

Oracle® Communications

Cloud Native Core, Security Edge Protection Proxy User Guide



Release 26.1.200
G59030-01
July 2026

ORACLE®

Copyright © 2020, 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Contents

1 Introduction

- 1.1 Overview
- 1.2 References

2 Security Edge Protection Proxy (SEPP) Architecture

- 2.1 N32c and N32f Call flows

3 SEPP Functionalities and Features

3.1 Core Functionalities

- 3.1.1 SEPP MNO functionality Support
- 3.1.2 NF Authentication using TLS certificate
- 3.1.3 Dedicated Gateways for IntraPLMN and InterPLMN Traffic
- 3.1.4 Support for Core Networks with or without SCP/ Routing Proxy
- 3.1.5 DNS Support for Remote SEPP Resolution

3.2 SEPP Features

- 3.2.1 SEPP Deployment Using Flux CD
- 3.2.2 Hosted SEPP Feature
- 3.2.3 Support for Max-Response Timer Header
- 3.2.4 Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers
- 3.2.5 Error Response and Logging Enhancement for SEPP
- 3.2.6 Lifecycle Management (LCM) Based Automation
- 3.2.7 Traffic Prioritization using Egress Gateway Pod Protection
- 3.2.8 SEPP and NRF Communication
 - 3.2.8.1 Static Host Based Selection
 - 3.2.8.2 DNS SRV Based Selection
 - 3.2.8.3 NRF Interface Registration and Deregistration
 - 3.2.8.4 NRF Selection Mechanisms using nrf client
 - 3.2.8.5 Upgrading SEPP for NRF Configuration
 - 3.2.8.6 Verifying NRF Routes Post Installation
- 3.2.9 Integrating SEPP with 5G Network Intelligence Fabric (5G NIF)
- 3.2.10 TLS 1.3 Support for Kubernetes API
- 3.2.11 LCI and OCI Headers in 5G Architecture

3.2.12	SEPP Dashboard Support for Detecting Vulnerable Messages	10
3.2.13	Security Countermeasure Features	10
3.2.13.1	Security Countermeasure Nonverbose Error Responses	10
3.2.13.2	Stateless Countermeasure Features	10
3.2.13.3	Stateful Countermeasure Features	12
3.2.14	Support for Dual Stack	13
3.2.15	Multiple SEPP Instances on Shared cnDBTier Cluster	14
3.2.16	Proactive Status Updates on SEPP	14
3.2.17	Traffic Segregation	15
3.2.18	Support for Originating Network Id Header Validation, Insertion, and Transposition	16
3.2.19	Support for Common Service APIs in CNC Console	16
3.2.20	Support for TLS	17
3.2.21	Support for Automated Certificate Lifecycle Management	17
3.2.22	Load Sharing among Multiple Remote SEPP Nodes	17
3.2.23	Separate Port Configurations for N32c and N32f on the Egress Routes	17
3.2.24	Support for cnDBTier APIs in CNC Console	18
3.2.25	5G SBI Message Mediation Support	18
3.2.26	Topology Hiding	19
3.2.27	Global Rate Limiting on Egress Gateway of SEPP	20
3.2.28	Global Rate Limiting on Ingress Gateway of SEPP	21
3.2.29	Multiple PLMN Support for Local and Remote SEPP	21
3.2.30	Support of TLS for Roaming Hubs and IPX Providers	22
3.2.31	Overload Control	22
3.2.32	Alternate Routing Across Remote SEPPs	23
3.2.33	Server Header Support	23
3.2.34	Support for Networks with/without 3GPP-SBI-Target-API-Root Header	23
3.2.35	Subject Alternate Name(SAN) Validation	24
3.2.36	SEPP Message Feed	24
3.2.37	Steering of Roaming (SOR) feature	25
3.2.38	Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set	25
3.2.39	Georedundancy	26
3.2.39.1	Two-Site Georedundancy	26
3.2.39.2	Three-Site Georedundancy	26
3.2.39.3	Four-Site Georedundancy	26
3.2.39.4	Managing the Feature	26
3.2.39.5	Adding a Site to an Existing SEPP Georedundant Site	27
3.2.39.6	Removing a Site from an Existing Georedundant Deployment	27
3.2.39.7	Recovering a Failed Site	27
3.2.40	Support for Kubernetes Resource	27
3.2.40.1	Support for Network Policies	27
3.2.40.2	Pod Disruption Budget	27

3.2.41	Alternate Routing and Load Sharing based on the DNS SRV Record for Home Network Functions	27
3.2.42	Rate Limiting for Egress Roaming Signaling per PLMN	28
3.3	Deployment Support	28
3.3.1	Helm Validation Framework	28
3.3.2	Automated Testing Suite Support	29

4 Configuring SEPP using CNC Console

4.1	CNC Console Interface	
4.2	SEPP Configuration	
4.2.1	Handshake Status	
4.2.2	Logging Config	
4.2.3	Remote SEPP	
4.2.4	Remote SEPP Set	1
4.2.5	Service APIs	1
4.2.6	System Options	1
4.2.7	NIF	1
4.2.8	Topology Hiding	1
4.2.9	Security Countermeasure	2
4.2.9.1	Nonverbose Error Response	2
4.2.9.2	Cat 1 -Service API Validation	2
4.2.9.3	Cat 2 – Network ID Validation	2
4.2.9.4	Cat 0 - SBI Message Schema Validation	3
4.2.9.5	Cat 3 - Previous Location Check	3
4.2.9.6	Cat-3 Time Location Check for Roaming Subscribers	4
4.2.9.7	Cat 1 - Service API Query Param Validation List Screen	4
4.2.10	Mediation	4
4.2.11	Ingress Rate Limiting	5
4.2.12	Egress Rate Limiting	5
4.2.13	SOR	5
4.2.14	Originating Network ID Header Support	5
4.2.15	Configurations	6
4.2.15.1	Gateway	6
4.2.15.2	cnDBTier	9
4.2.15.3	NRF-Client	10

5 SEPP Metrics, KPIs, and Alerts

5.1	SEPP Metrics
5.1.1	Configuring SEPP Metrics Dashboard in OCI
5.1.2	Common Metrics

- 5.1.2.1 cgroup_cpu_nanoseconds
- 5.1.2.2 cgroup_memory_bytes
- 5.1.2.3 nrfclient_nf_status_with_nrf
- 5.1.2.4 oc_ingressgateway_ip_addresses_fetch_failure
- 5.1.2.5 oc_egressgateway_ip_addresses_fetch_failure
- 5.1.2.6 oc_certificatemanagement_tls_certificate_info
- 5.1.2.7 oc_certificatemanagement_tls_secret_status
- 5.1.3 CN32F Common Metrics
 - 5.1.3.1 ocsepp_cn32f_jetty_request_stat_metrics_total
 - 5.1.3.2 ocsepp_cn32f_jetty_response_stat_metrics_total
 - 5.1.3.3 ocsepp_cn32f_connection_failure_total
 - 5.1.3.4 ocsepp_cn32f_requests_failure_total
 - 5.1.3.5 ocsepp_cn32f_response_failure_total
 - 5.1.3.6 ocsepp_cn32f_requests_total
 - 5.1.3.7 ocsepp_cn32f_response_total
 - 5.1.3.8 ocsepp_cn32f_latency_seconds_count
 - 5.1.3.9 ocsepp_cn32f_latency_seconds_max
 - 5.1.3.10 ocsepp_cn32f_latency_seconds_sum
 - 5.1.3.11 ocsepp_cn32f_outgoing_connections
 - 5.1.3.12 ocsepp_cn32f_server_latency
 - 5.1.3.13 ocsepp_cn32f_multipart_response_extraction_failure_total
 - 5.1.3.14 csepp_cn32f_multipart_requests_extraction_failure_total
- 5.1.4 PN32F Common Metrics
 - 5.1.4.1 ocsepp_pn32f_requests_total
 - 5.1.4.2 ocsepp_pn32f_requests_failure_total
 - 5.1.4.3 ocsepp_pn32f_responses_total
 - 5.1.4.4 ocsepp_pn32f_responses_failure_total
 - 5.1.4.5 ocsepp_pn32f_latency_seconds_count
 - 5.1.4.6 ocsepp_pn32f_latency_seconds_sum
 - 5.1.4.7 ocsepp_pn32f_latency_seconds_max
 - 5.1.4.8 ocsepp_pn32f_connection_failure_total
 - 5.1.4.9 ocsepp_pn32f_jetty_request_stat_metrics_total
 - 5.1.4.10 ocsepp_pn32f_jetty_response_stat_metrics_total
 - 5.1.4.11 ocsepp_pn32f_outgoing_connections
 - 5.1.4.12 ocsepp_pn32f_server_latency
- 5.1.5 N32C Handshake Procedure Metrics
 - 5.1.5.1 ocsepp_n32c_handshake_failure_attempts_total
 - 5.1.5.2 ocsepp_n32c_handshake_reInitiation_failure
 - 5.1.5.3 ocsepp_cn32c_handshake_requests_total
 - 5.1.5.4 ocsepp_cn32c_handshake_response_total
 - 5.1.5.5 ocsepp_cn32c_handshake_initiation_req_total
 - 5.1.5.6 ocsepp_cn32c_handshake_reinitiation_req_total

5.1.5.7	ocsepp_cn32c_handshake_delete_req_total	2
5.1.5.8	ocsepp_pn32c_handshake_requests_total	2
5.1.5.9	ocsepp_pn32c_handshake_response_total	2
5.1.5.10	ocsepp_n32c_handshake_status	2
5.1.6	Cat-1 NRF Service API Query Parameters Validation metrics	2
5.1.6.1	ocsepp_security_service_api_query_param_validation_success_total	2
5.1.6.2	ocsepp_security_service_api_query_param_validation_failure_total	2
5.1.6.3	ocsepp_configmgr_query_param_api_validation_config_failure_total	2
5.1.7	5G SBI Message Mediation Support Metrics	3
5.1.7.1	ocsepp_n32f_mediation_requests_total	3
5.1.7.2	ocsepp_n32f_mediation_not_applied_total	3
5.1.7.3	ocsepp_cn32f_mediation_response_total	3
5.1.7.4	ocsepp_cn32f_mediation_response_failure_total	3
5.1.7.5	ocsepp_pn32f_mediation_response_total	3
5.1.7.6	ocsepp_pn32f_mediation_response_failure_total	3
5.1.7.7	ocsepp_cn32f_mediation_latency_seconds_count	3
5.1.7.8	ocsepp_cn32f_mediation_latency_seconds_max	3
5.1.7.9	ocsepp_cn32f_mediation_latency_seconds_sum	3
5.1.7.10	ocsepp_pn32f_mediation_latency_seconds_count	3
5.1.7.11	ocsepp_pn32f_mediation_latency_seconds_max	3
5.1.7.12	ocsepp_pn32f_mediation_latency_seconds_sum	3
5.1.7.13	ocsepp_configmgr_mediation_rule_validation_failure_total	3
5.1.7.14	ocsepp_configmgr_mediation_feature_validation_failure_total	3
5.1.7.15	ocsepp_configmgr_mediation_rule_config_failure_total	3
5.1.8	Hosted SEPP Metrics	3
5.1.8.1	ocsepp_cn32c_handshake_requests_hosted_sepp_total	3
5.1.8.2	ocsepp_cn32f_sni_xfs_requests_hosted_sepp_total	3
5.1.8.3	ocsepp_cn32f_sni_xfs_requests_mismatch_hosted_sepp_total	3
5.1.8.4	ocsepp_pn32f_requests_hosted_sepp_total	4
5.1.9	Message Copy Metrics	4
5.1.9.1	oc_ingressgateway_msgcopy_requests_total	4
5.1.9.2	oc_ingressgateway_msgcopy_responses_total	4
5.1.9.3	oc_ingressgateway_dd_unreachable	4
5.1.9.4	oc_egressgateway_msgcopy_requests_total	4
5.1.9.5	oc_egressgateway_msgcopy_responses_total	4
5.1.9.6	oc_egressgateway_dd_unreachable	4
5.1.10	SOR Metrics	4
5.1.10.1	ocsepp_pn32f_sor_requests_total	4
5.1.10.2	ocsepp_pn32f_sor_responses_total	4
5.1.10.3	ocsepp_pn32f_sor_retry_to_producer_requests_total	4
5.1.10.4	ocsepp_pn32f_sor_back_to_consumer_responses_total	4
5.1.10.5	ocsepp_pn32f_sor_failure_total	4

5.1.10.6	ocsepp_pn32f_sor_timeout_failure_total	4
5.1.11	Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Metrics	4
5.1.11.1	oc_ingressgateway_rss_ratelimit_total	4
5.1.11.2	oc_ingressgateway_rss_ratelimit_total	4
5.1.11.3	ocsepp_configmgr_routefailure_total	4
5.1.11.4	oc_ingressgateway_rss_ratelimit_total	5
5.1.11.5	ocsepp_configmgr_ingress_rate_limit_validation_failure_total	5
5.1.12	Topology Hiding Metrics	5
5.1.12.1	ocsepp_topology_latency_seconds_count	5
5.1.12.2	ocsepp_topology_latency_seconds_max	5
5.1.12.3	ocsepp_topology_latency_seconds_sum	5
5.1.12.4	ocsepp_topology_header_success_total	5
5.1.12.5	ocsepp_topology_header_failure_total	5
5.1.12.6	ocsepp_topology_body_success_total	5
5.1.12.7	ocsepp_topology_body_failure_total	5
5.1.12.8	ocsepp_topology_success_total	5
5.1.12.9	ocsepp_topology_invalid_header_regex_configured_total	5
5.1.12.10	ocsepp_topology_header_regex_not_configured_total	5
5.1.13	Cat 0 - SBI Message Schema Validation Metrics	5
5.1.13.1	ocsepp_message_validation_applied_total	5
5.1.13.2	ocsepp_message_validation_on_body_failure_total	5
5.1.13.3	ocsepp_message_validation_on_header_failure_total	5
5.1.13.4	ocsepp_configmgr_message_validation_config_failure_total	5
5.1.14	Cat 1 - Service API Validation Metrics	5
5.1.14.1	ocsepp_security_service_api_failure_total	5
5.1.15	Cat 2 - Network ID Validation Metrics	5
5.1.15.1	ocsepp_originating_network_id_validation_success_total	5
5.1.15.2	ocsepp_network_id_validation_body_failure_total	6
5.1.15.3	ocsepp_network_id_validation_header_failure_total	6
5.1.15.4	ocsepp_configmgr_network_id_header_validation_failure_total	6
5.1.15.5	ocsepp_configmgr_network_id_rule_config_failure_total	6
5.1.15.6	ocsepp_configmgr_network_id_body_config_failure_total	6
5.1.15.7	ocsepp_configmgr_network_id_rule_validation_failure_total	6
5.1.15.8	ocsepp_configmgr_network_id_header_config_failure_total	6
5.1.16	Cat 3 - Previous Location Check Metrics	6
5.1.16.1	ocsepp_previous_location_exception_failure_total	6
5.1.16.2	ocsepp_previous_location_validation_success_total	6
5.1.16.3	ocsepp_previous_location_validation_failure_total	6
5.1.16.4	ocsepp_previous_location_validation_requests_total	6
5.1.16.5	ocsepp_pn32f_notification_fromNrf_total	6
5.1.17	Cat-3 Time Check for Roaming Subscribers	6
5.1.17.1	ocsepp_time_unauthenticated_location_validation_requests_total	6

5.1.17.2	ocsepp_time_unauthenticated_location_validation_success_total	6
5.1.17.3	ocsepp_time_unauthenticated_location_validation_failure_total	6
5.1.17.4	ocsepp_time_unauthenticated_location_exception_failure_total	6
5.1.17.5	ocsepp_time_unauthenticated_location_blocklist_requests_total	6
5.1.17.6	ocsepp_configmgr_country_config_failure_total	6
5.1.17.7	ocsepp_configmgr_exception_mcc_config_failure_total	6
5.1.17.8	ocsepp_configmgr_time_unauthenticated_config_failure_total	7
5.1.18	Rate Limiting for Egress Roaming Signaling per PLMN Metrics	7
5.1.18.1	oc_ingressgateway_plmn_egress_ratelimit_total	7
5.1.18.2	oc_ingressgateway_plmn_egress_ratelimit_total	7
5.1.18.3	oc_ingressgateway_plmn_egress_ratelimit_total	7
5.1.18.4	oc_ingressgateway_plmn_egress_ratelimit_total	7
5.1.18.5	oc_ingressgateway_plmn_egress_ratelimit_total	7
5.1.18.6	oc_ingressgateway_plmn_egress_ratelimit_total	7
5.1.19	Config Manager Metrics	7
5.1.19.1	ocsepp_configmgr_rpp_config_failure_total	7
5.1.19.2	ocsepp_configmgr_rpp_validation_failure_total	7
5.1.19.3	ocsepp_configmgr_routeupdate_total	7
5.1.19.4	ocsepp_configmgr_n32f_context_config_failure_total	7
5.1.19.5	ocsepp_configmgr_delete_route_failure_total	7
5.1.19.6	ocsepp_configmgr_put_route_failure_total	7
5.1.19.7	ocsepp_configmgr_delete_peer_set_failure_total	7
5.1.20	Support for Originating Network Id Header Validation, Insertion, and Transposition Metrics	7
5.1.20.1	ocsepp_originating_network_id_header_added_total	7
5.1.20.2	ocsepp_originating_network_id_header_transposed	7
5.1.20.3	ocsepp_originating_network_id_header_addition_failed	7
5.1.20.4	ocsepp_originating_network_id_header_add_or_transpose_failed	7
5.1.21	Support for TLS 1.3 Metrics	7
5.1.21.1	oc_ingressgateway_incoming_tls_connections	7
5.1.21.2	oc_egressgateway_outgoing_tls_connections	7
5.1.21.3	security_cert_x509_expiration_seconds	7
5.1.22	Proactive Status Updates on SEPP Metrics	7
5.1.22.1	oc_egressgateway_peer_health_status	7
5.1.22.2	oc_egressgateway_peer_health_ping_request_total	8
5.1.22.3	oc_egressgateway_peer_health_ping_response_total	8
5.1.22.4	oc_egressgateway_peer_health_status_transitions_total	8
5.1.22.5	oc_ingressgateway_health_check_incoming_ping_total	8
5.1.23	Multiple SEPP instances on shared cnDBTier cluster Metrics	8
5.1.23.1	ocsepp_cn32f_database_connectivity_healthy	8
5.1.23.2	ocsepp_pn32f_database_connectivity_healthy	8
5.1.23.3	ocsepp_cn32c_database_connectivity_healthy	8
5.1.23.4	ocsepp_pn32c_database_connectivity_healthy	8

5.1.23.5	ocsepp_configmgr_database_connectivity_healthy	8
5.1.23.6	ocsepp_coherence_database_connectivity_healthy	8
5.1.24	LCI and OCI Headers Metrics	8
5.1.24.1	oc_ingressgateway_headers_lci_total	8
5.1.24.2	oc_ingressgateway_headers_oci_total	8
5.1.25	Integrating SEPP with 5G Network Intelligence Fabric (5G NIF) Metrics	8
5.1.25.1	ocsepp_nif_discovery_requests_total	8
5.1.25.2	ocsepp_nif_discovery_responses_total	8
5.1.25.3	ocsepp_nif_registration_status	8
5.1.25.4	ocsepp_pn32f_nif_error_copy_requests_total	8
5.1.25.5	ocsepp_pn32f_nif_error_copy_responses_total	8
5.1.26	NRF Selection Mechanisms Using nrf client Metrics	8
5.1.26.1	nrfclient_nrf_operative_status	8
5.1.26.2	nrfclient_dns_lookup_request_total	8
5.1.27	Overload Control Metrics	8
5.1.27.1	service_resource_overload_level	8
5.1.27.2	oc_ingressgateway_route_overloadcontrol_discard_total	8
5.1.27.3	ingressgateway_svc_pending_count	8
5.1.28	Traffic Prioritization using Egress Gateway Pod Protection Metrics	8
5.1.28.1	nrfclient_nw_conn_out_request_total	8
5.1.28.2	oc_egressgateway_http_request_ratelimit_values_total	8
5.1.28.3	oc_egressgateway_http_request_ratelimit_reject_chain_length_histogram_bucket	8
5.1.28.4	oc_egressgateway_congestion_cpu_state	9
5.1.28.5	oc_egressgateway_congestion_system_state	9
5.1.28.6	oc_egressgateway_system_state_duration_percentage	9
5.1.28.7	oc_egressgateway_congestion_level_total	9
5.1.28.8	oc_egressgateway_congestion_level_bucket_total	9
5.1.28.9	oc_egressgateway_congestion_cpu_percentage_bucket	9
5.1.29	oc_egressgateway_http_request_ratelimit_reject_chain_duration_histogram_seconds_bucket	9
5.1.30	Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers Metrics	9
5.1.30.1	ocsepp_oauth_total_requests	9
5.1.31	Alternate Route Metrics	9
5.1.31.1	oc_dns_srv_lookup_total	9
5.1.31.2	oc_fqdn_alternate_route_total	9
5.1.31.3	oc_alternate_route_resultset	9
5.1.31.4	oc_configclient_request_total	9
5.1.31.5	oc_configclient_response_total	9
5.1.31.6	oc_configserver_reachability	9
5.1.32	Ingress and Egress Gateway Metrics	9
5.1.32.1	Ingress Gateway Metrics	9
5.1.32.2	Egress Gateway Metrics	10
5.2	SEPP KPIs	10

5.2.1	N32C Handshake Procedure KPIs	10
5.2.1.1	cn32c Handshake Requests Per Remote SEPP	10
5.2.1.2	cn32c Handshake Success Rate	10
5.2.1.3	cn32c Handshake Response Per Remote SEPP	11
5.2.1.4	cn32c Handshake Failure Per Remote SEPP	11
5.2.1.5	pn32c Handshake Requests Total Per Remote SEPP	11
5.2.1.6	pn32c Handshake Response Total Per Remote SEPP	11
5.2.1.7	pn32c Handshake Success rate	11
5.2.1.8	pn32c Handshake Failure Per Remote SEPP	11
5.2.2	SEPP Common KPIs	11
5.2.2.1	Memory Usage per POD	11
5.2.2.2	CPU Usage per POD	11
5.2.2.3	Total Ingress gateway requests	11
5.2.2.4	Total Egress gateway requests	11
5.2.2.5	Total Ingress gateway responses	11
5.2.2.6	Total Egress gateway responses	11
5.2.2.7	IGW Processing Time (ms)	11
5.2.2.8	PercentageDiscard	11
5.2.2.9	PriorityDiscard	11
5.2.3	CN32F Common KPIs	11
5.2.3.1	cn32f Routing Success Rate	11
5.2.3.2	Total cn32f Requests	11
5.2.3.3	cn32f Processing Time (ms)	11
5.2.3.4	Total cn32f Responses	11
5.2.3.5	cn32f Failures	11
5.2.4	PN32F Common KPIs	11
5.2.4.1	Total pn32f Requests	11
5.2.4.2	Total pn32f Responses	11
5.2.4.3	pn32f Processing Time (ms)	11
5.2.4.4	pn32f Failures	11
5.2.4.5	pn32f Routing Success Rate	11
5.2.5	Global Rate Limiting Feature KPIs	11
5.2.5.1	PLMN IGW Global Rate limit Traffic Rejected	11
5.2.5.2	N32 IGW Global Rate limit Traffic Rejected	11
5.2.6	Topology Hiding KPIs	11
5.2.6.1	CN32F Topology Egress Request Processing Time(ms)	11
5.2.6.2	CN32F Topology Ingress Response Processing Time	11
5.2.6.3	CN32F Topology Hiding Success	11
5.2.6.4	N32F Topology Success by headers	11
5.2.6.5	CN32F Topology Hiding Missing Regex Configuration	11
5.2.6.6	CN32F Topology Hiding Invalid Header Regex	11
5.2.6.7	PN32F Topology Ingress Request Processing Time(ms)	11

5.2.6.8	PN32F Topology Egress Response Processing Time(ms)	11
5.2.6.9	PN32F Topology Hiding Invalid Header Regex	11
5.2.6.10	PN32F Topology Hiding Missing Regex Configuration	11
5.2.6.11	PN32F Topology Hiding Success	12
5.2.7	5G SBI Message Mediation Support KPIs	12
5.2.7.1	Mediation Requests Counters - N32 Egress Request	12
5.2.7.2	Mediation Requests Counters - N32 Ingress Response	12
5.2.7.3	Mediation Requests Counters - N32 Ingress Request	12
5.2.7.4	Mediation Requests Counters - N32 Egress Response	12
5.2.7.5	Mediation Response Counters - N32 Egress Request	12
5.2.7.6	Mediation Response Counters - N32 Ingress Response	12
5.2.7.7	Mediation Response Counters - N32 Ingress Request	12
5.2.7.8	Mediation Response Counters - N32 Egress Response	12
5.2.7.9	Mediation Response Failure	12
5.2.7.10	Mediation Applied Total	12
5.2.7.11	Mediation Response Time At PN32F	12
5.2.7.12	Mediation Response Time At CN32F	12
5.2.8	Ingress Gateway Message Copy KPIs	12
5.2.8.1	Total Requests Data sent towards DD for Ingress Gateway	12
5.2.8.2	Total Ack received from DD for Requests for Ingress Gateway	12
5.2.9	Egress Gateway Message Copy KPIs	12
5.2.9.1	Total Requests Data sent towards DD for Egress Gateway	12
5.2.9.2	Total Ack received from DD for Requests for Egress Gateway	12
5.2.10	Hosted SEPP KPIs	12
5.2.10.1	CN32F Allowed P-RSS Validation Failure Count	12
5.2.10.2	PN32F Allowed P-RSS Validation Failure Count	12
5.2.11	SoR KPIs	12
5.2.11.1	Pn32f to SoR Request count total	12
5.2.11.2	SoR to Pn32f Response count total	12
5.2.12	Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set KPIs	12
5.2.12.1	Average No of messages discarded for a particular RSS	12
5.2.12.2	Average No of messages accepted for a particular RSS	12
5.2.12.3	Average No of messages for which feature not applied	12
5.2.12.4	Average of all messages by Status	12
5.2.12.5	List of Average number of messages dropped for all RSS	12
5.2.12.6	List of Average number of messages accepted for all RSS	12
5.2.13	Cat 0 - SBI Message Schema Validation KPIs	12
5.2.13.1	Message validation applied requests on cn32f	12
5.2.13.2	Cn32f message validation failure on request body	12
5.2.13.3	Cn32f message validation failures on request query parameter(s)	12
5.2.13.4	Message validation applied requests on pn32f	12
5.2.13.5	Pn32f message validation failure on request body	12

5.2.13.6	Pn32f message validation failures on request query parameter(s)	12
5.2.14	Rate Limiting for Egress Roaming Signaling per PLMN KPIs	12
5.2.14.1	Average Number of Messages Rejected for a Particular PLMN	12
5.2.14.2	Average Number of Messages Accepted for a Particular PLMN	12
5.2.14.3	Average Number of Messages for which Feature not Applied	12
5.2.14.4	Average of all Messages by Status	12
5.2.14.5	Average Number of Messages Rejected per PLMN	12
5.2.14.6	Average Number of Messages Accepted per PLMN	13
5.2.15	Integrating SEPP with 5G Network Intelligence Fabric (5G NIF) Feature KPIs	13
5.2.15.1	Discovery Requests Sent Towards NRF for NIF	13
5.2.15.2	Response Received from NRF for NIF Discovery	13
5.2.15.3	Rejected Message Copied towards NIF	13
5.2.15.4	Responses Received from NIF for Copied Messages	13
5.2.16	Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers KPIs	13
5.2.16.1	Total number of Get_ACCESS_TOKEN messages received from a particular SEPP	13
5.2.16.2	Number of requests without the requesterPLMN/requesterPLMNList attribute received from a particular SEPP	13
5.2.16.3	Number of requests to which SEPP responded with an error towards a particular SEPP	13
5.3	SEPP Alerts	13
5.3.1	System Level Alerts	13
5.3.1.1	SEPPPodMemoryUsageAlert	13
5.3.1.2	SEPPPodCpuUsageAlert	13
5.3.1.3	ocseppPodsRestart	13
5.3.1.4	ocseppAppinfoServiceDown	13
5.3.1.5	ocseppCn32cServiceDown	13
5.3.1.6	ocseppCn32fServiceDown	14
5.3.1.7	ocseppConfigMgrServiceDown	14
5.3.1.8	ocseppN32EgwServiceDown	14
5.3.1.9	ocseppN32lgwServiceDown	14
5.3.1.10	ocseppConfigserverServiceDown	14
5.3.1.11	ocseppPerfinfoServiceDown	14
5.3.1.12	ocseppPlmnEgwServiceDown	14
5.3.1.13	ocseppPlmnlgwServiceDown	15
5.3.1.14	ocseppPn32cServiceDown	15
5.3.1.15	ocseppPn32fServiceDown	15
5.3.1.16	ocseppNrfdiscServiceDown	15
5.3.1.17	ocseppNrfggmServiceDown	15
5.3.1.18	ocseppAlternrouteServiceDown	15
5.3.1.19	ocseppMediationServiceDown	16
5.3.1.20	ocseppCoherenceServiceDown	16
5.3.1.21	ocseppNfStatusUnavailable	16
5.3.2	Application Level Alerts	16

5.3.2.1	Common Alerts	16
5.3.2.2	Handshake Alerts	17
5.3.2.3	Upgrade Alerts	17
5.3.2.4	Rollback Alerts	17
5.3.2.5	Global Rate Limiting on Ingress Gateway of SEPP Alerts	17
5.3.2.6	Topology Hiding Alerts	18
5.3.2.7	5G SBI Message Mediation Support Alerts	18
5.3.2.8	Overload Control Alerts	18
5.3.2.9	Hosted SEPP Alerts	19
5.3.2.10	SEPP Message Feed Alerts	19
5.3.2.11	Steering of Roaming (SOR) Alerts	19
5.3.2.12	Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Alerts	20
5.3.2.13	Cat-0 SBI Message Schema Validation Alerts	21
5.3.2.14	Cat-1 Service API Validation Alerts	22
5.3.2.15	Cat-1 NRF Service API Query Parameters Validation Alerts	22
5.3.2.16	Cat-2 Network ID Validation Alerts	23
5.3.2.17	Cat-3 Previous Location Check Alerts	23
5.3.2.18	Cat-3 Time Check for Roaming Subscribers	24
5.3.2.19	Rate Limiting for Egress Roaming Signaling per PLMN Alerts	25
5.3.2.20	Separate Port Configurations for N32c and N32f on the Egress Routes Alerts	26
5.3.2.21	Support for TLS 1.3	26
5.3.2.22	Multiple SEPP Instances on Shared cnDBTier Cluster Alerts	26
5.3.2.23	Proactive Status Updates on SEPP Alerts	28
5.3.2.24	Integrating SEPP with 5G Network Intelligence Fabric (5G NIF) Feature Alerts	28
5.3.2.25	Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers Alerts	28
5.3.2.26	Support for Max-Response Timer Header Alerts	28
5.3.2.27	Egress Gateway Alerts	29
5.4	Alert Configuration	29

6 Protocol and Application Errors

7 SEPP Response Codes

Preface

- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [Conventions](#)

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

My Oracle Support

My Oracle Support (<https://support.oracle.com>) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <http://www.oracle.com/us/support/contact/index.html>. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request.
2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
 - For Technical issues such as creating a new Service Request (SR), select **1**.
 - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

Acronyms

The following table lists the acronyms and the terminologies used in the document:

Table Acronyms and Terminologies

Acronym	Description
Cat-0	Category 0 SBI Message Schema Validation Feature
Cat-1	Category 1 Service API Validation Feature
Cat-2	Category 2 Network ID Validation Feature
Cat-3	Category 3 Previous Location Check/ Category 3 Time location check
CNC Console	Oracle Communications Cloud Native Configuration Console
CRD	Custom Resource Definition
CNE	Oracle Communications Cloud Native Environment
cSEPP/C-SEPP	Consumer Security Edge Protection Proxy
DB	Database
DNS	Domain Name System
DRL	Drools Rule Language
EGW	Egress Gateway
FQDN	Fully Qualified Domain Name
GSMA	Groupe Speciale Mobile Association (GSMA). Represents the interests of mobile operators and the broader mobile industry worldwide.
Hosted SEPP	GSMA deployment model in which partners are hosted to connect to roaming partners.
IGW	Ingress Gateway
IPX	Internetwork Packet Exchange
K8s	Kubernetes
LCI Header	Load Control Information header
Local PLMN	PLMN managed by Local SEPP
Local SEPP	SEPP in Local PLMN
MNC	Mobile Network Codes
MCC	Mobile Country Codes
MNO	Mobile Network Operator
NDB	Network Database
NF	Network Function
Network Function	A functional building block within a network infrastructure, which has well defined external interfaces and well defined functional behavior. In practical terms, a network function is often a network node or physical appliance.
NF Consumer	A generic way to refer to an NF which consumes services provided by another NF. Example: An AMF acts as a Consumer NF that consumes AMPolicy services provided by the PCF.
NF Instance	A specific instance of a network function type.
NF Producer or NF Provider	A generic way to refer to an NF which provides services that can be consumed by another NF. Example: A PCF acts as a Producer NF that provides AMPolicy Services to the AMF.
NIF	Network Intelligence Fabric

Table (Cont.) Acronyms and Terminologies

Acronym	Description
NRF	Oracle Communications Cloud Native Core, Network Repository Function
OCI Header	Overload Control Information header
OSDC	Oracle Software Delivery Cloud
OCI	Oracle Cloud Infrastructure
OCIR	Oracle Cloud Infrastructure Registry
OHC	Oracle Help Center
OKE	Oracle Engine for Kubernetes
OTEL	OpenTelemetry
OSDC	Oracle Software Delivery Cloud
PDB	PodDisruptionBudget
PLMN	Public Land Mobile Network
pSEPP/P-SEPP	Producer Security Edge Protection Proxy
Remote PLMN	PLMN managed by Remote SEPP
Remote SEPP	SEPP in Remote PLMN
Remote SEPP Set	Set of Remote SEPPs to allow routing across Remote SEPPs
REST API	Representational State Transfer Application Programming Interface
Roaming Hub	Roaming Hub is the deployment mode of SEPP. Roaming Hub is used as an intermediate proxy. Each SEPP connects to the Roaming Hub which further connect to another SEPP. All the Remote SEPPs can communicate with each other through roaming hub.
SAN	Subject Alternate Name
SNI	Server Name Indication
Scaling	Ability to dynamically extend or reduce resources granted to the Virtual Network Function (VNF) as needed. This includes scaling out and in or scaling up and down.
SCM	Security Countermeasure
SEPP	Oracle Communications Cloud Native Core, Security Edge Protection Proxy
SUCI	Subscription Concealed Identifier
SUPI	Subscription Permanent Identifier
SVC	Service
TLS	Transport Layer Security
TH	Topology Hiding
TUH	Topology Unhiding
TPS	Transactions Per Second
UE	User Equipment
UDR	Oracle Communications Cloud Native Core, Unified Data Repository

What's New In This Guide

This section introduces the documentation updates for Release 26.1.2xx.

Release 26.1.200- G55168-01, July 2026

Feature Updates:

- Added a table with the details about configuration type and status of each feature in the [SEPP Functionalities and Features](#) section.
- Added the [SEPP Deployment Using Flux CD](#) feature.
- Updated the [Traffic Segregation](#) feature section with steps to onboard a new Remote SEPP.
- Updated the [SEPP Traffic Prioritization using Egress Gateway Pod Protection](#) feature section with the following:
 - Added the NRF Client REST API Configurations
 - Added the [NRF Client CNC Console Configurations](#)
- Added the [MultiFQDN CNC Console Configurations](#) for Hosted SEPP feature.

General Updates:

- Added the Gracefully Stop And Start Georeplication (Stop Georeplication and Start Georeplication) APIs and IP Preference For Replication Channel API in the following sections:
 - [Support for cnDBTier APIs in CNC Console](#)
 - [cnDBTier APIs](#)

1

Introduction

This document provides information about the role of Oracle Communications Cloud Native Core, Security Edge Protection Proxy (SEPP) in 5G Service Based Architecture and how to configure and use SEPP services and managed objects.

1.1 Overview

Oracle Communications Security Edge Protection Proxy (SEPP) is a Cloud native solution based on microservice architecture, which acts as a non-transparent proxy sitting at the perimeter of the PLMN network enabling secured inter-NF communication across PLMN networks.

Note

The performance and capacity of the SEPP system may vary based on the call model, Feature/Interface configuration, and underlying platform and hardware environment.

The SEPP installation is supported over the following platforms:

- – Oracle Communications Cloud Native Core, Cloud Native Environment (CNE)
- – Oracle Communications Cloud Native Core OCI Adaptor, NF Deployment in OCI

For more information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

SEPP supports the following functionalities:

- Protects the application layer control plane messages and sensitive data between two NFs belonging to different PLMNs that use the N32 interface to communicate with each other. The N32 interface is used between the SEPPs of a Visitor PLMN (VPLMN) and a Home PLMN (HPLMN) in roaming scenarios. 3GPP has specified N32 to be considered as two separate interfaces: N32-c and N32-f.
 - N32-c is the Control Plane interface between the SEPPs for performing the initial handshake and negotiating the parameters to be applied for the actual N32 message forwarding.
 - N32-f is the Forwarding interface between the SEPPs, that is used for forwarding the communication between the Network Function (NF) service consumer and the NF service producer after applying the application level security protection.
- Provides secure communication of Inter PLMN messages from Consumer NF to Producer NF using TLS protection mode (HTTP over TLS)
- Supports configuration of Remote SEPPs using REST API
- Performs mutual authentication and negotiation of cipher suites with the SEPP in the Remote SEPP.

- Handles key management aspects that involve setting up the required cryptographic keys needed for securing messages on the N32 interface between two SEPPs
- Provides a single point of access and control to internal NFs
- Validates inbound traffic as to whether it is from an authorized external PLMN
- Supports cross-layer validation of source and destination addresses and identifiers to provide anti-spoofing capabilities

SEPP Availability

Oracle Communications Cloud Native Core Security Edge Protection Proxy (SEPP) availability is dependent on many factors. SEPP applications are designed to achieve 99.999% availability, according to the applicable Telecommunications Industry Association TL9000 standards, with the following deployment requirements:

- Deploy on a Cloud Native Environment with at least 99.999% Availability.
- Deploy with $n + k$ application redundancy, where k is greater than or equal to one.
- Maintain production software within $n-3$ software releases, where n is the current general availability release.
- Apply bug fixes, critical patches, and configuration recommendations provided by Oracle promptly.
- Maintain disaster recovery procedures external to the applications for the reconstruction of lost or altered files, data, programs, or Cloud Native environment.
- Install, configure, operate, and maintain SEPP as per Oracle's applicable installation, operation, administration, and maintenance specifications.
- Maintain an active support contract and provide access to the deployed SEPP and your personnel to assist Oracle in addressing any outage.

SEPP availability is measured for each calendar year and is calculated as follows:

Table 1-1 Measuring SEPP Availability

Availability	Description
Planned Product Availability	(Product available time in each month) less (Excluded Time (defined below) in each month).
Actual Product Availability	(Planned Product Availability) less (any Unscheduled Outage)
Product Availability Level	(Actual Product Availability across all Production instances divided by Planned Product Availability across all Production instances) x 100

Note

Excluded Time means:

- Scheduled maintenance time.
- Lack of power or backhaul connectivity, except to the extent that such lack of backhaul connectivity was caused directly by the CNC NF.
- Hardware failure.
- Issues arising out of configuration errors or omissions.
- Failures caused by third-party equipment or software not provided by Oracle.
- Occurrence of any event under Force Majeure.
- Any time associated with failure to maintain the recommended architecture and redundancy model requirements above.

1.2 References

Refer the following documents for more information about Security Edge Protection Proxy User Guide:

- *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, cnDBTier User Guide*
- *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Operations Services Overlay Installation and Upgrade Guide*
- *Oracle Communications Cloud Native Core, Data Collector User Guide*
- *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*
- *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*
- *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Network Impact Report*
- *Oracle Communications Cloud Native Configuration Console User Guide*
- *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Automated Test Suite User Guide*
- *Oracle Communications Cloud Native Core, Certificate Management Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Certificate Management User Guide*
- *Oracle Communications Cloud Native Core, OCI Deployment Guide*

- *Oracle Communications Cloud Native Core, OCI Adaptor User Guide*
- *Oracle Communications Network Analytics Data Director Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core Release Notes*
- *Oracle Communications Cloud Native Core Licensing Information User Guide*
- *Oracle Communications Cloud Native Core Solution Upgrade Guide*
- *Oracle Communications Cloud Native Core Security Guide*

2

Security Edge Protection Proxy (SEPP) Architecture

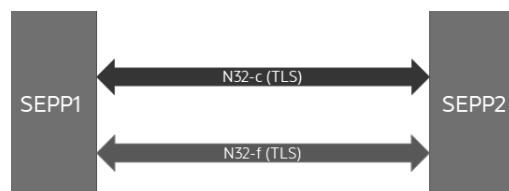
This section explains the Security Edge Protection Proxy (SEPP) Architecture.

Security Edge Protection Proxy (SEPP) is a proxy network function (NF) which is used for secured communication between inter-Public Land Mobile Network (PLMN) messages.

Security Edge Protection Proxy Interfaces

This section explains the Security Edge Protection Proxy (SEPP) interfaces.

Figure 2-1 Security Edge Protection Proxy Interfaces

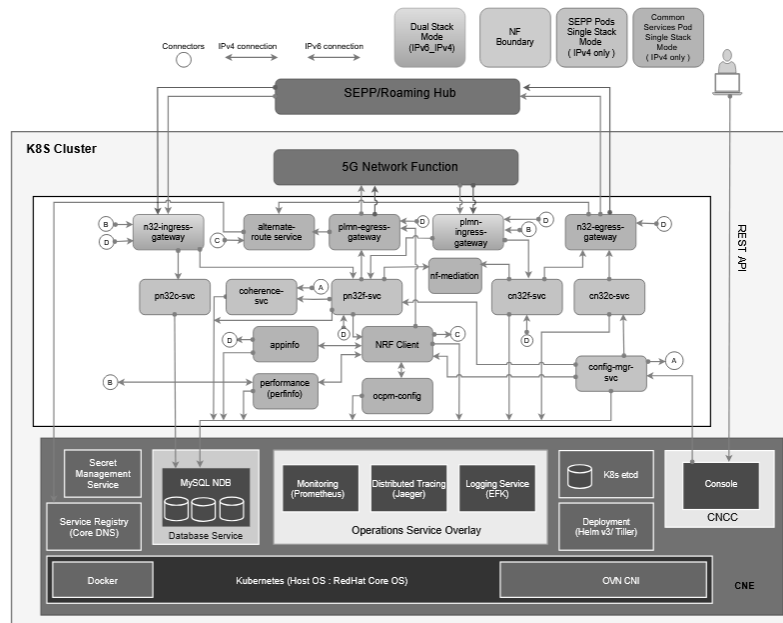


SEPP provides two interfaces, N32c and N32f:

- **N32c:** A control plane interface between the SEPPs for performing initial handshake and negotiating the parameters to be applied for the actual N32 message forwarding.
- **N32f:** A forwarding interface between the SEPPs for forwarding the communication between the NF service consumer and the NF service producer after applying application level security protection.

Security Edge Protection Proxy Architecture

Figure 2-2 Security Edge Protection Proxy Architecture



SEPP Architecture and Microservices Overview

This section explains the Security Edge Protection Proxy (SEPP) microservices:

Table 2-1 Microservices

Microservice	Description
config-mgr-svc	This microservice is an interface between CNC Console GUI and SEPP database which takes user configurations of Remote SEPP, Remote SEPP Set, and SEPP features and saves them in SEPP database. The microservice shares SEPP peer details to Gateway for creating routes. This microservice is used for configuring Remote SEPPs and retrieving handshake status. It also initiates a notification towards cn32c service to initiate handshake as soon as profile is enabled and notification to delete N32 context as soon as profile is disabled. Audit to re-initiate handshake notification in case of change in local profile is also handled by config-mgr.
cn32c-svc	This microservice performs context management and initiation of security capability negotiation. It creates an n32 context and maintains its state and negotiated security capability based on handshake procedure.
pn32c-svc	This microservice is used for context management and responds to security capability negotiation. It creates an n32 context and maintains its state and negotiated security capability based on handshake procedure.

Table 2-1 (Cont.) Microservices

Microservice	Description
cn32f-svc	This microservice receives messages from consumer NF and determines Remote SEPP address to forward the message over n32f interface to SEPP of producer NF network. Many SEPP features are implemented on this microservice. Example: Security Countermeasure, Topology hiding for header and body, Mediation etc.
pn32f-svc	This microservice receives inter-PLMN messages on N32 Interface from the Remote SEPP of consumer network over TLS connection and forwards to Producer NF. Example: Security Countermeasure, Topology hiding for header and body, Mediation etc.
plmn-ingress-gateway	Entry point for consumer NF to SEPP messages. PLMN Ingress Gateway exposes HTTP and HTTPS interfaces towards consumer NF network to receive messages required to be forwarded over inter-PLMN n32f interface.
plmn-egress-gateway	Exit point for SEPP to producer NF messages. PLMN Egress gateway exposes HTTP and HTTPS interface towards producer NF network to forward inter-PLMN messages received over inter-PLMN n32f interface.
n32-ingress-gateway	Entry point for N32 Ingress messages. N32 Ingress Gateway exposes HTTP over TLS interface on producer SEPP to receive n32c and n32f inter-PLMN messages.
n32-egress-gateway	Exit point for N32 Egress messages. N32 Egress gateway exposes HTTP over TLS interface on consumer SEPP to forward n32c and n32f inter-PLMN messages.
alternate-route-service	Exposes the Rest API to call DNS SRV requests to core-dns kubernetes service. Responsible for handling the static and dynamic retrieval of DNS records for defined virtual FQDN.
nrf-client-nfdiscovery or nrf-client-nfmanagement	NRF client application is responsible for performing NfRegistration, NfSubscription and NfDiscovery. The NRF Client interacts with the AppInfo and PerfInfo to check the status of the registered services as well as monitor the performance of the services.
Perf-info (Performance Monitor Service)	It is responsible for monitoring and analysis of the services to probe performance data, and provide analysis output including load, capacity. nrf client gets the performance data periodically from this service using the REST APIs exposed.
ocpm-config (config-server)	It is responsible to retrieve and store data by topic for various applications. It is the interface between nrf client and the Database. nrfclient does not directly use and interact with database, it uses the APIs of this service. There is a DB defined for nrfclient to save its state data right. you must be configuring the same
appinfo (Application Info Service)	This microservice acts as a helper service that monitors the health of all other microservices. It periodically checks the readiness and operational status of all configured microservices. Additionally, it tracks the status of the SEPPs registered services, whether they are Running, Suspended, or Deregistered, and continuously monitors their state. The nrf client periodically retrieves these service status details from the appInfo component through its REST API.

Table 2-1 (Cont.) Microservices

Microservice	Description
Mediation Service	- Mediation service allows to perform HTTP/2 signaling API message mediation between NFs. It resolves interoperability issues for 5GC inter-NF communication when there are vendor-specific implementations in NFs. It provides configuration framework to configure the mediation policy rules to be applied on the HTTP signaling messages. It allows dynamic configuration on mediation policies which will be applied on both request and response. - In SEPP mediation service allows MNOs to resolve the inter-op issues between PLMNs by manipulating the inter PLMN messages. The MNO shall rely on mediation service capabilities to provision the mediation rules. - This is a common microservice provided by SCP.
Coherence Service	- Coherence Service is responsible for storing UE Authentication status and UDR Profile data. - PN32F gets the UE Authentication status from UDR, and UDR Profile data from NRF Client using UDR Discovery request. - This service is used for stateful security countermeasure (Cat- 3 Previous location Check feature).

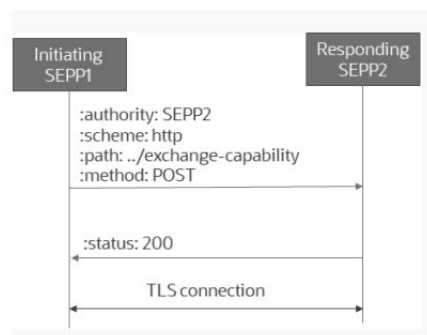
2.1 N32c and N32f Call flows

The following are the N32c and N32f call flows of SEPP:

N32c Call flow

Following diagram illustrates the N32c Call flow:

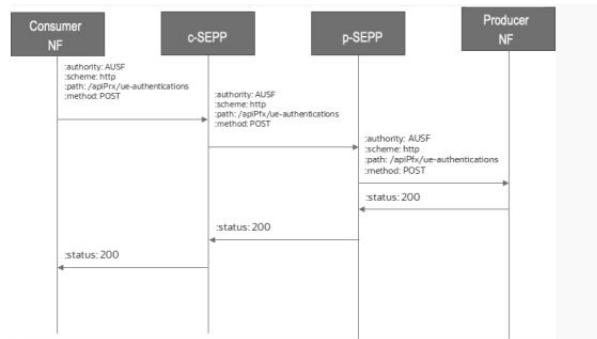
Figure 2-3 N32c Call flow



- The initiating SEPP issues a HTTP POST request towards the responding SEPP with the request body containing the "SecNegotiateReqData" with supporting security capabilities that is, Protocol for N32 Interconnect Security (PRINS) and/or Transport Layer Security (TLS).
- On successful processing of the request, the responding SEPP responds to the initiating SEPP1 with "200 OK" status code and a POST response body with selected security capability i.e. Protocol for N32 Interconnect Security (PRINS) and/or Transport Layer Security(TLS).

N32f Call flow: HTTP scheme is used by NF to communicate to SEPP

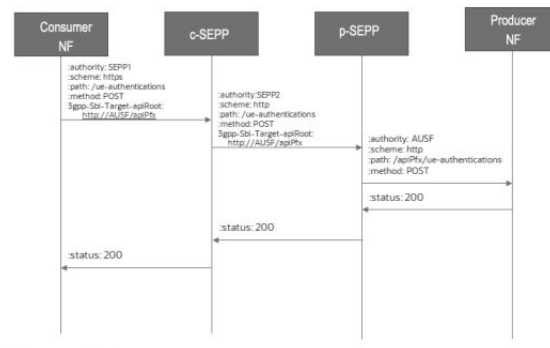
Following diagram illustrates the HTTP scheme is used by NF to communicate to SEPP in N32f Call flow:

Figure 2-4 N32f Call flow: HTTP scheme is used by NF to communicate to SEPP

- The SEPP on the NF service consumer (c-SEPP) and the SEPP on the NF service producer (p-SEPP) negotiate the security capabilities using the procedure defined in Handshake call flow. The SEPPs mutually negotiate to use Transport Layer Security (TLS) as the security policy.
- The NF service consumer is configured to route all HTTP messages with inter-PLMN FQDN as the "authority" part of the URI via the c-SEPP. The c-SEPP acts as a HTTP proxy.
- The c-SEPP forwards the HTTP service request within the N32-f TLS tunnel established.
- The p-SEPP forwards the HTTP service request to the NF service producer.
- The NF service producer sends the HTTP service response.
- The p-SEPP forwards the HTTP service response within TLS tunnel to the c-SEPP.
- The c-SEPP forwards the HTTP service response to the NF service consumer.

N32f Call flow: HTTPS with 3gpp-sbi-target-apiroot header is used by NF to communicate with SEPP

Following diagram illustrates the HTTPS with 3gpp-sbi-target-apiroot header is used by NF to communicate with SEPP in N32f Call flow:

Figure 2-5 N32f Call flow: HTTPS with 3gpp-sbi-target-apiroot header is used by NF to communicate with SEPP

- The SEPP on the NF service consumer (c-SEPP) and the SEPP on the NF service producer (p-SEPP) negotiate the security capabilities using the procedure defined in the handshake call flow. The SEPPs mutually negotiate to use TLS as the security policy.
- NF service consumer is configured with c-SEPP FQDN and sets up a TLS connection with c-SEPP.
- The NF service consumer sends the HTTP service request within the TLS connection to the c-SEPP, including a 3gpp-sbi-target-apiroot header set to the apiRoot of the p-NF.
- The c-SEPP forwards the HTTP service request within the N32-f TLS tunnel established.
- The p-SEPP extracts the HTTP message received on the TLS connection, and then seeing that the URI scheme of the NF service producer is "https" in 3gpp-sbi-target-apiroot header, the p-SEPP sets up a TLS connection with the NF service producer.
- The p-SEPP forwards the HTTP service request to the NF service producer.
- The NF service producer sends the HTTP service response.
- The p-SEPP forwards the HTTP service response within TLS tunnel to the c-SEPP.
- The c-SEPP upon receiving the HTTP response message within the TLS tunnel, forwards the response to the NF service consumer.

3

SEPP Functionalities and Features

SEPP includes the following core functionalities, features, and deployment support features:

Note

The performance and capacity of the SEPP system may vary based on the call model, Feature/Interface configuration, and underlying CNE and hardware environment.

This section describes the features and configuration details supported by SEPP.

Table 3-1 SEPP Features and Configuration details

Feature name	Enabled or Disabled by default	Configuration Type	Licensing SKU
SEPP Deployment Using Flux CD	N/A	Flux/GitOps, Helm	Oracle Communications Cloud Native Core, GitOps
Hosted SEPP	Disabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Support for Max-Response Timer Header	Disabled	Helm, REST API	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
PLMN Validation in Get Access Token Requests	Disabled	Helm	Oracle Communications Cloud Native Core, 5G Signaling Firewall
Error Response and Logging Enhancement	Enabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
LCM-Based Automation	Disabled	Helm	Oracle Communications Cloud Native Core, Automated Lifecycle Management
Traffic Prioritization / Egress Gateway Pod Protection	Disabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
SEPP and NRF Communication	Enabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
SEPP with 5G NIF	Disabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
TLS 1.3 Support for Kubernetes API	Disabled	Helm	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
LCI and OCI Headers	Disabled	Helm	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Dashboard Support for Detecting Vulnerable Messages	Enabled	No separate configuration stated	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual

Table 3-1 (Cont.) SEPP Features and Configuration details

Feature name	Enabled or Disabled by default	Configuration Type	Licensing SKU
Support for Dual Stack	Enabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Advanced Networking
Multiple SEPP Instances on Shared cnDBTier	Enabled	Helm, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Proactive Status Updates	Disabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Advanced Networking
Traffic Segregation	Disabled	Helm	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Originating Network ID Header Validation/ Insertion/Transposition	Disabled	REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Common Service APIs in CNC Console	Enabled	CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Support for TLS	Enabled	Helm	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Automated Certificate Lifecycle Management	N/A	No additional SEPP configuration	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Load Sharing among Multiple Remote SEPP Nodes	Disabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Separate N32c/N32f Port Configuration	Disabled	REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
cnDBTier APIs in CNC Console	Enabled when cnDBTier is deployed	CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
5G SBI Message Mediation	Disabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Topology Hiding	Disabled	REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual

Table 3-1 (Cont.) SEPP Features and Configuration details

Feature name	Enabled or Disabled by default	Configuration Type	Licensing SKU
Global Rate Limiting — Egress Gateway	Disabled	Helm, REST API	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Global Rate Limiting — Ingress Gateway	Enabled	Helm, REST API	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Multiple PLMN Support	Enabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
TLS for Roaming Hubs and IPX Providers	Enabled	REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Overload Control	Disabled	Helm, REST API	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Alternate Routing Across Remote SEPPs	Enabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Server Header Support	Disabled	Helm	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Networks with/without 3GPP-SBI-Target-API-Root Header	Enabled	Helm	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
SAN Validation	Enabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
SEPP Message Feed	Disabled	Helm	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Steering of Roaming	Disabled	REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Ingress Roaming Signaling Rate Limiting per Remote SEPP Set	Disabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Georedundancy	N/A – not a feature	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge

Table 3-1 (Cont.) SEPP Features and Configuration details

Feature name	Enabled or Disabled by default	Configuration Type	Licensing SKU
			Protection Proxy – 25K Active Subscribers Perpetual
Kubernetes Resource Support	N/A – not a feature	Helm	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Alternate Routing and Load Sharing via DNS SRV	Disabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Egress Roaming Signaling Rate Limiting per PLMN	Disabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Nonverbose Error Responses	Enabled	REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Cat-0 SBI Message Schema Validation	Disabled	REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Cat-1 Service API Validation	Disabled	REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Cat-1 NRF Query-Parameter Validation	Disabled; requires Cat-1 Service API Validation enabled	REST API, CNC Console	5G Signaling Firewall
Cat-2 Network ID Validation	Disabled	REST API, CNC Console	Oracle Communications Cloud Native Core, Security Edge Protection Proxy – 25K Active Subscribers Perpetual
Cat-3 Previous Location Check	Disabled	Helm, REST API, CNC Console	Oracle Communications Cloud Native Core, 5G Signaling Firewall
Cat-3 Time Check for Roaming Subscribers	Disabled	REST API, CNC Console	Oracle Communications Cloud Native Core, 5G Signaling Firewall

The following table depicts the SEPP features which support Hosted SEPP mode and Roaming Hub mode:

Table 3-2 SEPP Features that Support Hosted SEPP Mode and Roaming Hub mode

SL No	Feature Category	Feature	Support ed in SEPP Mode	Support ed in Hosted SEPP Mode	Support ed for Roamin g Hub
1	Core Functionalities	SEPP MNO functionality Support	Y	Y	N
2	Core Functionalities	NF Authentication using TLS certificate	Y	Y	Y
3	Core Functionalities	Dedicated Gateways for IntraPLMN and InterPLMN Traffic	Y	N	N (only supports InterPlm n Traffic)
4	Core Functionalities	Support for Core Networks with or without SCP / Routing Proxy	Y	N	N (only supports InterPlm n Traffic)
5	Core Functionalities	DNS Support for Remote SEPP Resolution	Y	Y	Y
6	SEPP Features	SEPP Deployment Using Flux CD	Y	Y	Y
7	SEPP Features	Hosted SEPP	Y	Y	Y
8	SEPP Features	Support for Max-Response Timer Header in SEPP	Y	Y	Y
9	SEPP Features	Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers	Y	N	N
10	SEPP Features	Error Response and Logging Enhancement for SEPP	Y	Y	Y
11	SEPP Features	Lifecycle Management (LCM) Based Automation	Y	Y	Y
12	SEPP Features	Traffic Prioritization using Egress Gateway Pod Protection	Y	N	N
13	SEPP Features	SEPP and NRF Communication	Y	N	N
14	SEPP Features	NRF Interface Registration and Deregistration	Y	N	N
15	SEPP Features	NRF Selection Mechanisms using nrf client	Y	N	N
16	SEPP Features	Integrating SEPP with 5G Network Intelligence Fabric (5G NIF)	Y	N	N
17	SEPP Features	TLS 1.3 Support for Kubernetes API	Y	Y	Y
18	SEPP Features	LCI and OCI Headers in 5G Architecture	Y	Y	N
19	SEPP Features	SEPP Dashboard Support for Detecting Vulnerable Messages	Y	Y	Y
20	SEPP Features/ Security Countermeasure Features	Security Countermeasure Nonverbose Error Responses	Y	Y	Y

Table 3-2 (Cont.) SEPP Features that Support Hosted SEPP Mode and Roaming Hub mode

SL No	Feature Category	Feature	Support ed in SEPP Mode	Support ed in Hosted SEPP Mode	Support ed for Roamin g Hub
21	SEPP Features/ Security Countermeasure Features	Cat-1 NRF Service API Query Parameters Validation	Y	N	N
22	SEPP Features/ Security Countermeasure Features	Cat-0 SBI Message Schema Validation Feature	Y	Y	Y
23	Security Countermeasure Features	Cat-1 Service API Validation Feature	Y	Y	Y
24	SEPP Features/ Security Countermeasure Features	Cat-2 Network ID Validation Feature	Y	Y	Y
25	SEPP Features/ Security Countermeasure Features	Cat-3 Previous Location Check Feature	Y	N	N
26	SEPP Features/ Security Countermeasure Features	Cat-3 Time Check for Roaming Subscribers	Y	N	N
27	SEPP Features	Support for Dual Stack	Y	Y	Y
28	SEPP Features	Multiple SEPP Instances on Shared cnDBTier Cluster	Y	N	N
29	SEPP Features	Proactive Status Updates on SEPP	Y	Y	Y
30	SEPP Features	Traffic Segregation	Y	Y	Y
31	SEPP Features	Support for Originating Network Id Header Validation, Insertion, and Transposition	Y	N	N
32	SEPP Features	Support for Common Service APIs in CNC Console	Y	Y	Y
33	SEPP Features	Support for TLS	Y	Y	Y
34	SEPP Features	Support for Automated Certificate Lifecycle Management	Y	Y	Y
35	SEPP Features	Load Sharing among Multiple Remote SEPP Nodes	Y	Y	Y
36	SEPP Features	Separate Port Configurations for N32c and N32f on the Egress Routes	Y	N	Y
37	SEPP Features	Support for cnDBTier APIs in CNC Console	Y	Y	Y
38	SEPP Features	5G SBI Message Mediation Support	Y	N	Y
39	SEPP Features	Topology Hiding	Y	N	Y

Table 3-2 (Cont.) SEPP Features that Support Hosted SEPP Mode and Roaming Hub mode

SL No	Feature Category	Feature	Support ed in SEPP Mode	Support ed in Hosted SEPP Mode	Support ed for Roamin g Hub
40	SEPP Features	Global Rate Limiting on Egress Gateway of SEPP	Y	Y	Y
41	SEPP Features	Global Rate Limiting on Ingress Gateway of SEPP	Y	Y	Y
42	SEPP Features	Multiple PLMN Support for Local and Remote SEPP	Y	Y	Y
43	SEPP Features	Support of TLS for Roaming Hubs and IPX Providers	Y	N	Y
44	SEPP Features	Overload Control	Y	Y	Y
45	SEPP Features	Alternate Routing Across Remote SEPPs	Y	Y	Y
46	SEPP Features	Server Header Support	Y	Y	Y
47	SEPP Features	Support for Networks with/without 3GPP-SBI-Target-API-Root Header	Y	Y	Y
48	SEPP Features	Subject Alternate Name (SAN) Validation	Y	Y	Y
49	SEPP Features	SEPP Message Feed	Y	N	Y
50	SEPP Features	Steering of Roaming (SOR) feature	Y	N	N
51	SEPP Features	Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set	Y	Y	Y
52	SEPP Features	Georedundancy	Y	Y	Y
53	SEPP Features	Support for Kubernetes Resource	Y	Y	Y
54	SEPP Features	Alternate Routing and Load Sharing based on the DNS SRV Record for Home Network Functions	Y	N	N
55	SEPP Features	Rate Limiting for Egress Roaming Signaling per PLMN	Y	Y	Y
56	Deployment features	Helm Validation Framework	Y	Y	Y
57	Deployment features	Automated Testing Suite Support	Y	Y	Y
58	Deployment features	Configuring SEPP using CNC Console	Y	Y	Y

3.1 Core Functionalities

The following are the core functionalities of SEPP:

3.1.1 SEPP MNO functionality Support

Feature Description

When mobile network operators (MNOs) need to connect with each other, they must ensure that their communication is secure and well-managed. Security Edge Protection Proxy (SEPP)

plays an important role in protecting and controlling these exchanges. SEPP acts as a protective gateway, allowing operators to safely exchange data while preventing direct access to their internal network functions.

SEPP is designed to enhance security and optimize efficiency in inter-operator communication.

1. Single point of entry for network traffic

SEPP serves as a central gateway for all external traffic entering or leaving an operator's network. By consolidating access points into a single controlled entry and exit, SEPP eliminates potential security risks associated with multiple access points. This approach simplifies network management while enhancing security.

2. Improved security and network protection

By routing all external traffic through SEPP, operators can safeguard their internal network functions from potential threats. Without SEPP, these functions might be directly exposed, increasing vulnerability to attacks. SEPP ensures that only authorized and encrypted communications are exchanged between networks, maintaining the security of the core network.

3. Connecting with multiple roaming partners

SEPP streamlines roaming and interconnectivity between different operators. Instead of establishing separate connections for each roaming partner, operators can utilize SEPP as a single, unified interconnect point. This allows secure communication with multiple networks at the same time, making it easier to manage roaming agreements.

SEPP has been enhanced to manage multiple connections across various operators and core networks, as outlined below:

1. Multiple SEPP connections to other operators

Operators often need to connect with several different networks for roaming and data exchange. The improved SEPP can now communicate with multiple SEPPs from different roaming partners at the same time. This allows for secure interconnection with many operators without needing separate systems for each one.

2. SEPP as a shared security gateway for multiple core networks

Many operators run multiple core networks. Instead of using a separate SEPP for each core, the enhanced SEPP can now act as a shared security gateway for multiple core networks. This reduces complexity and improves efficiency by centralizing security controls.

3. Enhanced N32-c handshake for secure communication

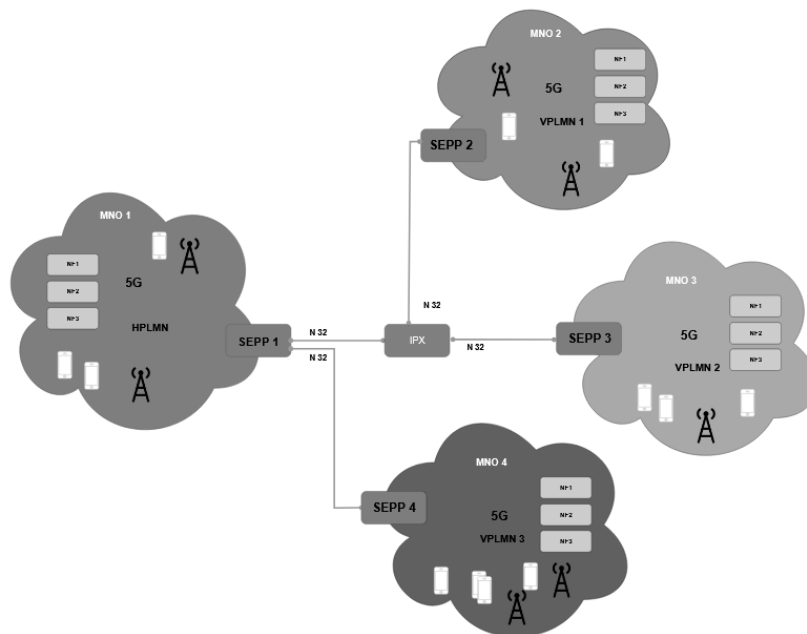
The N32-c interface manages secure communication between SEPPs in different networks. Previously, SEPP could only establish a single connection at a time. Now, it can perform multiple N32-c handshakes with different roaming partner SEPPs at the same time. This helps operators handle roaming connections more efficiently and improves overall network performance.

4. Support for multiple N32-f interfaces

The upgraded SEPP can now establish multiple N32-f connections with different roaming partners. This ensures that encrypted communication can be securely managed across several different operators, improving network security and reliability.

SEPP supports the following interfaces:

- N32-c interface (TLS mode only)
- N32-f interface (TLS mode only)
- TLS v1.2 and v1.3

Figure 3-1 SEPP MNO functionality Support Architecture

This diagram illustrates the SEPP MNO Functionality Support feature, demonstrating how different mobile network operators (MNOs) securely communicate over N32 interfaces using SEPPs. The IPX functions as a bridge between SEPPs, facilitating secure and controlled interconnectivity between the home and visited PLMNs. This configuration simplifies roaming agreements, boosts security, and enhances scalability, enabling operators to manage multiple roaming partners effectively while maintaining a single, controlled entry and exit point for all external traffic.

Managing MNO Functionality

Enable

This feature is the core functionality of SEPP. You do not need to enable or disable this feature. It is enabled by default.

Configure

You can configure the SEPP MNO functionality using REST API, CNC Console, or helm parameters:

- Configure SEPP MNO functionality using REST API: Perform the feature configurations as described *Remote SEPP* section in *Oracle Communications Security Edge Protection Proxy Rest API Guide*.
- Configure SEPP MNO functionality using CNC Console: Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Configuring using Helm parameters

Set the local profile for the SEPP Instance.

```
LocalProfile:
  name: "SEPP-SVC"
  #Size of PLMN list can be configured through Helm parameter. The default
  value for minimum is 1 and maximum is 30 for SEPP mode and 900 for IPX mode
  plmnIdList: [{"mcc":"333","mnc":"222"}, {"mcc":"444","mnc":"555"},
  {"mcc":"444","mnc":"55"}]
  # Do not change this value, this will be always true
  sbiTargetApiRootSupported: true
  # Enable PLMN ID List Validation in Exchange Capability Request, Default
  set to true
  n32cHandshakePlmnIdListValidationEnabled : true
  # PLMN ID List Validation Type in Exchange Capability Request, can be
  SUBSET or STRICT only
  n32cHandshakePlmnIdListValidationType: "SUBSET"
  # Enable PLMN ID List sending to Remote SEPP in Exchange Capability
  Request, Default set to true for SEPP Mode and false for RH Mode
  n32cHandshakePlmnIdListSend: true
  sanValidationRequired: true
  domain: "svc.cluster.com"
  seppViaVersion: "2.0"
  viaHeaderSeppViaInterFqdn: "2.0 SEPP-sepp2.inter.oracle.com"
  viaHeaderSeppViaIntraFqdn: "2.0 SEPP-ocsepp-plmn-ingress-gateway.seppsvc"
  interPlmnFqdn: *nfFqdnRef
  intraPlmnFqdn: "ocsepp-plmn-ingress-gateway.seppsvc"
  supportedSecurityCapabilityList:
    - "TLS"
  apiPrefix: ""
  retryInterval: 300000
  maxRetry: -1
  nfInstanceId: *nfInstIdRef
```

To make this functionality work, we would require the following:

- Remote SEPP
- Remote SEPP Set
- NRF Registration

For more details, see *Oracle Communications Security Edge Protection Proxy Rest API Guide*.

Observe

For more information on support of N32-c and N32-f interface feature metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.

2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.1.2 NF Authentication using TLS certificate

HTTPS support is a minimum requirement for 5G NFs as defined in 3GPP TS 33.501. HTTPS enables end to end encryption of messages to ensure security of data. HTTPS requires creation of TLS (Mutual TLS by 2 way exchange of ciphered keys).

Managing NF Authentication using TLS Certificate

Steps to Enable HTTPS in SEPP

Certificate Creation

To create certificate user must have the following files:

- ECDSA private key and CA signed certificate of OCSEPP (if initial algorithm is ES256)
- RSA private key and CA signed certificate of OCSEPP (if initial algorithm is RS256)
- TrustStore password file
- KeyStore password file
- CA certificate

Secret Creation

Execute the following command to create secret:

```
$ kubectl create secret generic ocseppaccesstoken-secret --from-  
file=ecdsa_private_key_pkcs8.pem --from-file=rsa_private_key_pkcs1.pem  
--from-file=trustStorePassword.txt --from-file=keyStorePassword.txt --  
from-file=ecdsa_ocsepp_certificate.crt--from-file=desertification -n  
ocsepp
```

Certificate and Key Exchange

Once the connection is established, both parties can use the agreed algorithm and keys to securely send messages to each other. The handshake has 3 main phases:

- Hello
 - Certificate Exchange
 - Key Exchange
1. **Hello:** The handshake begins with the client sending a ClientHello message. This contains all the information the server needs in order to connect to the client via SSL, including the various cipher suites and maximum SSL version that it supports. The server responds with a ServerHello, which contains similar information required by the client, including a decision based on the client's preferences about which cipher suite and version of SSL will be used.
 2. **Certificate Exchange:** Now that contact has been established, the server has to prove its identity to the client. This is achieved using its SSL certificate, which is a very tiny bit like its passport. An SSL certificate contains various pieces of data, including the name of the owner, the property (Example: domain) it is attached to, the certificate's public key, the digital signature and information about the certificate's validity dates. The client checks that it either implicitly trusts the certificate, or that it is verified and trusted by one of several

Certificate Authorities (CAs) that it also implicitly trusts. The server is also allowed to require a certificate to prove the client's identity, but this only happens in very sensitive applications.

3. **Key Exchange:** The encryption of the actual message data exchanged by the client and server will be done using a symmetric algorithm, the exact nature of which was already agreed during the Hello phase. A symmetric algorithm uses a single key for both encryption and decryption, in contrast to asymmetric algorithms that require a public/private key pair. Both parties need to agree on this single, symmetric key, a process that is accomplished securely using asymmetric encryption and the server's public/private keys.

The client generates a random key to be used for the main, symmetric algorithm. It encrypts it using an algorithm also agreed upon during the Hello phase, and the server's public key (found on its SSL certificate). It sends this encrypted key to the server, where it is decrypted using the server's private key, and the interesting parts of the handshake are complete. The parties are identified that they are talking to the right person, and have secretly agreed on a key to symmetrically encrypt the data that they are about to send each other. HTTP requests and responses can be sent by forming a plain text message and then encrypting and sending it. The other party is the only one who knows how to decrypt this message, and so Man In The Middle Attackers are unable to read or modify any requests that they may intercept.

SEPP supports following cipher suites

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

HTTPS Encrypted Communication

Once the HTTPS handshake is complete all communications between the client and the server are encrypted. This includes the full URL, data (plain text or binary), cookies and other headers.

The only part of the communication not encrypted is what domain or host the client requested a connection. This is because when the connection is initiated an HTTP request is made to the target server to create the secure connection. Once HTTPS is established the full URL is used.

This initialization only needs to occur once for each unique connection. This is why HTTP/2 has a distinct advantage over HTTP/1.1 since it multi-plexes connections instead of opening multiple connections.

Helm Configuration to enable HTTPS on SEPP:

Sample values.yaml to enable HTTPS on SEPP:

```
#Enabling it generates key and trust store for https support
  initssl: true          (Note: secret has to be created if its set to true)
#If true opens https port on egress gateway
  enableincominghttps: false
#Enabling it egress makes https request outside
  enableoutgoinghttps: true
  (Note: initssl should be set to true if either enableincominghttps or
  enableoutgoinghttps is enabled )
#KeyStore and TrustStore related private key and Certificate configuration
  (Note: The configuration names specified should be same as the      file names
  specified when creating secret)

  privateKey:
```

```
k8SecretName: ocsepp-n32-secret
k8Namespace: ocsepp
rsa:
  fileName: rsa_private_key_pkcs1.pem

certificate:
  k8SecretName: ocsepp-n32-secret
  k8Namespace: ocsepp
  rsa:
    fileName: ocsepp.cer

caBundle:
  k8SecretName: ocsepp-n32-secret
  k8Namespace: ocsepp
  fileName: caroot.cer

keyStorePassword:
  k8SecretName: ocsepp-n32-secret
  k8Namespace: ocsepp
  fileName: key.txt

trustStorePassword:
  k8SecretName: ocsepp-n32-secret
  k8Namespace: ocsepp
  fileName: trust.txt

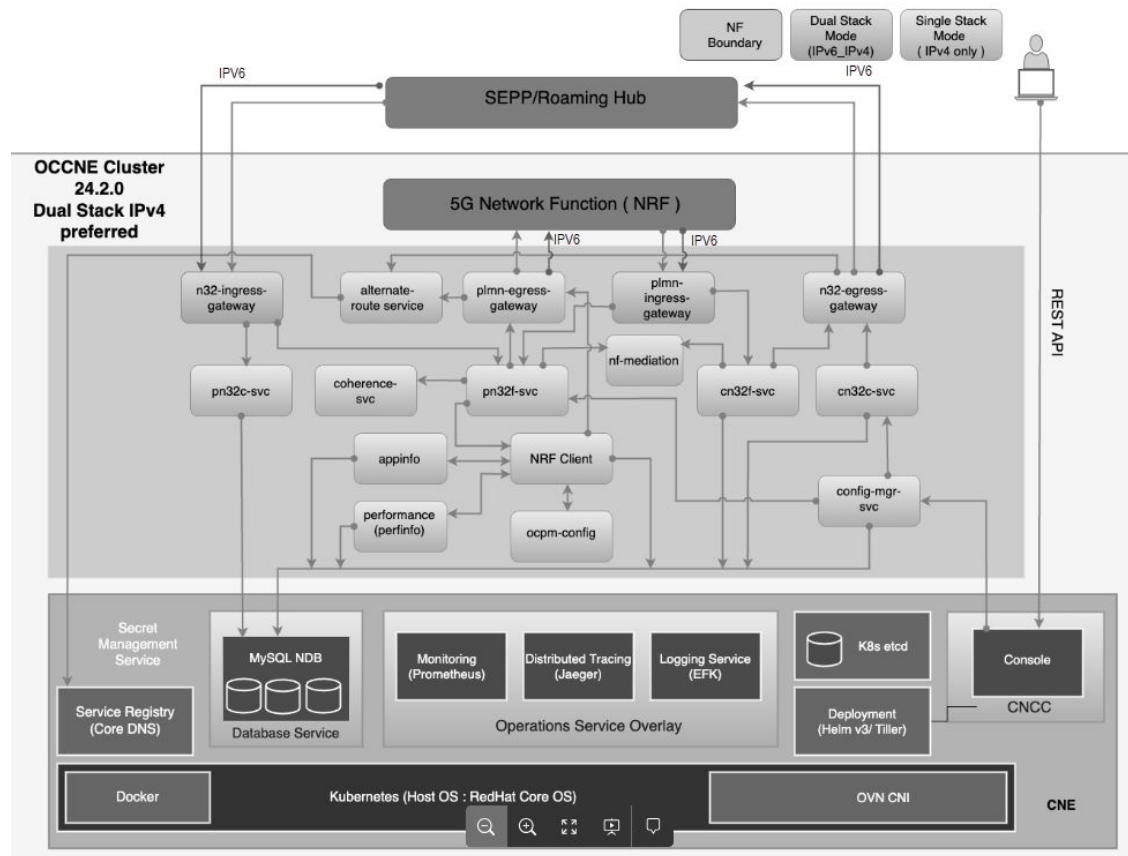
initialAlgorithm: RS256
```

3.1.3 Dedicated Gateways for IntraPLMN and InterPLMN Traffic

In the earlier releases, a single gateway was used to support both intraPLMN and interPLMN traffic. This led to performance related issues in supported capacity and error handling procedures. To overcome these issues, the architecture was modified. The modification involves establishing dedicated set of gateways for intraPLMN traffic and interPLMN traffic. It also involves optimization of resources to prevent utilization of more resources by these dedicated gateways. This brings in flexibility of scaling the resources independently based on either intraPLMN traffic or interPLMN traffic.

The following diagram depicts the topology for dedicated gateways for IntraPLMN InterPLMN traffic.

Figure 3-2 Dedicated gateways for IntraPLMN InterPLMN traffic



Managing Dedicated Gateways for IntraPLMN and InterPLMN traffic

Enable

This feature is the core functionality of SEPP. You do not need to enable or disable this feature. It is enabled by default.

Configure

You can configure the Dedicated Gateways for IntraPLMN and InterPLMN traffic using helm parameters.

Observe

For more information on dedicated gateways for IntraPLMN and InterPLMN traffic feature metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.

2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.1.4 Support for Core Networks with or without SCP/ Routing Proxy

The SEPP can be deployed in networks where SCP or Routing Proxy is deployed. Routing rules in such scenarios changes. To handle such deployments, a configuration is now provided at the Home SEPP to assist in identifying if the SEPP must route the message to the intended producer or just forward it to the SCP (if deployed in the network) such that the SCP could take the decision to route to the right producer.

Managing Support for Core Networks with or without SCP/ Routing Proxy

Enable

This feature is managed through the routes at the plmn-egress-gateway. A separate route must be created for the SCP/Routing Proxy to direct all incoming requests to the configured SCP/ Routing Proxy.

The priority/order of the route should be carefully selected to ensure that existing NRF routes are not impacted.

The SCP/Routing Proxy route should be configured with an order value of 100.

Configure

SCP/Routing Proxy Route Configuration

The SCP/Routing Proxy route needs to be configured at the plmn-egress-gateway with the following details:

```
[{
  "id": "scpRoute",
  "uri": "http://ocsepp.com",
  "order": 100,
  "filters": [{
    "args": {
      "errorHandling": [{
        "priority": 0,
        "actionSet": "scp_action_0",
        "errorCriteriaSet": "scp_criteria_1"
      }, {
        "priority": 1,
        "actionSet": "scp_action_0",
        "errorCriteriaSet": "scp_criteria_0"
      }
    ],
    "peerSetIdentifier": "scpPeer",
    "customPeerSelectorEnabled": true
  },
  "name": "SbiRouting"
}],
  "metadata": {
    "httpRuriOnly": true,
    "httpsTargetOnly": true,
    "sbiRoutingEnabled": true,
    "ignoreMaxRspTimeHeader": false,
    "sbiRoutingWeightBasedEnabled": true
  }
},
```

```

    "predicates": [{
      "args": {
        "pattern": "/*"
      },
      "name": "Path"
    }]
  }
}

```

For this SCP route, Peers, Peersset, SbiRoutingErrorCriteriaSets, and SbiRoutingErrorActionSets must also be configured at the plmn-egress-gateway as required.

Route Configuration for Producer NFs

Producer NFs (Example: UDR) can be statically configured using route configurations. Different path predicates are required for each Producer NF to ensure correct routing behavior.

For instance, to configure a UDR route, the following configuration needs to be applied. Please note that the order value for all Producer NFs (except NRF) should fall within the range of 1-100.

```

[
  {
    "id": "udrRoute",
    "uri": "http://ocsepp.com",
    "order": 10,
    "filters": [
      {
        "args": {
          "errorHandling": [
            {
              "priority": 0,
              "actionSet": "udr_action_0",
              "errorCriteriaSet": "udr_criteria_1"
            },
            {
              "priority": 1,
              "actionSet": "udr_action_0",
              "errorCriteriaSet": "udr_criteria_0"
            }
          ],
          "peerSetIdentifier": "udrPeer",
          "customPeerSelectorEnabled": true
        },
        "name": "SbiRouting"
      }
    ],
    "metadata": {
      "httpRuriOnly": true,
      "httpsTargetOnly": true,
      "sbiRoutingEnabled": true,
      "ignoreMaxRspTimeHeader": false,
      "sbiRoutingWeightBasedEnabled": true
    },
    "predicates": [
      {
        "args": {
          "pattern": "/nudr-*/"
        },
        "name": "Path"
      }
    ]
  }
]

```

In addition to the route configuration for the UDR, the following components need to be created for the route to function correctly:

- Peers
- Peersset
- SbiRoutingErrorCriteriaSets
- SbiRoutingErrorActionSets

① Note

Peer configuration and Peersset configuration also needs to be added for SCP and NRF. For more information, see Egress Gateway REST APIs in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Observe

For more information on Support for Core Networks with or without SCP/ Routing Proxy feature metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alerts still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.1.5 DNS Support for Remote SEPP Resolution

SEPP to SEPP communication was enforced through configuring the IP address for the far end SEPP. Operators may choose to use a DNS instead and so the SEPP must use the DNS and resolve the IP address of the far end SEPP. This feature allows configuring only the FQDN for the far end SEPP and subsequently needing DNS resolution for identifying the IP address. This feature enhances flexibility to operators now use DNS instead of providing a static IP address.

Managing DNS Support for Remote SEPP Resolution

Enable

This feature is the core functionality of SEPP. You do not need to enable or disable this feature. It is enabled by default.

Configure

DNS configuration can be done in Kubernetes DNS of the cluster. No HELM or REST based configurations are required.

Observe

For more information on DNS support for remote SEPP resolution feature metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2 SEPP Features

The following are SEPP features:

3.2.1 SEPP Deployment Using Flux CD

This feature enables fully automated, GitOps-driven deployment of SEPP using a centralized Flux CD setup. It leverages declarative manifests stored in Git to consistently install and manage SEPP on target Kubernetes clusters via Helm charts.

Key Features:

- **GitOps-Driven Deployment:** Uses declarative manifests stored in Git to manage SEPP on Kubernetes.
- **Pre-Packaged Templates:** Flux CD artifacts (occncc_fluxcd_artifacts_<version>.zip) provide ready-to-use repository structure.
- **Helm Integration:** A HelmRelease manifest defines installation and lifecycle management of SEPP.
- **Source Configuration:** Flux manifests specify the Git repo, branch, and target cluster access.
- **Kustomize Support:** Organizes and exposes resources for seamless deployment through Flux.

Benefits

- Declarative, version-controlled deployments
- Reduced manual intervention and configuration drift
- Faster setup using pre-built templates
- Scalable and maintainable multi-cluster deployment model

This feature simplifies SEPP deployment by combining Helm, Flux CD, and GitOps best practices into a cohesive and production-ready workflow.

For more details about the deployments using Flux Setup, refer to 'SEPP Deployment using Flux Setup Appendix ' section in the *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

3.2.2 Hosted SEPP Feature

Introduction

The Hosted SEPP feature enhances Oracle's Hosted SEPP deployment by enabling multi-tenancy support. This feature introduces the ability to handle multiple Fully Qualified Domain Names (FQDNs) and certificates for different tenants, providing improved flexibility in inter-network communications and partner routing.

Background

In previous deployments, Hosted SEPP supported only a single FQDN, which limited the flexibility of inter-network communications and routing decisions. In the earlier model (Hosted SEPP deployment model 2.2), it was difficult to differentiate between hosted and roaming partners based on FQDNs, leading to the grouping of SEPPs with similar PLMN IDs.

With the new model, the SEPP can handle multiple identities, facilitating seamless connectivity to different roaming and hosted partners using different FQDNs. This eliminates the constraints of using a single FQDN for all connections, thus improving the efficiency of traffic routing and reducing the need for manual intervention.

Detailed Description

The Hosted SEPP feature enhances Oracle's Hosted SEPP deployment by enabling multi-tenancy support. This feature introduces the ability to handle multiple Fully Qualified Domain Names (FQDNs) and certificates for different tenants, providing improved flexibility in inter-network communications and partner routing.

This feature introduces the ability to configure multiple FQDNs for a single Hosted SEPP instance, allowing operators to:

- Map each FQDN to specific PLMNs.
- Handle connections from multiple hosted and roaming partners efficiently.
- Improve N32 interconnection paths by managing different certificates for each FQDN.

Key enhancements include:

- Reduced TLS and N32c handshake information sent to remote SEPPs.
- Dynamic population of remote PLMN-IDs in the configuration, allowing for flexible interconnection setup.
- Support for multiple identities on both N32s and N32 interfaces, with improved TLS handshake processes for each identity.

Architecture

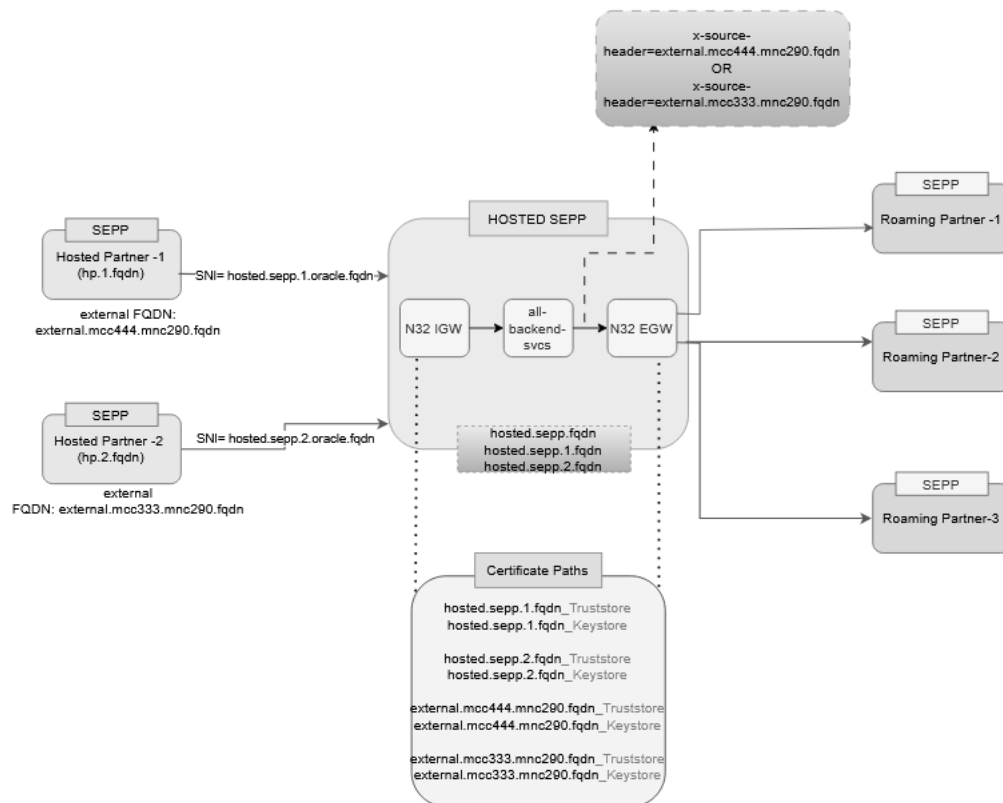
The architecture of the Hosted SEPP with Multiple FQDNs operates as follows:

1. **Hosted SEPP** can host multiple roaming and hosted partners, each with their own FQDN configuration.
2. **Internal and External FQDNs** are configured for each partner, where internal FQDNs are used for communication between the hosted partner and the Hosted SEPP, and external FQDNs are used for communication between the Hosted SEPP and roaming partners.
3. **N32s and N32 Interfaces:** The interface between the hosted partner and Hosted SEPP is N32s, while the interface to roaming partners is N32.

4. **TLS Handshake:** During the TLS handshake, the FQDN is included as part of the SNI (Server Name Indication). This allows the SEPP to identify which certificates to present based on the FQDN in use.
5. **Multiple FQDNs:** Each FQDN comes with its own associated certificates and configuration. The SEPP dynamically checks its keystore and trust store to ensure the correct certificates are used.

The following diagram represents the traffic representation in the Hosted SEPP:

Figure 3-3 Hosted SEPP Traffic Representation



This diagram outlines the architecture and communication flow for the Hosted SEPP and its interactions with Hosted Partners and Roaming Partners.

Step 1: Hosted SEPP Overview

- Hosted SEPP acts as a proxy for communication between hosted partners and roaming partners.
- Communication between hosted partners and Hosted SEPP occurs over n32s.
- Communication between Hosted SEPP and roaming partners happens over n32.

Step 2: Partner Configuration

- Hosted SEPP hosts multiple hosted partners.
- Each hosted partner supports:

- Internal fqdn (for n32s communication).
- External fqdn (for communication with roaming partners).
- Hosted SEPP supports multiple fqdns for each hosted partner.
- Each hosted partner uses distinct fqdns and certificates.

Step 3: N32s Handshake Process

- Hosted Partner sends a handshake capability request to the Hosted SEPP using its internal fqdn populated in the SNI (Server Name Indication) header.
- SNI in the TLS handshake is set to `hosted.sepp.1.inter.oracle.com`.
- N32 Ingress Gateway checks if the config and certificates for the fqdn are available:
 - If present, it responds with the appropriate certificates.
 - If absent, it sends default certificates.
- Once the TLS handshake is completed:
 - Hosted SEPP responds back with the fqdn `hosted.sepp.1.inter.oracle.com`.
 - No plmn list is included (the list is dynamic and changes with partner additions/removals).

Step 4: Roaming Partner Profile and Mapping

- Roaming profiles for each roaming partner are posted to Hosted SEPP before the n32 handshake.
 - These profiles contain the resolvable fqdn of the roaming partner (Example: `external.mcc444.mnc290.fqdn`).
- Mapping between hosted and roaming partners must be established before proceeding with handshakes.
- Until mapping is created, Hosted SEPP will not accept handshake requests from roaming partners.

Step 5: Capability Exchange and N32 Handshake

- Once mapping is complete, Hosted SEPP initiates the capability exchange.
 - The x-source-header header is populated with `external.mcc444.mnc290.fqdn`.
 - N32 Egress Gateway uses this to determine the client identity for the TLS handshake with the roaming partner.
- After capability exchange, N32 EGW completes the TLS handshake.
 - If the configuration is missing, default certificates are used.
- The n32 handshake with the roaming partner is established after successful TLS handshake.

Notes for Configuration and Operation

- Gateway pods take 130 seconds to come up; adjust ephemeral storage and probe settings.
 - Modify emptyDir configuration:

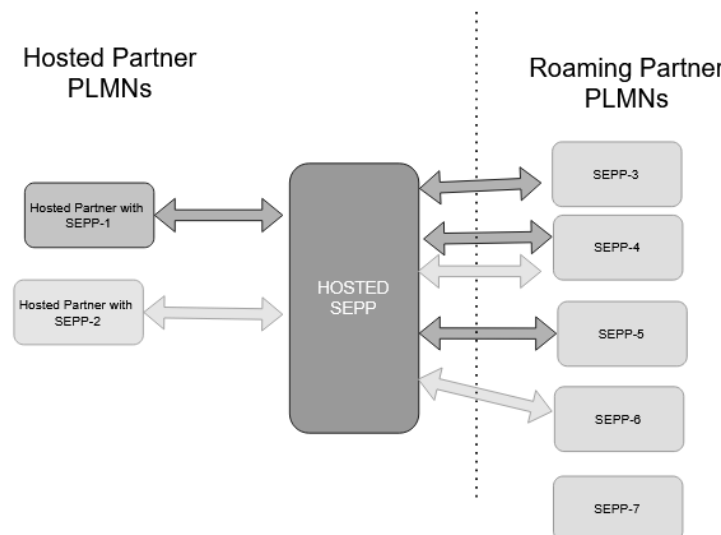
```
medium: Memory
sizeLimit: 10Mi # Change from 1Mi to 10Mi for 21 certificates
name: empty-dir
```

- If one hosted partner supports multiple roaming partners, increase remote-plmn-list size (default is 30).
- FQDN based interactions must use resolvable FQDNs. Dummy FQDNs are allowed only for communication between hosted partner and roaming partner.
- FQDN and IP combination for Remote SEPP Profile is not supported. Remote SEPP Profiles should use resolvable FQDNs in their configuration.
- PUT requests are used to modify roaming/hosted partner sets.
- SAN (Subject Alternate Name) in certificates should match the fqdn configuration. Mismatches prevent context establishment.
- Ensure that FQDN entry in multiFQDN section under FQDNDetails is present during fresh deployment or upgrade. The same is converted to DB entry and REST APIs can be used to add/modify/delete the multi-fqdn configurations.

Configuration and Connectivity for Hosted SEPP with Roaming Remote SEPP Sets

The following diagram depicts the connectivity between Hosted Partner Remote SEPP Sets and Roaming Partner Remote SEPP Sets:

Figure 3-4 Configuration and Connectivity for Hosted SEPP with Roaming Remote SEPP Sets



SEPP-1 can send SBI message to SEPP-3, SEPP-4 and SEPP-5
 SEPP-2 can send SBI message to SEPP-4 and SEPP-6
 SEPP-3 can send SBI message to Hosted SEPP-1
 SEPP-4 can send SBI message to Hosted SEPP-1 and Hosted SEPP-2
 SEPP-5 can send SBI message to Hosted SEPP-1
 SEPP-6 can send SBI message to Hosted SEPP-2.
 SEPP-7 cannot send and receive message to anyone yet.

Table 3-3 Connectivity Between Hosted Partner Remote SEPP Sets and Roaming Partner Remote SEPP Sets

Hosted Remote SEPP Set	Roaming Remote SEPP Set	Allowed (Yes/No)
Hosted SEPP Set 1	Roaming SEPP Set 3	YES
Hosted SEPP Set 1	Roaming SEPP Set 4	YES
Hosted SEPP Set 1	Roaming SEPP Set 5	YES
Hosted SEPP Set 1	Roaming SEPP Set 6	NO
Hosted SEPP Set 2	Roaming SEPP Set 3	NO
Hosted SEPP Set 2	Roaming SEPP Set 4	YES
Hosted SEPP Set 2	Roaming SEPP Set 5	NO
Hosted SEPP Set 2	Roaming SEPP Set 6	YES

A Hosted Partner can connect to multiple Roaming Remote SEPP Sets.

The Hosted SEPP may either be outside the partner's physical network or within the trust boundary of the partner.

You can configure the Remote SEPP Set according to the connectivity details outlined in the table.

Each Roaming Partner Remote SEPP must have an associated Hosted Partner Remote SEPP Set. The Roaming Partner's RPS should be configured with the "AllowedProducerRemoteSEPPSets" field, which lists the Hosted Remote SEPP Sets that are permitted by the Roaming Partner.

For instance, the SEPP-4 Remote SEPP Set should include (SEPP-1 and SEPP-2) in its list of allowed Remote SEPP Sets.

The handshake will originate from the Hosted SEPP towards the Roaming Partner, but only if a handshake has already been established between the Hosted SEPP and the Hosted Partner for that specific Roaming Partner. Each Hosted SEPP will have a unique handshake context with the Roaming Partner.

This setup enables the traffic to flow from the Roaming Partner to the Hosted Partner as well.

Configuration Limits for Hosted SEPP

The following table shows the configuration limits for Hosted SEPP:

Deployment	Max Remote PLMNs	Max PLMNs per Remote SEPP	Max Local PLMNs (Self PLMN) for Hosted SEPP
Hosted SEPP	900	900	900

Configuration Workflow

A new attribute, "AllowedProducerRemoteSeppSets," has been introduced in the Remote SEPP Sets configuration. This attribute determines whether the Hosted Remote SEPP Set can send traffic to the Roaming Remote SEPP Set. If the Hosted Partner's Remote SEPP Set is included in the Roaming Partner's RPS "AllowedProducerRemoteSEPPSets" list, then the Hosted SEPP can send traffic to the Roaming Partner's Remote SEPP Set, and vice versa.

Roaming Partner RPS

```
curl -v http://127.0.0.1:9090/sepp-configuration/v1/remotesepset -d
'{ "name": "csepp", "primary": "RPS4", "allowedListName": "Default",
"allowedProducerRemoteSeppSets" : ["HPS1","HPS2"] }' -H 'Content-
Type:application/json' -X POST
```

Managing Hosted SEPP Feature

Enable

This section describes the procedure to enable the feature.

You can enable the Supporting Multi-Tenancy in Hosted SEPP with Multiple FQDNs and Tenant-Specific Certificates feature using the following Helm Configuration:

The user can enable the Hosted SEPP using the following Helm Configuration:

To enable this feature, set the following Helm flags:

- `global.hostedSepp: true`
- `n32-ingress-gateway.multiFqdn.enabled: true`
- `n32-egress-gateway.multiFqdn.enabled: true`
- `n32cHandshakePlmnIdListSend: false`

Additionally, configure the following settings:

- `nrfClientSupport:false` and `seppCoherenceServiceEnabled: false` for Hosted SEPP deployments.
- `Values.global.localProfile.n32cHandshakePlmnIdListValidationEnabled:false` for Hosted SEPP deployments.

Configure

This section provides information about the configurations required to configure this feature.

For Hosted SEPP, ensure the following configurations:

- `nrfClientSupport` set to `false`
- `Values.global.localProfile.n32cHandshakePlmnIdListValidationEnabled` set to `false`.
- `n32-egress-gateway.globalRemoveRequestHeader: x-sni-value`
- `n32-ingress-gateway.globalRemoveRequestHeader: x-source-fqdn`

For non Hosted SEPP, ensure the following configurations:

- `n32-egress-gateway.globalRemoveRequestHeader: x-sni-value`
- `n32-ingress-gateway.globalRemoveRequestHeader: x-source-fqdn`
- `plmn-egress-gateway.globalRemoveRequestHeader:x-sni-value`

Multi-FQDN Configuration:

Enable multi-FQDN support by setting `multiFqdn.enabled: true` (in the N32 Ingress Gateway and N32 Egress Gateway) and providing the relevant FQDN details. Example configuration for FQDN details:

```
multiFqdn:
  enabled: true
  fqdnDetails:
    - fqdn: internal_fqdn.oracle.com
      caroot_file: bundle1.cer
      key_store_passwd_file: internal_fqdn.mcc111.mnc123.3gppnetwork_key.txt
      ocsepp_cert_file: internal_fqdn.mcc111.mnc123.3gppnetwork_ocsepp.cer
      rsa_certificate_file:
        internal_fqdn.mcc111.mnc123.3gppnetwork_rsa_certificate.crt
      rsa_private_key_file:
        internal_fqdn.mcc111.mnc123.3gppnetwork_rsa_private_key_pkcs1.pem
      ssl_ecdsa_certificate_file:
        internal_fqdn.mcc111.mnc123.3gppnetwork_ssl_ecdsa_certificate.crt
      ssl_ecdsa_private_key_file:
        internal_fqdn.mcc111.mnc123.3gppnetwork_ssl_ecdsa_private_key.pem
      trust_store_passwd_file:
        internal_fqdn.mcc111.mnc123.3gppnetwork_trust.txt
    - fqdn: external_fqdn1.mcc123.mnc90
      caroot_file: external_fqdn1.mcc123.mnc90.cer
      key_store_passwd_file: external_fqdn1.mcc123.mnc90_key.txt
      ocsepp_cert_file: external_fqdn1.mcc123.mnc90_ocsepp.cer
      rsa_certificate_file: external_fqdn1.mcc123.mnc90_rsa_certificate.crt
      rsa_private_key_file:
        external_fqdn1.mcc123.mnc90_rsa_private_key_pkcs1.pem
      ssl_ecdsa_certificate_file:
        external_fqdn1.mcc123.mnc90_ssl_ecdsa_certificate.crt
      ssl_ecdsa_private_key_file:
        external_fqdn1.mcc123.mnc90_ssl_ecdsa_private_key.pem
      trust_store_passwd_file: external_fqdn1.mcc123.mnc90_trust.txt
```

Note: The FQDN in the configuration must match the SAN value in the `ssl.conf` of the corresponding certificate.

Secret Creation for N32 Interface

For deployments with multi-FQDN flag enabled, create the required secret using the following command:

N32 Ingress Gateway

```
kubectl create secret generic ocsepp-n32-igw-secret
--from-file=ssl_ecdsa_private_key.pem --from-
file=rsa_private_key_pkcs1.pem --from-file=trust.txt --from-file=key.txt --
from-file=caroot.cer --from-file=rsa_certificate.crt --from-
file=ssl_ecdsa_certificate.crt --from-file=ocsepp.cer
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_ssl_ecdsa_private_key.pem
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_rsa_private_key_pkcs1.pem
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_trust.txt
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_key.txt
--from-file=bundle1.cer
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_rsa_certificate.crt
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_ssl_ecdsa_certificate.crt
```

```
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_ocsepp.cer
--from-file=external_fqdn1.mcc123.mnc90.cer -n <namespace>
```

N32 Egress Gateway

```
kubectl create secret generic ocsepp-n32-egw-secret
--from-file=ssl_ecdsa_private_key.pem --from-
file=rsa_private_key_pkcs1.pem --from-file=trust.txt --from-file=key.txt --
from-file=caroot.cer --from-file=rsa_certificate.crt --from-
file=ssl_ecdsa_certificate.crt --from-file=ocsepp.cer
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_ssl_ecdsa_private_key.pem
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_rsa_private_key_pkcs1.pem
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_trust.txt
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_key.txt
--from-file=bundle1.cer
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_rsa_certificate.crt
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_ssl_ecdsa_certificate.crt
--from-file=internal_fqdn.mcc111.mnc123.3gppnetwork_ocsepp.cer
--from-file=external_fqdn1.mcc123.mnc90.cer -n <namespace>
```

Handshake Establishment Process

Follow these steps for the handshake establishment between Hosted and Roaming Partners:

Note

The `plmnIDList` in the `/sepp-configuration/v1/remotesep` will contain PLMNs of the far-end peer, that is, Roaming /Hosted Partner PLMNs depending on the direction of the handshake (from hosted to roaming or vice versa).

1. Hosted Partner to Hosted SEPP: Post the remote SEPP profile of the hosted partner to the Hosted SEPP.

```
curl http://127.0.0.1:9090/sepp-configuration/v1/remotesep -d '{
  "name": "hosted-sepp",
  "plmnIdList": [{"mcc": "111", "mnc": "290"}, {"mcc": "111", "mnc": "291"}],
  #PLMN ID list of roaming partner SEPP
  "domain": "abc.svc",
  "seppFqdn": "hosted.sepp.1.fqdn",
  "securityCapabilityList": [ "TLS" ],
  "isEnabled": "TRUE",
  "apiPrefix": "",
  "apiVersion": "v1",
  "port": "443" }'
-H 'Content-Type:application/json' -X POST
```

2. Hosted SEPP to Roaming Partner: Post the remote SEPP profile of the roaming partner to the Hosted SEPP.

```
curl http://127.0.0.1:9090/sepp-configuration/v1/remotesep -d '{
  "name": "sepp4",
  "plmnIdList": [{"mcc": "444", "mnc": "290"}, {"mcc": "444", "mnc": "291"}],
```

```
"domain":"abc.svc",
"hostedPartner":"TRUE",
"seppFqdn":"hp1.fqdn",
"fqdnExposedTowardsPartnerProfile": "hosted.sepp.1.fqdn",
"partnerProfileFqdnTowardsRoamingPartner": "external.mcc444.mnc290.fqdn",
"securityCapabilityList": ["TLS"],
"isEnabled":"TRUE",
"apiPrefix":"",
"apiVersion":"v1",
"port": "443"
}' -H 'Content-Type:application/json' -X POST
```

3. Roaming Partner to Hosted SEPP: Post the remote SEPP profile of the Hosted SEPP to the roaming partner.

```
curl http://127.0.0.1:9090/sepp-configuration/v1/remotesep -d '
{ "name": "hosted-sepp",
  "plmnIdList": [{"mcc":"111","mnc":"290"}, {"mcc":"111","mnc":"291"}],
  #PLMN ID list of hosted partner SEPP
  "domain":"abc.svc",
  "seppFqdn":"roaming.partner.fqdn",
  "remoteSeppIpAddress": "1.2.3.4",
  "securityCapabilityList": [ "TLS" ],
  "isEnabled":"TRUE",
  "apiPrefix":"",
  "apiVersion":"v1",
  "port": "443" }'
-H 'Content-Type:application/json' -X POST
```

Profile posted of Hosted SEPP on Roaming Partner:

```
curl http://127.0.0.1:9090/sepp-configuration/v1/remotesep -d '
{ "name": "hosted-sepp",
  "plmnIdList": [{"mcc":"444","mnc":"290"}, {"mcc":"444","mnc":"291"}],
  #PLMN ID list of hosted partner SEPP
  "domain":"abc.svc",
  "seppFqdn":"external.mcc444.mnc290.fqdn",
  "securityCapabilityList": [ "TLS" ],
  "isEnabled":"TRUE",
  "apiPrefix":"",
  "apiVersion":"v1",
  "port": "443" }'
-H 'Content-Type:application/json' -X POST
```

4. Create Remote SEPP Sets:

- Create a remote SEPP set for hosted partners.

```
curl http://127.0.0.1:9090/sepp-configuration/v1/remotesepset?
hosted=true -XPOST -H 'Content-type: application/json' -d'
{"name":"HPS4","primary":"sepp4","serviceAPIAllowedListOptions":
{"serviceAPIAllowedListName":"Default"},"hostedRemotePartnerSet":"TRUE"}
'
```

```
curl http://127.0.0.1:9090/sepp-configuration/v1/remotesepppset -XPOST -
H 'Content-type: application/json' -d'
{"name":"RPS3","primary":"sepp8","allowedProducerRemoteSeppSets":
{"allowedProducerRemoteSeppSets":
["HPS4"]},"serviceAPIAllowedListOptions":
{"serviceAPIAllowedListName":"Default"},"hostedRemotePartnerSet":"FALSE"
}'
```

5. Add Multiple Hosted Partners to a Roaming Partner:

```
curl http://127.0.0.1:9090/sepp-configuration/v1/remotesepppset/RPS3 -XPUT -
H 'Content-type: application/json' -d'
{"name":"RPS3","primary":"sepp8","allowedProducerRemoteSeppSets":
{"allowedProducerRemoteSeppSets":
["HPS4","HPS2"]},"serviceAPIAllowedListOptions":
{"serviceAPIAllowedListName":"Default"},"hostedRemotePartnerSet":"FALSE"}
```

You can configure the feature using REST API and CNC Console:

- Configure using REST API: Perform the feature configurations as described in the Remote SEPP Set, Remote SEPP, and MultiFQDN Configurations sections in *Oracle Communications Cloud Native Core Security Edge Protection Proxy REST Specification Guide*.
- Configure using CNC Console: Perform the feature configurations as described Remote SEPP Set, Remote SEPP, and MultiFQDN Configurations sections in [Configuring SEPP using CNC Console](#).

Note

For upgrades from release 25.2.2xx to 26.1.2x, configuration is supported only through REST APIs in the target release. To preserve the configuration state from the earlier release, update the latest custom-values.yaml file before performing the upgrade:

- Configure all MultiFQDN settings that were configured in the earlier release.
- Copy the value of `ocsepp_cert_file` to `rsa_certificate_file`.

Configurations added through REST APIs after the upgrade are not preserved if the deployment is rolled back to the earlier release.

Observe

Following are the Hosted SEPP feature specific metrics:

- `ocsepp_cn32c_handshake_requests_hosted_sepp_total`
- `ocsepp_cn32f_sni_xfs_requests_hosted_sepp_total`
- `ocsepp_cn32f_sni_xfs_requests_mismatch_hosted_sepp_total`
- `ocsepp_pn32f_requests_hosted_sepp_total`

Following are the Hosted SEPP feature specific KPIs:

- CN32F Allowed P-RSS Validation Failure Count
- PN32F Allowed P-RSS Validation Failure Count

Following are the Hosted SEPP feature specific Alerts:

- SEPPn32fHSRoutingFailureAlert (Major, Critical, minor, warning)

For more information on Hosted SEPP feature related SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following steps:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.3 Support for Max-Response Timer Header

Overview

The feature enables the SEPP to validate, add, or modify timestamp headers in both ingress and egress request paths. Once enabled, it prevents delays and service degradation by enforcing a maximum allowable response time, ensuring that requests are processed within an acceptable time window. Any messages that exceed the defined response time threshold are automatically rejected, which enhances the overall efficiency and responsiveness of the network.

The feature works by embedding timestamps in the headers of messages sent from the 5G Core NFs to the SEPP. It compares the timestamp from the 5G Core NFs (T1) with the timestamp from the SEPP (T2). If the difference between these timestamps exceeds a predefined threshold, the message is considered expired and discarded.

Key functionalities include:

- **Timestamp-based Validation:** The 5G Core NFs sends messages with a timestamp and a maximum response time (typically 5 seconds). If the message is not processed within this window, it is discarded.
- **Timestamp Difference Rejection Logic:** If the difference between the 5G Core NFs timestamp (T1) and SEPP timestamp (T2) is negative or exceeds the 5-second threshold, the message is rejected to prevent unnecessary duplication or processing.
- **Flexible Timeout Settings:** The timeout is based on the minimum of two values: a fixed 5-second timeout from the 5G Core NFs and timestamp on SEPP. This flexibility allows the feature to adapt better to varying network conditions.
- **User Control Enablement:** By default, the feature is not enabled to prevent issues that could arise from clock skew across nodes, particularly in lab environment. It can be activated through Helm parameters as per user requirements.
- **Monitoring and Alerts:** The feature provides detailed metrics and customizable alert configurations, enabling network administrators to monitor its performance effectively and take corrective actions if necessary.

Background

The Max Response Header feature is introduced as part of the 5G core network to address issues related to message duplication and network inefficiencies, particularly in handling

roaming requests. The 3GPP specification suggests using timestamp headers to ensure that each message is processed only once. This feature enhances network reliability by preventing unnecessary message reprocessing and optimizing resource utilization. It also provides greater control over message timing, especially in environments with varying time synchronization conditions.

Header Format

Following is the SBI Timer Header format:

Table 3-4 SBI Timer Header Format

Header Name	Header Format
3gpp-Sbi-Sender-Timestamp	OWS day-name "," SP date1 SP time-of-day "." milliseconds SP GMT day-name, date1, time-of-day shall comply with the definition in clause 7.1.1.1 of IETF RFC 7231 [11]. day-name = %s"Mon" / %s"Tue" / %s"Wed" / %s"Thu" / %s"Fri" / %s"Sat" / %s"Sun" date1 = day SP month SP year ; e.g., 02 Jun 1982 day = 2DIGIT month = %s"Jan" / %s"Feb" / %s"Mar" / %s"Apr" / %s"May" / %s"Jun" / %s"Jul" / %s"Aug" / %s"Sep" / %s"Oct" / %s"Nov" / %s"Dec" year = 4DIGIT GMT = %s"GMT" time-of-day = hour ":" minute ":" second ; 00:00:00 - 23:59:60 (leap second) hour = 2DIGIT minute = 2DIGIT second = 2DIGIT milliseconds = 3DIGIT EXAMPLE: <i>3gpp-Sbi-Sender-Timestamp: Sun, 04 Aug 2019 08:49:37.845 GMT</i>

Detailed Description

This section details the functionality of the feature.

At Ingress Gateway

- When both 3gpp-Sbi-Sender-Timestamp and 3gpp-Sbi-Max-Rsp-Time are present:
If the calculated difference between the current time and the sender's timestamp (that is, max-rsp-header (CurrentTime - SenderTime)) is less than or equal to zero, the request will be rejected.
- When 3gpp-Sbi-Sender-Timestamp is absent but 3gpp-Sbi-Origination-Timestamp and 3gpp-Sbi-Max-Rsp-Time are present:
If the calculated difference between the current time and the origination time (i.e., max-rsp-header - (CurrentTime - OriginTime)) is less than or equal to zero, the request will be rejected.
- Final Request Timeout Calculation:
 - If the minimum of the calculated time difference (max-rsp-header - (CurrentTime - SenderTime)) and the configured request timeout (set at the route or global level) is

greater than or equal to zero, the final request timeout will be set to this minimum value.

- If the calculated value is less than zero, the request will be rejected.

4. Otherwise, allow the request for further processing by the backend.

At Egress Gateway

1. Final Request Timeout Calculation:

The final request timeout is calculated by subtracting the sender timestamp from the current time and comparing it to the maximum response time:

Max response time - (Current Time - Sender Timestamp).

2. If the calculated final request timeout is less than or equal to zero, the request will be rejected.

3. Otherwise, allow the request for backend processing.

At Backend

1. No Timestamp Check:

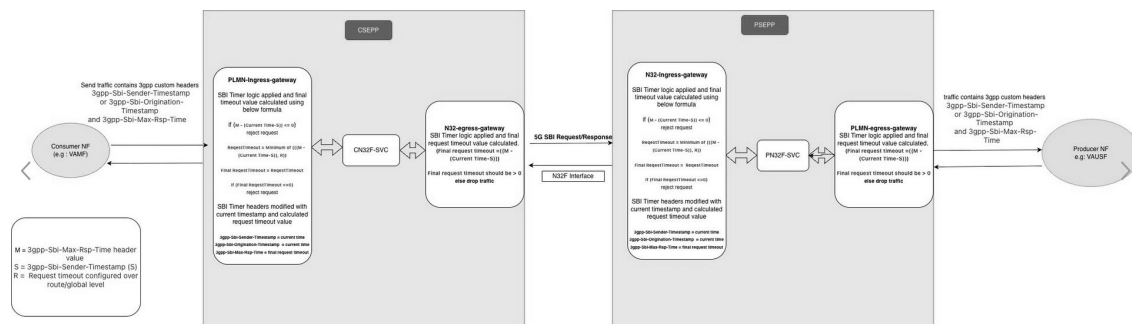
The backend processes the received request without performing any timestamp-related checks, as the validation and rejection occur earlier in the processing path.

Architecture

The following diagrams provide a high level view of Support for Max-Response Timer Header feature. Three scenarios are depicted here.

When consumer NF send request with 3GPP SBI Timer headers

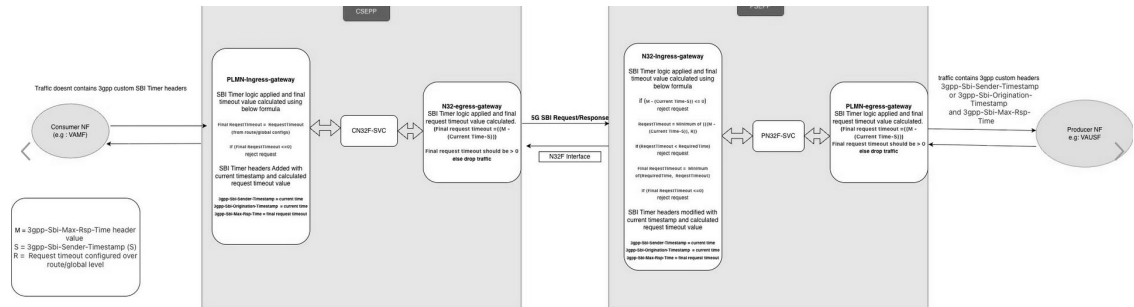
Figure 3-5 When consumer NF send request with 3GPP SBI Timer headers



- The image depicts the end-to-end flow of a 5G SBI request from a Consumer NF to a Producer NF through C-SEPP and P-SEPP, with the request containing 3GPP SBI Timer headers.
- At each Ingress and Egress Gateway, the SBI timer logic calculates the remaining valid request timeout.
- Requests are rejected early if the calculated timeout has already expired.
- Valid requests have their SBI headers updated with the current timestamp and the recalculated timeout.
- The updated requests are then forwarded to ensure only time-valid requests traverse the N32 interface and reach the producer.

When consumer NF send request without 3GPP SBI Timer headers

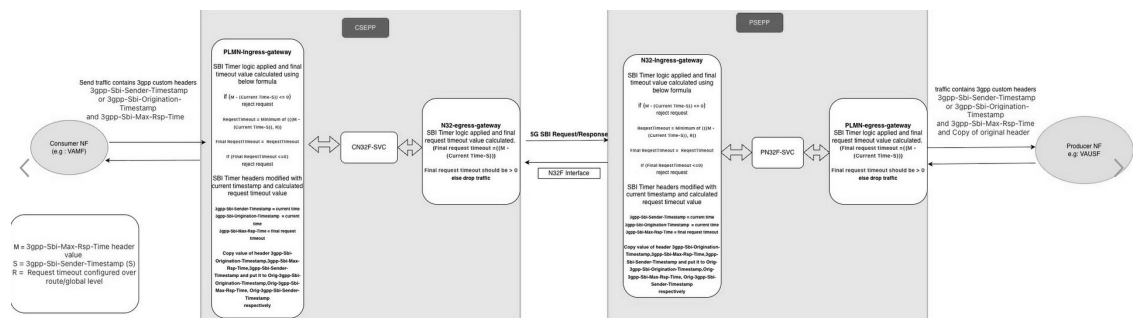
Figure 3-6 When consumer NF send request without 3GPP SBI Timer headers



- The image shows the flow when the Consumer NF sends a request without 3GPP SBI timer headers.
- At the CSEPP Ingress Gateway, a default request timeout (configured at the route or global level) is applied.
- CSEPP adds the SBI timer headers, including the current timestamp and the calculated timeout, before forwarding the request.
- The request then travels through the N32 interface to PSEPP, where the remaining timeout is recalculated and validated again.
- Only requests with a valid remaining timeout are forwarded to the Producer NF.
- Requests with expired timeouts are rejected early.

When consumer NF send request with 3GPP SBI Timer headers and publish header flag is set as true over PLMN Ingress Gateway

Figure 3-7 When consumer NF send request with 3GPP SBI Timer headers and publish header flag is set as true over PLMN Ingress Gateway



- The image shows the flow when the Consumer NF sends a request with 3GPP SBI timer headers and the publish header flag enabled on the PLMN Ingress Gateway.
- CSEPP validates the incoming timer headers, recalculates the remaining timeout, and updates the SBI headers with the current timestamp.

- A copy of the original SBI timer headers is preserved and forwarded along with the modified headers.
- At PSEPP, the request timeout is recalculated using the updated headers and validated before being forwarded.
- Only requests with a valid remaining timeout are forwarded to the Producer NF.

Managing the Feature

Enable

The user can enable the Support for Max-Response Timer Header feature using the Helm configurations.

Set `seppMaxRspTimerEnabled` to true in the global section of `ocsepp_custom_values_<version>.yaml` file

Configure

The following Helm parameters are added for Support for Max-Response Timer Header feature.

Table 3-5 Helm Configurations

Parameter	Description	Details
<code>global.seppMaxRspTimerEnabled</code>	This is an optional parameter. This global flag is used to enable or disable SBI Timer feature for calculating request timeout in OCSEPP for all gateway services	Data Type: boolean Default Value: false Range: NA
<code>global.seppSbiTimerTimezone</code>	This is an optional parameter. This flag is used to configure Timezone mode for SBI timer logic for all gateway services.	Data Type: String Default Value: GMT Range: GMT,ANY
<code>n32-ingress-gateway.isSbiTimerEnabled</code>	This is an optional parameter. This flag is used to enable or disable SBI Timer feature for calculating request timeout at n32-ingress-gateway	Data Type: boolean Default Value: false Range: NA
<code>n32-egress-gateway.isSbiTimerEnabled</code>	This is an optional parameter. This flag is used to enable or disable SBI Timer feature for calculating request timeout at n32-egress-gateway	Data Type: boolean Default Value: false Range: NA
<code>plmn-ingress-gateway.isSbiTimerEnabled</code>	This is an optional parameter. This flag is used to enable or disable SBI Timer feature for calculating request timeout at plmn-ingress-gateway.	Data Type: boolean Default Value: false Range: NA
<code>plmn-egress-gateway.isSbiTimerEnabled</code>	This is an optional parameter. This flag is used to enable or disable SBI Timer feature for calculating request timeout at plmn-egress-gateway.	Data Type: boolean Default Value: false Range: NA

Table 3-5 (Cont.) Helm Configurations

Parameter	Description	Details
n32-ingress-gateway.sbiTimerTimezone	This is an optional parameter. This flag is used to configure Timezone mode for SBI timer logic at n32-ingress-gateway.	Data Type: String Default Value: GMT Range: GMT,ANY
n32-egress-gateway.sbiTimerTimezone	This is an optional parameter. This flag is used to configure Timezone mode for SBI timer logic at n32-egress-gateway.	Data Type: String Default Value: GMT Range: GMT,ANY
plmn-ingress-gateway.sbiTimerTimezone	This is an optional parameter. This flag is used to configure Timezone mode for SBI timer logic at plmn-ingress-gateway.	Data Type: String Default Value: GMT Range: GMT,ANY
plmn-egress-gateway.sbiTimerTimezone	This is an optional parameter. This flag is used to configure Timezone mode for SBI timer logic at plmn-egress-gateway.	Data Type: String Default Value: GMT Range: GMT,ANY
plmn-ingress-gateway.publishHeaders	This is an optional parameter. publishHeaders flag decides if Orig-3gpp-Sbi-Sender-Timestamp , Orig-3gpp-Sbi-Max-Rsp-Time and Orig-3gpp-Sbi-Origination-Timestamp will be published and sent to backend service.	Data Type: boolean Default Value: false Range: NA
n32-ingress-gateway.publishHeaders	This is an optional parameter. publishHeaders flag decides if Orig-3gpp-Sbi-Sender-Timestamp , Orig-3gpp-Sbi-Max-Rsp-Time and Orig-3gpp-Sbi-Origination-Timestamp will be published and sent to backend service.	Data Type: boolean Default Value: false Range: NA

The configurations at the Ingress and Egress Gateways

This section provides information about the configurations required to configure this feature.

Helm Configurations

n32-ingress-gateway

```
isSbiTimerEnabled: *seppMaxRspTimerEnable
publishHeaders: false
sbiTimerTimezone: *seppSbiTimerTimezone
```

```
routesConfig:
- id: n32f
  uri: http://{ .Release.Name }-pn32f-svc:9090/
  path: /**
  order: 3
```

```

#Below field is used to provide an option to enable/disable route level
xfccHeaderValidation, it will override global configuration for
xfccHeaderValidation.enabled
metadata:
  requestTimeout: 1200
  xfccHeaderValidation:
    validationEnabled: false
  igwAtsSpecificRequestTimeout: *igwAtsSpecificRequestTimeoutRef
  igwAtsSpecificDeployment: *atsSpecificDeploymentRef
  svcName:

```

plmn-ingress-gateway

```

isSbiTimerEnabled: *seppMaxRspTimerEnable
publishHeaders: false
sbiTimerTimezone: *seppSbiTimerTimezone

```

```

routesConfig:
- id: cn32f
  uri: http://{ .Release.Name }}-cn32f-svc:9090/
  path: /**
  order: 2
  #Below field is used to provide an option to enable/disable route level
  xfccHeaderValidation, it will override global configuration for
  xfccHeaderValidation.enabled
  metadata:
    requestTimeout: 2600
    xfccHeaderValidation:
      validationEnabled: false
    igwAtsSpecificRequestTimeout: *igwAtsSpecificRequestTimeoutRef
    igwAtsSpecificDeployment: *atsSpecificDeploymentRef
    svcName:

```

n32-egress-gateway

```

isSbiTimerEnabled: *MaxRspTimerEnable
sbiTimerTimezone: *sbiTimerTimezone

```

plmn-egress-gateway

```

isSbiTimerEnabled: *MaxRspTimerEnable
sbiTimerTimezone: *sbiTimerTimezone

```

To enable this feature on the PLMN egress gateway, set the ignoreMaxRspTimeHeader metadata to "false" in the routeConfiguration section. If this flag is set to "true," the 3gpp-Max-Rsp-Time header will be ignored during the timeout calculation, and the feature will not be

applied. If not specified, the flag defaults to "true," meaning the 3gpp-Max-Rsp-Time header will be ignored unless explicitly set to "false." A sample route configuration is provided below.

```
routesConfig:
- id: scp_via_proxy
  uri: http://request.uri
  path: /**
  order: 1
  metadata:
    httpsTargetOnly: false    #In nrfClientDisable/ RH mode ,this flag should
    be true
    httpRuriOnly: false       #In nrfClientDisable/ RH mode ,this flag should
    be true
    sbiRoutingEnabled: false  #In nrfClientDisable/ RH mode ,this flag should
    be true
    ignoreMaxRspTimeHeader: false
```

When configuring specific routes over the PLMN egress gateway using the REST API, the user must ensure that the metadata is set to "false" for proper functionality of this feature. For more information on route configuration at PLMN Egress Gateway, refer to 'PLMN Egress Gateway' section of *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Observe

Metrics

The following metrics are available for this feature:

- oc_ingressgateway_request_failure_total
- oc_egressgateway_request_failure_total
- oc_egressgateway_sbitimer_timezone_mismatch
- oc_ingressgateway_sbitimer_timezone_mismatch

KPIs

There are no new updates to KPIs for this feature.

Alerts

The following alerts are available for this feature:

- ocseppMaxRespVldnFailedWarn
- ocseppMaxRespVldnFailedMinor
- ocseppMaxRespVldnFailedMajor
- ocseppMaxRespVldnFailedCritical

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.4 Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers

This feature enhances validation and error handling for OAuth Token Request (GET_Access_Token) messages processed by the SEPP. The primary goal is to ensure that only valid, authenticated messages with correct PLMN IDs are allowed into the local PLMN network, especially during inter-PLMN operations such as roaming.

Assumptions

The following are the assumptions for implementing the feature:

- This feature is only available in PSEPP and Hosted SEPP modes.
- The Cat-2 Network ID Validation feature must be enabled as a prerequisite for this functionality.
- All incoming Access Token requests must include the requesterPLMN/[List] and targetPLMN fields, as these are mandatory parameters in Inter-PLMN requests.
- The remote network must support the GET_ACCESS_TOKEN procedure as outlined in the 3GPP 29.510 specification.
- If the message is not received, the GET_ACCESS_TOKEN procedure will not be initiated on behalf of the remote network at the SEPP.

Core Functionality

- **PLMN Validation on Token Requests:**
The SEPP extracts the requester identity from the requesterPLMN or requesterPLMNList attribute in incoming OAuth GET_Access_Token messages and validates those against:
 - The PLMN-IDs negotiated during the n32c handshake, using strict or subset matching for requesterPLMN/requesterPLMNList.
 - The targetPLMN, ensuring it matches the local PLMN-IDs.
 - The request's content type, which must be application/x-www-form-urlencoded.
- **Mandatory Attribute Enforcement:**
The SEPP ensures all required OAuth parameters (including one among requesterPLMN or requesterPLMNList and targetPLMN) are present. Missing or invalid parameters trigger a configured error response.
- **Forwarding or Rejecting Requests:**
If validations pass, the SEPP forwards the request to the appropriate local network.
The response will have the content type application/json.
If validations fail, the SEPP returns an error response and logs the event.
- **KPI Measurement and Logging:**
Key metrics such as the total number of GET_Access_Token requests, requests missing PLMN information, and error responses are captured. Logs record the original and

validated values for targetPLMN, requesterPLMN, requesterPLMNList, and the actions taken (forwarded or rejected).

These logs adhere to existing Cat- 2 practices, ensuring compatibility during upgrades and rollbacks.

- **Alerting Based on Error Rate:**

The SEPP can trigger alerts based on error rate thresholds (Warn, Minor, Major, Critical), allowing operators to proactively address potential security or configuration issues. Alert levels adjust dynamically based on the error rate within the system.

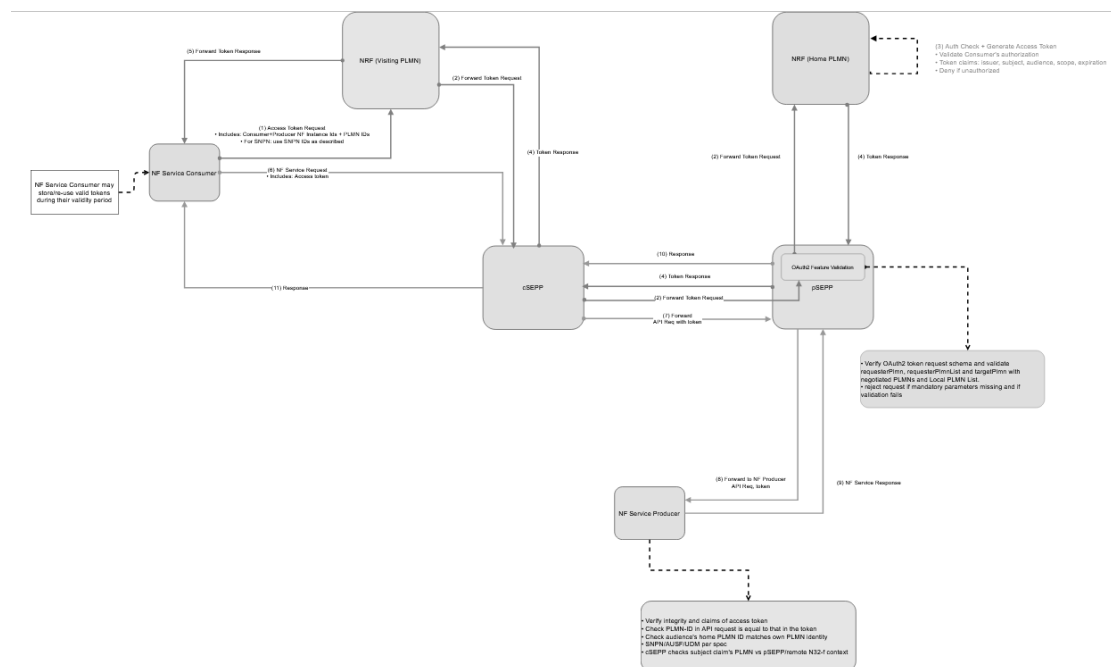
Activation and Compliance

The feature uses existing Cat-2 controls for activation and does not require a separate enable/disable switch. All validation and error-handling mechanisms comply with security and operational standards, supporting *application/x-www-form-urlencoded* for requests and *application/json* for responses.

Architecture

The following diagram provides a high level view of Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers feature.

Figure 3-8 Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers Architecture



The diagram illustrates a multi-step authentication process used by P-SEPP within a telecommunications network, specifically in the context of 5G roaming. The process involves several key components working together to ensure secure authorization for network resources and services between different PLMNs.

This process uses OAuth2 for token-based authentication and SEPP security features to safeguard communication. The goal is to ensure that only legitimate, authorized network functions can interact across PLMNs, providing a secure roaming authorization mechanism.

Key Components:

- **NF Service Consumer:** Initiates requests for network resources or services.
- **NRF (Visiting PLMN):** Network Repository Function in the visited PLMN that facilitates resource discovery and authorization.
- **cSEPP:** Security Edge Protection Proxy in the visited network, securing the connection between PLMNs.
- **NRF (Home PLMN):** Network Repository Function in the user's home PLMN that handles access permissions.
- **pSEPP:** Security Edge Protection Proxy in the home network, ensuring security and integrity of the request.
- **OAuth2 Feature Validation:** Logic for validating OAuth2 tokens.
- **NF Service Producer:** Provides the requested network resources or services.

Authentication Flow:

1. **Token Request:** The NF Service Consumer initiates a request for an access token through the NRF in the Visiting PLMN, including context information such as consumer/producer ID and PLMN IDs.
2. **Forwarding the Request:** The request is passed from the Visiting PLMN's NRF to the cSEPP, then on to the pSEPP in the Home PLMN for further processing.
3. **Token Validation at pSEPP:** The pSEPP in the home network validates the incoming token, checking its integrity, claims, PLMN identity, and other relevant parameters.
4. **Authorization Check:** The Home PLMN's NRF verifies the authorization of the requestor. If the requestor is authorized, an access token is issued; if not, access is denied.
5. **Token Return:** The access token is sent back to the NF Service Consumer through the reverse path, including pSEPP, cSEPP, and visiting PLMN's NRF.
6. **Service Request:** The NF Service Consumer sends an API request to the relevant NF Service Producer, using the access token and passing through the SEPP proxies.
7. **Service Response:** The requested network resource or service is returned to the NF Service Consumer following the reverse path.

Managing the Feature

Enable

You can enable the Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers feature using the following Helm Configuration:

Set the `oauthRequestValidation` parameter to true in the Pn32f section of the `values.yaml` file for the deployment.

For more information on the Helm parameters, refer to the *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Observe

Metrics

The following metrics are available for this feature:

- `ocsepp_oauth_total_requests`

KPIs

The following KPIs are available for this feature:

- Total number of Get_ACCESS_TOKEN messages received from a particular SEPP
- Number of requests without the requesterPLMN/requesterPLMNList attribute received from a particular SEPP
- Number of requests to which SEPP responded with an error towards a particular SEPP

Alerts

The following alerts are available for this feature:

- SEPPN32fNetworkID0authValidationFailureWarn
- SEPPN32fNetworkID0authValidationFailureMinor
- SEPPN32fNetworkID0authValidationFailureMajor
- SEPPN32fNetworkID0authValidationFailureCritical

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.5 Error Response and Logging Enhancement for SEPP

This feature includes both error enhancements and logging improvements.

Error Enhancements

The error response previously included only a generic error description in the details field, which was insufficient for troubleshooting. To provide more clarity, the error response now includes additional fields.

In the enhanced error response mechanism, SEPP now includes additional information in the details, such as the server FQDN, NF service name, vendor name, and error ID. These details help to find the specific reason of the error in the error responses.

The new format of the payload is as follows:

```
{
  "title":"[errorcodeProfiles[i].errorTitle]",
  "status": "[errorcodeProfiles[i].status]",
  "details":"<Server-FQDN>: <NF Service>: <Application specific problem
detail>: <VendorNF-App-Error-Id>",
  "cause":"[errorcodeProfiles[i].errorCause]"
}
```

Example:

```
{
  "title": "Resource Not Found",
  "status": 404,
  "detail": "sepp2.inter.oracle.com:cn32f-svc:SEPP function could not locate the
  required N32fContext or Remote SEPP configuration during the handling of an
  inter-PLMN signaling message:OSEPP-CN32F-CTX_ERR-E108",
  "cause": "N32F_CONTEXT_NOT_FOUND"
}
```

Logging Enhancements

The log messages enhancement feature adds the following additional attributes to the existing "ERROR" logs with appropriate values during the failure scenarios:

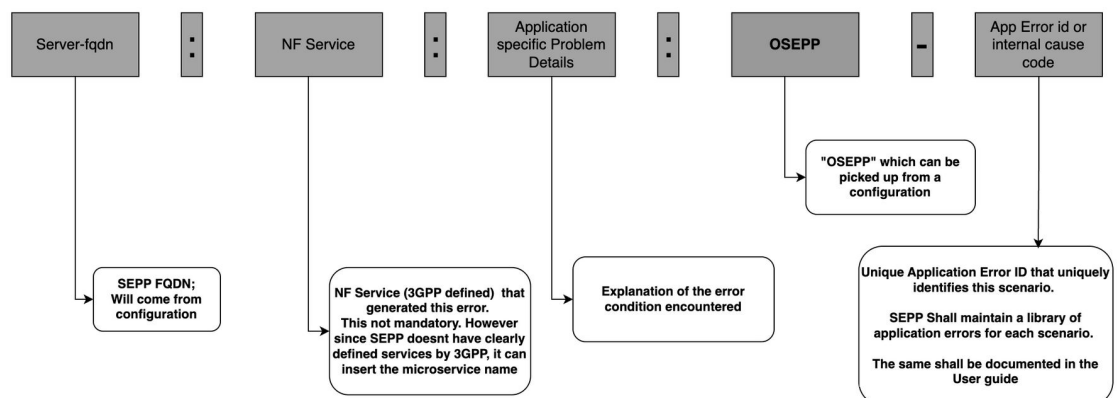
```
{
  "errorStatus": "404",
  "errorTitle": "Resource Not Found",
  "errorDetails": "ocsepp-plmn-ingress-gateway.vikashsepp1:cn32f-svc:SEPP
  function could not locate the required N32fContext or Remote SEPP
  configuration during the handling of an inter-PLMN signaling message:OSEPP-
  CN32F-CTX_ERR-E108",
  "errorCause": "N32F_CONTEXT_NOT_FOUND",
  "sender": "<namespace>"
}
```

Architecture

This feature enhances SEPP's error logging by adding additional information to the existing "ERROR" log messages, making it easier to identify the cause of issues and reduce troubleshooting time. The extra details are sourced from the ProblemDetails attribute, which SEPP generates or receives.

The following diagram represents the detailed format of the errors:

Figure 3-9 Error Format



SEPP sends error responses to consumer NFs when internal exceptions occur, such as connection failures, timeouts, or unknown hosts, while processing message requests from consumer NFs. These error responses include a payload with attributes like problem title, status, details, and the cause of the error, which help in investigating the issue. Previously, the error responses contained only a generic error description in the details field, making troubleshooting difficult. With the enhanced error response mechanism, SEPP now includes additional information, such as server FQDN, NF service name, vendor name, and error ID, in the details field. This improvement simplifies issue diagnosis by providing more specific information about the root cause, ultimately reducing troubleshooting time for users.

The following table describes the parameters of the details field of the payload:

Table 3-6 Parameters of the Details Field of the Payload

Parameter Name	Description
Server - FQDN	Indicates the SEPP FQDN, which is retrieved from the <code>nfFqdn</code> Helm chart parameter. Sample Value: <code>sepp2.inter.oracle.com</code>
NF-Service	Indicates the microservice name. It is the originator of the error response. This value is static and cannot be configured. Sample Value: • <code>cn32c-svc</code> • <code>pn32c-svc</code> • <code>cn32f-svc</code> • <code>pn32f-svc</code>
Application specific problem detail	Provides a short description of the error. Sample Value: <code>SEPP function could not locate the required N32fContext or Remote SEPP configuration during the handling of an inter-PLMN signaling message</code>
VendorNF	Indicates the vendor or NF type. This parameter is prefixed with "O", which indicates Oracle. For example, if the NF type is SEPP, the vendor name becomes OSEPP. It is obtained from the <code>EnhancedLoggingNFType</code> Helm Chart parameter. Sample Value: <code>OSEPP</code>

Table 3-6 (Cont.) Parameters of the Details Field of the Payload

Parameter Name	Description
App-Error-Id	Includes the microservice ID, category, and error ID. This value cannot be configured. For example, CN32F is the microservice ID name, UA is the category denoting unauthorised and E008 is the error code at SEPP. Sample Value: CN32F-CTX_ERR-E108 indicates the microservice ID and the error ID. The format is: App-Error ID is <nftype>-<serviceId>-<category>-E<XXX> Sample Value: OSEPP-CN32F-CTX_ERR-E108, where, • OSEPP is the vendor NF • CN32F is the microservice ID • CTX_ERR is the category • E108 is the app error code The details of App-Error ID format is as follows: nftype: Indicates the vendor or NF type. This parameter is prefixed with "O", which indicates Oracle. For example, if the NF type is SEPP, the vendor name becomes OSEPP. It is obtained from the nfType Helm Chart parameter. Sample Value: OSEPP • serviceId: It can be any of microservices supporting this feature. This value is set based on container name. • Category: Category to be used is fetched from error catalog. Errors are classified into categories based on Microservice id. • E<XXX>: It is the app error code id.
Microservice ID	Indicates the SEPP microservice IDs. The SEPP microservice IDs are denoted as follows: • CN32C for "cn32c-svc" microservice • PN32C for "pn32c-svc" microservice • CN32F for "cn32f-svc" microservice • PN32F for "pn32f-svc" microservice

Table 3-6 (Cont.) Parameters of the Details Field of the Payload

Parameter Name	Description																																																						
Category	Indicates the category under which the error comes up. This is an optional attribute and it does not require any configuration. Following are the list of categories: Table 3-7 Categories <table><tr><th>Category Code</th><th>Category Description</th><th>Sample Code</th></tr><tr><td>DB_ERR</td><td>Database Related Errors</td><td>CN32C-DB_ERR-E001</td></tr><tr><td>CFGN_ERR</td><td>Configuration Related Errors</td><td>CN32F-CFGN_ERR-E221</td></tr><tr><td>HS_ERR</td><td>Handshake flow Related Errors</td><td>PN32C-HS_ERR-E063</td></tr><tr><td>TH_ERR</td><td>Topology Hiding Errors</td><td>PN32F-TH_ERR-E163</td></tr><tr><td>TUH_ERR</td><td>Topology Unhiding Errors</td><td>PN32F-TUH_ERR-E166</td></tr><tr><td>PROTO_ERR</td><td>Protocol Specific Errors</td><td>CN32F-PROTO_ERR-E102</td></tr><tr><td>AUTHORIZATIO N_ERR</td><td>Authorization Errors</td><td>PN32F-AUTHORIZATIO N_ERR-E123</td></tr><tr><td>REQVLD_ERR</td><td>Request Validation Failed Error</td><td>CN32F-REQVLD_ERR-E147</td></tr><tr><td>RRMUT_ERR</td><td>Request Response Mutation Errors</td><td>PN32F-RRMUT_ERR-E161</td></tr><tr><td>MED_ERR</td><td>Mediation Errors</td><td>CN32F-MED_ERR-E182</td></tr><tr><td>TIMEOUT_ERR</td><td>Timeout Errors</td><td>PN32F-TIMEOUT_ERR-E242</td></tr><tr><td>REDIR_ERR</td><td>Redirection Errors</td><td>PN32F-REDIR_ERR-244</td></tr><tr><td>COMP_ERR</td><td>Compression Related Errors</td><td>CN32F-COMP_ERR-E205</td></tr><tr><td>DECODE_ERR</td><td>Decoding Errors</td><td>PN32F-DECODE_ERR-E204</td></tr><tr><td>SECNEG_ERR</td><td>Security Negotiation Errors</td><td>CN32F-SECNEG_ERR-E105</td></tr><tr><td>CTX_ERR</td><td>Context Specific Errors</td><td>CN32F-CTX_ERR-E108</td></tr><tr><td>APP_ERR</td><td>Application Errors</td><td>CN32F-APP_ERR-E111</td></tr></table>	Category Code	Category Description	Sample Code	DB_ERR	Database Related Errors	CN32C-DB_ERR-E001	CFGN_ERR	Configuration Related Errors	CN32F-CFGN_ERR-E221	HS_ERR	Handshake flow Related Errors	PN32C-HS_ERR-E063	TH_ERR	Topology Hiding Errors	PN32F-TH_ERR-E163	TUH_ERR	Topology Unhiding Errors	PN32F-TUH_ERR-E166	PROTO_ERR	Protocol Specific Errors	CN32F-PROTO_ERR-E102	AUTHORIZATIO N_ERR	Authorization Errors	PN32F-AUTHORIZATIO N_ERR-E123	REQVLD_ERR	Request Validation Failed Error	CN32F-REQVLD_ERR-E147	RRMUT_ERR	Request Response Mutation Errors	PN32F-RRMUT_ERR-E161	MED_ERR	Mediation Errors	CN32F-MED_ERR-E182	TIMEOUT_ERR	Timeout Errors	PN32F-TIMEOUT_ERR-E242	REDIR_ERR	Redirection Errors	PN32F-REDIR_ERR-244	COMP_ERR	Compression Related Errors	CN32F-COMP_ERR-E205	DECODE_ERR	Decoding Errors	PN32F-DECODE_ERR-E204	SECNEG_ERR	Security Negotiation Errors	CN32F-SECNEG_ERR-E105	CTX_ERR	Context Specific Errors	CN32F-CTX_ERR-E108	APP_ERR	Application Errors	CN32F-APP_ERR-E111
Category Code	Category Description	Sample Code																																																					
DB_ERR	Database Related Errors	CN32C-DB_ERR-E001																																																					
CFGN_ERR	Configuration Related Errors	CN32F-CFGN_ERR-E221																																																					
HS_ERR	Handshake flow Related Errors	PN32C-HS_ERR-E063																																																					
TH_ERR	Topology Hiding Errors	PN32F-TH_ERR-E163																																																					
TUH_ERR	Topology Unhiding Errors	PN32F-TUH_ERR-E166																																																					
PROTO_ERR	Protocol Specific Errors	CN32F-PROTO_ERR-E102																																																					
AUTHORIZATIO N_ERR	Authorization Errors	PN32F-AUTHORIZATIO N_ERR-E123																																																					
REQVLD_ERR	Request Validation Failed Error	CN32F-REQVLD_ERR-E147																																																					
RRMUT_ERR	Request Response Mutation Errors	PN32F-RRMUT_ERR-E161																																																					
MED_ERR	Mediation Errors	CN32F-MED_ERR-E182																																																					
TIMEOUT_ERR	Timeout Errors	PN32F-TIMEOUT_ERR-E242																																																					
REDIR_ERR	Redirection Errors	PN32F-REDIR_ERR-244																																																					
COMP_ERR	Compression Related Errors	CN32F-COMP_ERR-E205																																																					
DECODE_ERR	Decoding Errors	PN32F-DECODE_ERR-E204																																																					
SECNEG_ERR	Security Negotiation Errors	CN32F-SECNEG_ERR-E105																																																					
CTX_ERR	Context Specific Errors	CN32F-CTX_ERR-E108																																																					
APP_ERR	Application Errors	CN32F-APP_ERR-E111																																																					

Table 3-6 Parameters of the Details Field of the Payload

Parameter Name	Description									
	Table 3-7 Categories <table><tr><th>Category Code</th><th>Category Description</th><th>Sample Code</th></tr><tr><td>AUTH_ERR</td><td>Authentication Errors</td><td>PN32F-AUTH_ERR-E152</td></tr><tr><td>VLDN_ERR</td><td>Validation Errors</td><td>PN32F-VLDN_ERR-E113</td></tr></table>	Category Code	Category Description	Sample Code	AUTH_ERR	Authentication Errors	PN32F-AUTH_ERR-E152	VLDN_ERR	Validation Errors	PN32F-VLDN_ERR-E113
Category Code	Category Description	Sample Code								
AUTH_ERR	Authentication Errors	PN32F-AUTH_ERR-E152								
VLDN_ERR	Validation Errors	PN32F-VLDN_ERR-E113								
Error Code	Indicates the error code in the Exxx format. Sample Value: E101									

Logging Enhancements

The Error Log Messages Enhancement feature adds the following additional attributes to the existing "ERROR" logs with appropriate values during the failure scenarios:

Table 3-8 Log Attributes

Attribute	Attribute Description	Sample Value
errorStatus	Status code of the error	404, 500, etc.
errorTitle	Title of the error	Resource Not Found
errorDetails	Error detail produced by error handling framework in case of SEPP as producer. For errorDetails field to be populated in the required format, additional error handling framework should be enabled for the required service.	sepp2.inter.oracle.com:cn32f-svc:SEPP function could not locate the required N32fContext or Remote SEPP configuration during the handling of an inter-PLMN signaling message: OSEPP-CN32F-CTX_ERR-E108.
errorCause	Cause of the error	N32F_CONTEXT_NOT_FOUND
sender	Will be composed by the nfType(SEPP) plus the SEPP-FQDN	SEPP-sepp2.inter.oracle.com
receiver	Will be composed by the nfType(SEPP) plus the SEPP-FQDN	SEPP-sepp2.inter.oracle.com

Note

Among these, the errorStatus, errorTitle, errorDetails, and errorCause attributes are retrieved from the existing ProblemDetails attribute, while the sender and receiver details are fetched from the SEPP-FQDN.

Sample enhanced log details for the SEPP Subscription microservice:

```
{
  "_index": "logstash-2025.09.22", "_id":
  "RtdBcZkBnhkLPNhC6Iz0", "_version":
  1,
  "_score": null, "_source":
  {
    "stream": "stdout", "docker":
    {
      "container_id":
      "f035efb05505458fbaf2be9564884251dbc34e7633dfc6bf19cce0792bf5ceae"    },
    "kubernetes":
    {
      "container_name": "cn32f-svc", "namespace_name":
      "vikashsepp1", "pod_name":
      "ocsepp-release-vikash-cn32f-svc-8499d595-tbjld",
      "container_image":
      "occne-repo-host:5000/ocsepp-cn32f-svc:0.9.0-2",
      "container_image_id":
      "occne-repo-host:5000/ocsepp-cn32f-
      svc@sha256:7c4350fe349656d709602c83c855650bcdcbcefa78eff4470bea9b812d8d37f73",
      "pod_id":
      "28c1fd57-7f27-4519-9cd5-3762961a9fd8", "pod_ip":
      "10.233.76.80", "host":
      "occne-25-1-200-cluster-dg-k8s-node-4"    }, "instant":
    {
      "epochSecond": 1758541760, "nanoOfSecond":
      684000000    }, "thread": "default-nioEventLoopGroup-1-3",
    "level":
    "ERROR", "loggerName":
    "com.oracle.cgbu.cne.ocsepp.cn32f.handler.Cn32fSeppHandler",
    "endOfBatch":
    false,
    "loggerFqn": "org.apache.logging.log4j.spi.AbstractLogger",
    "threadId":
    19,
    "threadPriority": 5, "instanceType":
    "prod", "vendor": "Oracle", "messageTimestamp":
    "2025-09-22T11:49:20.684+0000", "processId":
    "${sys:PID}", "ocLogId":
    "1758541761456_80_ocsepp-release-vikash-plmn-ingress-
    gateway-64df94bc6-r92nc",
    "errorStatus": "404",
    "errorTitle": "Resource Not Found",
    "errorDetails": "'ocsepp-plmn-ingress-gateway.vikashsepp1:cn32f-
    svc:SEPP function
    could not locate the required N32fContext or Remote SEPP
    configuration during the
    handling of an inter-PLMN signaling
    message:OSEPP-CN32F-CTX_ERR-E108",
    "errorCause": "N32F_CONTEXT_NOT_FOUND",
    "sender": "ocsepp-plmn-ingress-gateway.vikashsepp1",
    "message": "N32fContext
    or Remote SEPP Set Not Found, problemDetails: {title: Resource Not
```



```

Found, status: 404,
    detail: ocsepp-plmn-ingress-gateway.vikashsepp1:cn32f-
svc:N32fContext or Remote SEPP Set
    Not Found:0SEPP:CN32F-UA-E008, cause: N32F_CONTEXT_NOT_FOUND}",
"@timestamp":
    "2025-09-22T11:49:20.706315512+00:00" }, "fields":
    {
        "@timestamp": [
            "2025-09-22T11:49:20.706Z" ], "messageTimestamp":
        [
            "2025-09-22T11:49:20.684Z" ] }, "highlight":
    {
        "level": [
            "@opensearch-dashboards-highlighted-field@ERROR@/opensearch-
dashboards-highlighted-field@" ],
        "kubernetes.container_name": [
            "@opensearch-dashboards-highlighted-field@cn32f@/opensearch-
dashboards-highlighted-field@-@opensearch-dashboards-highlighted-field@svc@/
opensearch-dashboards-highlighted-field@" ],
        "kubernetes.namespace_name": [
            "@opensearch-dashboards-highlighted-field@vikashsepp1@/opensearch-
dashboards-highlighted-field@" ] }, "sort":
    [
        1758541760706 ]]

```

In the example above, the following details have been added to enhance the existing error logs:

```

{
    "errorStatus": "404",
    "errorTitle": "Resource Not Found",
    "errorDetails": "ocsepp-plmn-ingress-gateway.vikashsepp1:cn32f-svc:SEPP
function could not locate the required N32fContext or Remote SEPP
configuration during the handling of an inter-PLMN signaling message:0SEPP-
CN32F-CTX_ERR-E108",
    "errorCause": "N32F_CONTEXT_NOT_FOUND",
    "sender": "<namespace>"
}

```

Note

For the detailed description about the enhanced Error codes, refer [Error Code Catalog Appendix](#).

Managing the Feature

Enable

This section describes the procedure to enable the feature.

You can enable the Error Response and Logging Enhancement for SEPP feature using the CNC Console or REST API:

Enable using REST API

Set `additionalErrorLogging` parameter to `ENABLED` in the logging section for each microservices.

```
# Logging level
logging:
  additionalErrorLogging: "DISABLED"
  level:
  root: "WARN"
```

Set `additionalErrorLogging` parameter to `ENABLED` in `sepp-configuration/sepp/nf-common-component/v1/pn32c/logging REST API`.

```
{
  "appLogLevel": "WARN",
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "WARN"
    }
  ],
  "additionalErrorLogging": "ENABLED"
}
```

Enable using CNC Console

1. From the left navigation menu, navigate to **SEPP** and then click **Logging Level Options**. The **Logging Level Options** page is displayed.
2. Click **Edit** icon to modify the Logging Level Options. The **Logging Level Configurations** page appears.
3. Set the **Additional Error Logging** to True.

Configure

You can configure the feature using REST API and CNC Console.

- **Configure using REST API:** Perform the feature configurations as described in Logging REST APIs in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- **Configure using CNC Console:** Perform the feature configurations as described in the 'Logging' section in [Configuring SEPP using CNC Console](#).

Observe

There are no specific metrics and alerts required for this feature.

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.6 Lifecycle Management (LCM) Based Automation

Lifecycle Management (LCM) Based Automation

In SEPP 25.2.100 and earlier releases, there was an option to provide a value for ServiceAccountName or leave it blank. If the name was provided, service account, role, and role binding used to be configured manually before providing the name in the custom-values.yaml file. If the name was not provided, SEPP used to create the default service account, role, and role binding based on the release name.

The Lifecycle Management (LCM) Automation feature optimizes deployment and upgrade processes of SEPP by automating service account creation. This enhancement allows you to automatically create user-defined service accounts without any manual intervention. The automation process includes:

- **Creating Service Account:** An automated resource service account creation has been introduced in this release to streamline the management of Kubernetes resources through Helm charts.
- **Applying NF Alert Configuration rules:** SEPP leverages the oso-alr-config Helm chart, introduced as part of the OSO package, to apply alert rules using Helm upgrades. While the oso-alr-config Helm chart is deployed automatically during the OSO installation, the SEPP performs Helm upgrades to apply or update the required alert rules. Both manual and automated configurations are supported.

Operational Service Overlay (OSO) Alerts Automation

Alerts are automated by using the Helm upgrade command with the Helm chart provided as part of the OSO software package. For more information, see OSO Alerts Automation section.

Managing LCM Based Automation

Enable

You can enable this feature using the following methods:

- **Manual:** see the "Manually Creating Service Account, Role, and Rolebinding" section in Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide.
- **Automatic:** see the "Automatically Creating Service Account, Role, and Rolebinding" section in Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide.

Configure

You can configure this feature using the following Helm parameters as described in the "Global Parameters" section of Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide:

Automating Service Account, Role and Role Binding

To automate the creation of service account, role, and role binding, enable the following parameters:

- Global Flag (`autoCreateResources.enabled`): Controls the overall automation of resource creation. This flag is disabled by default and must be set to true to enable automation.
- Resource-specific Flag (`autoCreateResources.serviceAccount.create`): Controls ServiceAccount creation at the resource level. This flag is enabled by default and make sure it is set to true alongside the global flag to enable ServiceAccount automation. This flag is conditional on the global flag and will only take effect if the global flag is set to true. If the flag is disabled, you must create the service accounts manually. The role and role binding resources are created along with the service account as part of this automation.

The following table provides the service account creation behavior using different combinations:

Table 3-9 Service Account Creation using Different Combinations

Parameter	serviceAccountName	Result
<code>autoCreateResources.enabled:true</code> <code>serviceAccount.create:true</code>	Provided	Service account is created or updated with the provided serviceAccountName.
<code>autoCreateResources.enabled:true</code> <code>serviceAccount.create:true</code>	Not provided	Service account is created or updated with .Release.name.
<code>autoCreateResources.enabled:true</code> <code>serviceAccount.create:false</code>	Provided	Service account is not created.
<code>autoCreateResources.enabled:true</code> <code>serviceAccount.create:false</code>	Not provided	The deployment fails. The serviceAccountName is mandatory.
<code>autoCreateResources.enabled:false</code> <code>serviceAccount.create:true</code> or <code>false</code>	Provided	The service account is not created. It must be created manually.
<code>autoCreateResources.enabled:false</code> <code>serviceAccount.create:true</code> or <code>false</code>	Not Provided	Service account is created or updated with .Release.name

Note**Upgrade Process:**

If a customer is upgrading from an older component version that uses manually created service accounts (SAs) to a version that supports automatic SA creation (where SAs are generated automatically for each component), and they want to transition to these automated service accounts, they must specify new SA names in the `ocsepp_custom_values_<version>.yaml` file. This triggers the automatic creation of the new service accounts, roles, and role bindings via Helm charts. The customer must manually remove the old service accounts.

Observe

There are no new metrics and KPIs added or updated for this feature.

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.7 Traffic Prioritization using Egress Gateway Pod Protection

The Ingress Gateway and Egress Gateway microservices handle incoming and outgoing traffic, respectively, for Network Functions (NFs). These microservices are crucial for maintaining network stability. However, they can be susceptible to overloads caused by unexpected traffic spikes, uneven distribution, or bursts due to fluctuating network conditions. Such overloads can lead to system instability or even network outages, highlighting the need for strong protection mechanisms to safeguard these critical gateway microservices.

The NRF-Client has implemented a mechanism that assigns a priority value to incoming and outgoing calls, considering the `3gpp-Sbi-Message-Priority` header (if present in the request) and the configuration provided through the REST APIs. With this feature in place, a rate-limiting mechanism is now applied to the Egress Gateway pods. This mechanism sets a maximum request limit for each pod. When the request rate exceeds the threshold, the pods take protective action by either rejecting additional requests with a custom error code or allowing them, depending on the configured settings.

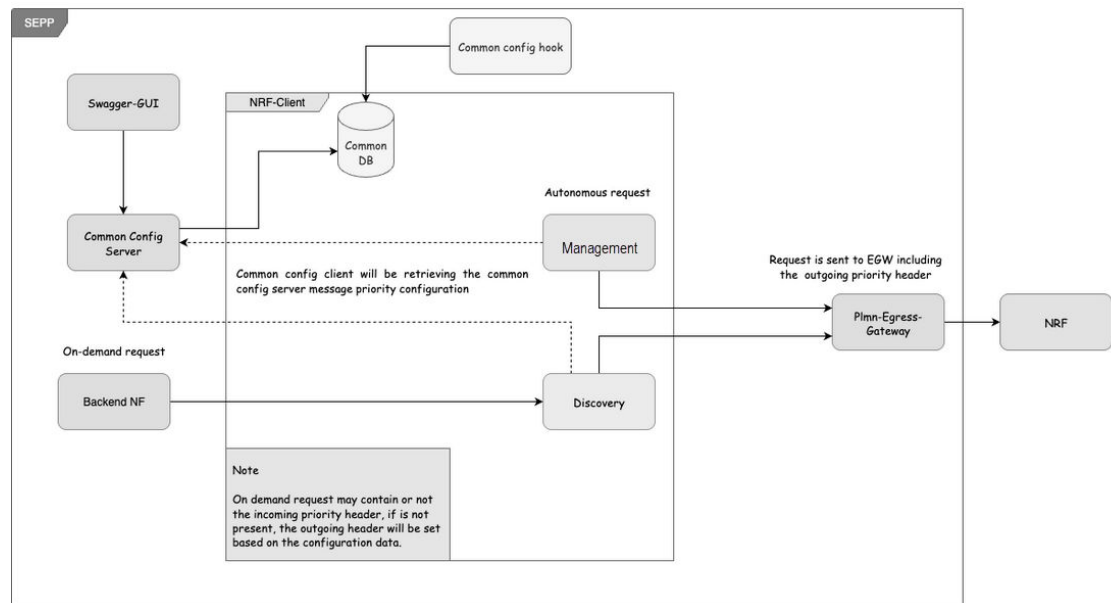
To ensure proper handling of NRF messages during overload situations at the PLMN egress gateway, SEPP must implement a mechanism that prioritizes these messages. In cases where the PLMN gateway reaches its maximum transactions per second (TPS) limit and enters an overload state, it is crucial that NRF messages continue to be processed and take precedence over other SBI messages.

The decision to accept or reject messages during overload will depend on the overall CPU usage at the PLMN Egress Gateway. To manage this, the system will define pod protection and congestion level configurations at the gateway, ensuring that NRF messages are prioritized for processing. This approach applies to both Inter-NRF and Intra-NRF messages, ensuring continuous and efficient handling under high load conditions.

Architecture

This section details the functionality of the feature.

Figure 3-10 SEPP Traffic Prioritization using Egress Gateway Pod Protection



The PLMN Egress Gateway microservice manages all outgoing traffic from Network Functions (NFs). The PLMN Egress Gateway microservice may become vulnerable due to overload conditions during unexpected traffic spikes, uneven traffic distribution, or bursts caused by network fluctuations. To ensure system stability and prevent potential network outages, a protection mechanism for the PLMN Egress Gateway microservice is necessary.

This feature introduces a rate-limiting system for PLMN Egress Gateway pods. The mechanism sets a maximum number of requests that each pod can handle. If the request rate exceeds this threshold, the pods take protective actions. Depending on their configuration, they may either reject excess requests with a custom error code or allow them through. Each PLMN Egress Gateway pod has its own Pod Protection Policier, which operates independently for each pod.

The plmn Egress Gateway's configuration for this feature includes:

- **Fill Rate:** This sets the maximum number of requests a pod can handle in a defined time interval. It is also possible to allocate the specific percentage for various path or method combinations.
- **Error Code Profiles:** This allows to reject the requests with a predefined error code profile.
- **Congestion Configuration:** This allows to configure the congestion levels based on CPU resource utilization.

- **Denied Request Actions:**

This setting determines the actions to be taken when the fill rate is exceeded. The action is based on the priority defined in the denied request actions and the configured response.

The available options for action are:

- **CONTINUE:** Allows requests to be processed even if the fill rate is exceeded.
- **REJECT:** Rejects requests and returns a specified error code.

By default, the Egress Gateway will reject all requests exceeding the fill rate, unless the 'CONTINUE' action is specifically defined in the denied request actions configuration.

The fill rate defines the maximum number of requests a pod can handle when traffic is evenly distributed over a one-second interval. However, in bursty traffic scenarios, where the traffic is unevenly distributed within that same one-second window, even if the average traffic falls within the fill rate limit, traffic failures may still occur. This is because bursts of high traffic within short periods can exceed the pod's capacity, leading to potential issues despite the overall traffic being within acceptable limits.

Congestion level is checked in a fixed interval (`congestionconfig.refreshInterval`) and the action will be decided based on the level in that interval. It is possible that the CPU levels might toggle between these intervals causing the congestion levels to switch frequently. To know the detected congestion level in the Egress Gateway, see the metrics `oc_egressgateway_congestion_level_bucket_total`.

The following table describes SEPP's configuration for different congestion levels for the Egress Gateway pods:

Table 3-10 Congestion Levels of Pods

Level Value	Abatement Threshold	Onset Threshold	Level Name	Description
1	58	65	Normal	The default congestion level of pod is "0". The pod moves from Normal to default when the CPU is less than the abatement of Normal (58). The pod moves from default to Normal state when the CPU exceeds onset of Normal (65). The pod moves from Danger of Congestion to Normal state, when the CPU is less than abatement of Danger of Congestion (70).
2	70	75	Danger of Congestion	The pod moves from Danger of Congestion to Normal, when the CPU is less than abatement of Danger of Congestion (70). The pod moves from Normal to Danger of Congestion state when the CPU exceeds onset of Danger of Congestion (75). The pod moves from Congested to Danger of Congestion when the CPU is less than abatement of Congested (80).

Table 3-10 (Cont.) Congestion Levels of Pods

Level Value	Abatement Threshold	Onset Threshold	Level Name	Description
3	80	85	Congested	The pod moves from Congested to Danger of Congestion when the CPU is less than abatement of Congested (80). The pod moves from Danger of Congestion to Congested state when the CPU exceeds onset of Congested (85).

Note

- If the current level is at L3 (Congested with onset value of 85) and the CPU value is 66, the level would be marked as L1 (Normal). The abatement value is considered only when the switching is between the adjacent levels (L2 (Danger of Congestion) to L1 (Normal) and L3 (Congested) to L2 (Danger of Congestion)).
- When SEPP Discovery microservice receives an overload response from Egress Gateway microservice, it does not retry or re-reroute this request. An error response will be generated.

Managing the Feature

You can enable the SEPP Traffic Prioritization using Egress Gateway Pod Protection feature using the Helm and CNC Console, or REST API.

Prerequisite

This feature is disabled by default. To enable the Egress Gateway Pod Protection using rate limiting, it is necessary to activate the feature and configure Traffic Prioritization for NRF-Client services (such as nf-management).

Enable using Helm (Before Release 26.1.200)

Set `trafficPrioritization.enabled` to `true` in the `nrf client` microservice section.

```
trafficPrioritization:
  featureEnabled: false
  incomingPriorityHeader: 3gpp-sbi-message-priority
  outgoingPriorityHeader: 3gpp-sbi-message-priority
  nfSubscribeMessageTypes:
    - messageType: AutonomousNfDiscovery
      subscribeNfTypes:
        - nfType: SEPP
          priority: 2
    - messageType: AutonomousNfSubscribe
      subscribeNfTypes:
        - nfType: SEPP
          priority: 2
    - messageType: AutonomousNfUnSubscribe
```



```

    subscribeNfTypes:
      - nfType: SEPP
        priority: 3
    - messageType: AutonomousNfSubscriptionUpdate
      subscribeNfTypes:
        - nfType: SEPP
          priority: 3
    messageType: AutonomousHealthCheck
      priority: 1
    - messageType: NfDiscovery
      priority: 2
    - messageType: AutonomousOnDemandNFRegistration
      priority: 1
    - messageType: NfHeartBeat
      priority: 1
    - messageType: AutonomousNfPatch
      priority: 3
    - messageType: NfProfileUpdate
      priority: 3
    - messageType: NfSubscribe
      priority: 4
    - messageType: NfSubscriptionUpdate
      priority: 3
    - messageType: NfUnSubscribe
      priority: 3
    - messageType: NfDeRegistration
      priority: 2
    - messageType: AutonomousNfDeRegistration
      priority: 2
    - messageType: NfProfileRetrieval
      priority: 3
    - messageType: NfListRetrieval
      priority: 4

```

Enable using REST API

This is applicable for on or after Release 26.1.200.

Set `trafficPrioritization.enabled` parameter to true in `{apiRoot}/sepp/nf-common-component/v1/nrf-client-nfmanagement/trafficPrioritization/node` REST API.

Set `podProtection.enabled` parameter to true in `{apiRoot}/sepp/nf-common-component/v1/egw/plmn/podProtectionByRateLimiting` REST API.

Note

NRF traffic can be prioritized at the PLMN Egress Gateway by adding following route in `PodProtectionByRateLimiting` configuration:

```

"routes": [{ "id": 1, "path": "/nnrf-*/**", "methods": [ "POST", "GET",
"PATCH", "DELETE", "PUT" ], "percentage": 10 }]

```

For more information about API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Enable using CNC Console

Enable through NRF Client

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**. The **Gateway**, **cnDBTier**, and **NRF-Client** options appear underneath.
2. Click **NRF-Client** tab. The **NRF-Client** page is displayed.
3. From the list of tabs, select the **Traffic Prioritization** tab.
4. Click **Edit** to edit the Traffic Prioritization configuration. The **Edit Traffic Prioritization** page is displayed.
5. Set the **Feature Enabled** to true.

Enable through PLMN Egress Gateway

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then **Egress Gateway**.
2. From the left navigation menu, navigate to **PLMN Egress Gateway**.
3. Click the **Pod Protection By Rate Limiting** option under the **PLMN Egress Gateway**. The **Pod Protection By Rate Limiting** page is displayed.
4. Click **Edit** to edit the configuration. The **Edit Pod Protection By Rate Limiting** page is displayed.
5. Set the **Enabled** to True.

For more details about CNC Console configurations, see *Configuring SEPP using CNC Console* section.

Configure

You can configure the feature using REST API and CNC Console.

- **Configure using REST API:** Perform the following configurations for this feature:
 - Configure the Error code profile using the `{apiRoot}/sepp/nf-common-component/v1/egw/plmn/errorcodeprofiles` API.
 - Configure the PodProtection configuration using the `{apiRoot}/sepp/nf-common-component/v1/egw/plmn/podProtectionByRateLimiting/node` API.
 - Configure the congestion level of the pods based on CPU using the `{apiRoot}/sepp/nf-common-component/v1/egw/plmn/congestionConfig/node` API.

For more information about these APIs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

- **Configure using CNC Console:** Perform the feature configurations as described in the 'NRF Client' and 'PLMN Egress Gateway' sections in [Configuring SEPP using CNC Console](#).

Observe

Metrics

The following metrics are available for this feature:

- `nrfclient_nw_conn_out_request_total`

- `oc_egressgateway_http_requests_total`
- `oc_egressgateway_http_request_ratelimit_values_total`
- `oc_egressgateway_http_request_ratelimit_reject_chain_duration_histogram_seconds_bucket`
- `oc_egressgateway_http_request_ratelimit_reject_chain_length_histogram_bucket`
- `oc_egressgateway_congestion_cpu_state`
- `oc_egressgateway_congestion_system_state`
- `oc_egressgateway_system_state_duration_percentage`
- `oc_egressgateway_congestion_level_total`
- `oc_egressgateway_congestion_level_bucket_total`
- `oc_egressgateway_congestion_cpu_percentage_bucket`

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.8 SEPP and NRF Communication

Overview

The Network Repository Function (NRF) plays a critical role in the interaction with the SEPP. It provides essential services that enable SEPP to securely and efficiently communicate with other network functions (NFs) in a 5G core network. Below are the key features and functionalities that the NRF offers to the SEPP:

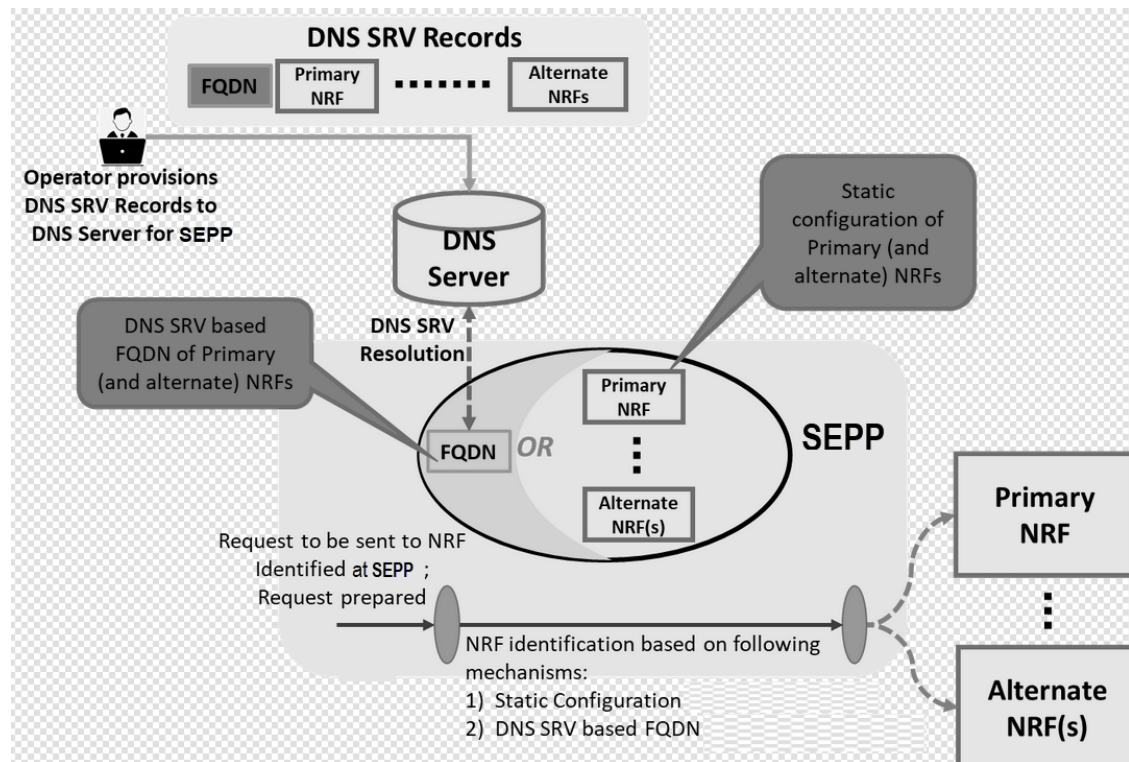
Key Features of NRF and SEPP Communication

1. **Service Discovery:** The NRF enables dynamic service discovery, allowing SEPP to automatically discover the addresses and capabilities of other NFs, both within its own network and in roaming partner networks. This eliminates the need for manual, static configurations, enhancing the flexibility of the cloud-native 5G architecture.
2. **NF Management and Status Monitoring:** The NRF maintains an up-to-date real-time database of all available NFs, tracking their status, capacity, and load. SEPP uses this information to select healthy and available NFs for communication, ensuring optimal network performance and reliability.
3. **Authorization and Security:** As part of the Service-Based Architecture (SBA) in 5G, the NRF functions as an authorization server. SEPP relies on the NRF to obtain security tokens based on the OAuth2.0 framework, ensuring that only authorized NFs can communicate with each other. This security mechanism is crucial for safeguarding inter-PLMN (Public Land Mobile Network) interfaces managed by SEPP.

4. Dynamic Configuration and Scalability: In a dynamic 5G environment, where NFs can be added, removed, or scaled, the NRF handles the registration and deregistration processes. This eliminates the need for complex, manual reconfigurations whenever changes occur, making the network more scalable and efficient. For more information about NRF interface registration and deregistration, refer [NRF Interface Registration and Deregistration](#) section.

Selecting NRF for SEPP Communication

Figure 3-11 Selecting NRF for SEPP Communication



NRF Request Handling

- SEPP communicates with the NRF through the nrf-client microservice. This handles various NRF requests, such as registration, heartbeat, discovery, and subscription.
- SEPP also manages inter-PLMN requests (e.g., discovery and subscription requests) from visited networks. These requests are directed to the plmn-egress-gateway microservice, where NRF selection is managed by route configurations within the gateway.

Multiple NRF Configurations and Routing

- Multiple NRFs can be configured for both inter-PLMN and intra-PLMN communication. The routing between NRFs is based on priority and weight. This allows flexible and efficient management of communication paths.
- In the case of virtual FQDN (Fully Qualified Domain Name) configurations, DNS requests are routed to the external DNS via the alternate-route-svc. This service provides flexibility by allowing you to configure static FQDNs for each virtual FQDN in the NRF configuration.

Routing Flexibility

- The alternate-route-svc service facilitates the use of static FQDNs for NRF routing, enhancing the flexibility of managing network paths and ensuring that SEPP can interact with the correct NRFs dynamically based on configuration.

Different Ways to Configure NRF:

Following are two primary methods for selecting and configuring NRF:

- [Static Host Based Selection](#)
- [DNS SRV Based Selection](#)

For more information about the features supported by nrf-client, refer [NRF client selection mechanism using nrf-client](#) section.

Configuring through CNC Console

The user can configure the error criteria or error action set to configure the exceptions for the alternate routing. For more information, refer CNC Console [PLMN Egress Gateway](#) section.

3.2.8.1 Static Host Based Selection

The Static Host Based Selection feature allows the SEPP to configure multiple static NRF instances for reliable and fault-tolerant communication. SEPP is set up with a primary NRF and one or more fallback NRFs. If the primary NRF becomes unreachable, SEPP will automatically switch to one of the fallback NRFs, ensuring uninterrupted service.

This feature is particularly useful for environments where network topology is stable and fixed, enabling a straightforward approach to NRF selection without the need for dynamic resolution.

Key Features:

- **Primary and Fallback NRF Configuration:** SEPP is configured with a primary NRF and one or more secondary NRFs. If the primary NRF is unavailable, the secondary NRF will take over, ensuring service continuity.
- **Direct NRF Registration:** If `enableVirtualNrfResolution` is set to false, the nrf-client will directly register with the specified primary and secondary NRFs using their configured addresses.
- **Helm Configurations:** This feature can be enabled by setting specific Helm parameters that define the primary and secondary NRF addresses, as well as the communication protocol (HTTP/HTTPS).

Helm Configuration Example:

To configure Static HostBased Selection for NRF in SEPP, the following Helm parameters need to be set:

Config-mgr-svc section:

```
nrfProfile:
  enableVirtualNrfResolution: false
  static:
    primaryNrfApiRoot: ocnrf-ingressgateway.ocnrf:80
    secondaryNrfApiRoot: ocnrf-2-ingressgateway.ocnrf-sepp:80
    nrfScheme: http
```

NRF Client section:

```
primaryNrfApiRoot: ocnrf-ingressgateway.ocnrf:80
secondaryNrfApiRoot: ocnrf-2-ingressgateway.ocnrf-sepp:80
nrfScheme: http
```

Both the config-mgr-svc and nrf-client sections must contain identical values for primaryNrfApiRoot, secondaryNrfApiRoot, and nrfScheme.

Retry Mechanism:

- If the primaryNrfApiRoot becomes unavailable, the nrf-client will attempt to connect to the secondaryNrfApiRoot. The connection type (HTTP or HTTPS) is determined by the nrfScheme parameter.
- The nrfRetryConfig parameter defines the retry logic and conditions under which retries will occur, specifying the number of retry attempts and error handling conditions. Below is an example of a retry configuration:

```
nrfRetryConfig: [
  {
    "serviceRequestType": "ALL_REQUESTS",
    "primaryNRFRetryCount": 1,
    "nonPrimaryNRFRetryCount": 1,
    "alternateNRFRetryCount": -1,
    "errorReasonsForFailure": [
      "503", "504", "500", "SocketTimeoutException",
      "JsonProcessingException", "UnknownHostException",
      "NoRouteToHostException", "IOException"
    ],
    "gatewayErrorCodes": ["503"],
    "requestTimeout": 10
  },
  {
    "serviceRequestType": "AUTONOMOUS_NFREGISTER",
    "primaryNRFRetryCount": 1,
    "nonPrimaryNRFRetryCount": 1,
    "alternateNRFRetryCount": -1,
    "errorReasonsForFailure": [
      "503", "504", "500", "SocketTimeoutException",
      "JsonProcessingException", "UnknownHostException",
      "NoRouteToHostException", "IOException"
    ],
    "gatewayErrorCodes": ["503"],
    "requestTimeout": 10
  }
]
```

For further details on Helm parameters and configurations, refer *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

3.2.8.2 DNS SRV Based Selection

SEPP can dynamically choose NRF instances based on real-time availability and site redundancy through DNS SRV configurations. This allows SEPP to resolve NRFs using DNS SRV based FQDNs, in addition to the static configurations defined by operators.

Internally, SEPP utilizes the Alternate Route Service, which enables the system to discover and select different NRF instances by performing DNS SRV-based lookups. This service translates FQDNs, or virtual FQDNs, into alternate NRF addresses, giving SEPP the flexibility to prioritize and adjust connections to NRFs depending on specific service requirements.

For further details on this functionality, refer to section "Alternate Routing and Load Sharing based on the DNS SRV Record for Home Network Functions".

Prerequisites:

- The Alternate-Route Service must be enabled and running to use DNS SRV-based selection for NRF.
- For detailed setup and activation, refer to the "Managing Alternate Routing and Load Sharing based on the DNS SRV Record for Home Network Functions" section.

Helm Configuration for DNS SRV-Based NRF Selection:

To configure DNS SRV based selection for NRF in SEPP, the following Helm parameters must be set in the config-mgr-svc and nrf-client sections.

Config-mgr-svc Configuration:

```
nrfProfile:
  enableVirtualNrfResolution: true
  virtual:
    virtualNrfFqdn: nrf.virtual.fqdn
    virtualNrfScheme: http
```

- The enableVirtualNrfResolution parameter must be set to true when using Virtual FQDN for NRF resolution. This configuration will create the required route configuration at the plmn-egress-gateway microservice.
- The PLMN-EGW peer configuration will include the virtualHost parameter.
- Once the plmn-egress-gateway identifies the virtualHost in the peer configuration:
 - It will connect to the alternate-route-svc for DNS SRV lookups.
 - The Alternate Route Service will retrieve the DNS records for the configured virtual FQDN.

NRF-Client Configuration:

```
enableVirtualNrfResolution: true
virtualNrfFqdn: nrf.virtual.fqdn
virtualNrfScheme: http
```

- The nrf-client section must also set enableVirtualNrfResolution to true when using Virtual FQDN for resolving NRF.
- Then nrf-client will connect to the alternate-route-svc for the DNS SRV records. Alternate Route service will fetch the DNS records for the configured virtual FQDN.

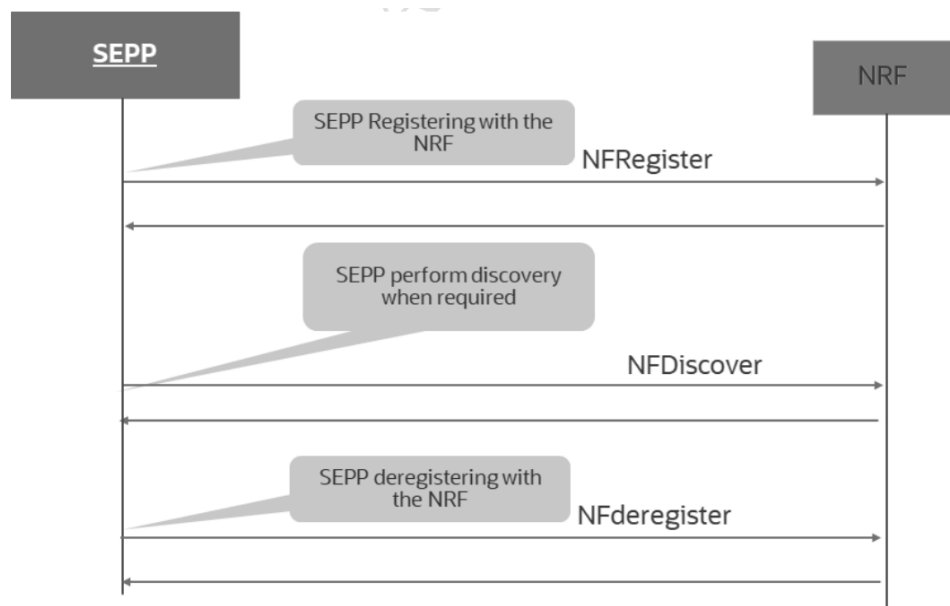
- The NRF configuration can be updated at both the nrf-client microservice and plmn-egress-gateway through Helm by upgrading SEPP with the new configurations specified in both the config-mgr-svc and nrf-client sections.
- SbiRouting Error-Criteria/Action-Sets at the plmn-egress-gateway microservice can be modified later through:
 - CNC Console
 - REST APIs (details available in the Egress Gateway REST API section of the SEPP Rest API Guide).

3.2.8.3 NRF Interface Registration and Deregistration

The Network Repository Function (NRF) stores all the information about NFs. SEPP supports registering into the NRF and deregistering from the NRF when required. This allows for NFs to discover the SEPP when required for routing interPLMN traffic.

The following diagram depicts the topology for NRF Interface Registration & Reregistration.

Figure 3-12 NRF Interface Registration and Reregistration



Managing NRF Interface Registration and Deregistration

Enable

```

nfName: sepp
# Global control to enable/disable deployment of NF Discovery service, enable
it if on demand discovery of NF is required.
nrfClientNfDiscoveryEnable: true
# Global control to enable/disable deployment of NF Management service.
nrfClientNfManagementEnable: true
  
```


Configure

You can configure the NRF Interface Registration and Deregistration using helm parameters:

Configuring using Helm parameters

profile:

```
[appcfg]
primaryNrfApiRoot= http://10.75.236.102:31294 #NRF Ingress Gateway IP
secondaryNrfApiRoot=
nrfScheme=http
retryAfterTime=PT120S
nrfClientType=SEPP
nrfClientSubscribeTypes=
appProfiles= [{"nfInstanceId":"9faf1bbc-6e4a-4454-a507-
aef01a101a06","nfType":"SEPP","nfStatus":"REGISTERED","fqdn":"ocsepp-release-
plmn-ingress-gateway.DEPLOYMENT_NAMESPACE","scheme":"http"}]

deploymentNrfClientService:
  # Services to be monitored by performance service
  # If no services are to be monitored,
  envNfNamespace,envNfType,envConsumeSvcName can be left blank
  envNfNamespace: csepp
  envNfType: 'sepp'
  envConsumeSvcName: 'appinfo'

# Service to be monitored by appinfo. We can add a specific SEPP service name
core_services:
sepp: []
```

Observe

For more information on NRF Interface Registration and Deregistration metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.8.4 NRF Selection Mechanisms using nrf client

Overview

In previous releases, the SEPP selects the Network Repository Function (NRF) for communication based on static configurations set by the operator. However, there are scenarios where the NRF may go down for various reasons. In such cases, the SEPP, which relies on the NRF for specific communications, may experience service disruptions.

Note

This feature is applicable only to INTRA-NRF requests. All INTER-PLMN (NRF and non-NRF) requests will be routed directly through PLMN-EGW, using EGW route configurations.

This feature allows the SEPP to dynamically select NRF instances based on real-time availability and site redundancy through DNS SRV configurations. In addition to the static configurations by operators, the SEPP can now resolve NRFs using DNS SRV-based Fully Qualified Domain Names (FQDNs). The SEPP is configured with a primary NRF and multiple fallback NRFs, which take over if the primary NRF becomes unreachable.

The nrf-client uses the Alternate Route Service, which helps the SEPP find and select different Network Repository Functions (NRFs) by using DNS SRV-based lookups. This service allows the SEPP to translate Fully Qualified Domain Names (FQDNs) or virtual FQDNs into alternate NRF addresses. This setup enables the SEPP to prioritize and adjust connections to different NRFs based on specific service needs.

The selection of the target NRF is determined by configurations set by the operator within the nrf-client. This selection can be:

- Static-based, using primary and secondary apiRoot values, or
- DNS-SRV-based, leveraging service discovery mechanisms.

Once all target NRFs have undergone health checks, the final route list is updated accordingly.

Call Flow of NRF Selection Mechanisms Using nrf client

1. **DNS SRV Record Lookup:** The SEPP starts by using DNS SRV records to fetch information about the FQDN of the primary NRF and alternate NRFs.
2. **DNS Query:** The DNS server receives a query from the SEPP to resolve the FQDN of the primary NRF or alternate NRFs.
3. **DNS Resolution:** The DNS server resolves the FQDN and returns the resolved address for the primary NRF or one of the alternate NRFs.
4. **NRF Identification by SEPP:** Based on the FQDN provided by DNS, the SEPP identifies the primary NRF to communicate with. If the primary NRF is unavailable, the SEPP uses one of the alternate NRFs.
5. **Static Configuration Option:** Alternatively, the SEPP can rely on a static configuration of the primary and alternate NRFs instead of querying DNS.
6. **Request Preparation:** Once an NRF (primary or alternate) is identified, the SEPP prepares the request to be sent to the selected NRF.
7. **Communication with NRF:** The SEPP sends the request to the identified NRF, enabling further processing or service access.

The Alternate Route Service utilizes DNS SRV records to look up virtual FQDNs. The Egress Gateway queries this service to retrieve a list of alternate NRFs along with their priorities. Based on these priorities, the Egress Gateway reroutes requests to other NRF instances as required.

Key Operations of nrf-client for High Availability

The SEPP uses a component called nrf-client to facilitate communication with NRFs. The nrf-client performs critical functions to ensure NRFs remain available and responsive:

1. **Traffic Routing:** The nrf-client directs all requests (for example, registration, updates, heartbeats, and discovery) to the primary NRF.
2. **Failure Detection:** If the primary NRF cannot be reached, the nrf-client detects the failure either through routing errors or by receiving a "503 Service Unavailable" response, which may include a "Retry-After" interval.
3. **Failover and Retry:** If the primary NRF fails, the nrf-client temporarily marks it as unavailable and reroutes traffic to a secondary NRF based on DNS SRV priority. The retry interval, specified in the "Retry-After" header or a preset interval, determines when the nrf-client reattempts contact with the primary NRF.
 - During this interval, all requests are routed to the secondary NRF.
 - Once the retry interval expires, the nrf-client resumes attempts to communicate with the primary NRF.

① Note

- The nrf-client only tries to connect with each NRF once. If it cannot send a request to an NRF, it returns a "503 Service Unavailable" response to the SEPP. This one-time attempt helps avoid repeated failures and ensures alternate routing to secondary NRFs when needed.
- When using nrf-client for NRF requests, note the following:
 - Messages with Content-Encoding: gzip are not supported.
 - Messages with Content-Type: multipart are not supported.

Managing Static Based Selection of NRF in SEPP

This section provides information about Helm configurations required to configure static based selection of NRF in SEPP.

If `enableVirtualNrfResolution` is set to `false`, the nrf-client will use the configured values of `primaryNrfApiRoot` and `secondaryNrfApiRoot` to register with the NRF (Network Repository Function) directly.

You can enable this feature using Helm configurations. To do so, you must set the following Helm parameters:

```
primaryNrfApiRoot= ocnrf-ingressgateway.ocnrf:80
secondaryNrfApiRoot= ocnrf-2-ingressgateway.ocnrf-sepp:80
nrfScheme=http
```

If the `primaryNrfApiRoot` is unavailable, then nrf-client shall retry to the configured `secondaryNrfApiRoot` FQDN. The parameter `nrfScheme` determines whether connection to NRF will be establish over HTTP or HTTPS.

The `nrfRetryConfig` parameter defines the conditions under which nrf-client will attempt retries with alternate NRFs. .

```
nrfRetryConfig=[{"serviceRequestType":"ALL_REQUESTS","primaryNRFRetryCount":1,
"nonPrimaryNRFRetryCount":1,"alternateNRFRetryCount":-1,"errorReasonsForFailure":
```

```
[
  "503", "504", "500", "SocketTimeoutException", "JsonProcessingException", "UnknownHostException", "NoRouteToHostException", "IOException"], "gatewayErrorCodes":
  ["503"], "requestTimeout": 10},
  {"serviceRequestType": "AUTONOMOUS_NFREGISTER", "primaryNRFRetryCount": 1, "nonPrimaryNRFRetryCount": 1, "alternateNRFRetryCount": -1, "errorReasonsForFailure":
  ["503", "504", "500", "SocketTimeoutException", "JsonProcessingException", "UnknownHostException", "NoRouteToHostException", "IOException"], "gatewayErrorCodes":
  ["503"], "requestTimeout": 10}]
```

Managing NRF Selection Mechanisms Using nrf client

This section provides information about Helm configurations required to configure DNS SRV based selection of NRF in SEPP.

Enable

You can enable this feature using Helm configurations. To enable this feature, it is mandatory to configure the following Helm parameters:

```
enableVirtualNrfResolution=true
virtualNrfFqdn=nrf.oracle.com
virtualNrfScheme=http
```

Note

If `enableVirtualNrfResolution` is set to `false`, the `nrf-client` uses `primaryNrfApiRoot` and `secondaryNrfApiRoot` configurations to register with the NRF. However, if `enableVirtualNrfResolution` is set to `true` and an incorrect `virtualNrfFqdn` is configured, the `nrf-client` does not fallback to `primaryNrfApiRoot` and `secondaryNrfApiRoot` for registration, resulting in a registration failure.

The parameter `nrfScheme` determines whether connection to NRF will be establish over HTTP or HTTPS.

Note

For ASM setups, scheme would always be set to 'http.'

For more details on additional Helm parameters related to this feature, refer to the Helm configuration section below.

Managing Health Check of NRF in SEPP

The `nrf` client performs health checks on all configured NRFs and keeps a list of healthy ones to use for service requests. NRFs can be configured in two ways:

- Statically (using direct FQDNs), or
- Dynamically (through DNS query using a virtual FQDN).

If the `HealthCheck` feature is enabled, it starts after the NF has successfully registered with an NRF. To ensure there is at least one healthy NRF available for session-based requests, the NRF used for the successful registration is automatically treated as healthy.

Enable

You can enable this feature using Helm configurations. To do so, you must set the following mandatory Helm parameters:

```
healthCheckConfig={"healthCheckCount":1,"healthCheckInterval":5,"requestTimeout":10,"errorReasonsForFailure":
["503","504","500","SocketTimeoutException","IOException"],"gatewayErrorCodes":[]}
```

Configure

Helm Configurations

To enable this feature, configure the following nrf-client Helm parameters in `ocsepp_custom_values_<versions>.yaml` file.

```
profile: |-
  [appcfg]
  primaryNrfApiRoot= ocnrf-ingressgateway.ocnrf:80
  secondaryNrfApiRoot=
  nrfScheme=http
  retryAfterTime=PT120S
  nrfClientType=SEPP
  nrfClientSubscribeTypes=
  appProfiles= [{"nfInstanceId":"9faf1bbc-6e4a-4454-a507-
aef01a101a06","nfType":"SEPP","nfStatus":"REGISTERED","heartBeatTimer":30,"fqdn":"ocsepp-plmn-ingress-gateway.DEPLOYMENT_NAMESPACE","plmnList": [{"mcc":
"333","mnc": "222"}, {"mcc": "444","mnc": "555"},
{"mcc": "444","mnc": "55"}], "capacity":500, "locality":"delhi", "priority":1 , "nfSetIdList":
["set001.seppset.5gc.mnc444.mcc555", "set001.seppset.5gc.mnc222.mcc333"]}]]
  registrationRetryInterval=5000
  subscriptionRetryInterval=5000
  enableDiscoveryRefresh=false
  discoveryRetryInterval=5000
  discoveryRefreshInterval=10
  discoveryDurationBeforeExpiry=90
  renewalTimeBeforeExpiry=3600
  validityTime=30
  enableSubscriptionAutoRenewal=true
  nfHeartbeatRate=80
  acceptAdditionalAttributes=false
  retryForCongestion=5
  supportedDataSetId=
  enableVirtualNrfResolution=false
  virtualNrfFqdn=nrf.oracle.com
  virtualNrfScheme=http
  subscriberNotificationInterval=1000
  requestTimeoutGracePeriod=2

nrfRetryConfig=[{"serviceRequestType":"ALL_REQUESTS","primaryNRFRetryCount":1,
"nonPrimaryNRFRetryCount":1,"alternateNRFRetryCount":-1,"errorReasonsForFailure":
["503","504","500","SocketTimeoutException","JsonProcessingException","Unknown
```

```
HostException", "NoRouteToHostException", "IOException"], "gatewayErrorCodes":
["503"], "requestTimeout": 10},
{"serviceRequestType": "AUTONOMOUS_NFREGISTER", "primaryNRFRetryCount": 1, "nonPri
maryNRFRetryCount": 1, "alternateNRFRetryCount": -1, "errorReasonsForFailure":
["503", "504", "500", "SocketTimeoutException", "JsonProcessingException", "Unknown
HostException", "NoRouteToHostException", "IOException"], "gatewayErrorCodes":
["503"], "requestTimeout": 10}}

healthCheckConfig={"healthCheckCount": 1, "healthCheckInterval": 5, "requestTimeou
t": 10, "errorReasonsForFailure":
["503", "504", "500", "SocketTimeoutException", "IOException"], "gatewayErrorCodes"
: []}

# Flag to determine if honoring Requester-Nf-Type is to be taken from query
parameter or not
discoveryHonoringRequesterNfTypeEnabled: true
```

Note

This is a sample configuration. Modify the parameters as per your setup. For detailed information about the Helm parameters, see the "Customising SEPP" section in the *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Table 3-11 nrf client Global Parameters

Name	Description	Details
discoveryHonoringRequesterNfTypeEnabled	This is a mandatory parameter. Flag to determine if the Requester -Nf- Type parameter is to be taken from query parameter or not.	Data Type: Boolean Range: true or false Default Value: true

Table 3-12 NRF Retry Configuration Parameters

Name	Description	Details
serviceRequestType	This is a mandatory parameter. The NRF service request type.	Data Type: String Range: ALL_REQUESTS, AUTONOMOUS_NFREGISTER, AUTONOMOUS_NFSTATUS_SUBSCRIBE, AUTONOMOUS_NFUNSUBSCRIBE, AUTONOMOUS_NFSUBSCRIBE_UPDATE, AUTONOMOUS_NFDISCOVER, AUTONOMOUS_NFHEARTBEAT, AUTONOMOUS_NFPATCH, NFREGISTER, NFUPDATE, NF_STATUS_SUBSCRIBE, NFDISCOVER, NF_SUBSCRIBE_UPDATE, NF_UNSUBSCRIBE, NFDEREGISTER, NF_PROFILE_RETRIEVAL, NF_LIST_RETRIEVAL, Note: ALL_REQUESTS is mandatory service type. All additional service types can be appended to the list based on requirement. Default Value: ALL_REQUESTS, AUTONOMOUS_NFREGISTER
primaryNrfRetryCount	This is an optional parameter. Number of times a service request shall be retried to the primary NRF in case of failure.	Data Type: Integer Range: NA Default Value: 1
nonPrimaryNrfRetryCount	This is an optional parameter. Number of times a service request shall be retried to the non-primary NRF in case of failure.	Data Type: Integer Range: NA Default Value: 1
alternateNRFRetryCount	This is an optional parameter. Number of alternate NRFs that shall be retried in case of failure.	Data Type: Integer Range: NA Default Value: -1 Note: A value of -1 indicates all available NRF instances are to be tried.

Table 3-12 (Cont.) NRF Retry Configuration Parameters

Name	Description	Details
errorReasonsForFailure	This is an optional parameter. The http status codes or exceptions for which retry shall be applied.	Data Type: Array[String] Range: [(All non 2xx HTTP status codes), "SocketTimeoutException", "JsonProcessingException", "UnknownHostException", "NoRouteToHostException"] Default Value: ["503", "504", "500", "SocketTimeoutException", "JsonProcessingException", "UnknownHostException", "NoRouteToHostException", "IOException"]
requestTimeout	This is an optional parameter. Indicates the timeout period where no response is received from the Egress Gateway. Unit: seconds	Data Type: Integer Range: NA Default Value: 10 seconds
gatewayErrorCodes	This is an optional parameter. The http status codes sent by the egress-gateway for which retry shall be applied.	Data Type: Array[String] Range: All HTTP Status codes Default Value: 303

Table 3-13 Health Check Parameters

Name	Description	Details
healthCheckCount	This is an optional parameter. The number of consecutive success or failures required to mark an NRF healthy or unhealthy.	Data Type: Integer Range: -1, Values greater than 0. -1 (denotes that the feature is disabled) Default Value: -1
healthCheckInterval	This is an optional parameter. The interval at which a health check of an NRF shall be performed. Unit: seconds.	Data Type: Integer Range: NA Default Value: 5 seconds
requestTimeout	This is an optional parameter. The timeout period where no response is received from the egress-gateway. Unit: seconds	Data Type: Integer Range: NA Default Value: 10 seconds
errorReasonsForFailure	This is an optional parameter. The http status codes or exceptions for which retry shall be applied.	Data Type: Array[String] Range: ["503", "500", "504", "SocketTimeoutException", "JsonProcessingException", "UnknownHostException", "NoRouteToHostException"] Default Value: ["503", "504", "500", "SocketTimeoutException", "IOException"]

Table 3-13 (Cont.) Health Check Parameters

Name	Description	Details
gatewayErrorCodes	This is an optional parameter. The http status codes sent by the Egress Gateway for which retry shall be applied.	Data Type: Array[String] Range: ["503","500",504","SocketTimeoutException","JsonProcessingException","UnknownHostException","NoRouteToHostException"] Default Value: []

REST API

There are no REST API configurations required for this feature.

CNC Console

There are no CNC Console configurations required for this feature.

Observe**Metrics**

The following metrics are available for this feature:

- nrfclient_nrf_operative_status
- nrfclient_dns_lookup_request_total

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.8.5 Upgrading SEPP for NRF Configuration

When upgrading SEPP, you can update the NRF configuration in the following microservices of `ocsepp_custom_values_<version>.yaml` file:

1. nrf-client configuration
2. config-mgr-svc section

Once the NRF configuration is updated, the routes for NRF at the plmn-egress-gateway microservice will be automatically created or updated when the `nrfAutoRouteCreation` flag is set to true.

Steps to Upgrade SEPP and Update NRF Configuration:

1. After updating the NRF configuration in the SEPP custom values file, run the following command:


```
helm upgrade -f <custom-values.yaml> ocsepp/ ocsepp-release-name -n <namespace>
```
2. User can still modify the Inter-PLMN NRF SBI Error criteria/Action sets configuration using the plmn-egress-gateway REST APIs or the CNC Console.

Do the following during SEPP upgrade:

- Upgrading from older SEPP releases to 25.2.2x: Set the `nrfAutoRouteCreation` flag to false to avoid impacting existing NRF traffic when upgrading.
- Enabling automatic NRF route creation:
 - To enable the automatic creation of NRF routes, set the `nrfAutoRouteCreation` flag to true and upgrade SEPP. This will create the necessary Inter-PLMN NRF and Intra-PLMN NRF routes during the upgrade on the same release.
- Removing old NRF routes: If previous NRF routes were manually created, they should be removed from plmn-egress-gateway through CNC Console or REST APIs.

For more information about PLMN Egress Gateway REST API Configurations, see 'Egress Gateway Configuration' section in the *Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Note

On upgrade, with `nrfAutoRouteCreation=true` enabled, the default REST API configuration values are as specified in the [Verifying NRF Routes Post Installation](#) section. Please refrain from altering the NRF routes configuration after the upgrade.

3.2.8.6 Verifying NRF Routes Post Installation

The following depicts the routes created at the plmn-egress-gateway microservice for both NRF and SCP:

Route Overview in Plmn-egress-gateway

Table 3-14 Route Overview in Plmn-egress-gateway

Route Type	Pattern	Priority	SCP	Non-SCP	Mandatory/Optional	Notes
Route 1: Inter PLMN NRF Traffic	/nnrf- */** + SAN header present	Priority: 1	Yes	Yes	Mandatory	Automatically created at startup

Table 3-14 (Cont.) Route Overview in Plmn-egress-gateway

Route Type	Pattern	Priority	SCP	Non-SCP	Mandatory/Optional	Notes
Route 2: Intra PLMN NRF Traffic	/nnrf- */** + SAN header absent	Priority: 1	Yes	Yes	Mandatory	Automatically created at startup
Route 3: Producer NF Traffic	/nudr - */** (Example for UDR)	Priority: 2-100	No	Yes	Optional	Manual creation through CNC Console / REST API
Route 4: SCP Route	- / ** (default route)	Priority: 100	Yes	No	Mandatory	Manual creation via CNC Console / REST API
Default Route (for direct routing)	NA	Priority: Default	Yes	Yes	Mandatory	Automatically created at startup

Default Values of REST API Configurations

When the feature is enabled, the following are the default values of the REST API configurations:

Routes Configuration

URL : `http://<config-mgr-svc>:<config-mgr-port>/sepp/nf-common-component/v1/egw/plmn/routesconfiguration`

Description: Routes are necessary to direct incoming requests at the plmn-egress-gateway to specific pre-defined host or virtualHost. These routes are defined based on matching criteria, such as path-based or predicate-based conditions.

Method: GET/PUT

Body

Following Routes will be created by default for the Inter-plmn and Intra-plmn NRF communication.

SCP Mode (`enableScpDeployment = true`)

```
[{"id":"default_route","uri":"egress://request.uri","order":100,"filters":
[{"name":"DefaultRouteRetry"}], "metadata":
{"ignoreMaxRspTimeHeader":false}, "predicates":[{"args":{"pattern":"/
**"},"name":"Path"}]}, {"id":"nrfInterRoute","uri":"http://
ocsepp.com","order":1,"filters":[{"args":{"errorHandling":
```

```
[{"priority":0,"actionSet":"nrf_action_0","errorCriteriaSet":"nrf_criteria_1"}
,
{"priority":1,"actionSet":"nrf_action_0","errorCriteriaSet":"nrf_criteria_0"}]
,"peerSetIdentifier":"nrfPeer","customPeerSelectorEnabled":true},"name":"SbiRo
uting"},"metadata":
{"httpRuriOnly":true,"httpsTargetOnly":true,"sbiRoutingEnabled":true,"ignoreMa
xRspTimeHeader":false,"sbiRoutingWeightBasedEnabled":true},"predicates":
[{"args":{"pattern":"/nnrf-*/**"},"name":"Path"}, {"args":{"header":"x-custom-
ingress-client-identity"},"name":"Header"}]],
{"id":"intraNrfRoute","uri":"egress://request.uri","order":1,"filters":
[{"args":null,"name":"DefaultRouteRetry"}],"predicates":[{"args":{"pattern":"/
nnrf-*/**"},"name":"Path"}, {"args":{"headerName":"x-custom-ingress-client-
identity"},"name":"HeaderAbsent"}]]}]
```

To configure SCP route, refer to [Support for Core Networks with or without SCP/ Routing Proxy](#) section.

Non-SCP Mode (enableScpDeployment = false)

```
[{"id":"default_route","uri":"egress://request.uri","order":100,"filters":
[{"name":"DefaultRouteRetry"}],"predicates":[{"args":{"pattern":"/
**"},"name":"Path"}]],{"id":"nrfInterRoute","uri":"http://
ocsepp.com","order":1,"filters":[{"args":{"errorHandling":
{"priority":0,"actionSet":"nrf_action_0","errorCriteriaSet":"nrf_criteria_1"}
,
{"priority":1,"actionSet":"nrf_action_0","errorCriteriaSet":"nrf_criteria_0"}]
,"peerSetIdentifier":"nrfPeer","customPeerSelectorEnabled":true},"name":"SbiRo
uting"},"metadata":
{"httpRuriOnly":true,"httpsTargetOnly":true,"sbiRoutingEnabled":true,"sbiRouti
ngWeightBasedEnabled":true},"predicates":[{"args":{"pattern":"/nnrf-*/
**"},"name":"Path"}, {"args":{"header":"x-custom-ingress-client-
identity"},"name":"Header"}]]}]
```

Peer configuration

URL : `http://<config-mgr-svc>:<config-mgr-port>/sepp/nf-common-component/v1/egw/plmn/peerconfiguration`

Description: Peers are endpoint FQDNs or IP addresses. Multiple peers can be defined, and each must be assigned a unique identity.

Method: GET/PUT

Body

```
[{"id":"1","host":"ocnrf-ingressgateway.ocnrf","port":"80","apiPrefix":""}]
```

Primary and secondary NRFs mentioned in the Helm configuration will be converted to REST routes at plmn-egress-gateway microservice at start.

Peerset configuration

URL : `http://<config-mgr-svc>:<config-mgr-port>/sepp/nf-common-component/v1/egw/plmn/peerssetconfiguration`

Description: A PeerSet is defined to group multiple peers together, with a priority assigned to each peer within the set.

Method: GET/PUT

Body

```
[{"id":"nrfPeer","httpConfiguration":
[{"priority":1,"peerIdentifier":"1"}], "httpsConfiguration":
[{"priority":1,"peerIdentifier":"1"}]]]
```

Primary and secondary NRFs mentioned in the Helm configuration will be converted to REST routes at plmn-egress-gateway microservice at start.

SbiRouting Error Criteria configuration

URL : `http://<config-mgr-svc>:<config-mgr-port>/sepp/nf-common-component/v1/egw/plmn/sbiroutingerrorcriteriasets`

Description: Error Criteria sets is defined for the rerouting based on the errors generated from the server/ exceptions generated .

Method: GET/PUT

Body

```
[{"id":"nrf_criteria_1","method":
["GET","POST","PUT","DELETE","PATCH"],"response":{"statuses":[{"status":
[500,503,504],"statusSeries":"5xx"}],"headersMatchingScript":"headerCheck, serv
er,via,.*NRF.*"}}, {"id":"nrf_criteria_0","method":
["GET","POST","PUT","DELETE","PATCH"],"exceptions":
["java.util.concurrent.TimeoutException","java.net.SocketException","java.net.
SocketTimeoutException","java.net.UnknownHostException","java.net.ConnectExcep
tion","java.net.NoRouteToHostException"]}]]]
```

The Error Criteria for alternate routing, as specified in the Helm configuration, will be converted into REST routes at the plmn-egress-gateway microservice during startup. These routes can be modified directly using REST APIs.

SbiRouting Error Action Sets configuration

URL : `http://<config-mgr-svc>:<config-mgr-port>/sepp/nf-common-component/v1/egw/plmn/sbiroutingerroractionsets`

Description: Action sets are configured for the routes to specify the actions to be taken if an error is received from the peer.

Method: GET/PUT

Body

```
[{"id":"nrf_action_0","action":"reroute","attempts":2,"blacklist":
{"enabled":false,"duration":60000}}]
```

Error Action Sets for alternate routing mentioned in the Helm configuration will be converted to REST routes at plmn-egress-gateway microservice at startup. User can change these routes directly through REST APIs.

For more information, refer "Egress Gateway REST API" section in the *Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

3.2.9 Integrating SEPP with 5G Network Intelligence Fabric (5G NIF)

To route traffic to a Network Function, SEPP has traditionally relied on configurations or destination headers found in incoming SBI requests. For integration with the customized 5G Network Intelligence Fabric (5GNIF), SEPP must now discover this custom NF through the NRF, which holds this information. Once discovered, SEPP uses all existing routing mechanisms (such as alternate routing) to direct traffic to the identified 5GNIF instance. Additionally, SEPP is required to send copies of error messages (triggered by countermeasures or failed checks) to the 5GNIF for analytics purposes.

The 5GNIF, which provides value-added services, also requires the ability to analyze errors generated by SEPP microservices, particularly those caused by security mechanisms such as SCM (Security Countermeasures), TH (Topology Hiding) or Mediation. To support this, any rejected message at SEPP will be encapsulated in a predefined format and forwarded to 5GNIF for analytical purposes.

5G NIF is an inline network function and resides in the home network. All requests and responses to home network functions will pass through the 5G NIF for roaming scenarios.

Feature Overview

The 5G Network Intelligence Fabric (5GNIF) is a vital component, enabling essential capabilities such as Steering of Roaming (SoR), mobility management, and threat intelligence.

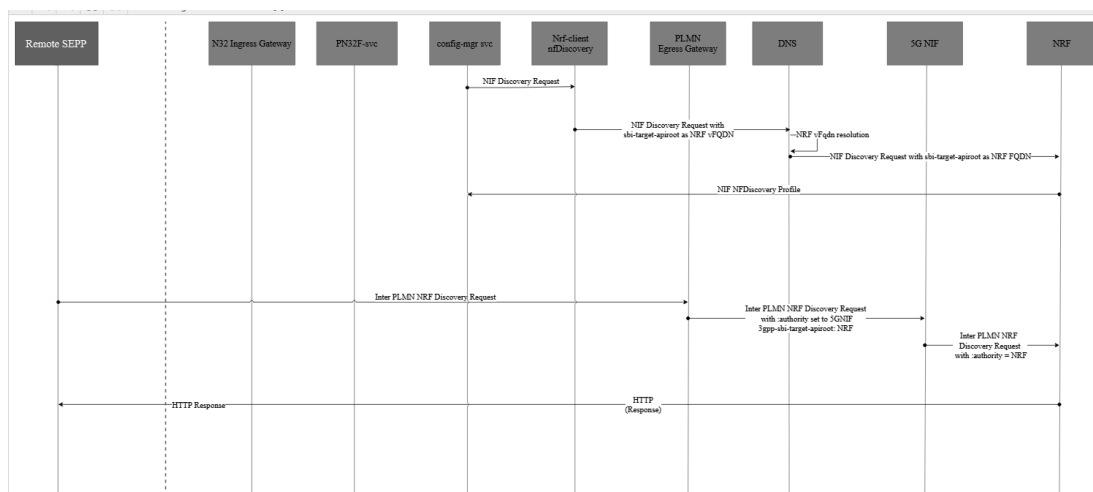
These features ensure that subscribers enjoy seamless, secure connectivity when accessing partner networks, delivering both a superior user experience and strong protection against evolving threats.

Below are the key functionalities enabled through 5GNIF integration:

Enabling NIF

When SEPP needs to route traffic, it begins by querying the NRF (Network Repository Function) to discover the 5GNIF. Once the 5GNIF is identified, SEPP manages incoming N32f/SBI traffic by setting routing headers—such as scheme, authority, and path, to ensure all messages pass securely and efficiently through the 5GNIF. This routing ensures secure and efficient message handling in both outbound and inbound communications.

Figure 3-13 Enabling NIF

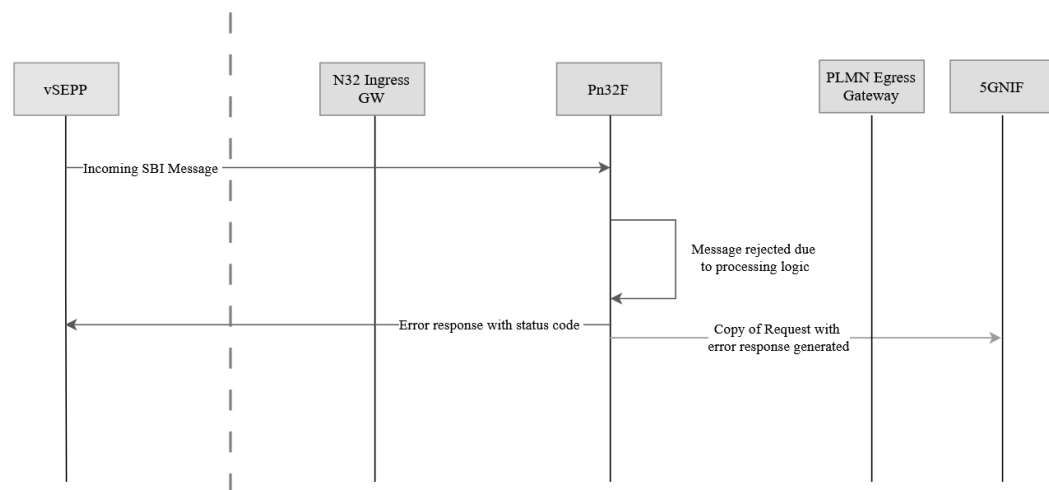


Error message forwarding to 5GNIF

When features like Security Countermeasures (SCM), Mediation, or Topology Hiding (TH) are enabled, SEPP performs strict validation on incoming SBI requests. If a request fails these checks, for example, due to an invalid header, incorrect URI, or malformed message body, SEPP rejects the request using a predefined error code and response.

In deployments where the 5GNIF is integrated, SEPP also sends a copy of both the rejected message and its corresponding error response to the 5GNIF. This enables the 5GNIF to analyze these messages for analytics, diagnostics, and operational optimization, supporting smarter decision-making and improved network performance.

Figure 3-14 Error message forwarding to 5GNIF



Detailed Description

To integrate 5GNIF with SEPP and enable dynamic discovery of 5GNIF instances, users must first enable the integration feature. Once enabled, the config-mgr-svc service periodically sends an NFDisccovery request (`/nnrf-disc/*`), with requester-nf-type set to SEPP and target-nf-type set to CUSTOM_5GNIF. Optionally, users can include a preferred-locality query parameter to prioritize specific regions.

This request is passed to the nrf-client-nfdisccovery microservice, which manages retrieving the configuration NRF (Network Repository Function). If a virtual FQDN is configured, the microservice first consults the alternate-route service to perform DNS resolution.

After successfully resolving the FQDN, the nrf-client-nfdisccovery forwards the request to the PLMN Egress Gateway, which selects the appropriate route based on the request and directs it to the correct NRF instance.

The NRF then returns an NFProfile, completing the discovery and enabling SEPP to dynamically configure routing to the available 5GNIF instances.

The NRF responds with an NFProfile response.

Example for NFProfile for NIF

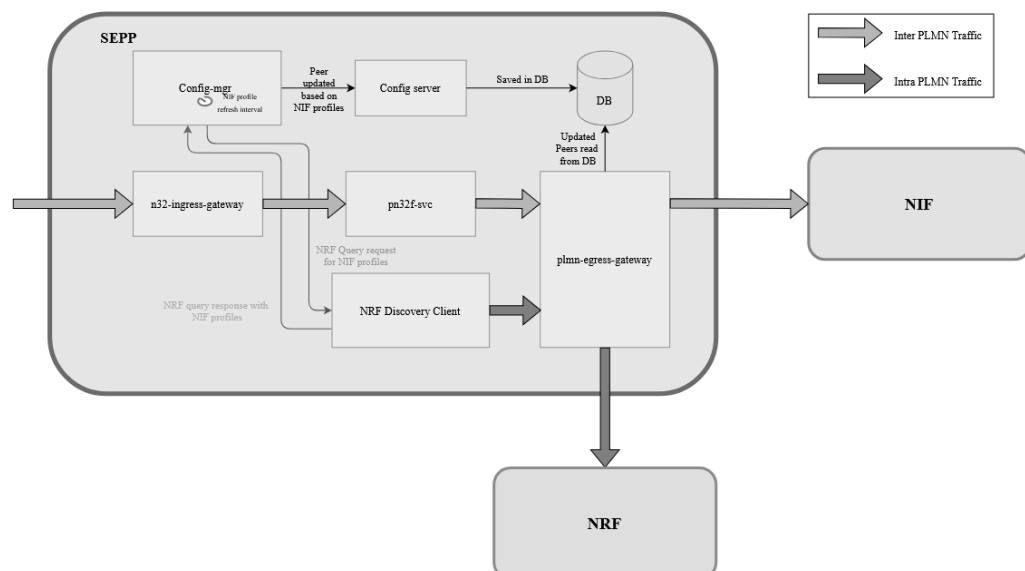
```

{
  "nfInstanceId": "cd2c976c-ba95-4d45-b71d-d68a3e761885",
  "nfType": "CUSTOM_5GNIF",
  "nfStatus": "REGISTERED",
  "heartBeatTimer": 30000,
  "plmnList": [
    {
      "mcc": "311",
      "mnc": "480"
    }
  ],
  "fqdn": "5gnif.mnc480.mcc311.3gppnetwork.org",
  "priority": 1,
  "capacity": 1000,
  "load": 10,
  "locality": "SouthLake",
  "customInfo": {
    "port": 8000,
    "scheme": "http"
  },
  "nfSetIdList": [
    "set001.nifset.5gc.mnc480.mcc311"
  ]
}

```

NIF Instance Selection and Traffic Routing

The following diagram provides a high level view of NIF Instance selection and traffic routing process:

Figure 3-15 NIF Instance Selection and Traffic Routing

NIF Selection and Routing Logic

After receiving one or more NFProfile responses from the NRF, the config-mgr-svc selects the appropriate NIF (Network Intelligence Fabric) instances based on the priority field in each profile. NIF peers and their corresponding routes are then established using a primary, secondary, and, tertiary hierarchy to ensure high availability and failover support.

Peer configuration is based on the following logic:

- If multiple NFProfiles are received, the priority field is used to determine the order of peers in the peer set.
- The fqdn field in the profile is used to populate the host field in the peer configuration.
- Port information is extracted from the customInfo block:
 - If the scheme is HTTP and no port is specified, port 80 is used by default.
 - If the scheme is HTTPS, the NFProfile is rejected, and an error is logged, as HTTPS is not supported for NIF peer connections.

Once the peers are configured, this information is forwarded to the PLMN Egress Gateway.

Traffic Routing via PLMN Egress Gateway

When an SBI request is received, the PLMN Egress Gateway selects the appropriate route to the NIF, starting with the primary peer:

- If the primary peer is unavailable or down, traffic is automatically rerouted to the secondary peer.
- If both primary and secondary are unavailable, the traffic is sent to the tertiary peer.

This hierarchical fallback mechanism ensures continued service availability and reliability.

Handling NIF Discovery Failures

If SEPP fails to discover any NIF instance from the NRF, or if all discovered instances transition into a SUSPENDED or UNDISCOVERABLE state, then all incoming SBI traffic is rejected with a configured error code.

Once the NIF instances become discoverable again, SEPP resumes normal processing of SBI traffic without requiring manual intervention.

Routing Information:

The following table contains summary for routes created for each event:

Table 3-15 Routing Information

S.No.	Event	Routes Present
1	Fresh installation	1. default_route
2	NIF feature enabled/NIF not discovered	1. nrf_route 2. nif_reject
3	NIF Discovered	1. nrf_route 2. nifPeer
4	NIF Disable	1. default_route

Table 3-15 (Cont.) Routing Information

S.No.	Event	Routes Present
5	NFDiscovery returns 200 OK Empty for NIF profile.	1. nrf_route 2. nif_reject
6	NRF change in NIF instances.	Only Peer Configuration change, No route change
7	Cat-3 feature is enabled when NIF is enabled (previous location or velocity).	1. nrf_route 2. nifPeer 3. cat3_udr_route 4. cat3_udm_route

Rejected Message Handling and Forwarding to 5GNIF

When features such as Security Countermeasure (SCM), Topology Hiding (TH), or Mediation are enabled, or if any SAN validation failures occur, SEPP performs strict validation on all incoming SBI requests. If a request contains invalid information in the header, URI, or body, SEPP rejects it using a configured error code and response.

If the 5GNIF has already been discovered in the deployment, SEPP also generates a copy of the rejected request and the corresponding error response. This copy is forwarded to the 5GNIF, which then processes the information for analytics, diagnostics, and other operational insights.

Managing Integration of SEPP with 5G NIF

This feature can be enabled by using the Helm parameters, CNC Console, or REST API.

Enable using Helm Parameter

Set the `sepp.headerAbsentPredicate` parameter to `true` in the PLMN Egress Gateway section of `ocsepp_custom_values_<version>.yaml` file.

```
sepp.headerAbsentPredicate: true
```

Set the `nif.enableNif` parameter to `true` in the Config Manager section of `ocsepp_custom_values_<version>.yaml` file.

```
nif.enableNif: true
```

For more information about the Helm parameters, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Enable using REST API**Enable the NIF feature:**

Set `enabled` to `true` in `/sepp-configuration/v1/nif/msg-copy/options` REST API.

```
curl 'http://<config-mgr-svc>:<port>/sepp-configuration/v1/nif/msg-copy/options' -XPUT -H 'Content-type: application/json' -d '{"enabled": "true"}'
```

Enable the Rejected Message copy feature:

Set `enabled` to `true` in `/sepp-configuration/v1/nif/msg-copy/options` REST API.

```
curl 'http://<config-mgr-svc>:<port>/sepp-configuration/v1/nif/msg-copy/
options -XPUT -H 'Content-type: application/json' -d '{"enabled" : "true"}'
```

For more information about API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Enable using CNC Console**Enable the NIF feature:**

1. From the left navigation menu, navigate to **SEPP** and then click **NIF**.
2. Click **NIF Option** under **NIF**. The **System Option** page appears at the right pane.
3. Click **Edit** icon to modify the Option. The **Edit System Option** page appears.
4. Set the **Enable NIF route selection** to `True`.

Enable the Rejected Message copy feature:

1. From the left navigation menu, navigate to **SEPP** and then click **NIF**.
2. Click **Message Copy Option** under **NIF**. The **System Option** page appears at the right pane.
3. Click **Edit** icon to modify the Option. The **Edit System Option** page appears.
4. Set the **Enable Error Message Copy** to `True`.

For more details about CNC Console configurations, see *Configuring SEPP using CNC Console* section in the *Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide*.

Configure

You can configure the feature using CNC Console, REST API and CNC Console.

- **Configure using REST API:** Perform the feature configurations as described in NIF REST APIs in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- **Configure using CNC Console:** Perform the feature configurations as described NIF section in [Configuring SEPP using CNC Console](#).

Observe

Following are the Integrating SEPP with 5G Network Intelligence Fabric (5G NIF) feature specific metrics:

- `ocsepp_nif_discovery_requests_total`
- `ocsepp_nif_discovery_responses_total`
- `ocsepp_nif_registration_status`
- `ocsepp_pn32f_nif_error_copy_requests_total`
- `ocsepp_pn32f_nif_error_copy_responses_total`

Following are the Integrating SEPP with 5G Network Intelligence Fabric (5G NIF) feature specific KPIs:

- Discovery Requests Sent Towards NRF for NIF
- Response Received from NRF for NIF Discovery
- Rejected Message Copied towards NIF
- Responses Received from NIF for Copied Messages

Following is the Integrating SEPP with 5G Network Intelligence Fabric (5G NIF) feature specific alert:

- `configMgrNoHealthyNIFAlert`

For more information on Support for Originating Network Id Header Validation, Insertion, and Transposition feature related SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.10 TLS 1.3 Support for Kubernetes API

Overview

In a Kubernetes-based 5G Core deployment, SEPP regularly interacts with the Kubernetes API server to retrieve the secrets and ConfigMap information. With this feature, SEPP supports TLS 1.3 for secure communication with the Kubernetes API server.

All pods can communicate with the Kubernetes API server using either TLS 1.3 or TLS 1.2. An application TLS server configured with TLS 1.3 requires a compatible CNE version that supports TLS 1.3.

This feature can be enabled or disabled using the `tlsEnableForKubeApiServer` parameter as follows:

- When you enable the `tlsVersionSupportForKubeApiServer.enabled` parameter, SEPP microservices communicate with the Kubernetes API server using only TLS 1.3.
- When you disable the `tlsVersionSupportForKubeApiServer.enabled` parameter, SEPP microservices communication with the Kubernetes API server will be through TLS 1.2.

Note

Enabling the TLS 1.3 Support for Kubernetes API Server Communication feature does not affect external TLS communication. However, you must update the `featureSecretsForKubeApiServer` parameter with the secrets used for external communication. The feature is disabled by default.

Managing TLS 1.3 Support for Kubernetes API Server Communication

Enable and Configure

Perform the following Helm configuration to enable this feature:

1. Open the `ocsepp_custom_values_<version>.yaml` file.
2. Set the value of `global.tlsVersionSupportForKubeApiServer.enabled` to `true` under the Global parameters section.
3. Set the value of `global.tlsVersionSupportForKubeApiServer.kubeApiServerTlsVersion` to `TLSv1.3`. This parameter defines the TLS version to be used. Here, the value must be either `TLSv1.3` or `TLSv1.2`. The comma-separated values are not supported.
4. Configure the value in `global.tlsVersionSupportForKubeApiServer.cipherSuites`. This parameter defines the CipherSuites used for the configured TLS version. If `TLSv1.2` is configured, comment out the first 3 `TLSv1.3` ciphers and uncomment the other five `TLSv1.2` ciphers. The list of CipherSuites is as follows:

```
- TLS_AES_256_GCM_SHA384
  - TLS_AES_128_GCM_SHA256
  - TLS_CHACHA20_POLY1305_SHA256

# - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
# - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
# - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
# - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
# - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
```

5. Configure the value in `pn32f-svc.tlsVersionSupportForKubeApiServer.cipherSuites`, `n32f-svc.tlsVersionSupportForKubeApiServer.cipherSuites`, `config-mgr-svc.tlsVersionSupportForKubeApiServer.cipherSuites`. This parameter defines the CipherSuites used for the configured TLS version.
 - a. If `TLSv1.3` is configured then the value must be one of the following:
 - i. `TLS_AES_256_GCM_SHA384`
 - ii. `TLS_AES_128_GCM_SHA256`
 - iii. `TLS_CHACHA20_POLY1305_SHA256`
 - b. If `TLSv1.2` is configured then value must be one of the following:
 - i. `TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384`
 - ii. `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`
 - iii. `TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256`
 - iv. `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`
 - v. `TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256`
6. Create the required secret and configure the values in the `tlsVersionSupportForKubeApiServer.featureSecrets` parameter under the Ingress and Egress Gateway parameter section as follows:

```
tlsVersionSupportForKubeApiServer:
  enabled: *kubeApiFlag
  kubeApiServerTlsVersion: *kubeApiVersion
```

```
cipherSuites: *kubeApiCiphers
# List of all the secrets to be volume mounted
featureSecrets: # When the above feature flag is enabled, configured
secrets in all the features shall be configured below, otherwise helm
upgrade/install will fail.
  - ocsepp-plmn-secret
```

Note

- a. Ingress Gateway and Egress Gateway must list all the secrets required for the external communication.
- b. SEPP uses different secrets for PLMN and N32 Gateways
 - i. For PLMN gateway, the value of featureSecrets must be - ocsepp-plmn-secret
 - ii. For N32 gateway, the value of featureSecrets must be - ocsepp-n32-secret
 - iii. If OCNADD is configured, then the following message-copy secret must be updated in featureSecrets: - message-copy-secret

7. Save the file.
8. Run the following command to enable the feature during Helm installation or upgrade:
For Helm installation:

```
helm install <release_name> <helm_chart> -f
<ocsepp_customized_values.yaml> --namespace <namespace-name>
```

For Helm upgrade:

```
helm upgrade <release_name> <helm_chart> -f
<ocsepp_customized_values.yaml> --namespace <namespace-name>
```

Helm Parameters

Parameter	Description	Details
<code>global.tlsVersionSupportForKubeApiServer.enabled</code>	This is a mandatory parameter. This parameter enables TLS 1.3 for communication between SEPP services and Kubernetes API server. When the parameter is set to true, SEPP Services communicate with the Kubernetes API server using configured version in <code>global.tlsVersionSupportForKubeApiServer.kubeApiServerTlsVersion</code> When the parameter is set to false, SEPP Services communicate with the Kubernetes API server using only TLS 1.2.	Data Type: Boolean Range: true or false Default Value: false
<code>global.tlsVersionSupportForKubeApiServer.kubeApiServerTlsVersion</code>	This is a mandatory parameter. This parameter defines the TLS version to be used.	Data Type: String Range: TLSv1.3 or TLSv1.2 Default Value: TLSv1.3

Parameter	Description	Details
<code>tlsVersionSupportForKubeApiServer.cipherSuites</code>	This is a mandatory parameter. This parameter list of CipherSuites used for the configured TLS version. Using this parameter, you can configure the required CipherSuites.	Data Type: String Range: If kubeApiServerTlsVersion is set to TLSv1.3, then use the following ciphers - TLS_AES_256_GCM_SHA384 - TLS_AES_128_GCM_SHA256 - TLS_CHACHA20_POLY1305_SHA256 If kubeApiServerTlsVersion is set to TLSv1.2, then use the following ciphers - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 Default Value: TLS_AES_256_GCM_SHA384,TLS_AES_128_GCM_SHA256,TLS_CHACHA20_POLY1305_SHA256
<code>tlsVersionSupportForKubeApiServer.featureSecrets</code>	This is a mandatory parameter. This parameter lists the secrets for external communication.	Data Type: String Range: N32 Gateways ocsepp-n32-secret PLMN Gateways ocsepp-plmn-secret

Observe**Metrics**

The following metrics are available for this feature:

- `oc_ingressgateway_ip_addresses_fetch_failure`
- `oc_egressgateway_ip_addresses_fetch_failure`
- `oc_certificatemanagement_tls_certificate_info`
- `oc_certificatemanagement_tls_secret_status`

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.11 LCI and OCI Headers in 5G Architecture

Background

In 5G architecture, network overloads can happen frequently because of the high volume of data exchange between producer and consumer Network Functions (NFs). This communication includes many messages and notifications, making efficient load balancing critical to prevent network failures. To manage this, it's important that consumer NFs or the Security Edge Protection Proxy (SEPP) are quickly notified in outgoing responses when a producer NF or SEPP is approaching an overload condition. This early notification allows consumer NFs or SEPP to take preventive actions and reduce the risk of service disruption.

Overview

This feature introduces LCI and OCI Headers in SEPP. These headers are designed to enhance the exchange of messages and notifications between producer resources and consumer Network Functions (NFs) within the 5G network, allowing an effective network management. These headers provide essential load and overload information for consumer NFs/SEPPs to optimize traffic distribution and take proactive measures during network overload scenarios.

This feature significantly enhances network efficiency, contributing to a more reliable and responsive network infrastructure. The headers are integrated into outgoing responses, based on load levels at the Ingress Gateway. Serving as communication tools, they allow Network Functions (NFs)/SEPPs to share crucial load information, ensuring an architecture where 5G Core Network remains stable and high performing even during heavy load conditions.

LCI Header

The LCI header shares the current load status of the SEPP. This helps the consumer NF or consumer SEPP manage and balance network traffic more effectively. The header is called `3gpp-Sbi-Lci`. When the conditions for generating LCI Header are met, the SEPP adds this header at the SEPP-level scope. The LCI header comprises overall load related information such as the timestamp of load data generation, the current load of the SEPP, and the scope of the load information.

Example of LCI Headers

SEPP Scope LCI Header:

3gpp-Sbi-Lci: Timestamp: "Tue, 04 Apr 2021 08:36:42 GMT"; Load-Metric: 25.0%; SEPP-FQDN: sepp1.example.com

- **SEPP scope LCI header:** The Ingress Gateway has a configurable polling interval for retrieving service-level load information from perf-info. It conducts load aggregation for supported SEPP services at the SEPP level, utilising a straightforward averaging logic. The determination of supported services relies on the NF service to instance ID mapping, carried out through Helm. When this feature is enabled over plmn-ingress-gateway, SEPP core service cn32f-svc is considered for load computation and when this feature is enabled over n32-ingress-gateway, SEPP core service pn32f-svc is considered for load computation. The load aggregation needs to be configured in NF service to instance ID mapping for both n32-ingress-gateway and plmn-ingress-gateway.

In SEPP, load mapping is specifically supported only for the cn32f-svc and pn32f-svc services. This means SEPP handles load information only for these two services. For example, when the Ingress Gateway receives load data for one of these services, it includes that information in the LCI or OCI header, specifically under the SEPP scope LCI header. The Ingress Gateway uses a configurable polling interval to regularly collect service-level load data from perf-info, which is then used to generate these headers.

OCI Header

The OCI header communicates information about overload conditions. This header sends the current overload information so that the Consumer NF or peer SEPP can reduce the traffic towards the overloaded NF. `3gpp-Sbi-Oci` denotes OCI Header. When the conditions for generating OCI Header are met, the SEPP includes this header at the SEPP scope.

The OCI header includes important details such as:

- The time the overload was detected
- The type of overload (for example, resource exhaustion)
- The recommended action, like reducing the number of incoming requests.

Both the LCI and OCI headers are incorporated in HTTP response messages without triggering additional signaling, ensuring a more efficient communication process. Here is how they help:

- The LCI header conveys the overall load of a SEPP, assisting in decisions regarding the acceptance or rejection of new requests to prevent further overload.
- In contrast, the OCI header communicates specific overload conditions, helping consumer NFs/consumer SEPP take informed actions to mitigate these conditions.
- The LCI and OCI headers complement each other, allowing a consumer NF/SEPP to reduce the number of requests it sends to other SEPPs in response to an OCI header, even if its overall load is not yet overloaded.
- This proactive measure prevents overload conditions from spreading.

Example of OCI Headers

SEPP Scope OCI Header:

Overload Control Information sent by an SEPP:

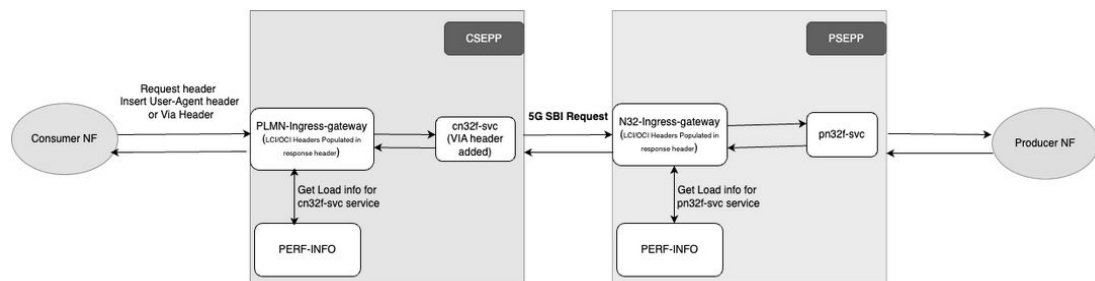
3gpp-Sbi-Oci: Timestamp: "Tue, 04 Feb 2020 08:49:37 GMT"; Period-of-Validity: 120s;
Overload-Reduction-Metric: 25.0%; SEPP-FQDN: sepp1.example.com

Note

For both LCI and OCI headers, load reporting is based on two factors: `localLciHeaderValidity` and `loadThreshold`. The `localLciHeaderValidity` determines how long the load information is valid. Once the validity period expires, the `loadThreshold` value is no longer considered. The threshold value will be validated only within the validity period.

Design and Architecture

The following diagram provides a high level view of LCI and OCI header support in consumer and producer SEPP.

Figure 3-16 Architecture

In the above diagram, the LCI/OCI Header support feature is enabled on the consumer SEPP over the PLMN-Ingress-gateway and towards the producer SEPP over the n32f-svc gateway.

The request and response flow:

1. Request Flow:

- The request is initiated from the consumer NF at the Initiator Side.
- The request passes through the Initiator SEPP, reaching the Responder SEPP, and finally the producer NF.

2. Response Flow:

- Once the producer NF processes the message, a response is generated and sent back to the consumer NF.
- The response passes through the Responder SEPP and then the Initiator SEPP.

3. LCI/OCI Header Integration:

- At the producer SEPP, the LCI and OCI headers are added to the outgoing response, carrying load information for the Responder SEPP. This load information is essential for the consumer SEPP to help with load balancing and traffic adjustments.
- As the response passes through the Initiator SEPP, the Initiator SEPP may add (depending on configurations) its load information in the LCI Header and overload reduction metrics in the OCI Header with SEPP Scope.
- The LCI/OCI headers added by the Initiator SEPP can be utilized by the consumer NF for load adjustments.

Example for the LCI header (Entire message body is not shown here):

```
HTTP/2 201
  content-length: 3953
  content-type: application/3gppHal+json
  3gpp-Sbi-Lci: Timestamp: "Fri, 07 Mar 2025 11:16:12 UTC"; Load-
Metric: 14.0%; SEPP-FQDN:
  sepp1.example.com
```

Example for the OCI header (Entire message body is not shown here):

```
HTTP/2 201
  content-length: 3953
  content-type: application/3gppHal+json
  3gpp-Sbi-Oci: Timestamp: "Fri, 07 Mar 2025 11:16:12 UTC"; Period-of-
Validity: 5s;
  Overload-Reduction-Metric: 5.0%; SEPP-FQDN: sepp1.example.com
```

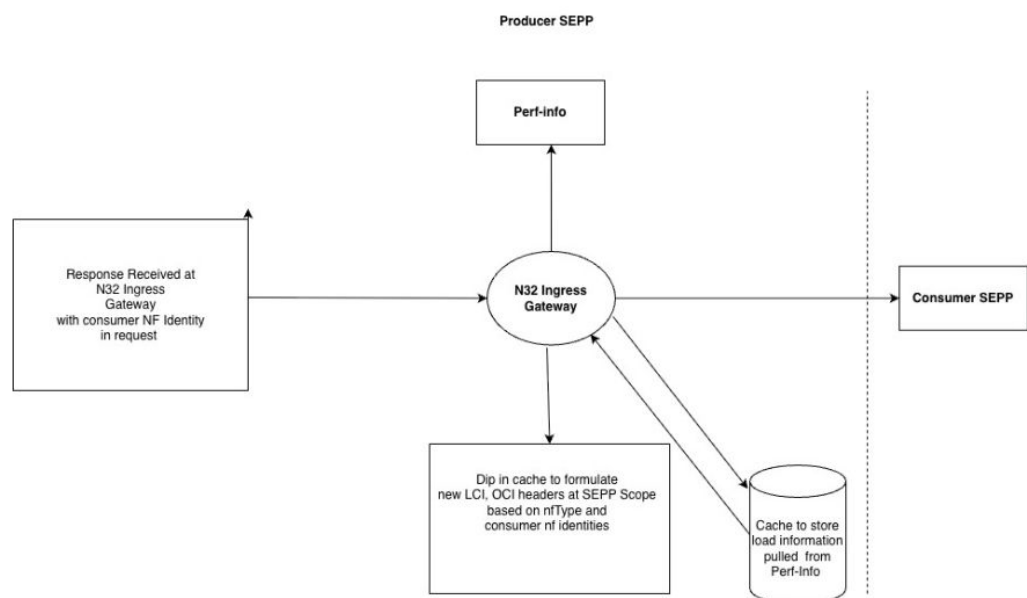
Note

When reporting OCI header, the LCI header may also be included.

This feature can be enabled either on the n32-ingress-gateway, PLMN-Ingress-gateway, or on both gateways. Load computation for SEPP is carried out by the ingress gateway using the perf-info load collection functionality. This approach helps in ensuring that load information is shared effectively between SEPPs.

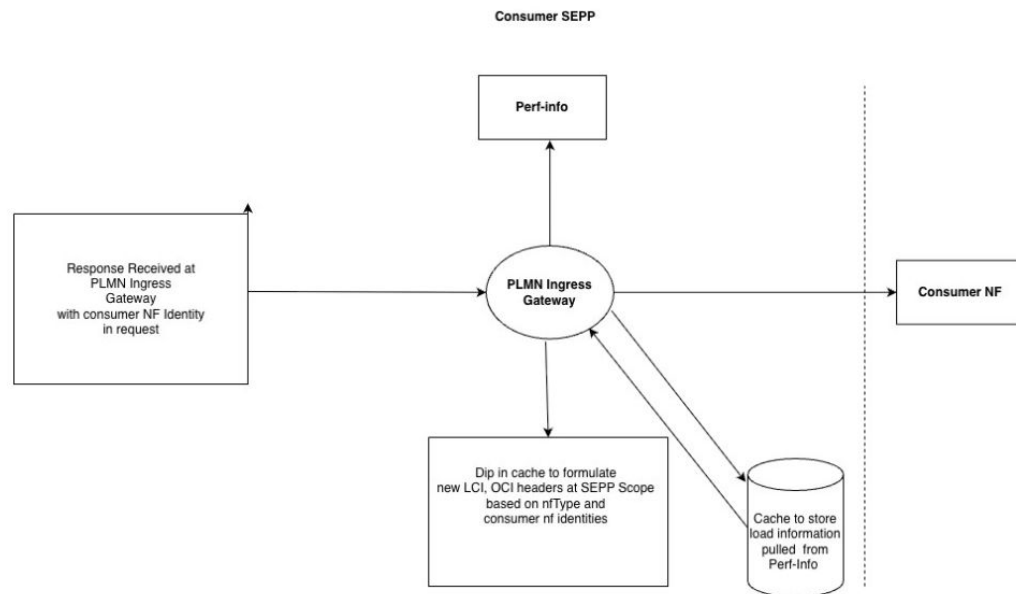
The following diagrams depict the LCI and OCI workflow:

Figure 3-17 At n32-ingress-gateway



The diagram above illustrates the workflow of LCI and OCI headers when enabled at the n32-ingress-gateway of the SEPP. The n32-ingress-gateway generates the LCI and OCI headers based on load data received from the perf-info service associated with the pn32f-svc microservice within the SEPP scope.

Figure 3-18 At PLMN-Ingress-gateway



The diagram above depicts the workflow of LCI and OCI headers when enabled at the PLMN Ingress Gateway of the SEPP. The PLMN Ingress Gateway generates the LCI and OCI headers using load information received from the perf-info service related to the cn32f-svc microservice within the SEPP scope.

LCI Headers and Their Workflow

LCI Headers are dedicated to providing real-time load information, using the identifier "3gpp-Sbi-Lci." This information provides consumer NFs with the essential knowledge needed to make informed decisions about distribution of network traffic efficiently. When specific conditions are met, LCI headers are included in the messages and notifications (notifications will be supported in future releases; not supported yet), extending the scope to network function, backend service, or both. The condition states that user-agent/via/oAuth header should be present in the request so that LCI is included in the response header.

When SEPP receives a request from an NF (for example, AMF), Ingress Gateway at SEPP checks if LCI is enabled. If it is not enabled, LCI header is not added as part of response. If it is enabled, LCI header is computed as follows:

- Ingress Gateway fetches the previous LCI Header from the cache.

Note

If no data is fetched regarding LCI Header from the cache, it indicates that LCI Header was not sent more than N seconds ago. In this case, a fresh LCI Header with the current information is included.

- If the current load metric is within the load metric threshold, LCI Header is not included in the message.
- If it is beyond the threshold, LCI Header is included in the message. The LCI Header information is updated in the coherence cache for the current destination and further processing is done.

Table 3-16 LCI header fields

Fields	Description
Load Control Timestamp	Human readable timestamp (date and time) indicating time when LCI header is sent.
Load Metric	Current load level for the scope of LCI, in terms of percentage, ranging from 0 to 100. 0 indicated 0% or no load, and 100 indicates 100.0% or maximum load.
Scope of LCI	The scope of LCI are nf Instance id, NF Set ID, NF Service Instance ID or NF Service Set ID. This scope depends on the scope of load info received from perf-info. The same scope is conveyed to the consumer NF.

OCI Headers and Their Workflow

OCI Headers operate under the identifier "3gpp-Sbi-Oci" and are designed to convey current overload information. This information is critical for consumer NFs, enabling them to take preemptive actions to reduce the traffic directed toward overloaded NFs. Like LCI headers, OCI headers are included in messages and notifications when specific conditions are met, spanning the NF scope, the backend service, or both.

When SEPP receives a request from an NF (for example, AMF), Ingress Gateway at SEPP checks if OCI is enabled. If it is not enabled, OCI header is not added as part of response. If it is enabled, OCI header is computed as follows:

- Ingress Gateway checks if the NF is overloaded as per the configured range. Then, it fetches the previous OCI Header sent from the cache.

Note

If no data is fetched regarding OCI Header, however, the NF is overloaded, even then OCI Header is prepared.

- If the overload is different from the previously computed value, the Ingress Gateway changes the overload reduction metric value and prepares a new OCI Header.
- If the overload is the same as the previous reported value, but the period of validity has expired, the OCI Header is included in the message.
- If the period of validity has not expired, the Ingress Gateway continues with further processing. The OCI Header information is updated in the coherence cache for the current destination and further processing is done.

Overload Control based on OCI Header

Overload information need to be sent from producer NF to consumer NF in "3gpp-Sbi-Oci" header. The fields included in this header are described below:

Table 3-17 OCI header fields

Fields	Description
Overload Control Timestamp	Human readable timestamp (date and time) indicating time when OCI header is sent.
Overload Reduction Metric	Indicates the percentage of traffic reduction that this NF expects from the receiver of OCI Header. The value ranges from 0-100, 0 indicating no traffic reduction towards sender of OCI.
Overload Control Period of Validity	Indicates a timer within which the information conveyed in OCI Header shall be considered valid (unless overridden by a newer OCI Header)
Scope of OCI	The scope of OCI can be one of below: - NFInstanceID or NF-SET or NF-Service-instance or NF-Service-Set (if NF is a producer) - NFInstanceID or NF-SET or NF-Service-instance or NF-Service-Set or Call-Back URI (if NF is a consumer) The scope depends on the scope of load info received from perf-info. The same scope is conveyed to the destination NF.

Load Computation

The Ingress Gateway features a configurable polling interval used to retrieve service-level load information from perf-info. The Ingress Gateway conducts load aggregation for supported SEPP services at the NF (Network Function) level, using a straightforward averaging logic. The supported services are decided based on the NF service to instance ID mapping, which is done through Helm. For example, in SEPP, the mapping is done for cn32f-svc and pn32f-svc services. Hence, only these two services are supported in SEPP.

For instance, when the Ingress Gateway receives two pieces of load information for a particular service, it adds these values together and then divides the sum by two to calculate the average load at the NF level.

Peer Identity

If multiple fields are available to extract peer identity, the priority to extract this identity will be in below order:

1. OAuth token (1st priority)
2. User Agent header (2nd priority)
3. VIA header (3rd priority)

When multiple headers are present at the N32 or PLMN Ingress Gateway, the consumer NF identity must be determined by extracting it from the headers according to the priority order outlined above.

The **nf Instance id** is retrieved from these headers and serves as the key identifier for the consumer NF. Examples of how to extract this ID from various headers are as follows:

- **OAuth Token Header**
Authorization → The nf Instance id is extracted by decoding the token,
Example:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiI2ZmFmMWJiYy02ZTRhLTQ0NTQtYTUwNy1hMTRlZjhlMWJjMTEiLCJzdWIiOiJmZTdka0tkYyYi0wNTQxLTRjN2Q0YWI4NC1jNmQ3MGIXY
```

```
jAxYjEiLCJhdWQiOiJTTUYiLCJzY29wZSI6Im5zbWYtcGR1c2Vzc2lvbiIsImV4cCI6MTY0NDQ5
MjExOH0.xGud7dIwEugyXV-I8M-xld90Ar_rKbB3y4_SWVVAwHyb6DxtWUz5uB-
KtXwozbRJorMnKF6rwz5D0G6VAm9WLw
```

- **User-Agent Header**
User-Agent → Format: NFType-NFInstanceId fqdn

Example:

```
SEPP-9faf1bbc-6e4a-4454-a507-aef01a101a06
sepp1.example.com
```

- **Via Header**
Via → Format: HttpVersion fqdn

Example:

```
via: 2.0 sepp1.example.com
```

Note

Since the nf Instance id is not included in the Via header, the FQDN is used as the key identifier in this case.

Validity Period

If the same peer sends multiple requests within the validity period and there is no breach of configured thresholds, then SEPP will not add LCI or OCI headers.

Use Cases

- Currently, LCI and OCI headers are only supported at the Ingress Gateway.
- The Initiator SEPP can use the LCI header to share its load information with the consumer NF. This helps the consumer NF decide whether to send new inter-PLMN requests to other available SEPPs, if any, or reduce the number of requests sent to the Initiator SEPP.
- The Responder SEPP can use the LCI header to share its load information with the Initiator SEPP. This helps the Initiator SEPP decide whether to route new requests to other available peer SEPPs, if any, or to reduce the number of requests sent to the Responder SEPP.
- On the other hand, the SEPP can use the OCI header to inform the consumer NF or Initiator SEPP about an overload condition. This allows them to reduce the number of requests being sent to the overloaded SEPP.

Overload Reduction Metric

The Overload-Reduction-Metric value (which indicates the percentage of traffic the OCI sender wants the receiver to reduce) is calculated based on the current load of the SEPP's core microservices.

When this feature is enabled on the n32-ingress-gateway, it monitors the load of the pn32f-svc microservice to report the metric to the consumer SEPP. Similarly, when enabled on the plmn-Ingress-Gateway, it monitors the load of the cn32f-svc microservice for the same purpose.

The following tables indicate the overload config range and Overload-Reduction-Metric:

Table 3-18 Feature Enabled on n32-ingress-gateway

SEPP core service monitored for load/overload condition	overload config range	overload reduction metric	Details
pn32f-svc	minor: "[60-70]" major: "[70-80]" critical: "[80-100]"	minor: 5 major: 10 critical: 30	When SEPP is acting as a Responder SEPP during live traffic, it monitors the CPU load of the pn32f-svc microservice. If an overload is detected, it is reported using the OCI header with an appropriate Overload-Reduction-Metric based on the load level: 60.0% to 70.0% load → <i>Minor overload</i>: 5.0% traffic reduction is requested. 70.0% to 80.0% load → <i>Major overload</i>: 10.0% traffic reduction is requested. 80.0% to 100.0% load → <i>Critical overload</i>: 30.0% traffic reduction is requested. 0% to 60.0% load → <i>No overload</i>: OCI header is not sent.

Table 3-19 Feature Enabled on plmn-ingress-gateway

SEPP core service monitored for load/overload condition	overload config range	overload reduction metric	Details
cn32f-svc	minor: "[60-70]" major: "[70-80]" critical: "[80-100]"	minor: 5 major: 10 critical: 30	When SEPP acts as an Initiator SEPP during live traffic, it monitors the CPU load of the cn32f-svc microservice. If an overload is detected, it is reported using the OCI header with the following Overload-Reduction-Metric based on the load level: 60.0% to 70.0% load → <i>Minor overload</i>: 5.0% traffic reduction is requested. 70.0% to 80.0% load → <i>Major overload</i>: 10.0% traffic reduction is requested. 80.0% to 100.0% load → <i>Critical overload</i>: 30.0% traffic reduction is requested. 0% to 60.0% load → <i>No overload</i>: OCI header is not sent.

LCI and OCI Header Priorities

The SEPP must extract the consumer NF identity from one of the following request headers at the n32 or plmn ingress gateway:

1. oAuth token (1st priority)
2. User-Agent (2nd priority)
3. Via (3rd priority)

If multiple headers are available, the consumer NF identity should be extracted based on the priority order listed above.

The nf Instance id is extracted from these headers, and this Instance id serves as the key to identify the consumer NF. Below are examples of how to extract the nf Instance id:

oAuth Token Header

- Authorization: nf Instance id is extracted by decoding the following token:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiI2ZmFmMWJiYy02ZTRhLTQ0NTQtYTUwNy1hMTRlZjhlMWJjMTEiLCJzdWIiOiJmZTdkOTkyYi0wNTQxLTRjN2QtYWI4NC1jNmQ3MGIXYjAxYjEiLCJhdWQiOiJTTUYiLCJzY29wZSI6Im5zbWYtcGR1c2Vzc2lvbiIsImV4cCI6MTY0NDQ5MjExOjE0LmVudG90Ar_rKbB3y4_SWVVAwHyb6DxtWUz5uB-KtXwozbRJorMNkF6rwz5D0G6VAm9WLw
```

User-Agent Header

- User-Agent: NF Type and Instance ID are extracted in the format:

```
NFType-NFInstanceId fqdn SEPP-9faf1bbc-6e4a-4454-a507-aef01a101a06  
sepp1.example.com
```

Via Header

- Via: nf Instance id isn't part of the Via header, so the fqdn is used as the required key:

```
via: 2.0 sepp1.example.com
```

In this case, the fqdn (Fully Qualified Domain Name) from the Via header acts as the identifying key.

Managing LCI and OCI Headers

Enable

The LCI and OCI Headers feature can be enabled by performing the following Helm configurations globally.

Global Helm Configuration

User can enable LCI and OCI headers globally at the SEPP level for the gateways by setting specific configuration parameters:

- To enable them at the **n32-ingress-gateway** level, set:

```
seppLciEnabledN32Ingress = true
```

```
seppOciEnabledN32Ingress = true
```

- To enable them globally at the **plmn-ingress-gateway** level, set:

```
seppLciEnabledPlmnIngress = true
```

```
seppOciEnabledPlmnIngress = true
```

The following settings control whether LCI and OCI headers are included in messages handled by each gateway:

```
#####LCI/OCI header Global Values#####
#Enabling LCI over N32-Ingress
seppLciEnabledN32Ingress: &n32IngressLciEnable true

#Enabling OCI over N32-Ingress
seppOciEnabledN32Ingress: &n32IngressOciEnable true

#Enabling LCI over PLMN-Ingress
seppLciEnabledPlmnIngress: &plmnIngressLciEnable true

#Enabling OCI over PLMN-Ingress
seppOciEnabledPlmnIngress: &plmnIngressOciEnable true
#####
```

Configure

In the `perf-info` section, modify the parameter `configmapPerformance.prometheus` according to the prometheus service deployed in the cluster.

If the Prometheus UI has cluster-name appended to it, modify the parameter as follows:

```
http://occne-kube-prom-stack-kube-prometheus.occne-infra:80/<clustername>
```

Example:

```
http://occne-kube-prom-stack-kube-prometheus.occne-infra:80/cluster1
```

① Note

Perform a Helm upgrade after applying the above customizations, if SEPP has already been deployed.

n32-ingress-gateway Helm Configuration

OCI configuration

```
ociHeaderConfig:
  enabled: *n32IngressOciEnable
  validityPeriod: 5000 #(value in milliseconds)
  overloadConfigRange: #Note - minor, major and critical conditions
  should be broken down in buckets of range [0 to 100] only both inclusive for
  it to be a valid config
  minor: "[60-70]"
```

```

        major: "[70-80]"
        critical: "[80-100]"
    reductionMetrics:
        minor: 5 #(Possible values 1 to 9 both inclusive)
        major: 10 #(Possible values 5 to 15 both inclusive)
        critical: 30 #(Possible values 10 to 50 both inclusive)

    nfInstanceId: *nfInstIdRef #Producer nf Instance id
    nfType: "SEPP"
    nfFqdn: *nfFqdnRef

    svcToSvcInstanceIdMapping:
        - svcName: "pn32f-svc"
          serviceInstanceId: *nfInstIdRef

LCI configuration

    lciHeaderConfig:
        enabled: *n32IngressLciEnable
        loadThreshold: 40
        localLciHeaderValidity: 10000 #(value in milliseconds)

perf-info configuration

    perfInfoConfig:
        pollingInterval: 5000 #(value in milliseconds)
        serviceName: ""
        host: <release-name>-sepp-perf-info # it should be replaced by correct
release name
        port: 5905
        perfInfoRequestMap: "/load"

```

PLMN Ingress Gateway Helm Configuration

```

OCI configuration

    ociHeaderConfig:
        enabled: *plmnIngressOciEnable
        validityPeriod: 5000 #(value in milliseconds)
        overloadConfigRange: #Note - minor, major and critical conditions
should be broken down in buckets of range [0 to 100] only both inclusive for
it to be a valid config
        minor: "[60-70]"
        major: "[70-80]"
        critical: "[80-100]"
    reductionMetrics:
        minor: 5 #(Possible values 1 to 9 both inclusive)
        major: 10 #(Possible values 5 to 15 both inclusive)
        critical: 30 #(Possible values 10 to 50 both inclusive)

    nfInstanceId: *nfInstIdRef #Producer nf Instance id
    nfType: "SEPP"
    nfFqdn: *nfFqdnRef

```

```

svcToSvcInstanceIdMapping:
  - svcName: "cn32f-svc"
    serviceInstanceId: *nfInstIdRef

```

LCI configuration

```

lciHeaderConfig:
  enabled: *plmnIngressLciEnable
  loadThreshold: 40
  localLciHeaderValidity: 10000 #(value in milliseconds)

```

perf-info configuration

```

perfInfoConfig:
  pollingInterval: 5000 #(value in milliseconds)
  serviceName: ""
  host: <release-name>-sepp-perf-info #it should be replaced by correct
release name
  port: 5905
  perfInfoRequestMap: "/load"

```

Note

When configuring the minor, major, or critical overload ranges, make sure the ranges do not overlap. For example, overlapping ranges like 75–85 and 80–90 can cause incorrect behavior or results in overload detection and response.

For more information about the Helm parameters, see "Customizing SEPP" section in *Oracle Communications Cloud Native Core, SEPP Installation, Upgrade, and Fault Recovery Guide*.

Observe

Following are the LCI and OCI Headers specific metrics:

- oc_ingressgateway_headers_lci_total
- oc_ingressgateway_headers_oci_total

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following steps:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.12 SEPP Dashboard Support for Detecting Vulnerable Messages

In earlier releases, SEPP metrics could not be filtered using the originating PLMN ID, making it difficult to segregate and identify the source of messages. This feature enhances the metrics by enabling segregation based on PLMN ID, thereby improving operational efficiency in tracking message origins.

At CSEPP, the existing metric `ocsepp_cn32f_requests_total` has been enhanced to include `source_plmn_id` as a new dimension. If the source PLMN ID is not present in the incoming header fields, either `3gpp-sbi-originating-network-id` or `3gpp-sbi-asserted-plmn-id`, it is recorded with the value 'Unknown'.

At PSEPP, a new metric `ocsepp_originating_network_id_validation_success_total` has been introduced. This metric is updated only after successful Cat -2 header validation.

Managing SEPP Dashboard Support for Detecting Vulnerable Messages feature.

Prerequisites:

- The `3gpp-Sbi-Originating-Network-Id` shall be present in the ingress message, populated either by the VPLMN or by the SEPP.
- The Cat -2 Network ID Validation feature must be enabled.

Enable

This feature is enabled by default.

Observe

Following are the SEPP Dashboard Support for Detecting Vulnerable Messages feature specific metrics:

- `ocsepp_cn32f_requests_total` (existing metrics updated)
- `ocsepp_originating_network_id_validation_success_total` (new metrics)

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following steps:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.13 Security Countermeasure Features

The SEPP security countermeasure features are classified into the following:

- Stateless Countermeasure features
- Stateful Countermeasure features

3.2.13.1 Security Countermeasure Nonverbose Error Responses

Introduction

In earlier releases, SEPP's security countermeasure features (Cat-0, Cat-1, Cat-2, and Cat-3) provided detailed error messages explaining the reason for rejection and other information while rejecting an invalid request. The details in the error messages can weaken the effectiveness of the security countermeasures. Hence, the details should be reduced to prevent attackers from gaining useful information.

With the implementation of this feature, the user can configure whether the details in the error messages can either be detailed (verbose) or simplified (nonverbose). The error responses for the following features are enhanced:

- Cat-0 SBI Message Schema Validation Feature
- Cat-1 Service API Validation Feature
- Cat-2 Network ID Validation Feature
- Cat-3 Previous Location Check Feature

Note: Verbose error responses are disabled by default.

- When verbose error responses are disabled, attributes such as title, detail, and cause in the error response are filled with less detailed, user-configured values. By default, these attributes are set to "Rejected" for title, "Server Error" for detail, and "Unknown" for cause. Http status code remains same as earlier. Additionally, attributes like invalidParams and instances are not included in the generated error responses.
- When verbose error responses are enabled, all security countermeasure features display verbose (detailed) error responses, similar to those in previous SEPP releases.

Note

Nonverbose error response in error reason header of Egress Gateway:

Earlier, when n32-egress-gateway or plmn-egress-gateway encountered exceptions or errors, it added a custom "error-reason" header to the responses. This header contained verbose (detailed) information about the exceptions and errors.

With this enhancement, SEPP's Ingress Gateway has been configured to remove the "error-reason" header at the route level, which was generated by Egress Gateway.

Cat-0 SBI Message Schema Validation Feature

The following are the verbose and nonverbose error response examples:

Table 3-20 SCM Error Responses

Cat 0 Error Responses Attributes	Attribute Value with Verbose (Detailed) Error Responses Enabled	Attribute Default Value with Verbose Error Responses Disabled (Concise Error Message)
title	"Message validation failed at P-SEPP"	"Rejected"

Table 3-20 (Cont.) SCM Error Responses

Cat 0 Error Responses Attributes	Attribute Value with Verbose (Detailed) Error Responses Enabled	Attribute Default Value with Verbose Error Responses Disabled (Concise Error Message)
detail	"Message validation for request / nnrf-disc/v1/nf-instances failed for remote sepp set: SEPP-3"	"Server Error"
cause	"Error in Request Parameter(s)"	"Unknown"
status	406	406
invalid Params	["parameters.requester-plmn-list: Invalid JSON syntax", "parameters.target-plmn-list: Invalid JSON syntax", "service-names[0]: integer found, string expected", "parameters.target-nf-type: is missing but it is required"]	This attribute is not part of error response.
instance	"/nnrf-disc/v1/nf-instances"	This attribute is not part of error response.

For more information about Cat-0 SBI Message Schema Validation Feature, see [Cat-0 SBI Message Schema Validation feature](#).

Cat-1 Service API Validation Feature

The following are the verbose and nonverbose error response examples:

Error Response Examples:

Table 3-21 SCM Error Responses

Cat 1 Error Responses Attributes	Attribute Value with Verbose (Detailed) Error Responses Enabled	Attribute Default Value with Verbose Error Responses Disabled (Concise Error Message)
title	"Service API not in allowed list"	"Rejected"
detail	"Service API /namf-evts/v1/subscriptions/73ce9152d6bc4033bfc9296190c62--394 from Remote SEPP SEPP-4 is not in allowed list"	"Server Error"
cause	"Service API=/namf-evts/v1/subscriptions/73ce9152d6bc4033bfc9296190c62--394,remoteSepp=SEPP-4"	"Unknown"
status	406	406
instance	"/namf-evts/v1/subscriptions/73ce9152d6bc4033bfc9296190c62--394"	This attribute is not part of error response.

For more information about Cat-1 Service API Validation Feature, see [Cat-1 Service API Validation Feature](#) section.

Cat -2 Network ID Validation Feature

The following are the verbose and nonverbose error response examples:

Table 3-22 SCM Error Responses

Cat 2 Error Responses Attributes	Attribute Value with Verbose (Detailed) Error Responses Enabled	Attribute Default Value with Verbose Error Responses Disabled (Concise Error Message)
title	"Network ID Validation Failed"	"Rejected"
detail	"Service API /namf-loc/v1/imsi-987654300000001/provide-loc-info from Network ID Validation List Name 'default' is not having expected Network ID in header"	"Server Error"
cause	"Service API= /namf-loc/v1/imsi-987654300000001/provide-loc-info , Network ID Validation List Name = default"	"Unknown"
status	406	406
instance	"/namf-loc/v1/imsi-987654300000001/provide-loc-info,POST"	This attribute is not part of error response.

For more information about Cat -2 Network ID Validation Feature, see [Cat -2 Network ID Validation Feature](#) section.

Cat-3 Previous Location Check Feature

The following are the verbose and nonverbose error response examples:

Table 3-23 SCM Error Responses

Cat 3 Error Responses Attributes	Attribute Value with Verbose (Detailed) Error Responses Enabled	Attribute Default Value with Verbose Error Responses Disabled (Concise Error Message)
title	"CAT 3 Previous Location check failed due to exception"	"Rejected"
detail	"Service API /nsmf-pdusession/v1/pdu-sessions Allowed List Name 'newcustomlist4' is not authenticated due to exception :UDR Response Is Not Success"	"Server Error"
cause	"Service API= /nsmf-pdusession/v1/pdu-sessions , Allowed List Name = newcustomlist4"	"Unknown"

Table 3-23 (Cont.) SCM Error Responses

Cat 3 Error Responses Attributes	Attribute Value with Verbose (Detailed) Error Responses Enabled	Attribute Default Value with Verbose Error Responses Disabled (Concise Error Message)
status	406	406
instance	"Service API= /nsmf-pdusession/v1/pdu-sessions , Allowed List Name = newcustomlist4"	This attribute is not part of error response.

For more information about Cat-3 Previous Location Check Feature, see [Cat-3 Previous Location Check Feature](#) section.

Managing Security Countermeasure Nonverbose Error Responses

Enable

You can enable or disable the feature using the CNC Console or REST API.

Note

This feature is enabled by default. If the user has to see the verbose error response, the feature must be disabled at global and Remote SEPP Set level.

To enable or disable the feature, perform the following steps:

CNC Console

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and click **Security Countermeasure**.
2. Click **Non Verbose Error Measure Configuration** under **Security Countermeasure**, the **Option** appears.
3. The Security countermeasure Nonverbose Configuration feature details are available on the **Option** screen.
4. Click **Edit** icon to modify the Option. The **Edit Option** page appears.
5. Set **Verbose Error Response Enabled** to True or False on the right pane.

REST API

Set `VerboseScmErrRspConfigEnabled` as true or False using the Verbose Error Response Configuration Options REST API.

```
{
  "VerboseScmErrRspConfig": {
    "enabled": false,
    "title": "Rejected",
    "cause": "Unknown",
    "detail": "Server Error"
  }
}
```

Enabling and Configuring Security countermeasure Nonverbose Error Responses in Remote SEPP Set

The user selects the Remote SEPP Set screen on CNC Console and can set the Verbose Error Response Enabled parameter to true or false, and configure **Security Counter Measure (SCM) - Non Verbose Error Response Configuration**.

Configure

You can configure the feature using REST API and CNC Console.

- **Configure using REST API:** Perform the feature configurations as described in Verbose Error Response Configuration Options REST APIs in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- **Configure using CNC Console:** Perform the feature configurations as described in the [Nonverbose Error Response Configuration](#) and [Remote SEPP Set](#) sections of [Configuring SEPP using CNC Console](#) chapter.

Observe

There are no new metrics and KPIs added or updated for this feature.
For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.13.2 Stateless Countermeasure Features

The following are the SEPP stateless countermeasure features:

3.2.13.2.1 Cat-1 NRF Service API Query Parameters Validation

Background

In the existing implementation of the Cat-1 feature, the SEPP cannot filter the traffic based on the NF type in the discovery request. This limitation makes it difficult to manage access to services based on specific NF types. For example, while allowing an AMF from a foreign network to request UDM services may be necessary, it may not be suitable for a Policy to access those services.

Currently, the SEPP enforces Cat-1 rules on an API-wide basis. It can either block the entire API or allow its verification, but it lacks the flexibility to apply filtering using the NF type information included in the query parameters of incoming discovery requests. This results in limited control for network operators, preventing them from applying specific access control rules based on the requesting and target NF types.

Overview

This feature is designed to address the gap in the current implementation by leveraging the NF Type details provided in the discovery request. By integrating the requester-nf-type and target-nf-type information, the feature allows filtering of requests based on the specific NF Type combination configured by the user. This enhancement provides finer control, enabling operators to refine Cat-1 rules based on specific NF Type information.

Key benefits of the feature include:

- **Controlled access:** Provides NF types specific access to nnrf-disc service, blocking any unauthorized requests.
- **Improved security:** Blocks unapproved access, it improves security and protects the system from unwanted or harmful interactions.
- **Targeted filtering:** Operators can set rules based on NF Types, allowing for more specific control over which network functions can access the service.
- **Reduced risk:** The firewall reduces the chances of untrusted network functions connecting to important APIs, making the system safer.

This feature enhances the security by allowing only trusted network functions to access the nnrf-disc service.

Feature Description

The Cat-1 NRF Service API Query Parameters Validation enhancement in SEPP adds a filtering system for NRF discovery requests. With this feature, SEPP checks the requester-nf-type and target-nf-type parameters in incoming discovery requests to see if they match with a preconfigured list of allowed NF-Type combinations. This helps to ensure that only authorized network functions (NFs) can access the NRF, blocking unauthorized requests and improving network security.

SEPP processes the discovery incoming requests as follows:

- SEPP extracts the requester-nf-type and target-nf-type from the incoming request.
- It compares these extracted values with the list of allowed combinations that the users have configured.
- For example, a valid request could look like this:

```
/nnrf-disc/v1/nf-instances/. *target-nf-type=SMF.$ & requester-nf-type=AMF.$
```

SEPP uses these validation rules to check if the request matches the allowed combinations. If it does, the request is processed; if not, SEPP rejects it and sends an error message. This feature makes firewall rules more specific and effective in managing network traffic.

If the requester-nf-type and target-nf-type are not configured in SEPP, the system will continue with its default behavior and allow the request without additional validation. However, if these parameters are missing from a request and SEPP is set to check for them, SEPP will treat the request as invalid and return an error.

This feature can be controlled by the `Enable CAT1-Query Parameter Validation` parameter. Operators can choose to enable or disable it as needed. When enabled, SEPP enforces NF-Type validation along with the existing Cat-1 firewall rules. The system checks whether the request is allowed based on the service and roaming partner, and if the additional NF-Type validation is enabled. If everything matches with the request is allowed; otherwise, SEPP responds with an error message or problem details, depending on the verbosity setting.

Managing Cat-1 NRF Service API Query Parameters Validation

This section provides information about the configurations required to enable and configure this feature.

Enable

You can enable the Cat-1 NRF Service API Query Parameters Validation feature using the CNC Console or REST API.

Enable using REST API

Set `queryParamValidationEnabled` to `true` in `/sepp-configuration/v1/security-counter-measure/` feature REST API.

```
{
  "messageFilteringOnResourceUriAndHttpMethodEnabled":
  {
    "queryParamValidationEnabled": true,
    "enabled": true
  }
}
```

For more information about API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Enable using CNC Console

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat 1 -Service API Validation** under **Security Countermeasure**. **Option, Service API Allowed List**, and **Query Parameter Validation List** appears.
3. Click **Option**, the **Options** page appears on the right pane.
4. Set **Enable Cat 1-Query Parameter Validation** parameter to **True** to enable the feature.

Note

Enable Cat 1-Service API Validation parameter must be set to **True** to enable this feature.

For more details about CNC Console Configurations, see *Configuring SEPP using CNC Console* section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide*.

Configure

You can configure the Cat-1 NRF Service API Query Parameters Validation using REST API and CNC Console.

- Configure Cat-1 NRF Service API Query Parameters Validation using REST API: Perform the feature configurations as described **Security Counter Measure Service API Validation**, **Security Countermeasure Service API Allowed List Name**, and **Supported Service API's** sections in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- Configure Cat-1 NRF Service API Query Parameters Validation using CNC Console: Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Observe

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.13.2.2 Cat-0 SBI Message Schema Validation Feature

Overview

The Cat-0 SBI Message Schema Validation feature intends to support stateless security counter measures, which enhance the security measures of roaming partners by allowing only valid requests to be processed at SEPP. Cat-0 SBI Message Schema Validation feature supports the checks that are carried out to investigate if a message is valid or not. It also checks for the correctness of the message format.

This feature is designed to support the following:

- Validating the headers present in the SBI Message. Example, it checks whether the header name spelling is correct.
- Verifying the combination of request body and query parameters and compare it with the respective schema.
- Routing of only valid SBI messages to egress or ingress the network.

Feature Description

The Cat-0 SBI Message Schema Validation feature intends to support stateless security counter measures, which enhance roaming partners to handle security (Example: only valid requests are allowed to be processed at SEPP).

This feature is to support request validation at N32F.

As many requests traverse the SEPP, there is a security concern that a spoofing NF or an NF, which may cause not forming the messages as per specification. In such a scenario, the SEPP, which is at the boundary of the network could do a first pass for malformed or incorrect messages.

Malformed messages can be of following types:

- The JSON message may contain data that is malformed. A formatted JSON could still contain malformed data that can cause errors in the server code.
- The JSON message may contain repeated IEs or missing IEs that make the message noncompliant with its schema.

This feature intends to support message validation in the following two ways.

- Request's body value is validated against the corresponding schema as provided in 3GPP OpenAPI specifications or configured by user.

- Request's query parameters values are validated against their corresponding schema(s) as provided in 3GPP OpenAPI specifications or configured by user.

Traffic needs to be rejected or forwarded from C-SEPP or P-SEPP based on the validation results as configured by the user.

As part of the message validation feature, the following major scenarios are getting validated:

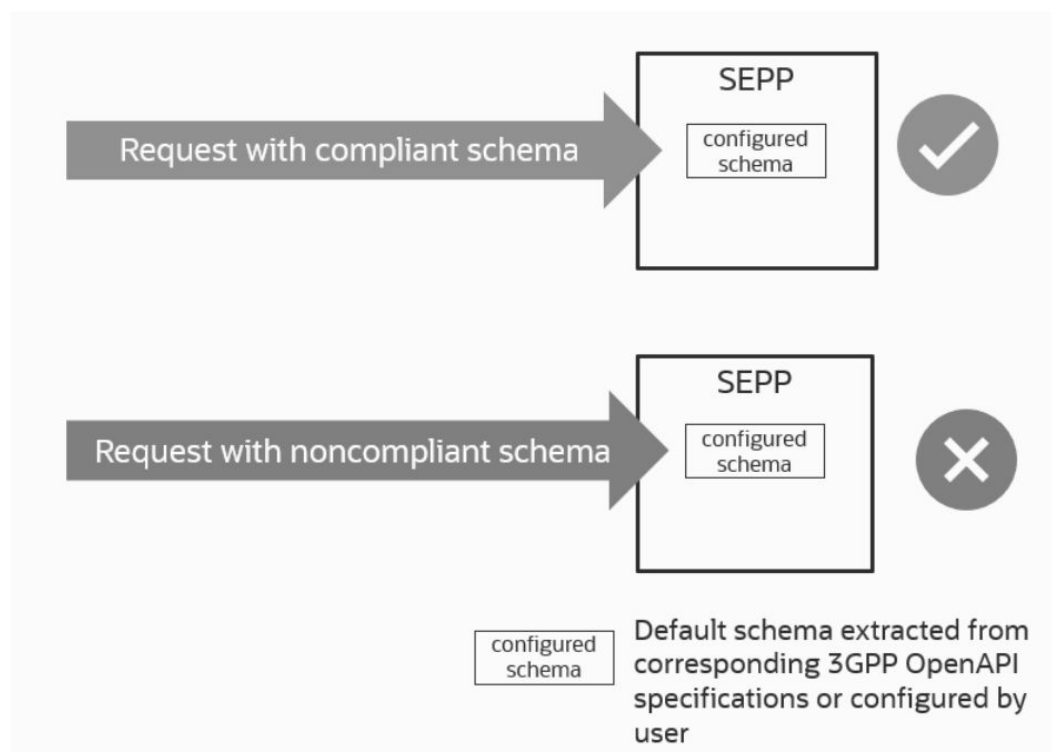
- If any SBI request JSON schema's mandatory parameter is missing, it is validated.
- If any SBI request JSON schema's mandatory parameter under an optional parameter is missing, it is validated.
- In SBI request, the regex of both the mandatory and optional parameters are validated against the regex defined in the given schema configurations.
- The data types of the IEs in SBI request are validated against the given schema configurations.
- SBI request's additional attribute is allowed or disallowed as provided in the given schema configurations.
- Any invalid JSON structure (syntax) either present in SBI request body or in request query parameters value is validated.
- Repeated IEs, either present in SBI request body or in request query parameters value, are validated.

Design and Architecture

This feature can be enabled on C-SEPP and P-SEPP. For all the supported default resource URI and HTTP methods, SEPP provides the corresponding default schema(s) (JSON format) as per 3GPP OpenAPI specifications, which is used to validate the request at run time. The user is also allowed to add or update the default schema. As per the error configurations done by the user, traffic is rejected or forwarded from SEPP based on the validation results.

Note

In SEPP Mode, the Egress and Ingress requests (as per configured egress/ingress rules) are validated against the configured schema at C-SEPP and P-SEPP. In roaming hub mode, only the Ingress requests (as per configured Ingress rules) are validated.

Figure 3-19 Cat-0 Functionality

Feature Configurations

The following diagram specifies the CNC Console options available for Cat-0 SBI Message Schema Validation feature.

Figure 3-20 CNC Console options available for Cat-0 SBI Message Schema Validation feature in SEPP Mode

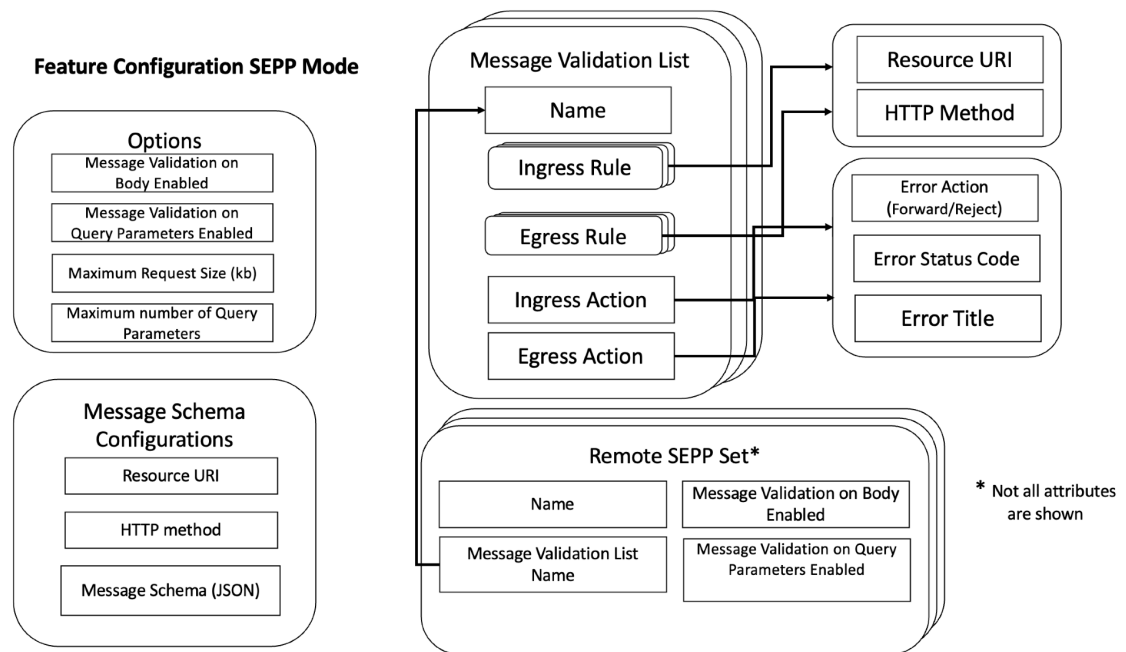
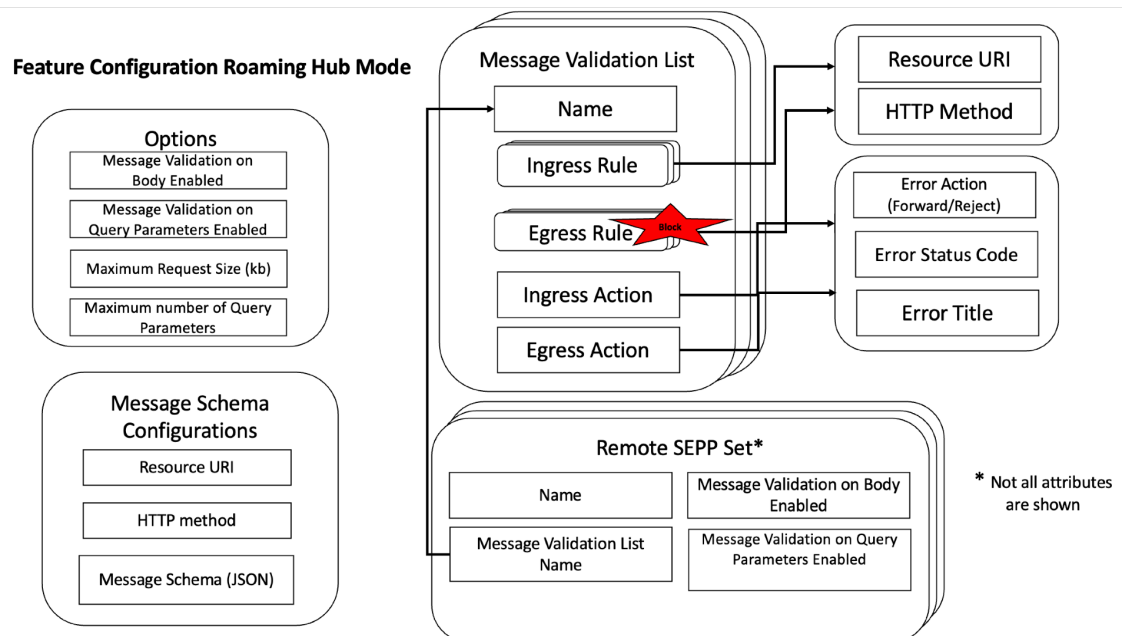


Figure 3-21 CNC Console options available for Cat - 0 SBI Message Schema Validation feature in Roaming Hub Mode



The following section explains the configurations in detail:

Options

- Message Validation on Body Enabled enables or disables the feature on request body.

- Message Validation on Query Parameters Enabled enables or disables the feature on request query parameters.
- Maximum Request Size(kb) provides the maximum allowed request body size (Default is 40 KB)
- Maximum Number Of Query Parameters provides the maximum number of allowed query parameters (Default is 100).

Message Validation List

- As part of this feature configuration, multiple Message Validation Lists can be configured.
- Default Message Validation List with the name Default is configured during SEPP installation which can be used as a sample to visualize the rules.
- Default list uses exhaustive number of rules, and its usage impacts the performance.
- Ingress and Egress Rules define the SBI requests to be validated.
- SEPP in Roaming Hub mode supports Ingress Rules only.
- Ingress and Egress Action defines the Action (Forward or Reject), Status Code, and Title to be used in case message validation gets failed for C-SEPP or P-SEPP.
- Any one of the existing Message Validation Lists can be configured for a Remote SEPP Set. By default, the default Message Validation List is configured and the feature is disabled.

Message Validation Configuration in Remote SEPP Set

The user needs to associate the required message validation list to the Remote SEPP Set for which the feature has to be applied.

Note

To enable the feature, the user needs to enable the Message Validation on Body Enabled and Message Validation on Query Parameters Enabled options available at **Option** screen and Remote SEPP Set CNC Console screens or REST APIs.

Message Schema Configurations

For the supported resource URI and HTTP methods, SEPP provides the default JSON schemas (application or json) that are used to do the message schema validation. The user can do any changes in the default schema or delete it using CNC Console or REST APIs. Before overwriting the default schema, creating a backup of existing schema is advisable, as once overwritten, SEPP will not provide the default schema. User provided custom schema should be fully resolved, that is, it should not have any internal references to other schemas.

To add a new resource URI, HTTP method and corresponding schema, resource URI, and HTTP method need to be present in SEPP's default service APIs.

Sample Request Body Schema

Request body schema must be provided in the valid JSON format.

Example:

```
{
  "required" : [ "amfStatusUri" ],
  "type" : "object",
```

```

"properties" : {
  "amfStatusUri" : {
    "type" : "string"
  },
  "guamiList" : {
    "minItems" : 1,
    "type" : "array",
    "items" : {
      "required" : [ "amfId", "plmnId" ],
      "type" : "object",
      "properties" : {
        "plmnId" : {
          "required" : [ "mcc", "mnc" ],
          "type" : "object",
          "properties" : {
            "mcc" : {
              "pattern" : "^\\d{3}$",
              "type" : "string"
            },
            "mnc" : {
              "pattern" : "^\\d{2,3}$",
              "type" : "string"
            }
          }
        },
        "nid" : {
          "pattern" : "[A-Fa-f0-9]{11}$",
          "type" : "string"
        }
      }
    }
  },
  "amfId" : {
    "pattern" : "[A-Fa-f0-9]{6}$",
    "type" : "string"
  }
}
}
}
}
}
}
}

```

Sample Query Parameters Schema

Query parameter and corresponding schema should be provided in the form of JSON array where each array element will provide information for one query parameter. This information must have the following mandatory attributes:

name: name of the query parameter as used in the request URI.

in: type of parameter, it should be 'query' to indicate that its is a query parameter, any other type will not be supported.

required: true/false, provides mandatory or optional parameter information.

content or schema:

- JSON objects and arrays of JSON objects: They shall be formatted using the JSON syntax, by including a "**content**:" block, and specifying the "application/json" media type, followed by the json **schema** of the object.

- Arrays of simple types: They shall be formatted using the OpenAPI "style" set as "form" and the "explode" set as "false", and specifying the **schema** of the object.

The following example shows content and schema of two query parameters, that is, 'supported-features' and 'plmn-id':

```
[ {
  "name" : "supported-features",
  "in" : "query",
  "required" : false,
  "schema" : {
    "pattern" : "[A-Za-f0-9]*$",
    "type" : "string"
  }
}, {
  "name" : "plmn-id",
  "in" : "query",
  "required" : false,
  "content" : {
    "application/json" : {
      "schema" : {
        "required" : [ "mcc", "mnc" ],
        "type" : "object",
        "properties" : {
          "mcc" : {
            "pattern" : "\\d{3}$",
            "type" : "string"
          },
          "mnc" : {
            "pattern" : "\\d{2,3}$",
            "type" : "string"
          }
        }
      }
    }
  }
} ]
```

Managing Cat-0 SBI Message Schema Validation Feature

Enable

You can enable the Cat-0 SBI Message Schema Validation feature using the CNC Console or REST API.

Enable using REST API

Set `MessageValidationOnBodyEnabled` and `MessageValidationOnQueryParametersEnabled` to true in Message Validation REST API.

```
curl -X 'PUT' \
  'http://<SEPP- Configuration IP>:<PORT>/sepp-configuration/v1/security-counter-measure/
  message-validation/options' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "messageValidationOption": {
      "messageValidationOnBodyEnabled": true,
```

```

    "messageValidationOnQueryParametersEnabled": true,
    "maxRequestSize": 0,
    "maxNumberOfQueryParams": 0
  }
}'

```

For more information about API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Enable using CNC Console

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat 0 - SBI Message Validation** under Security Countermeasure. The **Option** appears underneath.
3. Click **Option**, the option screen appears at the right pane. The Cat 0 - SBI Message Validation feature details are available in the screen.
4. Click **Edit** icon to modify the Option. The **Edit Option** page appears.
5. Set the **Message Validation on Body Enabled** and **Message Validation on Query Parameters Enabled** to True.

For more details about CNC Console configurations, see [Configuring SEPP using CNC Console](#) section in the *Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide*.

Note

To enable the feature, the user also needs to enable the Message Validation on Body Enabled and Message Validation on Query Parameters Enabled options available at Remote SEPP Set, using CNC Console screen or REST APIs.

Observe

The following are the Cat-0 SBI Message Schema Validation feature specific metrics:

- ocsepp_message_validation_applied_total
- ocsepp_message_validation_on_body_failure_total
- ocsepp_message_validation_on_header_failure_total

The following are the Cat-0 SBI Message Schema Validation feature specific KPIs:

- Message validation applied requests on cn32f
- Cn32f message validation failure on request body
- Cn32f message validation failures on request query parameter(s)
- Message validation applied requests on pn32f
- Pn32f message validation failure on request body
- Pn32f message validation failures on request query parameter(s)

For more information on Cat-0 SBI Message Schema Validation feature related SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.13.2.3 Cat-1 Service API Validation Feature

Overview

Interface blocking refers to checks for a type of a message to investigate whether it is allowed on a specific reference point. Typically based on the combination of path and operation type and sometimes operation Id for 5GC messages.

In 5G interconnect scenarios, the operator may prohibit certain interfaces from traversing the networks. In 5G interconnect, the recommended practice for Category 1 filtering is to block all 5G interconnect messages except those explicitly required for a given interface, and to allow/activate on the SEPP only the required interface(s) needed for roaming scenarios. This avoids the risk of roaming partners extending the roaming services or features unilaterally by using interfaces and sending messages that are not covered by an existing roaming agreement. In general, the SEPP needs to be able to differentiate between the different interfaces and methods encapsulated in the N32f packets and treat them separately. This is achieved by using a combination of “Method Types” and URIs in Cat -1 feature.

The Cat-1 Service API Validation feature allows the operator to configure the stateless security measures at the SEPP. This feature intends to support this in two ways:

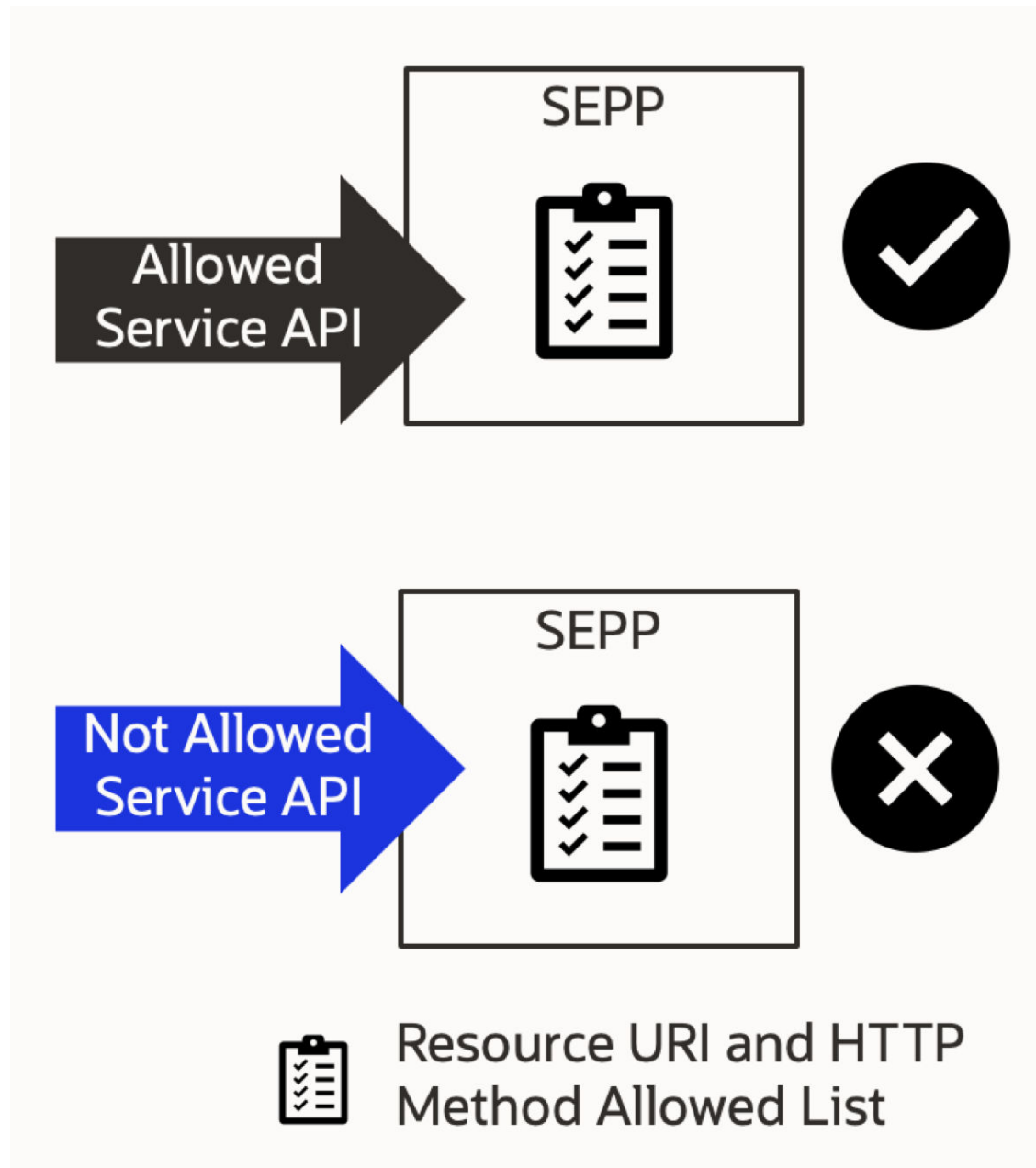
- Check of combination of HTTP Method and Service requested (Example: POST cannot be used in combination with a particular service).
- Allows only certain SBI messages to either egress or ingress the network (Example: AMF to UDM messages)

Design and Architecture

SEPP allows a list of allowed Resource URIs and HTTP Method to be configured. Every Ingress and egress messages is validated against the configured allowed list on P-SEPP and C-SEPP.

The configuration is available at both C-SEPP and P-SEPP. If the Resource URI and HTTP Method matches with the configured one, that Service API is allowed to pass through the SEPP and if the Resource URI and Method in the ingress and egress messages does not matches, then as per the configured action in the error configuration by the user, the Service API is rejected.

Figure 3-22 Security Counter Measure Service



To add a new Resource URI along with HTTP Method which is not present in the Allowed List, add the new Resource URI first in Service API section (on the Main Menu of SEPP in CNC Console GUI) and add it in the Allowed List. The newly added Resource URI and Method automatically be updated in the drop down of Ingress Rule and Egress Rule.

By default the Resource URI list in CNC Console Screen will be populated with some default Resource URIs and HTTP Methods which is user configurable and can be added or deleted as required.

User can consider the following points while configuring the Cat-1 Service API Validation feature:

- Multiple Allowed List can be configured by the user.

- Default Allowed List with name *Default* is configured during SEPP installation.
- An allowed list needs to be configured for a Remote SEPP Set mandatorily.
- Ingress and Egress Rules define the combination of allowed Resource URIs and HTTP Methods.
- Ingress and Egress Action defines the Action, Status Code, and Title to be used in case the Service API is not allowed.

Managing Cat-1 Service API Validation Feature

Enable

You can enable Cat-1 Service API Validation feature using the REST API or CNC Console.

- Enable using REST API: Set Enable Security Counter Measure to True in **Security Counter measure Service API Validation Feature State API**. For more information about API path, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy REST Specification Guide*.
- Enable using CNC Console: Set Enable Service API Validation to True on the **Security Counter Measure** page. For more information about enabling the feature using CNC Console, see **Security Counter Measure Options**.

Configure

You can configure the Cat-1 Service API Validation using REST API and CNC Console.

- Configure Cat-1 Service API Validation using REST API: Perform the feature configurations as described Security Counter Measure Service API Validation, Security Countermeasure Service API Allowed List Name, and Supported Service API's sections in *Oracle Communications Security Edge Protection Proxy Rest API Guide*.
- Configure Cat-1 Service API Validation using CNC Console: Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Observe

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.13.2.4 Cat -2 Network ID Validation Feature

Overview

Cat -2 Network ID Validation feature refers to the checks carried out to investigate if a message is sent from home or the allocated network (for example, in case of a shared infrastructure). This feature also determines if there are any inconsistencies between the information in the lower level and in the embedded packets and IEs.

An example of this is the MCC/MNC combination in the incoming SUPI/SUCI that could be validated against Home PLMN IDs that are supported by the P-SEPP. Also, any message configured to validate CSEPP PLMN ID, that contains the servingNetworkName Body IE parameter could be validated against Visited network PLMN ID supported by C-SEPP.

Cat- 2 Network ID Validation feature allows the user to validate the PLMN ID of Producer SEPP and Consumer SEPP in the SBI Request Messages over N32 Interface in SEPP Mode as well as in Roaming Hub Mode.

This feature is not applicable to SBI responses that is, only SBI requests are validated.

This feature can be used for:

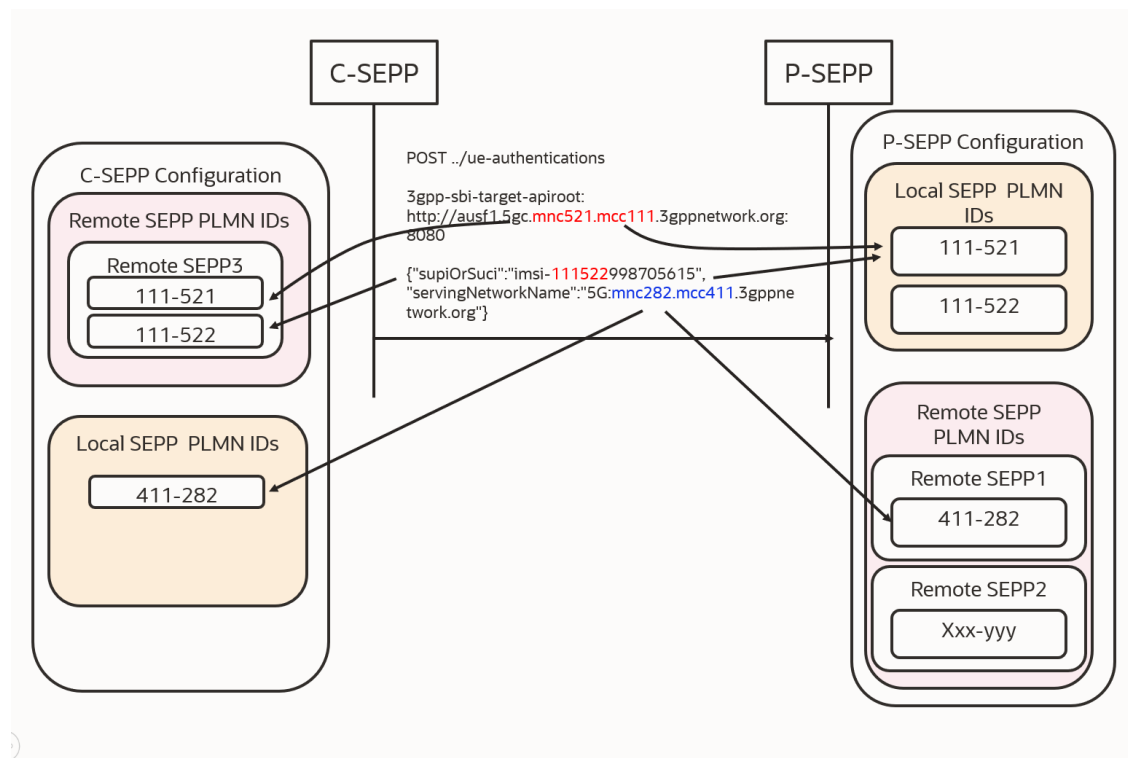
- The validation of subscribers that are allowed into the home network.
- Validation of PLMN ID of Ingress messages.

Design and Architecture

The following diagram represents the Cat-2 Network ID validation feature in SEPP mode.

SEPP Mode

Figure 3-23 SEPP Mode



In SEPP Mode, this feature can be applied on Producer SEPP (P-SEPP) as well as on Consumer SEPP (C-SEPP). At Producer SEPP, this feature validates the P-SEPP PLMN ID against the configured Local SEPP PLMN IDs in the N32 Ingress SBI requests (header and JSON IE) and C-SEPP PLMN ID against the configured Remote SEPP PLMN IDs.

Where as, at Consumer SEPP, this feature validates the P-SEPP PLMN ID against the configured Producer Remote SEPP PLMN IDs in the N32 Egress SBI requests (header and JSON IE) and C-SEPP PLMN ID against Local SEPP PLMN IDs configured at C-SEPP.

When the feature is enabled at C-SEPP:

- If the user wants to validate the PLMN ID in *3gpp-sbi-target-apiroot* header, then user can add a configuration for this header in Header Configuration for ALL Service APIs or for a particular Service API. Once the configuration is done, and if it is configured for ALL Service APIs, then for every SBI request message containing *3gpp-sbi-target-apiroot* header, PLMN ID is extracted from MCC and MNC part of this header and is compared with the Remote SEPP PLMN IDs. If a match is found, then that request is allowed and if a match is not found, then that request is rejected or forwarded as per user configured Error Configuration.
- If the user wants to validate the PLMN ID in *3gpp-Sbi-Originating-Network-Id* or *3gpp-Sbi-Asserted-Plmn-Id* headers, then user can add a configuration for this header in header configuration for all Service APIs or for a particular Service API. Once the configuration is done, and if it is configured for all Service APIs, then for every SBI request message containing *3gpp-Sbi-Originating-Network-Id* or *3gpp-Sbi-Asserted-Plmn-Id* header, PLMN is extracted from this header value and is compared with the local SEPP PLMN IDs. If a match is found, then that request is allowed and if a match is not found, then that request is rejected or forwarded as per user configured error configuration.

Note

It is recommended to use the Regex `(\d{3})-(\d{2,3})` for configuring *3gpp-Sbi-Originating-Network-Id* and *3gpp-Sbi-Asserted-Plmn-Id* headers.

- If user wants to validate *supiOrSuci* value in UE Authentication SBI request message, then user can add a configuration for this JSON IE in Body Configuration for UE Authentication SBI request API. Once the configuration is done, then for every UE Authentication SBI request message, PLMN ID is extracted from *supiOrSuci* IE and is compared with the Remote SEPP PLMN IDs. If a match is found, then that request is allowed and if a match is not found then that request is rejected or forwarded as per user configured Error Configuration.
- If user wants to validate *servingNetworkName* value in UE Authentication SBI request message, then user can add a configuration for this JSON IE in Body Configuration for UE Authentication SBI request API. Once the configuration is done, then for every UE Authentication SBI request message, PLMN ID is extracted from *servingNetworkName* IE and is compared with the Local SEPP PLMN IDs. If a match is found, then that request is allowed and if a match is not found, then that request is rejected or forwarded as per user configured Error Configuration.

When the feature is enabled at P-SEPP:

- If the user wants to validate the PLMN ID in *3gpp-sbi-target-apiroot* header, then user can add a configuration for this header in Header Configuration for all Service APIs or for a particular Service API. Once the configuration is done, and if it is configured for all Service APIs, then for every SBI request Message containing *3gpp-sbi-target-apiroot* header, PLMN ID is extracted from MCC and MNC part of this header and is compared with the Local SEPP PLMN IDs. If a match is found, then that request is allowed and if a match is not found, then that request is rejected or forwarded as per user configured Error Configuration.
- If the user wants to validate the PLMN ID in *3gpp-Sbi-Originating-Network-Id* or *3gpp-Sbi-Asserted-Plmn-Id* headers, then the user can add a configuration for this header in Header Configuration for all Service APIs or for a particular Service API. Once the configuration is done, and if it is configured for all Service APIs, then for every SBI request Message containing *3gpp-Sbi-Originating-Network-Id* or *3gpp-Sbi-Asserted-Plmn-Id* header, PLMN is extracted from this header value and is compared with the remote SEPP PLMN IDs. If a

match is found, then that request is allowed and if a match is not found, then that request is rejected or forwarded as per user configured error configuration.

Note

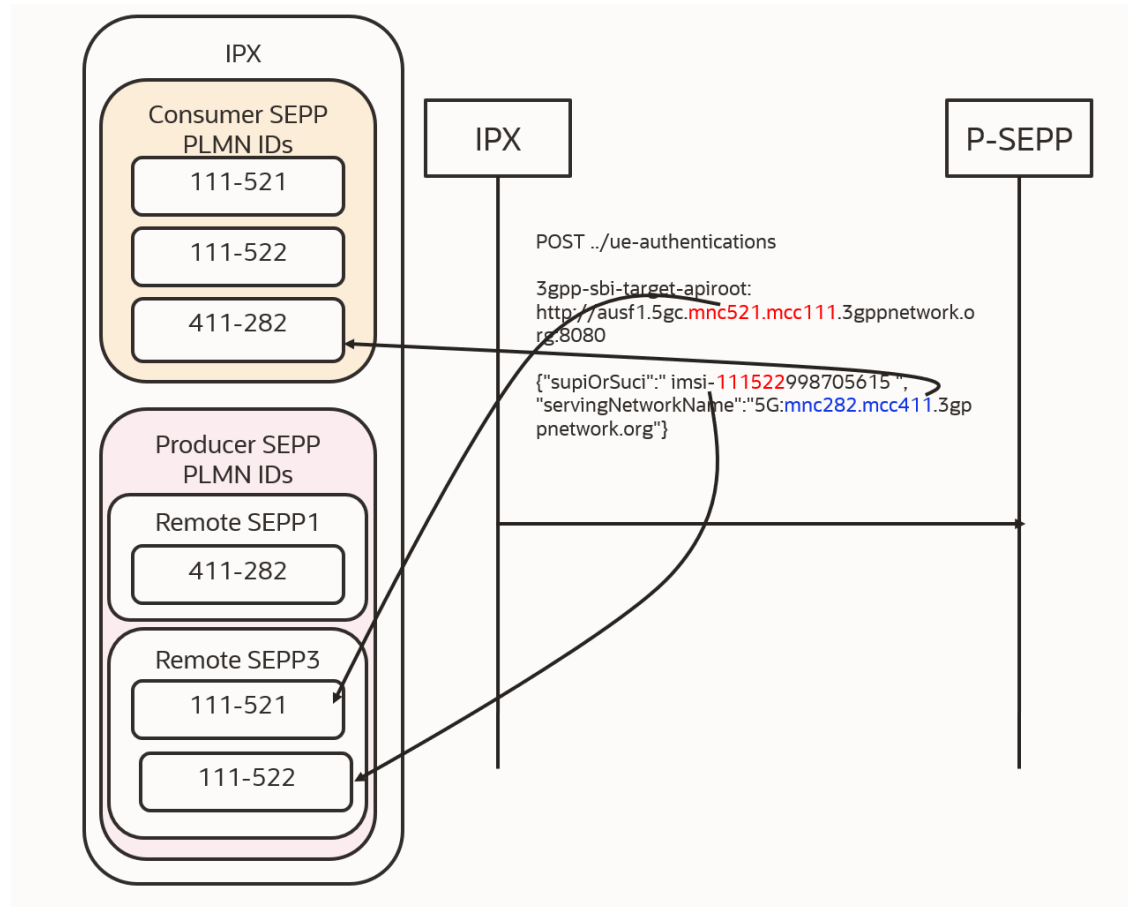
It is recommended to use the Regex `(\d{3})-(\d{2,3})` for configuring 3gpp-Sbi-Originating-Network-Id and 3gpp-Sbi-Asserted-Plmn-Id headers.

- If the user wants to validate *supiOrSuci* value in UE Authentication SBI request message, then the user can add a configuration for this JSON IE in Body Configuration for UE Authentication SBI request API. Once the configuration is done, then for every UE Authentication SBI request message, PLMN ID is extracted from *supiOrSuci* IE and is compared with the Local SEPP PLMN IDs. If a match is found, then that request is allowed and if a match is not found, then that request is rejected or forwarded as per user configured Error Configuration.
- If the user wants to validate *servingNetworkName* value in UE Authentication SBI request message, then user can add a configuration for this JSON IE in Body Configuration for UE Authentication SBI request API. Once the configuration is done, then for every UE Authentication SBI request message, PLMN ID is extracted from *servingNetworkName* IE and is compared with the Remote SEPP PLMN IDs. If a match is found, then that request is allowed and if a match is not found then that request is rejected or forwarded as per user configured Error configuration.

In Roaming Hub mode

The following diagram represents the Cat 2 – Network ID validation feature in Roaming Hub mode.

Figure 3-24 Roaming Hub mode



In Roaming Hub mode, this feature validates the P-SEPP PLMN ID (received in the incoming message in *3gpp-sbi-target-apiroot* and *supiOrSuci*) against the configured Producer Remote SEPP PLMN IDs in the N32 Egress SBI requests (header and JSON IE) and C-SEPP PLMN ID against the Consumer Remote SEPP PLMN IDs. In this mode, only Egress SBI requests are validated. Ingress SBI requests are not validated. So, user is not allowed to add Ingress Rules in Network ID Validation List of the feature.

When the feature is enabled at Roaming Hub:

- If the user wants to validate the PLMN ID in *3gpp-sbi-target-apiroot* header, then user can add a configuration for this header in Header Configuration for all Service APIs or for a particular Service API. Once the configuration is done, and if it is configured for all Service APIs then for every SBI request Message containing *3gpp-sbi-target-apiroot* header, PLMN ID is extracted from MCC and MNC part of this header and is compared with the Producer SEPP PLMN IDs. If a match is found, then that request is allowed and if a match is not found then that request is rejected or forwarded as per user configured Error Configuration.
- If the user wants to validate the PLMN ID in *3gpp-Sbi-Originating-Network-Id* or *3gpp-Sbi-Asserted-Plmn-Id* headers, then user can add a configuration for this header in Header Configuration for all Service APIs or for a particular Service API. Once the configuration is done, and if it is configured for all Service APIs then for every SBI request Message containing *3gpp-Sbi-Originating-Network-Id* or *3gpp-Sbi-Asserted-Plmn-Id* header, PLMN ID is extracted from the header and is compared with the consumer SEPP PLMN IDs. If a

match is found, then that request is allowed and if a match is not found then that request is rejected or forwarded as per user configured error configuration.

Note

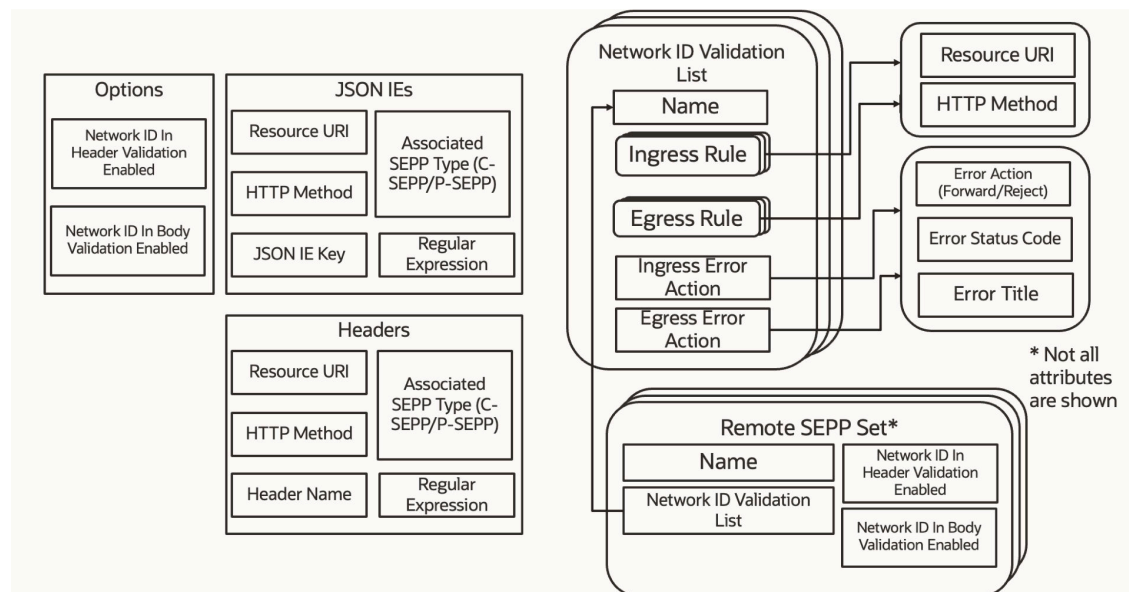
It is recommended to use the Regex $(\backslash d\{3\})-(\backslash d\{2,3\})$ for configuring 3gpp-Sbi-Originating-Network-Id and 3gpp-Sbi-Asserted-Plmn-Id headers.

- If the user wants to validate *supiOrSuci* value in UE Authentication SBI request message, then user can add a configuration for this JSON IE in Body Configuration for UE Authentication SBI request API. Once the configuration is done, then for every UE Authentication SBI request message, PLMN ID is extracted from *supiOrSuci* IE and is compared with the Producer SEPP PLMN IDs. If a match is found, then that request is allowed and if a match is not found then that request is rejected or forwarded as per user configured Error Configuration.
- If user wants to validate *servingNetworkName* value in UE Authentication SBI request message, then user can add a configuration for this JSON IE in Body Configuration for UE Authentication SBI request API. Once the configuration is done, then for every UE Authentication SBI request message, PLMN ID is extracted from *servingNetworkName* IE and is compared with the Consumer SEPP PLMN IDs. If a match is found, then that request is allowed and if a match is not found then that request is rejected or forwarded as per user configured Error Configuration.

Configurations

The following diagram specifies the CNC Console options available for Cat 2 – Network ID Validation.

Figure 3-25 CNC Console options available for Cat 2 – Network ID Validation



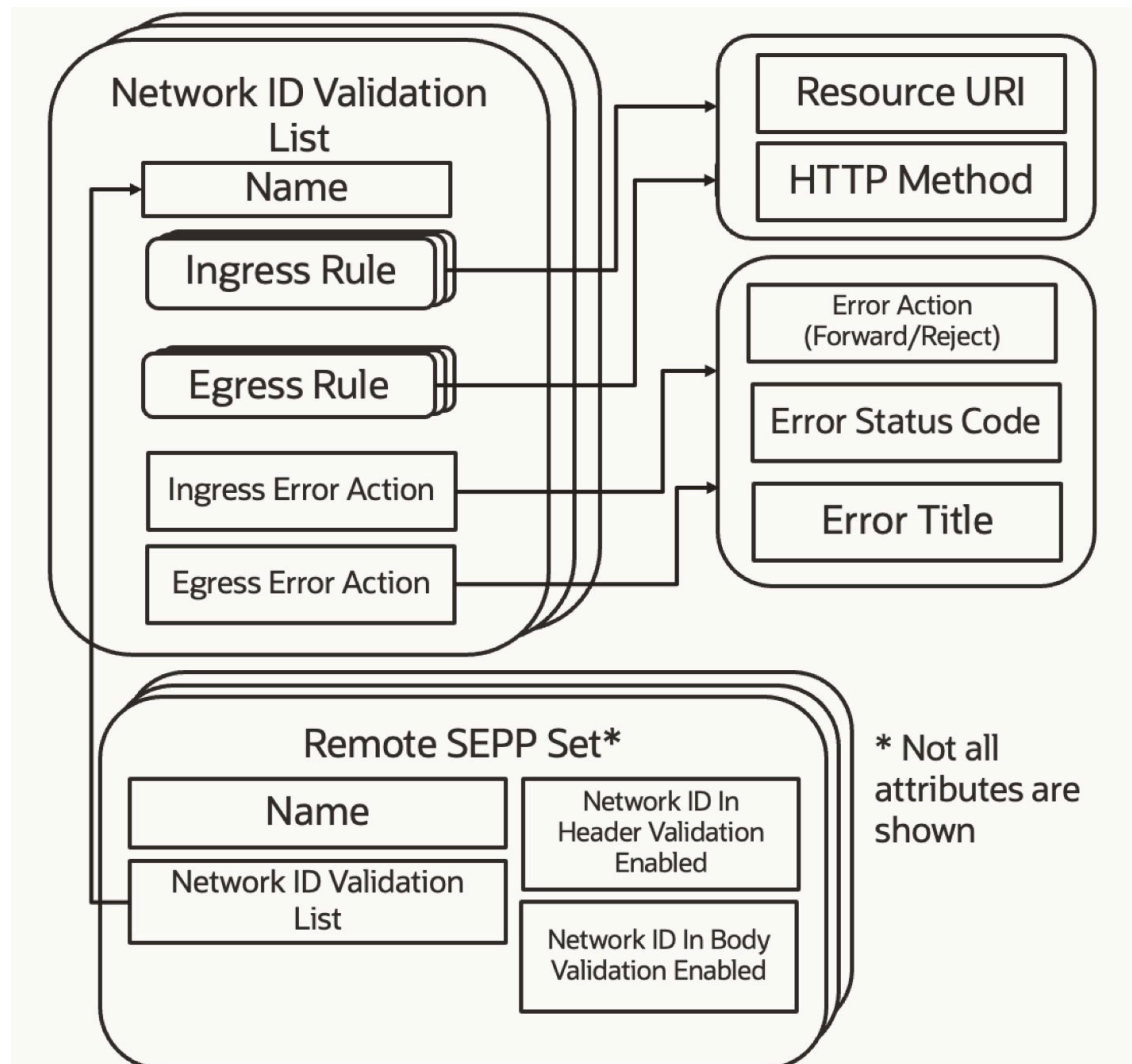
The list of options available in CNC Console is as follows:

- Options

- JSON IEs
- Headers
- Network ID Validation List
- Remote SEPP Set

Feature Configuration –Network ID Validation List

Figure 3-26 Feature Configuration – Network ID Validation List



- As part of this feature configuration, multiple Network ID Validation Lists can be configured for a Remote SEPP Set.
- Default Network ID Validation List with the name Default is configured during SEPP installation which can be used as a sample to visualize the rules.
- Default list uses exhaustive number of rules, and its usage impacts the performance.
- A Network ID Validation List is configured for a Remote SEPP Set if the feature is enabled.
- By default, the feature is disabled and the default Network ID Validation List is configured.

- Ingress and Egress Rules define the SBI requests to be validated.
- SEPP in Roaming Hub mode supports Egress Rules only.
- Ingress and Egress Action defines the Action (Forward/Reject), Status Code, and Title to be used in case SEPP fails to validate the SBI request for the C-SEPP or P-SEPP PLMN ID.

Feature Configuration –Options,Headers, JSON IEs

- Options Configuration
 - Allows feature to be enabled for validating Network ID in Header or Body.
- Headers Configuration
 - SBI request is identified by Resource URI and HTTP Method.
 - Header is identified by name.
 - Regular Expression extracts the PLMN from the header value.
- JSON IEs Configuration
 - SBI request is identified by Resource URI and HTTP Method.
 - JSON Body IE key is identified by name.
 - Regular Expression extracts the PLMN from the IE value.

Note

If JSON IE name is present in the root path of the JSON as well as present in the nested object, then it is recommended to keep the JSON IE which is present in the root if we want to validate it. If the child JSON IE has to be validating, it is recommended to configure it is own parent JSON IE.

Example:

```
{
  "plmnId": {
    "mcc": "111",
    "mnc": "111"
  },
  "ueLocation": {
    "eutraLocation": {
      "tai": {
        "plmnId": {
          "mcc": "111",
          "mnc": "111"
        },
        "tac": "string",
        "nid": "string"
      }
    }
  }
}
```

In the above example, if a user wants to validate `plmnId` which is present inside `tai` object, then the user has to give identifier name as `tai` in Body IE Configuration. If the user has configured `plmnId` in Body IE Configuration, then as per this example, the `oplevel plmnId` gets validated.

Note

- From 25.1.200 onwards, in CSEPP and PSEPP, the length of mnc will be used from PLMN table where mnc length is configured according to the PLMNs given in Remote SEPP.
- The configured regex uses the length of the mnc configured in the PLMN table for forwarding or rejecting the traffic.
- From 25.1.200 release onwards, for the list of given headerNames (supi, suci, supiOrSuci, ueld, ueContextId, and 3gpp-Sbi-Correlation-Info), the user can configure two rows, given that associated SEPP type (PSEPP or CSEPP) remains same and the regex is different for the combination of resourceURI, headerName, and httpMethod.
- From 25.1.200 release onwards, for the list of given identifiers (supi, suci, supiOrSuci, ueld, and ueContextId), the user can configure two rows, given that associated SEPP type (PSEPP or CSEPP) remains same and the regex is different for the combination of resourceURI, identifier, and httpMethod.

Managing Cat -2 Network ID Validation Feature**Enable**

You can enable the Cat -2 Network ID Validation feature using the CNC Console or REST API.

- **Enable using REST API:** Set **messageFilteringOnNetworkIdHeaderValidation** and **messageFilteringOnNetworkIdBodyValidation** to true in Security Counter Measure REST API.

```
curl -X 'PUT' \
  'http://<SEPP- Configuration IP>:<PORT>/sepp-configuration/v1/security-counter-measure/network-id-validation/options' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "networkIdValidationEnabled": {
      "messageFilteringOnNetworkIdHeaderValidation": true,
      "messageFilteringOnNetworkIdBodyValidation": true
    }
  }'
```

For more information about API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

- **Enable using CNC Console:**
 1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
 2. Select **Cat 2 – Network ID Validation**.
 3. The **Option** appears underneath.
 4. Click **Option**, the option screen appears at the right pane. The Cat 2 – Network ID Validation Feature details are available in the screen.
 5. Click **Edit** icon to modify the Option. The Edit Option page appears.

6. Set **Network ID in Header Validation Enabled** and **Network ID in Body Validation Enabled** to True or False on the right pane.

Configure

You can configure the feature using REST API and CNC Console:

- Configure using REST API: Perform the feature configurations as described in *Cat 2– Network ID Validation System Option* section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- Configure using CNC Console: Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Observe

Following are the CAT 2- Network ID Validation feature specific metrics:

- `ocsepp_network_id_validation_header_failure_total`
- `ocsepp_network_id_validation_body_failure_total`

For more information on CAT 2- Network ID Validation feature related SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following steps:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.13.3 Stateful Countermeasure Features

The following are the SEPP stateful countermeasure features:

3.2.13.3.1 Cat-3 Previous Location Check Feature

Cat-3 Previous Location Check feature is a part of stateful security countermeasures. This feature verifies that only messages from an authenticated user equipment (UE) and originating from the same location where the UE was previously authenticated are permitted to traverse the network.

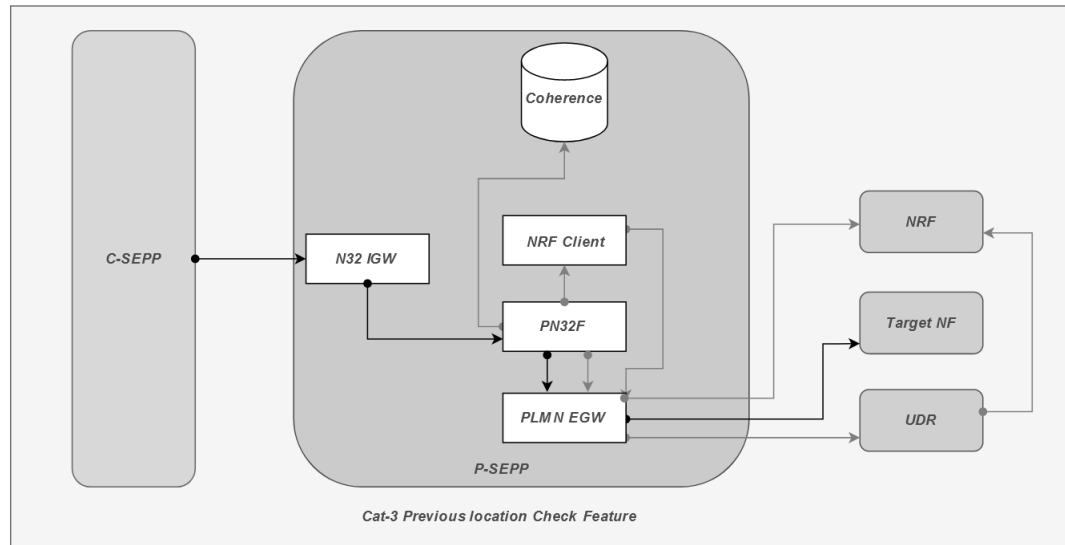
The authentication status is retrieved from the UDR in the home network for a particular User Equipment (UE) and the same is verified for every incoming message reaching the Producer SEPP (P-SEPP). When the UE is verified from the UDR response, that message is allowed to pass through P-SEPP. If the UE is not authenticated or the location of the incoming message do not match with the location registered in UDR at the time of UE authentication process, the message is rejected and displays the user configurable error response.

Note

This feature is available only on P-SEPP.

Design and Architecture

Figure 3-27 Design and Architecture



The diagram shows the functionality of Cat-3 Previous Location Check feature.

At the INIT time of SEPP, or when the SEPP is up and running, PN32F microservice sends a trigger to nrf-client microservice to discover the UDR profile registered with the NRF of home network (P-SEPP) through the PLMN Egress Gateway. The UDR registers its profile with NRF where SEPP is in the allowedNFtype list. Once the UDR profile is received by nrf-client, PN32F microservice stores the details (UDR profile details like FQDN, SUPI range, Port Number, Scheme (HTTP or HTTPS) so on) in coherence cache.

Whenever an incoming Ingress request message reaches P-SEPP, the PN32F microservice fetches the UE ID and the serving network ID from the Ingress request (from the Header or Body JSON) and searches for the UDR FQDN in UDR profile received in the UDR Discovery response. Once UDR FQDN is found in coherence cache map, PN32F microservice sends a request to UDR through PLMN Egress Gateway for finding the authentication status of the UE ID received in the ingress request message. UDR responds with the authentication status of the UE along with the previously authenticated location. PN32F now checks for the authentication status received from UDR. If the UE is authenticated, it then matches the serving network PLMN received in the Ingress request message along with the PLMN in Serving Network Name received in UE authentication message response. If both match, then the request is forwarded to the target NF. SEPP also subscribes for UDR profile change in NRF.

Note

After the upgrade, the Cat-3 Previous Location Check feature must be restored to its previous state, if the feature was enabled previously.

Managing Cat-3 Previous Location Check Feature

You can enable the Cat-3 Previous Location Check feature using the Helm parameters, CNC Console, or REST API.

Enable using Helm Parameter

Set the `seppCoherenceServiceEnabled` parameter to `true` in the N32 Ingress Gateway section of `ocsepp_custom_values_<version>.yaml` file.

```
seppCoherenceServiceEnabled: true
```

For more information about the Helm parameters, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Enable using REST API

Set `messageFilteringOnPreviousLocationCheckValidation` to `true` in Previous Location Check REST API.

```
curl -X 'PUT' \
  'http://<SEPP- Configuration IP>:<PORT>/sepp-configuration/v1/security-counter-measure/previous-location-validation/options' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "previousLocationCheckValidationEnabled": {
      "cacheRefreshTimer": 120000,
      "messageFilteringOnPreviousLocationCheckValidation": false
    }
  }'
```

For more information about API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Enable using CNC Console

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat 3 - Previous Location Check** under Security Countermeasure. The **Option** appears underneath.
3. Click **Option**, the option screen appears at the right pane. The Cat 3 - Previous Location Check feature details are available on the screen.
4. Click **Edit** icon to modify the Option. The **Edit Option** page appears.
5. Set the **Previous Location Check Enabled** to True.

For more details about CNC Console configurations, see [Configuring SEPP using CNC Console](#).

Observe

Following are the Cat-3 Previous Location Check feature specific metrics:

- `ocsepp_previous_location_exception_failure_total`
- `ocsepp_previous_location_validation_success_total`

- ocsepp_previous_location_validation_failure_total
- ocsepp_previous_location_validation_requests_total
- ocsepp_pn32f_notification_fromNrf_total

For more information on Cat-3 Previous Location Check feature related SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.13.3.2 Cat-3 Time Check for Roaming Subscribers

Overview

In roaming scenarios, customers enter different time zones or regions, which must be verified. To ensure the same, SEPP uses PLMN-IDs (Public Land Mobile Network identifiers) to check if the subscriber has entered a new time zone.

The SEPP compares the last recorded PLMN-ID from the subscriber's previous registration with the new PLMN-ID. By doing this, the SEPP can confirm if the subscriber has moved to a new time zone or area. This process ensures accurate monitoring and verification of subscriber movements while roaming. The process starts with the P-SEPP checking if the feature is enabled for the specified consumer Remote SEPP Set. This feature is specifically for AUSF UE Authentication messages and only works when the Subscription Permanent Identifier/Subscription Concealed Identifier (SUPI/SUCI) is included in the incoming message. When the SEPP receives the SBI request, it extracts key information, such as the SUPI/SUCI (UE ID), the Serving PLMN, and the Sender Timestamp.

The Sender Timestamp is taken from the 3gpp-Sbi-Sender-Timestamp value in the header, if available. If unavailable, the timestamp is set to the exact time when the message reaches the SEPP.

First, the SUPI and SUCI are extracted from the incoming message. If the SUCI is present, the SEPP queries the UDM to get the corresponding SUPI. Once the SUPI is retrieved, the SEPP queries the UDR to check the subscriber's authentication status. If the SUPI is already in the incoming message, the SEPP directly queries the UDR for the authentication status.

For better performance, the SEPP caches results from the UDR and UDM in the Coherence Cache. A configurable timeout ensures that frequently requested data is quickly available. If the extracted Serving PLMN is different from the previously authenticated PLMN, an extra check is made: the Sender Timestamp is compared with the Authentication Timestamp.

The minimum transition time is the shortest time it takes for a UE device to travel from one location to another. This estimate considers configurable factors like distance and average flight velocity, providing a baseline for the minimum time the device would need to make a legitimate move.

This comparison ensures that the time difference between the Sender Timestamp and the Authentication Timestamp is equal to or greater than the minimum transition time, thereby

confirming a legitimate transition. The minimum transition time is calculated using the Haversine formula, which calculates the distance between two geographical points based on their latitude and longitude. The result is combined with the average flight velocity (configured by the user) to estimate the time needed for the transition. If this check passes, the request proceeds; if it fails, the request is rejected.

Assumptions

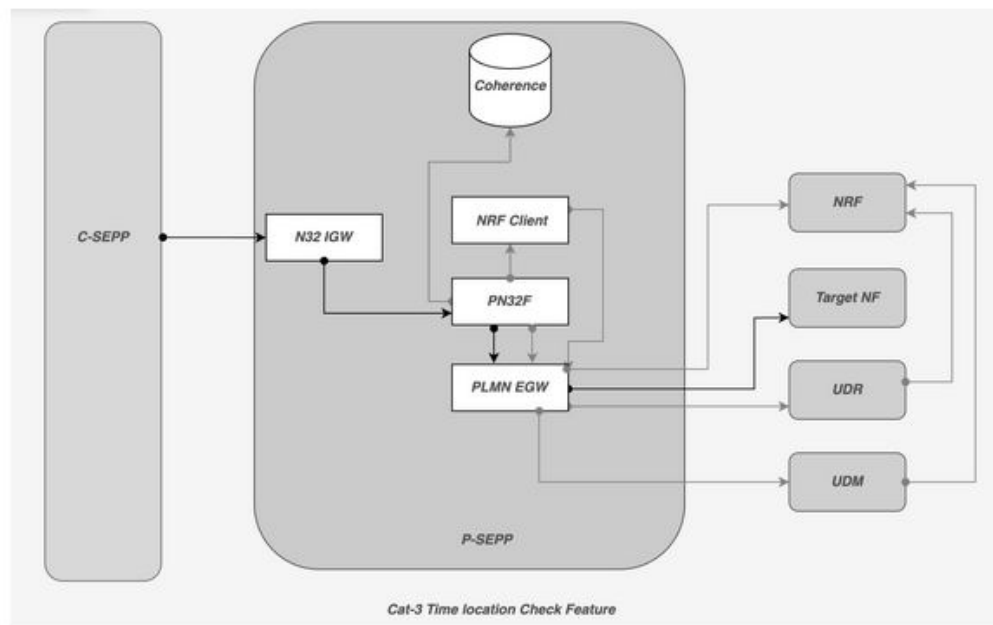
The following are the assumptions for the feature:

- The subscriber has already authenticated in the home network before entering the foreign network.
- UDM and UDR must be accessible.
- The Cat-3 time check for roaming subscribers will only apply to the /nausf-auth/v1/ue-authentications API.
- The UDM and UDR must register their profiles with NRF, and SEPP must be added to the allowed NF type list.

Architecture

The following diagram represents the detailed architecture of this feature.

Figure 3-28 Cat-3 Time Check for Roaming Subscribers Feature



Note

This feature is available only on P-SEPP.

The diagram shows the functionality of Cat-3 Time check for Roaming Subscribers.

Initial Setup and UDR/UDM Discovery

1. **Initialization:** When the system starts up, the PN32F microservice triggers the nrf-client to discover the UDR and UDM profiles registered with the home network's NRF via the PLMN Egress Gateway.
2. **Profile Registration:** The UDR and UDM register their profiles with the NRF, ensuring that the SEPP is included in the allowed NF types list.
3. **Profile Reception:** The nrf-client receives the UDR and UDM profiles from the NRF.
4. **Storing Profile Details:** The nrf-client stores the following details in the coherence cache:
 - UDR Profile: FQDN, SUPI range, port number, scheme (HTTP/HTTPS), etc.
 - UDM Profile: SUPI-SUCI mapping, routing indicators, UDM FQDN, etc.

The coherence cache ensures that these critical details are readily available for quick lookup and further processing as the system handles incoming requests.

Incoming Ingress Request Processing

1. Extracting UE ID and Serving Network ID:

- The PN32F microservice extracts the UE ID (SUPI) and the serving network ID from the incoming UE Authentication request.

2. Handling SUCI (if present):

- If a SUCI is present in the request:
 - The PN32F retrieves the SUCI and its associated routing indicator from the coherence cache.
 - The UDM FQDN is fetched based on the routing indicator from the UDM profile in the cache.
 - The PN32F then queries the UDM using the `/nudm-ueau/v1/{supiOrSuci}/security-information-rg` API to retrieve the corresponding SUPI.

3. Retrieving Authentication Status:

- Once the SUPI is obtained, the UDR FQDN is fetched from the UDR profile in the coherence cache.
- The PN32F then queries the UDR using the `/nudr-dr/v2/subscription-data/{ueId}/authentication-data/authentication-status` API to retrieve the authentication status and the previously authenticated location for the UE.

4. Authentication Status Check:

- The PN32F verifies the UE's authentication status.
 - If the UE is authenticated, the microservice compares the serving network PLMN from the ingress request with the PLMN in the UE authentication response.
 - * If the PLMNs match, the Cat-3 Time Check is not applied, and the request is forwarded to the target NF.
 - * If the PLMNs do not match, the Cat-3 Time Check is applied.

5. Cat- 3 Time Check:

- The Sender Timestamp is extracted from the `3gpp-Sbi-Sender-Timestamp` header, or it is set to the current time if the header is missing.
- The Minimum Transition Time is calculated using the Haversine formula, which computes the distance between two geographical points based on their latitude and longitude. This is

combined with the average flight velocity, a user-configured parameter, to estimate the time needed for the UE to move from the previous location to the new one.

6. Validating the Transition Time:

- The PN32F compares the calculated time (Authentication Timestamp – Sender Timestamp) with the Minimum Transition Time:
 - If the calculated time exceeds the Minimum Transition Time, the Cat-3 Time Check passes, and the request is forwarded to the target NF.
 - If the calculated time is less than the Minimum Transition Time, the request is either:
 - * Rejected, or
 - * Forwarded based on the user-configured error action.

This process ensures that only legitimate transitions based on geographical movement are allowed, preventing fraud or invalid requests from being processed.

Detailed Description

At the start (INIT) of the SEPP, or when the SEPP is running, the PN32F microservice sends a trigger to the nrf-client microservice to discover the UDM and UDR profiles registered with the NRF of the home network (P-SEPP) through the PLMN Egress Gateway. The UDM and UDR register their profiles with the NRF, ensuring that SEPP is included in the allowedNFtype list.

Once the nrf-client receives the UDR profile, the PN32F microservice stores the details (such as FQDN, SUPI range, port number, and scheme type—HTTP or HTTPS) in the coherence cache. After the UDM profile is received, the PN32F microservice also stores the UDM profile details (like SUPI/SUCI mapping, routing indicator, and UDM FQDN) in the coherence cache.

When an incoming Ingress request message reaches the P-SEPP, the PN32F microservice retrieves the UE ID (SUPI) and the serving network ID from the UE Authentication request (/nausf-auth/v1/ue-authentications) in the Body JSON. If a SUCI is present in the request, the PN32F microservice extracts the SUCI and looks for the routing indicator associated with the SUCI in the coherence cache for UDM details.

The PN32F microservice then searches for the UDM FQDN in the UDM profile received in the UDM Discovery response. Once it finds the UDM FQDN in the coherence cache, it sends a request to the UDM using the /nudm-ueau/v1/{supi0rSuci}/security-information-rg API through the PLMN Egress Gateway to retrieve the SUPI associated with the SUCI.

The UDM then responds with the SUPI for the SUCI provided in the ingress request.

Once the SUPI is available, the PN32F microservice searches for the UDR FQDN in the UDR profile received from the UDR Discovery response. When the UDR FQDN is found in the coherence cache, the PN32F microservice sends a request to the UDR through the PLMN Egress Gateway to get the authentication status of the UE ID from the incoming request message. This is done using the /nudr-dr/v2/subscription-data/{ueId}/authentication-data/authentication-status API.

The UDR responds with the authentication status of the UE, along with the previously authenticated location and timestamp. The PN32F microservice then checks the authentication status received from the UDR.

- If the UE is authenticated, PN32F compares the serving network PLMN from the Ingress request message with the PLMN from the Serving Network Name in the UE authentication response from the UDR.
 - If both PLMNs match, the Cat-3 Time check for the Roaming Subscribers feature is not applied, and the request is forwarded to the target NF.

- If the PLMNs do not match, the Cat-3 Time check for the Roaming Subscribers feature is applied, and additional checks are performed before proceeding with the request.

To apply the Cat-3 Time check for Roaming Subscribers feature, the PN32F microservice validates whether the calculated timestamp (Authentication Timestamp – Sender Timestamp) is greater than the Minimum Transition Time.

- If the calculated timestamp is greater than the Minimum Transition Time, the Cat-3 Time check passes, and the request is forwarded to the target NF.
- If the calculated timestamp is less than the Minimum Transition Time, the Cat-3 Time check fails. Depending on the user configuration in the Error Action, the request is either rejected or forwarded.

The PN32F microservice extracts the Sender Timestamp from the 3gpp-Sbi-Sender-Timestamp header. If this header is not present, the Sender Timestamp is set to the exact time when the message reaches the SEPP. The Authentication Timestamp is retrieved from the UDR Response.

To calculate the Minimum Transition Time, the MCC (Mobile Country Code) is extracted from both the serving network PLMN in the Ingress request and the PLMN in the Serving Network Name from the UE authentication response. Once both MCCs are available, their location coordinates (latitude and longitude) are retrieved from the configuration.

Using the Haversine formula and the average flight velocity (a user-configured parameter), the Minimum Transition Time is calculated.

Once the Cat-3 Time check for Roaming Subscribers feature fails for a UE, the UE is added to a blocklist for a configured period, during which all subsequent requests for this UE will be rejected. The blocklist check is applied based only on the SUPI/SUCI information found in the path, query parameters, or message headers. If the SUPI/SUCI is found in the message body, given it is not present in path or query, it will not be considered for the blocklist check. Once the blocklist timer expires or the UE passes the Cat-3 Time check, the UE is removed from the blocklist, and future requests will be forwarded as usual.

Additionally, the SEPP subscribes to receive notifications for any changes to the UDR and UDM profiles in the NRF.

Note

The default value of average flight velocity is set as 12,000 km/hr.

Managing Cat-3 Time Check for Roaming Subscribers Feature

You can enable the Cat-3 Time Check for Roaming Subscribers feature using CNC Console or REST API.

Enable using REST API

Set `messageFilteringOnUnAuthLocationEnabled` to `true` in `/sepp-configuration/v1/security-counter-measure/time-location-check/unauthenticated-location` REST API.

```
{
  "timeUnAuthenticatedCheckValidationEnabled": {
    "unAuthLocationId": 1,
    "messageFilteringOnUnAuthLocationEnabled": true,
    "avgFlightVelocity": 70,
    "blockListRefreshTimer": 11,
    "timeUnit": "SECONDS",
```



```
"cacheRefreshTimer": 11,
"errorAction": {
  "title": "Unauthenticated location check failed",
  "statusCode": 406,
  "action": "REJECT"
},
```

For more information about API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Configure

Enable using CNC Console

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat-3 Time Location Check** under Security Counter Measure, **Unauthenticated Location** page appears underneath.
3. Click **Unauthenticated Location** under Security Countermeasure. The **Option** appears underneath.
4. Click **Option**, the option screen appears at the right pane. The Cat 3 - Previous Location Check feature details are available on the screen.
5. Click **Edit** icon to modify the Option. The **Edit Option** page appears.
6. Set the **Cat 3 Time Check Unauthenticated Location Enabled** to True.

Note

To enable the feature, the user also needs to enable the Cat3-Time Location Check - Unauthenticated Location Enabled parameter available at Remote SEPP Set, using CNC Console screen or REST APIs.

For more details about CNC Console configurations, see *Configuring SEPP using CNC Console* section in the *Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide*.

You can configure the feature using REST API and CNC Console:

- Configure using REST API: Perform the feature configurations as described in the Cat-3 Time Check for Roaming Subscribers section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- Configure using CNC Console: Perform the feature configurations as described in the Cat-3 Time Check for Roaming Subscribers section in [Configuring SEPP using CNC Console](#).

Observe

Following are the Cat-3 Previous Location Check feature specific metrics:

- ocsepp_time_unauthenticated_location_validation_requests_total
- ocsepp_time_unauthenticated_location_validation_success_total
- ocsepp_time_unauthenticated_location_validation_failure_total
- ocsepp_time_unauthenticated_location_exception_failure_total
- ocsepp_time_unauthenticated_location_blocklist_requests_total

- ocsepp_configmgr_country_config_failure_total
- ocsepp_configmgr_exception_mcc_config_failure_total
- ocsepp_configmgr_time_unauthenticated_config_failure_total

For more information on Cat-3 Previous Location Check feature related SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.14 Support for Dual Stack

Overview

According to RFC 4213, Dual stack provides implementation of both versions of the Internet Protocol (IPv4 and IPv6). As there are chances of IPv4 exhaustion, Communication Service Providers (CSP) are seeking support for IPv6 addressing. Moreover, a dual stack networking enables users to use IPv4 or IPv6 or both simultaneously.

Using the dual stack mechanism, applications or NFs can establish connections with pods and services in a Kubernetes cluster using IPv4 or IPv6 or both simultaneously.

Dual stack provides:

- coexistence strategy that allows hosts to reach IPv4 and IPv6 simultaneously.
- IPv4 and IPv6 allocation to the Kubernetes clusters during cluster creation. This allocation is applicable for all the Kubernetes resources unless explicitly specified during the cluster creation.

SEPP can be deployed on CNE that supports dual stack networking. In case both IPv4 and IPv6 addresses are present in the DNS response, the IPv6 interface will be preferred over IPv4 over external interfaces. Several REST API configurations have been modified to cater to the IPv6 configuration, For more information , see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

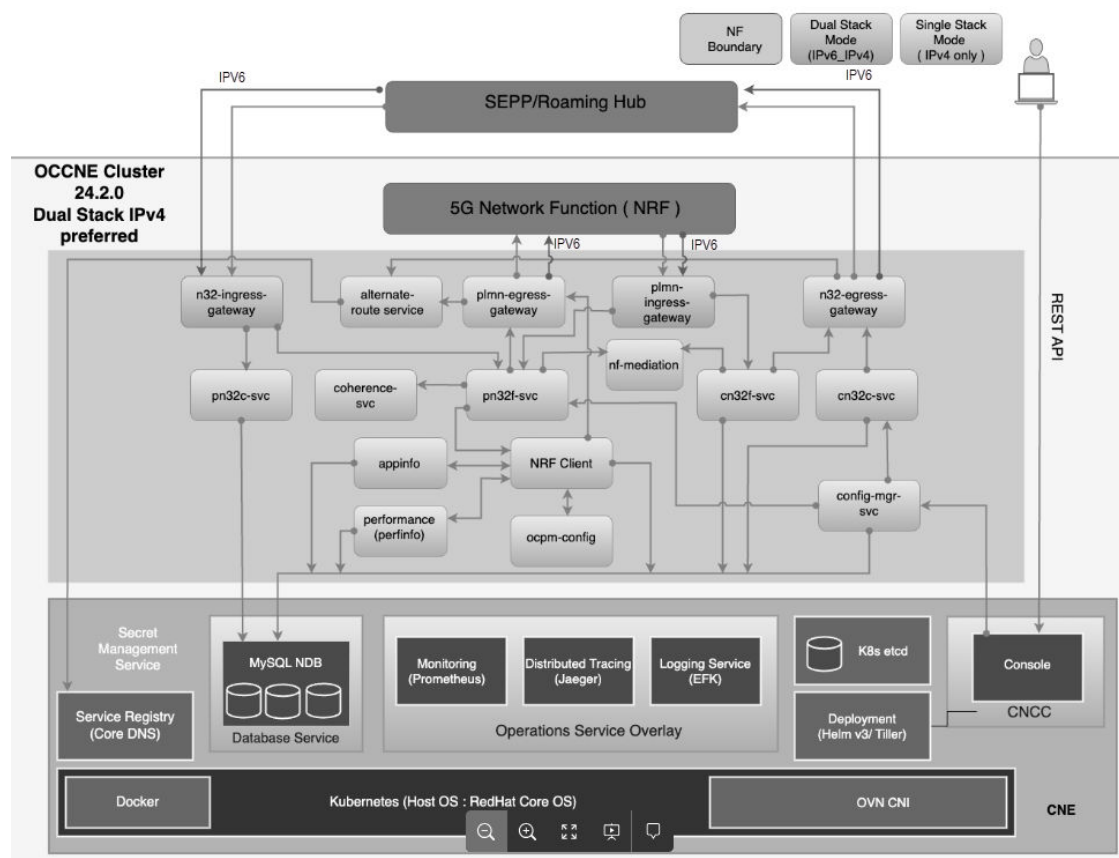
The SEPP now supports dual stack communication, that is, it can handle both IPv4 and IPv6 on its external interfaces. However, it still uses IPv4 exclusively for communication with internal microservices.

- SEPP supports both IPv4 and IPv6 for external communications, allowing the SEPP to connect with external entities such as Remote SEPPs and other network functions (NFs).
- Communication with internal microservices within the SEPP cluster is limited to IPv4 only.
- SEPP still uses IPv4 to communicate with common cluster services, including the cnDBTier and CNC Console, even in a dual stack environment.
- Communication with the Data director is currently tested on the IPv4 interface only.
- When SEPP needs to interface with Remote SEPPs or other intra-NFs, it checks the DNS response. If the DNS response contains both IPv4 and IPv6 addresses, SEPP prefers the

IPv6 address for communication. If the DNS response contains only one address family (either IPv4 or IPv6), SEPP uses that address to connect to the endpoint.

Dual Stack Architecture

Figure 3-29 Dual Stack Architecture



Dual Stack Call Flows

Edge Microservices

The edge microservice refers to the Ingress Gateway microservice, which handles incoming requests from other Network Functions (NFs). SEPP supports dual-stack operation for Ingress Gateway microservices, allowing them to accept both IPv4 and IPv6 address types from external NFs.

Note

The Ingress Gateway microservice connects to SEPP backend microservices using a single-stack IPv4 connection.

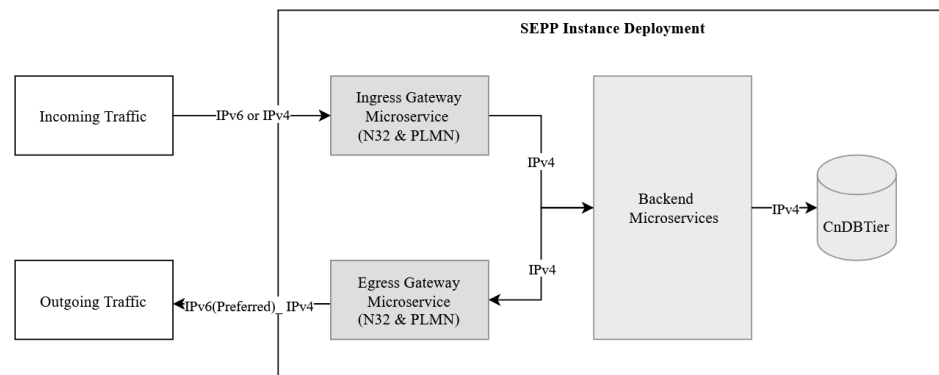
Backend Microservices

All SEPP microservices other than the Ingress Gateway microservice are considered backend microservices. SEPP supports single-stack IPv4 for all backend microservices, which communicate with each other over this single-stack network.

The Egress Gateway microservice handles outgoing requests and supports both IPv4 and IPv6. The preferred routing mode can be configured using the `global.egressRoutingMode` parameter.

For more details on configuring this parameter, refer to the "Customizing SEPP" section in the *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Figure 3-30 Dual Stck Call Flows



Ingress Gateway Communication (N32 and PLMN)

1. The requester NF sends requests to the Ingress Gateway microservice.
2. The Ingress Gateway microservice accepts traffic over either IPv4 or IPv6.
3. The Ingress Gateway forwards the requests to the backend microservices using single stack IPv4 only.
4. Backend microservices process the incoming requests and send responses back to the Ingress Gateway using single-stack IPv4 only.
5. The Ingress Gateway returns the response to the requester NF using the same IP interface received in Step 2.

Egress Gateway Communication (N32 and PLMN)

1. Backend microservices send requests to the Egress Gateway using single stack IPv4.
2. The Egress Gateway resolves the peer NF address via DNS.
 - If the peer NF has an IPv6 address, the Egress Gateway sends the request over IPv6.
 - If not, the request is sent over IPv4.

IP Address Allocation to Pods

IP address allocation to pods depends on the IP address preference set in the Kubernetes cluster. Pods do not have the privilege to choose an IP address. For example, if the Kubernetes cluster has IPv4 preferred configuration, both IPv4 and IPv6 are allocated to the pod, but the primary IP address is IPv4.

Example of a pod deployed in an IPv4 preferred infrastructure:

Status: Running

```
IP: 10.233.105.175
IPs:
IP: 10.233.105.175
IP: fd00::1:1c6b:9742:b7f1:f1a6
```

Similarly, if the Kubernetes cluster has IPv6 preferred configuration, both IPv4 and IPv6 are allocated to the pod, but the primary IP address is IPv6.

Example of a pod deployed in an IPv6 preferred infrastructure:

```
Status: Running
IP: fd00::1:1c6b:9742:b7f1:f1a6
IPs:
IP: fd00::1:1c6b:9742:b7f1:f1a6
IP: 10.233.105.175
```

The following table describes how IP address allocation, IP Family Policy, and IP Families vary based on the DeploymentMode Helm parameter configuration for services:

Table 3-24 IP Address Allocation

Infrastructure Preference	Application Preference (DeploymentMode Helm Parameter)	IP Family Policy Attribute	IP Families Attribute	Pod IP	Service IP	Endpoints
IPv4 Preferred	IPv4	SingleStack	IPv4	IPv4, IPv6	IPv4	IPv4
IPv6 Preferred	IPv4	SingleStack	IPv4	IPv6, IPv4	IPv4	IPv4
IPv4 Preferred	IPv6	SingleStack	IPv6	IPv4, IPv6	IPv6	IPv6
IPv6 Preferred	IPv6	SingleStack	IPv6	IPv6, IPv4	IPv6	IPv6
IPv4 Preferred	IPv4_IPv6 (IPv4Preferred)	RequireDualStack	IPv4 Preferred	IPv4, IPv6	IPv4, IPv6	IPv4
IPv6 Preferred	IPv4_IPv6 (IPv4Preferred)	RequireDualStack	IPv4 Preferred	IPv6, IPv4	IPv6, IPv4	IPv4
IPv4 Preferred	IPv6_IPv4 (IPv6Preferred)	RequireDualStack	IPv6 Preferred	IPv4, IPv6	IPv4, IPv6	IPv6
IPv6 Preferred	IPv6_IPv4 (IPv6Preferred)	RequireDualStack	IPv6 Preferred	IPv6, Pv4	IPv6, IPv4	IPv6

Upgrade Impact

In a dual stack environment, all the SEPP backend services are configured as Single Stack IPv4 by default. However, this is not applicable for the edge microservices (Ingress Gateway microservice) as they can choose to increase the list of ipFamilies by setting it as dual stack according to the cluster preference.

Note

The `.spec.ipFamilies` field is conditionally mutable; you can add or remove a secondary IP address family, but you cannot change the primary IP address family of an existing service.

Supported Upgrade Paths

In a dual stack environment, upgrade from single stack to dual stack is supported, provided the preferred IP family remains unchanged.

The following table summarizes all possible upgrade scenarios from single stack to dual stack deployments, along with their support status:

Table 3-25 Supported Upgrade Paths

From	To	Supported/Not supported
IPv4	IPv4-IPv6	Supported
IPv4	IPv6-IPv4	Not supported
IPv6	IPv6-IPv4	Supported
IPv6	IPv4-IPv6	Not supported

Managing Dual Stack Support**Enable**

The Dual Stack Support is a core functionality of SEPP. You do not need to enable or disable this feature.

Prerequisites:

- CNE or any CNE cloud network with Dual Stack Support (IPv4 preferred) must be enabled.
- All cluster nodes must come with IPv6 address.

Configure**Helm Configuration**

Configure the following in `n32-ingress-gateway` and `plmn-ingress-gateway`:

Single Stack IPv4

```
#deployment mode for dual stack support >> Possible values : IPv6_IPv4 (Dual Stack Mode) and IPv4 (Non-dual stack/RH mode )
deploymentMode: IPv4
```

Dual Stack IPv4/IPv6 preferred

```
#deployment mode for dual stack support >> Possible values : IPv6_IPv4 (Dual Stack Mode) and IPv4 (Non-dual stack/RH mode )
deploymentMode: IPv6_IPv4
```

N32-egress-gateway and Plmn-egress-gateway Single Stack/Dual Stack

#egressRouting mode for dual stack support >> Possible values :
IPv4, IPv6, IPv4_IPv6, IPv6_IPv4, None
egressRoutingMode: IPv6_IPv4

Note

Currently, for the gateways to assign IPv4 and IPv6 addresses to the load balancer service, `staticIpEnabled` should be set to false and the following annotation must be added to the service annotations in the `ocsepp_custom_values_<version>.yaml` file:

```
metallb.universe.tf/loadBalancerIPs: <IPv4>,<IPv6>
```

Note

In dual stack environment, if multiple (comma separated) IP addresses has to be given, do the following configurations in the `ocsepp_custom_values_<version>.yaml` file:

1. set `staticIpAddressEnabled` to false
2. Add an annotation to the N32 Ingress Gateway:

Example:

```
metallb.universe.tf/loadBalancerIPs:  
10.75.181.25,2606:b400:605:b830::8
```

For more information about above mentioned Helm parameters, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Configure

You can configure the feature using REST API and CNC Console:

- Configure using REST API: Perform the feature configurations as described in the Remote SEPP section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- Configure using CNC Console: Perform the feature configurations as described in the Remote SEPP section in [Configuring SEPP using CNC Console](#).

Observe

Following are the Support for Dual Stack specific Gateway metrics:

- Egress Gateway metrics:
 - `oc_egressgateway_incoming_ip_type`
 - `oc_egressgateway_outgoing_ip_type`
 - `oc_egressgateway_dualstack_ip_rejected_total`

- Ingress Gateway metrics:
 - oc_ingressgateway_incoming_ip_type
 - oc_ingressgateway_outgoing_ip_type

For more information on IPv6 Dual Stack Support feature related SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following steps:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.15 Multiple SEPP Instances on Shared cnDBTier Cluster

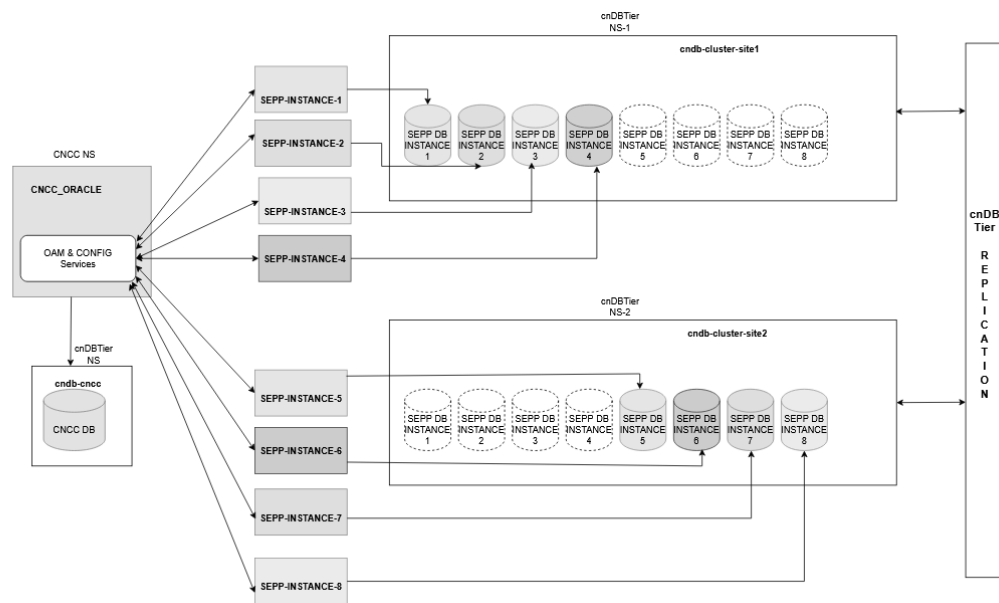
Oracle Communications Cloud Native Core, Security Edge Protection Proxy (SEPP) now supports a new deployment model that allows multiple SEPP instances to be deployed on a shared cnDBTier cluster. This approach optimizes resource utilization, as SEPP is a stateless service, and the existing cnDBTier often operates below capacity.

By leveraging a shared cnDBTier for multiple SEPP instances, significant savings in CPU resources can be achieved, compared to maintaining separate databases for each instance. This shared cnDBTier is designed with security in mind, restricting data access for each SEPP instance through the use of distinct logins and credentials during deployment. As a result, each SEPP instance can securely access its dedicated database while utilising the same cnDBTier within the Kubernetes cluster. For instance, deploying four SEPP instances within the same cluster can lead to considerable resource efficiency compared to deploying each instance with its own database.

In this deployment model, 1+1 GR redundancy only supported with maximum four SEPP instances in one cluster to avoid operational scaling issues. This deployment model can only be used for SEPP instances deployed on the same CNE cluster.

Detailed Description

In this deployment model, multiple SEPPs are deployed on a shared cnDBTier cluster. The following diagram illustrates the comprehensive architecture of the deployment model.

Figure 3-31 Multiple SEPP instances on shared cnDBTier cluster architecture

This deployment model supports a two-site georeplication setup with four SEPP instances deployed per site and per cluster. Each cnDBTier is configured with a single replication channel.

The Key Concepts of the feature are the following:

- **Shared cnDBTier in the same Cluster:** A single database cluster is used to streamline data storage and management across multiple SEPP instances.
- **Credential-based Segregation:** Seamlessly segregate data access and permissions based on unique credentials or access keys assigned to each NF user.

Shared cnDBTier in the same Cluster:

• cnDB-cluster-site1

The MySQL databases for SEPP-INSTANCE-1, SEPP-INSTANCE-2, SEPP-INSTANCE-3, and SEPP-INSTANCE-4 are configured on the cnDB-cluster-site1 in a shared mode, separated by SEPP user credentials. The databases of SEPP-INSTANCE-5, SEPP-INSTANCE-6, SEPP-INSTANCE-7, and SEPP-INSTANCE-8 are replicated on this cluster.

• cnDB-cluster-site2

The MySQL databases for SEPP-INSTANCE-5, SEPP-INSTANCE-6, SEPP-INSTANCE-7, and SEPP-INSTANCE-8 are configured on the cnDB-cluster-site2 in shared mode, separated by SEPP user credentials. The databases of SEPP-INSTANCE-1, SEPP-INSTANCE-2, SEPP-INSTANCE-3, and SEPP-INSTANCE-4 are replicated on this cluster.

Credential-based Segregation:

Access control for each SEPP database is managed by the root user on the cnDBTier cluster. All SEPP databases are replicated to another configured site to enable recovery in case the database becomes corrupted. All SEPP instances are configured on CNC Console, and users can be assigned instance-specific roles to restrict access to other SEPP instances.

Installing and Managing multiple SEPP instances on shared cnDBTier cluster

Multiple SEPP instances on the shared cnDBTier cluster deployment model is enabled by default on SEPP. To enable the access restriction per SEPP instance on CNC Console, do the following:

1. Set the CNC Console parameter `instanceLevelAuthorizationEnabled` to true in the `occncc_custom_values_.yaml` file.
2. Run Helm upgrade command to upgrade CNC console.
3. When Helm upgrade completes, all existing users will have a new role "INSTANCE_ALL" assigned by default.

To assign a NF-specific role, the admin must unassign the "INSTANCE_ALL" role from the user and map the user to the desired NF role.

cnDBTier Resource Profile Updates

The following profile updates are required in the `occndbtier_custom_values_<version>.yaml` file for this feature.

- Install the cnDBTier using the 'New Values' mentioned in the following table. The 'New Values' must be updated in the `occndbtier_custom_values_<version>.yaml` file.
- If the cnDBTier is already installed, perform an upgrade using the procedure mentioned in the *Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

Table 3-26 cnDBTier Parameter Values

Parameter	Default Values (in CV file)	New Values (to be Updated in CV file)
MaxNoOfTables	1024	3000
MaxNoOfAttributes	5000	24000
MaxNoOfOrderedIndexes	1024	3700

Configure

The following are the configurations for the multiple SEPP instances on shared cnDBTier cluster deployment model:

Helm Configuration

This section provides the multiple SEPP instances on shared cnDBTier cluster deployment model related Helm configurations.

The following timeout parameter must be configured to check whether database connectivity with Service is healthy or not in the microservices such as `cn32f-svc`, `cn32c-svc`, `pn32f-svc`, `pn32c-svc` and `config-mgr-svc`:

```
configs:
  domainLabels: 3
  rateLimitingConfigMap: rss-ratelimit-map
  egressRateLimitingConfigMap: egress-ratelimit-map
  dbCheckRefreshTimeout: 30000
```

Table 3-27 Details of Helm Parameter

Parameter	Description	Details
dbCheckRefreshTimeout	This is a mandatory parameter. This value represents time interval that checks whether database connectivity with Service is healthy or not.	Data Type: Integer Default Value: 30000ms

MySQL Configurations

This section provides the information about the MySQL configurations required to configure this deployment model.

While configuring SEPP, user has to create MySQL credential secrets. The user can configure unique credentials while creating MySQL secrets, followed by configuring unique credentials for every SEPP instance.

Note

SEPP database and SEPP BACKUP DB can be created with NF_INSTANCE_ID in db name, whereas "-" is replaced by "_".

To create the user and the database, see *Configuring database, Creating Users, and Granting Permissions* section of *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Observe

CNC Console IAM Configurations

This section provides the information about the CNC Console IAM configurations required to configure this deployment model.

Log in to the CNC Console using admin credentials to create user and assign NF specific roles to the user.

The following roles are predefined in the CNC Console IAM:

- **SEPP_READ:** The assigned user can perform the read operation for NFs with this permission. If the user has a SEPP_READ role, then the user can only read configurations of any MOs configurations within the Policy and cannot write or update or delete any record.
- **SEPP_WRITE:** With this permission, the assigned user can perform create, read, update, and delete operations for NFs. If the user has SEPP_WRITE then the user can read, write, update, or delete any MOs configurations within the NF.
- **Admin:** The user assigned this role has access to all resources (NF resources and CS resources) within the CNC Console application. The user can create, read, update, and delete MO configurations for all NFs supported by CNC Console.
- **INSTANCE_ALL:** A user with INSTANCE_ALL role can access all instances, provided they have ADMIN, <NF>_READ, or <NF>_WRITE role, and Cluster level role in case it is a multicluster deployment. This role is automatically assigned to all local users during the

first upgrade when this deployment model is enabled. If an operator wants to restrict a user to a particular instance, then they have to unassign INSTANCE_ALL role and assign any of the instance level roles to the user.

- **Instance specific role:** Instance Level roles allow users to have access to specific NF instances. A user can be associated with single or multiple instance roles. Instance Role name must match the instance ID of that particular instance given in Helm configuration `inglobal.instances[i].id`.

Combinations of SEPP user roles

- ADMIN/SEPP_READ/SEPP_WRITE + INSTANCE_ALL: Here user will have access to all SEPP Instances.
- ADMIN/SEPP_READ/SEPP_WRITE + sepp_instance_specific role: Here user will have access to a specific SEPP instance.

For more details about CNC Console Configurations, see "Types of Roles in CNC Console" section in the *Cloud Native Configuration Console User Guide*.

Following are the deployment model specific metrics:

- ocsepp_cn32f_database_connectivity_healthy
- ocsepp_pn32f_database_connectivity_healthy
- ocsepp_cn32c_database_connectivity_healthy
- ocsepp_pn32c_database_connectivity_healthy
- ocsepp_configmgr_database_connectivity_healthy

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.16 Proactive Status Updates on SEPP

Oracle Communications Cloud Native Core, Security Edge Protection Proxy (SEPP) supports Proactive Status Updates on SEPP feature.

In the previous releases, consumer NFs sent Service Based Interface (SBI) message requests to Remote SEPP without checking the status of Remote SEPP because there was no health check mechanism implemented at SEPP.

Using this feature, consumer NFs can determine the health or connection status of Remote SEPP before forwarding any SBI message request to it. This feature allows consumer NFs or Consumer SEPPs to perform the following health check verification on Remote SEPPs:

- Determines if SEPP is reachable.

- Determines if SEPP is available to respond to SBI message requests.

Assumptions

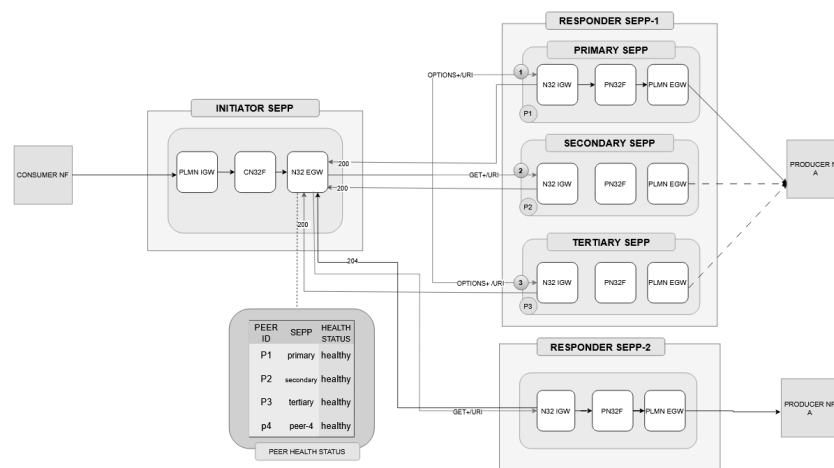
The following are the assumptions of the feature:

- This feature applies only to the N32 Egress gateway and N32 Ingress gateway.
- The user can enable or disable the feature for each Remote SEPP Peer. If the healthAPIURI and healthAPIMethod fields are blank in the Remote SEPP configuration, the feature will be disabled for that peer.
- On N32 Ingress Gateway, the response codes allowed are 200,201,202,203,204,205,206,207,208, and 226. This list is exhaustive.
- On N32 Egress Gateway, the response codes that will be used to mark peers as healthy are 200,204,400, 401,403,404, 405, 501, and 503. This list is exhaustive.

Detailed Description

The following diagram represents the detailed architecture of this feature.

Figure 3-32 Proactive Status Updates on SEPP Architecture



1. The initiator SEPP has 4 Remote Peer SEPPs configured; Primary, Secondary, and Tertiary - (associated with the Responder SEPP-1 Remote SEPP set), and Peer-4, associated with Responder SEPP-2.
2. In the default case, when the feature is not present or is disabled, the initiator SEPP communicates with the Primary SEPP. If the Primary SEPP goes down for some reason, it then reroutes to the Secondary SEPP. If the Secondary SEPP also goes down, the initiator SEPP reroutes traffic to the Tertiary SEPP.
3. The above process introduces latency for the outbound traffic of the SEPP due to the failed peer SEPPs.
4. With the proactive status feature enabled, the N32 Egress gateway sends a health check request using a configured Health API URI and method combination (GET and OPTIONS methods supported) towards the Peer SEPPs. The response code received will be compared against a list of pre-configured response codes to determine the health status of each Remote Peer SEPP. These messages will be sent at a configured periodic interval.

5. The N32 Ingress gateway will receive the URI and method and compare them against the configured settings. It will then respond with the configured response code, which the N32 Egress Gateway will use to proactively determine the health status of each peer SEPP. The user can configure the response sent from the N32 Ingress gateway within the range of 2xx.
6. If the feature is enabled as described in point 2, initially the traffic will be routed to the Primary SEPP. If the Primary SEPP goes down, the health API will receive a response code not configured on the N32 Egress gateway, and the Primary SEPP will be marked as UNHEALTHY.
7. Considering the Responder SEPP-1 peers in this example, as soon as the Primary SEPP is marked as UNHEALTHY, the initiator SEPP routes the traffic towards the next healthy peer (Secondary SEPP) without any re-routing. If the Secondary SEPP also goes down, it will be marked as UNHEALTHY and traffic will be directly routed to the Tertiary SEPP without any re-routing. This reduces latency and improves the overall efficiency of the system.

Managing Proactive Status Updates on SEPP Feature

Proactive Status Updates on the SEPP feature is disabled by default. It is enabled using Helm and CNC Console or REST based configurations on N32 Egress Gateway and N32 Ingress Gateway.

To enable the feature, do the following:

1. Set the `healthCheckMonitoring.enabled` parameter to true in the N32 Ingress Gateway section of `ocsepp_custom_values_<version>.yaml` file.

```
#To enable feature on IGW
healthCheckMonitoring:
  enabled: true
```

2. Set the `seppPeerHealthCheck` parameter to true in the N32 Egress Gateway section of `ocsepp_custom_values_<version>.yaml` file.

```
#To enable feature on EGW for
  seppPeerHealthCheck: true
#Response codes that will be considered to mark peer healthy
seppPeerHealthCheckCodes: "200,204,400,401,403,404,405,501,503"
```

At CNC Console:

① Note

To enable the feature at the CNC console, ensure that the feature parameters on Helm are set to true.

Perform the following procedure to enable the support for the Proactive Status Updates on the SEPP feature:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.

2. The **Gateway** and **cnDBTier** appear underneath. Select **Gateway** and navigate to the **IGW** option to configure the Ingress Gateway APIs. The **PLMN Ingress Gateway** and **N32 Ingress Gateway** appear underneath.
3. Click **Health Check Configuration** under **N32 Ingress Gateway**. The Health Check Configuration page is displayed.
4. Click the Edit icon to modify the **Health Check Configuration**. The **Health Check ConfigurationOptions** page appears.
5. Set **Enabled** to True.

Perform the following procedure to configure the Remote SEPP:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Remote SEPP**.
2. The list of all the Remote SEPPs configured along with the parameters appears on the right pane.
3. Click the Edit icon to modify a specific parameter. The page is enabled for modification.
4. Add the **healthApiPath** and **healthApiMethod** in the **Remote SEPP**. The **healthApiMethod** drop-down has only GET and OPTIONS.

For more details about CNC Console Configurations, see Configuring SEPP using CNC Console section in the *Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide*.

At REST API:

- To enable on N32 Ingress Gateway, Set enabled to true in healthCheckConfigOptions REST API.

```
curl http://<release-name>-config-mgr-svc:<port>/sepp/nf-common-component/v1/igw/n32/healthCheckConfigOptions -XPUT -H 'Content-type: application/json' -d '{
```

```
  "enabled":true,
  "healthCheckConfigObject": {
    "healthCheckResponseCode": 200,
    "healthCheckRequestMethod":"OPTIONS",
    "healthCheckResourceUri": "/health-check/v1/status"
  }
}'
```

or

```
curl http://<release-name>-config-mgr-svc:<port>/sepp/nf-common-component/v1/igw/n32/healthCheckConfigOptions -XPUT -H 'Content-type: application/json' -d '{
```

```
  "enabled":true,
  "healthCheckConfigObject": {
    "healthCheckResponseCode": 200,
    "healthCheckRequestMethod":"GET",
    "healthCheckResourceUri": "/health-status/v1/status"
  }
}'
```

- Content-Type: User must send Content-Type: application/json when configuring N32 Ingress Gateway health checks through ConfigMgr. For backward compatibility, ConfigMgr must process a valid JSON request when this header is missing or malformed.
- To enable the feature on the N32 Egress gateway for a particular Remote SEPP, add the parameters healthApiPath and healthApiMethod in the Remote SEPP. The healthApiMethod parameter supports only GET and OPTIONS.

Note

By default, these parameter values are null. After enabling the "seppPeerHealthCheck" Helm parameter, the user needs to add the above parameters to the Remote SEPP for the peer whose health status they want to monitor.

```
curl http://<release-name>-config-mgr-svc:<port>/sepp-configuration/v1/
remotesep -XPOST -H 'Content-type: application/json' -d '{
  "name": "SEPP-2",
  "plmnIdList": [{"mcc": "111", "mnc": "345"}],
  "domain": "mnc345.mcc111.3gppnetwork.org",
  "seppFqdn": "sepp-ats-rel-stubserver",
  "securityCapabilityList": ["TLS"],
  "remoteSeppIpAddress": "10.233.52.22",
  "port": 8443,
  "apiPrefix": "",
  "isEnabled": "TRUE",
  "apiVersion": "v1",
  "healthApiPath": "/health-check/v1/status",
  "healthApiMethod": "OPTIONS"}
```

or

```
curl http://<release-name>-config-mgr-svc:<port>/sepp-configuration/v1/
remotesep -XPOST -H 'Content-type: application/json' -d '{
  "name": "SEPP-2",
  "plmnIdList": [{"mcc": "311", "mnc": "345"}],
  "domain": "mnc345.mcc311.3gppnetwork.org",
  "seppFqdn": "sepp-rel-stubserver",
  "securityCapabilityList": ["TLS"],
  "remoteSeppIpAddress": "10.233.52.255",
  "port": 443,
  "apiPrefix": "",
  "isEnabled": "TRUE",
  "apiVersion": "v1",
  "healthApiPath": "/health-status/v1/status",
  "healthApiMethod": "GET"}
```


The user can also configure the feature during runtime (that is, without deleting the Remote SEPP Set). To do so, run the following command:

```
curl http://<release-name>-config-mgr-svc:<port>/sepp-configuration/v1/
remotesep/SEPP-2 -XPUT -H 'Content-type: application/json' -d '
{
  "name": "SEPP-2",
  "plmnIdList": [
    {
      "mcc": "111",
      "mnc": "345"
    }
  ],
  "domain": "mnc345.mcc111.3gppnetwork.org",
  "seppFqdn": "sepp-ats-rel-stubserver",
  "securityCapabilityList": [
    "TLS"
  ],
  "remoteSeppIpAddress": "10.233.52.22",
  "port": 8443,
  "apiPrefix": "",
  "isEnabled": "TRUE",
  "apiVersion": "v1"
}
```

Note

Add the following two parameters if not present:

- "healthApiPath": "/health-check/v1/status"
- "healthApiMethod": "OPTIONS"

Note

To disable the feature on a Remote SEPP Peer, run the following steps:

1. Delete the Remote SEPP Set associated with the Peer.
2. Run the following command to modify the Remote SEPP Peer:

```
curl http://<release-name>-config-mgr-svc:<port>/sepp-configuration/v1/remotesep/SEPP-2 -XPUT -H 'Content-type: application/json' -d '{
  "name": "SEPP-2",
  "plmnIdList": [{"mcc": "111", "mnc": "345"}],
  "domain": "mnc345.mcc111.3gppnetwork.org",
  "seppFqdn": "sepp-ats-rel-stubserver",
  "securityCapabilityList": ["TLS"],
  "remoteSeppIpAddress": "10.233.52.22",
  "port": 8443,
  "apiPrefix": "",
  "isEnabled": "TRUE",
  "apiVersion": "v1"
}'
```

3. Create the Remote SEPP Set again.

Run the following command to modify frequency, timeout, failureThreshold, or SuccessThreshold parameters of the health check requests being sent:

```
curl http://127.0.0.1:9090/sepp/nf-common-component/v1/egw/n32/peermonitoringconfiguration -XPUT -H 'Content-type: application/json' -d '{
  "enabled": false, "timeout": 1000, "frequency": 6000, "failureThreshold": 3, "successThreshold": 3}'
```

Note

Do not change the value for enabled for this feature. It must remain as false.

Configure

You can configure the feature using REST API and CNC Console.

- **Configure using REST API:** Perform the feature configurations as described in Health Check Configuration (N32 IGW), Peer monitoring configuration and Remote SEPP REST APIs in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- **Configure using CNC Console:** Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Observe

Following are the feature specific metrics:

- `oc_egressgateway_peer_health_status`
- `oc_egressgateway_peer_health_ping_request_total`
- `oc_egressgateway_peer_health_ping_response_total`
- `oc_egressgateway_peer_health_status_transitions_total`
- `oc_ingressgateway_health_check_incoming_ping_total`

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.17 Traffic Segregation

This feature provides end-to-end traffic segregation to SEPP based on traffic types. Within a Kubernetes cluster, traffic segregation can divide applications or workloads into distinct sections such as OAM, SBI, Kubernetes control traffic, etc. The Multus CNI container network interface (CNI) plugin for Kubernetes enables attaching multiple network interfaces to pods to help segregate traffic from each SEPP microservice.

This feature addresses the challenge of logically separating IP traffic of different profiles, which are typically handled through a single network (Kubernetes overlay). The new functionality ensures that critical networks are not cross-connected or sharing the same routes, thereby preventing network congestion.

With traffic segregation, operators can segregate traffic to external feeds and applications more effectively. Previously, all external traffic was routed through the same external network, but now, egress traffic from the SEPP pods can be directed through non-default networks to third-party applications. This separation is achieved by leveraging cloud-native infrastructure and the load balancing algorithms in OCCNE.

The feature supports the configuration of separate networks, Network Attachment Definitions (NADs), and the Cloud Native Load Balancer (CNLB). These configurations are crucial for enabling cloud native load balancing, facilitating ingress-egress traffic separation, and optimizing load distribution within SEPP.

Prerequisites

The CNLB feature is only available in SEPP if OCCNE is installed with CNLB and Multus.

Assumptions

- The CNLB cluster must be installed with the required additional signaling and OAM interfaces during the initial installation phase.
- The target subnets/CIDR ranges for Remote SEPP Sets must be defined in the `cnlb.ini` configuration file to enable traffic segregation across different remote SEPPs.

- For onboarding a Remote SEPP, the partner will provide the relevant FQDNs and target CIDR ranges, which can then be dynamically added to the CNLB configuration at runtime.
- The CNLB feature is only available in SEPP, Roaming Hub, and Hosted SEPP, if OCCNE is installed with CNLB and Multus.

Cloud Native Load Balancer (CNLB)

CNE provides Cloud Native Load Balancer (CNLB) for managing the ingress and egress network as an alternate to the existing LBVM, lb-controller, and egress-controller solutions. You can enable or disable this feature only during a fresh CNE installation. When this feature is enabled, CNE automatically uses CNLB to control ingress traffic. To manage the egress traffic, you must preconfigure the egress network details in the `cnlb.ini` file before installing CNE.

For more information about enabling and configuring CNLB, see *Oracle Communications Cloud Native Core, Cloud Native Environment User Guide*, and *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

Network Attachment Definitions for CNLB

A Network Attachment Definition (NAD) is a resource used to set up a network attachment, in this case, a secondary network interface to a pod. SEPP supports two types of CNLB NADs:

1. Ingress Network Attachment Definitions

Ingress NADs are used to handle inbound traffic only. This traffic enters the CNLB application through an external interface service IP address and is routed internally using interfaces within CNLB networks.

- **Naming Convention:** `nf-<service_network_name>-int`

2. Egress Only Network Attachment Definitions

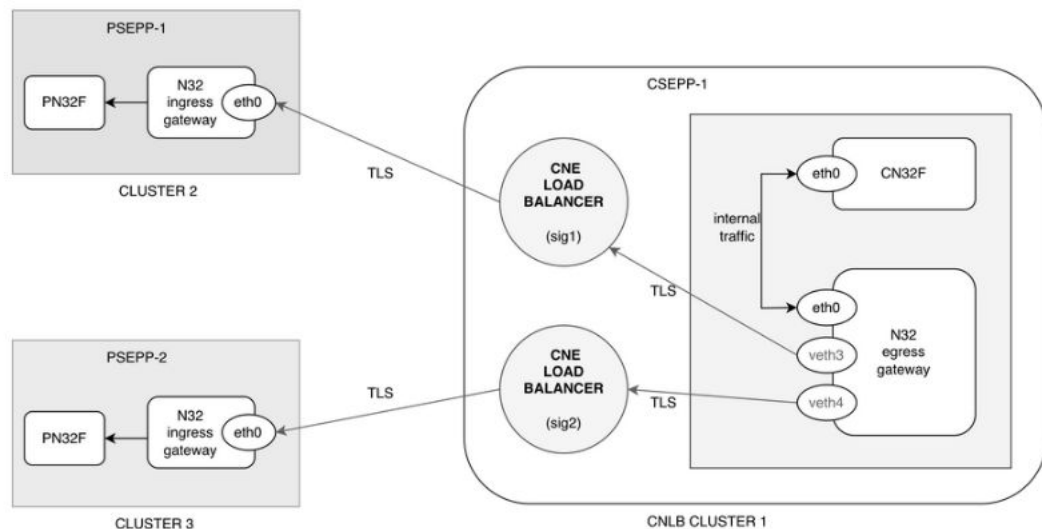
Egress Only NADs enable outbound traffic only. An NF pod can initiate traffic and route it through a CNLB application, translating the source IP address to an external egress IP address. An egress NAD contains network information to create interfaces for NF pods and routes to external subnets.

- **Requirements:**
 - Ingress NADs are already created for the desired internal networks.
 - Destination (egress) subnet addresses are known beforehand and defined under the `cnlb.ini` file's `egress_dest` variable to generate NADs.
 - The use of an Egress NAD on a deployment can be combined with Ingress NADs to route traffic through specific CNLB apps.

Egress Traffic Segregation

The Egress Traffic manages the flow of N32 signaling traffic from the Core Network (CN32F) within the CSEPP to Remote PSEPPs. This process ensures secure and efficient routing of signaling data across the network while maintaining proper segregation between multiple PSEPP peers.

The following diagram represents the traffic egressing from `n32-egress-gateway` to multiple Remote SEPPs.

Figure 3-33 Egress Traffic Segregation**Key Components:**

- **CN32F (Core Network N32 Forwarding Function):** Generates outbound N32 requests intended for remote PSEPPs, initiating the signaling flow.
- **N32 Egress Gateway:** Handles the outbound N32 traffic from the CSEPP, applying security protocols such as TLS encapsulation before forwarding the traffic.
- **eth0 (N32 Egress Gateway Interface):** The physical or virtual network interface on the N32 Egress Gateway used to transmit TLS-secured N32 traffic out of the CSEPP.
- **veth3/veth4 (Virtual Interfaces):** Dedicated virtual interfaces on the N32 Egress Gateway, each used for traffic directed to specific remote PSEPP peers (PSEPP1 and PSEPP2, respectively).
- **CNE Load Balancer (sig1/sig2):** A dedicated load balancer for each remote PSEPP. Each load balancer instance is bound to a unique signaling interface (sig1 for PSEPP1, sig2 for PSEPP2), ensuring traffic is routed correctly and prevents mixing of traffic between different PLMNs.

Traffic Flow (Egress):

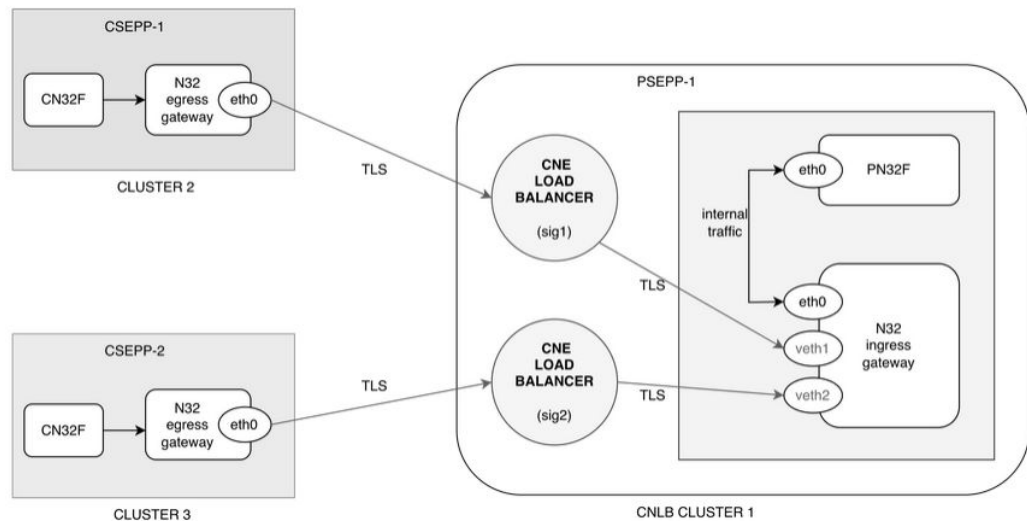
1. **Generation of Request:** The CN32F generates an N32 request targeted at a remote PLMN.
2. **Forwarding to Egress Gateway:** The request is sent to the N32 Egress Gateway within the CSEPP.
3. **TLS Encapsulation and Forwarding:** The N32 Egress Gateway encapsulates the request in a secure TLS session and forwards it via the appropriate virtual interface (veth3 or veth4) to the corresponding CNE Load Balancer at the remote PSEPP.
4. **Segregation of Traffic:** To maintain proper traffic segregation, each outbound request from the CSEPP is directed to its corresponding CNE Load Balancer instance (sig1 for PSEPP1, sig2 for PSEPP2), ensuring that no cross-PLMN traffic mixing occurs at the load balancer level.

This feature provides secure and efficient handling of N32 egress traffic, ensuring clear traffic paths and enhanced network performance.

Ingress Traffic Segregation

The following diagram represents the traffic ingressing from the external cluster (different NFs like AMF) to Plmn-ingress-gateway.

Figure 3-34 Ingress Traffic Segregation



The Ingress Traffic feature manages the secure processing and forwarding of N32 signaling traffic arriving at the PSEPP from remote CSEPPs. This process ensures that incoming signaling data is properly validated, decrypted, and routed to the appropriate Producing Network N32 Forwarding function (PN32F) within the home PLMN (Public Land Mobile Network).

Key Components:

- **CNE Load Balancer (sig1/sig2):** Each CNE Load Balancer instance is dedicated to a specific CSEPP peer, receiving TLS-terminated N32 traffic and forwarding it to the appropriate N32 Ingress Gateway. The load balancer uses sig1 and sig2 interfaces to direct traffic from CSEPP-1 and CSEPP-2, respectively.
- **N32 Ingress Gateway:** Handles inbound N32 traffic, performing security validations such as JWS (JSON Web Signature) verification before forwarding the traffic to the PN32F for processing.
- **veth1 / veth2 (Virtual Ethernet Interfaces):** These virtual interface pairs route traffic from the CNE Load Balancer to the N32 Ingress Gateway. veth1 handles traffic from CSEPP-1 (via sig1), while veth2 handles traffic from CSEPP-2 (via sig2), ensuring network-level isolation within the PSEPP pod or namespace.
- **eth0 (N32 Ingress Gateway Interface):** The network interface through which validated N32 traffic is forwarded as internal traffic within the PSEPP to the PN32F.
- **PN32F (Producing Network N32 Forwarding Function):** Receives the validated and decrypted N32 request and processes it in the context of the home PLMN, initiating the appropriate actions.

Traffic Flow:

1. **Reception by CNE Load Balancer:** The CNE Load Balancer (sig1 or sig2) receives an inbound TLS session from the corresponding remote CSEPP peer.
2. **Forwarding to N32 Ingress Gateway:** The TLS traffic is forwarded from the CNE Load Balancer to the N32 Ingress Gateway via the appropriate virtual Ethernet interface — veth1 for CSEPP-1 traffic (sig1) or veth2 for CSEPP-2 traffic (sig2).
3. **Security Processing:** The N32 Ingress Gateway performs necessary security checks (e.g., JWS verification) on the inbound N32 message before forwarding it onward.
4. **Forwarding to PN32F:** Once validated, the N32 Ingress Gateway forwards the traffic via eth0 as internal traffic to the PN32F.
5. **Processing by PN32F:** The PN32F processes the N32 request in the context of the home PLMN, handling the appropriate network operations.

This feature ensures secure and efficient processing of inbound N32 traffic, facilitating reliable signaling between remote CSEPPs and the home network.

Managing Ingress and Egress Traffic Segregation

Enable

This feature is disabled by default. To enable this feature, you must configure the network attachment annotations in the `ocsepp_custom_values_<version>.yaml` file.

Configuration

This section outlines the steps to configure additional subnets/interfaces on the CNLB (Core Network Load Balancer) cluster. This configuration is cluster-dependent and can be customized according to your specific requirements.

1. **CNLB Configuration**
This step is to configure the CNLB cluster by editing the CNLB configuration file.
 - a. Navigate to the CNLB configuration directory:
`/var/ocne/cluster/{OCCNE_CLUSTER}`
 - b. Edit the CNLB configuration file:
`vi cnlb.ini`
 - c. Define network sections for your desired subnets. For example, you can define sections for oam, sig1, and sig2. The following is an example configuration. Be sure to replace all IDs, CIDRs, gateways, and IP addresses with those specific to your environment.

Example: cnlb.ini

```
[cnlb]
[cnlb:vars]
cnlb_replica_cnt = 4
ext_net_attach_def_plugin = "macvlan"
cnlb_node_label = True

[cnlb:children]
# Network names to be used for pod and external networking, these
# networks and ports
# will be created and attached to hosts in cnlb host group.

oam
```

```
sig1
sig2
```

```
[oam:vars]
service_network_name = "oam"
subnet_id = "d377f04e-baef-4473-a992-d9e344ad0205"
network_id = "06effdde-458c-401b-a29f-27eaab11e46a"
external_network_range = "10.75.152.0/23"
external_default_gw = "10.75.152.1"
egress_ip_addr = ["10.75.152.178", "10.75.153.148"]
internal_network_range = "171.16.0.0/24"
service_ip_addr =
["10.75.152.240", "10.75.152.155", "10.75.152.175", "10.75.152.103", "10.75.
152.124", "10.75.152.203"]
egress_dest = ["10.75.151.0/24"]
```

```
[sig1:vars]
service_network_name = "sig1"
subnet_id = "d377f04e-baef-4473-a992-d9e344ad0205"
network_id = "06effdde-458c-401b-a29f-27eaab11e46a"
external_network_range = "10.75.152.0/23"
external_default_gw = "10.75.152.1"
egress_ip_addr = ["10.75.152.168", "10.75.153.133"]
internal_network_range = "172.16.0.0/24"
service_ip_addr = ["10.75.152.146", "10.75.152.83"]
egress_dest = ["10.75.151.0/24"]
shared_internal = True
```

```
[sig2:vars]
service_network_name = "sig2"
subnet_id = "d377f04e-baef-4473-a992-d9e344ad0205"
network_id = "06effdde-458c-401b-a29f-27eaab11e46a"
external_network_range = "10.75.152.0/23"
external_default_gw = "10.75.152.1"
egress_ip_addr = ["10.75.153.1", "10.75.153.186"]
internal_network_range = "173.16.0.0/24"
service_ip_addr = ["10.75.152.68", "10.75.153.115"]
egress_dest = ["10.75.214.0/23"]
shared_internal = True
```

2. Apply Network Updates

Once the CNLB configuration is complete, the next step is to apply the network updates to provision and/or attach the required interfaces, ensuring proper network segregation as defined in the cnlb.ini file.

a. Source the environment:

To begin, source the environment to load the necessary variables and configurations:

```
source opener.sh
```

b. Run the network update script:

After sourcing the environment, execute the network update script, which will apply the configuration and provision the interfaces accordingly:

```
./installers/updNetworks.py
```


This script will ensure the interfaces are created and attached as per the specifications in the cnlb.ini file, providing the required network segregation.

For detailed instructions and parameter details, refer to the *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

Note

The configuration for adding interfaces may vary and change depending on your environment or requirements. For the latest information, refer to the *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

SEPP Configurations for Egress Gateway

To attach multiple interfaces corresponding to different PSEPP subnets in the n32-egress-gateway deployment, annotations must be configured in the `ocsepp_custom_values_<version>.yaml` file.

Sample Configurations for n32-egress-gateway:

1. Single Interface:

For attaching a single interface (nf-sig1-egr1), use the following annotation:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig1-egr1@nf-sig1-egr1
```

2. Multiple Interfaces:

For attaching multiple interfaces (nf-sig1-egr1 and nf-sig2-egr2), use the following annotation:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig1-egr1@nf-sig1-egr1, default/nf-sig2-egr2@nf-sig2-egr2
```

Note

To add or update any annotations, a Helm upgrade will be required to apply the changes.

Dynamic NAD Addition, Deletion, and Modification (Recommended)

To avoid pod restarts when adding, deleting, or modifying Network Attachment Definitions (NADs) in a deployment, the OCCNE team provides the reconciler-manager.py utility.

The `reconciler-manager.py` script is a command-line tool for managing Dynamic Network Attachments (NADs) and Cloud Native Load Balancer (CNLB) annotations through the k8s-reconciler ConfigMap. It offers a safe and validated interface for performing operations that would otherwise require complex kubectl patch commands with error-prone JSON escaping.

Usage

```
python provision/platform/common/scripts/reconciler-manager.py <command> [flags]
```

Available Commands

- `check` – Verify that the ConfigMap is available and readable.

- list-nads – Display the configured NADs and destination routes.
- list-apps – List the existing application entries and their annotations.
- config-app – Create or update application annotations.
- delete-app – Remove an application entry from a namespace block.

Configure an Application (config-app)

The config-app command supports multiple modes.

Configure NADs Only

```
python provision/platform/common/scripts/reconciler-manager.py
config-app \
--namespace sepp \
--app n32-egress-gateway \
--nad sig-deg1 --nad sig-deg2
```

Example Output:

Delete App Entry

```
Namespace block:  sepp
App label:       n32-egress-gateway
Current config:
{
  "annotations": {
    "k8s.v1.cni.cncf.io/networks": "default/sig-deg1@sig-deg1"
  }
}
```

Delete this app entry? This cannot be undone. [y/N]: y

✓ App 'n32-egress-gateway' deleted from namespace block 'sepp'.

Note

- Apps are added to the reconciler script with "app" label of the deployment. For example, for adding/modification/deletion of NADs on n32-egress-gateway app=n32-egress-gateway must be used.
- Reconciler script takes precedence over the manual patching of the annotations in the deployment. For manual patching only, you need to make sure 'app' is removed from the reconciler script list.

SEPP Configurations for Ingress Gateway

Similarly, annotations need to be configured in the n32-ingress-gateway deployment to attach multiple interfaces corresponding to different PSEPP subnets. These annotations should also be included in the ocsepp_custom_values_<version>.yaml file.

Sample Configurations for n32-ingress-gateway:**1. Single Interface:**

For attaching a single interface (nf-sig1-ie1), use the following annotation:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig1-ie1@nf-sig1-ie1
oracle.com.cnc/cnlb: '[{"backendPortName": "igw-https", "cnlbIp": "10.75.152.83",
"cnlbPort": "443"}]'
```

2. Multiple Interfaces:

For attaching multiple interfaces (nf-sig1-ie1 and nf-sig2-ie2), use the following annotation:

```
k8s.v1.cni.cncf.io/networks: default/nf-sig1-ie1@nf-sig1-ie1, default/nf-sig2-ie2@nf-
sig2-ie2
oracle.com.cnc/cnlb: '[{"backendPortName": "igw-https", "cnlbIp": "nf-sig1-
ie1/10.75.152.83, nf-sig2-ie2/10.75.152.68", "cnlbPort": "443"}]'
```

```
oracle.com.cnc/ingressMultiNetwork: "true"
```

Note

To add or update annotations in the n32-ingress-gateway, a helm upgrade is required.

Onboarding New Remote SEPP Partner on CNLB Cluster**Manual patching of Annotations**

Following steps must be followed to onboard the new Remote SEPP Partner

1. The target CIDR range for the Remote partner must be defined in the cnlb.ini file. Corresponding external egress IP must also be defined for the target CIDR range. New interface will be defined in the pod for the CIDR block defined in the cnlb.ini file.

```
[sig3:vars]
service_network_name = "sig2"
subnet_id = "d377f04e-baef-4473-a992-d9e344ad0205"
network_id = "06effdde-458c-401b-a29f-27eaab11e46a"
external_network_range = "10.75.152.0/23"
external_default_gw = "10.75.152.1"
egress_ip_addr = ["10.75.153.1", "10.75.153.186"]
internal_network_range = "173.16.0.0/24"
service_ip_addr = ["10.75.152.68", "10.75.153.115"]
egress_dest = ["10.75.214.0/23"]
shared_internal = True
```

2. Update the annotation field in the deployment section for the egress-gateway. Sample configuration is provided below:

```
deployment:
  customExtension:
    labels: {}
    annotations:
      k8s.v1.cni.cncf.io/networks: default/nf-sig1-egr1@nf-sig1-egr1,
default/nf-sig2-egr1@nf-sig2-egr1
```

Here, sig2 is the new interface defined in the cnlb.ini file. egr1 and egr2 are the egress IPs configured for this interface. You can assign either of these egress IPs to the interface, and all traffic originating from sig2 will be routed through the selected egress IP address.

3. Perform an upgrade the SEPP application to reload the annotation changes. For more details about upgrade, refer "Upgrading SEPP" section in the *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Both gateways require specific configuration in the `ocsepp_custom_values_<version>.yaml` file to ensure network interfaces are properly assigned.

For more information about Traffic Segregation configuration, see "Configuring Traffic Segregation" section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Observe

There are no Metrics, KPIs, or Alerts available for this feature.

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.18 Support for Originating Network Id Header Validation, Insertion, and Transposition

This feature enables the insertion or transposition of either the 3gpp-Sbi-Originating-Network-Id header or the 3gpp-Sbi-Asserted-Plmn-Id header into Sbi request messages. It is expected that the originator of a request can be easily identified, but there are some scenarios where the originating network information may not be conveyed in the Sbi requests to the home network. In such scenarios, this feature infer the originating PLMN ID and populates the required header in the Sbi request.

Feature Overview

The feature supports the following three functionalities:

- Header Value Validation using Cat 2 Network ID Validation feature
- Header Addition
- Header Transposition

Header Value Validation

If 3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id header is already present in the incoming request and the user wants to validate the PLMN ID in this header, then the user has to enable and configure the Cat 2 - Network ID validation feature.

- At C-SEPP (consumer SEPP), Cat 2 feature validates the PLMN ID value present in the header against the configured local SEPP PLMN IDs.

- At P-SEPP (producer SEPP), Cat 2 feature validates the PLMN ID value present in the header against the configured remote SEPP PLMN IDs.

Request is either allowed and forwarded or discarded as per the validation result and the Cat 2 error configurations.

Note

- The user must enable the Cat 2 feature at global and Remote SEPP Set level.
- The user must add the required headers (3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id) in the Cat 2 header configurations.
- If the Cat 2 is disabled or required header configuration is not done, header validation will not be performed.
For more details about the Cat 2 - Network ID Validation feature, refer to [Cat 2 - Network ID Validation feature](#) section.

Header Addition

The 3gpp-Sbi-Originating-Network-Id header or 3gpp-Sbi-Asserted-Plmn-Id header is added to the request according to the user configuration. The header addition occurs on both P-SEPP and C-SEPP when the incoming request does not have any header present.

Header Transposition

If the incoming request at producer SEPP has a header (3gpp-Sbi-Originating-Network-Id header or 3gpp-Sbi-Asserted-Plmn-Id) but the producer SEPP's supported header name is different from this request header, then the original header present in the request is replaced with the header supported by producer SEPP. SEPP supported header name is user configurable.

For example, if the header in the incoming request is 3gpp-Sbi-Asserted-Plmn-Id, but the user has configured the 3gpp-Sbi-Originating-Network-Id header at producer SEPP, then request header 3gpp-Sbi-Asserted-Plmn-Id header is replaced with the 3gpp-Sbi-Originating-Network-Id header.

Note

- After the header name transposition, its value will be the same as the value that was present in the original request header.
- If the request header is 3gpp-Sbi-Originating-Network-Id and its value is in the format MCC-MNC";src:"srcType"-"srcfqdn (Here srcType = "SCP"/"SEPP"), then during transposition to 3gpp-Sbi-Asserted-Plmn-Id header, the value will only have MCC-MNC and remaining part from the header value will be stripped off.

Example:

Header before transposition: 3gpp-Sbi-Originating-Network-Id:123-456;src:SEPP sepp001.sepp.5gc.mnc333.mcc222.3gppnetwork.org

Header after transposition: 3gpp-Sbi-Asserted-Plmn-Id:123-456

Detailed Description

Functioning at C-SEPP

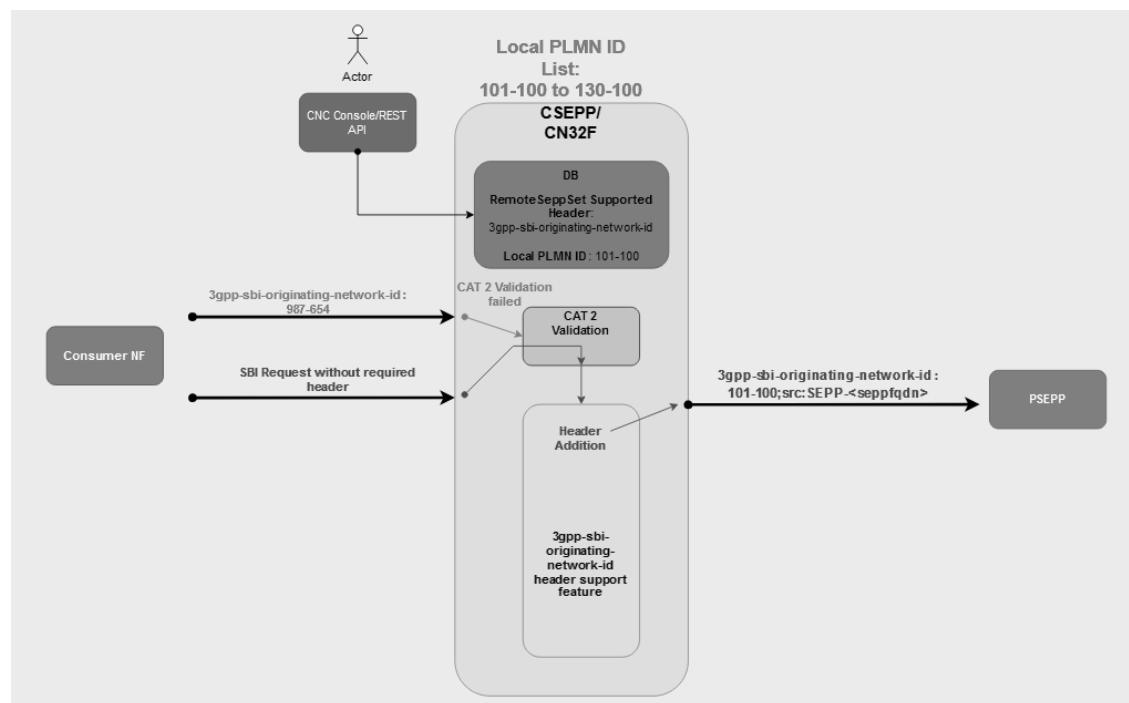
If the 3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id header is present in the incoming request, the header value validation is performed using the Cat 2 - Network ID Validation feature. This functionality extracts the PLMN ID value from the header and matches it against the local SEPP PLMN IDs supported by the C-SEPP. The user has to enable and configure the Cat 2 feature functionalities through CNC Console or REST API to do the validation.

If the 3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id header is missing in the request, it will be added in the request by C-SEPP. The user has to enable and configure the feature through CNC Console or REST API.

C-SEPP adds the header that its Remote SEPP (P-SEPP) supports, so the header name will be taken from user's configured header name at Remote SEPP Set level. Header value will be populated with the user configured local PLMN ID at C-SEPP. If the user has not configured the local PLMN ID, then one of the local PLMN ID's values supported by the C-SEPP will be used to populate the missing header.

The following diagram represents the functionality of the feature on the C-SEPP side.

Figure 3-35 C-SEPP side functionality



Instance 1: Request comes with 3gpp-Sbi-Originating-Network-Id: 987-654

In this example, the user has enabled the Cat 2 feature and done the header configuration for '3gpp-Sbi-Originating-Network-Id' header. Request comes at C-SEPP and as the PLMN ID value (987-654) extracted from the header is not present in the SEPP local PLMN ID List (101-100 to 130-100), request is discarded.

Instance 2: (Sbi request comes without any of the 3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id headers)

In this example, the user has enabled the Cat 2 feature and done the header configuration for '3gpp-Sbi-Originating-Network-Id' and '3gpp-Sbi-Asserted-Plmn-Id' headers. As request is not having any of the headers, Cat 2 feature bypasses the validation here. The user has enabled the feature, configured the '111-100' as local PLMN Id and header name as '3gpp-Sbi-Originating-Network-Id' at Remote SEPP Set level.

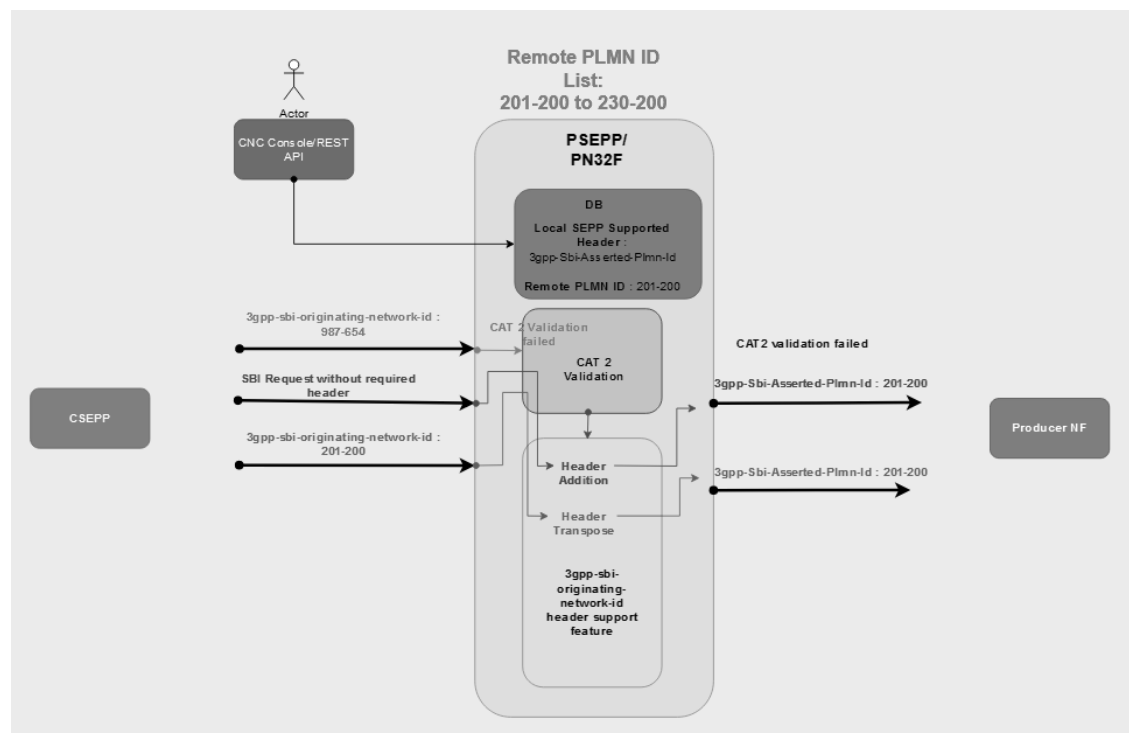
Header '3gpp-Sbi-Originating-Network-Id' with value '111-100;src:SEPP-<seppfqdn>' is added successfully in the request as per this header 3gpp specification.

Note

If the user configures the '3gpp-Sbi-Asserted-Plmn-Id' header here, then '3gpp-Sbi-Asserted-Plmn-Id' header is added with just having the PLMN ID value i.e. '111-100' as per this header 3gpp specifications.

Functioning at P-SEPP

The following diagram represents the functionality of the feature on the P-SEPP side.

Figure 3-36 P-SEPP side functionality

If the 3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id header is present in the incoming request, the header value validation is performed using the Cat 2 - Network ID Validation feature. This functionality extracts the PLMN ID value from the header and matches it against the remote SEPP PLMN IDs supported by the P-SEPP. The user has to enable and configure the Cat 2 feature functionalities through CNC Console or REST API to do the validation.

If the 3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id header is missing in the request, it will be added in the request by P-SEPP. The user has to enable and configure the feature through CNC Console or REST API.

P-SEPP adds the header supported by the producer SEPP network, and the header name will be the SEPP supported header name, configured by the user. Header value will be populated with the user configured remote PLMN ID value per Remote SEPP Set level. If this PLMN ID is not configured, then SEPP takes one of the PLMN ID value from the remote list of supported PLMN IDs per Remote SEPP Set.

If the incoming request at producer SEPP has one header (3gpp-Sbi-Originating-Network-Id header or 3gpp-Sbi-Asserted-Plmn-Id) but the producer SEPP supported header name is different from this request header, then the original header present in the request is replaced with the header supported by producer SEPP. SEPP supported header name is user configurable.

Instance 1: Request comes with '3gpp-Sbi-Originating-Network-Id: 987-654'

In instance 1, the user has enabled the Cat-2 feature and done the header configuration for 3gpp-Sbi-Originating-Network-Id header. The request comes at P-SEPP and as the PLMN ID value (987-654) extracted from the header is not present in the Remote PLMN ID List (201-200 to 230-200), request is discarded.

Instance 2: (Sbi request comes without 3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id)

In this example, the user has enabled the Cat 2 feature and performed the header configuration for 3gpp-Sbi-Originating-Network-Id and 3gpp-Sbi-Asserted-Plmn-Id headers. As the request does not have any of the headers, Cat 2 features bypass the validation here.

Here, user has enabled the feature, configured the '201-200' as remote PLMN ID and set 3gpp-Sbi-Asserted-Plmn-Id as SEPP supported header name.

Header 3gpp-Sbi-Asserted-Plmn-Id with value '201-200' is added successfully in the request as per the format defined in the 3gpp specification of '3gpp-Sbi-Asserted-Plmn-Id' header.

Note

If the user configures the '3gpp-Sbi-Originating-Network-Id' header, then '3gpp-Sbi-Originating-Network-Id' header is added as '201-200;src:SEPP-<seppfqdn>' as per the format defined in the 3gpp specifications of the header.

Instance 3: (Sbi request comes with 3gpp-Sbi-Originating-Network-Id: 201-200)

In this instance, the request comes with 3gpp-Sbi-Originating-Network-Id but the header supported at the P-SEPP is 3gpp-Sbi-Asserted-Plmn-Id. Hence, header transposition occurs and the incoming original header, which is 3gpp-Sbi-Originating-Network-Id: 201-200 is replaced by 3gpp-Sbi-Asserted-Plmn-Id: 201-200.

Note

- The headers 3gpp-Sbi-Originating-Network-Id and 3gpp-Sbi-Asserted-Plmn-Id wont be present in the incoming Sbi request at the same time.
- If the Cat 2 feature is disabled or not configured for the 3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id headers, the no validation will be performed for the PLMN ID in these headers.
- If an incoming Sbi request does not have either of the 3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id headers, then the Cat 2 validation will be bypassed and the header is added by feature as per the user's configurations.
- When the SEPP is deployed in the roaming hub mode, header Addition or transposition is not performed.
- SNPN-ID in 3gpp-Sbi-Originating-Network-Id is presently not supported.

Managing the Support for Originating Network Id Header Validation, Insertion, and Transposition Feature

You can enable the Support for Originating Network Id Header Validation, Insertion, and Transposition feature using the CNC Console, or REST API.

Enable using REST API

Set `originatingNetworkIdSupportEnabled` to `true` in Originating Network Id Header Support Option REST API.

```
{ "originatingNetworkIdHeaderSupportOption": {  
  "originatingNetworkIdSupportEnabled": true,  
  "seppSupportedLocalPlmnId": "444-555",  
  "seppSupportedOrigNetworkIdHeader": "3gpp-Sbi-Asserted-Plmn-Id"  
}
```

For more information about the API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Enable using CNC Console

Perform the following procedure to enable the support for the 3gpp-Sbi-Originating-Network-Id header feature:

1. From the left navigation menu, navigate to **SEPP** and click **Originating Network ID Header Support**. The **Option** appears underneath.
2. Click **Option**, the options screen appears on the right pane with parameters to enable the feature.
3. Click the **Edit** icon to modify the **Option**. The **Edit Option** page appears.
4. Set **Enabled** to True or False.

For more details about CNC Console configurations, see [Configuring SEPP using CNC Console](#).

Configure

You can configure the feature using REST API and CNC Console.

- **Configure using REST API:** Perform the feature configurations as described in 3gpp-Sbi-Originating-Network-Id Header Support Options REST API and Originating Network Id Header Support Option Parameters REST API in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- **Configure using CNC Console:** Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Observe

Following are the Support for Originating Network Id Header Validation, Insertion, and Transposition feature specific metrics:

- ocsepp_originating_network_id_header_added_total
- ocsepp_originating_id_header_transposed_total
- ocsepp_originating_header_addition_failed
- ocsepp_originating_header_add_or_transpose_failed

For more information on Support for Originating Network Id Header Validation, Insertion, and Transposition feature related SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.19 Support for Common Service APIs in CNC Console

The configuration for common service APIs was supported only using REST. With the implementation of this feature, SEPP now supports the configuration of Ingress Gateway and Egress Gateway parameters using the CNC Console. You can perform HTTP methods using the Console.

Managing common service APIs in the CNC Console

Enable

This feature is enabled automatically along with the SEPP instance deployment.

Configure

You can configure the common services APIs in the CNC Console. For more information, see Common Services Configuration section.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

1. Collect the logs: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. Raise a service request: See [My Oracle Support](#) for more information on how to raise a service request.

3.2.20 Support for TLS

Network Functions (NFs) or peers can use Hypertext Transfer Protocol Secure (HTTPS) to establish secured ingress and egress connections with consumer NFs and producer NFs, respectively. These communication protocols are encrypted using Transport Layer Security (TLS).

TLS comprises the following components:

- **Handshake Protocol:** Exchanges the security parameters of a connection. Handshake messages are supplied to the TLS record layer.
- **Record Protocol:** Receives the messages to be transmitted, fragments the data into multiple blocks, secures the records, and then transmits the result. Received data is delivered to higher-level peers.

From Release 24.1.0 onwards, SEPP supports TLS 1.3 for all functions and interfaces that are supported by TLS 1.2. With this feature, SEPP supports creation of TLS 1.3 and TLS 1.2 connections and mandatory ciphers and extensions.

The following table provides comparison of TLS 1.2 with TLS 1.3:

Table 3-28 Comparison of TLS 1.2 with TLS 1.3

Feature	TLS 1.2	TLS 1.3
TLS Handshake	This is less efficient as it requires more round-trips to complete the handshake process.	This is more efficient as it requires less round-trips to complete the handshake process.
Cipher Suites	This has less secured Cipher suites.	This has more secured Cipher suites. They supports the following ciphers: • TLS_CHACHA20_POLY1305_SHA256 • TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_AES_128_CCM_8_SHA256: This Cipher is not supported by Java library. • TLS_AES_128_CCM_SHA256: This Cipher is not supported by Java library.
Round-Trip Time (RTT)	This has higher RTT during TLS handshake.	This has low RTT.
Performance	This has higher latency during TLS handshake.	This has low latency during TLS handshake.

The following digital signature algorithms of TLS 1.2 and TLS 1.3 are supported in TLS handshake:

Table 3-29 Digital Signature Algorithms

Algorithm	Key Size (Bits)	Elliptic Curve (EC)
RS256 (RSA)	2048	NA
	4096 This is the recommended value.	NA

Table 3-29 (Cont.) Digital Signature Algorithms

Algorithm	Key Size (Bits)	Elliptic Curve (EC)
ES256 (ECDSA)	NA	SECP384r1 This is the recommended value.

Note

The following functionalities from TLS 1.3 specifications are not supported:

- Zero round-trip time (0-RTT) mode.
- Pre-Shared Key (PSK) exchange.

Enable

This feature is enabled by default at the time of Gateway Services deployment by completing the configurations as described in the Helm Parameters section.

Configure

You can configure this feature using Helm parameters. The following parameters in the Gateway Services microservices such as n32-egress-gateway and PLMN-egress-gateway, must be customized to support TLS 1.3.

The following table describes the newly introduced and updated parameters for TLS 1.3:

Table 3-30 Helm Parameters

Parameter Name	Description	Mandatory/ Optional/ Conditional	Details
clientDisabledExtension	Disables the extension sent by messages originated by clients (ClientHello).	O	Data Type: String Range: NA Default Value: ec_point_formats
serverDisabledExtension	Disables the extension sent by messages originated by servers (ServerHello).	O	Data Type: String Range: NA Default Value: null
tlsNamedGroups	Provides a list of values sent in the supported_groups extension. These are comma-separated values.	O	Data Type: String Range: NA Default Value: null
clientSignatureSchemes	Provides a list of values sent in the signature_algorithms extension. These are comma-separated values.	O	Data Type: String Range: NA Default Value: null

Table 3-30 (Cont.) Helm Parameters

Parameter Name	Description	Mandatory/ Optional/ Conditional	Details
service.ssl.tlsVersion	Indicates the TLS version.	M	Data Type: String Range: • TLSv1.2, TLSv1.3 • TLSv1.2 • TLSv1.3 Default Value: TLSv1.2, TLSv1.3
allowedCipherSuites	Indicates allowed Ciphers suites.	O	Data Type: String Range: NA Default Values: • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_AES_128_GCM_SHA256 • TLS_CHACHA20_POLY1305_SHA256
cipherSuites	Indicates supported cipher suites.	O	Data Type: String Range: NA Default Values: • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_AES_128_GCM_SHA256 • TLS_CHACHA20_POLY1305_SHA256
messageCopy.security.tlsVersion	Indicates the TLS version for establishing communication between Kafka and NF when security is enabled.	C	Data Type: String Range: • TLSv1.3 • TLSv1.2 Default Value: TLSv1.3

The following sample Helm configuration is required for TLS 1.3:

```
clientDisabledExtension: ec_point_formats #comma-separated-values To disable
extension being sent in ClientHello
serverDisabledExtension: null #comma-separated-values To disable extension
being sent from server originated messages
tlsNamedGroups: null #comma-separated-values to whitelist the
supported_groups extension values
clientSignatureSchemes: null #comma-separated-values to whitelist the
signature_algorithms extension values
service:
  ssl:
    tlsVersion: TLSv1.2,TLSv1.3
allowedCipherSuites:
  - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
  - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
  - TLS_AES_256_GCM_SHA384
  - TLS_AES_128_GCM_SHA256
  - TLS_CHACHA20_POLY1305_SHA256
cipherSuites:
  - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
  - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
  - TLS_AES_256_GCM_SHA384
  - TLS_AES_128_GCM_SHA256
  - TLS_CHACHA20_POLY1305_SHA256
messageCopy:
  security:
    tlsVersion: TLSv1.2
```

Note

- SEPP does not prioritize cipher suites on the basis of priorities. To select cipher on the basis of priorities, you must write the cipher suites in the decreasing order of priority.
- SEPP does not prioritize supported groups on the basis of priorities. To select supported group on the basis of priorities, you must write the supported group values in the decreasing order of priority.
- If you want to provide values for the signature_algorithms extension using the `clientSignatureSchemes` parameter, the following comma-separated values must be provided to deploy the pods:
 - `rsa_pkcs1_sha512`
 - `rsa_pkcs1_sha384`
 - `rsa_pkcs1_sha256`
- The mandatory extensions as listed in RFC 8446 cannot be disabled on the client or server side. The following is the list of the extensions that cannot be disabled:
 - `supported_versions`
 - `key_share`
 - `supported_groups`
 - `signature_algorithms`
 - `pre_shared_key`

Observe

The following metrics are available for this feature:

- `oc_ingressgateway_incoming_tls_connections`
- `oc_egressgateway_outgoing_tls_connections`
- `security_cert_x509_expiration_seconds`

For more information about metrics, see [SEPP Metrics](#) section.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

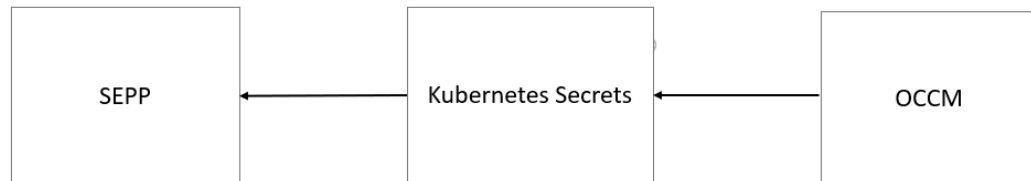
1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.21 Support for Automated Certificate Lifecycle Management

In SEPP 23.4.x and earlier, X.509 and Transport Layer Security (TLS) certificates were managed manually. When multiple instances of NFs were deployed in a 5G network, certificate management, such as certificate creation, renewal, removal, and so on, became tedious and error-prone.

Starting with SEPP 24.1.x, you can integrate with Oracle Communications Cloud Native Core, Certificate Management (OCCM) to support automation of certificate lifecycle management. OCCM manages TLS certificates stored in Kubernetes secrets by integrating with Certificate Authority (CA) using the Certificate Management Protocol Version 2 (CMPv2) protocol in the Kubernetes secret. OCCM obtains and signs TLS certificates within the SEPP namespace. For more information about OCCM, see *Oracle Communications Cloud Native Core, Certificate Management User Guide*.

Figure 3-37 Support for OCCM



The above diagram indicates that OCCM writes the keys to the certificates and SEPP reads these keys to establish a TLS connection with other NFs.

OCCM can automatically manage the following TLS certificates:

- 5G Service Based Architecture (SBA) client TLS certificates
- 5G SBA server TLS certificates

This feature allows to monitor, create, recreate, and renew TLS certificates using OCCM, based on their validity. For information about enabling HTTPS, see "NF Authentication using TLS certificate" in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Install Guide Considerations

- **Upgrade:** When SEPP is deployed with OCCM, follow the specific upgrade procedure. For information about the upgrade strategy, see "Upgrade Strategy" in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.
- **Rollback:** For more information on migrating the secrets from SEPP to OCCM and removal of Kubernetes secrets from the yaml file, see "Postupgrade Task" in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Configure

There are no additional configuration changes required at SEPP.

Maintain

If you encounter any OCCM-specific alerts, see the "OCCM Alerts" section in *Oracle Communications Cloud Native Core, Certificate Management User Guide*.

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.22 Load Sharing among Multiple Remote SEPP Nodes

Introduction

Until the current release, SEPP could only redirect traffic to the secondary remote SEPP when the primary SEPP was unavailable, allowing for the use of only one SEPP at a time.

With load sharing among multiple Remote SEPP nodes feature, SEPP can now efficiently distribute incoming traffic among multiple remote SEPPs. To enable this, the operator needs to configure a single virtual FQDN for the remote SEPPs. SEPP retrieves multiple Remote SEPP records from the DNS server and distributes the traffic according to the priority and weight associated with each record. Hence, SEPP can simultaneously route the incoming requests to multiple Remote SEPPs.

Feature Overview

This feature enables SEPP to perform DNS SRV Query to discover Remote SEPPs. This enhances the capability of DNS SRV which already supports DNS Query capability. This feature eases the operation and maintenance for the customer when the failover nodes are configured in the DNS server for different SEPPs. Previously, users were configuring the FQDNs locally for each SEPP. This configuration also allows the operator to establish a backup/failover pair of a notify consumer or producer within DNS. In case of a notify or service request failure, SEPP can then choose this backup configuration for retry purposes.

The following is the format of SRV records defined at DNS server:

`_Service._Proto.Name TTL Class SRV Priority Weight Port Target`

`_http._tcp.example.com 86400 IN SRV 1 10 80 blr.example.com`

In this example, the virtual FQDN "example.com" returns the single FQDN record "blr.example.com" with priority 1 and weight 10.

Similarly, multiple records can be defined against the single virtual FQDN. Each FQDN returned can be resolved to multiple IP address in the DNS server.

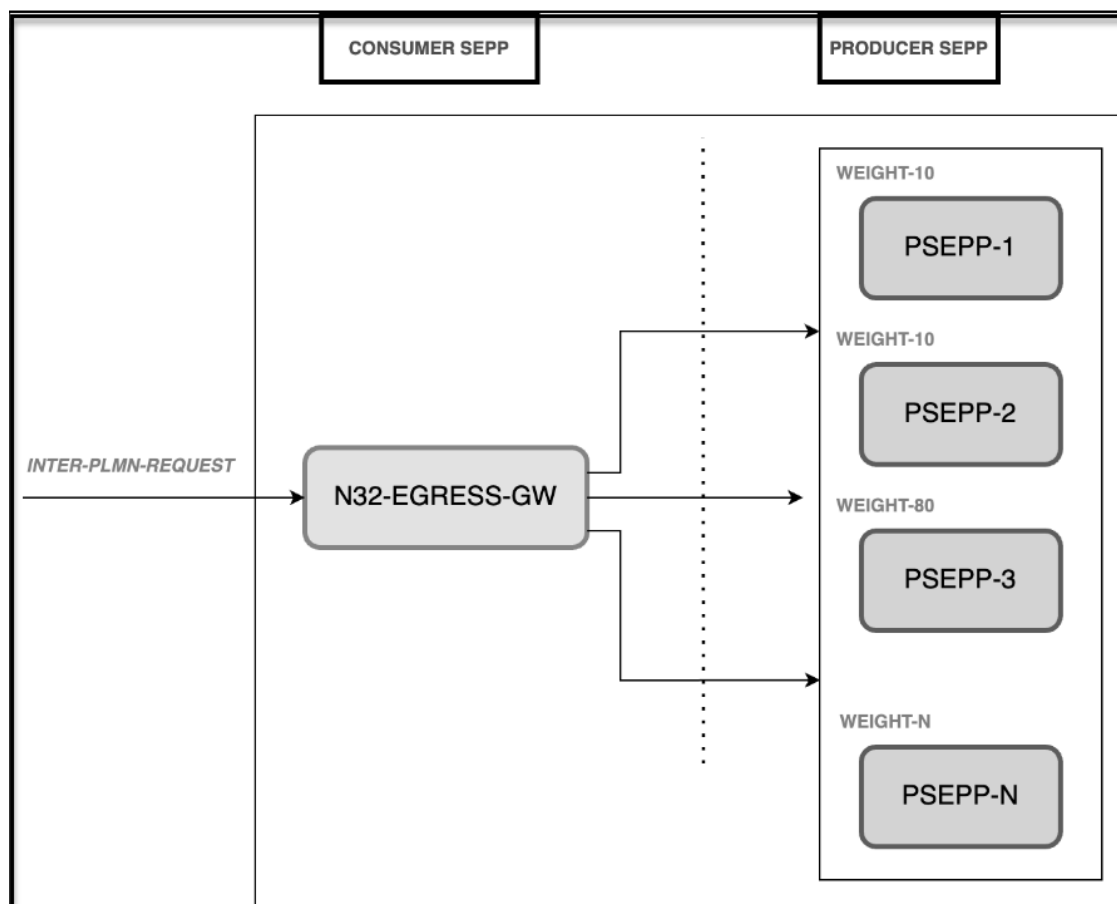
Assumptions

- SEPP FQDN and Virtual FQDN should be mapped to a valid IP address or multiple IP addresses. SEPP FQDN is used for the N32c handshake procedure.
- The load sharing feature is only used for the N32F traffic at n32-egress-gateway.
- Remote SEPP Set cannot have multiple peers with different Virtual FQDNs.
- Only a single Remote SEPP from the multiple Remote SEPPs with the same virtual FQDNs needs to be associated with Remote SEPP Set.
- SEPPs sharing the same Virtual FQDN should belong to the same set of PLMNs.
- Deployment of Alternate Route Service is mandatory for using load sharing feature at both the Egress Gateways.
- Correct DNS configuration is up to the discretion of the user. It is possible that the SEPP may or may not have handshake established within its database.
- Selective routing to Remote SEPPs based on server header is currently not supported.

Detailed Description

The following diagram represents the load sharing among multiple Remote SEPP nodes feature:

Figure 3-38 Architecture



Oracle SEPP supports load sharing of incoming 5G traffic to multiple Remote SEPPs with the help of load sharing feature at Egress Gateway. Single virtual host is configured for all the Remote SEPPs that share the same PLMN ID list. Virtual Host is mapped to the Remote SEPPs' FQDN records, and each having its own priority and weight values. Egress Gateway retrieves the records from the DNS server and arranges them in the order of priority and weight. If the priorities of all the fetched records against the virtual host are same, then the requests are load-shared among the Remote SEPPs on the basis of weight value assigned to each record. If any of the peers are unavailable, then the load gets redistributed to the remaining peers based on their priority and weight factor. This redistribution depends on whether the error code received from the peer is correctly configured in the database.

Managing Load Sharing among Multiple Remote SEPP Nodes Feature

Enable

This section describes the procedure to enable the feature.

You can enable the load sharing among multiple Remote SEPP nodes feature using the CNC Console or REST API by adding the *Virtual Host* parameter at the Remote SEPP Profile.

Note

If the *Virtual Host* parameter is left blank, then the feature is disabled.

Remote SEPP (with virtual host configuration) must be associated with the Remote SEPP Set.

For more details about CNC Console Configurations, see the [Configuring SEPP using CNC Console](#) section.

For more information about API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Configure

You can configure the feature using Helm, REST API, and CNC Console.

Configure Using Helm:

The following are the Helm configurations required to enable and configure the feature:

Configure the following Helm parameters in the `ocsepp_custom_values_<version>.yaml` file.

```
dnsSrvEnabled: true
alternateRouteServiceEnable: true
dnsSrvFqdnSetting.enabled: false
dnsSrvFqdnSetting.pattern: "_{scheme}._tcp.{fqdn}."
```

Note

Scheme must be always be "http".

The following parameter must be enabled at the `nrf-client` section under global parameters:

```
alternateRouteServiceEnable: true
```

The following parameter must also be enabled at `n32-egress-gateway` section under global parameters:

```
checkAltRouteSvcReady: true
```

For more information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

- **Configure using REST API:** Perform the feature configurations as described in the Remote SEPP REST API in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- **Configure using CNC Console:** Perform the feature configurations as described in the Remote SEPP in the [Configuring SEPP using CNC Console](#) section.

Observe

Following are the feature specific metrics:

- `oc_egressgateway_sbiRouting_http_responses_total`

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.23 Separate Port Configurations for N32c and N32f on the Egress Routes

In some deployments, there is a need to have different service IPs and ports for n32c and n32f interfaces although the endpoint SEPP may have the same name. Separate ports for n32c and n32f interfaces provide the flexibility to configure different IP address or port for the n32c and n32f interfaces.

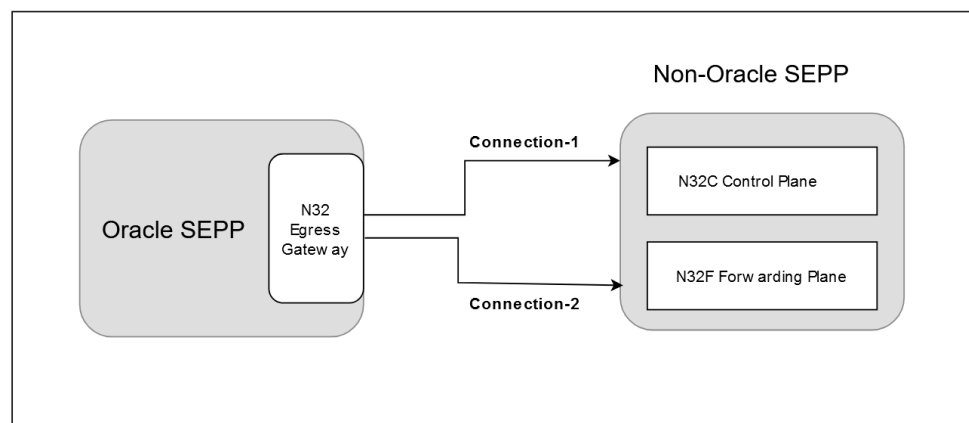
SEPP was previously configured to support a single port or connection for both the control plane interface (n32c) and the forwarding interface (n32f). The Egress Gateway was set up to establish the connection of n32c and n32f on the same port, making it challenging to segregate the traffic. This also prevented SEPP from initiating connections to Remote SEPPs that required a different address for the n32c and n32f interfaces.

To improve traffic segregation, the Egress Gateway is enhanced by configuring different ports for the n32c and n32f connections on both the Remote SEPP Set and its local configurations.

Design and Architecture

The following diagram represents the separate port configurations for n32c and n32f on the Egress routes feature:

Figure 3-39 Separate Port Configurations Design Diagram



As represented in the diagram, the n32 Egress Gateway in the Oracle SEPP is configured to use different IP addresses or ports for connecting on the n32c and n32f interfaces on the same Remote SEPP. The n32f configuration in the Remote SEPP profile initiates separate connections toward the forwarding pane and control pane of the Remote SEPP. The forwarding plane can have the distinct FQDN and port.

If the n32f configuration is not present, then it is assumed that the control plane and forwarding plane have the same FQDN, IP address, and port combination.

Managing Separate Port Configurations for N32c and N32f on the Egress Routes

Enable

This section describes the procedure to enable the feature.

You can enable separate port configurations for n32c and n32f on the egress routes feature using the CNC Console or REST API by adding the n32 configuration parameters at the Remote SEPP profile. The parameters are **N32F FQDN**, **N32F IP Address**, and **N32F Port**.

Note

If the **N32F FQDN**, **N32F IP Address**, and **N32F Port** parameters are left blank, then the feature is disabled.

For more details about CNC Console Configurations, see *Configuring SEPP using CNC Console* section in the *Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide*.

For more information about REST API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Configure

You can configure the feature using REST API and CNC Console.

- **Configure using REST API:** Perform the feature configurations as described in the Remote SEPP REST API in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- **Configure using CNC Console:** Perform the feature configurations as described in the Remote SEPP in the [Configuring SEPP using CNC Console](#) section.

Observe

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and troubleshooting scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.24 Support for cnDBTier APIs in CNC Console

With the implementation of this feature, cnDBTier APIs are integrated into the CNC Console. SEPP users can view specific cnDBTier APIs such as checking the cnDBTier version, status of cnDBTier clusters, and georeplication status on the CNC Console.

The following cnDBTier APIs can be viewed (read only) from the CNC Console:

Table 3-31 Georeplication Recovery

Console Parameter	cnDBTier API name	Description	Supported cnDBTier Version
cnDBTier Backup Status	<code>http://<base-uri>/ocdbtier/backup/status</code>	This is a read-only API. This API displays the current backup status of the cnDBTier. It displays the following: - current system time - current cnDBTier backup status - next scheduled backup time	25.2.100 onwards.
Update Cluster As Failed	<code>http://<base-uri>/ocdbtier/markcluster/failed</code>	This API is used to mark a cnDBTier cluster as failed.	24.2.x onwards
Start Georeplication Recovery	<code>http://<base-uri>/ocdbtier/faultRecovery/start</code>	This API is used to start the georeplication recovery.	24.2.x onwards
Georeplication Recovery Status	<code>http://<base-uri>/ocdbtier/faultrecovery/status</code>	This API is used to monitor the georeplication recovery status of FAILED and ACTIVE cnDBTier sites.	24.2.x onwards

Table 3-32 cnDBTier Health `http://base-uri/{unique_prefix}/ocdbtier/health-info`

Console Parameter	cnDBTier API name	Description	Supported cnDBTier Version
Backup Manager Health Status	<code>http://base-uri/{unique_prefix}/ocdbtier/health-info/backup-mgr-svc/status</code>	This is a read-only API. This API displays the health status of the backup manager service. It checks the following: - if the backup manager service is up or not - if the service can connect to database or not	24.1.x onwards

Table 3-32 (Cont.) cnDBTier Health `http://base-uri/{unique_prefix}/ocdbtier/health-info`

Console Parameter	cnDBTier API name	Description	Supported cnDBTier Version
Monitor Health Status	<code>http://base-uri/{unique_prefix}/ocdbtier/health-info/monitor-svc/status</code>	This is a read-only API. This API displays the health status of the monitor service. It checks the following: • if the monitor service is up or not • if the service can connect to database or not • if the metrics are fetched or not (the metrics are fetched when the service is up and vice versa)	24.1.x onwards
NDB Health Status	<code>http://base-uri/{unique_prefix}/ocdbtier/health-info/ndb-svc/status</code>	This is a read-only API. This API displays the health status of the network database. It checks the following: • if the pod is connected to PVC or not • if the pods status is up or not Note:PVC is not applicable to some of the pods.	24.1.x onwards
Replication Health Status	<code>http://base-uri/{unique_prefix}/ocdbtier/health-info/replication-svc/status</code>	This is a read-only API. This API displays the health status of the replication service. It checks the following: • if the replication service is up or not • if the service can connect to database or not	24.1.x onwards
cnDBTier Version	<code>http://<base-uri>/ocdbtier/version</code>	This is a read-only API. This API displays the cnDBTier version.	23.4.x onwards
Backup List	<code>http://<base-uri>/ocdbtier/list/backups</code>	This is a read-only API. This API displays the details of completed backups along with backup ID, backup creation timestamp, and backup size.	23.4.x onwards
Database Statistics Report	<code>http://<base-uri>/ocdbtier/statistics/report/dbinfo</code>	This is a read-only API. This API displays the number of available databases.	23.4.x onwards

Table 3-33 Geo Replication Status

Console Parameter	cnDBTier API name	Description	Supported cnDBTier Version
Real Time Overall Replication Status	http://<base-uri>/ocdbtier/status/replication/realtime	This is a read-only API. This API displays the overall replication status in multisite deployments. For example, in a four-site deployment, it provides the replication status between the following sites: site1-site2, site1-site3, site1-site4. This is applicable for all other sites, such as, site 2, site 3, and site 4.	23.4.x onwards
Site Specific Real Time Replication Status	http://<base-uri>/ocdbtier/status/replication/realtime/{remoteSiteName}	This is a read-only API. This API displays the site-specific replication status.	23.4.x onwards
HeartBeat Status	http://<base-uri>/ocdbtier/heartbeat/status	This is a read-only API. This API displays the connectivity status between the local site and the remote site name to which SEPP is connected.	23.4.x onwards
Local Cluster Status	http://<base-uri>/db-tier/status/cluster/local/realtime	This is a read-only API. This API displays the status of the local cluster.	23.4.x onwards
On Demand Backup	http://<base-uri>/ocdbtier/on-demand/backup/initiate	This API can be edited to create a new backup. This API displays the status of initiated on-demand backups and helps to create a new backup.	23.4.x onwards
Georeplication Status Across All Sites	http://<base-uri>/ocdbtier/replication/status/realtime	This is a read only API. This API is used to retrieve the georeplication status across all the configured sites, offering a consolidated view of the overall replication status.	25.2.2xx onwards

Table 3-34 Georeplication Recovery

Console Parameter	cnDBTier API name	Description	Supported cnDBTier Version
Update Cluster As Failed	http://<base-uri>/ocdbtier/markcluster/failed	This API is used to mark a cnDBTier cluster as failed.	24.2.x onwards

Table 3-34 (Cont.) Georeplication Recovery

Console Parameter	cnDBTier API name	Description	Supported cnDBTier Version
Start Georeplication Recovery	<code>http://<base-uri>/ocdbtier/faultRecovery/start</code>	This API is used to start the georeplication recovery.	24.2.x onwards
Georeplication Recovery Status	<code>http://<base-uri>/ocdbtier/faultrecovery/status</code>	This API is used to monitor the georeplication recovery status of FAILED and ACTIVE cnDBTier sites.	24.2.x onwards

Table 3-35 Gracefully Stop And Start Georeplication APIs and IP Preference For Replication Channel API

Console Parameter	cnDBTier API name	Description	Supported cnDBTier Version
IP Family Preference for Replication Channel	<code>http://base-uri/ocdbtier/preferred/ip-family/preferenceshttp://<base-uri>/ocdbtier/preferred/ip-family/preferences/{siteNameAndRemoteSiteName}</code>	This API is used to configure or update the preferred IP family (IPv4 or IPv6) used for communication between the replication channels.	26.1.2xx onwards
Gracefully Start Georeplication	<code>http://base-uri /ocdbtier/georeplication/start</code>	This API is used to initiate the georeplication between a primary site and one or more remote sites.	26.1.2xx onwards
Gracefully Stop Georeplication	<code>http://base-uri /ocdbtier/georeplication/stop</code>	This API is used to stop georeplication between a primary site and specified remote sites.	26.1.2xx onwards

Note

This feature works when cnDBTier is configured as an instance during the CNC Console deployment. For more information about integrating cnDBTier APIs in CNC Console, see *Oracle Communications Cloud Native Core, cnDBTier User Guide*.

You can view the cnDBTier GUI in the CNC Console. For more information, see cnDBTier in the CNC Console.

Managing cnDBTier APIs in CNC Console

Enable

This feature is available by default, when cnDBTier is configured as an instance during the CNC Console deployment. For more information about integrating cnDBTier APIs in CNC Console, see the "NF Single Cluster Configuration With cnDBTier Menu Enabled" section in the *Oracle Communications, Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*.

There is no option to disable this feature.

Configure

You can view the cnDBTier APIs in the CNC Console. For more information, see cnDBTier APIs in the CNC Console.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.25 5G SBI Message Mediation Support

In the earlier releases, SEPP supports Mediation as an independent service to modify 5G Service Based Interface (SBI) message content, which includes HTTP2 header values and JSON message body, based on the user-defined mediation rule sets.

The message mediation feature at SEPP allows MNOs to resolve the inter-op issues between PLMNs by manipulating the inter PLMN messages. SEPP uses the mediation service to support message mediation. The MNO relies on mediation service capabilities to provide the mediation rules.

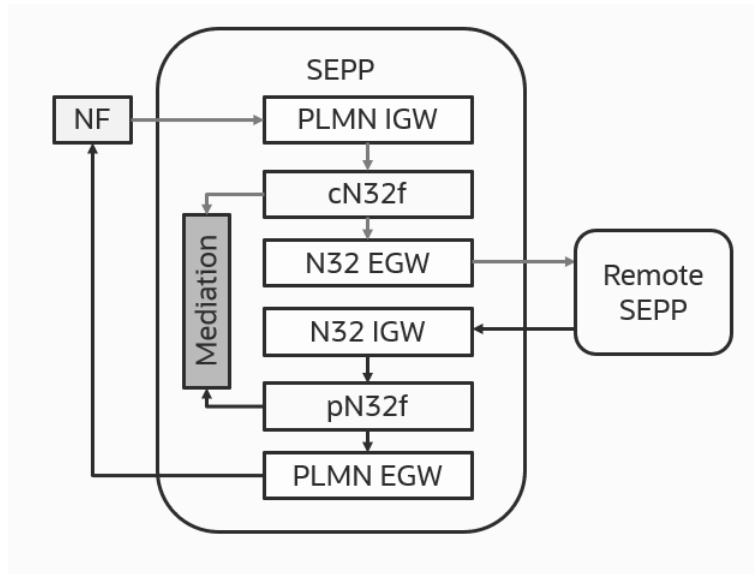
SEPP is enhanced to include mediation as a microservice, that applies user-configured message mediation rules to ingress 5G SBI messages to perform message mediation transformation. Mediation is a common microservice reused across Oracle NFs. The mediation feature allows user-defined business rules to filter or manipulate headers, Query Parameters and Body Parameters in SBI request and response messages. Mediation is invoked by CN32f and PN32f microservices when the request and response matches the configured trigger rules. Trigger Rule Configuration is managed by SEPP. Mediation Rule Configuration used by Mediation microservice is managed by Mediation microservice.

Note

Multipart messages and compressed messages are presently not supported by SEPP mediation.

Mediation Representation with SEPP

Figure 3-40 Mediation Representation with SEPP



Managing Mediation

Enable

- To deploy this service, set the `mediationService` parameter to `true` in the global parameters section of the `custom value` file.

Example:

```

nf-mediation:
  # mediation Global configuration
  global:
    mediationService: true

```

For more information about this parameter, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

You can enable the feature using REST API and the CNC Console.

To enable the feature, set the `Enable Mediation` parameter to `true` in the **Options** screen under the **Mediation** section of the CNC Console GUI.

Following Configuration are required at SEPP to apply 5G SBI Message Mediation Support feature:

- Trigger Rule Configuration
- Mediation Rule Configuration

Trigger Rule Configuration

Mediation Trigger Rule List for SEPP

SEPP allows user to configure Trigger Rule List as a filtering criteria, based on which the mediation functionality is invoked. 5G SBI Message Mediation Support functionality can be applied on a Local SEPP and on a Remote SEPP Set, hence Trigger Rules can be configured for Local and Remote SEPP Set. That is, the user can apply mediation rule on a request going out or a response coming in for this particular local SEPP or the user can apply mediation rule on a request going out or a response coming in for this particular Remote SEPP Set. Every Trigger Rule List must contain a unique name.

SEPP also extends its functionality for the user to apply 5G SBI Message Mediation Support functionality on every request and response that is received on SEPP by selecting Match All option on the console.

Filtering Criteria for Message Mediation functionality to be invoked

SEPP allows user to configure Multiple Trigger Rules in Trigger Rule List which then includes *Trigger Points*, *Resource URI*, *HTTP Method*, and *Group ID*. In this *Trigger Points*, *Resource URI*, and *HTTP Method* are used as a filtering Condition by SEPP to select if a particular request needs Message Mediation functionality invocation or not. Here User is allowed to configure unique combination of *Trigger points*, *Resource URI*, and *HTTP Method* for every Trigger Rule in a Trigger Rule list.

Mediation Trigger Points for SEPP

SEPP supports the following mediation trigger points to invoke mediation service:

- Ingress inter-PLMN and core network messages
- Egress inter-PLMN messages

Hence, Trigger Point can be any of the following:

- N32 EGRESS REQUEST
- N32 INGRESS RESPONSE
- N32 INGRESS REQUEST
- N32 EGRESS RESPONSE

The Trigger Points are available only in the context of the N32 interface/traffic.

Resource URI

There are certain SBI messages which traverse through networks. On the basis of the default list of all the Inter PLMN SBI Messages, SEPP Message Mediation feature allows the user to configure only the allowed Inter PLMN SBI Messages. User is allowed to add any new Resource URI as per their requirement. To add any new Resource URI and HTTP Method, user can add the same in SEPP [Service APIs](#) section in CNC Console. Once an API is added, the same can be reflected under Resource URI of Trigger Rule.

HTTP Method

SEPP allows the user to configure the following predefined HTTP Methods for Message Mediation feature:

- POST
- PUT
- GET
- PATCH
- DELETE
- OPTIONS

Group ID

SEPP uses *Group ID* for which mediation configuration is to be done. This is passed to the Mediation Service for grouping similar rules. It is of type string.

Agenda Group Naming

User can combine mediation rules by using Agenda Group in the mediation rule definition.

Agenda group is the combination of Group id and Trigger Point (configured in Trigger Rule in [Configuring SEPP using CNC Console](#)).

Agenda group name in mediation rules should be <Group Id>-<triggerpoint> as shown in the following example:

agenda-group "AUSF-N32_Ingress_Request"

```
function String modifySupiOrSuci(String str) {
    String supiOrSuci = str.substring(0,15)+"0000"+str.substring(16);
    return supiOrSuci;
}

rule "Match SUPI_SUCI Modify"
    agenda-group "AUSF-N32_Ingress_Request"
    when
        req : Request(body.has("$.supiOrSuci"))
    then

        req.body.put("$", "supiOrSuci", modifySupiOrSuci(req.body.get("$.supiOrSuci")).toString())
    end
```

User can refer the [Mediation Service Rule Book in the Appendix](#) to configure the Mediation Rules.

Examples of Use Cases supported by SEPP

- Incoming SUCI (Subscription Concealed Identifier) format modification and then using modified SUPI (Subscription Permanent Identifier) for selection:
 - Example: "supiOrSuci": "suci-0-310-260-0-0-0-173416067" is first modified to "supiOrSuci": "suci-0-310-260-0000-0-0-173416067"
 - Example: "imei-9844312345123456" is modified to "imei-984431234512345"
- Validating SUCI for format and sending Custom error code
 - Example: "supiOrSuci": "suci-0-310-260-0-0-0-173416067" is checked and instead of 404 Not found, error sent to AMF is modified to 403 "invalid SUCI format".
- When ingress request doesn't come in a supported scheme
 - Example: Visited Network works only with http, but home network requires https.

Only following combination are allowed to be configured in 5G SBI Message Mediation Support feature for local and Remote SEPP with **Match All** Configuration.

Table 3-36 Match All Configurations

Match ALL	Local/ Remote	Description
True	Local	This combination is used for applying mediation rules for all the inter PLMN SBI requests or responses for all the peer SEPPs.
True	Remote	This combination is used for applying mediation rules for all the inter PLMN SBI requests or responses for a particular peer SEPP for which trigger rule list is configured.
False	Local	This combination is used for applying mediation rules on selected (using Trigger points, Resource URI, and HTTP Method) inter PLMN SBI requests or responses for all the peer SEPPs.
False	Remote	This combination is used for applying mediation rules on selected (using Trigger points, Resource URI, and HTTP Method) inter PLMN SBI requests or responses for a particular peer SEPP for which trigger rule list is configured.

Configuring Trigger Rule List in Remote SEPP Set

Once a Trigger Rule list is created with Trigger Rules, user is allowed to associate a Trigger Rule List with the Configured Remote SEPP Sets.

User can select the Remote SEPP Set screen on CNC Console and can select Trigger Rule List under Trigger Rule List Name field.

Note

The user can associate a Trigger Rule list created exclusively for the Remote SEPP Set but is not permitted to associate a Trigger Rule list created for a Local SEPP with the Remote SEPP Set.

Error Configuration

Error action (Reject or Continue) with status code and title is configurable and used for error response. When an error response is received from Mediation microservice the configured error action by user is applied on message.

User is allowed to configure Error action with status code and title to be used with Errors. When an error response is received from the Mediation microservice, the configured error action is applied on message.

Error Action can be:

- Continue: SEPP persists in processing the message, even if an error response is received from the Mediation microservice.
- Reject: SEPP will decline the message and provide a user-configured status code and title.

Mediation Rule Configuration

You can configure the Mediation rules either using config map, or using REST API and CNC Console.

Configuring Mediation Rules using ConfigMap

The following are the steps to configure the rules:

1. Download the configMap into a file using the following command:

```
kubectl get configmap ocsepp-release-nf-mediation-config-active -n sepp-med3 -oyaml > rule_configmap.txt
```

2. At the end of the file **rule_configmap.txt** , append the below **data** block along with needed rules:

```
data:
  rule.drl: |
    package com.oracle.cgbu.ocmediation.nfmediation;

    import com.oracle.cgbu.ocmediation.nfruleengine.NFDroolsRuleEngine;
    import com.oracle.cgbu.ocmediation.factdetails.Request;
    import com.oracle.cgbu.ocmediation.factdetails.Response;
    import java.util.Map;
    import java.util.HashMap;

    dialect "mvel"

    //rules
```

Example of updated file:

```
apiVersion: v1
kind: ConfigMap
metadata:
  annotations:
    meta.helm.sh/release-name: ocsepp-release
    meta.helm.sh/release-namespace: sepp-med3
  creationTimestamp: "2022-08-21T09:37:03Z"
  labels:
    app.kubernetes.io/managed-by: Helm
  name: ocsepp-release-nf-mediation-config-active
  namespace: sepp-med3
  resourceVersion: "15059111"
  uid: 10baf42f-3582-49a7-8cd1-7228357c9a54
data:
  rule.drl: |
    package com.oracle.cgbu.ocmediation.nfmediation;

    import com.oracle.cgbu.ocmediation.nfruleengine.NFDroolsRuleEngine;
    import com.oracle.cgbu.ocmediation.factdetails.Request;
    import com.oracle.cgbu.ocmediation.factdetails.Response;
    import java.util.Map;
    import java.util.HashMap;

    dialect "mvel"

    rule "Rule_show_add_header"
    when
      req : Request(headers.has("3gpp-Sbi-Message-Priority")) == true)
    then
```

```

    req.headers.add("3gpp-Sbi-Message-Priority","10")
    req.headers.del("x-test-req-header1")
end

rule "Rule_Request_body_add_delete"
when
    req : Request(headers.has("x-forwarded-NF","NRF"))
then
    req.body.put("$","nfId","ccc5ccbb-5bb9-465f-9ace-0faf08cb4223")
    req.body.del("$.supiOrSuci")
end

```

3. Run the following command to update the configmap with the updated rules:

```
kubectl replace ocsepp-release-nf-mediation-config-active -n sepp-med3 -f
rule_configmap.txt
```

Note

- If rules changed on nf active mediation then use **"\$releaseName-nf-mediation-config-active"** as the name of the configmap.
- It is recommended to maintain the back up of the rules as it can lead to rule loss if the config map deleted.

Mediation Rules using CNC Console and Rest APIs

The above process of creating mediation rules in mediation-nf using a configmap with Kubernetes has several limitations, such as configuring the code to create and update the mediation rules using a configmap, whose size is limited to 1 MB.

With this feature enhancement in release 23.4.0, mediation rules are being stored in the database, these rules can be retrieved, modified, deleted, new rules can be added and applied to mediation microservice using the CNC Console or REST API, leveraging improved database storage capacity.

The following are the user configurable Mediation Rules Parameters:

- Rule Name: Unique mediation rule name for identification.
- Status: APPLIED or DRAFT. New mediation rule(s) are always created and stored with DRAFT status into the database. Once the rules are saved again using Apply state, they will be applied to the mediation microservice and the status will be APPLIED.
- State: Possible rule state values are Save, Compile, Apply, Clone, and Draft. New rule(s) are always created using Save state. Once created, user can select the other states.
- Mediation Mode: Possible values of mode are MEIDATION_ACTIVE and MEIDATION_TEST. The user is required to configure the mediation rules using MEDIATION_ACTIVE mediation mode. MEDIATION_TEST mode is only for internal purpose.
 - MEIDATION_ACTIVE: It is applicable only to mediation microservice active mode.
 - MEDIATION_TEST: It is applicable only to mediation microservice test mode.

- **Code:** Mediation rule code. The user needs to perpend the following data block along with the needed rules. User will be restricted to apply a rule that is having any compilation or syntax error in it. Any special character in code results in a rule compilation error.

```
package com.oracle.cgbu.ocmediation.nfmediation;

import com.oracle.cgbu.ocmediation.nfruleengine.NFDroolsRuleEngine;
import com.oracle.cgbu.ocmediation.factdetails.Request;
import com.oracle.cgbu.ocmediation.factdetails.Response;
import java.util.Map;
import java.util.HashMap;

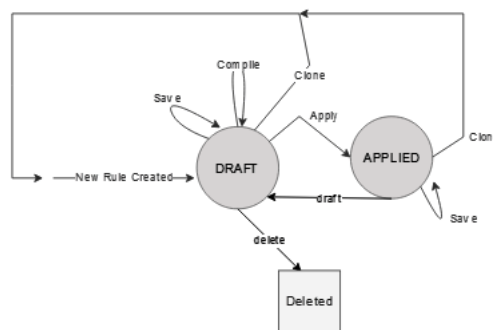
dialect "mvel"

//rules
```

- **Format:** Rule format. Only DRL is supported currently.
- **New Rule Name:** New mediation rule name is to be given only when an existing rule is cloned using Clone state.

The following diagram represents the rule state and status flow:

Figure 3-41 Rule Status Flow



Mediation Rules States for DRAFT status rules

- **Save:** It saves an existing rule into the database.
- **Compile:** It only compiles the rule and saves it again into the data base.
- **Clone:** It works same as "save as". That is, saves an existing rule with a new given name into the database.
- **Apply:** It compiles the rule, saves it again in APPLIED status into the database, and applies the rule to mediation microservice.
- **Draft:** Its not supported as rule status is already DRAFT.

Mediation Rule States for APPLIED status rules

- **Save:** It applies the rule with changed mediation mode to the mediation microservice. While using this state, user can only change the rule mode, that is, `MEDIATION_ACTIVE` to `MEDIATION_TEST` and vice versa. Modification in rules code is not allowed.

- **Compile:** Its not allowed as rule had been already compiled while applying it earlier.
- **Clone:** It works same as "save as", that is, saves an existing rule with a new given name into the database in DRAFT status.
- **Apply:** Its not allowed as rule is already in APPLIED status.
- **Draft:** It removes the APPLIED rule from mediation microservice, and saves again the rule in database in DRAFT status. While using this state, any change in the rule code is not allowed. Any code changes can be done once the rule status has been successfully changed to DRAFT status.

① Note

1. In SEPP release 23.4.0, the user is recommended to configure the mediation rules using CNC Console or REST APIs. However, configuring the mediation rules using config map is still supported and might be removed in the future releases.
2. The `mediationConfig.ruleApi.enabled` helm flag is used to enable or disable the CNC Console or REST API based rules configurations feature implementation. If the flag value is true, mediation microservice uses the rules from the database configured using CNC Console or REST APIs. If the flag value is false, mediation microservice uses the rules configured in the config map. The default value of the flag is true.
3. If an upgrade is performed from earlier SEPP releases to 23.4.0 release, the earlier releases' mediation rules that are available in the config map will be copied into the database table in **APPLIED** state and that can be viewed, modified, or deleted using CNC console or REST APIs in 23.4.0 release.
4. After an upgrade, if a rollback is performed from SEPP release 23.4.0 to earlier releases, the config map will have the same rules that were present before the upgrade.

For more information on creating mediation rules using CNC Console or REST API, see the [Mediation Rules Configuration](#) or *Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Custom Headers

Custom headers can be used if the user want to apply any mediation ruels on the service request message headers. The following are the custom headers supported by the mediation service.

Table 3-37 Custom Headers

CustomHeader	Original Message Header
x-original-method	":method"
x-original-scheme	":scheme"
x-original-authority	":authority"
x-original-path	URI of original message
x-original-status	":status"
x-message-type	value can be "request" or "response" Default value : "request"

Note

In case of invalid http status code configured in x-original-status header in Mediation rule, the original status code will be returned from SEPP response.

Mediation Header

In cn32f and pn32f section, user can set mediation Request timeout time. N32f service wait for mediation svc response for mediationRequestTimeout time before sending the error message, if mediation service is unreachable.

In mediation rules, user can reject any incoming request. To reject request, user has to add some headers in mediation response which helps n32f services in making the decision. Two header Parameters are user configurable for this purpose.

- **mediationRequestRejectStatusCodeHeaderName:** The name of the header added in response to depict that n32f service has to reject this request and the value of this header indicates the error code to be returned.
- **mediationRequestRejectReasonHeaderName:** The name of the header added in response to depict that n32f service has to reject this request and the value of this header indicates the error reason to be returned.

Example:

```
mediation:
  mediationRequestTimeout: 900
  header:
    mediationRequestRejectStatusCodeHeaderName: "ocsepp-reject-status"
    mediationRequestRejectReasonHeaderName: "ocsepp-reject-reason"
```

Note

SEPP mediation service does not support multipart messages.

Observe

For more information the Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.26 Topology Hiding

Oracle Communications Security Edge Protection Proxy (SEPP) is a proxy network function (NF) which is used for secured communication between inter-Public Land Mobile Network

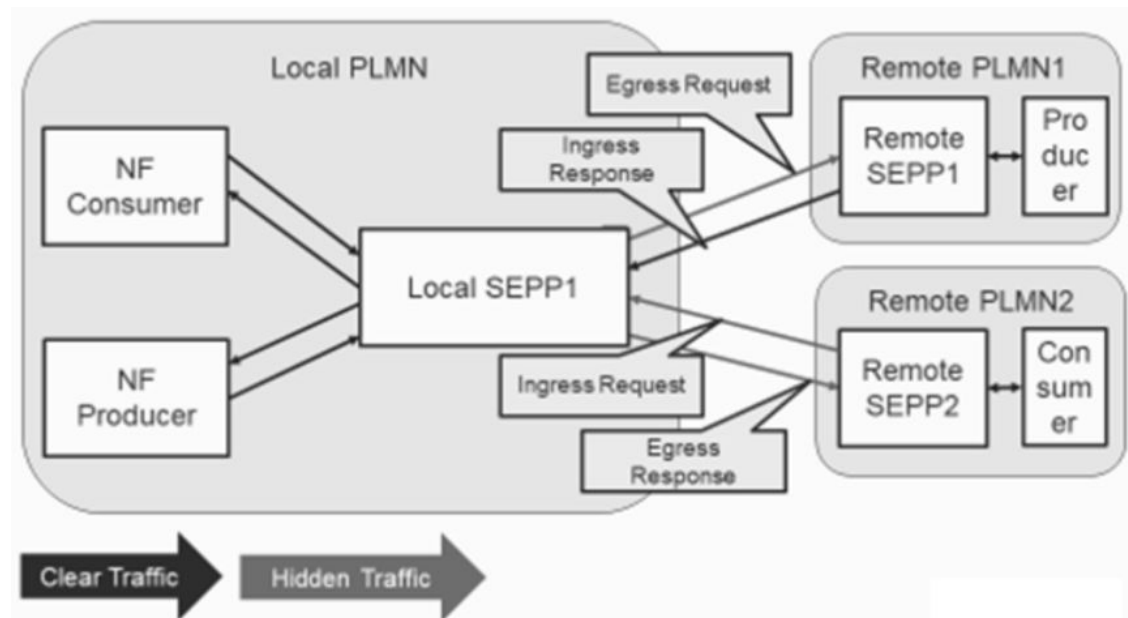
(PLMN) messages. SEPP provides message filtering and policing on inter-PLMN control plane interfaces and topology hiding.

Topology hiding is a security feature in 5G that secures the address of the network elements and can prevent the attacks intended for unauthorized access to network element or interruption of the network service. The purpose of the feature is to enable topology hiding of information from the home or visited PLMN to the visited or Home PLMN. Topology Hiding conceals identity information from all messages leaving a PLMN.

Design and Architecture

The following diagram represents the detailed architecture of the Topology Hiding feature.

Figure 3-42 Topology Hiding Architecture



If the Topology Hiding feature is enabled, based on the header, body, and path table configuration named as header, topology_body, and topology_path respectively, either the message will be masked with the hidden value (if the operation is Topology Hiding (TH)) or the message will be represented by the actual value (if the operation is Topology Recovery (TUH)), where the hidden value is mapped to the actual value.

Hidden traffic: The traffic is marked as hidden traffic, if the message (header,body, and path) is masked with the hidden value (operation is TH) or the message is represented by the actual value (operation is TUH).

Clear traffic: The traffic is marked as clear traffic, if the request or response does not have any of the header, body, and path configurations mentioned in the header, body, or path configuration table.

Note

Topology Hiding is applicable only to N32f and not to N32c.

Note

Only FQDN, NF_INSTANCE_ID, and NF_SERVICE_INSTANCE_ID format values are supported and they can be updated or modified for hiding or recovery in the message header, body, and path identifiers.

Actual to Pseudo Mapping

User can perform actual to pseudo mapping through CNC Console GUI. While configuring the pseud values, user has to add the specific unique prefix and suffix to make pseudo names globally unique. This is necessary to make topology hiding functionality efficient.

Figure 3-43 Actual to Pseudo Mapping

Actual Value	Pseudo Values	Value Type	Actions
udm1.5gc.mmc340.mcc313.3gppnetwork.org	cyc010-cnfrk.udm-001.5gc.mmc340.mcc313.3gpp...	FQDN	[Delete] [Edit]

The actual values and pseudo values are mapped and stored in the actual_values and pseudo_values tables.

The Helm parameter, maxPseudoValue is by default set to 7. This parameter defines the maximum number of pseudo values that can be associated with a single actual value in the system.

Pseudo Value Mapping Rules

The following are the pseudo value mapping rules:

- For each actual value, the user can configure a minimum of 1 and a maximum of 7 pseudo values.
- If an actual value needs to be hidden, one of the configured pseudo values is selected randomly for hiding or recovery.
- The mapping between actual and pseudo values is used during the TH (Topology Hiding) and TUH (Topology Recovery) operations.
- These operations rely on the configured actual-to-pseudo mappings to determine the appropriate substitution.

Constraints

The following are the constraints:

- A pseudo value must be unique across all actual values, that is, the same pseudo value cannot be reused for multiple actual values.
- An actual value cannot be used for any pseudo value.
- Duplicate pseudo values for the same actual value are also not allowed.

Supported Actual Value Types

The following actual value types are currently supported for mapping:

- FQDN (Fully Qualified Domain Name)
- NF_INSTANCE_ID (Network Function Instance ID)
- NF_SERVICE_INSTANCE_ID (Network Function Service Instance ID)

Header, Body, and Path Configurations

Header, body, and Path configuration tables contain the information that is used for Topology hiding and recovery. The four types of configured traffic are:

- Egress Request
- Egress Response
- Ingress Request
- Ingress Response

Ingress response has a clear traffic, as any messages for TH or TUH are not defined in the ingress response.

Note

In Topology Hiding feature, each request and response are considered as separate operations. Each header, body identifiers, and path identifiers cannot be related to each other in any of the operations. Both request (Ingress/Egress) and response (Ingress/Egress) should be treated as separate isolated transactions.

The following is the default header configuration that is available to the user:

Table 3-38 Topology Hiding Header Configurations

Header Name	Regular Expression	Trigger Point	Operation
via	(?<=SCP-).*	Request Egress	Topology Hiding
via	(?<=SCP-).*	Response Egress	Topology Hiding
location	(?<=http:// https://)[^/?]+	Response Egress	Topology Hiding
server	(?<=SCP- UDM-)([^\s]*)	Response Egress	Topology Hiding
3gpp-sbi-target-apiroot	(?<=http:// https://)[^/?]+	Request Ingress	Topology Recovery
3gpp-sbi-routing-binding	(?<=nfinst= nfservinst= backupamfinst= backuppnf=)([^\s;]+)	Request Ingress	Topology Recovery
3gpp-sbi-binding	(?<=nfinst= nfservinst= backupamfinst= backuppnf=)([^\s;]+)	Request Egress	Topology Hiding
3gpp-sbi-binding	(?<=nfinst= nfservinst= backupamfinst= backuppnf=)([^\s;]+)	Response Egress	Topology Hiding

Note

The user must configure the actual to pseudo mappings for the all the headers mentioned in the above table. The above mentioned are the default header configuration that is added by default and user must configure the actual to pseudo mappings for the all the headers mentioned in the above table.

Messages passing through the system check all the above configured conditions. The direction and the message type for the header configured in the table are validated and perform the TH or TUH based on the operation configured in the table.

When the incoming message or outgoing message has the header that is eligible for performing TH/TUH, the actual or pseudo value of that specific header is extracted and compared it with the actual to pseudo mapping. Once the mapping operation to be performed is identified, if it is TH, the actual value is masked from the pseudo mapping and if it is TUH, recover the pseudo value with the actual value received from mapping.

Query Parameters

When making a request, the request URL may include query parameters. These parameters can be of types such as FQDN, NF_INSTANCE_ID, or NF_SERVICE_INSTANCE_ID, and should be eligible for TH/ TUH operations. To support TH/TUH operations on query parameters, the system allows users to configure query parameters in header table only. Query parameters are part of the URL, the header mechanism treats them as headers internally, enabling consistent handling across different parts of the message. Users can define a query parameter entry in the header table with the following configuration:

- Header Name: Logical name used to refer the query parameter.
- Direction: Specifies the message flow (Example: Ingress or Egress).
- Regular Expression: To extract the value from the URL.
- MessageType: Type of message.
- Operations: TH, TUH

Note

If the key is added as header in the header table, then every request will be processed for these keys as header (if available in request). Configuration done at header table configuration is applicable for all request and response.

Note

User must do the actual to pseudo mapping, else hiding and recovery operations fails.

Example:

```
curl -v --http2-prior-knowledge 'http://127.0.0.1:9090/nnrf-disc/v1/nf-  
instances?service-names=nausf-auth&target-nf-type=AUSF&requester-nf-  
type=NRF&target-plmn-
```

```
list=%5B%7B%22mcc%22%3A%22310%22%2C%20%22mnc%22%3A%22300%22%7D%5D&requester-
plmn-
list=%5B%7B%22mcc%22%3A%22310%22%2C%22mnc%22%3A%22310%22%7D%2C%7B%22mcc%22%3A%
22311%22%2C%22mnc%22%3A%22490%22%7D%5D&routing-indicator=0000&requester-nf-
instance-fqdn=https://pseudo.mnc330.mcc310.BE4NRF06.rs.nokia.com:9090' -H
'Content-Type: application/json' -H 'x-forwarded-
host:ausf.default.mnc444.mcc444.3pnetwork.org' -H 'x-forwarded-port:8080' -H
'3gpp-sbi-target-apiroot: http://ausf.default.mnc444.mcc444.3pnetwork.org'
```

In the above request if the user wants to hide or recover the *requester-nf-instance-fqdn*, then this parameter can be added as a header in the header table configuration and based on that hiding or recovery performed.

For the above example, the Header table will be as follows:

Table 3-39 Example of Header Table

Header Name	Regular Expression	Trigger Point	Operation
requester-nf-instance-fqdn	((?<=http:// https://)[^:/?]+)	Request Egress	Topology Hiding

Topology Hiding Body Configurations

When the request or response message is passing through the system then the request body or response body is processed and its identifiers are hidden or unhidden based on the rules defined in the topology body table.

Every message passing through the system is having a unique apiUrl. Based on that rules for TH/TUH are defined. For every apiUrl, there are a list of identifiers that can be part of request or response based on that user can define the identifiers eligible for TH/TUH. User can configure like what operation needs to be performed when that body is processed.

Table 3-40 Example Body Configurations

Method	API Response	Identifier	Regular Expression	Trigger Point	Operation
POST	/nnrf-nfm/v1/subscriptions	reqNfInstanceId	^[0-9A-Fa-f]{8}-[0-9A-Fa-f]{4}-4[0-9A-Fa-f]{3}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{12}\$	N32_Egress_Request	Topology Hiding
POST	/nnrf-nfm/v1/subscriptions	nfStatusNotificationUri	(?<=http:// https://)[^:/?]+	N32_Egress_Request	Topology Hiding
POST	/nnrf-nfm/v1/subscriptions	reqNfFqdn	^(?!:\/)(?=.{1,255}\$)((.{1,63}.){1,127}(?![0-9]*\$)[a-z0-9-]?)	N32_Egress_Request	Topology Hiding
POST	/nnrf-nfm/v1/subscriptions	reqNfInstanceId	^[0-9A-Fa-f]{8}-[0-9A-Fa-f]{4}-4[0-9A-Fa-f]{3}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{12}\$	N32_Ingress_Response	Topology Recovery

Table 3-40 (Cont.) Example Body Configurations

Method	API Response	Identifier	Regular Expression	Trigger Point	Operation
POST	/nnrf-nfm/v1/subscriptions	nfStatusNotificationUri	(?<=http:// https://)[^:/?]+	N32_Ingress_Response	Topology Recovery
POST	/nnrf-nfm/v1/subscriptions	reqNfFqdn	^(?!://)(?=[{1,255}\$])(.{1,63}.){1,127}(?![0-9]*\$)[a-z0-9-]?)	N32_Ingress_Response	Topology Recovery
POST	/nnrf-nfm/v1/subscriptions	hnrfUri	(?<=http:// https://)[^:/?]+	N32_Ingress_Request	Topology Recovery
POST	/nnrf-nfm/v1/subscriptions	hnrfUri	(?<=http:// https://)[^:/?]+	N32_Egress_Response	Topology Hiding

If the same apiUri having more than one identifiers then the separate rows of each identifiers must be in the database. One row always corresponds to single identifier configuration. If in case user enters the wrong or inappropriate configuration for any identifier then that particular configuration needs to be deleted and reenter back into the system

Note

If the user wants to add the apiUri and that is not present in the default table, exception or error occurs. Ensure to add the default configuration before configuring topology body.

Note

The actual to pseudo mapping should be done as mentioned in the above mentioned table. All the configuration related to actual pseudo mapping should apply in the same way.

Topology Hiding Path Configurations

As messages (requests or responses) pass through the system, their paths are evaluated using regular expressions configured in the topology path table. These expressions are used to extract specific values from the request or response, which can then be hidden (TH) or recover (TUH) based on the predefined rules. Each message is uniquely identified by an API URL, which acts as the basis for determining the applicable TH/TUH rules. For every API URL, corresponding regular expressions are configured to target relevant parts of the request or response payload. Customers can define which operation (TH/TUH) should be applied when a specific path is processed. This flexible configuration allows better control over data visibility as messages traverse the system.

Table 3-41 Example Path Configurations

Method	API Response	Regular Expression	Trigger Point	Operation
POST	/nnrf-nfm/v1/subscriptions	^[0-9A-Fa-f]{8}-[0-9A-Fa-f]{4}-4[0-9A-Fa-f]{3}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{12}\$	N32_Egress_Req est	Topology Hiding
POST	/nnrf-nfm/v1/subscriptions	(?<=http:// https://)[^/?]+	N32_Egress_Req est	Topology Hiding

Example for Topology Path:

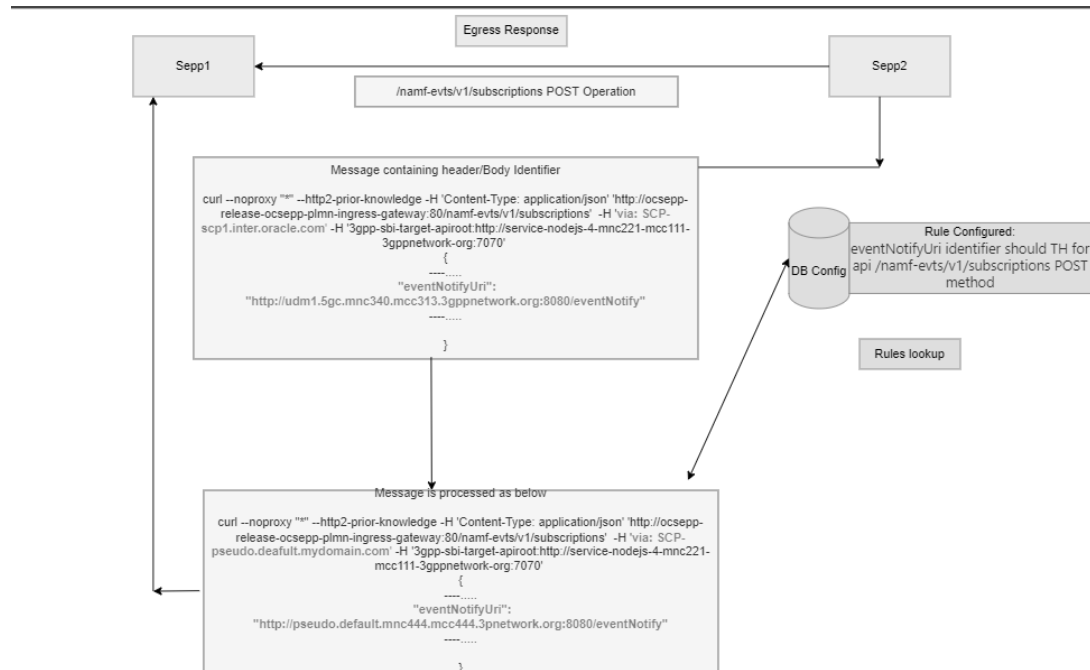
```
curl --noproxy "*" --http2-prior-knowledge -X GET -H '3gpp-sbi-target-
apiRoot: http://pseudo-perfgo-mnc100-mcc111-3gppnetwork-org.user-sepp2:9814' -
H "Content-type: application/json" 'http://ocsepp-release-user-plmn-ingress-
gateway:80/npcf-ue-policy-control/v1/policies/54804518-4191-46b3-955c-
ac631f953ed8?service-names=nnrf-disc&target-nf-type=NRF&req-nf-
type=NRF&target-plmn-
list=%5B%7B%22mcc%22%3A%22310%22%2C%20%22mnc%22%3A%22300%22%7D%5D&requester-
plmn-
list=%5B%7B%22mcc%22%3A%22310%22%2C%22mnc%22%3A%22310%22%7D%2C%7B%22mcc%22%3A%
22311%22%2C%22mnc%22%3A%22490%22%7D%5D&routing-indicator=0000&req-nf-instance-
id=44804518-4191-46b3-955c-ac631f953ed8&nrf-disc-uri=http://
nrf.5gc.mnc222.mcc333.3gppnetwork.org:80/nnrf-disc/v1/nf-instances&req-nf-
instance-fqdn=https://nrf.5gc.mnc330.mcc310.BE4NRF06.rs.oracle.com:7070'
```

In the above request, if the user wants to hide 54804518-4191-46b3-955c-ac631f953ed8 value present in the path which is in valid format (NF_INSTANCE_ID), then the path configuration for this URI is required in the path table and based on that, hiding or recovery will be performed along with actual-pseudo configuration.

Example for Topology Body:

Consider the message request body going out from the system from CN32F (Ingress response) and having the apiUrl as */namf-evts/v1/subscriptions* and containing *eventNotifyUri* identifier as *http://udm1.5gc.mnc340.mcc313.3gppnetwork.org:8080/eventNotify* and api method is of type POST. In this case, topology body table has the identifier *eventNotifyUri* configured for INGRESS RESPONSE as operation TH. This table checks the apiUrl which is actually called exists as a configuration in the table. Also, this method should be POST, so extract the *eventNotifyUri* from this message and consider this for TH.

Then look up the actual to pseudo mappings and try to find actual value as *udm1.5gc.mnc340.mcc313.3gppnetwork.org* and retrieve all the possible pseudo values associated with it. Once seven pseudo values against this actual value are identified, the system randomly select any one value and mask *eventNotifyUri* with pseudo value like *eventNotifyUri: http://pseudo.default.mnc444.mcc444.3pnetwork.org:8080/eventNotify*.

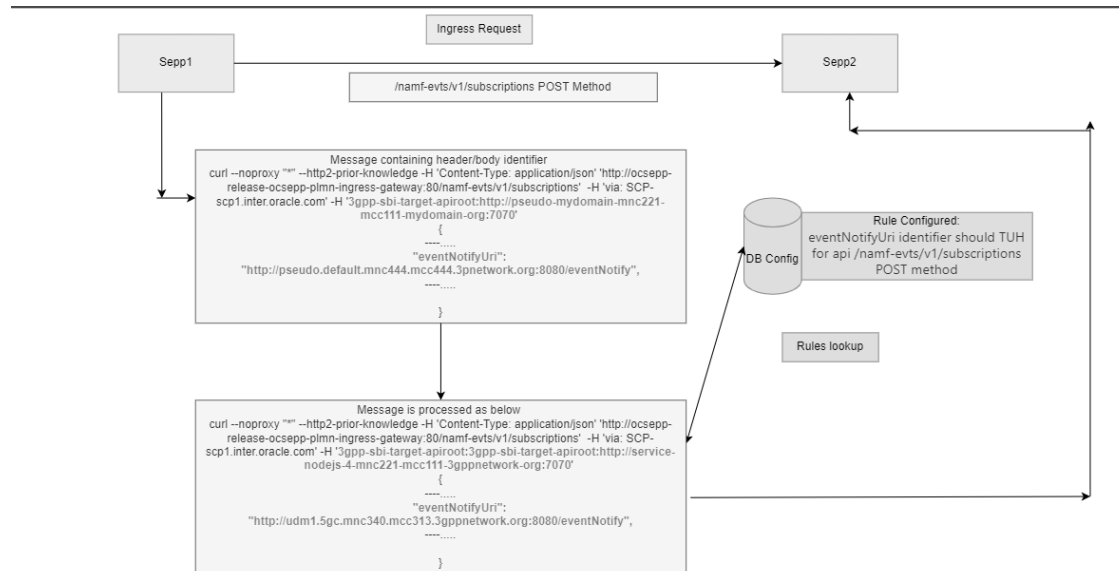
Figure 3-44 Example for Hiding the message

In the example via header is configured as hidden in the header table configuration and also eventNotifyUri as Body identifier for resource `URI /namf-evts/v1/subscriptions` in topology body configuration for the POST response. For the Egress Request message, via header is passed as actual value and eventNotifyUri as body identifier is passed as actual value. Since we have configured that TH operation should be performed therefore as per the actual/pseudo mapping mapped, TH should happen and actual values should convert into pseudo value. In the below example, it explains how we look up the database and convert the actual to pseudo value.

Example of Recovery

Consider the message request body going out from the system from CN32F (Egress Request) and having the `apiUrl` as `/namf-evts/v1/subscriptions` and containing eventNotifyUri identifier as `http://pseudo.default.mnc444.mcc444.3pnetwork.org:8080/eventNotify` and api method is of type POST. In this case, topology body table has the identifier eventNotifyUri configured for EGRESS REQUEST as operation TUH further this table checks that `apiUrl` which is actually called exists as a configuration in the table. Also, this method should be POST, so extract the eventNotifyUri from this message and consider this for TUH.

Look up the actual to pseudo mappings and find pseudo value as `pseudo.default.mnc444.mcc444.3pnetwork.org` and retrieve the actual value associated with it. System replaces the pseudo value with actual value `udm1.5gc.mnc340.mcc313.3gppnetwork.org`

Figure 3-45 Example of Recovery

In the example, 3gpp-sbi-target-apiroot header is configured as recovered in the header table configuration and also eventNotifyUri as Body identifier for resource URI */namf-evts/v1/subscriptions* in topology body configuration for the POST request. For the Ingress Request message, 3gpp-sbi-target-apiroot header is passed as pseudo value and eventNotifyUri as body identifier is passed as pseudo value. Since we have configured that TUH operation should be performed therefore as per the actual/pseudo mapping mapped, TUH should happen and pseudo values should convert into actual value. In the below example, it explains how we look up the database and convert the pseudo to actual value.

Default Configuration

The following is the default table that have all the resource URIs supported by the SEPP.

Table 3-42 Default Configuration

id	resourceURI	method	regex
1	/nudm-sdm/v2/{supi}	GET	^/nudm-sdm/v2/((imsi-[0-9]{5,15})nai-.+ gci-.+ gli-.+ .+)/?\$
2	/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation	PUT	^/nausf-auth/v1/ue-authentications/([0-9a-zA-Z-]+ .+)/5g-aka-confirmation/?\$
3	/nudm-sdm/v2/{ueId}/sdm-subscriptions	POST	^/nudm-sdm/v2/((imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ .+)/sdm-subscriptions/?\$
4	/oauth2/token	POST	^/oauth2/token/?\$
5	/nsmf-pdusession/v1/pdu-sessions/{pduSessionRef}/transfer-mo-data	POST	^/nsmf-pdusession/v1/pdu-sessions/([0-9a-zA-Z-]+ .+)/transfer-mo-data/?\$
6	/nudm-sdm/v2/{supi}/smf-select-data	GET	^/nudm-sdm/v2/((imsi-[0-9]{5,15})nai-.+ gci-.+ gli-.+ .+)/smf-select-data/?\$
7	/nnrf-nfm/v1/subscriptions	POST	^/nnrf-nfm/v1/subscriptions/?\$

Table 3-42 (Cont.) Default Configuration

id	resourceURI	method	regex
8	/nudm-uecm/v1/{ueld}/registrations/smsf-3gpp-access	GET	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ .+)/registrations/smsf-3gpp-access/?\$
9	/nudm-sdm/v2/{supi}/sms-mng-data	GET	^/nudm-sdm/v2/(imsi-[0-9]{5,15})nai-.+ gci-.+ gli-.+ .+)/sms-mng-data/?\$
10	/nudm-sdm/v2/{supi}/sm-data	GET	^/nudm-sdm/v2/(imsi-[0-9]{5,15})nai-.+ gci-.+ gli-.+ .+)/sm-data/?\$
11	/namf-mt/v1/ue-contexts/{ueContextId}	GET	^/namf-mt/v1/ue-contexts/(imsi-[0-9]{5,15})nai-.+ gli-.+ gci-.+ .+)/?\$
12	/namf-evts/v1/subscriptions/{subscriptionId}	DELETE	^/namf-evts/v1/subscriptions/(((0-9){5,6})-)?[^\-]+)/?\$
13	/namf-evts/v1/subscriptions/{subscriptionId}	PATCH	^/namf-evts/v1/subscriptions/(((0-9){5,6})-)?[^\-]+)/?\$
14	/namf-evts/v1/subscriptions	POST	^/namf-evts/v1/subscriptions/?\$
15	/nudm-sdm/v2/{supi}/am-data/sor-ack	PUT	^/nudm-sdm/v2/(imsi-[0-9]{5,15})nai-.+ gci-.+ gli-.+ .+)/am-data/sor-ack/?\$
16	/nudm-uecm/v1/{ueld}/registrations/amf-3gpp-access	PUT	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ .+)/registrations/amf-3gpp-access/?\$
17	/namf-loc/v1/{ueContextId}/provide-loc-info	POST	^/namf-loc/v1/(imsi-[0-9]{5,15})nai-.+ gli-.+ gci-.+ imei-[0-9]{15} imeisv-[0-9]{16} .+)/provide-loc-info/?\$
18	/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation	POST	^/nausf-auth/v1/ue-authentications/([0-9a-zA-Z-]+ .+)/5g-aka-confirmation/?\$
19	/nsmf-pdusession/v1/pdu-sessions/{pduSessionRef}/transfer-mt-data	POST	^/nsmf-pdusession/v1/pdu-sessions/([0-9a-zA-Z-]+ .+)/transfer-mt-data/?\$
20	/nudm-sdm/v2/{ueld}/sdm-subscriptions/{subscriptionId}	PATCH	^/nudm-sdm/v2/(imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ .+)/sdm-subscriptions/(((0-9){5,6})-)?[^\-]+)/?\$
21	/nudm-sdm/v2/{supi}/sms-data	GET	^/nudm-sdm/v2/(imsi-[0-9]{5,15})nai-.+ gci-.+ gli-.+ .+)/sms-data/?\$
22	/namf-comm/v1/subscriptions/{subscriptionId}	DELETE	^/namf-comm/v1/subscriptions/(((0-9){5,6})-)?[^\-]+)/?\$
23	/nnssf-nssselection/v2/network-slice-information	GET	^/nnssf-nssselection/v2/network-slice-information/?\$
24	/nudm-uecm/v1/{ueld}/registrations/amf-3gpp-access	GET	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ .+)/registrations/amf-3gpp-access/?\$
25	/nudm-uecm/v1/{ueld}/registrations/smsf-3gpp-access	DELETE	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ .+)/registrations/smsf-3gpp-access/?\$

Table 3-42 (Cont.) Default Configuration

id	resourceURI	method	regex
26	/nsmf-pdusession/v1/pdu-sessions/{pduSessionRef}/modify	POST	^/nsmf-pdusession/v1/pdu-sessions/([0-9a-zA-Z-]+ .+)/modify/?\$
27	/nnrf-nfm/v1/subscriptions/{subscriptionID}	PATCH	^/nnrf-nfm/v1/subscriptions/([0-9]{5,6}-)?[^-+]/?\$
28	/nudm-sdm/v2/shared-data	GET	^/nudm-sdm/v2/shared-data/?\$
29	/npcf-ue-policy-control/v1/policies	POST	^/npcf-ue-policy-control/v1/policies/?\$
30	/nnrf-disc/v1/nf-instances	GET	^/nnrf-disc/v1/nf-instances/?\$
31	/nudm-sdm/v2/{supi}/am-data/upu-ack	PUT	^/nudm-sdm/v2/(imsi-[0-9]{5,15})nai-.[+ gci-.[+ gli-.[+ .+)/am-data/upu-ack/?\$
32	/npcf-ue-policy-control/v1/policies/{polAssold}	GET	^/npcf-ue-policy-control/v1/policies/([0-9a-zA-Z-]+ .+)/?\$
33	/nudm-sdm/v2/{ueld}/sdm-subscriptions/{subscriptionId}	DELETE	^/nudm-sdm/v2/(imsi-[0-9]{5,15})nai-.[+ msisd-[0-9]{5,15} extid-[^-+] gci-.[+ gli-.[+ .+)/sdm-subscriptions/([0-9]{5,6}-)?[^-+]/?\$
34	/nudm-sdm/v2/{supi}/am-data	GET	^/nudm-sdm/v2/(imsi-[0-9]{5,15})nai-.[+ gci-.[+ gli-.[+ .+)/am-data/?\$
35	/nudm-uecm/v1/{ueld}/registrations/smsf-non-3gpp-access	GET	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.[+ msisd-[0-9]{5,15} extid-[^-+] gci-.[+ gli-.[+ .+)/registrations/smsf-non-3gpp-access/?\$
36	/nnrf-nfm/v1/subscriptions/{subscriptionID}	DELETE	^/nnrf-nfm/v1/subscriptions/([0-9]{5,6}-)?[^-+]/?\$
37	/nudm-uecm/v1/{ueld}/registrations/amf-non-3gpp-access	GET	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.[+ msisd-[0-9]{5,15} extid-[^-+] gci-.[+ gli-.[+ .+)/registrations/amf-non-3gpp-access/?\$
38	/nudm-uecm/v1/{ueld}/registrations/amf-non-3gpp-access	PUT	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.[+ msisd-[0-9]{5,15} extid-[^-+] gci-.[+ gli-.[+ .+)/registrations/amf-non-3gpp-access/?\$
39	/nudm-uecm/v1/{ueld}/registrations/smsf-non-3gpp-access	PUT	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.[+ msisd-[0-9]{5,15} extid-[^-+] gci-.[+ gli-.[+ .+)/registrations/smsf-non-3gpp-access/?\$
40	/namf-comm/v1/subscriptions/{subscriptionId}	PUT	^/namf-comm/v1/subscriptions/([0-9]{5,6}-)?[^-+]/?\$
41	/namf-comm/v1/subscriptions	POST	^/namf-comm/v1/subscriptions/?\$
42	/nausf-auth/v1/ue-authentications	POST	^/nausf-auth/v1/ue-authentications/?\$
43	/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation	DELETE	^/nausf-auth/v1/ue-authentications/([0-9a-zA-Z-]+ .+)/5g-aka-confirmation/?\$
44	/nudm-sdm/v2/{supi}/nssai	GET	^/nudm-sdm/v2/(imsi-[0-9]{5,15})nai-.[+ gci-.[+ gli-.[+ .+)/nssai/?\$
45	/npcf-ue-policy-control/v1/policies/{polAssold}	DELETE	^/npcf-ue-policy-control/v1/policies/([0-9a-zA-Z-]+ .+)/?\$

Table 3-42 (Cont.) Default Configuration

id	resourceURI	method	regex
46	/nsmf-pdusession/v1/pdu-sessions/{pduSessionRef}/release	POST	^/nsmf-pdusession/v1/pdu-sessions/([0-9a-zA-Z-]+ . +)/release/?\$
47	/nudm-uecm/v1/{ueId}/registrations/smsf-3gpp-access	PUT	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ . +)/registrations/smsf-3gpp-access/?\$
48	/nsmf-pdusession/v1/pdu-sessions	POST	^/nsmf-pdusession/v1/pdu-sessions/?\$
49	/nudm-uecm/v1/{ueId}/registrations/smsf-non-3gpp-access	DELETE	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ . +)/registrations/smsf-non-3gpp-access/?\$
50	/nudm-sdm/v2/{supi}/ue-context-in-smf-data	GET	^/nudm-sdm/v2/(imsi-[0-9]{5,15})nai-.+ gci-.+ gli-.+ . +)/ue-context-in-smf-data/?\$
51	/nudm-uecm/v1/{ueId}/registrations/amf-non-3gpp-access	PATCH	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ . +)/registrations/amf-non-3gpp-access/?\$
52	/nudm-uecm/v1/{ueId}/registrations/smf-registrations/{pduSessionId}	DELETE	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ . +)/registrations/smf-registrations/([01]?[0-9][0-9]?[2][0-4][0-9] 25[0-5])+/?\$
53	/nudm-sdm/v2/{supi}/ue-context-in-smf-data	GET	^/nudm-sdm/v2/(imsi-[0-9]{5,15})nai-.+ gci-.+ gli-.+ . +)/ue-context-in-smf-data/?\$
54	/npcf-ue-policy-control/v1/policies/{polAssId}/update	POST	^/npcf-ue-policy-control/v1/policies/([0-9a-zA-Z-]+ . +)/update/?\$
55	/nudm-uecm/v1/{ueId}/registrations/amf-3gpp-access	PATCH	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ . +)/registrations/amf-3gpp-access/?\$
56	/nudm-uecm/v1/{ueId}/registrations/smf-registrations/{pduSessionId}	PUT	^/nudm-uecm/v1/(imsi-[0-9]{5,15})nai-.+ msisdn-[0-9]{5,15} extid-[^\@]+\@[^\@]+ gci-.+ gli-.+ . +)/registrations/smf-registrations/([01]?[0-9][0-9]?[2][0-4][0-9] 25[0-5])+/?\$

- If the user wants to create any rules base on the resource URI check, then the resource URI should be present in the above table.
- If in case, the resource URI does not exist and user wants that SEPP should support that resource URI from topology hiding perspective then using the CNC Console GUI, the user must first enter the resource URI in this table and then only start the body configuration of topology body as explained in the topology body configuration.
- If in case resource URI is not present in the default table and user wants to add the resource URI in topology configuration, then user to see the api URI in the dropdown menu of CNC Console.

This table is populated from the set of identified resource URIs at the time of start-up and if any additional resource URI needs to be added then user can add that resource URI from the CNCC Screen.

Managing Topology Hiding

Enable

Topology Hiding feature is disabled by default. You can enable Topology Hiding feature using the REST API or CNC Console.

- Enable using REST API: Set ENABLED to True in Topology Hiding API. For more information about API path, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy REST Specification Guide*.
- Enable using CNC Console: Set ENABLED to True on the Topology Hiding page. For more information about enabling the feature using CNC Console, see [Configuring SEPP using CNC Console](#).

Configure

You can configure the Topology Hiding using REST API and CNC Console.

- Configure Topology Hiding using REST API: Perform the feature configurations as described *Topology hiding* section in *Oracle Communications Security Edge Protection Proxy Rest API Guide*.
- Configure Topology Hiding using CNC Console: Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Observe

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.27 Global Rate Limiting on Egress Gateway of SEPP

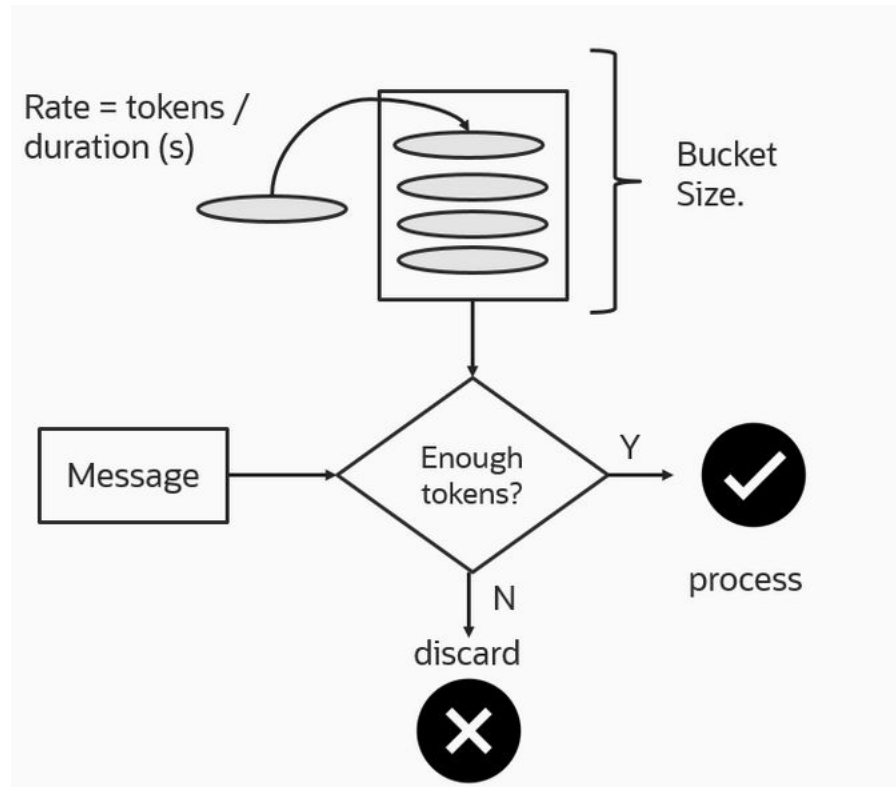
Feature Overview

Global rate limiting is a feature provided by Egress Gateway. There are cases where the messages flood into SEPP. With this feature, SEPP secures the network when aggregated egress traffic exceeds the allowed traffic rate limit. If the traffic exceeds the allowed traffic rate limit, SEPP does not process the traffic and responds with an error code. Egress global rate limiting functionality of the SEPP allows the user to configure the acceptable egress traffic rate to any registered NF instance. SEPP enables users to configure the maximum number of outgoing messages at a given duration.

Algorithm

Global Rate limiting on Egress Gateway is implemented using Bucket4j which uses Token Bucket Algorithm. The token bucket algorithm is as follows:

Figure 3-46 Algorithm Overview



Token bucket algorithm process messages only if there are tokens in the bucket. The bucket is asynchronously filled with a configurable rate and tokens are removed from the bucket on message processing.

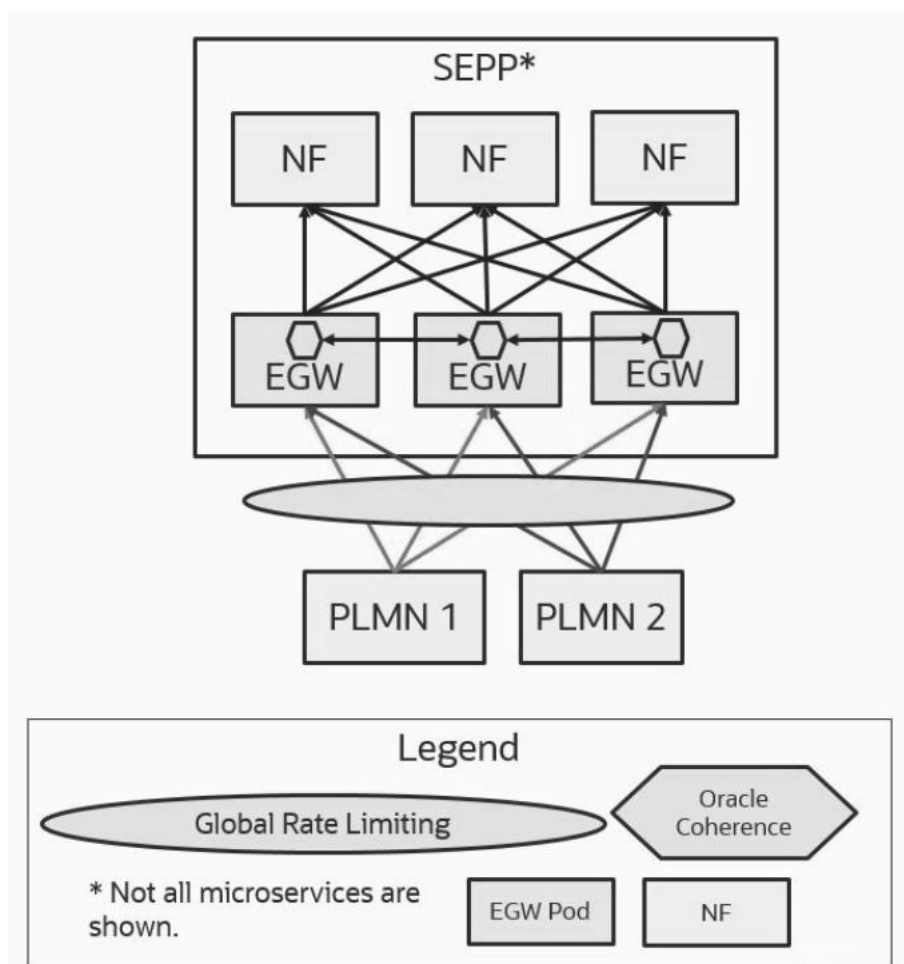
Three configurations are required for this algorithm:

- Bucket size in which capacity to handle traffic burst is defined.
- Duration to decide how frequently to refill bucket.
- Tokens (number of) which should be added to refill the bucket.

Design and Architecture

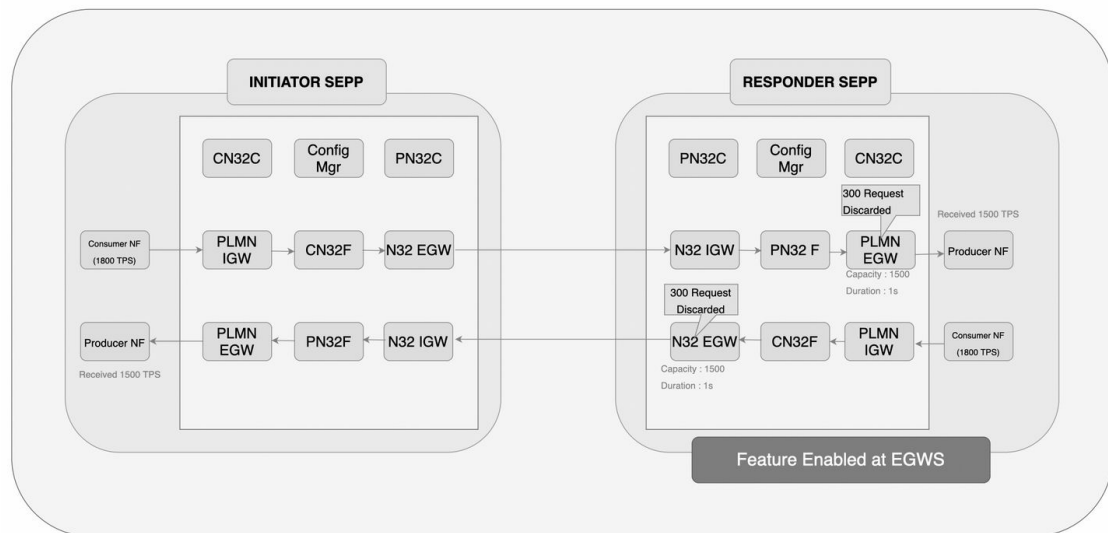
Using Egress global rate limiting, the user can configure capacity for SEPP. The incoming traffic management at the gateway level and any traffic beyond the capacity of the backend service get rejected with the error response as configured by the user. The rules for global rate limiting apply to all traffic, irrespective of headers

Figure 3-47 Architecture



The following diagram represents the functionality of the global rate limiting feature with the example:

Figure 3-48 Example



In the above diagram, the global rate limiting feature is enabled on the Responder SEPP Egress Gateway with a capacity of 1500 requests with one second duration. The requests are initiated from the customer NF at initiator side which passes through the initiator SEPP and reach to the responder SEPP Egress Gateway.

If the incoming requests on Egress Gateway of Responder SEPP are more than the configured capacity of 1500 request per second, the requests above the configured value are rejected with an error response as configured by the user. In the above example, the consumer NF is generating 1800 TPS and as per the capacity set at Egress Gateway, 300 requests are rejected with an configured error response. Similarly, if requests are initiated from producer NF at responder side, then the Egress Gateway at responder SEPP will discard the excess request and allow only the configured limit.

Managing Global Rate Limiting on Egress Gateway of SEPP

Enable

The Egress Rate Limiting feature is disabled by default in SEPP at the time of deployment. To enable the feature, set `egressRateLimiting.enabled` to true in the PLMN Egress Gateway and/or N32 Egress Gateway of `ocsepp-custom-values-<version>.yaml` file.

`ocsepp-custom-values-<version>.yaml`

The following parameters are to be configured:

```
###In the EGW section
egressRateLimiting:
  enabled: true
  duration: 1 # in seconds
  burstCapacity: 4000
  refillRate: 4000
  errorCodeOnRateLimit: 429
```

Configure

1. Configure the following parameters to configure the given range of min and max token requests into equal separated intervals.

ocsepp-custom-values-<version>.yaml

```
###In the EGW section of custom-values
egressRateLimiting:
  enabled: true
  duration: 1
  bucketCapacity: 4000
  refillRate: 4000
  errorCodeOnRateLimit: 429
###In the EGW section for egressRateLimiting of values.yaml
minTokenRequest: 5
maxTokenRequest: 10
rangePoint: 6
```

2. Add configurable error codes for Egress Gateway with exceptionType with the value EGRESS_RATE_LIMITING as given below example:

```
curl 127.0.0.1:9090/sepp/nf-common-component/v1/egw/<GW>/
configurableerrorcodes -XPUT -H 'Content-type: application/json' -d'
{"enabled": true,"errorScenarios":
[{"exceptionType":"EGRESS_RATE_LIMITING","errorProfileName":"ERR_1200"},
{"exceptionType":"CONNECTION_TIMEOUT","errorProfileName":"ERR_100"},
{"exceptionType":"REQUEST_TIMEOUT","errorProfileName":"ERR_REQUEST_TIMEOUT"
}],
{"exceptionType":"UNKNOWN_HOST_EXCEPTION","errorProfileName":"ERR_UNKNOWN_H
OST"}, {"exceptionType":"CONNECT_EXCEPTION","errorProfileName":"ERR_400"},
{"exceptionType":"REJECTED_EXECUTION_EXCEPTION","errorProfileName":"ERR_500
"},
{"exceptionType":"CLOSED_CHANNEL_EXCEPTION","errorProfileName":"ERR_600"},
{"exceptionType":"SSL_HANDSHAKE_EXCEPTION","errorProfileName":"ERR_700"},
{"exceptionType":"INTERNAL_ERROR","errorProfileName":"ERR_800"},
{"exceptionType":"NOT_FOUND_EXCEPTION","errorProfileName":"ERR_NOT_FOUND"},
{"exceptionType":"LATE_ARRIVAL_EXCEPTION","errorProfileName":"ERR_1000"},
{"exceptionType":"INVALID_HTTP_REQUEST_EXCEPTION","errorProfileName":"ERR_1
100"},
{"exceptionType":"NOTIFICATION_RATE_LIMIT","errorProfileName":"ERR_1100"},
{"exceptionType":"RETRY_AFTER_ERROR","errorProfileName":"ERR_1100"},
{"exceptionType":"GRACEFUL_SHUTDOWN","errorProfileName":"ERR_1100"},
{"exceptionType":"VIRTUAL_HOST_RESOLUTION_ERROR","errorProfileName":"ERR_11
00"},
{"exceptionType":"INVALID_OAUTH_TOKEN_REQUEST","errorProfileName":"ERR_1100
"}, {"exceptionType":"OAUTH_INTERNAL_ERROR","errorProfileName":"ERR_1100"},
{"exceptionType":"OAUTH_TOKEN_RETRIEVAL_FAILURE","errorProfileName":"ERR_11
00"},
{"exceptionType":"OAUTH_NRF_RESPONSE_FAILURE","errorProfileName":"ERR_1100"
}]}'
```

Example:

```
curl 127.0.0.1:9090/sepp/nf-common-component/v1/egw/plmn/
configurableerrorcodes -XPUT -H 'Content-type: application/json' -d'
{"enabled": true,"errorScenarios":
[{"exceptionType":"EGRESS_RATE_LIMITING","errorProfileName":"ERR_1200"},
{"exceptionType":"CONNECTION_TIMEOUT","errorProfileName":"ERR_100"},
{"exceptionType":"REQUEST_TIMEOUT","errorProfileName":"ERR_REQUEST_TIMEOUT"
},
{"exceptionType":"UNKNOWN_HOST_EXCEPTION","errorProfileName":"ERR_UNKNOWN_H
OST"}, {"exceptionType":"CONNECT_EXCEPTION","errorProfileName":"ERR_400"},
{"exceptionType":"REJECTED_EXECUTION_EXCEPTION","errorProfileName":"ERR_500
"},
{"exceptionType":"CLOSED_CHANNEL_EXCEPTION","errorProfileName":"ERR_600"},
{"exceptionType":"SSL_HANDSHAKE_EXCEPTION","errorProfileName":"ERR_700"},
{"exceptionType":"INTERNAL_ERROR","errorProfileName":"ERR_800"},
{"exceptionType":"NOT_FOUND_EXCEPTION","errorProfileName":"ERR_NOT_FOUND"},
{"exceptionType":"LATE_ARRIVAL_EXCEPTION","errorProfileName":"ERR_1000"},
{"exceptionType":"INVALID_HTTP_REQUEST_EXCEPTION","errorProfileName":"ERR_1
100"},
{"exceptionType":"NOTIFICATION_RATE_LIMIT","errorProfileName":"ERR_1100"},
{"exceptionType":"RETRY_AFTER_ERROR","errorProfileName":"ERR_1100"},
{"exceptionType":"GRACEFUL_SHUTDOWN","errorProfileName":"ERR_1100"},
{"exceptionType":"VIRTUAL_HOST_RESOLUTION_ERROR","errorProfileName":"ERR_11
00"},
{"exceptionType":"INVALID_OAUTH_TOKEN_REQUEST","errorProfileName":"ERR_1100
"}, {"exceptionType":"OAUTH_INTERNAL_ERROR","errorProfileName":"ERR_1100"},
{"exceptionType":"OAUTH_TOKEN_RETRIEVAL_FAILURE","errorProfileName":"ERR_11
00"},
{"exceptionType":"OAUTH_NRF_RESPONSE_FAILURE","errorProfileName":"ERR_1100"
}]]'
```

3. Configure the errorcode profile mentioned in the configurable error codes mentioned in the step 2. This step is to configure the errorCode, errorCause, errorTitle, and errorDescription parameters to be sent in response when the message is discarded.

```
curl 127.0.0.1:9090/sepp/nf-common-component/v1/egw/<GW>/errorcodeprofiles
-XPUT -H 'Content-type: application/json' -d'
[{"name":"ERR_NOT_FOUND","errorCode":404,"errorCause":"Cause Not
Found","errorTitle":"errorTitle Not
Found","errorDescription":"errorDescription Not Found"},
{"name":"ERR_REQUEST_TIMEOUT","errorCode":503},
{"name":"ERR_UNKNOWN_HOST","errorCode":400,"errorCause":"Unknown Host
Exception at EGW","errorTitle":"Unknown Host
Exception","errorDescription":"Unknown Host Exception"},
{"name":"ERR_1200","errorCode":428,"errorCause":"Discarded due to rate
limit","errorTitle":"Discard Rate limit","errorDescription":"Egress rate
limiting discard"}]
```

Example:

```
curl 127.0.0.1:9090/sepp/nf-common-component/v1/egw/plmn/errorcodeprofiles
-XPUT -H 'Content-type: application/json' -d'
[{"name":"ERR_NOT_FOUND","errorCode":404,"errorCause":"Cause Not
Found","errorTitle":"errorTitle Not
Found","errorDescription":"errorDescription Not Found"},
{"name":"ERR_REQUEST_TIMEOUT","errorCode":503},
{"name":"ERR_UNKNOWN_HOST","errorCode":400,"errorCause":"Unknown Host
Exception at EGW","errorTitle":"Unknown Host
Exception","errorDescription":"Unknown Host Exception"},
{"name":"ERR_1200","errorCode":428,"errorCause":"Discarded due to rate
limit","errorTitle":"Discard Rate limit","errorDescription":"Egress rate
limiting discard"}]
```

```
curl 127.0.0.1:9090/sepp/nf-common-component/v1/egw/n32/errorcodeprofiles -
XPUT -H 'Content-type: application/json' -d'
[{"name":"ERR_NOT_FOUND","errorCode":404,"errorCause":"Cause Not
Found","errorTitle":"errorTitle Not
Found","errorDescription":"errorDescription Not Found"},
{"name":"ERR_REQUEST_TIMEOUT","errorCode":503},
{"name":"ERR_UNKNOWN_HOST","errorCode":400,"errorCause":"Unknown Host
Exception at EGW","errorTitle":"Unknown Host
Exception","errorDescription":"Unknown Host Exception"},
{"name":"ERR_1200","errorCode":428,"errorCause":"Discarded due to rate
limit","errorTitle":"Discard Rate limit","errorDescription":"Egress rate
limiting discard"}]
```

Observability

There are no new metrics and KPIs are added or updated in as part of this feature.

For more information on SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.28 Global Rate Limiting on Ingress Gateway of SEPP

Feature Overview

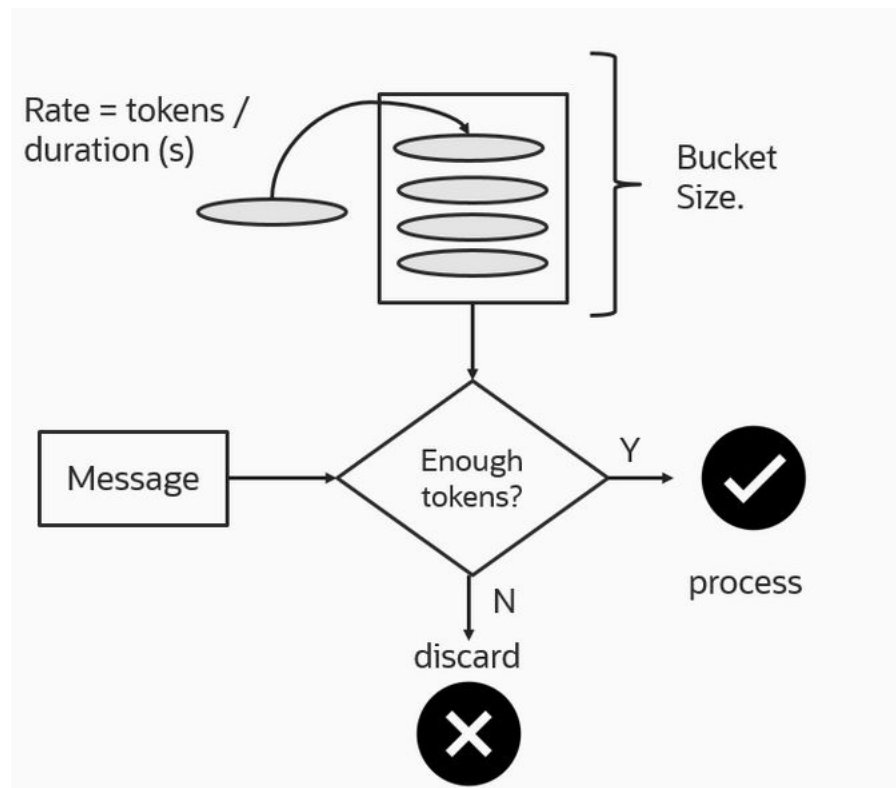
Global rate limiting is a feature provided by Ingress Gateway. There are cases where the messages floods into SEPP. With this feature, SEPP secures the network when aggregated ingress traffic from any registered NF instance exceeds the allowed traffic rate limit. If the traffic

exceeds the allowed traffic rate limit, SEPP does not process the traffic and responds with an error code. Ingress global rate limiting functionality of the SEPP allows the user to configure the acceptable traffic rate from a consumer NF instance. SEPP enables users to configure the maximum number of incoming messages at a given duration.

Algorithm

Global Rate limiting on ingress gateway is implemented using Bucket4j which uses Token Bucket Algorithm. The token bucket algorithm is as follows:

Figure 3-49 Algorithm Overview



Token bucket algorithm process messages only if there are tokens in the bucket. The bucket is asynchronously filled with a configurable rate and tokens are removed from the bucket on message processing.

Three configurations are required for this algorithm:

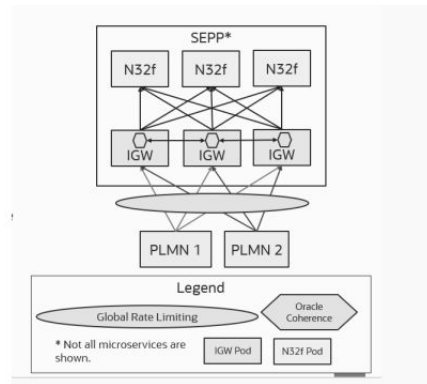
- Bucket size in which capacity to handle traffic burst is defined.
- Duration to decide how frequently to refill bucket.
- Tokens (number of) which should be added to refill the bucket.

Design and Architecture

Using Global rate limiting, the user can configure capacity for backend service (n32f service). The incoming traffic management at the gateway level and any traffic beyond the capacity of

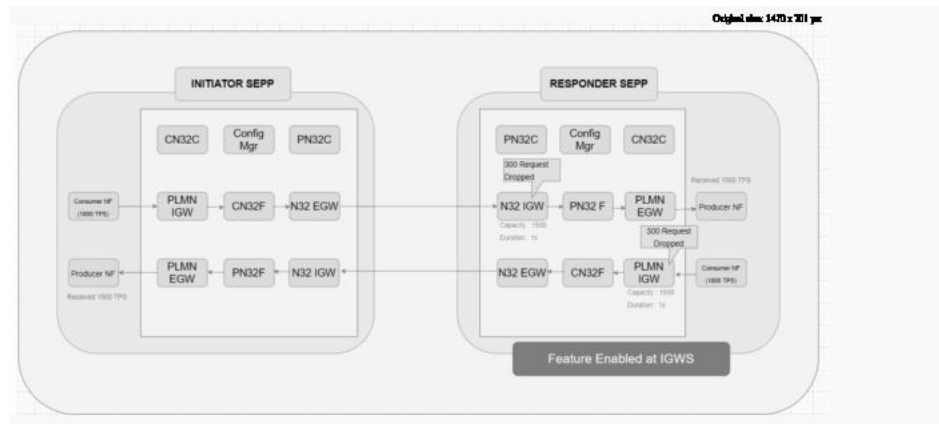
the backend service get rejected with the error response as configured by the user. The rules for global rate limiting apply to all traffic, irrespective of headers

Figure 3-50 Architecture



The following diagram represents the functionality of the global rate limiting feature with the example:

Figure 3-51 Example



In the above diagram, the global rate limiting feature is enabled on the Responder SEPP Ingress Gateway with a capacity of 1500 requests with one second duration. The requests are initiated from the customer NF at initiator side which passes through the initiator SEPP and reach to the responder SEPP Ingress Gateway.

If the incoming requests on Ingress Gateway of Responder SEPP are more than the configured capacity of 1500 request per second, the requests above the configured value are rejected with an error response as configured by the user. In the above example, the consumer NF is generating 1800 TPS and as per the capacity set at Ingress Gateway, 300 requests are rejected with an configured error response. Similarly, if requests are initiated from producer NF at responder side, then the Ingress Gateway at responder SEPP will discard the excess request and allow only the configured limit.

Managing Global Rate Limiting on Ingress Gateway of SEPP

Enable

The Ingress Rate Limiting feature is a core functionality of SEPP. This feature is automatically enabled in SEPP at the time of deployment.

ocsepp_custom_values_<version>.yaml

The following parameters are to be configured:

```
###In the IGW section
rateLimiting:
  enabled: true

  globalIngressRateLimiting:
    enabled: true
    duration: 1 # in seconds
    burstCapacity: 4000
    refillRate: 4000
    errorCodeOnRatelimit: 429
```

Configure

Configure the following parameters to configure the given range of min and max token requests into equal separated intervals.

ocsepp/values.yaml

```
###In the IGW section
minTokenRequest: 5
maxTokenRequest: 10
rangePoint: 6
```

Observability

This section describes the information about the KPIs and metrics.

Metrics

The following table describes the newly introduced metrics as part of this feature.

For PLMN Ingress Gateway and N32 Ingress Gateway:

Table 3-43 oc_ingressgateway_global_ratelimit_total

Metric Type	Tags Available
oc_ingressgateway_global_ratelimit_total	- Method - Route_path - Scheme - InstanceIdentifier - Status

KPIs

The following tables describe the newly introduced KPIs as part of this feature:

Table 3-44 N32 IGW Global Rate limit Traffic Rejected

Field	Details
KPI Detail	Measures the N32 IGW Global rate limit traffic rejected
Metric Used for KPI	sum(irate(oc_ingressgateway_global_ratelimit_total{namespace=~"\$Namespace",app="n32-ingress-gateway", Status="dropped"}[2m])) No. of messages rejected for traffic initiated from consumer side

Table 3-45 PLMN IGW Global Rate limit Traffic Rejected

Field	Details
KPI Detail	Measures the PLMN IGW Global rate limit traffic rejected
Metric Used for KPI	sum(irate(oc_ingressgateway_global_ratelimit_total{namespace=~"\$Namespace",app="plmn-ingress-gateway", Status="dropped"}[2m])) No. of messages rejected for traffic initiated from producer side

For more information on Global Rate Limiting on Ingress Gateway of SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.29 Multiple PLMN Support for Local and Remote SEPP

In earlier releases, the local SEPP could route to only a single roaming partner and a particular PLMNID for that roaming partner.

Generally, it is preferred to use a single SEPP to route to multiple roaming partners and to support a list of PLMN IDs. This feature allows the local SEPP to communicate with multiple remote partners simultaneously. With this feature, SEPP can route to a minimum of 1 and a maximum of 30 PLMN IDs at the same time.

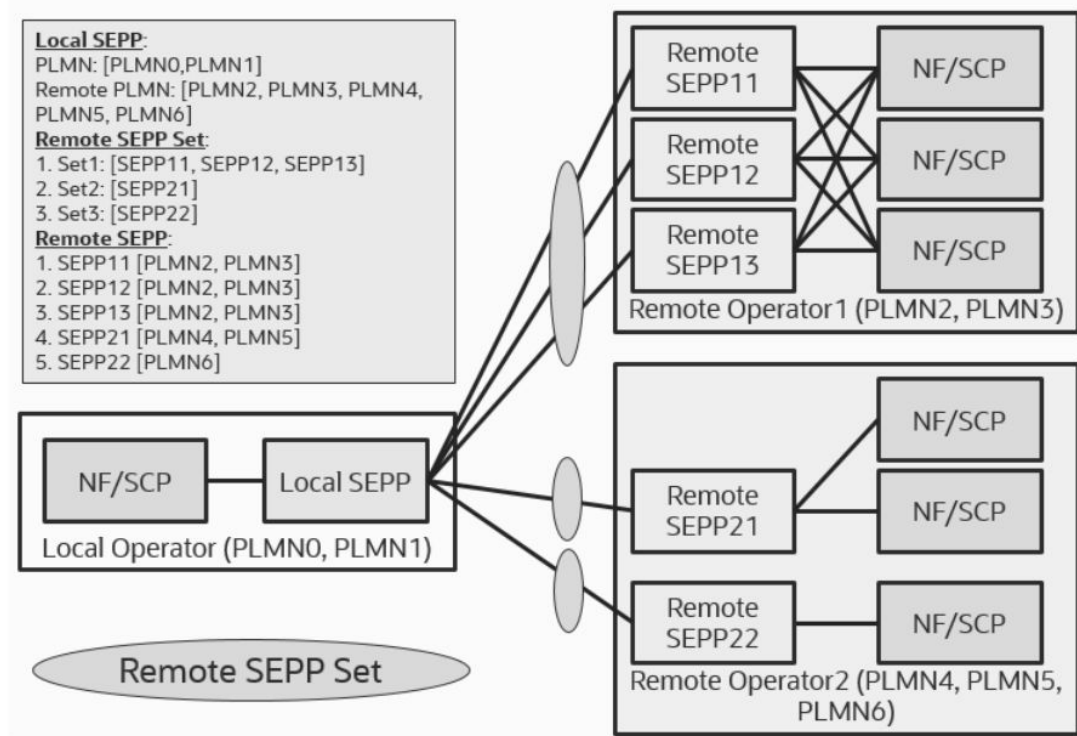
Feature Overview

The purpose of implementing this feature is to configure a single SEPP instance to support multiple PLMNs.

Design and Architecture

In this feature, one SEPP instance supports multiple PLMNs. The following diagram represents the detailed architecture of the Multiple PLMN Support for Local and Remote SEPP.

Figure 3-52 Multiple PLMN Support



Note

At present, in MNO SEPP mode, the deployment and configuration support a single serving PLMN, with up to 30 PLMN IDs mapped to that PLMN for identification and routing purposes.

Configuration

The SEPP can be configured as Local SEPP (the SEPP at the consumer side, cSEPP) and as Remote SEPP (the SEPP at the producer side, pSEPP). To initiate the communication between a Local SEPP and Remote SEPP, the Local and Remote SEPP sides must be configured to have the list of SEPPs that are assigned as primary, secondary, or tertiary SEPPs to support the alternate routing as per the availability of SEPPs. Each Remote SEPP Set must have unique list of PLMNs supported by that particular set.

In the above diagram the SEPPs, Remote SEPP Sets, and PLMNs are configured as follows:

- Local SEPP supports PLMN0 and PLMN1.
- Remote SEPP11, SEPP12, and SEPP13 shares the PLMN2 and PLMN3 networks.
- The Remote SEPP11, Remote SEPP12, and Remote SEPP13 are working as primary, secondary and tertiary SEPPs, as they are part of the same remote SEPP set.

- Remote SEPP21 and Remote SEPP22 belongs to the remote operator 2. Remote SEPP 21 supports PLMN4 and PLMN5. Remote SEPP 22 supports PLMN 6.
- Remote SEPP 21 and 22 belongs to same operator, but they are not sharing the same PLMN list. So they are reachable through different remote SEPP sets.

Functioning and Routing

Once the SEPPs are installed and configured, the communication between the two SEPPs are through two interfaces, N32c and N32f:

- N32c is interface between the SEPPs for performing initial handshake and negotiating the parameters to be applied for the actual N32 message forwarding. The handshake initiation is modified to support the Multiple PLMN For Local and Remote SEPP.
- N32f is the interface between the SEPPs for forwarding the communication between the NF service consumer and the NF service producer. While sending the request, the plmnid configured at handshake must be validated by comparing the plmnid in the *fqdn* of the *3gpp-sbi-target-apiRoot* or *Authority header*. Once the plmnid is validated, then the messages are forwarded to the destined NFs.

Note

The following table lists the Range of RemoteSepp Set and PLMN ID:

Table 3-46 Range of RemoteSepp Set and PLMN ID

Mode/Parameter	PLMN ID per RemoteSepp	RemoteSepp Set	Local PLMN ID List (Self PLMN)
SEPP Mode	Minimum = 1 Maximum = 30	Minimum = 1 Maximum = 1000	Minimum = 1 Maximum = 30

RemoteSepp = SEPP in Remote PLMN

RemoteSepp Set = RemoteSepp sets are created per PLMN. It consists of minimum 1 and maximum 3 RemoteSepp in Primary-Secondary-Tertiary mode and used for N32F SBI message routing.

Managing Multiple PLMN support for local and remote SEPP

Enable

This feature is the core functionality of SEPP. You do not need to enable or disable this feature. It is enabled by default.

Note

If plmnList is updated in *NrfClient: profile: appProfiles* section, after performing the helm-upgrade user needs to restart the NF Management pods (nfmanagement and nfdiscovery) to load the latest profile in NRF.

Configure

You can configure the Multiple PLMN support for local and remote SEPP using REST API and CNC Console.

- Configure Multiple PLMN support for local and remote SEPP using REST API: Perform the feature configurations as described *Remote SEPP Set* and *Remote SEPP* section in *Oracle Communications Security Edge Protection Proxy Rest API Guide*.
- Configure Multiple PLMN support for local and remote SEPP using CNC Console: Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Observe

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.30 Support of TLS for Roaming Hubs and IPX Providers

Oracle SEPP is deployed by both Mobile Network Operators (MNO) and Roaming Hub providers. MNOs deploy SEPP acting as 3GPP 5GC SEPP NEF which enables inter PLMN communication between two networks via N32 interface.

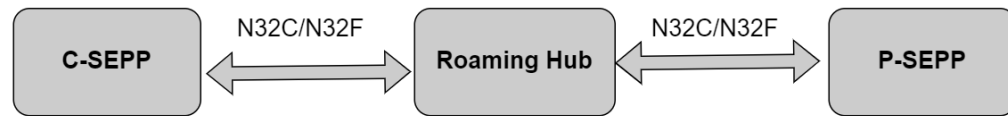
Roaming Hub providers deploy SEPP acting as Roaming Hub enabling inter PLMN communication between MNOs through N32 interface supporting Application Layer Security (ALS) as specified in TS 33.501. Roaming Hub provider is able to deploy SEPP in either of one mode i.e. functionality of 5GC SEPP NF or Roaming Hub. Roaming Hub providers are able to host 5GC SEPP NF functionality for few of the MNO's and host Roaming Hub functionality for rest of the MNOs.

There are scenarios where the user wants to deploy an intermediate proxy. Currently, for SEPP to be a viable proxy 3GPP specifies that a mode is known as PRINS be supported on the Roaming Hub.

PRINS mode has implementation challenges. To overcome this, there is an option where SEPP can be deployed as an interconnect proxy without the use of PRINS. This is commonly called as the Supporting Roaming Hub using TLS mode.

The following diagram represents a high-level overview of Support of TLS for Roaming Hubs and IPX Providers.

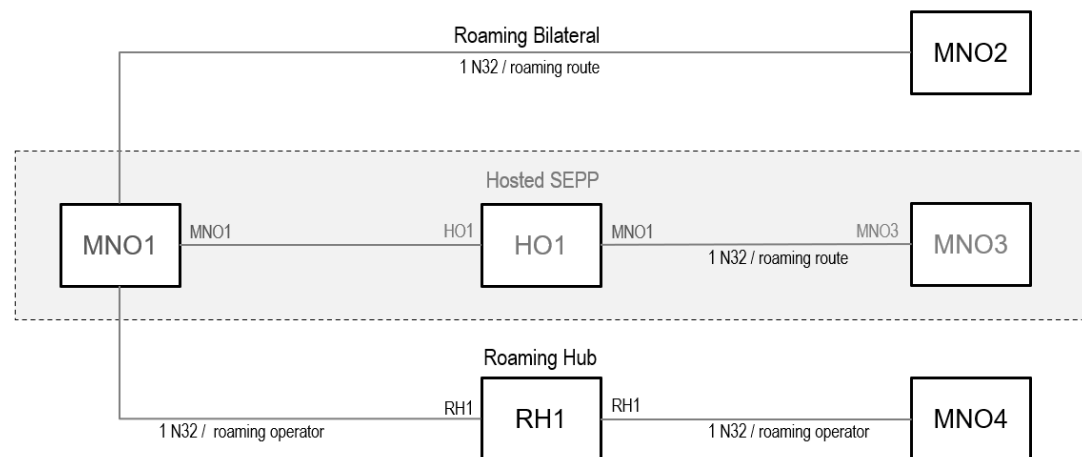
Figure 3-53 Roaming Hub Overview



SEPP supports inter PLMN traffic for both SEPP MNO and Roaming Hub modes.

Detailed Description

Figure 3-54 Detailed Description



2

In previous releases, when operators require to have interconnect with other operators, a Security Edge Protection Proxy allows for the following:

- A single point of entry into and out of the network
- This single point of entry can help in having a single point of interconnect
- Security to networks (not exposing all the NF's to the external network)

With the Support of TLS for Roaming Hubs and IPX Providers feature,

- The interconnect operators functions as an aggregator for roaming interconnects.
- The Operators can scale out as interconnect and can enable interconnects to become a roaming HUB.
- To support this, multiple SEPPs can be deployed by a Roaming Hub.
- Each SEPP within the Roaming Hub Provider needs to implement complex routing feature in order to route message from one consumer SEPP to a producer SEPP within the Roaming Hub.

Note

The following table lists the Range of RemoteSepp Set and PLMN ID:

Table 3-47 Range of RemoteSepp Set and PLMN ID

Mode/Parameter	PLMN ID per RemoteSepp	RemoteSepp Set	Local PLMN ID List (Self PLMN)
Roaming Hub Mode	Minimum = 1 Maximum = 20	Minimum = 1 Maximum = MaxRSS The value can be derived from : $\text{MaxRSS} == (400 - \text{total number of PLMNs used}) / \text{Maximum PLMN per RemoteSeppSet}$	Minimum = 1 Maximum = 400 Local PLMN is consolidate list of PLMN IDs of all the RemoteSepp(s) associated with this Roaming Hub

RemoteSepp = SEPP in Remote PLMN

RemoteSepp Set = RemoteSepp sets are created per PLMN. It consists of minimum 1 and maximum 3 RemoteSepp in Primary-Secondary-Tertiary mode and used for N32F SBI message routing.

Managing Support of TLS for Roaming Hubs and IPX Providers

Enable

This feature is the core functionality of SEPP. You do not need to enable or disable this feature. It is enabled by default.

Configure

You can configure the Support of TLS for Roaming Hubs and IPX Providers feature using REST API, and CNC Console.

- Configure Support of TLS for Roaming Hubs and IPX Providers feature using REST API: Perform the feature configurations as described *logging Configurations* section in *Oracle Communications Security Edge Protection Proxy Rest API Guide*.
- Configure Support of TLS for Roaming Hubs and IPX Providers feature using CNC Console: Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Observe

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.31 Overload Control

Overview

SEPP processes 5G Service Based Interface (SBI) messages using its microservices. To protect the microservices from the overload, SEPP supports Overload Control feature. The feature provided by Ingress Gateway and perf-info. This section describes the SEPP support for this feature.

Overload occurs when 100% of the planned capacity for cpu and memory for service is exhausted or message queue is filled due to unavailability of microservices. It can be due to uneven distribution of traffic towards a given SEPP service instance, network fluctuations leading to traffic bursts, or unexpected high traffic volume at any given point of time. During overload conditions, the service's response time may grow to unacceptable levels, and exhaustion of resources may force service to unexpected behavior or even downtime. Overload control helps to control services downtime and ensures service availability during extreme overload conditions. Using Ingress Gateway and perf-info microservices, SEPP provides options to control network congestion and handle the overload scenarios.

Detailed Description

The perf-info calculates CPU and memory usage. Based on the configured threshold levels, the perf-info applies the threshold level defined. The Ingress Gateway allows or discards incoming requests, based on the overload threshold applied. There are 2 types of discard policies supported; Percentage based and Priority based.

In percentage based discard policy, discard the certain percentage of messages once the traffic is reached threshold level. In priority based discard policy, if SEPP is overloaded, only the high priority messages are processed. Low priority traffic is discarded with the configured error code.

SEPP allows the user to configure the threshold overload levels.

Setting the Threshold Level

User can configure the threshold level for CPU and Memory usage.

Threshold levels defined are Warning , Minor, Major, and Critical. When values for CPU and Memory reach the configured limits, data is discarded on the basis of configured values for the threshold level defined.

Overload Control uses the following metrics to determine and apply suitable load level :

- CPU
- memory
- svc_pending_count

After calculation of all metrics, an appropriate load level is selected. Corresponding to the selected load level , appropriate discard actions are applied on incoming traffic. The discard actions are applied in subsequent sampling periods.

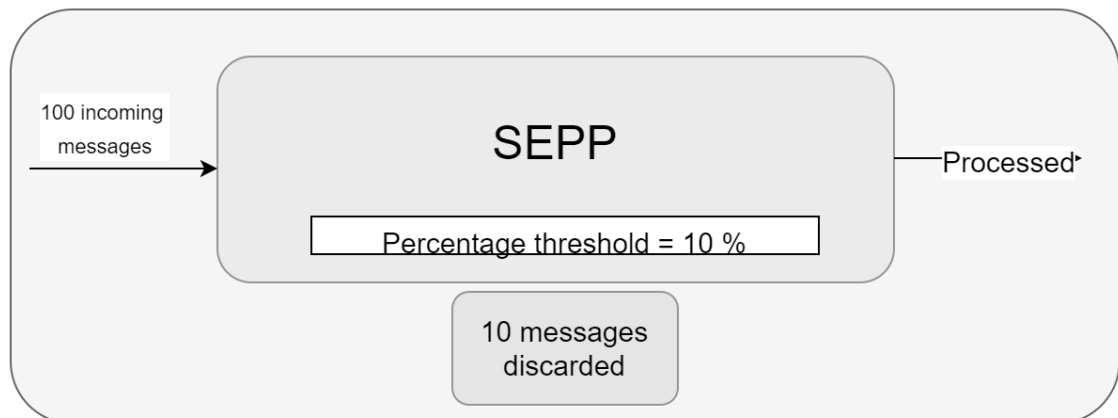
The following table represents the example for threshold set for CPU, memory, pending count, and the discard values.

Table 3-48 Threshold Level

Threshold Level	CPU onset	CPU abatement	Memory onset	Memory abatement	Pending count onset	Pending count abatement	Percentage Discard	Priority threshold
Warning	65	60	50	45	100	60	0%	30
Minor	75	70	70	65	200	125	5%	24
Major	85	80	80	75	260	201	10%	15
Critical	95	90	90	85	400	300	25%	10

Percentage based discard policy

The percentage based discard policy is applied on the incoming traffic. When SEPP is overloaded and if the discard scheme is Percentage based, the number of messages discarded corresponds to the threshold percentage level defined.

Figure 3-55 Percentage based discard policy

In the above diagram, percentage threshold level is set as 10%. Therefore for incoming traffic rate at 100 TPS, 10 messages will be discarded with the configured error code.

Note

When the percentage-based overload control discarding policy is enabled, the number of requests to be dropped in the current sampling period is computed based on the configured percentage discard and the "rate of requests outgoing of Ingress Gateway" in the previous sampling period for the service. Sampling Period can have max value as 60 seconds.

Once the number of requests to be dropped in the current sampling period is computed, the gateway does not drop all the new traffic to meet the discard count. Instead, Ingress Gateway executes a random function to decide if a request is to be discarded or not. If the random function returns true, the request is discarded in the current sampling period with the discard action "RejectWithErrorCode". This ensures there is a spread of discard requests in a sampling period.

Since we are calculating the number of requests to be dropped in the current sampling period based on the number of requests sent to the backend service in the previous sampling period and not on the total requests received at Ingress Gateway, the percentage dropped is not exactly the percentage configured.

Priority Based Discard Policy

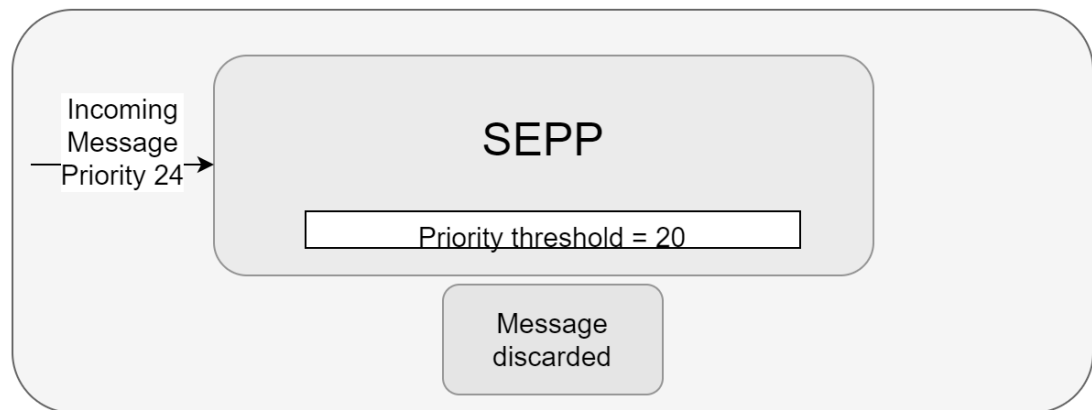
The priority based discard policy is applied on the incoming traffic . When application is overloaded and if the discard scheme is priority based, the messages are discarded on the basis of the priority threshold defined. The priority is determined by the "3gpp-sbi-message-priority" header present in the incoming message.

Higher the value of "3gpp-sbi-message-priority" lower is the priority and vice-versa.

Figure 3-56 Priority Based Discard Policy



In the above diagram, the threshold priority set is 20. The incoming message has 3gpp-sbi-message-priority header value as 10. As it is a high priority message, the message will be processed.

Figure 3-57 Priority Based Discard Policy

In the above diagram, the threshold priority set is 20. The incoming message has 3gpp-sbi-message-priority header value as 24. As it is a low priority message, the message will be discarded.

Managing Overload Control Feature

Enable

This feature is disabled by default.

Note

This feature is supported only on inter PLMN traffic, that is N32 Ingress Gateway and PN32F-svc.

You can enable the Overload Control feature using the Helm and REST API.

- In Helm configuration: In the perf-info section, set `overloadManager.enabled` to true.
-
- Configure the `configmapPerformance.prometheus` in the perf-info section to the Prometheus IP:port or service name. This is required for CPU and memory calculations.

```
configmapPerformance
  prometheus: "http://ocoso-prom-svr.ocsepp:80"
```

- **Note**

The prometheus UI can also have cluster name appended after slash("/"). The configmapPerformance.prometheus should be configured in the following manner:

```
<IP>:<port>/<clusterName>
```

- In REST API: Set svcName to <release-name>-pn32f-svc and set enabled to true in Overload Control REST API. For more information about API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Configure

You can configure the feature using REST API:

- Configure using Helm: Refer the Steps to configure overload control section for configuring using Helm parameters.
- Configure using REST API: Perform the feature configurations as described in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Note

The feature is supported on Inter PLMN, that is, N32 Ingress Gateway only. Only PN32F-svc will be mapped as service.

Configuring Overload Control

1. In the N32 Ingress gateway section of ocsepp_custom_values_<version>.yamlfile, set the parameter enableIncomingHttp to true.
 - a. This is required as https communication is not allowed between the perf-info and the n32-ingress-gateway service.
 - b. Network policy can be applied on the namespace to restrict external http incoming traffic through n32-ingress-gateway.
2. In the perf-info section, modify the parameter configmapPerformance.prometheus according to the prometheus service deployed in the cluster. If the Prometheus UI has cluster-name appended to it, modify the parameter as follows:
http://occne-kube-prom-stack-kube-prometheus.occne-infra:80/<clustername>

Example: http://occne-kube-prom-stack-kube-prometheus.occne-infra:80/cluster1

Note

Perform the Helm upgrade after the above customizations, if OCSEPP has already been deployed.

3. By default, the feature will be disabled. To enable the feature run the Overload Control REST API to change the svcName to `<release-name>-pn32f-svc` and **enabled** to true.

```
curl -X PUT http://<config-server>:port/sepp/nf-common-
component/v1/igw/n32/ocpolicymapping -H 'Content-type: application/json' -
d '
{
  "enabled": true,
  "mappings": [
    {
      "svcName": "ocsepp-release-pn32f-svc",
      "policyName": "Policy2"
    }
  ],
  "samplingPeriod": 5000
}'
```

4. To define CPU, Memory, and pending count threshold limits, run the following code. Change the svcname as done as step 1:

```
curl -X PUT http://<config-server>:port/sepp/nf-common-component/v1/perf-
info/overloadLevelThreshold -H 'Content-type: application/json' -d '
[
  {
    "svcName": "ocsepp-release-pn32f-svc",
    "metricsThresholdList": [
      {
        "metricsName": "memory",
        "levelThresholdList": [
          {
            "level": "Warning",
            "onsetValue": 50,
            "abatementValue": 45
          },
          {
            "level": "Minor",
            "onsetValue": 70,
            "abatementValue": 65
          },
          {
            "level": "Major",
            "onsetValue": 80,
            "abatementValue": 75
          },
          {
            "level": "Critical",
            "onsetValue": 90,
            "abatementValue": 85
          }
        ]
      },
      {
        "metricsName": "cpu",
        "levelThresholdList": [
          {
```

```

        "level": "Warning",
        "onsetValue": 65,
        "abatementValue": 60
      },
      {
        "level": "Minor",
        "onsetValue": 75,
        "abatementValue": 70
      },
      {
        "level": "Major",
        "onsetValue": 85,
        "abatementValue": 80
      },
      {
        "level": "Critical",
        "onsetValue": 95,
        "abatementValue": 90
      }
    ]
  },
  {
    "metricsName": "svc_pending_count",
    "levelThresholdList": [
      {
        "level": "Warning",
        "onsetValue": 100,
        "abatementValue": 60
      },
      {
        "level": "Minor",
        "onsetValue": 200,
        "abatementValue": 125
      },
      {
        "level": "Major",
        "onsetValue": 260,
        "abatementValue": 201
      },
      {
        "level": "Critical",
        "onsetValue": 400,
        "abatementValue": 300
      }
    ]
  }
]

```

Note

The above mentioned values are the recommended values for each threshold level. User can configure them as required.

5. By default the discard policy is Percentage Based. To change the discard policy to Priority based, run the following code with svcName same as in step1:

```
curl -X PUT http://<config-server>:port/sepp/nf-common-
component/v1/igw/n32/ocpolicymapping -H 'Content-type: application/json' -
-d '
{
  "enabled": true,
  "mappings": [
    {
      "svcName": "ocsepp-release-pn32f-svc",
      "policyName": "Policy1"
    }
  ],
  "samplingPeriod": 5000
}'

#Policy1 is PriorityBased
```

6. By default the error code configured is 429. To change the error code, change the errorCode and run the following code:

```
curl -X PUT http://<config-server>:port/sepp/nf-common-
component/v1/igw/n32/errorcodeprofiles -H 'Content-type: application/json'
-d '
[
  {
    "name": "error429",
    "errorCode": 503,
    "errorCause": "",
    "errorTitle": "",
    "errorDescription": ""
  }
]
```

7. To get the details of discard policy defined, run the following command:

```
curl -X GET http://<config-server>:port/sepp/nf-common-
component/v1/igw/n32/ocdiscardpolicies
```

Example:
DiscardPolicies.json:

```
[{
  "name": "Policy1",
  "scheme": "PriorityBased",
  "policies": [{
    "level": "Warning",
    "value": 30,
    "action": "RejectWithErrorCode",
    "errorCodeProfile": "error429"
  },
  {
    "level": "Minor",
    "value": 24,
```

```
"action": "RejectWithErrorCode",
"errorCodeProfile": "error429"
},
{
"level": "Major",
"value": 15,
"action": "RejectWithErrorCode",
"errorCodeProfile": "error429"
},
{
"level": "Critical",
"value": 10,
"action": "RejectWithErrorCode",
"errorCodeProfile": "error429"
}
]
},
{
"name": "Policy2",
"scheme": "PercentageBased",
"policies": [{
"level": "Warning",
"value": 0,
"action": "RejectWithErrorCode",
"errorCodeProfile": "error429"
},
{
"level": "Minor",
"value": 5,
"action": "RejectWithErrorCode",
"errorCodeProfile": "error429"
},
{
"level": "Major",
"value": 10,
"action": "RejectWithErrorCode",
"errorCodeProfile": "error429"
},
{
"level": "Critical",
"value": 25,
"action": "RejectWithErrorCode",
"errorCodeProfile": "error429"
}
]
}
]
```


Note

- The above are recommended values. The user can change them as per their preference.
- **value** for Percentage Based policy indicates the percentage of incoming traffic that will be discarded.
- **value** for Priority Based policy indicates the threshold priority value, equal to or higher which the incoming traffic will be discarded.

8. To change the discard policies, run the following command with your changes in DiscardPolicies.json:

```
curl -X PUT http://<config-server>:port/sepp/nf-common-component/v1/igw/n32/ocdiscardpolicies -H 'Content-type: application/json' -d @DiscardPolicies.json
```

Observe

Following are the overload control feature specific metrics:

- oc_ingressgateway_route_overloadcontrol_discard_total
- service_resource_overload_level
- ingressgateway_svc_pending_count

Following are the overload control feature specific KPIs:

- PercentageDiscard
- PriorityDiscard

For more information on Overload Control feature related SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.32 Alternate Routing Across Remote SEPPs

In earlier SEPP releases, Consumer SEPP could communicate with single Producer SEPP and if this single Producer SEPP was not available for any reason, n32f traffic would drop and not reach its home PLMN.

The SEPP cannot route to an alternate SEPP when the primary path is down. The reasons for the primary path being down can vary from routing path having issues, to the actual P-SEPP not being up and running. Some of the reasons can be as follows:

- Remote SEPP or NF is not reachable due to network issue
- Remote SEPP or NF unavailable
- There is an internal server error
- Bad request
- FQDN/IP or port is incorrect
- HTTP2 Error codes, 400, 404, 500, 503, and 504

With Alternate Routing Across Remote SEPPs feature, the consumer SEPP routes the n32f traffic messages based on the availability of Primary or Secondary or Tertiary producer SEPPs present in a Remote SEPP Set (sets will be uniquely identified based on the home PLMN).

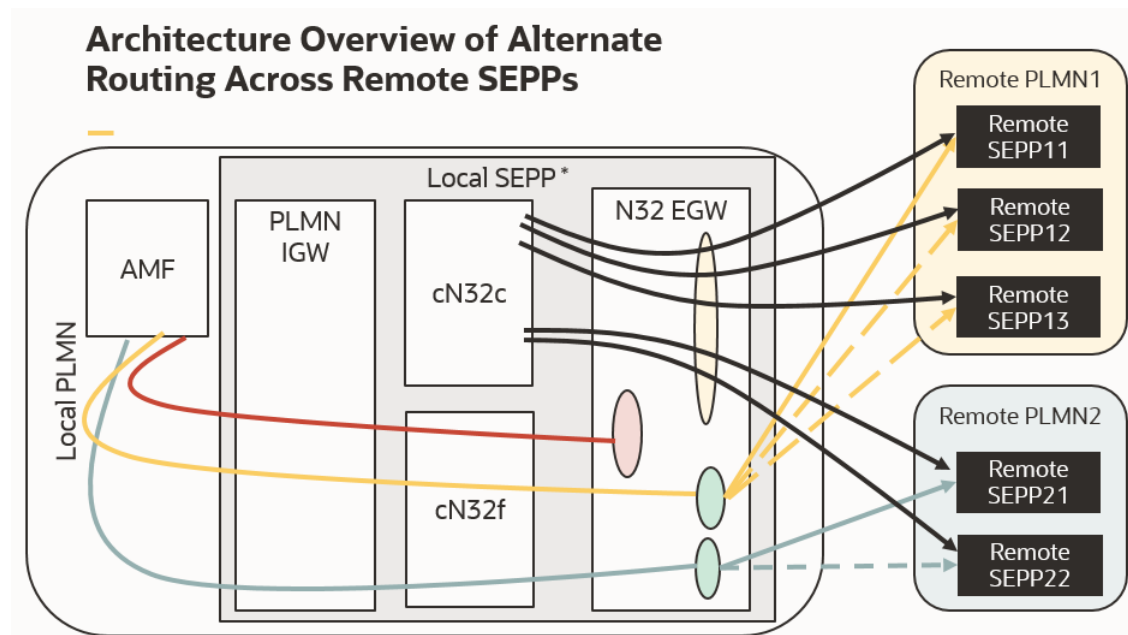
Feature Overview

The purpose of implementing this feature is to support and provide a reliable path always between two PLMNs through the SEPPs.

Design and Architecture

In this feature, the consumer SEPP end routes the n32f traffic messages based on the availability of Primary or Secondary or Tertiary producer SEPPs present in a set. The following diagram represents the detailed architecture of the Alternate Routing Across Remote SEPPs.

Figure 3-58 Architecture



- The remote SEPP sets are created per PLMN. In this diagram there are two remote sets, remote SEPP set for Home PLMN1 and remote SEPP set for Home PLMN2.
- These Remote SEPP sets can vary from 1 to N (unique for each PLMN).
- Each set can have up to three producer SEPP's based on their priority as primary, secondary, or tertiary.
- N32 EGW routes each n32f traffic message to the producer based on the priority assigned in the Remote SEPP.

- N32 EGW first routes each n32f traffic message towards primary SEPP, if it is not available it then routes to secondary SEPP, and similarly towards the tertiary SEPP.
- The call will be dropped only if primary, secondary, and tertiary SEPPs are not available. This increases the reliability of the calls.

The following configuration is mandatory and needs to be done through CNC console or REST APIs.

Managing Alternate Routing Across Remote SEPPs

Enable

This feature is the core functionality of SEPP. You do not need to enable or disable this feature. It is enabled by default.

Configure

You can configure the Alternate Routing Across Remote SEPPs using REST API, and CNC Console.

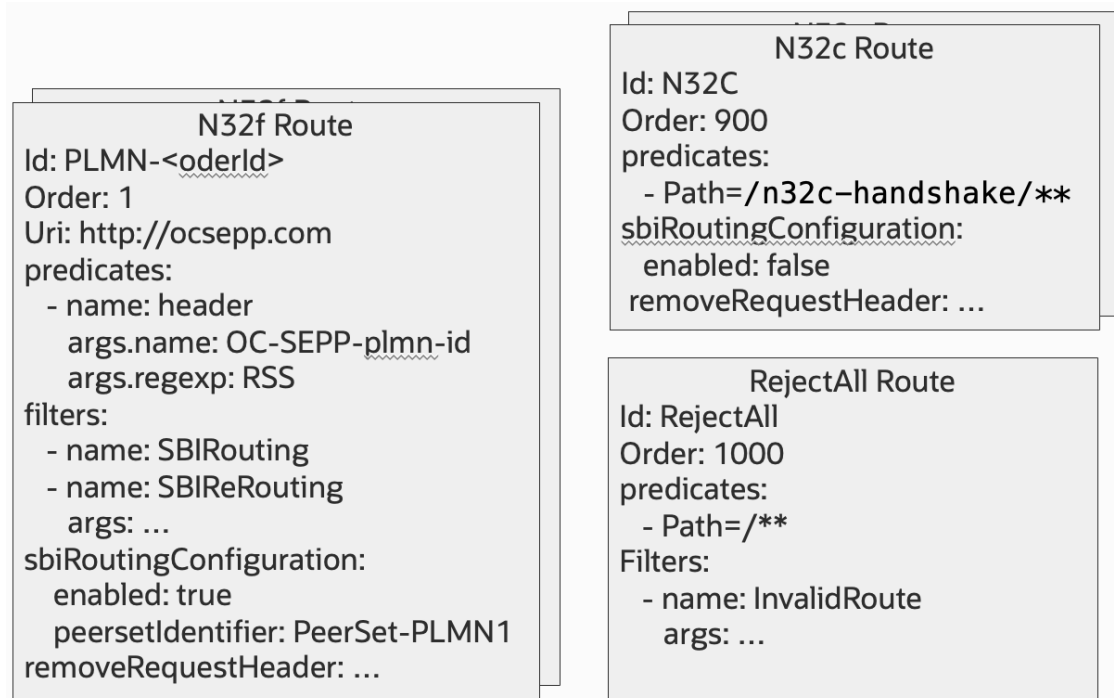
- Configure Alternate Routing Across Remote SEPPs using REST API: Perform the feature configurations as described in *Remote SEPP Set* section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Default Routes

When SEPP is installed, the following three default routes are created on n32-egress-gateway. These routes are for internal use only. They aren't available in the custom values, and users don't need to modify them.

- **N32c** with path set to `/n32c-handshake/**`: This route helps in the routing of n32c handshake exchange-capability message without apiPrefix.
- **N32d** with path set to `/*/n32c-handshake/**`: This route helps in the routing of n32c handshake exchange-capability message with apiPrefix.
- **ocseppRejectAll** (default route): This is the default route. If any message except Handshake and n32f arrives, it is dropped at Egress Gateway and is not routed towards the Producer SEPP.

The following diagram represents the default routes:

Figure 3-59 Default Routes

Example of the default route created automatically on n32-egress-gateway:

```
routesconfiguration: [{"id": "ocseppRejectAll", "uri": "https://ocsepp.com",
"order": 903, "filters": [{"args": {"errorCodeOnInvalidRoute": "500",
"errorCauseOnInvalidRoute": "No Matching Route.", "errorTitleOnInvalidRoute":
"No Matching Route.", "errorDescriptionOnInvalidRoute": "No Matching
Route."}, "name": "InvalidRouteFilter"}], "predicates": [{"args": {"pattern":
""}, "name": "Path"}]}, {"id": "n32c", "uri": "https://ocsepp.com", "order":
901, "predicates": [{"args": {"pattern": "/n32c-handshake/**"}, "name":
"Path"}]}, {"id": "n32d", "uri": "https://ocsepp.com", "order": 902,
"predicates": [{"args": {"pattern": "/*n32c-handshake/**"}, "name":
"Path"}]}],
```

Routes for SBI Message (n32f interface)

- Remote SEPP is added at initiator and responder side. Once security capability procedure is successful, Peer is added at n32-egress-gateway.
- Remote SEPP Set is added to SEPP having information about primary, secondary, and tertiary SEPPs. Based on the Remote SEPP Set information, n32-egress-gateway also configures the **Peer Set** at its end, and tries to route the N32f messages based on the priority configured in the Peer Set.
- N32f route helps in routing of n32f traffic messages towards producer SEPP based on the PLMN of the Producer SEPP.

The user configurable Helm parameters and can be configured through ocsepp/charts/config-mgr-svc/values.yaml file.

For more information on the Helm parameters, see the *config-mgr-svc* section of *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Example for n32f routes automatically created on n32-egress-gateway:

```
{
  "id": "plmn-357",
  "uri": "https://ocsepp.com",
  "order": 357,
  "filters": [
    {
      "args": {
        "errorHandling": [
          {
            "priority": 2,
            "actionSet": "action_0",
            "actionSet": "criteria_1",
            {
              "priority": 1,
              "actionSet": "action_0",
              "errorCriteriaSet": "criteria_0"
            }
          ],
        "peerSetIdentifier": "RS-1",
        "customPeerSelectorEnabled": true,
        "name": "SbiRouting",
        {
          "args": {
            "name": "OC-SEPP-PLMN-ID",
            "name": "RemoveRequestHeader"
          },
          "metadata": {
            "httpRuriOnly": false,
            "httpsTargetOnly": false,
            "sbiRoutingEnabled": true
          },
          "predicates": [
            {
              "args": {
                "header": "OC-SEPP-PLMN-ID",
                "regex": "RS-1"
              },
              "name": "Header"
            }
          ],
          "forwardheaderdetails": {
            "enabled": false,
            "forwardHeaderValue": ""
          }
        }
      ]
    }
  ]
}
```

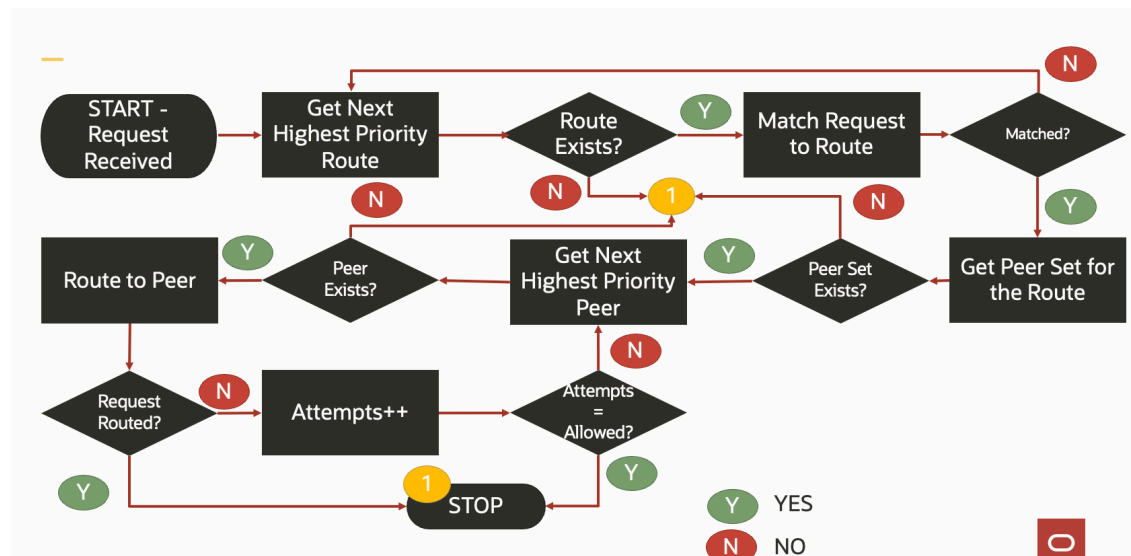
Alternate Routing Flow

When the SBI message reached n32-egress-gateway, the relevant route is selected. The traffic is routed towards the primary peer mentioned in the peerset associated with the route. On receiving the response from primary peer:

- If the response is successful, message is routed towards Remote SEPP.
- If the response is an error response, the response is validated against the errorCriteriaSet mentioned in the route. If the error received matched the criteria mentioned in the criteria set, action is performed on the basis of actionSet.

The following diagram represents the Alternate Routing flow:

Figure 3-60 Alternate Routing Flow Chart Representation



Response Validation against the Error Handling

The error handling section within the SbiRouting filter is an array of mapping between criteriaset and actionset, tagged with a priority. It is similar to peer and peerset configurations tagging. The criteriaset and actionset ids are tagged in the error handling section along with a

priority for each mapping. The actual criteriaset and actionset configuration has to be done in `sbiRoutingErrorCriteriaSets` and `sbiRoutingErrorActionSets` respectively.

sbiRoutingErrorCriteriaSets

It is a combination of HTTP method and HTTP status code or HTTP method and exception. When an error response is received, HTTP status code and HTTP method or exception is received in the response is validated with the configured values.

- If the match is found, action is taken on the basis of erroractionset configuration.
- If the match is not found, error response is sent to the originating NF.

Example of `SbiRoutingErrorCriteriaSets` automatically created through Helm configuration on `n32-egress-gateway`:

```
"sbiroutingerrorcriteriasets": [{"id": "criteria_1", "method": ["GET",
"POST", "PUT", "DELETE", "PATCH"], "response": {"statuses": [{"status": [400,
404], "statusSeries": "4xx"}, {"status": [500, 503, 504], "statusSeries":
"5xx"}]}}, {"id": "criteria_0", "method": ["GET", "POST", "PUT", "DELETE",
"PATCH"], "exceptions": ["java.util.concurrent.TimeoutException",
"java.net.SocketException", "java.net.SocketTimeoutException",
"java.net.UnknownHostException", "java.net.ConnectException",
"java.net.NoRouteToHostException"]}]}
```

SbiRoutingErrorActionSets

After the criteriaset conditions is validated and met, the corresponding actionset is selected to decide what will be the next course of action.

- **action:** SEPP currently supports only reroute action which means try secondary or tertiary peer when the error is received from primary or secondary peer.
- **attempts:** Maximum no of retries to either same or different peer in case of errors or failures from received backend.
 - The action is reroute and no of reroute attempts is 2. Hence, the request would be rerouted to the secondary peer of peerset and then the same steps would be repeated for tertiary once the error response from secondary peer is recieved.
 - If only primary peer is configured and if an error is received, three retries are attempted on the primary peer.
 - If the primary and secondary peers are configured and if an error is received, the retries are attempted on the secondary peer.

Note

To enable alternate routing feature, change `alternateRoute.sbiReRoute.sbiRoutingErrorActionSets[].Attempts` parameter value to 2. This will allow switching from primary to secondary to tertiary remote SEPPs.

- **blacklist:** At present, this is not used on `n32-egress-gateway`.

Example of SbiRoutingErrorActionSets automatically created through Helm configuration on n32-egress-gateway:

```
"sbiroutingerroractionsets": [{"id": "action_0", "action": "reroute",
"attempts": 2, "blacklist": {"enabled": false, "duration": 60000}}]
```

Observe

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.33 Server Header Support

SEPP supports server header in the error responses generated for SEPP services. Using the server header, the user can find out if the error is generated either by the SEPP or some other NF. As per 3gpp specs, format of server-header is SEPP-<fqdn>.

If an error condition occurred at SEPP, SEPP services adds the server header in the error response message. This is added to inform the user that error is generated by SEPP. All the intra PLMN traffic in SEPP is received by plmn-ingress-gateway and inter PLMN traffic is handled by n32-ingress-gateway. If any of the ingress-gateway in SEPP is not reachable or in error state, error response would not contain server-header parameter.

Gateways also support adding server-header for which user has to enable server-header through helm configuration.

Error scenario examples: n32f-context not found, 3gpp-header is not in correct format, Ingress Gateway is down.

Helm Configuration

Configure using Helm: To configure server header at Egress Gateway, you need to perform the helm configurations:

```
# All attributes under "serverHeaderDetails" will need to be configured only
if "serverHeaderConfigMode" is set as "HELM"
serverHeaderDetails:
  # If enabled at Global level, Global conf will be used by default if no
  conf exists at Route level.
  enabled: true
  # Mandatory attribute if not defined at Route level. By default used for
  Global level conf. Value need to be one among "errorCodeSeriesList" resource
  defined below.
  errorCodeSeriesId: E1
  configuration:
    # Mandatory attribute. This value is common across Global and Route
```

```

level conf. If not defined, server header will not be included in response.
  nfType: SEPP
  # Optional attribute. This value is common across Global and Route
level conf. If not defined, only "nfType" will be used for server header
value.
  nfInstanceId: 9faf1bbc-6e4a-4454-a507-aef01a101a06

# Use below configuration to define errorCodeSeries list
errorCodeSeriesList:
  # Value of "id" attribute will need to used for assigning
"errorCodeSeriesId" either at Global or Route level conf for Server header.
- id: E1
  errorCodeSeries:
    # Possible values for "errorSet" attribute: 5xx, 4xx, 3xx, 2xx, 1xx
    - errorSet: 4xx
    # Possible values include all error codes in the respective
HttpSeries(Ex: 4xx) value assigned for "errorSet". Use single value of -1 if
all error codes are to be considered.
    errorCodes:
      - 400
      - 408
      - 404
    - errorSet: 5xx
    errorCodes:
      - 500
      - 503
      - 504
- id: E2
  errorCodeSeries:
    - errorSet: 4xx
    errorCodes:
      - -1

```

① Note

Egress Gateway won't add the server header in self-generated error responses.
 Example: The server header won't be added if the error is generated due to DNS SRV records not present in the DNS.

3.2.34 Support for Networks with/without 3GPP-SBI-Target-API-Root Header

There are some networks which may not be supporting the Rel 16 header 3GPP-SBI-Target-APIroot header. In such scenarios, SEPP being deployed at the edge of the network is required to make the routing possible in the core network. SEPP modifies the respective headers such that routing doesn't fail in the core network.

Note

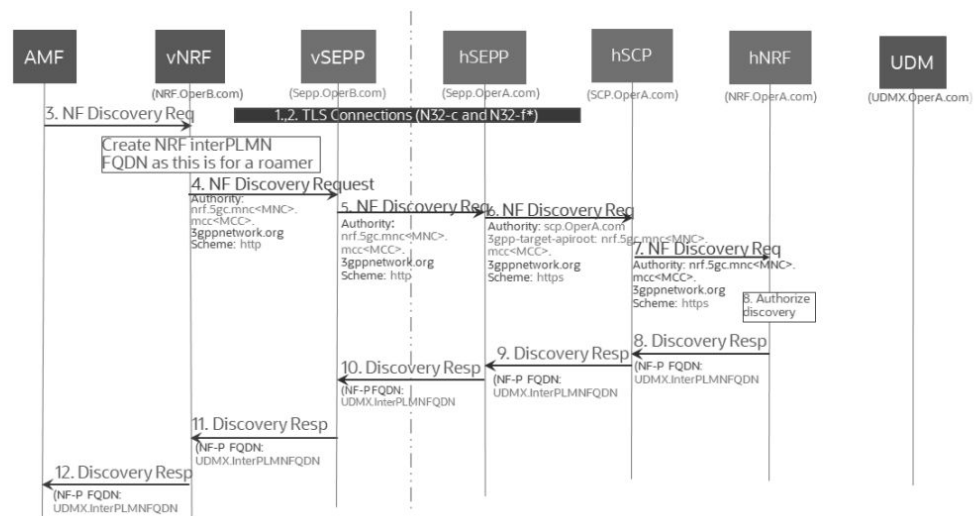
When an incoming message doesn't have the 3gpp-sbi-target-apiroot header but the PLMN network needs it, SEPP uses the information in the authority header to populate the 3gpp-sbi-target-apiroot header.

There are two possible scenarios:

1. Network that is sending in the request does not use the 3GPP-SBI-Target-Apiroot header, however the core Network behind the SEPP will require this header. In this case the SEPP sets the 3GPP-SBI-Target-Apiroot header to the intended producer by using the value presented to it in the authority header.

The following diagram represents this scenario:

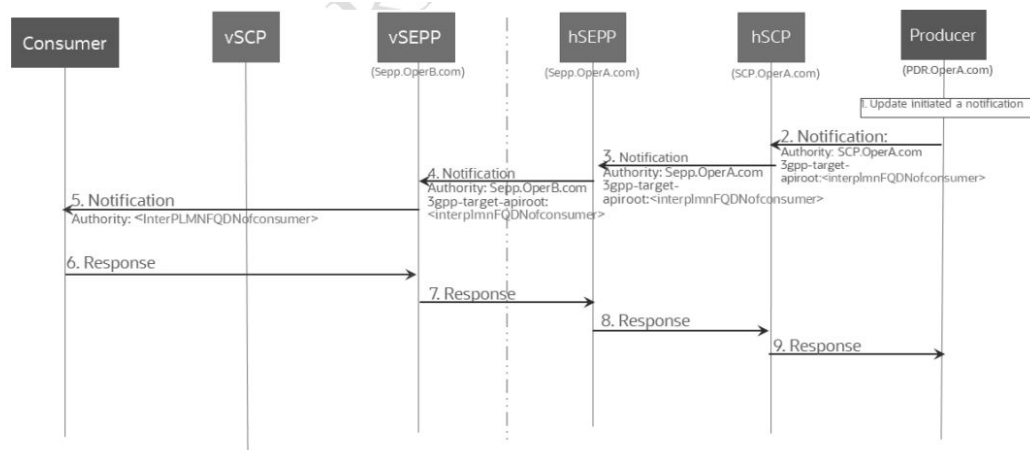
Figure 3-61 SEPP sets the 3GPP-SBI-Target-Apiroot header



2. Network that is sending in the request uses the 3GPP-SBI-Target-Apiroot header, however the core network behind the SEPP will not be able to understand this header.

In this case the SEPP removes the 3GPP-SBI-Target-Apiroot header and set the intended producer in the authority header.

The following diagram represents this scenario:

Figure 3-62 SEPP removes the 3GPP-SBI-Target-Apiroot header

Managing Support for networks with/without 3GPP-SBI-Target-API-Root header

Enable

This feature is enabled by default. This is on the basis of query consumer NF is sending to producer NF.

Configure

You can configure the Support for networks with/without 3GPP-SBI-Target-API-Root header using helm parameters:

Configuring using Helm parameters

```
sbiRouting:
# Default scheme applicable when 3gpp-sbi-target-apiroot header is missing
sbiRoutingDefaultScheme: https
```

In pn32f svc ->

```
configs:
is3gppSbiTargetApiRootSchemeHttp: true
```

Observe

For more information on Support for networks with/without 3GPP-SBI-Target-API-Root header feature metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.

2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.35 Subject Alternate Name(SAN) Validation

SAN (Subject Alternate Names) validation validates whether the SAN in the SEPP certificate matches with the FQDN in the Remote SEPP.

In earlier releases, the SAN Validation is a mandatory configuration for the regular SEPP deployments.

With this enhancement, the SAN validation is made optional. By making SAN validation option, user get the flexibility to bypass this check.

The following procedures can be used to make it optional :

- `sanValidationRequired` is set to false in the roaming partner configuration. This disables san validation for the specific roaming partner.
- `sanValidationRequired` is set to false in helm configuration. This disables san validation for all the Remote SEPPs.

Managing Subject Alternate Name(SAN) Validation Feature

Enable

Subject Alternate Name(SAN) Validation Feature is enabled by default. You can disable Subject Alternate Name(SAN) Validation feature using the helm parameters and REST API or CNC Console .

- To disable the feature, set `sanValidationRequired` to False in the `ocsepp-custom-values-<version>.yaml` file.
- Disable using REST API: Set `sanValidationRequired` to False in Remote SEPP. For more information about API path, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy REST Specification Guide*.
- Disable using CNC Console: Set `sanValidationRequired` to False on the Remote SEPP page. For more information about enabling the feature using CNC Console, see [Configuring SEPP using CNC Console](#).
- **SAN Validation Enable/Disable Table**

Value of <code>sanValidationRequired</code> in HELM	Value of <code>sanValidationRequired</code> in REST API/ CNC Console (Remote SEPP)	SAN Validation of incoming N32C handshake request
TRUE	TRUE	SAN Validation occurs
TRUE	FALSE	No SAN Validation
FALSE	TRUE	No SAN Validation
FALSE	FALSE	No SAN Validation

Observe

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.36 SEPP Message Feed

Overview

SEPP supports copying of the incoming and outgoing messages passing through Ingress and Egress Gateways to a Data Director. Data Director receives the messages from Gateways with a correlation-id and feeds the data securely to an external monitoring system.

The correlation-id in the messages is used as a unique identifier for every transaction. If the request does not have the correlation-id, the Gateway generates the correlation-id and then passes it to Data Director.

Upon receiving an Ingress Gateway request containing a correlation-Id, the SEPP microservices include this correlation-Id in upcoming related Egress Gateway requests that may be generated by the SEPP based on the Ingress Gateway transaction.

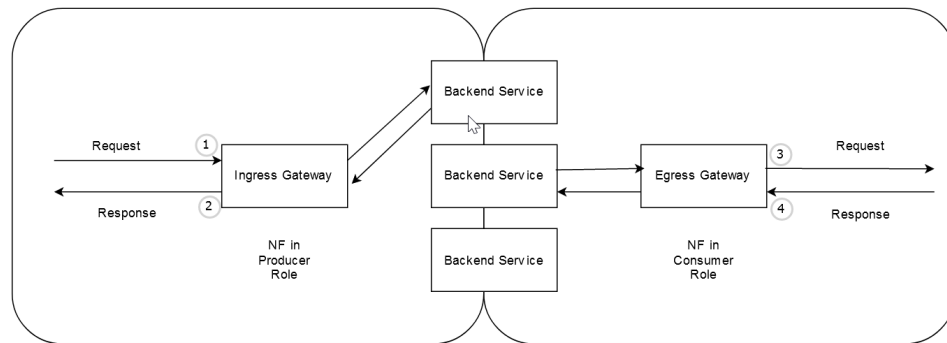
From Gateway Services 25.1.100, Gateway Services is enhanced to capture all messages within a transaction on to the same Kafka partition based on the keybasedKafkaProducer Helm configuration. If keybasedKafkaProducer is set to true, correlation-id metadata is used as the message key for the key based Kafka producer. This ensures that the same transaction messages are routed to the same partition. If keybasedKafkaProducer is set to false, all the messages are distributed across all the available partitions in a round robin order.

Note

This feature is not applicable for ASM deployments.

Detailed Description

In order to capture incoming and outgoing traffic, there are four tapping points for Ingress Gateway and Egress Gateway.

Figure 3-63 Overview of Message Feed**Ingress Gateway Tapping point**

1. As soon as the request arrives at Ingress Gateway, but before passing through the chain of filters for post-processing.
2. When the response is about to be sent by Ingress Gateway to the consumer NF, after all the post-processing is done by the chain of filters on the response.

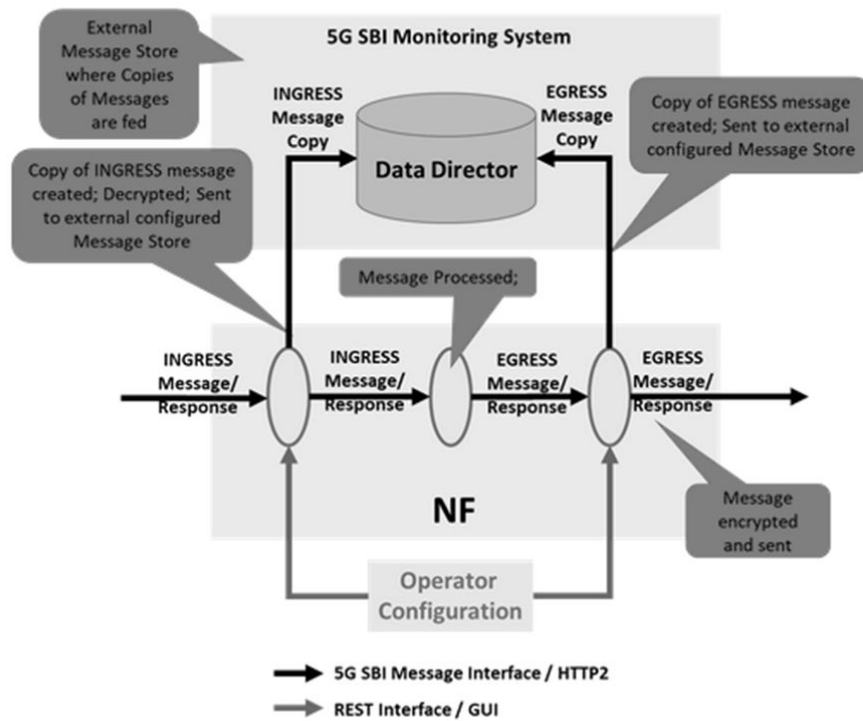
Egress Gateway Tapping point

3. After the request is received at Egress Gateway and has undergone preprocessing and before the request is sent out of Egress Gateway to the producer NF.
4. As soon as the response is received at Egress Gateway and before passing it through the chain of filters for post-processing.

In above diagram, 1, 2, 3, and 4 indicate the tapping points at which gateways copy the messages. The message copy can be configured and enabled/disabled in pairs for points (1, 2) and points (3, 4).

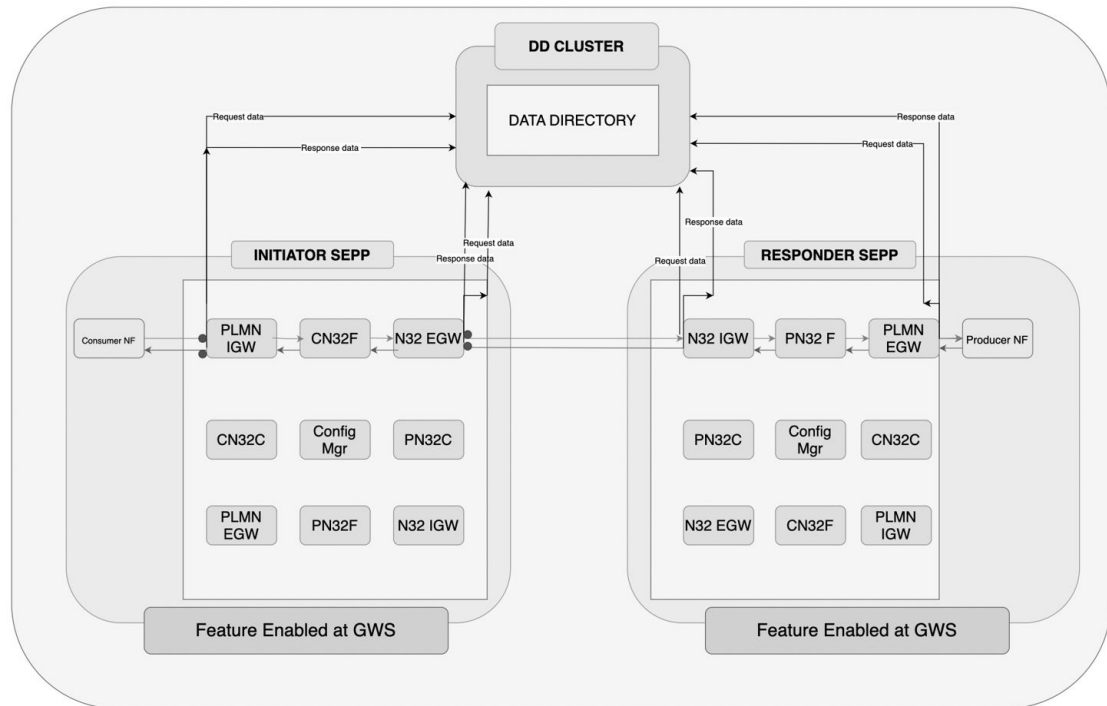
The following diagram represents the SEPP representation of Message feed feature:

Figure 3-64 SEPP representation of Message Feed feature



Detailed Description

The following diagram indicates the detailed view of message feed feature in SEPP:

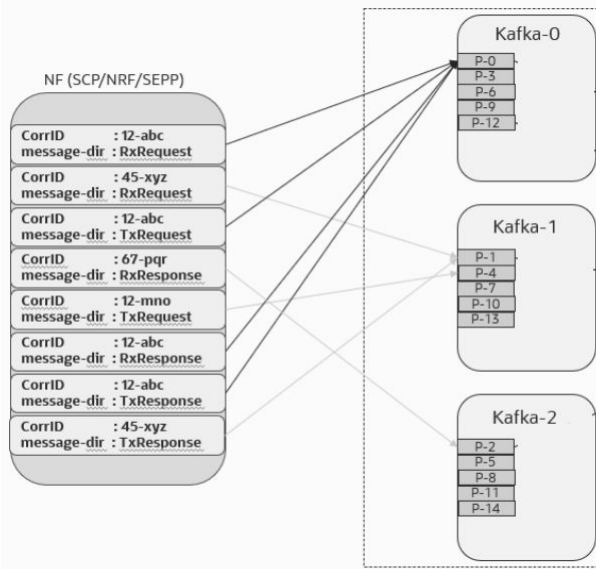
Figure 3-65 Detailed view of message feed feature in SEPP

- When SEPP receives an incoming message at the Ingress Gateway, before processing the message, the data sends to the DD.
- After processing the request at DD, the request is sent towards the SEPP microservices which forward to the Egress Gateway.
- After the message has been processed at the Egress Gateway, the data is again sent to DD before being sent to the consumer NF.
- Similarly, when a response is received from end consumer NF, it is received at the Egress Gateway.
- When the Egress Gateway receives the response, it sends the message to DD, which processes it and sends to the SEPP microservice.
- The response, when received at Ingress Gateway is processed and then sent towards DD before before it is sent to the consumer NF.
- The messages received at DD are correlated against the correlation-id parameter that gateways add.
- The correlation-id parameter is either extracted from the "x-request-id" header present in the incoming request/response or is autogenerated by gateways, if the header is not present.

When an incoming request or response is received at the gateway, the header details and body of the message are copied to the DD. Along with these details, additional details about the message (Example: timestamp) are also copied as part of the metadata list.

The following diagram represents the enhancement of selecting the same Kafka partition for the messages of the same transaction type:

Figure 3-66 Same Kafka partition for the messages of the same transaction type



The following are the descriptions of the attributes that are copied to the DD (Data Director) as part of metadata:

Table 3-49 Attributes Copied to DD

Attribute name	Data type	Description	Source of data
correlation-id	String	This is a mandatory parameter. This is a unique identifier for the message (both request and response). Below are the possibilities to populate this field in the metadata: 1. If the x-request-id header is present in the message, then the correlation id shall be extracted from this field. 2. If this header is not present, then the below processing shall be done: a. At IGW: As soon as the message is received and the x-request-id header is not present, then IGW shall generate a UUID, and set correlation-id to this UUID. Copy the message towards DD with correlation id in metadata, but no x-request-id header. After copying the message to DD, this custom header shall be added by IGW to the incoming request. For response coming from the backend, IGW shall again pick the same UUID as set in the request from its context and set in the metadata. For response, it does not matter if the x-request-id header is present or not, as IGW already has it in its request context which is re-used while processing the corresponding response. b. At EGW: After receiving a request from the backend, if the x-request-id header is not present, EGW shall first generate a UUID and set correlation-id metadata to this UUID, and also add the x-request-id header to the	x-request-id header or autogenerate

Table 3-49 (Cont.) Attributes Copied to DD

Attribute name	Data type	Description	Source of data
		request message. Then it should copy the message to DD and then send it out on the wire. In the response received from outside NF, EGW shall reuse the UUID it generated during the request and set it to correlation-id of response metadata. EGW need not check or add the x-request-id header in the response.	
consumer-id	String	This is an optional Parameter. 5G nf Instance id of the NF originated the received message For IGW, this should be peer-nf-id For EGW, this should be self-nf-id For Notifications at EGW, this should be peer-nf-id	peer-nf-id for IGW is the NF-Instance ID extracted from the user-agent header if available in the request. User-Agent: <NF Type>-<nf Instance id> <NF FQDN> self-nf-id will be extracted from Helm configurations (nfInstanceId).
producer-id	String	This is an optional Parameter. 5G nf Instance id of the next destination NF	self-nf-id will be extracted from Helm configurations (nfInstanceId). Note: EGW doesnt populate producer-id.
consumer-fqdn	String	This is an optional parameter. It is the FQDN of the orginating NF. For IGW, this should be peer-nf-fqdn. For EGW, this should be self-nf-fqdn. For Notifications at EGW, this should be peer-nf-fqdn.	peer-nf-fqdn for IGW is the NF-FQDN extracted from user-agent header. User-Agent: <NF Type>-<nf Instance id> <NF FQDN> self-nf-fqdn will be extracted from Helm configurations (nfFqdn). peer-nf-fqdn for EGW is extracted from 3gpp-sbi-target-apiroot header.
producer-fqdn	String	This is an optional parameter. FQDN of the destination NF For EGW, this should be peer-nf-fqdn For IGW, this should be self-nf-fqdn For Notifications at EGW, this should be self-nf-fqdn	peer-nf-fqdn is extracted from 3gpp-sbi-target-apiroot header. self-nf-fqdn will be extracted from Helm configurations (nfFqdn).
hop-by-hop-id	String	This is an optional parameter. This should be set if it is present in the request/response headers.	hop-by-hop-id header (this header is added by SCP).

Table 3-49 (Cont.) Attributes Copied to DD

Attribute name	Data type	Description	Source of data
source-ip	String	This is an optional parameter. The origination IP address of the message.	NA
destination-ip	String	This is an optional parameter. The destination IP address of the message.	NA
source-port	String	This is an optional parameter. The port that receives the message or response from the destination. TCP port of the NF on the origination IP address.	NA
destination-port	String	This is an optional parameter. The port that delivers the message or response to the source. TCP port of the NF on the destination IP address.	NA
timestamp	String	This is a mandatory parameter. Identifies the timestamp when the message arrives at the producer. The timestamp format would be long and in nanoseconds.	The time at which the message is created with nanoseconds precision. Example: 1656499195571109210 nanoseconds
message-direction	String	This is a mandatory parameter. The direction of the message can be either incoming or outgoing.	Parameter to indicate whether a message is an ingress to or egress from NF. It can be indicated by putting the traffic feed trigger point name. • RxRequest (Ingress Request) • TxRequest (Egress Request) • RxResponse (Egress Response) • TxResponse (Ingress Response)
feed-source	FeedSource	This is a mandatory parameter. Information about the NF details. Note: The details are given in the following table.	The details are given in the following table.

FeedSource

Table 3-50 FeedSource

Attribute name	Data type	Description	Source of data
nf-type	String	This is a mandatory parameter. Identifies a type of producer NF.	The information from the Helm configuration from the parameter <code>global.nfTypeMsgCpy</code> . If the NFs haven't configured this, then unknown would be attached as a value.
nf-instance-id	String	This is an optional Parameter. Identifies a producer NF Instance.	The information from Helm configuration from the parameter <code>global.nfTypeMsgCpy</code> .
nf-fqdn	String	This is an optional Parameter. Identifies a producer NF FQDN.	The information from Helm configuration from the parameter <code>global.nfFqdn</code> .
pod-instance-id	String	Identifies a producer NF's pod. This is the unique identifier or name assigned to the pod. This is an optional parameter.	NA

Managing SEPP Message Feed Feature**Enable**

You can enable SEPP Message Feed feature using the following Helm configuration:

1. Open the `ocsepp_custom_values_<version>.yaml` file.
2. To enable feature on PLMN Ingress Gateway, in the PLMN Ingress Gateway section, set `messageCopy.enabled` to true.
3. To enable feature on N32 Ingress Gateway, in the N32 Ingress Gateway section, set `messageCopy.enabled` to true.
4. To enable feature on PLMN Egress Gateway, in the PLMN egress Gateway section, set `messageCopy.enabled` to true.
5. To enable feature on N32 Egress Gateway, in the N32 Egress Gateway section, set `messageCopy.enabled` to true.
6. To enable copy message payload as well, in the sections mentioned in steps 2-4, set `messageCopy.copyPayload` to true.
7. If you want to disable security protocol settings on messageCopy set `messageCopy.security.enabled` to false.
8. set `Kafka.bootstrapAddress` to DD service and port, where port will be 9092 (security disabled).
9. set `keybasedKafkaProducer` to true on all or required gateway if we want to enable the functionality of message copy on same partition.
10. Skip to step 12 if security is disabled.

11. If `messageCopy.security.enabled` is true, do the following steps:
 - a. Ensure DD is deployed prior to deploying SEPP.
 - b. Follow the NF Integration Steps for generating Gateway secrets.
 - i. Copy the certificate from `ssl_certs/demoCA/cacert.pem` (BEGIN CERTIFICATE to END CERTIFICATE).
 - ii. Copy the sepp-certificates in a temporary new folder.
 - iii. Edit the existing `caroot.cer` to append the `cacert.pem` in the following manner:

```

-----BEGIN CERTIFICATE-----
<existing caroot-certificate content>
-----END CERTIFICATE-----
-----
-----BEGIN CERTIFICATE-----
<DD caroot-certificate content>
-----END CERTIFICATE-----

```

- iv. Create a new file `sasl.txt` in the `seppcertificates` folder and run the following:

```
echo (JAAS secret password) > sasl.txt
```

- v. Run the following command to generate the Gateway secret:

```
kubectl create secret generic ocsepp-plmn-secret --from-
file=ecdsa_private_key.pem --from-file=rsa_private_key_pkcs1.pem --
from-file=trust.txt --from-file=key.txt --from-file=sasl.txt --from-
file=caroot.cer --from-file=rsa_certificate.crt --from-
file=ecdsa_certificate_pkcs1.crt --from-file=ocsepp.cer -n
<namespace>
```

```
kubectl create secret generic ocsepp-n32-secret --from-
file=ecdsa_private_key.pem --from-file=rsa_private_key_pkcs1.pem --
from-file=trust.txt --from-file=key.txt --from-file=sasl.txt --from-
file=caroot.cer --from-file=rsa_certificate.crt --from-
file=ecdsa_certificate_pkcs1.crt --from-file=ocsepp.cer -n
<namespace>
```

- c. Set `security.saslConfiguration.username` to username used to configure DD.
 - d. Set the `security.saslConfiguration.password.k8NameSpace` where DD is deployed.
 - e. set the parameter `security.protocol` to `SSL` or `SASL_SSL`.
 - f. Set the parameter `Kafka.bootstrapAddress` to DD service and port which have port number 9093(security enabled, SSL mode) or 9094 (security enabled SASL_SSL mode).
12. Save the file.
 13. Run Helm install. For more information about installation procedure, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Note

Currently SEPP does not support dynamic reloading of certificates for Kafka.

If you are enabling this parameter after SEPP deployment, run *helm upgrade*. For more information about the upgrade procedure, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Configure

You can configure the feature using Helm:

Configure the following SEPP parameters to configure nfType, nfFQDN, and nfInstanceId.

You can configure the feature using Helm:

Configure the following SEPP parameters to configure nfType, nfFQDN, and nfInstanceId.

ocsepp_custom_values_<version>.yaml file.

```
global:
  nfType:SEPP
  nfFQDN: <FQDN address>
  nfInstanceId: < nf Instance id>
```

Sample Ingress Gateway Configuration:

```
messageCopy:
  enabled: false
  copyPayload: false
  topicName: message.copy
  ackRequired: false
  retryOnFailure: 0
  retryBackoffTime: 100 #in milliseconds
  requestTimeout: 100 #in milliseconds
  keybasedKafkaProducer: false
  security:
    enabled: true
    protocol: SASL_SSL
    tlsVersion: TLSv1.3
    saslConfiguration:
      userName: test
      password:
        k8SecretName: message-copy-secret #relevant n32 or plmn secret
        k8Namespace: ocegress
        fileName: password.txt
  threadPoolConfigurations:
    coreSize: 16
    maxSize: 16
    queueCapacity: 4200
#kafka-details
kafka:
  bootstrapAddress: 10.75.175.246:30608
```

Sample Egress Gateway Configuration:

```
messageCopy:
  enabled: false
  copyPayload: false
  topicName: message.copy
  ackRequired: false
  retryOnFailure: 0
  retryBackoffTime: 100 #in milliseconds
  requestTimeout: 100 #in milliseconds
  keybasedKafkaProducer: false
  security:
    enabled: false
    protocol: SASL_SSL
    tlsVersion: TLSv1.3
    saslConfiguration:
      userName: test
      password:
        k8SecretName: message-copy-secret
        k8NameSpace: ocegress
        fileName: password.txt
  threadPoolConfigurations:
    coreSize: 16
    maxSize: 16
    queueCapacity: 4200
#kafka-details
kafka:
  bootstrapAddress: 10.75.175.246:30608
```

Observe

Following are the SEPP Message Feed feature specific metrics:

- oc_ingressgateway_msgcopy_requests_total
- oc_ingressgateway_msgcopy_responses_total
- oc_ingressgateway_dd_unreachable
- oc_egressgateway_msgcopy_requests_total
- oc_egressgateway_msgcopy_responses_total
- oc_egressgateway_dd_unreachable

Following are the SEPP Message Feed feature specific KPIs:

- Total Requests Data sent towards DD for Ingress Gateway
- Total Ack received from DD for Requests for Ingress Gateway
- Total Requests Data sent towards DD for Egress Gateway
- Total Ack received from DD for Requests for Egress Gateway

For more information on SEPP Message Feed feature related SEPP Metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs:** For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

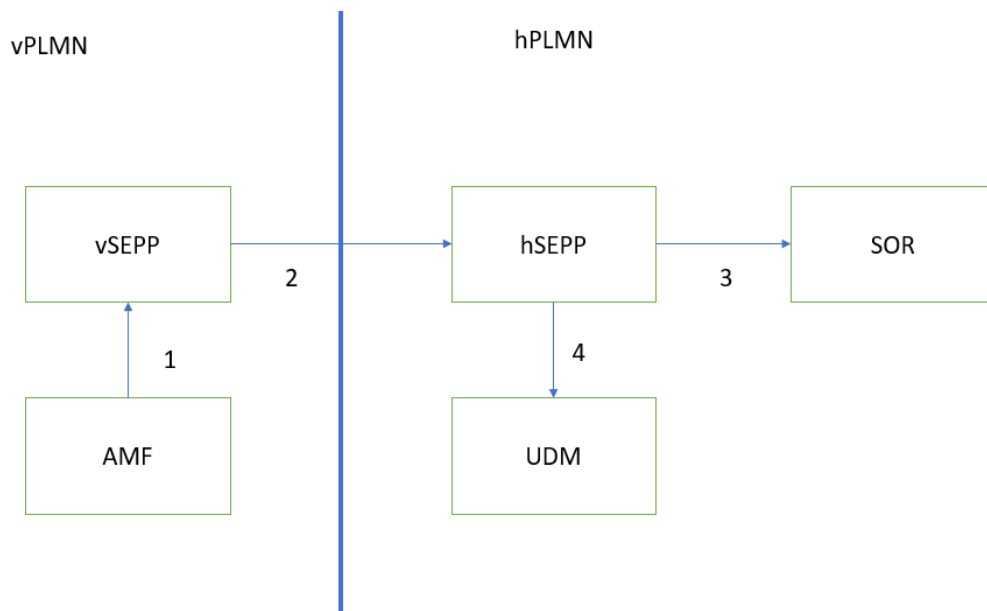
3.2.37 Steering of Roaming (SOR) feature

SOR is deployed as a roaming management solution intended for optimizing roaming cooperation between operators. It allows flexible network selection management for outbound roamers to stimulate an appropriate roaming network choice for subscribers.

SEPP supports the SOR feature to provide better roaming experience to the customers. With this feature, instead of connecting to the required NF, SEPP sends the request to SOR. SOR either connects to the required subscriber or returns the location of the required subscriber to the home SEPP.

The following diagram represents the overview of SOR functionality:

Figure 3-67 SOR Functionality



Here,

1. AMF initiates request towards vSEPP for roaming subscriber.
2. vSEPP forwards the request towards the hSEPP.
3. Hsepp checks method type and RURI (request URI) and if it matches, then forwards the request to the SOR platform.
4. Based on the response, hSEPP forwards the request to the UDM.

The SOR platform receives the standard SBI messages as the input to decide the redirection. The SOR platform responds with a standard HTTP/2 defined response code for redirection.

The SOR platform can either send a response to the SEPP or can directly forward the request to the required UDM. The customer can choose one of the two modes.

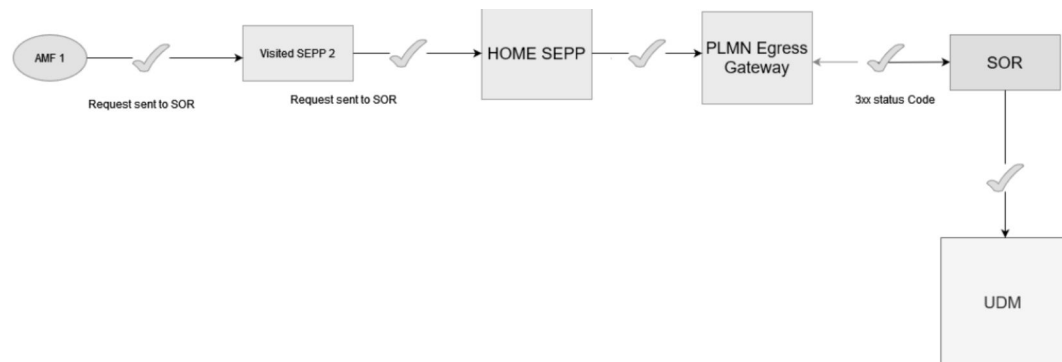
Based on the redirection of response from SOR platform, the two modes are:

- Redirection Disabled Mode (Passive Mode)
- Redirection Enabled Mode (Active Mode)

Redirection Disabled Mode

In this mode, SOR automatically connects to the required subscriber. In this case, producer SEPP receives the required response from the SOR Server and there is no redirection logic is applied at SEPP level. In this mode SEPP does not support redirection codes 3xx.

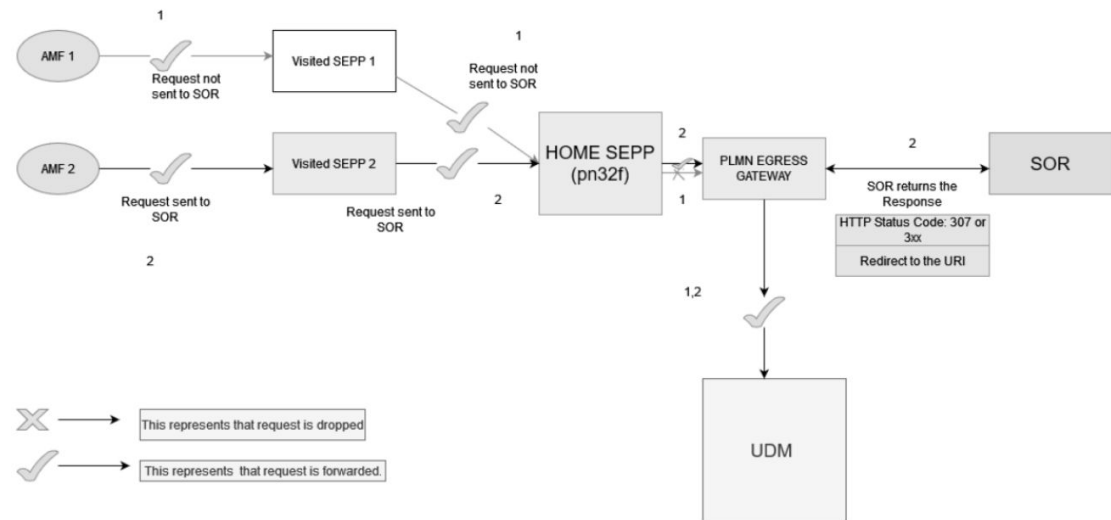
Figure 3-68 Redirection Disabled Mode



Here, SOR platform is acting as a front end. Hence, the search time is less and the SOR platform does the load balance.

Redirection Enabled Mode

In this mode, SOR returns the location of the required subscriber to the home SEPP. In this mode, the producer SEPP receives the 3xx redirection http status code. If it matches the redirection code configured at CNC Console or REST API, then producer SEPP reads the location header and connects to the producer NF. In this mode, configuring SOR is easier.

Figure 3-69 Redirection Enabled Mode

Managing Steering of Roaming (SOR) Feature

Enable

You can enable the SOR feature using the CNC Console or REST API.

Enable using REST API

Set enabled to true in SOR REST API.

```
{
  "enabled": true,
  "retryToProducer": true,
  "redirection": {
    "enabled": true,
    "codes": "308"
  },
  "servers": [
    {
      "priority": "PRIMARY",
      "httpScheme": "http",
      "sorFqdn": "service-nodejs-2-mnc100-mcc111-3gppnetwork-org.adity-sepp2",
      "sorPort": "7070",
      "apiPrefix": "sor",
      "serverHeader": "SOR-service-nodejs-2-mnc100-mcc111-3gppnetwork-
org.adity-sepp2"
    }
  ],
  "retryWait": {
    "retryAltSor": false,
    "retryTime": 1000,
    "retryCount": 0
  },
  "httpCodes": {
    "errorCodes": "501",

```

```

    "exceptions": "java.util.concurrent.TimeoutException"
  },
  "errorConfiguration": {
    "errorCodesToConsumer": "408",
    "errorMessageToConsumer": "Timeout at SOR"
  }
}

```

For more information about API path, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

Enable using CNC Console

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **SOR**.
2. The **Option** appears underneath.
3. Click **Option**, the option screen appears at the right pane. The SOR Feature details are available in the screen.
4. Click **Edit** icon to modify the Option. The **Edit Option** page appears.
5. Set **SOR Enabled** to True on the right pane.

More details about CNC Console Configurations, see [Configuring SEPP using CNC Console](#) section.

Configure

You can configure the feature using REST API and CNC Console.

- **Configure using REST API:** Perform the feature configurations as described in SOR and SOR Config Allowed List REST APIs in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- **Configure using CNC Console:** Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Observe

Following are the SOR feature specific metrics:

- ocsepp_pn32f_sor_requests_total
- ocsepp_pn32f_sor_responses_total
- ocsepp_pn32f_sor_retry_to_producer_requests_total
- ocsepp_pn32f_sor_back_to_consumer_responses_total
- ocsepp_pn32f_sor_failure_total
- ocsepp_pn32f_sor_timeout_failure_total

Following are the SOR feature specific KPIs:

- Pn32f to SoR Request count total
- SoR to Pn32f Response count total

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.38 Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set

Overview

This feature provides ingress request rate limiting on N32 Interface based on Remote SEPP Set to throttle messages from the Remote SEPP sending more than configured limit for the Remote SEPP Set.

There can be scenarios where the flood of messages at Remote SEPP Set level need to be controlled. With this feature, SEPP secures the network when aggregated traffic at the Remote SEPP level exceeds the allowed traffic rate limit.

If the traffic exceeds the allowed traffic rate limit, SEPP does not process the traffic and responds with an error code. The rate limiting functionality at the SEPP allows an operator to configure the acceptable traffic rate from a consumer NF instance.

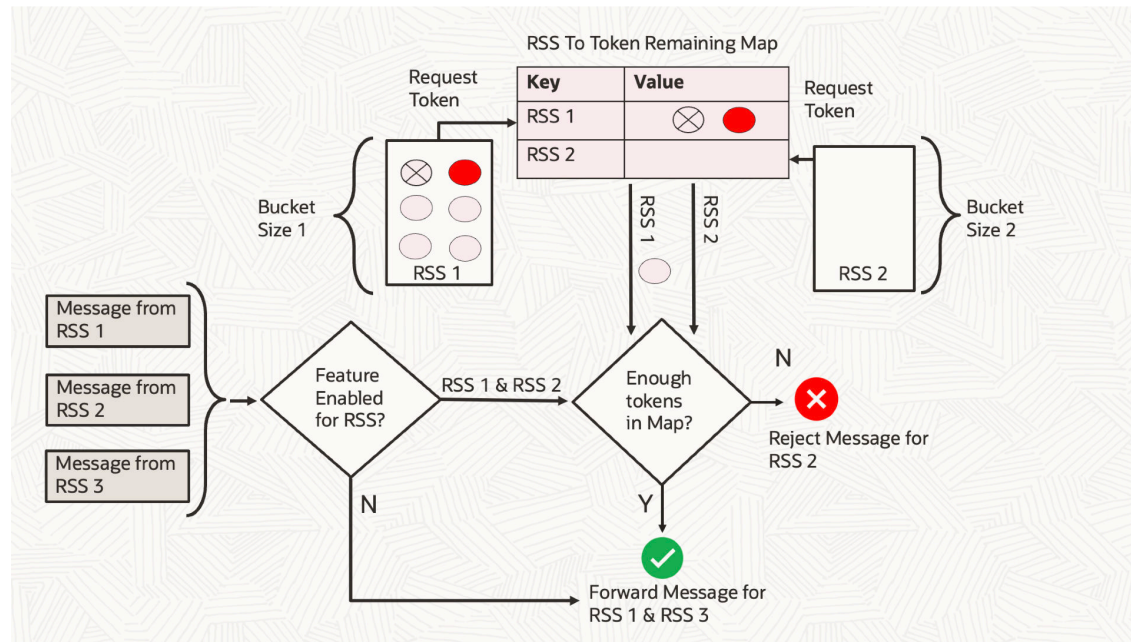
SEPP enables users to configure the maximum number of incoming messages at a given duration.

Algorithm

The Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set feature is implemented using customised Bucket4j which uses TokenBucket Algorithm.

The following is the token bucket algorithm:

- Token bucket algorithm processes messages only if there are tokens in the bucket.
- Bucket is filled with a configurable rate during token consumption or asynchronously.
- Tokens are removed from bucket on message processing.
- The following four configurations are required:
 - Bucket Capacity - Bucket size defined the capacity to handle traffic burst.
 - Refill Rate - Tokens (number of) which should be added to refill the bucket.
 - Refill Duration - to decide how frequently to refill bucket.
 - Request Token - to define the batch size of token requested from the corresponding bucket. It is recommended to have the Request Token value between 3 to 5% of the refill rate.
- Tokens and duration defines the traffic rate.

Figure 3-70 Token Bucket Algorithm

The above algorithm shows Ingress Rate limiting for message received from three RSS; RSS 1, RSS 2, and RSS 3.

When the message is received at N32 Ingress Gateway, if the Ingress rate limiting feature is not enabled for any RSS, then message is simply forwarded from that RSS. In our example, Ingress rate limiting feature is not enabled for RSS 3 and enabled for RSS 1 and RSS 2, So message coming from RSS 3 will be forwarded.

Now different buckets will be created for both RSS 1 and RSS 2 as per user configuration of bucket capacity and request token.

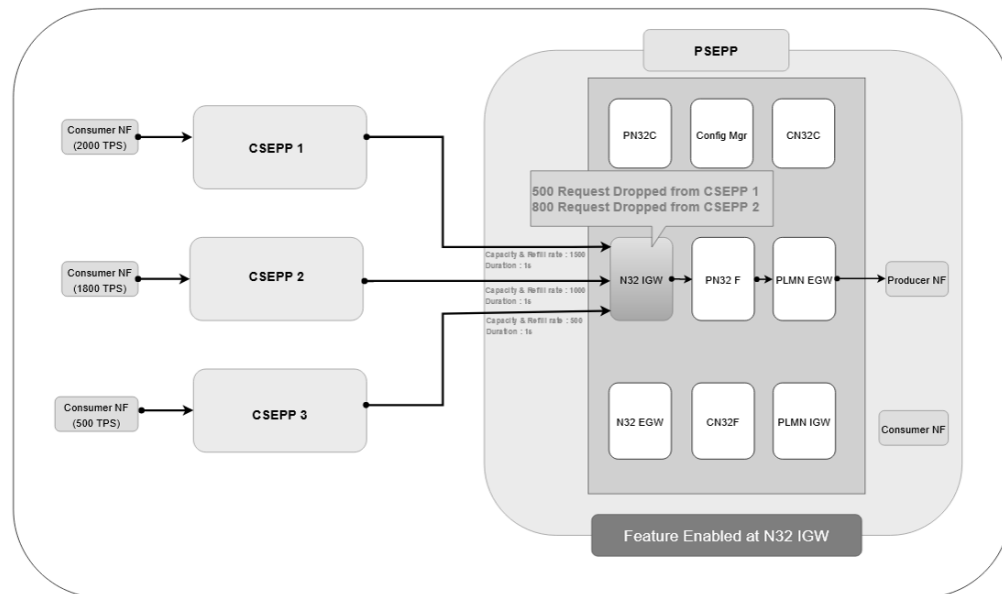
In above case, request token is set to 2 and from request token, RSS to token remaining mapping is filled. In this mapping one message is processed (shown in pink colour with a cross inside) and one token (with red colour) in RSS to token remaining mapping for RSS 1 is still remaining. When message from RSS 1 is processed, It is checked if there are enough tokens present in RSS to token remaining mapping for RSS 1. Since there are 2 tokens available, the message is forwarded.

When message from RSS 2 is processed, It is checked if there are enough tokens present in RSS to Token remaining mapping for RSS 2. Since there no tokens available for RSS 2, this message will be rejected with user configurable status code and title.

Detailed Description

The Rate limiting for Ingress Roaming Signaling per Remote SEPP Set feature protects the network from unsecured NF consumers that flood the SEPP microservices with excessive requests.

The following diagram represents the rate limiting for ingress roaming signaling per Remote SEPP Set feature:

Figure 3-71 Rate limiting for Ingress Roaming Signaling per Remote SEPP Set Architecture

In the above diagram, the ingress rate limiting per Remote SEPP Set feature is enabled on the Producer SEPP (PSEPP) N32 Ingress Gateway with a capacity of 1500, 1000, and 500 requests with one second duration for Consumer SEPP (CSEPP 1), CSEPP 2, and CSEPP 3. The requests are initiated from the customer NF at initiator side which passes through the Consumer SEPP and reach to the Producer SEPP N32 Ingress Gateway.

If the incoming requests from CSEPP 1, CSEPP 2, and CSEPP 3 on N32 Ingress Gateway of Producer SEPP are more than the configured capacity of 1500, 1000 and 500 request per second, the requests above the configured value are rejected with an error response as configured by the user. In the above example, the consumer NFs are generating 2000 TPS, 1800 TPS and 500 TPS and as per the capacity set at Producer SEPP N32 Ingress Gateway, 500 and 800 requests are rejected respectively with an configured error response.

Managing Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Feature

Enable

You can enable the feature using the Helm parameter and CNC Console or REST API.

Note

Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set feature is disabled by default.

To enable the feature, perform the following steps:

1. In the Yaml file:

Set the `rssRateLimiting.enabled` parameter to `true` in the N32 Ingress Gateway section of `ocsepp-custom-values- <version>.yaml` file.

The following parameters are to be configured:
###In the IGW section

```
rateLimiting:
enabled: true
  rssRateLimiting:
    enabled: false
```

2. To configure using CNC Console or REST API:

CNC Console

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Ingress Rate Limiting**.
2. Select **Remote SEPP Set** which is defined under **Ingress Rate Limiting** screen.
3. The **Option** appears underneath.
4. Click **Option**, the option screen appears at the right pane. The Ingress Rate Limiting Feature details are available in the screen.
5. Click **Edit** icon to modify the Option. The Edit Option page appears.
6. Set **RSS Ingress Rate Limiting Enabled** to True on the right pane.

REST API

Set `ingressRateLimitingEnabled` as `True` at the Ingress Rate Limiting REST API. Set `ingressRateLimitingEnabled` as `true` in the Remote SEPP Set REST API.

Note

In CNC Console and REST API, Ingress Rate Limiting feature can be enabled or disabled globally for the complete system at runtime. The Ingress Rate Limiting feature can be enabled for a particular RSS.

- If the Helm parameter `rssRateLimiting` is set to `false`, then Ingress Rate Limiting feature is globally disabled.
- If the Ingress Rate Limiting feature is disabled on CNC Console Option screen of Ingress Rate Limiting, then Ingress Rate Limiting feature is globally disabled.
- If it is required to disable or enable Ingress Rate Limiting feature for a particular RSS, then the helm parameter `rssRateLimiting` and **Enable Ingress Rate Limiting** option in CNC Console Ingress Rate Limiting feature Option screen must be enabled.

Configure

You can configure the feature using REST API and CNC Console.

- **Configure using REST API:** Perform the feature configurations as described in Ingress Rate Limiting System Option and Remote SEPP Set REST APIs in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

- **Configure using CNC Console:** Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Observe

Following are the feature specific metrics:

- `oc_ingressgateway_rss_ratelimit_total` (Added with three different metric filter status)

Following are the feature specific KPIs:

- Average No of messages discarded for a particular RSS
- Average No of messages accepted for a particular RSS
- Average No of messages for which feature not applied
- Average of all messages by Status
- List of Average number of messages accepted for all RSS
- List of Average number of messages dropped for all RSS

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

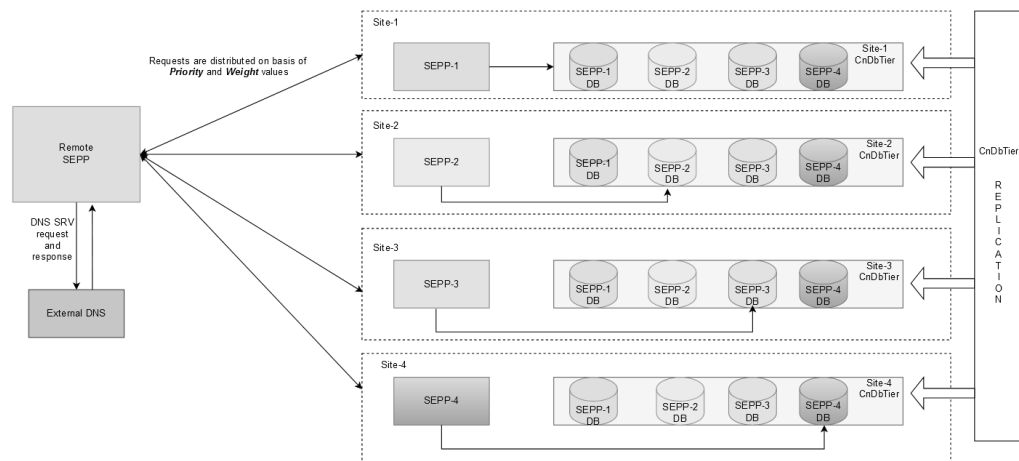
In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.39 Georedundancy

Oracle Communications Cloud Native Core, Security Edge Protection Proxy (SEPP) supports georedundancy to ensure high availability and redundancy. It offers a two, three, or four-sites georedundancy to ensure service availability when one of the SEPP sites is down. When SEPP is deployed as georedundant site, all the sites work in an active state and the same data is available at all the sites.

The NFs send service requests to their primary SEPP. When the primary SEPP site is unavailable, the NFs redirect the service requests to the alternate SEPP site. In this case, NFs get the same information from alternate SEPP, and each site maintains and uses its set of configurations.

Figure 3-72 Georedundancy Overview

The prerequisites for georedundancy feature are as follows:

- Each site must configure the remote georedundant SEPP sites' details.
- Georedundancy feature cannot be disabled once enabled on a site.
- Georedundant sites must be time synchronized.
- Georedundant sites must be reachable from NFs or peers on all the sites.

Detailed Description

The specifications for georedundancy feature are as follows:

- All the georedundant sites must have same Helm and REST based configurations except NF Instanced Id, SEPP Endpoint, and port.
- The georedundant SEPP sites must be reachable from NFs or peers on all the sites.
- The same NFs or peers must not communicate to other georedundant SEPP sites simultaneously for the same session.
- When SEPP is deployed in three-site georedundant mode:
 - In case of three-site georedundant mode, NFs or peers can configure the three SEPP instances in Remote SEPP Set as primary, secondary, and tertiary. When the primary instance is available, the NFs send service requests to the primary instance.
 - All the sites register with NRF independently and work in an active state.
 - All SEPP instances share the context data by using the replication service of cnDBTier. This enables service continuity during the failure of any of the sites.
 - The NFs in a given site can discover SEPP instances through NRF.
- When SEPP is deployed in four-site georedundant mode:
 - All the four SEPP instances need to be configured using DNS mechanism in communication between different SEPPs.
 - All the sites work in an active state.
 - All instances share context data using the cnDBTier's replication service to enable service continuity during site failure.

- When the peer detects the failure of the primary instance, the peer redirects its traffic to the secondary instance while the primary instance remains unavailable until it becomes available again.
- If the priorities are same for all the SEPP instances, then load sharing can be done between SEPP-SEPP communication on the basis of weight factor.

① Note

Rollback sequence for n-site geo-replicated setup (n = 2, 3, 4)

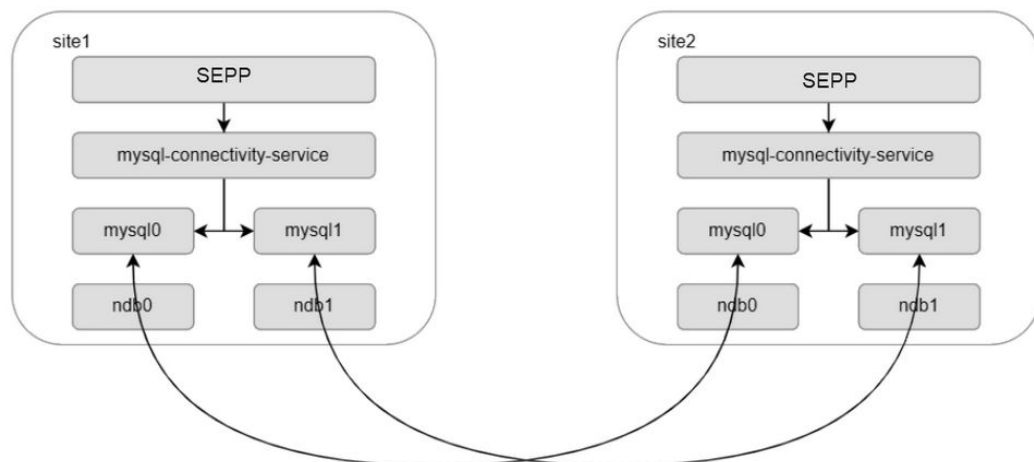
1. Begin by rolling back cnDBTier-S(n), followed by SEPP-S(n) and CNC Console-S(n) for the most recent site (Site n).
2. Next, move to cnDBTier-S(n-1) and initiate Geo Replication Recovery to sync it with the healthy cnDBTier-S(n). Then, rollback SEPP-S(n-1) and CNC Console-S(n-1).
3. Continue the rollback down to Site 1, starting with cnDBTier-S(1) and performing Geo Replication Recovery with the healthy cnDBTier-S(2), followed by rolling back SEPP-S(1) and CNC Console-S(1).

SEPP supports the following types of georedundant deployments:

3.2.39.1 Two-Site Georedundancy

The following diagram depicts the topology for two-site georedundant SEPP deployment:

Figure 3-73 Two-Site Georedundancy



The two-site georedundancy is established when the cnDBTier at both sites is configured with georedundancy configurations. The cnDBTier provides bi-directional replication between both sites. When the two-sites replication is up and running, the updates done in the database at one site are replicated to the other sites in real time. The changes include creating, changing, or deleting a record.

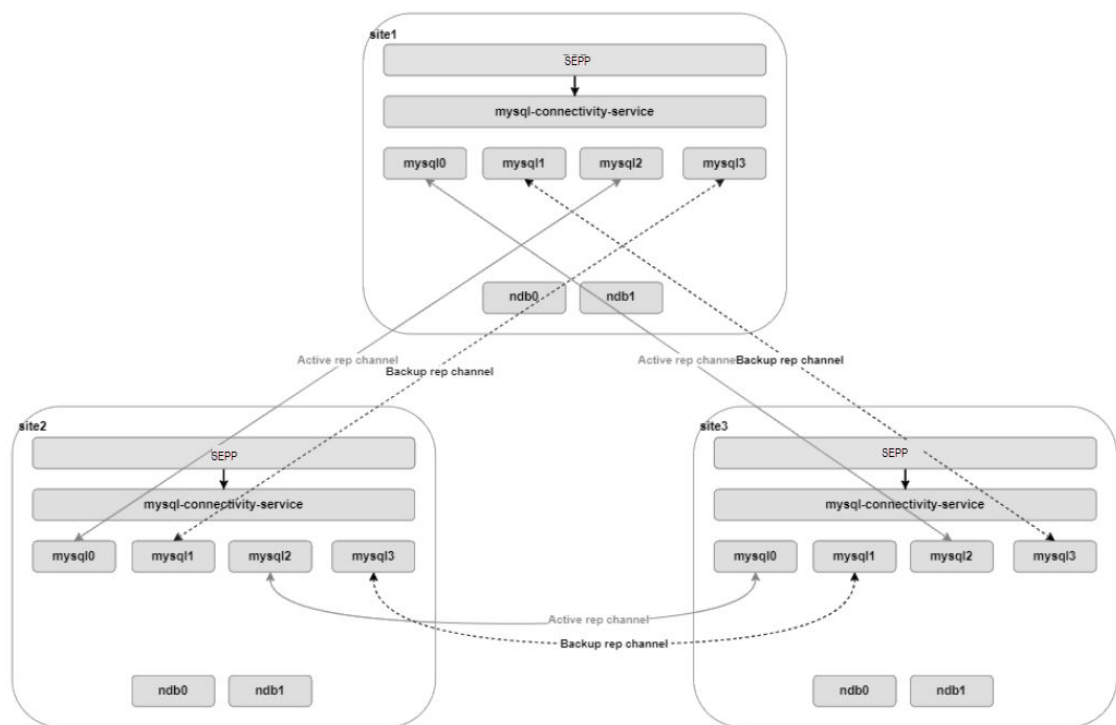
The benefits of two-site georedundancy are:

- In case of a single site failure, the alternate site takes up the load of the failed site.
- In case of single or dual site failure (or connection failure or any other failure), the SEPP at the active site(s) serves the consumer NFs. For example, in a two-site redundant system with Site 1 and Site 2,
 - If Site1 fails, SEPP at Site 2 serves the Consumer NF of Site 1.
 - If Site2 fails, SEPP at Site 1 serves the Consumers NF of Site 2.

3.2.39.2 Three-Site Georedundancy

The following diagram depicts the topology for three-site georedundant SEPP deployment:

Figure 3-74 Three-Site Georedundancy



In case of three-site georedundancy, bi-directional replication is established from each site to the other two sites. The database updates from each site are replicated to the other two sites over the replication channel.

The benefits of three-site georedundancy are:

- In case of a single site failure, the remaining two sites take up the load of the failed site.
- Each site uses the SQL nodes for active and standby replication channels for high availability of the replication channels.
- In case of single or dual site failure (or connection failure or any other failure), the SEPP in active state serves the consumer NFs. For example, in a three-site redundant system with Site 1, Site 2, and Site 3,
 - If Site 1 fails, SEPP at Site 2 and Site 3 serve the Consumer NF of Site1.

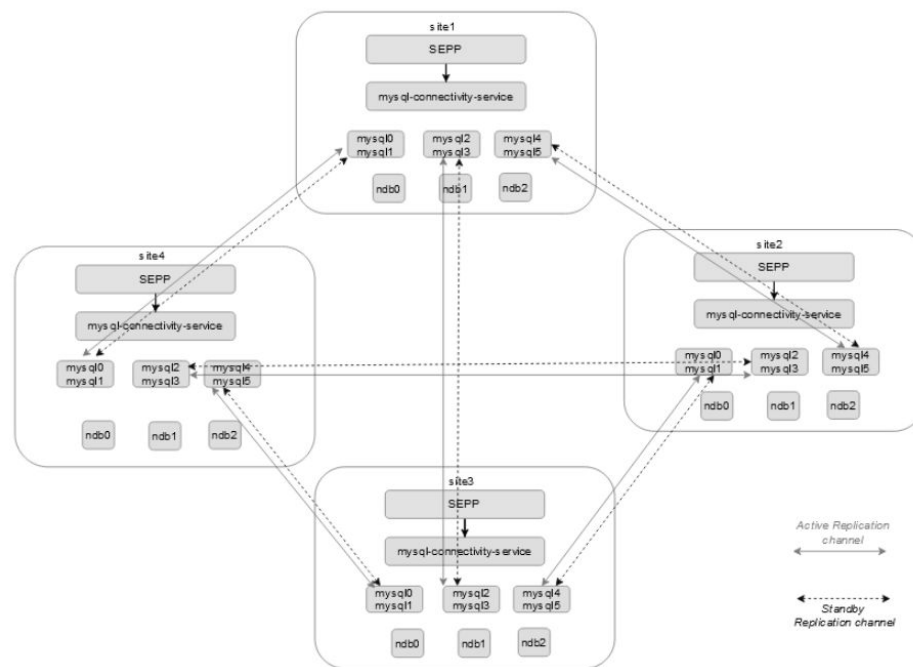
- If Site 1 and Site 2 fail, SEPP at Site 3 serves Consumer NF of Site 1 and Site 2.

When the three-sites replication is up and running, the updates done in the database at one site are replicated to the other sites in real time. The changes include creating, changing, or deleting a record.

3.2.39.3 Four-Site Georedundancy

The following diagram depicts the topology for four-site georedundant SEPP deployment:

Figure 3-75 Four-site Georedundancy



SEPP supports the four-site georedundant deployment. In case of four-site georedundancy, each site participates in a 4-way replication. The database updates from each site are replicated to the other three sites over the replication channels.

The advantages of four-site georedundancy are:

- In case of a single site failure, the remaining three sites keep establishing the bi-directional replication.
- Requires 6 SQL pods and 3 db-rep-svc at each site.
- Each site uses two SQL nodes for active and standby replication channels for high availability of the replication channels.
- In case of single or dual site failure (or connection failure or any other failure), the SEPP in active state serves the consumer NFs. For example, in a 4-site redundant system with Site 1, Site 2, Site 3, and Site 4,
 1. If Site 1 fails, SEPP at Site 2, Site 3, and Site 4 serve the Consumer NF of Site1.
 2. If Site 1 and Site 2 fail, SEPP at Site 3 and Site 4 serve the Consumer NF of Site 1 and Site 2.

3. If , Site 2, and Site 3 fail, SEPP at Site4 serves the Consumer NF of Site 1, Site 2, and Site 3.

When the four-site replication is up and running, the updates done in the database at one site are replicated to the other sites in real time. The changes include creating, changing, or deleting a record.

3.2.39.4 Managing the Feature

This section provides information about Helm, REST API, and Cloud Native Configuration Console (CNC Console) configurations required for enabling and configuring this feature.

3.2.39.4.1 Prerequisites

To deploy SEPP in a georedundant environment:

1. cnDBTier must be installed and configured for each site. For the installation procedure, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
2. The Database Replication Channels between the sites must be up. For information about installing cnDBTier and parameters required for setting the Database replication in cnDBTier custom values.yaml, see "Installing cnDBTier" and "Customizing cnDBTier" in *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.
3. Configure MySQL Database, Users, and Secrets. For the configuration procedure, see "Preinstallation Tasks" in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.
4. Deploy SEPP over the replicated cnDBTier sites. For information about installing and deploying SEPP, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

3.2.39.4.2 Enable

Configure the following to enable the georedundancy feature:

Note

Configuring these attributes during deployment is mandatory before enabling the georedundancy feature. Otherwise, georedundancy cannot be enabled, and SEPP at the site will act as a standalone SEPP.

Helm Configuration for Database:

Configure the following parameters in the ocsepp-custom values-<version>.yaml file for all four sites:

- global.seppDbName
- global.seppBackupDbName
- global.mysql.primary.host
- global.mysql.primary.port
- global.nameSpace

- `global.mysql.primary.host`
- `global.localProfile.name`
- `global.localProfile.nfInstanceId`

At Site 1:

```
global:
  # Kubernetes Secret containing DB credentials
  dbCredSecretName: &dbCredSecretNameRef 'ocsepp-mysql-cred'

  # Namespace where secret is deployed
  nameSpace: DEPLOYMENT_NAMESPACE

  mysql:
    primary:
      host: &mySqlHostRef "mysql-connectivity-service.site1"
      port: &mySqlPortRef 3306
    # Name of Sepp database
    seppDbName: &dbNameRef "seppdbSite1DB"
    # Name of Sepp backup database
    seppBackupDbName: "seppbackupdbSite1DB"

  nfFqdn: &nfFqdnRef "site1sepp1.inter.oracle.com"

  localProfile:
    name: "SEPP-1"
    nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a06"
```

At Site 2:

```
global:

  # Kubernetes Secret containing DB credentials
  dbCredSecretName: &dbCredSecretNameRef 'ocsepp-mysql-cred'

  # Namespace where secret is deployed
  nameSpace: DEPLOYMENT_NAMESPACE

  mysql:
    primary:
      host: &mySqlHostRef "mysql-connectivity-service.site2"
      port: &mySqlPortRef 3306
    # Name of Sepp database
    seppDbName: &dbNameRef "seppdbSite2DB"
    # Name of Sepp backup database
    seppBackupDbName: "seppbackupdbSite2DB"

  nfFqdn: &nfFqdnRef "site2sepp2.inter.oracle.com"

  localProfile:
    name: "SEPP-2"
    nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a07"
```

At Site 3:

```

global:
  # Kubernetes Secret containing DB credentials
  dbCredSecretName: &dbCredSecretNameRef 'ocsepp-mysql-cred'

  # Namespace where secret is deployed
  nameSpace: DEPLOYMENT_NAMESPACE
  mysql:
    primary:
      host: &mySqlHostRef "mysql-connectivity-service.site3"
      port: &mySqlPortRef 3306
    # Name of Sepp database
    seppDbName: &dbNameRef "seppdbSite3DB"
    # Name of Sepp backup database
    seppBackupDbName: "seppbackupdbSite3DB"

  nfFqdn: &nfFqdnRef "site3sepp3.inter.oracle.com"

localProfile:
  name: "SEPP-3"
  nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a08"

```

At Site 4:

```

global:
  # Kubernetes Secret containing DB credentials
  dbCredSecretName: &dbCredSecretNameRef 'ocsepp-mysql-cred'
  # Namespace where secret is deployed
  nameSpace: DEPLOYMENT_NAMESPACE
  mysql:
    primary:
      host: &mySqlHostRef "mysql-connectivity-service.site4"
      port: &mySqlPortRef 3306
  # Name of Sepp database
  seppDbName: &dbNameRef "seppdbSite4DB"
  # Name of Sepp backup
  database seppBackupDbName: "seppbackupdbSite4DB"
  nfFqdn: &nfFqdnRef "site4sepp4.inter.oracle.com"
  localProfile:
    name: "SEPP-4"
    nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a09"

```

For more information about configuring the parameters, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Following is the sample configuration at Site named "**site1**" (SEPPInstanceId: 99faf1bbc-6e4a-4454-a507-aef01a101a07), which is georedundant with Sites, "**site2**" (SEPPInstanceId: 99faf1bbc-6e4a-4454-a507-aef01a101a07) and "**site3**" (SEPPInstanceId: 9faf1bbc-6e4a-4454-a507-aef01a101a08):

3.2.39.5 Adding a Site to an Existing SEPP Georedundant Site

This section describes the procedure to add a site to an existing SEPP site.

Prerequisites

1. SEPP connected to a cnDBTier is up and running. This is referred as Site 1 or Site 2.
2. SEPP and cnDBTier versions used for Site 1 and Site 2 installation must be identified and same must be used for adding another site.

Adding a site

1. Install a new cnDBTier. This cnDBTier must act as a georedundant database to the cnDBTier in Site-1 or Site-2. For more information to install a cnDBTier, see *Oracle Communications Cloud Native Core cnDBTier User Guide*.
2. Verify the replication channel between the cnDBTier sites by sending the following request to the dbMonitor service of both the cnDBTier sites. The responses from both the cnDBTier sites must show the status of replication channel as up:

```
curl http://<mysql-db-monitor-service>:8080/db-tier/status/replication/
realtime
```

Sample command:

```
curl http://mysql-cluster-db-monitor-svc:8080/db-tier/status/replication/
realtime
```

Sample output:

```
[{"localSiteName":"Site-1","remoteSiteName":"Site-2","remoteSiteName":"Site-3",
"replicationStatus":"UP","secondsBehindRemote":0}]
```

3. Create the required secrets to install the Site-2 or Site-3 SEPP in the same namespace as the new cnDBTier installed in step-1, by following the steps described in the "Configuring Kubernetes Secret for Accessing SEPP Database" subsection from "Installing SEPP" chapter in *Oracle Communications Cloud Native Core Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.
4. Run the following command to verify the secret created:

```
$ kubectl describe secret <database secret name> -n <Namespace of SEPP deployment>
```
5. Install the Site-2 or Site-3 SEPP using the updated ocsepp-custom values-<version>.yaml file. For more information on installation procedure, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.
Verify the installation as mentioned in *Oracle Communications Cloud Native Core Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.
6. After the installation is complete, make sure that the "DbReplicationStatusInactive" alert is not raised in the alert dashboard. If the alert exists, verify the above configuration steps. If the alert is not present, it means the DBReplication is up and the georedundant deployment of all sites is successful.

3.2.39.6 Removing a Site from an Existing Georedundant Deployment

This section describes the procedure to remove a site from an existing SEPP deployment.

Prerequisites

1. SEPP connected to a cnDBTier is up and running. This is referred as Site 1 (SEPP-1 and SEPP-2).

Procedure

This section describes the procedure to remove a site (Site1 or Site 2) from an existing georedundant SEPP deployment.

1. Uninstall the SEPP instance from the site to be removed.
For more information on uninstallation procedure, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.
2. Remove the DB Replication and the NDBCluster on the site to be removed. For more information on this procedure, see *Oracle Communications Cloud Native Core, cnDBTier User Guide*.

3.2.39.7 Recovering a Failed Site

This section describes the procedure of recovering a failed site in SEPP deployment.

When any of the four sites is down or replication is broken with other sites, then perform the following procedure to restore the failed site from healthy remote site backup.

Mark the site as FAILED in remote healthy cluster

```
curl -i -X POST http://<failed site replication svc IP: port>/db-tier/gr-recovery/remotesite/<failed site name>/failed
```

```
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 0
Date: Fri, 19 Apr 2024 08:14:48 GMT
```

Start recovery of failed site from healthy cluster

```
curl -i -X POST http://<failed site replication sec IP:port>/db-tier/gr-recovery/site/<failed site name>/grbackupsite/<remote Healthy site name>/start
```

```
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
```

```
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 0
Date: Fri, 19 Apr 2024 08:46:13 GMT
```

Monitor the fault recovery state

```
curl -i -X GET http://<failed site replication sec IP:port>/db-tier/gr-
recovery/site/<failed site name>/status
```

```
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Type: application/json
Transfer-Encoding: chunked
Date: Fri, 19 Apr 2024 09:04:41 GMT

{"localSiteName": "<failed site
name>", "geoReplicationRecoverystatus": "COMPLETED", "geoReplicationRecoverystate
": "COMPLETED", "remotesitesGrInfo": [{"remoteSiteName": "<remote site
name1>", "replchannel_group_id": "1", "gr_state": "COMPLETED"},
{"remoteSiteName": "<remote site
name2>", "replchannel_group_id": "1", "gr_state": "COMPLETED"},
{"remoteSiteName": "<remote site
name3>", "replchannel_group_id": "1", "gr_state": "COMPLETED"}]}
```

For more information about installation and recovery procedures, see *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide*.

3.2.40 Support for Kubernetes Resource

3.2.40.1 Support for Network Policies

Network Policies are an application-centric construct that allows you to specify how a pod is allowed to communicate with various network entities. To control communication between the cluster's pods and services and to determine which pods and services can access one another inside the cluster, it creates pod-level rules.

Previously, SEPP had the privilege to communicate with other namespaces, and pods of one namespace could communicate with others without any restriction. Now, namespace-level isolation is provided for the SEPP pods, and some scope of communications is allowed between the SEPP and pods outside the cluster. The network policies enforce access restrictions for all the applicable data flows except communication from Kubernetes node to pod for invoking container probe.

Managing Support for Network Policies

Enable

To use this feature, network policies need to be applied to the namespace in which SEPP is applied.

Configure

You can configure this feature using Helm. For information about configuring Network Policy for SEPP Deployment, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

Observe

There are no specific metrics, KPIs, and alerts required for the Support of Network Policy functionality.

3.2.40.2 Pod Disruption Budget

Background

Pods are the basic units of Kubernetes and are used to deploy Cloud Native Core (CNC) Network Functions (NFs).

Pod disruption occurs due to voluntary and involuntary disruptions.

Voluntary Disruptions

Voluntary pod disruption occurs when:

- pods are removed from a cluster.
- SEPP worker nodes are drained at the time of cluster upgrade and running pods are evicted from these nodes.
- load balancing of worker nodes is performed by evenly distributing pods among worker nodes.

Involuntary Disruptions

Involuntary pod disruption occurs when:

- the system hardware or software fails.
- SEPP worker nodes are removed accidentally.
- pods are evicted due to insufficient resources on SEPP worker nodes.

Kubernetes has introduced the PodDisruptionBudgets (PDB) concept to manage pod disruption outages in the clusters. PDB is applicable only for voluntary disruptions. It is used to avoid service impact due to voluntary disruption and ensures high availability of NF.

PodDisruptionBudget SEPP Implementation Overview

PodDisruptionBudget (PDB) is a Kubernetes resource that allows you to achieve high availability of scalable application services when the cluster administrators perform voluntary disruptions to manage the cluster nodes. PDB restricts the number of pods that are down simultaneously from voluntary disruptions. Defining PDB is helpful to keep the services running undisturbed when a pod is deleted accidentally or deliberately. PDB can be defined for high available and scalable SEPP services such as cn32f-svc, pn32f-svc, cn32c-svc, pn32c-svc, n32-ingress-gateway, n32-egress-gateway, plmn-ingress-gateway, plmn-egress-gateway, nfmanagement-svc, nfdiscovery-svc, and perf-info services.

It allows safe eviction of pods when a Kubernetes node is drained to perform maintenance on the node. It uses the `maxUnavailable` parameter specified in the Helm chart to determine the maximum number of pods that can be unavailable during a voluntary disruption. For example, when a `maxUnavailable` is 25%, pod evictions are allowed as long as no more than 25% of the desired pods are in unhealthy state.

For more information about this functionality, see <https://kubernetes.io/docs/concepts/workloads/pods/disruptions/#pod-disruption-budgets>.

Default PodDisruptionBudget for SEPP Deployment

Table 3-51 Default PodDisruptionBudget for SEPP Deployment

Microservice	PDB Value Specified (maxUnavailable)
cn32c-svc	25%
pn32c-svc	25%
cn32f-svc	25%
pn32f-svc	25%
config-mgr-svc	Not Applicable
plmn-ingress-gateway	25%
plmn-egress-gateway	25%
n32-ingress-gateway	25%
n32-egress-gateway	25%
app-info	25%
config-server	50%
perf-info	25%
nf-mediation	25%

Managing Pod Disruption BudgetEnable

This feature is enabled automatically if you are deploying SEPP with Release 16.

Configure

You can configure this feature using Helm. For information about configuring PDB, see *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*.

Observe

There are no specific metrics, KPIs, and alerts required for the support of PDB functionality.

3.2.41 Alternate Routing and Load Sharing based on the DNS SRV Record for Home Network Functions

Overview

SEPP selects SCP and NRF for indirect communication of 5G messages using the static configurations done by the operator. This feature enables SEPP to route the messages to the SCP, NRF or Producer NFs using DNS SRV records, in addition to already existing static manual configuration by the operator.

SEPP supports DNS SRV based selection of SCP, NRF or Producer NFs through Egress Gateway (EGW), which supports Alternate Route Service (ARS). It enables SEPP to resolve the FQDN or Virtual FQDN to alternative FQDNs of 5G NFs. EGW uses the virtual FQDN of

NF instances to query the Alternate Route Service and retrieve the list of alternative FQDNs with priorities and weight assigned to each of them. Based on the priorities, EGW selects the NF instances for rerouting attempts and distributes the incoming traffic in proportion to the weight value assigned. If the priorities of the records are same, then the traffic will be load shared among the FQDNs received based on the weight factor.

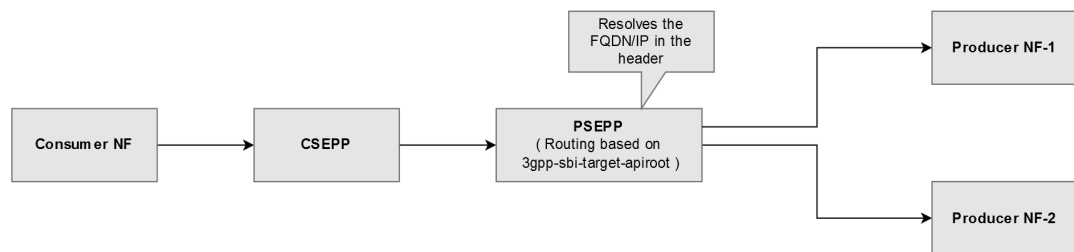
Using this feature, SEPP performs alternate routing of messages by querying SRV records with the DNS server to find an alternate producer NF and route the service request. SEPP supports load sharing among multiple NRF, SCP, and Producer NFs' records using DNS SRV queries at plmn-egress-gateway. SEPP uses either NF Set or DNS SRV records to perform alternate routing.

Support for Direct and Indirect Communication

The Consumer NF messages are routed through plmn-egress-gateway based on the 3gpp-sbi-target-apiroot header.

In the case of direct communication, the gateway service routes the messages to the FQDN or IP defined in the header. In the case of indirect communication, static peers are defined on the gateway service and the incoming requests are routed to the defined hosts if they match the routing criteria.

Figure 3-76 Direct Communication



Routing to NRF, SCP, and Producer NFs Through DNS SRV

SEPP supports priority and weight based load sharing of the Consumer NFs requests through plmn-egress-gateway. SEPP sends a DNS SRV query request based on the virtual FQDN configured to identify peer NF instances. The DNS server returns the response with the multiple FQDNs associated with the virtual FQDN, and each FQDN can have multiple IP addresses associated to each. Further based on the DNS SRV algorithm defined at the plmn-egress-gateway, it selects the peer with the highest priority. If the priorities are same for multiple FQDN records, then the incoming requests are load shared based on the weight value assigned to them.

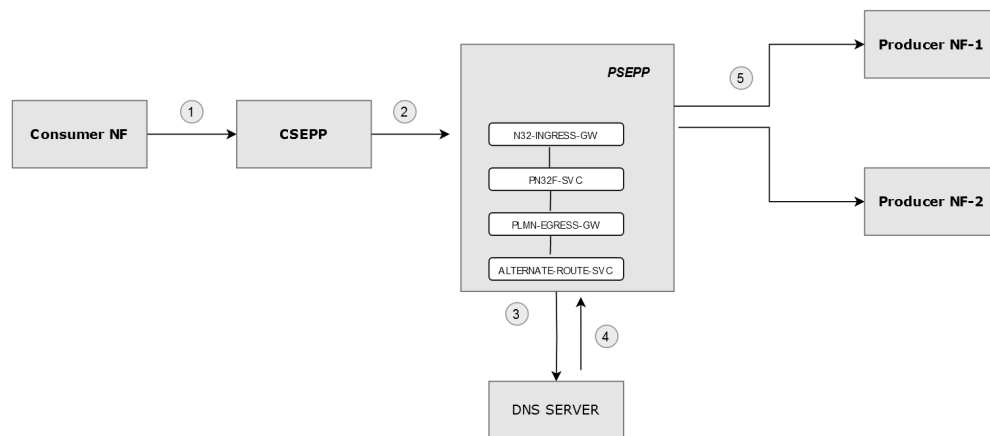
Below is the format of SRV records defined at DNS server:

_Service._Proto.Name TTL Class SRV Priority Weight Port Target

_http._tcp.example.com 86400 IN SRV 1 10 80 blr.example.com

In this example, the virtual FQDN "example.com" will return the single FQDN record "blr.example.com" with priority 1 and weight 10.

Similarly, multiple records can be defined against the single virtual FQDN. Each FQDN returned can be resolved to an IP address in the DNS server.

Figure 3-77 Routing to Producer NFs Through DNS SRV

The above diagram represents the routing to Producer NFs Through DNS SRV. The flow of routing is represented by the numbering.

SEPP supports multiple deployments for NFs (Example:SCP) for routing of Consumer NF queries. When SEPP receives a request, it processes the request from Consumer NF based on the API root header configuration. PLMN Egress Gateway (EGW) performs the routing based on the SbiRouting functionality at plmn-egress-gateway and forwards the requests to the peer configuration defined in the route. This process of routing the request between SEPP through SCP is known as indirect communication.

SEPP supports alternate routing and weight based routing (load sharing) for virtual peer instances. In the case of virtual peer instances, SEPP identifies peer SCP instances based on the virtual FQDN and distributes the 5G traffic to NF instances based on DNS SRV algorithm defined at plmn-egress-gateway. Lower the priority, higher is the chance of selection and if multiple instances have same priority, then distribution or load sharing will be based on the weight factor assigned to each of the records. In the case of failures, SEPP also supports alternate routing to the peers based on their priority.

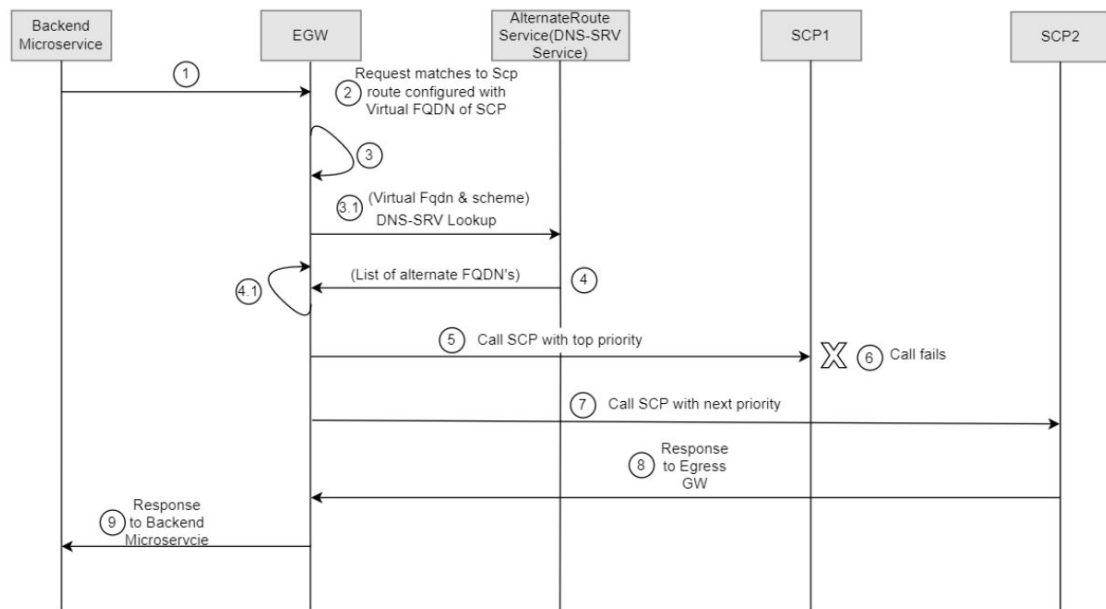
Alternate Route Service uses the DNS SRV records for the resolution of virtual FQDN. PLMN Egress Gateway uses the virtual FQDN of peer instances to query the Alternate Route Service and get the list of alternate FQDNs with a priority and weight assigned to each one of them. The Egress Gateway selects the peer instances for routing or re-routing attempts based on this priority value assigned to the peers and distribute the traffic based on weight value if priorities are same.

The routing can be:

- Priority Based
- Weight Based

Priority Based Routing

In priority based routing, the virtual FQDN configured for SCP instances with different priorities.

Figure 3-78 Priority Based Routing

1. PLMN Egress Gateway (EGW) receives a request from the backend microservice.
2. PLMN EGW maps the request to a route configured with virtual FQDN of SCP. Since the route is configured with virtual FQDN of SCP, the virtual FQDN is resolved to the list of alternate FQDNs.
3. PLMN EGW gets the list of FQDNs from cache data. If cached data has not expired, Step 3 (a) and Step 4 are skipped. If the cache has expired data in it, EGW continues with the Step 3 (a).
 - a. EGW calls the Alternate Route Service to resolve virtual FQDN to alternate FQDNs and puts it in the cached memory.
4. PLMN EGW performs the DNS-SRV query using the virtual FQDN & scheme.
 - a. EGW caches the resolved virtual FQDN details.
5. PLMN EGW selects the SCP instance based on the priority, and it calls the primary SCP.
6. The call fails if the primary SCP is unavailable.
7. Based on the error codes configured in the route, EGW selects the SCP instance with next priority to initiate a call.
 - a. PLMN EGW repeats this step for all available SCP instances in the order of their priority as per the number of reroute attempts configured by the operator.
8. PLMN EGW receives the response.
9. PLMN EGW forwards the response to the backend micro-service as it is received.

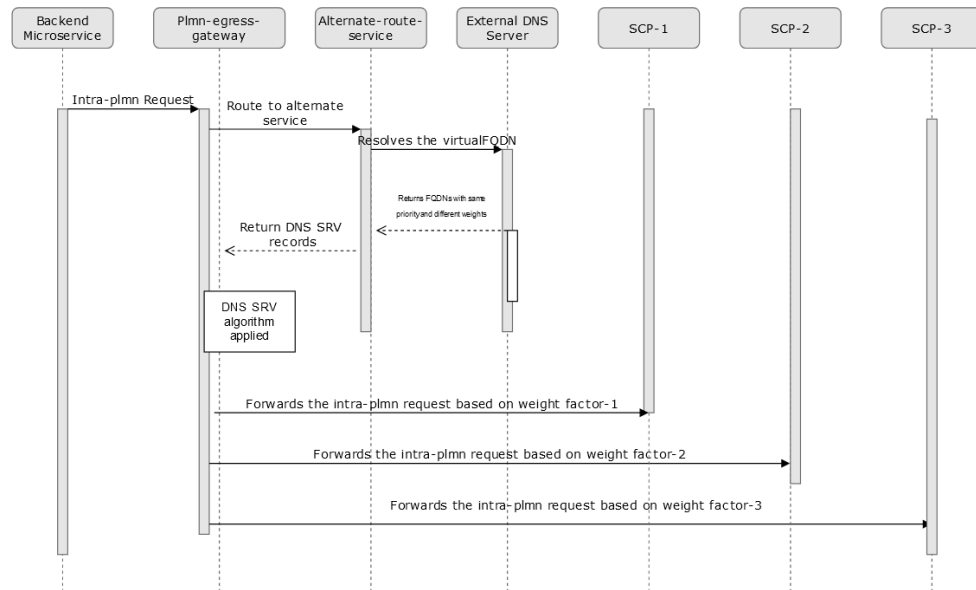
Note:

In Step 2, if Static SCP instances are configured instead of virtual FQDN, then Step 3 & Step 4 are skipped and Steps 5 through 9 are followed.

Weight Based Routing or Load Sharing

In weight based routing (Load Sharing) , virtual FQDN configured for SCP instances with same priorities and different weights.

Figure 3-79 Weight Based Routing or Load Sharing



1. PLMN Egress Gateway (EGW) receives a request from the backend microservice.
2. PLMN EGW maps the request to a route configured with virtual FQDN of SCP. Since the route is configured with virtual FQDN of SCP, the virtual FQDN is resolved to the list of alternate FQDNs.
3. PLMN EGW gets the list of FQDNs from cache data. If cached data has not expired, Step 3 (a) and Step 4 are skipped. If the cache has expired data in it, EGW continues with the Step 3 (a).
 - a. EGW calls the AlternateRoute Service to resolve virtual FQDN to alternate FQDNs and puts it in the cached memory.
4. PLMN EGW performs the DNS-SRV query using the virtual FQDN and scheme.
 - a. EGW caches the resolved virtual FQDN details.
5. PLMN EGW selects the SCP instances based on the weight factor because priorities are same, and it load shares the traffic to SCP instances in proportion to the weight factor defined.
6. The traffic is redistributed to next available peer if the primary SCPs are unavailable.
7. Based on the error codes configured in the route, EGW selects the SCP instance with next priority to initiate a call.
 - a. PLMN EGW repeats this step for all available SCP instances in the order of their priority as per the number of reroute attempts configured by the operator.

8. PLMN EGW receives the response.
9. PLMN EGW forwards the response to the backend microservice as it is received.

Note:

In Step 2, if Static SCP instances are configured instead of virtual FQDN, then Step 3 and Step 4 are skipped and Steps 5 through 9 are followed.

Managing Alternate Routing and Load Sharing based on the DNS SRV Record for Home Network Functions

Enable

This section describes the procedure to enable the feature.

This feature is enabled by setting the following helm parameters configurations in the `ocsepp_custom_values_<version>.yaml` file.

```
dnsSrvEnabled: true
alternateRouteServiceEnable: true
dnsSrvFqdnSetting.enabled: false
dnsSrvFqdnSetting.pattern: "_{scheme}._tcp.{fqdn}."
routeConfigMode: REST
```

The following parameter must also be enabled at `nrf-client` section under global parameters:

```
alternateRouteServiceEnable: true
```

The following parameter must also be enabled at `plmn-egress-gateway` section under global parameters:

```
checkAltRouteSvcReady: true
```

For more information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Configure

You can configure the feature using Helm, REST API, and CNC Console.

- **Configure HELM based static FQDNs against Producer NF virtual Host:**

To set static FQDNs for routing, update the configuration as follows:

```
# Static Virtual FQDN Configuration
staticVirtualFqdns:
  - name: https://udr.test.com
    alternateFqdns:
      - target: udr1.test.com
        port: 5060
        priority: 10
        weight: 20
      - target: udr2.test.com
        port: 5060
        priority: 20
        weight: 30
```

Note

- Multiple FQDNs with different priorities and weights can be configured for a single virtual FQDN.
- If the virtual FQDN is not found in the predefined list, the alternate-route service will automatically send a DNS SRV query to the external DNS configured for that domain.

- **Configure using REST API:** Perform the feature configurations as described in Egress Gateway REST APIs in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- **Configure using CNC Console:** Perform the feature configurations as described in the PLMN Egress Gateway section under [Configuring SEPP using CNC Console](#) chapter.

Note

NRF virtual path route is mandatory to route the intra-plmn request to NRF.

Observe

Following are the feature specific metrics:

- oc_fqdn_alternate_route_total
- oc_dns_srv_lookup_total
- oc_alternate_route_resultset
- oc_configclient_request_total
- oc_configclient_response_total
- oc_egressgateway_sbiRouting_http_responses_total

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.2.42 Rate Limiting for Egress Roaming Signaling per PLMN

Overview

In some scenarios, SEPP may be forwarding more traffic towards a particular PLMN ID resulting in the producers of that PLMN ID getting overloaded. To control this excess traffic,

SEPP has been enhanced to provide egress rate limiting based on PLMN ID, which ensures that the core network is not overloaded with traffic. Using PLMN ID to filter the number of messages limits the traffic towards the network. It prevents network congestion and improves the overall network performance.

This feature allows to configure the egress request rate on N32F interface based on the destination PLMN IDs. This controls the messages sent out from the local SEPP to a Remote SEPP.

With this feature, the number of egress requests to one or more PLMN IDs is restricted based on a configured count using token bucket algorithm. The same bucket is applicable to one or more configured PLMN IDs. The messages are counted based on the destination PLMN ID only and not destination Remote SEPP or Remote SEPP Set.

SBI Messages are discarded based on the 3gpp-Sbi-Message-Priority. Only low priority messages are subject to being discarded. The high priority messages are not discarded even if they breach the egress rate limiting priority threshold. The user can configure the threshold for 3gpp-Sbi-Message-Priority to be used for discarding.

PLMN is identified from 3gpp-Sbi-Target-apiRoot header or :authority header. The :authority header is used only in absence of 3gpp-Sbi-Target-apiRoot header.

Token Bucket Algorithm

This feature is implemented using Bucket4j, which uses Token Bucket Algorithm.

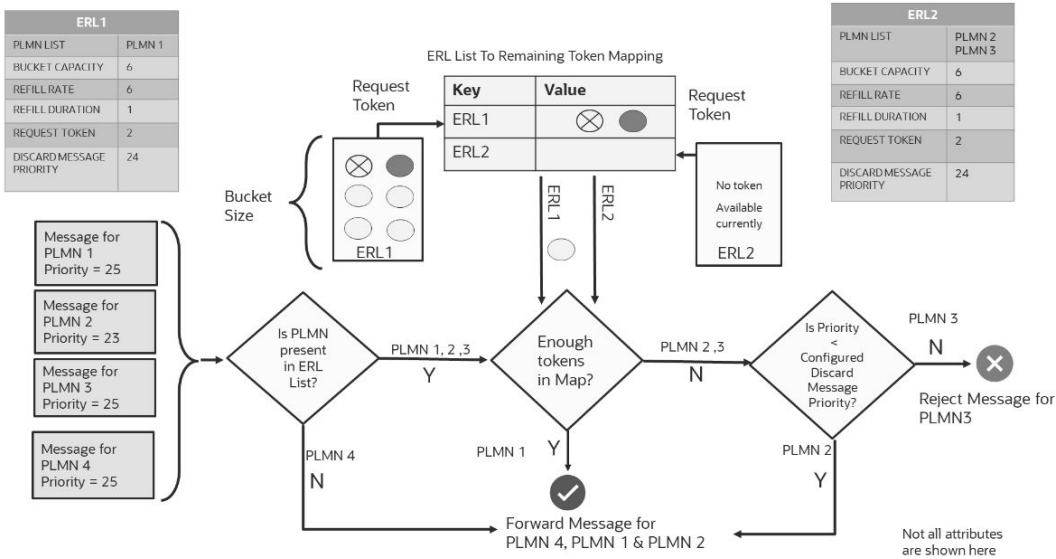
The token bucket algorithm can be described as the following:

- Token bucket algorithm processes messages only if there are tokens in the bucket.
- Bucket is filled with tokens at a user configurable rate during token consumption asynchronously.
- Tokens are removed from the bucket once the messages are processed.
- The following four configurations are required:
 - Bucket Capacity - Bucket size defines the capacity to handle traffic burst.
 - Refill Rate - The number of tokens which should be added to refill the bucket.
 - Refill Duration - It decides how frequently the bucket needs to be refilled.
 - Request Token - It defines the batch size of token requested from the corresponding bucket.
- Tokens divided by refill duration defines the traffic rate.

Detailed Description

The following diagram represents how the egress rate limiting is implemented using token bucket algorithm.

Figure 3-80 Token Bucket Algorithm



The above diagram represents the egress rate limiting for the messages received from four different PLMNS, that is, PLMN 1, PLMN 2, PLMN 3, PLMN 4.

In the above diagram, the user has configured the Bucket Capacity, Refill Rate, Refill Duration, and Request Token for the two Egress Rate Limit lists; Egress Rate Limit1 (ERL1) and ERL2. Two separate buckets are created corresponding to each list.

The "ERL List to Remaining Token Mapping" is populated from the corresponding bucket (for ERL List1 and ERL List2) using number of configured request tokens.

The PLMNs are configured as follows:

- PLMN 1 is configured under ERL List 1; the list has 6 tokens and number of request tokens is set to 2. When the request for PLMN 1 arrives, the "ERL List to Remaining Token Mapping" is populated with 2 tokens and the request consumes one of the available tokens and gets forwarded. One token remains in the mapping.
- PLMN 2 and PLMN 3 are configured under ERL List 2 where corresponding mapping or bucket has no token available. When the requests for PLMN 2 and PLMN 3 are raised, as there are no tokens available, the messages are checked for their priority as follows:
 - Message priority for PLMN 2 (configured as 23 in the example) is less than the configured discard message priority threshold for ERL2 (configured as 24), it is a high priority message and thus it gets forwarded.
 - Message priority for PLMN 3 (configured as 25 in the example) is greater than or equal to the configured discard message priority threshold for ERL2 (configured as 24), it is a low priority message and thus it gets rejected by the rate limit feature with the user configured status code and title.
- PLMN 4 is not configured under any of the ERL lists and thus when the request for PLMN 4 arrives, the rate limit feature is considered as false for this request and thus it gets forwarded.

The following table represents the summary of the algorithm:

Table 3-52 Summary

PLMN	Tokens in ERL List	Message Priority	Rejected/Forwarded
PLMN 1	Token available in ERL1's bucket	Ignore priority	Forwarded
PLMN 2	No token available in ERL2's bucket	High	Forwarded
PLMN 3		Low	Rejected
PLMN 4	No corresponding ERL list is configured	Ignore priority	Forwarded

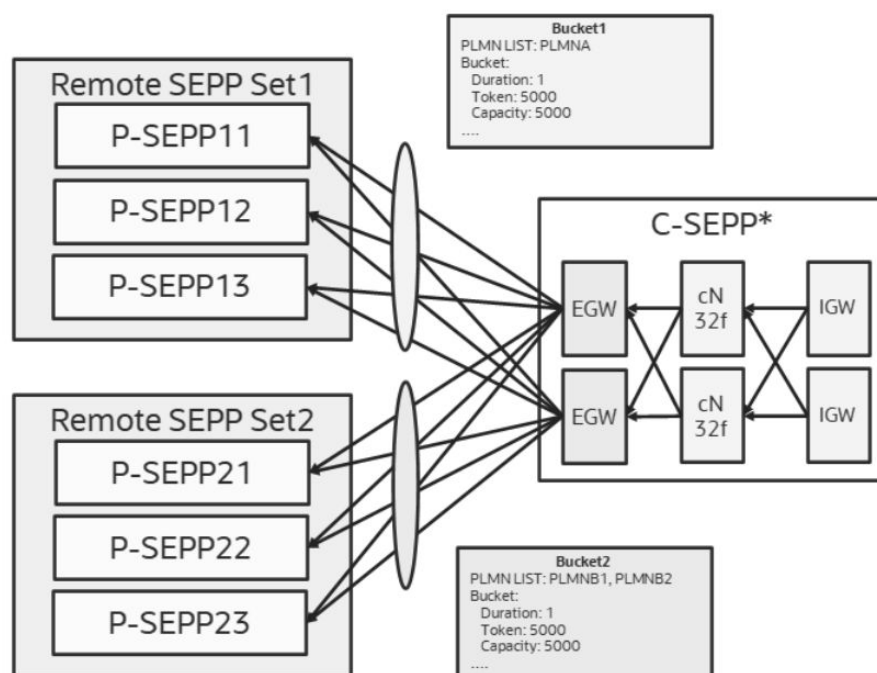
Note

The request message priority is derived as follows:

- Check the request for 3gpp-Sbi-Message-Priority header.
- If not present, check for helm configured route message priority.
- If not present, take the default priority.

Implementation

This feature is implemented at PLMN IGW in SEPP mode at the C-SEPP side. This provides egress rate limiting for all the messages originating from Local SEPP based on PLMN IDs. The diagram shows two separate token buckets applicable for Remote SEPP Set 1 and Remote SEPP Set 2. One Remote SEPP Set can cater to multiple PLMNs.

Figure 3-81 Implementation at C-SEPP

Note

This feature is not applicable to P-SEPP in SEPP mode. The feature is implemented at N32 IGW in Roaming Hub mode.

Managing Rate Limiting for Egress Roaming Signaling per PLMN Feature**Enable**

This section describes the procedure to enable the feature. You can enable the feature using the Helm parameters and REST API or CNC Console.

The feature is disabled by default.

This feature is enabled by setting the following Helm parameters configurations in the Ingress Gateway section:

```
rateLimiting:
  enabled: true
  egressRateLimiter:
    enabled: true
```

Note

- In SEPP mode, set the `egressRateLimiter.enabled` parameter to true in the PLMN Ingress Gateway section of `ocsepp_custom_values_<version>.yaml` file.
- In Roaming Hub mode, set the `egressRateLimiter.enabled` parameter to true in the N32 Ingress Gateway section of `ocsepp_custom_values_roaming_hub_<version>.yaml` file.

For more information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Perform the following procedure to enable this feature using CNC Console or REST API:

CNC Console

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Rate Limiting**.
2. Select **Egress Rate Limiting** which is defined under **Rate Limiting** screen.
3. The **Option and EgressRateLimitingList** appears underneath.
4. Click **Option** the option screen appears at the right pane. The Egress Rate Limiting Feature details are available in the screen.
5. Set **Egress Rate Limiting Enabled** to True on the right pane.

REST API

Set the `egressRateLimitingEnabled` as Enabled at the Egress Rate Limiting REST API.

Note

In CNC Console or REST API, Egress Rate Limiting feature can be enabled and disabled globally for the complete system at run time and this can be enabled or disabled for a particular Egress Rate Limiting List also.

- If the Helm parameters `rateLimiting` and `egressRateLimiter` are set to false, then the feature is globally disabled.
- If feature is disabled on CNC Console **Option** screen or REST API, then the feature is globally disabled.
- If feature is disabled on CNC Console **EgressRateLimitingList** screen or REST API, then Egress Rate Limiting feature is disabled for that ERL list (group of PLMN IDs).

Configure

You can configure the feature using REST API and CNC Console.

- **Configure using REST API:** Perform the feature configurations as described in Egress Gateway REST APIs in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.
- **Configure using CNC Console:** Perform the feature configurations as described in [Configuring SEPP using CNC Console](#).

Observe

Following are the feature specific metrics:

- `oc_ingressgateway_plmn_egress_ratelimit_total` (Added with six different metric filter status)

Following are the feature specific KPIs:

- Average Number of Messages Rejected for a Particular PLMN
- Average Number of Messages Accepted for a Particular PLMN
- Average Number of Messages for which Feature not Applied
- Average of all Messages by Status
- Average Number of Messages Rejected per PLMN
- Average Number of Messages Accepted per PLMN

For more information about metrics and KPIs, see [SEPP Metrics](#) and [SEPP KPIs](#) sections.

Maintain

If you encounter alerts at system or application levels, see [SEPP Alerts](#) section for resolution steps.

In case the alert still persists, perform the following:

1. **Collect the logs and Troubleshooting Scenarios:** For more information on how to collect logs and troubleshooting information, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.

2. **Raise a service request:** See [My Oracle Support](#) for more information on how to raise a service request.

3.3 Deployment Support

The following are the deployment support functionalities of SEPP:

3.3.1 Helm Validation Framework

SEPP supports helm validation framework enhancement. This enhancement is supported to validate the values in a `custom_values.yaml` file with JSON schemas.

It allows the user to perform the following validations:

Table 3-53 Types of Validation checks

Types of Validation checks	Example
Requirement checks	An <code>API_KEY</code> environment variable is set
Type validation	The image tag is a string such as "1.5" and not the number 1.5
Range validation	The value for a CPU utilization percentage key is between 1 and 100
Constraint Validation	The pullPolicy is IfNotPresent, Always, or Never. The URL has the format <code>http(s)://<host>:<port></code> <code>custom_values.yaml</code>

The schema is automatically validated while running the following commands:

- `helm install`
- `helm upgrade`
- `helm lint`
- `helm template`

With Helm 3, user can create a file called `values.schema.json` which allows to set value requirements and constraints in a single location.

The `values.schema.json` file is written using the JSON Schema vocabulary. According to json-schema.org, JSON Schema is a vocabulary that allows you to annotate and validate JSON documents.

Example for `values.schema.json` file.

```
{
  "$schema": "http://json-schema.org/draft-07/schema",
  "required": [
    "image",
    "serviceType",
    "port"
  ],
  "properties": {
    "image": {
      "type": "object",
      "required": [
```



```

        "repository",
        "tag"
    ],
    "properties": {
        "repository": {
            "type": "string"
        },
        "tag": {
            "type": "string"
        }
    }
},
"serviceType": {
    "type": "string",
    "enum": ["ClusterIP", "LoadBalancer"]
},
"port": {
    "type": "integer",
    "minimum": 8080
}
}
}

```

This example provides a schema for the values `image.repository`, `image.tag`, `serviceType`, and `port`. This can be seen by reviewing the keys under the **properties** keyword.

This example also uses the **required**, **type**, **enum**, and **minimum** keywords, which are explained in the following sections.

Required

The **required** keyword is used to fail chart rendering if specific values are not provided.

Example:

```

"required": [
    "image",
    "serviceType",
    "port"
],

```

The values listed ("image", "serviceType", and "port") must be provided in order to install or upgrade the chart. If you are try to install this chart without any of these values, you get the following error:

```

$ helm template validation-test test-chart
Error: values don't meet the specifications of the schema(s) in the following
chart(s):
test-chart:
- (root): image is required
- (root): serviceType is required
- (root): port is required

```

Validating User Input against Constraints

JSON Schema provides several different keywords which can be used for validating user inputs against specific constraints. One common keyword is **type**, which is used to ensure that the input is a String, Integer, Boolean, Object, or another type.

```
"image": {  
  "type": "object",
```

```
- image: Invalid type. Expected: object, given: string
```

Another useful keyword to validate user input is **enum**. This keyword is used for validating against a list of supported inputs. In the values.schema.json example, enum is used to restrict the serviceType to "ClusterIP" and "LoadBalancer":

```
"serviceType": {  
  "type": "string",  
  "enum": ["ClusterIP", "LoadBalancer"]  
},
```

If the wrong serviceType is provided, you can see the following error:

```
$ helm template validation-test test-chart --set serviceType=NodePort  
Error: values don't meet the specifications of the schema(s) in the following  
chart(s):  
test-chart:  
- serviceType: serviceType must be one of the following: "ClusterIP",  
"LoadBalancer"
```

minimum. This keyword is a good way to ensure that integer values are not set below a certain threshold. The below example shows how you can set a minimum value for a port number:

```
"port": {  
  "type": "integer",  
  "minimum": 8080  
}
```

If you attempt to install the chart with a port number smaller than 8080, you'll see the following error:

```
$ helm template validation-test test-chart --set port=8000  
Error: values don't meet the specifications of the schema(s) in the following  
chart(s):  
test-chart:  
- port: Must be greater than or equal to 8080/1
```

There are other keywords which can be used for validating input, including (but not limited to):

- **maximum**: Set a maximum threshold for a numeric value
- **pattern**: Set a regex that a value must conform to
- **additionalProperties**: Determines if the user can provide values that are not defined in the values.schema.json file. By default, users can provide any additional value without

restriction, but this would not impact chart rendering if the additional value isn't used in your templates.

Note

The helm validation framework is enabled by default for helm v3. To disable the feature, the user has to delete `values.schema.json` file from the charts folder.

3.3.2 Automated Testing Suite Support

SEPP provides Automated Testing Suite for validating the functionalities. Through Automated Testing Suite (ATS), Oracle Communications aims at providing an end-to-end solution to its customers for deploying and testing the 5G NFs. Refer to *Oracle Communications Automated Testing Suite User Guide* for more information.

4

Configuring SEPP using CNC Console

This chapter describes how to configure different services in SEPP using Oracle Communications Cloud Native Configuration (CNC) Console.

4.1 CNC Console Interface

You can use the SEPP integrated with CNC Console after logging in to the CNC Console application. To successfully log in to the CNC Console, you need to make the following updates to the hosts file available at the **C:\Windows\System32\drivers\etc** location.

1. In the Windows system, open the hosts file in the notepad as an Administrator and append the following set of lines at the end:

```
<CNCC Node IP> cncc-iam-ingress-gateway.cncc.svc.cluster.local  
<CNCC Node IP> cncc-core-ingress-gateway.cncc.svc.cluster.local
```

For example:

```
10.75.212.88 cncc-iam-ingress-gateway.cncc.svc.cluster.local  
10.75.212.88 cncc-core-ingress-gateway.cncc.svc.cluster.local
```

Note

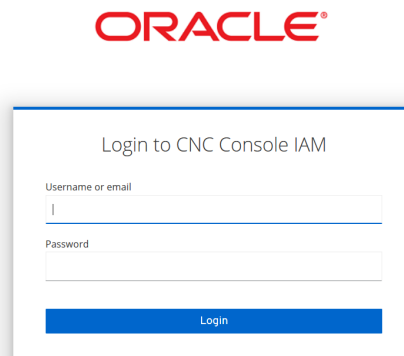
The IP Address in the above lines may change when deployment cluster changes.

2. Save and close the hosts file.
Ensure that a CNC user and password are created before logging into the CNC Console. For more information on how to create a CNC Console user and password, see *Oracle Communications Cloud Native Core Console Installation and Upgrade Guide*.

Log in to CNC Console

The procedure to log in to the CNC Console is as follows:

1. Open any browser.
2. Enter the URL: **http://<host name>:<port number>**. The **Login** screen appears:

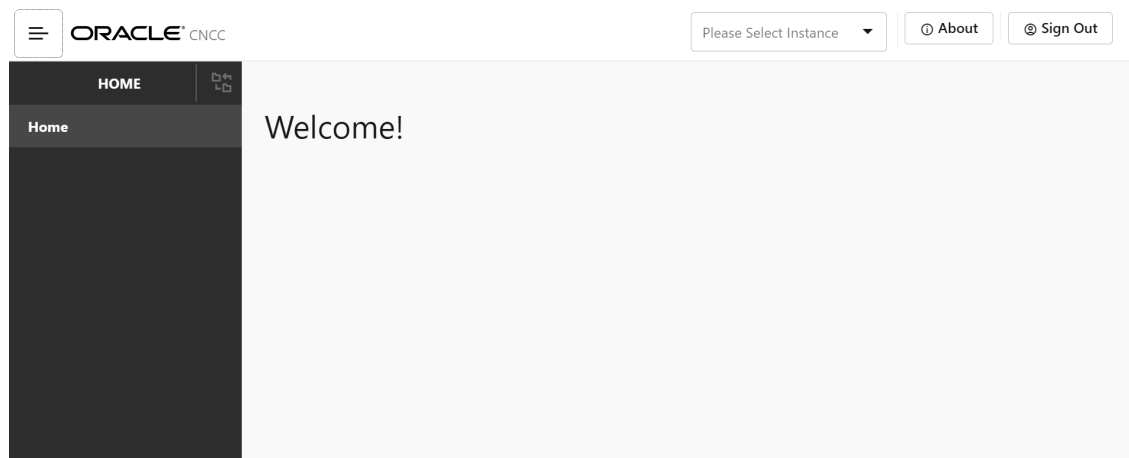
Figure 4-1 Login Screen

The login screen features the ORACLE® logo at the top. Below it is a white box with a blue border containing the text "Login to CNC Console IAM". Inside this box are two input fields: "Username or email" and "Password". At the bottom of the box is a blue button labeled "Login".

Note

<host name> is *cncc-iam-ingress-ip* and <port number> is *cncc-iam-ingress-port*.

3. Enter the valid credentials.
4. Click **Login**. The Welcome screen of CNC Console interface appears.

Figure 4-2 Welcome Page of the CNC Console

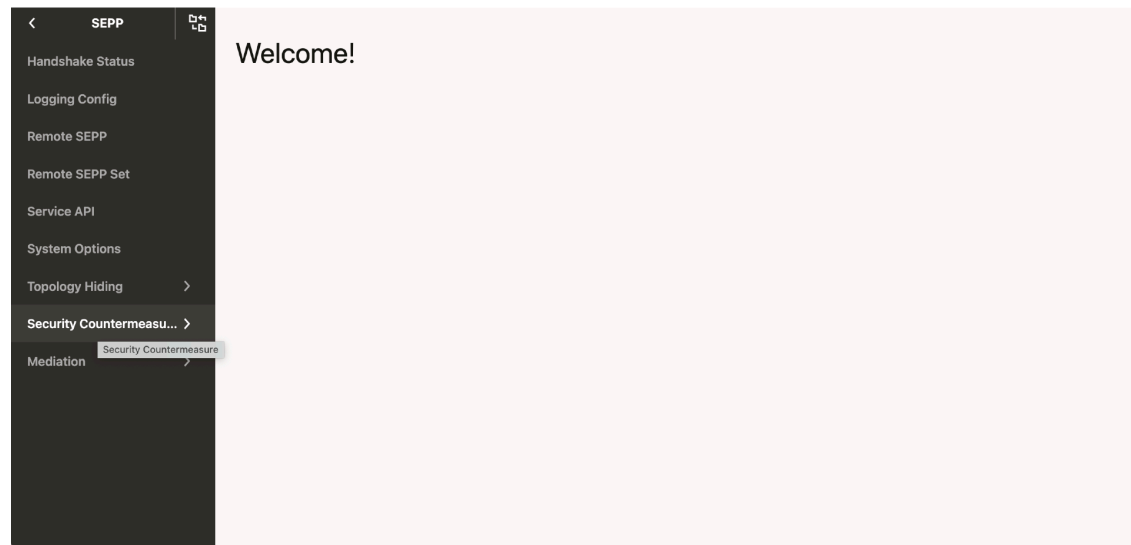
Select the required NF instance from the **Please Select Instance** drop-down list. The left pane displays the selected network function and on clicking the network function the corresponding APIs and configurations appears underneath.

4.2 SEPP Configuration

This section describes how to configure different SEPP features and services using CNC Console.

On selecting SEPP from the drop-down list, the following screen appears:

Figure 4-3 SEPP Welcome Screen



4.2.1 Handshake Status

Handshake Status Rest API returns the handshake status corresponding to each Remote SEPP Name.

Perform the following procedure to view the Handshake Status:

1. From the left navigation menu, navigate to SEPP and then click **Handshake Status**. The list of all the handshake status corresponding to each SEPP Name appears on the right pane.

The parameters are:

Table 4-1 Handshake Status Parameters

Parameter Name	Mandatory(M) / Optional(O) Conditional(C)	Datatype	Description
Remote SEPP	M	String	Name of Remote SEPP
State	M	Enum	N32F Context State
3GPP SBI Target API Root Header Supported	M	Boolean	Indicates whether 3GPP SBI Target API Root Header Supported or not by Remote SEPP.

Table 4-1 (Cont.) Handshake Status Parameters

Parameter Name	Mandatory(M) / Optional(O) Conditional(C)	Datatype	Description
Local PLMN ID(s)	M	Object	List of Local PLMN ID(s) supported
Handshake Reinit Status	M	Enum	Status of Reinitiated Handshake when changing local PLMN ID(s) or editing Remote SEPP configuration.
HandshakeTimestamp	M	Time format	This parameter displays time of handshake and time is updated if handshake is reinitiated.

Note

Possible handshake states are CAPABILITY_EXCHANGE_STATE and N32F_ESTABLISHED_STATE.

- CAPABILITY_EXCHANGE_STATE - Handshake initiated with the remote SEPP.
- N32F_ESTABLISHED_STATE - Handshake completed and TLS connection is established with remote SEPP.

4.2.2 Logging Config

Logging Config allows the user to configure the application based log level and package based log level.

The Logging Config can be configured into SEPP mode and Roaming Hub mode. To enable the SEPP mode, set the `operationMode` flag to true. To enable the IPX mode, set the `operationMode` flag to false. In Roaming Hub mode, the user cannot enable or disable the log level for `nfmanagement` and `nfdiscovery` services.

Perform the following procedure to configure the log levels:

1. From the left navigation menu, navigate to **SEPP** and then click **Logging Config**.
2. On selecting **Logging Config**, the list of all application logs, package logs, and their levels configured in the system appear on the right pane.
3. Click **Edit** icon to modify the log level. The page is enabled for modification.
4. Click **View** to view the details of the log level List.

Table 4-2 Logging Config Parameters

Parameter Name	Description	Details
ServiceType	This is a mandatory parameter. Name of the Common Service - N32 Egress Gateway, N32 Ingress Gateway, PLMN Egress Gateway, PLMN Ingress Gateway.	Data type: Enum

Table 4-2 (Cont.) Logging Config Parameters

Parameter Name	Description	Details
Application Log Level	This is a mandatory parameter. Log level for the application	Data type: Enum
Package Log Level	This is a mandatory parameter. Log level of each corresponding packages. Example: For Package root, the loglevel can be ERROR.	Data type: String

The supported log levels are ERROR, WARN, INFO, DEBUG, and TRACE.

Table 4-3 Log levels

Log Level	Description
ERROR	Designates error events that might still allow the application to continue running.
WARN	Designates potentially harmful situations.
INFO	Designates informational messages that highlight the progress of the application at a coarse-grained level.
DEBUG	Designates fine-grained informational events that are most useful to debug an application.
TRACE	Designates finer-grained informational events than the DEBUG log level.

Configuring Logging Level for SEPP Microservices

Perform the steps below to configure the Logging Level for SEPP Microservices:

The Logging Level Configuration page allows users to view and edit log level details at both the application and package levels for each service.

- From the left navigation menu, navigate to **SEPP** and then click **Logging Level Configuration**.
- On selecting **Logging Level Configuration**, the list of all application logs, package logs, and their levels configured in the system appear on the right pane.
The Logging Level Configuration page displays the following log level details:
 - Service:** Displays the name of the SEPP microservice.
 - Application Log Level:** Shows the log level of the microservice.
 - Additional Error Logging:** This flag enables or disables the error logging feature, depending on whether it is set to "enable" or "disable".
- Click **Edit** icon to modify the log level. The page is enabled for modification.
- Select a **Service Type** from the drop-down list to choose one of the following NF Service types:
 - n32-ingress-gateway
 - n32-egress-gateway
 - plmn-ingress-gateway
 - plmn-egress-gateway

- alternate-route
- nrf-client-nfdiscovery
- nrf-client-nfmanagement
- nf-mediation
- config-mgr-svc
- cn32c-svc
- pn32c-svc
- cn32f-svc
- pn32f-svc
- app-info
- perf-info
- coherence

5. Select the **Additional Error Logging** status from the drop-down list.
6. Click Save.

The following table outlines which microservices are supported by the logging enhancements:

Table 4-4 Supported Microservices

Microservice	Description	Whether Enhanced Error Logging Supported.
n32-ingress-gateway	Specifies the log level options for n32-ingress-gateway microservice	Yes
n32-egress-gateway	Specifies the log level options for n32-egress-gateway microservice	Yes
plmn-ingress-gateway	Specifies the log level options for plmn-ingress-gateway microservice	Yes
plmn-egress-gateway	Specifies the log level options for plmn-egress-gateway microservice	Yes
alternate-route	Specifies the log level options for alternate-route microservice	Yes
nrf-client-nfdiscovery	Specifies the log level options for nrf-client-nfdiscovery microservice	Yes
nrf-client-nfmanagement	Specifies the log level options for nrf-client-nfmanagement microservice	Yes
nf-mediation	Specifies the log level options for nf-mediation microservice	No
config-mgr-svc	Specifies the log level options for config-mgr-svc microservice	No
cn32c-svc	Specifies the log level options for cn32c-svc microservice	Yes
pn32c-svc	Specifies the log level options for pn32c-svc microservice	Yes

Table 4-4 (Cont.) Supported Microservices

Microservice	Description	Whether Enhanced Error Logging Supported.
cn32f-svc	Specifies the log level options for cn32f-svc microservice	Yes
pn32f-svc	Specifies the log level options for pn32f-svc microservice	Yes
app-info	Specifies the log level options for app-info microservice	No
perf-info	Specifies the log level options for perf-info microservice	No
coherence	Specifies the log level options for coherence microservice	No

4.2.3 Remote SEPP

Remote SEPP returns all the configured Remote SEPP profiles.

Perform the following procedure to configure the Remote SEPP:

1. From the left navigation menu, navigate to SEPP and then click **Remote SEPP**. The list of all the Remote SEPPs configured along with the parameters appear on the right pane.
2. Click **Edit** icon to modify a specific parameter. The page is enabled for modification.

Note

- **SEPP Name** and **FQDN** cannot be edited.
- **N32F FQDN**, **N32F IP Address**, and **N32F Port** values must be added in the Remote SEPP to enable separate port configurations for n32c and n32f on the Egress routes feature.
- **Virtual Host** value must be added in the Remote SEPP to enable load sharing among multiple Remote SEPP nodes feature.
- Remote SEPP also means an intermediary supporting hop-by-hop TLS (Roaming Hub Deployment Model).

The parameters are:

Table 4-5 Remote SEPP Parameters

Parameter Name	Mandatory(M)/Optional(O)/Conditional(C)	Data Type	Description
Name	M	String	Name of the Remote Sepp
PLMN ID(s)	M	Object	List of PLMN ID - MCC and MNC
Domain	O	String	Domain for routing.

Table 4-5 (Cont.) Remote SEPP Parameters

Parameter Name	Mandatory(M)/Optional(O)/Conditional(C)	Data Type	Description
SEPP FQDN	M	String	Fully Qualified Domain Name for SEPP
Security Capability List	O	List <Security Capability>	Type of security capability supported - TLS and PRINS. Default value: TLS
Is Enabled	O	Boolean	Remote SEPP is True (enabled) or False (disabled). Default value: True
Port	O	Integer	Port for SEPP NF
API Prefix	O	String	API Prefix Default Value: ""
API Version	O	String	API Version
priority	O	Integer	This parameter is currently not in use and is reserved for future release.
Remote SEPP IP Address	O	String	This is the Remote SEPP IP Address. The IP Address can be IPv6 or IPv4. If Remote SEPP IP is provided, it will be added in authority header while sending HTTP2 headers towards Remote SEPP. If Remote SEPP IP is not provided, FQDN is resolved to the corresponding IP endpoint using DNS. Hence, DNS configuration should be present for the FQDN. In this case, the authority header contains FQDN. Note: If IP is provided, the Server Name Indication (SNI) field in the TLS handshake message will not be populated with the SEPP FQDN. For SNI, only SEPP FQDN should be present and not the remote SEPP IP address.
sanValidationRequired	O	Boolean	San validation is enabled for incoming N32C handshake request or not. Default Value: True
N32F FQDN	C	String	This is a conditional parameter. This parameter describes the FQDN used for the forwarding plane. This is a mandatory parameter if user wants to use port segregation feature and configure different control plane and forward plane.

Table 4-5 (Cont.) Remote SEPP Parameters

Parameter Name	Mandatory(M)/Optional(O)/Conditional(C)	Data Type	Description
N32F IP Address	O	String	This parameter describes the IP Address used for the forwarding plane. This will be configured only if control plane and forwarding plane configuration are different. The IP Address can be IPv6 or IPv4. Note: If IP is provided, the Server Name Indication (SNI) field in the TLS handshake message will not be populated with the SEPP FQDN. For SNI, only SEPP FQDN should be present and not the remote SEPP IP address.
N32F Port	C	String	This is a conditional parameter. This parameter describes the port used for the forwarding plane. This is a mandatory parameter if user wants to use port segregation feature and configure different control plane and forward plane.
Virtual Host	O	String	This parameter describes the virtual FQDN used for the load sharing between the remote SEPPs.
HealthAPIPath	C	String	Conditional parameter. If present, HealthAPIMethod also needs to be present. default value is null. URI sent as part of proactive monitoring of remote peer.
HealthAPIMethod	C	String	Conditional parameter. If present, HealthAPIPath also needs to be present. default value is null. URI sent as part of proactive monitoring of remote peer. The healthApiMethod parameter supports only GET and OPTIONS.
partnerProfileFqdnTowardsRoamingPartner	C	String	This is the external FQDN of the hosted partner exposed towards the roaming partner. Note: Applicable for Hosted SEPP mode.
fqdnExposedTowardsPartnerProfile	C	String	This is the internal FQDN of the hosted partner. Note: Applicable for Hosted SEPP mode.
Hosted Partner SEPP	M	boolean	This is a mandatory parameter for hosted SEPP mode. This parameter will be set to true when peer associated is a hosted partner. The value will be false when peer associated is a roaming partner.

- Click **Delete** icon to delete a specific Remote SEPP Profile.

4. Click **Add** from the top right side to add a new Remote SEPP Profile.
5. Click **Save**.

Note

Name, **SEPP FQDN**, and **PLMN ID(s)** are mandatory parameters.

4.2.4 Remote SEPP Set

Remote SEPP Set (RSS) allows the user to configure the Remote SEPP Sets.

Perform the following procedure to configure the Remote SEPP Set:

1. From the left navigation menu, navigate to SEPP and then click **Remote SEPP Set**. On selecting **Remote SEPP Set**, the list of all the Remote SEPP Sets configured in the system appears on the right pane.
2. Click **Edit** icon to modify the parameters. The page is enabled for modification.
3. Click **Add** to add a new Remote SEPP Set. PrimarySepp, SecondarySepp, and TertiarySepp can be added.

Note

- One Remote SEPP Set will be created per PLMN and this set can have up to three producer SEPPs sharing same PLMN. The three producer SEPPs can be configured in this set based on their priority as either Primary, Secondary, or Tertiary.
- The user must configure atleast one of the Primary, Secondary, or Tertiary parameters.

Note

Set the Hosted Partner SEPP flag to true for Hosted SEPP mode.

The parameters are:

Table 4-6 Remote SEPP Set Parameters

Parameter Name	Description	Details
Name	This is a mandatory parameter. Name of Remote SEPP Set.	Data Type: String
Primary SEPP	This is a conditional parameter. The name of Primary SEPP configured in the Remote SEPP Set, which is treated as the primary route in the forward plane.	Data Type: String

Table 4-6 (Cont.) Remote SEPP Set Parameters

Parameter Name	Description	Details
Secondary SEPP	This is a conditional parameter. The name of Secondary SEPP configured in the Remote SEPP Set, which is treated as the secondary route in the forward plane.	Data Type: String
Tertiary SEPP	This is a conditional parameter. The name of Tertiary SEPP configured in the Remote SEPP Set, which is treated as the tertiary route in the forward plane.	Data Type: String
CAT 1 Service API Validation	This is a mandatory parameter. Name of the allowed list supported by Remote SEPP Set.	Data Type: String
CAT 2 Network ID Validation	This is a mandatory parameter. It contains Cat 2 network ID Validation Configuration parameter for this RSS.	Data Type: String
Network ID in Header Validation Enabled	This is a mandatory parameter. A boolean value to enable and disable the network ID in the header validation at RSS level. True indicates that the validation is enabled. False indicates that the validation is disabled.	Data Type: Boolean
Network ID in Body Validation Enabled	This is a mandatory parameter. A boolean value to enable and disable the network ID in body validation at RSS level. True indicates that the validation is enabled. False indicates that the validation is disabled.	Data Type: Boolean
Network ID Validation List Name	This is a mandatory parameter. It contains the network id validation list name supported by the particular Remote SEPP Set.	Data Type: String
Ingress Rate Limiting	This is a mandatory parameter. It contains Ingress Rate Limiting Configuration parameters for this RSS.	Data Type: Object
RSS Ingress Rate Limiting Enabled	This is an optional Parameter. A boolean value to enable and disable the Ingress Rate Limiting feature at the RSS level. True indicates that the feature is enabled. False indicates that the feature is disabled.	Data Type: Boolean
Bucket Capacity	This is a conditional parameter. It is the integer number for setting the bucket capacity for token bucket algorithm. Bucket size defines the capacity to handle traffic burst.	Data Type: Integer
Refill Rate	This is a conditional parameter. This parameter defines the number of tokens to be added to refill the bucket.	Data Type: Integer

Table 4-6 (Cont.) Remote SEPP Set Parameters

Parameter Name	Description	Details
Refill Duration	This is a conditional parameter. This parameter indicates the duration to decide how frequently to refill the bucket.	Data Type: Integer
Request Token	This is a conditional parameter. This parameter defines the pre loaded tokens to refill the bucket.	Data Type: Integer
Error Configuration	This is a mandatory parameter. This parameter contains the error configurations for Ingress Rate Limiting feature limited to the particular RSS.	Data Type: Object
Action	This is a conditional parameter. The action can be Reject or Forward. By default, Reject is selected and SBI request is rejected with the user configured error configuration when the number of SBI requests is above the configured limit.	Data Type: String
Status code	This is a conditional parameter. Error status code to be used in the Error Response for discarding the SBI requests when the number of SBI requests is above the configured limit.	Data Type: Integer Default Value: 429
Title	This is a conditional parameter. Error Title to be used in the Error Response for discarding the SBI requests when the number of SBI requests is above the configured limit.	Data Type: String Default Value: Too many requests sent
Mediation	This is a mandatory parameter. It contains mediation configuration parameters for the particular RSS.	Data Type: Object
Trigger Rule List Name	This is a mandatory parameter. This parameter contains the trigger list name supported by the particular Remote SEPP Set.	Data Type: String
SoR	This is a mandatory parameter. It contains SoR Configuration parameters for this RSS.	Data Type: Object
SoR Trigger Rule Enabled	This is a mandatory parameter. Enables or disables the SoR trigger rules.	Data Type: Boolean
SoR Trigger Rule List Name	This is a mandatory parameter. SOR trigger rule list name supported by the particular Remote SEPP Set.	Data Type: String
PLMN ID(s)	This is a mandatory parameter. PLMN ID(s) supported by Remote SEPPs in Set. PLMN ID includes MCC and MNC.	Data Type: Integer
Allowed List Name	This is a mandatory parameter. Allowed list name supported by Remote SEPP Sets.	Data Type: string

Table 4-6 (Cont.) Remote SEPP Set Parameters

Parameter Name	Description	Details
Trigger Rule List Name	This is a mandatory parameter. Trigger List Name supported by Remote SEPP Sets.	Data Type: string
Message Validation on Body Enabled	This is a mandatory parameter. A Boolean value to enable or disable the message validation on body at RSS level. True indicates that the feature is enabled. False indicates that the feature is disabled.	Data Type: boolean
Message Validation On Query Parameters Enabled	This is a mandatory parameter. A Boolean value to enable and disable the message validation on query parameters at RSS level. True indicates that the message validation on query parameters is enabled. False indicates that the message validation on query parameters is disabled.	Data Type: boolean
Message Validation List	This is a mandatory parameter. Message validation list name supported by Remote SEPPs Set.	Data Type: String
Allowed List Name	This is a mandatory parameter. Allowed List Name supported by Remote SEPPs Set.	Data Type: String
Trigger Rule List Name	This is a mandatory parameter. Trigger List Name supported by Remote SEPPs Set.	Data Type: String
Previous Location check Enabled	This is a conditional parameter. A Boolean value to enable and disable the Cat 3 - Previous Location check feature at RSS Level. True indicates that the feature is enabled. False indicates that the feature is disabled.	Data Type: Boolean
Previous Location Trigger List	This is a conditional parameter. Previous Location Trigger List Name supported by Remote SEPP Set.	Data Type: String
Enabled	This is a mandatory parameter. A boolean value to enable or disable the 'Originating Network ID Header Support' feature at Remote SEPP Set level. True indicates that the feature is enabled. False indicates that the feature is disabled.	Data Type: Boolean

Table 4-6 (Cont.) Remote SEPP Set Parameters

Parameter Name	Description	Details
Supported Header Name	This is a mandatory parameter. User needs to provide one of the two header's values from the drop-down to select the Remote SEPP Set supported header name. '3gpp-Sbi-Originating-Network-Id' is selected by default. Note: - When 3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id header is missing in incoming request at C SEPP then this configured header at Remote SEPP Set level is added in the request. - This configuration is not used by producer SEPP.	Data Type: String
PLMN ID [mcc-mnc] from below list	This is an optional Parameter. User needs to provide a single remote PLMN ID (mcc-mnc) from one of the supported PLMN IDs by this RSS. Note: User can only configure this after the RSS have been created and have been assigned the PLMN IDs. While creating the RSS, user needs to provide this as empty value.	Data Type: String
Verbose Error Response Enabled	This is an optional parameter. A boolean value to enable or disable the security countermeasure features error responses verbosity at Remote SEPP Set level. True indicates verbosity enabled. False indicates verbosity disabled.	Data Type: Boolean Default Value: False
Cat 3 - Time Location Check - Unauthenticated Location Enabled	This is a mandatory parameter. A boolean value to enable or disable the Cat-3 Time Location Check Feature at the global level. True indicates that the feature is enabled. False indicates that the feature is disabled.	Data Type: Boolean Default Value: False

Table 4-7 Hosted SEPP Mode

Attribute	Description	Details
Hosted Partner SEPP	This is a mandatory parameter for hosted SEPP mode. This parameter will be set to true when peer associated is a hosted partner. The value will be false when peer associated is a roaming partner.	Data Type: boolean Default Value: False

Table 4-7 (Cont.) Hosted SEPP Mode

Attribute	Description	Details
Allowed Producer Remote SEPP Sets	This is a conditional parameter. List of allowed producer Hosted SEPP Sets that can send or receive the N32F traffic for the specific peer. Note: Applicable for Hosted SEPP mode.	Data Type: Array

Note

- Remote SEPP Sets are created per PLMN. Every Remote SEPP Set consists of minimum 1 and maximum 3 Remote SEPP in primary, secondary, tertiary mode, and is used for N32F SBI message routing.
- The user must configure atleast one of the primary, secondary, or tertiary parameters.

4.2.5 Service APIs

The Service API is used to add, view, and delete the complete set of REST APIs supported by SEPP.

1. From the left navigation menu, navigate to SEPP and then click **Service APIs**. The list of all the REST APIs supported by SEPP appears on the right pane.
2. Click **Add** from the top right side to add or delete the supported REST APIs.

The parameters are:

Table 4-8 Service APIs

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
Resource URIs	M	String	5G Service Based Resource URI
HTTP Method	M	Enum	Resource URI Method (GET,POST,PUT,PATCH,DELETE,OPTIONS, HEAD)
Regular Expression	M	String	Regular Expression for matching Resource URI

4.2.6 System Options

The **System Options** and **Remote SEPP Set** option allows the user to enable and configure the Hosted SEPP feature.

Note

This configuration is not supported by Hosted SEPP deployment from Release 25.2.200 onwards.

Perform the following procedure to do the Hosted SEPP configurations:

1. From the left navigation menu, navigate to SEPP and then click **System Options**.
2. Click **Allowed P-RSS Validation Options** under **System Options**, the **System Options** page appears on the right pane.
3. Set **Enable Allowed P-RSS Validation** to **True** to enable the Hosted SEPP feature.
4. From the left navigation menu, navigate to SEPP and then click **Remote SEPP Set** option for configuring Hosted SEPP feature.
5. Click **Edit** icon to modify the specific parameter. User can add or delete **Allowed Producer Remote SEPP Sets**. This is the list of Remote SEPP Sets that are allowed for communication with Consumer Remote SEPP Set.

The parameters are:

Table 4-9 System Options Parameter

Parameter Name	Mandatory(M) / Optional(O) Conditional(C)	Datatype	Description
Enable Allowed P-RSS Validation	O	Boolean	A Boolean value to enable and disable the Hosted SEPP Feature. True indicates Enabled False indicates Disabled. The feature is disabled (set to false) by default.

Table 4-10 Remote SEPP Set Parameters

Parameter Name	Mandatory(M) / Optional(O) Conditional(C)	Datatype	Description
Allowed Producer Remote SEPP Sets	O	List of String	List of Remote SEPP Sets which are allowed for communication with Consumer Remote SEPP Set

4.2.7 NIF

The **NIF** option is used to enable and configure the NIF feature and rejected message copy functionality.

Perform the following procedure to do the **NIF** feature configurations:

1. From the left navigation menu, navigate to **SEPP** and then click **NIF**.
2. Click **NIF Option** under **NIF**. The **System Option** page appears at the right pane.

3. Click **Edit** icon to modify the Option. The **Edit System Option** page appears.
4. Set the **Enable NIF route selection** to True.

Note

The NRF Query Parameters screen will be disabled until NIF feature is enabled.

The **NIF Option** page parameters are:

Table 4-11 NIF Option Page Parameters

Parameter Name	Data Type	Description
Enable NIF route selection	Boolean	This is a mandatory parameter. This is used to enable or disable the NIF feature. feature. True indicates that the feature is enabled and false indicates that the feature is disabled.

NRF Query Parameters Configurations:

Perform the following procedure to do the NRF Query Parameters Configurations:

1. From the left navigation menu, navigate to **SEPP** and then click **NIF**.
2. Click **NRF Query Parameters** under **NIF**. The **NRF Query Params** page appears at the right pane.
3. Click **Edit** icon to modify the Option. The **Edit NRF Query Params** page appears.
4. User can configure the following parameters:

Table 4-12 NRF Query Parameters

Parameter Name	Data Type	Parameter Description
Query Param	String	This is a mandatory parameter. This parameter indicates the value of queryparameters as defined in 3GPP, which used to perform NF Discovery.
Query Param Value	String	This is a mandatory parameter. This parameter indicates the value of the query parameter set basis which NF is to be discovered.

Perform the following procedure to do the configurations for Rejected Message Copy for NIF feature:

1. From the left navigation menu, navigate to **SEPP** and then click **NIF**.
2. Click **Message Copy Option** under **NIF**, **System Options** page appears on the right pane.
3. Set **Enable Error Message Copy** parameter to true to enable the Rejected Message Copy feature.

- Click **Message Copy Params** under NIF, the **NIF Message Copy Params** page appears on the right pane.
- Click **Edit** from the top right side to update the URI on which the copy of the rejected message is sent towards NIF.

Note

The Message Copy Parameters screen will be disabled until NIF and Message Copy Options features are enabled.

Table 4-13 Message Copy Option Parameters

Parameter Name	Data Type	Description
Enable Error Message Copy	Boolean	This is a mandatory parameter. This is used to enable or disable the rejected message copy feature. True indicates that the feature is enabled and false indicates that the feature is disabled.

Table 4-14 Message Copy Parameters

Parameter Name	Data Type	Parameter Description
URI	String	This is a mandatory parameter. This parameter indicates the value of URI that will be used to send the copy of rejected message towards nif Default value: nnif-sbi-error
API version	String	This is a mandatory parameter. This parameter indicates the API version used to send the copy of rejected message towards nif. Default value: v1

4.2.8 Topology Hiding

Topology Hiding enables or disables the topology hiding feature. The user can configure the topology header, body, and path configurations.

Topology Hiding option allows the user to set Topology Hiding feature as ENABLED or DISABLED and configure the topology options.

Perform the following procedure to enable or disable the Topology Hiding feature:

- From the left navigation menu, navigate to SEPP and then click **Topology Hiding**. Select the **Option**.
- Click Edit icon to modify the Option. The **Edit Option** page appears.
- Set the **Topology Hiding** to True or False.

The parameters are:

Table 4-15 Topology Hiding Parameters

Parameter Name	Datatype	Description
Topology Hiding	Boolean	This is an optional parameter. A Boolean value to enable and disable the Topology Hiding feature. True indicates Enabled False indicates Disabled. The feature is disabled (set to false) by default.

Topology Configuration Options

Topology framework provides the options at the CNC Console screen while processing the request/response json format messages.

Perform the following procedure to configure the message route and Enable Multi PsuedoValue:

1. From the left navigation menu, navigate to SEPP and then select **Topology Hiding**. The **Option** appears underneath the topology hiding on the left menu.
2. Click on **Option**, the **Option** screen appears.
3. User can configure **Action** and **Enable Multiple PsuedoValue**.

Table 4-16 Topology Hiding Parameters

Parameter Name	Datatype	Description
Enable Multiple Pseudo Value	Boolean	This is an optional parameter. This is a boolean field. If set to true, signifies that if more than one actual value exists in the request/response then every actual value occurrence is replaced by unique pseudo value. The value is disabled (set to false) by default.
Action	Enum	This is a mandatory parameter. Action has two possible values FORWARD and REJECT. By default we have FORWARD enabled.
Status Code	Integer	This is a mandatory parameter. User can configure the required HTTP error code when exceptions arise due to the TH operation failures.
Title	String	This is a mandatory parameter. User can configure the required Title when exceptions arise due to the TH operation failures.

Note

Action has two possible values FORWARD and REJECT. By default we have FORWARD enabled.

FORWARD: While any exception occurs processing the message for TH/TUH, the original message is forwarded as if no TH/TUH is enabled and operation should be success.

REJECT: While any exception occurs processing the message for TH/TUH, the original message is dropped with the error body having status code and error description as configured in the CNCC screen.

If in case REJECT is selected and statusCode and Error Description is not given or left empty, then status code is set as 500 and error description as "Internal error" by default.

Note

Enable Multiple PseudoValue is set as false by default and user can set to true to enable the special processing. If the system has many occurrence of same actual value in the request/response then this property gives the flexibility that each same actual value must be replaced with the unique different pseudo value. This can only be possible if we define at least 7 different pseudo values in actual to pseudo mappings as we pick different values from this mappings only. Also if same actual value occurs more than 7 times then there is a possibility of repetition since we only have maximum of 7 distinct values.

Enable Multiple PseudoValue property works on request and response separately. Request and response processing are two different operations and should be treated as the isolated operations.

Pseudo Values

The **Pseudo Values** option appears underneath the Topology Hiding. This Pseudo Values option allows the user to set the pseudo values against an actual value.

Perform the following procedure to set the pseudo values against an actual value:

1. From the left navigation menu, navigate to SEPP and then select **Topology Hiding**. The **Pseudo Values** appears underneath the topology hiding on the left menu.
2. Click on **Pseudo Values**, the list of all the actual values and corresponding pseudo values configured in the system along with their Value Type.
3. Click **Add** to add the actual value and corresponding pseudo values.

Note

Actual Value, **Pseudo Value**, and **Value Type** are mandatory parameters.

Note

If the actual value contains **mnc** and **mcc** values as in **3gpp-sbi-target-apiRoot**, then pseudo values must also contain **mnc** and **mcc**.

4. Click **Save**.

The parameters are:

Table 4-17 Pseudo Value Configuration Parameters

Attribute	Datatype	Description
actualValue	string	This is a mandatory parameter. Refers to the actual FQDN of network functions.
pseudoValues	string	This is a mandatory parameter. Refers to the pseudo value corresponding to a configured actual value of network functions.
value type	Enum	This is a mandatory parameter. Refers to the type of actual and pseudo values. Example: FQDN, NF SERVICE ID, NF SERVICE INSTANCE ID, OTHERS. OTHERS is for the values which do not fit in first three categories.

Header and Body Configurations

1. From the left navigation menu, navigate to SEPP and then select **Topology Hiding**. The **Header** and **Body IE** options appears underneath the topology hiding on the left menu.
2. Click **Header**, the **Header** screen appears on the right pane.
3. Click **Add**, the **Create Header** appears and user can add the header information.
4. User can add the new header parameters.

Note

Header Name, Regular Expression, Trigger Point, and **Operation** are the header parameters.

5. Click **Body IE**, the **Topology Body** screen appears on the right pane.
6. Click **Add**, the **Topology Body Configuration** appears and user can add the body information.

Note

Method, API Resource, Identifier, Regular Expression, Trigger Point, and **Operation** are the body parameters.

7. Click **Save**.

Table 4-18 Header Configuration Parameters

Attribute	Data type	Description
Header Name	String	This is a mandatory parameter. Name of the header
Regular Expression	String	This is a mandatory parameter. Regular Expression for the header
Trigger Point	Enum	This is a mandatory parameter. Request Ingress, Response Ingress, Request Egress, Response Egress
Operation	Enum	This is a mandatory parameter. Topology Hiding or Topology Recovery

Table 4-19 Body IE Configuration Parameters

Attribute	Data type	Description
API Resource	String	This is a mandatory parameter. API Resource that comes from default table.
Identifier	String	This is a mandatory parameter. Body IE Key Identifier.
Regular Expression	String	This is a mandatory parameter. Regular Expression for the Body IE.
Trigger Point	Enum	This is a mandatory parameter. Request Ingress, Response Ingress, Request Egress, Response Egress.
Operation	Enum	This is a mandatory parameter. Topology Hiding /Topology Recovery.
Method	Enum	This is a mandatory parameter. GET/PUT/POST/DELETE/PATCH

Path Configurations

1. From the left navigation menu, navigate to **SEPP** and then select **Topology Hiding**. The **Path IE** appears underneath the topology hiding on the left menu.
2. Click **Path IE**, the **Path IE** screen appears on the right pane.
3. Click **Add**, the **Topology Path Configuration** appears and user can add the path information.
4. User can add the new path parameters.

Note

Method, API Resource, Regular Expression, Trigger Point, and Operation are the path parameters.

- Click **Save**.

Table 4-20 Path IE Configuration Parameters

Attribute	Data type	Description
Method	Enum	This is a mandatory parameter. GET/PUT/POST/DELETE/PATCH
API Resource	String	This is a mandatory parameter. API Resource that comes from default table.
Regular Expression	String	This is a mandatory parameter. Regular Expression for the Path IE.
Trigger Point	Enum	This is a mandatory parameter. Request Ingress, Response Ingress, Request Egress, Response Egress.
Operation	Enum	This is a mandatory parameter. Topology Hiding /Topology Recovery.

4.2.9 Security Countermeasure

The **Security Countermeasure** option is used to enable and configure the Cat-0 SBI Message Schema Validation feature, Cat 1 -Service API Validation feature, Cat 2 – Network ID Validation feature, and Cat 3 – Previous Location Check feature.

The **Service API Allowed List** REST API is used to do the configurations on the allowed list of REST APIs.

4.2.9.1 Nonverbose Error Response

SEPP's security counter measure features have been improved to provide either detailed (verbose) or simplified (nonverbose) error responses, depending on user configurations. The error configurations of the Cat-0, Cat-1, Cat-2, and Cat-3 features are enhanced.

To enable the feature, perform the following steps:

- In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
- Click **Non Verbose Error Measure Configuration** under **Security Countermeasure**, the **Option** appears.
- The Security Counter Measure Nonverbose Configuration feature details are available on the **Option** screen.
- Click **Edit** icon to modify the Option. The **Edit** Option page appears.

5. Set **Verbose Error Response Enabled** to True on the right pane.

Perform the following procedure to configure feature:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Non Verbose Error Measure Configuration** under **Security Countermeasure**, the **Option** appears underneath.
3. Click **Option**, the option screen appears at the right pane. The Security Counter Measure Verbose Configuration feature details are available in the screen.
4. Click **Edit** icon to modify the Option. The Edit Option page appears.
5. Set **Verbose Error Response Enabled** to True on the right pane.
6. Update the "**Title**" to required error title.
7. Update the "**Cause**" to required error cause.
8. Update the "**Detail**" to required error detail.

Table 4-21 Parameters

Parameter Name	Datatype	Mandatory(M) / Conditional(C) Optional(O)	Description
Verbose Error Response Enabled	Boolean	M	A boolean value to enable or disable the security counter measure features error response verbosity at global level True indicates verbosity enabled. False indicates verbosity disabled. Default Value: False
title	String	O	Non verbose error response title for security counter measure features. Its value is 'Rejected' by default.
cause	String	O	Non verbose error response cause for security counter measure features. Its value is 'Unknown' by default.
detail	String	O	Non verbose error response detail for security counter measure features. Its value is 'Server Error' by default.

4.2.9.2 Cat 1 -Service API Validation

The **Security Countermeasure** option is used to enable the Cat 0 - SBI Message Schema Validation feature, Cat 1 -Service API Validation feature, Cat 2 – Network ID Validation feature, and Cat 3 - Previous Location Check feature.

Perform the following procedure to do the **Cat 1 -Service API Validation** configurations:

1. From the left navigation menu, navigate to SEPP and then click **Security Countermeasure**.
2. Click **Cat 1 -Service API Validation** under **Security Countermeasure**, **Option** page appears on the right pane.

3. Set **Security Countermeasure parameter** to **True** to enable the Cat 1 -Service API Validation feature.
4. Click **Service API Allowed List** under **Cat 1 -Service API Validation**, the **Service API Allowed List** page appears on the right pane.
5. Click **Add** from the top right side to add or update the allowed REST APIs and supported methods.

The parameters are:

Table 4-22 Cat 1 -Service API Validation

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
Enable Cat 1 - Service API Validation	O	Boolean	A Boolean value to enable and disable the Cat 1 - Service API Validation feature. true indicates enabled false indicates disabled. The feature is disabled (set to false) by default.

Table 4-23 Service API Allowed List

Parameter Name	Sub Parameter	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
Service API Allowed List Name		M	String	Allowed list name per Remote SEPP Set
N32 Ingress		M	Object	Ingress Direction
	Resource URI	M	String	Resource URI
	HTTP Method	M	Enum	Resource URI Method (GET,POST,PUT,PATCH,DELETE,OPTIONS, HEAD)
N32 Egress		M	Object	Egress Direction
	Resource URI	M	String	Resource URI
	HTTP Method	M	Enum	Resource URI Method (GET,POST,PUT,PATCH,DELETE,OPTIONS, HEAD)
N32 Ingress Action		M	Object	Ingress Action
	Title	M	String	Title for the Error Configuration
	Status Code	M	Integer	Default Value 406

Table 4-23 (Cont.) Service API Allowed List

Parameter Name	Sub Parameter	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
	Action	M	Enum	Whenever a failure happens, request will be rejected with the user configured action. Default value is Reject.
N32 Egress Action		M	Object	Egress Action
	Title	M	String	Title for the Error Configuration
	Status Code	M	Integer	Default Value 406
	Action	M	Enum	Whenever a failure happens, request will be rejected with the user configured action. Default value is Reject

4.2.9.3 Cat 2 – Network ID Validation

The **Security Countermeasure** option is used to enable the Cat 0 - SBI Message Schema Validation feature, Cat 1 -Service API Validation feature, Cat 2 – Network ID Validation feature, and Cat 3 - Previous Location Check feature.

Perform the following procedure to enable or disable the **Cat 2 -Network ID Validation** feature:

1. From the left navigation menu, navigate to **SEPP**, and then click **Security Countermeasure**.
2. Click **Cat 2 -Network ID Validation** under **Security Countermeasure**, the **Option**, **Cat 2 - Network ID Validation List**, **Header**, and **Body IE** appears underneath.
3. Click **Option**, the **Option** page appears on the right pane. The Cat 2 – Network ID Validation feature enabling details are available on the screen.
4. Click **Edit** icon to modify the Option. The **Edit Option** page appears.
5. Set the **Network ID in Header Validation Enabled** to True or False.
6. Set the **Network ID in Body Validation Enabled** to True or False.

The parameters are:

Table 4-24 Cat 2 -Network ID Validation

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
Network ID in Header Validation Enabled	M	Boolean	A Boolean value to enable and disable the Network ID in Header Validation feature at global level. True- Enabled False- Disabled
Network ID in Body Validation Enabled	M	Boolean	A Boolean value to enable and disable the Network ID in Body Validation feature at global level. True- Enabled False- Disabled

Configuring Cat 2 -Network ID Validation Feature

1. From the left navigation menu, navigate to **SEPP**, and then click **Security Countermeasure**.
2. Click **Cat 2 -Network ID Validation** under **Security Countermeasure**, the **Option, Cat 2 - Network ID Validation List , Header**, and **Body IE** appears underneath.
3. Click **Cat 2 -Network ID Validation List** , the **Cat 2 -Network ID Validation List** page appears on the right pane.
4. Click **Add** to add a new Cat 2 -Network ID Validation List. The **Add Cat 2 -Network ID Validation List** page appears and user can add the Network ID Validation List information.
5. Enter **Network ID Validation List Name**.
6. Enter **Ingress Rules** with **HTTP Method** and **Resource URI**.
7. Enter **Egress Rules** with **HTTP Method** and **Resource URI**.
8. Enter **Ingress Error Action** and **Egress Error Action**.

Table 4-25 Network ID Validation List

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
Network ID Validation List Name	M	String	A string value to represent a Network ID Validation List Name

Table 4-26 Ingress and Egress Error Action Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
Action	M	Enum	Error Action in case of Network ID Validation Failure (REJECT, FORWARD)
Status Code	M	Integer	Error Status Code to be returned in case of Network ID Validation Failure
Title	M	String	Error Title in case of Network ID Validation Failure

Adding Ingress Rules and Egress Rules

1. Click **Network ID Validation List**, the **Network ID Validation List** page appears on the right pane.
2. Click **Add** to add a new Network ID Validation List. The **Add Network ID Validation List** page appears and user can add the Network ID Validation List information.
3. To add Ingress Rules, click **Add** icon for the Ingress rules.
4. A new page, **Add Ingress Rules**, opens to Add Ingress Rules with **HTTP Method** and **Resource URI** as configurable parameters. Select the desired HTTP Method and Resource URI from the drop down menu.
5. To add Egress Rules, click **Add** icon for the Egress rules.
6. A new page, **Add Egress Rules**, opens to Add Egress Rules with **HTTP Method** and **Resource URI** as configurable parameters. Select the desired HTTP Method and Resource URI from the drop-down menu.

Table 4-27 Ingress Rules and Egress Rules Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
HTTP Method	M	Enum	Enums with the following allowed values: POST, PUT, GET, PATCH, DELETE, OPTION, HEAD
Resource URI	M	String	Resource URI for which PLMN ID Validation will take place.

Header Configuration

Perform the following procedure to View Header Configuration.

1. From the left navigation menu, navigate to **SEPP**, and then select **Security Countermeasure**.
2. Then select **Cat 2 – Network ID Validation**.
3. The **Header** option appears underneath.
4. Click **Header**, the **Header** screen appears at the right pane. The Header details are available on the screen.

The parameters are:

Table 4-28 Header Configuration

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
Resource URI	M	String	Resource URI for which PLMN ID Validation happens.
HTTP Method	M	Enum	Enums with the following allowed values: POST, PUT, GET, PATCH, DELETE, OPTION, HEAD
Header Name	M	String	Header Name for which PLMN ID validation should happen
Regular Expression	M	String	Regular Expression to fetch PLMN ID (MCC & MNC)
Associated SEPP Type	M	Enum	CSEPP or PSEPP

Perform the following procedure to Add Header Configuration.

1. From the left navigation menu, navigate to **SEPP**, and then select **Security Countermeasure**.
2. Select **Cat 2 – Network ID Validation**.
3. The **Header** screen appears underneath.
4. Click **Header**, the Header screen appears at the right pane. The Header details are available on the screen.
5. Click **Add** to add a new Header. The **Add Header** screen appears and user can add the Header information.

The parameters are:

Table 4-29 Add Header Configuration

Parameter Name	Enabled Listing Screen	Enabled Edit Screen	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
Resource URI	Yes	Yes	M	String	Resource URI for which PLMN ID Validation will happen
HTTP Method	Yes	Yes	M	Enum	Enums with the following allowed values: POST, PUT, GET, PATCH, DELETE, OPTION, HEAD
Header Name	Yes	Yes	M	String	Header Name for which PLMN ID validation should happen
Regular Expression	Yes	Yes	M	String	Regular Expression to Fetch PLMN ID (MCC & MNC)
Associated SEPP Type	Yes	Yes	M	Enum	CSEPP or PSEPP

Body IE Configuration

Perform the following procedure to View Body IE Configuration.

1. From the left navigation menu, navigate to **SEPP**, then select **Security Countermeasure**.
2. Then select **Cat 2 – Network ID Validation**.
3. The **Body IE** screen appears underneath.
4. Click **Body IE**, the Body IE screen appears at the right pane. The Body IE details are available on the screen.

The parameters are:

Table 4-30 Body IE Configuration

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
Resource URI	M	String	Resource URI for which PLMN ID Validation will happen
HTTP Method	M	Enum	Enums with the following allowed values: POST, PUT, GET, PATCH, DELETE, OPTION, HEAD
Body IE Key	M	String	Body IE Key for which PLMN ID validation should happen
Associated SEPP Type	M	Enum	CSEPP or PSEPP
Regular Expression	M	String	Regular Expression to Fetch PLMN ID (MCC and MNC)

Perform the following procedure to Add Body IE Configuration:

1. From the left navigation menu, navigate to **SEPP**, and then select **Security Countermeasure**.
2. Then select **Cat 2 – Network ID Validation**.
3. The **Body IE** screen appears underneath.
4. Click **Body IE**, the Body IE screen appears at the right pane. The Body IE details are available on the screen.
5. Click **Add** to add a new Body IE. The **Add Body IE** screen appears and user can add the Body IE information.

The parameters are:

Table 4-31 Add Body IE Configuration Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
Resource URI	M	String	Resource URI for which PLMN ID Validation will happen
HTTP Method	M	Enum	Enums with the following allowed values: POST, PUT, GET, PATCH, DELETE, OPTION, HEAD
Body IE Key	M	String	Body IE Key for which PLMN ID validation should happen
Associated SEPP Type	M	Enum	CSEPP or PSEPP
Regular Expression	M	String	Regular Expression to fetch PLMN ID (MCC and MNC)

4.2.9.4 Cat 0 - SBI Message Schema Validation

The **Security Countermeasure** option is used to enable the Cat 0 - SBI Message Schema Validation feature, Cat 1 -Service API Validation feature, Cat 2 – Network ID Validation feature, and Cat 3 - Previous Location Check feature.

Perform the following procedure to do the **Cat 0- SBI Message Schema Validation feature** configurations:

Options screen

1. From the left navigation menu, navigate to SEPP and then click **Security Countermeasure**.
2. Click **Cat 0 - SBI Message Schema Validation feature** under **Security Counter Measure**, the **Option** appears underneath.
3. Click **Option**, the option screen appears at the right pane. The Cat 0 - SBI Message Validation feature details are available in the screen.
4. Click Edit icon to modify the Option. The **Edit Option** page appears

5. Set the **Message Validation on Body Enabled** and **Message Validation on Query Parameters Enabled** to True.
6. Set the **Maximum Request Size (KB)** as per the requirement, default value is set as 40 KB.
7. Set the **Maximum Number of Query parameters** as per the requirement, default value is set as 100.

The parameters are:

Table 4-32 Cat 0 - SBI Message Schema Validation feature (Options Screen) Parameters

Parameter Name	Datatype	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Message Validation On Body Enabled	boolean	O	A boolean value to enable or disable the message validation on body at global level. true indicates enabled false indicates disabled
Message Validation On Query Parameters Enabled	boolean	O	A boolean value to enable and disable the message validation on Query Parameters at global level. true indicates enabled false indicates disabled
Maximum Request Size (KB)	integer	O	Provides maximum allowed request body size. Default value: 40 KB
Maximum Number of Query parameters	integer	O	Provides maximum number of allowed query parameters. Default Value: 100

Message Validation List Screen

1. From the left navigation menu, navigate to SEPP and then click **Security Countermeasure**.
2. Click **Cat 0 - SBI Message Schema Validation feature** under **Security Countermeasure**, the **Message Validation List** appears underneath.
3. Click **Message Validation List**, the Message Validation List screen appears at the right pane.
4. Click Edit icon to modify the Option. The **Edit Option** page appears
5. The user can edit or add the Message Validation List.
6. Click Edit icon to modify the Option. The **Edit Option** page appears. The Message Validation List can be edited.
7. Click **Add** to add a new Message Validation List. The **Add Message Validation List** page appears, and the user can add the new Message Validation List information.

8. The user can add **Message Validation List Name**, **Ingress Rules** with **HTTP Method** and **Resource URI**, **Egress Rules** with **HTTP Method** and **Resource URI** (Not allowed in Roaming Hub mode), **Ingress Error Action**, and **Egress Error Action**.

The parameters are:

Table 4-33 Message Validation List Parameters

Parameter Name	Datatype	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Message Validation List Name	string	M	Represents a Message Validation List Name

Ingress Error Action and Egress Error Action Parameters:

Table 4-34 Ingress Error Action and Egress Error Action Parameters

Parameter Name	Datatype	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Action	Enum	M	Error action in the case of Message Validation failure. Range: REJECT, FORWARD
Status Code	String	M	Error status code to be returned in case of Message Validation failure.
Title	String	M	Error Title in case of Message Validation failure.

Adding Ingress Rules and Egress Rules

1. Click **Message Validation List**, the **Message Validation List** page appears on the right pane.
2. Click **Add** to add a new Message Validation List. The **Add Message Validation List** page appears and user can add the Message Validation List information.
3. To add Ingress Rules, click **Add** icon for the Ingress rules.
4. A new page, **Add Ingress Rules**, opens to Add Ingress Rules with **HTTP Method** and **Resource URI** as configurable parameters. Select the desired HTTP Method and Resource URI from the drop down menu.
5. To add Egress Rules, click **Add** icon for the Egress rules.
6. A new page, **Add Egress Rules**, opens to Add Egress Rules with **HTTP Method** and **Resource URI** as configurable parameters. Select the desired HTTP Method and Resource URI from the drop down menu.

Table 4-35 Ingress Rules and Egress Rules Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
HTTP Method	M	Enum	Enums with the following allowed values: POST, PUT, GET, PATCH, DELETE, OPTION, HEAD
Resource URI	M	String	Resource URI for which Message validation happens.

Message Schema Configuration Screen

Perform the following procedure to view and update Message Schema Configuration.

1. From the left navigation menu, navigate to **SEPP** and then select **Security Countermeasure**.
2. Select **Cat 0 - SBI Message Validation**.
3. The **Message Schema Configuration** option appears underneath.
4. Click **Message Schema Configuration**, the **Message Schema Configuration** screen appears at the right pane. The Message Schema Configuration details are available in the screen.
5. Click **Add** to add a new Resource URI, HTTP Method, and corresponding JSON schema.
6. Select a **Resource URI** from dropdown.
7. Select a **HTTP Method** from dropdown.
8. Enter Corresponding resolved **Message Schema** in JSON format.

Perform the following procedure to delete a existing Resource URI and HTTP Method and corresponding Message Schema:

1. From the left navigation menu, navigate to **SEPP** and then select **Security Countermeasure**.
2. Select **Cat 0 - SBI Message Validation**.
3. The **Message Schema Configuration** option appears underneath.
4. Click **Message Schema Configuration**, the **Message Schema Configuration** screen appears at the right pane. The Message Schema Configuration details are available in the screen.
5. Select HTTP Method, and Resource URI to be deleted and click **Delete** to delete a Resource URI, HTTP Method, and corresponding JSON schema.
6. The message "Do you want to delete the record" appears. Click **OK**.

Table 4-36 Message Schema Configuration Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
Resource URI	M	String	Resource URI
HTTP Method	M	Enum	Resource URI Method (GET,POST,PUT,PATCH ,DELETE,OPTIONS, HEAD)
Message Schema(JSON)	M	Object	Message Schema

4.2.9.5 Cat 3 - Previous Location Check

The **Security Countermeasure** option is used to enable the Cat 0 - SBI Message Schema Validation feature, Cat 1 -Service API Validation feature, Cat 2 – Network ID Validation feature, and Cat 3 - Previous Location Check feature.

Perform the following procedure to do the **Cat 3 - Previous Location Check feature** configurations (The **Option** and **Trigger List** appears underneath) :

Option Screen Configuration

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat 3 - Previous Location Check feature** under **Security Countermeasure**, the **Option** appears underneath.
3. Click **Option**, the option screen appears at the right pane. The Cat 3 - Previous Location Check feature details are available in the screen.
4. Click Edit icon to modify the Option. The **Edit Option** page appears
5. Set the **Previous Location Check Enabled** to True.
6. Set the **Cache Refresh Timer (milliseconds)** as per the requirement. The default value is set as 5000.

The parameters are:

Table 4-37 Cat 3 - Previous Location Check feature (Option Screen) Parameters

Parameter Name	Datatype	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Previous Location Check Enabled	boolean	M	A boolean value to enable or disable the Cat 3 - Previous Location Check feature at global level. true indicates enabled false indicates disabled Default Value: false

Table 4-37 (Cont.) Cat 3 - Previous Location Check feature (Option Screen) Parameters

Parameter Name	Datatype	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Cache Refresh Timer (milliseconds)	integer	M	An integer value to set the cache refresh timer. After this timer expiry, PN32F fetch the UE authentication status from UDR for the UE ID received in Ingress Request. Default value: 5000

Trigger List Screen Configuration

Trigger List screen allows the user to configure a set of rules for which Cat-3 Previous Location Check happens.

Perform the following procedure to view a Previous Location Trigger List:

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat 3 - Previous Location Check feature** under **Security Countermeasure**, the **Trigger List** appears underneath.
3. Click **Trigger List**, the **Cat 3 - Previous Location Trigger List** screen appears at the right pane. The Cat 3 - Previous Location Check feature details are available on the screen.
4. Click **Add** to add a new Previous Location Trigger List. The **Create Cat 3- Previous Location Trigger List** page appears and user can add the Previous Location Trigger List information.
5. Enter **Name**, **N32 Ingress Rules** with **HTTP Method**, **Resource URI**, **Error Action**, and **Exception Action**.

Trigger List parameters are:

Table 4-38 Trigger List parameters

Parameter Name	Data Type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Name	String	M	Represents a Previous Location Trigger List Name. Default Value: Blank

Error Action Parameters:

Table 4-39 Error Action Parameters

Parameter Name	Data Type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Action	Enum	M	Error action, in case of Previous Location Check Validation failure. Range: REJECT, FORWARD. Default Value: REJECT
Status Code	Integer	M	Error Status Code to be returned, in case of Previous Location Check Validation failure. Status codes 401 and 407 are not supported at present. Default Value: 406
Title	String	M	Error title, in case of Previous Location Check Validation failure. Default Value: CAT 3 Previous Location Check Failed

Exception Action Parameters:**Table 4-40 Exception Action Parameters**

Parameter Name	Data Type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Action	Enum	M	Exception action, in case of Previous Location Check Exception failure Range: REJECT, FORWARD. Default Value: REJECT
Status Code	Integer	M	Exception Status Code to be returned, in case of Previous Location Check Exception failure. Status codes 401 and 407 are not supported at present. Default Value: 406
Title	String	M	Exception title, in case of Previous Location Check Exception failure. Default Value: CAT 3 Previous Location Check Failed due to exception

Add N32 Ingress Rules screen

1. To add N32 Ingress Rules, click **Add**.
2. **Add N32 Ingress** page opens to add Ingress Rules with **HTTP Method** and **Resource URI** as configurable parameters.
3. Select the desired HTTP Method and Resource URI from the drop-down menu.

The parameters are:

Table 4-41 HTTP Method and Resource URI

Parameter Name	Data Type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
HTTP Method	Enum	M	Enums with the following allowed values: POST, PUT, GET, PATCH, DELETE, OPTION, HEAD
Resource URI	String	M	Resource URI for which Previous Location Check validation happens.

Header configuration

Header Configuration screen allows the user to configure Headers for which Previous Location Check Validation happens:

Perform the following procedure to configure Serving Network ID Header Configuration.

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat 3 - Previous Location Check feature** under **Security Countermeasure**, the **Trigger List** appears underneath.
3. Click **Header**, the **Serving Network ID** and **UE ID** appears underneath.
4. Click **Serving Network ID**, the **Serving Network ID Header** details are available in the screen.
5. Click **Add** to add a new header. The **Add Serving Network ID Header** page appears and user can add the Header information.

The parameters are:

Table 4-42 Header Parameters

Parameter Name	Data Type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Resource URI	String	M	Resource URI for which Previous Location Check validation happens.

Table 4-42 (Cont.) Header Parameters

Parameter Name	Data Type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
HTTP Method	ENUM	M	Enums with the following allowed values: POST, PUT, GET, PATCH, DELETE, OPTION, HEAD.
Header Name	String	M	Header Name for which Previous Location Check validation happens.
Regular Expression	String	M	Regular expression to fetch Serving Network ID (MCC and MNC).

UE ID Header

Perform the following procedure to configure the UE ID Header Configuration:

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat 3 - Previous Location Check feature** under **Security Countermeasure**, the **Trigger List** appears underneath.
3. Click **Header**, the **Serving Network ID** and **UE ID** appears underneath.
4. Click **Serving Network ID**, the **Serving Network ID Header** details are available in the screen.
5. Click **Add** to add a new Body IE. The **Add Serving Network ID Header** page appears and user can add the Header information.
6. Click **UE ID**, the **UE ID Header** details are available in the screen.
7. Click **Add** to add a new Header. The **Add UE ID** page appears and user can add the UE ID information.

The parameters are:

Table 4-43 UE ID Header Parameters

Parameter Name	Data Type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Resource URI	String	M	Resource URI for which Previous Location Check validation happen.
HTTP Method	ENUM	M	Enums with the following allowed values: POST, PUT, GET, PATCH, DELETE, OPTION, HEAD.

Table 4-43 (Cont.) UE ID Header Parameters

Parameter Name	Data Type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Header Name	String	M	Header Name for which Previous Location Check validation should happen.
Regular Expression	String	M	Regular Expression to fetch UE ID.

Body IE Configuration

Perform the following procedure to configure the Serving Network ID Body IE Configuration:

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat 3 - Previous Location Check feature** under **Security Countermeasure**, the **Body IE** appears underneath.
3. Click **Body IE**, the **Serving Network ID** and **UE ID** appears underneath.
4. Click **Serving Network ID**, the **Serving Network ID Body IE** details are available in the screen.
5. Click **Add** to add a new Body IE. The **Add Serving Network ID Body IE** page appears and user can add the Body IE information.

The parameters are:

Table 4-44 Serving Network ID Body IE Parameters

Parameter Name	Data Type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Resource URI	String	M	Resource URI for which Previous Location Check validation will happen
HTTP Method	ENUM	M	Enums with the following allowed values: POST, PUT, GET, PATCH, DELETE, OPTION, HEAD
Body IE Key	String	M	Body IE Key Name for which Previous Location Check validation should happen
Regular Expression	String	M	Regular Expression to Fetch Serving Network ID (MCC & MNC)

Perform the following procedure to configure the UE ID Body IE Configuration:

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.

- Click **Cat 3 - Previous Location Check** feature under **Security Countermeasure**, the **Body IE** appears underneath.
- Click **Body IE**, the **Serving Network ID** and **UE ID** appears underneath.
- Click **UE ID**, the **UE ID Body IE** details are available in the screen.
- Click **Add** to add a new UE ID. The **Add UE ID Body IE** page appears and user can add the UE ID information.

The parameters are:

Table 4-45 UE ID Body IE Configuration Parameters

Parameter Name	Data Type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Resource URI	String	M	Resource URI for which Previous Location Check validation happens.
HTTP Method	ENUM	M	Enums with the following allowed values: POST, PUT, GET, PATCH, DELETE, OPTION, HEAD.
Body IE Key	String	M	Body IE key name for which Previous Location Check validation should happen.
Regular Expression	String	M	Regular Expression to fetch UE ID.

4.2.9.6 Cat-3 Time Location Check for Roaming Subscribers

Perform the following procedure to configure the Cat-3 Time check for Roaming Subscribers feature:

Option Screen Configuration:

- From the left navigation menu, navigate to **SEPP** and then click **Security Counter Measure**.
- Click **Cat-3 Time Location Check** under Security Counter Measure. **Unauthenticated Location** page appears underneath.
- Click **Unauthenticated Location**. The **Option** appears underneath.
- Click **Option**. The option screen appears on the right pane. The Cat-3 Time check for Roaming Subscribers feature details are available on the screen.
- Click **Edit** icon to modify the Option. The **Edit Option** page appears.
- Set the **Cat 3 Time Check Unauthenticated Location Enabled** to True.
- Set the **Average Flight Velocity (km/hr)** to calculate the distance as per the requirement (must be set in kilometers per hour (km/hr)).
- Set the **Blocklist Refresh Timer Value** and **Blocklist Refresh Time Unit** as per the requirement.
- Set the **Cache Refresh Timer (milliseconds)**.

The following are the Cat-3 Time Location Check for Roaming Subscribers parameters:

Table 4-46 Cat-3 Time Location Check for Roaming Subscribers Parameters

Parameter Name	Data Type	Description
Cat 3 - Time Location Check - Unauthenticated Location Enabled	Boolean	This is a mandatory parameter. A boolean value to enable or disable the Cat-3 Time Location Check feature at the global level. True indicates that the feature is enabled. False indicates that the feature is disabled.
Average Flight Velocity (km/hr)	Integer	This is a mandatory parameter. The average flight velocity of the user while travelling from one location to the other. Note: The default value is set as 12000 km/hr.
Blocklist Refresh Timer Value	Integer	This is a mandatory parameter. In case a user is blocklisted because of malicious activity, the blocklist will be refreshed according to the value given for 'Blocklist Refresh Timer Value'.
Blocklist Refresh Time Unit	Enum	This is a mandatory parameter. 'Blocklist Refresh Time Unit' specifies the unit for the value given in 'Blocklist Refresh Timer Value'.
Cache Refresh Timer	Integer	This is a mandatory parameter. The parameter specifies the time after which the cache is refreshed. Note: The default value is 120000 ms.

Error Action Parameters:

Table 4-47 Error Action Parameters

Parameter Name	Data Type	Description
Action	Enum	This is a mandatory parameter. Determines the action to be taken, either reject or forward if an error is encountered. Range: REJECT, FORWARD. Default Value: REJECT
Status Code	Integer	This is a mandatory parameter. Indicates if an error has occurred with the specified error code. Default Value: 406

Table 4-47 (Cont.) Error Action Parameters

Parameter Name	Data Type	Description
Title	String	This is a mandatory parameter. Indicates if an error has occurred with the specified title for the error.

Exception Action Parameters:

Table 4-48 Exception Action Parameters

Parameter Name	Data Type	Description
Action	Enum	This is a mandatory parameter. Determines the action to be taken, either reject or forward if an error is encountered. Range: REJECT, FORWARD. Default Value: REJECT
Status Code	Integer	This is a mandatory parameter. Indicates if an error has occurred with the specified error code. Default Value: 406
Title	String	This is a mandatory parameter. Indicates if an error has occurred with the specified title for the error.

Country Screen Configuration:

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat-3 Time Location Check** under Security Counter Measure. **Unauthenticated Location** page appears underneath.
3. Click **Unauthenticated Location**. The **Country Configurations** appears underneath.
4. Click **Country Configurations**. The **Add Country Configurations** screen appears on the right pane.
5. Set the **Country Name**, **MCC**, **Latitude**, and **Longitude**.
6. Click **Save**.

The parameters are:

Table 4-49 Country Screen Parameters

Parameter Name	Data Type	Description
Country Name	String	This is a mandatory parameter. The input takes the name of the country the user wishes to add.

Table 4-49 (Cont.) Country Screen Parameters

Parameter Name	Data Type	Description
MCC	string	This is a mandatory parameter. The input takes the MCC of the country the user wishes to add.
Latitude	String	This is a mandatory parameter. The input takes the latitude of the country the user wishes to add.
Longitude	String	This is a mandatory parameter. The input takes the longitude of the country the user wishes to add.

Exception MCC List Screen Configuration:

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat-3 Time Location Check** under Security Counter Measure. The **Unauthenticated Location** page appears underneath.
3. Click **Unauthenticated Location**. The **Exception MCC list configurations** appears underneath.
4. Click **Exception MCC list configurations**. The **Add Exception MCC list configurations** screen appears on the right pane.
5. Click **Edit** icon to modify the Exception MCC list configurations.
6. Set the **MCC** and **Exception MCC**.
7. Click **Save**.

The parameters are:

Table 4-50 Exception MCC List Parameters

Parameter Name	Data Type	Description
MCC	Integer	This is a mandatory parameter. The input takes the MCC of the country against which the exception MCCs are to be configured.
Exception MCC	Integer	This is a mandatory parameter. The input takes the MCC of the countries against which the time-check feature should not be applied. Multiple exception mcc values can be configured against the same value for MCC.

4.2.9.7 Cat 1 - Service API Query Param Validation List Screen

The **Security Countermeasure** option is used to enable the **Cat 0 - SBI Message Schema Validation** feature, **Cat 1 -Service API Validation** feature, **Cat 2 – Network ID Validation**

feature, **Cat 3 - Previous Location Check feature**, and **Cat 3 - Time Location Check feature**.

Perform the following procedure to do the **Cat-1 Service API Query Parameters Validation** configurations:

1. From the left navigation menu, navigate to **SEPP** and then click **Security Countermeasure**.
2. Click **Cat 1 -Service API Validation** under **Security Countermeasure**. The following fields will appear:
 - **Option**
 - **Service API Allowed List**
 - **Cat-1 Query Parameter Validation List**
3. Click **Option**, the **Options** page appears on the right pane.
4. Set **Enable Cat 1-Query Parameter Validation** parameter to *True* to enable the feature.

Note

Enable Cat 1 - Service API Validation parameter must be set to *True* to enable this feature.

5. Click **Cat-1 Query Parameter Validation List** under **Cat 1 -Service API Validation**, the **Create Cat-1 Query Parameter Validation List** page appears on the right pane.
6. Click **Add** to add or update in sthe allowed REST APIs and supported methods.
7. Click **Save**.

The **Options** page parameters are:

Table 4-51 Options Page Parameters

Parameter Name	Data Type	Description
Enable Cat 1 - Service API Validation	Boolean	This is a mandatory parameter. This is used to enable or disable the Cat 1 - Service API Validation feature. True indicates that the feature is enabled and false indicates that the featur is disabled. The feature is disabled (set to false) by default.
Enable Cat 1-Query Parameter Validation	Boolean	This is a mandatory parameter. This is used to enable or disable the Cat-1 Service API query parameters validation feature. True indicates that the feature is enabled and false indicates that the feature is disabled. By default, the feature is disabled.

The **Create Cat-1 Query Parameter Validation List** parameters are:

Table 4-52 Create Cat-1 Query Parameter Validation List

Parameter Name	Data Type	Parameter Description
Service API Query Param Validation List Name	Enum	This is a mandatory parameter. This parameter indicates the name of the Service API Query Param Validation List. Note: Only those list names are visible in the dropdown which have already been configured on the 'Cat-1 Service API Allowed List' screen as it provides mapping between Cat-1 Service API Validation feature and Cat-1 Query Parameter Validation feature.
Service API - Method	Enum	This is a mandatory parameter. This parameter indicates the method for the resource URI that the user wants to add. Only GET method is allowed to be configured.
Service API - Resource URI	Enum	This is a mandatory parameter. This parameter takes only NRF Discovery URI as input to which the query parameter validation needs to be done.
Query Parameters - Query Parameter Name	String	This is a mandatory parameter. This parameter takes the name of the query parameter for which the validation needs to be done. Default value is requester-nf-type.
Query Parameters - Query Parameter Value	String	This is a mandatory parameter. This parameter takes the value of the query parameter for which the validation needs to be done which will be requester-nf-type in this case.
Query Parameters - Paired Query Parameter Name	String	This is a mandatory parameter. This parameter takes the name of the paired query parameter for which the validation needs to be done. Default value: target-nf-type.
Query Parameters - Paired Query Parameter Value	String	This is a mandatory parameter. This parameter takes the value of the paired query parameter for which the validation needs to be done which will be 'target-nf-type' in this case.

4.2.10 Mediation

The mediation option allows the user to set Mediation feature as ENABLED or DISABLED and configure the mediation options.

Perform the following procedure to enable or disable the mediation feature:

1. From the left navigation menu, navigate to SEPP and then click **Mediation**. The **options** appears underneath.
2. Click **Options**, the options screen appears at the right pane. The **Mediation Feature** and **Error Configuration** details are available in the screen.
3. Click **Edit** icon to modify the Options. The **Edit Options** page appears.
4. Set the **Mediation Feature** to True or False and configure the **Error Configuration** parameters **Action**, **Status Code**, and **Title**.

The parameters are:

Mediation Options

Table 4-53 Mediation Options

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
Enable Mediation	O	Boolean	A Boolean value to enable and disable the Mediation feature. true indicates Enabled false indicates Disabled. The feature is disabled (set to false) by default.
Error Configuration	M	Object	Error Configuration details

Mediation Trigger Rule List Configuration

Mediation Trigger Rule Configuration screen allows the user to configure a set of trigger rules which act as a filtering criteria for SEPP to send a particular request for mediation.

Perform the following procedure to configure the trigger rules:

1. From the left navigation menu, navigate to SEPP and then select **Mediation**. The **Trigger Rule List** appears underneath.
2. Click **Trigger Rule List**, the list of all the existing rules and corresponding configurations appears.
3. Click **Add** to add the Trigger Rule List. The **Add Trigger Rules** page appears and user can add the Trigger Rule information.

Note

HTTP Method, Resource URI, Trigger Point, and Group Id are the Trigger Rule parameters.

- Click **Edit** to modify the rule list. The **Edit Trigger Rules** page appears and user can update the Trigger Rule information.

Error Configuration Parameters

Table 4-54 Error Configuration Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
Title	M	String	Error Title in case of getting error from mediation service
Action	M	Enum	Error Action to be performed in case of getting error from mediation service (Reject, Continue).
StatusCode	M	Integer	Error Status Code to be returned in case of getting error from mediation service.

Mediation Trigger Rules List Parameters

Table 4-55 Mediation Trigger Rules List Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
Name	M	String	Name of Trigger Rule List
Trigger Rules	M	Object	List of Trigger Rules
Trigger Rule List Type	M	Enum	Type of TRL (Local / Remote)
Match All Enabled	M	Boolean	Match All Enabled (true/false)
Match All GroupId	M	String	Match All Group Id
Match All Trigger Points	M	Enum	Match All Trigger Points

Trigger Rule Parameters

Table 4-56 Trigger Rule Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
HTTP Method:	M	Enum	Resource URI Method (GET,POST,PUT,PATCH ,DELETE,OPTIONS)
Resource URIs:	M	String	Resource URI

Table 4-56 (Cont.) Trigger Rule Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
Trigger Points:	M	Enum	List of Trigger Point ("N32_Egress_Request", "N32_Ingress_Response", "N32_Ingress_Request", "N32_Egress_Response")
Group Id:	M	String	Group ID for which mediation configuration is to be done. This is passed to the Mediation Service for grouping similar rules.

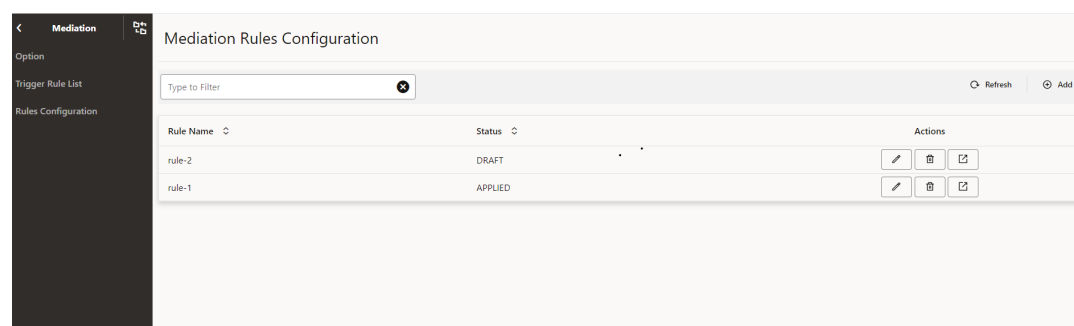
Mediation Rules Configuration

Rules configuration screen allows the user to compile, create, update, delete, clone and apply the mediation rules using CNCC.

Perform the following procedure to configure the mediation rules:

1. From the left navigation menu, navigate to **SEPP** and then select **Mediation**. The **Rules Configuration** screen appears underneath.
2. Click **Rules Configuration**, the list of all the existing rules with corresponding status appears.

Figure 4-4 Mediation Rules Configuration Screen



3. Click **Add** to add a new rule. The **Add Mediation Rule** page appears and user can add the mediation rule information.

Figure 4-5 Add Mediation Rule Screen

Add Mediation Rule

Option

Trigger Rule List

Rules Configuration

Rule Name: rule-3

Format: DRL

Status: DRAFT

Mediation Mode: MEDIATION_ACTIVE

Code: `package com.oracle.cgbu.ocmediation.nfmediation;
import com.oracle.cgbu.ocmediation.nfruleengine.NFDroolsRuleEngine;
import com.oracle.cgbu.ocmediation.factdetails.Request;
import com.oracle.cgbu.ocmediation.factdetails.Response;
import java.util.Map;`

State: Save

Save Cancel

Note

Rule Name, Format, Status, Mediation Mode, Code, and State are the mediation rule parameters.

- Click **Edit** button of an existing rule to modify that rule. The **Edit Mediation Rule** page appears and user can edit the rule information.

Figure 4-6 Edit Mediation Rule Screen

Edit Mediation Rule

Option

Trigger Rule List

Rules Configuration

Rule Name: rule-2

Format: DRL

Status: DRAFT

Mediation Mode: MEDIATION_ACTIVE

Code: `rule "Rule_show_add_header"
when
req : Request(headers.has("3gpp-Sbi-Message-Priority") == true)
then
req.headers.add("3gpp-Sbi-Message-Priority","10")
end`

State: Compile

Save Cancel

- Click **Delete** button of an existing rule to delete that rule, confirmation dialog box appears and user can click **OK** to delete the rule or **Cancel** to cancel the deletion.
- Click **View** button of an existing rule to view the rule's details.

Rules Configuration Parameters

Table 4-57 Rules Configuration Parameters

Parameter Name	Mandatory(M) / Optional(O) Conditional(C)	Datatype	Description
Rule Name	M	String	Mediation rule name
Status	M	Enum	Mediation rule status: APPLIED or DRAFT
State	M	Enum	Compile, Clone, Save, Draft, Apply The user can select one of these states.
Mediation Mode	M	Enum	MEIDATION_ACTIVE: its only applicable to mediation microservice active mode MEDIATION_TEST: its only applicable to mediation microservice test mode The user is required to configure the mediation rules using MEDIATION_ACTIVE mediation mode. MEDIATION_TEST mode is only for internal purpose.
Code	M	String	Mediation rule code content. The user has to prepend the following data block along with the needed rules in the code section and save to create the rule in DRAFT status. If the user wants to apply the rule on mediation microservice then user can edit the rule and save it again with Apply state. <pre>package com.oracle.cgbu.ocmediation.n fmediation; import com.oracle.cgbu.ocmediation.n fruleengine.NFDroolsRuleEngin e; import com.oracle.cgbu.ocmediation.f actdetails.Request; import com.oracle.cgbu.ocmediation.f actdetails.Response; import java.util.Map; import java.util.HashMap; dialect "mvel"</pre>
Format	M	Enum	Rule format. Only DRL is supported currently.
New Rule Name	C	String	New rule name is to be given only when state is clone.

Note

- The rule name for a rule in the **DRAFT** status must be unique. Otherwise, the new rule overwrites the old one.
- The new rule is always created and saved with **DRAFT** status in the database. User needs to save the rule with **APPLY** state to apply the rule to mediation microservice.

4.2.11 Ingress Rate Limiting

Ingress Rate Limiting screen allows the user to configure the global parameters for ingress rate limiting feature. Perform the following procedure to enable and configure the ingress rate limiting feature:

Ingress Rate Limiting: Remote SEPP Set

1. From the left navigation menu, navigate to SEPP and then click **Ingress Rate Limiting**. The **Remote SEPP Set** option appears underneath.
2. Click **Remote SEPP Set** under **Ingress Rate Limiting**, the **Options** appears underneath.
3. Click **Options** under **Remote SEPP Set**, the **Options** page appears on the right pane.
4. Click **Edit** icon to modify the Option. The **Edit Option** page appears. The Ingress Rate Limiting feature configurations are available in the screen.
5. Set **Originating Network ID Header** as “3gpp-Sbi-Asserted-Plmn-Id” or “3gpp-Sbi-Originating-Network-Id” or both.
6. Set **Remote SEPP Set Ingress Rate Limiting Enabled** as true or false.
7. Enter **Bucket Capacity**, **Refill Rate**, **Refill Duration**, and **Request Tokens**.
8. Under **Error Configuration**, Enter **Action** as Reject.
9. Enter **Status Code** and **Title**.

Table 4-58 Ingress Rate Limiting: Remote SEPP Set Parameters

Parameter Name	Datatype	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Originating Network ID Header	Enum	M	This parameter can have the following allowed values: 3gpp-Sbi-Asserted-Plmn-Id, 3gpp-Sbi-Originating-Network-Id

Table 4-58 (Cont.) Ingress Rate Limiting: Remote SEPP Set Parameters

Parameter Name	Datatype	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Remote SEPP Set Ingress Rate Limiting Enabled	Boolean	O	A Boolean value to enable and disable the Ingress Rate Limiting feature at global level. true indicates Enabled. false indicates Disabled. The feature is disabled (set to false) by default.
Bucket Capacity	Integer	M	Integer Number for setting the Bucket Capacity as an input for Token Bucket Algorithm. Bucket size defined the capacity to handle traffic burst.
Refill Rate	Integer	M	Refill Rate to define the number of tokens to be added to refill the bucket
Refill Duration	Integer	M	Duration to decide how frequently to refill bucket
Request Token	Integer	M	Request Token to define the Pre loaded tokens to refill the bucket
Action	String	M	By Default Reject is selected as SBI Request will be rejected with the user configured Error Configuration when the number of SBI requests is above the configured limit.
Status Code	Integer	M	Error Status Code to be used in the Error Response for discarding the SBI requests when the number of SBI requests is above the configured limit.
Title	String	M	Error Title to be used in the Error Response for discarding the SBI requests when the number of SBI requests is above the configured limit.

4.2.12 Egress Rate Limiting

Egress Rate Limiting screen allows the user to configure the global parameters for egress rate limiting feature. Perform the following procedure to enable and configure the egress rate limiting feature:

Egress Rate Limiting

Perform the following procedure to enable the Egress Rate Limiting:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Rate Limiting**.
2. Select **Egress Rate Limiting** which is defined under **Rate Limiting** screen.
3. The **Option and EgressRateLimitingList** appears underneath.
4. Click **Option**, the option screen appears at the right pane. The Egress Rate Limiting Feature details are available in the screen.
5. Set **Egress Rate Limiting Enabled** to True on the right pane.

Egress rate limiting Option Parameters

Table 4-59 Egress rate limiting Option Parameters

Attribute	Data Type	Description
EgressRateLimitingEnabled	boolean	This is a mandatory parameter. Enables or disables the Egress Rate limiting. true indicates Enabled. false indicates Disabled. Default Value: False

Perform the following procedure to view a Egress Rate Limiting List:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Rate Limiting**.
2. Select **Egress Rate Limiting** which is defined under **Rate Limiting** screen.
3. The **Option and EgressRateLimitingList** appears underneath.
4. Click **EgressRateLimitingList**, the EgressRateLimitingList screen appears at the right pane.

Perform the following procedure to add an Egress Rate Limiting List:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Rate Limiting**.
2. Select **Egress Rate Limiting** which is defined under **Rate Limiting** screen.
3. The **Option and EgressRateLimitingList** appears underneath.
4. Click **EgressRateLimitingList**, the EgressRateLimitingList screen appears at the right pane. The Egress Rate Limiting feature details are available on the screen.
5. Click **Add** to add a new Egress Rate Limiting List. The **Create Egress Rate Limiting List** page appears. User can add the Egress Rate Limiting List information.
6. Enter **Egress Rate Limiting List Name**.

7. Under **Egress Rate Limiting Configurations** section, enter the list configurations; enter **Enabled**, **Discard Message Priority**, **Bucket Capacity**, **Refill Rate**, **Refill duration**, and **Request Tokens**.
8. Under **Error Configuration** section, enter the error configuration details; enter **Action**, **Enter Status Code**, and **Title**.
9. Under **Remote Sepp Set** or **PLMN ID(s)** section, select the **Remote SEPP Set** name or add PLMN IDs for the Egress Rate Limiting List.

Table 4-60 Parameter List

Parameter Name	Data Type	Description
Name	String	This is a mandatory parameter. Indicates the Egress Rate Limiting List Name
Enabled	boolean	This is a mandatory parameter. A boolean value to enable or disable the feature at egress rate limiting list level. This will be disabled by default.
Discard Message Priority	integer	This is a mandatory parameter. Integer value to indicate the message priority used to decide if a message shall be dropped or not when rate limiting is enforced. Messages with higher or equal priority than Discard Message Priority are dropped. It the value is not provided in REST API , 0 will be used as default value of integer.
Bucket Capacity	integer	This is a mandatory parameter. Integer number for setting the Bucket Capacity as an input for Token Bucket Algorithm. Bucket size defined the capacity to handle traffic burst.
Refill Rate	integer	This is a mandatory parameter. Refill Rate to define the number of tokens to be added to refill the bucket Its value can't be greater than configured Bucket Capacity value
Refill Duration	integer	This is a mandatory parameter. Duration to decide how frequently to refill bucket.
Request Tokens	integer	This is a mandatory parameter. Request Tokens defines the batch size of token requested from the corresponding bucket. Its recommended that the value should be configured as: - bucket capacity divided by number of plmn-ingress gateway pods (SEPP mode). - bucket capacity divided by number of n32-ingress gateway pods (Roaming Hub mode).
Action	Enum	This is a mandatory parameter. Error Action to be used in the Error Response while discarding the requests for Egress Rate Limiting Action supported is 'REJECT'

Table 4-60 (Cont.) Parameter List

Parameter Name	Data Type	Description
Status Code	integer	This is a mandatory parameter. Error Status Code to be used in the Error Response for discarding the requests due to Egress Rate Limiting Configured Status Code is sent back in the HTTP/2 response message
Title	String	This is a mandatory parameter. Error Title to be used in the Error Response for discarding the requests for Egress Rate Limiting. Configured Title is sent back in the HTTP/2 response message.
Remote Sepp Set	Enum	This is a conditional parameter. Remote SEPP Set name in the place of PLMN ID(s), in this case the feature is applied to the PLMN IDs of the given Remote SEPP Set. Either RSS or PLMN IDs can be selected.
PLMN ID(s)	Enum	This is a conditional parameter. PLMN IDs which need to be part of Egress Rate Limiting List on which feature needs to be applied. Either PLMN IDs or RSS can be selected. It's value must be in mcc-mnc format.

Perform the following procedure to edit an Egress Rate Limiting List:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Rate Limiting**.
2. Select **Egress Rate Limiting** which is defined under **Rate Limiting** screen.
3. The **Option and EgressRateLimitingList** appears underneath.
4. Click **EgressRateLimitingList** the option screen appears at the right pane. The Egress Rate Limiting Feature details are available in the screen.
5. Click **Edit** option. The **Edit Egress Rate Limiting List** page appears and user can edit the Egress Rate Limiting List information.
6. You can edit the Egress Rate Limiting List Configurations and click **Save**.

Perform the following procedure to delete an Egress Rate Limiting List:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Rate Limiting**.
2. Select **Egress Rate Limiting** which is defined under **Rate Limiting** screen.
3. The **Option and EgressRateLimitingList** appears underneath.
4. Click **EgressRateLimitingList** the option screen appears at the right pane. The Egress Rate Limiting Feature details are available in the screen.
5. Click **Delete** option in front of an existing Egress Rate Limiting List that you want to delete.
6. "Do you want to delete the record" message appears. Click **OK**.

4.2.13 SOR

The **SOR** option allows the user to set SOR feature as enabled or disabled and configure the SOR options. Perform the following procedure to enable and configure the SOR feature:

SOR Options

1. From the left navigation menu, navigate to **SEPP** and then click **SOR**. The **Options** and **Trigger Rule List** appear underneath.
2. Click **Options** under **SOR**, the **Options** page appears on the right pane.
3. Click **Edit** icon to modify the Option. The **Edit Option** page appears.
4. Set **SOR Enabled** parameter to **True** to enable the SOR feature.
5. Set **Redirection enabled** to **true** or **false** to enable or disable the redirection and user can configure the http code.
6. Set **Alternative Routing enabled** to **true** or **false** to enable or disable the redirection and user can configure the http code.
7. Set **Retry to NF on SOR Server Error Enabled** to **true** or **false** to enable or disable the retry to SOR.
8. Configure **SOR Server(s)** by configuring **Priority**, **Http Scheme**, and other parameters..
9. Configure the **Redirection**, **Alternative Routing Options**, **SOR Server Error**, and **Custom Error**.

Options Parameters

Table 4-61 Options Parameters

Attribute	Data type	Mandatory(M) / Conditional(C) Optional(O)	Description
SOR Enabled	Boolean	O	To enable or disable the SOR feature. true indicates enabled and false indicates disabled. The feature is disabled (set to false) by default.
Alternate Routing Enabled	Boolean	O	To enable or disable the alternative SOR option. true indicates enabled and false indicates disabled. The value is disabled (set to false) by default.
Retry to NF on SOR Server Error Enabled	Boolean	O	To enable or disable the retry, in case of an error from SOR producer NF occurs. true indicates enabled and false indicates disabled.

Table 4-61 (Cont.) Options Parameters

Attribute	Data type	Mandatory(M) / Conditional(C) Optional(O)	Description
Redirection: Enabled	Boolean	O	If this parameter is enabled, then SOR responds to SEPP and SEPP sends request to producer. If this parameter is disabled, then SOR sends request to producer directly. true indicates enabled and false indicates disabled.
Redirection: codes	String	M	This parameter is used to configure all the 3xx HTTP response code that contain valid location header parameter in response, for which user wants to get the redirection applied.
servers: priority	Enum	M	SOR server priority must be Primary, Secondary, and Tertiary. The first entry will always be saved as PRIMARY in database.
servers: httpScheme	Enum	M	The scheme can be http or https.
servers: sorFqdn	String	M	Indicates the FQDN of SOR Server
servers: sorPort	String	M	Indicates the Port of SOR Server
servers: apiPrefix	String	M	Indicates the API Prefix for SOR Server
servers: serverHeader	String	M	Server header that is expected to be received from SOR Server. Typically, SOR-<SOR FQDN>
Alt Routing options: Timeout	integer	M	Indicates the timer to set for retries towards SOR.
Alt Routing options: MaxRetry	integer	M	Indicates the number of times to retry towards the primary SOR end point.
SoR Server Errors: errorCodes	string	M	This is the list of expected response codes (multiple 5xx response codes seperated by (,)) can be configured. Example: 501,504,510) from SOR that will need SEPP to contact the Producer directly in case of retry to producer is enabled.
SoR Server Errors: exceptions	string	M	The exceptions that are expected to be returned from SoR in case of timeout.
Custom Error: status code	string	M	Error code that will be relayed to consumer in case of SOR Timeout.
Custome Error: Title	string	M	Error Message that will be relayed to consumer in case of SOR Timeout.

SOR Trigger Rule List

1. From the left navigation menu, navigate to **SEPP** and then click **SOR**. The **Options** and **Trigger Rule List** appear underneath.
2. Click **Trigger Rule List** under **SOR**, the **Trigger Rule List** page appears on the right pane.
3. Click **Add** to add the Trigger Rule List.
4. Select a combination of HTTP Method and Resource **URI** from the drop down list and **Save**.

Trigger Rule List Parameters

Table 4-62 Trigger Rule List Parameters

Attribute	Data type	Mandatory(M) / Conditional(C) Optional(O)	Description
Trigger Rule List	String	M	Name of the SOR Trigger List
URI List: resourceURI	String	M	List of Resource URI
URI List: httpMethod	String	M	Request URI httpMethod. The Range is: POST, PUT, GET, PATCH, DELETE, OPTIONS, and HEAD.

4.2.14 Originating Network ID Header Support

Originating Network ID Header Support allows the user to set the Support for Originating Network Id Header Validation, Insertion, and Transposition feature as Enabled or Disabled and configure the Originating Network ID Header options.

Perform the following procedure to enable, disable, and configure the Support for Originating Network Id Header Validation, Insertion, and Transposition feature:

1. From the left navigation menu, navigate to SEPP and then click **Originating Network ID Header Support**. The **option** appears underneath.
2. Click **Option**, the option screen appears at the right pane with the parameters to enable the feature.
3. Click **Edit** icon to modify the Option. The **Edit Option** page appears.
4. Set **Enabled** to **True** or **False**.
5. Select **SEPP's Local PLMN ID[mcc-mnc]** from the drop-down menu.
6. Select **Supported Header Name** from the drop down menu.
7. Click **Save**.

The following table lists the parameters:

Table 4-63 Parameters

Parameter Name	Description	Details
Enabled	This is a mandatory parameter. A boolean value to enable or disable the support for Originating Network Id Header Validation, Insertion, and Transposition feature at the global level. True indicates the feature is enabled. False indicates the feature is disabled.	Datatype: Boolean
SEPP's Local PLMN ID [mcc-mnc]	This is an optional parameter. This value indicates SEPP's supported Local PLMN ID which is used for missing header (3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id) population at C SEPP. The user can provide this value by selecting one of the values from the drop-down. SEPP selects one of the supported local PLMN IDs when this parameter is not configured.	Datatype: String
Supported Header Name	This is a mandatory parameter. This value indicates the supported header name. The user can select either "3gpp-Sbi-Originating-Network-Id" or "3gpp-Sbi-Asserted-Plmn-Id" from the drop-down. P SEPP uses this header to either add the missing request header (3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id) or header (already present in the request) transposition.	Datatype: String

4.2.15 Configurations

This section explains how to configure cnDBTier, Egress Gateway, and Ingress Gateway APIs.

4.2.15.1 Gateway

This section explains how to configure the Common Gateway (Ingress Gateway and Egress Gateway) APIs.

4.2.15.1.1 IGW (Ingress Gateway)

This section contains the PLMN Ingress Gateway and N32 Ingress Gateway APIs.

4.2.15.1.1.1 PLMN Ingress Gateway

This section contains the PLMN Ingress Gateway APIs.

Error Code Series List

This configuration is used to map errorCodeSeries id and the corresponding errorCodeSeries configuration. By default this configuration is null.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **IGW** option to configure the Ingress Gateway APIs. The **PLMN Ingress Gateway** and **N32 Ingress Gateway** appears underneath.
3. Click **Error Code Series List** option under **PLMN Ingress Gateway**. The **Error Code Series** page is displayed.
4. Click **Add** to add the error code series.
The **Create Error Code Series** page is displayed.
5. Configure the fields in the **Create Error Code Series** page.
6. Click **Add** in the **Error Code Series** section.
The **Add Error Code Series** page is displayed.
7. Configure the fields in the **Add Error Code Series** page.
8. Click **Save** on the **Add Error Code Series** page to save the details. Click **Cancel** to discard your progress and go back to **Create Error Code Series** page.
9. Click **Save** on the **Create Error Code Series** page to save the details. Click **Cancel** to discard your progress and go back to **Error Code Series** page.

Note

- Use the Edit icon or Delete available in the next column of the specific entry to update or delete the error code series information.
- Use Refresh icon to refresh the error code series information.

Table 4-64 Error Code Series List

Field Name	Description	Details
ID	Indicates the error code identifier.	DataType: string Constraints: NA Default Value: NA Presence: M
Exception List	Lists the configurable exception or error for an error scenario in Ingress Gateway. The only supported values are: ConnectionTimeout, RequestTimeout, UnknownHostException, ConnectException, RejectedExecutionException, InternalError and NotFoundException, ClosedChannelException, BlackListIpException	DataType: string Constraints: NA Default Value: NA Presence: M

Table 4-64 (Cont.) Error Code Series List

Field Name	Description	Details
Error Code Series	Lists the error codes for a specific service. Note: "ErrorCodeSeries" is configured only if a set of error responses with specific error codes is expected in server header. If it is not configured then all the error responses will have server header.	DataType: string Constraints: NA Default Value: NA Presence: M

Table 4-65 Error Code Series Configuration

Field Name	Description	Details
Error Set	Possible values for "errorSet" attribute: 5xx, 4xx, 3xx, 2xx, 1xx	DataType: string Constraints: NA Default Value: NA Presence: M
Error Codes	Possible values include all error codes in the respective HttpSeries value assigned for "errorSet". Note: Use single value of "-1" if all error codes in that HttpSeries are to be considered.	DataType: string Constraints: NA Default Value: NA Presence: M

Server Header Details

This API can be used for adding Server Header in the error responses sent from Ingress Gateway. By default, this feature is disabled. To enable the feature, invoke the following REST API and update the enable switch.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **IGW** option to configure the Ingress Gateway APIs. The **PLMN Ingress Gateway** and **N32 Ingress Gateway** appears underneath.
3. Click **Server Header Details** option under **PLMN Ingress Gateway**. The **Server Header Details** page is displayed.
4. Click **Edit** to update the Server Header Details. The **Edit Server Header Details** page is displayed.
5. Configure the fields in the **Edit Server Header Details** page.
6. Click **Save** on the **Edit Server Header Details** page to save the details. Click **Cancel** to discard your progress and go back to Server Header Details page.

Table 4-66 Server Header Details

Field Name	Description	Details
Enabled	Use the switch to enable or disable Server Header	DataType: boolean (true, false) Constraints: NA Default Value: NA
Error Code Series Id	Specify the error list ID.	DataType: string Constraints: NA Default Value: NA Presence: M
Configuration	Configure the following fields in Configuration section	DataType: NA Constraints: NA Default Value: NA
NF Type	Specify the type of network function. In this case, it is SEPP.	DataType: string Constraints: NA Default Value: NA
NF Instance Id	Enter the SEPP instance ID. It represents the UUID of the SEPP deployment that is used to generate the Server Header.	DataType: integer Constraints: NA Default Value: NA

Note

Ensure that an errorCodeSeries exists corresponding to the errorCodeSeriesId.

4.2.15.1.1.2 N32 Ingress Gateway

This section contains the N32 Ingress Gateway APIs.

Error Code Profiles

This configuration is used to update the errorcodeprofiles that is used in Overload Control feature for populating details in error responses when a request is discarded.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **IGW** option to configure the Egress Gateway APIs. The **PLMN Ingress Gateway** and **N32 Ingress Gateway** appears underneath.
3. Click **Error Code Profiles** option under **N32 Ingress Gateway**. The **Error Code Profiles** page is displayed with default values for few fields.
4. Click **Add** to add the profiles.
The **Create Error Code Profiles** page is displayed.
5. Configure the fields in the **Create Error Code Profiles** page.
6. Click **Save** on the Create Error Code Profiles page to save the details. Click **Cancel** to discard your progress and go back to **Error Code Profiles** page.

Note

- Use the Edit icon available in the next column of the specific entry to update the route error profile information.
- Use Refresh icon to refresh the route error profile information.

Table 4-67 Error Code Profile Configuration

Field Name	Description	Details
Name	Indicate the error name.	DataType: string Constraints: NA Default Value: NA Presence: M
Error Code	Error Code field in an errorScenario determines the HttpStatusCode that needs to be populated in ProblemDetails (HttpStatus field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType field.	DataType: integer Constraints: NA Default Value: NA Presence: M
Error Cause	Error Cause field in an errorScenario determines the error cause that needs to be populated in ProblemDetails (Cause field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter.	DataType: integer Constraints: NA Default Value: NA Presence: O
Error Title	Error Title field in an errorScenario determines the title that needs to be populated in ProblemDetails (Title field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter.	DataType: integer Constraints: NA Default Value: NA Presence: O

Table 4-67 (Cont.) Error Code Profile Configuration

Field Name	Description	Details
Redirect URL	Redirect URL field in an errorScenario determines the redirection URL, this value is populated in LOCATION header while sending response from Ingress Gateway. The header is populated only when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter, the errorCode configured for the particular errorScenario lies in 3xx error series and the redirectUrl field for the particular errorScenario is configured appropriately.	DataType: integer Constraints: NA Default Value: NA Presence: O
Retry After	Retry After field in an errorScenario determines the value in seconds or particular date after which the service should be retried, this value is populated in Retry-After header while sending response from Ingress Gateway. The header is populated only when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter, the errorCode configured for the particular errorScenario lies in 3xx error series and the retry-after field for the particular errorScenario is configured appropriately in seconds.	DataType: integer Constraints: NA Default Value: NA Presence: O
Error Description	Error Description field in an errorScenario determines the description that needs to be populated in ProblemDetails (Detail field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType field.	DataType: integer Constraints: NA Default Value: NA Presence: O

Error Code Series List

This configuration is used to map errorCodeSeries id and the corresponding errorCodeSeries configuration. By default this configuration is null.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.

2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **IGW** option to configure the Ingress Gateway APIs. The **PLMN Ingress Gateway** and **N32 Ingress Gateway** appears underneath.
3. Click **Error Code Series List** option under **N32 Ingress Gateway**. The **Error Code Series List** page is displayed.
4. Click **Add** to add the error code series.
The **Create Error Code Series** page is displayed.
5. Configure the fields in the **Create Error Code Series** page.
6. Click **Add** in the **Error Code Series** section.
The **Add Error Code Series** page is displayed.
7. Configure the fields in the **Add Error Code Series** page.
8. Click **Save** on the **Add Error Code Series** page to save the details. Click **Cancel** to discard your progress and go back to **Create Error Code Series** page.
9. Click **Save** on the **Create Error Code Series** page to save the details. Click **Cancel** to discard your progress and go back to **Error Code Series** page.

Note

- Use the Edit icon or Delete available in the next column of the specific entry to update or delete the error code series information.
- Use Refresh icon to refresh the error code series information.

Table 4-68 Error Code Series List

Field Name	Description	Details
ID	Indicates the error code identifier.	DataType: string Constraints: NA Default Value: NA Presence: M
Exception List	Lists the configurable exception or error for an error scenario in Ingress Gateway. The only supported values are: ConnectionTimeout, RequestTimeout, UnknownHostException, ConnectException, RejectedExecutionException, InternalError and NotFoundException, ClosedChannelException, BlackListIpException	DataType: string Constraints: NA Default Value: NA Presence: M

Table 4-68 (Cont.) Error Code Series List

Field Name	Description	Details
Error Code Series	Lists the error codes for a specific service. Note: "ErrorCodeSeries" is configured only if a set of error responses with specific error codes is expected in server header. If it is not configured then all the error responses will have server header.	DataType: string Constraints: NA Default Value: NA Presence: M

Table 4-69 Error Code Series Configuration

Field Name	Description	Details
Error Set	Possible values for "errorSet" attribute: 5xx, 4xx, 3xx, 2xx, 1xx	DataType: string Constraints: NA Default Value: NA Presence: M
Error Codes	Possible values include all error codes in the respective HttpSeries value assigned for "errorSet". Note: Use single value of "-1" if all error codes in that HttpSeries are to be considered.	DataType: string Constraints: NA Default Value: NA Presence: M

OverloadControlDiscardPolicies

This configuration is used to update service names and corresponding policy names for the service which is mapped to ocDiscardPolicies based on "policyName" and also to enable or disable the Overload Control feature and the sampling period in overload control. By default, the Overload Control feature is disabled and the sampling period is 6000.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **IGW** option to configure the Ingress Gateway APIs. The **PLMN Ingress Gateway** and **N32 Ingress Gateway** appears underneath.
3. Click the **Overload Control Discard Policy** option under **N32 Ingress Gateway** to configure peers. The **Overload Control Discard Policy** page is displayed.
4. Click **Add** to add the configuration. The **Create Overload Control Discard Policy** page is displayed.
5. Configure the fields in the Create Overload Control Discard Policy page.
6. Click **Add** in the Policies section. The **Add Policies** page is displayed.
7. Configure the fields under the **Add Policies** page.

8. Click **Save** on the **Add Policies** page to save the details. Click **Cancel** to discard your progress and go back to Create Overload Control Discard Policy page.
9. Click **Save** on the Create **Overload Control Discard Policy** page to save the details. Click **Cancel** to discard your progress and go back to Overload Control Discard Policy page.

Note

- Use the Edit icon available in the next column of the specific entry to update the policy information.
- Use Refresh icon to refresh the policy information.

Table 4-70 Discard Policy Configuration

Attribute Name	Description	Details
Name	Name of the discarded policy. Note: name must be the value configured in policyName under ocpolicymapping.	DataType: string Constraints: NA Default Value: NA Presence: M
Scheme	Discarded policy scheme based on percentage.	DataType: string Constraints: NA Default Value: NA Presence: M

Table 4-71 Policies

Attribute Name	Description	Details
Value	Value of priority above which requests are considered as potential candidates for drop. Percentage of requests to drop in the current sampling period over the calculated rate in the previous sampling period.	DataType: string Constraints: NA Default Value: NA Presence: M
Action	Defines the action to be taken on selected requests rejection based on error code.	DataType: string The value can be: RejectWithErrorCode Constraints: NA Default Value: NA Presence: M
Level	Defines the overload level.	DataType: string Constraints: NA Default Value: NA Presence: M
Error Code Profile	Define the error code profiles.	DataType: string Constraints: NA Default Value: NA Presence: M

Discard Policy Mapping

This URI can be used to update service names and corresponding policy name for the service which is mapped to "ocDiscardPolicies" based on "policyName" and also to enable/disable the Overload Control feature and the sampling period in overload control. By default, the Overload Control feature is disabled and the sampling period is the 5s and max limit to 60s. To enable the feature, REST API needs to be invoked and update the enabled flag to true.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **IGW** option to configure the Ingress Gateway APIs. The **PLMN Ingress Gateway** and **N32 Ingress Gateway** appears underneath.
3. Click the **Discard Policy Mapping** option under **N32 Ingress Gateway** to manage discard policies. The **Discard Policy Mapping** page is displayed.
4. Click **Edit**. This opens the **Edit Discard Policy Mapping** page.
5. Click **Add** to add the mappings. The **Add Mappings** page is displayed.
6. Configure the fields under the **Add Mappings** page.
7. Click **Save** on the **Add Mappings** page to save the details.
8. Click **Save** on the **Edit Discard Policy Mapping** page to save the details.

Note

- Use the Edit icon available in the next column of the specific entry to update the policy information.
- Use Refresh icon to refresh the policy information.

Table 4-72 Discard Policy Mapping Configurations

Attribute Name	Description	Details
Enable Overload Control	Specifies whether to enable or disable overload control.	DataType: Boolean Constraints: NA Default Value: False
Sampling Period (in milliseconds)	Specifies the time frame for each cycle of overload control per service. Its default value is 5 Seconds. The max value is 60 seconds.	DataType: int Constraints: NA Default Value: 5 seconds

Table 4-73 Mappings Configurations

Attribute Name	Description	Details
Service Name	Specifies the name of the microservice that is further used to determine a mapping between service and discard policy name per service.	DataType: string Presence: M
Policy Name	Specifies the name of the discard policy that is used to determine a mapping between service and discard policy name per service. The drop-down list shows the policies configured using the Discard Policy page.	DataType: string The value can be: RejectWithErrorCode Presence: M

Server Header Details

This API can be used for adding Server Header in the error responses sent from Ingress Gateway. By default, this feature is disabled. To enable the feature, invoke the following REST API and update the enable switch.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **IGW** and navigate to **IGW** option to configure the Ingress Gateway APIs. The **PLMN Ingress Gateway** and **N32 Ingress Gateway** appears underneath.
3. Click **Server Header Details** option under **N32 Ingress Gateway**. The **Server Header Details** page is displayed.
4. Click **Edit** to update the Server Header Details. The **Edit Server Header Details** page is displayed.
5. Configure the fields in the **Edit Server Header Details** page.
6. Click **Save** on the **Edit Server Header Details** page to save the details. Click **Cancel** to discard your progress and go back to Server Header Details page.

Table 4-74 Server Header Details

Field Name	Description	Details
Enabled	Use the switch to enable or disable Server Header	DataType: boolean (true, false) Constraints: NA Default Value: NA
Error Code Series Id	Specify the error list ID.	DataType: string Constraints: NA Default Value: NA Presence: M
Configuration	Configure the following fields in Configuration section	DataType: NA Constraints: NA Default Value: NA

Table 4-74 (Cont.) Server Header Details

Field Name	Description	Details
NF Type	Specify the type of network function. In this case, it is SEPP.	DataType: string Constraints: NA Default Value: NA
NF Instance Id	Enter the SEPP instance ID. It represents the UUID of the SEPP deployment that is used to generate the Server Header.	DataType: integer Constraints: NA Default Value: NA

Note

Ensure that an errorCodeSeries exists corresponding to the errorCodeSeriesId.

4.2.15.1.1.2.1 Health Check Configuration

Health Check Configuration is used to enable Ingress gateway to understand incoming health API and respond with a configured response code to denote it is healthy.

Perform the following procedure to configure the Proactive Status Updates on the SEPP feature:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appear underneath. Select **Gateway** and navigate to the **IGW** option to configure the Ingress Gateway APIs. The **PLMN Ingress Gateway** and **N32 Ingress Gateway** appear underneath.
3. Click **Health Check Configuration** under **N32 Ingress Gateway**. The Health Check Configuration page is displayed.
4. Click the Edit icon to modify the **Health Check Configuration**. The **Health Check ConfigurationOptions** page appears.
5. Set **Enabled** to True or False.
6. Enter **Request URI**, **Response Code**, and **Request Method**.

Table 4-75 Health Check Configuration Parameters

Parameter Name	Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
enabled	Boolean	M	enable feature on IGW.
Request URI	String	M	URI to match with incoming health check URI. If matches with method, the configured response code sent If no match is found, the health check request is forwarded to the backend service.

Table 4-75 (Cont.) Health Check Configuration Parameters

Parameter Name	Data Type	Mandatory(M)/ Optional(O)/ Conditional(C)	Description
Request Method	String	M	method to match with the incoming health check Method. If matches with URI, the configured response code sent Only GET and OPTIONS are supported.
Response Code	Numerical	M	Response code to send when URI and Method are a match. Range of values: 200,201,202,203,204,205,206,207,208,226.

4.2.15.1.2 EGW (Egress Gateway)

This section contains the PLMN Egress Gateway and N32 Egress Gateway APIs.

4.2.15.1.2.1 PLMN Egress Gateway

This section contains the PLMN Egress Gateway APIs.

Congestion Level Configuration

This URI is used to configure the congestion level of the pods. The feature is disabled by default.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.
3. Click the **Congestion Config** option under **PLMN Egress Gateway** to configure the congestion level of the pods. The **Congestion Config** page is displayed.
4. Click **Edit** to edit the Congestion Level Configuration. The **Edit Congestion Config** page is displayed.
5. Configure the fields in the **Edit Congestion Config** page.
6. Click **Add** to add the configuration levels. The **Edit levels** page is displayed.
7. You can add **Level Name**, **Level Value**, and **Resources**.
8. Click **Add** to add the Resource parameters. The **Edit Resources** page is displayed.
9. Click **Save** on the **Edit Resources** page. Click **Cancel** to discard your progress and go back to the **Edit Congestion Config** page.

Table 4-76 Congestion Config Configuration

Field Name	Description	Details
Refresh Interval (seconds)	This attribute indicates the refresh interval for the scheduler to calculate congestion level in milliseconds. CPU levels are stable when tested with values more than 500 ms as a refresh interval. Any value less than this, CPU spikes are observed for a shorter period of time, and denied requests are processed based on the action defined for that congestion level.	Data Type: boolean Constraints: NA Default Value: false
Exception Type	Specific exception for which error profile name is configured	Data Type: string Constraints: NA Default Value: NA
Error Profile Name	Specific Name for a particular exception occurs	Data Type: string Constraints: NA Default Value: NA
Level Name	Indicates the name of the congestion level. Note: This value should be unique.	Data Type: String Constraints: NA Default Value: NA
Level Value	Indicates numerical values for each congestion level. The value ranges from 1-10. The values should be defined from 1 to 10 in increasing order. The default level of the system is 0. This is the state of the system when the CPU is less than the onset of L1 defined in the configuration.	Data Type: Integer Constraints: 1 -10 Default Value: 0
Resources	Indicates the list of resources.	Data Type: Object Constraints: NA Default Value: NA

Table 4-77 Resources

Field Name	Description	Details
Name	Indicates the resource names. The possible value is CPU.	Data Type: String Constraints: NA Default Value: CPU
Onset Threshold	Indicates the onset threshold. Note: Onset should be more than the abatement of the level. Denoted by percentage.	Data Type: Integer Constraints: NA Default Value: NA

Table 4-77 (Cont.) Resources

Field Name	Description	Details
Abatement Threshold	Indicates the abatement threshold. Denoted by percentage. Note: - Abatement should be less than the onset of the same level. - Abatement should be more than the onset of the previous level.	Data Type: Integer Constraints: NA Default Value: NA

Error Code Profile

This configuration is used to update the errorcodeprofiles that is used in Overload Control for populating details in error responses when a request is discarded.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configuration**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.
3. .
4. Click **Error Code Profiles** option under **PLMN Egress Gateway**. The **Error Code Profiles** page is displayed with default values for few fields.
5. Click **Add** to add the profiles.
The **Create Error Code Profiles** page is displayed.
6. Configure the fields in the **Create Error Code Profiles** page.
7. Click **Save** on the Create Error Code Profiles page to save the details. Click **Cancel** to discard your progress and go back to **Error Code Profiles** page.

Note

- Use the Edit icon available in the next column of the specific entry to update the route error profile information.
- Use Refresh icon to refresh the route error profile information.

Table 4-78 Error Code Profile Configuration

Field Name	Description	Details
Name	Indicate the error name.	Data Type: string Constraints: NA Default Value: NA Presence: M

Table 4-78 (Cont.) Error Code Profile Configuration

Field Name	Description	Details
Error Code	Error Code field in an errorScenario determines the HttpStatusCode that needs to be populated in ProblemDetails (HttpStatus field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType field.	DataType: integer Constraints: NA Default Value: NA Presence: M
Error Cause	Error Cause field in an errorScenario determines the error cause that needs to be populated in ProblemDetails (Cause field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter.	DataType: integer Constraints: NA Default Value: NA Presence: O
Error Title	Error Title field in an errorScenario determines the title that needs to be populated in ProblemDetails (Title field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter.	DataType: integer Constraints: NA Default Value: NA Presence: O
Redirect URL	Redirect URL field in an errorScenario determines the redirection URL, this value is populated in LOCATION header while sending response from Ingress Gateway. The header is populated only when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter, the errorCode configured for the particular errorScenario lies in 3xx error series and the redirectUrl field for the particular errorScenario is configured appropriately.	DataType: integer Constraints: NA Default Value: NA Presence: O

Table 4-78 (Cont.) Error Code Profile Configuration

Field Name	Description	Details
Retry After	Retry After field in an errorScenario determines the value in seconds or particular date after which the service should be retried, this value is populated in Retry-After header while sending response from Ingress Gateway. The header is populated only when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType parameter, the errorCode configured for the particular errorScenario lies in 3xx error series and the retry-after field for the particular errorScenario is configured appropriately in seconds.	DataType: integer Constraints: NA Default Value: NA Presence: O
Error Description	Error Description field in an errorScenario determines the description that needs to be populated in ProblemDetails (Detail field) response from Ingress Gateway when the exception occurred at Ingress Gateway matches the configured errorScenario's exceptionType field.	DataType: integer Constraints: NA Default Value: NA Presence: O

Pod Protection By Rate Limiting

Perform the following procedure to enable and identify the pods that must be protected by Rate Limiting feature.

1. In the **CNC Console**, click **SEPP** and then **Ingress Gateway**.
2. From the left navigation menu, navigate to **PLMN Ingress Gateway**.
3. Click the **Pod Protection By Rate Limiting** option under the **PLMN Ingress Gateway**. The **Pod Protection By Rate Limiting** page is displayed.
4. Click **Edit** to edit the configuration. The **Edit Pod Protection By Rate Limiting** page is displayed.
5. Configure the fields in the **Edit Pod Protection By Rate Limiting** page as mentioned in the following table.

Table 4-79 Pod Protection By Rate Limiting

Field Name	Description	Details
Enabled	The attribute indicates if rate limiting feature is enabled for the Ingress Gateway pod protection.	Data Type: Boolean Constraints: true or false Default Value: false

Table 4-79 (Cont.) Pod Protection By Rate Limiting

Field Name	Description	Details
Fill Rate	This attribute indicates the number of requests to be processed by an Ingress Gateway pod in one second (1000 ms).	Data Type: Integer Constraints: NA Default Value: 1000 Unit: milliseconds
Routes	This attribute indicates the route levels for the rate limiting.	Data Type: Array (object of Routes) Constraints: NA Default Value: <pre>[{ "Id": 1, "routeId": "", "Methods": "POST,GET", "Path": "/npcf- smpolicycontrol/*/sm- policies", "Percentage": 13.2, "defaultPriority": 13 }, { "Id": 2, "routeId": "", "Methods": "DELETE,PUT", "Path": "/npcf- smpolicycontrol/*/sm- policies", "Percentage": 10, "defaultPriority": }]</pre>
Error Code Profile	This attribute indicates the error profile to be sent for the rejected requests.	Data Type: String Constraints: HELM, REST, ERR_RATE_LIMIT Default Value: ERR_POP_503
Denied Request Actions	This attribute indicates the request actions to be performed for the rejected requests.	Data Type: Object Constraints: NA Default Value: See Denied Request Actions Table.
Priority Header Name	Determines the priority of the request using the configured header. Indicates the name of the header..	Data Type: String Constraints: NA Default Value: 3gpp-sbi-message-priority

Table 4-79 (Cont.) Pod Protection By Rate Limiting

Field Name	Description	Details
Default Priority	This attribute determines the priority of the request.	Data Type: Integer Constraints: NA Default Value: 24

6. Click **Add** in the Routes section to configure the different routes for **Pod Protection By Rate Limiting** feature.
7. The page is displayed.
8. Configure the fields in the Add Routes page as follows:

Table 4-80 Routes

Field Name	Description	Details
Id	Indicates the route ID that processes the request. The number of route IDs can be more than 1.	Data Type: Integer Constraints: NA Default Value: NA
Path	Indicates the path to be matched to apply fill rate on the route.	Data Type: String Constraints: NA Default Value: NA
Methods	Indicates the HTTP method of the route for which the rate limiting is defined. All valid HTTP methods are allowed, for example: • POST • PUT • PATCH • DELETE • GET	Data Type: String Constraints: • POST • PUT • PATCH • DELETE • GET Default Value: NA
Percentage	This attribute indicates the percentage of requests for a route. If the fill rate is 1000 and <code>routes.percentage</code> is 40%, the route process 400 requests in 1000 ms. The minimum value is 0.1. Note: The total percentage across all routes must not exceed 100.	Data Type: Integer Constraints: NA Default Value: NA
Default Priority	This attribute determines the priority of the request.	Data Type: Integer Constraints: NA Default Value: NA

9. Click **Save** on the Add Routes page to save the details. Click **Cancel** to discard your progress and go back to **Edit Pod Protection By Rate Limiting** page.

10. Click **Add** in the **Denied Request Actions** section to configure the actions for rejected requests.
11. The **Add Denied Request Actions** page is displayed.
12. Configure the fields in the Add Denied Request Actions page as follows:

Table 4-81 Denied Request Actions

Field Name	Description	Details
Id	Indicates the unique ID of the rejected request.	Data Type: Integer Constraints: NA Default Value: NA
Congestion Level	Indicates the congestion level of the pods. This value depends on the value field in the congestion level configuration or the default value 0. For more information about congestion level configuration, see Congestion Level Configuration.	Data Type: Integer Constraints: 1-10 Default Value: 0,1,2,3,4
Action	Rejects or processes a denied request based on the value of <code>congestionLevel</code> attribute in the <code>deniedRequestActions</code> object. The possible values are REJECT and CONTINUE.	Data Type: String Constraints: REJECT, CONTINUE Default Value: NA
Error Code Profile	Indicates a valid error code for the rejected requests based on the action.	Data Type: String Constraints: Any error code profile that is configured using <code>errorcodeprofiles</code> API Default Value: NA
Priority	Indicates the priority of the requests which should be processed for the CONTINUE action.	Data Type: Integer Constraints: NA Default Value: NA

13. Click **Save** on the **Add Denied Request Actions** page to save the details. Click **Cancel** to discard your progress and go back to **Edit Pod Protection By Rate Limiting** page.
14. Click **Save** on the **Edit Pod Protection By Rate Limiting** page to save the details. Click **Cancel** to discard your progress and go back to Pod Protection By Rate Limiting page.

Configurable Error Codes

This URI is used to configure an appropriate error response (error code, error cause, error title, and error description) for exceptions occurring at N32 Egress Gateway. The feature is disabled by default.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.
3. Click the **Configurable Error Codes** option under **PLMN Egress Gateway** to configure Error Code Profiles. The **Error Code Profiles** page is displayed.
4. Click **Add** to add the configurable error codes. The **Edit Configurable Error Codes** page is displayed.
5. Configure the fields in the **Edit Configurable Error Codes** page.
6. Click **Save** on the **Edit Configurable Error Codes** page to save the configurable error codes. Click **Cancel** to discard your progress and go back to the **Edit Error Code Profiles** page.

Table 4-82 Configurable Error Codes Configuration

Field Name	Description	Details
Enabled	Enables or disables configurable error codes functionality	DataType: boolean Constraints: NA Default Value: false
Exception Type	Specific exception for which error profile name is configured	DataType: string Constraints: NA Default Value: NA
Error Profile Name	Specific Name for a particular exception occurs	DataType: string Constraints: NA Default Value: NA

Peer Configuration

This configuration is used to add or update the list of peers wherein each peer consists of ID, host, port or virtualHost, and apiPrefix. The ID of each peer is mapped to Peer Identifier in Peer Set Configuration. The default value is null.

Perform the following configurations to configure the peers:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.
3. Click the **Peer Configuration** option under **PLMN Egress Gateway** to configure peers. The **Peer Configuration** page is displayed.

- Click **Add** to add the peer configuration. The **Create Peer Configuration** page is displayed.
- Configure the fields in the **Create Peer Configuration** page.
- Click **Save** on the **Create Peer Configuration** page to save the peer configuration. Click **Cancel** to discard your progress and go back to Peer Configuration page.

Note

- Use the Edit icon available in the next column of the specific entry to update the Peer Configuration information.
- Use Refresh icon to refresh the list of peers configured.

Table 4-83 Peer Configuration

Field Name	Description	Details
ID	Enter an unique peer identifier value.	DataType: string Constraints: NA Default Value: NA
Host	Host details of a local peer. It can be IPv4, IPv6 and FQDN details.	DataType: string Constraints: NA Default Value: NA
Port	Port details of a local host peer.	DataType: string Constraints: NA Default Value: NA
Virtual Host	Host details of a remote peer. This FQDN is sent to Alternate Route Service for DNS SRV resolution.	DataType: string Constraints: NA Default Value: NA
API Prefix	API prefix details of a peer. Note: The recommended value is / .	DataType: string Constraints: NA Default Value: NA
healthApiPath	Include the SEPP API details.	DataType: string Constraints: NA Default Value: /{seppApiRoot}/{apiVersion}/status

Peer Set Configuration

This configuration is used to add or update the list of peer sets wherein each peer set consists of **ID** and list of HTTP/HTTPS instances. Each instance consists of priority and peer identifier that is mapped to id in peerconfiguration resource. The **ID** of each peer set is mapped to **Peer Set Identifier** in **Routes Configuration** resource. The default value is null.

Perform the following configurations to configure the peer set:

- In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
- The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.

3. Click the **Peer Set Configuration** option under **PLMN Egress Gateway** to configure peers.
4. The **Peer Set Configuration** page is displayed.
5. Click **Add** to add the peer set configuration.
6. Configure the fields in the **Create Peer Set Configuration** page.
7. Click **Add** to add HTTP Configuration. The **Add HTTP Configuration** page is displayed.
8. In the **Add HTTP Configuration** page, configure the fields.
9. Click **Save** to save the details. Click **Cancel** to discard your progress, close the dialog box, and go back to **Create Peer Set Configuration** page.
10. Click **Add** to add HTTPS Configuration. The **Add HTTPS Configuration** page is displayed.
11. In the **Add HTTPS Configuration** page, configure the fields.
12. Click **Save** to save the details. Click **Cancel** to discard your progress, close the dialog box, and go back to **Create Peer Set Configuration** page.
13. Click **Save** on the **Create Peer Set Configuration** page to save the details. Click **Cancel** to discard your progress and go back to **Peer Set Configuration** page.

Note

- Use the **Edit** icon available in the next column of the specific entry to update the peer set configuration information.
- Use **Refresh** icon to refresh the list of peers configured.

Table 4-84 Peer Set Configuration

Field Name	Description	Details
ID	Enter the unique identifier for Peer Set.	DataType: string Constraints: Unique value in peer set configuration. Default Value: NA
HTTP Configuration	Configuration for HTTP based Peers. This value will be selected, if 3GPPAPIRootScheme value is http.	DataType: array () Constraints: NA Default Value: NA
HTTPS Configuration	Configuration for HTTPS based Peers. This value will be selected, if 3GPPAPIRootScheme value is https.	DataType: array () Constraints: NA Default Value: NA

Table 4-85 Peer Identifier Configuration

Field Name	Description	Details
Priority	Priority of peer to be used in a peer set.	DataType: integer Constraints: Priority must be unique. Default Value: NA

Table 4-85 (Cont.) Peer Identifier Configuration

Field Name	Description	Details
Peer Identifier	Peer identifier is the value of peer configured during PeerConfiguration.	Data Type: string Constraints: NA Default Value: NA

Routes Configuration

This configuration is used to fetch and update the list of routes configuration.

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.
3. Click the **Routes Configuration** option under **PLMN Egress Gateway** to configure peers. The **Routes Configuration** page is displayed.
4. Click **Add** to add the peer configuration. The **Create Routes Configuration** page is displayed.
5. Configure the fields in the **Create Routes Configuration** page.
6. Configure the fields in the **metadata** section.
7. Click **Add** in the right side column to add predicates. The **Add predicates** page is displayed.
8. Configure the fields in the **Add predicates** page .
9. Click **Save** to save predicates configuration. Click **Cancel** to discard your progress, close the window, and go back to **Create Routes Configuration** page.
10. Configure the fields in the **Filters section** as mentioned in the table.
11. Click **Add** in the right side column to add filters. The **Add Filters window** is displayed.
12. Configure the fields in the **Add Filters** window.
13. Click **Add** at the bottom of the window to save filters configuration. Click **Remove** to remove filter configuration.
14. Click **Save** in **Add Filters** window to save the details. Click **Cancel** to discard your progress, close the window, and go back to **Create Routes Configuration** page.
15. Click **Save** on the **Create Routes Configuration** page to save the details. Click **Cancel** to discard your progress and go back to **Routes Configuration** page.

Note

- Use the Edit icon available in the next column of the specific entry to update the routes configuration.
- Use Refresh icon to refresh the routes configuration.

Table 4-86 Routes Configuration

Field Name	Description	Details
ID	Indicates the unique route configuration identifier	DataType: string Constraints: Unique value of route Default Value: NA
URI	Provide any dummy URL, or leave the existing URL with the existing value.	DataType: string Constraints: NA Default Value: NA
Order	Provide the order of the execution of this route. Note: The value of the order attribute must be unique for each routing configuration.	DataType: integer Constraints: NA Default Value: NA
httpRuriOnly	This flag indicates the scheme of the outgoing request from OCSEPP. If the value is set to true, the scheme of RURI is changed to http. If the value is set to false, no change occurs to the scheme. Note: In case of non-ASM configuration and 3GPPAPIRootScheme in Roaming Options is set to https, set the value as false.	DataType: boolean (true,false) Constraints: NA Default Value: NA
httpsTargetOnly	For SEPP, the value of this flag must always be set to true. Note: This is a read-only attribute.	DataType: boolean (true, false) Constraints: NA Default Value: NA
sbiRoutingEnabled	Enables or disables SBI routing true: SbiRouting functionality is enabled false: SbiRouting functionality is disabled	DataType: boolean (true, false) Constraints: NA Default Value: false

Table 4-86 (Cont.) Routes Configuration

Field Name	Description	Details
Predicates	Header predicate details for matching target PLMN mapped to this SBIRoute rule. Note: The predicates can be combined in a single configuration as shown in the Body, or only the required configuration can be retained for processing the message. Sample value:- <pre>"predicates": [{ "args": { "header": "OC-MCCMNC", "regexp": "310014" }, "name": "Header" }]</pre> Note: "header": "OC-MCCMNC" must not be changed. Only "regexp": "310014" can be modified. regexp consists of MCC and MNC values. In this example, value of MCC and MNC is 310 and 014. Multiple values can be provided for regexp as shown below: <pre>"predicates": [{ "args": { "header": "OC-MCCMNC", "regexp": "310014" }, "name": "Header" }, { "args": { "header": "OC-MCCMNC", "regexp": "315012" }, "name": "Header" }]</pre>	DataType: Predicate structure. See description for more details. Constraints: NA Default Value: NA

Table 4-87 Filters Configuration

Field Name	Description	Details
Name	Provide filtername as "SbiRouting".	DataType: boolean (true, false) Constraints: NA Default Value: NA
PeerSetIdentifier	This flag maps to id of peerSetConfiguration.	DataType: String Constraints: NA Default Value: NA
CustomPeerSelectorEnabled	This flag allows the user to send request to a particular instance directly when enabled according to "ocalternaterouteattempt" header.	DataType: boolean (true, false) Constraints: NA Default Value: NA

SBI Error Action Sets

This configuration is used to list or update the error action set configuration at Egress Gateway. Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configuration**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.
3. Click **SBI Error Action Sets** option under **PLMN Egress Gateway** to configure action sets. The **SBI Error Action Sets** page is displayed.
4. Click **Add** to add an error action set configuration. The Create **SBI Error Action Sets** page is displayed.
5. Configure the fields in the **Create SBI Error Action Sets** page.
6. Configure the fields under the **Black List** section.
7. Click **Save** on the **Create SBI Error Action Sets** page to save the details. Click **Cancel** to discard your progress and go back to **SBI Error Action Sets** page.

Note

- Use the Edit icon available in the next column of the specific entry to update the action sets information.
- Use Refresh icon to refresh the list of action sets.

Table 4-88 SBI Error Action Sets

Field Name	Description	Details
ID	Indicate the unique Id for SBI routing error action set.	DataType: string Constraints: Default Value: NA
Action	Action that needs to be taken when specific criteria set is matched.	DataType: string Constraints: reroute, retry Default Value: reroute
Attempts	Maximum number of retries to either same or different peer in case of error or failures from backend.	DataType: string Constraints: NA Default Value: 3

Table 4-89 Blacklist

Field Name	Description	Details
Enabled	This flag enables the peer blacklist feature using the server headers received in the response.	DataType: boolean Constraints: true, false Default Value: false
Duration	The duration for which the peer is blacklisted and no traffic is routed to that peer for this period.	DataType: integer Constraints: NA Default Value: 60000

SBI Error Criteria Sets

This URI is used to list or update SBI error criteria sets configuration at Egress Gateway. By default, this configuration is disabled.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.
3. Click the **SBI Error Criteria Sets** option under **PLMN Egress Gateway** to configure peers. The **SBI Error Criteria Sets** page is displayed.
4. Click **Add** to add an error action set configuration. The **Create SBI Error Criteria Sets** page is displayed.
5. Configure the fields in the **Create SBI Error Criteria Sets** page.
6. Click **Save** in **Add Statuses** window to save statuses configuration. Click **Cancel** to discard your progress, close the window, and go back to **Create SBI Error Criteria Sets** page.
7. Click **Add** in **Cause** section to configure cause details. Configure the fields under Cause section.
8. **Headers Matching Script**: Enter headers matching script.
9. Click **Save** on the **Create SBI Error Criteria Sets** page to save the details. Click **Cancel** to discard your progress and go back to **Create SBI Error Criteria Sets** page.

 Note

- Use Edit icon available in the next column of the specific entry to update the SBI Error Criteria Sets information.
- Use Refresh icon to refresh the configuration.

10. Click **Add** in **Cause** section to configure cause details. Configure the fields under **Cause** section.
11. Enter headers matching script.
12. Click **Save** on the **Create SBI Error Criteria Sets** page to save the details. Click **Cancel** to discard your progress and go back to **Create SBI Error Criteria Sets** page.

 Note

- Use Edit icon available in the next column of the specific entry to update the SBI Error Criteria Sets information.
- Use Refresh icon to refresh the configuration.

Table 4-90 SbiErrorCriteriaSets

Field Name	Description	Details
ID	Enter an unique ID for SBI routing error action set	DataType: Integer Constraints: NA Default Value: NA
Method	Indicate the action that needs to be taken when specific criteria set is matched.	DataType: String Constraints: NA Default Value: NA
Exceptions	Enter the maximum number of retries to either same or different peer in case of error or failures from backend.	DataType: Integer Constraints: NA Default Value: NA
Response	Configure the following fields under Statuses section. Click Add to add HTTP status details	DataType: String Constraints: NA Default Value: NA
Status Series	Enter the HTTP status series for which reroute or retry is triggered, when the error response is received from downstream.	DataType: Integer Constraints: NA Default Value: NA
Status	Specify HTTP statuses that belongs to above mentioned status series for which reroute or retry is triggered. To enable retry or reroute for all the HTTP status belonging to a status series, configure this as -1.	DataType: Integer Constraints: NA Default Value: NA

Table 4-91 Cause Configuration

Field Name	Description	Details
ignoreCauselfMissing	Enable the switch to ignore cause if it is missing.	DataType: boolean (true, false) Constraints: NA Default Value: NA
Path	Path of the cause.	DataType: String Constraints: NA Default Value: NA
Reason	Reason of the cause.	DataType: string Constraints: NA Default Value: NA

Retry Profile

This URI is used to configure the conditions for retry to particular peer. If a request to a particular peer fails, the PLMN Egress Gateway then checks this retryProfile and then retry the request till the retry count exhausts or we receive success response. For the first request, Retry Profile configurations will be considered. These configurations can also be done at N32 Egress Gateway and can be used if we did not receive any retry profile and default retry is enabled at EGW.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.
3. Click the **Retry Profile** option under **PLMN Egress Gateway** to configure retry profile. The **Retry Profile** page is displayed.
4. Click **Add** to add the retry profile configuration. The **Edit Retry Profile** page is displayed.
5. Configure the fields in the **Create Retry Profile** page.
6. Click **Save** on the **Create Retry Profile** screen to save the details. Click **Cancel** to discard your progress and go back to **Edit Retry Profile** screen.

Table 4-92 Retry Profile Configurations

Field Name	Description	Details
retryCount	The number of retries which happens at EGW when a request fails. (This count is excluding the first request. Example: retryCount is 3 then a total of 4 attempts will be made)	DataType: INTEGER Constraints: NA Default Value: 3
requestTimeout	This is the configuration for request timeout for the call that happened from EGW to producer NF.	DataType: INTEGER Constraints: NA Default Value: 3,000
statuses	This configuration is the list of HTTP error codes to match when we get the response from producer NF. It can be defined as either 4xx or a specific error code like 404. We should not define both like ["4xx", "400"] . If defined then which condition is first met will be served. If any unexpected error occurs while reading retryProfile shared by the backend service then the normal call flow would continue and retry conditions will not be considered.	DataType: LIST Constraints: NA Default Value: 4xx, 5xx

Table 4-92 (Cont.) Retry Profile Configurations

Field Name	Description	Details
exceptions	This configuration is the list of exceptions to match when we get the response from producer NF. Following is the set of possible exceptions: • java.net.UnknownHostException • javax.net.ssl.SSLHandshakeException • java.nio.channels.ClosedChannelException • java.net.ConnectException • java.net.SocketTimeoutException • java.util.concurrent.TimeoutException • java.util.concurrent.RejectedExecutionException	DataType: LIST Constraints: NA Default Value: java.net.ConnectException

Note

If SBI routing is enabled for the route then this default route retry will not work even if we receive the retry profile as part of the request header for that route.

4.2.15.1.2.2 N32 Egress Gateway

This section contains the N32 Egress Gateway APIs.

SBI Error Criteria Sets

This URI is used to list or update SBI error criteria sets configuration at Egress Gateway. By default, this configuration is disabled.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.
3. Click the **SBI Error Criteria Sets** option under **N32 Egress Gateway** to configure peers. The **SBI Error Criteria Sets** page is displayed.
4. Click **Add** to add an error action set configuration. The **Create SBI Error Criteria Sets** page is displayed.
5. Configure the fields in the **Create SBI Error Criteria Sets** page.
6. Click **Save** in **Add Statuses** window to save statuses configuration. Click **Cancel** to discard your progress, close the window, and go back to **Create SBI Error Criteria Sets** page.

7. Click **Add** in **Cause** section to configure cause details. Configure the fields under Cause section.
8. Headers Matching Script: Enter headers matching script.
9. Click **Save** on the **Create SBI Error Criteria Sets** page to save the details. Click **Cancel** to discard your progress and go back to **Create SBI Error Criteria Sets** page.

Note

- Use Edit icon available in the next column of the specific entry to update the SBI Error Criteria Sets information.
- Use Refresh icon to refresh the configuration.

10. Click **Add** in **Cause** section to configure cause details. Configure the fields under **Cause** section.
11. Enter headers matching script.
12. Click **Save** on the **Create SBI Error Criteria Sets** page to save the details. Click **Cancel** to discard your progress and go back to **Create SBI Error Criteria Sets** page.

Note

- Use Edit icon available in the next column of the specific entry to update the SBI Error Criteria Sets information.
- Use Refresh icon to refresh the configuration.

Table 4-93 SbiErrorCriteriaSets

Field Name	Description	Details
ID	Enter an unique ID for SBI routing error action set	DataType: Integer Constraints: NA Default Value: NA
Method	Indicate the action that needs to be taken when specific criteria set is matched.	DataType: String Constraints: NA Default Value: NA
Exceptions	Enter the maximum number of retries to either same or different peer in case of error or failures from backend.	DataType: Integer Constraints: NA Default Value: NA
Response	Configure the following fields under Statuses section. Click Add to add HTTP status details	DataType: String Constraints: NA Default Value: NA
Status Series	Enter the HTTP status series for which reroute or retry is triggered, when the error response is received from downstream.	DataType: Integer Constraints: NA Default Value: NA

Table 4-93 (Cont.) SbiErrorCriteriaSets

Field Name	Description	Details
Status	Specify HTTP statuses that belongs to above mentioned status series for which reroute or retry is triggered. To enable retry or reroute for all the HTTP status belonging to a status series, configure this as -1.	DataType: Integer Constraints: NA Default Value: NA

Table 4-94 Cause Configuration

Field Name	Description	Details
ignoreCauselfMissing	Enable the switch to ignore cause if it is missing.	DataType: boolean (true, false) Constraints: NA Default Value: NA
Path	Path of the cause.	DataType: String Constraints: NA Default Value: NA
Reason	Reason of the cause.	DataType: string Constraints: NA Default Value: NA

SBI Error Action Sets

This configuration is used to list or update the error action set configuration at Egress Gateway. Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.
3. Click **SBI Error Action Sets** option under **N32 Egress Gateway** to configure action sets. The **SBI Error Action Sets** page is displayed.
4. Click **Add** to add an error action set configuration. The Create **SBI Error Action Sets** page is displayed.
5. Configure the fields in the **Create SBI Error Action Sets** page.
6. Configure the fields under the **Black List** section.
7. Click **Save** on the **Create SBI Error Action Sets** page to save the details. Click **Cancel** to discard your progress and go back to **SBI Error Action Sets** page.

Note

- Use the Edit icon available in the next column of the specific entry to update the action sets information.
- Use Refresh icon to refresh the list of action sets.

Table 4-95 SBI Error Action Sets

Field Name	Description	Details
ID	Indicate the unique Id for SBI routing error action set.	DataType: string Constraints: Default Value: NA
Action	Action that needs to be taken when specific criteria set is matched.	DataType: string Constraints: reroute, retry Default Value: reroute
Attempts	Maximum number of retries to either same or different peer in case of error or failures from backend.	DataType: string Constraints: NA Default Value: 3

Table 4-96 Blacklist

Field Name	Description	Details
Enabled	This flag enables the peer blacklist feature using the server headers received in the response.	DataType: boolean Constraints: true, false Default Value: false
Duration	The duration for which the peer is blacklisted and no traffic is routed to that peer for this period.	DataType: integer Constraints: NA Default Value: 60000

Configurable Error Codes

This URI is used to configure an appropriate error response (error code, error cause, error title, and error description) for exceptions occurring at N32 Egress Gateway. The feature is disabled by default.

Perform the following configurations:

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**.
2. The **Gateway** and **cnDBTier** appears underneath. Select **Gateway** and navigate to **EGW** option to configure the Egress Gateway APIs. The **PLMN Egress Gateway** and **N32 Egress Gateway** appears underneath.
3. Click the **Configurable Error Codes** option under **PLMN Egress Gateway** to configure Error Code Profiles. The **Error Code Profiles** page is displayed.
4. Click **Add** to add the configurable error codes. The **Edit Configurable Error Codes** page is displayed.
5. Configure the fields in the **Edit Configurable Error Codes** page.
6. Click **Save** on the **Edit Configurable Error Codes** page to save the configurable error codes. Click **Cancel** to discard your progress and go back to the **Edit Error Code Profiles** page.

Table 4-97 Configurable Error Codes Configuration

Field Name	Description	Details
Enabled	Enables or disables configurable error codes functionality	DataType: boolean Constraints: NA Default Value: false
Exception Type	Specific exception for which error profile name is configured	DataType: string Constraints: NA Default Value: NA
Error Profile Name	Specific Name for a particular exception occurs	DataType: string Constraints: NA Default Value: NA

Note

Modifying the `sbiRoutingErrorActionSets` and `sbiRoutingErrorCriteriaSets` is not supported directly. If users wish to update these configurations, they can edit the `ocsepp_custom_values_<version>.yaml` file by adjusting the parameters `sbiRoutingErrorCriteriaSets` and `sbiRoutingErrorActionSets` under the `config-mgr-svc` section, then perform a helm upgrade.

4.2.15.1.3 MultiFQDN Configurations

MultiFQDN Configurations allows you to configure multiple Fully Qualified Domain Names (FQDNs) and their associated certificates for Hosted SEPP deployments. Configure the FQDN details for both the N32 Ingress Gateway and N32 Egress Gateway to ensure that the appropriate certificate is selected during the TLS handshake.

Perform the following procedure to configure MultiFQDN in N32 Egress Gateway:

1. From the left navigation menu, navigate to **SEPP**, and then click **Configurations** and select **Gateway**.
2. Navigate to **N32 Egress Gateway**.
3. Configure the MultiFQDN settings for N32 Egress Gateway.
4. Set `n32-egress-gateway.multiFqdn.enabled` to true to enable multiple FQDN support.
5. Add an FQDN details entry for each FQDN or Subject Alternative Name (SAN) that requires a separate certificate configuration.
6. Specify the certificate, private-key, CA-root, keystore-password, and truststore-password files for each FQDN. The specified files must be available in the corresponding secret.
7. Click Save to apply the configuration.

Perform the following procedure to configure MultiFQDN N32 Ingress Gateway:

1. From the left navigation menu, navigate to **SEPP**, and then click **Configurations** and select **Gateway**.
2. Navigate to **N32 Ingress Gateway**.
3. Configure the MultiFQDN settings for the N32 Ingress Gateway.
4. Set `multiFqdn.enabled` to true to enable multiple FQDN support.

5. Add an FQDN details entry for each FQDN or Subject Alternative Name (SAN) that requires a separate certificate configuration.
6. Specify the certificate, private-key, CA-root, keystore-password, and truststore-password files for each FQDN. The specified files must be available in the corresponding secret.
7. Click Save to apply the configuration.

Note

- The certificate, key, CA root, and password files must be available at the expected mounted or configured location before saving the Multi FQDN configuration.

Table 4-98 Parameters

Parameter Name	Description	Details
fqdn	This is a conditional parameter. Specifies the FQDN or SAN for which certificates must be configured.	Datatype: String Range: String Default Value: dummy.com
caroot_file	This is a conditional parameter. Specifies the CA root file name for the corresponding FQDN. The file must be present in the secret.	Datatype: String Range: String Default Value: dummy.com_caroot.cer
key_store_passwd_file	This is a conditional parameter. Specifies the keystore password file name for the corresponding FQDN. The file must be present in the secret.	Datatype: String Range: String Default Value: dummy.com_key.txt
ocsepp_cert_file	This is a conditional parameter. Specifies the Oracle Communications SEPP certificate file name for the corresponding FQDN. The file must be present in the secret.	Datatype: String Range: String Default Value: dummy.com_ocsepp.cer
rsa_certificate_file	This is a conditional parameter. Specifies the RSA certificate file name for the corresponding FQDN. The file must be present in the secret.	Datatype: String Range: String Default Value: dummy.com_rsa_certificate.crt
rsa_private_key_file	This is a conditional parameter. Specifies the RSA private-key file name for the corresponding FQDN. The file must be present in the secret.	Datatype: String Range: String Default Value: dummy.com_rsa_private_key_pkcs1.pem
ssl_ecdsa_certificate_file	This is a conditional parameter. Specifies the SSL ECDSA certificate file name for the corresponding FQDN. The file must be present in the secret.	Datatype: String Range: String Default Value: dummy.com_ssl_ecdsa_certificate.crt

Table 4-98 (Cont.) Parameters

Parameter Name	Description	Details
ssl_ecdsa_private_key_file	This is a conditional parameter. Specifies the SSL ECDSA private-key file name for the corresponding FQDN. The file must be present in the secret.	Datatype: String Range: String Default Value: dummy.com_ssl_ecdsa_private_key.pem

4.2.15.2 cnDBTier

Perform the following procedure to view the cnDBTier version, status of cnDBTier clusters, and georeplication status on the CNC Console. The cnDBTier APIs are read only APIs that can be viewed in CNC Console.

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**. The **Gateway** and **cnDBTier** appears underneath.
2. click **cnDBTier** tab. The **cnDBTier** page is displayed.
3. **IP Preference for Replication Channel**
 - a. From the left navigation menu, navigate to **cnDBTier** and then click **IP Preference for Replication Channel**. The **IP Preference for Replication Channel** page is displayed.
 - b. Click **Actions** to view the IP preference set for the sites. The **Edit IP Preference for Replication Channel** page is displayed.

Table 4-99 IP Preference for Replication Channel

Attribute	Description
Local Site Name	This attribute displays the name of the current site to which NRF is connected.
Remote Site Name	This attribute displays the remote site name.
Preferred Family Preferred IP Family Type	This attribute displays the current site's preferred IP family type. IPV4- Indicates the IP Type selected is IPV4. IPV6- Indicates the IP Type selected is IPV6.
Status	This attribute displays the current site's preferred IP family status.

Gracefully Stop and Start Georeplication

- a. From the left navigation menu, navigate to **cnDBTier** and then click **Gracefully Stop and Start Georeplication**. The **Gracefully Stop and Start Georeplication** page is displayed.
- b. Click **Stop Georeplication** from the left navigation menu to stop the georeplication between the sites. The **Edit Gracefully Stop Georeplication** page is displayed.

Table 4-100 Gracefully Stop and Start Georeplication

Attribute	Description
Site Name	This attribute displays the name of the current site to which NRF is connected.
Remote Site Name	This attribute displays the list of the remote site name.
Replication Channel Group Ids	This attribute displays the list of replication group ids of the current site.
Georeplication Details	This attribute displays the details of the georeplication sites and their status.

Georeplication Details**Table 4-101 Georeplication Details**

Attribute	Description
Site Name	This attribute displays the name of the current site to which NRF is connected.
Remote Site Names	This attribute displays the name of the remote site.
Replication Channel Group Id	This attribute displays the replication group id.
Replication Status	This attribute displays the status of the georeplication between sites. - UP- Indicates the georeplication between the sites is up. - DOWN- Indicates the georeplication between the sites is down.
Switch Over Status	This attribute displays the switch-over status of the georeplication between sites. - SWITCH_OVER_ENABLED- Indicates the switch-over status of the georeplication between the sites is up. - SWITCH_OVER_DISABLED- Indicates the switch-over status of the georeplication between the sites is disabled.

3. Click **Stop Georeplication** to stop the replication between the sites.

Start Georeplication

- a. From the left navigation menu, navigate to **to cnDBTier** and then click **Gracefully Stop and Start Georeplication**. The **Gracefully Stop and Start Georeplication** page is displayed.
- b. Click **Start Georeplication** from the left navigation menu to start the georeplication between the sites. The **Start Georeplication** page is displayed.

Table 4-102 Start Georeplication

Attribute	Description
Site Name	This attribute displays the name of the current site to which NRF is connected.

Table 4-102 (Cont.) Start Georeplication

Attribute	Description
Remote Site Name	This attribute displays the list of the remote site name.
Replication Channel Group Ids	This attribute displays the list of replication group ids of the current site.
Georeplication Details	This attribute displays the details of the georeplication sites and their status.

Georeplication Details**Table 4-103 Georeplication Details**

Attributes	Description
Site Name	This attribute displays the name of the current site to which NRF is connected.
Remote Site Names	This attribute displays the name of the remote site.
Replication Channel Group Id	This attribute displays the replication group id.
Replication Status	This attribute displays the status of the georeplication between sites. • UP- Indicates the georeplication between the sites is up. • DOWN- Indicates the georeplication between the sites is down.
Switch Over Status	This attribute displays the switch-over status of the georeplication between sites. • SWITCH_OVER_ENABLED- Indicates the switch-over status of the georeplication between the sites is up. • SWITCH_OVER_DISABLED- Indicates the switch-over status of the georeplication between the sites is disabled.

3. Click **Start Georeplication** to start the replication between the sites.

Georeplication Recovery Across All Sites

4. Click **Georeplication Status Across All Sites** to access the georeplication status across all configured sites in the cnDBTier cluster.
The **Georeplication Status Across All Sites** page is displayed.

Table 4-104 Georeplication Status Across All Sites

Attribute	Description
Remote Site Name	This field displays the remote site involved in replication with the local site.
Replication Details Per Group ID	This field displays the list of replication-group-specific details with the site name and the remote site name.
Site Name	This field displays the primary site for which the georeplication status is being reported.

Table 4-104 (Cont.) Georeplication Status Across All Sites

Attribute	Description
Site To Mate Site Replication Status	This field displays the overall georeplication status with all the groups.

- Click the view icon option to view the **View Georeplication Status Across All Sites** attributes. The **View Georeplication Status Across All Sites** page is displayed.

Table 4-105 View Replication Details Per Group ID

Attribute	Description
All SQL Status Details	This field displays the detailed SQL and replication state information for all SQL nodes for the specified replication group.
Replication Channel Group Status	This field displays the replication status for the specific replication group.
Replication Group ID	This field displays the unique identifier assigned to the replication group.

- Click the view icon option to view the **View All SQL Status Details** attributes. The **View All SQL Status Details** page is displayed.

Attribute	Description
SQL Node	This field displays the SQL node whose replication metrics are being reported.
Source Host	This field displays the remote host from which the SQL node pulls replication data.
External IP	This field displays the external endpoint through which the SQL node is accessible within the site.
Replica I/O Running Status	This field displays whether the replication I/O thread is currently active or not. Possible values: Yes or No.
Replica SQL Running Status	This field displays whether the SQL thread responsible for applying replicated changes is running or not. Possible values: Yes or No.
Server ID	This field displays the unique server identifier used in the replication topology.
Role	This field displays the role of the SQL node within the replication setup (for example, ACTIVE or STANDBY).
Last Error Code	This field displays the most recent SQL thread error code encountered during replication.
Last Error Message	This field displays the most recent error message reported by the SQL thread.
Last I/O Error Code	This field displays the most recent I/O thread error code encountered during replication.
Last I/O Error Message	This field displays the most recent error message reported by the I/O thread.
Last SQL Error Code	This field displays the most recent SQL execution error code encountered while applying replicated transactions.
Last SQL Error Message	This field displays the last SQL execution error message reported.

Attribute	Description
SQL Remaining Delay	This field displays the replication lag or delay (if any) between the source site and this SQL node. Unit: seconds.

cnDBTier Backup Status

- Click the **cnDBTier Backup Status** to view cnDBTier backup status. The **cnDBTier Backup Status** page is displayed.

Note

The parameters listed in the below table are read-only.

Table 4-106 cnDBTier Backup Status

Attribute	Description
Current Timestamp	This attribute displays the current system time.
Is Backup In Progress	This attribute displays the current cnDBTier backup status.
Next Backup Scheduled Timestamp	This attribute displays the time for the next scheduled backup.

cnDBTier Health

- Click the **Backup List** to view the list of completed backups along with backup ID, backup size, and backup creation timestamp. The **Backup List** page is displayed.

Note

The following are read-only APIs.

Table 4-107 Backup List

Fields	Description
Site Name	This attribute displays the name of the current site to which SEPP is connected.
Backup Details	This attribute displays the following information like backup id, backup size, and backup creation timestamp.
Backup Id	This attribute displays the ID of the stored backup.
Backup Size (bytes)	This attribute displays the size of the stored backup.
Creation TimeStamp	This attribute displays the time recorded when the backup was stored.

- Click **cnDBTier Health** to view the health status of the microservices such as replication, backup manager, monitor services, NDB services. The **cnDBTier Health** page is displayed.

- a. Click the **Backup Manager Health Status** to view the health status of the backup manager.
The **Backup Manager Health Status** page is displayed.

Note

The following are read-only APIs.

Table 4-108 Backup Manager Health Status

Attribute	Description
Service Name	This attribute displays the service name of the backup manager microservice.
Service Status	This attribute displays the service status of the backup manager microservice. Possible values are UP and DOWN.
DB Connection Status	This attribute displays the database connection status of the backup manager microservice. Possible values are UP and DOWN.
Overall Backup Manager Service Health	This attribute displays the overall health status of the backup manager microservice. Possible values are UP and DOWN.
Backup Executor Health Status	This attribute displays the following information like node id and DB connection status of the backup executor.
Node Id	This attribute displays the id of the node.
DB Connection Status	This attribute displays the backup executor database connection status with the nodes. Possible values are UP and DOWN.

- b. Click the **Monitor Health Status** to view the health status of the services.
The **Monitor Health Status** page is displayed.

Note

The following are read-only APIs.

Table 4-109 Monitor Health Status details

Attribute	Description
Service Name	This attribute displays the service name of the monitor microservice.
DB Connection Status	This attribute displays the database connection status of the monitor microservice. Possible values are UP and DOWN.
Metric Scrape Status	This attribute displays the status of the metric scrape, that is if the metrics are fetched or not. If the metrics are fetched then the service is up and vice versa. Possible values are UP and DOWN.

Table 4-109 (Cont.) Monitor Health Status details

Attribute	Description
Overall Monitor Service Health	This attribute displays the overall health status of the monitor microservice. Possible values are UP and DOWN.

- c. Click the **NDB Health Status** to view the health status of the network database. The **NDB Health Status** page is displayed.

Note

The following are read-only APIs.

Table 4-110 NDB Health Status details

Attribute	Description
Local Site Name	This attribute displays the name of the current site (Site 1, Site 2).
NDB Health Status Details	This attribute displays the health status of the network database such as the name of the NDB service, status of the service, health status of PVC.
Service Name	This attribute displays the service name.
Service Status	This attribute displays the status of the service. Possible values are UP and DOWN.
PVC Health Status	This attribute displays the health status of the PVC. Possible values are UP, DOWN, and NA. Note: This attribute is set to NA when some of the database pods are not connected to the PVC.

- d. Click the **Replication Health Status** to view the health status of the replication sites. The **Replication Health Status** page is displayed.

Note

The following are read-only APIs.

Table 4-111 Replication Health Status details

Attribute	Description
Local Site Name	This attribute displays the name of the current site (Site 1, Site 2).

Table 4-111 (Cont.) Replication Health Status details

Attribute	Description
Health Status Details	This attribute displays the health status details of the local site like replication service name, replication service status, database connection status of the replication service, and the overall health status of the replication microservices. The number of rows in this table varies depending on the type of deployment (for example, two-site, three-site deployments).
Service Name	This attribute displays the name of the available replication service.
Service Status	This attribute displays the status of the available replication service. Possible values are UP and DOWN.
DB Connection Status	This attribute displays the database connection status of the replication microservice. Possible values are UP and DOWN.
Overall Replication Service Health	This attribute displays the overall health status of the replication microservice. Possible values are UP and DOWN.

10. Click **cnDBTier Version** to view the version.
The **cnDBTier Version** page is displayed.

Note

The following are read-only APIs.

Table 4-112 cnDBTier Version Attributes

Attribute	Description
cnDBTier Version	This attribute displays the cnDBTier version.
NDB Version	This attribute displays the network database (NDB) version.

11. Click the **Database Statistics Report** to view the available databases.
The **Database Statistics Report** page is displayed.

Note

The following are read-only APIs.

Table 4-113 Database Statistics Report

Fields	Description
Database Count	This attribute displays the number of available database.

Table 4-113 (Cont.) Database Statistics Report

Fields	Description
Database Tables Count	This attribute displays the available database names and their table count.
Database Name	This attribute displays the database name.
Table Count	This attribute displays the table count for each database.
Database Table Rows Count	This attribute displays the rows present in each table.
Database Name	This attribute displays the database name.



- a. Click the  icon available next to the database name to view the **View Database Table Rows Count** screen.
The **View Database Table Rows Count** page is displayed.

Table 4-114 View Database Table Rows Count

Fields	Description
Database Name	This attribute displays the database name.
Tables	This attribute displays the table names and the corresponding rows in each table.
Table Name	This attribute displays the table name.
Row Count	This attribute displays the table rows present in each table.

12. Click **Georeplication Recovery** to view the Georeplication Recovery Status of cnDBTier cluster. The **Georeplication Recovery** page is displayed. The **Update Cluster As Failed**, **Start Georeplication Recovery**, and **Georeplication Recovery Status** appears underneath the **Georeplication Recovery Status** on the left menu.
 - Click the **Update Cluster As Failed** to mark the cluster as FAILED.
The **Update Cluster As Failed** page is displayed.

Table 4-115 Update Cluster As Failed

Fields	Description
Cluster Names	This field displays a list of cnDBTier clusters that can be marked as failed.
Failed Cluster Names	This field displays a cnDBTier cluster that is marked as failed.

Click **Update Cluster**. The selected cluster name is updated in the **Failed Cluster Names** field.

13. Click the **Start Georeplication Recovery** to start the georeplication recovery process for a failed site.
The **Start Georeplication Recovery** page is displayed.

Table 4-116 Start Georeplication Recovery

Attribute	Description
Failed Cluster Name	This field displays a list of all the clusters that have been marked as failed.
Backup Cluster Name (Optional)	This field displays a list of all the healthy clusters. If no cluster is selected, the system uses the first available healthy cluster for the backup.

14. Click **Start Georeplication Recovery** to initiate georeplication recovery.
15. Click the **Georeplication Recovery Status** to view the status of georeplication recovery for cnDBTier clusters.
The **Georeplication Recovery Status** page is displayed.

Table 4-117 Georeplication Recovery Status

Attribute	Description
Local Cluster Name	This field displays the name of the local cluster.
Georeplication Recover Status Details	This field displays the details of the georeplication recovery status of cnDBTier clusters.
Cluster Name	This field displays the clusters by name.
Georeplication Recovery Status	This field displays the current georeplication recovery status of the corresponding cluster.

The following are the statuses of Georeplication Recovery:

Table 4-118 Georeplication Recovery Status

Georeplication Recovery Status	Description
ACTIVE	The cluster is in a healthy state, and replication is up and running with its respective mate cluster.
REINSTALLED	The cluster enters this state during fatal error recovery when the end user reinstalls the cluster.
STARTDRRESTORE	When Georeplication recovery is started, the cluster will transition into this state.
INITIATEBACKUP	When Georeplication recovery is started, the cluster will identify a healthy cluster for backup initiation and transition into this state.
CHECKBACKUP	When the backup is initiated, the georeplication recovery cluster will monitor the progress of the backup until its completion. If the backup fails, the cluster will restart the backup.
COPY_BACKUP	Upon completion of the backup, the georeplication recovery cluster will request the transfer of the backup from the healthy cluster to the georeplication recovery cluster.
CHECK_BACKUP_COPY	When backup copy is started georeplication recovery cluster will monitor for the backup transfer progress till it's completion and if it's fails the cluster will re-initiates the backup transfer.
BACKUPCOPIED	When the backup copy is started, the georeplication recovery cluster will monitor the progress of the backup transfer until its completion. If the transfer fails, the cluster will restart the backup transfer.

Table 4-118 (Cont.) Georeplication Recovery Status

Georeplication Recovery Status	Description
BACKUPEXTRACTED	This state indicates that the backup has been successfully extracted at the georeplication recovery cluster, allowing the restoration of the backup to start.
FAILED	This state is used by end user to mark specific cluster as failed and hence georeplication recovery is essential to recover the cluster. This state can also indicate that georeplication recovery started and the database is restored using the healthy cluster backup.
UNKNOWN	This state is used by the end user to mark a specific cluster as failed, necessitating georeplication recovery for cluster recovery. Additionally, this state can indicate that georeplication recovery has started and the database has been restored using the backup from the healthy cluster.
RECONNECTSQLNODES	This state is used to instruct SQL nodes to be offline during backup restoration to prevent any records from entering the binlog of the georeplication recovery cluster.
BACKUPRESTORE	This state indicates that the backup, successfully copied from the healthy cluster, is currently being used to restore the georeplication recovery cluster.
RESTORED	When the backup is successfully restored in the georeplication recovery cluster, the cluster will enter this state to start the reestablishment of replication channels.
BINLOGINITIALIZED	This state indicates the start of binlogs for the restoration of replication channels, necessary to start the restore process
RECONFIGURE	When the binlog is restarted, the georeplication recovery cluster will reestablish the replication channels with respect to all its mate clusters.

16. Click **Georeplication Status** to view the local site and remote site name to which SEPP is connected.

The **Georeplication Status** page is displayed.

Note

The following are read-only APIs.

Table 4-119 Georeplication Status

Attribute	Description
Local Site Name	This attribute displays the local site name to which SEPP is connected. Note: The number of local site names may vary depending on the type of georeplication used in SEPP.
Remote Site Name	This attribute displays the remote site name. Note: The number of remote site names may vary depending on the type of georeplication used in SEPP.
Replication Status	This attribute displays the replication status with corresponding sites. Note: The number of replication statuses may vary depending on the type of georeplication used in SEPP.

Table 4-119 (Cont.) Georeplication Status

Attribute	Description
Seconds Behind Remote Site	This attribute displays the number of seconds that the last record read by the local site is behind the latest record written by the remote site for all the replication groups. Note: The number of replication statuses may vary depending on the type of georeplication used in SEPP.



- a. Click the icon in the **Actions** menu to view the **View Georeplication Status** screen.
The **Georeplication Status** page is displayed.

Table 4-120 Georeplication Status

Attribute	Description
Replication Group Delay	This attribute displays the delay in seconds for the remote site for individual replication groups.
Replication Channel Group Id	This attribute displays the ID of the replication channel group.



- b. Click the icon to view the **Replication Group Delay** attributes.
The **Replication Group Delay** page is displayed.

Table 4-121 View Replication Group Delay

Attribute	Description
Channel Details	This attribute displays the channel details such as Remote Replication IP and Role.
Remote Replication IP	This attribute displays the IP of the remote replication channel.
Role	This attribute displays the role of the replication channel IP.

17. Click the **HeartBeat Status** to view the connectivity between local site and remote site to which SEPP is connected.
The **HeartBeat Status** page is displayed.

Note

The following are read-only APIs.

Table 4-122 HeartBeat Status Details

Fields	Description
Site Name	This attribute displays the name of the current site to which SEPP is connected.

Table 4-122 (Cont.) HeartBeat Status Details

Fields	Description
HeartBeat Details	This attribute displays the following information such as the remote site name, heart beat status, heart beat lag, and replication channel group id.
Remote Site Name	This attribute displays the remote site name.
Heartbeat Status	This attribute displays the connectivity status with corresponding sites.
Heartbeat Lag	This attribute displays the lag or latency in seconds it took to synchronize between sites.
Replication channel Group Id	This attribute displays the ID of the replication channel group.

18. Click **Local Cluster Status** to view the local cluster status for the current site. The **Local Cluster Status** page is displayed.

Note

The following are read-only APIs.

Table 4-123 Local Cluster Status

Attribute	Description
Site Name	This attribute displays the name of the current site to which SEPP is connected.
Cluster Status	This attribute displays the local cluster status for the current site.

19. Click the **On Demand Backup** to create a new backup and view the status of initiated on-demand backups. The **On Demand Backup** page is displayed.

Note

The following are read-only APIs.

Table 4-124 On Demand Backup Details

Fields	Description
Site Name	This attribute displays the name of the current site to which SEPP is connected.
DR Status	This attribute displays the disaster recovery status .
Backup Id	This attribute displays the ID of the stored backup.
Backup Status	This attribute displays the status of backup.
Remote Transfer Status	The attribute displays the status of remote transfer.

Table 4-124 (Cont.) On Demand Backup Details

Fields	Description
Initiate Backup	The attribute displays whether the backup is initiated or not. Note: You can read and write this API.

20. Click **Edit**. The **Edit On Demand Backup** page appears.

Note

The **Edit** mode is available only for Initiate Backup.

21. Use the Toggle button to initiate the backup and click **Save**. A confirmation message "Saved successfully" appears.
22. Click **Cancel** to navigate back to the On Demand Backup page.
23. Click **Refresh** to reload the On Demand Backup page.

4.2.15.3 NRF-Client

This section contains the NF-management APIs. Perform the following steps to configure the priority values for the incoming and autonomous messages in the nrf-client microservice for the traffic prioritization feature.

1. In the CNC Console GUI, from the left navigation menu, navigate to **SEPP** and then click **Configurations**. The **Gateway**, **cnDBTier**, and **NRF-Client** options appear underneath.
2. Click **NRF-Client (NF-Management)** tab. The **NRF-Client** page is displayed.
3. From the list of tabs, select the **Traffic Prioritization** tab.
4. Click **Edit** to edit the Traffic Prioritization configuration. The **Edit Traffic Prioritization** page is displayed.
5. Set the **Feature Enabled** to true.
6. Set the names for the priority headers both for incoming requests and outgoing requests.
7. Edit the **NF Subscribe Message Types**. Default NF subscribe message types can be modified or we can add a new messageType by clicking **Add** option.
8. Edit **NFType** and priority of the NF Subscribe message types.
9. Click **Save** for applying the changes.
10. Edit the **NF Message Types**. Default NF message types can be modified or we can add a new messageType by clicking Add Option.
11. Edit **messageType** and **priority**.
12. Click **Save** for applying the changes.

Table 4-125 nrf-client Parameters

Parameter	Description	Details
trafficPrioritization.enabled	This is a mandatory parameter. This parameter is used to enable or disable the message priority feature.	Data Type: Boolean Range: true or false Default Value: false

Table 4-125 (Cont.) nrf-client Parameters

Parameter	Description	Details
trafficPrioritization.incomingPriorityHeader	This is a mandatory parameter. This is the header name for the incoming requests.	DataType: String Default Value: 3gpp-sbi-message-priority
trafficPrioritization.outgoingPriorityHeader	This is a mandatory parameter. This is the header name for the outgoing request.	DataType: String Default Value: 3gpp-sbi-message-priority
trafficPrioritization.messageTypes	This is a mandatory parameter. This is the message priority value for rest of the message request types that are not nfSubscribe Message types.	DataType: Object Default Value: NA
trafficPrioritization.nfSubscribeMessageTypes	This is a mandatory parameter. This is Message priority value for subscribe autonomous message request types, all autonomous operation associated to nrfClientSubscribeTypes.	DataType: Object Default Value: NA
trafficPrioritization.nfSubscribeMessageTypes.messageType	This is a mandatory parameter. This is MessageType value for the nfSubscribeMessageTypes.	DataType: String Default Value: NA
trafficPrioritization.messageTypes.messageType	This is a mandatory parameter. This is MessageType value of the messageType object.	DataType: String Default Value: NA
trafficPrioritization.nfSubscribeMessageTypes.subscribeNfTypes.nfType	This is a mandatory parameter. This is NfType value for the subscribeNfType object.	DataType: String Default Value: NA
trafficPrioritization.nfSubscribeMessageTypes.subscribeNfTypes.priority	This is a mandatory parameter. This is Priority value for the subscribeNfType object.	DataType: String Default Value: NA
trafficPrioritization.messageTypes.priority	This is a mandatory parameter. This is Priority value of the messageType object.	DataType: String Default Value: NA
trafficPrioritization.nfSubscribeMessageTypes.subscribeNfTypes	This is a mandatory parameter. This is subscribeNfTypes list of nfSubscribeMessageTypes.	DataType: Object Default Value: NA

5

SEPP Metrics, KPIs, and Alerts

5.1 SEPP Metrics

This section provides information about the SEPP metrics.

The following table describes the SEPP metric types used to measure the health and performance of SEPP and its core functionalities:

Table 5-1 Metric Type

Metric Type	Description
Counter	Represents the total number of occurrences of an event or traffic, such as measuring the total amount of traffic received and transmitted by SEPP, and so on.
Gauge	Represents a single numerical value that changes randomly. This metric type is used to measure various parameters, such as SEPP load values, memory usage, and so on.
Histogram	A histogram is a combination of various counters. The histogram metrics are used to track the size of events, usually how long they take, via their observe method.

The following table describes the SEPP dimensions:

Table 5-2 Dimensions

Dimensions	Details	Values
apiUrl	Resource URIs passing across Inter PLMN Via SEPP	Example: /namf-comm/v1/subscriptions
app	SEPP Service names	Examples: • cn32f-svc • cn32c-svc • pn32f-svc • pn32c-svc • Plmn Ingress Gateway • N32 Ingress Gateway
application	application name, here, it is ocsepp.	Example: ocsepp

Table 5-2 (Cont.) Dimensions

Dimensions	Details	Values
cause	Indicates the reason of failure	Examples: - Network ID is not present in PLMN ID List - UDR Response Is Not Success - UE is either not present or unable to extract UE from message - Previous Location Check Validation Failed
chart	Indicates the SEPP microservice release names	Examples: - pn32f-svc-2.23.3-0 - cn32f-svc-2.23.3-0
container	Indicates the name of the container. It is part of each metrics. The app and container contains the same value.	Examples: - cn32f-svc - cn32c-svc - pn32f-svc - pn32c-svc - Plmn Ingress Gateway - N32 Ingress Gateway
DestinationHost	Indicates the destination host for Jetty client on PN32F or CN32F	Examples: - ocsepp-release-plmn-egress-gateway - ocsepp-release-n32-egress-gateway
direction	Direction of the request or response. In Gateway Metrics, the values are egress and egressOut. In N32F Metrics the values are ingress and egress.	Examples: - Ingress - Egress
Egress Rate Limit List	The list that contains the PLMN for applying Egress Rate Limit.	Example: ERL1
engVersion	The SEPP Release version	Example: 23.3.0
error_action	The action needs to be taken when there is a validation failure in SEPP.	Examples: REJECT FORWARD
ErrorOriginator	Name of service that originates the error.	Example: PN32F
event	The event that occurred on request processing or completion.	Examples: - onBegin - onHeaders - onQueued - onContent - onCommit - onFailure
handshake_procedure	The type of the handshake operation at cSEPP or pSEPP.	Example: capability-exchange

Table 5-2 (Cont.) Dimensions

Dimensions	Details	Values
header	SBI Headers	Examples: - via - server
Host	FQDN of the target host	Example: ocsepp-release-n32-egress-gateway.
http_error_message	Reason for failure response received.	Examples: - Context Not Established - Destination URI contain invalid PLMN ID - Message validation failed - N32fContext Not Found - org.springframework.web.reactive.function.client.WebClientRequestException: Connect Timeout
http_method	HTTP Method Name	Examples: - GET - PUT - POST - PATCH - DELETE
http_status	HTTP Status Code in response (404 NOT_FOUND, 429 TOO_MANY_REQUESTS, 200 OK)	Examples: 2xx, 4xx, 5xx
namespace	Name of the Kubernetes namespace on which microservice is running.	Example: seppsvc
nf_instance_id	Unique identity of the NF Instance sending request to OCSEPP.	Example: 9faf1bbc-6e4a-4454-a507-aef01a101a06
NfServiceType	Name of target network function service	Example: nausf-auth
NfType	Name of target network function	Examples: - ausf - udm - nrf
node	Name of the Kubernetes worker node on which microservice is running.	Example: k8s-node-13.chase1.lab.in.oracle.com
peer_domain	Domain of Remote SEPP	Example: svc.cluster.local
peer_fqdn	FQDN of peer present in Remote SEPP	
peer_plmn_id	Supported PLMN list of Remote SEPP	Example: "[Plmn [mcc=123, mnc=456]]"
plmn_identifier	In CAT 2 Network ID Validation feature, PLMN is extracted from this identifier.	Examples: - supi - addUeLocation - guamiList
pod	Name of the pod of SEPP microservice	Example: ocsepp-release-cn32f-svc-6fd6ccfd4b-hkgqb

Table 5-2 (Cont.) Dimensions

Dimensions	Details	Values
Port	Port number	Example: 443
release	Name of the SEPP release deployed.	Example: ocsepp-release
releaseVersion	Indicates the current release version of SEPP.	Example: 23.4.0
remote_sepp_name	Name of the SEPP from where message is received or destined to	Example: SEPP-1
remote_sepp_set_name	Name of the Remote SEPP Set from where message is received or destined to	Example: RPS-3
request_path	Resource URI as per defined in 3GPP specifications for 5G.	Example: /nudm-sdm/v2/imsi-987654000000008
ruleApplied	Rules Applied on Local SEPP or Remote SEPP.	Examples: • REMOTE • LOCAL
Scheme	Indicates the HTTP Scheme	Examples: • HTTPS • HTTP
sepp_type	SEPP that acts as Producer SEPP or Consumer SEPP	Examples: • Consumer • Producer
sourceRss	only if Allowed P-RSS Validation is enabled	Example:
Status	The status of the feature or microservice.	Examples: • accepted • dropped • ratelimit not applied
vendor	For OCSEPP, vendor Value must be set to "oracle"	Example: oracle
PLMN_ID	This dimension indicates the PLMN ID of corresponding CSEPP/ PSEPP.	Example: PLMN_ID="111-200"
InstanceIdentifier	Prefix of the pod configured in helm when there are multiple instances in same deployment.	Example: InstanceIdentifier="n32_ingressgateway"
assertedPlmnValue	This dimension indicates the PLMN ID of corresponding CSEPP.	Example: assertedPlmnValue="123-345"
reason	This dimension indicates the reason for the metric being pegged.	Example: reason="Time Validation Is Successful"
Database	This dimension indicates the name of the associated database.	Example: database="seppdb_sepp_1"
method	This dimension indicates the method present in the incoming request.	Example: method="GET"

5.1.1 Configuring SEPP Metrics Dashboard in OCI

This section describes the steps to upload the `ocsepp_oci_dashboard_<version>.json` file on OCI Logging Analytics Dashboard. As OCI doesn't support Grafana, OCI uses the Logging Analytics Dashboard Service for visualizing the metrics and logs.

The steps are:

1. Log in to OCI Console.

Note

For more details about logging in to the OCI, refer to [Signing In to the OCI Console](#).

2. Open the navigation menu and click **Observability & Management**.
3. Under **Logging Analytics**, click **Dashboards**. The **Dashboards** page appears.
4. Choose the **Compartment** on the left pane.
5. Click **Import dashboards**.
6. User can select and upload the `ocsepp_oci_dashboard_<version>.json` file. The following three parameters of json file must be customized before uploading it:
 - a. `##COMPARTMENT_ID`: The OCID of the compartment.
 - b. `##METRIC_NAMESPACE`: The metrics namespace that the user provided while deploying OCI adaptor.
 - c. `##K8_NAMESPACE`: Kubernetes namespace where SEPP is deployed.
7. **Import dashboard** page appears. Click **Import** button on the page.
User can view the imported dashboard and can view the metrics in the dashboard.

Note

SEPP has organized the panels or widgets in five dashboards to support the SEPP metrics and all the five dashboards have been clubbed into a single JSON file.

For more details, see *Oracle Communications Cloud Native Core, OCI Adaptor Deployment Guide*.

5.1.2 Common Metrics

5.1.2.1 cgroup_cpu_nanoseconds

Table 5-3 cgroup_cpu_nanoseconds

Field	Details
Metric Details	Total CPU time consumed by service in nanoseconds.
Microservice	Consumer N32f, Producer N32f

Table 5-3 (Cont.) cgroup_cpu_nanoseconds

Field	Details
Type	Hologram
Dimensions	• app (Consumer, Producer) • chart • container • namespace • pod • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision

5.1.2.2 cgroup_memory_bytes

Table 5-4 cgroup_memory_bytes

Field	Details
Metric Details	Total memory consumed by service in bytes.
Microservice	Consumer N32f, Producer N32f
Type	Gauge
Dimensions	• app (Consumer, Producer) • chart • container • namespace • pod • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision

5.1.2.3 nrfclient_nf_status_with_nrf

Table 5-5 nrfclient_nf_status_with_nrf

Field	Details
Metric Details	The operative status of the NF communicated to the NRF. The gauge indicates status as follows: • 0 - REGISTERED • 1 - DEREGISTERED • 2 - SUSPENDED • 3 - UNDISCOVERABLE • 4 - UNKNOWN
Microservice	nrf client
Metric type	Gauge
Dimensions	• NfInstanceId • NfType • NfFqdn

Note

This metrics is not applicable for the Roaming Hub mode.

5.1.2.4 oc_ingressgateway_ip_addresses_fetch_failure

Table 5-6 oc_ingressgateway_ip_addresses_fetch_failure

Field	Details
Metric Details	This metrics will be pegged when exception occurs while fetching IP address of the services from kube-api-server.
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Counter
Dimensions	NA

5.1.2.5 oc_egressgateway_ip_addresses_fetch_failure

Table 5-7 oc_egressgateway_ip_addresses_fetch_failure

Field	Details
Metric Details	This metrics will be pegged when exception occurs while fetching IP address of the services from kube-api-server.
Microservice	Plmn Egress Gateway, N32 Egress Gateway
Type	Counter
Dimensions	NA

5.1.2.6 oc_certificatemanagement_tls_certificate_info

Table 5-8 oc_certificatemanagement_tls_certificate_info

Field	Details
Metric Details	This metric pegs the status as 1 if any file in the configured secret fails to load its name, and Reason is pegged in dimension. When the secret is loaded again, the status of the current entry is changed to 0. If the certificate is loaded without any issue, there will be no entry in Prometheus and the previous entry value will be changed to 0.
Microservice	Plmn Egress Gateway, N32 Egress Gateway
Type	Gauge
Dimensions	- Name - SecretName - Reason (CERT_CORRUPT, CERT_MISSING, CERT_EXPIRED) - Service

5.1.2.7 oc_certificatemanagement_tls_secret_status

Table 5-9 oc_certificatemanagement_tls_secret_status

Field	Details
Metric Details	This metric is used to determine whether the configured secret is available in the namespace. If the secret is available in the namespace, the metric is pegged with value 0. If the secret is unavailable or if it is removed, then the metric is pegged with status 1.
Microservice	Plmn Egress Gateway, N32 Egress Gateway
Type	Gauge
Dimensions	• Secret • Namespace • Service

5.1.3 CN32F Common Metrics

5.1.3.1 ocsepp_cn32f_jetty_request_stat_metrics_total

Table 5-10 ocsepp_cn32f_jetty_request_stat_metrics_total

Field	Details
Metric Details	This metric will be pegged for every event occurred when a request is sent to CN32F
Microservice	Consumer N32f
Type	Counter
Dimensions	• app (Consumer) • chart • client_type • container • DestinationHost • event • nf_instance_id • namespace • pod

5.1.3.2 ocsepp_cn32f_jetty_response_stat_metrics_total

Table 5-11 ocsepp_cn32f_jetty_response_stat_metrics_total

Field	Details
Metric Details	This metric will be pegged for every event occurred when a response is received from CN32F
Microservice	Consumer N32f
Type	Counter

Table 5-11 (Cont.) ocsepp_cn32f_jetty_response_stat_metrics_total

Field	Details
Dimensions	• app (Consumer) • chart • client_type • container • DestinationHost • event • nf_instance_id • namespace • pod

5.1.3.3 ocsepp_cn32f_connection_failure_total

Table 5-12 ocsepp_cn32f_connection_failure_total

Field	Details
Metric Details	This metric will be pegged in the customized Jetty Client as soon as it fails to connect to the destination service.
Microservice	Consumer N32f
Type	Counter
Dimensions	• app (Consumer) • chart • container • Direction • ErrorOriginator • http_error_message • Host • nf_instance_id • namespace • pod • Port

5.1.3.4 ocsepp_cn32f_requests_failure_total

Table 5-13 ocsepp_cn32f_requests_failure_total

Field	Details
Metric Details	Total number of requests failed to be sent from cn32f to Remote SEPP. Condition: When any error or exception occurs on cn32f side because of which request is not sent to pn32f.
Microservice	Consumer N32f
Type	Counter

Table 5-13 (Cont.) ocsepp_cn32f_requests_failure_total

Field	Details
Dimensions	• app(consumer) • chart • container • namespace • nf_instance_id • pod • http_status • vendor • peer_domain • peer_plmn_id • peer_fqdn • remote_sepp_name

Note

- The dimension peer_domain is optional and will be present only when destination URI contains PLMN ID.
- The dimension peer_plmn_id is optional and will be present only when destination URI contains PLMN ID.
- The dimension peer_fqdn is optional and will be present only when n32f context is found.
- The dimension remote_sepp_name is optional and will be present only when n32f context is found.

5.1.3.5 ocsepp_cn32f_response_failure_total

Table 5-14 ocsepp_cn32f_response_failure_total

Field	Details
Metric Details	Total number of response failed to be sent from cn32f pod to NF. Condition: When any error or exception occurs on cn32f and request is not sent to NF.
Microservice	Consumer N32f
Type	Counter

Table 5-14 (Cont.) ocsepp_cn32f_response_failure_total

Field	Details
Dimensions	• app(consumer) • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • vendor

5.1.3.6 ocsepp_cn32f_requests_total

Table 5-15 ocsepp_cn32f_requests_total

Field	Details
Metric Details	Total number of requests sent or received through the cn32f interface. Count is incremented when a request is received on the InboundInterface or sent from the OutboundInterface of cn32f.
Microservice	Consumer N32f
Type	Counter
Dimensions	• app (consumer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • sourceRss (only if Allowed P-RSS Validation is enabled) • source_plmn_id • vendor

5.1.3.7 ocsepp_cn32f_response_total

Table 5-16 ocsepp_cn32f_response_total

Field	Details
Metric Details	Total number of responses sent or received through the cn32f interface. Count is incremented when a response is received on the OutboundInterface or sent from the InboundInterface of cn32f.
Microservice	Consumer N32f
Type	Counter
Dimensions	• app(consumer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • sourceRss (only if Allowed P-RSS Validation is enabled) • http_status • vendor

5.1.3.8 ocsepp_cn32f_latency_seconds_count

Table 5-17 ocsepp_cn32f_latency_seconds_count

Field	Details
Metric Details	This metric is used to display the number of ingress requests processed at cn32f in a particular time span (in seconds).
Microservice	Consumer N32f
Type	Histogram

Table 5-17 (Cont.) ocsepp_cn32f_latency_seconds_count

Field	Details
Dimensions	• app(consumer) • chart • container • direction • message_type • namespace • nf_instance_id • nf_service_type • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • vendor

5.1.3.9 ocsepp_cn32f_latency_seconds_max

Table 5-18 ocsepp_cn32f_latency_seconds_max

Field	Details
Metric Details	This metrics is used to display the maximum of processing time of an ingress request at cn32f in seconds.
Microservice	Consumer N32f
Type	Histogram
Dimensions	• app(consumer) • chart • container • direction • message_type • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • nf_service_type • vendor

5.1.3.10 ocsepp_cn32f_latency_seconds_sum

Table 5-19 ocsepp_cn32f_latency_seconds_sum

Field	Details
Metric Details	This metrics is used to display the average of processing time of all the ingress request at cn32f for a particular time.
Microservice	Consumer N32f
Type	Histogram
Dimensions	• app(consumer) • chart • container • direction • message_type • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • nf_service_type • vendor

5.1.3.11 ocsepp_cn32f_outgoing_connections

Table 5-20 ocsepp_cn32f_outgoing_connections

Field	Details
Metric Details	Gauge metric that will peg active outgoing connections from CN32F to destination
Microservice	Consumer N32f
Type	Gauge
Dimensions	• app (Consumer) • chart • container • direction • Host • nf_instance_id • namespace • pod

5.1.3.12 ocsepp_cn32f_server_latency

Table 5-21 ocsepp_cn32f_server_latency

Field	Details
Metric Details	This metric will be pegged in Jetty response listener that captures the amount of time taken for processing of the request by jetty client
Microservice	Consumer N32f
Type	Gauge
Dimensions	• Method • InstanceIdentifier • Host

5.1.3.13 ocsepp_cn32f_multipart_response_extraction_failure_total

Table 5-22 ocsepp_cn32f_multipart_response_extraction_failure_total

Field	Details
Metric Details	This metric is used to determine the number of multipart responses for which request body extraction failed.
Microservice	cn32f
Type	counter
Dimensions	• app (consumer, producer) • chart • container • http_status • method • namespace • nf_instance_id • pod • release • request_uri • resource_uri • vendor

5.1.3.14 csepp_cn32f_multipart_requests_extraction_failure_total

Table 5-23 csepp_cn32f_multipart_requests_extraction_failure_total

Field	Details
Metric Details	This metric is used to determine the number of multipart requests for which request body extraction failed.
Microservice	cn32f
Type	counter

Table 5-23 (Cont.) csepp_cn32f_multipart_requests_extraction_failure_total

Field	Details
Dimensions	• app (consumer, producer) • chart • container • http_status • method • namespace • nf_instance_id • pod • release • request_uri • vendor

5.1.4 PN32F Common Metrics

5.1.4.1 ocsepp_pn32f_requests_total

Table 5-24 ocsepp_pn32f_requests_total

Field	Details
Metric Details	Total number of requests sent to or received from the peer SEPP through the pn32f interface. The counter is incremented when a request is received on the InboundInterface or sent from the OutboundInterface of pn32f.
Microservice	Producer N32f
Type	Counter
Dimensions	• app(producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor

Note

The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.4.2 ocsepp_pn32f_requests_failure_total

Table 5-25 ocsepp_pn32f_requests_failure_total

Field	Details
Metric Details	Total number of requests failed to be sent to NF. This metrics can be used in Hosted SEPP mode. Condition: When any error or exception occurs on pn32f side because of which request is not sent to NF.
Microservice	Producer N32f
Type	Counter
Dimensions	• app(producer) • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • vendor • http_error_message

Note

The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.4.3 ocsepp_pn32f_responses_total

Table 5-26 ocsepp_pn32f_responses_total

Field	Details
Metric Details	Total number of responses sent to or received from peer SEPP through the pn32f interface. This counter is incremented when a response is received on the InboundInterface or sent via the OutboundInterface of pn32f.
Microservice	Producer N32f
Type	Counter

Table 5-26 (Cont.) ocsepp_pn32f_responses_total

Field	Details
Dimensions	• app(producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status (201 CREATED, 404 NOT_FOUND) • vendor

Note

The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.4.4 ocsepp_pn32f_responses_failure_total

Table 5-27 ocsepp_pn32f_responses_failure_total

Field	Details
Metric Details	Number of responses failed to be sent to Consumer SEPP (cSEPP). Condition: When a response is failed to be sent to cSEPP.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (producer) • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • vendor

Note

The dimension `peer_plmn_id` is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.4.5 ocsepp_pn32f_latency_seconds_count

Table 5-28 ocsepp_pn32f_latency_seconds_count

Field	Details
Metric Details	This metric is used to display the number of ingress requests processed at pn32f in a particular time span (in seconds).
Microservice	Producer N32f
Type	Histogram
Dimensions	• app (producer) • chart • container • direction • message_type • namespace • nf_instance_id • nf_service_type • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • vendor

Note

The dimension `peer_plmn_id` is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.4.6 ocsepp_pn32f_latency_seconds_sum

Table 5-29 ocsepp_pn32f_latency_seconds_sum

Field	Details
Metric Details	This metrics is used to display the average processing time of all the ingress request at pn32f for a particular time.
Microservice	Producer N32f
Type	Histogram

Table 5-29 (Cont.) ocsepp_pn32f_latency_seconds_sum

Field	Details
Dimensions	• app (producer) • chart • container • direction • message_type • namespace • nf_instance_id • nf_service_type • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type

Note

The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.4.7 ocsepp_pn32f_latency_seconds_max

Table 5-30 ocsepp_pn32f_latency_seconds_max

Field	Details
Metric Details	This metrics is used to display the maximum processing time of all the ingress request at pn32f for a particular time.
Microservice	Producer N32f
Type	Histogram
Dimensions	• vendor • nf_instance_id • nf_type • peer_fqdn • peer_domain • peer_plmn_id • direction • message_type • namespace • nf_type • nf_service_type • remote_sepp_name • app • container • pod • release

Note

The dimension `peer_plmn_id` is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.4.8 ocsepp_pn32f_connection_failure_total

Table 5-31 ocsepp_pn32f_connection_failure_total

Field	Details
Metric Details	This metric will be pegged in the customized Jetty Client as soon as it fails to connect to the destination service.
Microservice	Producer N32f
Type	Counter
Dimensions	- Host - port - Direction - http_error_message - errorOriginator - nf_instance_id

5.1.4.9 ocsepp_pn32f_jetty_request_stat_metrics_total

Table 5-32 ocsepp_pn32f_jetty_request_stat_metrics_total

Field	Details
Metric Details	This metric will be pegged for every event occurred when a request is sent to PN32F
Microservice	Producer N32f
Type	Counter
Dimensions	- app (Producer) - chart - client_type - container - DestinationHost - event - nf_instance_id - namespace - pod

5.1.4.10 ocsepp_pn32f_jetty_response_stat_metrics_total

Table 5-33 ocsepp_pn32f_jetty_response_stat_metrics_total

Field	Details
Metric Details	This metric will be pegged for every event occurred when a response is received from PN32F
Microservice	Producer N32f

Table 5-33 (Cont.) ocsepp_pn32f_jetty_response_stat_metrics_total

Field	Details
Type	Counter
Dimensions	• app (Producer) • chart • client_type • container • DestinationHost • event • nf_instance_id • namespace • pod

5.1.4.11 ocsepp_pn32f_outgoing_connections

Table 5-34 ocsepp_pn32f_outgoing_connections

Field	Details
Metric Details	Gauge metric that will peg active outgoing connections from PN32F to destination
Microservice	Producer N32f
Type	Gauge
Dimensions	• app (Producer) • chart • container • direction • Host • nf_instance_id • namespace • pod

5.1.4.12 ocsepp_pn32f_server_latency

Table 5-35 ocsepp_pn32f_server_latency

Field	Details
Metric Details	This metric will be pegged in Jetty response listener that captures the amount of time taken for processing of the request by jetty client
Microservice	Producer N32f
Type	Gauge
Dimensions	• Method • InstanceIdentifier • Host

5.1.5 N32C Handshake Procedure Metrics

5.1.5.1 ocsepp_n32c_handshake_failure_attempts_total

Table 5-36 ocsepp_n32c_handshake_failure_attempts_total

Field	Details
Metric Details	If N32c Handshake procedure fails, this metrics will be pegged and corresponding alarm will be raised. This metrics can be used in Hosted SEPP mode.
Microservice	Producer and Consumer N32c
Type	Counter
Dimensions	• app(consumer, producer) • chart • container • handshake_procedure • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor

5.1.5.2 ocsepp_n32c_handshake_reInitiation_failure

Table 5-37 ocsepp_n32c_handshake_reInitiation_failure

Field	Details
Metric Details	If N32c Handshake Reinitiation procedure fails, this metrics will be pegged and corresponding alarm will be raised.
Microservice	Consumer N32c, Producer N32c
Type	Gauge
Dimensions	• app(consumer, producer) • chart • container • handshake_procedure • namespace • peer_domain (optional) • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor

5.1.5.3 ocsepp_cn32c_handshake_requests_total

Table 5-38 ocsepp_cn32c_handshake_requests_total

Field	Details
Metric Details	Total number of requests sent over n32c for handshake procedure. Condition: When SEPP initiates any handshake procedure requests towards peer SEPP.
Microservice	Consumer N32c
Type	Counter
Dimensions	• app(consumer) • chart • container • handshake_procedure • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor

5.1.5.4 ocsepp_cn32c_handshake_response_total

Table 5-39 ocsepp_cn32c_handshake_response_total

Field	Details
Metric Details	Total number of responses received over n32c for handshake procedure. Condition: When SEPP receives any handshake procedure response from peer SEPP. It can be successful or failure based on response code.
Microservice	Consumer N32c
Type	Counter
Dimensions	• app(consumer) • chart • container • handshake_procedure • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status (2xx,4xx,5xx) • vendor

5.1.5.5 ocsepp_cn32c_handshake_initiation_req_total

Table 5-40 ocsepp_cn32c_handshake_initiation_req_total

Field	Details
Metric Details	Total number of Handshake initiation requests received from config-mgr. Condition: When handshake initiation requests are received from config-mgr.
Microservice	Consumer N32c
Type	Counter
Dimensions	• app(consumer) • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor

5.1.5.6 ocsepp_cn32c_handshake_reinitiation_req_total

Table 5-41 ocsepp_cn32c_handshake_reinitiation_req_total

Field	Details
Metric Details	Total number of Handshake ReInitiation requests received from config-mgr. Condition: When handshake Reinitiation requests received from config-mgr.
Microservice	Consumer N32C
Type	Counter
Dimensions	• app(consumer) • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor

5.1.5.7 ocsepp_cn32c_handshake_delete_req_total

Table 5-42 ocsepp_cn32c_handshake_delete_req_total

Field	Details
Metric Details	Total number of Handshake context delete requests received from config-mgr. Condition: When handshake context delete requests are received from config-mgr.
Microservice	Consumer N32c
Type	Counter
Dimensions	• app(consumer) • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor

5.1.5.8 ocsepp_pn32c_handshake_requests_total

Table 5-43 ocsepp_pn32c_handshake_requests_total

Field	Details
Metric Details	Total number of requests received over n32c for handshake procedure. Condition: When any handshake procedure request is received from peer SEPP.
Microservice	Producer N32c
Type	Counter
Dimensions	• app(producer) • chart • container • handshake_procedure • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor

5.1.5.9 ocsepp_pn32c_handshake_response_total

Table 5-44 ocsepp_n32c_handshake_response_total

Field	Details
Metric Details	Total number of responses sent over n32c for handshake procedure. Condition: When SEPP sends response to handshake procedure received. It can be a success response or failure response based on success code.
Microservice	Producer N32c
Type	Counter
Dimensions	• app(producer) • chart • container • handshake_procedure • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status (200 OK,4xx,5xx) • vendor

5.1.5.10 ocsepp_n32c_handshake_status

Table 5-45 ocsepp_n32c_handshake_status

Field	Details
Metric Details	This metric describes the current status of the Remote partner handshake procedure.
Microservice	N32c
Type	Gauge
Dimensions	• Remote_partner_name • Remote_fqdn • Namespace • Nf_instance_id

Note

Following are the possible values:

- 0: CONTEXT_NOT_FOUND
- 1: CAPABILITY_EXCHANGE_STATE
- 2: PARAMETER_EXCHANGE_STATE
- 3: N32F_ESTABLISHED_STATE

If no entry is found for a specific Remote SEPP, the N32c context is never established.

5.1.6 Cat-1 NRF Service API Query Parameters Validation metrics

5.1.6.1 ocsepp_security_service_api_query_param_validation_success_total

Table 5-46 ocsepp_security_service_api_query_param_validation_success_total

Field	Details
Metric Details	Total number of successful requests when the Cat-1 NRF Service API Query Parameters Validation feature is enabled.
Microservice	N32f
Dimensions	• app • chart • container • direction • http_method • namespace • nf_instance_id • peer_domain • peer_fqdn • pod • release • remote_sepp_name • remote_sepp_set_name • requester_nf_type • resource_uri • target_nf_type • vendor
Type	Counter

5.1.6.2 ocsepp_security_service_api_query_param_validation_failure_total

Table 5-47 ocsepp_security_service_api_query_param_validation_failure_total

Field	Details
Metric Details	Total number of failure requests when the feature is enabled.
Microservice	N32f

Table 5-47 (Cont.) ocsepp_security_service_api_query_param_validation_failure_total

Field	Details
Dimensions	• app • chart • container • direction • http_method • namespace • nf_instance_id • peer_domain • peer_fqdn • pod • release • remote_sepp_name • remote_sepp_set_name • resource_uri • vendor
Type	Counter

5.1.6.3 ocsepp_configmgr_query_param_api_validation_config_failure_total

Table 5-48 ocsepp_configmgr_query_param_api_validation_config_failure_total

Field	Details
Metric Details	This metrics is pegged whenever the configurations related to query parameter api validation fails.
Microservice	Config Manager
Type	Counter
Dimensions	• http_error_message • http_status • http_method • chart • namespace • pod • container

5.1.7 5G SBI Message Mediation Support Metrics

5.1.7.1 ocsepp_n32f_mediation_requests_total

Table 5-49 ocsepp_n32f_mediation_requests_total

Field	Details
Metric Details	Metric is common for both CN32F and PN32F. Separation happens based on "app" tag. Number of requests in which Trigger Rule Applied at SEPP end for Mediation, based on configuration.
Microservice	Consumer N32f, Producer N32f
Type	Counter
Dimensions	• app • chart • container • namespace • pod • ruleApplied • peer_fqdn • peer_domain • peer_plmn_id • direction • http_method • NfType • NfServiceType • vendor • nf_instance_id • message_type

Note

The dimension "peer_plmn_id" is applicable only for Consumer N32F.

5.1.7.2 ocsepp_n32f_mediation_not_applied_total

Table 5-50 ocsepp_n32f_mediation_not_applied_total

Field	Details
Metric Details	Metric is common for both CN32F and PN32F. Separation happens based on "app" tag. Number of requests for which Trigger Rule do not match at SEPP and request is not forwarded to Mediation. For Match all configurations, the trigger points will be matched.
Microservice	Consumer N32f, Producer N32f
Type	Counter

Table 5-50 (Cont.) ocsepp_n32f_mediation_not_applied_total

Field	Details
Dimensions	• app (Consumer, Producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • message_type • ruleApplied • vendor

Note

The dimension "peer_plmn_id" is applicable only for Consumer N32F.

5.1.7.3 ocsepp_cn32f_mediation_response_total

Table 5-51 ocsepp_cn32f_mediation_response_total

Field	Details
Metric Details	Number of requests in which CN32F service of SEPP get Response from Mediation Service.
Microservice	Consumer N32f
Type	Counter
Dimensions	• app (Producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • http_status • remote_sepp_name • vendor

5.1.7.4 ocsepp_cn32f_mediation_response_failure_total

Table 5-52 ocsepp_cn32f_mediation_response_failure_total

Field	Details
Metric Details	Number of requests in which CN32F service of SEPP get failure response from mediation service.
Microservice	Consumer N32f
Type	Counter
Dimensions	• peer_fqdn • peer_domain • plmn_id • statusCode • direction • error_msg • vendor • nfInstanceId • ruleApplied • requestType • method • request_path

5.1.7.5 ocsepp_pn32f_mediation_response_total

Table 5-53 ocsepp_pn32f_mediation_response_total

Field	Details
Metric Details	Number of requests in which PN32F service of SEPP get response from Mediation Service.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (Producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • message_type • ruleApplied • http_status • vendor

Note

The dimension `peer_plmn_id` is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.7.6 ocsepp_pn32f_mediation_response_failure_total

Table 5-54 ocsepp_pn32f_mediation_response_failure_total

Field	Details
Metric Details	Number of requests in which PN32F service of SEPP get failure response from mediation service.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (Producer) • chart • container • direction • http_error_message • http_method • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • request_path • message_type • ruleApplied • http_status • vendor

Note

The dimension "peer_plmn_id" is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.7.7 ocsepp_cn32f_mediation_latency_seconds_count

Table 5-55 ocsepp_cn32f_mediation_latency_seconds_count

Field	Details
Metric Details	This metric is used to display the number of ingress requests processed at cn32f in a particular time span (in seconds).
Microservice	Consumer N32f
Type	Histogram

Table 5-55 (Cont.) ocsepp_cn32f_mediation_latency_seconds_count

Field	Details
Dimensions	• app (Consumer) • chart • container • direction • message_type • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • vendor

5.1.7.8 ocsepp_cn32f_mediation_latency_seconds_max

Table 5-56 ocsepp_cn32f_mediation_latency_seconds_max

Field	Details
Metric Details	Total time taken for processing a message (from sending a message to receiving the response).
Microservice	Consumer N32f
Type	Histogram
Dimensions	• app (Consumer) • chart • container • direction • message_type • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • vendor

5.1.7.9 ocsepp_cn32f_mediation_latency_seconds_sum

Table 5-57 ocsepp_cn32f_mediation_latency_seconds_sum

Field	Details
Metric Details	Total time taken for processing a message (from sending a message to receiving the response).

Table 5-57 (Cont.) ocsepp_cn32f_mediation_latency_seconds_sum

Field	Details
Microservice	Consumer N32f
Type	Histogram
Dimensions	• app (Consumer) • chart • container • direction • message_type • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • vendor

5.1.7.10 ocsepp_pn32f_mediation_latency_seconds_count

Table 5-58 ocsepp_pn32f_mediation_latency_seconds_count

Field	Details
Metric Details	Total time taken for processing a message (from sending a message to receiving the response).
Microservice	Producer N32f
Type	Histogram
Dimensions	• app (Producer) • chart • container • direction • message_type • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • vendor

Note

The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.7.11 ocsepp_pn32f_mediation_latency_seconds_max

Table 5-59 ocsepp_pn32f_mediation_latency_seconds_max

Field	Details
Metric Details	This metrics is used to display the maximum processing time of all the ingress request at pn32f for a particular time.
Microservice	Producer N32f
Type	Histogram
Dimensions	• app (Producer) • chart • container • direction • message_type • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • vendor

Note

The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.7.12 ocsepp_pn32f_mediation_latency_seconds_sum

Table 5-60 ocsepp_pn32f_mediation_latency_seconds_sum

Field	Details
Metric Details	Time taken by Mediation Service to process request after getting called from PN32F service.
Microservice	PN32f
Type	Histogram

Table 5-60 (Cont.) ocsepp_pn32f_mediation_latency_seconds_sum

Field	Details
Dimensions	• app (Producer) • chart • container • direction • message_type • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • vendor

Note

The dimension "peer_plmn_id" is optional and will be available only when SAN contains the PLMN in FQDN.

5.1.7.13 ocsepp_configmgr_mediation_rule_validation_failure_total

Table 5-61 ocsepp_configmgr_mediation_rule_validation_failure_total

Field	Details
Metric Details	This metrics is pegged whenever the rule validation fails related to mediation feature.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • chart • pod • namespace • microservice • node • container • vendor

5.1.7.14 ocsepp_configmgr_mediation_feature_validation_failure_total

Table 5-62 ocsepp_configmgr_mediation_feature_validation_failure_total

Field	Details
Metric Details	This metrics is pegged whenever there is a failure related to mediation feature.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • chart • pod • namespace • microservice • node • container • vendor

5.1.7.15 ocsepp_configmgr_mediation_rule_config_failure_total

Table 5-63 ocsepp_configmgr_mediation_rule_config_failure_total

Field	Details
Metric Details	This metrics is pegged whenever there is a configuration failure for mediation feature.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • http_method • chart • pod • namespace • microservice • node • container • vendor

5.1.8 Hosted SEPP Metrics

5.1.8.1 ocsepp_cn32c_handshake_requests_hosted_sepp_total

Table 5-64 ocsepp_cn32c_handshake_requests_hosted_sepp_total

Field	Details
Metric Details	Total number of handshake requests sent from Hosted SEPP.
Microservice	Consumer N32C
Type	Counter
Dimensions	- remote_sepp_name - peer_plmn_id - self_fqdn - x_source_fqdn - handshake_procedure - peer_fqdn

5.1.8.2 ocsepp_cn32f_sni_xfs_requests_hosted_sepp_total

Table 5-65 ocsepp_cn32f_sni_xfs_requests_hosted_sepp_total

Field	Details
Metric Details	Total number of requests for which SNI and XFS populated correctly.
Microservice	Consumer N32F
Type	Counter
Dimensions	- remote_sepp_name - peer_fqdn - http_status - x_source_fqdn - sourceRss - source_plmn_id - server_name_indication

5.1.8.3 ocsepp_cn32f_sni_xfs_requests_mismatch_hosted_sepp_total

Table 5-66 ocsepp_cn32f_sni_xfs_requests_mismatch_hosted_sepp_total

Field	Details
Metric Details	Total number of requests for which SNI and XFS populated incorrectly.
Microservice	Consumer N32F
Type	Counter

Table 5-66 (Cont.) ocsepp_cn32f_sni_xfs_requests_mismatch_hosted_sepp_total

Field	Details
Dimensions	- remote_sepp_name - peer_fqdn - http_status - x_source_fqdn - sourceRss - source_plmn_id - server_name_indication

5.1.8.4 ocsepp_pn32f_requests_hosted_sepp_total

Table 5-67 ocsepp_pn32f_requests_hosted_sepp_total

Field	Details
Metric Details	Total number of requests for which pn32f is able to forward request in Hosted SEPP.
Microservice	Consumer N32F
Type	Counter
Dimensions	- http_status - http_method - server_name_indication - subject_alt_name

5.1.9 Message Copy Metrics

5.1.9.1 oc_ingressgateway_msgcopy_requests_total

Table 5-68 oc_ingressgateway_msgcopy_requests_total

Field	Details
Metric Details	This is incremented whenever request message is sent or acknowledged from Data Director.
Microservice	PLMN Ingress gateway, N32 Ingress Gateway
Type	Counter
Dimensions	- app (PLMN Ingress gateway, N32 Ingress Gateway) - chart - container - namespace - pod - type

5.1.9.2 oc_ingressgateway_msgcopy_responses_total

Table 5-69 oc_ingressgateway_msgcopy_responses_total

Field	Details
Metric Details	This is incremented whenever response message is sent or acknowledged from DD.
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Counter
Dimensions	• app (PLMN Ingress gateway, N32 Ingress Gateway) • chart • container • namespace • pod • type

5.1.9.3 oc_ingressgateway_dd_unreachable

Table 5-70 oc_ingressgateway_dd_unreachable

Field	Details
Metric Details	This indicates whether DD is reachable or not 0 - reachable, 1 - unreachable
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Gauge
Dimensions	• app

5.1.9.4 oc_egressgateway_msgcopy_requests_total

Table 5-71 oc_egressgateway_msgcopy_requests_total

Field	Details
Metric Details	This is incremented whenever request message is sent or acknowledged from DD.
Microservice	Plmn Egress Gateway, N32 Egress Gateway
Type	Counter
Dimensions	• app (PLMN Ingress gateway, N32 Ingress Gateway) • chart • container • namespace • pod • type

5.1.9.5 oc_egressgateway_msgcopy_responses_total

Table 5-72 oc_egressgateway_msgcopy_responses_total

Field	Details
Metric Details	This is incremented whenever response message is sent or acknowledged from DD.
Microservice	Plmn Egress Gateway, N32 Egress Gateway
Type	Counter
Dimensions	• app (PLMN Ingress gateway, N32 Ingress Gateway) • chart • container • namespace • pod • type

5.1.9.6 oc_egressgateway_dd_unreachable

Table 5-73 oc_egressgateway_dd_unreachable

Field	Details
Metric Details	This indicates whether DD is reachable or not 0 - reachable, 1 - unreachable
Microservice	Plmn Egress Gateway, N32 Egress Gateway
Type	Gauge
Dimensions	• app

5.1.10 SOR Metrics

5.1.10.1 ocsepp_pn32f_sor_requests_total

Table 5-74 ocsepp_pn32f_sor_requests_total

Field	Details
Metric Details	Number of requests sent to SOR
Microservice	Producer N32f
Type	Counter

Table 5-74 (Cont.) ocsepp_pn32f_sor_requests_total

Field	Details
Dimensions	• app (Producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • resource_uri • sorFqdn • sorPort • vendor

Note

- The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.
- If Alternate Routing for SOR (Steering of Roaming) servers is enabled, the fqdn and port tags will contain multiple values.
Example:

fqdn = A,B,C

port = A,B,C

In this context:

A refers to the Primary SOR Server/Port

B refers to the Secondary SOR Server/Port

C refers to the Tertiary SOR Server/Port

Each position in the list corresponds to the same level of priority across both the FQDN and Port fields.

5.1.10.2 ocsepp_pn32f_sor_responses_total

Table 5-75 ocsepp_pn32f_sor_responses_total

Field	Details
Metric Details	Number of responses received from SOR.
Microservice	Producer N32f

Table 5-75 (Cont.) ocsepp_pn32f_sor_responses_total

Field	Details
Type	Counter
Dimensions	• app (Producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • resource_uri • sorFqdn • sorPort • vendor

Note

- The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.
- If Alternate Routing for SOR (Steering of Roaming) servers is enabled, the fqdn and port tags will contain multiple values.

Example:

fqdn = A,B,C

port = A,B,C

In this context:

A refers to the Primary SOR Server/Port

B refers to the Secondary SOR Server/Port

C refers to the Tertiary SOR Server/Port

Each position in the list corresponds to the same level of priority across both the FQDN and Port fields.

5.1.10.3 ocsepp_pn32f_sor_retry_to_producer_requests_total

Table 5-76 ocsepp_pn32f_sor_retry_to_producer_requests_total

Field	Details
Metric Details	Number of requests sent to producer based on 3gpp header.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (Producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • sorFqdn • sorPort • vendor

Note

- The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.
- If Alternate Routing for SOR (Steering of Roaming) servers is enabled, the fqdn and port tags will contain multiple values.
Example:

fqdn = A,B,C

port = A,B,C

In this context:

A refers to the Primary SOR Server/Port

B refers to the Secondary SOR Server/Port

C refers to the Tertiary SOR Server/Port

Each position in the list corresponds to the same level of priority across both the FQDN and Port fields.

5.1.10.4 ocsepp_pn32f_sor_back_to_consumer_responses_total

Table 5-77 ocsepp_pn32f_sor_back_to_consumer_responses_total

Field	Details
Metric Details	Number of responses received from SoR sent back to consumer.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (Producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • sorFqdn • sorPort • vendor

Note

- The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.
- If Alternate Routing for SOR (Steering of Roaming) servers is enabled, the fqdn and port tags will contain multiple values.
Example:

fqdn = A,B,C

port = A,B,C

In this context:

A refers to the Primary SOR Server/Port

B refers to the Secondary SOR Server/Port

C refers to the Tertiary SOR Server/Port

Each position in the list corresponds to the same level of priority across both the FQDN and Port fields.

5.1.10.5 ocsepp_pn32f_sor_failure_total

Table 5-78 ocsepp_pn32f_sor_failure_total

Field	Details
Metric Details	Number of 4xx or 5xx responses received from SOR.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (Producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • sorFqdn • sorPort • vendor

Note

- The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.
- If Alternate Routing for SOR (Steering of Roaming) servers is enabled, the fqdn and port tags will contain multiple values.
Example:

fqdn = A,B,C

port = A,B,C

In this context:

A refers to the Primary SOR Server/Port

B refers to the Secondary SOR Server/Port

C refers to the Tertiary SOR Server/Port

Each position in the list corresponds to the same level of priority across both the FQDN and Port fields.

5.1.10.6 ocsepp_pn32f_sor_timeout_failure_total

Table 5-79 ocsepp_pn32f_sor_timeout_failure_total

Field	Details
Metric Details	Number of requests which are request timeout while connecting to SOR.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (Producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • NfType • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • sorFqdn • sorPort • vendor

Note

- The dimension peer_plmn_id is optional and will be available only when SAN contains the PLMN in FQDN.
- If Alternate Routing for SOR (Steering of Roaming) servers is enabled, the fqdn and port tags will contain multiple values.
Example:

fqdn = A,B,C

port = A,B,C

In this context:

A refers to the Primary SOR Server/Port

B refers to the Secondary SOR Server/Port

C refers to the Tertiary SOR Server/Port

Each position in the list corresponds to the same level of priority across both the FQDN and Port fields.

5.1.11 Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Metrics

5.1.11.1 oc_ingressgateway_rss_ratelimit_total

Table 5-80 oc_ingressgateway_rss_ratelimit_total

Field	Details
Metric Details	Number of request for which RSS based rate limiting was applied and request was successfully forwarded.
Microservice	N32 Ingress Gateway
Type	Counter
Dimensions	- ErrorOriginator - InstanceIdentifier - Method - PLMN_ID - Remote_SEPP_Set - Scheme - Status
Metric filter	Status = accepted

5.1.11.2 oc_ingressgateway_rss_ratelimit_total

Table 5-81 oc_ingressgateway_rss_ratelimit_total

Field	Details
Metric Details	Number of request for which RSS based rate limiting was not applied.
Microservice	N32 Ingress Gateway
Type	Counter
Dimensions	- Method - Status - Scheme - InstanceIdentifier - ErrorOriginator
Metric filter	Status = ratelimit not applied

5.1.11.3 ocsepp_configmgr_routefailure_total

Table 5-82 ocsepp_configmgr_routefailure_total

Field	Details
Metric Details	Metric pegged due to route sync issue in SEPP.
Microservice	Config Manager
Type	Counter

Table 5-82 (Cont.) ocsepp_configmgr_routefailure_total

Field	Details
Dimensions	• http_status • http_error_message • app

5.1.11.4 oc_ingressgateway_rss_ratelimit_total

Table 5-83 oc_ingressgateway_rss_ratelimit_total

Field	Details
Metric Details	Number of request for which RSS based rate limiting was applied but request had to be dropped.
Microservice	N32 Ingress Gateway
Type	Counter
Dimensions	• ErrorOriginator • InstanceIdentifier • Method • PLMN_ID • Remote_SEPP_Set • Scheme • Status
Metric filter	Status = dropped

5.1.11.5 ocsepp_configmgr_ingress_rate_limit_validation_failure_total

Table 5-84 ocsepp_configmgr_ingress_rate_limit_validation_failure_total

Field	Details
Metric Details	This metrics is pegged whenever there is a failure related to Ingress Rate Limiting.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • pod • namespace • microservice • node • container • vendor

5.1.12 Topology Hiding Metrics

5.1.12.1 ocsepp_topology_latency_seconds_count

Table 5-85 ocsepp_topology_latency_seconds_count

Field	Details
Metric Details	This metric is used to display the number of ingress requests processed at cn32f and pn32f in a particular time span (in seconds).
Microservice	Consumer N32f, Producer N32f
Type	Histogram
Dimensions	• app (consumer, producer) • chart • container • direction • message_type • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • nf_service_type • vendor

Note

The dimension "peer_plmn_id" is applicable only for Consumer N32F.

5.1.12.2 ocsepp_topology_latency_seconds_max

Table 5-86 ocsepp_topology_latency_seconds_max

Field	Details
Metric Details	This metrics is used to display the maximum processing time of an ingress request at cn32f and pn32f in seconds.
Microservice	Consumer N32f, Producer N32f
Type	Histogram

Table 5-86 (Cont.) ocsepp_topology_latency_seconds_max

Field	Details
Dimensions	• app (consumer, producer) • chart • container • direction • message_type • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_service_type • nf_type • vendor

Note

The dimension "peer_plmn_id" is applicable only for Consumer N32F.

5.1.12.3 ocsepp_topology_latency_seconds_sum

Table 5-87 ocsepp_topology_latency_seconds_sum

Field	Details
Metric Details	This metrics is used to display the average processing time of all the ingress request at cn32f and pn32f for a particular time.
Microservice	Consumer N32f, Producer N32f
Type	Histogram
Dimensions	• app (consumer, producer) • chart • container • direction • message_type • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • nf_service_type • vendor

Note

The dimension "peer_plmn_id" is applicable only for Consumer N32F.

5.1.12.4 ocsepp_topology_header_success_total

Table 5-88 ocsepp_topology_header_success_total

Field	Details
Metric Details	Count of headers for which topology hiding and recovery was successful
Microservice	Consumer N32f, Producer N32f
Type	Counter
Dimensions	• app (consumer, producer) • chart • container • header • namespace • nf_instance_id • pod • vendor

5.1.12.5 ocsepp_topology_header_failure_total

Table 5-89 ocsepp_topology_header_failure_total

Field	Details
Metric Details	Count of headers for which topology hiding and recovery failed
Microservice	Consumer N32f, Producer N32f
Type	Counter
Dimensions	• app (consumer, producer) • chart • container • header • namespace • nf_instance_id • http_error_message • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor

Note

The dimension "peer_plmn_id" is applicable only for Consumer N32F.

5.1.12.6 ocsepp_topology_body_success_total

Table 5-90 ocsepp_topology_body_success_total

Field	Details
Metric Details	Count of body attributes for which topology hiding and recovery was successful.
Microservice	Consumer N32f, Producer N32f
Type	Counter
Dimensions	• app (Consumer, Producer) • chart • container • http_method • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor

Note

The dimension "peer_plmn_id" is applicable only for Consumer N32F.

5.1.12.7 ocsepp_topology_body_failure_total

Table 5-91 ocsepp_topology_body_failure_total

Field	Details
Metric Details	Count of body for which topology hiding and recovery failed
Microservice	Consumer N32f, Producer N32f
Type	Counter
Dimensions	• app (Consumer, Producer) • api_url • http_error_message • chart • container • http_method • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • http_status • vendor

Note

The dimension "peer_plmn_id" is applicable only for Consumer N32F.

5.1.12.8 ocsepp_topology_success_total

Table 5-92 ocsepp_topology_success_total

Field	Details
Metric Details	Count of messages for which topology hiding or recovery was successful
Microservice	Consumer N32f, Producer N32f
Type	Counter
Dimensions	• app (consumer, producer) • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor

Note

The dimension "peer_plmn_id" is applicable only for Consumer N32F.

5.1.12.9 ocsepp_topology_invalid_header_regex_configured_total

Table 5-93 ocsepp_topology_invalid_header_regex_configured_total

Field	Details
Metric Details	If configured header regex pattern is invalid, this metric will be pegged.
Microservice	Consumer N32f, Producer N32f
Type	Counter
Dimensions	• vendor • nf_instance_id • http_error_message

5.1.12.10 ocsepp_topology_header_regex_not_configured_total

Table 5-94 ocsepp_topology_header_regex_not_configured_total

Field	Details
Metric Details	If header regex pattern is not configured, this metric will be pegged.

Table 5-94 (Cont.) ocsepp_topology_header_regex_not_configured_total

Field	Details
Microservice	Consumer N32f, Producer N32f
Type	Counter
Dimensions	• vendor • nf_instance_id • http_error_message

5.1.13 Cat 0 - SBI Message Schema Validation Metrics

5.1.13.1 ocsepp_message_validation_applied_total

Table 5-95 ocsepp_message_validation_applied_total

Field	Details
Metric Details	Total number of requests for which message validation is applied.
Microservice	Producer N32f, Consumer N32F
Type	Counter
Dimensions	• app (Consumer, Producer) • chart • container • http_method • namespace • nf_instance_id • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • remote_sepp_set_name • vendor

Note

- The dimension "peer_plmn_id" is applicable only for Consumer N32F.
- An additional dimension "sourceRss" is applicable for Consumer N32F in Roaming Hub mode.

5.1.13.2 ocsepp_message_validation_on_body_failure_total

Table 5-96 ocsepp_message_validation_on_body_failure_total

Field	Details
Metric Details	Number of requests in which message validation failed on body at SEPP end.

Table 5-96 (Cont.) ocsepp_message_validation_on_body_failure_total

Field	Details
Microservice	Producer N32f, Consumer N32F
Type	Counter
Dimensions	• app • chart • container • http_method • namespace • nf_instance_id • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • remote_sepp_set_name • request_path • vendor • http_status

Note

- The dimension "peer_plmn_id" is applicable only for Consumer N32F.
- An additional dimension "sourceRss" is applicable for Consumer N32F in Roaming Hub mode.

5.1.13.3 ocsepp_message_validation_on_header_failure_total

Table 5-97 ocsepp_message_validation_on_header_failure_total

Field	Details
Metric Details	Number of requests for which message validation failed on query parameters at SEPP end.
Microservice	Producer N32f, Consumer N32F
Type	Counter

Table 5-97 (Cont.) ocsepp_message_validation_on_header_failure_total

Field	Details
Dimensions	• app • chart • container • http_method • namespace • nf_instance_id • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • remote_sepp_set_name • request_path • vendor • http_status

Note

- The dimension "peer_plmn_id" is applicable only for Consumer N32F.
- An additional dimension "sourceRss" is applicable for Consumer N32F in Roaming Hub mode.

5.1.13.4 ocsepp_configmgr_message_validation_config_failure_total

Table 5-98 ocsepp_configmgr_message_validation_config_failure_total

Field	Details
Metric Details	This metrics is pegged whenever an error/exception occurs while doing configurations for message schema validation.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • chart • namespace • microservice • node • pod • container • vendor

5.1.14 Cat 1 - Service API Validation Metrics

5.1.14.1 ocsepp_security_service_api_failure_total

Table 5-99 ocsepp_security_service_api_failure_total

Field	Details
Metric Details	Metric is common for both CN32F and PN32F. Separation happens based on "app" tag. Number of requests failed as Method and Resource URI were not Allowed.
Microservice	N32f
Type	Counter
Dimensions	• app (Consumer, Producer) • chart • container • http_method • namespace • nf_instance_id • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • resource_uri • http_status • vendor • nf_service_type • nf_type

Note

The dimension "peer_plmn_id" is applicable only for Consumer N32F.

5.1.15 Cat 2 - Network ID Validation Metrics

5.1.15.1 ocsepp_originating_network_id_validation_success_total

Table 5-100 ocsepp_originating_network_id_validation_success_total

Field	Details
Metric Details	Total number of requests for Cat-2 Network id header validation successful. Condition : This metric will be pegged only when cat-2 is enabled and header validation is successful.
Microservice	Pn32f
Metric type	Counter

Table 5-100 (Cont.) ocsepp_originating_network_id_validation_success_total

Field	Details
Dimensions	• app • chart • container • header_type • method • peer_plmn_id • namespace • NfServiceType • NfType • peer_domain • peer_fqdn • pod • remote_sepp_name • remote_sepp_set_name • vendor

5.1.15.2 ocsepp_network_id_validation_body_failure_total

Table 5-101 ocsepp_network_id_validation_body_failure_total

Field	Details
Metric Details	Number of requests for which the Network ID body validation feature failed.
Microservice	Producer N32f, Consumer N32F
Type	Counter
Dimensions	• app (Consumer, Producer) • cause • chart • container • error_action • http_method • namespace • nf_instance_id • peer_domain • peer_fqdn • plmn_identifier • pod • remote_sepp_name • remote_sepp_set_name • resource_uri • sepp_type • vendor

5.1.15.3 ocsepp_network_id_validation_header_failure_total

Table 5-102 ocsepp_network_id_validation_header_failure_total

Field	Details
Metric Details	Number of request for which Network ID header validation feature were failed.
Microservice	Producer N32f, Consumer N32F
Type	Counter
Dimensions	• app (Consumer, Producer) • cause • chart • container • error_action • http_method • namespace • nf_instance_id • peer_domain • peer_fqdn • plmn_identifier • pod • remote_sepp_name • remote_sepp_set_name • resource_uri • sepp_type • vendor

5.1.15.4 ocsepp_configmgr_network_id_header_validation_failure_total

Table 5-103 ocsepp_configmgr_network_id_header_validation_failure_total

Field	Details
Metric Details	This metrics is pegged whenever there is a header validation failure for Network ID Validation feature.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • vendor • chart • pod • namespace • microservice • node • pod • container

5.1.15.5 ocsepp_configmgr_network_id_rule_config_failure_total

Table 5-104 ocsepp_configmgr_network_id_rule_config_failure_total

Field	Details
Metric Details	This metrics is pegged whenever there is a configuration related error for Network ID Validation feature.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • http_method • chart • pod • namespace • microservice • node • container • vendor

5.1.15.6 ocsepp_configmgr_network_id_body_config_failure_total

Table 5-105 ocsepp_configmgr_network_id_body_config_failure_total

Field	Details
Metric Details	This metrics is pegged whenever there is an error related to body configurations for Network ID Validation feature.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • http_method • chart • pod • namespace • microservice • node • container • vendor

5.1.15.7 ocsepp_configmgr_network_id_rule_validation_failure_total

Table 5-106 ocsepp_configmgr_network_id_rule_validation_failure_total

Field	Details
Metric Details	This metrics is pegged whenever validation fails for Network ID Validation feature.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • chart • namespace • microservice • node • pod • container • vendor

5.1.15.8 ocsepp_configmgr_network_id_header_config_failure_total

Table 5-107 ocsepp_configmgr_network_id_header_config_failure_total

Field	Details
Metric Details	This metrics is pegged whenever there is an error related to header configurations for Network ID Validation feature.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • chart • namespace • microservice • node • pod • container • vendor

5.1.16 Cat 3 - Previous Location Check Metrics

5.1.16.1 ocsepp_previous_location_exception_failure_total

Table 5-108 ocsepp_previous_location_exception_failure_total

Field	Details
Metric Details	Number of requests, for which previous location validation check failed due to exceptions.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (Consumer, Producer) • cause • chart • container • error_action • http_method • namespace • nf_instance_id • peer_domain • peer_fqdn • pod • remote_sepp_name • remote_sepp_set_name • resource_uri • vendor

5.1.16.2 ocsepp_previous_location_validation_success_total

Table 5-109 ocsepp_previous_location_validation_success_total

Field	Details
Metric Details	Number of requests, for which previous location validation feature is successful.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (Consumer, Producer) • chart • container • http_method • namespace • nf_instance_id • peer_domain • peer_fqdn • pod • remote_sepp_name • remote_sepp_set_name • resource_uri • vendor

5.1.16.3 ocsepp_previous_location_validation_failure_total

Table 5-110 ocsepp_previous_location_validation_failure_total

Field	Details
Metric Details	Number of requests, for which previous location validation check failed.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (Consumer, Producer) • cause • chart • container • error_action • http_method • namespace • nf_instance_id • peer_domain • peer_fqdn • pod • remote_sepp_name • remote_sepp_set_name • resource_uri • vendor

5.1.16.4 ocsepp_previous_location_validation_requests_total

Table 5-111 ocsepp_previous_location_validation_requests_total

Field	Details
Metric Details	Number of requests, for which previous location validation feature is applied.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (Consumer, Producer) • chart • container • http_method • namespace • nf_instance_id • peer_domain • peer_fqdn • pod • remote_sepp_name • remote_sepp_set_name • resource_uri • vendor

5.1.16.5 ocsepp_pn32f_notification_fromNrf_total

Table 5-112 ocsepp_pn32f_notification_fromNrf_total

Field	Details
Metric Details	It is pegged every time the notification is received on pn32f from NRF for UDR profile change.
Microservice	Producer N32f
Type	Counter
Dimensions	• app (Consumer, Producer) • chart • container • namespace • nf_instance_id • nf_type • notification_status • notification_type • pod • vendor

5.1.17 Cat-3 Time Check for Roaming Subscribers

5.1.17.1 ocsepp_time_unauthenticated_location_validation_requests_total

Table 5-113 ocsepp_time_unauthenticated_location_validation_requests_total

Field	Details
Metric Details	Total number of requests received by the Cat-3 time check feature.
Microservice	Producer N32f
Type	Counter
Dimensions	• app • assertedPlmnValue • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • pod • release • remote_sepp_name • remote_sepp_set_name • vendor

5.1.17.2 ocsepp_time_unauthenticated_location_validation_success_total

Table 5-114 ocsepp_time_unauthenticated_location_validation_success_total

Field	Details
Metric Details	Total number of requests received by the Cat-3 time check feature with successful validation.
Microservice	Producer N32f
Type	Counter
Dimensions	• app • assertedPlmnValue • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • pod • reason • release • remote_sepp_name • remote_sepp_set_name • vendor

5.1.17.3 ocsepp_time_unauthenticated_location_validation_failure_total

Table 5-115 ocsepp_time_unauthenticated_location_validation_failure_total

Field	Details
Metric Details	Total number of requests received by the Cat-3 time check feature with failed validation.
Microservice	Producer N32f
Type	Counter
Dimensions	• app • assertedPlmnValue • chart • container • error_action • namespace • nf_instance_id • peer_domain • peer_fqdn • pod • release • remote_sepp_name • remote_sepp_set_name • vendor

5.1.17.4 ocsepp_time_unauthenticated_location_exception_failure_total

Table 5-116 ocsepp_time_unauthenticated_location_exception_failure_total

Field	Details
Metric Details	Total number of requests received by the Cat-3 time check feature where exception was encountered.
Microservice	Producer N32f
Type	Counter
Dimensions	• app • assertedPlmnValue • chart • container • error_action • namespace • nf_instance_id • peer_domain • peer_fqdn • pod • release • remote_sepp_name • remote_sepp_set_name • vendor

5.1.17.5 ocsepp_time_unauthenticated_location_blocklist_requests_total

Table 5-117 ocsepp_time_unauthenticated_location_blocklist_requests_total

Field	Details
Metric Details	Total number of requests received by the Cat-3 time check feature which were blacklisted.
Microservice	Producer N32f
Type	Counter
Dimensions	• app • assertedPlmnValue • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • pod • release • remote_sepp_name • remote_sepp_set_name • vendor

5.1.17.6 ocsepp_configmgr_country_config_failure_total

Table 5-118 ocsepp_configmgr_country_config_failure_total

Field	Details
Metric Details	Total number of exceptions that occurred while saving default country configurations or deleting the same for the Cat-3 time check feature.
Microservice	Producer N32f
Type	Counter
Dimensions	• app • application • chart • container • http_error_message • http_status • http_method • namespace • node • pod • release • vendor

5.1.17.7 ocsepp_configmgr_exception_mcc_config_failure_total

Table 5-119 ocsepp_configmgr_exception_mcc_config_failure_total

Field	Details
Metric Details	This metrics is pegged whenever the configurations related to exception mcc fails.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • chart • namespace • microservice • node • pod • container • vendor

5.1.17.8 ocsepp_configmgr_time_unauthenticated_config_failure_total

Table 5-120 ocsepp_configmgr_time_unauthenticated_config_failure_total

Field	Details
Metric Details	This metrics is pegged whenever the configurations related to time unauthenticated check fails.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_status • chart • namespace • microservice • node • pod • container • vendor

5.1.18 Rate Limiting for Egress Roaming Signaling per PLMN Metrics

5.1.18.1 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-121 oc_ingressgateway_plmn_egress_ratelimit_total

Field	Details
Metric Details	Number of requests for which Egress Rate Limiting was applied and request was successfully forwarded because tokens were available for the Egress Rate Limit List.
Microservice	Ingress Gateway
Type	Counter
Dimensions	• Status • PLMN_ID • Scheme • Egress Rate Limit List • InstanceIdentifier • ErrorOriginator
Status	ERL_MATCH_TOKEN_AVAILABLE_FWD

5.1.18.2 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-122 oc_ingressgateway_plmn_egress_ratelimit_total

Field	Details
Metric Details	Number of requests for which Egress Rate Limiting was applied, here tokens were not available to process the request, request was rejected as its priority was low (above than the configured cutoff).
Microservice	Ingress Gateway
Type	Counter
Dimensions	• Status • PLMN_ID • Scheme • Egress Rate Limit List • InstanceIdentifier • ErrorOriginator
Status	ERL_MATCH_NO_TOKEN_LOW_PRI_REJECT

5.1.18.3 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-123 oc_ingressgateway_plmn_egress_ratelimit_total

Field	Details
Metric Details	The number of requests for which egress rate limiting was applied, here tokens were not available to process the request, but the request was forwarded as its priority was high (less than the configured cutoff).
Microservice	Ingress Gateway
Type	Counter
Dimensions	• Status • PLMN_ID • Scheme • Egress Rate Limit List • InstanceIdentifier • ErrorOriginator
Status	ERL_MATCH_NO_TOKEN_HIGH_PRI_FWD

5.1.18.4 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-124 oc_ingressgateway_plmn_egress_ratelimit_total

Field	Details
Metric Details	Number of requests for which rate limiting could not be applied as invalid PLMN ID was sent in the request. The request was forwarded.

Table 5-124 (Cont.) oc_ingressgateway_plmn_egress_ratelimit_total

Field	Details
Microservice	Ingress Gateway
Type	Counter
Dimensions	- Status - PLMN_ID - Scheme - Egress Rate Limit List - InstanceIdentifier - ErrorOriginator
Status	ERROR_UNABLE_TO_EXTRACT_PLMN_FWD

5.1.18.5 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-125 oc_ingressgateway_plmn_egress_ratelimit_total

Field	Details
Metric Details	Number of requests for which rate limiting could not be applied as none of the Egress Rate Limit List contains the corresponding PLMN ID. The request was forwarded.
Microservice	Ingress Gateway
Type	Counter
Dimensions	- Status - PLMN_ID - Scheme - Egress Rate Limit List - InstanceIdentifier - ErrorOriginator
Status	ERL_NO_MATCH_FWD

5.1.18.6 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-126 oc_ingressgateway_plmn_egress_ratelimit_total

Field	Details
Metric Details	Number of requests for which rate limiting could not be applied as some unexpected exception was raised during the execution of the rate limit filter for the request. The request was forwarded.
Microservice	Ingress Gateway
Type	Counter

Table 5-126 (Cont.) oc_ingressgateway_plmn_egress_ratelimit_total

Field	Details
Dimensions	- Status - PLMN_ID - Scheme - Egress Rate Limit List - InstanceIdentifier - ErrorOriginator
Status	ERROR_INTERNAL_FWD

5.1.19 Config Manager Metrics

5.1.19.1 ocsepp_configmgr_rpp_config_failure_total

Table 5-127 ocsepp_configmgr_rpp_config_failure_total

Field	Details
Metric Details	This metrics is pegged whenever there is a Remote SEPP configuration failure.
Microservice	Config Manager
Type	Counter
Dimensions	- app - http_error_message - http_method - http_status - vendor - chart - pod - namespace - container

5.1.19.2 ocsepp_configmgr_rpp_validation_failure_total

Table 5-128 ocsepp_configmgr_rpp_validation_failure_total

Field	Details
Metric Details	This metric is pegged whenever there is a change in mandatory parameter of Remote SEPP Profile or mandatory parameter is missing.
Microservice	Config Manager
Type	Counter

Table 5-128 (Cont.) ocsepp_configmgr_rpp_validation_failure_total

Field	Details
Dimensions	• app • http_error_message • http_status • vendor • chart • pod • namespace • container

5.1.19.3 ocsepp_configmgr_routeupdate_total

Table 5-129 ocsepp_configmgr_routeupdate_total

Field	Details
Metric Details	This metric shows the total number of times config-mgr microservice has tried to update the route details for Egress Gateway microservices.
Microservice	Config Manager
Type	Counter
Dimensions	• app • running • chart • pod • namespace • container

5.1.19.4 ocsepp_configmgr_n32f_context_config_failure_total

Table 5-130 ocsepp_configmgr_n32f_context_config_failure_total

Field	Details
Metric Details	This metric is pegged when an HS context is fetched by name or FQDN and the query fails or returns empty.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • vendor • chart • pod • namespace • container • http_error_message • http_method • http_status • microservice • node

5.1.19.5 ocsepp_configmgr_delete_route_failure_total

Table 5-131 ocsepp_configmgr_delete_route_failure_total

Field	Details
Metric Details	This metrics is pegged whenever the gateway routes are deleted.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_request_message • chart • pod • namespace • microservice • node • container • vendor

5.1.19.6 ocsepp_configmgr_put_route_failure_total

Table 5-132 ocsepp_configmgr_put_route_failure_total

Field	Details
Metric Details	This metrics is pegged whenever the creation of gateway routes fails.
Microservice	Config Manager
Type	Counter
Dimensions	• app • application • http_error_message • http_request_message • chart • pod • namespace • microservice • node • container • vendor

5.1.19.7 ocsepp_configmgr_delete_peer_set_failure_total

Table 5-133 ocsepp_configmgr_delete_peer_set_failure_total

Field	Details
Metric Details	This metrics is pegged whenever the peer set deletion fails.
Microservice	Config Manager
Type	Counter

Table 5-133 (Cont.) ocsepp_configmgr_delete_peer_set_failure_total

Field	Details
Dimensions	• app • application • http_error_message • http_request_message • chart • pod • namespace • microservice • node • container • vendor

5.1.20 Support for Originating Network Id Header Validation, Insertion, and Transposition Metrics

5.1.20.1 ocsepp_originating_network_id_header_added_total

Table 5-134 ocsepp_originating_network_id_header_added_total

Field	Details
Metric Details	Total number of requests on which "3gpp-sbi-originating-network-id" or "3gpp-sbi-asserted-plmn-id" header is added.
Microservice	PN32F CN32F
Type	Counter
Dimensions	CN32F: • header_name • header_value • remote_sepp_name • remote_sepp_set_name • peer_fqdn • peer_plmn_id • request_path • vendor • nf_instance_id PN32F: • header_name • header_value • remote_sepp_name • remote_sepp_set_name • peer_fqdn • request_path • vendor • nf_instance_id

5.1.20.2 ocsepp_originating_network_id_header_transposed

Table 5-135 ocsepp_originating_network_id_header_transposed

Field	Details
Metric Details	Total number of requests in which the header name ("3gpp-sbi-originating-network-id" or "3gpp-sbi-asserted-plmn-id") is transposed to another header name. Example: "3gpp-sbi-originating-network-id" is transposed into "3gpp-sbi-asserted-plmn-id" or "3gpp-sbi-asserted-plmn-id" is transposed into "3gpp-sbi-originating-network-id"
Microservice	PN32F
Type	Counter
Dimensions	- transposed_header - header_value - header_from_request - remote_sepp_name - remote_sepp_set_name - peer_fqdn - request_path - vendor - nf_instance_id Note: The header_value dimension will not be populated in cases where SEPP does not make any modifications in incoming request header value for 3gpp-Sbi-Originating-Network-Id or 3gpp-Sbi-Asserted-Plmn-Id header while header transposition.

5.1.20.3 ocsepp_originating_network_id_header_addition_failed

Table 5-136 ocsepp_originating_network_id_header_addition_failed

Field	Details
Metric Details	Total number of requests in which missing header ("3gpp-sbi-originating-network-id" or "3gpp-sbi-asserted-plmn-id") could not be added due to any internal error.
Microservice	CN32F
Type	Counter
Dimensions	- request_path - vendor - nf_instance_id

5.1.20.4 ocsepp_originating_network_id_header_add_or_transpose_failed

Table 5-137 ocsepp_originating_network_id_header_add_or_transpose_failed

Field	Details
Metric Details	Total number of requests for which the missing header ("3gpp-sbi-originating-network-id" or "3gpp-sbi-asserted-plmn-id") could not be added, or header transposition was failed due to any internal error.
Microservice	PN32F
Type	Counter
Dimensions	- request_path - vendor - nf_instance_id

5.1.21 Support for TLS 1.3 Metrics

5.1.21.1 oc_ingressgateway_incoming_tls_connections

Table 5-138 oc_ingressgateway_incoming_tls_connections

Field	Details
Metric Details	Number of TLS connections received on the Ingress Gateway and their negotiated versions. The version can be TLS 1.2 or TLS 1.3.
Microservice	PLMN Ingress Gateway N32 Ingress Gateway
Type	Gauge
Dimensions	- NegotiatedTLSVersion - Host - Direction - InstanceIdentifier

5.1.21.2 oc_egressgateway_outgoing_tls_connections

Table 5-139 oc_egressgateway_outgoing_tls_connections

Field	Details
Metric Details	Number of TLS connections sent on the Egress Gateway and their negotiated versions. The version can be TLS 1.2 or TLS 1.3.
Microservice	PLMN Egress Gateway N32 Egress Gateway
Type	Gauge

Table 5-139 (Cont.) oc_egressgateway_outgoing_tls_connections

Field	Details
Dimensions	- NegotiatedTLSVersion - Host - Direction - InstanceIdentifier

5.1.21.3 security_cert_x509_expiration_seconds

Table 5-140 security_cert_x509_expiration_seconds

Field	Details
Metric Details	Time to certificate expiry in epoch seconds.
Microservice	PLMN Ingress Gateway N32 Ingress Gateway PLMN Egress Gateway N32 Egress Gateway
Type	Histogram
Dimensions	- app - chart - container - namespace - pod - serialNumber - subject - CN (CommonName) - O (Organization) - L (Locality) - ST (State or ProvinceName) - C (CountryName)

5.1.22 Proactive Status Updates on SEPP Metrics

5.1.22.1 oc_egressgateway_peer_health_status

Table 5-141 oc_egressgateway_peer_health_status

Field	Details
Metric Details	Set to 1, if a peer is unhealthy and reset to 0, when it becomes healthy again. Set to -1, if peer is removed from peerconfiguration.
Microservice	N32 Egress Gateway, PLMN Egress Gateway
Metric type	Gauge
Dimensions	- peer - vfqdn

5.1.22.2 oc_egressgateway_peer_health_ping_request_total

Table 5-142 oc_egressgateway_peer_health_ping_request_total

Field	Details
Metric Details	Incremented every time a health ping is sent toward a peer.
Microservice	N32 Egress Gateway, PLMN Egress Gateway
Metric type	Counter
Dimensions	- peer - vfqdn

5.1.22.3 oc_egressgateway_peer_health_ping_response_total

Table 5-143 oc_egressgateway_peer_health_ping_response_total

Field	Details
Metric Details	Incremented every time a health ping response(irrespective of success or failure) is received from a peer.
Microservice	N32 Egress Gateway, PLMN Egress Gateway
Metric type	Counter
Dimensions	- peer - vfqdn - status Note: vfqdn will be populated if virtual FQDN is configured.

5.1.22.4 oc_egressgateway_peer_health_status_transitions_total

Table 5-144 oc_egressgateway_peer_health_status_transitions_total

Field	Details
Metric Details	Incremented every time a peer transitions from AVAILABLE to UNAVAILABLE or UNAVAILABLE to AVAILABLE.
Microservice	N32 Egress Gateway, PLMN Egress Gateway
Metric type	Counter
Dimensions	- peer - vfqdn - from - to

5.1.22.5 oc_ingressgateway_health_check_incoming_ping_total

Table 5-145 oc_ingressgateway_health_check_incoming_ping_total

Field	Details
Metric Details	Incremented every time a health ping is received from a peer.
Microservice	N32 Ingress Gateway
Metric type	Counter
Dimensions	- method - resource_uri

5.1.23 Multiple SEPP instances on shared cnDBTier cluster Metrics

5.1.23.1 ocsepp_cn32f_database_connectivity_healthy

Table 5-146 ocsepp_cn32f_database_connectivity_healthy

Field	Details
Metric Details	This metric will be pegged when database connectivity with cn32f service is established and healthy and when connection breaks then set to 0.
Microservice	cn32f
Metric type	Gauge
Dimensions	- app - chart - container - database - namespace - nf_instance_id - pod - release - vendor

5.1.23.2 ocsepp_pn32f_database_connectivity_healthy

Table 5-147 ocsepp_pn32f_database_connectivity_healthy

Field	Details
Metric Details	This metric will be pegged when database connectivity with pn32f service is established and healthy and when connection breaks then set to 0.
Microservice	pn32f
Metric type	Gauge

Table 5-147 (Cont.) ocsepp_pn32f_database_connectivity_healthy

Field	Details
Dimensions	• app • chart • container • database • namespace • nf_instance_id • pod • release • vendor

5.1.23.3 ocsepp_cn32c_database_connectivity_healthy

Table 5-148 ocsepp_cn32c_database_connectivity_healthy

Field	Details
Metric Details	This metric will be pegged when database connectivity with cn32c service is established and healthy and when connection breaks then set to 0.
Microservice	cn32c
Metric type	Gauge
Dimensions	• app • chart • container • database • namespace • nf_instance_id • pod • release • vendor

5.1.23.4 ocsepp_pn32c_database_connectivity_healthy

Table 5-149 ocsepp_pn32c_database_connectivity_healthy

Field	Details
Metric Details	This metric will be pegged when database connectivity with pn32c service is established and healthy and when connection breaks then set to 0.
Microservice	pn32c
Metric type	Gauge

Table 5-149 (Cont.) ocsepp_pn32c_database_connectivity_healthy

Field	Details
Dimensions	• app • chart • container • database • namespace • nf_instance_id • pod • release • vendor

5.1.23.5 ocsepp_configmgr_database_connectivity_healthy

Table 5-150 ocsepp_configmgr_database_connectivity_healthy

Field	Details
Metric Details	This metric will be pegged when database connectivity with config manager service is established and healthy and when connection breaks then set to 0.
Microservice	config-mgr
Metric type	Gauge
Dimensions	• app • application • chart • container • database • namespace • nf_instance_id • node • pod • release • vendor

5.1.23.6 ocsepp_coherence_database_connectivity_healthy

Table 5-151 ocsepp_coherence_database_connectivity_healthy

Field	Details
Metric Details	This metric will be pegged when database connectivity with coherence service is established and healthy and when connection breaks then set to 0..
Microservice	coherence-svc
Metric type	Gauge

Table 5-151 (Cont.) ocsepp_coherence_database_connectivity_healthy

Field	Details
Dimensions	• app • chart • container • database • namespace • nf_instance_id • pod • release • vendor

5.1.24 LCI and OCI Headers Metrics

5.1.24.1 oc_ingressgateway_headers_lci_total

Table 5-152 oc_ingressgateway_headers_lci_total

Field	Details
Metric Details	This counts the total number of lci header attached.
Microservice	n32-ingress-gateway, plmn-ingress-gateway
Metric type	Counter
Dimensions	• scope • app • chart • container • namespace • pod

5.1.24.2 oc_ingressgateway_headers_oci_total

Table 5-153 oc_ingressgateway_headers_oci_total

Field	Details
Metric Details	This counts the total number of oci header attached.
Microservice	n32-ingress-gateway, plmn-ingress-gateway
Metric type	Counter
Dimensions	• scope • app • chart • container • namespace • pod

5.1.25 Integrating SEPP with 5G Network Intelligence Fabric (5G NIF) Metrics

5.1.25.1 ocsepp_nif_discovery_requests_total

Table 5-154 ocsepp_nif_discovery_requests_total

Field	Details
Metric Details	Total number of requests received to NRF when querying for NIF.
Microservice	Config-mgr
Type	Counter
Dimensions	• app • chart • container • namespace • nf_instance_id • pod • release • vendor

5.1.25.2 ocsepp_nif_discovery_responses_total

Table 5-155 ocsepp_nif_discovery_responses_total

Field	Details
Metric Details	Total number of responses received from NRF when querying for NIF.
Microservice	Config-mgr
Type	Counter
Dimensions	• app • chart • container • namespace • nf_instance_id • pod • release • status_code • vendor

5.1.25.3 ocsepp_nif_registration_status

Table 5-156 ocsepp_nif_registration_status

Field	Details
Metric Details	Total number of active NIFs for routing.
Microservice	Config-mgr
Type	Gauge

Table 5-156 (Cont.) ocsepp_nif_registration_status

Field	Details
Dimensions	• app • chart • container • namespace • nf_instance_id • pod • release • status_code • vendor

5.1.25.4 ocsepp_pn32f_nif_error_copy_requests_total

Table 5-157 ocsepp_pn32f_nif_error_copy_requests_total

Field	Details
Metric Details	Total number of message copy requests sent to NIF when error response is generated.
Microservice	pn32f-svc
Type	Counter
Dimensions	• app • chart • container • namespace • nf_instance_id • uri • pod • release • vendor

5.1.25.5 ocsepp_pn32f_nif_error_copy_responses_total

Table 5-158 ocsepp_pn32f_nif_error_copy_responses_total

Field	Details
Metric Details	Total number of message copy responses received from NIF when error response is generated.
Microservice	pn32f-svc
Type	Counter

Table 5-158 (Cont.) ocsepp_pn32f_nif_error_copy_responses_total

Field	Details
Dimensions	• app • chart • container • namespace • nf_instance_id • pod • release • status_code • vendor

5.1.26 NRF Selection Mechanisms Using nrf client Metrics

5.1.26.1 nrfclient_nrf_operative_status

Table 5-159 nrfclient_nrf_operative_status

Field	Details
Description	The current operative status of the NRF Instance. Note: The HealthCheck mechanism is an important component that allows monitoring and managing the health of NRF services. When enabled, it makes periodic HTTP requests to NRF services to check their availability and updates their status accordingly so that the metric <code>nrfclient_nrf_operative_status</code> updates properly. When disabled, for each NRF route, it is checked whether the retry time has expired. If so, the health state is reset to "HEALTHY", and the retry time is cleared. If the metric value is : • 0 - NRF is unavailable/unhealthy • 1- NRF is available/healthy
Type	Gauge
Dimension	NrfUri - URI of the NRF Instance

5.1.26.2 nrfclient_dns_lookup_request_total

Table 5-160 nrfclient_dns_lookup_request_total

Field	Details
Description	Total number of times a DNS lookup request is sent to the alternate route service. Note: The metric will be pegged only if <code>enableVirtualNrfResolution</code> is set to true.
Type	Counter
Dimension	• Scheme • VirtualFqdn

5.1.27 Overload Control Metrics

5.1.27.1 service_resource_overload_level

Table 5-161 service_resource_overload_level

Field	Details
Metric Details	Overload level value for Warning, Minor, Major, and Critical.
Microservice	Performance
Type	Gauge
Dimensions	• namespace • app • value • metric

5.1.27.2 oc_ingressgateway_route_overloadcontrol_discard_total

Table 5-162 oc_ingressgateway_route_overloadcontrol_discard_total

Field	Details
Metric Details	This metric is pegged when the incoming request is discarded by the Overload filter on priority or percentage basis.
Microservice	N32 Ingress Gateway
Type	Counter
Dimensions	• namespace • app • DiscardAction • Status

5.1.27.3 ingressgateway_svc_pending_count

Table 5-163 ingressgateway_svc_pending_count

Field	Details
Metric Details	The number of pending requests count on Ingress Gateway.
Microservice	N32 IGW
Type	Gauge
Dimensions	• namespace • app • value • servicenow

5.1.28 Traffic Prioritization using Egress Gateway Pod Protection Metrics

5.1.28.1 nrfclient_nw_conn_out_request_total

Table 5-164 nrfclient_nw_conn_out_request_total

Field	Details
Description	Total Number of request sent out from NRF-Client to the NRF (network) (inclusive of autonomous and on-demand).
Type	Counter
Dimensions	• MessageType • NfType • NfInstanceId • NfFqdn • NrfURI • OutgoingPriorityHeader • Priority • TargetNfType

5.1.28.2 oc_egressgateway_http_request_ratelimit_values_total

Table 5-165 oc_egressgateway_http_request_ratelimit_values_total

Field	Details
Description	This metric tracks the total number of requests handled by the rate limiting filter, categorized by whether each request was allowed or denied. If there was no route configuration for the dropped message, the Id attribute is set to "NA".
Type	Counter
Dimensions	• Id • Allowed • RouteLevelColor

5.1.28.3

oc_egressgateway_http_request_ratelimit_reject_chain_length_histogram_bucket

Table 5-166 oc_egressgateway_http_request_ratelimit_reject_chain_length_histogram_bucket

Field	Details
Description	This metric records the number of consecutive requests rejected before a successful acceptance and adds it to the corresponding bucket. This metric provides the number of failed requests due to bursty traffic.
Type	Histogram
Dimensions	NA

5.1.28.4 oc_egressgateway_congestion_cpu_state

Table 5-167 oc_egressgateway_congestion_cpu_state

Field	Details
Description	Records the CPU levels at the congestion state.
Type	Gauge
Dimensions	NA

5.1.28.5 oc_egressgateway_congestion_system_state

Table 5-168 oc_egressgateway_congestion_system_state

Field	Details
Description	Records the congestion level of the pod based on all the resources. The highest congestion level recorded for all resources in that interval is the system congestion level. However, currently, the congestion level is calculated based on CPU usage. Therefore, it records the system level based on CPU.
Type	Gauge
Dimensions	NA

5.1.28.6 oc_egressgateway_system_state_duration_percentage

Table 5-169 oc_egressgateway_system_state_duration_percentage

Field	Details
Description	Records the percentage of time the system was at a particular congestion level.
Type	Gauge
Dimensions	level

5.1.28.7 oc_egressgateway_congestion_level_total

Table 5-170 oc_egressgateway_congestion_level_total

Field	Details
Description	Counts the number of times a level was identified. This congestion level value varies between 0-10.
Type	Counter
Dimensions	NA

5.1.28.8 oc_egressgateway_congestion_level_bucket_total

Table 5-171 oc_egressgateway_congestion_level_bucket_total

Field	Details
Description	Identifies the number of times the congestion level was observed. Note: This metric also records any increase in CPU, which sometimes is not recorded in the oc_egressgateway_congestion_cpu_state metric. Refer to both the metrics to obtain the congestion level of the pod. 0,1,2,3...10 based on configured levels.
Type	Histogram
Dimensions	NA

5.1.28.9 oc_egressgateway_congestion_cpu_percentage_bucket

Table 5-172 oc_egressgateway_congestion_cpu_percentage_bucket

Field	Details
Description	Records the CPU utilization percentages, tracking the distribution of values across predefined ranges. For example, 0 to 10 %, 11 to 20 %, 21 to 30 %, 31 to 40 %, 41 to 50 %, 51 to 60 %, 61 to 70 %, 71 to 80 %, 81 to 90 %, and 91 to 100%.
Type	Histogram
Dimensions	NA

5.1.29

oc_egressgateway_http_request_ratelimit_reject_chain_duration_histogram_seconds_bucket

Table 5-173 oc_egressgateway_http_request_ratelimit_reject_chain_duration_histogram_seconds_bucket

Field	Details
Description	Records the reject chain duration.
Type	Histogram
Dimensions	NA

5.1.30 Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers Metrics

5.1.30.1 ocsepp_oauth_total_requests

Table 5-174 ocsepp_oauth_total_requests

Field	Details
Metric Details	Total number of OAuth Get token request received at Pn32f microservice.
Microservice	Producer N32f
Type	Counter
Dimensions	• peer_domain • peer_fqdn • nf_instance_id • remote_sepp_name • vendor

5.1.31 Alternate Route Metrics

5.1.31.1 oc_dns_srv_lookup_total

Table 5-175 oc_dns_srv_lookup_total

Field	Details
Metric Details	Track the number of times the DNS SRV lookup was done for a given scheme and FQDN.
Microservice	alternate route, Egress Gateway
Type	Counter
Dimensions	• Fqdn • Scheme

5.1.31.2 oc_fqdn_alternate_route_total

Table 5-176 oc_fqdn_alternate_route_total

Field	Details
Metric Details	Track number of registration, de-registration, and lookup calls received for a given scheme+FQDN. Note: Registration does not reflect active registration numbers. It captured number of registration requests received.
Microservice	alternate route, Egress Gateway
Type	Counter

Table 5-176 (Cont.) oc_fqdn_alterate_route_total

Field	Details
Dimensions	• Method • EndpointName • Scheme • Fqdn

5.1.31.3 oc_alterate_route_resultset

Table 5-177 oc_alterate_route_resultset

Field	Details
Metric Details	Provides number of alternate routes known for a given scheme and FQDN. Whenever DNS SRV lookup or static configuration is done, this metric provides number of known alternate route for a given pair. For example, <"http", "abc.oracle.com">: 2.
Microservice	alternate route, Egress Gateway
Type	Gauge
Dimensions	• Fqdn • Scheme

5.1.31.4 oc_configclient_request_total

Table 5-178 oc_configclient_request_total

Field	Details
Metric Details	This metric is pegged whenever a polling request is made from config client to the server for configuration updates.
Microservice	alternate route, Plmn Ingress Gateway, N32 Ingress Gateway
Type	Counter
Dimensions	Tags: releaseVersion, configVersion. • releaseVersion tag indicates the current chart version of alternate route service deployed. • configVersion tag indicates the current configuration version of alternate route service.

5.1.31.5 oc_configclient_response_total

Table 5-179 oc_configclient_response_total

Field	Details
Metric Details	This metric is pegged whenever a response is received from the server to client.
Microservice	alternate route, Plmn Ingress Gateway, N32 Ingress Gateway
Type	Counter

Table 5-179 (Cont.) oc_configclient_response_total

Field	Details
Dimensions	Tags: releaseVersion, configVersion, updated. - releaseVersion tag indicates the current chart version of alternate route service deployed. - configVersion tag indicates the current configuration version of alternate route service. - updated tag indicates whether there is a configuration update or not.

5.1.31.6 oc_configserver_reachability

Table 5-180 oc_configserver_reachability

Field	Details
Metric Details	Gauge metric to peg the reachability of config server.
Microservice	alternate route, PLMN Ingress Gateway, N32 Ingress Gateway
Type	Gauge
Dimensions	NA

5.1.32 Ingress and Egress Gateway Metrics

Table 5-181 Ingress and Egress Gateway Dimensions

Dimension	Details
NFType	Name of the NF Type in path. For Example: Path is /nxxx-yyy/vz/..... Where XXX(Upper Case) is NFType UNKNOWN if unable to extract NFType from the path
NFServiceType	Name of the Service with in the NF. Example: Path is /nxxx-yyy/vz/..... Where nxxx-yyy is NFServiceType UNKNOWN if unable to extract NFServiceType from the path
receivedResponseCode (Pod readiness state metric)	receivedResponseCode (Pod readiness state metric)
id (Pod readiness state metric)	Service profile Id of the backend svc
uri (Pod readiness state metric)	Service profile Uri of the backend svc
event	This tag captures the lifecycle event processed during the jetty request processing with the back-end svc
Host	(Ip or fqdn) : port of ingress gateway
DestinationHost	Destination ip/fqdn
client_type	client_type
HttpVersion	Http protocol version
oc_ingressgateway_pod_resource_stress_Type	The type of resource for which the pod protection threshold has reached.e.g. CPU, MEMORY, PENDING_REQUEST
XfccHeaderPresent	XfccHeaderPresent
consumerNFType	consumerNFType

Table 5-181 (Cont.) Ingress and Egress Gateway Dimensions

Dimension	Details
Scheme	Http protocol scheme
Path	Path predicate that matched the current request.
ClientCertIdentity	Certificate Identity of the client
content_available	content_available
Route_Path	Path predicate/Header predicate that matched the current request.
InstanceIdentifier	Prefix of the pod configured in helm when there are multiple instances in same deployment.
jetty_request_timeout	Tag to capture if a request at IGW failed due to request timeout at jetty level.
Virtual Host	The fqdn which requires alternate route svc resolution.
error_reason	Reason for failure response received. If message is sent in the response, then it is filled with the message otherwise exception class is filled. In case of successful response it is filled with "no-error".
ErrorOriginator	This tag captures the ErrorOriginator
quantile	This tag captures the latency values with ranges as 10ms, 20ms, 40ms, 80ms, 100ms, 200ms, 500ms, 1000ms, and 5000ms.
oc_ingressgateway_xfcc_header_validate_ServiceType	Name of the Service with in the NF.
oc_ingressgateway_dns_resolution_Status	oc_ingressgateway_dns_resolution_Status
oc_ingressgateway_global_ratelimit_Status	Request accepted or dropped
oc_ingressgateway_global_ratelimit_total_app	Application at which traffic rejection occurs - n32-ingress-gateway or plmn-ingress-gateway
oc_ingressgateway_global_ratelimit_total_Method	Request method received (POST , PUT , GET , PATCH , DELETE)
oc_ingressgateway_connection_failure_Host	destination ip/fqdn
oc_ingressgateway_connection_failure_Port	destination port
oc_ingressgateway_connection_failure_Direction	This tag determines the direction in which there is connection failure at IGW
oc_ingressgateway_xfcc_header_validate_Status	oc_ingressgateway_xfcc_header_validate_Status
oc_ingressgateway_xfcc_header_validate_Cause	This tag determines the validation cause for the xfcc header validation metric being pegged
oc_ingressgateway_incoming_pod_connections_rejected_Direction	The incoming connections rejected at IGW handled in pod protection
oc_ingressgateway_xfcc_header_validate_CertsCompared	This tag captures the total number of certificates compared in XFCC header at IGW during the header validation
oc_configclient_request_total_releaseVersion	This tag indicates the current release version of ingress gateway
oc_configclient_request_total_configVersion	This tag indicates the configuration version that ingress gateway is currently maintaining
oc_configclient_response_total_releaseVersion	This tag indicates the configuration version that ingress gateway is currently maintaining

Table 5-181 (Cont.) Ingress and Egress Gateway Dimensions

Dimension	Details
oc_configclient_response_total_configurationVersion	This tag indicates the configuration version that ingress gateway is currently maintaining
oc_configclient_response_total_updated	This tag indicates whether the configuration was updated or not
oc_ingressgateway_incoming_connections_Direction	This tag indicates the direction of connection established i.e, whether it is incoming or outgoing
oc_ingressgateway_incoming_connections_Host	This tag indicates the remote address of client connected to ingress gateway
oc_ingressgateway_outgoing_connections_Direction	This tag indicates the direction of connection established i.e, whether it is incoming or outgoing
oc_ingressgateway_going_connections_Host	This tag indicates the address of destination
Proxy	Value received for "x-custom-egress-proxy-header".
ConnectedHostIp	This tag captures the IP of destination host to which EGW sends ping requests
ConnectedHostFqdn	This tag captures the fqdn of destination host to which EGW sends ping requests
ConnectedHostPort	This tag captures the port of destination host to which EGW sends ping requests
oc_egressgateway_connection_failure_Host	destination ip/fqdn
oc_egressgateway_connection_failure_Port	destination port
oc_egressgateway_incoming_connections_Direction	This tag indicates the direction of connection established i.e, whether it is incoming or outgoing
oc_egressgateway_incoming_connections_Host	This tag indicates the remote address of client connected to ingress gateway
oc_egressgateway_outgoing_connections_Direction	This tag indicates the direction of connection established i.e, whether it is incoming or outgoing
oc_egressgateway_outgoing_connections_Host	This tag indicates address of destination
EndpointName	Request sent for
Reroute_Path	Path that matched the request to over corresponding route Example : /nef/**
Attempt	Attempt number for scp re-route. Example : 1 , 2 etc.,
peer	A peer is a single upstream endpoint defined in sbiRouting.peerConfiguration.
vfqdn	Virtual FQDN (vFQDN) is the logical DNS name used by Egress Gateway to represent a peer endpoint instance for routing and health checks.

5.1.32.1 Ingress Gateway Metrics

This section provides information about the Ingress Gateway metrics used in SEPP.

5.1.32.1.1 oc_ingressgateway_request_failure_total

Table 5-182 oc_ingressgateway_request_failure_total

Field	Details
Metric Details	This metric will be pegged when the request is discarded due to late arrival at Ingress Gateway.
Microservice	n32-ingress-gateway, plmn-ingress-gateway
Type	Counter
Dimensions	- ErrorDetail - ErrorOriginator - HttpStatus - InstanceIdentifier - Method - Scheme

5.1.32.1.2 oc_ingressgateway_sbitimer_timezone_mismatch

Table 5-183 oc_ingressgateway_sbitimer_timezone_mismatch

Field	Details
Metric Details	When the global.seppSbiTimerTimezone is set to ANY and timezone is not specified in the header then this metric is pegged at Ingress Gateway.
Microservice	n32-ingress-gateway, plmn-ingress-gateway
Type	Gauge
Dimensions	- Route path - Method - app - chart - container - method - namespace - path - pod - release

5.1.32.1.3 oc_ingressgateway_http_requests_total

Table 5-184 oc_ingressgateway_http_requests_total

Field	Details
Metric Details	This metric will be pegged as soon as the request reaches the Ingress gateway in the first custom filter of the application.
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Counter

Table 5-184 (Cont.) oc_ingressgateway_http_requests_total

Field	Details
Dimensions	• Method • NFType • NFServiceType • Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • ClientCertIdentity • consumerInstanceId • ConsumerFqdn Note: The dimension ClientCertIdentity is optional. The dimension will be populated once the custom values are finalized.

5.1.32.1.4 oc_ingressgateway_http_responses_total

Table 5-185 oc_ingressgateway_http_responses_total

Field	Details
Metric Details	This metric will be pegged in the last custom filter of the Ingress gateway while the response is being sent back to the consumer NF.
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Counter
Dimensions	• Status • Method • NFType • NFServiceType • Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • ClientCertIdentity Note: The dimension ClientCertIdentity is optional. The dimension will be only populated once the custom values are finalized.

5.1.32.1.5 oc_ingressgateway_request_latency_seconds

Table 5-186 oc_ingressgateway_request_latency_seconds

Field	Details
Metric Details	This metric will be pegged in the last custom filter of the Ingress gateway while the response is being sent back to the consumer NF. This metric tracks the amount of time taken for processing the request. It starts as soon the request reaches the first custom filter of the application and lasts till the response is sent back to the consumer NF from the last custom filter of the application.

Table 5-186 (Cont.) oc_ingressgateway_request_latency_seconds

Field	Details
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Histogram
Dimensions	- quantile - InstanceIdentifier - Route_path - Method

5.1.32.1.6 oc_ingressgateway_request_latency_seconds_count

Table 5-187 oc_ingressgateway_request_latency_seconds_count

Field	Details
Metric Details	This metric is used to display the number of ingress requests processed in a particular time span (in seconds).
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Histogram
Dimensions	- InstanceIdentifier - Route_path - Method

5.1.32.1.7 oc_ingressgateway_request_latency_seconds_sum

Table 5-188 oc_ingressgateway_request_latency_seconds_sum

Field	Details
Metric Details	This metrics is used to display the average of processing time of all the ingress request for a particular time.
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Histogram
Dimensions	- InstanceIdentifier - Route_path - Method

5.1.32.1.8 oc_configclient_request_total

Table 5-189 oc_configclient_request_total

Field	Details
Metric Details	This metric will be pegged whenever config client is polling for configuration update from common configuration server.
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Counter
Dimensions	- configVersion - releaseVersion

5.1.32.1.9 oc_configclient_response_total

Table 5-190 oc_configclient_response_total

Field	Details
Metric Details	This metrics will be pegged whenever config client receives response from common configuration server.
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Counter
Dimensions	• configVersion • releaseVersion • updated

5.1.32.1.10 oc_configserver_reachability

Table 5-191 oc_configserver_reachability

Field	Details
Metric Details	Gauge metric to peg the reachability of config server.
Microservice	alternate route, PLMN Ingress Gateway, N32 Ingress Gateway
Type	Gauge
Dimensions	NA

5.1.32.1.11 oc_ingressgateway_incoming_connections

Table 5-192 oc_ingressgateway_incoming_connections

Field	Details
Metric Details	Gauge metric that will peg active incoming connections from client to ingress gateway
Microservice	PLMN Ingress Gateway, N32 Ingress Gateway
Type	Gauge
Dimensions	• Host • Direction • InstanceIdentifier

5.1.32.1.12 oc_ingressgateway_outgoing_connections

Table 5-193 oc_ingressgateway_outgoing_connections

Field	Details
Metric Details	Gauge metric that will peg active outgoing connections from ingress gateway to destination
Microservice	PLMN Ingress Gateway, N32 Ingress Gateway
Type	Gauge

Table 5-193 (Cont.) oc_ingressgateway_outgoing_connections

Field	Details
Dimensions	• Host • Direction • InstanceIdentifier

5.1.32.1.13 oc_ingressgateway_connection_failure_total

Table 5-194 oc_ingressgateway_connection_failure_total

Field	Details
Metric Details	This metric will be pegged in the customized Jetty Client as soon as it fails to connect to the destination service with direction as ingressOut. Here in case of Ingress gateway, the destination service will be a backend microservice of the NF. And TLS connection failure metrics when connecting to ingress with direction as ingress.
Microservice	PLMN Ingress Gateway, N32 Ingress Gateway
Type	Counter
Dimensions	• Host • port • Direction • InstanceIdentifier • errorReason • errorOriginator Note: • The dimensions Host, port, and, errorReason are optional. The dimensions are only populated once the custom values are finalized. • addErrorReasonToConnectionFailureMetric to true metric must be true for the metric to be pegged with error reason and alert SEPPConnectionFailureN32IGWAlert to be generated.

5.1.32.1.14 oc_ingressgateway_global_ratelimit_total

Table 5-195 oc_ingressgateway_global_ratelimit_total

Field	Details
Metric Details	This metric will be pegged in the custom filter implemented to check the global rate limit conditions.
Microservice	PLMN Ingress Gateway, N32 Ingress Gateway
Type	Counter
Dimensions	• Method • Route_path • Scheme • InstanceIdentifier • Status

5.1.32.1.15 oc_ingressgateway_request_content_metrics_total

Table 5-196 oc_ingressgateway_request_content_metrics_total

Field	Details
Metric Details	This metric will be pegged by default filter RequestContentMetrics. It pegs whether request has request body or not.
Microservice	PLMN Egress Gateway, N32 Egress Gateway
Type	Counter
Dimensions	- method - content_available - InstanceIdentifier

5.1.32.1.16 oc_ingressgateway_request_processing_latency_seconds

Table 5-197 oc_ingressgateway_request_processing_latency_seconds

Field	Details
Metric Details	This metric will be pegged in the last custom filter of the Ingress gateway while the response is being sent back to the consumer NF. This metric captures the amount of time taken for processing of the request only within Ingress gateway. It starts as soon the request reaches the first custom filter of the application and lasts till the request is forwarded to the destination.
Microservice	PLMN Egress Gateway, N32 Egress Gateway
Type	Histogram
Dimensions	- quantile - InstanceIdentifier - Route_path - Method

5.1.32.1.17 oc_ingressgateway_incoming_ip_type

Table 5-198 oc_ingressgateway_incoming_ip_type

Field	Details
Metric Details	Pegs IP address type of the active incoming connections from the client to Ingress Gateway. Example: <pre>oc_ingressgateway_incoming_ip_type{Host="10.233.109.0", ReceivedAddressType="IPv4", } 0.0</pre>
Microservice	N32 Ingress Gateway, PLMN Ingress Gateway
Type	Gauge
Dimensions	- Host - ReceivedAddressType

5.1.32.1.18 oc_ingressgateway_outgoing_ip_type

Table 5-199 oc_ingressgateway_outgoing_ip_type

Field	Details
Metric Details	Pegs IP address type of the active outgoing connections from Ingress Gateway to the destination. Example: <code>oc_ingressgateway_outgoing_ip_type{BackendSvc="n32-egress-gateway", BackendSvcAddressType="IPv6",} 4.0</code>
Microservice	N32 Ingress Gateway, PLMN Ingress Gateway
Type	Gauge
Dimensions	- BackendSvc - BackendSvcAddressType

5.1.32.2 Egress Gateway Metrics

This section provides information about the Egress Gateway metrics used in SEPP.

5.1.32.2.1 oc_egressgateway_request_failure_total

Table 5-200 oc_egressgateway_request_failure_total

Field	Details
Metric Details	This metric will be pegged when the request is discarded due to late arrival at Egress Gateway.
Microservice	n32-egress-gateway, plmn-egress-gateway
Type	Counter
Dimensions	- ErrorDetail - ErrorOriginator - HttpStatus - InstanceIdentifier - Method - Scheme

5.1.32.2.2 oc_egressgateway_sbitimer_timezone_mismatch

Table 5-201 oc_egressgateway_sbitimer_timezone_mismatch

Field	Details
Metric Details	When the global.seppSbiTimerTimezone is set to ANY and timezone is not specified in the header then this metric is pegged at Egress Gateway.
Microservice	n32-egress-gateway, plmn-egress-gateway
Type	Gauge

Table 5-201 (Cont.) oc_egressgateway_sbitimer_timezone_mismatch

Field	Details
Dimensions	- Route path - Method - app - chart - container - method - namespace - path - pod - release

5.1.32.2.3 oc_egressgateway_http_requests_total

Table 5-202 oc_egressgateway_http_requests_total

Field	Details
Metric Details	This metric will be pegged as soon as the request reaches the Egress Gateway in the first custom filter of the application with direction as egress. This will also be pegged when the request goes out of egress in Jetty Request Listener with direction as Egress Out.
Microservice	Plmn Egress Gateway, N32 Egress Gateway
Type	Counter
Dimensions	- Method - NFType - NFServiceType - Host - HttpVersion - Scheme - Proxy - InstanceIdentifier - Direction

5.1.32.2.4 oc_egressgateway_http_responses_total

Table 5-203 oc_egressgateway_http_responses_total

Field	Details
Metric Details	This metric will be pegged in the last custom filter of the Egress Gateway while the response is being sent back to backend NF microservice with direction as egress. This will also be pegged when the response is fetched in Jetty responseListener with direction as egressOut. BlacklistedFqdn tag will be filled with BlacklistedFqdn when request is sent with blacklisted producer
Microservice	Plmn Egress Gateway, N32 Egress Gateway
Type	Counter

Table 5-203 (Cont.) oc_egressgateway_http_responses_total

Field	Details
Dimensions	• HttpVersion • Scheme • Direction • error_reason • ErrorOriginator

5.1.32.2.5 oc_egressgateway_incoming_connections

Table 5-204 oc_egressgateway_incoming_connections

Field	Details
Metric Details	Gauge metric that will peg active incoming connections from client to egress gateway
Microservice	PLMN Egress Gateway, N32 Egress Gateway
Type	Gauge
Dimensions	• Host • Direction • InstanceIdentifier

5.1.32.2.6 oc_egressgateway_outgoing_connections

Table 5-205 oc_egressgateway_outgoing_connections

Field	Details
Metric Details	Gauge metric that will peg active outgoing connections from egress gateway to destination
Microservice	PLMN Egress Gateway, N32 Egress Gateway
Type	Gauge
Dimensions	• Host • Direction • InstanceIdentifier

5.1.32.2.7 oc_egressgateway_connection_failure_total

Table 5-206 oc_egressgateway_connection_failure_total

Field	Details
Metric Details	This metric will be pegged by jetty client when the destination is not reachable by egress gateway. Here the destination is producer NF.
Microservice	PLMN Egress Gateway, N32 Egress Gateway
Type	Counter

Table 5-206 (Cont.) oc_egressgateway_connection_failure_total

Field	Details
Dimensions	• Host • port • Direction • InstanceIdentifier • errorReason • errorOriginator

Note

The dimensions Host, port, and errorReason are optional. The dimensions are only populated once the custom values are finalized.

5.1.32.2.8 oc_egressgateway_sbiRouting_http_requests_total

Table 5-207 oc_egressgateway_sbiRouting_http_requests_total

Field	Details
Metric Details	This metric is pegged in the SBIRoutingFilter only when SBIRouting feature is enabled for a route to which request is sent to Egress Gateway.
Microservice	Plmn Egress Gateway, N32 Egress Gateway
Type	Counter
Dimensions	• Sbi_Fqdn • Reroute_Path • Response_Code (This would be populated as blank for requests) • Attempt • HttpVersion • Scheme • InstanceIdentifier

5.1.32.2.9 oc_egressgateway_sbiRouting_http_responses_total

Table 5-208 oc_egressgateway_sbiRouting_http_responses_total

Field	Details
Metric Details	This metric will be pegged in the SBIRoutingFilter only when SBI Routing feature is enabled for a route to which request is sent to EGW and when sbiRerouteEnabled is set to true and reroute mechanism is executed.
Microservice	PLMN Egress Gateway, N32 Egress Gateway
Type	Counter

Table 5-208 (Cont.) oc_egressgateway_sbiRouting_http_responses_total

Field	Details
Dimensions	• Sbi_Fqdn • Reroute_Path • Status • Attempt • HttpVersion • Scheme • InstanceIdentifier • ErrorOriginator

5.1.32.2.10 oc_egressgateway_server_latency_seconds

Table 5-209 oc_egressgateway_server_latency_seconds

Field	Details
Metric Details	This metric will be pegged in Jetty response listener that captures the amount of time taken for processing of the request by jetty client.
Microservice	PLMN Egress Gateway, N32 Egress Gateway
Type	Histogram
Dimensions	• quantile • InstanceIdentifier • Method

5.1.32.2.11 oc_egressgateway_incoming_ip_type

Table 5-210 oc_egressgateway_incoming_ip_type

Field	Details
Metric Details	Pegs IP address type of the active incoming connections from the client to Egress Gateway. Example: <pre>oc_egressgateway_incoming_ip_type{Host="fd00:0:0:1:809e:4529:947c:bea8%0",ReceivedAddressType="IPv6",} 4.0</pre>
Microservice	N32 Egress Gateway, PLMN Egress Gateway
Type	Gauge
Dimensions	• Host • receivedAddressType

5.1.32.2.12 oc_egressgateway_outgoing_ip_type

Table 5-211 oc_egressgateway_outgoing_ip_type

Field	Details
Metric Details	Pegs IP address type of the active outgoing connections from Egress Gateway to the destination. Example: <pre>oc_egressgateway_outgoing_ip_type{DestinationHost="sepp-stub",DestinationHostAddressType="IPv6",} 4.0</pre>
Microservice	N32 Egress Gateway, PLMN Egress Gateway
Type	Gauge
Dimensions	• app • chart • container • namespace • pod • DestinationHost • DestinationHostAddressType

5.1.32.2.13 oc_egressgateway_dualstack_ip_rejected_total

Table 5-212 oc_egressgateway_dualstack_ip_rejected_total

Field	Details
Metric Details	Counts the total number of IP address rejections because the IP address type configured in the egressRoutingMode does not match the IP address type returned by DNS resolution. Example: <pre>oc_egressgateway_dualstack_ip_rejected_total{"service":"sepp-svc1", "egressRoutingMode":"IPv6", "dnsReceivedIpType":"IPv4"} 10.0</pre> <pre>oc_egressgateway_dualstack_ip_rejected_total{"service":"sepp-svc1", "egressRoutingMode":"IPv4", "dnsReceivedIpType":"IPv6"} 5.0</pre>
Microservice	N32 Egress Gateway, PLMN Egress Gateway
Type	Counter
Dimensions	• service • egressRoutingMode • dnsReceivedIpType

5.1.32.2.14 oc_egressgateway_global_ratelimit_dropped_message_total

Table 5-213 oc_egressgateway_global_ratelimit_dropped_message_total

Field	Details
Metric Details	This depicts the total requests dropped when the traffic exceeds configured rate limiting values.
Microservice	N32 Egress Gateway, PLMN Egress Gateway
Type	Counter
Dimensions	- Method - Scheme

5.2 SEPP KPIs

This section provides information about the SEPP KPIs.

5.2.1 N32C Handshake Procedure KPIs

5.2.1.1 cn32c Handshake Requests Per Remote SEPP

Table 5-214 cn32c Handshake Requests Per Remote SEPP

Field	Details
KPI Detail	Measures the cn32c handshake requests per remote SEPP.
Metric Used for KPI	"sum(ocsepp_cn32c_handshake_requests_total{namespace=~\"\$Namespace\"})by(peer_domain, peer_fqdn, peer_plmn_id , remote_sepp_name)"
Service Operation	n32c Handshake Request
Response Code	NA

5.2.1.2 cn32c Handshake Success Rate

Table 5-215 cn32c Handshake Success Rate

Field	Details
KPI Detail	Measures the cn32c handshake success rate.
Metric Used for KPI	(sum(ocsepp_cn32c_handshake_response_total{namespace=~\"\$Namespace\",http_status=\"200 OK\"})/sum(ocsepp_cn32c_handshake_requests_total{namespace=~\"\$Namespace\"}))*100
Service Operation	n32c handshake success rate
Response Code	200 OK

5.2.1.3 cn32c Handshake Response Per Remote SEPP

Table 5-216 cn32c Handshake Response Per Remote SEPP

Field	Details
KPI Detail	Measures the cn32c handshake response per remote SEPP.
Metric Used for KPI	"sum(ocsepp_cn32c_handshake_response_total{namespace=~\"\$Namespace\"})by(peer_domain, peer_fqdn, peer_plmn_id, remote_sepp_name)"
Service Operation	n32c Handshake
Response Code	All

5.2.1.4 cn32c Handshake Failure Per Remote SEPP

Table 5-217 cn32c Handshake Failure Per Remote SEPP

Field	Details
KPI Detail	Measures the cn32c handshake failure per remote SEPP
Metric Used for KPI	"sum(ocsepp_n32c_handshake_failure_attempts_total{namespace=~\"\$Namespace\",app=\"cn32c-svc\"})by(peer_domain, peer_fqdn, peer_plmn_id, remote_sepp_name)"
Service Operation	n32c Handshake
Response Code	4xx and 5xx

5.2.1.5 pn32c Handshake Requests Total Per Remote SEPP

Table 5-218 pn32c Handshake Requests Total Per Remote SEPP

Field	Details
KPI Detail	Measures the pn32c handshake requests total per remote SEPP
Metric Used for KPI	"sum(ocsepp_pn32c_handshake_requests_total{namespace=~\"\$Namespace\"})by(peer_domain, peer_fqdn, peer_plmn_id, remote_sepp_name)"
Service Operation	n32c Handshake
Response Code	All

5.2.1.6 pn32c Handshake Response Total Per Remote SEPP

Table 5-219 pn32c Handshake Response Total Per Remote SEPP

Field	Details
KPI Detail	Measures the pn32c handshake response total per remote SEPP
Metric Used for KPI	"sum(ocsepp_pn32c_handshake_response_total{namespace=~\"\$Namespace\"})by(peer_domain, peer_fqdn, peer_plmn_id, remote_sepp_name)"
Service Operation	n32c Handshake
Response Code	All

5.2.1.7 pn32c Handshake Success rate

Table 5-220 pn32c Handshake Success rate

Field	Details
KPI Detail	Measures the pn32c handshake success rate.
Metric Used for KPI	(sum(ocsepp_pn32c_handshake_response_total{namespace=~"\$Namespace",http_status="200 OK"})/sum(ocsepp_pn32c_handshake_requests_total{namespace=~"\$Namespace"}))*100
Service Operation	n32c Handshake
Response Code	200

5.2.1.8 pn32c Handshake Failure Per Remote SEPP

Table 5-221 pn32c Handshake Failure Per Remote SEPP

Field	Details
KPI Detail	Measures the pn32c handshake failure total
Metric Used for KPI	sum(ocsepp_n32c_handshake_failure_attempts_total{namespace=~"\$Namespace",app="pn32c-svc"})by(peer_domain, peer_fqdn, peer_plmn_id)
Service Operation	n32c Handshake
Response Code	4xx and 5xx

5.2.2 SEPP Common KPIs

5.2.2.1 Memory Usage per POD

Table 5-222 Memory Usage per POD

Field	Details
KPI Detail	Measures the memory usage per POD
Metric Used for KPI	sum(container_memory_usage_bytes{namespace=~"\$Namespace",image!=""}) by (pod)/(1024*1024*1024)
Service Operation	NA
Response Code	NA

5.2.2.2 CPU Usage per POD

Table 5-223 CPU Usage per POD

Field	Details
KPI Detail	Measures the CPU usage per POD

Table 5-223 (Cont.) CPU Usage per POD

Field	Details
Metric Used for KPI	sum(rate(container_cpu_usage_seconds_total{namespace=~"\$Namespace",image!=""}[2m])) by (pod) * 1000
Service Operation	N/A
Response Code	N/A

5.2.2.3 Total Ingress gateway requests

Table 5-224 Total Ingress gateway requests

Field	Details
KPI Detail	Measures the total Ingress gateway requests
Metric Used for KPI	sum((oc_ingressgateway_http_requests_total{namespace=~"\$Namespace"}))by(app,HttpVersion,Direction)

5.2.2.4 Total Egress gateway requests

Table 5-225 Total Egress gateway requests

Field	Details
KPI Detail	Measures the total egress gateway requests
Metric Used for KPI	sum((oc_egressgateway_http_requests_total{namespace=~"\$Namespace"}))by(app,HttpVersion,Direction)

5.2.2.5 Total Ingress gateway responses

Table 5-226 Total Ingress gateway responses

Field	Details
KPI Detail	Measures the total Ingress gateway responses
Metric Used for KPI	sum((oc_ingressgateway_http_responses_total{namespace=~"\$Namespace"}))by(app,HttpVersion,Direction)

5.2.2.6 Total Egress gateway responses

Table 5-227 Total Egress gateway responses

Field	Details
KPI Detail	Measures the total Egress gateway responses
Metric Used for KPI	sum((oc_egressgateway_http_responses_total{namespace=~"\$Namespace"}))by(app,HttpVersion,Direction)

5.2.2.7 IGW Processing Time (ms)

Table 5-228 IGW Processing Time (ms)

Field	Details
KPI Detail	Measures the IGW Processing Time
Metric Used for KPI	<code>sum(irate(oc_ingressgateway_request_latency_seconds_sum{namespace=~"\$Namespace"}[2m])) by(Method,app) / sum(irate(oc_ingressgateway_request_latency_seconds_count{namespace=~"\$Namespace"}[2m])) by(Method,app)</code>

5.2.2.8 PercentageDiscard

Table 5-229 PercentageDiscard

Field	Details
KPI Detail	Measures the number of Discard requests for Percentage based scheme
Metric Used for KPI	<code>oc_ingressgateway_route_overloadcontrol_discard_total sum(irate(oc_ingressgateway_route_overloadcontrol_discard_total{DiscardAction="PercentageBased",Status="DISCARDED",namespace="\$Namespace"}[2m]))</code>

5.2.2.9 PriorityDiscard

Table 5-230 PriorityDiscard

Field	Details
KPI Detail	Measures the number of Discard requests for Priority based scheme
Metric Used for KPI	<code>oc_ingressgateway_route_overloadcontrol_discard_total sum(irate(oc_ingressgateway_route_overloadcontrol_discard_total{DiscardAction="PriorityBased",Status="DISCARDED",namespace="\$Namespace"}[2m]))</code>

5.2.3 CN32F Common KPIs

5.2.3.1 cn32f Routing Success Rate

Table 5-231 cn32f Routing Success Rate

Field	Details
KPI Detail	Measures the cn32f routing success rate.

Table 5-231 (Cont.) cn32f Routing Success Rate

Field	Details
Metric Used for KPI	$\frac{\text{sum(ocsepp_cn32f\_response\_total\{namespace=\sim\$Namespace\}})}{\text{sum(ocsepp_cn32f\_requests\_total\{namespace=\sim\$Namespace\}})} * 100$
Service Operation	n32f message forward
Response Code	All

5.2.3.2 Total cn32f Requests

Table 5-232 Total cn32f Requests

Field	Details
KPI Detail	Measures the cn32f requests rate per remote SEPP.
Metric Used for KPI	$\text{sum}((\text{ocsepp\_cn32f\_requests\_total\{namespace=\sim\$Namespace\}, \text{direction="egress"}})) \text{ by (PEER\_DOMAIN, PEER\_FQDN, PLMN\_ID)}$
Service Operation	n32f message forward
Response Code	All

5.2.3.3 cn32f Processing Time (ms)

Table 5-233 cn32f Processing Time (ms)

Field	Details
KPI Detail	Measures the cn32f processing time (ms)
Metric Used for KPI	$\frac{\text{sum(irate(ocsepp\_cn32f\_latency\_seconds\_sum\{namespace=\sim\$Namespace\}\{2m\}) \text{ by (peer\_fqdn, remote\_sepp\_name)}}}{\text{sum(irate(ocsepp\_cn32f\_latency\_seconds\_count\{namespace=\sim\$Namespace\}\{2m\}) \text{ by (peer\_fqdn, remote\_sepp\_name)}}$
Service Operation	n32f message forward
Response Code	All

5.2.3.4 Total cn32f Responses

Table 5-234 Total cn32f Responses

Field	Details
KPI Detail	Measures the cn32f response rate per remote SEPP
Metric Used for KPI	$\text{sum}((\text{ocsepp\_cn32f\_response\_total\{namespace=\sim\$Namespace\}, \text{direction="egress"}})) \text{ by (PEER\_DOMAIN, PEER\_FQDN, PLMN\_ID)}$
Service Operation	n32f message forward
Response Code	All

5.2.3.5 cn32f Failures

Table 5-235 cn32f Failures

Field	Details
KPI Detail	Measures the total cn32f request failures.
Metric Used for KPI	sum(ocsepp_cn32f_requests_failure_total{namespace=~"\$Namespace"}) by (PEER_DOMAIN, PEER_FQDN, PLMN_ID, statusCode)
Service Operation	n32f message forward
Response Code	5xxx

5.2.4 PN32F Common KPIs

5.2.4.1 Total pn32f Requests

Table 5-236 Total pn32f Requests

Field	Details
KPI Detail	Measures the total pn32f requests
Metric Used for KPI	sum((ocsepp_pn32f_requests_total{namespace=~"\$Namespace", direction="egress"}))by(PEER_DOMAIN, PEER_FQDN, PLMN_ID)
Service Operation	n32f message forward
Response Code	All

5.2.4.2 Total pn32f Responses

Table 5-237 Total pn32f Responses

Field	Details
KPI Detail	Measures the pn32f response rate per remote SEPP.
Metric Used for KPI	sum((ocsepp_pn32f_responses_total{namespace=~"\$Namespace", direction="egress"})) by(PEER_DOMAIN, PEER_FQDN, PLMN_ID)
Service Operation	n32f message forward
Response Code	All

5.2.4.3 pn32f Processing Time (ms)

Table 5-238 pn32f Processing Time (ms)

Field	Details
KPI Detail	Measures the pn32f processing time in milli seconds

Table 5-238 (Cont.) pn32f Processing Time (ms)

Field	Details
Metric Used for KPI	<code>sum(irate(ocsepp_pn32f_latency_seconds_sum{namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name) / sum(irate(ocsepp_pn32f_latency_seconds_count{namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name)</code>
Service Operation	n32f message forward
Response Code	All

5.2.4.4 pn32f Failures

Table 5-239 pn32f Failures

Field	Details
KPI Detail	Measures the pn32f request failures in total
Metric Used for KPI	<code>sum(ocsepp_pn32f_requests_failure_total{namespace=~"\$Namespace"}) by (PEER_DOMAIN, PEER_FQDN, PLMN_ID)</code>
Service Operation	n32f message forward
Response Code	4xx and 5xx

5.2.4.5 pn32f Routing Success Rate

Table 5-240 pn32f Routing Success Rate

Field	Details
KPI Detail	Measures the pn32f routing success rate
Metric Used for KPI	<code>(sum(ocsepp_pn32f_responses_total{namespace=~"\$Namespace"}) / sum(ocsepp_pn32f_requests_total{namespace=~"\$Namespace"})) * 100</code>
Service Operation	n32f message forward
Response Code	All

5.2.5 Global Rate Limiting Feature KPIs

5.2.5.1 PLMN IGW Global Rate limit Traffic Rejected

Table 5-241 PLMN IGW Global Rate limit Traffic Rejected

Field	Details
KPI Detail	Measures the PLMN IGW Global rate limit traffic rejected
Metric Used for KPI	<code>sum(irate(oc_ingressgateway_global_ratelimit_total{namespace=~"\$Namespace",app="plmn-ingress-gateway",Status="dropped"}[2m]))</code> No. of messages rejected for traffic initiated from producer side

5.2.5.2 N32 IGW Global Rate limit Traffic Rejected

Table 5-242 N32 IGW Global Rate limit Traffic Rejected

Field	Details
KPI Detail	Measures the N32 IGW Global rate limit traffic rejected
Metric Used for KPI	sum(irate(oc_ingressgateway_global_ratelimit_total{namespace=~"\$Namespace",app="n32-ingress-gateway", Status="dropped"}[2m])) No. of messages rejected for traffic initiated from consumer side

5.2.6 Topology Hiding KPIs

5.2.6.1 CN32F Topology Egress Request Processing Time(ms)

Table 5-243 CN32F Topology Egress Request Processing Time(ms)

Field	Details
KPI Detail	Measures the cn32f topology Egress request processing time
Metric Used for KPI	sum(irate(ocsepp_topology_latency_seconds_sum{app="cn32f-svc",direction="egress",message_type="request",namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name) / sum(irate(ocsepp_topology_latency_seconds_count{app="cn32f-svc",direction="egress",message_type="request",namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name)
Service Operation	n32f message forward
Response Code	NA

5.2.6.2 CN32F Topology Ingress Response Processing Time

Table 5-244 CN32F Topology Ingress Response Processing Time

Field	Details
KPI Detail	Measures the cn32f topology Egress response processing time
Metric Used for KPI	sum(irate(ocsepp_topology_latency_seconds_sum{app="cn32f-svc",direction="ingress",message_type="response",namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name) / sum(irate(ocsepp_topology_latency_seconds_count{app="cn32f-svc",direction="ingress",message_type="response",namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name)
Service Operation	n32f message forward
Response Code	NA

5.2.6.3 CN32F Topology Hiding Success

Table 5-245 CN32F Topology Hiding Success

Field	Details
KPI Detail	Measures the n32f topology success by messages
Metric Used for KPI	sum(ocsepp_topology_success_total{app="cn32f-svc", namespace=~"\$Namespace"})
Service Operation	n32f message forward
Response Code	NA

5.2.6.4 N32F Topology Success by headers

Table 5-246 N32F Topology Success by headers

Field	Details
KPI Detail	Measures N32F Topology success by headers
Metric Used for KPI	sum(ocsepp_topology_header_success_total{app="cn32f-svc"}) by(header) Note: Update label app to "app=pn32f-svc" for PN32F microservice.
Service Operation	n32f message forward
Response Code	NA

5.2.6.5 CN32F Topology Hiding Missing Regex Configuration

Table 5-247 CN32F Topology Hiding Missing Regex Configuration

Field	Details
KPI Detail	Measures the cn32f topology hiding missing regex configuration
Metric Used for KPI	sum(ocsepp_topology_header_regex_not_configured_total{app="cn32f-svc", namespace=~"\$Namespace"}) by(error_msg)
Service Operation	n32f message forward
Response Code	NA

5.2.6.6 CN32F Topology Hiding Invalid Header Regex

Table 5-248 CN32F Topology Hiding Invalid Header Regex

Field	Details
KPI Detail	Measures the cn32f topology hiding invalid header regex
Metric Used for KPI	sum(ocsepp_topology_invalid_header_regex_configured_total{app="cn32f-svc", namespace=~"\$Namespace"}) by(error_msg)
Service Operation	n32f message forward
Response Code	NA

5.2.6.7 PN32F Topology Ingress Request Processing Time(ms)

Table 5-249 PN32F Topology Ingress Request Processing Time(ms)

Field	Details
KPI Detail	Measures the pn32f topology Ingress request processing time
Metric Used for KPI	sum(irate(ocsepp_topology_latency_seconds_sum{app="pn32f-svc",direction="ingress",message_type="request",namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name) / sum(irate(ocsepp_topology_latency_seconds_count{app="pn32f-svc",direction="ingress",message_type="request",namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name)
Service Operation	n32f message forward
Response Code	NA

5.2.6.8 PN32F Topology Egress Response Processing Time(ms)

Table 5-250 PN32F Topology Egress Response Processing Time(ms)

Field	Details
KPI Detail	Measures the pn32f topology Egress response processing time
Metric Used for KPI	sum(irate(ocsepp_topology_latency_seconds_sum{app="pn32f-svc",direction="egress",message_type="response",namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name) / sum(irate(ocsepp_topology_latency_seconds_count{app="pn32f-svc",direction="egress",message_type="response",namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name)
Service Operation	n32f message forward
Response Code	NA

5.2.6.9 PN32F Topology Hiding Invalid Header Regex

Table 5-251 PN32F Topology Hiding Invalid Header Regex

Field	Details
KPI Detail	Measures the pn32f topology hiding invalid header Regex
Metric Used for KPI	sum(ocsepp_topology_invalid_header_regex_configured_total{app="pn32f-svc", namespace=~"\$Namespace"}) by(error_msg)
Service Operation	n32f message forward
Response Code	NA

5.2.6.10 PN32F Topology Hiding Missing Regex Configuration

Table 5-252 PN32F Topology Hiding Missing Regex Configuration

Field	Details
KPI Detail	Measures the pn32f topology hiding missing Regex configuration

Table 5-252 (Cont.) PN32F Topology Hiding Missing Regex Configuration

Field	Details
Metric Used for KPI	sum(ocsepp_topology_header_regex_not_configured_total{app="pn32f-svc", namespace=~"\$Namespace"}) by(error_msg)
Service Operation	n32f message forward
Response Code	NA

5.2.6.11 PN32F Topology Hiding Success

Table 5-253 PN32F Topology Hiding Success

Field	Details
KPI Detail	Measures the pn32f topology hiding success
Metric Used for KPI	sum(ocsepp_topology_success_total{app="pn32f-svc", namespace=~"\$Namespace"})
Service Operation	n32f message forward
Response Code	NA

5.2.7 5G SBI Message Mediation Support KPIs

5.2.7.1 Mediation Requests Counters - N32 Egress Request

Table 5-254 Mediation Requests Counters - N32 Egress Request

Field	Details
KPI Detail	Measures the Mediation Requests Counters for N32 Egress Request
Metric Used for KPI	sum(ocsepp_n32f_mediation_requests_total{direction="N32_Egress_Request", namespace=~"\$Namespace"})

5.2.7.2 Mediation Requests Counters - N32 Ingress Response

Table 5-255 Mediation Requests Counters - N32 Ingress Response

Field	Details
KPI Detail	Measures the Mediation Requests Counters for N32 Ingress Response
Metric Used for KPI	sum(ocsepp_n32f_mediation_requests_total{direction="N32_Ingress_Response", namespace=~"\$Namespace"})

5.2.7.3 Mediation Requests Counters - N32 Ingress Request

Table 5-256 Mediation Requests Counters - N32 Ingress Request

Field	Details
KPI Detail	Measures the Mediation Requests Counters for N32 Ingress Request
Metric Used for KPI	sum(ocsepp_n32f_mediation_requests_total{direction="N32_Ingress_Request", namespace=~"\$Namespace"})

5.2.7.4 Mediation Requests Counters - N32 Egress Response

Table 5-257 Mediation Requests Counters - N32 Egress Response

Field	Details
KPI Detail	Measures the Mediation Requests Counters for N32 Egress Response
Metric Used for KPI	sum(ocsepp_n32f_mediation_requests_total{direction="N32_Egress_Response", namespace=~"\$Namespace"})

5.2.7.5 Mediation Response Counters - N32 Egress Request

Table 5-258 Mediation Response Counters - N32 Egress Request

Field	Details
KPI Detail	Measures the Mediation Response Counters for N32 Egress Request
Metric Used for KPI	sum(ocsepp_n32f_mediation_response_total{direction="N32_Egress_Request", namespace=~"\$Namespace"})

5.2.7.6 Mediation Response Counters - N32 Ingress Response

Table 5-259 Mediation Response Counters - N32 Ingress Response

Field	Details
KPI Detail	Measures the Mediation Response Counters for N32 Ingress Response
Metric Used for KPI	sum(ocsepp_cn32f_mediation_response_total{direction="N32_Ingress_Response", namespace=~"\$Namespace"})

5.2.7.7 Mediation Response Counters - N32 Ingress Request

Table 5-260 Mediation Response Counters - N32 Ingress Request

Field	Details
KPI Detail	Measures the Mediation Response Counters for N32 Ingress Request

Table 5-260 (Cont.) Mediation Response Counters - N32 Ingress Request

Field	Details
Metric Used for KPI	sum(ocsepp_pn32f_mediation_response_total{direction="N32_Ingress_Request",namespace=~"\$Namespace"})

5.2.7.8 Mediation Response Counters - N32 Egress Response

Table 5-261 Mediation Response Counters - N32 Egress Response

Field	Details
KPI Detail	Measures the Mediation Response Counters for N32 Egress Response
Metric Used for KPI	sum(ocsepp_pn32f_mediation_response_total{direction="N32_Egress_Response",namespace=~"\$Namespace"})

5.2.7.9 Mediation Response Failure

Table 5-262 Mediation Response Failure

Field	Details
KPI Detail	Measures the Mediation Response Failure
Metric Used for KPI	sum(ocsepp_cn32f_mediation_response_failure{namespace=~"\$Namespace"}) by (Direction, status_code) sum(ocsepp_pn32f_mediation_response_failure{namespace=~"\$Namespace"}) by (Direction, status_code)

5.2.7.10 Mediation Applied Total

Table 5-263 Mediation Applied Total

Field	Details
KPI Detail	Measures the Mediation Applied Total
Metric Used for KPI	(sum(ocsepp_n32f_mediation_requests_total{namespace=~"\$Namespace"})*100)/ (sum(ocsepp_n32f_mediation_not_applied_total{namespace=~"\$Namespace"})) +sum(ocsepp_n32f_mediation_requests_total{namespace=~"\$Namespace"})

5.2.7.11 Mediation Response Time At PN32F

Table 5-264 Mediation Response Time At PN32F

Field	Details
KPI Detail	Measures the Mediation Response Time at PN32F

Table 5-264 (Cont.) Mediation Response Time At PN32F

Field	Details
Metric Used for KPI	<code>sum(irate(ocsepp_pn32f_mediation_latency_seconds_sum{namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name) / sum(irate(ocsepp_pn32f_mediation_latency_seconds_count{namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name)</code>

5.2.7.12 Mediation Response Time At CN32F

Table 5-265 Mediation Response Time At CN32F

Field	Details
KPI Detail	Measures the Mediation Response Time at CN32F
Metric Used for KPI	<code>sum(irate(ocsepp_cn32f_mediation_latency_seconds_sum{namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name) / sum(irate(ocsepp_cn32f_mediation_latency_seconds_count{namespace=~"\$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name)</code>

5.2.8 Ingress Gateway Message Copy KPIs

5.2.8.1 Total Requests Data sent towards DD for Ingress Gateway

Table 5-266 Total Requests Data sent towards DD for Ingress Gateway

Field	Details
KPI Detail	Measures the total requests data sent towards DD for Ingress Gateway.
Metric Used for KPI	<code>sum(irate(oc_egressgateway_msgcopy_requests_total{namespace=~"\$Namespace",type="req"}[2m])) by(app)</code>

5.2.8.2 Total Ack received from DD for Requests for Ingress Gateway

Table 5-267 Total Ack received from DD for Requests for Ingress Gateway

Field	Details
KPI Detail	Measures the total Ack received from DD for requests for Ingress Gateway.
Metric Used for KPI	<code>sum(irate(oc_egressgateway_msgcopy_requests_total{namespace=~"\$Namespace",type="ack"}[2m])) by(app)</code>

5.2.9 Egress Gateway Message Copy KPIs

5.2.9.1 Total Requests Data sent towards DD for Egress Gateway

Table 5-268 Total Requests Data sent towards DD for Egress Gateway

Field	Details
KPI Detail	Measures the total Requests Data sent towards DD for Egress Gateway.
Metric Used for KPI	<code>sum(irate(oc_egressgateway_msgcopy_requests_total{namespace="\$Namespace",type="req"}[2m])) by(app)</code>

5.2.9.2 Total Ack received from DD for Requests for Egress Gateway

Table 5-269 Total Ack received from DD for Requests for Egress Gateway

Field	Details
KPI Detail	Measures the total acknowledgement received from DD on Egress Gateway.
Metric Used for KPI	<code>sum(irate(oc_egressgateway_msgcopy_requests_total{namespace="\$Namespace",type="ack"}[2m])) by(app)</code>

5.2.10 Hosted SEPP KPIs

5.2.10.1 CN32F Allowed P-RSS Validation Failure Count

Table 5-270 CN32F Allowed P-RSS Validation Failure Count

Field	Details
KPI Detail	Measures the number of messages failed due to incorrect routing rules configured at cn32f microservice.
Metric Used for KPI	<code>sum(ocsepp_allowed_p_rss_routing_failure_total{app="cn32f-svc", namespace=~"\$Namespace"}) by (app)</code>

5.2.10.2 PN32F Allowed P-RSS Validation Failure Count

Table 5-271 PN32F Allowed P-RSS Validation Failure Count

Field	Details
KPI Detail	Measures the number of messages failed due to incorrect routing rules configured at pn32f microservice
Metric Used for KPI	<code>sum(ocsepp_allowed_p_rss_routing_failure_total{app="pn32f-svc", namespace=~"\$Namespace"}) by (app)</code>

5.2.11 SoR KPIs

5.2.11.1 Pn32f to SoR Request count total

Table 5-272 Pn32f to SoR Request count total

Field	Details
KPI Detail	Number of messages sent to SOR from SEPP
Metric Used for KPI	sum(ocsepp_pn32f_sor_requests_total{direction="egress", namespace=~"\$Namespace"})

5.2.11.2 SoR to Pn32f Response count total

Table 5-273 SoR to Pn32f Response count total

Field	Details
KPI Detail	Number of responses received from SOR to SEPP
Metric Used for KPI	sum(ocsepp_pn32f_sor_responses_total{direction="ingress", namespace=~"\$Namespace"})

5.2.12 Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set KPIs

5.2.12.1 Average No of messages discarded for a particular RSS

Table 5-274 Average No of messages discarded for a particular RSS

Field	Details
KPI Detail	Measures the average number of messages discarded for a particular RSS.
Metric used for KPI	sum(irate(oc_ingressgateway_rss_ratelimit_total{namespace="namespace",Remote_SEPP_Set="<Remote SEPP Set name>", Status="dropped"}[2m]))

5.2.12.2 Average No of messages accepted for a particular RSS

Table 5-275 Average No of messages accepted for a particular RSS

Field	Details
KPI Detail	Measures the average number of messages accepted for a particular RSS.
Metric used for KPI	sum(irate(oc_ingressgateway_rss_ratelimit_total{namespace="namespace",Remote_SEPP_Set="<Remote SEPP Set name>", Status="accepted"}[2m]))

5.2.12.3 Average No of messages for which feature not applied

Table 5-276 Average No of messages for which feature not applied

Field	Details
KPI Detail	Measures the average number of messages for which feature not applied.
Metric used for KPI	sum(irate(oc_ingressgateway_rss_ratelimit_total{namespace="namespace",Status="ratelimit not applied"}[2m]))

5.2.12.4 Average of all messages by Status

Table 5-277 Average of all messages by Status

Field	Details
KPI Detail	Measures the average of all messages by Status
Metric used for KPI	sum(irate(oc_ingressgateway_rss_ratelimit_total{namespace="namespace"}[2m])) by (Status)

5.2.12.5 List of Average number of messages dropped for all RSS

Table 5-278 List of Average number of messages dropped for all RSS

Field	Details
KPI Detail	Lists the average number of messages dropped for all RSS
Metric used for KPI	sum(irate(oc_ingressgateway_rss_ratelimit_total{namespace="namespace", Status="dropped"}[2m])) by (Remote_SEPP_Set)

5.2.12.6 List of Average number of messages accepted for all RSS

Table 5-279 List of Average number of messages accepted for all RSS

Field	Details
KPI Detail	Lists the average number of messages accepted for all RSS
Metric used for KPI	sum(irate(oc_ingressgateway_rss_ratelimit_total{namespace="namespace", Status="accepted"}[2m])) by (Remote_SEPP_Set)

5.2.13 Cat 0 - SBI Message Schema Validation KPIs

5.2.13.1 Message validation applied requests on cn32f

Table 5-280 Message validation applied requests on cn32f

Field	Details
KPI Detail	Measures the total number of requests at CN32F on which message validation has been applied by request path.
Metric Used for KPI	sum(ocsepp_message_validation_applied_total{namespace=~"\$Namespace",app="cn32f-svc"}) by (requestPath)

5.2.13.2 Cn32f message validation failure on request body

Table 5-281 Cn32f message validation failure on request body

Field	Details
KPI Detail	Measures the total number of message validation failure(s) on request body by request path.
Metric Used for KPI	sum(ocsepp_message_validation_on_body_failure_total{namespace=~"\$Namespace",app="cn32f-svc"}) by (request_path)

5.2.13.3 Cn32f message validation failures on request query parameter(s)

Table 5-282 Cn32f message validation failures on request query parameter(s)

Field	Details
KPI Detail	Measures the total number of message validation failures on request query parameter(s) by request path.
Metric Used for KPI	sum(ocsepp_message_validation_on_header_failure_total{namespace=~"\$Namespace",app="cn32f-svc"}) by (request_path)

5.2.13.4 Message validation applied requests on pn32f

Table 5-283 Message validation applied requests on pn32f

Field	Details
KPI Detail	Measures the total number of requests at pn32f on which message validation has been applied by request path.
Metric Used for KPI	sum(ocsepp_message_validation_applied_total{namespace=~"\$Namespace",app="pn32f-svc"}) by (requestPath)

5.2.13.5 Pn32f message validation failure on request body

Table 5-284 Pn32f message validation failure on request body

Field	Details
KPI Detail	Measures the total number of message validation failure(s) on request body by request path.
Metric Used for KPI	sum(ocsepp_message_validation_on_body_failure_total{namespace=~"\$Namespace",app="pn32f-svc"}) by (request_path)

5.2.13.6 Pn32f message validation failures on request query parameter(s)

Table 5-285 Pn32f message validation failures on request query parameter(s)

Field	Details
KPI Detail	Measures the total number of message validation failures on request query parameter(s) by request path.
Metric Used for KPI	sum(ocsepp_message_validation_on_header_failure_total{namespace=~"\$Namespace",app="pn32f-svc"}) by (request_path)

5.2.14 Rate Limiting for Egress Roaming Signaling per PLMN KPIs

5.2.14.1 Average Number of Messages Rejected for a Particular PLMN

Table 5-286 Average Number of Messages Rejected for a Particular PLMN

Field	Details
KPI Detail	Measures the average number of messages rejected for a particular PLMN
Metric used for KPI	sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total{namespace="namespace",PLMN_ID="PLMN ID",Status="ERL_MATCH_NO_TOKEN_LOW_PRI_REJECT"}[2m]))

5.2.14.2 Average Number of Messages Accepted for a Particular PLMN

Table 5-287 Average Number of Messages Accepted for a Particular PLMN

Field	Details
KPI Detail	Measures the average number of messages accepted for a particular PLMN

Table 5-287 (Cont.) Average Number of Messages Accepted for a Particular PLMN

Field	Details
Metric used for KPI	sum(rate(oc_ingressgateway_plmn_egress_ratelim it_total{namespace="namespace",PLMN_ID="PLM N ID", Status=~"ERL_MATCH_TOKEN_AVAILABLE_FWD ERL_MATCH_NO_TOKEN_HIGH_PRI_FWD"} [2m]))

5.2.14.3 Average Number of Messages for which Feature not Applied

Table 5-288 Average Number of Messages for which Feature not Applied

Field	Details
KPI Detail	Measures the average number of messages for which feature not applied
Metric used for KPI	sum(rate(oc_ingressgateway_plmn_egress_ratelim it_total{namespace="namespace",Status! ~"ERL_MATCH_TOKEN_AVAILABLE_FWD ERL_MATCH_NO_TOKEN_HIGH_PRI_FWD ERL_MATCH_NO_TOKEN_LOW_PRI_REJECT"} [2m]))

5.2.14.4 Average of all Messages by Status

Table 5-289 Average of all Messages by Status

Field	Details
KPI Detail	Measures the average of all messages by status
Metric used for KPI	sum(rate(oc_ingressgateway_plmn_egress_ratelim it_total{namespace="namespace"}[2m])) by (Status)

5.2.14.5 Average Number of Messages Rejected per PLMN

Table 5-290 Average Number of Messages Rejected per PLMN

Field	Details
KPI Detail	Measures the average number of messages rejected per PLMN
Metric used for KPI	sum(rate(oc_ingressgateway_plmn_egress_ratelim it_total{namespace="namespace", Status="ERL_MATCH_NO_TOKEN_LOW_PRI_RE JECT"}[2m])) by (PLMN_ID)

5.2.14.6 Average Number of Messages Accepted per PLMN

Table 5-291 Average Number of Messages Accepted per PLMN

Field	Details
KPI Detail	Measures the average number of messages accepted per PLMN
Metric used for KPI	sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total{namespace="namespace", Status=~"ERL_MATCH_TOKEN_AVAILABLE_FW D ERL_MATCH_NO_TOKEN_HIGH_PRI_FWD"}[2m])) by (PLMN_ID)

5.2.15 Integrating SEPP with 5G Network Intelligence Fabric (5G NIF) Feature KPIs

5.2.15.1 Discovery Requests Sent Towards NRF for NIF

Table 5-292 Discovery Requests Sent Towards NRF for NIF

Field	Details
KPI Detail	sum(irate(ocsepp_nif_requests_total{namespace="seppsvc"}[2m]))
Metric Used for KPI	ocsepp_nif_requests_total
Service Operation	Discovery requests sent towards NRF for NIF.
Response Code	NA

5.2.15.2 Response Received from NRF for NIF Discovery

Table 5-293 Response Received from NRF for NIF Discovery

Field	Details
KPI Detail	sum(irate(ocsepp_nif_responses_total{namespace="seppsvc"}[2m]))
Metric Used for KPI	ocsepp_nif_responses_total
Service Operation	Total Response received from NRF for NIF Discovery.
Response Code	NA

5.2.15.3 Rejected Message Copied towards NIF

Table 5-294 Rejected Message Copied towards NIF

Field	Details
KPI Detail	sum(irate(ocsepp_pn32f_nif_error_copy_requests_total{namespace="seppsvc"}[2m]))
Metric Used for KPI	ocsepp_pn32f_nif_error_copy_requests_total

Table 5-294 (Cont.) Rejected Message Copied towards NIF

Field	Details
Service Operation	Total messages copied towards NIF.
Response Code	NA

5.2.15.4 Responses Received from NIF for Copied Messages

Table 5-295 Responses Received from NIF for Copied Messages

Field	Details
KPI Detail	sum(irate(ocsepp_pn32f_nif_error_copy_responses_total{namespace="seppsvc"}[2m])) by (status_code)
Metric Used for KPI	ocsepp_pn32f_nif_error_copy_responses_total
Service Operation	Responses received from NIF for copied messages.
Response Code	NA

5.2.16 Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers KPIs

5.2.16.1 Total number of Get_ACCESS_TOKEN messages received from a particular SEPP

Table 5-296 Total number of Get_ACCESS_TOKEN messages received from a particular SEPP

Field	Details
KPI Detail	Total number of Get_ACCESS_TOKEN messages received from a particular SEPP.
Metric Used for KPI	ocsepp_oauth_total_requests_total Expression: ocsepp_oauth_total_requests_total{namespace="sepp-2",remote_sepp_name="sepp1"}, andKPI for a duration -> sum(irate(ocsepp_oauth_total_requests_total{namespace="sepp-2",remote_sepp_name="sepp1"}[2m]))
Service Operation	NA
Response Code	NA

5.2.16.2 Number of requests without the requesterPLMN/requesterPLMNList attribute received from a particular SEPP

Table 5-297 Number of requests without the requesterPLMN/requesterPLMNList attribute received from a particular SEPP

Field	Details
KPI Detail	Number of requests without the requesterPLMN/requesterPLMNList attribute received from a particular SEPP to which either the PLMN-ID was inferred or responded with an error (based on configuration).
Metric Used for KPI	ocsepp_network_id_validation_body_failure_total Expression: {namespace="sepp-2",resource_uri="/oauth2/token",remote_sepp_set_name="RPS1",cause="Missing mandatory fields"}
Service Operation	NA
Response Code	NA

5.2.16.3 Number of requests to which SEPP responded with an error towards a particular SEPP

Table 5-298 Number of requests to which SEPP responded with an error towards a particular SEPP

Field	Details
KPI Detail	Number of requests to which SEPP responded with an error towards a particular SEPP.
Metric Used for KPI	ocsepp_network_id_validation_body_failure_total Expression: {namespace="sepp-2",resource_uri="/oauth2/token",remote_sepp_set_name="RPS1"} ocsepp_network_id_validation_body_failure_total{namespace="sepp-2",resource_uri="/oauth2/token",remote_sepp_name="sepp1"}
Service Operation	NA
Response Code	NA

5.3 SEPP Alerts

This section provides information about the SEPP alerts and their configuration.

Note

To configure alerts in SEPP when using environment other than CNE:

- namespace: {{\$labels.kubernetes_namespace}}
- podname: {{\$labels.kubernetes_pod_name}}

To configure alerts in SEPP when using CNE 25.2.2xx version:

- namespace: {{\$labels.namespace}}
- podname: {{\$labels.pod}}

Note

Several Alert labels had extra spaces surrounding their values, which have now been corrected. The affected labels include: namespace, app, pod, header, api_url, requestPath, source, and nf_instance_id.

For example:

Previously, label values contained extra spaces due to the template formatting, such as:

```
namespace: ' {{ $labels.namespace }} '
```

which produced values like:

```
namespace: " sepp-namespace "
```

.

The formatting has been updated to remove these spaces:

```
namespace: '{{ $labels.namespace }}'
```

resulting in properly trimmed values like:

```
namespace: "sepp-namespace"
```

.

5.3.1 System Level Alerts

5.3.1.1 SEPPPodMemoryUsageAlert

Table 5-299 SEPPPodMemoryUsageAlert

Field	Details
Trigger Condition	Pod memory usage is above the threshold (70%)
Severity	Warning

Table 5-299 (Cont.) SEPPPodMemoryUsageAlert

Field	Details
Alert details provided	Summary <pre>'namespace: {{\$labels.namespace}}, podname: {{\$labels.pod}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Memory usage is {{ \$value printf "%.2f" }} which is above 70% (current value is: {{ \$value }})'</pre> Expression: <pre>(sum by(namespace,container,pod) (container_memory_usage_bytes{namespace="sepp- namespace",container=~".*cn32c-svc.* .pn32c- svc.* .cn32f-svc.* .pn32f-svc.* .config-mgr- svc.* .n32-egress-gateway.* .n32-ingress- gateway.* .plmn-egress-gateway.* .plmn-ingress- gateway.* .nf-mediation.* .nrf-client- nfmanagement.* .nrf-client-nfdiscovery.*"})) / (sum by (namespace,container,pod) (kube_pod_container_resource_limits{resource="memory" ,namespace="sepp-namespace",container=~".*cn32c- svc.* .pn32c-svc.* .cn32f-svc.* .pn32f- svc.* .config-mgr-svc.* .n32-egress- gateway.* .n32-ingress-gateway.* .plmn-egress- gateway.* .plmn-ingress-gateway.* .nf- mediation.* .nrf-client-nfmanagement.* .nrf-client- nfdiscovery.*"})) * 100 >= 70</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4003
Metric Used	kube_pod_container_resource_limits Note: This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert gets cleared when the memory utilization falls below the critical threshold. Note: The threshold is configurable in the <i>SeppAlertrules.yaml</i> file. If guidance is required, contact My Oracle Support.

5.3.1.2 SEPPPodCpuUsageAlert

Table 5-300 SEPPPodCpuUsageAlert

Field	Details
Trigger Condition	Pod CPU usage is above the threshold (70%)
Severity	Warning

Table 5-300 (Cont.) SEPPPodCpuUsageAlert

Field	Details
Alert details provided	Summary <pre>'namespace: {{\$labels.namespace}}, podname: {{\$labels.pod}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: CPU usage is {{ \$value printf "%.2f" }} which is usage is above 70% (current value is: {{ \$value }})'</pre> Expression: <pre>(sum by (namespace,container,pod) (rate(container_cpu_usage_seconds_total{namespace="se pp-namespace",container=~".*cn32c-svc.* .pn32c- svc.* .cn32f-svc.* .pn32f-svc.* .config-mgr- svc.* .n32-egress-gateway.* .n32-ingress- gateway.* .plmn-egress-gateway.* .plmn-ingress- gateway.* .nf-mediation.* .nrf-client- nfmanagement.* .nrf-client-nfdiscovery.*"} [2m]))) / (sum by (namespace, container,pod) (kube_pod_container_resource_limits{resource="cpu",na mespace="sepp-namespace",container=~".*cn32c- svc.* .pn32c-svc.* .cn32f-svc.* .pn32f- svc.* .config-mgr-svc.* .n32-egress- gateway.* .n32-ingress-gateway.* .plmn-egress- gateway.* .plmn-ingress-gateway.* .nf- mediation.* .nrf-client-nfmanagement.* .nrf-client- nfdiscovery.*"}))) * 100 >= 70</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4002
Metric Used	container_cpu_usage_seconds_total Note : This is a Kubernetes metric used for instance availability monitoring. If the metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert gets cleared when the CPU utilization is below the critical threshold. Note: The threshold is configurable in the <i>SeppAlertrules.yaml</i> file. If guidance is required, contact My Oracle Support.

5.3.1.3 ocseppPodsRestart

Table 5-301 ocseppPodsRestart

Field	Description
Trigger Condition	Triggered when a pod belonging to any of the SEPP services in a namespace has been restarted.
Severity	Major

Table 5-301 (Cont.) ocseppPodsRestart

Field	Description
Alert Details Provided	Summary <pre>namespace: {{\$labels.namespace}}, kubernetes_pod_name: {{\$labels.pod}}, Container: {{\$labels.container}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }} {{ end }}</pre> Expression: <pre>increase(kube_pod_container_status_restarts_total{namespace="sepp-namespace"}[2m]) > 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4071
Metric Name	kube_pod_container_status_restarts_total Note: This is a Kubernetes metric. If this metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert is cleared automatically if the specific pod is up. Steps: 1. Refer to the application logs on Kibana and filter based on the pod name. Check for database related failures such as connectivity, Kubernetes secrets, and so on. 2. Run the following command to check orchestration logs for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <desired full pod name> -n <namespace></pre> 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.4 ocseppAppinfoServiceDown

Table 5-302 ocseppAppinfoServiceDown

Field	Description
Trigger Condition	Triggered when appinfo services is unavailable.
Severity	Critical
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="appinfo", namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="appinfo", namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4072
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-302 (Cont.) ocseppAppinfoServiceDown

Field	Description
Resolution	The alert is cleared when the appinfo services is available. Steps: 1. Run the following command to check the orchestration log of appinfoservice and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.5 ocseppCn32cServiceDown

Table 5-303 ocseppCn32cServiceDown

Field	Description
Trigger Condition	Triggered when cn32c-svc service is unavailable.
Severity	Critical

Table 5-303 (Cont.) ocseppCn32cServiceDown

Field	Description
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="cn32c-svc",namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="cn32c-svc",namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4073
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert is cleared when the cn32c-svc services are available. Steps: 1. Run the following command to check the orchestration log of cn32c-svc and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.6 ocseppCn32fServiceDown

Table 5-304 ocseppCn32fServiceDown

Field	Description
Trigger Condition	Triggered when cn32f-svc service is unavailable.
Severity	Critical
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="cn32f-svc",namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="cn32f-svc",namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4074
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-304 (Cont.) ocseppCn32fServiceDown

Field	Description
Resolution	The alert is cleared when the cn32f-svc services are available. Steps: 1. Run the following command to check the orchestration log of cn32f-svc and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.7 ocseppConfigMgrServiceDown

Table 5-305 ocseppConfigMgrServiceDown

Field	Description
Trigger Condition	Triggered when config-mgr-svc services is unavailable.
Severity	Critical

Table 5-305 (Cont.) ocseppConfigMgrServiceDown

Field	Description
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="config-mgr-svc", namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="config-mgr-svc", namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4075
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert is cleared when the config-mgr-svc services are available. Steps: 1. Run the following command to check the orchestration log of config-mgr-svc and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.8 ocseppN32EgwServiceDown

Table 5-306 ocseppN32EgwServiceDown

Field	Description
Trigger Condition	Triggered when n32-egress-gateway is unavailable.
Severity	Critical
Alert Details Provided	Summary <pre>'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down'</pre> Expression: <pre>absent(up{app_kubernetes_io_name="n32-egress-gateway",namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="n32-egress-gateway",namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4076
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-306 (Cont.) ocseppN32EgwServiceDown

Field	Description
Resolution	The alert is cleared when the n32-egress-gateway services are available. Steps: 1. Run the following command to check the orchestration logs of N32 EGW Service and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.9 ocseppN32IgwServiceDown

Table 5-307 ocseppN32IgwServiceDown

Field	Description
Trigger Condition	Triggered when n32-ingress-gateway is unavailable.
Severity	Critical

Table 5-307 (Cont.) ocseppN32lgwServiceDown

Field	Description
Alert Details Provided	Summary <pre>'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down'</pre> Expression: <pre>absent(up{app_kubernetes_io_name="n32- ingress-gateway",namespace="sepp- namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="n32- ingress-gateway",namespace="sepp- namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4077
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-307 (Cont.) ocseppN32lgwServiceDown

Field	Description
Resolution	The alert is cleared when the n32-ingress-gateway services are available. Steps: 1. Run the following command to check the orchestration log of n32-ingress-gateway and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.10 ocseppConfigserverServiceDown

Table 5-308 ocseppConfigserverServiceDown

Field	Description
Trigger Condition	Triggered when config-server(ocpm) service is unavailable.
Severity	Critical

Table 5-308 (Cont.) ocseppConfigserverServiceDown

Field	Description
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="config-server",namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="config-server",namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4078
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert is cleared when the config-server(ocpm) services are available. Steps: 1. Run the following command to check the orchestration log of config-server(ocpm) and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.11 ocseppPerfinfoServiceDown

Table 5-309 ocseppPerfinfoServiceDown

Field	Description
Trigger Condition	Triggered when perf-info service is unavailable.
Severity	Critical
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="perf-info",namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="perf-info",namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4079
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-309 (Cont.) ocseppPerfinfoServiceDown

Field	Description
Resolution	The alert is cleared when the perf-info services are available. Steps: 1. Run the following command to check the orchestration log of perf-info services and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.12 ocseppPlmnEgwServiceDown

Table 5-310 ocseppPlmnEgwServiceDown

Field	Description
Trigger Condition	Triggered when plmn-egress-gateway service is unavailable.
Severity	Critical

Table 5-310 (Cont.) ocseppPlmnEgwServiceDown

Field	Description
Alert Details Provided	Summary <pre>'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down'</pre> Expression: <pre>absent(up{app_kubernetes_io_name="plmn-egress-gateway",namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="plmn-egress-gateway",namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4080
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-310 (Cont.) ocseppPlmnEgwServiceDown

Field	Description
Resolution	The alert is cleared when the plmn-egress-gateway services are available. Steps: 1. Run the following command to check the orchestration log of plmn-egress-gateway services and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.13 ocseppPlmnIgwServiceDown

Table 5-311 ocseppPlmnIgwServiceDown

Field	Description
Trigger Condition	Triggered when plmn-ingress-gateway service is unavailable.
Severity	Critical

Table 5-311 (Cont.) ocseppPlmnlgwServiceDown

Field	Description
Alert Details Provided	Summary <pre>'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down'</pre> Expression: <pre>absent(up{app_kubernetes_io_name="plm n-ingress-gateway",namespace="sepp- namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="plmn- ingress-gateway",namespace="sepp- namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4081
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-311 (Cont.) ocseppPlmnlgwServiceDown

Field	Description
Resolution	The alert is cleared when the plmn-ingress-gateway services are available. Steps: 1. Run the following command to check the orchestration log of plmn-ingress-gateway services and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.14 ocseppPn32cServiceDown

Table 5-312 ocseppPn32cServiceDown

Field	Description
Trigger Condition	Triggered when pn32c-svc service is unavailable.
Severity	Critical

Table 5-312 (Cont.) ocseppPn32cServiceDown

Field	Description
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="pn32c-svc",namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="pn32c-svc",namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4082
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert is cleared when the pn32c-svc services are available. Steps: 1. Run the following command to check the orchestration log of pn32c-svc services and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.15 ocseppPn32fServiceDown

Table 5-313 ocseppPn32fServiceDown

Field	Description
Trigger Condition	Triggered when pn32f-svc service is unavailable.
Severity	Critical
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="pn32f-svc",namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="pn32f-svc",namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4083
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-313 (Cont.) ocseppPn32fServiceDown

Field	Description
Resolution	The alert is cleared when the pn32f-svc services are available. Steps: 1. Run the following command to check the orchestration log of pn32f-svc services and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.16 ocseppNrfdiscServiceDown

Table 5-314 ocseppNrfdiscServiceDown

Field	Description
Trigger Condition	Triggered when nrf-client-nfdiscovery service is unavailable.
Severity	Critical

Table 5-314 (Cont.) ocseppNrfdiscServiceDown

Field	Description
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="nrf-client-nfdiscovery",namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="nrf-client-nfdiscovery",namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4084
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert is cleared when the nrf-client-nfdiscovery services are available. Steps: 1. Run the following command to check the orchestration log of nrf-client-nfdiscovery services and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.17 ocseppNrfmgmServiceDown

Table 5-315 ocseppNrfmgmServiceDown

Field	Description
Trigger Condition	Triggered when nrf-client-nfmanagement service is unavailable.
Severity	Critical
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="nrf-client-nfmanagement", namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="nrf-client-nfmanagement", namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4085
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-315 (Cont.) ocseppNrfgmServiceDown

Field	Description
Resolution	The alert is cleared when the nrf-client-nfmanagement services are available. Steps: 1. Run the following command to check the orchestration log of nrf-client-nfmanagement services and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.18 ocseppAlternrouteServiceDown

Table 5-316 ocseppAlternrouteServiceDown

Field	Description
Trigger Condition	Triggered when alternate-route service is unavailable.
Severity	Critical

Table 5-316 (Cont.) ocseppAlternrouteServiceDown

Field	Description
Alert Details Provided	Summary <pre>'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down'</pre> Expression: <pre>absent(up{app_kubernetes_io_name="alt ernate-route",namespace="sepp- namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="alternate -route",namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4086
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-316 (Cont.) ocseppAlternrouteServiceDown

Field	Description
Resolution	The alert is cleared when the alternate-route services are available. Steps: 1. Run the following command to check the orchestration log of alternate-route services and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.19 ocseppMediationServiceDown

Table 5-317 ocseppMediationServiceDown

Field	Description
Trigger Condition	Triggered when nf-mediation service is unavailable.
Severity	Critical

Table 5-317 (Cont.) ocseppMediationServiceDown

Field	Description
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="nf-mediation",namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="nf-mediation",namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4087
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.
Resolution	The alert is cleared when the nf-mediation services are available. Steps: 1. Run the following command to check the orchestration log of nf-mediation services and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.20 ocseppCoherenceServiceDown

Table 5-318 ocseppCoherenceServiceDown

Field	Description
Trigger Condition	Triggered when coherence-svc service is unavailable.
Severity	Critical
Alert Details Provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : OCSEPP {{\$labels.app_kubernetes_io_name}} service down' Expression: <pre>absent(up{app_kubernetes_io_name="coherence-svc", namespace="sepp-namespace"}) or sum by(namespace, app_kubernetes_io_name) (up{app_kubernetes_io_name="coherence-svc", namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4088
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-318 (Cont.) ocseppCoherenceServiceDown

Field	Description
Resolution	The alert is cleared when the coherence-svc services are available. Steps: 1. Run the following command to check the orchestration log of coherence-svc services and check for liveness or readiness probe failures: <pre>kubectl get po -n <namespace></pre> Note the full name of the pod that is not running, and use it in the following command: <pre>kubectl describe pod <specific desired full pod name> -n <namespace></pre> 2. Refer to the application logs on Kibana and filter based on above service name. Check for ERROR and WARNING logs related to this service. 3. Check the database status. For more information, see <i>Oracle Communications Cloud Native Core, cnDBTier User Guide</i>. 4. Depending on the failure reason, take the resolution steps. 5. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.1.21 ocseppNfStatusUnavailable

Table 5-319 ocseppNfStatusUnavailable

Field	Description
Trigger Condition	Triggered when all the SEPP services are unavailable, either because the SEPP is getting deployed or purged.
Severity	Critical

Table 5-319 (Cont.) ocseppNfStatusUnavailable

Field	Description
Alert Details Provided	Summary namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : All OCSEPP services are unavailable.' Expression: <pre>absent(up{app_kubernetes_io_part_of="ocsepp",namespace="sepp-namespace"}) or sum(up{app_kubernetes_io_part_of="ocsepp", namespace="sepp-namespace"}) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4089
Metric Name	up Note: This is a Prometheus metric used for instance availability monitoring. If this metric is not available, use the similar metric as exposed by the monitoring system.

Table 5-319 (Cont.) ocseppNfStatusUnavailable

Field	Description
Resolution	The alert is cleared automatically when the SEPP services start becoming available. Steps: 1. Check for service specific alerts which may be causing the issues with service exposure. 2. Run the following command to check if the pod's status is in "Running" state: <pre>kubectl -n <namespace> get pod</pre> If it is not in running state, capture the pod logs and events. 3. Run the following command to fetch the events as follows: <pre>kubectl get events --sort-by=.metadata.creationTimestamp -n <namespace></pre> 4. Refer to the application logs on Kibana and check for database related failures such as connectivity, invalid secrets, and so on. The logs can be filtered based on the services. 5. Run the following command to check Helm status and make sure there are no errors: <pre>helm status <helm release name of the desired NF> -n <namespace></pre> If it is not in "STATUS: DEPLOYED", then again capture logs and events. 6. If the issue persists, capture all the outputs from the above steps and contact My Oracle Support.

5.3.2 Application Level Alerts

5.3.2.1 Common Alerts

5.3.2.1.1 SEPPN32fRoutingFailure

Table 5-320 SEPPN32fRoutingFailure

Field	Details
Trigger Condition	N32f service not able to forward message
Severity	Info

Table 5-320 (Cont.) SEPPN32fRoutingFailure

Field	Details
Alert details provided	Summary <pre>namespace: {{ \$labels.namespace}}, podname: {{ \$labels.pod}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }} {{ end }}</pre> Expression: <pre>idelta(ocsepp_cn32f_requests_failure_total{namespace= "sepp-namespace"}[2m]) > 0 or (ocsepp_cn32f_requests_failure_total{namespace="sepp- namespace"} unless ocsepp_cn32f_requests_failure_total{namespace="sepp- namespace"} offset 2m)</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4001
Metric Used	ocsepp_cn32f_requests_failure_total
Resolution	The alert gets cleared when Consumer SEPP accepts request only if producer NF domain and PLMN match the Remote SEPP configured. Steps: The failure reason is present in the alert. Possible Resolutions : 1. Check whether the Remote SEPP is present in database. 2. Validate the Remote SEPP PLMN which is configured. 3. Validate the handshake is completed with the remote SEPP and context is present in database. 4. Validate the producer NF Domain. 5. Check whether the Remote SEPP Set for required Remote SEPP is present in the database. 6. Check whether the N32F route is present in database (common_configuration table).

5.3.2.1.2 SEPPConfigMgrRouteFailureAlert

Table 5-321 SEPPConfigMgrRouteFailureAlert

Field	Details
Trigger Condition	When routing failure occurs while posting remote SEPP or roaming partner set, this alert will be raised.
Severity	Major

Table 5-321 (Cont.) SEPPConfigMgrRouteFailureAlert

Field	Details
Alert Details Provided	Summary <pre>namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Route Failure has occurred because {{ \$labels.http_error_message }}</pre> Expression: <pre>sum(increase(ocsepp_configmgr_routefailure_total{namespace="sepp-namespace",app="config-mgr-svc"}[5m]) >0 or (ocsepp_configmgr_routefailure_total{namespace="sepp-namespace",app="config-mgr-svc"} unless ocsepp_configmgr_routefailure_total{namespace="sepp-namespace",app="config-mgr-svc"} offset 5m)) by (namespace,http_status,http_error_message) > 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4038
Metric Name	ocsepp_configmgr_routefailure_total
Resolution	The alert is cleared if no new failures are observed in 5 minutes window. Possible resolutions: 1. Ensure that the correct Remote Partner Profile (RPP) profile has been used. 2. Ensure that the correct Remote SEPP Set (RSS) profile has been used.

5.3.2.1.3 EgressSbiErrorRateAbove1Percent

Table 5-322 EgressSbiErrorRateAbove1Percent

Field	Details
Trigger Condition	Sbi Transaction Error Rate exceeded configured threshold
Severity	Major

Table 5-322 (Cont.) EgressSbiErrorRateAbove1Percent

Field	Details
Alert details provided	Summary "Sbi Transaction Error Rate detected above 1 Percent of Total Sbi Transactions" Expression <pre>"sum(rate(oc_egressgateway_sbiRouting_http_responses_total{namespace="sepp-namespace",Status!~\"2.*\"}[5m])) by (namespace,app, pod) / sum(rate(oc_egressgateway_sbiRouting_http_responses_total{namespace="sepp-namespace"}[5m])) by (namespace, app, pod) *100 >= 1"</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7001
Metric Used	oc_egressgateway_sbiRouting_http_responses_total
Resolution	This alert will be raised when the total SBI transaction error rate will be above 1% of the total transaction done during 5 minute time period. Metric will be cleared when the error rate will be below 1%.

5.3.2.1.4 ocseppNfProfileStatusInNRFDn

Table 5-323 ocseppNfProfileStatusInNRFDn

Field	Details
Trigger Condition	SEPP registration with configured NRF gets failed.
Severity	Critical
Alert details provided	Summary namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }} : The OCSEPP NF profile status with the NRF is currently down. Expression: <pre>nrfclient_nf_status_with_nrf{app_kubernetes_io_part_of="ocsepp",namespace="sepp-namespace"} != 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4090
Metric Used	nrfclient_nf_status_with_nrf
Resolution	The alert will be raised if the SEPP does not get registered in the configured NRF . The alert will be cleared when the SEPP status with NRF getsregistered.

Note

This alert is not applicable for the Roaming Hub mode.

5.3.2.2 Handshake Alerts

5.3.2.2.1 SEPPCn32cHandshakeFailureAlert

Table 5-324 SEPPCn32cHandshakeFailureAlert

Field	Details
Trigger Condition	Handshake procedure has failed on Consumer SEPP
Severity	Major
Alert details provided	Summary <pre>'namespace: {{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}'</pre> Expression: <pre>sum(increase(ocsepp_n32c_handshake_failure_attempts_total{app="cn32c-svc",namespace="sepp-namespace"}[5m]) >0 or (ocsepp_n32c_handshake_failure_attempts_total{app="cn32c-svc",namespace="sepp-namespace"} unless ocsepp_n32c_handshake_failure_attempts_total{app="cn32c-svc",namespace="sepp-namespace"} offset 5m)) by (namespace,remote_sepp_name,nf_instance_id,peer_fqdn,app,container,reason) > 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.2001
Metric Used	ocsepp_n32c_handshake_failure_attempts_total filtered by app=cn32c-svc

Table 5-324 (Cont.) SEPPCn32cHandshakeFailureAlert

Field	Details
Resolution 1	The alert gets cleared when the N32C Handshake is established after successful TCP connection to remote SEPP. Failure reason: Release name used while helm installation is other than ocsepp-release. Error Verification: Check the failure reason in the alert. If the failure reason is <i>404 –route not found</i> or <i>Route not found</i>, follow the recovery steps: - Run the following command to get pod details: \$ kubectl get pods -n <namespace> Example: # kubectl get pods -n csepp NAME READY STATUS RESTARTS AGE ocsepp-release-appinfo-6cdc48fc47-c9gfv 1/1 Running 0 8d ocsepp-release-cn32c-svc-6547db777d-76gwd 1/1 Running 0 8d ocsepp-release-cn32f-svc-7cd54bdf68-czbnb 1/1 Running 0 8d ocsepp-release-config-mgr-svc-79c95d4b9d-8stk7 1/1 Running 0 8d ocsepp-release-n32-egress-gateway-54c658b947-s5f9m 0/2 Pending 0 23h ocsepp-release-n32-egress-gateway-54c658b947-scvvp 2/2 Running 0 7d23h ocsepp-release-n32-ingress-gateway-777c68cb9-8jsdc 0/2 Pending 0 23h ocsepp-release-n32-ingress-gateway-777c68cb9-98t7x 0/2 Init:ImagePullBackOff 0 23h ocsepp-release-pn32c-svc-58bff857f-jmfdd 1/1 Running 0 8d ocsepp-release-pn32f-svc-784d5c7568-rh24g - Run the following command to navigate to the pod: \$ kubectl exec -it <config-mgr-pod name> -n <namespace> bash Example:

Table 5-324 (Cont.) SEPPCn32cHandshakeFailureAlert

Field	Details
	<pre>\$ kubectl exec -it ocsepp-release-config-mgr-svc-79c95d4b9d-8stk7 -n csepp bash</pre> - Run the command to get the existing route details present on N32 Egress Gateway: <pre>curl -X GET http://<config-manager-service-name>:9090/sepp/nf-common-component/v1/egw/n32/routesconfiguration</pre> Example: <pre>curl -X GET http://ocsepp-release-config-mgr-svc:9090/sepp/nf-common-component/v1/egw/n32/routesconfiguration</pre> - If this output is null, add the configuration details in config-mgr - svc deployment. For more information about the configuration details, see the <i>Deployment Configuration for Config-mgr-svc</i> section in <i>Oracle Communications Cloud Native Core Security Edge Protection Proxy Installation Guide</i>. - After the config-mgr - svc pod is restarted, run the step1 to step3 again. After adding the configuration, rerun the curl command mentioned in step3 to get the route details. - Delete and add the RemoteSepp and reinitiate the handshake. If the value is still null, contact My Oracle Support.
Resolution 2	The alert gets cleared when the N32C Handshake is established after successful TCP connection to remote SEPP. Steps: The failure reason is present in the alert. Possible Resolutions: - Disable the Remote SEPP. - Delete the Remote SEPP. - Update and reinitiate Handshake.

5.3.2.2.2 SEPPPN32cHandshakeFailureAlert

Table 5-325 SEPPPN32cHandshakeFailureAlert

Field	Details
Trigger Condition	Handshake procedure has failed on Producer sepp
Severity	Major

Table 5-325 (Cont.) SEPPN32cHandshakeFailureAlert

Field	Details
Alert details provided	Summary 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Handshake procedure has failed on Producer side because {{ \$labels.http_error_message }}' Expression: sum(increase(ocsepp_n32c_handshake_failure_attempts_total{app="pn32c-svc",namespace="sepp-namespace"}[5m]) >0 or (ocsepp_n32c_handshake_failure_attempts_total{app="pn32c-svc",namespace="sepp-namespace"} unless ocsepp_n32c_handshake_failure_attempts_total{app="pn32c-svc",namespace="sepp-namespace"} offset 5m)) by (namespace,remote_sepp_name,nf_instance_id,peer_fqdn,app,reason) > 0
OID	1.3.6.1.4.1.323.5.3.46.1.2.3001
Metric Used	ocsepp_n32c_handshake_failure_attempts_total filtered by app=pn32c-svc
Resolution	The alert gets cleared when the N32C Handshake is successful due to TCP connection success of Producer to consumer SEPP. Steps: The failure reason is present in the alert. Possible Resolution: Update and reinitiate the Handshake.

5.3.2.3 Upgrade Alerts

5.3.2.3.1 SEPPUpgradeStartedAlert

Table 5-326 SEPPUpgradeStartedAlert

Field	Details
Trigger Condition	Rest API trigger at start of Upgrade
Severity	NA

Table 5-326 (Cont.) SEPPUpgradeStartedAlert

Field	Details
Alert details provided	applicationname alertname servicename releasename namespace oid severity vendor sourcerelease targetrelease
OID	1.3.6.1.4.1.323.5.3.46.1.2.8001
Metric Used	NA
Resolution	If a success alert is generated then start and failure alerts will be cleared.

5.3.2.3.2 SEPPUpgradeFailedAlert

Table 5-327 SEPPUpgradeFailedAlert

Field	Details
Trigger Condition	Rest API trigger at failure of Upgrade
Severity	NA
Alert details provided	applicationname alertname servicename releasename namespace oid severity vendor sourcerelease targetrelease
OID	1.3.6.1.4.1.323.5.3.46.1.2.8002
Metric Used	NA

Table 5-327 (Cont.) SEPPUpgradeFailedAlert

Field	Details
Resolution	If a success alert is generated then start and failure alerts will be cleared. Possible resolutions: 1. Check the pre or post upgrade logs in Kibana to analyze the cause of failure. 2. Filter the upgrade logs using the pod name filter. Example: ocsepp-release-update-db. 3. If the cause of upgrade failure is database or network connectivity issue, contact the system administrator. 4. If the upgrade failure occurs during the preupgrade phase, resolve the issue, then perform an upgrade. Do not perform rollback because SEPP deployment remains in the source or older release. 5. If the upgrade failure occurs during the postupgrade phase, for example, post upgrade hook failure due to target release pod not moving to ready state, then perform a rollback.

5.3.2.3.3 SEPPUpgradeSuccessfulAlert

Table 5-328 SEPPUpgradeSuccessfulAlert

Field	Details
Trigger Condition	Rest API trigger at success of Upgrade
Severity	NA
Alert details provided	applicationname alertname servicename releasename namespace oid severity vendor sourcerelease targetrelease
OID	1.3.6.1.4.1.323.5.3.46.1.2.8003
Metric Used	NA
Resolution	This is an information alert raised when software upgrade is successful. This alert will auto clear after the duration set in resolve_timeout value of AlertManager.

5.3.2.4 Rollback Alerts

5.3.2.4.1 SEPPRollbackStartedAlert

Table 5-329 SEPPRollbackStartedAlert

Field	Details
Trigger Condition	Rest API trigger at start of Rollback
Severity	NA
Alert details provided	applicationname alertname servicename releasename namespace oid severity vendor sourcerelease targetrelease
OID	1.3.6.1.4.1.323.5.3.46.1.2.8004
Metric Used	NA
Resolution	If a success alert is generated then start and failure alerts will be cleared.

5.3.2.4.2 SEPPRollbackFailedAlert

Table 5-330 SEPPRollbackFailedAlert

Field	Details
Trigger Condition	Rest API trigger at failure of Rollback
Severity	NA
Alert details provided	applicationname alertname servicename releasename namespace oid severity vendor sourcerelease targetrelease
OID	1.3.6.1.4.1.323.5.3.46.1.2.8005
Metric Used	NA

Table 5-330 (Cont.) SEPPRollbackFailedAlert

Field	Details
Resolution	If a success alert is generated then start and failure alerts will be cleared. Possible resolutions: 1. Check the rollback logs in Kibana to analyze the cause of failure. 2. Filter the rollback logs using the pod name filter. Example: ocsepp-release-update-db. 3. If the cause of rollback failure is database or network connectivity issue, contact the system administrator.

5.3.2.4.3 SEPPRollbackSuccessfulAlert

Table 5-331 SEPPRollbackSuccessfulAlert

Field	Details
Trigger Condition	Rest API trigger at success of Rollback
Severity	NA
Alert details provided	applicationname alertname servicename releasename namespace oid severity vendor sourcerelease targetrelease
OID	1.3.6.1.4.1.323.5.3.46.1.2.8006
Metric Used	NA
Resolution	Cleared after DEFAULT_DURATION_FOR_ALERT_EXPIRY minutes

5.3.2.5 Global Rate Limiting on Ingress Gateway of SEPP Alerts

5.3.2.5.1 IngressGlobalMessageDropAbovePointOnePercent

Table 5-332 IngressGlobalMessageDropAbovePointOnePercent

Field	Details
Trigger Condition	Ingress Global Message Drop Rate detected greater than or equal to 0.1 Percent of Total Transactions.
Severity	Warning

Table 5-332 (Cont.) IngressGlobalMessageDropAbovePointOnePercent

Field	Details
Alert details provided	Summary "Ingress Global Message Drop Rate detected above 0.1 Percent of Total Transactions" Expression <pre>sum(rate(oc_ingressgateway_global_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,app)/ sum(rate(oc_ingressgateway_global_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,app) *100 >= 0.1 < 1</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7002
Metric Used	oc_ingressgateway_global_ratelimit_total
Resolution	The alert will be raised when the percentage of messages rejected for Global Rate Limit will be greater than or equal to 0.1% of the total messages received. This will get cleared once percentage of message rejected is below 0.1% or greater than or equal to 1%. Possible resolutions: 1. If the message drop percentage is different from the expected value, verify the configuration. 2. For all the configuration related information, refer 'Troubleshooting Steps for Rate Limiting Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.5.2 IngressGlobalMessageDropAbove1Percent

Table 5-333 IngressGlobalMessageDropAbove1Percent

Field	Details
Trigger Condition	Ingress Global Message Drop Rate detected greater than or equal to 1 Percent of Total Transactions.
Severity	Warning

Table 5-333 (Cont.) IngressGlobalMessageDropAbove1Percent

Field	Details
Alert details provided	Summary "Ingress Global Message Drop Rate detected above 1 Percent of Total Transactions" Expression <pre>sum(rate(oc_ingressgateway_global_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,app)/ sum(rate(oc_ingressgateway_global_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,app) *100 >= 1 < 10</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7003
Metric Used	oc_ingressgateway_global_ratelimit_total
Resolution	The alert will be raised when the percentage of messages rejected for Global Rate Limit will be greater than or equal to 1% of the total messages received. This will get cleared once percentage of message rejected is below 1% greater than or equal to 10%. Possible resolutions: 1. If the message drop percentage is different from the expected value, verify the configuration. 2. For all the configuration related information, refer 'Troubleshooting Steps for Rate Limiting Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.5.3 IngressGlobalMessageDropAbove10Percent

Table 5-334 IngressGlobalMessageDropAbove10Percent

Field	Details
Trigger Condition	Ingress Global Message Drop Rate detected greater than or equal to 10 Percent of Total Transactions.
Severity	Minor

Table 5-334 (Cont.) IngressGlobalMessageDropAbove10Percent

Field	Details
Alert details provided	Summary "Ingress Global Message Drop Rate detected above 10 Percent of Total Transactions" Expression <pre>sum(rate(oc_ingressgateway_global_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,app)/ sum(rate(oc_ingressgateway_global_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,app) *100 >= 10 < 25</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7004
Metric Used	oc_ingressgateway_global_ratelimit_total
Resolution	The alert will be raised when the percentage of messages rejected for Global Rate Limit will be greater than or equal to 10% of the total messages received. This will get cleared once percentage of message rejected is below 10% or greater than or equal to 25% . Possible resolutions: 1. If the message drop percentage is different from the expected value, verify the configuration. 2. For all the configuration related information, refer 'Troubleshooting Steps for Rate Limiting Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.5.4 IngressGlobalMessageDropAbove25Percent

Table 5-335 IngressGlobalMessageDropAbove25Percent

Field	Details
Trigger Condition	Ingress Global Message Drop Rate detected greater than or equal to 25 Percent of Total Transactions
Severity	Major

Table 5-335 (Cont.) IngressGlobalMessageDropAbove25Percent

Field	Details
Alert details provided	Summary "Ingress Global Message Drop Rate detected above 25 Percent of Total Transactions" Expression <pre>sum(rate(oc_ingressgateway_global_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,app)/ sum(rate(oc_ingressgateway_global_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,app) *100 >= 25 < 50</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7005
Metric Used	oc_ingressgateway_global_ratelimit_total
Resolution	The alert will be raised when the percentage of messages rejected for Global Rate Limit will be greater than or equal to 25% of the total messages received. This will get cleared once percentage of message rejected is below 25% or greater than or equal to 50%. Possible resolutions: 1. If the message drop percentage is different from the expected value, verify the configuration. 2. For all the configuration related information, refer 'Troubleshooting Steps for Rate Limiting Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.5.5 IngressGlobalMessageDropAbove50Percent

Table 5-336 IngressGlobalMessageDropAbove50Percent

Field	Details
Trigger Condition	Ingress Global Message Drop Rate detected greater than or equal to 50 Percent of Total Transactions
Severity	Critical

Table 5-336 (Cont.) IngressGlobalMessageDropAbove50Percent

Field	Details
Alert details provided	Summary "Ingress Global Message Drop Rate detected above 50 Percent of Total Transactions" Expression <pre>sum(rate(oc_ingressgateway_global_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,app)/ sum(rate(oc_ingressgateway_global_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,app) *100 >= 50</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7006
Metric Used	oc_ingressgateway_global_ratelimit_total
Resolution	The alert will be raised when the percentage of messages rejected for Global Rate Limit will be greater than or equal to 50% of the total messages received. This will get cleared once percentage of message rejected is below 50%. Possible resolutions: 1. If the message drop percentage is different from the expected value, verify the configuration. 2. For all the configuration related information, refer 'Troubleshooting Steps for Rate Limiting Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.6 Topology Hiding Alerts

5.3.2.6.1 SEPPN32fTopologyOperationFailureAlert

Table 5-337 SEPPN32fTopologyOperationFailureAlert

Field	Details
Trigger Condition	Topology Hiding or Recovery Failure exceeded configured threshold (1%)
Severity	Major

Table 5-337 (Cont.) SEPPN32fTopologyOperationFailureAlert

Field	Details
Alert details provided	Summary "Topology hiding/recovery operation failures reached more than configured threshold" Expression: <pre>delta(ocsepp_topology_header_failure_total{namespace="sepp-namespace"}[2m]) > 0 or (ocsepp_topology_header_failure_total{namespace="sepp-namespace"} unless ocsepp_topology_header_failure_total{namespace="sepp-namespace"} offset 2m)</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4004
Metric Used	ocsepp_topology_header_failure_total, ocsepp_topology_header_success_total
Resolution	This alert will be raised when the total Topology Hiding or Recovery failures reach more than 1%. Alert will be cleared when the error rate is below 1%. Possible Resolutions: 1. Check the header for which alert is raised, header name present in alert label. 2. Verify the error_msg using "ocsepp_topology_header_failure_total" metric and KPI. 3. Fix or add configuration for the header. Note: The alert will be cleared only if the corresponding success metric is pegged.

5.3.2.6.2 SEPPN32fTopologyBodyOperationFailureAlert

Table 5-338 SEPPN32fTopologyBodyOperationFailureAlert

Field	Details
Trigger Condition	Topology Operation failed and exceeds defined threshold
Severity	Major

Table 5-338 (Cont.) SEPPN32fTopologyBodyOperationFailureAlert

Field	Details
Alert details provided	Summary "Topology Hiding/Recovery Operation failures reached more than configured threshold" Expression: delta(ocsepp_topology_body_failure_total{namespace="sepp-namespace"}[2m]) > 0 or (ocsepp_topology_body_failure_total{namespace="sepp-namespace"} unless ocsepp_topology_body_failure_total{namespace="sepp-namespace"} offset 2m)
OID	1.3.6.1.4.1.323.5.3.46.1.2.4006
Metric Used	ocsepp_topology_body_failure_total ocsepp_topology_body_success_total
Resolution	This alert will be raised when the total Topology Hiding or Recovery for message body failures reach more than 1%. Alert will be cleared when the error rate will be below 1%. Possible Resolutions: 1. Check the apiUrl, method for which alert is raised, apiUrl present in alert label. 2. Verify the error_msg using "ocsepp_topology_body_failure_total" metric and KPI. 3. Fix or add configuration for the body Identifiers. Note: The alert will be cleared only if the corresponding success metric is pegged.

5.3.2.7 5G SBI Message Mediation Support Alerts

5.3.2.7.1 SEPPCN32fMediationFailure

Table 5-339 SEPPCN32fMediationFailure

Field	Details
Trigger Condition	Mediation processing Failure
Severity	Info

Table 5-339 (Cont.) SEPPCN32fMediationFailure

Field	Details
Alert details provided	Summary namespace: {{ \$labels.namespace}}, podname: {{ \$labels.pod}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Mediation processing Failed because {{ \$labels.http_error_message }} Expression: <pre>increase(ocsepp_cn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status!="504 GATEWAY_TIMEOUT"}[10m]) > 0 or (ocsepp_cn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status!="504 GATEWAY_TIMEOUT"} unless ocsepp_cn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status!="504 GATEWAY_TIMEOUT"} offset 10m)</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4007
Metric Used	ocsepp_cn32f_mediation_response_failure
Resolution	This alert will be raised when Mediation microservice is unable to apply rules on the incoming request & response from SEPP. Possible Resolution: 1. Check if the Mediation Rules exist. 2. Check the Agenda Group in the mediation rule is matching from the request and response sent from SEPP.

5.3.2.7.2 SEPPCN32fMediationUnreachable

Table 5-340 SEPPCN32fMediationUnreachable

Field	Details
Trigger Condition	Mediation service is not accessible
Severity	Critical

Table 5-340 (Cont.) SEPPCN32fMediationUnreachable

Field	Details
Alert details provided	Summary <pre>namespace: {{ \$labels.namespace}}, podname: {{ \$labels.pod}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Mediation service is not accessible because {{ \$labels.http_error_message }}</pre> Expression: <pre>increase(ocsepp_cn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status="504 GATEWAY_TIMEOUT"}[10m]) > 0 or (ocsepp_cn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status="504 GATEWAY_TIMEOUT"} unless ocsepp_cn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status="504 GATEWAY_TIMEOUT"} offset 10m)</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4008
Metric Used	ocsepp_cn32f_mediation_response_failure
Resolution	This alert will be raised when Mediation microservice is not accessible. Possible Resolution: 1. Check if the Mediation microservice pod is up. 2. Check if Mediation Service Name and servicePort number is correct.

5.3.2.7.3 SEPPPN32fMediationFailure

Table 5-341 SEPPPN32fMediationFailure

Field	Details
Trigger Condition	Mediation processing Failure
Severity	Info

Table 5-341 (Cont.) SEPPPN32fMediationFailure

Field	Details
Alert details provided	Summary namespace: {{ \$labels.namespace}}, podname: {{ \$labels.pod}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Mediation processing Failed because {{ \$labels.http_error_message }} Expression: <pre>increase(ocsepp_pn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status!="504 GATEWAY_TIMEOUT"}[10m]) > 0 or (ocsepp_pn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status!="504 GATEWAY_TIMEOUT"} unless ocsepp_pn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status!="504 GATEWAY_TIMEOUT"} offset 10m)</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4009
Metric Used	ocsepp_pn32f_mediation_response_failure
Resolution	This alert will be raised when Mediation microservice is unable to apply rules on the incoming request & response from SEPP. Possible Resolution: 1. Check if the Mediation Rules exist. 2. Check the Agenda Group in the mediation rule is matching from the request and response sent from SEPP.

5.3.2.7.4 SEPPPN32fMediationUnreachable

Table 5-342 SEPPPN32fMediationUnreachable

Field	Details
Trigger Condition	Mediation service is not accessible
Severity	Critical

Table 5-342 (Cont.) SEPPPN32fMediationUnreachable

Field	Details
Alert details provided	Summary <pre>namespace: {{ \$labels.namespace}}, podname: {{ \$labels.pod}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Mediation service is not accessible because {{ \$labels.http_error_message }}</pre> Expression: <pre>increase(ocsepp_pn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status="504 GATEWAY_TIMEOUT"}[10m]) > 0 or (ocsepp_pn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status="504 GATEWAY_TIMEOUT"} unless ocsepp_pn32f_mediation_response_failure_total {namespace="sepp-namespace",http_status="504 GATEWAY_TIMEOUT"} offset 10m)</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4010
Metric Used	ocsepp_pn32f_mediation_response_failure
Resolution	This alert will be raised when Mediation microservice is not accessible. Possible Resolution: 1. Check if the Mediation microservice pod is up. 2. Check if Mediation Service Name and servicePort number is correct.

5.3.2.8 Overload Control Alerts

5.3.2.8.1 SEPPServiceOverloadWarn

Table 5-343 SEPPServiceOverloadWarn

Field	Details
Trigger Condition	CPU memory of pn32f-svc more than 65%
Severity	Warning

Table 5-343 (Cont.) SEPPServiceOverloadWarn

Field	Details
Alert details provided	Summary Backend service is in overload with load level in warning state Expression <code>service_resource_overload_level{namespace="sepp-namespace"} == 1</code>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7007
Metric Used	service_resource_overload_level
Resolution	The alert will be cleared when CPU Memory for backend-svc goes below 65%. Possible resolutions: 1. Users can configure threshold levels for CPU and memory usage. When usage reaches the defined limits, data is discarded based on the configured thresholds. If the discarded data differs from what is expected, please verify the threshold configurations. 2. To monitor current CPU and memory usage, check the statistics on Grafana or Prometheus. 3. For configuration-related details and other information, refer to the 'Overload Control Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.8.2 SEPPServiceOverloadMinor

Table 5-344 SEPPServiceOverloadMinor

Field	Details
Trigger Condition	CPU memory of pn32f-svc more than 70%
Severity	Minor
Alert details provided	Summary Backend service is in overload with load level in minor state Expression <code>service_resource_overload_level{namespace="sepp-namespace"} == 2</code>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7008
Metric Used	service_resource_overload_level

Table 5-344 (Cont.) SEPPServiceOverloadMinor

Field	Details
Resolution	The alert will be cleared when CPU Memory for backend-svc goes below 70%. Possible resolutions: 1. Users can configure threshold levels for CPU and memory usage. When usage reaches the defined limits, data is discarded based on the configured thresholds. If the discarded data differs from what is expected, please verify the threshold configurations. 2. To monitor current CPU and memory usage, check the statistics on Grafana or Prometheus. 3. For configuration-related details and other information, refer to the 'Overload Control Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.8.3 SEPPServiceOverloadMajor

Table 5-345 SEPPServiceOverloadMajor

Field	Details
Trigger Condition	CPU memory of pn32f-svc more than 80%
Severity	Major
Alert details provided	Summary Backend service is in overload with load level in major state Expression <pre>service_resource_overload_level{namespace="sepp-namespace"} == 3</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7009
Metric Used	service_resource_overload_level

Table 5-345 (Cont.) SEPPServiceOverloadMajor

Field	Details
Resolution	The alert will be cleared when CPU Memory for backend-svc goes below 80%. Possible resolutions: 1. Users can configure threshold levels for CPU and memory usage. When usage reaches the defined limits, data is discarded based on the configured thresholds. If the discarded data differs from what is expected, please verify the threshold configurations. 2. To monitor current CPU and memory usage, check the statistics on Grafana or Prometheus. 3. For configuration-related details and other information, refer to the 'Overload Control Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.8.4 SEPPServiceOverloadCritical

Table 5-346 SEPPServiceOverloadCritical

Field	Details
Trigger Condition	CPU memory of pn32f-svc more than 90%
Severity	Critical
Alert details provided	Summary Backend service is in overload with load level in critical state Expression <pre>service_resource_overload_level{namespace="sepp-namespace"} == 4</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7010
Metric Used	service_resource_overload_level

Table 5-346 (Cont.) SEPPServiceOverloadCritical

Field	Details
Resolution	The alert will be cleared when CPU Memory for backend-svc goes below 90%. Possible resolutions: 1. Users can configure threshold levels for CPU and memory usage. When usage reaches the defined limits, data is discarded based on the configured thresholds. If the discarded data differs from what is expected, please verify the threshold configurations. 2. To monitor current CPU and memory usage, check the statistics on Grafana or Prometheus. 3. For configuration-related details and other information, refer to the 'Overload Control Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.9 Hosted SEPP Alerts

5.3.2.9.1 SEPPpN32fHSRoutingFailureAlert

Table 5-347 SEPPpN32fHSRoutingFailureAlert

Field	Details
Trigger Condition	When the routing failure rate at Pn32f service is greater than 20 percentage.
Severity	Major
Alert details provided	Summary namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Routing has failed due to {{ \$labels.http_error_message }} Expression: <pre>((sum by(namespace, app, nf_instance_id, pod) (ocsepp_allowed_p_rss_routing_failure_total{namespace="sepp-namespace"})) / (sum by(namespace, app, nf_instance_id, pod) (ocsepp_pn32f_requests_total{namespace="sepp-namespace"}))) > 0.2</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4013
Metric Used	ocsepp_allowed_p_rss_routing_failure_total , ocsepp_pn32f_requests_total

Table 5-347 (Cont.) SEPPn32fHSRoutingFailureAlert

Field	Details
Resolution	The alert gets automatically cleared when the failure rate at pn32f microservice goes below 20 percent. Possible resolutions: 1. If a request gets rejected due to Remote SEPP Set not being present in allowed list, add the corresponding entry for the same. 2. For further details, refer 'Hosted SEPP' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.9.2 SEPPCn32fHSRoutingFailureAlertMinor

Table 5-348 SEPPCn32fHSRoutingFailureAlertMinor

Field	Details
Trigger Condition	When the routing failure rate at Cn32f service is greater than 50 percentage.
Severity	Minor
Alert details provided	Summary namespace: {{ \$labels.namespace }}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Routing has failed due to {{ \$labels.http_error_message }} for {{ \$labels.sourceRss }} Expression: <pre>((sum by(namespace, app, nf_instance_id, pod, sourceRss) (ocsepp_allowed_p_rss_routing_failure_total{namespace="sepp-namespace"})) / (sum by(namespace, app, nf_instance_id, pod, sourceRss) (ocsepp_cn32f_requests_total{namespace="sepp-namespace"}))) > 0.5</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4014
Metric Used	ocsepp_allowed_p_rss_routing_failure_total , ocsepp_cn32f_requests_total
Resolution	The alert gets automatically cleared when the failure rate at cn32f microservice goes below 50 percent. Possible resolutions: 1. If a request gets rejected due to Remote SEPP Set not being present in allowed list, add the corresponding entry for the same. 2. For further details, refer 'Hosted SEPP' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.9.3 SEPPCn32fHSRoutingFailureAlertMajor

Table 5-349 SEPPCn32fHSRoutingFailureAlertMajor

Field	Details
Trigger Condition	When the routing failure rate at Cn32f service is greater than 60 percentage.
Severity	Major
Alert details provided	Summary namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Routing has failed due to {{ \$labels.http_error_message }} for {{ \$labels.sourceRss }} Expression: <pre>((sum by(namespace, app, nf_instance_id, pod, sourceRss) (ocsepp_allowed_p_rss_routing_failure_total{namespace="sepp-namespace"})) / (sum by(namespace, app, nf_instance_id, pod, sourceRss) (ocsepp_cn32f_requests_total{namespace="sepp-namespace"}))) > 0.6</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4015
Metric Used	ocsepp_allowed_p_rss_routing_failure_total, ocsepp_cn32f_requests_total
Resolution	The alert gets automatically cleared when the failure rate at cn32f microservice goes below 60 percent. Possible resolutions: 1. If a request gets rejected due to Remote SEPP Set not being present in allowed list, add the corresponding entry for the same. 2. For further details, refer 'Hosted SEPP' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.9.4 SEPPCn32fHSRoutingFailureAlertCritical

Table 5-350 SEPPCn32fHSRoutingFailureAlertCritical

Field	Details
Trigger Condition	When the routing failure rate at Cn32f service is greater than 65 percentage.
Severity	Critical

Table 5-350 (Cont.) SEPPCn32fHSRoutingFailureAlertCritical

Field	Details
Alert details provided	Summary namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Routing has failed due to {{ \$labels.http_error_message }} for {{ \$labels.sourceRss }} Expression: <pre>((sum by(namespace, app, nf_instance_id, pod, sourceRss) (ocsepp_allowed_p_rss_routing_failure_total{namespace="sepp-namespace"})) / (sum by(namespace, app, nf_instance_id, pod, sourceRss) (ocsepp_cn32f_requests_total{namespace="sepp-namespace"}))) > 0.65</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4016
Metric Used	ocsepp_allowed_p_rss_routing_failure_total, ocsepp_cn32f_requests_total
Resolution	The alert gets automatically cleared when the failure rate at cn32f microservice goes below 65 percent. Possible resolutions: 1. If a request gets rejected due to Remote SEPP Set not being present in allowed list, add the corresponding entry for the same. 2. For further details, refer 'Hosted SEPP' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.9.5 SEPPCn32fHSRoutingFailureAlertWarning

Table 5-351 SEPPCn32fHSRoutingFailureAlertWarning

Field	Details
Trigger Condition	When the routing failure rate at Cn32f service is greater than 25 percentage.
Severity	Warning

Table 5-351 (Cont.) SEPPCn32fHSRoutingFailureAlertWarning

Field	Details
Alert details provided	Summary namespace: {{ \$labels.namespace }}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Routing has failed due to {{ \$labels.http_error_message }} for {{ \$labels.sourceRss }} Expression: <pre>((sum by(namespace, app, nf_instance_id, pod, sourceRss) (ocsepp_allowed_p_rss_routing_failure_total{namespace="sepp-namespace"})) / (sum by(namespace, app, nf_instance_id, pod, sourceRss) (ocsepp_cn32f_requests_total{namespace="sepp-namespace"}))) > 0.25</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4017
Metric Used	ocsepp_allowed_p_rss_routing_failure_total, ocsepp_cn32f_requests_total
Resolution	The alert gets automatically cleared when the failure rate at cn32f microservice goes below 25 percent. Possible resolutions: 1. If a request gets rejected due to Remote SEPP Set not being present in allowed list, add the corresponding entry for the same. 2. For further details, refer 'Hosted SEPP' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.10 SEPP Message Feed Alerts

5.3.2.10.1 DDUnreachableFromN32IGW

Table 5-352 DDUnreachableFromN32IGW

Field	Details
Trigger Condition	This alarm is raised when Data Director is not reachable from N32 Ingress Gateway.
Severity	major

Table 5-352 (Cont.) DDUnreachableFromN32IGW

Field	Details
Alert details provided	Summary DD Unreachable from N32 Ingress Gateway Expression: <pre>(oc_ingressgateway_dd_unreachable{namespace="sepp-namespace",app="n32-ingress-gateway"} == 1)</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4018
Metric Used	oc_ingressgateway_dd_unreachable
Resolution	Alert gets cleared automatically when the connection with Data Director is established. Possible Resolutions: 1. Check whether Kafka broker pods are up and running. 2. Kafka.bootstrapAddress parameter must be set to correct listener IP and port. 3. All the values must be checked for DD configuration in ocsepp_custom_values_<version>.yaml file, as this is a Helm based feature. 4. For more details and configuration related issues, refer to 'Message Feed Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.10.2 DDUnreachableFromPLMNIGW

Table 5-353 DDUnreachableFromPLMNIGW

Field	Details
Trigger Condition	This alarm is raised when Data Director is not reachable from PLMN Ingress Gateway.
Severity	major
Alert details provided	Summary DD Unreachable from PLMN Ingress Gateway Expression: <pre>(oc_ingressgateway_dd_unreachable{namespace="sepp-namespace",app="plmn-ingress-gateway"} == 1)</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4019
Metric Used	oc_ingressgateway_dd_unreachable

Table 5-353 (Cont.) DDUnreachableFromPLMNIGW

Field	Details
Resolution	Alert gets cleared automatically when the connection with Data Director is established. Possible Resolutions: 1. Check whether Kafka broker pods are up and running. 2. Kafka.bootstrapAddress parameter must be set to correct listener IP and port. 3. All the values must be checked for DD configuration in ocsepp_custom_values_<version>.yaml file, as this is a Helm based feature. 4. For more details and configuration related issues, refer to 'Message Feed Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.10.3 DDUnreachableFromN32EGW

Table 5-354 DDUnreachableFromN32EGW

Field	Details
Trigger Condition	This alarm is raised when Data Director is not reachable from N32 Egress Gateway.
Severity	major
Alert details provided	Summary DD Unreachable from N32 Egress Gateway Expression: <pre>(oc_egressgateway_dd_unreachable{namespace="sepp-namespace",app="n32-egress-gateway"} == 1)</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4020
Metric Used	oc_egressgateway_dd_unreachable
Resolution	Alert gets cleared automatically when the connection with Data Director is established. Possible Resolutions: 1. Check whether Kafka broker pods are up and running. 2. Kafka.bootstrapAddress parameter must be set to correct listener IP and port. 3. All the values must be checked for DD configuration in ocsepp_custom_values_<version>.yaml file, as this is a Helm based feature. 4. For more details and configuration related issues, refer to 'Message Feed Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.10.4 DDUnreachableFromPLMNEGW

Table 5-355 DDUnreachableFromPLMNEGW

Field	Details
Trigger Condition	This alarm is raised when Data Director is not reachable from PLMN Egress Gateway.
Severity	major
Alert details provided	Summary DD Unreachable from PLMN Egress Gateway Expression: <pre>(oc_egressgateway_dd_unreachable{namespace="sepp-namespace",app="plmn-egress-gateway"} == 1)</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4021
Metric Used	oc_egressgateway_dd_unreachable
Resolution	Alert gets cleared automatically when the connection with Data Director is established. Possible Resolutions: 1. Check whether Kafka broker pods are up and running. 2. <code>Kafka.bootstrapAddress</code> parameter must be set to correct listener IP and port. 3. All the values must be checked for DD configuration in <code>ocsepp_custom_values_<version>.yaml</code> file, as this is a Helm based feature. 4. For more details and configuration related issues, refer to 'Message Feed Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.11 Steering of Roaming (SOR) Alerts

5.3.2.11.1 SEPPN32fSORFailureAlertPercent30to40

Table 5-356 SEPPN32fSORFailureAlertPercent30to40

Field	Details
Trigger Condition	30% to 40% of SOR traffic results in failure.
Severity	Minor

Table 5-356 (Cont.) SEPPn32fSORFailureAlertPercent30to40

Field	Details
Alert details provided	Summary: 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{ . first value humanizeTimestamp }}{ end }' Expression: <pre>sum(rate(ocsepp_pn32f_sor_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app)/sum(rate(ocsepp_pn32f_sor_requests_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app)>=0.3 and sum(rate(ocsepp_pn32f_sor_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app)/sum(rate(ocsepp_pn32f_sor_requests_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app)<0.4</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4022
Metric Used	ocsepp_pn32f_sor_failure_total and ocsepp_pn32f_sor_requests_total
Resolution	This alert will be raised when the percentage failure of SOR responses is in the range 30%-40%, in the sample collected in last 2 min. Possible Resolutions : 1. Check the below headers in the response coming from SOR server. If any of these is missing, it will cause SOR Failure: a. Server Header b. Location Header 2. Check if the redirection code (3xx) received from SOR should be the same as the one configured through CNC Console. This code can be viewed in the metric ocsepp_pn32f_sor_failure_total. 3. Check if the SOR Server is sending the response code 5xx and whether the code is not configured through CNC Console or retry to Producer NF is disabled. This code can be viewed in the metric ocsepp_pn32f_sor_failure_total. 4. Check if any client error(4xx) is coming while connecting to SoR. This code can be viewed in the metric ocsepp_pn32f_sor_failure_total.

5.3.2.11.2 SEPPn32fSORFailureAlertPercent40to50

Table 5-357 SEPPn32fSORFailureAlertPercent40to50

Field	Details
Trigger Condition	40% to 50% of SOR traffic results in failure.
Severity	Major
Alert details provided	Summary: 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{ . first value humanizeTimestamp }}{ end }' Expression: <pre>sum(rate(ocsepp_pn32f_sor_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app)/sum(rate(ocsepp_pn32f_sor_requests_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app)>=0.4 and sum(rate(ocsepp_pn32f_sor_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app)/sum(rate(ocsepp_pn32f_sor_requests_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app)<0.5</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4023
Metric Used	ocsepp_pn32f_sor_failure_total and ocsepp_pn32f_sor_requests_total
Resolution	This alert will be raised when the percentage failure of SOR responses is in the range 40%-50%, in the sample collected in last 2 min. Possible Resolutions: 1. Check the below headers in the response coming from SoR server, if any of these is missing, it will cause SOR Failure: a. Server Header b. Location Header 2. Check if the redirection code (3xx) received from SOR should be same as one configured through CNC Console. This code can be viewed in the metricocsepp_pn32f_sor_failure_total. 3. Check if SOR Server is sending response code 5xx and the code is not configured through CNC Console or Retry to Producer NF is disabled. This code can be viewed in the metricocsepp_pn32f_sor_failure_total. 4. Check if any client error (4xx) is coming while connecting to SOR. This code can be viewed in the metricocsepp_pn32f_sor_failure_total.

5.3.2.11.3 SEPPn32fSORFailureAlertPercentAbove50

Table 5-358 SEPPn32fSORFailureAlertPercentAbove50

Field	Details
Trigger Condition	50% of SOR traffic results in failure
Severity	Critical
Alert details provided	Summary: 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}' Expression: <pre>sum(rate(ocsepp_pn32f_sor_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app)/ sum(rate(ocsepp_pn32f_sor_requests_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app)>=0.5</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4024
Metric Used	ocsepp_pn32f_sor_failure_total and ocsepp_pn32f_sor_requests_total
Resolution	This alert will be raised when the percentage failure of SOR responses is above 50%, in the sample collected in last 2 min. Possible Resolutions: 1. Check the below headers in the response coming from SOR server, if any of these is missing, it will cause SOR Failure: a. Server Header b. Location Header 2. Check if the redirection code(3xx) received from SOR should be same as one configured via CNC Console. This code can be viewed in the metricocsepp_pn32f_sor_failure_total. 3. Check if SOR Server is sending response code 5xx and the code is not configured through CNC Console or retry to Producer NF is disabled. This code can be viewed in the metricocsepp_pn32f_sor_failure_total. 4. Check if any client error(4xx) is coming while connecting to SOR. This code can be viewed in the metricocsepp_pn32f_sor_failure_total.

5.3.2.11.4 SEPPn32fSORTimeoutFailureAlert

Table 5-359 SEPPn32fSORTimeoutFailureAlert

Field	Details
Trigger Condition	Increase of more than five timeout errors in last two minutes for SOR.
Severity	critical
Alert details provided	Summary: 'namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }}{ . first value humanizeTimestamp }}{ end }' Expression: <pre> idelta(ocsepp_pn32f_sor_timeout_failure_total{namespace="sepp-namespace"}[2m]) > 5 or (ocsepp_pn32f_sor_timeout_failure_total{namespace="sepp-namespace"} unless ocsepp_pn32f_sor_timeout_failure_total{namespace="sepp-namespace"} offset 2m) </pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4025
Metric Used	ocsepp_pn32f_sor_timeout_failure_total
Resolution	This alert will be raised when the response received from SOR Server suggests that server is either down or unreachable for more than five error counts in the sample collected in last two minutes. Possible Resolutions : 1. Check and fix if the SOR server is unreachable. 2. Check and fix if the configuration made through CNC Console has wrong values for server. Check if the FQDN and port configured are correct. 3. The scheme selected must be supported by SOR server.

5.3.2.12 Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Alerts

5.3.2.12.1 IngressRssRateLimitPerRSSMessageDropAbovePointOnePercent

Table 5-360 IngressRssRateLimitPerRSSMessageDropAbovePointOnePercent

Field	Details
Trigger Condition	If a request has to be dropped when all the tokens in the bucket are exhausted and drop rate per RSS is detected above 0.1 percent of total transactions of that RSS, this metric will be pegged and corresponding alert will be raised.
Severity	Warning
Alert Details Provided	Summary: Ingress RSS Based Rate Limiting Message Drop Rate per RSS detected above 0.1 Percent of Total Transactions of that RSS. Expression: <pre>sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,Remote_SEPP_Set)/ sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,Remote_SEPP_Set) *100 >= 0.1 < 10</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7011
Metric Name	oc_ingressgateway_rss_ratelimit_total
Resolution	The alerts gets cleared when the drop rate per RSS is detected below 0.1 percent of total transactions. Possible resolutions: 1. In the CNC Console GUI, navigate to SEPP and then click Rate limiting and then Ingress Rate Limiting. The Remote SEPP Set appears underneath, click Remote SEPP Set, the Options appears. Check the header configured in Originating Network ID Header parameter. 2. For all configuration related and additional information, refer 'Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.12.2 IngressRssRateLimitPerRSSMessageDropAbove10Percent

Table 5-361 IngressRssRateLimitPerRSSMessageDropAbove10Percent

Field	Details
Trigger Condition	If a request has to be dropped when all the tokens in the bucket are exhausted and drop rate per RSS is detected above 10 percent of total transactions of that RSS, this metric will be pegged and corresponding alert will be raised.

Table 5-361 (Cont.) IngressRssRateLimitPerRSSMessageDropAbove10Percent

Field	Details
Severity	Minor
Alert Details Provided	Summary: Ingress RSS Based Rate Limiting Message Drop Rate per RSS detected above 10 Percent of Total Transactions of that RSS. Expression: <pre>sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,Remote_SEPP_Set)/ sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,Remote_SEPP_Set) *100 >= 10 < 25</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7012
Metric Name	oc_ingressgateway_rss_ratelimit_total
Resolution	The alerts gets cleared when the drop rate per RSS is detected below 10 percent of total transactions. Possible resolutions: 1. In the CNC Console GUI, navigate to SEPP and then click Rate limiting and then Ingress Rate Limiting. The Remote SEPP Set appears underneath, click Remote SEPP Set, the Options appears. Check the header configured in Originating Network ID Header parameter. 2. For all configuration related and additional information, refer 'Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.12.3 IngressRssRateLimitPerRSSMessageDropAbove25Percent

Table 5-362 IngressRssRateLimitPerRSSMessageDropAbove25Percent

Field	Details
Trigger Condition	If a request has to be dropped when all the tokens in the bucket are exhausted and drop rate per RSS is detected above 25 percent of total transactions of that RSS, this metric will be pegged and corresponding alert will be raised.
Severity	Major

Table 5-362 (Cont.) IngressRssRateLimitPerRSSMessageDropAbove25Percent

Field	Details
Alert Details Provided	Summary: Ingress RSS Based Rate Limiting Message Drop Rate per RSS detected above 25 Percent of Total Transactions of that RSS Expression: <pre>sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,Remote_SEPP_Set)/ sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,Remote_SEPP_Set) *100 >= 25 < 50</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7013
Metric Name	oc_ingressgateway_rss_ratelimit_total
Resolution	The alerts gets cleared when the drop rate per RSS is detected below 25 percent of total transaction. Possible resolutions: 1. In the CNC Console GUI, navigate to SEPP and then click Rate limiting and then Ingress Rate Limiting. The Remote SEPP Set appears underneath, click Remote SEPP Set, the Options appears. Check the header configured in Originating Network ID Header parameter. 2. For all configuration related and additional information, refer 'Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.12.4 IngressRssRateLimitPerRSSMessageDropAbove50Percent

Table 5-363 IngressRssRateLimitPerRSSMessageDropAbove50Percent

Field	Details
Trigger Condition	If a request has to be dropped when all the tokens in the bucket are exhausted and drop rate per RSS is detected above 50 percent of total transactions of that RSS, this metric will be pegged and corresponding alert will be raised.
Severity	Critical

Table 5-363 (Cont.) IngressRssRateLimitPerRSSMessageDropAbove50Percent

Field	Details
Alert Details Provided	Summary: Ingress RSS Based Rate Limiting Message Drop Rate per RSS detected above 50 Percent of Total Transactions of that RSS. Expression: <pre>sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,Remote_SEPP_Set)/ sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,Remote_SEPP_Set) *100 >= 50</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7014
Metric Name	oc_ingressgateway_rss_ratelimit_total
Resolution	The alerts gets cleared when the drop rate per RSS is detected below 50 percent of total transactions. Possible resolutions: 1. In the CNC Console GUI, navigate to SEPP and then click Rate limiting and then Ingress Rate Limiting. The Remote SEPP Set appears underneath, click Remote SEPP Set, the Options appears. Check the header configured in Originating Network ID Header parameter. 2. For all configuration related and additional information, refer 'Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.12.5 IngressRssRateLimitMessageDropAbovePointOnePercent

Table 5-364 IngressRssRateLimitMessageDropAbovePointOnePercent

Field	Details
Trigger Condition	If a request has to be dropped when all the tokens in the bucket are exhausted and drop rate is detected above 0.1 percent of total transactions, this metric will be pegged and corresponding alert will be raised.
Severity	Warning

Table 5-364 (Cont.) IngressRssRateLimitMessageDropAbovePointOnePercent

Field	Details
Alert Details Provided	Summary: Ingress RSS Based Rate Limiting Message Drop Rate detected above 0.1 Percent of Total Transaction Expression: <pre>sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,app)/ sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,app) *100 >= 0.1 < 1</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7015
Metric Name	oc_ingressgateway_rss_ratelimit_total
Resolution	The alerts gets cleared when the drop rate is detected below 0.1 percent of total transactions. Possible resolutions: 1. In the CNC Console GUI, navigate to SEPP and then click Rate limiting and then Ingress Rate Limiting. The Remote SEPP Set appears underneath, click Remote SEPP Set, the Options appears. Check the header configured in Originating Network ID Header parameter. 2. For all configuration related and additional information, refer 'Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.12.6 IngressRssRateLimitMessageDropAboveOnePercent

Table 5-365 IngressRssRateLimitMessageDropAboveOnePercent

Field	Details
Trigger Condition	If a request has to be dropped when all the tokens in the bucket are exhausted and drop rate is detected above 1 percent of total transactions, this metric will be pegged and corresponding alert will be raised.
Severity	Warning

Table 5-365 (Cont.) IngressRssRateLimitMessageDropAboveOnePercent

Field	Details
Alert Details Provided	Summary: Ingress RSS Based Rate Limiting Message Drop Rate detected above 1 Percent of Total Transactions Expression: <pre>sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,app)/ sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,app) *100 >= 1 < 10</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7016
Metric Name	oc_ingressgateway_rss_ratelimit_total
Resolution	The alerts gets cleared when the drop rate is detected below 1 percent of total transactions. Possible resolutions: 1. In the CNC Console GUI, navigate to SEPP and then click Rate limiting and then Ingress Rate Limiting. The Remote SEPP Set appears underneath, click Remote SEPP Set, the Options appears. Check the header configured in Originating Network ID Header parameter. 2. For all configuration related and additional information, refer 'Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.12.7 IngressRssRateLimitMessageDropAbove10Percent

Table 5-366 IngressRssRateLimitMessageDropAbove10Percent

Field	Details
Trigger Condition	If a request has to be dropped when all the tokens in the bucket are exhausted and drop rate is detected above 10 percent of total transactions, this metric will be pegged and corresponding alert will be raised.
Severity	Minor

Table 5-366 (Cont.) IngressRssRateLimitMessageDropAbove10Percent

Field	Details
Alert Details Provided	Summary: Ingress RSS Based Rate Limiting Message Drop Rate detected above 10 Percent of Total Transactions. Expression: <pre>sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,app)/ sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,app) *100 >= 10 < 25</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7017
Metric Name	oc_ingressgateway_rss_ratelimit_total
Resolution	The alerts gets cleared when the drop rate is detected below 10 percent of total transactions. Possible resolutions: 1. In the CNC Console GUI, navigate to SEPP and then click Rate limiting and then Ingress Rate Limiting. The Remote SEPP Set appears underneath, click Remote SEPP Set, the Options appears. Check the header configured in Originating Network ID Header parameter. 2. For all configuration related and additional information, refer 'Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.12.8 IngressRssRateLimitMessageDropAbove25Percent

Table 5-367 IngressRssRateLimitMessageDropAbove25Percent

Field	Details
Trigger Condition	If a request has to be dropped when all the tokens in the bucket are exhausted and drop rate is detected above 25 percent of total transactions, this metric will be pegged and corresponding alert will be raised.
Severity	Major

Table 5-367 (Cont.) IngressRssRateLimitMessageDropAbove25Percent

Field	Details
Alert Details Provided	Summary: Ingress RSS Based Rate Limiting Message Drop Rate detected above 25 Percent of Total Transactions Expression: <pre>sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,app)/ sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,app) *100 >= 25 < 50</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7018
Metric Name	oc_ingressgateway_rss_ratelimit_total
Resolution	The alerts gets cleared when the drop rate is detected below 25 percent of total transactions. Possible resolutions: 1. In the CNC Console GUI, navigate to SEPP and then click Rate limiting and then Ingress Rate Limiting. The Remote SEPP Set appears underneath, click Remote SEPP Set, the Options appears. Check the header configured in Originating Network ID Header parameter. 2. For all configuration related and additional information, refer 'Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.12.9 IngressRssRateLimitMessageDropAbove50Percent

Table 5-368 IngressRssRateLimitMessageDropAbove50Percent

Field	Details
Trigger Condition	If a request has to be dropped when all the tokens in the bucket are exhausted and drop rate is detected above 50 percent of total transactions, this metric will be pegged and corresponding alert will be raised.
Severity	Critical

Table 5-368 (Cont.) IngressRssRateLimitMessageDropAbove50Percent

Field	Details
Alert Details Provided	Summary: Ingress RSS Based Rate Limiting Message Drop Rate detected above 50 Percent of Total Transactions. Expression: <pre>sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace",Status="dropped"}[5m])) by (namespace,app)/ sum(rate(oc_ingressgateway_rss_ratelimit_total{namespace="sepp-namespace"}[5m])) by (namespace,app) *100 >= 50</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.7019
Metric Name	oc_ingressgateway_rss_ratelimit_total
Resolution	The alerts gets cleared when the drop rate is detected below 50 percent of total transactions. Possible resolutions: 1. In the CNC Console GUI, navigate to SEPP and then click Rate limiting and then Ingress Rate Limiting. The Remote SEPP Set appears underneath, click Remote SEPP Set, the Options appears. Check the header configured in Originating Network ID Header parameter. 2. For all configuration related and additional information, refer 'Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.13 Cat-0 SBI Message Schema Validation Alerts

5.3.2.13.1 SEPPN32fMessageValidationOnHeaderFailureMinorAlert

Table 5-369 SEPPN32fMessageValidationOnHeaderFailureMinorAlert

Field	Details
Trigger Condition	Message validation failed for request query parameters for 40 % of requests (on which message validation was applied) in last 2 minutes.
Severity	minor

Table 5-369 (Cont.) SEPPN32fMessageValidationOnHeaderFailureMinorAlert

Field	Details
Alert Details Provided	Summary: Namespace: {{ \$labels.kubernetes_namespace }}, Podname: {{ \$labels.kubernetes_pod_name }}, App: {{ \$labels.app }}, Nfinstanceid: {{ \$labels.nfInstanceId }} Expression: (sum(rate(ocsepp_message_validation_on_header_failure_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id))*100 >= 40 < 60
OID	1.3.6.1.4.1.323.5.3.46.1.2.4026
Metric Used	ocsepp_message_validation_on_header_failure_total

Table 5-369 (Cont.) SEPPN32fMessageValidationOnHeaderFailureMinorAlert

Field	Details
Resolution	The alerts gets cleared when the count is not between 40 to 60. Possible Resolutions: 1. Check Logs or Metrics: Review the following metrics for message validation failures: • ocsepp_message_validation_on_body_failure • ocsepp_message_validation_on_header_failure 2. To identify the Failing Resource URI and HTTP Method, do the following: • For request body validation failures, search for the text: "Message validation failed for request body for request" • For query parameter validation failures, search for: "Message validation failed for request query parameter(s) for request" • For more detailed information about logs, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 3. In CNC Console GUI, navigate to SEPP and select Security Countermeasure from the left-hand menu. • Click Cat 0 - SBI Message Schema Validation to open the Message Validation List. • Search for the relevant resource URI to retrieve the corresponding schema. • Compare the request body or query parameters against the schema to ensure the request complies with the schema. If necessary, update the schema to reflect the correct structure.

5.3.2.13.2 SEPPN32fMessageValidationOnHeaderFailureMajorAlert

Table 5-370 SEPPN32fMessageValidationOnHeaderFailureMajorAlert

Field	Description
Trigger Condition	Message validation failed for request query parameters for 60 % of requests (on which message validation was applied) in last 2 minutes.
Severity	major

Table 5-370 (Cont.) SEPPN32fMessageValidationOnHeaderFailureMajorAlert

Field	Description
Alert Details Provided	Summary: Namespace: {{ \$labels.kubernetes_namespace }}, Podname: {{ \$labels.kubernetes_pod_name }}, App: {{ \$labels.app }}, Nfinstanceid: {{ \$labels.nfinstanceid }} Expression: <pre>(sum(rate(ocsepp_message_validation_on_header_failure_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id))*100 >= 60 < 80</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4027
Metric Name	ocsepp_message_validation_on_header_failure_total

Table 5-370 (Cont.) SEPPN32fMessageValidationOnHeaderFailureMajorAlert

Field	Description
Resolution	The alerts gets cleared when the count is not between 60 to 80. Possible Resolutions: 1. Check Logs or Metrics: Review the following metrics for message validation failures: • ocsepp_message_validation_on_body_failure • ocsepp_message_validation_on_header_failure 2. To identify the Failing Resource URI and HTTP Method, do the following: • For request body validation failures, search for the text: "Message validation failed for request body for request" • For query parameter validation failures, search for: "Message validation failed for request query parameter(s) for request" • For more detailed information about logs, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 3. In CNC Console GUI, navigate to SEPP and select Security Countermeasure from the left-hand menu. • Click Cat 0 - SBI Message Schema Validation to open the Message Validation List. • Search for the relevant resource URI to retrieve the corresponding schema. • Compare the request body or query parameters against the schema to ensure the request complies with the schema. If necessary, update the schema to reflect the correct structure.

5.3.2.13.3 SEPPN32fMessageValidationOnHeaderFailureCriticalAlert

Table 5-371 SEPPN32fMessageValidationOnHeaderFailureCriticalAlert

Field	Description
Trigger Condition	Message validation failed for request query parameters for 80 % of requests (on which message validation was applied) in last 2 minutes.
Severity	critical

Table 5-371 (Cont.) SEPPN32fMessageValidationOnHeaderFailureCriticalAlert

Field	Description
Alert Details Provided	Summary: Namespace: {{ \$labels.kubernetes_namespace }}, Podname: {{ \$labels.kubernetes_pod_name }}, App: {{ \$labels.app }}, Nfinstanceid: {{ \$labels.nfInstanceId }} Expression: (sum(rate(ocsepp_message_validation_on_header_failure_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id))*100 >= 80
OID	1.3.6.1.4.1.323.5.3.46.1.2.4028
Metric Name	ocsepp_message_validation_on_header_failure_total

Table 5-371 (Cont.) SEPPN32fMessageValidationOnHeaderFailureCriticalAlert

Field	Description
Resolution	This alert is cleared when the count is less than 80. Possible Resolutions: 1. Check Logs or Metrics: Review the following metrics for message validation failures: • ocsepp_message_validation_on_body_failure • ocsepp_message_validation_on_header_failure 2. To identify the Failing Resource URI and HTTP Method, do the following: • For request body validation failures, search for the text: "Message validation failed for request body for request" • For query parameter validation failures, search for: "Message validation failed for request query parameter(s) for request" • For more detailed information about logs, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 3. In CNC Console GUI, navigate to SEPP and select Security Countermeasure from the left-hand menu. • Click Cat 0 - SBI Message Schema Validation to open the Message Validation List. • Search for the relevant resource URI to retrieve the corresponding schema. • Compare the request body or query parameters against the schema to ensure the request complies with the schema. If necessary, update the schema to reflect the correct structure.

5.3.2.13.4 SEPPN32fMessageValidationOnBodyFailureMinorAlert

Table 5-372 SEPPN32fMessageValidationOnBodyFailureMinorAlert

Field	Description
Trigger Condition	Message validation failed for request body for 40 % of requests (on which message validation was applied) in last 2 minutes.
Severity	minor

Table 5-372 (Cont.) SEPPN32fMessageValidationOnBodyFailureMinorAlert

Field	Description
Alert Details Provided	Summary: Namespace: {{ \$labels.kubernetes_namespace }}, Podname: {{ \$labels.kubernetes_pod_name }}, App: {{ \$labels.app }}, Nfinstanceid: {{ \$labels.nfInstanceId }} Expression: (sum(rate(ocsepp_message_validation_on_body_failure_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id))*100 >= 40 < 60
OID	1.3.6.1.4.1.323.5.3.46.1.2.4029
Metric Name	ocsepp_message_validation_on_body_failure_total

Table 5-372 (Cont.) SEPPN32fMessageValidationOnBodyFailureMinorAlert

Field	Description
Resolution	The alerts gets cleared when the count is not between 40 to 60. Possible Resolutions: 1. Check Logs or Metrics: Review the following metrics for message validation failures: • ocsepp_message_validation_on_body_failure • ocsepp_message_validation_on_header_failure 2. To identify the Failing Resource URI and HTTP Method, do the following: • For request body validation failures, search for the text: "Message validation failed for request body for request" • For query parameter validation failures, search for: "Message validation failed for request query parameter(s) for request" • For more detailed information about logs, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 3. In CNC Console GUI, navigate to SEPP and select Security Countermeasure from the left-hand menu. • Click Cat 0 - SBI Message Schema Validation to open the Message Validation List. • Search for the relevant resource URI to retrieve the corresponding schema. • Compare the request body or query parameters against the schema to ensure the request complies with the schema. If necessary, update the schema to reflect the correct structure.

5.3.2.13.5 SEPPN32fMessageValidationOnBodyFailureMajorAlert

Table 5-373 SEPPN32fMessageValidationOnBodyFailureMajorAlert

Field	Details
Trigger Condition	Message validation failed for request body for 60 % of requests (on which message validation was applied) in last 2 minutes.
Severity	major

Table 5-373 (Cont.) SEPPN32fMessageValidationOnBodyFailureMajorAlert

Field	Details
Alert Details Provided	Summary: Namespace: <pre>{{ \$labels.kubernetes_namespace }}</pre>, Podname: <pre>{{ \$labels.kubernetes_pod_name }}</pre>, App: <pre>{{ \$labels.app }}</pre>, Nfinstanceid: <pre>{{ \$labels.nfInstanceId }}</pre> Expression: <pre>(sum(rate(ocsepp_message_validation_on_body_failure_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id))*100 >= 60 < 80</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4030
Metric Name	ocsepp_message_validation_on_body_failure_total

Table 5-373 (Cont.) SEPPN32fMessageValidationOnBodyFailureMajorAlert

Field	Details
Resolution	The alerts gets cleared when the count is not between 60 to 80. Possible Resolutions: 1. Check Logs or Metrics: Review the following metrics for message validation failures: • ocsepp_message_validation_on_body_failure • ocsepp_message_validation_on_header_failure 2. To identify the Failing Resource URI and HTTP Method, do the following: • For request body validation failures, search for the text: "Message validation failed for request body for request" • For query parameter validation failures, search for: "Message validation failed for request query parameter(s) for request" • For more detailed information about logs, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 3. In CNC Console GUI, navigate to SEPP and select Security Countermeasure from the left-hand menu. • Click Cat 0 - SBI Message Schema Validation to open the Message Validation List. • Search for the relevant resource URI to retrieve the corresponding schema. • Compare the request body or query parameters against the schema to ensure the request complies with the schema. If necessary, update the schema to reflect the correct structure.

5.3.2.13.6 SEPPN32fMessageValidationOnBodyFailureCriticalAlert

Table 5-374 SEPPN32fMessageValidationOnBodyFailureCriticalAlert

Field	Details
Trigger Condition	Message validation failed for request body for 80 % of requests (on which message validation was applied) in last 2 minutes.
Severity	critical

Table 5-374 (Cont.) SEPPN32fMessageValidationOnBodyFailureCriticalAlert

Field	Details
Alert Details Provided	Summary: Namespace: {{ \$labels.kubernetes_namespace }}, Podname: {{ \$labels.kubernetes_pod_name }}, App: {{ \$labels.app }}, Nfinstanceid: {{ \$labels.nfInstanceId }} Expression: (sum(rate(ocsepp_message_validation_on_body_failure_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total{namespace="sepp-namespace"}[2m])) by (namespace, app, pod, nf_instance_id))*100 >= 80
OID	1.3.6.1.4.1.323.5.3.46.1.2.4031
Metric Name	ocsepp_message_validation_on_body_failure_total

Table 5-374 (Cont.) SEPPN32fMessageValidationOnBodyFailureCriticalAlert

Field	Details
Resolution	This alert is cleared when the count is less than 80. Possible Resolutions: 1. Check Logs or Metrics: Review the following metrics for message validation failures: • ocsepp_message_validation_on_body_failure • ocsepp_message_validation_on_header_failure 2. To identify the Failing Resource URI and HTTP Method, do the following: • For request body validation failures, search for the text: "Message validation failed for request body for request" • For query parameter validation failures, search for: "Message validation failed for request query parameter(s) for request" • For more detailed information about logs, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 3. In CNC Console GUI, navigate to SEPP and select Security Countermeasure from the left-hand menu. • Click Cat 0 - SBI Message Schema Validation to open the Message Validation List. • Search for the relevant resource URI to retrieve the corresponding schema. • Compare the request body or query parameters against the schema to ensure the request complies with the schema. If necessary, update the schema to reflect the correct structure.

5.3.2.14 Cat-1 Service API Validation Alerts

5.3.2.14.1 SEPPN32fServiceApiValidationFailureAlert

Table 5-375 SEPPN32fServiceApiValidationFailureAlert

Field	Details
Trigger Condition	Service API not in allowed list
Severity	Major

Table 5-375 (Cont.) SEPPN32fServiceApiValidationFailureAlert

Field	Details
Alert details provided	Summary N32f : Service API not in allowed list Expression: <pre>delta(ocsepp_security_service_api_failure_total{namespace="sepp-namespace"}[2m]) > 0 or (ocsepp_security_service_api_failure_total{namespace="sepp-namespace"} unless ocsepp_security_service_api_failure_total{namespace="sepp-namespace"} offset 2m)</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4005
Metric Used	ocsepp_security_service_api_failure_total
Resolution	This alert will be raised when there is difference of at least 1 between first and last data point in sample collected in last 2 minutes. Alert will be cleared after 2 minutes. Possible Resolutions: 1. Check the Resource URI + Method for which alert is raised. 2. Verify the error_msg using "ocsepp_security_service_api_failure_total" metric and KPI. 3. Fix or add configuration for the Resource URI + Method in Service API's and Allowed List.

5.3.2.15 Cat-1 NRF Service API Query Parameters Validation Alerts

5.3.2.15.1 seppN32fSrvApiQryPrmValFailAltWarn

Table 5-376 seppN32fSrvApiQryPrmValFailAltWarn

Field	Description
Trigger Condition	Triggered as a warning in case of a failure of Cat-1 NRF service API query parameters validation feature when the rate of failure goes above 5.
Severity	Warning

Table 5-376 (Cont.) seppN32fSvcApiQryPrmValFailAltWarn

Field	Description
Alert Details Provided	Summary namespace: {{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Service Api Query Param Validation has failed because {{ \$labels.cause }} Expression: <pre>sum(rate(ocsepp_security_service_api_query_param_validation_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app,pod,resource_uri) > 0 and sum(rate(ocsepp_security_service_api_query_param_validation_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app,pod,resource_uri) <= 5</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4067
Metric Name	ocsepp_security_service_api_query_param_validation_failure_total
Resolution	The alert gets automatically cleared when the rate of failure goes down below 5. Possible Resolutions: 1. Verify that both the Cat-1 NRF Service API Query Parameters Validation feature and the Cat-1 Service API Validation feature are enabled. For configuration-related issues, refer to 'Cat-1 NRF Service API Query Parameters Validation Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. Ensure that the same list-name is selected across all lists configured for query parameter validation. 3. Check the pair of Network Functions and the values included in the request. Make sure they match the configuration defined for this feature. 4. Confirm that the request is an NRF discovery request using the GET method.

5.3.2.15.2 seppN32fSrvApiQryPrmValFailAltMinor

Table 5-377 seppN32fSrvApiQryPrmValFailAltMinor

Field	Description
Trigger Condition	Triggered as a minor alert in case of a failure of Cat-1 NRF service API query parameters validation feature when the rate of failure goes between 5 to 10.
Severity	Minor
Alert Details Provided	Summary namespace: {{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{ . first value humanizeTimestamp }}{{ end }}: Service Api Query Param Validation has failed because {{ \$labels.cause }} Expression: <pre>sum(rate(ocsepp_security_service_api_query_param_validation_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app,pod,resource_uri) > 5 and sum(rate(ocsepp_security_service_api_query_param_validation_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app,pod,resource_uri) <= 10</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4068
Metric Name	ocsepp_security_service_api_query_param_validation_failure_total

Table 5-377 (Cont.) seppN32fSrvApiQryPrmValFailAltMinor

Field	Description
Resolution	The alert gets automatically cleared when the rate of failure goes down below 5 or over 10. Possible Resolutions: 1. Verify that both the Cat-1 NRF Service API Query Parameters Validation feature and the Cat-1 Service API Validation feature are enabled. For configuration-related issues, refer to 'Cat-1 NRF Service API Query Parameters Validation Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. Ensure that the same list-name is selected across all lists configured for query parameter validation. 3. Check the pair of Network Functions and the values included in the request. Make sure they match the configuration defined for this feature. 4. Confirm that the request is an NRF discovery request using the GET method.

5.3.2.15.3 seppN32fSrvApiQryPrmValFailAltMajor

Table 5-378 seppN32fSrvApiQryPrmValFailAltMajor

Field	Description
Trigger Condition	Triggered as a major alert in case of a failure of Cat-1 NRF service API query parameters validation feature when the rate of failure goes between 10 to 100.
Severity	Major

Table 5-378 (Cont.) seppN32fSvcApiQryPrmValFailAltMajor

Field	Description
Alert Details Provided	Summary namespace:{{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Service Api Query Param Validation has failed because {{ \$labels.cause }} Expression: <pre>sum(rate(ocsepp_security_service_api_query_param_validation_failure_total{ namespace="sepp-namespace"} [2m]))by(namespace,nf_instance_id,app ,pod,resource_uri) > 10 and sum(rate(ocsepp_security_service_api_query_param_validation_failure_total{ namespace="sepp-namespace"} [2m]))by(namespace,nf_instance_id,app ,pod,resource_uri) <= 100</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4069
Metric Name	ocsepp_security_service_api_query_param_validation_failure_total
Resolution	The alert gets automatically cleared when the rate of failure goes down below 10 or over 100. Possible Resolutions: 1. Verify that both the Cat-1 NRF Service API Query Parameters Validation feature and the Cat-1 Service API Validation feature are enabled. For configuration-related issues, refer to 'Cat-1 NRF Service API Query Parameters Validation Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. Ensure that the same list-name is selected across all lists configured for query parameter validation. 3. Check the pair of Network Functions and the values included in the request. Make sure they match the configuration defined for this feature. 4. Confirm that the request is an NRF discovery request using the GET method.

5.3.2.15.4 seppN32fSrvApiQryPrmValFailAltCritical

Table 5-379 seppN32fSrvApiQryPrmValFailAltCritical

Field	Description
Trigger Condition	Triggered as a critical alert in case of a failure of Cat-1 NRF service API query parameters validation feature when the rate of failure goes above 100.
Severity	Critical
Alert Details Provided	Summary namespace:{{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Service Api Query Param Validation has failed because {{ \$labels.cause }} Expression: <pre>sum(rate(ocsepp_security_service_api_query_param_validation_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app,pod,resource_uri) > 100</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4070
Metric Name	ocsepp_security_service_api_query_param_validation_failure_total
Resolution	The alert gets automatically cleared when the rate of failure goes down below 100. Possible Resolutions: 1. Verify that both the Cat-1 NRF Service API Query Parameters Validation feature and the Cat-1 Service API Validation feature are enabled. For configuration-related issues, refer to 'Cat-1 NRF Service API Query Parameters Validation Feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. Ensure that the same list-name is selected across all lists configured for query parameter validation. 3. Check the pair of Network Functions and the values included in the request. Make sure they match the configuration defined for this feature. 4. Confirm that the request is an NRF discovery request using the GET method.

5.3.2.16 Cat-2 Network ID Validation Alerts

5.3.2.16.1 SEPPN32fNetworkIDValidationHeaderFailureAlert

Table 5-380 SEPPN32fNetworkIDValidationHeaderFailureAlert

Field	Details
Trigger Condition	If Network ID Validation for Header fails, this metrics will be pegged and corresponding alert will be raised.
Severity	Major
Alert details provided	Summary: 'namespace: {{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }}{{ end }}: Network ID Validation has failed because {{ \$labels.cause }}' Expression: <pre>sum(increase(ocsepp_network_id_validation_header_failure_total{namespace="sepp-namespace"}[2m]) >0 or (ocsepp_network_id_validation_header_failure_total{namespace="sepp-namespace"} unless ocsepp_network_id_validation_header_failure_total{namespace="sepp-namespace"} offset 2m)) by (namespace, remote_sepp_name, nf_instance_id, peer_fqdn, plmn_identififer, app, resource_uri, pod) > 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4011
Metric Used	ocsepp_network_id_validation_header_failure_total
Resolution	The alerts gets cleared when the count goes below 0. Possible Resolutions: 1. Review CN32F logs for error codes SEPP-CN32FSEPP-ERROR-0013 or SEPP-CN32FSEPP-ERROR-0014 (indicating the incoming request was rejected with error code 406 by the CN32F microservice). 2. Similarly, check PN32F logs for error codes SEPP-PN32FSEPP-ERROR-0016 or SEPP-PN32FSEPP-ERROR-0017 (indicating the request was rejected with error code 406 by the PN32F microservice). 3. Ensure the correct regular expression is configured under Header/ Body IE in the Cat 2 – Network ID Validation section (located under the Security Countermeasure tab in SEPP). 4. Confirm that the PLMN ID specified in the request headers is included in the SEPP's configured PLMN ID List. 5. For issues related to invalid PLMN ID configurations in the body or headers, refer to 'Cat-2 Network ID Validation Feature' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.16.2 SEPPN32fNetworkIDValidationBodyIEFailureAlert

Table 5-381 SEPPN32fNetworkIDValidationBodyIEFailureAlert

Field	Details
Trigger Condition	If Network ID Validation for Body fails, this metrics will be pegged and corresponding alert will be raised.
Severity	Major
Alert details provided	Summary: 'namespace: {{ \$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}: Network ID Body Validation has failed because {{ \$labels.cause }}' Expression: <pre>sum(increase(ocsepp_network_id_validation_body_failure_total{namespace="sepp-namespace"}[2m]) >0 or (ocsepp_network_id_validation_body_failure_total{namespace="sepp-namespace"} unless ocsepp_network_id_validation_body_failure_total{namespace="sepp-namespace"} offset 2m)) by (namespace, remote_sepp_name, nf_instance_id, peer_fqdn, plmn_identifier, app, resource_uri, pod) > 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4012
Metric Used	ocsepp_network_id_validation_body_failure_total
Resolution	The alerts gets cleared when the count goes below 0. Possible Resolutions: 1. Review CN32F logs for error codes SEPP-CN32FSEPP-ERROR-0013 or SEPP-CN32FSEPP-ERROR-0014 (indicating the incoming request was rejected with error code 406 by the CN32F microservice). 2. Similarly, check PN32F logs for error codes SEPP-PN32FSEPP-ERROR-0016 or SEPP-PN32FSEPP-ERROR-0017 (indicating the request was rejected with error code 406 by the PN32F microservice). 3. Ensure the correct regular expression is configured under Header/ Body IE in the Cat 2 – Network ID Validation section (located under the Security Countermeasure tab in SEPP). 4. Confirm that the PLMN ID specified in the request headers is included in the SEPP's configured PLMN ID List. 5. For issues related to invalid PLMN ID configurations in the body or headers, refer to 'Cat-2 Network ID Validation Feature' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.17 Cat-3 Previous Location Check Alerts

5.3.2.17.1 SEPPN32fPreviousLocationCheckValidationFailureAlertMinor

Table 5-382 SEPPn32fPreviousLocationCheckValidationFailureAlertMinor

Field	Details
Trigger Condition	When previous location check validation failure error is detected between 30 to 40 Percent of total transactions, this alert will be raised.
Severity	Minor
Alert Details Provided	Summary namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }} Expression: <pre>sum(rate(ocsepp_previous_location_val idation_failure_total{namespace="sepp -namespace"} [2m]))by(namespace,nf_instance_id,app ,pod)/ sum(rate(ocsepp_previous_location_val idation_requests_total{namespace="sep p-namespace"} [2m]))by(namespace,nf_instance_id,app ,pod)>=0.3 and sum(rate(ocsepp_previous_location_val idation_failure_total{namespace="sepp -namespace"} [2m]))by(namespace,nf_instance_id,app ,pod)/ sum(rate(ocsepp_previous_location_val idation_requests_total{namespace="sep p-namespace"} [2m]))by(namespace,nf_instance_id,app ,pod)<0.4</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4032
Metric Name	ocsepp_previous_location_validation_failure_total

Table 5-382 (Cont.) SEPPN32fPreviousLocationCheckValidationFailureAlertMinor

Field	Details
Resolution	The alerts gets cleared when the previous location check validation failure error does not lie between 30 to 40 percent of total transactions. Possible Resolutions: - Look for the following error codes in the PN32F microservice logs: - SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-ERROR-0019 - SEPP-PN32FSEPP-ERROR-0018 - SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-EXCEPTION-0020 - Ensure that the correct regex is configured for UE ID and Serving Network ID under the Header or Body IE configuration within the Cat 3 – Previous Location Check section of the Security Countermeasure tab in the SEPP CNC Console. - Check that the MCC and MNC values (configured in either the header or body) match the serving network name. These values are returned in the UDR response, so also confirm that the UDR response is successful. - If Cat 3 – Previous Location Check is enabled, ensure the SUPI is present in the incoming request message. - Verify the following: - Confirm the UDR discovery procedure completes successfully. - Ensure the FQDN or IP of the UDR is reachable. - Verify the Coherence service is up and running. - Make sure the SUPI falls within the IMSI range defined in the UDR profile. - Ensure that DNS resolution is properly configured for the UDR discovery call, specifically for the pn32f-svc service used in subscription use cases.

5.3.2.17.2 SEPPN32fPreviousLocationCheckValidationFailureAlertMajor

Table 5-383 SEPPN32fPreviousLocationCheckValidationFailureAlertMajor

Field	Details
Trigger Condition	When previous location check validation failure error is detected between 40 to 50 Percent of total transactions, this alert will be raised.
Severity	Major

Table 5-383 (Cont.) SEPPn32fPreviousLocationCheckValidationFailureAlertMajor

Field	Details
Alert Details Provided	Summary <pre>namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}</pre> Expression: <pre>sum(rate(ocsepp_previous_location_val idation_failure_total{namespace="sepp -namespace"} [2m]))by(namespace,nf_instance_id,app ,pod)/ sum(rate(ocsepp_previous_location_val idation_requests_total{namespace="sep p-namespace"} [2m]))by(namespace,nf_instance_id,app ,pod)>=0.4 and sum(rate(ocsepp_previous_location_val idation_failure_total{namespace="sepp -namespace"} [2m]))by(namespace,nf_instance_id,app ,pod)/ sum(rate(ocsepp_previous_location_val idation_requests_total{namespace="sep p-namespace"} [2m]))by(namespace,nf_instance_id,app ,pod)<0.5</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4033
Metric Name	ocsepp_previous_location_validation_failure_total

Table 5-383 (Cont.) SEPPn32fPreviousLocationCheckValidationFailureAlertMajor

Field	Details
Resolution	The alerts gets cleared when the previous location check validation failure error does not lie between 40 to 50 percent of total transactions. Possible Resolutions: 1. Look for the following error codes in the PN32F microservice logs: • SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-ERROR-0019 • SEPP-PN32FSEPP-ERROR-0018 • SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-EXCEPTION-0020 2. Ensure that the correct regex is configured for UE ID and Serving Network ID under the Header or Body IE configuration within the Cat 3 – Previous Location Check section of the Security Countermeasure tab in the SEPP CNC Console. 3. Check that the MCC and MNC values (configured in either the header or body) match the serving network name. These values are returned in the UDR response, so also confirm that the UDR response is successful. 4. If Cat 3 – Previous Location Check is enabled, ensure the SUPI is present in the incoming request message. 5. Verify the following: • Confirm the UDR discovery procedure completes successfully. • Ensure the FQDN or IP of the UDR is reachable. • Verify the Coherence service is up and running. • Make sure the SUPI falls within the IMSI range defined in the UDR profile. 6. Ensure that DNS resolution is properly configured for the UDR discovery call, specifically for the pn32f-svc service used in subscription use cases.

5.3.2.17.3 sEPPn32fPrevLocChkValFailAlrtCritical

Table 5-384 sEPPn32fPrevLocChkValFailAlrtCritical

Field	Details
Trigger Condition	When previous location check validation failure error is detected above 50 Percent of total transactions, this alert will be raised.

Table 5-384 (Cont.) sEPPn32fPrevLocChkValFailAlrtCritical

Field	Details
Severity	Critical
Alert Details Provided	Summary <pre>namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}</pre> Expression: <pre>sum(rate(ocsepp_previous_location_val idation_failure_total{namespace="sepp -namespace"} [2m]))by(namespace,nf_instance_id,app ,pod)/ sum(rate(ocsepp_previous_location_val idation_requests_total{namespace="sep p-namespace"} [2m]))by(namespace,nf_instance_id,app ,pod)>=0.5</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4034
Metric Name	ocsepp_previous_location_validation_failure_total

Table 5-384 (Cont.) sEPPnPn32fPrevLocChkValFailAlrtCritical

Field	Details
Resolution	The alerts gets cleared when the previous location check validation failure error does not lie above 50 percent of total transactions. Possible Resolutions: 1. Look for the following error codes in the PN32F microservice logs: • SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-ERROR-0019 • SEPP-PN32FSEPP-ERROR-0018 • SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-EXCEPTION-0020 2. Ensure that the correct regex is configured for UE ID and Serving Network ID under the Header or Body IE configuration within the Cat 3 – Previous Location Check section of the Security Countermeasure tab in the SEPP CNC Console. 3. Check that the MCC and MNC values (configured in either the header or body) match the serving network name. These values are returned in the UDR response, so also confirm that the UDR response is successful. 4. If Cat 3 – Previous Location Check is enabled, ensure the SUPI is present in the incoming request message. 5. Verify the following: • Confirm the UDR discovery procedure completes successfully. • Ensure the FQDN or IP of the UDR is reachable. • Verify the Coherence service is up and running. • Make sure the SUPI falls within the IMSI range defined in the UDR profile. 6. Ensure that DNS resolution is properly configured for the UDR discovery call, specifically for the pn32f-svc service used in subscription use cases.

5.3.2.17.4 SEPPnPn32fPreviousLocationCheckExceptionFailureAlertMinor

Table 5-385 SEPPnPn32fPreviousLocationCheckExceptionFailureAlertMinor

Field	Details
Trigger Condition	When previous location check exception failure is detected between 30 to 40 Percent of total transactions, this alert will be raised.

Table 5-385 (Cont.) SEPPn32fPreviousLocationCheckExceptionFailureAlertMinor

Field	Details
Severity	Minor
Alert Details Provided	Summary namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }} Expression: <pre>sum(rate(ocsepp_previous_location_exception_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app,pod)/ sum(rate(ocsepp_previous_location_validation_requests_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app,pod)/ sum(rate(ocsepp_previous_location_exception_failure_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app,pod)/ sum(rate(ocsepp_previous_location_validation_requests_total{namespace="sepp-namespace"}[2m]))by(namespace,nf_instance_id,app,pod)<0.4</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4035
Metric Name	ocsepp_previous_location_exception_failure_total

Table 5-385 (Cont.) SEPPN32fPreviousLocationCheckExceptionFailureAlertMinor

Field	Details
Resolution	The alerts gets cleared when the previous location check exception failure does not lie between 30 to 40 percent of total transactions. Possible Resolutions: 1. Look for the following error codes in the PN32F microservice logs: • SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-ERROR-0019 • SEPP-PN32FSEPP-ERROR-0018 • SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-EXCEPTION-0020 2. Ensure that the correct regex is configured for UE ID and Serving Network ID under the Header or Body IE configuration within the Cat 3 – Previous Location Check section of the Security Countermeasure tab in the SEPP CNC Console. 3. Check that the MCC and MNC values (configured in either the header or body) match the serving network name. These values are returned in the UDR response, so also confirm that the UDR response is successful. 4. If Cat 3 – Previous Location Check is enabled, ensure the SUPI is present in the incoming request message. 5. Verify the following: • Confirm the UDR discovery procedure completes successfully. • Ensure the FQDN or IP of the UDR is reachable. • Verify the Coherence service is up and running. • Make sure the SUPI falls within the IMSI range defined in the UDR profile. 6. Ensure that DNS resolution is properly configured for the UDR discovery call, specifically for the pn32f-svc service used in subscription use cases.

5.3.2.17.5 SEPPN32fPreviousLocationCheckExceptionFailureAlertMajor

Table 5-386 SEPPN32fPreviousLocationCheckExceptionFailureAlertMajor

Field	Details
Trigger Condition	When previous location check exception failure error is detected between 40 to 50 Percent of Total Transactions , this alert will be raised.
Severity	Major

Table 5-386 (Cont.) SEPPn32fPreviousLocationCheckExceptionFailureAlertMajor

Field	Details
Alert Details Provided	Summary <pre>namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}</pre> Expression: <pre>sum(rate(ocsepp_previous_location_exception_failure_total{namespace="sepp-namespace"} [2m]))by(namespace,nf_instance_id,app,pod)/ sum(rate(ocsepp_previous_location_validation_requests_total{namespace="sepp-namespace"} [2m]))by(namespace,nf_instance_id,app,pod)>=0.4 and sum(rate(ocsepp_previous_location_exception_failure_total{namespace="sepp-namespace"} [2m]))by(namespace,nf_instance_id,app,pod)/ sum(rate(ocsepp_previous_location_validation_requests_total{namespace="sepp-namespace"} [2m]))by(namespace,nf_instance_id,app,pod)<0.5</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4036
Metric Name	ocsepp_previous_location_exception_failure_total

Table 5-386 (Cont.) SEPPn32fPreviousLocationCheckExceptionFailureAlertMajor

Field	Details
Resolution	The alerts gets cleared when the previous location check exception failure error does not lie between 40 to 50 percent of total transactions. Possible Resolutions: 1. Look for the following error codes in the PN32F microservice logs: • SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-ERROR-0019 • SEPP-PN32FSEPP-ERROR-0018 • SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-EXCEPTION-0020 2. Ensure that the correct regex is configured for UE ID and Serving Network ID under the Header or Body IE configuration within the Cat 3 – Previous Location Check section of the Security Countermeasure tab in the SEPP CNC Console. 3. Check that the MCC and MNC values (configured in either the header or body) match the serving network name. These values are returned in the UDR response, so also confirm that the UDR response is successful. 4. If Cat 3 – Previous Location Check is enabled, ensure the SUPI is present in the incoming request message. 5. Verify the following: • Confirm the UDR discovery procedure completes successfully. • Ensure the FQDN or IP of the UDR is reachable. • Verify the Coherence service is up and running. • Make sure the SUPI falls within the IMSI range defined in the UDR profile. 6. Ensure that DNS resolution is properly configured for the UDR discovery call, specifically for the pn32f-svc service used in subscription use cases.

5.3.2.17.6 sEPPn32fPrevLoChkExcepFailAlrtCritical

Table 5-387 sEPPn32fPrevLoChkExcepFailAlrtCritical

Field	Details
Trigger Condition	When previous location check exception failure error is detected above 50 Percent of total transactions, this alert will be raised.
Severity	Critical

Table 5-387 (Cont.) sEPPn32fPrevLoChkExcepFailAlrtCritical

Field	Details
Alert Details Provided	Summary <pre>namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}</pre> Expression: <pre>sum(rate(ocsepp_previous_location_exception_failure_total{namespace="sepp-namespace"} [2m]))by(namespace,nf_instance_id,app,pod)/ sum(rate(ocsepp_previous_location_validation_requests_total{namespace="sepp-namespace"} [2m]))by(namespace,nf_instance_id,app,pod)>=0.5</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4037
Metric Name	ocsepp_previous_location_exception_failure_total

Table 5-387 (Cont.) sEPPn32fPrevLoChkExcepFailAlrtCritical

Field	Details
Resolution	The alerts gets cleared when the previous location check exception failure error does not lie above 50 percent of total transactions. Possible Resolutions: 1. Look for the following error codes in the PN32F microservice logs: • SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-ERROR-0019 • SEPP-PN32FSEPP-ERROR-0018 • SEPP-PREVIOUS-LOCATION-CHECK-VALIDATION-EXCEPTION-0020 2. Ensure that the correct regex is configured for UE ID and Serving Network ID under the Header or Body IE configuration within the Cat 3 – Previous Location Check section of the Security Countermeasure tab in the SEPP CNC Console. 3. Check that the MCC and MNC values (configured in either the header or body) match the serving network name. These values are returned in the UDR response, so also confirm that the UDR response is successful. 4. If Cat 3 – Previous Location Check is enabled, ensure the SUPI is present in the incoming request message. 5. Verify the following: • Confirm the UDR discovery procedure completes successfully. • Ensure the FQDN or IP of the UDR is reachable. • Verify the Coherence service is up and running. • Make sure the SUPI falls within the IMSI range defined in the UDR profile. 6. Ensure that DNS resolution is properly configured for the UDR discovery call, specifically for the pn32f-svc service used in subscription use cases.

5.3.2.18 Cat-3 Time Check for Roaming Subscribers

5.3.2.18.1 pn32fTimeUnauthLocChkValFailAlrtMinor

Table 5-388 pn32fTimeUnauthLocChkValFailAlrtMinor

Field	Details
Trigger Condition	Triggered in case of a minor failure for Cat-3Time Unauthenticated Location Check.
Severity	Minor
Alert Details Provided	Summary <pre>namespace: {{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }} {{ end }}</pre> Expression: <pre>sum(increase(ocsepp_time_unauthenticat ed_location_validation_failure_tota l{namespace="sepp-namespace"}[2m]) or ocsepp_time_unauthenticated_location_ validation_failure_total{namespace="s epp-namespace"} unless ocsepp_time_unauthenticated_location_ validation_failure_total{namespace="s epp-namespace"} offset 2m) by (namespace,nf_instance_id,app,pod) >=1 and sum(increase(ocsepp_time_unauthenticat ed_location_validation_failure_tota l{namespace="sepp-namespace"}[2m]) or ocsepp_time_unauthenticated_location_ validation_failure_total{namespace="s epp-namespace"} unless ocsepp_time_unauthenticated_location_ validation_failure_total{namespace="s epp-namespace"} offset 2m) by (namespace,nf_instance_id,app,pod) <=10</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4055
Metric Name	ocsepp_time_unauthenticated_location_validation_failure_total

Table 5-388 (Cont.) pn32fTimeUnauthLocChkValFailAlrtMinor

Field	Details
Resolution	The alert gets cleared when the failure count is above 10. Possible Resolutions: 1. Ensure that the Cat-3 Time Check for Roaming Subscribers is enabled at both the global and remote levels. For configuration issues and detailed analysis, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. Confirm that SUPI or SUCI is included in the incoming message. The UDM and UDR discovery procedures (FQDN or IP of UDR) must be successful and reachable. 3. Verify the following: • Verify that the Coherence service is operational. • Ensure the SUPI falls within the IMSI range defined in the UDR profile. • Confirm that the routing indicator from the SUCI matches the routingIndicator in the UDM profile. 4. DNS Resolution Confirm proper DNS resolution for the UDR discovery call, particularly for the pn32f-svc service used in subscription scenarios. 5. Ensure the supi Or Suci and servingNetworkName (including MCC/MNC) are present in the NAUSF request JSON body. Verify that the UDR response is successful and that the servingNetworkName values match. 6. If the request is rejected, verify whether the rejection might be caused by mismatched units of measurement.

5.3.2.18.2 pn32fTimeUnauthLocChkValFailAlrtMajor

Table 5-389 pn32fTimeUnauthLocChkValFailAlrtMajor

Field	Details
Trigger Condition	Triggered in case of a major failure for Cat-3 Time Unauthenticated Location Check.
Severity	Major

Table 5-389 (Cont.) pn32fTimeUnauthLocChkValFailAlrtMajor

Field	Details
Alert Details Provided	Summary <pre>namespace: {{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }} {{ end }}</pre> Expression: <pre>sum(increase(ocsepp_time_unauthenticat ed_location_validation_failure_tota l{namespace="sepp-namespace"}[2m]) or ocsepp_time_unauthenticated_location_ validation_failure_total{namespace="s epp-namespace"} unless ocsepp_time_unauthenticated_location_ validation_failure_total{namespace="s epp-namespace"} offset 2m) by (namespace,nf_instance_id,app,pod) >=11 and sum(increase(ocsepp_time_unauthenticat ed_location_validation_failure_tota l{namespace="sepp-namespace"}[2m]) or ocsepp_time_unauthenticated_location_ validation_failure_total{namespace="s epp-namespace"} unless ocsepp_time_unauthenticated_location_ validation_failure_total{namespace="s epp-namespace"} offset 2m) by (namespace,nf_instance_id,app,pod) <=50</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4056
Metric Name	ocsepp_time_unauthenticated_location_validation_failure_total

Table 5-389 (Cont.) pn32fTimeUnauthLocChkValFailAlrtMajor

Field	Details
Resolution	The alert gets cleared when the failure count is not in between 10 and 50. Possible Resolutions: 1. Ensure that the Cat-3 Time Check for Roaming Subscribers is enabled at both the global and remote levels. For configuration issues and detailed analysis, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. Confirm that SUPI or SUCI is included in the incoming message. The UDM and UDR discovery procedures (FQDN or IP of UDR) must be successful and reachable. 3. Verify the following: • Verify that the Coherence service is operational. • Ensure the SUPI falls within the IMSI range defined in the UDR profile. • Confirm that the routing indicator from the SUCI matches the routingIndicator in the UDM profile. 4. DNS Resolution Confirm proper DNS resolution for the UDR discovery call, particularly for the pn32f-svc service used in subscription scenarios. 5. Ensure the supi Or Suci and servingNetworkName (including MCC/MNC) are present in the NAUSF request JSON body. Verify that the UDR response is successful and that the servingNetworkName values match. 6. If the request is rejected, verify whether the rejection might be caused by mismatched units of measurement.

5.3.2.18.3 pn32fTimeUnauthLocChkValFailAlrtCritical

Table 5-390 pn32fTimeUnauthLocChkValFailAlrtCritical

Field	Details
Trigger Condition	Triggered in case of a critical failure for Cat-3 Time Unauthenticated Location Check.
Severity	Critical

Table 5-390 (Cont.) pn32fTimeUnauthLocChkValFailAlrtCritical

Field	Details
Alert Details Provided	Summary <pre>namespace: {{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }} {{ end }}</pre> Expression: <pre>sum(increase(ocsepp_time_unauthenticat ed_location_validation_failure_tota l{namespace="sepp-namespace"}[2m]) or ocsepp_time_unauthenticated_location_ validation_failure_total{namespace="s epp-namespace"} unless ocsepp_time_unauthenticated_location_ validation_failure_total{namespace="s epp-namespace"} offset 2m) by (namespace,nf_instance_id,app,pod) >=51</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4057
Metric Name	ocsepp_time_unauthenticated_location_validation_failure_total

Table 5-390 (Cont.) pn32fTimeUnauthLocChkValFailAlrtCritical

Field	Details
Resolution	The alert gets cleared when the failure count is below 51. Possible Resolutions: 1. Ensure that the Cat-3 Time Check for Roaming Subscribers is enabled at both the global and remote levels. For configuration issues and detailed analysis, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. Confirm that SUPI or SUCI is included in the incoming message. The UDM and UDR discovery procedures (FQDN or IP of UDR) must be successful and reachable. 3. Verify the following: • Verify that the Coherence service is operational. • Ensure the SUPI falls within the IMSI range defined in the UDR profile. • Confirm that the routing indicator from the SUCI matches the routingIndicator in the UDM profile. 4. DNS Resolution Confirm proper DNS resolution for the UDR discovery call, particularly for the pn32f-svc service used in subscription scenarios. 5. Ensure the supi Or Suci and servingNetworkName (including MCC/MNC) are present in the NAUSF request JSON body. Verify that the UDR response is successful and that the servingNetworkName values match. 6. If the request is rejected, verify whether the rejection might be caused by mismatched units of measurement.

5.3.2.18.4 pn32fTimeUnauthLocChkExcepFailAlrtMinor

Table 5-391 pn32fTimeUnauthLocChkExcepFailAlrtMinor

Field	Details
Trigger Condition	Triggered in case of a minor exception for Cat-3 Time Unauthenticated Location Check.
Severity	Minor

Table 5-391 (Cont.) pn32fTimeUnauthLocChkExcepFailAlrtMinor

Field	Details
Alert Details Provided	Summary <pre>namespace: {{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }} {{ end }}</pre> Expression: <pre>sum(increase(ocsepp_time_unauthenticat ed_location_exception_failure_total{ namespace="sepp-namespace"}[2m]) or ocsepp_time_unauthenticated_location_ exception_failure_total{namespace="se pp-namespace"} unless ocsepp_time_unauthenticated_location_ exception_failure_total{namespace="se pp-namespace"} offset 2m) by (namespace,nf_instance_id,app,pod) >=1 and sum(increase(ocsepp_time_unauthenticat ed_location_exception_failure_total{ namespace="sepp-namespace"}[2m]) or ocsepp_time_unauthenticated_location_ exception_failure_total{namespace="se pp-namespace"} unless ocsepp_time_unauthenticated_location_ exception_failure_total{namespace="se pp-namespace"} offset 2m) by (namespace,nf_instance_id,app,pod) <=10</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4058
Metric Name	ocsepp_time_unauthenticated_location_exception_failure_total

Table 5-391 (Cont.) pn32fTimeUnauthLocChkExcepFailAlrtMinor

Field	Details
Resolution	The alert gets cleared when the exception count is above 10. Possible Resolutions: 1. Ensure that the Cat-3 Time Check for Roaming Subscribers is enabled at both the global and remote levels. For configuration issues and detailed analysis, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. Confirm that SUPI or SUCI is included in the incoming message. The UDM and UDR discovery procedures (FQDN or IP of UDR) must be successful and reachable. 3. Verify the following: • Verify that the Coherence service is operational. • Ensure the SUPI falls within the IMSI range defined in the UDR profile. • Confirm that the routing indicator from the SUCI matches the routingIndicator in the UDM profile. 4. DNS Resolution Confirm proper DNS resolution for the UDR discovery call, particularly for the pn32f-svc service used in subscription scenarios. 5. Ensure the supi Or Suci and servingNetworkName (including MCC/MNC) are present in the NAUSF request JSON body. Verify that the UDR response is successful and that the servingNetworkName values match. 6. If the request is rejected, verify whether the rejection might be caused by mismatched units of measurement.

5.3.2.18.5 pn32fTimeUnauthLocChkExcepFailAlrtMajor

Table 5-392 pn32fTimeUnauthLocChkExcepFailAlrtMajor

Field	Details
Trigger Condition	Triggered in case of a major exception for Cat-3 Time Unauthenticated Location Check.
Severity	Major

Table 5-392 (Cont.) pn32fTimeUnauthLocChkExcepFailAlrtMajor

Field	Details
Alert Details Provided	Summary <pre>namespace: {{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }} {{ end }}</pre> Expression: <pre>sum(increase(ocsepp_time_unauthenticat ed_location_exception_failure_total{ namespace="sepp-namespace"}[2m]) or ocsepp_time_unauthenticated_location_ exception_failure_total{namespace="se pp-namespace"} unless ocsepp_time_unauthenticated_location_ exception_failure_total{namespace="se pp-namespace"} offset 2m) by (namespace,nf_instance_id,app,pod) >=11 and sum(increase(ocsepp_time_unauthenticat ed_location_exception_failure_total{ namespace="sepp-namespace"}[2m]) or ocsepp_time_unauthenticated_location_ exception_failure_total{namespace="se pp-namespace"} unless ocsepp_time_unauthenticated_location_ exception_failure_total{namespace="se pp-namespace"} offset 2m) by (namespace,nf_instance_id,app,pod) <=50</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4059
Metric Name	ocsepp_time_unauthenticated_location_exception_failure_total

Table 5-392 (Cont.) pn32fTimeUnauthLocChkExcepFailAlrtMajor

Field	Details
Resolution	The alert gets cleared when the exception count is not in between 10 and 50. Possible Resolutions: 1. Ensure that the Cat-3 Time Check for Roaming Subscribers is enabled at both the global and remote levels. For configuration issues and detailed analysis, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. Confirm that SUPI or SUCI is included in the incoming message. The UDM and UDR discovery procedures (FQDN or IP of UDR) must be successful and reachable. 3. Verify the following: • Verify that the Coherence service is operational. • Ensure the SUPI falls within the IMSI range defined in the UDR profile. • Confirm that the routing indicator from the SUCI matches the routingIndicator in the UDM profile. 4. DNS Resolution Confirm proper DNS resolution for the UDR discovery call, particularly for the pn32f-svc service used in subscription scenarios. 5. Ensure the supi Or Suci and servingNetworkName (including MCC/MNC) are present in the NAUSF request JSON body. Verify that the UDR response is successful and that the servingNetworkName values match. 6. If the request is rejected, verify whether the rejection might be caused by mismatched units of measurement.

5.3.2.18.6 pn32fTimeUnauthLocChkExcepFailAlrtCritical

Table 5-393 pn32fTimeUnauthLocChkExcepFailAlrtCritical

Field	Details
Trigger Condition	Triggered in case of a critical exception for Cat-3 Time Unauthenticated Location Check.
Severity	Critical

Table 5-393 (Cont.) pn32fTimeUnauthLocChkExcepFailAlrtCritical

Field	Details
Alert Details Provided	Summary <pre>namespace: {{ \$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . first value humanizeTimestamp }} {{ end }}</pre> Expression: <pre>sum(increase(ocsepp_time_unauthenticat ed_location_exception_failure_total{ namespace="sepp-namespace"}[2m]) or ocsepp_time_unauthenticated_location_ exception_failure_total{namespace="se pp-namespace"} unless ocsepp_time_unauthenticated_location_ exception_failure_total{namespace="se pp-namespace"} offset 2m) by (namespace,nf_instance_id,app,pod) >=51</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4060
Metric Name	ocsepp_time_unauthenticated_location_exception_failure_total

Table 5-393 (Cont.) pn32fTimeUnauthLocChkExcepFailAlrtCritical

Field	Details
Resolution	The alert gets cleared when the exception count is below 51. Possible Resolutions: 1. Ensure that the Cat-3 Time Check for Roaming Subscribers is enabled at both the global and remote levels. For configuration issues and detailed analysis, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. Confirm that SUPI or SUCI is included in the incoming message. The UDM and UDR discovery procedures (FQDN or IP of UDR) must be successful and reachable. 3. Verify the following: • Verify that the Coherence service is operational. • Ensure the SUPI falls within the IMSI range defined in the UDR profile. • Confirm that the routing indicator from the SUCI matches the routingIndicator in the UDM profile. 4. DNS Resolution Confirm proper DNS resolution for the UDR discovery call, particularly for the pn32f-svc service used in subscription scenarios. 5. Ensure the supi Or Suci and servingNetworkName (including MCC/MNC) are present in the NAUSF request JSON body. Verify that the UDR response is successful and that the servingNetworkName values match. 6. If the request is rejected, verify whether the rejection might be caused by mismatched units of measurement.

5.3.2.19 Rate Limiting for Egress Roaming Signaling per PLMN Alerts

5.3.2.19.1 EgressRequestRateLimitDropAbove10Percent

Table 5-394 EgressRequestRateLimitDropAbove10Percent

Field	Details
Trigger Condition	If a request is dropped due to the tokens in the bucket are exhausted and drop rate per PLMN is detected above 10 percent of total transactions of that PLMN, oc_ingressgateway_plmn_egress_ratelimit_total metric will be pegged and corresponding alert will be raised.
Severity	Minor
Alert Details Provided	Summary Egress Rate Limiting Request Drop Rate detected per PLMN above 10 Percent of Total Transactions Expression: <pre>sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total{namespace="sepp-namespace",Status="ERL_MATCH_NO_TOKEN_LOW_PRI_REJECT"}[5m])) by (EgressRateLimitList,PLMN_ID,namespace)/ sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total{namespace="sepp-namespace"}[5m])) by (EgressRateLimitList,PLMN_ID,namespace) *100 >= 10 < 25</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4039
Metric Name	oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-394 (Cont.) EgressRequestRateLimitDropAbove10Percent

Field	Details
Resolution	The alerts gets cleared when the count goes down. Possible resolutions: 1. If the traffic is being forwarded even if tokens for the Egress Rate Limiting List are exhausted, do the following: a. Verify the 3gpp-Sbi-Message-Priority header of the request. If the priority in the header is less than Discard Message Priority property of the message in the Egress Rate Limiting List, then the message is not dropped. b. If the 3gpp-Sbi-Message-Priority header is not present, then the priority is checked in the route configuration. If a value for 3gpp-Sbi-Message-Priority is present in the route configuration, then the above mentioned condition is considered and the same solution is applied. c. If the priority is unknown for the request, 24 is considered as the default value for the request priority, then, the same condition as above is applied. 2. For further details and configuration related issues, please <refer troubleshooting> 2. For all configuration related and additional information, refer 'Rate Limiting for Egress Roaming Signaling per PLMN feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.19.2 EgressRequestRateLimitDropAbove25Percent

Table 5-395 EgressRequestRateLimitDropAbove25Percent

Field	Details
Trigger Condition	If a request is dropped due to the tokens in the bucket are exhausted and drop rate per PLMN is detected above 25 percent of total transactions of that PLMN, <code>oc_ingressgateway_plmn_egress_ratelimit_total</code> metric will be pegged and corresponding alert will be raised.
Severity	Major

Table 5-395 (Cont.) EgressRequestRateLimitDropAbove25Percent

Field	Details
Alert Details Provided	Summary Egress Rate Limiting Request Drop Rate detected per PLMN above 25 Percent of Total Transactions Expression: <pre>sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total{namespace="sepp-namespace",Status="ERL_MATCH_NO_TOKEN_LOW_PRI_REJECT"}[5m])) by (EgressRateLimitList,PLMN_ID,namespace)/ sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total{namespace="sepp-namespace"}[5m])) by (EgressRateLimitList,PLMN_ID,namespace) *100 >= 25 < 50</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4040
Metric Name	oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-395 (Cont.) EgressRequestRateLimitDropAbove25Percent

Field	Details
Resolution	The alerts gets cleared when the count goes down. Possible resolutions: 1. If the traffic is being forwarded even if tokens for the Egress Rate Limiting List are exhausted, do the following: a. Verify the 3gpp-Sbi-Message-Priority header of the request. If the priority in the header is less than Discard Message Priority property of the message in the Egress Rate Limiting List, then the message is not dropped. b. If the 3gpp-Sbi-Message-Priority header is not present, then the priority is checked in the route configuration. If a value for 3gpp-Sbi-Message-Priority is present in the route configuration, then the above mentioned condition is considered and the same solution is applied. c. If the priority is unknown for the request, 24 is considered as the default value for the request priority, then, the same condition as above is applied. 2. For further details and configuration related issues, please <refer troubleshooting> 2. For all configuration related and additional information, refer 'Rate Limiting for Egress Roaming Signaling per PLMN feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.19.3 EgressRequestRateLimitDropAbove50Percent

Table 5-396 EgressRequestRateLimitDropAbove50Percent

Fields	Details
Trigger Condition	If a request is dropped due to the tokens in the bucket are exhausted and the drop rate per PLMN is detected above 50 percent of total transactions of that PLMN, oc_ingressgateway_plmn_egress_ratelimit_total metric will be pegged and corresponding alert will be raised.
Severity	Critical

Table 5-396 (Cont.) EgressRequestRateLimitDropAbove50Percent

Fields	Details
Alert Details Provided	Summary Egress Rate Limiting Request Drop Rate detected per PLMN above 50 Percent of Total Transactions Expression: <pre>sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total{namespace="sepp-namespace",Status="ERL_MATCH_NO_TOKEN_LOW_PRI_REJECT"}[5m])) by (EgressRateLimitList,PLMN_ID,namespace)/sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total{namespace="sepp-namespace"}[5m])) by (EgressRateLimitList,PLMN_ID,namespace) *100 >= 50</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4041
Metric Name	oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-396 (Cont.) EgressRequestRateLimitDropAbove50Percent

Fields	Details
Resolution	The alerts gets cleared when the count goes down. Possible resolutions: 1. If the traffic is being forwarded even if tokens for the Egress Rate Limiting List are exhausted, do the following: a. Verify the 3gpp-Sbi-Message-Priority header of the request. If the priority in the header is less than Discard Message Priority property of the message in the Egress Rate Limiting List, then the message is not dropped. b. If the 3gpp-Sbi-Message-Priority header is not present, then the priority is checked in the route configuration. If a value for 3gpp-Sbi-Message-Priority is present in the route configuration, then the above mentioned condition is considered and the same solution is applied. c. If the priority is unknown for the request, 24 is considered as the default value for the request priority, then, the same condition as above is applied. 2. For further details and configuration related issues, please <refer troubleshooting> 2. For all configuration related and additional information, refer 'Rate Limiting for Egress Roaming Signaling per PLMN feature' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.20 Separate Port Configurations for N32c and N32f on the Egress Routes Alerts

5.3.2.20.1 EgressInterfaceConnectionFailure

Table 5-397 EgressInterfaceConnectionFailure

Field	Details
Trigger Condition	If the destination host and port mentioned in the Remote profile are unreachable or not available, then the alert will be raised.
Severity	Major

Table 5-397 (Cont.) EgressInterfaceConnectionFailure

Field	Details
Alert Details Provided	Summary: Egress connection failure on the interface Expression: <pre>sum(increase(oc_egressgateway_connection_failure_total{namespace="sepp-namespace", app="n32-egress-gateway"}[5m])) by (namespace, app, Host, Port) >0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4042
Metric Name	oc_egressgateway_connection_failure_total
Resolution	Possible resolutions: 1. If the Remote SEPP is changed with new fields of N32F configuration, and the traffic is uneven after changing the profile, check: a. whether the Remote SEPP Set is created correctly. b. if the configuration is stored inside the database correctly. c. Verify in the database that the routes at n32-egress-gateway are updated according to the new configuration done at Remote SEPP. 2. For more details, refer 'Separate Port Configurations for N32c and N32f on the Egress Routes' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.21 Support for TLS 1.3

5.3.2.21.1 SEPPConnectionFailurePLMNIGWAlert

Table 5-398 SEPPConnectionFailurePLMNIGWAlert

Field	Details
Trigger Condition	Connection failure occurs for incoming traffic at PLMN Ingress Gateway
Severity	Major

Table 5-398 (Cont.) SEPPConnectionFailurePLMNIGWAlert

Field	Details
Alert details provided	Summary: <pre>namespace:{{\$labels.namespace}}, timestamp:{{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end}}: Incoming connection failure on plmn- ingress-gateway due to {{\$labels.error_reason }}</pre> Expression: <pre>sum(increase(oc_ingressgateway_conne- tion_failure_total{namespace="sepp- namespace",app="plmn-ingress- gateway"}[5m]) >0 or (oc_ingressgateway_connection_failure_ total{namespace="sepp- namespace",app="plmn-ingress- gateway"} unless oc_ingressgateway_connection_failure_ total{namespace="sepp- namespace",app="plmn-ingress- gateway"} offset 5m)) by (namespace,app) > 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4043
Metric used	oc_ingressgateway_connection_failure_total
Resolution	After resolving the reason for the connection failure, this alert will be removed. Possible resolutions: 1. If SSLHandshakeException or SSLException error is observed, verify that both the server and client have one common TLS version supported. 2. If a common TLS version is supported, then check if both the client and server have compatible cipher suites. 3. Also, ensure that CA certificate of server must be present inside client's truststore. 4. For more details, refer 'Support for TLS 1.3' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.21.2 SEPPConnectionFailureN32IGWAlert

Table 5-399 SEPPConnectionFailureN32IGWAlert

Field	Details
Trigger Condition	Connection failure occurs for incoming traffic at N32 Ingress Gateway
Severity	Major
Alert details provided	Summary: <pre>namespace: {{\$labels.namespace}}, timestamp: {{ with query "time()" }} {{ . first value humanizeTimestamp }}{{ end }}: Incoming connection failure on n32- ingress-gateway due to {{ \$labels.error_reason }}</pre> Expression: <pre>sum(increase(oc_ingressgateway_conne- ction_failure_total{namespace="sepp- namespace",app="n32-ingress-gateway"} [5m]) >0 or (oc_ingressgateway_connection_failure_ total{namespace="sepp- namespace",app="n32-ingress- gateway"} unless oc_ingressgateway_connection_failure_ total{namespace="sepp- namespace",app="n32-ingress- gateway"} offset 5m)) by (namespace,app) > 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4044
Metric used	oc_ingressgateway_connection_failure_total

Table 5-399 (Cont.) SEPPConnectionFailureN32IGWAlert

Field	Details
Resolution	After resolving the reason for connection failure, this alert will be removed. Possible resolutions: 1. If SSLHandshakeException or SSLEException error is observed, verify that both the server and client have one common TLS version supported. 2. If a common TLS version is supported, then check if both the client and server have compatible cipher suites. 3. Also, ensure that CA certificate of server must be present inside client's truststore. 4. For more details, refer 'Support for TLS 1.3' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

Note

- addErrorReasonToConnectionFailureMetric to true metric must be true for the metric to be pegged with error reason and alert SEPPConnectionFailureN32IGWAlert to be generated.

5.3.2.21.3 SEPPX509CertificateExpiryAlertMinor

Table 5-400 SEPPX509CertificateExpiryAlertMinor

Field	Details
Trigger Condition	When TLS certificate is valid for only 6 months before expiration.
Severity	Minor
Alert details provided	Summary: Certificate expiry in less than 6 months Expression: <pre>security_cert_x509_expiration_seconds{namespace="sepp-namespace"} - time() <= 15724800</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4045
Metric used	security_cert_x509_expiration_seconds

Table 5-400 (Cont.) SEPPX509CertificateExpiryAlertMinor

Field	Details
Resolution	Only after certificates have been updated, this alert will be removed. Possible resolutions: 1. For more information on certificate lifecycle process, refer Support for 'Automated Certificate Lifecycle Management' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide</i>.

5.3.2.21.4 SEPPX509CertificateExpiryAlertMajor

Table 5-401 SEPPX509CertificateExpiryAlertMajor

Field	Details
Trigger Condition	When TLS certificate is valid for only 3 months before expiration.
Severity	Major
Alert details provided	Summary: Certificate expiry in less than 3 months Expression: <pre>security_cert_x509_expiration_seconds{namespace="sepp-namespace"} - time() <= 7862400</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4046
Metric used	security_cert_x509_expiration_seconds
Resolution	Only after certificates have been updated, this alert will be removed. Possible resolutions: 1. For more information on certificate lifecycle process, refer 'Support for Automated Certificate Lifecycle Management' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide</i>.

5.3.2.21.5 SEPPX509CertificateExpiryAlertCritical

Table 5-402 SEPPX509CertificateExpiryAlertCritical

Field	Details
Trigger Condition	When TLS certificate is valid for only 1 month before expiration.
Severity	Critical
Alert details provided	Summery: Certificate expiry in less than 1 month Expression: <pre>security_cert_x509_expiration_seconds{namespace="sepp-namespace"} - time() <= 2592000</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4047
Metric used	security_cert_x509_expiration_seconds
Resolution	Only after certificates have been updated, this alert will be removed. Possible resolutions: For more information on certificate lifestyle process, refer 'Support for Automated Certificate Lifecycle Management' section in the <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide</i>.

5.3.2.22 Multiple SEPP Instances on Shared cnDBTier Cluster Alerts

5.3.2.22.1 Cn32fConnectionFailureWithDatabaseAlert

Table 5-403 Cn32fConnectionFailureWithDatabaseAlert

Field	Details
Trigger Condition	ocsepp_cn32f_database_connectivity_healthy = 0
Severity	Major

Table 5-403 (Cont.) Cn32fConnectionFailureWithDatabaseAlert

Field	Details
Alert Details Provided	Summary: Alert is raised when connectivity is broken between CN32f and cnDBTier. Metric value is pegged as 0 and then alert is raised. Expression: <pre>ocsepp_cn32f_database_connectivity_healthy{namespace="sepp-namespace"} == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4050
Metric Name	ocsepp_cn32f_database_connectivity_healthy
Resolution	Possible resolutions: 1. Verify services status to ensure that all services are running in the namespace where cnDBTier is deployed. 2. Check table creation to confirm that required tables are created in the database. 3. To verify whether the services are connected to correct database and namespace, run <pre>kubectl get deploy deploymentName -n namespace</pre> , and search for <code>spring.datasource.url</code> parameter and check the value. 4. For more details, refer 'Multiple SEPP instances on Shared cnDBTier Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.22.2 Cn32cConnectionFailureWithDatabaseAlert

Table 5-404 Cn32cConnectionFailureWithDatabaseAlert

Field	Details
Trigger Condition	ocsepp_cn32c_database_connectivity_healthy == 0
Severity	Major

Table 5-404 (Cont.) Cn32cConnectionFailureWithDatabaseAlert

Field	Details
Alert Details Provided	Summary: Alert is raised when connectivity is broken between CN32c and cnDBTier for more than 30 seconds. Metric value is pegged as 0 and then alert is raised. Expression: <pre>ocsepp_cn32c_database_connectivity_healthy{namespace="sepp-namespace"} == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4051
Metric Name	ocsepp_cn32c_database_connectivity_healthy
Resolution	Possible resolutions: 1. Verify services status to ensure that all services are running in the namespace where cnDBTier is deployed. 2. Check table creation to confirm that required tables are created in the database. 3. To verify whether the services are connected to correct database and namespace, run <pre>kubectl get deploy deploymentName -n namespace</pre> , and search for <code>spring.datasource.url</code> parameter and check the value. 4. For more details, refer 'Multiple SEPP instances on Shared cnDBTier Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.22.3 Pn32fConnectionFailureWithDatabaseAlert

Table 5-405 Pn32fConnectionFailureWithDatabaseAlert

Field	Details
Trigger Condition	ocsepp_pn32f_database_connectivity_healthy == 0
Severity	Major

Table 5-405 (Cont.) Pn32fConnectionFailureWithDatabaseAlert

Field	Details
Alert Details Provided	Summary: Alert is raised when connectivity is broken between PN32F and cnDBTier for more than 30 seconds. Metric value is pegged as 0 and then alert is raised. Expression: <pre>ocsepp_pn32f_database_connectivity_healthy{namespace="sepp-namespace"} == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4052
Metric Name	ocsepp_pn32f_database_connectivity_healthy
Resolution	Possible resolutions: 1. Verify services status to ensure that all services are running in the namespace where cnDBTier is deployed. 2. Check table creation to confirm that required tables are created in the database. 3. To verify whether the services are connected to correct database and namespace, run <pre>kubectl get deploy deploymentName -n namespace</pre> , and search for <code>spring.datasource.url</code> parameter and check the value. 4. For more details, refer 'Multiple SEPP instances on Shared cnDBTier Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.22.4 Pn32cConnectionFailureWithDatabaseAlert

Table 5-406 Pn32cConnectionFailureWithDatabaseAlert

Field	Details
Trigger Condition	ocsepp_pn32c_database_connectivity_healthy == 0
Severity	Major

Table 5-406 (Cont.) Pn32cConnectionFailureWithDatabaseAlert

Field	Details
Alert Details Provided	Summary: Alert is raised when connectivity is broken between PN32C and cnDBTier for more than 30 seconds. Metric value is pegged as 0 and then alert is raised. Expression: <pre>ocsepp_pn32c_database_connectivity_healthy{namespace="sepp-namespace"} == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4053
Metric Name	ocsepp_pn32c_database_connectivity_healthy
Resolution	Possible resolutions: 1. Verify services status to ensure that all services are running in the namespace where cnDBTier is deployed. 2. Check table creation to confirm that required tables are created in the database. 3. To verify whether the services are connected to correct database and namespace, run <pre>kubectl get deploy deploymentName -n namespace</pre> , and search for <code>spring.datasource.url</code> parameter and check the value. 4. For more details, refer 'Multiple SEPP instances on Shared cnDBTier Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.22.5 cfgMgrConnFailureWithDBAlert

Table 5-407 cfgMgrConnFailureWithDBAlert

Field	Details
Trigger Condition	ocsepp_configmgr_database_connectivity_healthy == 0
Severity	Major

Table 5-407 (Cont.) cfgMgrConnFailureWithDBAlert

Field	Details
Alert Details Provided	Summary: Alert is raised when connectivity is broken between PN32C and cnDBTier for more than 30 seconds. Metric value is pegged as 0 and then alert is raised. Expression: <pre>oocsepp_configmgr_database_connectivity_healthy{namespace="sepp-namespace"} == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4054
Metric Name	ocsepp_configmgr_database_connectivity_healthy == 0
Resolution	Possible resolutions: 1. Verify services status to ensure that all services are running in the namespace where cnDBTier is deployed. 2. Check table creation to confirm that required tables are created in the database. 3. To verify whether the services are connected to correct database and namespace, run <pre>kubectl get deploy deploymentName -n namespace</pre> , and search for <code>spring.datasource.url</code> parameter and check the value. 4. For more details, refer 'Multiple SEPP instances on Shared cnDBTier Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.22.6 cn32fIncorrectDbConf

Table 5-408 cn32fIncorrectDbConf

Field	Details
Trigger Condition	This alert will be raised when incorrect database configuration provided for cn32f service and resulting in connection failure with database.
Severity	Major

Table 5-408 (Cont.) cn32fIncorrectDbConf

Field	Details
Alert Details Provided	Summary: Due to incorrect database configuration, connection failed with database. Expression: <pre>(up{app="cn32f-svc", namespace="sepp-namespace"} unless on (namespace) absent(hikaricp_connections{app="cn32f-svc", namespace="sepp-namespace"})) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4063
Metric Name	NA
Resolution	Possible resolutions: 1. Verify the <code>ocsepp_custom_values_<version>.yaml</code> file to ensure that the correct namespace where <code>cnDBTier</code> is installed is specified, and the database names are also correct. Example: <pre>seppDbName: &dbNameRef "seppdb_user1_sepp" seppBackupDbName: &backupDbNameRef "seppbackupdb_user1_sepp" mysql: primary: host: &mySqlHostRef "mysql- connectivity-service.cndb-sepp" # Ensure the namespace is correct in which cndb should get installed.</pre> 2. For more details, refer 'Multiple SEPP instances on Shared <code>cnDBTier</code> Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.22.7 cn32cIncorrectDbConf

Table 5-409 cn32cIncorrectDbConf

Field	Details
Trigger Condition	This alert will be raised when incorrect database configuration provided for cn32c service and resulting in connection failure with database.
Severity	Major
Alert Details Provided	Summary: Due to incorrect database configuration, connection failed with database. Expression: <pre>(up{app="cn32c-svc", namespace="sepp-namespace"} unless on (namespace) absent(hikaricp_connections{app="cn32c-svc", namespace="sepp-namespace"})) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4062
Metric Name	NA
Resolution	Possible resolutions: 1. Verify the <code>ocsepp_custom_values_<version>.yaml</code> file to ensure that the correct namespace where cnDBTier is installed is specified, and the database names are also correct. Example: <pre>seppDbName: &dbNameRef "seppdb_user1_sepp" seppBackupDbName: &backupDbNameRef "seppbackupdb_user1_sepp" mysql: primary: host: &mySqlHostRef "mysql-connectivity-service.cndb-sepp" # Ensure the namespace is correct in which cndb should get installed.</pre> 2. For more details, refer 'Multiple SEPP instances on Shared cnDBTier Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.22.8 pn32fIncorrectDbConf

Table 5-410 pn32fIncorrectDbConf

Field	Details
Trigger Condition	This alert will be raised when incorrect database configuration provided for pn32f service and resulting in connection failure with database.
Severity	Major
Alert Details Provided	Summary: Due to incorrect database configuration, connection failed with database. Expression: <pre>(up{app="pn32f-svc", namespace="sepp-namespace"} unless on (namespace) absent(hikaricp_connections{app="pn32f-svc", namespace="sepp-namespace"})) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4064
Metric Name	NA
Resolution	Possible resolutions: 1. Verify the <code>ocsepp_custom_values_<version>.yaml</code> file to ensure that the correct namespace where <code>cnDBTier</code> is installed is specified, and the database names are also correct. Example: <pre>seppDbName: &dbNameRef "seppdb_user1_sepp" seppBackupDbName: &backupDbNameRef "seppbackupdb_user1_sepp" mysql: primary: host: &mySqlHostRef "mysql-connectivity-service.cndb-sepp" # Ensure the namespace is correct in which cndb should get installed.</pre> 2. For more details, refer 'Multiple SEPP instances on Shared <code>cnDBTier</code> Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.22.9 pn32cIncorrectDbConf

Table 5-411 pn32cIncorrectDbConf

Field	Details
Trigger Condition	This alert will be raised when incorrect database configuration provided for pn32c service and resulting in connection failure with database.
Severity	Major
Alert Details Provided	Summary: Due to incorrect database configuration, connection failed with database. Expression: <pre>(up{app="pn32c-svc", namespace="sepp-namespace"} unless on (namespace) absent(hikaricp_connections{app="pn32c-svc", namespace="sepp-namespace"})) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4065
Metric Name	NA
Resolution	Possible resolutions: 1. Verify the <code>ocsepp_custom_values_<version>.yaml</code> file to ensure that the correct namespace where cnDBTier is installed is specified, and the database names are also correct. Example: <pre>seppDbName: &dbNameRef "seppdb_user1_sepp" seppBackupDbName: &backupDbNameRef "seppbackupdb_user1_sepp" mysql: primary: host: &mySqlHostRef "mysql-connectivity-service.cndb-sepp" # Ensure the namespace is correct in which cndb should get installed.</pre> 2. For more details, refer 'Multiple SEPP instances on Shared cnDBTier Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.22.10 cfgMgrIncorrectDbConf

Table 5-412 cfgMgrIncorrectDbConf

Field	Details
Trigger Condition	This alert will be raised when incorrect database configuration provided for config manager service and resulting in connection failure with database.
Severity	Major
Alert Details Provided	Summary: Due to incorrect database configuration, connection failed with database. Expression: <pre>(up{app="config-mgr-svc", namespace="sepp-namespace"} unless on (namespace) absent(hikaricp_connections{app="config-mgr-svc", namespace="sepp-namespace"})) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4061
Metric Name	NA
Resolution	Possible resolutions: 1. Verify the <code>ocsepp_custom_values_<version>.yaml</code> file to ensure that the correct namespace where <code>cnDBTier</code> is installed is specified, and the database names are also correct. Example: <pre>seppDbName: &dbNameRef "seppdb_user1_sepp" seppBackupDbName: &backupDbNameRef "seppbackupdb_user1_sepp" mysql: primary: host: &mySqlHostRef "mysql-connectivity-service.cndb-sepp" # Ensure the namespace is correct in which cndb should get installed.</pre> 2. For more details, refer 'Multiple SEPP instances on Shared <code>cnDBTier</code> Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.22.11 coherenceIncorrectDbConf

Table 5-413 coherenceIncorrectDbConf

Field	Details
Trigger Condition	This alert will be raised when incorrect database configuration provided for coherence service and resulting in connection failure with database.
Severity	Major
Alert details provided	Summary: Due to incorrect database configuration, connection failed with database. Expression: <pre>(up{app="coherence-svc",namespace="sepp-namespace"} unless on (namespace) absent(hikaricp_connections{app="coherence-svc"})) == 0</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4091
Metric Used	NA
Resolution	Possible Resolutions: 1. Verify the <code>ocsepp_custom_values_<version>.yaml</code> file to ensure that the correct namespace where <code>cnDBTier</code> is installed is specified, and the database names are also correct. Example: <pre>seppDbName: &dbNameRef "seppdb_user1_sepp" seppBackupDbName: &backupDbNameRef "seppbackupdb_user1_sepp" mysql: primary: host: &mySqlHostRef "mysql-connectivity-service.cndb-sepp" # Ensure the namespace is correct in which cndb should get installed.</pre> 2. For more details, refer 'Multiple SEPP instances on Shared cnDBTier Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.22.12 coherenceConnFailureWithDBAlert

Table 5-414 coherenceConnFailureWithDBAlert

Field	Details
Trigger Condition	This alert will be raised when connectivity is broken between coherence and cnDBTier for more than 30 seconds. Metric value is pegged as 0 and then alert is raised.
Severity	Major
Alert details provided	Expression: ocsepp_coherence_database_connectivity_healthy{namespace="sepp-namespace"} == 0
OID	1.3.6.1.4.1.323.5.3.46.1.2.4092
Metric Used	NA
Resolution	Possible Resolutions: 1. Verify services status to ensure that all services are running in the namespace where cnDBTier is deployed. 2. Check table creation to confirm that required tables are created in the database. 3. Run the following command to check table creation to confirm that required tables are created in the database: kubect1 get deploy deploymentName -n namespace and search for spring.datasource.url parameter and check the value. 4. For more details, refer 'Multiple SEPP instances on Shared cnDBTier Cluster' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.23 Proactive Status Updates on SEPP Alerts

5.3.2.23.1 EgressGatewayPeerUnhealthyAlert

Table 5-415 EgressGatewayPeerUnhealthyAlert

Field	Details
Trigger Condition	When a peer becomes unhealthy or oc_egressgateway_peer_health_status for a peer value = 1
Severity	Major

Table 5-415 (Cont.) EgressGatewayPeerUnhealthyAlert

Field	Details
Alert Details Provided	Summary Peer is unhealthy Expression: <code>sum(oc_egressgateway_peer_health_status{namespace="sepp-namespace",app="n32-egress-gateway"}) by (namespace,app,peer) >0</code>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4048
Metric Name	oc_egressgateway_peer_health_status
Resolution	Possible resolutions: 1. Check whether the peer SEPP is unreachable. If it's reachable and still it's showing as unhealthy, Refer to 'Proactive status updates on SEPP' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. The alert gets cleared once the particular peer becomes reachable, that is, the <code>oc_egressgateway_peer_health_status</code> for the peer becomes 0.

5.3.2.23.2 EgressGatewayAllPeersUnhealthyAlert

Table 5-416 EgressGatewayAllPeersUnhealthyAlert

Field	Details
Trigger Condition	When all peers in a peerset become unhealthy.
Severity	Critical
Alert Details Provided	Summary All peers unhealthy Expression: <code>(sum(oc_egressgateway_peer_count{namespace="sepp-namespace"}) by (namespace) - sum(oc_egressgateway_peer_available_count{namespace="sepp-namespace"}) by (namespace))=sum(oc_egressgateway_peer_count{namespace="sepp-namespace"}) by (namespace)</code>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4049

Table 5-416 (Cont.) EgressGatewayAllPeersUnhealthyAlert

Field	Details
Metric Name	oc_egressgateway_peer_count, oc_egressgateway_peer_available_count
Resolution	Possible resolutions: 1. Check if all the peers are unreachable. If any peer is reachable and reachable and still this alert is raised, Refer to 'Proactive status updates on SEPP' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>. 2. The alert gets cleared once all peers become reachable, that is, the difference between the available peer count and total peer count becomes less than total peer count.

5.3.2.24 Integrating SEPP with 5G Network Intelligence Fabric (5G NIF) Feature Alerts

5.3.2.24.1 configMgrNoHealthyNIFAlert

Table 5-417 configMgrNoHealthyNIFAlert

Field	Details
Trigger Condition	Raised when there is no healthy NIF found from NRF discovery for NIF.
Severity	Critical
Alert details provided	Expression: (sum(ocsepp_nif_registration_status{namespace="sepp-namespace"}) by (namespace)) <= 0
OID	1.3.6.1.4.1.323.5.3.46.1.2.4070
Metric Used	ocsepp_network_id_validation_header_failure_total

Table 5-417 (Cont.) configMgrNoHealthyNIFAlert

Field	Details
Resolution	Cleared when there is at least one healthy NIF for forwarding traffic, Possible Resolutions: 1. Review CN32F logs for error codes SEPP-CN32FSEPP-ERROR-0013 or SEPP-CN32FSEPP-ERROR-0014 (indicating the incoming request was rejected with error code 406 by the CN32F microservice). 2. Similarly, check PN32F logs for error codes SEPP-PN32FSEPP-ERROR-0016 or SEPP-PN32FSEPP-ERROR-0017 (indicating the request was rejected with error code 406 by the PN32F microservice). 3. Ensure the correct regular expression is configured under Header/Body IE in the Cat 2 – Network ID Validation section (located under the Security Countermeasure tab in SEPP). 4. Confirm that the PLMN ID specified in the request headers is included in the SEPP's configured PLMN ID List. 5. For issues related to invalid PLMN ID configurations in the body or headers, refer to 'Cat-2 Network ID Validation Feature' section in <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.25 Support for PLMN Validation in Get Access Token Requests for Roaming Subscribers Alerts

5.3.2.25.1 SEPPN32fNetworkIDOAuthValidationFailureWarn

Table 5-418 SEPPN32fNetworkIDOAuthValidationFailureWarn

Field	Description
Trigger Condition	When the request for /oauth2/token fails for less than 5 requests.
Severity	Warning
Alert Details Provided	Expression: <pre>sum by (namespace, app, pod, nf_instance_id, resource_uri) (increase(ocsepp_network_id_validation_body_failure_total{namespace="sepp-2",resource_uri="/oauth2/token"}[2m])) > 0 and sum by (namespace, app, pod, nf_instance_id, resource_uri) (increase(ocsepp_network_id_validation_body_failure_total{namespace="sepp-2",resource_uri="/oauth2/token"}[2m])) <= 5</pre>

Table 5-418 (Cont.) SEPPN32fNetworkIDOAuthValidationFailureWarn

Field	Description
OID	1.3.6.1.4.1.323.5.3.46.1.2.4098
Metric Name	SEPPN32fNetworkIDOAuthValidationFailureWarn
Resolution	When the request for /oauth2/token stops failing or of ocsepp_network_id_validation_body_failure_total value becomes 0.

5.3.2.25.2 SEPPN32fNetworkIDOAuthValidationFailureMinor

Table 5-419 SEPPN32fNetworkIDOAuthValidationFailureMinor

Field	Description
Trigger Condition	When the request for /oauth2/token fails for more than 5 request and less than 100 requests.
Severity	Minor
Alert Details Provided	Expression: sum by (namespace, app, pod, nf_instance_id, resource_uri) (increase(ocsepp_network_id_validation_body_failure_total{namespace="sepp-2",resource_uri="/oauth2/token"}[2m])) > 5 and sum by (namespace, app, pod, nf_instance_id, resource_uri) (increase(ocsepp_network_id_validation_body_failure_total{namespace="sepp-2",resource_uri="/oauth2/token"}[2m])) <= 100
OID	1.3.6.1.4.1.323.5.3.46.1.2.4099
Metric Name	SEPPN32fNetworkIDOAuthValidationFailureMinor
Resolution	When the request for /oauth2/token requests failure goes below count of 5 or of ocsepp_network_id_validation_body_failure_total value becomes less than 5.

5.3.2.25.3 SEPPN32fNetworkIDOAuthValidationFailureMajor

Table 5-420 SEPPN32fNetworkIDOAuthValidationFailureMajor

Field	Description
Trigger Condition	When the request for /oauth2/token fails for more than 25 request and less than 500 requests.

Table 5-420 (Cont.) SEPPN32fNetworkIDOAuthValidationFailureMajor

Field	Description
Severity	Major
Alert Details Provided	Expression: sum by (namespace, app, pod, nf_instance_id, resource_uri) (increase(ocsepp_network_id_validation_body_failure_total{namespace="sepp-2",resource_uri="/oauth2/token"}[2m])) > 25 and sum by (namespace, app, pod, nf_instance_id, resource_uri) (increase(ocsepp_network_id_validation_body_failure_total{namespace="sepp-2",resource_uri="/oauth2/token"}[2m])) <= 500
OID	1.3.6.1.4.1.323.5.3.46.1.2.4100
Metric Name	SEPPN32fNetworkIDOAuthValidationFailureMajor
Resolution	When the request for /oauth2/token requests failure goes below count of 25 or of ocsepp_network_id_validation_body_failure_total value becomes less than 25.

5.3.2.25.4 SEPPN32fNetworkIDOAuthValidationFailureCritical

Table 5-421 SEPPN32fNetworkIDOAuthValidationFailureCritical

Field	Description
Trigger Condition	When the request for /oauth2/token fails for more than 50 request and less than 1000 requests.
Severity	Critical
Alert Details Provided	Expression: sum by (namespace, app, pod, nf_instance_id, resource_uri) (increase(ocsepp_network_id_validation_body_failure_total{namespace="sepp-2",resource_uri="/oauth2/token"}[2m])) > 50 and sum by (namespace, app, pod, nf_instance_id, resource_uri) (increase(ocsepp_network_id_validation_body_failure_total{namespace="sepp-2",resource_uri="/oauth2/token"}[2m])) <= 1000

Table 5-421 (Cont.) SEPPN32fNetworkIDOAuthValidationFailureCritical

Field	Description
OID	1.3.6.1.4.1.323.5.3.46.1.2.4101
Metric Name	SEPPN32fNetworkIDOAuthValidationFailureCritical
Resolution	When the request for /oauth2/token requests failure goes below count of 50 or of ocsepp_network_id_validation_body_failure_total value becomes less than 50.

5.3.2.26 Support for Max-Response Timer Header Alerts

5.3.2.26.1 ocseppMaxRespVldnFailedWarn

Table 5-422 ocseppMaxRespVldnFailedWarn

Field	Details
Trigger Condition	Gateway Late Arrival Exceptions >= 4/s and < 50/s for the past 5 minutes.
Severity	Warn
Alert details provided	Expression: <code>sum(rate(oc_ingressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m]) or rate(oc_egressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m])) by (namespace) >= 4 and sum(rate(oc_ingressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m]) or rate(oc_egressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m])) by (namespace) < 50</code>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4102
Metric used	oc_egressgateway_request_failure_total, oc_ingressgateway_request_failure_total

Table 5-422 (Cont.) ocseppMaxRespVldnFailedWarn

Field	Details
Resolution	Possible resolutions: 1. Increase the timeout duration for the maximum response timer to allow the network additional time for processing and response. 2. Monitor for network latency or congestion to identify any performance issues. 3. Evaluate resource limitations within the core network that may be impacting performance.

5.3.2.26.2 ocseppMaxRespVldnFailedMinor

Table 5-423 ocseppMaxRespVldnFailedMinor

Field	Details
Trigger Condition	Gateway Late Arrival Exceptions >= 50/s for the past 5 minutes.
Severity	Minor
Alert details provided	Expression: <pre>sum(rate(oc_ingressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m]) or rate(oc_egressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m])) by (namespace) >= 50 and sum(rate(oc_ingressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m]) or rate(oc_egressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m])) by (namespace) < 150</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4103
Metric used	oc_egressgateway_request_failure_total, oc_ingressgateway_request_failure_total

Table 5-423 (Cont.) ocseppMaxRespVldnFailedMinor

Field	Details
Resolution	Possible resolutions: 1. Increase the timeout duration for the maximum response timer to allow the network additional time for processing and response. 2. Monitor for network latency or congestion to identify any performance issues. 3. Evaluate resource limitations within the core network that may be impacting performance.

5.3.2.26.3 ocseppMaxRespVldnFailedMajor

Table 5-424 ocseppMaxRespVldnFailedMajor

Field	Details
Trigger Condition	Gateway Late Arrival Exceptions >= 150/s for the past 5 minutes.
Severity	Major
Alert details provided	Expression: <pre>sum(rate(oc_ingressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m]) or rate(oc_egressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m])) by (namespace) >= 150 and sum(rate(oc_ingressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m]) or rate(oc_egressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m])) by (namespace) < 500</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4104
Metric used	oc_egressgateway_request_failure_total, oc_ingressgateway_request_failure_total

Table 5-424 (Cont.) ocseppMaxRespVldnFailedMajor

Field	Details
Resolution	Possible resolutions: 1. Increase the timeout duration for the maximum response timer to allow the network additional time for processing and response. 2. Monitor for network latency or congestion to identify any performance issues. 3. Evaluate resource limitations within the core network that may be impacting performance.

5.3.2.26.4 ocseppMaxRespVldnFailedCritical

Table 5-425 ocseppMaxRespVldnFailedCritical

Field	Details
Trigger Condition	Gateway Late Arrival Exceptions >= 500/s for the past 5 minutes.
Severity	Critical
Alert details provided	Expression: <pre>sum(rate(oc_ingressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m]) or rate(oc_egressgateway_request_failure_total{namespace="sepp-namespace",ErrorDetail="LATE_ARRIVAL_EXCEPTION"}[5m])) by (namespace) > 500</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4105
Metric used	oc_egressgateway_request_failure_total, oc_ingressgateway_request_failure_total
Resolution	Possible resolutions: 1. Increase the timeout duration for the maximum response timer to allow the network additional time for processing and response. 2. Monitor for network latency or congestion to identify any performance issues. 3. Evaluate resource limitations within the core network that may be impacting performance.

5.3.2.27 Egress Gateway Alerts

5.3.2.27.1 egressGlobalMessageDropInfo

Table 5-426 egressGlobalMessageDropInfo

Field	Details
Trigger Condition	Egress global message drop rate detected greater than or equal to point 1 percent of total transactions.
Severity	Warn
Alert details provided	Expression: <code>sum(rate(oc_egressgateway_global_ratelimit_dropped_message_total{kubernetes_namespace="sepp-namespace"}[5m])) by (app)/sum(rate(oc_egressgateway_http_requests_total{kubernetes_namespace="sepp-namespace"}[5m])) by (app) *100 >= 0.1 < 1</code>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4091
Metric Used	oc_egressgateway_global_ratelimit_dropped_message_total
Resolution	Possible Resolutions: 1. The alert will be raised when the percentage of messages rejected for global rate limit will be greater than or equal to 0.1% of the total messages received. This will get cleared once percentage of message rejected is below 0.1% or greater than or equal to 1%. 2. If the message drop percentage different from expected value, please verify the configuration. 3. For all configuration related details, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.27.2 egressGlobalMessageDropWarn

Table 5-427 egressGlobalMessageDropWarn

Field	Details
Trigger Condition	Egress global message drop rate detected greater than or equal to 1 percent of total transactions.
Severity	Warn

Table 5-427 (Cont.) egressGlobalMessageDropWarn

Field	Details
Alert details provided	Expression: <code>sum(rate(oc_egressgateway_global_ratelimit_dropped_message_total{kubernetes_namespace="sepp-namespace"}[5m])) by (app)/sum(rate(oc_egressgateway_http_requests_total{kubernetes_namespace="sepp-namespace"}[5m])) by (app) *100 >= 1 < 10</code>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4092
Metric Used	oc_egressgateway_global_ratelimit_dropped_message_total
Resolution	Possible Resolutions: 1. The alert will be raised when the percentage of messages rejected for global rate limit will be greater than or equal to 1% of the total messages received. This will get cleared once percentage of message rejected is below 1% or greater than or equal to 10%. 2. If the message drop percentage different from expected value, please verify the configuration. 3. For all configuration related details, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.27.3 egressGlobalMessageDropMinor

Table 5-428 egressGlobalMessageDropMinor

Field	Details
Trigger Condition	Egress global message drop rate detected greater than or equal to 10 percent of total transactions.
Severity	Minor
Alert details provided	Expression: <code>sum(rate(oc_egressgateway_global_ratelimit_dropped_message_total{kubernetes_namespace="sepp-namespace"}[5m])) by (app)/sum(rate(oc_egressgateway_http_requests_total{kubernetes_namespace="sepp-namespace"}[5m])) by (app) *100 >= 10 < 25</code>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4093
Metric Used	oc_egressgateway_global_ratelimit_dropped_message_total

Table 5-428 (Cont.) egressGlobalMessageDropMinor

Field	Details
Resolution	Possible Resolutions: 1. The alert will be raised when the percentage of messages rejected for Global Rate Limit will be greater than or equal to 10% of the total messages received. This will get cleared once percentage of message rejected is below 10% or greater than or equal to 25%. 2. If the message drop percentage different from expected value, please verify the configuration. 3. For all configuration related details, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.27.4 egressGlobalMessageDropMajor

Table 5-429 egressGlobalMessageDropMajor

Field	Details
Trigger Condition	Egress global message drop rate detected greater than or equal to 25 percent of total transactions.
Severity	Major
Alert details provided	Expression: <pre>sum(rate(oc_egressgateway_global_ratelimit_dropped_message_total{kubernetes_namespace="sepp-namespace"}[5m])) by (app)/ sum(rate(oc_egressgateway_http_requests_total{kubernetes_namespace="sepp-namespace"}[5m])) by (app) *100 >= 10 < 25</pre>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4094
Metric Used	oc_egressgateway_global_ratelimit_dropped_message_total
Resolution	Possible Resolutions: 1. The alert will be raised when the percentage of messages rejected for global rate limit will be greater than or equal to 25% of the total messages received. This will get cleared once percentage of message rejected is below 25% or greater than or equal to 50%. 2. If the message drop percentage different from expected value, please verify the configuration. 3. For all configuration related details, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.3.2.27.5 egressGlobalMessageDropCritical

Table 5-430 egressGlobalMessageDropCritical

Field	Details
Trigger Condition	Egress global message drop rate detected greater than or equal to 50 percent of total transactions.
Severity	Critical
Alert details provided	Expression: <code>sum(rate(oc_egressgateway_global_ratelimit_dropped_message_total{kubernetes_namespace="sepp-namespace"}[5m])) by (app)/sum(rate(oc_egressgateway_http_requests_total{kubernetes_namespace="sepp-namespace"}[5m])) by (app) *100 >= 50</code>
OID	1.3.6.1.4.1.323.5.3.46.1.2.4095
Metric Used	oc_egressgateway_global_ratelimit_dropped_message_total
Resolution	Possible Resolutions: 1. The alert will be raised when the percentage of messages rejected for global rate limit will be greater than or equal to 50% of the total messages received. This will get cleared once percentage of message rejected is below 50%. 2. If the message drop percentage different from expected value, please verify the configuration. 3. For all configuration related details, refer to <i>Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide</i>.

5.4 Alert Configuration

This section describes how to configure alert rules for the SEPP in both Prometheus and OCI environments. It provides guidance on setting up measurement-based alert rules, where the alerting system evaluates metrics reported by SEPP microservices against specified rule conditions to generate alerts as needed.

Prometheus Alert Configuration

In a Prometheus environment, SEPP alert rules are configured based on metrics reported by SEPP components. The alerting workflow monitors these metrics and issues notifications when the defined conditions are met.

For more information about configuring SEPP alerts in Prometheus, see the “Alert Configuration” section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

OCI Alert Configuration

In OCI environments, SEPP alert rules are defined using Metric Query Language (MQL). Configuring these alerts requires creating a dedicated SEPP alert file for use with the OCI observability platform.

For more information about configuring SEPP alerts in OCI, see the “Configuring Alerts in OCI” section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

6

Protocol and Application Errors

Following are the SEPP Protocol and Application Errors:

CN32F Errors

Table 6-1 CN32F Errors

Parameters	HTTP status code	Description
INVALID_DESTINATION_URI_EXCEPTION	400 Bad Request	Invalid destination host URI
EMPTY_FORWARDED_HOST_EXCEPTION	400 Bad Request	Empty forwarded Host Header
SBI_TARGET_HEADER_NOT_FOUND_EXCEPTION	400 Bad Request	3gpp-sbi-target-apiroot header is absent
HOSTED_SEPP_EXCEPTION	400 Bad Request	Allowed Producer Remote SEPP Set routing failure Request not allowed as no consumer Remote SEPP Set found in hosted SEPP mode
SECURITY_CAPABILITY_EXCEPTION	401 Unauthorized	Invalid Security Capability negotiated
CONTEXT_NOT_ESTABLISHED_EXCEPTION	401 Unauthorized	N32fContext not established
TOPOLOGY_HIDING_EXCEPTION	403 Forbidden	Exception in Topology Hiding/ Recovery
N32F_CONTEXT_NOT_FOUND_EXCEPTION	404 Not Found	N32fContext or Remote SEPP Set not found
SERVICE_API_VALIDATION_EXCEPTION	406 Not Acceptable or User configured reject status code	Service API not in allowed list
SEPP_MEDIATION_REQUEST_REJECT	406 Not Acceptable or User configured reject status code	Request is rejected by mediation service
SEPP_MEDIATION_RESPONSE_REJECT	406 Not Acceptable or User configured reject status code	Error in response received from mediation service
NETWORK_ID_HEADER_VALIDATION_EXCEPTION	406 Not Acceptable	Network ID validation in header failed
NETWORK_ID_HEADER_VALIDATION_EXCEPTION	406 Not Acceptable or User configured reject status code	Network ID header validation failed
NETWORK_ID_BODY_VALIDATION_EXCEPTION	406 Not Acceptable or User configured reject status code	Network ID body validation failed
CN32F_MESSAGE_VALIDATION_EXCEPTION	406 Not Acceptable or User configured reject status code.	Error in request body

PN32F Errors

Table 6-2 PN32F Errors

Parameters	HTTP status code	Description
SEPP_EMPTY_CLIENT_IDENTITY_HEADER	400 Bad Request	Client identity header is absent in request
SEPP_INVALID_DESTINATION_URI	400 Bad Request	SBI target header/authority header not in right syntax/invalid destination host format
INVALID_PLMN	400 Bad Request	Invalid PLMN List in request or PLMN is in top level domain of destination FQDN
HOSTED_SEPP_EXCEPTION	400 Bad Request	Allowed producer Remote SEPP Set routing failure Request not allowed as no consumer Remote SEPP Set found in Hosted SEPP mode
SEPP_INVALID_DESTINATION_URI	400 Bad Request	SBI target header/authority header not in right syntax/invalid destination host format
SEPP_N32FCONTEXT_NOT_ESTABLISHED	401 Unauthorized	Context not in established state
TOPOLOGY_HIDING_EXCEPTION	403 Forbidden	Internal error
N32F_CONTEXT_NOT_FOUND	404 Not Found	n32fcontext not found at PSEPP or TLS identity validation failed
SERVICE_API_ALLOWED_LIST_EXCEPTION	406 Not Acceptable	Service API not in allowed list
MESSAGE_VALIDATION_FAILED_EXCEPTION	406 Not Acceptable	Message validation failed at PSEPP
SEPP_MEDIATION_REQUEST_REJECT	406 Not Acceptable or User configured reject status code	Request is rejected by mediation service
SEPP_MEDIATION_RESPONSE_REJECT	406 Not Acceptable or User configured reject status code	Error in response received from mediation service
PN32F_NETWORK_ID_HEADER_EXCEPTION	406 Not Acceptable	Network ID header validation failed
PN32F_NETWORK_ID_BODY_EXCEPTION	406 Not Acceptable	Network ID body validation failed
PREVIOUS_LOCATION_CHECK_VALIDATION_EXCEPTION	406 Not Acceptable or User configured reject status code	CAT 3 Previous Location check failed
PREVIOUS_LOCATION_CHECK_VALIDATION_EXCEPTION_ACTION	406 Not Acceptable or User configured reject status code	CAT 3 Previous location check failed due to exception

CN32C Errors

Table 6-3 CN32C Errors

Parameters	HTTP status code	Description
INTERNAL_SERVER_ERROR_EXCEPTION	500 Internal Server Error	1. Error while deleting n32f context or RPP delete. 2. Error while accessing the data base or any other unexpected coding exception.

PN32C Errors**Table 6-4 PN32C Errors**

Parameters	HTTP status code	Description
ROAMING_PARTNER_PROFILE_EXCEPTION	403 Forbidden	1. Handshake failure due to Remote SEPP is not enabled. 2. Sender Roaming Partner does not exist in the database. 3. Sender Roaming Partner FQDN is not present in SAN header FQDNs.
INCOMPLETE_REQUEST_BODY_EXCEPTION	400 Bad Request	Handshake failure due to Sender fqdn or capability list is empty.
INTERNAL_SERVER_ERROR_EXCEPTION	500 Internal Server Error	Error displayed while accessing the database or any other unexpected coding exception.

7

SEPP Response Codes

Following are the SEPP Response Codes:

N32c Handshake Response Codes

Table 7-1 N32c Handshake Response Codes

Request	Response Code	ResponseBody	Reference	Use Case	Comment/Remarks
Capability Exchange	200 OK	SecNegotiateResp Body	TS29573_N32_H andshake.yaml	Handshake success	Handshake Success
	400 Bad Request	Problem Details	TS29573_N32_H andshake.yaml	- Invalid RequestBody - Mandatory IE Missing	Handshake fail
	404 Not Found	Problem Details	3gpp Spec 29500	Incorrect API	
	405 Method Not Allowed	Problem Details	3gpp Spec 29500	When request method other than POST is used	
	411 Length Required	Problem Details	TS29573_N32_H andshake.yaml	When Content Length header is not present in request.	
	413 Payload Too Large	Problem Details	TS29573_N32_H andshake.yaml	When request entity is larger than limits defined by server.	
	415 Unsupported Media Type	Problem Details	TS29573_N32_H andshake.yaml	When payload format is not in json. When "Content-Type:application/json" is not present.	
	429 Too Many Requests	Problem Details	TS29573_N32_H andshake.yaml	Too many requests in a given amount of time ("rate limiting").	

Table 7-1 (Cont.) N32c Handshake Response Codes

Request	Response Code	ResponseBody	Reference	Use Case	Comment/Remarks
	500 Internal Server Error	Problem Details	TS29573_N32_Handshake.yaml	1. DB Access Fail during execution of handshake request in server side. 2. Any other internal error in server during execution of handshake request.	
	503 Service Unavailable	null	TS29573_N32_Handshake.yaml	When sepp backend service is down.	
	501 Not Implemented	Problem Details		When capability mismatch happens.	
	401 Unauthorized	Problem Details		When request received for unknown peer sepp.	
	403 Forbidden	Problem Details		When request received for disabled peer sepp.	

N32f Message Forwarding Response Codes**Table 7-2 N32f Message Forwarding Response Codes**

Pod Name	Response Code	ResponseBody	Use Case	Reference	Comment
cN32f-Sepp	200 OK	Success Body	Success	TS29573_JOSEProtectedMessageForwarding.yaml	NA
	404 Not Found	Problem Details	N32f Context Not Found	3gpp Spec 29500	
	401 Unauthorized	Problem Details	Context Not Established	3gpp Spec 29500	
	401 Unauthorized	Problem Details	Security Capability apart from TLS	3gpp Spec 29500	

Table 7-2 (Cont.) N32f Message Forwarding Response Codes

Pod Name	Response Code	ResponseBody	Use Case	Reference	Comment
	400 Bad Request	Problem Details	3GPP-target-sbi-apiroot/destination producer NF is absent	TS29573_JOSEP roTECTEDMessage Forwarding.yaml	
	503 Service Unavailable	Problem Details	Service Unavailable	TS29573_JOSEP roTECTEDMessage Forwarding.yaml	
pN32f-Sepp	400 Bad Request	Problem Details	3GPP-target-sbi-apiroot/destination producer NF is absent	TS29573_JOSEP roTECTEDMessage Forwarding.yaml	
	503 Service Unavailable	Problem Details	Service Unavailable	TS29573_JOSEP roTECTEDMessage Forwarding.yaml	

Note

All the exceptions/error response codes sent by the N32 Egress Gateway at consumer and PLM Egress Gateway at producer will be same and we will be adding a problem statement which contains the reason why it occurred.

A.1 Error Code Catalog

The following are available error codes categorized based on SEPP microservices and Ingress and Egress Gateways.

A.1.1 SEPP Microservice Error Catalog

The following are the available error responses in SEPP microservices.

Table 3 cn32c-svc

Error Code	Details	Cause	Possible Resolution
CN32C-DB_ERR-E001	DB Access Failed. Could not perform the query	The SEPP could not reach the database due to connectivity issues. Wrong database username/password or expired credentials. Incorrect SQL query or unsupported query syntax used.	-Check if the database server is running and reachable from the SEPP host. -Verify connection parameters (host, port, DB name, username, password). -Confirm that credentials are valid and not expired. -Look into database logs for query-related errors
CN32C-DB_ERR-E010	Unsupported PLMN Format	PLMN is not in the correct/expected format like an incorrect MCC-MNC structure	-Ensure PLMN values follow the correct format and standard. -Resend corrected request
CN32C-CFGN_ERR-E022	Roaming Partner not found or is disabled	The requested roaming partner does not exist, or it is currently disabled.	-Verify roaming partner configuration. -Enable the partner if it is required for communication
CN32C-CFGN_ERR-E023	Roaming Partner is disabled	The roaming partner exists but has been disabled.	Enable the roaming partner in SEPP configuration
CN32C-HS_ERR-E051	Could not get the next state from state machine	This indicates that the state machine couldn't transition to the next state, an error is logged, and an internal server error is thrown.	Ensure that the state machine's transition logic allows for a transition from the current state to next state

Table 3 (Cont.) cn32c-svc

Error Code	Details	Cause	Possible Resolution
CN32C-HS_ERR-E061	Duplicate Request for this SEPP	Same request was already processed	Review logs and metrics for insights into what might be causing the race conditions. Look at the frequency of context duplication errors, and check if there's a way to prevent these from occurring.
CN32C-HS_ERR-E063	Incomplete Request Body	Required parameters missing from request body.	Ensure all mandatory parameters are included in the request.

Table 4 pn32c-svc

Error Code	Details	Cause	Possible Resolution
PN32C-DB_ERR-E001	DB Access Failed. Could not perform the query	The SEPP could not reach the database due to connectivity issues. Wrong database username/password or expired credentials. Incorrect SQL query or unsupported query syntax used.	-Check if the database server is running and reachable from the SEPP host. -Verify connection parameters (host, port, DB name, username, password). -Confirm that credentials are valid and not expired. -Look into database logs for query-related errors
PN32C-DB_ERR-E010	Unsupported PLMN Format	PLMN is not in the correct/expected format like an incorrect MCC-MNC structure	Ensure PLMN values follow the correct format and standard. Resend corrected request
PN32C-CFGN_ERR-E022	Roaming Partner not found or is disabled	The requested roaming partner does not exist, or it is currently disabled.	Verify roaming partner configuration. Enable the partner if it is required for communication
PN32C-CFGN_ERR-E023	Roaming Partner is disabled	The roaming partner exists but has been disabled.	Enable the roaming partner in SEPP configuration
PN32C-HS_ERR-E051	Could not get the next state from state machine	This indicates that the state machine couldn't transition to the next state, an error is logged, and an internal server error is thrown.	Ensure that the state machine's transition logic allows for a transition from the current state to next state

Table 4 (Cont.) pn32c-svc

Error Code	Details	Cause	Possible Resolution
PN32C-HS_ERR-E061	Duplicate Request for this SEPP	Same request was already processed	Review logs and metrics for insights into what might be causing the race conditions. Look at the frequency of context duplication errors, and check if there's a way to prevent these from occurring.
PN32C-HS_ERR-E063	Incomplete Request Body	Required parameters missing from request body.	Ensure all mandatory parameters are included in the request.

Table 5 cn32f-svc

Error Code	Details	Cause	Possible Resolution
CN32F-VLDN_ERR-E101	The destination URI contains a PLMN ID that is invalid or not recognized according to 3GPP specifications.	PLMN ID format or value is incorrect or not compliant with 3GPP standards.	The 'Forwarded' header is missing the required 'host' parameter or it is empty, so the authority could not be determined.
CN32F-PROTO_ERR-E102	The 'Forwarded' header is missing the required 'host' parameter or it is empty, so the authority could not be determined	Missing or misconfigured 'Forwarded' header in the request.	Ensure the 'Forwarded' header includes a valid 'host' parameter.
CN32F-PROTO_ERR-E103	The mandatory 'Client-Identity' header is absent from the request. Authentication cannot proceed without this information	Request missing required 'Client-Identity' header.	Ensure the Client-Identity header is included and correctly populated for authentication.
CN32F-PROTO_ERR-E104	x-forwarded-host header is Absent	The request does not contain the required 'x-forwarded-host' header.	Verify the inclusion of the 'x-forwarded-host' header.
CN32F-SECNEG_ERR-E105	The negotiated security capabilities are invalid or unsupported, preventing secure communication	Security capabilities in handshake or request are not supported by SEPP or remote SEPP.	Verify supported security capabilities in SEPP configuration and align with remote SEPP.
CN32F-PROTO_ERR-E106	Request rejected because the mandatory HTTP header '3gpp-sbi-target-apiroot' is missing	Required HTTP header '3gpp-sbi-target-apiroot' not present in the request.	Include the '3gpp-sbi-target-apiroot' header in the request with correct value.
CN32F-PROTO_ERR-E107	The provided value for '3gpp-sbi-target-apiroot' is not a valid URI or does not comply with 3GPP SBI conventions	The header value is invalid or does not follow the required URI format per 3GPP SBI specifications.	Ensure the '3gpp-sbi-target-apiroot' value is a valid URI and complies with 3GPP SBI format rules.

Table 5 (Cont.) cn32f-svc

Error Code	Details	Cause	Possible Resolution
CN32F-CTX_ERR-E108	SEPP function could not locate the required N32fContext or Remote SEPP configuration during the handling of an inter-PLMN signaling message	-Missing or incorrect N32fContext configuration for the remote SEPP. -Remote SEPP configuration not set up or deleted.	Verify and configure the correct N32fContext and Remote SEPP
CN32F-CTX_ERR-E109	The forwarding context (N32fContext) required for inter-PLMN SEPP communication over the N32 interface has not been established	N32fContext creation procedure failed or was not performed before the request.	Establish the N32fContext properly before sending inter-PLMN messages.
CN32F-VLDN_ERR-E113	The destination host URI provided in the request is invalid or malformed	Destination URI is invalid or malformed	Ensure URI follows proper syntax and includes required components per SEPP configuration.

Table 5 (Cont.) cn32f-svc

Error Code	Details	Cause	Possible Resolution
CN32F-REQVLD_ERR-E141	Cat-0 Message Validation failed as request query param is having noncompliance schema. Please make the changes as per user configured or default schema	The schema for query parameters is non compliant.	To fix the issue, 1. Check Logs or Metrics: Review the following metrics for message validation failures: - ocsepp_message_validation_on_body_failure - ocsepp_message_validation_on_header_failure 2. To identify the Failing Resource URI and HTTP Method, do the following: - For request body validation failures, search for the text: "Message validation failed for request body for request" - For query parameter validation failures, search for: "Message validation failed for request query parameter(s) for request" - For more detailed information about logs, refer to <i>Troubleshooting Guide</i>. 3. In CNC Console GUI, navigate to SEPP and select Security Countermeasure from the left-hand menu. - Click Cat 0 - SBI Message Schema Validation to open the Message Validation List. - Search for the relevant resource URI to retrieve the corresponding schema. - Compare the request body or queryparameters against the schema to ensure the request complies with the schema. If necessary, update the schema to

Table 5 (Cont.) cn32f-svc

Error Code	Details	Cause	Possible Resolution
			reflect the correct structure

Table 5 (Cont.) cn32f-svc

Error Code	Details	Cause	Possible Resolution
CN32F-REQVLD_ERR-E142	Cat-0 Message Validation failed as request body is having noncompliance schema. For a successful validation, please make the request body changes to make it compliant with the user configured or default schema or turn off the Cat0 feature	The schema for request body is non compliant.	To fix the issue, - Check Logs or Metrics: Review the following metrics for message validation failures: - ocsepp_message_validation_on_body_failure - ocsepp_message_validation_on_header_failure - To identify the Failing Resource URI and HTTP Method, do the following: - For request body validation failures, search for the text: "Message validation failed for request body for request" - For query parameter validation failures, search for: "Message validation failed for request query parameter(s) for request" - For more detailed information about logs, refer to <i>Troubleshooting Guide</i>. - In CNC Console GUI, navigate to SEPP and select Security Countermeasure from the left-hand menu. - Click Cat 0 - SBI Message Schema Validation to open the Message Validation List. - Search for the relevant resource URI to retrieve the corresponding schema. - Compare the request body or query parameters against the schema to ensure the request complies with the schema. If necessary, update the

Table 5 (Cont.) cn32f-svc

Error Code	Details	Cause	Possible Resolution
			schema to reflect the correct structure
CN32F-REQVLD_ERR-E143	Cat-1 Service API validation failed	The Resource URI + Method in the incoming message is different from the configured value.	To fix the issue, 1. Check the Resource URI + Method for which the issue is observed. 2. Verify the error_msg using "ocsepp_security_service_api_failure_total" metric and KPI. 3. Fix or add configuration for the Resource URI + Method in Service API's and Allowed List.
CN32F-REQVLD_ERR-E144	Cat-1 Request Query Param validation failed	The query parameters in the incoming message is different from the configured value.	To fix the issue: 1. Verify that both the Cat-1 NRF Service API Query Parameters Validation feature and the Cat-1 Service API Validation feature are enabled. 2. Ensure that the same list-name is selected across all lists configured for query parameter validation. 3. Check the pair of Network Functions and the values included in the request. Make sure they match the configuration defined for this feature. 4. Confirm that the request is an NRF discovery request using the GET method.

Table 5 (Cont.) cn32f-svc

Error Code	Details	Cause	Possible Resolution
CN32F-REQVLD_ERR-E145	Cat-2 Network Id Validation failed on request header(s)	Invalid PLMN ID in Header configurations is received on SEPP.	To fix the issue, 1. Verify if correct regex is configured against the header identifier in Header Configuration. 2. If error is thrown from CN32F microservices, verify if MCC and MNC combination is present in the PLMN ID List based on the associated SEPP configurations. 3. If error is thrown from PN32F microservices, verify proper configurations PLMN ID list based on the associated SEPP configurations.
CN32F-REQVLD_ERR-E146	Cat-2 Network Id Validation failed on request body	Invalid PLMN ID in body configurations is received on SEPP	To fix the issue, 1. Verify if correct regex is configured against the body IE in Body IE Configuration. 2. If the error is from CN32F microservices, verify MCC and MNC combination is present in the PLMN ID List based on associated SEPP configured. 3. If the error is from PN32F microservices, verify if MCC and MNC combination is present in the PLMN ID list based on associated SEPP configured.
CN32F-TH_ERR-E162	The expected topology value was not found during the topology processing or recovery operation	Required topology data is missing or misconfigured in SEPP.	Verify topology configuration.
CN32F-TH_ERR-E163	A required pseudo value or placeholder was not found during topology processing or feature operation	Missing placeholder/ pseudo value configuration for topology hiding/ unhiding.	Check and configure missing pseudo values/ placeholders in the topology configuration.
CN32F-TH_ERR-E165	An error occurred during the topology hiding operation		

Table 5 (Cont.) cn32f-svc

Error Code	Details	Cause	Possible Resolution
CN32F-TUH_ERR-E166	An error occurred while performing the topology unhiding operation	Misconfiguration or missing data needed to unhide topology.	Review topology unhiding configuration.
CN32F-TH_ERR-E167	Invalid or missing query parameters caused an error during topology hiding or recovery operations	Required query parameters are missing or incorrectly formatted in the request.	Verify query parameter configuration and format.
CN32F-TH_ERR-E168	The provided URL is invalid or malformed.	URL syntax error or non-compliance with expected SEPP API format.	Check and correct the API URL according to SEPP specifications.
CN32F-TH_ERR-E169	The request was rejected based on policy or validation rules.		
CN32F-MED_ERR-E181	The mediation function rejected the egress request message due to policy or validation failure.	Mediation processing Failure	To fix the issue, 1. Check if the Mediation Rules exist. 2. Check the Agenda Group in the mediation rule is matching from the request and response sent from SEPP.
CN32F-MED_ERR-E182	The mediation function rejected the ingress response message due to policy or validation failure.	Mediation processing Failure	To fix the issue, 1. Check if the Mediation Rules exist. 2. Check the Agenda Group in the mediation rule is matching from the request and response sent from SEPP.
CN32F-MED_ERR-E183	The mediation function did not respond within the expected time frame, causing a timeout.	Mediation service is not accessible	To fix the issue, 1. Check if the Mediation microservice pod is up. 2. Check if Mediation Service Name and servicePort number is correct.
CN32F-MED_ERR-E184	An unexpected error occurred within the mediation function, causing the request to fail.	Mediation service is not accessible	To fix the issue, 1. Check if the Mediation microservice pod is up. 2. Check if Mediation Service Name and servicePort number is correct.
CN32F-COMP_ERR-E201	Failed to decompress the gzip-encoded content due to invalid or corrupted data.	ZipException during gzipDecompress	Inspect the gzip-encoded data

Table 5 (Cont.) cn32f-svc

Error Code	Details	Cause	Possible Resolution
CN32F-COMP_ERR-E202	An I/O exception occurred while decompressing gzip-encoded content, indicating a read/write failure.	IOException while unzipping	Ensure Permissions, system resources are correct.
CN32F-COMP_ERR-E203	An IndexOutOfBoundsException occurred during gzip decompression.	IndexOutOfBoundsException while unzipping	Make sure that the data is correctly compressed in the GZIP format
CN32F-DECODE_ERR-E204	Failed to decode the GZIP-compressed request body.	Error decoding GZIP body	Ensure that the header is correctly set to gzip if the request body is compressed using GZIP. Ensure that the GZIP compression process was done correctly before sending the request.
CN32F-COMP_ERR-E205	An I/O exception occurred while compressing data with gzip, indicating a failure in the compression process.	IOException during gzipCompress	Check if system is running low on disk space, memory, or file handles
CN32F-CFGN_ERR-E221	The request was denied because the Remote SEPP Set is not configured.	Denied request because Remote SEPP Set has not been configured	Ensure correct configurations are done for Remote SEPP Set.
CN32F-REQVLD_ERR-E231	Multipart request contains invalid header/content-type/boundary.	- Invalid or Missing Content-Type Header - Incorrect Boundary Definition - Malformed Multipart Body - Inconsistent Boundary in Content-Type and Request Body:	- Ensure Correct Content-Type Header: - Boundary Consistency - Verify the Content-Length - Check for Special Characters in Boundary - Check Multipart Body Formatting:

Table 5 (Cont.) cn32f-svc

Error Code	Details	Cause	Possible Resolution
CN32F-REQVLD_ERR-E232	Error in processing multipart request body.	- Malformed Multipart Body - Incorrect Boundary Handling - Missing or Incorrect Content-Type for Individual Parts - Special Characters or Encoding Issues in Body Content - Multipart Body Too Large - Server Timeout or Network Issues	- Inspect the Multipart Body - check service logs and metrics

Table 6 pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-VLDN_ERR-E101	The destination URI contains a PLMN ID that is invalid or not recognized according to 3GPP specifications	PLMN ID format or value is incorrect or not compliant with 3GPP standards.	-Verify PLMN ID format (MCC+MNC) matches 3GPP requirements.- Update configuration to use correct PLMN IDs.
PN32F-PROTO_ERR-E102	The 'Forwarded' header is missing the required 'host' parameter or it is empty, so the authority could not be determined	Missing or misconfigured 'Forwarded' header in the request.	Ensure the 'Forwarded' header includes a valid 'host' parameter.
PN32F-PROTO_ERR-E103	The mandatory 'Client-Identity' header is absent from the request. Authentication cannot proceed without this information.	Request missing required 'Client-Identity' header.	Ensure the Client-Identity header is included and correctly populated for authentication.
PN32F-PROTO_ERR-E104	x-forwarded-host header is Absent	The request does not contain the required 'x-forwarded-host' header.	Verify the inclusion of the 'x-forwarded-host' header.
PN32F-SECNEG_ERR-E105	The negotiated security capabilities are invalid or unsupported, preventing secure communication.	Security capabilities in handshake or request are not supported by SEPP or remote SEPP.	Verify supported security capabilities in SEPP configuration and align with remote SEPP.
PN32F-PROTO_ERR-E106	Request rejected because the mandatory HTTP header '3gpp-sbi-target-apiroot' is missing.	Required HTTP header '3gpp-sbi-target-apiroot' not present in the request.	Include the '3gpp-sbi-target-apiroot' header in the request with correct value.
PN32F-PROTO_ERR-E107	The provided value for '3gpp-sbi-target-apiroot' is not a valid URI or does not comply with 3GPP SBI conventions.	The header value is invalid or does not follow the required URI format per 3GPP SBI specifications.	Ensure the '3gpp-sbi-target-apiroot' value is a valid URI and complies with 3GPP SBI format rules.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-CTX_ERR-E108	SEPP function could not locate the required N32fContext or Remote SEPP configuration during the handling of an inter-PLMN signaling message.	- Missing or incorrect N32fContext configuration for the remote SEPP. - Remote SEPP configuration not set up or deleted.	Verify and configure the correct N32fContext and Remote SEPP
PN32F-CTX_ERR-E109	The forwarding context (N32fContext) required for inter-PLMN SEPP communication over the N32 interface has not been established.	N32fContext creation procedure failed or was not performed before the request.	Establish the N32fContext properly before sending inter-PLMN messages.
PN32F-VLDN_ERR-E113	The destination host URI provided in the request is invalid or malformed.	Destination URI is invalid or malformed	Ensure URI follows proper syntax and includes required components per SEPP configuration.
PN32F-PROTO_ERR-E121	The required 'Client-Identity' HTTP header is missing from the request. This header is necessary to identify and authenticate the client.	'Client-Identity' HTTP header is missing from the request.	Ensure that the client identity header is included in the request as it is required to identify and authenticate the client.
PN32F-AUTH_ERR-E122	The 'Client-Identity' header is present but does not conform to the required format. Ensure it follows the expected syntax and includes valid values.	'Client-Identity' HTTP header does not have the required format.	Ensure that the client identity header is included with expected syntax and also has valid values.
PN32F-PROTO_ERR-E124	The required 'X-Forwarded-Port' HTTP header is missing from the request. This header is needed to determine the original client port.	The required 'X-Forwarded-Port' HTTP header is missing from the request	Ensure that the X-Forwarded-Port header is included as it is required to determine the original client port.
PN32F-PROTO_ERR-E125	The required 'X-Forwarded-Host' HTTP header is missing from the request. This header is necessary to determine the original host requested by the client	The required 'X-Forwarded-Host' HTTP header is missing from the request	Ensure that the X-Forwarded-Host header is included as it is required to determine the original host requested by the client

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-PROTO_ERR-E126	The '3gpp-sbi-target-apiroot' or 'authority' header is invalid or improperly formatted. Ensure it includes a valid URI with scheme, host, and optional path components as per 3GPP specifications.	The '3gpp-sbi-target-apiroot' or 'authority' header is invalid or improperly formatted.	Ensure that the '3gpp-sbi-target-apiroot' or 'authority' header is invalid or improperly formatted are having the correct format as well as a valid URI including scheme, host, optional path components as per the 3GPP specifications
PN32F-VLDN_ERR-E127	The PLMN list in the request is invalid, or the PLMN appears as the top-level domain in the destination FQDN, which is not allowed.	The PLMN list in the request is invalid, or the PLMN appears as the top-level domain in the destination	Please ensure correct PLMN list is included with valid values. Ensure PLMN does not appear as the top-level domain in destination FQDN.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-REQVLD_ERR-E141	Cat-0 Message Validation failed as request query param is having noncompliance schema. Please make the changes as per user configured or default schema.	The schema for query parameters is non compliant.	To fix the issue, 1. Check Logs or Metrics: Review the following metrics for message validation failures: - ocsepp_message_validation_on_body_failure - ocsepp_message_validation_on_header_failure 2. To identify the Failing Resource URI and HTTP Method, do the following: - For request body validation failures, search for the text: "Message validation failed for request body for request". - For query parameter validation failures, search for: "Message validation failed for request query parameter(s) for request". - For more detailed information about logs, refer to <i>Troubleshooting Guide</i>. 3. In CNC Console GUI, navigate to SEPP and select Security Countermeasure from the left-hand menu. - Click Cat 0 - SBI Message Schema. - Validation to open the Message Validation List. - Search for the relevant resource URI to retrieve the corresponding schema. - Compare the request body or query parameters against the schema to ensure the request complies

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
			with the schema. If necessary, update the schema to reflect the correct structure.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-REQVLD_ERR-E142	Cat 0 Message Validation failed as request body is having noncompliance schema. For a successful validation, please make the request body changes to make it compliant with the user configured or default schema or turn off the Cat0 feature	The schema for request body is non compliant.	To fix the issue, do the following: 1. Check Logs or Metrics: Review the following metrics for message validation failures: • ocsepp_message_validation_on_body_failure • ocsepp_message_validation_on_header_failure 2. To identify the Failing Resource URI and HTTP Method, do the following: • For request body validation failures, search for the text: "Message validation failed for request body for request". • For query parameter validation failures, search for: "Message validation failed for request query parameter(s) for request." For more detailed information about logs, refer to Troubleshooting Guide. 3. In CNC Console GUI, navigate to SEPP and select Security Countermeasure from the left-hand menu:

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
			- Click Cat 0 - SBI Message Schema. - Validation to open the Message Validation List. - Search for the relevant resource URI to retrieve the corresponding schema. - Compare the request body or queryparameters against the schema to ensure the request complies with the schema. If necessary, update the schema to reflect the correct structure
PN32F-REQVLD_ERR-E143	Cat-1 Service API validation failed	The Resource URI + Method in the incoming message is different from the configured value.	To fix the issue, do the following: 1. Check the Resource URI + Method for which the issue is observed. 2. Verify the error_msg using "ocsepp_security_service_api_failure_total" metric and KPI. 3. Fix or add configuration for the Resource URI + Method in Service API's and Allowed List.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-REQVLD_ERR-E144	Cat-1 Request Query Param validation failed	The query parameters in the incoming message is different from the configured value.	To fix the issue, do the following: 1. Verify that both the Cat-1 NRF Service API Query Parameters Validation feature and the Cat-1 Service API Validation feature are enabled. 2. Ensure that the same list-name is selected across all lists configured for query parameter validation. 3. Check the pair of Network Functions and the values included in the request. Make sure they match the configuration defined for this feature. 4. Confirm that the request is an NRF discovery request using the GET method.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-REQVLD_ERR-E145	Cat-2 Network Id Validation failed on request header(s)	Invalid PLMN ID in Header configurations is received on SEPP.	To fix the issue, do the following: 1. Verify if correct regex is configured against the header identifier in Header Configuration. 2. If error is thrown from CN32F microservices, verify if MCC and MNC combination is present in the PLMN ID List based on the associated SEPP configurations. 3. If error is thrown from PN32F microservices, verify proper configurations PLMN ID list based on the associated SEPP configurations.
PN32F-REQVLD_ERR-E146	Cat-2 Network Id Validation failed on request body	Invalid PLMN ID in body configurations is received on SEPP	To fix the issue, do the following: 1. Verify if correct regex is configured against the body IE in Body IE Configuration. 2. If the error is from CN32F microservices, verify if MCC and MNC combination is present in the PLMN ID List based on associated SEPP configured. 3. If the error is from PN32F microservices, verify if MCC and MNC combination is present in the PLMN ID list based on associated SEPP configured.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-TH_ERR-E162	The expected topology value was not found during the topology processing or recovery operation.	Required topology data is missing or misconfigured in SEPP.	Verify topology configuration.
PN32F-TH_ERR-E163	A required pseudo value or placeholder was not found during topology processing or feature operation.	Missing placeholder/ pseudo value configuration for topology hiding/ unhiding.	Check and configure missing pseudo values/ placeholders in the topology configuration.
PN32F-TH_ERR-E165	An error occurred during the topology hiding operation.	Incorrect or Incomplete Topology Configuration: Network Configuration or Routing Issues	Check the configuration related to topology hiding Review the network topology and routing configurations
PN32F-TUH_ERR-E166	An error occurred while performing the topology unhiding operation.	Misconfiguration or missing data needed to unhide topology.	Review topology unhiding configuration.
PN32F-TH_ERR-E167	Invalid or missing query parameters caused an error during topology hiding or recovery operations.	Required query parameters are missing or incorrectly formatted in the request.	Verify query parameter configuration and format.
PN32F-TH_ERR-E168	The provided URL is invalid or malformed.	URL syntax error or non-compliance with expected SEPP API format.	Check and correct the API URL according to SEPP specifications.
PN32F-TH_ERR-E169	The request was rejected based on policy or validation rules.	This indicate that it violated a specific policy or validation rule within the system	Review the logs to identify the specific policy or validation rule that caused the rejection. Validate the input parameters in the request and ensure they conform to the expected format or data requirements. Verify authentication credentials, permissions, and roles to ensure they have the necessary access for the request.
PN32F-MED_ERR-E181	The mediation function rejected the egress request message due to policy or validation failure.	Mediation processing Failure	To fix the issue, 1. Check if the Mediation Rules exist. 2. Check the Agenda Group in the mediation rule is matching from the request and response sent from SEPP.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-MED_ERR-E182	The mediation function rejected the ingress response message due to policy or validation failure.	Mediation processing Failure	To fix the issue, 1. Check if the Mediation Rules exist. 2. Check the Agenda Group in the mediation rule is matching from the request and response sent from SEPP.
PN32F-MED_ERR-E183	The mediation function did not respond within the expected time frame, causing a timeout.	Mediation service is not accessible	To fix the issue, 1. Check if the Mediation microservice pod is up. 2. Check if Mediation Service Name and servicePort number is correct.
PN32F-MED_ERR-E184	An unexpected error occurred within the mediation function, causing the request to fail	Mediation service is not accessible	To fix the issue, 1. Check if the Mediation microservice pod is up. 2. Check if Mediation Service Name and servicePort number is correct.
PN32F-COMP_ERR-E201	Failed to decompress the gzip-encoded content due to invalid or corrupted data	ZipException during gzipDecompress	Inspect the gzip-encoded data
PN32F-COMP_ERR-E202	An I/O exception occurred while decompressing gzip-encoded content, indicating a read/write failure	IOException while unzipping	Ensure Permissions, system resources are correct.
PN32F-COMP_ERR-E203	An IndexOutOfBoundsException occurred during gzip decompression	IndexOutOfBoundsException while unzipping	Make sure that the data is correctly compressed in the GZIP format
PN32F-DECODE_ERR-E204	Failed to decode the GZIP-compressed request body	Error decoding GZIP body	Ensure that the Content-Encoding header is correctly set to gzip if the request body is compressed using GZIP. Ensure that the GZIP compression process was done correctly before sending the request.
PN32F-COMP_ERR-E205	An I/O exception occurred while compressing data with gzip, indicating a failure in the compression process.	IOException during gzipCompress	Check if system is running low on disk space, memory, or file handles

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-CFGN_ERR-E221	The request was denied because the Remote SEPP Set is not configured.	Denied request because Remote SEPP Set has not been configured	Ensure correct configurations are done for Remote SEPP Set.
PN32F-REQVLD_ERR-E231	Multipart request contains invalid header/content-type/boundary.	- Invalid or Missing Content-Type Header - Incorrect Boundary Definition - Malformed Multipart Body - Inconsistent Boundary in Content-Type and Request Body:	- Ensure Correct Content-Type Header: - Boundary Consistency - Verify the Content-Length - Check for Special Characters in Boundary - Check Multipart Body Formatting:
PN32F-REQVLD_ERR-E232	Error in processing multipart request body.	- Malformed Multipart Body - Incorrect Boundary Handling - Missing or Incorrect Content-Type for Individual Parts - Special Characters or Encoding Issues in Body Content - Multipart Body Too Large - Server Timeout or Network Issues	- Inspect the Multipart Body - check service logs and metrics

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-TIMEOUT_ERR-E241	The request timed out while processing at the SOR or PLMN-EGW component.	- High processing delay at SOR or PLMN-EGW due to load or performance issues. - Network latency or congestion between SEPP and SOR/PLMN-EGW.	1. Check whether the SOR feature is enabled at CNC Console or REST API, check the Remote SEPP Set, validate SOR is enabled for the given RSS. 2. Check Roaming Hub is disabled, and SEPP is deployed in SEPP Mode. 3. Check the SOR list name associated with RSS. Verify the method plus URI that is passed in the message request exists in the SOR List. 4. Verify that the server header value given at the time of configuration matches the value that reaches in server header in the response. In this case, if retry is true then retry will be performed. If server header value is not matched, even when retry is true, retry will not be performed.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-TIMEOUT_ERR-E242	Request Timeout: SOR did not respond within the designated timeframe.	- SOR was overloaded or unresponsive. - Network issues caused delayed or dropped responses.	1. Check whether the SOR feature is enabled at CNC Console or REST API, check the Remote SEPP Set, validate SOR is enabled for the given RSS. 2. Check Roaming Hub is disabled, and SEPP is deployed in SEPP Mode. 3. Check the SOR list name associated with RSS. Verify the method plus URI that is passed in the message request exists in the SOR List 4. Verify that the server header value given at the time of configuration matches the value that reaches in server header in the response. In this case, if retry is true then retry will be performed. If server header value is not matched, even when retry is true, retry will not be performed.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-CFGN_ERR-E243	An error was encountered due to incorrect or missing configuration in SOR.	- Missing or invalid parameters in SOR configuration. - Refer <i>SEPP Troubleshooting Guide</i>.	1. Check whether the SOR feature is enabled at CNC Console or REST API, check the Remote SEPP Set, validate SOR is enabled for the given RSS. 2. Check Roaming Hub is disabled, and SEPP is deployed in SEPP Mode. 3. Check the SOR list name associated with RSS. Verify the method plus URI that is passed in the message request exists in the SOR List. 4. Verify that the server header value given at the time of configuration matches the value that reaches in server header in the response. In this case, if retry is true then retry will be performed. If server header value is not matched, even when retry is true, retry will not be performed.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-REDIR_ERR-E244	SOR 3xx response received while redirection is disabled.	SOR returned a redirection response (3xx) but SEPP redirection mode is disabled.	1. Check whether the SOR feature is enabled at CNC Console or REST API, check the Remote SEPP Set, validate SOR is enabled for the given RSS. 2. Check Roaming Hub is disabled, and SEPP is deployed in SEPP Mode. 3. Check the SOR list name associated with RSS. Verify the method plus URI that is passed in the message request exists in the SOR List. 4. Verify that the server header value given at the time of configuration matches the value that reaches in server header in the response. In this case, if retry is true then retry will be performed. If server header value is not matched, even when retry is true, retry will not be performed.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-REQVLD_ERR-E150	The location check validation failed because the time authentication was not successful.	Time authentication was not successful	1. Confirm that the country related parameters such as longitude, latitude, and MCC are correctly configured to ensure accurate location validation. 2. Ensure that the incoming request includes a valid SUPI or SUCI. 3. If the incoming request contains a SUCI, the message will be forwarded to the UDM, which must be correctly discovered through NRF. 4. Check that the UDM is present and reachable in the network. 5. The UDM discovery procedure must be successful for the feature to function correctly. 6. Verify that the FQDN or IP address of the UDR is reachable from the network. This can be checked through configuration details if verbosity is disabled. 7. Verify all configurations and units of measurement.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-REQVLD_ERR-E151	An internal exception occurred during the time unauthenticated location check process.	Time unauthenticated location check failed due to internal exception.	1. Confirm that the country related parameters such as longitude, latitude, and MCC are correctly configured to ensure accurate location validation. 2. Ensure that the incoming request includes a valid SUPI or SUCI. 3. If the incoming request contains a SUCI, the message will be forwarded to the UDM, which must be correctly discovered through NRF. 4. Check that the UDM is present and reachable in the network. 5. The UDM discovery procedure must be successful for the feature to function correctly. 6. Verify that the FQDN or IP address of the UDR is reachable from the network. This can be checked through configuration details if verbosity is disabled. 7. Verify all configurations and units of measurement.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-AUTH_ERR-E152	Previous Location Check failed. Service API is not Authenticated.	Previous Location Check failed as Service API is not Authenticated.	1. Search for error codes in PN32F microservice logs. 2. Verify if the correct regex is configured in Header or Body IE configuration for UE ID and Serving Network ID under Cat-3 Previous Location Check Section under Security Countermeasure of SEPP CNC Console. 3. Verify if the MCC and MNC from serving network configured in either Header or Body is matching with the serving network name. The MCC and MNC values are part of the UDR response. Check whether the UDR response is success. 4. SUPI must be present in the incoming message, if it is configured for Cat-3 Previous Location Check. 5. UDR discovery procedure must be successful. 6. Coherence service must be up and running. 7. SUPI must be part of the IMSI range coming as part of the UDR profile received in UDR discovery response. 8. FQDN or IP of UDR must be reachable.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
			9. Proper DNS resolutions must be done for UDR discovery call, pn32f-svc for subscription use case.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
PN32F-AUTH_ERR-E153	Previous Location Check failed. Service API is not Authenticated due to internal server error or exception.	Authentication failed due to internal server error or exception.	1. Look for error codes in the PN32F microservice logs. 2. Ensure that the correct regex is configured for UE ID and Serving Network ID under the Header or Body IE configuration within the Cat - 3 Previous Location Check section of the Security Countermeasure tab in the SEPP CNC Console. 3. Check that the MCC and MNC values match the serving network name. These values are returned in the UDR response, so also confirm that the UDR response is successful. 4. If Cat - 3 Previous Location Check is enabled, ensure the SUPI is present in the incoming request message. 5. Verify the following: • Confirm the UDR discovery procedure completes successfully. • Ensure the FQDN or IP of the UDR is reachable. • Verify the Coherence service is up and running. • Make sure the SUPI falls within the IMSI range defined in the UDR profile.

Table 6 (Cont.) pn32f-svc

Error Code	Details	Cause	Possible Resolution
			6. Ensure that DNS resolution is properly configured for the UDR discovery call, specifically for the pn32f-svc service used in subscription use cases.

A.1.2 Ingress Gateway Error Catalog

The following are the available error responses in PLMN and N32 Ingress Gateways.

Table 7 Ingress Gateway Error Catalog (PLMN/N32 Ingress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
IGW-E001	CONNECTION_TIMEOUT	This error code indicates that Ingress Gateway is unable to establish connection with the back-end service in the configured connectionTimeout period.	The server is down, or it takes too long to respond	Increase the timeout value or check if the server is up and running.
IGW-E002	REQUEST_TIMEOUT	This error code indicates that request processing time is more than the configured request timeout at Ingress Gateway.	A successful TCP connection was established, but no response was received within the expected time frame.	To resolve this issue, increase the request timeout setting on the Ingress Gateway or optimize the back-end services to handle the request faster.

Table 7 (Cont.) Ingress Gateway Error Catalog (PLMN/N32 Ingress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
IGW-E003	UNKNOWN_HOST_EXCEPTION	This error code indicates that ingress-gateway is unable to resolve the configured backend service host name.	The error message indicates that the ingress gateway is unable to resolve the configured backend service hostname. This could be due to several reasons such as incorrect DNS settings, network issues, or misconfigured hostnames.	Check if the FQDN of the backend service is correctly resolved from the ingress gateway's perspective. If necessary, update the DNS records or configure the ingress gateway to use alternative routing servers. Additionally, ensure that the hostname provided in the configuration file matches the actual hostname of the backend service.
IGW-E004	CONNECT_EXCEPTION	This error code indicates that Ingress Gateway is unable to establish the connection with the back-end service.	Peer or NF service is refused due to incorrect network configuration and inactive service.	Verify peer or NF service availability, network routing, and ensure that the correct IP address and port are being used.
IGW-E005	BLACKLIST_IP_EXCEPTION	This error code indicates that BlackList IP exception has occurred while establishing new connection with blacklisted IP at ingress-gateway. IP would be blacklisted if TCP connection to this IP failed for more than 10 times.	Gateway is not able to establish TCP connection for more than 10 times.	Verify why the IP is not accepting TCP Connection
IGW-E006	CLOSED_CHANNEL_EXCEPTION	This error code indicates that closed channel exception has occurred as incoming TCP connection was abruptly terminated before ingress gateway had send back the response.	Unexpected Behaviour within the Network.	Capture Network Traces to identify the root cause of why the connection terminated abruptly.

Table 7 (Cont.) Ingress Gateway Error Catalog (PLMN/N32 Ingress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
IGW-E051	REJECTED_EXECUTION_EXCEPTION	This error code indicates that RejectedExecutionException has occurred at ingress-gateway as the incoming no. of requests are more than the queueCapacity configured in applicationThreadPoolConfig.	Value set for applicationThreadPoolConfig in configuration of Ingress Gateway is low for incoming traffic.	Increase the value of applicationThreadPoolConfig in Ingress Gateway Configuration or reduce the number of requests flowing to Ingress Gateway for incoming traffic
IGW-E052	NOT_FOUND_EXCEPTION	This error code indicates that ingress-gateway received request path with an endpoint that is not implemented in the backend service.	URI of the Request has not been configured in the routesConfig for Ingress Gateway.	Configure the required URI in routesConfig or correct the URI of the request
IGW-E053	SOCKET_EXCEPTION	This error indicates that SocketException has occurred as ingress-gateway was unable to create TCP connection.	Network connectivity issues, network restrictions, or misconfigured endpoints.	Verify if there are any network policies blocking the communication between the ingress gateway and the backend service. Confirm that the endpoint URL provided in the configuration file is correct and reachable from the ingress gateway
IGW-E054	INTERNAL_ERROR	This error code indicates that some unknown internal error occurred in ingress-gateway while processing the request.	Internal errors occur when there's an issue with improper configurations, mismatch in helm charts and versions.	Identify the cause from the error message sent back in the response and refer to the documentation for the specific feature that is being used and re verify all the configurations. Helm Charts and respective versions can also be checked.

Table 7 (Cont.) Ingress Gateway Error Catalog (PLMN/N32 Ingress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
IGW-E102	LATE_ARRIVAL_EXCEPTION	This error code indicates that late arrival exception occurred at ingress-gateway as the request has already expired as part of SbiTimer feature.	• Timezone mismatch or 12H/24H time format mismatch is leading to some time in the past in SBI Timer header for SBI Timer feature. • Header of MaxRspTime might have been configured with the wrong value. • The calculated request timeout value based on SBI Timer Headers is less than the configured requiredTime value at the route level.	Ensure that the timestamp and maximum response time attached in the header are of the correct time with respect to the time zone and not a time in the past. Verify if the requiredTime is accommodating to the calculated timeout value from SBI Timer header values.
IGW-E112	USERAGENT_VALIDATION_FAILURE_EXCEPTION	This error code indicates that validation of user-agent header failed at ingress-gateway.	The NF configured in the user-agent header of the request does not match with the configured NFs in the Ingress Gateway.	Either verify if the request has the correct NF Value configured, or if the intended NF as been added into the list of acceptable NFs at the Ingress Gateway end.
IGW-E122	GRACEFUL_SHUTDOWN	This error code indicates that requests received at ingress-gateway rejected gracefully during the ingress-gateway shutdown.	This indicates that the service/pod is shutting down and responses sent include this Error Response.	(Re) Deploy Ingress Gateway, or prevent shutdown

Table 7 (Cont.) Ingress Gateway Error Catalog (PLMN/N32 Ingress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
IGW-E132	GLOBAL_RATELIMIT	This error code indicates that requests are being rejected after exceeding the burstCapacity when global ratelimit feature enabled.	Number of requests are exceeding the bucket size set in the configurations for Ingress Gateway, which is calculated on the basis of maxTokenRequest and burstCapacity.	Modify the globalIngressRateLimiting parameter of Gateway configurations to accommodate incoming requests.
IGW-E152	RSS_BASED_RATELIMIT	This error code indicates that requests are being rejected after exceeding the burstCapacity when rss ratelimit feature enabled.	When egressrateLimiting ConfigMap is configured with burstCapacity and ratelimiting enabled, this exception is thrown when the requests count exceeds this burstCapacity and it is also based on the MNC and MCC values.	If this is unexpected behaviour, change the values/ attributes in egressrateLimiting ConfigMap.
IGW-E162	XFCC_HEADER_VALIDATION_FAILURE	This error code indicates that validation of XFCC header failed at ingress-gateway.	Indicates that the XFCC Header in the request is in the correct format but is not validated as it is the wrong value for the request.	Verify if the right value is updated in the XFCC Header based on the Ingress Gateway Deployment configuration.
IGW-E163	XFCC_HEADER_NOT_PRESENT_OR_EMPTY	This error code indicates that a request received at Ingress Gateway has no XFCC header or an empty XFCC header.	XFCC Header is absent.	Include the XFCC header in the request.
IGW-E164	XFCC_MATCHCERTCOUNT_GREATER_THAN_CERTS_IN_HEADER	This error code indicates that XFCC header validation failed due to XFCC certificate count miss match in ingress-gateway.	Ingress Gateway Configuration requires more certs than the number of certs sent in the request.	Check the number of certificates in the request sent and attach the certificates matching the configuration set for Ingress Gateway
IGW-E165	XFCC_HEADER_INVALID	This error code indicates that invalid XFCC header received at ingress-gateway.	XFCC Header present in request is not of the right format.	Fix the XFCC Header of the request and send.

Table 7 (Cont.) Ingress Gateway Error Catalog (PLMN/N32 Ingress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
IGW-E171	OAuthTokenValidationFailure	This error code indicates that validation of OAuth token failed at ingress-gateway.	OAuth token passed with the request is enclosed with values which are incorrect with respect to the configuration present at the gateway.	Verify and include the OAuth Token with the correct set of values related to the deployed Ingress Gateway config.
IGW-E172	OAuthCertExpired	This error code indicates that OAuth cert expired received at ingress-gateway.	Indicates that all the attributes are present and in the correct format, but the token has expired with respect to Ingress Gateway's configuration.	Attach an OAuth Token in the Request which hasn't expired
IGW-E173	OAuthMismatchInKid	This error code indicates that OAuth miss match in KID at ingress-gateway.	Incorrect issuer and incorrect KID Value is included in the request.	Check the issuer attribute and KID Value passed in the request.
IGW-E173	OAuthMismatchAlgorithmForKid	This error code indicates that the algorithm used while generating the token does not match the configured algorithm when the oauthValidationMode is "KID_ONLY".	The algorithm used while generating the token does not match the configured algorithm.	Check the issuer attribute and KID value passed in the request.
IGW-E174	OAuthProducerScopeNotPresent	This error code indicates that producer scope not present in the OAuth token received at ingress-gateway.	Producer Scope is not present in the token of the request	Check for Producer Scope value in the token used in the request
IGW-E175	OAuthProducerScopeMismatch	This error code indicates that producer scope present in the OAuth token miss match with producer scope configured in ingress gateway.	Producer Scope sent through the token either has wrong format, has been used incorrectly, or incorrect value is present.	Verify the Producer Scope Attribute of the token sent through the request

Table 7 (Cont.) Ingress Gateway Error Catalog (PLMN/N32 Ingress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
IGW-E176	OAUTH_MISMATCH_IN_NRF_INSTANCEID	This error code indicates that nrf instance id present in the OAuth token miss match with nrf instance id configured in ingress gateway.	Wrong issuer attribute is passed along with the access token.	Check the issue attribute attached along with the request
IGW-E176	OAUTH_MISMATCH_IN_ALGORITHM	This error code indicates that the algorithm used while generating the token does not match the configured algorithm when the oauthValidationMode is "KID_PREFERRED" or "INSTANCEID_ONLY".	The algorithm used while generating the token does not match the configured algorithm.	Correct the algorithm used.
IGW-E177	OAUTH_PRODUCER_PLMNID_MISMATCH	This error code indicates that the received OAuth token miss match with producer plmn configured in ingress gateway.	Configured PLMN ID in IGW, is not matching with the PLMN ID received through the OAuth Token.	Correct the PLMN ID in the configuration for Ingress Gateway or correct the ID in the OAuth Token itself.
IGW-E178	OAUTH_AUDIENCE_NOT_PRESENT_OR_INVALID	This error code indicates that the received OAuth token does not contain an audience or invalid audience at ingress gateway.	Token is passed with Audience parameter containing only correct nftype but in a different Case. An Audience Parameter with missing or wrong values for Nftype and NfInstanceId can also raise this error.	Verify and include the correct Nftype or/and NfInstanceId in the request.
IGW-E179	OAUTH_TOKEN_INVALID	This error code indicates that invalid OAuth token received at ingress-gateway.	Indicates that the OAuth Token received at Ingress Gateway is not valid, i.e., not in the correct format.	Verify if the OAuth Token is in the right format and configured with the right set of values with respect to the configuration at the gateway.

Table 7 (Cont.) Ingress Gateway Error Catalog (PLMN/N32 Ingress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
IGW-E180	OAUTH_TOKEN_ABSENCE	This error code indicates that a request was received at Ingress Gateway with no OAuth token or an empty OAuth token.	The OAuth Token is missing in the request.	Include the OAuth token in the request.
IGW-E182	IGW_OVERLOAD_CONTROL_EXCEPTION	This error code indicates that overload control exception occurred at Ingress Gateway.	Occurs when the OverloadControl feature is enabled and configured at the requests per second being sent.	Modify the configuration for acceptable requests per second.
IGW-E183	RESPONSE_STATUS_EXCEPTION	This error code indicates that ResponseStatusException was created by Spring boot as Ingress Gateway exception handler was unable to handle an exception.	NA	NA
IGW-E191	CCA_HEADER_VALIDATION_FAILURE	This error code indicates that validation of CCA header failed at ingress-gateway.	CCA Header sent through the request, failed while validation.	Include the correct value for CCA Header in the request
IGW-E192	PARTIAL_SHUTDOWN	This error code indicates that requests are being discarded partially at ingress-gateway when Control shutdown feature enabled.	Behaviour is observed when ControlledShutdown is enabled.	Disable ControlledShutdown
IGW-E193	COMPLETE_SHUTDOWN	This error code indicates that requests are being discarded completely at ingress-gateway when Control shutdown feature enabled.	Behaviour is observed when ControlledShutdown is enabled.	Disable ControlledShutdown
IGW-E200	IGW_TRAFFIC_REJECTION	This error code indicates all requests will be rejected at ingress-gateway when traffic reject feature enabled.	trafficrejectmode has been enabled through REST configuration for Ingress Gateway.	Disable trafficrejectmode through REST configuration for Ingress Gateway

Table 7 (Cont.) Ingress Gateway Error Catalog (PLMN/N32 Ingress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
IGW-E201	IGW_OVERLOAD_CONTROL_PERCENTAGE_DISCARD	This error code indicates that requests are being rejected when load level matches with configured percentage in discard policy mapping of overload control feature.	The number of requests have reached the configured percentage for OverloadControl. When this value has reached, the respective Error Code is displayed.	Change the percentage value to accommodate more requests and not to display the error code.
IGW-E202	IGW_OVERLOAD_CONTROL_PRIORITY_DISCARD	This error code indicates that requests with sbi-priority header with value greater than configured number of discarded load level matches in discard policy mapping of overload control feature.	The number of requests have reached the configured priority for OverloadControl. When this value has reached, the respective Error Code is displayed.	Change the priority value to accommodate more requests and not to display the error code.
IGW-E211	IGW_ROUTE_RATELIMITER_PRIORITY_DISCARD	This error code indicates that requests are being rejected when priority level matches with configured priority in discard policy mapping of route level ratelimiting feature.	Request is rejected because priority level set in the SBI header is greater than the configured priority in the route level rate limiting, discard policy mapping.	Increase the Priority in discard policy mapping for route level rate limiting, so requests with higher priority are accepted.
IGW-E220	IGW_PLMN_EGRESS_RATELIMIT	This error code indicates that requests are being rejected after exceeding the burstCapacity when plmn egress ratelimit feature enabled	When egressrateLimiting ConfigMap is configured with burstCapacity and ratelimiting enabled, this exception is thrown when the requests count exceeds this burstCapacity and it is also based on the MNC and MCC values.	If this is unexpected behaviour, change the values/ attributes in egressrateLimiting ConfigMap.

Table 7 (Cont.) Ingress Gateway Error Catalog (PLMN/N32 Ingress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
IGW-E221	INVALID_ROUTE_EXCEPTION	This error code indicates invalid route exception has occurred as invalidRouteFilter was configured in the routesConfiguration.	Invalid Route Exception is a filter enabled at route level in routesConfig.	If this is unexpected behaviour, check the path if that is the expected path that was supposed to be marked as an invalid route.
IGW-E223	OAuth_ALGORITHM_INVALID	This error code indicates that the algorithm used while generating the token is not a valid algorithm.	Algorithm used while generating token is not a valid algorithm.	Provide the valid algorithm while generating the token.

A.1.3 Egress Gateway Error Catalog

The following are the available error responses in PLMN and N32 Egress Gateways.

Table 8 Egress Gateway Error Catalog (PLMN/N32 Egress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
EGW-E001	CONNECTION_TIMEOUT	This error code indicates that Egress Gateway is unable to establish the connection with peer NF in the configured connectionTimeout period.	The server is down, or it takes too long to respond.	Increase the timeout value or check if the server is up and running.
EGW-E002	REQUEST_TIMEOUT	This error code indicates that request processing time is more than the configured request timeout at Egress Gateway.	A successful TCP connection was established, but no response was received within the expected time frame.	To resolve this issue, increase the request timeout setting on the Egress Gateway or optimize the back-end services to handle the request faster.

Table 8 (Cont.) Egress Gateway Error Catalog (PLMN/N32 Egress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
EGW-E003	UNKNOWN_HOST_EXCEPTION	This error code indicates that Egress Gateway is unable to resolve the peer NF service host name.	The error message indicates that Egress Gateway is unable to resolve the configured back-end service hostname. This could be due to several reasons, such as incorrect DNS settings, network issues, or misconfigured hostnames.	Ensure that the hostname provided in the configuration file matches the actual hostname of the back-end service. If required, update the DNS records or configure Gateway Services to use Alternative Routing Service.
EGW-E004	CONNECT_EXCEPTION	This error code indicates that Egress Gateway is unable to establish the connection with peer NF service.	Peer or NF service is refused due to incorrect network configuration and inactive service.	Verify peer or NF service availability, network routing, and ensure that the correct IP address and port are being used.
EGW-E005	INVALID_HTTP_REQUEST_EXCEPTION	This error code indicates an exception occurred because HTTP1.1 is disabled, but the oc-http-version header has a value http1.	The enableOutgoingHttp1 parameter of Egress Gateway is set to false.	Set the enableOutgoingHTTP1 parameter in Egress Gateway config to true.
EGW-E006	CLOSED_CHANNEL_EXCEPTION	This error code indicates that a closed channel exception has occurred as the incoming TCP connection was abruptly terminated before Egress Gateway sent the response.	Unexpected behavior is observed within the network.	Capture network traces to identify the cause why the connection was terminated abruptly.
EGW-E007	SSL_HANDSHAKE_EXCEPTION	This error code indicates the SSL handshake exception occurred in Egress Gateway.	Incorrect certificates are used to establish the connection. Use of incorrect certificates or cipher suites mismatch or TLS version incompatibility.	Check certificates, Cipher suites, and TLS version match between the client and server.

Table 8 (Cont.) Egress Gateway Error Catalog (PLMN/N32 Egress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
EGW-E051	REJECTED_EXECUTION_EXCEPTION	This error code indicates that RejectedExecutionException has occurred at Egress Gateway as the number of incoming requests is more than queueCapacity configured in applicationThreadPoolConfig.	Value set for the applicationThreadPoolConfig configuration in Egress Gateway is low for incoming traffic.	Increase the value of applicationThreadPoolConfig in Egress Gateway Configuration or reduce the number of requests to Egress Gateway for incoming traffic.
EGW-E052	NOT_FOUND_EXCEPTION	This error code indicates that Egress Gateway received a request path with an endpoint that is not implemented in the peer NF service.	URI of the request has not been configured in the routesConfig for Egress Gateway.	Configure the required URI in routesConfig or correct the URI of the request.
EGW-E054	INTERNAL_ERROR	This error code indicates that some unknown internal error occurred in Egress Gateway while processing the request.	NA	NA

Table 8 (Cont.) Egress Gateway Error Catalog (PLMN/N32 Egress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
EGW-E102	LATE_ARRIVAL_EXCEPTION	This error code indicates that late arrival exception has occurred at Egress Gateway as the request has already expired as part of the SbiTimer feature.	• Timezone mismatch or 12H/24H time format mismatch is leading to some time in the past in SBI Timer header for SBI Timer feature. • Header of MaxRspTime might have been configured with the wrong value. • The calculated request timeout value based on SBI Timer Headers is less than the configured requiredTime value at the route level.	Ensure that the timestamp and maximum response time attached in the header are of the correct time with respect to the time zone and not a time in the past. Verify if the requiredTime is accommodating to the calculated timeout value from SBI Timer header values.
EGW-E112	NOTIFICATION_RATE_LIMIT	This error code indicates that requests are being rejected after exceeding the bucketCapacity when the notificationRateLimit feature is enabled.	Request count is higher than the configured bucketCapacity at notificationRateLimit in Egress Gateway configurations.	Either increase the bucketCapacity or reduce the number of requests.
EGW-E122	RETRY_AFTER_ERROR	This error code indicates that the retry-after feature is enabled and producer FQDN is blocklisted, therefore the request cannot be fulfilled.	The RetryAfter feature is enabled, and the blacklistPeriod is completed for the peer. But when the peer is accessed when it is blocklisted, this error is displayed.	If not the intended behavior, disable the RetryAfter feature.

Table 8 (Cont.) Egress Gateway Error Catalog (PLMN/N32 Egress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
EGW-E127	GRACEFUL_SHUTDOWN	This error code indicates that the service shutdown is in progress.	During the processing of requests, if the IP changes and DNS refresh occurs.	Wait for the IP reset and DNS refresh to complete. After establishing connection to the service, try sending the request again.
EGW-E132	VIRTUAL_HOST_RESOLUTION_ERROR	This error code indicates that the DNS resolution of virtual FQDN failed.	The vFQDN provided for the virtual host cannot be resolved by the DNS servers	• Verify that the vFQDN is correctly spelled and formatted. • Ensure that the DNS server is operational. • Confirm that the FQDN exists in the DNS records.
EGW-E152	INVALID_OAUTH_TOKEN_REQUEST	This error code indicates the exception occurred because of a bad AccessToken request received.	The OAuth token received might be of the incorrect format or value.	Check the correct format or value on the basis of the configurations at Egress Gateway.
EGW-E153	OAUTH_INTERNAL_ERROR	This error code indicates that Egress Gateway is unable to send an AccessToken request to NRF.	InternalServerError because of error while processing or sending a request to NRF.	NA
EGW-E154	OAUTH_TOKEN_RETRIEVAL_FAILURE	This error code indicates an error in retrieving a token from NRF.	Error while retrieving the token from a response received from NRF.	NA
EGW-E155	OAUTH_NRF_RESPONSE_FAILURE	This error code indicates response failure from NRF (invalid/bad response) while trying to fetch OAuth token.	Error while processing the response received from NRF.	NA
EGW-E172	EGRESS_RATE_LIMITING	This error code indicates that requests are being rejected after exceeding the bucketCapacity when the egressRateLimiting feature is enabled.	The number of requests is exceeding the bucket size set in the configurations for Egress Gateway.	Edit the bucket size in Egress Gateway configurations, or reduce the rate of requests.

Table 8 (Cont.) Egress Gateway Error Catalog (PLMN/N32 Egress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
EGW-E182	EGW_CUSTOM_EXCEPTION	This error code indicates that custom exception occurred in some feature at Egress Gateway because some conditions failed to fulfill at runtime, such as SbiRouting, SeppProxyFilter, PeerBlackListHelper, EgressGatewayFilter, and so on.	NA	Identify the error from the response received, and verify with the respective documentation.
EGW-E183	RESPONSE_STATUS_EXCEPTION	This error code indicates that ResponseStatusException was created by Spring boot as Egress Gateway exception handler was unable to handle an exception.	NA	NA
EGW-E192	PARTIAL_SHUTDOWN	This error code indicates that requests are being discarded partially at Egress Gateway when the control shutdown feature is enabled.	Behavior is observed when ControlledShutdown is enabled.	Disable the ControlledShutdown configuration.
EGW-E193	COMPLETE_SHUTDOWN	This error code indicates that requests are being discarded completely at Egress Gateway when the control shutdown feature is enabled.	Behavior is observed when ControlledShutdown is enabled.	Disable the ControlledShutdown configuration.
EGW-E202	INVALID_ROUTE_EXCEPTION	This error code indicates that an invalid route exception has occurred as invalidRouteFilter was configured in routesConfiguration.	Invalid route exception filter is enabled at the route level in routesConfig.	If this is unexpected behavior, check if the expected path was supposed to be marked as an invalid route.

Table 8 (Cont.) Egress Gateway Error Catalog (PLMN/N32 Egress Gateway)

Error Code	Details	Description	Cause	Possible Resolution
EGW-E206	IP_TYPE_MISMATCH_EXCEPTION	This error code indicates that there is a mismatch in the configured EgressRoutingMode IP type and the DNS resolved IP type.	The type of IP resolved by DNS is different from the IP type configured at Egress Gateway.	Verify if Gateway Services is deployed in a dual stack IP setup.

A.2 5G CNC Mediation Service RuleBook

Header Rule APIs

This section describes about the 5G CNC Mediation Service RuleBook.

Table 9 Header Rule APIs

S.no	API	Return type	Description	Rule Example
1	get(String key)	String	Returns a header value with that key.	Request(headers.get("x-number"))
2	has(String key)	Boolean	Returns true/false depending on whether the header is present or not.	Request(headers.has("3gpp-Sbi-Producer-Id"))
3	has(String key, String value)	Boolean	Returns true/false depending on whether the header is present with that value or not.	Request(headers.has("3gpp-Sbi-Producer-Id","nfinst=1faf1bbc-6e4a-3994-a507-a14ef8e1bc23"))
4	put(String key, String value)	Boolean	Add the header with given key and value if does not find or updates the header with value if the given key found.	req.headers.put("x-original-authority", "10.172.19.110:8080")
5	add(String key, String value);	Boolean	Add the header with given key and value if does not find	req.headers.add("Content-type","application/json")
6	set(String key, String value);	Boolean	Updates the header with value if the given key found	req.headers.set("3gpp-Sbi-Producer-Id","nfinst=1faf1bbc-6e4a-3994-a507-a14ef8e1bc25")
7	set(String key, String oldValue, String newValue);	Boolean	Update the header with new value if the existing value matches with oldValue	req.headers.set("x-number", "2", "3")
8	del(String key)	Boolean	Delete the header	req.headers.del("3gpp-Sbi-Message-Priority")
9	del(String key, String value)	Boolean	Delete the header if it matches with given key and value	req.headers.del("x-forwarded-NF","NRF")
10	count();	Integer	Returns the count of total numbers of all headers.	Request(headers.count())

Table 9 (Cont.) Header Rule APIs

S.no	API	Return type	Description	Rule Example
11	count(String key);	Integer	Returns the count of total numbers of headers with the key.	Request(headers.count("Accept"))
12	count(String key, String value)	Integer	Returns the count of total numbers of headers with the key and value	Request(headers.count("Accept", "application/json"))

Example:

Rules configured:

```

rule "New Rule1"
when
    req : Request(headers.get("x-number")== 2)
then
    req.headers.put("x-original-authority", "10.172.19.110:8080")
end
rule "New Rule2"
when
    req : Request(headers.has("3gpp-Sbi-Producer-Id") && headers.has("3gpp-Sbi-Producer-Id", "nfinst=1faf1bbc-6e4a-3994-a507-a14ef8e1bc23"))
then
    req.headers.set("3gpp-Sbi-Producer-Id", "nfinst=1faf1bbc-6e4a-3994-a507-a14ef8e1bc25")
    req.headers.set("x-number", "2", "3")
end
rule "New Rule3"
when
    req : Request(headers.count() == 6)
then
    req.headers.del("3gpp-Sbi-Message-Priority")
    req.headers.del("x-forwarded-NF", "NRF")
end
rule "New Rule4"
when
    req : Request(headers.count("Accept") == 2 && headers.count("Accept", "application/json") == 1)
then
    req.headers.add("Content-type", "application/json")
end

```

Request:

```

curl -v -H 'x-forwarded-NF: NRF' -H 'x-number: 2' -H 'Accept: application/json' -H 'Accept: application/xml' -H '3gpp-Sbi-Producer-Id:nfinst=1faf1bbc-6e4a-3994-a507-a14ef8e1bc23' -H '3gpp-Sbi-Message-Priority: true' 'http://localhost:9090/nmediation-http/v1/'

```

Response:

```

HTTP/1.1 200 OK

```

Connection: keep-alive
 Content-Type: application/json
 Content-Length: 339

```
{"mediationStatus":"SUCCESS","updatedHeaders":{"x-number":[["2","3"]],"3gpp-Sbi-Producer-Id":["nfinst=1faf1bbc-6e4a-3994-a507-a14ef8e1bc23","nfinst=1faf1bbc-6e4a-3994-a507-a14ef8e1bc25"]}}, "uri":"http://localhost:9090/nmediation-http/v1/", "addedHeaders":{"x-original-authority":["10.172.19.110:8080"],"Content-type":["application/json"]}}
```

Body Rule APIs

The following are the new body APIs which are being supported in mediation from release 22.2.0. The paths in all the parameters should always be in the JSONPath format.

Table 10 Body Rule APIs

S.no	API	Return type	Description	Example
1	get(String path)	Object	Returns the value present at that path.	body.get("\$.service Name")
2	getAll(String path)	List<Object>	Returns the list of objects present at the path given.	body.getAll("\$.service Name")
3	has(String path);	boolean	Checks if the key is present at that path or not.	body.has("\$.service Name")
4	has(String path, Object value)	boolean	Checks if the key has the value present at that path or not.	body.get("\$.service Name","value")
5	absPath(String arrayPath, String elem, Object value)	Object	Returns the whole path from head to the elem if the value is present.	req.body.absPath("\$.payload","iePath","/a")
6	indexOf(String arrayPath, String elem, Object value)	Integer	Returns the index of the value in that JSONArray	req.body.indexOf("\$.dataToIntegrityProtectBlock.payload","value", 100)
7	put(String path, String key, T value)	boolean	Add the body with given key and value at path if does not find or updates the body with value at the path if the given key found.	req.body.put("\$","key","value")
8	add(String path, T value)	boolean	Adds the key embedded in the path with value given if element is not present.	req.body.add("\$.ip EndPoints","value")

Table 10 (Cont.) Body Rule APIs

S.no	API	Return type	Description	Example
9	set(String path, T value)	boolean	Updates the key embedded in the path with value given if element is present.	rsp.body.add("\$.ipEndpoints","value")
10	del(String path, T value)	boolean	Deletes the element at that path with the value present. Not supported. User can use the del (String path) api for delete.	req.body.del("\$.path","value")
11	del(String path)	boolean	Deletes the element at that path	req.body.del("\$.path")

Example:

Rules configured:

```
function Map<String,Object> ipEndpoints(String ipv4Address, String
ipv6Address){
    Map< String,Object> ipEndPointObj = new HashMap< String,Object>();
    ipEndPointObj.put("ipv4Address", ipv4Address);
    ipEndPointObj.put("ipv6Address", ipv6Address);
    return ipEndPointObj;
}
rule "Rule_Request_body_1"
when
    req : Request(body.has("$.ipEndpoints"))
then
    req.body.put("$", "dnn", "d6n1")
    req.body.set("$.fqdn", "amf.oracle.com")

req.body.add("$.ipEndpoints",ipEndpoints("10.75.243.193", "2001:0db8:85a3:0000:
0000:8a2e:0370:7335"))
end
rule "Rule_Request_body_2"
when
    req : Request(body.get("$.supi")=="imsi-1100000001" &&
body.has("$.suppFeat", "All"))
then
    req.body.del("$.supiOrSuci")
end
```

Request:

```
curl -v -H 'x-forwarded-NF: NRF' -H 'Accept: application/json' -H 'Content-
type: application/json' -d '{"ipEndpoints":
[{"ipv4Address":"10.75.243.203", "ipv6Address":"2001:db8:85a3:8d3:1319:8a2e:370
:7348"}], "servingNetworkName":"5G:mnc654.mcc987.3gppnetwork.org", "supiOrSuci":
"suci-0-987-654-0064-0-0-000000001", "fqdn":
"pcf.oracle.com", "notificationUri": "http://
```

```
pcf1svc.trafficiot.odyssey.lab.us.oracle.com:80/notification/update",
"pduSessionId": "2347", "sliceInfo": "adfg127", "supi": "imsi-1100000001",
"suppFeat": "All" }' 'http://localhost:9090/nmediation-http/v1/'
```

Response:

HTTP/1.1 200 OK

Connection: keep-alive

```
{"mediationStatus":"SUCCESS","data":
```

```
{"servingNetworkName":"5G:mnc654.mcc987.3gppnetwork.org","ipEndPoints":
```

```
[{"ipv6Address":"2001:db8:85a3:8d3:1319:8a2e:370:7348","ipv4Address":"10.75.243.203"},
```

```
{"ipv6Address":"2001:0db8:85a3:0000:0000:8a2e:0370:7335","ipv4Address":"10.75.243.193"}], "fqdn":"amf.oracle.com", "notificationUri":"http://
```

```
pcf1svc.trafficiot.odyssey.lab.us.oracle.com:80/notification/
update", "dnn":"d6n1", "pduSessionId":"2347", "sliceInfo":"adfg127", "supi":"imsi-1100000001", "suppFeat":"All"}, "uri":"http://localhost:9090/nmediation-
http/v1/"}
```

Http Message Common APIs

The following are the common APIs supported for request and response body.

Table 11 Http Message Common APIs

S.no	API	Return Type	Description	Example
1	getUri()	String	returns the URI of the Request	req.getUri()
2	setUri(String URI)	void	Sets the URI in the body of request	req.setUri(" http://oracle.com ")
3	getHttpStatusCode()	Integer	returns the StatusCode from the body of Response	rsp.getHttpStatusCode()
4	setHttpStatusCode(Integer httpStatusCode)	void	sets the StatusCode in the body of Response	rsp.setHttpStatusCode(201)
5	getRejectMessage()	String	returns the Reject Message from the body of Response	rsp.getRejectMessage()
6	setRejectMessage(String rejectMessage)	void	sets the Reject Message in the body of Response	rsp.setRejectMessage("Message")

Other Support for Rules

This section describes methods to create Mediation rules.

Regex Support:

The following example defines rules using regular expression on header API.

```
rule "regex rule"
when
  req : Request(headers.get("x-number") matches "[0-9]")
then
  req.headers.put( "Header name", "value")
```

end

```
rule "regex rule2"
when
    req : Request(body.get("$.serviceName").toString() matches
"suci-.*0-0-0.*")
then
    req.body.put("$", "new key", "value")
end
```

Example:

```
suci-0-123-45-0-0-0-0-1-17-
e9b9916c911f448d8792e6b2f387f85d3ecab9040049427d9edbb5431b0bc711023be6a057f349
56ba21b45d936238aeb7
```

Operator on header API

The following example defines rules using in operator on header API.

```
rule "New Rule5"
when
    req : Request(headers.get("x-number") in (1,2,3,4,5 ))
then
    req.headers.put( "Header name", "value")
end
```

Function Support

The following example defines rules using functions:

```
function String update(String str){
    int a = Integer.parseInt(str);
    a = a+1;
    str = ""+a;
    return str;
}

rule "Function"
when
    req : Request(headers.get("ADD")==1)
then
    req.headers.set("ADD",update(req.headers.get("ADD")))
end
```

Configuring Mediation Rules

Configuring Mediation Rules using ConfigMap

The following are the steps to configure the rules:

1. Download the configMap into a file using the following command:

```
kubectll get configmap ocsepp-nf-mediation-config-active -n seppsvc -oyaml
>> rule_configmap.txt
```

2. At the end of the file **rule_configmap.txt** , append the below **data** block along with needed rules:

```
data:
  rule.drl: |
    package com.oracle.cgbu.ocmediation.nfmediation;

    import com.oracle.cgbu.ocmediation.nfruleengine.NFDroolsRuleEngine;
    import com.oracle.cgbu.ocmediation.factdetails.Request
    import com.oracle.cgbu.ocmediation.factdetails.Response;
    import java.util.Map;
    import java.util.HashMap;

    dialect "mvel"

    //rules
```

Example of updated file:

```
apiVersion: v1
kind: ConfigMap
metadata:
  creationTimestamp: "2022-08-10T14:01:27Z"
  name: ocsepp-nf-mediation-config-active
  namespace: seppsvc
  resourceVersion: "1635604"
  uid: b4ff6628-4e84-4aa7-8116-4daecbff8712
data:
  rule.drl: |
    package com.oracle.cgbu.ocmediation.nfmediation;

    import com.oracle.cgbu.ocmediation.nfruleengine.NFDroolsRuleEngine;
    import com.oracle.cgbu.ocmediation.factdetails.Request
    import com.oracle.cgbu.ocmediation.factdetails.Response;
    import java.util.Map;
    import java.util.HashMap;

    dialect "mvel"

    rule "Rule_show_add_header"
    when
      req : Request(headers.has("3gpp-Sbi-Message-Priority") == true)
    then
      req.headers.add("3gpp-Sbi-Message-Priority","10")
      req.headers.del("x-test-req-header1")
    end

    rule "Rule_Request_body_add_delete"
    when
      req : Request(headers.has("x-forwarded-NF","NRF"))
    then
      req.body.put("$","nfId","ccc5ccb-5bb9-465f-9ace-0faf08cb4223")
      req.body.del("$.supiOrSuci")
    end
```

3. Run the following command to update the configmap with the updated rules:

```
kubectl replace ocsepp-nf-mediation-config-active -n seppsvc -f
rule_configmap.txt
```

Note

- If rules changed on nf active mediation then use **"\$releaseName-nf-mediation-config-active"** as the name of the configmap.
- It is recommended to maintain the back up of the rules as it can lead to rule loss if the config map deleted

Rules for NF-Mediation:

```
function Map<Object, Object> addObject() {
    return new HashMap<Object, Object>();
}

function ArrayList<Object> addArray() {
    return new ArrayList<Object>();
}

function Map<String, Object> ipEndpoints(String ipv4Address, String
ipv6Address, String ipv6Prefix){
    Map< String, Object> ipEndPointObj = new HashMap< String, Object>();
    ipEndPointObj.put("ipv4Address", ipv4Address);
    ipEndPointObj.put("ipv6Address", ipv6Address);
    ipEndPointObj.put("ipv6Prefix", ipv6Prefix );
    return ipEndPointObj;
}

rule "New Rule1"
    agenda-group "scp"
    when
        req : Request(body.get("$.apiPrefix")=="/mediation")
    then
        req.headers.put("x-dest-path", req.body.get("$.apiPrefix").toString())
    end

rule "New Rule2"
    when
        req :
        Request(body.has("$.defaultNotificationSubscriptions[*].callbackUri", "pcf.orac
le.com/pcs/v1.0/nrf/notifcallback"))
    then

        req.body.add("$.ipEndpoints", ipEndpoints("10.75.243.193", "2001:0db8:85a3:0000:
0000:8a2e:0370:7335", "2001:0db8:85a3"))

    end

rule "New Rule3"
    when
```

```

        req : Request(headers.has("x-forwarded-NF", "PCF") &&
body.has("$.fqdn", "pcf.oracle.com"))
    then
        req.headers.put("x-forwarded-NF", "AMF")
        req.body.put("$", "fqdn", "amf.oracle.com")
    end

rule "New Rule4"
when
    req : Request(body.get("$.serviceName")== "svc1")
then
    req.headers.del("x-service-name")
end

rule "New Rule5"
when
    req : Request(headers.get("x-number") in (1,2,3,4,5 ))
then
    req.headers.put( "x-original-authority", "10.172.19.110:8080")
end
Configuring rules for PT

rule "pt_d2h_to_nf_rule"
salience 1
when
    req : Request(headers.has("pt_dest_uri"))
then

    req.setUri(req.headers.get("pt_dest_uri").replace("pcf.com", "10.75.203.74:1000
/simulation"))
    req.headers.del("pt_dest_uri")
end

rule "pt_rar_to_pcf_rule"
salience 1
when
    req : Request(req.getUri().toString() matches ".*(npcf-
policyauthorization)*(v1)*(notification)*(notify).*")
then
    req.getHttpMessageFactImpl().forwardPath = IWFCOnsts.FORWARD_TO_H2D
    req.headers.add("diameterApplicationId", "16777236")
    req.headers.add("diameterCommandCode", "258")
    req.headers.add("original-req-uri", req.getUri())
end

rule "pt_aar_to_pcf_rule"
salience 1
when
    req : Request(req.body.has("$.ascReqData.notifUri"))
then

    req.body.add("$.ascReqData.notifUri", req.body.get("$.ascReqData.notifUri").toS
tring().replace("iwf.com", "10.75.203.74:30079"))

    req.body.add("$.ascReqData.evSubsc.notifUri", req.body.get("$.ascReqData.evSubs
c.notifUri").toString().replace("iwf.com", "10.75.203.74:30079"))

```

```

end

rule "pt_asr_to_pcf_rule"
salience 1
when
  req : Request(getUri().toString() matches ".*(npcf-
policyauthorization)*(v1)*(notification)*(terminate).*")
then
  req.getMessageFactImpl().forwardPath=IWFCnsts.FORWARD_TO_H2D
  req.headers.add("diameterApplicationId", "16777236")
  req.headers.add("diameterCommandCode", "274")
  req.headers.add("original-req-uri", req.getUri())
end

rule "pt_ccri_to_h2d_rule"
salience 1
when
  req : Request(req.getUri() matches ".*(npcf-smpolicycontrol/v1/sm-policies)
(/$|$)")
then
  req.getMessageFactImpl().forwardPath=IWFCnsts.FORWARD_TO_H2D
  req.headers.put("diameterApplicationId", "16777238")
  req.headers.put("diameterCommandCode", "272")
  req.headers.put("requestType", "CREATE")
end

rule "pt_ccru_to_h2d_rule"
salience 1
when
  req : Request(req.getUri() == ".*(npcf-smpolicycontrol/v1/sm-policies/)(.*/
update)/($|$)")
then
  req.getMessageFactImpl().forwardPath=IWFCnsts.FORWARD_TO_H2D
  req.headers.put("diameterApplicationId", "16777238")
  req.headers.put("diameterCommandCode", "272")
  req.headers.put("requestType", "UPDATE")
  req.headers.put("original-req-uri", req.getUri())
end

rule "pt_ccrt_to_h2d_rule"
salience 1
when
  req : Request(req.getUri() matches ".*(npcf-smpolicycontrol/v1/sm-policies/)
(.*/delete)/($|$)")
then
  req.getMessageFactImpl().forwardPath=IWFCnsts.FORWARD_TO_H2D
  req.headers.put("diameterApplicationId", "16777238")
  req.headers.put("diameterCommandCode", "272")
  req.headers.put("requestType", "DELETE")
  req.headers.put("original-req-uri", req.getUri())
end

```

A.3 Default SEPP Service APIs

The following are the list of default SEPP Service APIs supported by SEPP features:

Table 12 Default SEPP Service APIs

Sr. No	HTTP Method	Resource URI	Reference (GSMA FS.36 Version 2.2) or 3GPP Release Version
1	POST	/namf-evts/v1/subscriptions	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29518-ge0)
2	DELETE	/namf-evts/v1/subscriptions/{subscriptionId}	3GPP Release Version: 16.14.0
3	PATCH	/namf-evts/v1/subscriptions/{subscriptionId}	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0
4	POST	/namf-loc/v1/{ueContextId}/provide-loc-info	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29.518)
5	POST	/nausf-auth/v1/ue-authentications	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.10.0 (29509-ga0)
6	PUT	/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.10.0 (29509-ga0)
7	DELETE	/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation	3GPP Release Version: 16.10.10 (29.509)
8	GET	/nnrf-disc/v1/nf-instances	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.13.0 (29510-gd0)
9	POST	/nnrf-nfm/v1/subscriptions	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.13.0 (29510-gd0)
10	DELETE	/nnrf-nfm/v1/subscriptions/{subscriptionID}	3GPP Release Version: 16.13.0 (29.510)

Table 12 (Cont.) Default SEPP Service APIs

Sr. No	HTTP Method	Resource URI	Reference (GSMA FS.36 Version 2.2) or 3GPP Release Version
11	PATCH	/nnrf-nfm/v1/subscriptions/{subscriptionID}	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29.518)
12	GET	/nnsf-nsselection/v1/network-slice-information	3GPP Release Version: 15.2.0 (29531-f20)
13	POST	/npcf-ue-policy-control/v1/policies	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.12.0 (29525-gc0)
14	DELETE	/npcf-ue-policy-control/v1/policies/{polAssold}	3GPP Release Version: 16.14.0 (29.518) Specification #: 29.525 Release 16.12.0
15	GET	/npcf-ue-policy-control/v1/policies/{polAssold}	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29.518)
16	POST	/npcf-ue-policy-control/v1/policies/{polAssold}/update	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.12.0 (29525-gc0)
17	POST	/nsmf-pdusession/v1/pdu-sessions	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29502-ge0)
18	POST	/nsmf-pdusession/v1/pdu-sessions/{pduSessionRef}/modify	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29502-ge0)
19	POST	/nsmf-pdusession/v1/pdu-sessions/{pduSessionRef}/release	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29502-ge0)
20	POST	/nsmf-pdusession/v1/pdu-sessions/{pduSessionRef}/retrieve	3GPP Release Version: 16.14.0 (29502-ge0)
21	POST	/nsmf-pdusession/v1/pdu-sessions/{pduSessionRef}/transfer-mo-data	3GPP Release Version: 16.14.0 (29502-ge0)
22	GET	/nudm-sdm/v1/{supi}	3GPP Release Version: 15.2.1 (29503-f21) Specification #: 29.503 Release 15.2.1

Table 12 (Cont.) Default SEPP Service APIs

Sr. No	HTTP Method	Resource URI	Reference (GSMA FS.36 Version 2.2) or 3GPP Release Version
23	GET	/nudm-sdm/v1/{supi}/am-data	3GPP Release Version: 15.2.1 (29503-f21)
24	PUT	/nudm-sdm/v1/{supi}/am-data/sor-ack	3GPP Release Version: 15.2.1 (29503-f21)
25	GET	/nudm-sdm/v1/{supi}/nssai	3GPP Release Version: 15.2.1 (29503-f21)
26	POST	/nudm-sdm/v1/{supi}/sdm-subscriptions	3GPP Release Version: 15.2.1 (29503-f21)
27	DELETE	/nudm-sdm/v1/{supi}/sdm-subscriptions/{subscriptionId}	3GPP Release Version: 16.14.0 (29.518)
28	GET	/nudm-sdm/v1/{supi}/sm-data	3GPP Release Version: 15.2.1 (29503-f21)
29	GET	/nudm-sdm/v1/{supi}/ue-context-in-smf-data	3GPP Release Version: 15.2.1 (29503-f21)
30	GET	/nudm-sdm/v1/shared-data	3GPP Release Version: 15.2.1 (29503-f21)
31	GET	/nudm-sdm/v2/{supi}	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29503-ge0)
32	GET	/nudm-sdm/v2/{supi}/am-data	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29503-ge0)
33	PUT	/nudm-sdm/v2/{supi}/am-data/sor-ack	3GPP Release Version: 16.14.0 (29503-ge0) Specification #: 29.503 Release 16.14.0
34	PUT	/nudm-sdm/v2/{supi}/am-data/upu-ack	3GPP Release Version: 16.14.0 (29503-ge0) Specification #: 29.503 Release 16.14.0
35	GET	/nudm-sdm/v2/{supi}/nssai	FS.36 v2.2 3GPP Release Version: 16.14.0 (29503-ge0)
36	POST	/nudm-sdm/v2/{ueId}/sdm-subscriptions	3GPP Release Version: 16.14.0 (29503-ge0)
37	DELETE	/nudm-sdm/v2/{ueId}/sdm-subscriptions/{subscriptionId}	3GPP Release Version: 16.14.0 (29.518)

Table 12 (Cont.) Default SEPP Service APIs

Sr. No	HTTP Method	Resource URI	Reference (GSMA FS.36 Version 2.2) or 3GPP Release Version
38	PATCH	/nudm-sdm/v2/{ueld}/sdm-subscriptions/{subscriptionId}	3GPP Release Version: 16.14.0 (29503-ge0)
39	GET	/nudm-sdm/v2/{supi}/sm-data	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29503-ge0)
40	GET	/nudm-sdm/v2/{supi}/ue-context-in-smf-data	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29503-ge0)
41	GET	/nudm-sdm/v2/shared-data	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29503-ge0)
42	GET	/nudm-uecm/v1/{ueld}/registrations/amf-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
43	PATCH	/nudm-uecm/v1/{ueld}/registrations/amf-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
44	PUT	/nudm-uecm/v1/{ueld}/registrations/amf-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
45	GET	/nudm-uecm/v1/{ueld}/registrations/amf-non-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
46	PATCH	/nudm-uecm/v1/{ueld}/registrations/amf-non-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
47	PUT	/nudm-uecm/v1/{ueld}/registrations/amf-non-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
48	DELETE	/nudm-uecm/v1/{ueld}/registrations/smf-registrations/{pduSessionId}	3GPP Release Version: 16.14.0 (29503-ge0)
49	PUT	/nudm-uecm/v1/{ueld}/registrations/smf-registrations/{pduSessionId}	3GPP Release Version: 16.14.0 (29503-ge0)
50	DELETE	/nudm-uecm/v1/{ueld}/registrations/smsf-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
51	GET	/nudm-uecm/v1/{ueld}/registrations/smsf-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)

Table 12 (Cont.) Default SEPP Service APIs

Sr. No	HTTP Method	Resource URI	Reference (GSMA FS.36 Version 2.2) or 3GPP Release Version
52	PUT	/nudm-uecm/v1/{ueld}/registrations/smsf-3gpp-access	3GPP Release Version: 16.14.0 (29.518) Specification #: 29.503 Release 16.14.0
53	DELETE	/nudm-uecm/v1/{ueld}/registrations/smsf-non-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
54	GET	/nudm-uecm/v1/{ueld}/registrations/smsf-non-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
55	PUT	/nudm-uecm/v1/{ueld}/registrations/smsf-non-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
56	POST	/oauth2/token	3GPP Release Version: 16.13.0 (29510-gd0)
57	GET	/nudm-sdm/v2/{supi}/smf-select-data	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29503-ge0)
58	GET	/nudm-sdm/v2/{supi}/sms-data	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29503-ge0)
59	GET	/nudm-sdm/v2/{supi}/ue-context-in-smsf-data	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29503-ge0)
60	GET	/nudm-sdm/v2/{supi}/sms-mng-data	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29503-ge0)
61	GET	/nnssf-nssselection/v2/network-slice-information	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.8.0 (29531-g80)
62	DELETE	/namf-comm/v1/subscriptions/{subscriptionId}	3GPP Release Version: 16.14.0 (29518-ge0)
63	POST	/namf-comm/v1/subscriptions	3GPP Release Version: 16.14.0 (29518-ge0)
64	PUT	/namf-comm/v1/subscriptions/{subscriptionId}	3GPP Release Version: 16.14.0 (29518-ge0)
65	GET	/namf-mt/v1/ue-contexts/{ueContextId}	3GPP Release Version: 16.14.0 (29518-ge0)

The following list of APIs is obsolete, as they are not compliant with 3GPP Release 16:

- "/nnrf-nfm/v1/NfStatusChangeNotification" and method = "POST";
- "/nudm-sdm/v2/{supi}/am-data" and method = "POST";
- "/nausf-auth/v1/ue-authentications/{authCtxId}/5g-aka-confirmation" and method = "POST";
- "/nsmf-pdusession/v1/pdu-sessions/{pduSessionRef}" and method = "POST";
- "/nsmf-pdusession/v1/pdu-sessions/{pduSessionRef}/transfer-mt-data" and method = "POST";
- "/nudm-sdm/v1/{supi}/am-data/upu-ack" and method = "PUT";

Note

The user is required to delete them from the default SERVICE API list or any custom list if upgrading from previous versions to SEPP 23.2.0 or later versions.