Oracle® Communications

Cloud Native Core, Security Edge Protection Proxy User Guide

Release 23.4.3

F89215-09

October 2024

Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide, Release 23.4.3

F89215-09

Contents

1 Introduction

2 Security Edge Protection Proxy(SEPP) Architecture

3 SEPP Features

5 SEPP Metrics, KPIs, and Alerts

6 SEPP Response Codes

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 provides information about the acronyms and the terminologies used in the document.

Table Acronyms and Terminologies

Acronym	Description
CRD	Custom Resource Definition
CNE	Cloud Native Environment
CSEPP	Consumer Security Edge Protection Proxy
DNS	Domain Name System
DD	Data Director
EGW	Egress Gateway
FQDN	Fully Qualified Domain Name
Hosted SEPP	Hosted SEPP functionality provides selective routing in Roaming Hub Mode
IGW	Ingress Gateway
IPX	Internetwork Packet Exchange
K8s	Kubernetes
Local PLMN	PLMN managed by Local SEPP
Local SEPP	SEPP in Local PLMN
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.
NRF	Network Repository Function
OHC	Oracle Help Center
OSDC	Oracle Software Delivery Cloud
PDB	PodDisruptionBudget
PLMN	Public Land Mobile Network
PSEPP	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 alternate routing across Remote SEPPs
REST API	Representational State Transfer Application Programming Interface

Table (Cont.) Acronyms and Terminologies

Acronym	Description
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 talk with each other through roaming hub.
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.
SEPP	Security Edge Protection Proxy
SOR	Steering Of Roaming
SVC	Service
TLS	Transport Layer Security
TH	Topology Hiding
TUH	Topology Unhiding

What's New In This Guide

This section introduces the documentation updates for Release 23.4.x.

Release 23.4.3- F89215-09, October 2024

There are no updates made to this document in this release.

Release 23.4.2- F89215-08, July 2024

Updated descriptions of the following CNC Console parameters in the Remote SEPP section:

- `32F FQDN`
- `N32F Port`

Release 23.4.1- F89215-07, June 2024

Added the table number for the SEPPPn32fPreviousLocationCheckValidationFailureAlertPercentAbove50 alert.

Release 23.4.0- F89215-06, June 2024

Added the table number for the SEPPPn32fPreviousLocationCheckValidationFailureAlertPercentAbove50 alert.

Release 23.4.1- F89215-05, April 2024

There are no updates made to this document in this release.

Release 23.4.0- F89215-04, March 2024

Added the metric type to all the metrics in the SEPP Metrics section.

Release 23.4.0- F89215-03, February 2024

- Added the Separate Port Configurations for n32c and n32f on the Egress Routes feature in the SEPP Features chapter.
- Added the "Separate Port Configurations for n32c and n32f on the Egress Routes" CNC Console parameters in Remote SEPP section under the Configuring SEPP using CNC Console chapter.
- Added the"EgressInterfaceConnectionFailure" alert to support in the SEPP Alerts section to support the "Separate port configurations for n32c and n32f on the Egress routes" feature.

Release 23.4.0- F89215-02, January 2024

- Removed the Separate Port Configurations for n32c and n32f on the Egress Routes feature from the SEPP Features chapter.
- Removed the "Separate Port Configurations for n32c and n32f on the Egress Routes" CNC Console parameters from Remote SEPP section in the Configuring SEPP using CNC Console chapter.
- Removed the"EgressInterfaceConnectionFailure" alert to support in the SEPP Alerts section to support the "Separate port configurations for n32c and n32f on the Egress routes" feature.

Release 23.4.0- F89215-01, December 2023

- Added the following new features in SEPP Features chapter:
  - Separate Port Configurations for n32c and n32f on the Egress Routes feature
  - Load Sharing among Multiple Remote SEPP Nodes feature
  - Support for cnDBTier APIs in CNC Console feature.
- Updated the 5G SBI Message Mediation Support section and added the Mediation Rules using CNC Console and Rest APIs section.
- Updated the header and body configuration tables in the Topology Hiding feature.
- Added a note about the `alternateRoute.sbiReRoute.sbiRoutingErrorActionSets[].Attempts` parameter in the Alternate Routing Across Remote SEPPs feature.
- Added the following section in the Configuring SEPP using CNC Console section.
  - cnDBTier
- Updated the following sections in the Configuring SEPP using CNC Console section.
  - Updated the Remote SEPP section to support Separate Port Configurations for n32c and n32f on the Egress Routes and Load Sharing among Multiple Remote SEPP Nodes features.
  - Updated the Mediation to support Mediation Rules Configuration using CNC Console.
- Restructured the SEPP Application Alerts section to categorize the alerts based on the microservices and features.
- Added the following alert to support in the SEPP Alerts section to support the separate port configurations for n32c and n32f on the Egress routes feature:
  - EgressInterfaceConnectionFailure
- Updated the expressions of the following alerts in the SEPP Alerts section:
  - SEPPPn32fSORFailureAlertPercent30to40
  - SEPPPn32fSORFailureAlertPercent40to50
  - SEPPPn32fSORFailureAlertPercentAbove50
  - SEPPPn32fSORTimeoutFailureAlert
  - SEPPN32fMessageValidationOnHeaderFailureMinorAlert
  - SEPPN32fMessageValidationOnHeaderFailureMajorAlert
  - SEPPN32fMessageValidationOnHeaderFailureCriticalAlert
  - SEPPN32fMessageValidationOnBodyFailureMinorAlert
  - SEPPN32fMessageValidationOnBodyFailureMajorAlert
  - SEPPN32fMessageValidationOnBodyFailureCriticalAlert
  - SEPPN32fTopologyOperationFailureAlert
  - SEPPN32fTopologyBodyOperationFailureAlert

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 CNE and hardware environment.

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, Security Edge Protection Proxy REST Specification 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 Troubleshooting Guide*
- *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Network Impact Report*
- *Oracle Communications Cloud Native Configuration Console User Guide*

2

Security Edge Protection Proxy(SEPP) Architecture

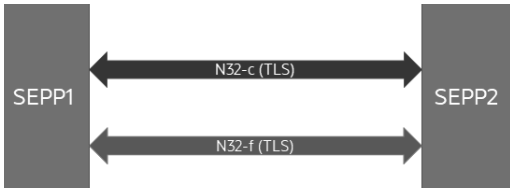
This section explains the Security Edge Protection Proxy(SEPP) Architecture.

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

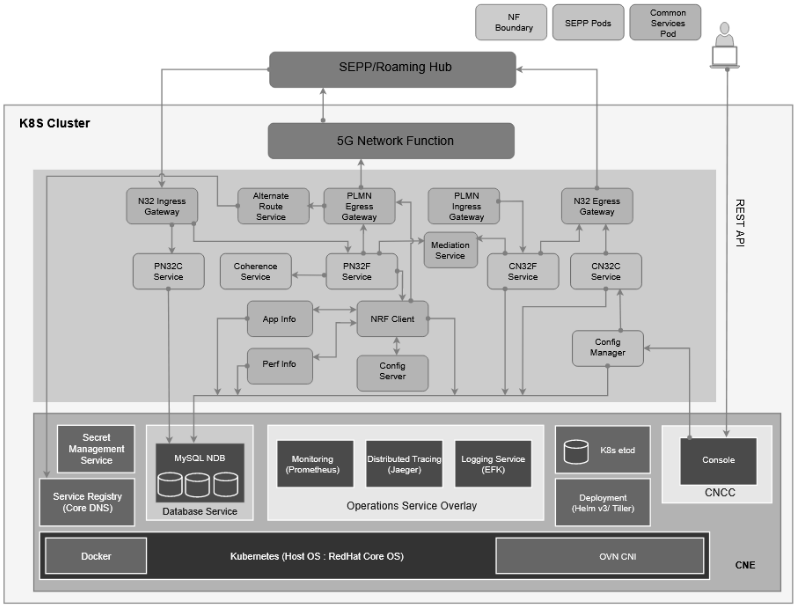


SEPP provides two interfaces, N32c and N32f:

- **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) architecture and microservices overview.

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.

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-PLAN 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.

ocpm-config (Performance Monitor Service)

It is responsible for monitoring and analyzing the services to probe performance data and provide analysis output, including load and capacity. NRF-Client periodically gets the performance data from this service using the exposed REST APIs.

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 exposed REST APIs.

appinfo (Application Info Service)
It is responsible for checking the status of the NF's registered services (Deregistration or Running or Not_Running) and periodically monitoring the status. NRF-Client gets the service status of the services periodically from the AppInfo using its REST API.

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.

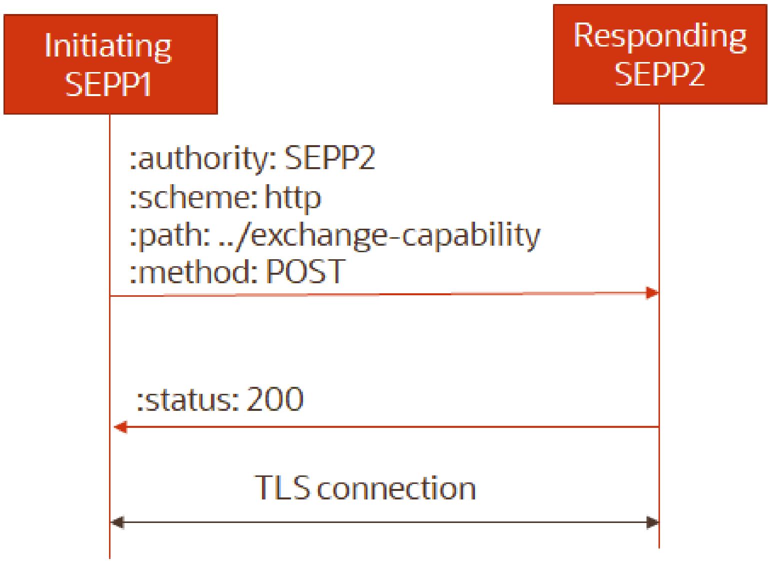
2.1 N32c and N32f Call flows

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

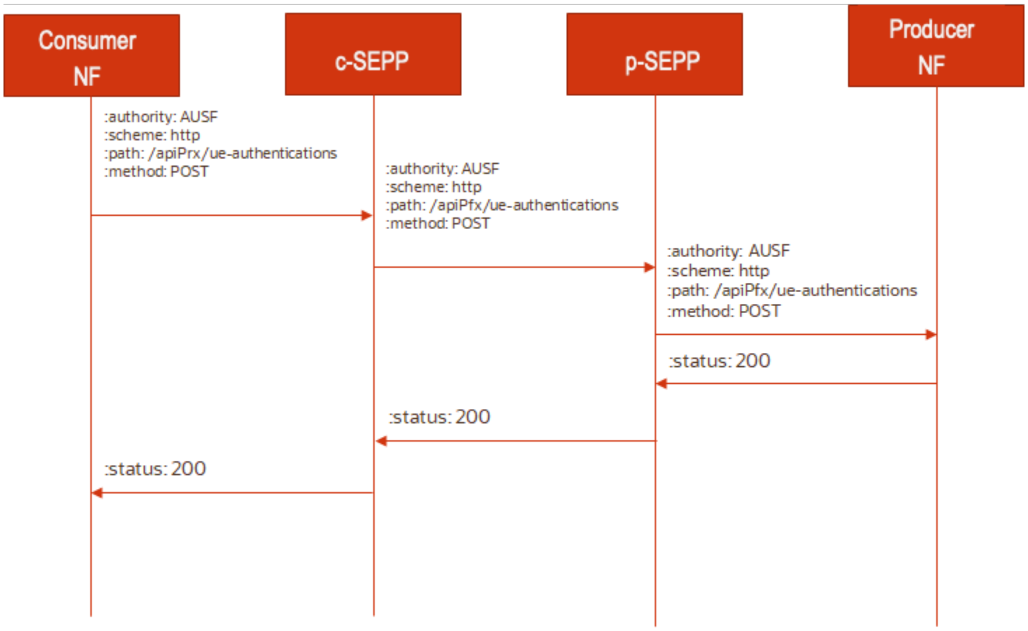


- The initiating SEPP issues a HTTP POST request towards the responding SEPP with the request body containing the "SecNegotiateReqData" with supporting security capabilities that is, PRotocol for N32 INterconnect Security (PRINS) and/or Transport Layer Security (TLS).
- On successful processing of the 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

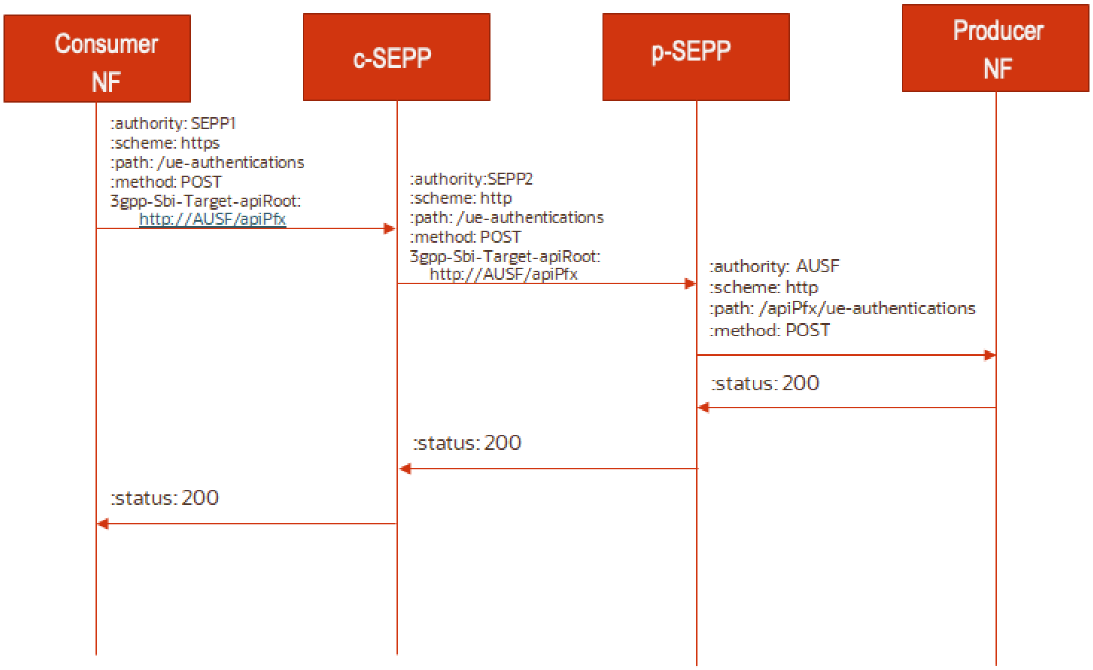


- The SEPP on the NF service consumer (c-SEPP) and the SEPP on the NF service producer (p-SEPP) negotiate the security capabilities using the procedure defined in Handshake call flow. The SEPPs mutually negotiate to use Transport Layer Security (TLS) as the security policy.
- The NF service consumer 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 3pp-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 Features

This section explains the SEPP 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.

3.1 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

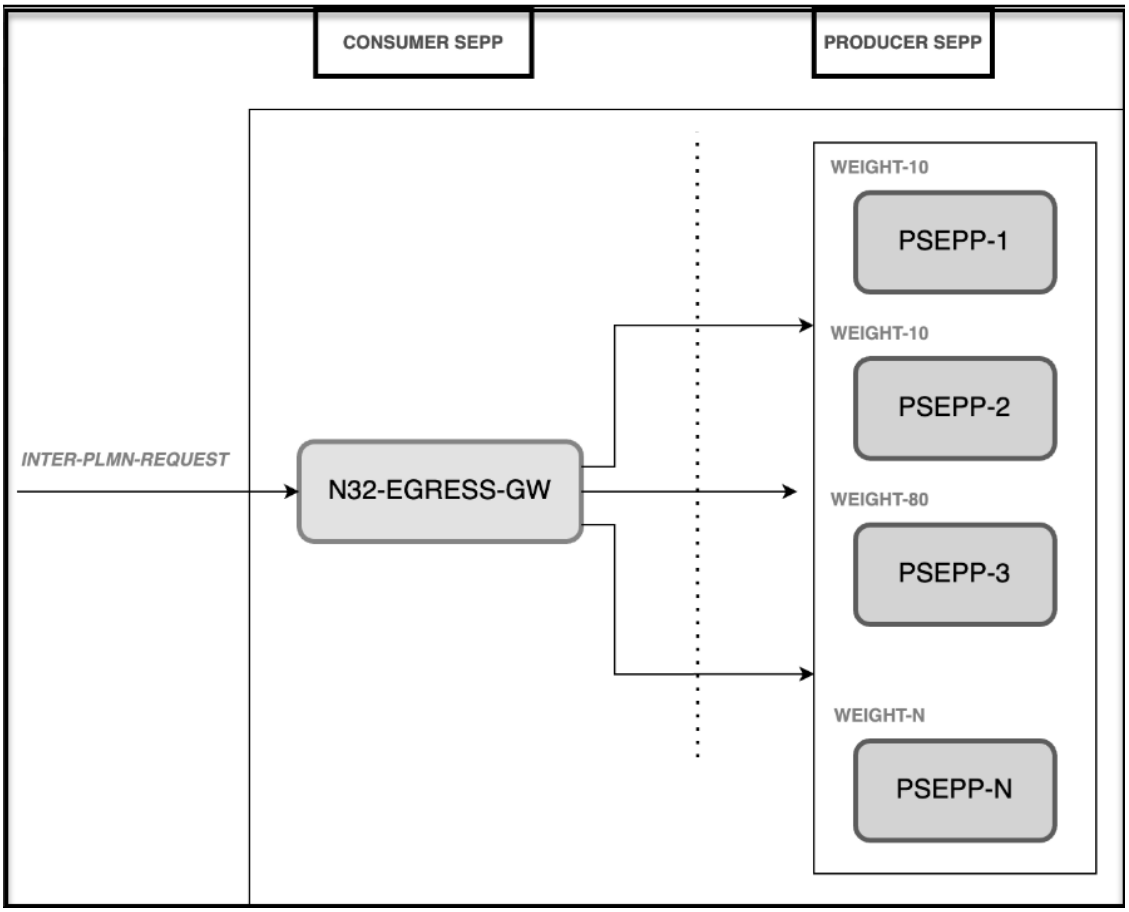
- SEPP FQDN and Virtual FQDN should be mapped to a valid IP address or multiple IP addresses. SEPP FQDN is used for the N32c handshake procedure.
- The load sharing feature is only used for the N32F traffic at N32-egress-gateway.
- Remote SEPP Set cannot have multiple peers with different Virtual FQDNs.

- 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 3-1 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.

Prerequisites
The following are the prerequisites 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
```

> **Note**
>
> Additionally, for Roaming Hub deployment, enable the `global.appinfoServiceEnable: true`.

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

Enabling
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 *Configuring SEPP using CNC Console* section in the *Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide*.

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 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

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 Separate Port Configurations for N32c and N32f on the Egress Routes

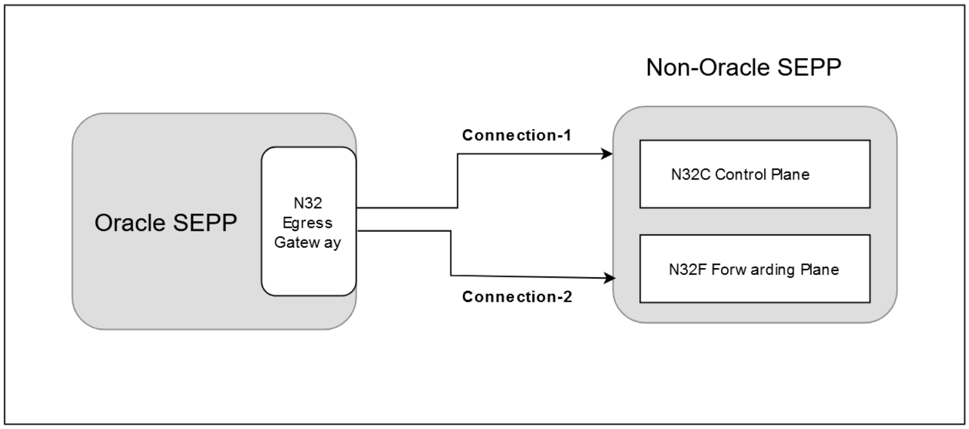
In some deployments, there is a need to have different service IPs and ports for n32c and n32f interfaces although the endpoint SEPP may have the same name. Separate ports for n32c and n32f interfaces provide the flexibility to configure different IP address or port for the n32c and n32f interfaces.
SEPP was previously configured to support a single port or connection for both the control plane interface (n23c) 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 3-2   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.

3.3 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:

- Backup List: This API displays the details of stored backups, such as the ID and size of the backup.
- cnDBTier version: This API displays the cnDBTier version.
- Database Statistics Report: This API displays the number of available database.
- Geo Replication Status:
  - Real Time Overall Replication Status: 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, site2-site3, site2-site4, and site2-site1. This is applicable for all other sites.
  - Site Specific Real Time Replication Status: This API displays the site-specific replication status.
- HeartBeat Status: This API displays the connectivity status between the local site and the remote site name to which SEPP is connected.
- Local Cluster Status: This API displays the status of the local cluster.
- On-Demand Backup: This API displays the status of initiated on-demand backups.

> **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.

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.

3.4 5G SBI Message Mediation Support

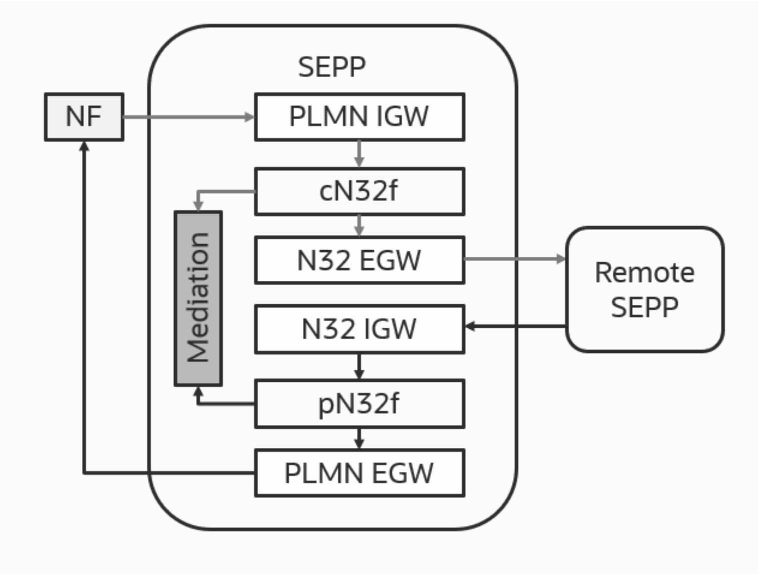
In the earlier releases, SEPP supports Mediation as an independent service to modify 5G Service Based Interface (SBI) message content, which includes HTTP2 header values and JSON message body, based on the user-defined mediation rule sets.

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.

Mediation Representation with SEPP

Figure 3-3 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").to
String()))
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 3-1 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 capacitiy.

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.

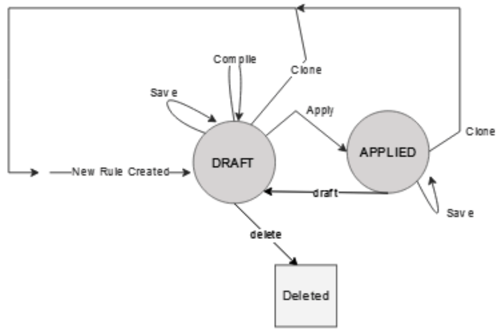
- Code: Mediation rule code. The user needs to perpend the following data block along with the needed rules. User will be restricted to apply a rule that is having any compilation or syntax error in it. Any special character in code results in a rule compilation error.

```
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 3-4 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.

> ⓘ **Note**
>
> 1. In SEPP release 23.4.0, the user is recommented 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 3-2 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"

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"
```

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.

3.5 Topology Hiding

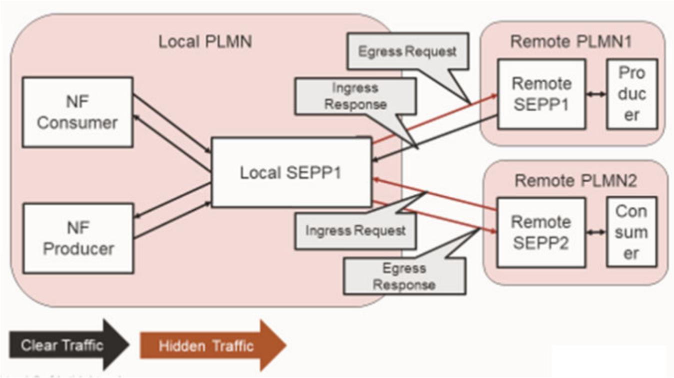
Oracle Communications Security Edge Protection Proxy (SEPP) is a proxy network function (NF) which is used for secured communication between inter-Public Land Mobile Network (PLMN) messages. SEPP provides message filtering and policing on inter-PLMN control plane interfaces and topology hiding.

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 3-5 Topology Hiding Architecture



If the Topology Hiding feature is enabled, based on the header and body table configuration, either the message will be masked with the hidden value (if the operation is TH -Topology Hiding) or the message will be represented by the actual value (if the operation is TUH - Topology Recovery), where the hidden value is mapped to the actual value.

Hidden traffic: The traffic is marked as hidden traffic, if the message (header and body) 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 and body configurations mentioned in the header or body 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 can be updated or modified for hiding or recovery in the message header or body identifier.

Header and Body Table Configuration

Header and body 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

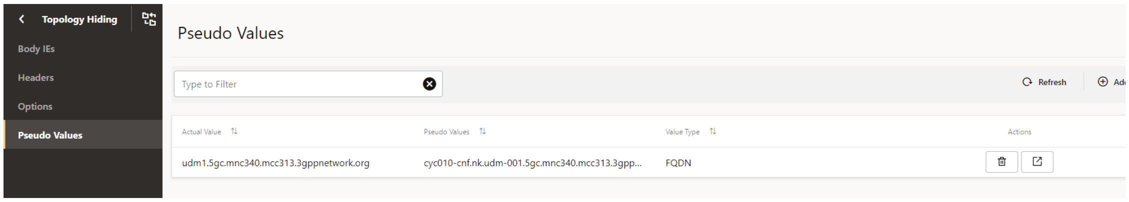
Ingress response has a clear traffic, as any messages for TH or TUH are not defined in the ingress response.

> **Note**
>
> In Topology Hiding feature, each request and response are considered as separate operations. Neither request header/identifiers nor response header/identifiers can be related to each other in any of the operations. Both request and response should be treated as separate isolated transactions.

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 3-6   Actual to Pseudo Mapping



Actual Value	Pseudo Values	Value Type	Actions
udm1.5gc.mnc340.mcc313.3gppnetwork.org	cyc010-cnf.nk.udm-001.5gc.mnc340.mcc313.3gpp...	FQDN	

The actual values and pseudo values are mapped and stored in the **actual_values** and **pseudo_values** tables.

The following is the default header configuration that is available to the user:

Table 3-3   Header configuration

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-)([^]*)	Response Egress	Topology Hiding
3gpp-sbi-target-apiroot	(?<=http://|https://)[^:/?]+	Request Ingress	Topology Recovery
3gpp-sbi-routing-binding	(?<=nfinst=|nfservinst=|backupamfinst=|backupnf=)([^;]+)	Request Ingress	Topology Recovery
3gpp-sbi-binding	(?<=nfinst=|nfservinst=|backupamfinst=|backupnf=)([^;]+)	Request Egress	Topology Hiding
3gpp-sbi-binding	(?<=nfinst=|nfservinst=|backupamfinst=|backupnf=)([^;]+)	Response Egress	Topology Hiding

Topology Hiding Body
TH: Topology Hiding

TUH: Topology Recovery

ⓘ 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.

The Helm parameter, **TOPOLOGY_HIDING_MAX_PSEUDO_VALUE** is by default set to 7. So user can add minimum one and maximum 7 pseudo values in the system. For any actual value, user can configure 7 pseudo values. If this actual value has to be hidden, then a random value is being picked up out of these 7 pseudo values and will be used for hiding.

TH and TUH operation is run based on the actual/pseudo mapping that user has configured in the system. By default, for any one actual value we can map at most 7 pseudo values. If the user perform TH and TUH, then value that is eligible for TH/TUH will be looked up in the mapping and accordingly TH/TUH operation is being performed.

In the actual to pseudo mapping ,for every actual value user can map multiple pseudo value. By default, user map minimum 1 and maximum 7 pseudo values. Actual values type that are currently supported are:

- FQDN
- NF_INSTANCE_ID
- NF_SERVICE_INSTANCE_ID.

  The user cannot add same pseudo value for the different actual values and duplicate pseudo values are not allowed.

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

With the Resource URI request, there can be a possibility where request URL contains the key value pair. These values can be one of FQDN, NF_INSTANCE_ID, or NF_SERVICE_INSTANCE_ID types and should be eligible for TH/TUH.

User can add the path header in the header table and specify the relevant configuration such as headerName, DIRECTION, Regular Expression, MessageType, operations. Then the keys coming in the path will be treated as headers.

ⓘ 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 3-4 Example of Header Table

Header Name	Regular Expression	Trigger Point	Operation
requester-nf-instance-fqdn	((?<=http://|https://)[^:/?]+)	Request Egress	Topology Hiding

Body Configuration

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 3-5 Example Body Configuration

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

Table 3-5 (Cont.) Example Body Configuration

Method	API Response	Identifier	Regular Expression	Trigger Point	Operation
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
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 apiUrl 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 apiUrl 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.

Path Parameters

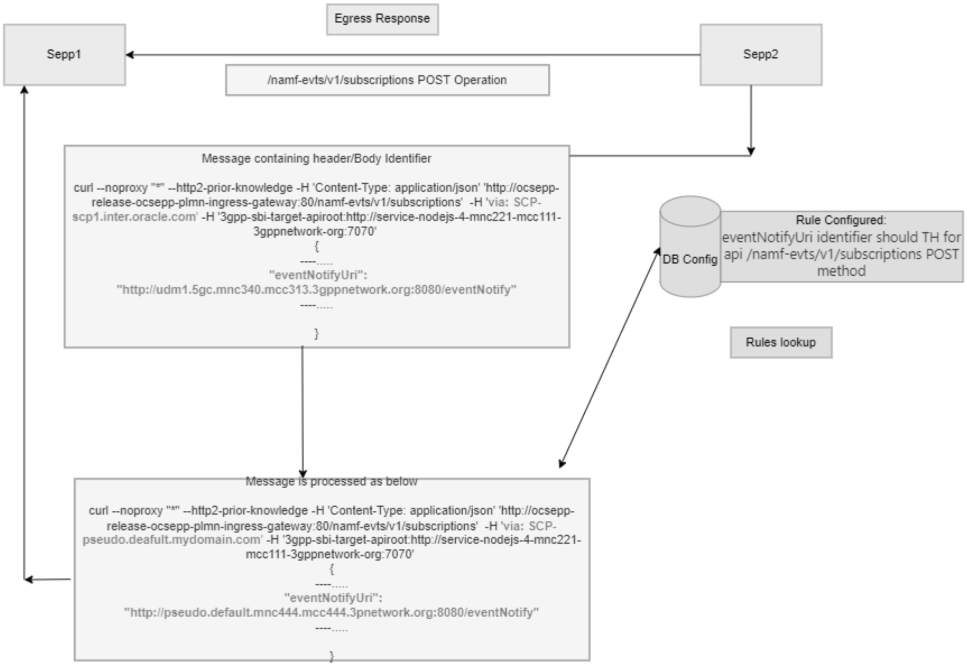
Resource URIs or messages , where path is having some variable identifier that keeps on changing for the different request. Such APIs are mapped in the default table with syntax of the path variable and same syntax to be used in the topology body configuration if identifiers for that API needs to be part of TH/TUH.

Example:

Consider the message request body going out from the system from CN32F (Ingress response) and having the apiUrl as */nnrf-disc/v1/nf-instances* 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.*

Example for Hiding the message


Egress Response
Sepp1
Sepp2
/namf-evts/v1/subscriptions POST Operation
Message containing header/Body Identifier
DB Config
Rule Configured:
eventNotifyUri identifier should TH for api /namf-evts/v1/subscriptions POST method
Rules lookup
Message is processed as below

```
curl --noproxy "*" --http2-prior-knowledge -H 'Content-Type: application/json' 'http://ocsepp-release-ocsepp-plmn-ingress-gateway:80/namf-evts/v1/subscriptions'  -H 'via: SCP-scp1.inter.oracle.com' -H '3gpp-sbi-target-apiroot:http://service-nodejs-4-mnc221-mcc111-3gppnetwork-org:7070'
{
----.....
"eventNotifyUri": "http://udm1.5gc.mnc340.mcc313.3gppnetwork.org:8080/eventNotify"
----.....

}
```

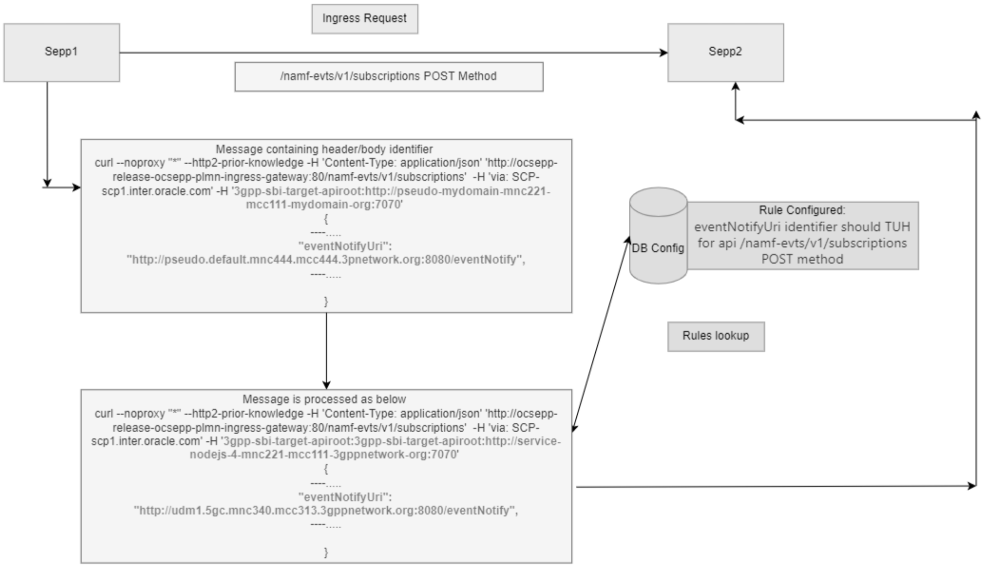
```
curl --noproxy "*" --http2-prior-knowledge -H 'Content-Type: application/json' 'http://ocsepp-release-ocsepp-plmn-ingress-gateway:80/namf-evts/v1/subscriptions'  -H 'via: SCP-pseudo.deafult.mydomain.com' -H '3gpp-sbi-target-apiroot:http://service-nodejs-4-mnc221-mcc111-3gppnetwork-org:7070'
{
----.....
"eventNotifyUri": "http://pseudo.default.mnc444.mcc444.3pnetwork.org:8080/eventNotify"
----.....

}
```

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/subscription*s 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 */nnrf-disc/v1/nf-instances* 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*


Ingress Request
Sepp1
Sepp2
/namf-evts/v1/subscriptions POST Method
DB Config
Rule Configured:
eventNotifyUri identifier should TUH for api /namf-evts/v1/subscriptions POST method
Rules lookup

Message containing header/body identifier

```
curl --noproxy "*" --http2-prior-knowledge -H 'Content-Type: application/json' 'http://ocsepp-release-ocsepp-plmn-ingress-gateway:80/namf-evts/v1/subscriptions'  -H 'via: SCP-scp1.inter.oracle.com' -H '3gpp-sbi-target-apiroot:http://pseudo-mydomain-mnc221-mcc111-mydomain-org:7070'
{
----.....
"eventNotifyUri":
"http://pseudo.default.mnc444.mcc444.3pnetwork.org:8080/eventNotify",
----.....

}
```

Message is processed as below

```
curl --noproxy "*" --http2-prior-knowledge -H 'Content-Type: application/json' 'http://ocsepp-release-ocsepp-plmn-ingress-gateway:80/namf-evts/v1/subscriptions'  -H 'via: SCP-scp1.inter.oracle.com' -H '3gpp-sbi-target-apiroot:3gpp-sbi-target-apiroot:http://service-nodejs-4-mnc221-mcc111-3gppnetwork-org:7070'
{
----.....
"eventNotifyUri":
"http://udm1.5gc.mnc340.mcc313.3gppnetwork.org:8080/eventNotify",
----.....

}
```

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 U*RI /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 psuedo 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 3-6 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/?$

Table 3-6 (Cont.) Default Configuration

id	resourceURI	method	regex
7	/nnrf-nfm/v1/subscriptions	POST	^/nnrf-nfm/v1/subscriptions/?$
8	/nudm-uecm/v1/{ueId}/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/{ueId}/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/{ueId}/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-nsselection/v2/network-slice-information	GET	^/nnssf-nsselection/v2/network-slice-information/?$
24	/nudm-uecm/v1/{ueId}/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/{ueId}/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 3-6 (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/{polAssoId}	GET	^/npcf-ue-policy-control/v1/policies/([0-9a-zA-Z-]+|.+)/?$
33	/nudm-sdm/v2/{ueId}/sdm-subscriptions/{subscriptionId}	DELETE	^/nudm-sdm/v2/(imsi-[0-9]{5,15}|nai-.+|msisdn-[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/{ueId}/registrations/smsf-non-3gpp-access	GET	^/nudm-uecm/v1/(imsi-[0-9]{5,15}|nai-.+|msisdn-[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/{ueId}/registrations/amf-non-3gpp-access	GET	^/nudm-uecm/v1/(imsi-[0-9]{5,15}|nai-.+|msisdn-[0-9]{5,15}|extid-[^@]+@[^@]+|gci-.+|gli-.+|.+)/registrations/amf-non-3gpp-access/?$
38	/nudm-uecm/v1/{ueId}/registrations/amf-non-3gpp-access	PUT	^/nudm-uecm/v1/(imsi-[0-9]{5,15}|nai-.+|msisdn-[0-9]{5,15}|extid-[^@]+@[^@]+|gci-.+|gli-.+|.+)/registrations/amf-non-3gpp-access/?$
39	/nudm-uecm/v1/{ueId}/registrations/smsf-non-3gpp-access	PUT	^/nudm-uecm/v1/(imsi-[0-9]{5,15}|nai-.+|msisdn-[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/{polAssoId}	DELETE	^/npcf-ue-policy-control/v1/policies/([0-9a-zA-Z-]+|.+)/?$

Table 3-6 (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-smsf-data	GET	^/nudm-sdm/v2/(imsi-[0-9]{5,15}|nai-.+|gci-.+|gli-.+|.+)/ue-context-in-smsf-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/{polAssoId}/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.

3.6 Global Rate Limiting on Egress Gateway of SEPP

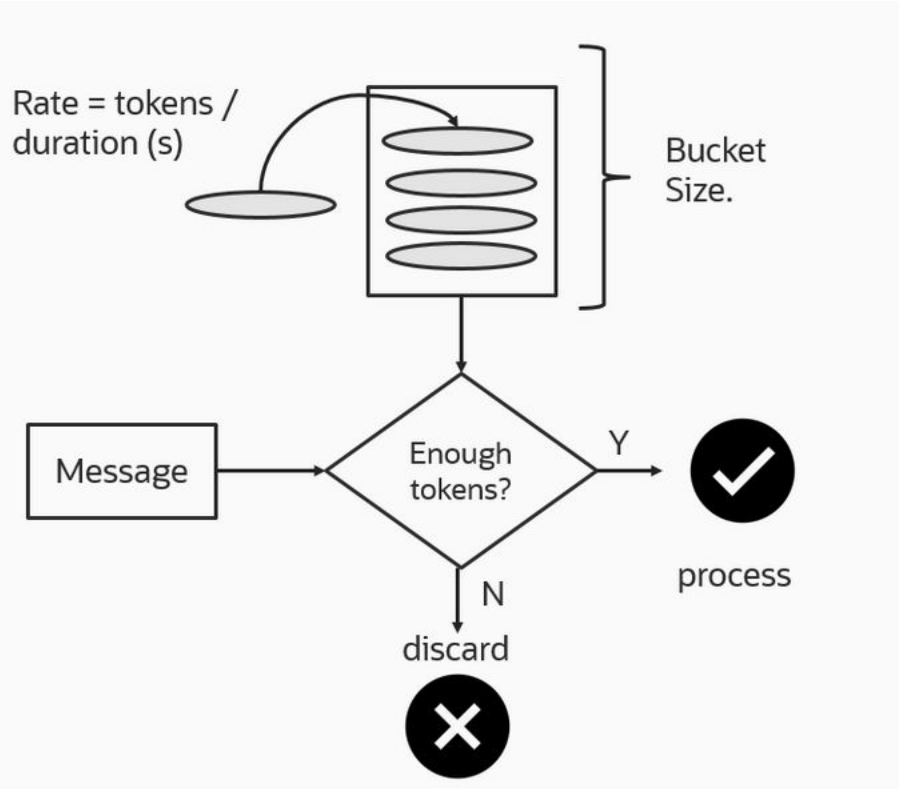
Feature Overview

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 3-7 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.

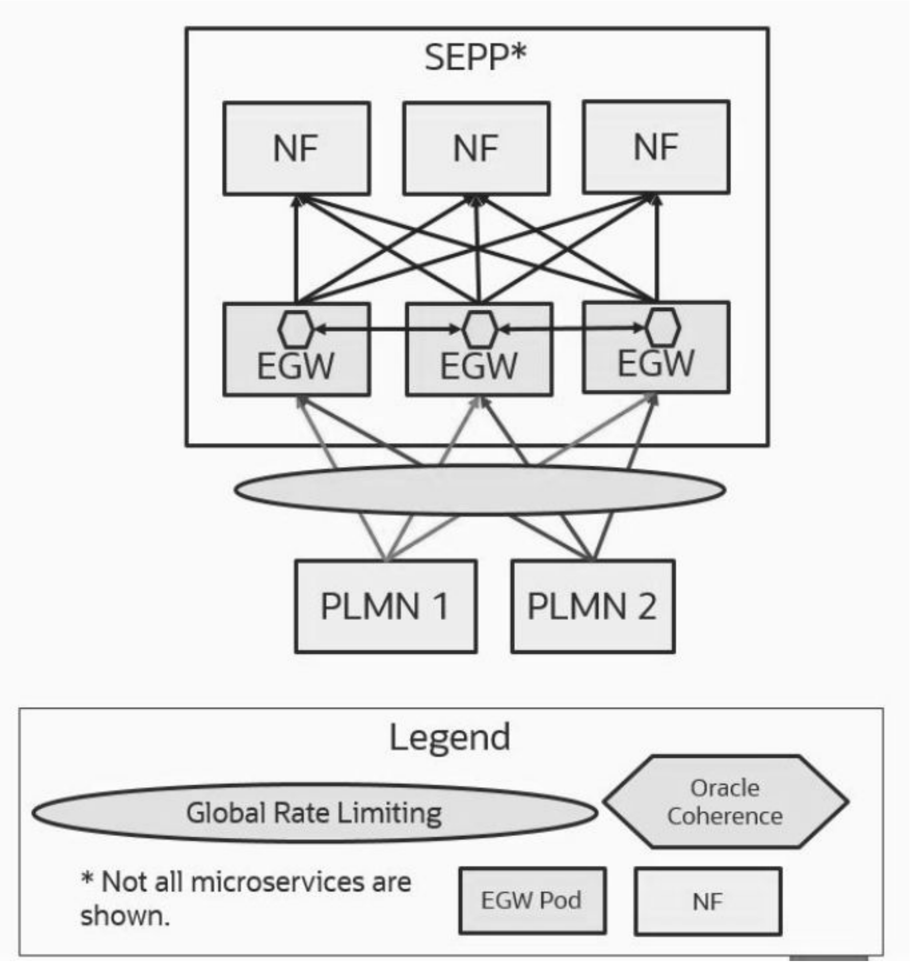
Three configurations are required for this algorithm:

- 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

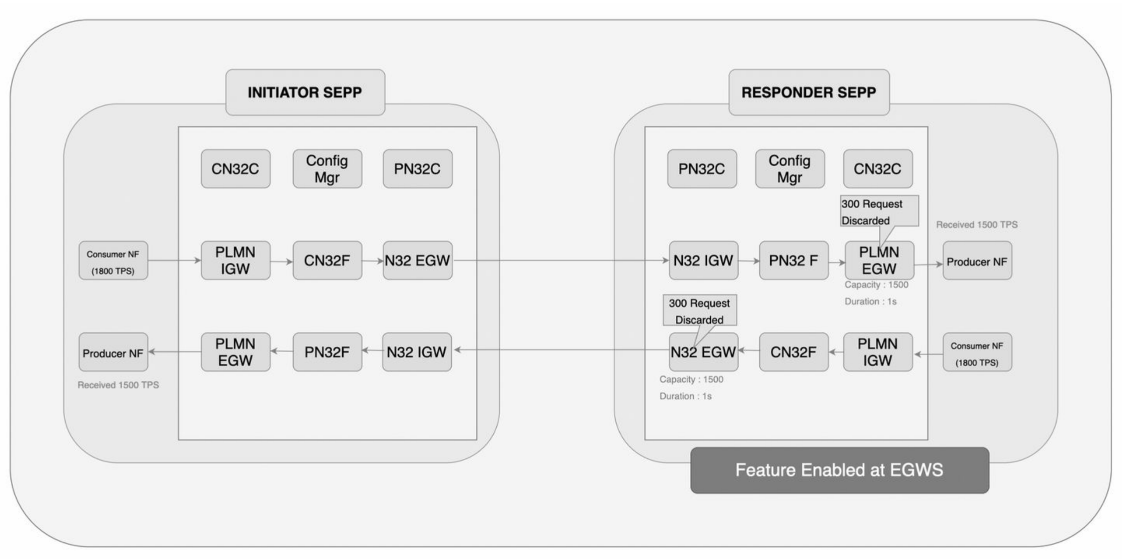
Using Egress global rate limiting, the user can configure capacity for SEPP. The incoming traffic management at the gateway level and any traffic beyond the capacity of the backend service get rejected with the error response as configured by the user. The rules for global rate limiting apply to all traffic, irrespective of headers

Figure 3-8 Architecture



The following diagram represents the functionality of the global rate limiting feature with the example:

Figure 3-9 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

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
```

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.

3.7 Global Rate Limiting on Ingress Gateway of SEPP

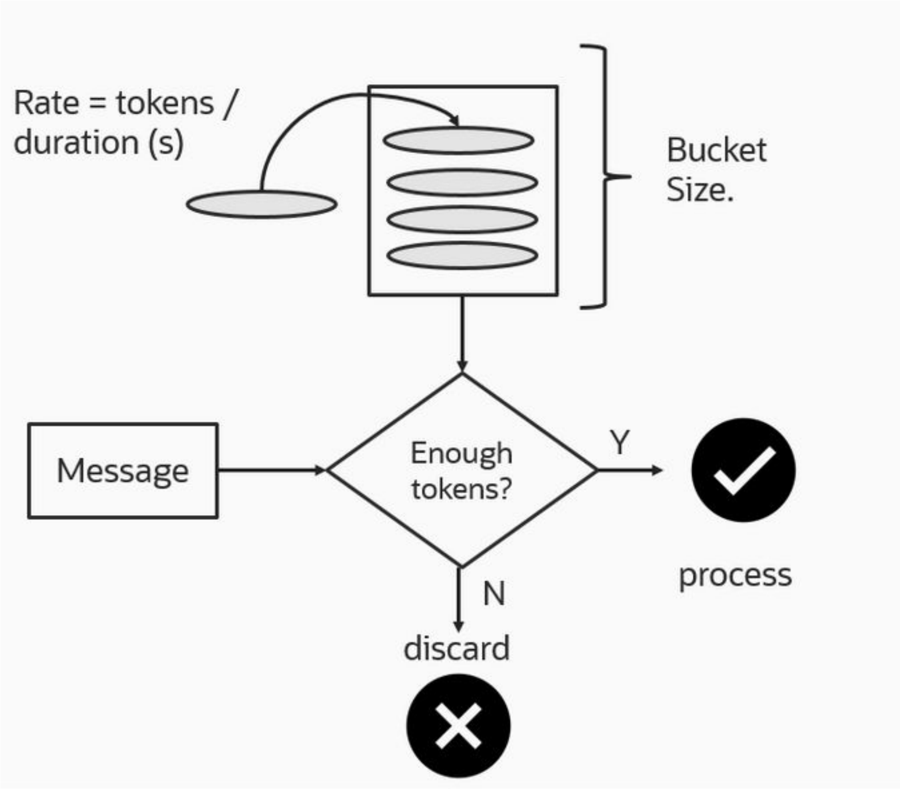
Feature Overview

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 3-10 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.

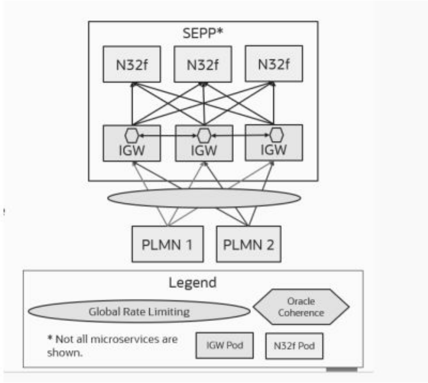
Three configurations are required for this algorithm:

- 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

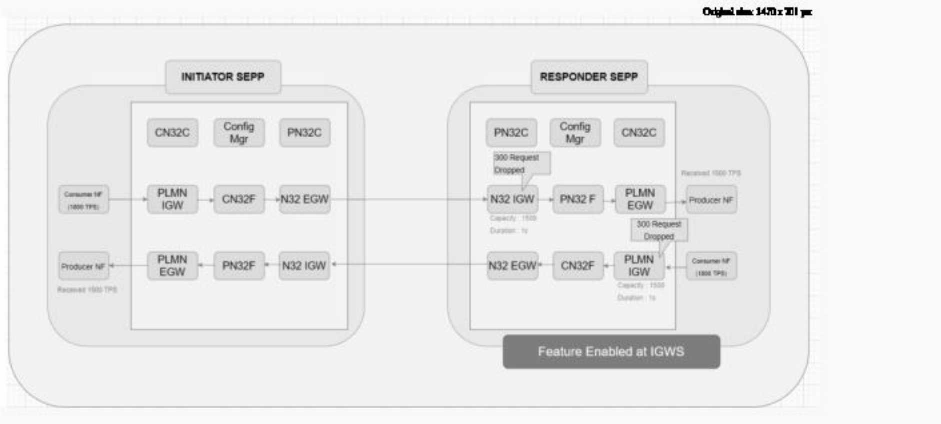
Using Global rate limiting, the user can configure capacity for backend service (n32f service). The incoming traffic management at the gateway level and any traffic beyond the capacity of the backend service get rejected with the error response as configured by the user. The rules for global rate limiting apply to all traffic, irrespective of headers

Figure 3-11 Architecture



The following diagram represents the functionality of the global rate limiting feature with the example:

Figure 3-12 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_custom_values_<version>.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 3-7 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 3-8 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 3-9 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.

3.8 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.

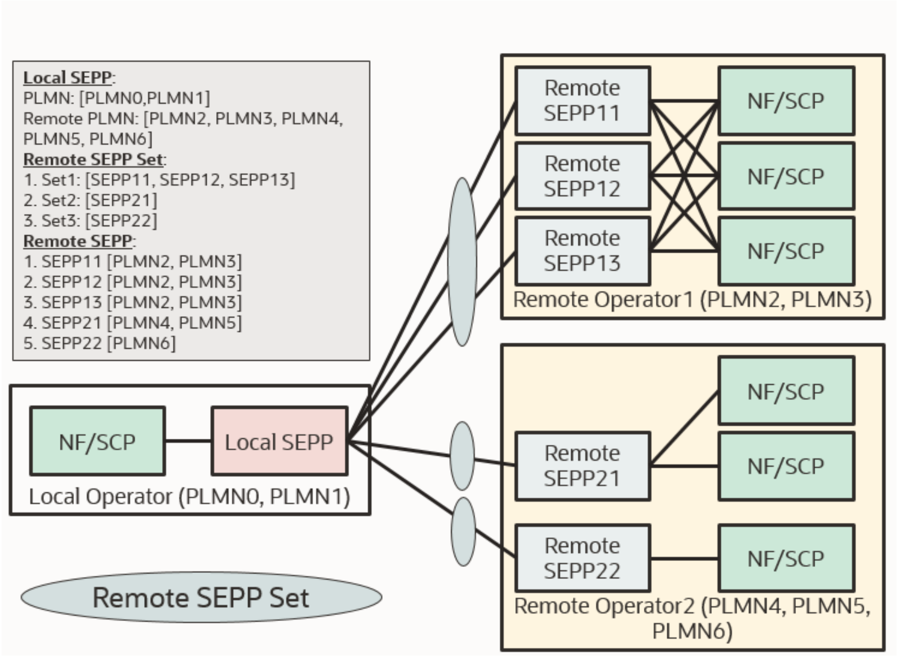
Generally, it is preferred to use a single SEPP to route to multiple roaming partners and to support a list of PLMN IDs. This feature allows the local SEPP to communicate with multiple remote partners simultaneously. With this feature, SEPP can route to a minimum of 1 and a maximum of 30 PLMN IDs at 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.



Local SEPP:
PLMN: [PLMN0,PLMN1]
Remote PLMN: [PLMN2, PLMN3, PLMN4, PLMN5, PLMN6]
Remote SEPP Set:
1. Set1: [SEPP11, SEPP12, SEPP13]
2. Set2: [SEPP21]
3. Set3: [SEPP22]
Remote SEPP:
1. SEPP11 [PLMN2, PLMN3]
2. SEPP12 [PLMN2, PLMN3]
3. SEPP13 [PLMN2, PLMN3]
4. SEPP21 [PLMN4, PLMN5]
5. SEPP22 [PLMN6]

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 3-10 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.

3.9 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.

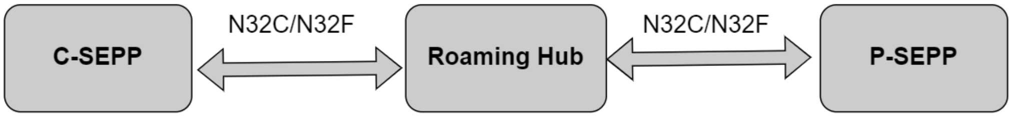
Roaming Hub providers deploy SEPP acting as Roaming Hub enabling inter PLMN communication between MNOs through N32 interface supporting Application Layer Security (ALS) as specified in TS 33.501. Roaming Hub provider is able to deploy SEPP in either of one mode i.e. functionality of 5GC SEPP NF or Roaming 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.

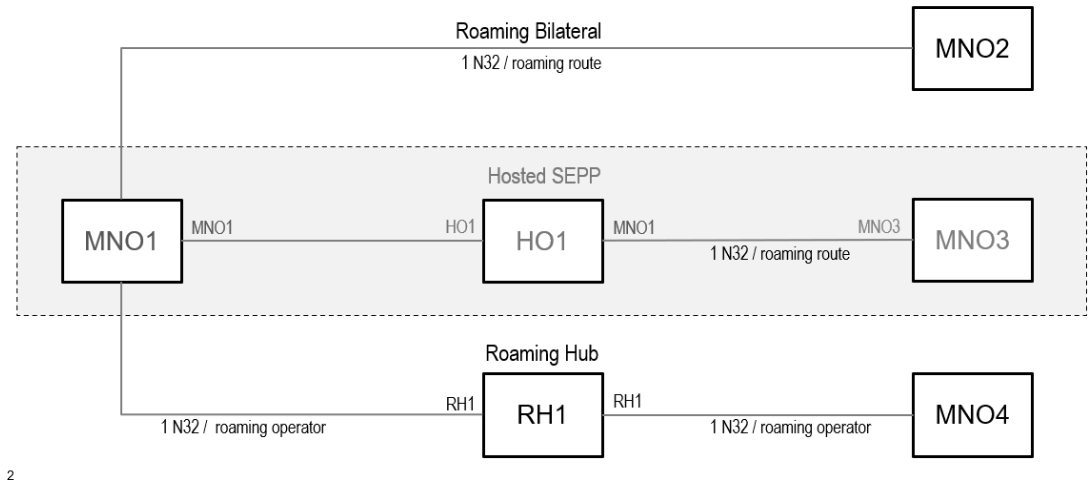
Figure 3-13 Roaming Hub Overview



SEPP supports inter PLMN traffic for both SEPP MNO and Roaming Hub modes.

Detailed Description

Figure 3-14 Detailed Description



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 3-11 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 :
		MaxRSS == (400 – total number of PLMNs used) / 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.

3.10 Cat-1 Service API Validation 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:

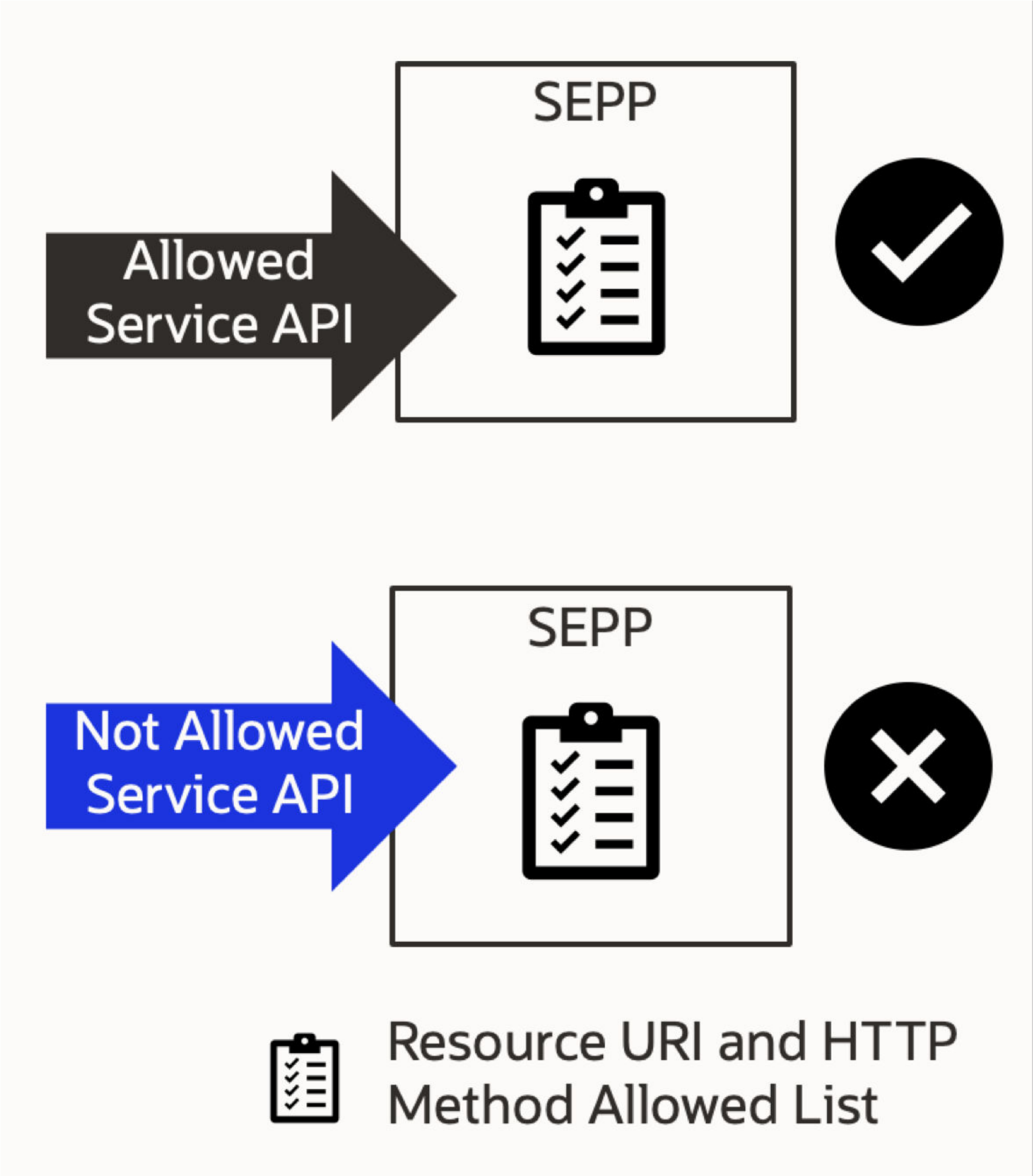
- Check of combination of HTTP Method and Service requested (Example: POST cannot be used in combination with a particular service).
- Allows only certain SBI messages to either egress or ingress the network (Example: AMF to UDM messages)

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 3-15 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 Validationfeature:

- 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.

3.11 Overload Control Feature

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 is exhausted. 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.

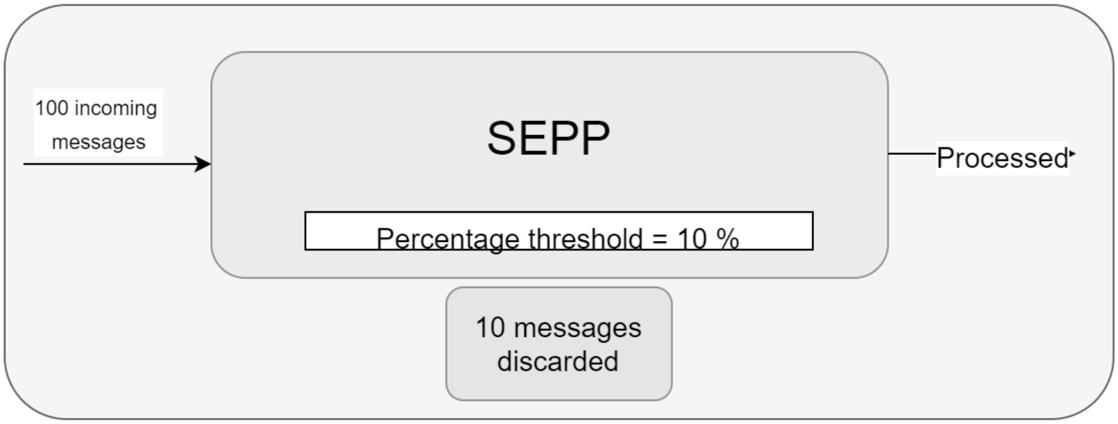
The following table represents the example for threshold set for CPU and memory and the discard values.

Table 3-12 Threshold Level

Threshold Level	CPU onset	CPU abatement	Memory onset	Memory abatement	Percentage Discard	Priority threshold
Warning	65	60	50	45	0%	30
Minor	75	70	70	65	5%	24
Major	85	80	80	75	10%	15
Critical	95	90	90	85	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.

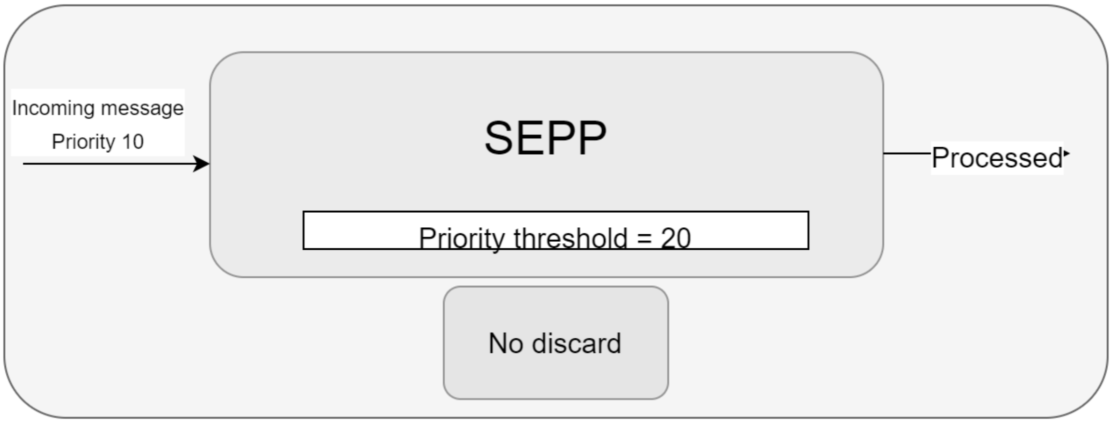


In the above diagram, percentage threshold level is set as 10%. Therefore for incoming traffic rate at 100 TPS, 10 messages will be discarded with the configured error code.

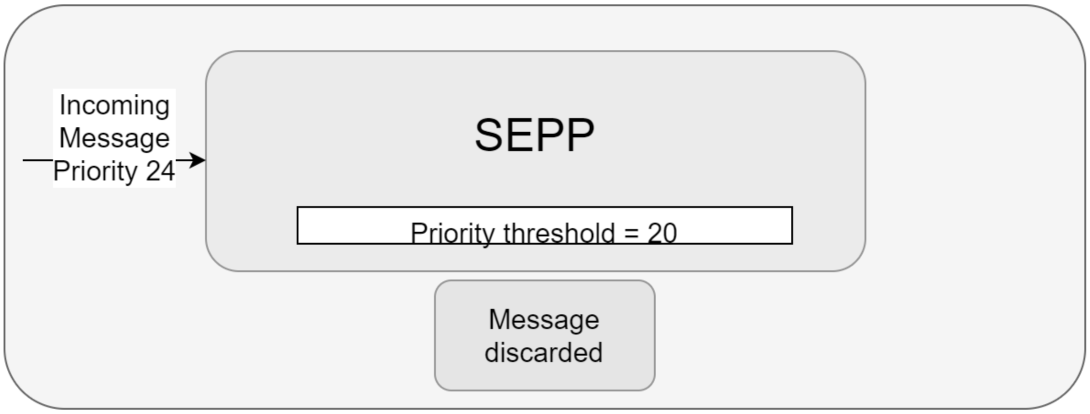
Priority based discard policy

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.



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.



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 REST API.

- **Enable using 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 REST API: Perform the feature configurations as described in *Oracle Communications Cloud Native Core Security Edge Protection Proxy REST Specification Guide.*

Steps to configure overload control

1. 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
}'
```

2. To define CPU and Memory 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
        }]
    }]
}]'
```

> **Note**
>
> The above mentioned values are the recommended values for each threshold level. User can configure them as required.

3. 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
   ```

4. 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": ""
       }
    ]
   ```

5. 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.

6. 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_total
- service_resource_overload_level

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.

3.12 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 SEPPor 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).

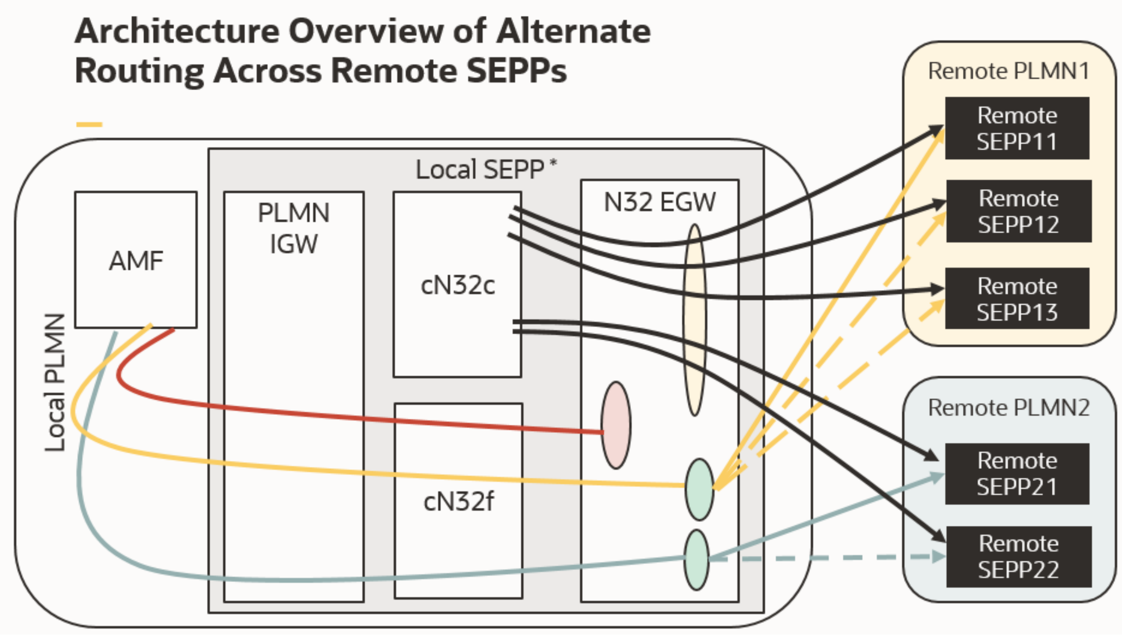
Feature Overview

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 3-16 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.

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*.

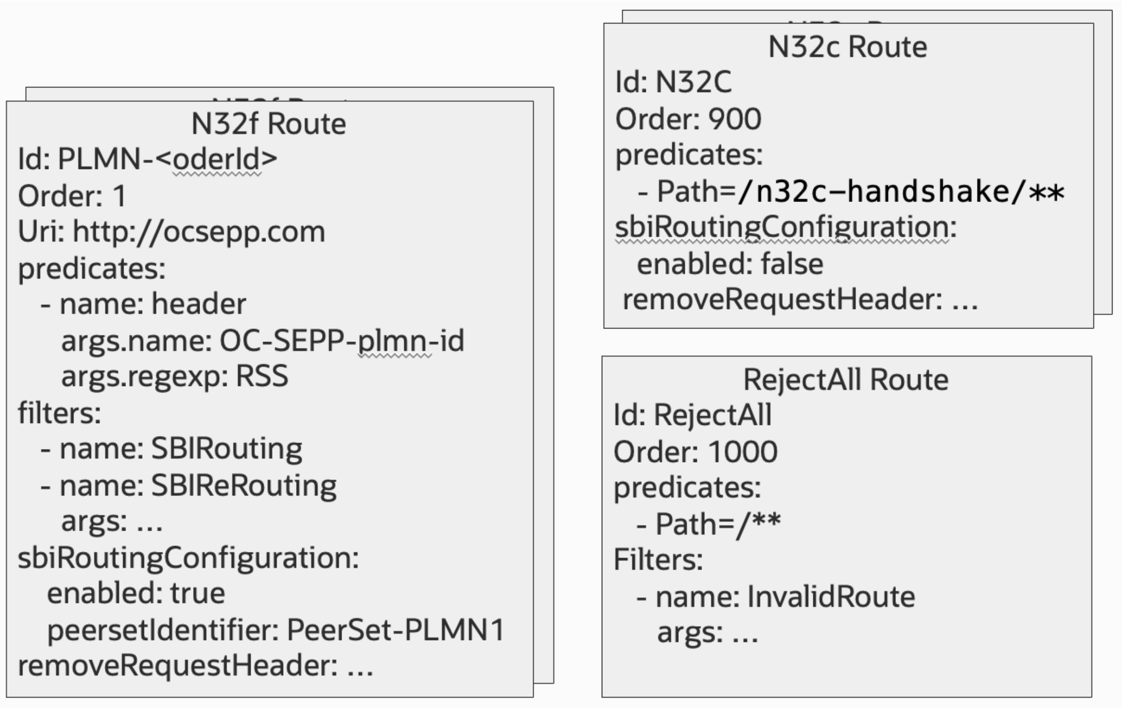
Default Routes

When SEPP is installed, the following three default routes are created on n32-egress-gateway:

- **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 3-17 Default Routes



```
N32f Route
Id: PLMN-<oderId>
Order: 1
Uri: http://ocsepp.com
predicates:
  - name: header
    args.name: OC-SEPP-plmn-id
    args.regexp: RSS
filters:
  - name: SBIRouting
  - name: SBIReRouting
    args: ...
sbiRoutingConfiguration:
   enabled: true
   peersetIdentifier: PeerSet-PLMN1
removeRequestHeader: ...
```

```
N32c Route
Id: N32C
Order: 900
predicates:
  - Path=/n32c-handshake/**
sbiRoutingConfiguration:
  enabled: false
 removeRequestHeader: ...
```

```
RejectAll Route
Id: RejectAll
Order: 1000
predicates:
  - Path=/**
Filters:
  - name: InvalidRoute
    args: ...
```

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", "regexp": "RS-1"},
"name": "Header"}]}], "forwardheaderdetails": {"enabled": false,
"forwardHeaderValue": ""}
```

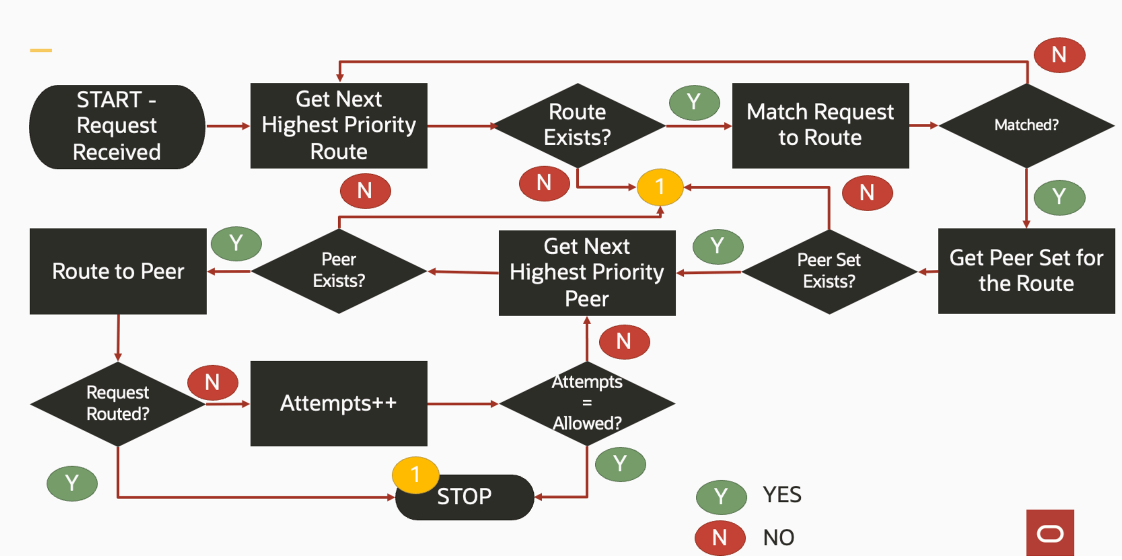
Alternate Routing Flow

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 3-18 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.

3.13 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 assinged 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
```

3.14 SEPP MNO functionality Support

SEPP supports the following interfaces:

- N32-c interface (TLS mode only)
- N32-f interface (TLS mode only)
- TLS 1.2

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)

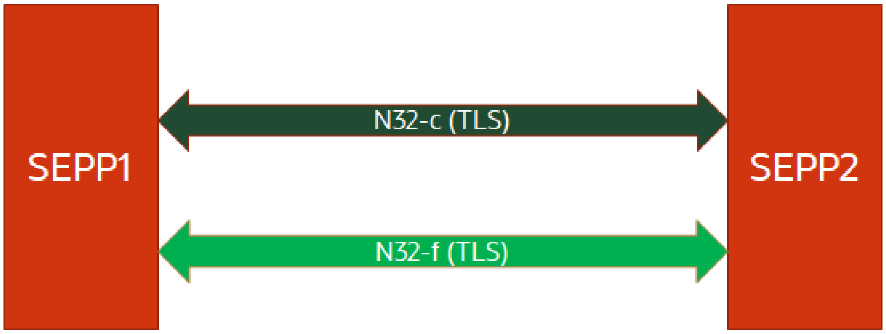
Additionally the SEPP can assist is being a single point of interconnect to multiple networks on the core and inter-PLMN side. To assist in such a deployment, the SEPP has been modified to support the following:

- Multiple Remote SEPPs on the egress side towards other PLMNs.
- Multiple core networks can be configured to route to the SEPP.

- The N32-c handshake procedures have also been modified to support multiple N32-c handshakes with other roaming partner SEPPs, that is the N32-f interfaces can be established across multiple partners from a single SEPP.

The following diagram depicts the topology for SEPP MNO functionality Support.

Figure 3-19 Topology for SEPP MNO functionality



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-1"
    plmn:
      mcc: "311"
      mnc: "282"
    domain: "oracle.com"
    interPlmnFqdn: "sepp1.inter.oracle.com"
    intraPlmnFqdn: "ocsepp-plmn-ingress-gateway.DEPLOYMENT_NAMESPACE"
    supportedSecurityCapabilityList:
     - "TLS"
    apiPrefix:  ""
    retryInterval: 300000
    maxRetry: -1
    nfInstanceId: "9faf1bbc-6e4a-4454-a507-aef01a101a06"
```

```
   #configuration in n32-egress-gateway, n32-ingress-gateway, plmn-egress-
gateway, plmn-ingress-gateway section in ocsepp-custom-values.yaml file

 ssl:
     tlsVersion: TLSv1.2
sepp:
forwardProxy: true
egressProxyHeader: X-Next-Hop
customAuthorityHeader: X-Next-Hop-Authority
removeUnusedProxyAfter: 30
# set true only if sepp.forwardProxy -> true,otherwise it wont have effect
tlsConnectionMode: true
```

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:

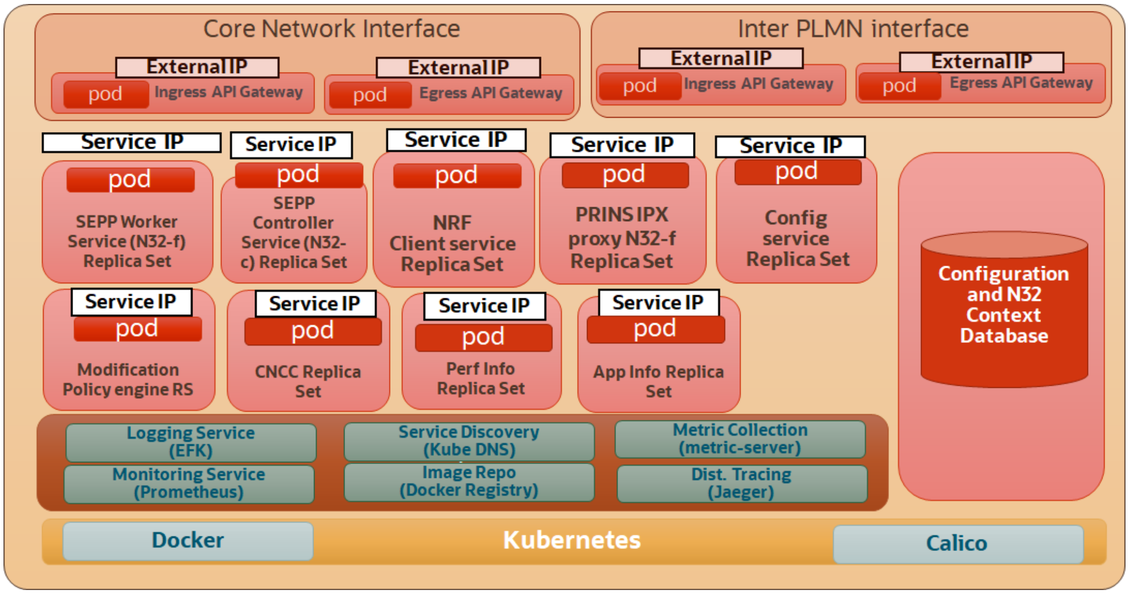
1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request**: See My Oracle Support for more information on how to raise a service request.

3.15 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 3-20 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.

3.16 Support for Core Networks with or without SCP

The SEPP can be deployed in networks where SCP 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

Enable

This feature can be enabled or disabled though HELM. The configuration must be set in PLMN Egress gateway section.

Configure

You can configure the Support for Core Networks with or without SCP using helm parameters:

Configuring using Helm parameters

```
    routesConfig:
  - id: scp_via_proxy
    uri: http://request.uri
    path: /**
    order: 1
    metadata:
      httpsTargetOnly: false
      httpRuriOnly: false
      sbiRoutingEnabled: true
```

Observe

For more information on Support for Core Networks with or without SCP 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.

3.17 Support for Networks with/without 3GPP-SBI-Target-API-Root Header

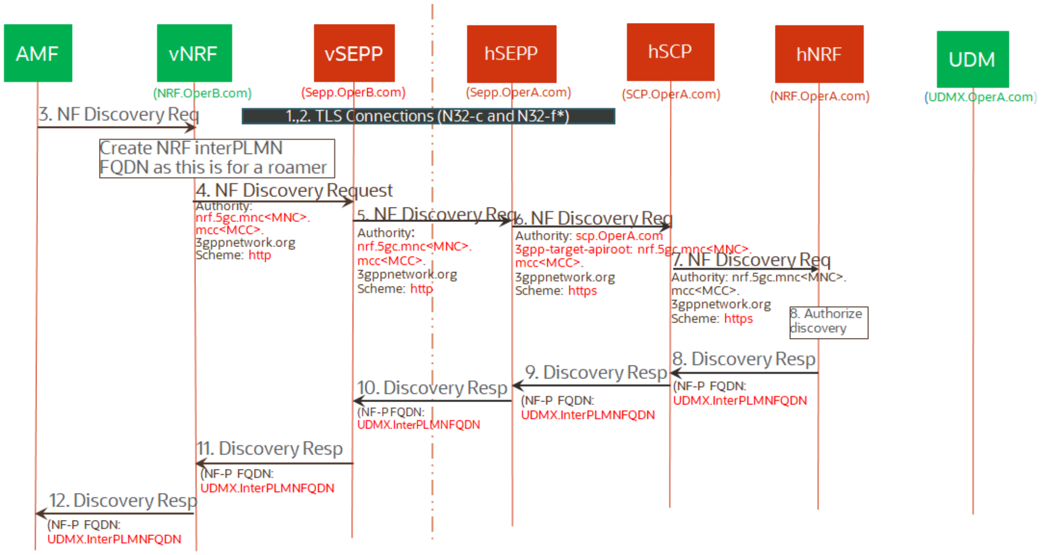
There are some networks which may not be supporting the Rel 16 header 3GPP-SBI-Target-Apiroot header. In such scenarios, SEPP being deployed at the edge of the network is required to make the routing possible in the core network. SEPP modifies the respective headers such that routing doesn't fail in the core network.

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 3-21   SEPP sets the 3GPP-SBI-Target-Apiroot header

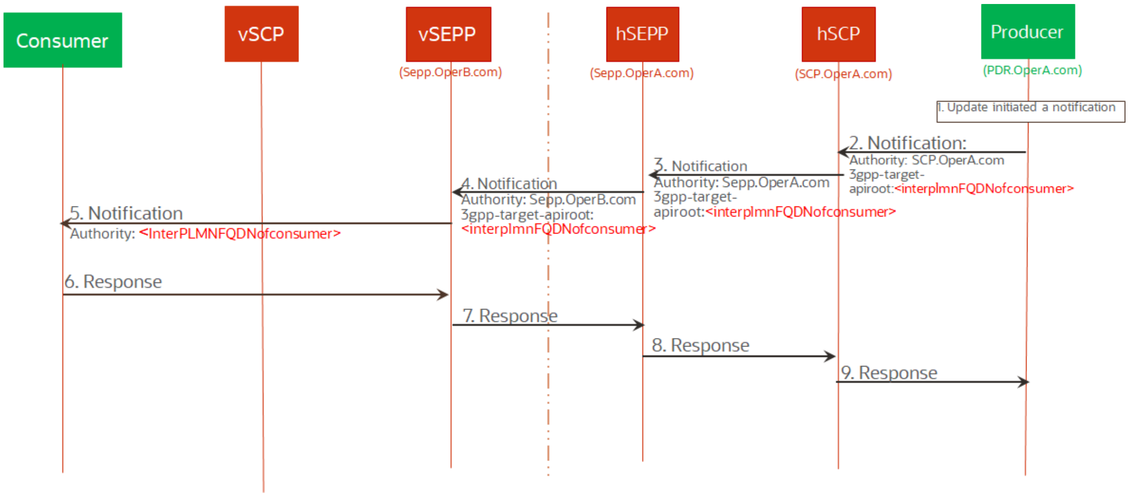


2. Network that is sending in the request uses the 3GPP-SBI-Target-Apiroot header, however the core network behind the SEPP will not be able to understand this header.

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 3-22   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.

3.18 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:

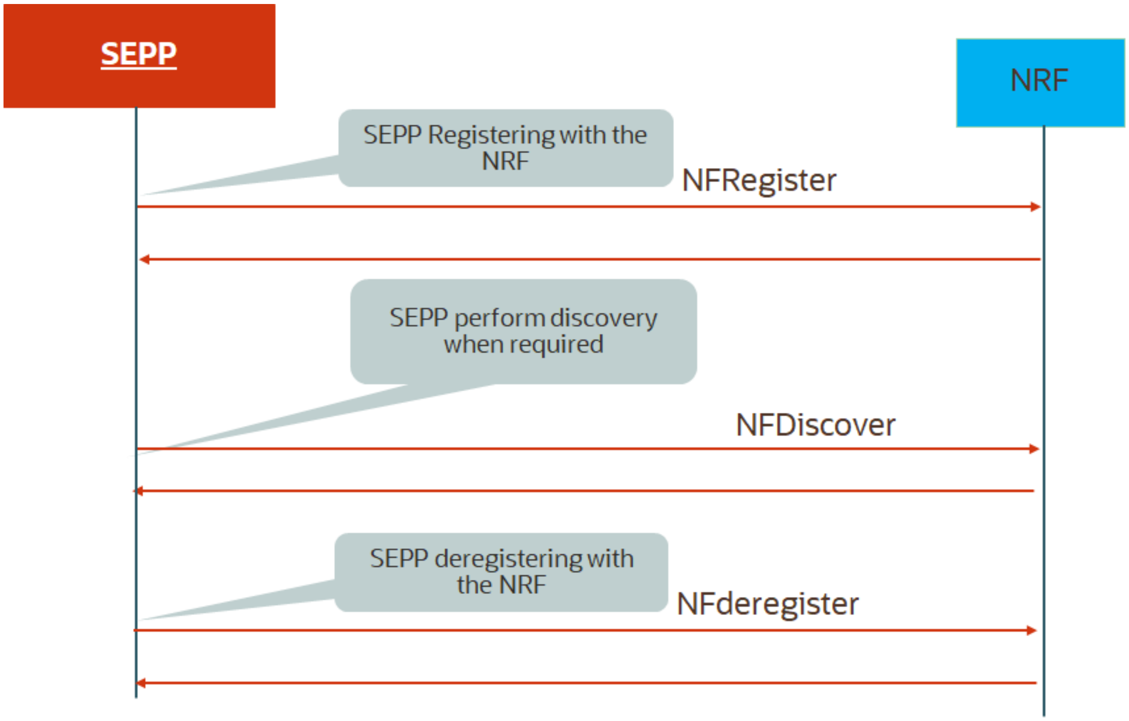
1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request**: See My Oracle Support for more information on how to raise a service request.

3.19 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 3-23 NRF Interface Registration & Reregistration.


SEPP
NRF
SEPP Registering with the NRF
NFRegister
SEPP perform discovery when required
NFDiscover
SEPP deregistering with the NRF
NFderegister

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.

3.20 Helm Validation Framework

SEPP supports helm validation framework enhancement.This enahancement 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 3-13 Types of Validation checks

Types of Validation checks	Example
Requirement checks	An API_KEY 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 http(s)://<host>:<port> custom_values.yaml

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 Schemavocabulary. 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.

3.21 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 sanValidationRequired in HELM	Value of sanValidationRequired 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

Value of sanValidationRequired in HELM	Value of sanValidationRequired in REST API/ CNC Console (Remote SEPP)	SAN Validation of incoming N32C handshake request
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.

3.22 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 OCNRF (if initial algorithm is ES256)
- RSA private key and CA signed certificate of OCNRF (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
```

3.23 SEPP Message Feed Feature

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.

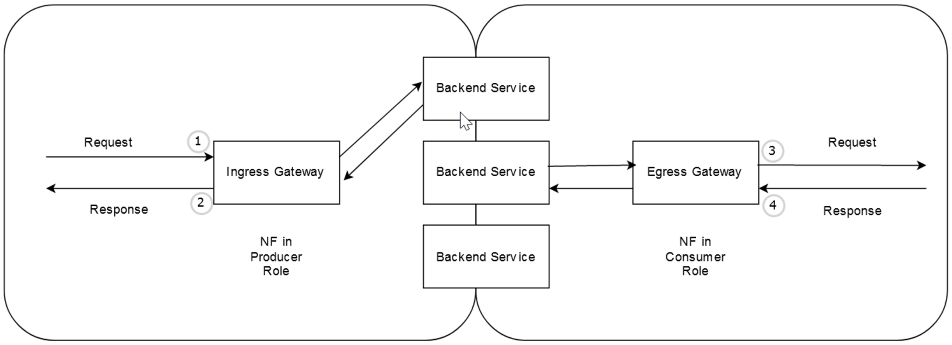
The correlation-id in the messages is used as a unique identifier for every transaction. If the request does not have the correlation-id, the Gateway generates the correlation-id and then passes it to Data Director.

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.

Overview

In order to capture incoming and outgoing traffic, there are four tapping points for Ingress Gateway and Egress Gateway.

Figure 3-24 Overview of Message Feed



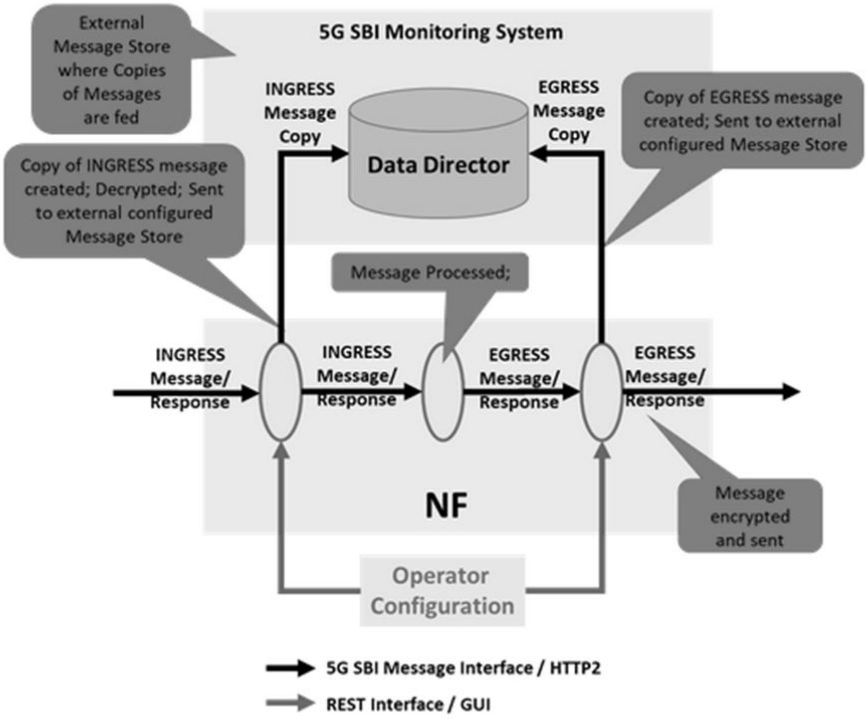
Ingress Gateway Tapping point

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 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:

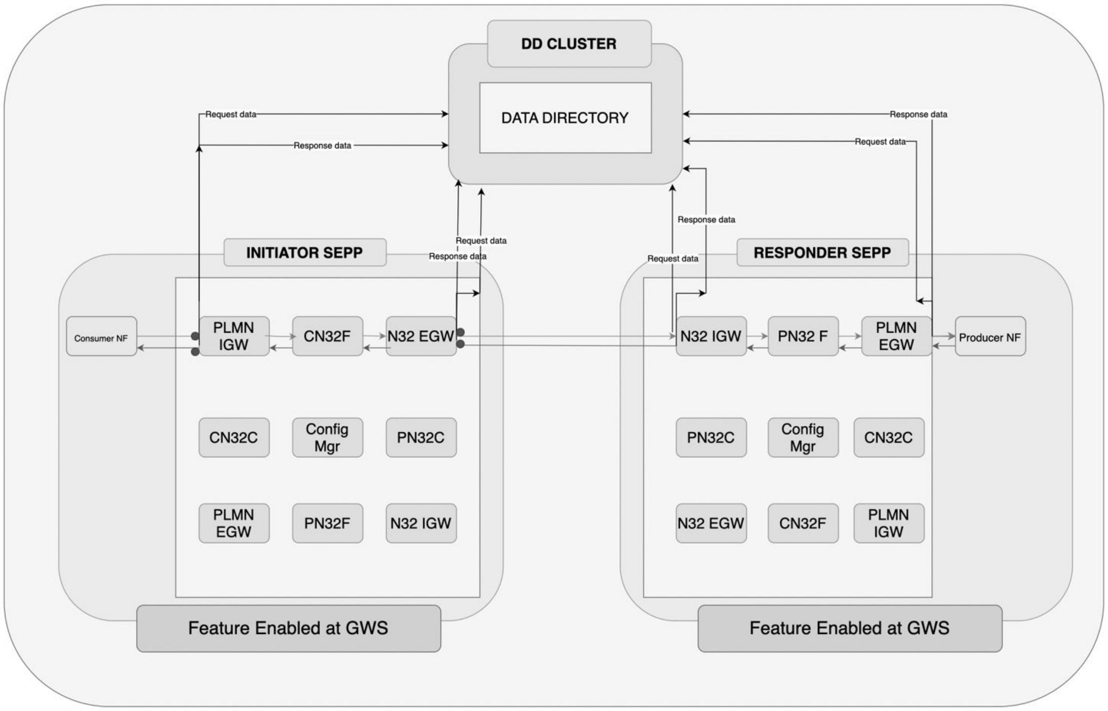
Figure 3-25   SEPP representation of Message Feed feature


External Message Store where Copies of Messages are fed
5G SBI Monitoring System
INGRESS Message Copy
Data Director
EGRESS Message Copy
Copy of EGRESS message created; Sent to external configured Message Store
Copy of INGRESS message created; Decrypted; Sent to external configured Message Store
Message Processed;
INGRESS Message/ Response
INGRESS Message/ Response
EGRESS Message/ Response
EGRESS Message/ Response
NF
Message encrypted and sent
Operator Configuration
5G SBI Message Interface / HTTP2
REST Interface / GUI

Detailed Description

The following diagram indicates the detailed view of message feed feature in SEPP:

Figure 3-26 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 are the descriptions of the attributes that are copied to the DD (Data Director) as part of metadata:

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 request message. Then it should copy the message to DD and then send it out	x-request-id header or autogenerate

Attribute name	Data type	Description	Source of data
		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 from3gpp-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).
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

Attribute name	Data type	Description	Source of data
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

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 `global.nfTypeMsgCpy`. 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 `global.nfTypeMsgCpy`.
nf-fqdn	String	This is an optional Parameter. Identifies a producer NF FQDN.	The information from Helm configuration from the parameter `global.nfFqdn.`

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. Skip to step 11 if security is disabled.
10. 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 temperory 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 **Kafka.bootstrapAddress** to DD service and port which will have port 9094 (security enabled).

11. Save the file.

12. Run *helm install.* For more information about installation procedure, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation and Upgrade 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.

ocsepp-custom-values-<version>.yaml

```
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
  threadPoolConfigurations:
    coreSize: 8
    maxSize: 8
    queueCapacity: 1000
     security:
    enabled: false
    protocol: SASL_SSL
    tlsVersion: TLSv1.2
    saslConfiguration:
      userName: test
      password:
        k8SecretName: message-copy-secret           #For plmn gateways it
will be: ocsepp-plmn-secret
        k8NameSpace: ocegress
        fileName: password.txt


#Kafka related configuration:
```

```
Kafka:
  bootstrapAddress: 10.75.175.246:30608
```

Sample Egress Gateway Configuration:

```
messageCopy:
 enabled: false
 copyPayload: false
 topicName: message.copy
 ackRequired: false
  retryOnFailure: 0
  threadPoolConfigurations:
    coreSize: 8
    maxSize: 8
    queueCapacity: 1000
  security:
    enabled: false
    protocol: SASL_SSL
    tlsVersion: TLSv1.2
    saslConfiguration:
      userName: test
      password:
        k8SecretName: message-copy-secret           #For plmn gateways it
will be: ocsepp-plmn-secret
        k8NameSpace: ocegress
        fileName: password.txt
#Kafka related configuration:
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.

3.24 Hosted SEPP Feature

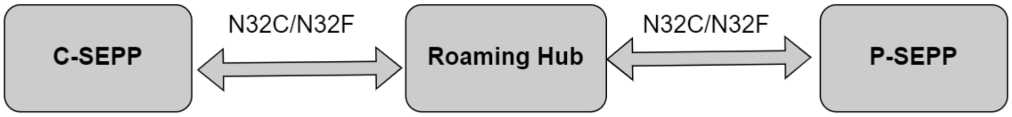
General Description

This section provides a general overview of the Hosted SEPP Feature.

Background

SEPP is deployed by both Mobile Network Operators (MNO) and Roaming Hub providers. MNOs deploy SEPP acting as 3GPP 5G SEPP NF which enables inter-PLMN communication between two networks through N32 interface. Oracle SEPP can be deployed in either of the two modes, that is, functionality of 5GC SEPP NF or Roaming Hub. Roaming Hub providers are able to host 5GC SEPP NF functionality for few of the MNOs and host Roaming Hub functionality for rest of the MNOs. All the Remote SEPPs that are connected to the Roaming Hub are allowed to communicate with each other.

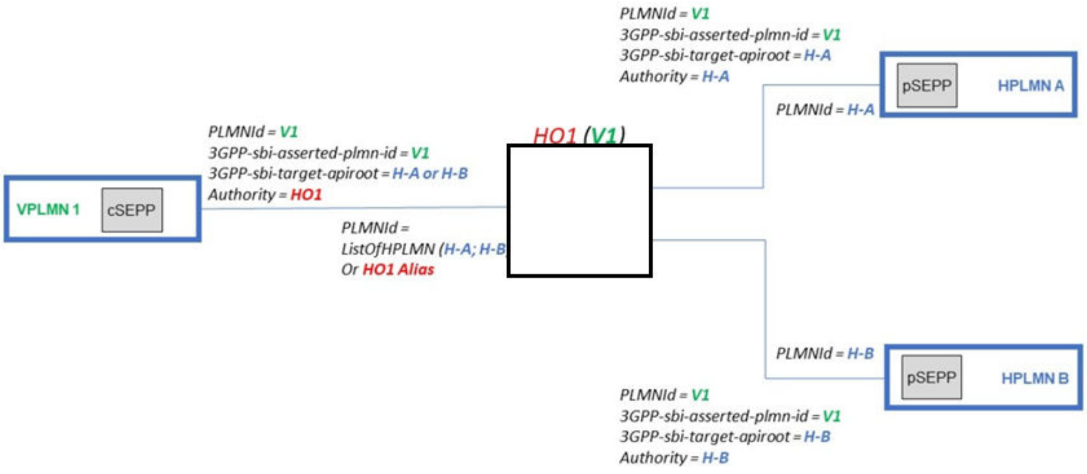
Figure 3-27 Remote SEPPs Connected to Roaming Hub



Hosted SEPP

Selective routing was not supported by Roaming Hub phase 1. The Hosted SEPP functionality supports selective routing between Remote SEPP Sets. There are operators who do not have a dedicated SEPP for each of the roaming partners. They can approach an aggregator to enable them to have bilateral roaming agreements on their behalf with the roaming partner. These SEPPs are for the aggregator and may be shared with other MNOs and can connect to multiple partners. Smaller operators who have roaming agreements with a lot of partners have a challenge in managing roaming connections. Additionally, they would prefer to have a lesser OPEX by not having to invest in multiple SEPPs. They would reach out to aggregators who would assist in managing the connections. This brings the following simplification for them, by delegating roaming traffic to external Roaming Hub provider (shown as HO1 in picture below). HO1 hosts all the roaming agreements on behalf of PLMNs.

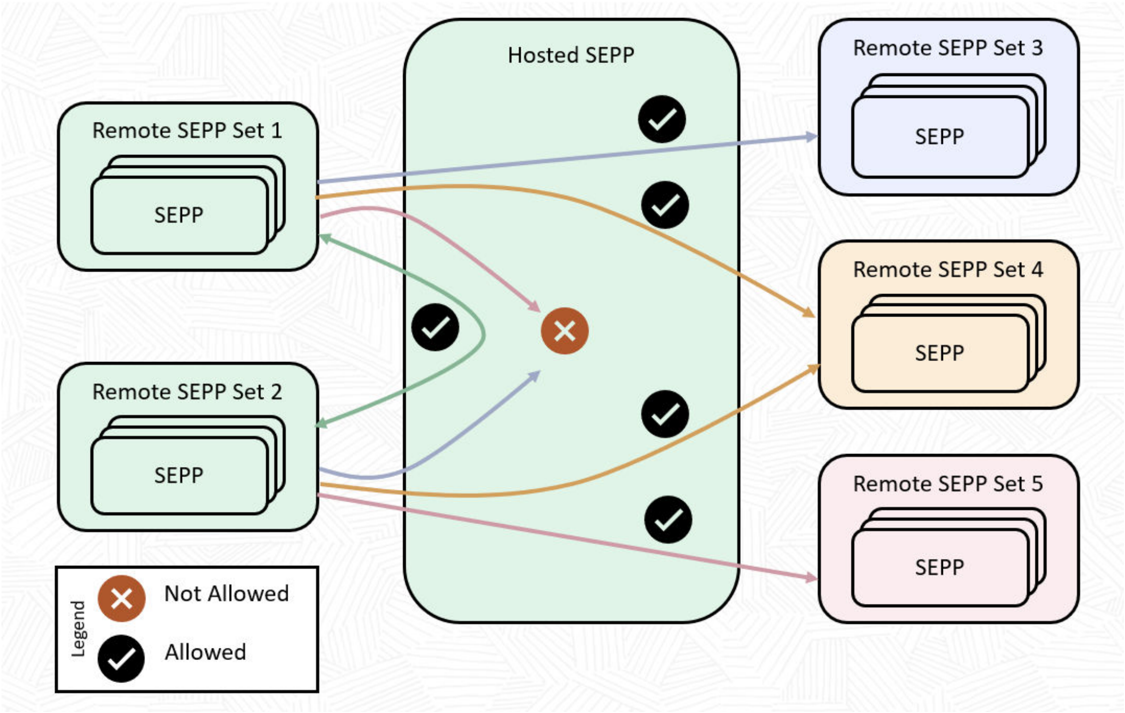
Figure 3-28   Hosted SEPP



Feature Overview

The following diagram represents the overview of Hosted SEPP functionality.

Figure 3-29   Hosted SEPP Overview



Connectivity Between Remote SEPP Sets Allowed by Hosted SEPP:

Consumer Remote SEPP Set	Producer Remote SEPP Set	Allowed – Yes/No
Remote SEPP Set 1	Remote SEPP Set 2	YES
Remote SEPP Set 1	Remote SEPP Set 3	YES
Remote SEPP Set 1	Remote SEPP Set 4	YES
Remote SEPP Set 1	Remote SEPP Set 5	NO
Remote SEPP Set 2	Remote SEPP Set 1	YES
Remote SEPP Set 2	Remote SEPP Set 3	NO
Remote SEPP Set 2	Remote SEPP Set 4	YES
Remote SEPP Set 2	Remote SEPP Set 5	YES
Remote SEPP Set 3	ANY	NO
Remote SEPP Set 4	ANY	NO
Remote SEPP Set 5	ANY	NO

- Hosted SEPP can connect to multiple Remote SEPP Sets.
- Selective routing is allowed from cSEPP (Remote SEPP Set 1, Remote SEPP Set 2) to pSEPP ((Remote SEPP Set 3, Remote SEPP Set 4, and Remote SEPP Set 5).
- Hosted SEPP, while not in the physical network of the partner, or can be in the trust boundary of the partner.
- Global Error Response can be configured on the Hosted SEPP, if routing is not allowed to remote SEPP.

You can configure the Remote SEPP Set in accordance with the connectivity mentioned in the table. Each of the Remote SEPPs should have an associated Remote SEPP Set. Consumer Remote SEPP Set should be configured with the "AllowedProducerRemoteSEPPSets". This field should contain the list of the Producer Remote SEPP Sets which are allowed from the consumer. For example : PSEPP-1 Remote SEPP Set should have the list containing (PSEPP-3 and PSEPP-4) Remote SEPP Sets.

Configuration limit for Hosted SEPP:

The following table indicates the configuration limit for Hosted SEPP:

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

Configuration Work flow

Hosted SEPP feature can be enabled through REST APIs or through CNC Console Allowed P-RSS Validation feature option. Hosted SEPP provides the flexibility to selectively route from one SEPP to the other SEPP. A new attribute has been introduced in the RemoteSEPPSets configuration, that is "AllowedProducerRemoteSeppSets", to decide if the source Remote SEPP Set can send the traffic to destination Remote SEPP Set or not. If the destination Remote SEPP Set is in the list of the source RSS "AllowedProducerRemoteSEPPSets", then only source SEPP can send the traffic to the destination Remote SEPP Set. For example:

```
curl -v http://127.0.0.1:9090/sepp-configuration/v1/remoteseppset -d '{
    "name": "csepp", "primary": "remote", "allowedListName": "Default",
    "allowedProducerRemoteSeppSets" : ["psepp1,psepp2"] }' -H 'Content-
Type:application/json' -X
    POST
```

Source Remote SEPP Set

Here, Remote SEPP Set (csepp) has two allowedProducerRemoteSeppSets "psepp1,psepp2" in the list. Therefore, Remote SEPP Set-csepp can send the traffic to the Remote SEPP Set(psepp1) and Remote SEPP Set (psepp2) only. Note that vice-versa is not applicable in this scenario. Remote SEPP Set(psepp), still cannot send the traffic to the Remote SEPP Set(csepp) if not configured.

> **Note**
>
> Remote SEPP Operators (in SEPP mode) need to configure the combined (incoming and outgoing) PLMN-ID List in the remote profile while establishing TLS handshake with Hosted SEPP.

In case of Hosted SEPP, consumer Remote SEPP shall be a part of a Remote SEPP Set as the selective routing is applicable between Remote SEPP Sets.

Managing Hosted SEPP Feature

Enable

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

Prerequisite: The parameter **nrfClientEnabled** must be set to false in the custom values.

- **Enable using REST API:** Set **enabled** to true in Hosted SEPP REST API.

  ```
  '{
   "hostedSepp": {
    "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:** In the CNC Console GUI, from the left navigation menu, navigate to SEPP and then click **System Options**. Select **Allowed P-RSS Validation** Options. Set **Enable Allowed P-RSS Validation** as **True** or **False** on the right pane.

  More details about CNC Console Configurations, see Configuring SEPP using CNC Console section.

> **Note**
>
> **nrfClientEnabled** parameter is used to enable the Roaming Hub Mode as Hosted SEPP is applicable for Roaming Hub Mode only.

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 Set* 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 Hosted SEPP feature specific metrics:

- ocsepp_allowed_p_rss_routing_failure_total

Following are the Hosted SEPP feature specific KPIs:

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

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.

3.25 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.

This feature is designed to support the following:

- Validating the headers present in the SBI Message.
- 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.

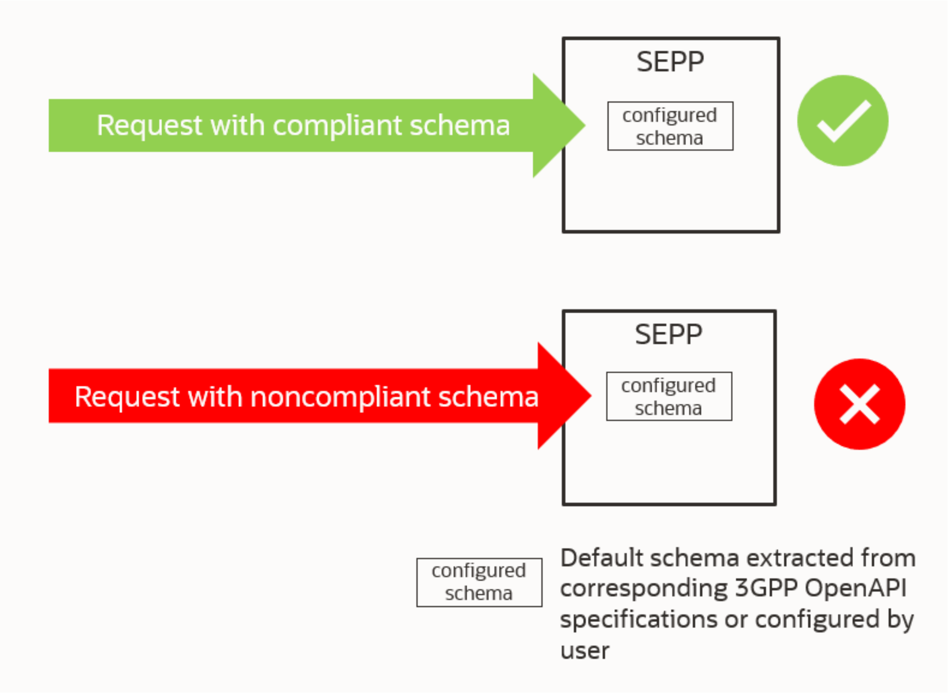
As part of the message validation feature, the following major scenarios are getting validated:

- 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 3-30 Cat-0 Functionality


SEPP
configured schema
Request with compliant schema
SEPP
configured schema
Request with noncompliant schema
configured schema
Default schema extracted from corresponding 3GPP OpenAPI specifications or configured by user

Feature Configurations

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

Figure 3-31 CNC Console options available for Cat-0 SBI Message Schema Validation feature in SEPP Mode

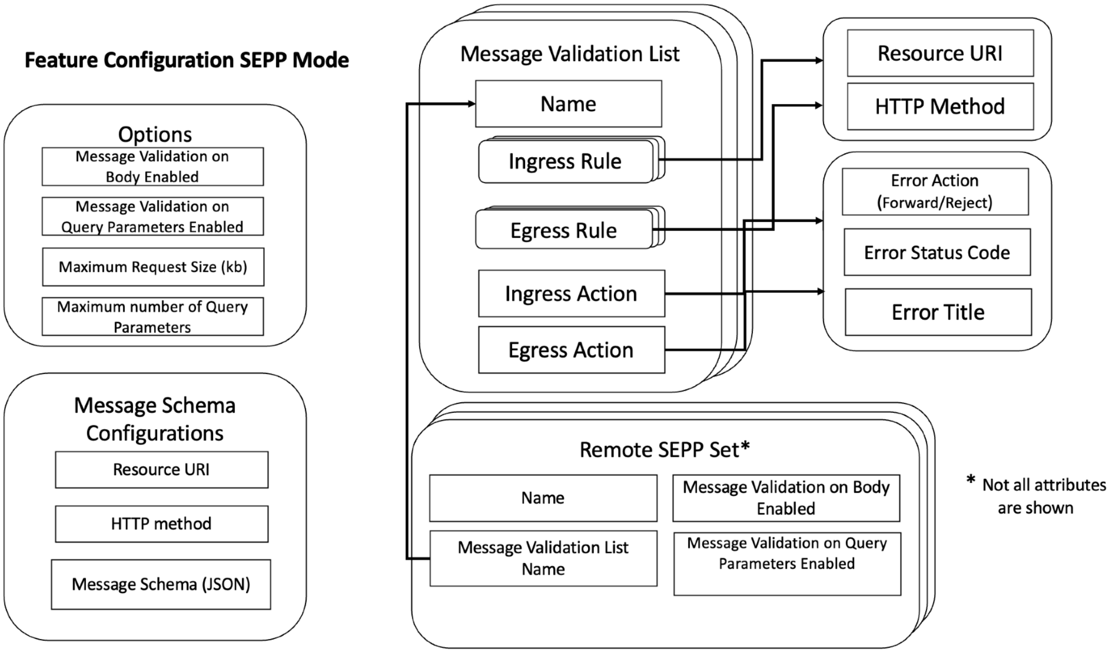
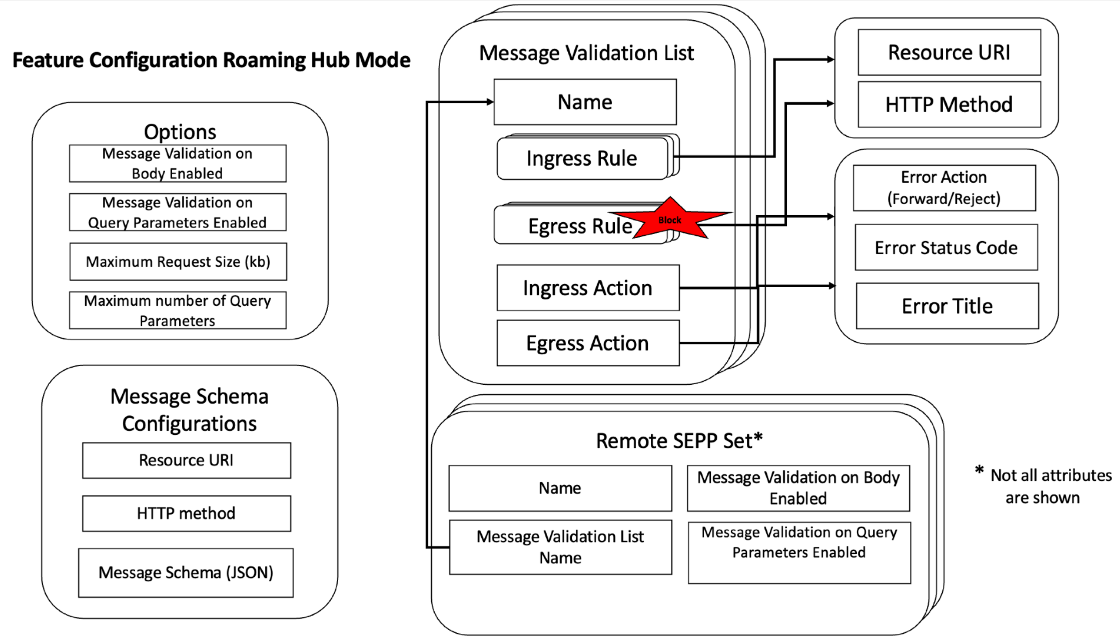

Feature Configuration SEPP Mode
Options
Message Validation on Body Enabled
Message Validation on Query Parameters Enabled
Maximum Request Size (kb)
Maximum number of Query Parameters
Message Schema Configurations
Resource URI
HTTP method
Message Schema (JSON)
Message Validation List
Name
Ingress Rule
Egress Rule
Ingress Action
Egress Action
Resource URI
HTTP Method
Error Action (Forward/Reject)
Error Status Code
Error Title
Remote SEPP Set*
Name
Message Validation on Body Enabled
Message Validation List Name
Message Validation on Query Parameters Enabled
* Not all attributes are shown

Figure 3-32 CNC Console options available for Cat - 0 SBI Message Schema Validation feature in Roaming Hub Mode


Feature Configuration Roaming Hub Mode
Options
Message Validation on Body Enabled
Message Validation on Query Parameters Enabled
Maximum Request Size (kb)
Maximum number of Query Parameters
Message Schema Configurations
Resource URI
HTTP method
Message Schema (JSON)
Message Validation List
Name
Ingress Rule
Egress Rule
Block
Ingress Action
Egress Action
Resource URI
HTTP Method
Error Action (Forward/Reject)
Error Status Code
Error Title
Remote SEPP Set*
Name
Message Validation on Body Enabled
Message Validation List Name
Message Validation on Query Parameters Enabled
* Not all attributes are shown

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 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.

schema or content (application/json): json schema for the parameter.

The following example shows the schemas (JSON array format) of two query parameters (supported-features and plmn-id):

```
[ {
  "name" : "supported-features",
  "in" : "query",
  "required" : false,
  "schema" : {
    "pattern" : "^[A-Fa-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.

3.26 Cat 2 - Network ID Validation Feature

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.

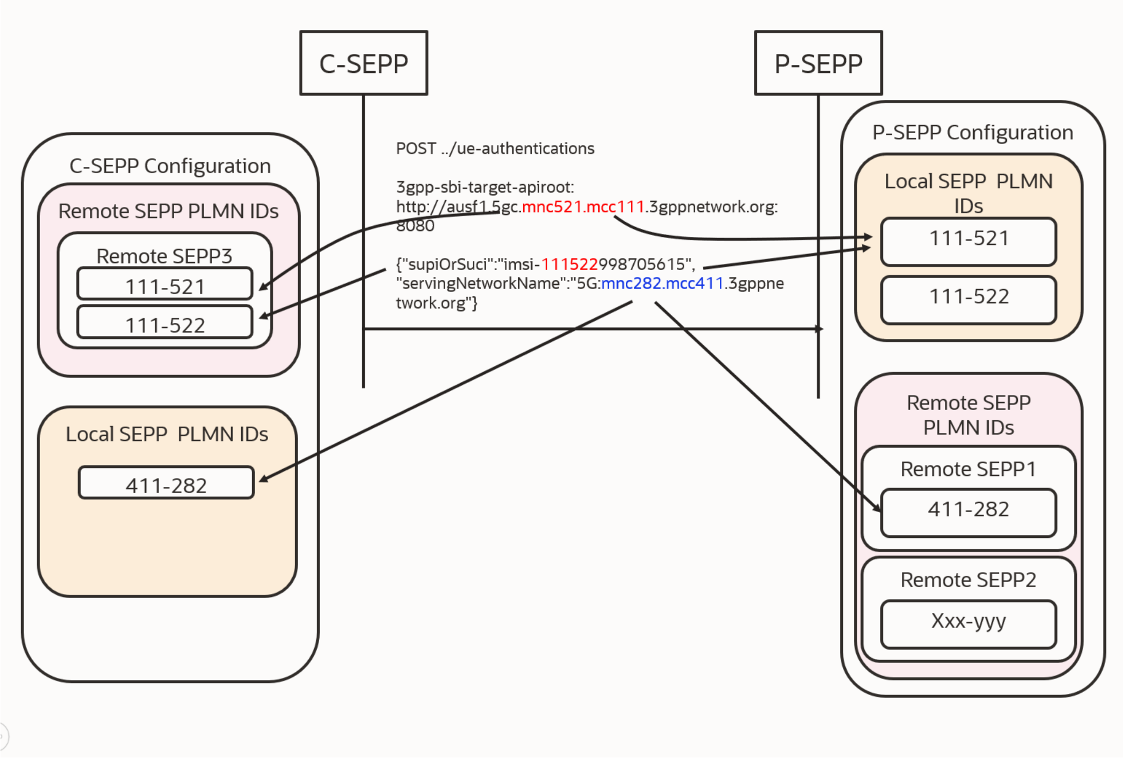
This feature is not applicable to SBI responses that is, only SBI requests are validated.

Design and Architecture

The following diagram represents the Cat 2 – Network ID validation feature in SEPP mode.

SEPP Mode

Figure 3-33 SEPP Mode



```
POST ../ue-authentications

3gpp-sbi-target-apiroot:
http://ausf1.5gc.mnc521.mcc111.3gppnetwork.org:
8080

{"supiOrSuci":"imsi-111522998705615",
"servingNetworkName":"5G:mnc282.mcc411.3gppne
twork.org"}
```

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 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.

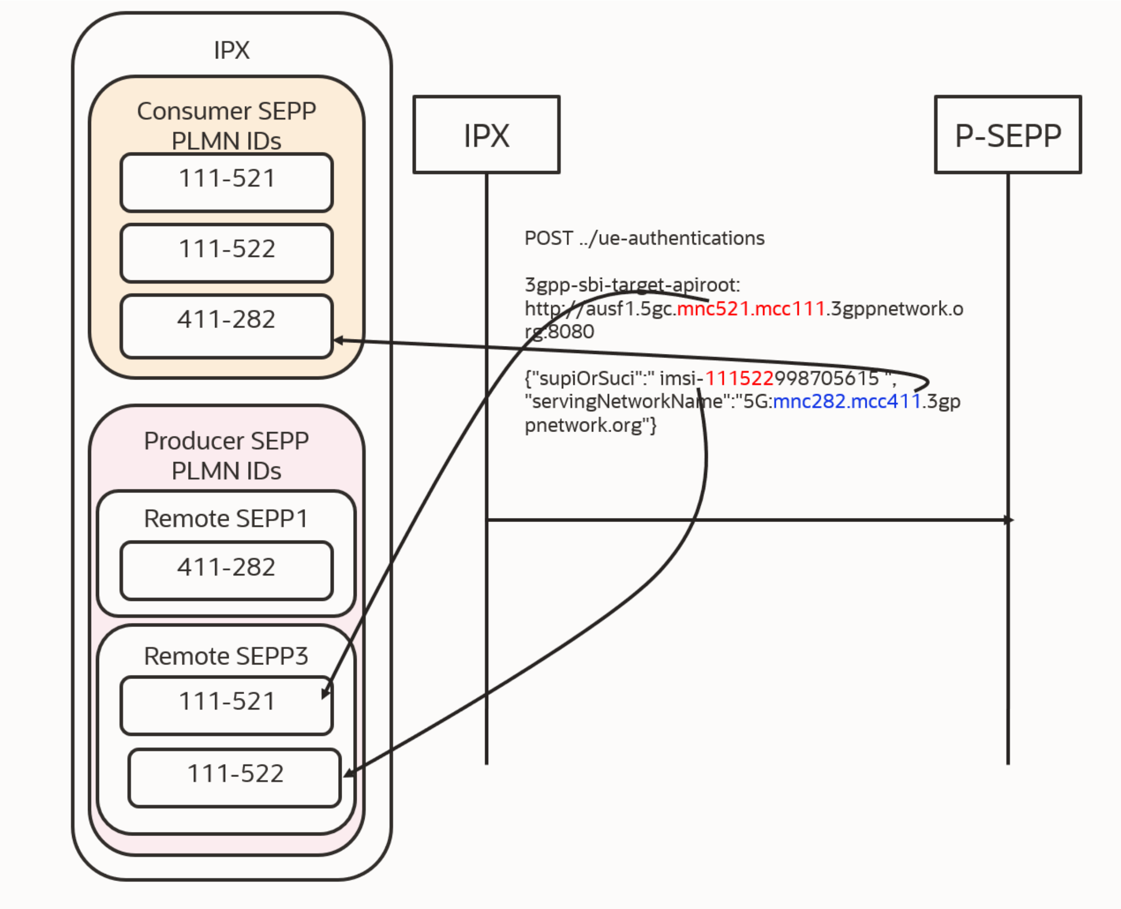
When the feature is enabled at P-SEPP:

- 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 *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 3-34 Roaming Hub mode



```
POST ../ue-authentications

3gpp-sbi-target-apiroot:
http://ausf1.5gc.mnc521.mcc111.3gppnetwork.org:8080

{"supiOrSuci":" imsi-111522998705615 ",
"servingNetworkName":"5G:mnc282.mcc411.3gppnetwork.org"}
```

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:

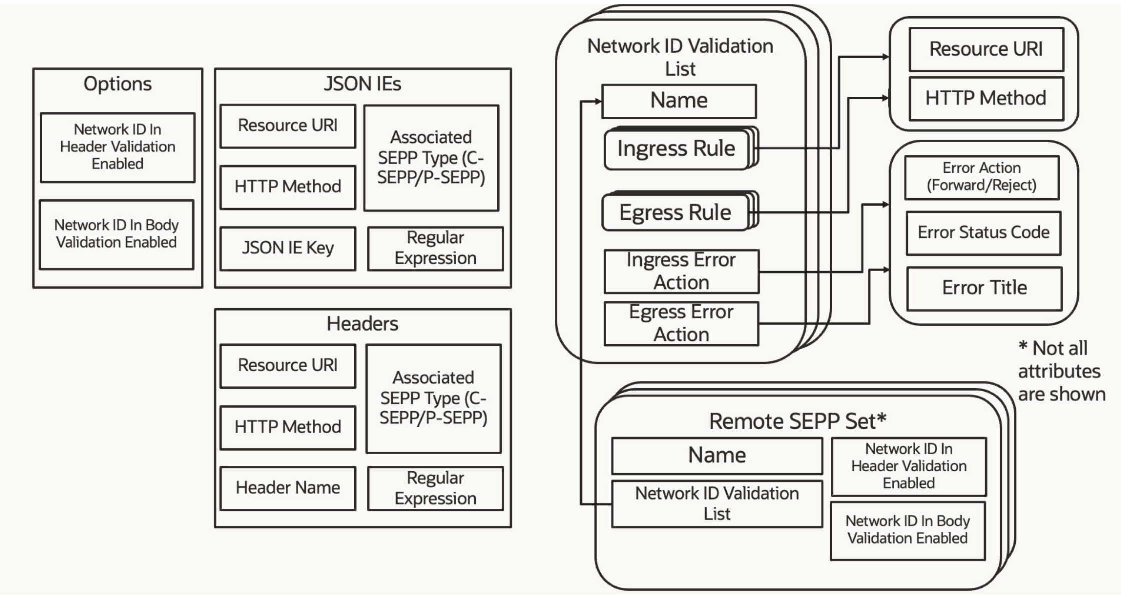
- If the user wants to validate the PLMN ID in *3gpp-sbi-target-apiroot* header, then user can add a configuration for this header in Header Configuration for all Service APIs or for a particular Service API. Once the configuration is done, and if it is configured for all Service APIs then for every 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 *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 3-35 CNC Console options available for Cat 2 – Network ID Validation

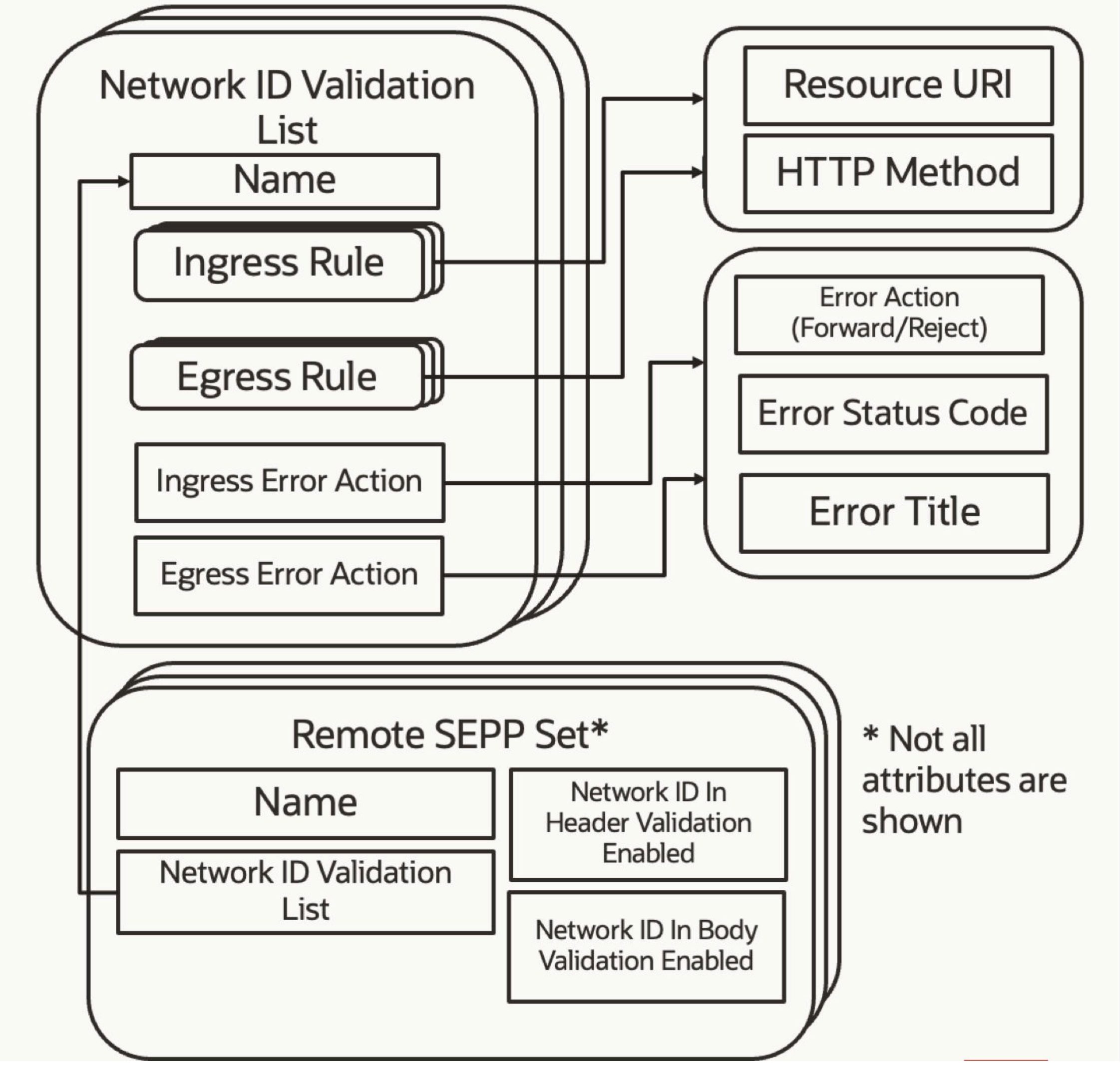

Options
Network ID In Header Validation Enabled
Network ID In Body Validation Enabled
JSON IEs
Resource URI
HTTP Method
JSON IE Key
Associated SEPP Type (C-SEPP/P-SEPP)
Regular Expression
Headers
Resource URI
HTTP Method
Header Name
Associated SEPP Type (C-SEPP/P-SEPP)
Regular Expression
Network ID Validation List
Name
Ingress Rule
Egress Rule
Ingress Error Action
Egress Error Action
Resource URI
HTTP Method
Error Action (Forward/Reject)
Error Status Code
Error Title
* Not all attributes are shown
Remote SEPP Set*
Name
Network ID Validation List
Network ID In Header Validation Enabled
Network ID In Body Validation Enabled

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 3-36   Feature Configuration – Network ID Validation List


Network ID Validation List
Name
Ingress Rule
Egress Rule
Ingress Error Action
Egress Error Action
Resource URI
HTTP Method
Error Action (Forward/Reject)
Error Status Code
Error Title
Remote SEPP Set*
Name
Network ID Validation List
Network ID In Header Validation Enabled
Network ID In Body Validation Enabled
* Not all attributes are shown

- 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 used as a sample to visualize the rules.
- Default list uses exhaustive number of rules, and it is 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 toplevel `plmnId` gets validated.

Note

- In C-SEPP and P-SEPP, by default, the three digit mnc is supported for identifiers that have associated SEPP type configured as C-SEPP and P-SEPP in Header and Body table configurations.
- If the user wants to support two digit mnc, then the user can configure the regex in Header and Body table configurations for that identifier. The two digit mnc configuration will be applicable for all the Remote SEPP Sets.

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 **messageFilteringOnPlmnHeaderValidation** and **messageFilteringOnPlmnBodyValidation** 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.

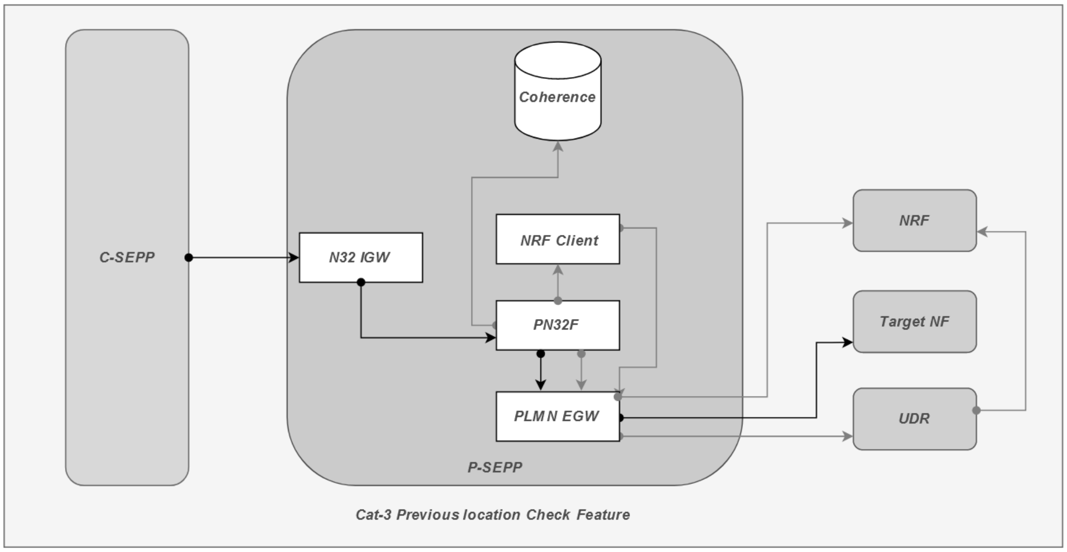
3.27 Cat-3 Previous Location Check Feature

Cat-3 Previous Location Check feature is a part of stateful security countermeasures. This feature verifies that only those messages are allowed to traverse the network, which are from an authenticated UE and from the same location from where the UE got authenticated previously.

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



Cat-3 Previous location Check Feature

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.

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": 5000,
    "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

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:

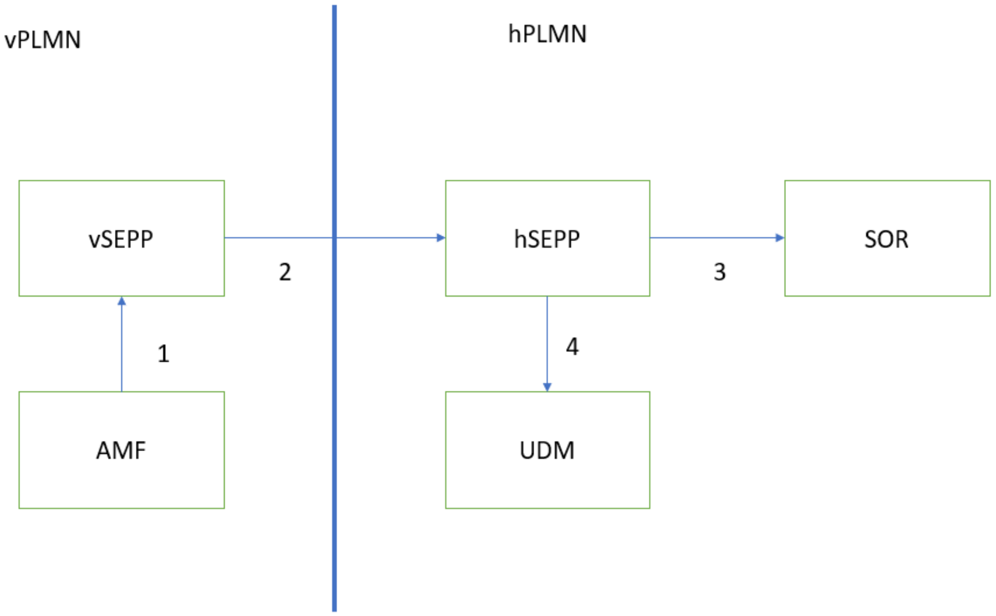
1. **Collect the logs**: For more information on how to collect logs, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*.
2. **Raise a service request**: See My Oracle Support for more information on how to raise a service request.

3.28 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.



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.

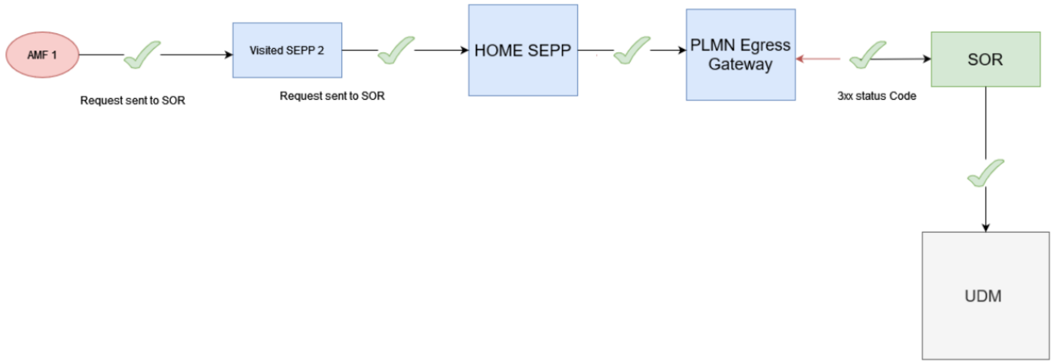
Based on the redirection of response from SOR platform, the two modes are:

- 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 3-37 Redirection Disabled Mode

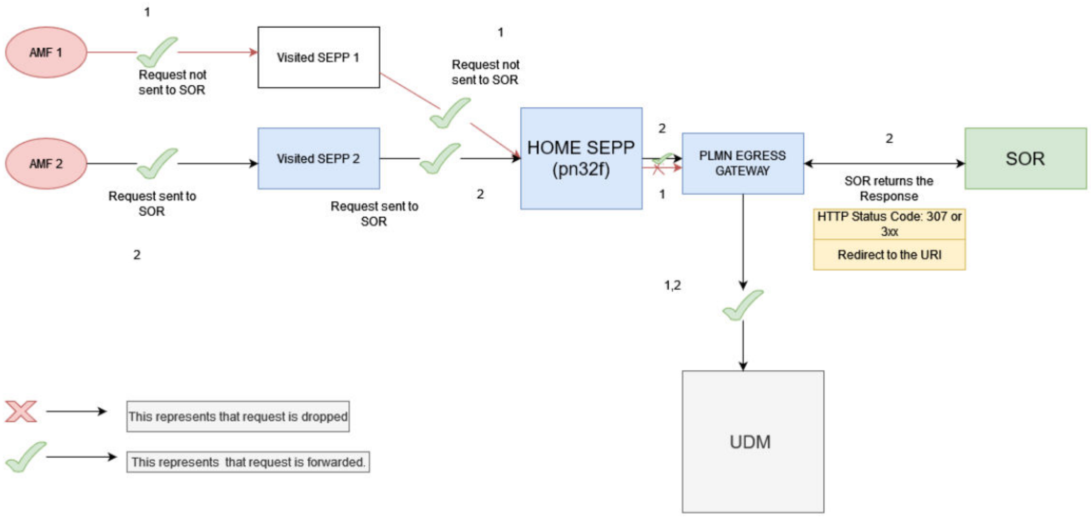


Here, SOR platform is acting as a front end. Hence, the search time is less and the SOR platform does the load balance.

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 3-38 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.

3.29 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.

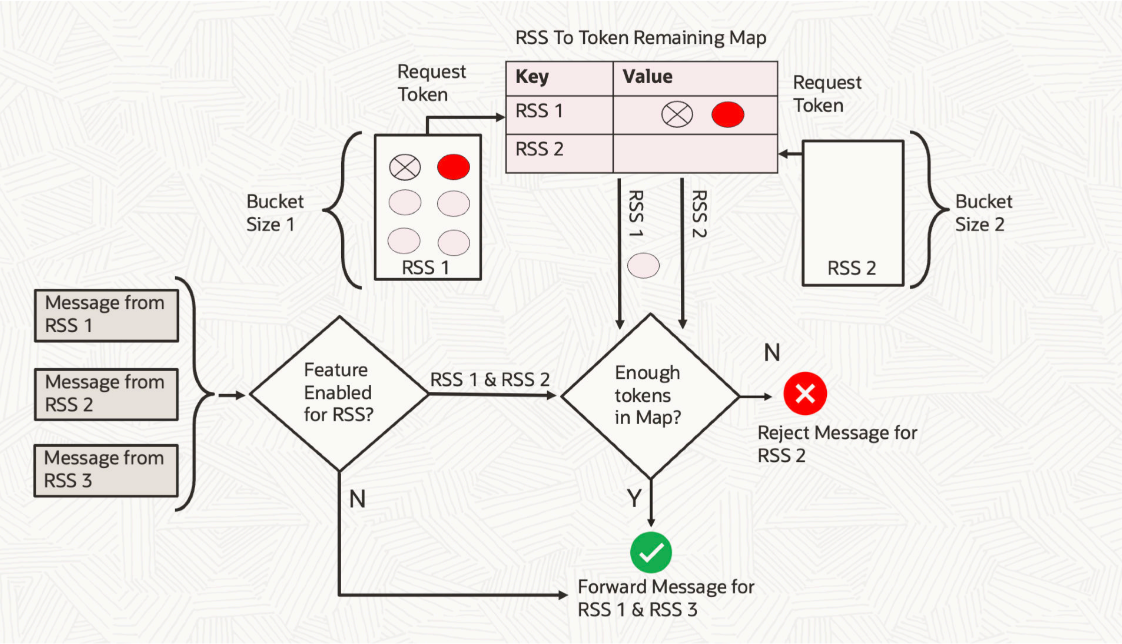
Algorithm

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 recommented to have the Request Token value between 3 to 5% of the refill rate.
- Tokens and duration defines the traffic rate.

Figure 3-39 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.

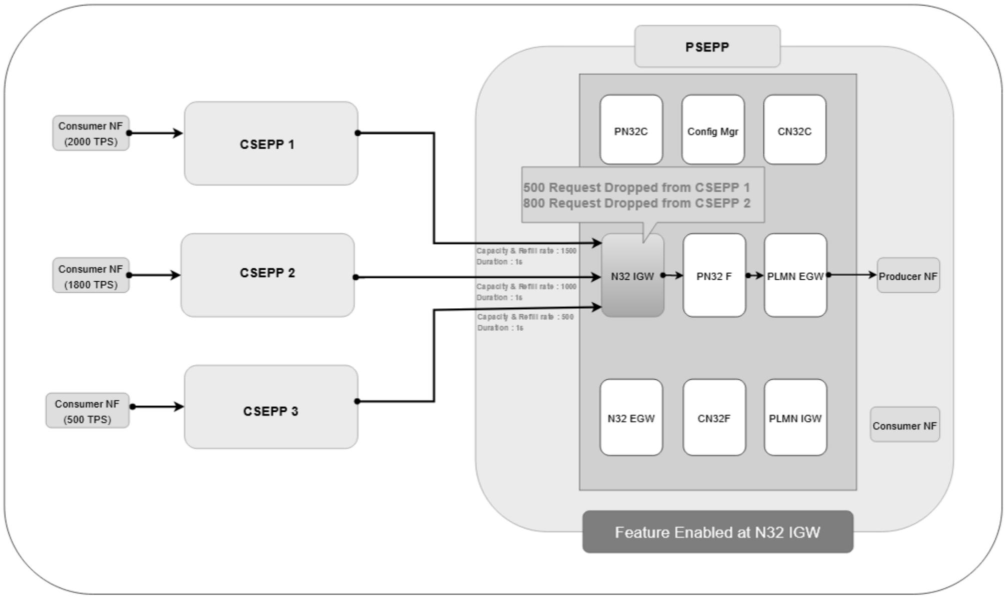
When message from RSS 2 is processed, It is checked if there are enough tokens present in RSS to Token remaining mapping for RSS 2. Since there no tokens available for RSS 2, this message will be rejected with user configurable status code and title.

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 3-40 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.

3.30 Support for 3-Site Georedundancy

The SEPP architecture supports Geographical Redundant (Georedundant) deployments to ensure high availability and redundancy. It offers three-sites georedundancy to ensure service availability when one of the SEPP sites is down.

- All the sites that register with NRF work independently and are in Active state.
- Based on the Rank, each SEPP site subscribes to NRF for any state change of other SEPP sites and gets notified when an SEPP site goes down.
- The NFs in a given site need to configure one of the georedundant SEPP as the primarySEPP and others as secondarySEPP and tertiarySEPP, respectively.
- When the primarySEPP is available, the NFs send service requests to the primarySEPP. When the SEPP at the primary site is unavailable, the NFs redirect service requests to the secondarySEPP or tertiarySEPP, until the primarySEPP's Active status is restored.
-
- The SEPP's data gets replicated between the georedundant sites by using cnDBtier replication service.

  With SEPP georedundant feature, the SEPP services continues to work as independent service operations.

Following are the prerequisites for georedundancy:

- Each site is independent of other sites.
- Each site must configure remote SEPP sites as georedundant mates.
- The configurations at each site must be same. The SEPP at all sites must handle the NFs in the same manner.
- NFs need to configure georedundant SEPP details as Primary, Secondary, and TertiarySEPPs.
- Name of SEPP database must be unique in each site.
- Once the Georedundancy feature is enabled on a site, it cannot be disabled.
- The user must replicate the configuration done at one site to all the other sites for seamlessly performing the georedeundancy.
- Georedundant sites must have REST based configuration as explained in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy REST Specification Guide*.

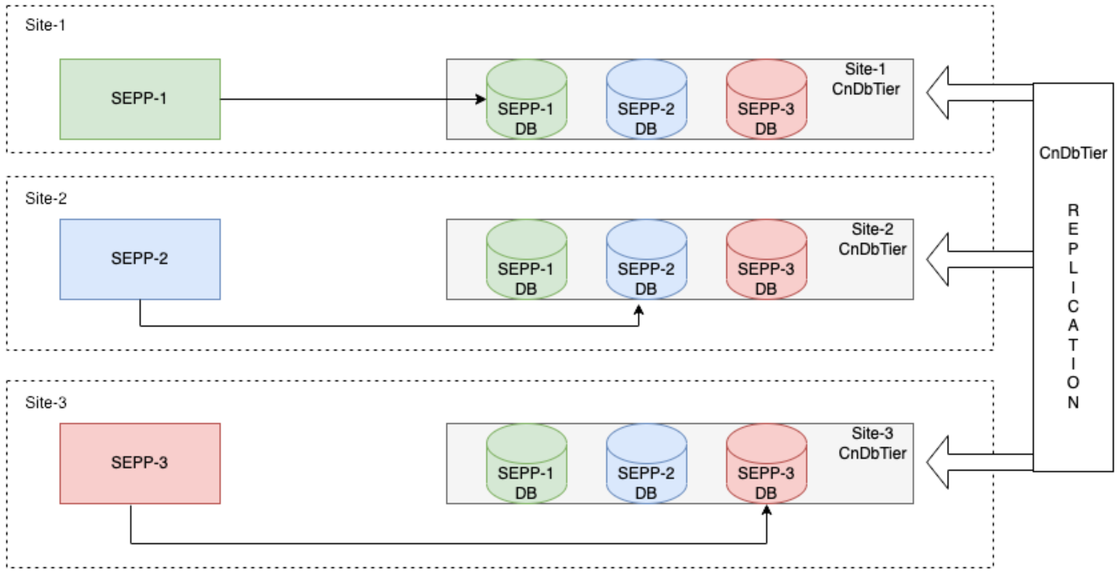
Three-Site Georedundancy Deployment

When SEPP is deployed as three-sites georedundant instances, then:

- Every active site contains one instance of SEPP
- Each SEPP instance has its own cnDBTier.
- SEPP database for each SEPP must be unique.

The following diagram depicts the topology for three-site georedundant SEPP deployment:

Figure 3-41 Three-Site Georedundancy Deployment



The topology has the following configurations:

Site 1: SEPP-1 communicates with SEPP-1 DB deployed in Site-1 cnDBTier.

Site 2: SEPP-2 communicates with SEPP-2 DB deployed in Site-2 cnDBTier.

Site 3: SEPP-1 communicates with SEPP-3 DB deployed in Site-3 cnDBTier.

The replication between all the cnDBTier at multiple sites are enabled.

All the configurations on each site is managed by site owners. Each site should have the same configurations for smooth switchover in case of failure.

The georedundant SEPP site processes the request only if the below conditions are met:

- The consumer SEPP ensures that the original subscribing SEPP is down.
  To keep track of peer SEPP status, every SEPP instance in the georedundant deployment subscribes with NRF for receiving notifications about the change in the status of peer instances. Based on this subscription, whenever the peer SEPP comes up or goes down, NRF notifies all the other instances of the deployment.
- The DB replication must be in good health.

Managing Support for Georedundancy

Deploy

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 DBTier Installation and Upgrade Guide.*
2. The Database Replication Channels between the sites must be up. For information about installing cnDBTier and parameters required for setting the DB 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*.

Configure

Enable Georedundancy Feature

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 stand-alone SEPP.

Helm Configuration for Database:

Configure the following parameters in the `sepp-23.1.0.0.0.custom-values.yaml` file for all three 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"
```

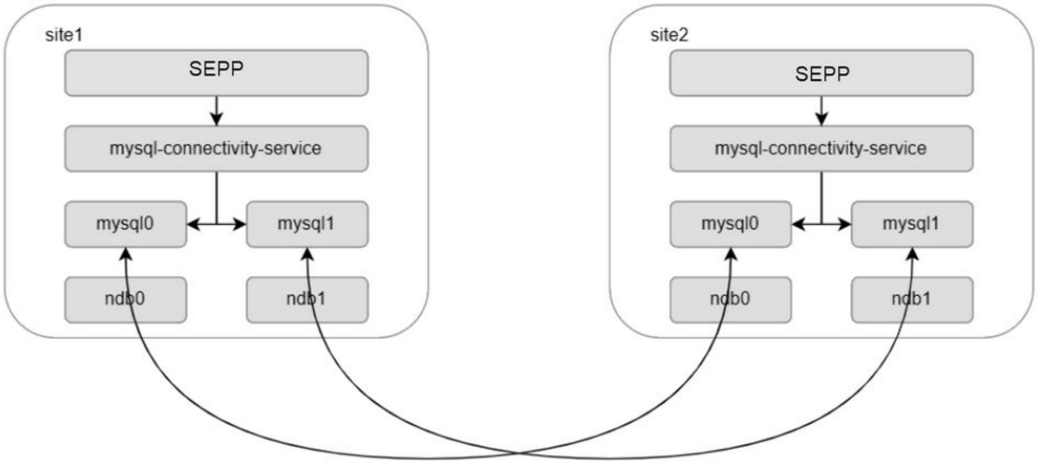
For more information about configuring the parameters, see *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

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`):

3.30.1 Two-Site Georedundancy

The following diagram depicts the topology for two-site georedundant SEPP deployment:

Figure 3-42 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.

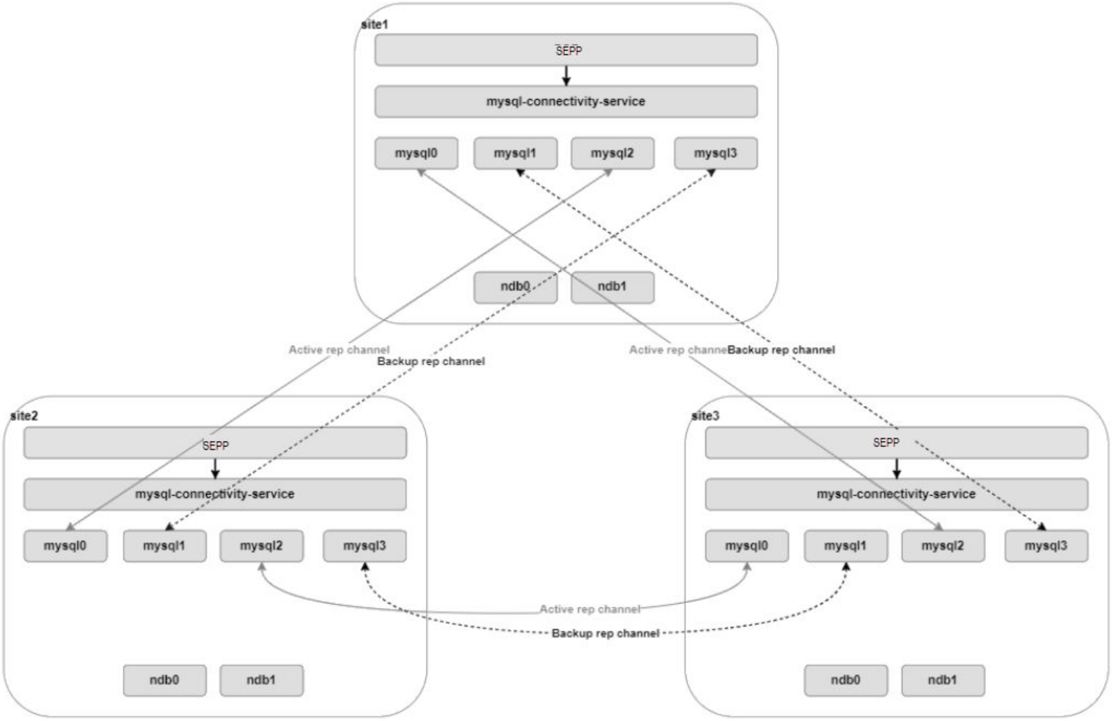
The benefits of two-site georedundancy are:

- 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.

3.30.2 Three-Site Georedundancy

The following diagram depicts the topology for three-site georedundant SEPP deployment:

Figure 3-43 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:

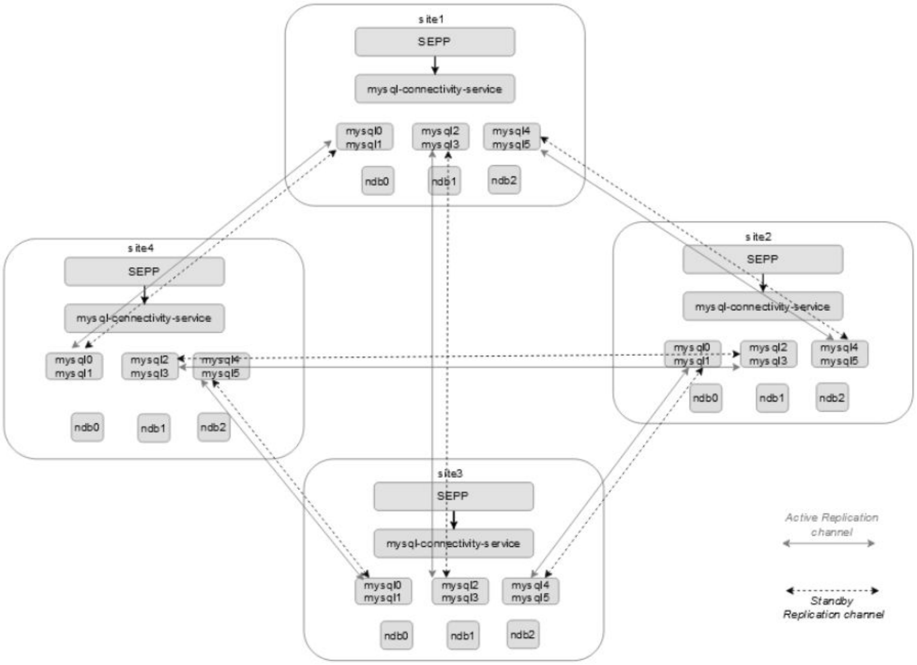
- In case of a single site failure, the remaining two sites take up the load of the failed site.
- Each site uses the SQL nodes for active and standby replication channels for high availability of the replication channels.
- In case of single or dual site failure (or connection failure or any other failure), 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.

3.30.3 Four-Site Georedundancy

The following diagram depicts the topology for four-site georedundant SEPP deployment:

Figure 3-44 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.

3.30.4 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.

3.30.4.1 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*.

3.30.4.2 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`):

3.30.5 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.

3.30.6 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*.

3.30.7 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*.

3.31 Support for Kubernetes Resource

3.31.1 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.

3.31.2 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 undisrupted 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 3-14 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.

3.32 Alternate Routing 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.

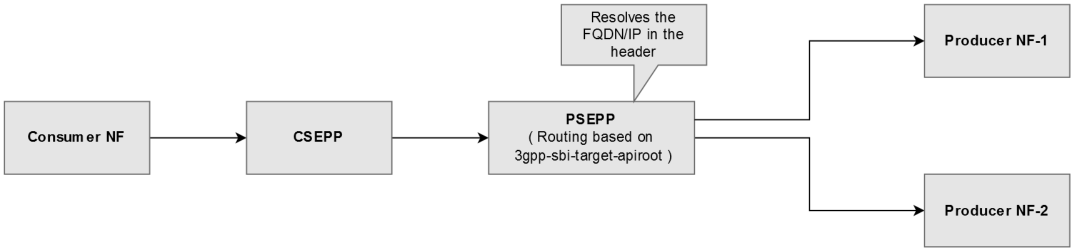
Using this feature, SEPP performs alternate routing of messages by querying SRV records with the DNS server to find an alternate producer NF and route the service request. SEPP supports retrieval of NRF, SCP, and Producer NFs' records using DNS SRV queries at plmn-egress-gateway. SEPP uses either NF 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 3-45 Direct Communication


Resolves the FQDN/IP in the header
Consumer NF
CSEPP
PSEPP
(Routing based on 3gpp-sbi-target-apiroot)
Producer NF-1
Producer NF-2

Routing to NRF, SCP, and Producer NFs Through DNS SRV

SEPP supports priority and weight based routing 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 distributed based on the weight value assigned to them.

Below is the format of SRV records defined at DNS server:

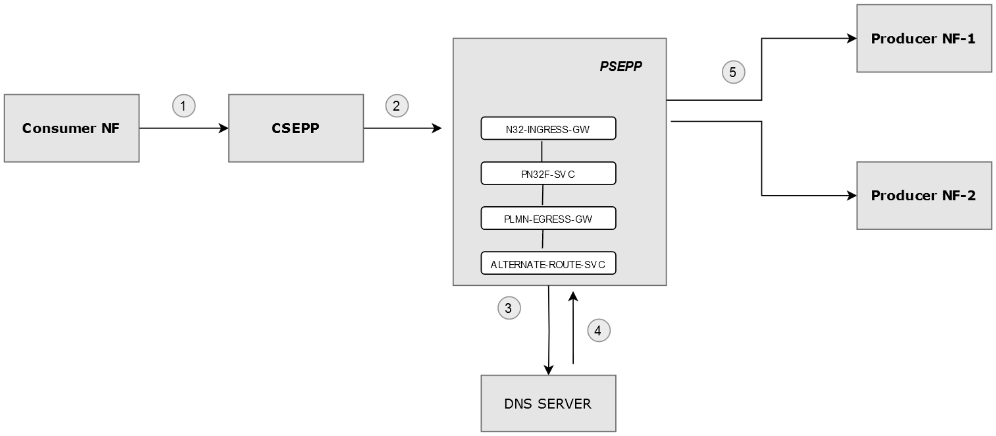
_Service._Proto.Name TTL Class SRV Priority Weight Port Target

_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 multiple IP address in the DNS server.

Figure 3-46   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.

Design and Architecture

SEPP supports multiple deployments for 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 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 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.

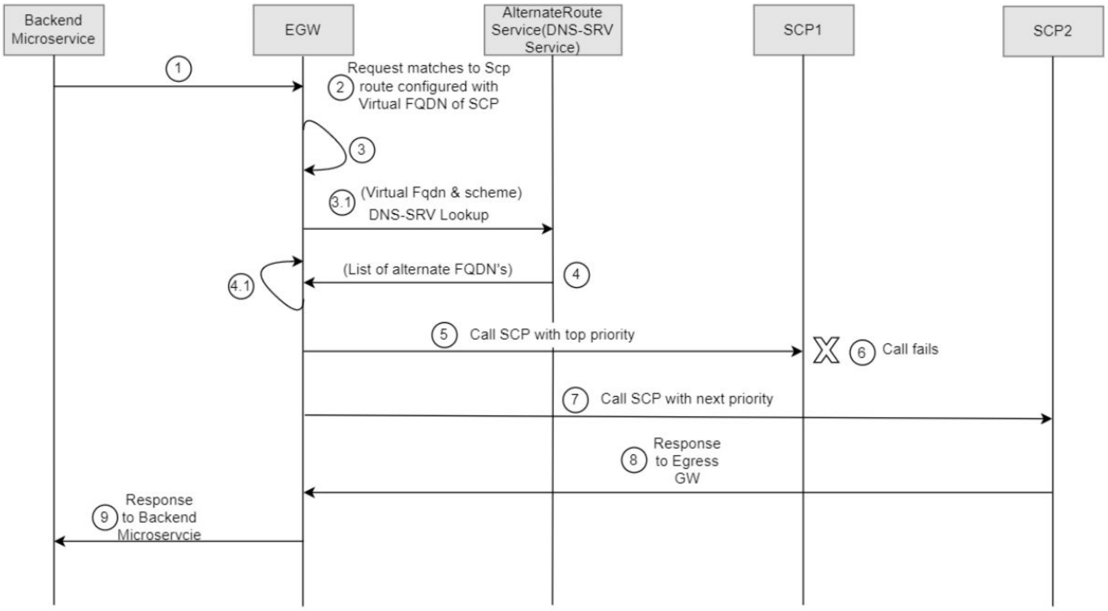
Alternate Route Service uses the DNS SRV records for the resolution of virtual FQDN. PLMN Egress Gateway uses the virtual FQDN of peer instances to query the Alternate Route Service and get the list of alternate FQDNs with a priority and weight assigned to each one of them. The Egress Gateway selects 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.


Backend Microservice
EGW
AlternateRoute Service(DNS-SRV Service)
SCP1
SCP2
1
2 Request matches to Scp route configured with Virtual FQDN of SCP
3
3.1 (Virtual Fqdn & scheme) DNS-SRV Lookup
4.1
(List of alternate FQDN's)
4
5 Call SCP with top priority
6 Call fails
7 Call SCP with next priority
8 Response to Egress GW
9 Response to Backend Microservcie

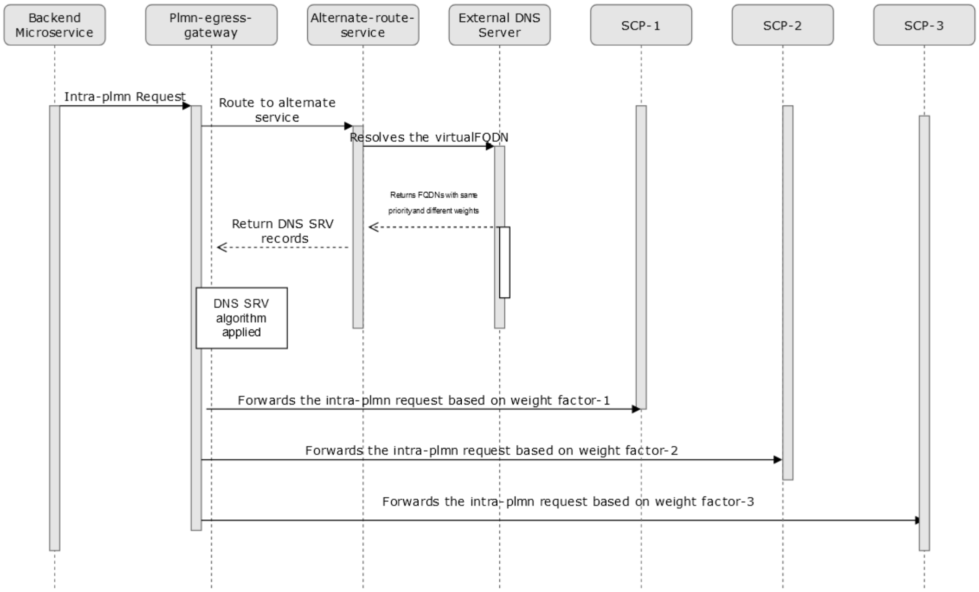
1. PLMN Egress Gateway (EGW) receives a request from the backend microservice.
2. PLMN EGW maps the request to a route configured with virtual FQDN of SCP. Since the route is configured with virtual FQDN of SCP, the virtual FQDN is resolved to the list of alternate FQDNs.
3. PLMN EGW gets the list of 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

In weight based routing, virtual FQDN configured for SCP instances with same priorities and different weights.


Backend Microservice
Plmn-egress-gateway
Alternate-route-service
External DNS Server
SCP-1
SCP-2
SCP-3
Intra-plmn Request
Route to alternate service
Resolves the virtualFQDN
Returns FQDNs with same priorityand different weights
Return DNS SRV records
DNS SRV algorithm applied
Forwards the intra-plmn request based on weight factor-1
Forwards the intra-plmn request based on weight factor-2
Forwards the intra-plmn request based on weight factor-3

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 distributes the traffic to SCP instances in proportion to the weight factor defined.
6. The call is re-routed to next 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 micro-service 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.

> **Note**
>
> This feature is not applicable for the following:
>
> - SOR (Steering of Roaming)
> - Roaming Hub mode
> - N32 Egress Gateway

Managing Alternate Routing based on the DNS SRV Record for Home Network Functions Feature

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}."
configureDefaultRoute: true
sbiRoutingConfigMode: REST
routeConfigMode: REST
```

The following parameter must also be enabled at nrf-client section under global parameters:

```
alternateRouteServiceEnable: 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 REST API.

- **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.*

> **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.

3.33 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.

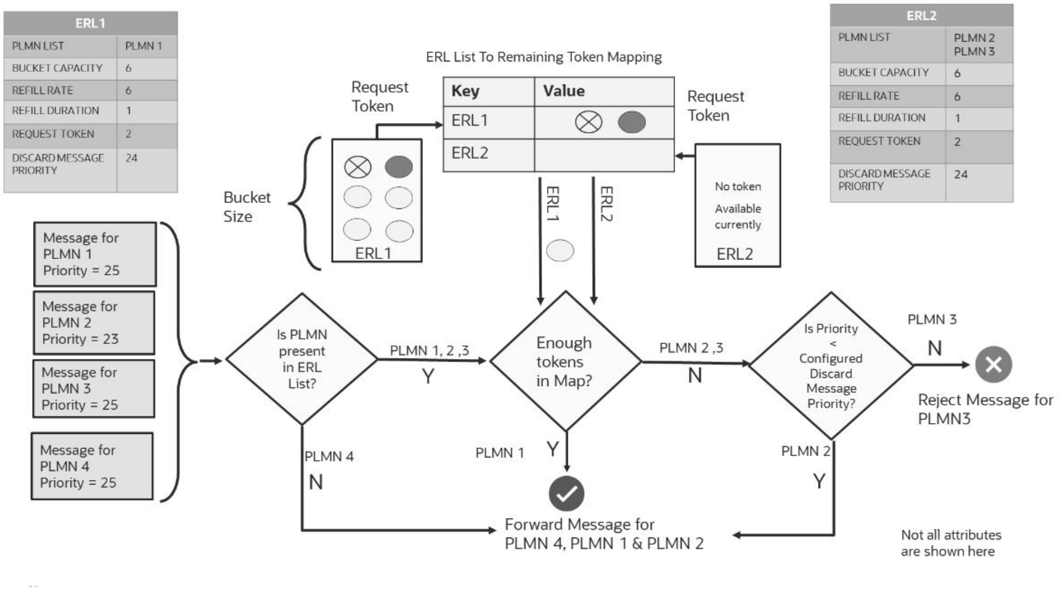
The token bucket algorithm can be described as the following:

- 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 3-47   Token Bucket Algorithm



ERL1	
PLMN LIST	PLMN 1
BUCKET CAPACITY	6
REFILL RATE	6
REFILL DURATION	1
REQUEST TOKEN	2
DISCARD MESSAGE PRIORITY	24

ERL List To Remaining Token Mapping

Key	Value
ERL1	⊗ ●
ERL2	

ERL2	
PLMN LIST	PLMN 2 PLMN 3
BUCKET CAPACITY	6
REFILL RATE	6
REFILL DURATION	1
REQUEST TOKEN	2
DISCARD MESSAGE PRIORITY	24

Request Token
Bucket Size
ERL1
Request Token
No token Available currently
ERL2
ERL1
ERL2
Message for PLMN 1 Priority = 25
Message for PLMN 2 Priority = 23
Message for PLMN 3 Priority = 25
Message for PLMN 4 Priority = 25
Is PLMN present in ERL List?
PLMN 1, 2 ,3
Y
Enough tokens in Map?
PLMN 2 ,3
N
Is Priority < Configured Discard Message Priority?
PLMN 3
N
Reject Message for PLMN3
PLMN 4
N
PLMN 1
Y
PLMN 2
Y
Forward Message for PLMN 4, PLMN 1 & PLMN 2
Not all attributes are shown here

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 3-15 Summary

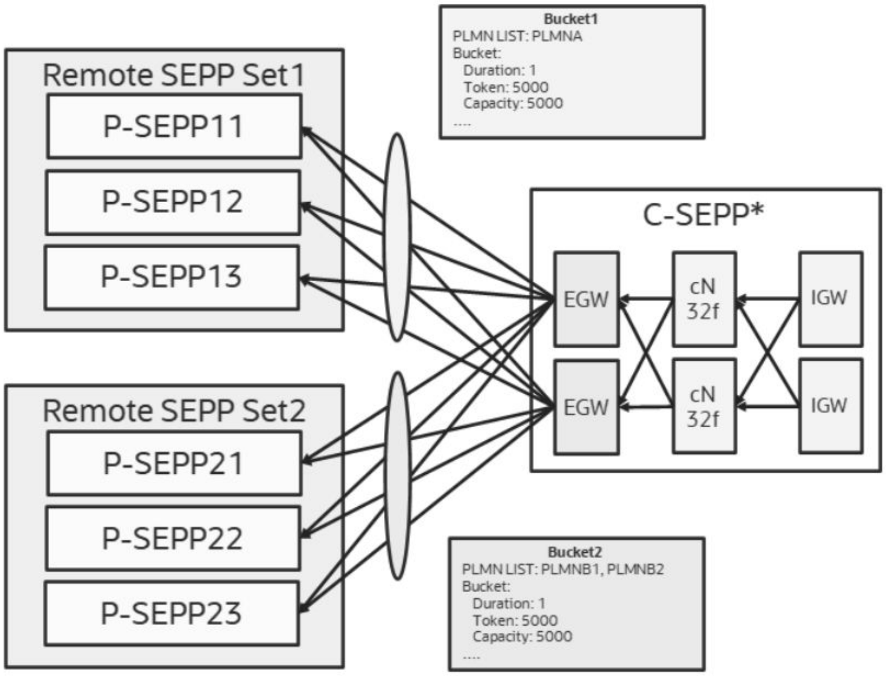
PLMN	Tokens in ERL List	Message Priority	Rejected/Forwarded
PLMN 1	Token available in ERL1's bucket	Ignore priority	Forwarded
PLMN 2	No token available in ERL2's bucket	High	Forwarded
PLMN 3		Low	Rejected
PLMN 4	No corresponding ERL list is configured	Ignore priority	Forwarded

> **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 3-48 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 **Engress 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.

3.34 Service mesh for intra-NF communication

Oracle SEPP leverages the Istio or Envoy service mesh (Aspen Service Mesh) for all internal and external communication. The service mesh integration provides inter-NF communication and allows API gateway co-working with service mesh. The service mesh integration supports the services by deploying a special sidecar proxy in the environment to intercept all network communication between microservices.

For more information on configuring ASM, see *Oracle Communications Cloud Native Core Network Repository Function Installation and Upgrade Guide.*

3.35 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 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.

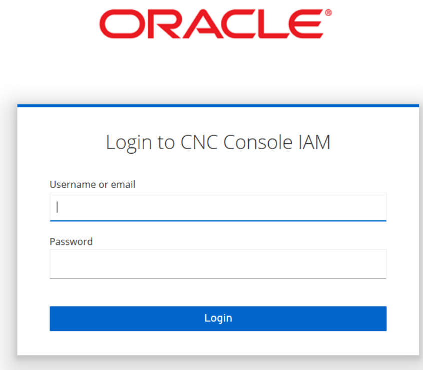
2. Save and close the hosts file.
   Ensure that a CNC user and password are created before logging into the CNC Console. For more information on how to create a CNC Console user and password, see *Oracle Communications Cloud Native Core Console Installation and Upgrade Guide*.

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

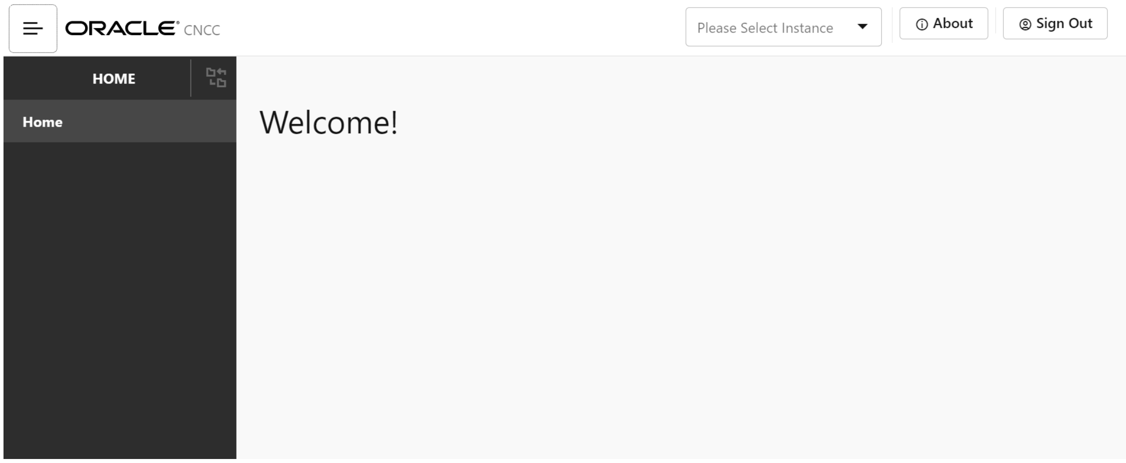


> **Note**
>
> <host name> is *cncc-iam-ingress-ip* and <port number> is *cncc-iam-ingress-port.*

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



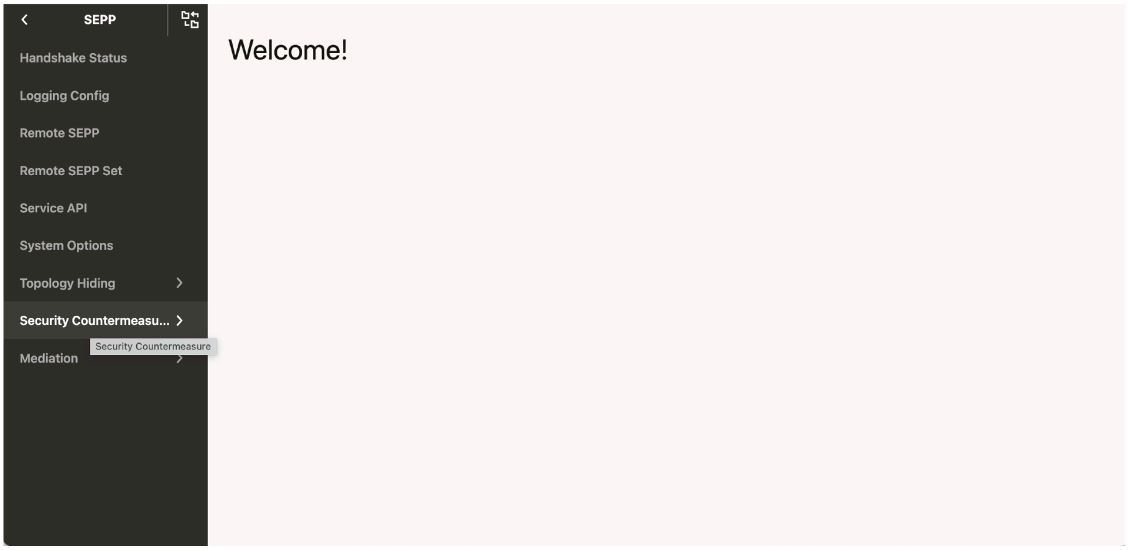
Select the required NF instance from the **Please Select Instance** drop-down list. The left pane displays the selected network function and on clicking the network function the corresponding APIs and configurations appears underneath.

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 IPX mode. To enable the SEPP mode, set the *operationMode* flag to true. To enable the IPX mode, set the *operationMode* flag to false. In IPX mode,the user cannot enable or disable the log level for nfmanagement and nfdscovery 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	Enabled/Available		Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
	Listing Screen	Edit Screen			
ServiceType	Yes	Yes	M	Enum	Name of the Common Service - N32 EGW, N32 IGW, PLMN EGW, PLMN IGW
Application Log Level	Yes	Yes	M	Enum	Log level for the application
Package Log Level	Yes	Yes	M	String	Log level of each corresponding packages. Example: For Package root, the loglevel can be ERROR.

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.

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.

The parameters are:

Table 4-4 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
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
Is Enabled	O	Boolean	Remote SEPP is True (enabled) or False (disabled).
Port	O	Integer	Port for SEPP NF
API Prefix	M	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. 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.
sanValidationRequired	O	Boolean	San validation is enabled for incoming N32C handshake request or not.

Table 4-4 (Cont.) Remote SEPP Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
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.
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.
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.

3. 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 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 parameters are:

Table 4-5 Remote SEPP Set Parameters

Parameter Name	Data type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Name	String	M	Name of Remote SEPP Set
Primary SEPP	String	M	The name of Primary SEPP Configured in Set, which is treated as the primary route in the forward plane.
Secondary SEPP	String	O	The name of Secondary SEPP configured in Set, which is treated as the secondary route in the forward plane.
Tertiary SEPP	String	O	The name of Tertiary SEPP configured in Set, which is treated as the tertiary route in the forward plane.
Allowed List Name	String	M	Allowed List Name supported by Remote SEPPs Set.
CAT 2 Network ID Validation	String	M	It contains CAT 2 Network ID Validation Configuration parameters for this RSS
Network ID in Header Validation Enabled	Boolean	M	A Boolean value to enable and disable the Network ID in Header Validation feature at RSS level. true indicates Enabled false indicates Disabled
Network ID in Body Validation Enabled	Boolean	M	A Boolean value to enable and disable the Network ID in Body Validation feature at RSS level. true indicates Enabled false indicates Disabled

Table 4-5 (Cont.) Remote SEPP Set Parameters

Parameter Name	Data type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Network ID Validation List Name	String	M	Network ID Validation List Name supported by this Remote SEPP Set
Hosted SEPP	Object	M	It contains Hosted SEPP Configuration parameters for this RSS
Allowed Producer Remote SEPP Sets	String	M	List of Allowed Producer Remote SEPP Sets
Ingress Rate Limiting	Object	M	It contains Ingress Rate Limiting Configuration parameters for this RSS
RSS Ingress Rate Limiting Enabled	Boolean	O	A Boolean value to enable and disable the Ingress Rate Limiting feature at this RSS level. true indicates Enabled false indicates Disabled
Bucket Capacity	Integer	C	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	C	Refill Rate to define the number of tokens to be added to refill the bucket
Refill Duration	Integer	C	Duration to decide how frequently to refill bucket
Request Token	Integer	C	Request Token to define the Pre loaded tokens to refill the bucket
Error Configuration	Object	M	Error Configuration for Ingress Rate Limiting feature limited to this RSS
Action	String	C	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.

Table 4-5 (Cont.) Remote SEPP Set Parameters

Parameter Name	Data type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Status code	Integer	C	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	C	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.
Mediation	Object	M	It contains Mediation Configuration parameters for this RSS
Trigger Rule List Name	String		Trigger List Name supported by this Remote SEPP Set.
SoR	Object	M	It contains SOR Configuration parameters for this RSS
SoR Trigger Rule Enabled	Boolean	M	Enables or disables the SOR trigger rules.
SoR Trigger Rule List Name	Integer	M	SOR Trigger Rule List Name supported by this Remote SEPP Set
PLMN ID(s)	Integer	M	PLMN ID(s) supported by Remote SEPPs in Set. PLMN ID includes MCC and MNC
Allowed List Name	string	M	Allowed List Name supported by Remote SEPPs Set.
Trigger Rule List Name	string	M	Trigger List Name supported by Remote SEPPs Set.
Message Validation on Body Enabled	boolean	M	A Boolean value to enable or disable the Message Validation on Body at RSS level. true indicates enabled false indicates disabled

Table 4-5 (Cont.) Remote SEPP Set Parameters

Parameter Name	Data type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Message Validation On Query Parameters Enabled	boolean	M	A Boolean value to enable and disable the Message Validation on Query Parameters at RSS level. true indicates enabled false indicates disabled
Message Validation List	String	M	MessageValidation List Name supported by Remote SEPPs Set.
Allowed List Name	String	M	Allowed List Name supported by Remote SEPPs Set.
Trigger Rule List Name	String	M	Trigger List Name supported by Remote SEPPs Set.
Previous Location check Enabled	Boolean	C	A Boolean value to enable and disable the Cat 3 - Previous Location check feature at RSS Level. true indicates Enabled false indicates Disabled
Previous Location Trigger List	String	C	Previous Location Trigger List Name supported by Remote SEPPs Set.

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 4-6 Service APIs

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
Resource URIs	M	String	5G Service Based Resource URI

Table 4-6 (Cont.) Service APIs

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
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.

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 4-7 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 4-8 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 Topology Hiding

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:

1. From the left navigation menu, navigate to SEPP and then click **Topology Hiding**. Select the **Option**.
2. Click Edit icon to modify the Option. The **Edit Option** page appears.
3. Set the **Topology Hiding** to True or False.

   The parameters are:

   Table 4-9 Topology Hiding Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
Topology Hiding	O	Boolean	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 4-10 Topology Hiding Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
Enable Multiple Pseudo Value	O	Boolean	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 occurence is replaced by unique pseudo value. The value is disabled (set to false) by default.
Action	M	Enum	Action has two possible values FORWARD and REJECT. By default we have FORWARD enabled.
Status Code	M	Integer	User can configure the required HTTP error code when exceptions arise due to the TH operation failures.
Title	M	String	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 3**gpp-sbi-target-apiRoot**, then pseudo values must also contain **mnc** and **mcc**.

4. Click **Save.**

The parameters are:

Table 4-11 Pseudo Value Configuration Parameters

Attribute	Mandatory(M)/Optional(O)/Conditional(C)	Datatype	Description
actualValue	M	string	Refers to the actual FQDN of network functions.

Table 4-11 (Cont.) Pseudo Value Configuration Parameters

Attribute	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
pseudoValues	M	string	Refers to the pseudo value corresponding to a configured actual value of network functions.
value type	M	Enum	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 4-12 Header Configuration Parameters

Attribute	Mandatory(M)/ Optional(O)/ Conditional(C)	Data type	Description
Header Name	M	String	Name of the header
Regular Expression	M	String	Regular Expression for the header

Table 4-12 (Cont.) Header Configuration Parameters

Attribute	Mandatory(M)/ Optional(O)/ Conditional(C)	Data type	Description
Trigger Point	M	Enum	Request Ingress, Response Ingress, Request Egress, Response Egress
Operation	M	Enum	Topology Hiding or Topology Recovery

Table 4-13 Body IE Configuration Parameters

Attribute	Mandatory(M)/ Optional(O)/ Conditional(C)	Data stype	Description
API Resource	M	String	API Resource that comes from default table
Identifier	M	String	Body IE Key Identifier
Regular Expression	M	String	Regular Expression for the Body IE
Trigger Point	M	Enum	Request Ingress, Response Ingress, Request Egress, Response Egress
Operation	M	Enum	Topology Hiding / Topology Recovery
Method	M	Enum	GET/PUT/POST/ DELETE/PATCH

4.2.8 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.

4.2.8.1 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 4-14 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 4-15 Service API Allowed List

Parameter Name	Sub Parameter	Enabled		Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
		Listing Screen	Edit Screen			
Allowed List Name		Yes	Yes	M	String	Allowed list name per Remote SEPP Set
N32 Ingress		Yes	Yes	M	Object	Ingress Direction
	Resource URI	No	Yes	M	String	Resource URI
	HTTP Method	No	Yes	M	Enum	Resource URI Method (GET,POST, PUT,PATCH, DELETE,OPTIONS, HEAD)
N32 Egress		Yes	Yes	M	Object	Egress Direction
	Resource URI	No	Yes	M	String	Resource URI

Table 4-15   (Cont.) Service API Allowed List

Parameter Name	Sub Parameter	Enabled		Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
	HTTP Method	No	Yes	M	Enum	Resource URI Method (GET,POST, PUT,PATCH, DELETE,OPTIONS, HEAD)
N32 Ingress Action		Yes	Yes	M	Object	Ingress Action
	Title	No	Yes	M	String	Title for the Error Configuration
	Status Code	No	Yes	M	Integer	Default Value 406
	Action	No	Yes	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	No	Yes	M	String	Title for the Error Configuration
	Status Code	No	Yes	M	Integer	Default Value 406
	Action	No	Yes	M	Enum	Whenever a failure happens, request will be rejected with the user configured action. Default value is Reject

4.2.8.2 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 4-16 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 theNetwork 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 4-17 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 4-18 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 4-19 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

Table 4-19 (Cont.) Ingress Rules and Egress Rules Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
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 4-20 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
MNC Length	M	Integer	Indicates the length of mnc. It can be two or three.

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 4-21 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
MNC Length	Yes	Yes	M	Integer	Indicates the length of mnc. It can be two or three.

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 4-22 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

Table 4-22 (Cont.) Body IE Configuration

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
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)
MNC Length	M	Integer	Indicates the length of mnc. It can be two or three.

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 4-23 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)

Table 4-23 (Cont.) Add Body IE Configuration Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Data Type	Description
MNC Length	M	Integer	Indicates the length of mnc. It can be two or three.

4.2.8.3 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 4-24 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

Table 4-24 (Cont.) Cat 0 - SBI Message Schema Validation feature (Options Screen) Parameters

Parameter Name	Datatype	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
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 4-25 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 4-26 Ingress Error Action and Egress Error Action Parameters

Parameter Name	Datatype	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
Action	String	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 4-27 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 4-28 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

4.2.8.4 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 4-29 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
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 4-30 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 4-31 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 4-32 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 401and 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 4-33 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 4-34 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.
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 4-35 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.
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 4-36 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

Table 4-36 (Cont.) Serving Network ID Body IE Parameters

Parameter Name	Data Type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
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**.
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 **UE ID**, the **UE ID Body IE** details are available in the screen.
5. 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 4-37 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 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

Parameter Name	Enabled		Mandatory(M) /Optional(O)/ Conditional(C)	Datatype	Description
	Listing Screen	Edit Screen			
Enable Mediation	Yes	Yes	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	Yes	Yes	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.

4. 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 4-38 Error Configuration Parameters

Parameter Name	Enabled		Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
	Listing Screen	Edit Screen			
Title	Yes	Yes	M	String	Error Title in case of getting error from mediation service
Action	Yes	Yes	M	Enum	Error Action to be performed in case of getting error from mediation service (Reject, Continue).
StatusCode	Yes	Yes	M	Integer	Error Status Code to be returned in case of getting error from mediation service.

Mediation Trigger Rules List Parameters

Table 4-39 Mediation Trigger Rules List Parameters

Parameter Name	Enabled		Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
	Listing Screen	Edit Screen			
Name	Yes	Yes	M	String	Name of Trigger Rule List
Trigger Rules	No	Yes	M	Object	List of Trigger Rules
Trigger Rule List Type	Yes	Yes	M	Enum	Type of TRL (Local /Remote)
Match All Enabled	Yes	Yes	M	Boolean	Match All Enabled (true/ false)
Match All GroupId	No	Yes	M	String	Match All Group Id
Match All Trigger Points	No	Yes	M	Enum	Match All Trigger Points

Trigger Rule Parameters

Table 4-40 Trigger Rule Parameters

Parameter Name	Enabled		Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description
	Listing Screen	**Edit Screen**			
HTTP Method:	No	Yes	M	Enum	Resource URI Method (GET,POST,PUT,PATCH,DELETE,OPTIONS)
Resource URIs:	No	Yes	M	String	Resource URI
Trigger Points:	No	Yes	M	Enum	List of Trigger Point ("N32_Egress_Request", "N32_Ingress_Response", "N32_Ingress_Request", "N32_Egress_Response")
Group Id:	No	Yes	M	String	Group ID for which mediation configuration is to be done. This is passed to the Mediation Service for grouping similar rules.

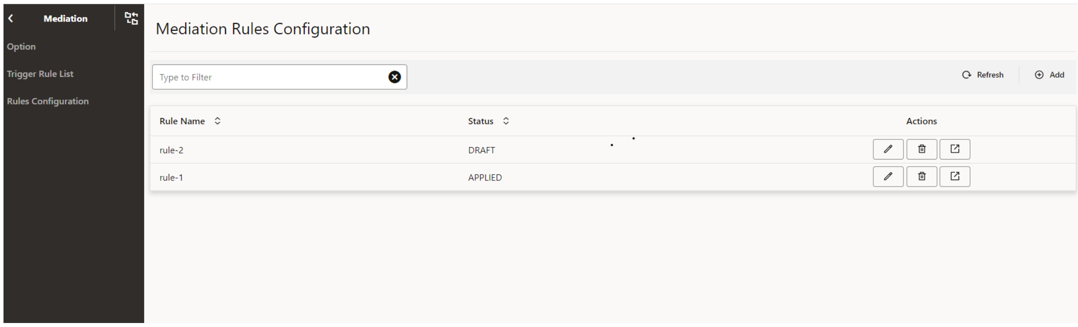
Mediation Rules Configuration

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.

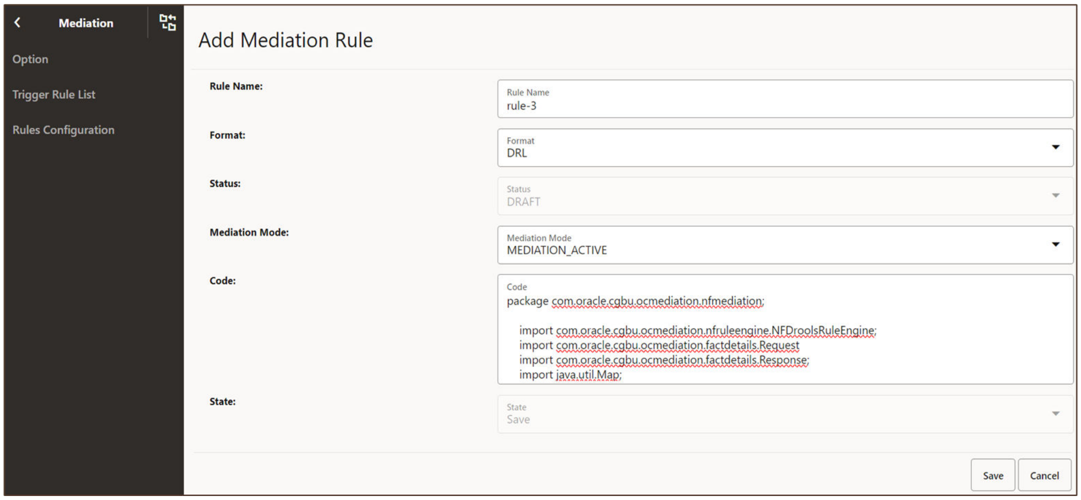
Figure 4-4 Mediation Rules Configuration Screen



Rule Name	Status	Actions
rule-2	DRAFT	
rule-1	APPLIED	

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



Rule Name: Rule Name rule-3

Format: Format DRL

Status: Status DRAFT

Mediation Mode: Mediation Mode MEDIATION_ACTIVE

Code: 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;
```

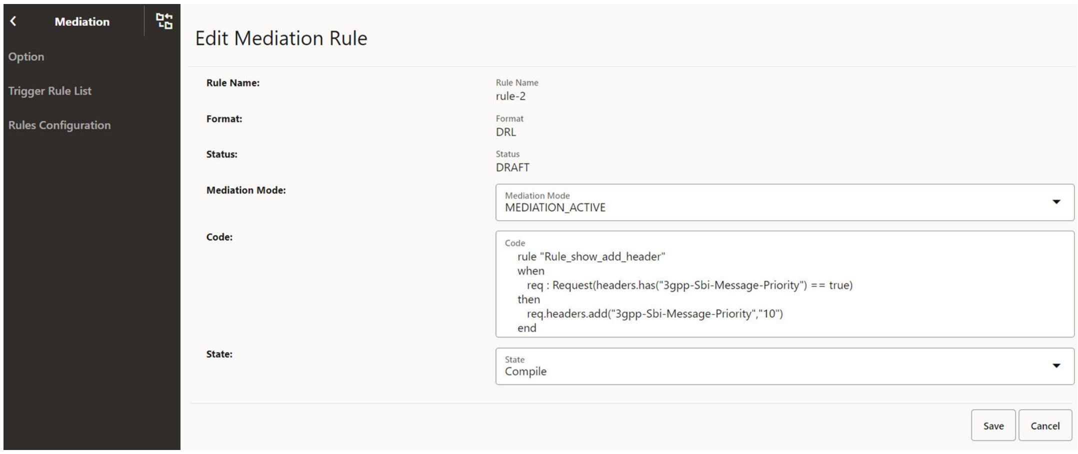
State: State Save

> **Note**
>
> **Rule Name, Format, Status, Mediation Mode, Code**, and **State** are the mediation rule parameters.

.

4. 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

Rule Name: Rule Name rule-2

Format: Format DRL

Status: Status DRAFT

Mediation Mode: Mediation Mode MEDIATION_ACTIVE

Code:

```
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: State Compile

5. 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.
6. Click **View** button of an existing rule to view the rule's details.

Rules Configuration Parameters

Table 4-41   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.

Table 4-41 (Cont.) Rules Configuration Parameters

Parameter Name	Mandatory(M)/ Optional(O)/ Conditional(C)	Datatype	Description

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.
			`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"`

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.

4.2.10 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 liming 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 4-42 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
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.

Table 4-42 (Cont.) Ingress Rate Limiting: Remote SEPP Set Parameters

Parameter Name	Datatype	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
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.

4.2.11 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 liming 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 **Engress 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 4-43 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 4-44 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.

Table 4-44 (Cont.) Parameter List

Parameter Name	Data Type	Description
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'
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.

Table 4-44 (Cont.) Parameter List

Parameter Name	Data Type	Description
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**.

4.2.12 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 4-45 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.
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.

Table 4-45 (Cont.) Options Parameters

Attribute	Data type	Mandatory(M)/ Conditional(C)/ Optional(O)	Description
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 4-46 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.

4.2.13 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. From the left navigation pane, click the **SEPP** tab, and then click the **cnDBTier** tab.
2. Click the **Backup List** to create and check the status of on-demand backups in the DB Tier.

Table 4-47 Backup List

Fields	Description
Site Name	This field displays the name of the current site to which SEPP is connected.
Backup Id	This field displays the ID of the stored backup.
Backup Size (bytes)	This field displays the size of the stored backup.
Creation TimeStamp	This field displays the time recorded when the backup was stored.

3. Click the **cnDBTier Version** to view the version.

Table 4-48 cnDBTier Version Attributes

Fields	Description
cnDBTier Version	This field displays the cnDBTier version.
NDB Version	This field displays the network database (NDB) version.

4. Click the **Database Statistics Report** to view the available database.

Table 4-49 Database Statistics Report

Fields	Description
Database Count	This field displays the number of available database.
Table Count	This field displays the table count for each database.

Table 4-49 (Cont.) Database Statistics Report

Fields	Description
Database Table Rows Count	This field displays the table rows present in each table.

5. Click the **Geo Replication Status** to view the local site and remote site name to which SEPP is connected.

Table 4-50 GeoReplication Status

Fields	Description
Local Site Name	This field displays the local site name to which SEPP is connected.
Remote Site Name	This field displays the remote site name.
Replication Status	This field displays the replication status with corresponding sites.
Seconds Behind Remote Site	This field 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.

a. Click the **View** icon in the **Actions** menu to view the **View Geo Replication Status** screen.
As part of **Replication Group Delay** details, user can view the following details:

Table 4-51 Geo Replication Status

Fields	Description
Replication Channel Group Id	This field displays the ID of the replication channel group.
Seconds Behind Remote Site	This field 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

i. Click the **View** icon to view the **Replication Group Delay** attributes:

As part of **Channel** details, user can view the following details:

Table 4-52 View Replication Group Delay

Fields	Description
Remote Replication IP	This field displays the IP of the remote replication channel.
Role	This field displays the role of the replication channel IP.

6. Click the **HeartBeat Status** to view the connectivity between local site and remote site name to which SEPP is connected.

Table 4-53 HeartBeat Status Details

Fields	Description
Remote Site Name	This field displays the remote site name.
Heartbeat Status	This field displays the connectivity status with corresponding sites.

Table 4-53 (Cont.) HeartBeat Status Details

Fields	Description
Heartbeat Lag	This field displays the lag or latency in seconds it took to syncronize between sites.
Replication channel Group Id	This field displays the ID of the replication channel group.

7. Click the **Local Cluster Status** to view the local cluster status for the current site:

Table 4-54 Local Cluster Status

Fields	Description
Site Name	This field displays the name of the current site to which SEPP is connected.
Cluster Status	This field displays the local cluster status for the current site.

8. Click the **On Demand Backup** to view the initiated on-demand backups to store data.

Table 4-55 On Demand Backup Details

Fields	Description
Site Name	This field displays the name of the current site to which SEPP is connected.
DR Status	This field displays the status of DR.
Backup Id	This field displays the ID of the stored backup.
Backup Status	This field displays the status of backup.
Remote Transfer Status	The field displays the status of remote transfer.
Initiate Backup	The field displays whether the backup is initiated or not.

5

SEPP Metrics, KPIs, and Alerts

5.1 SEPP Metrics

This section provides information about the SEPP metrics.

Table 5-1 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
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

Table 5-1 (Cont.) Dimensions

Dimensions	Details	Values
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
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

Table 5-1 (Cont.) Dimensions

Dimensions	Details	Values
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
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:

Table 5-1 (Cont.) Dimensions

Dimensions	Details	Values
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

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 5-2 cgroup_cpu_nanoseconds

Field	Details
Metric Details	Total CPU time consumed by service in nanoseconds
Microservice	Consumer N32f, Producer N32f
Type	Hologram
Dimensions	• app (Consumer, Producer) • chart • service_resource_overload_level • container • engVersion • exported_application • exported_microservice • exported_namespace • exported_pod • exported_vendor • microservice • mktgVersion • namespace • node • pod • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision • vendor "Oracle"

5.1.2.2 cgroup_memory_bytes

Table 5-3 cgroup_memory_bytes

Field	Details
Metric Details	Total memory consumed by service in bytes
Microservice	Consumer N32f, Producer N32f
Type	Gauge

Table 5-3   (Cont.) cgroup_memory_bytes

Field	Details
Dimensions	• app (Consumer, Producer) • chart • container • engVersion • exported_application • exported_microservice • exported_namespace • exported_pod • exported_vendor • microservice • mktgVersion • namespace • node • pod • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision • vendor "Oracle"

5.1.2.3 oc_configclient_request_total

Table 5-4   oc_configclient_request_total

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	• app (Plmn Ingress Gateway, N32 Ingress Gateway) • application • chart • configVersion • container • engVersion • microservice • mktgVersion • namespace • pod • releaseVersion • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision • vendor "Oracle"

5.1.2.4 oc_configclient_response_total

Table 5-5 oc_configclient_response_total

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	• app (Plmn Ingress Gateway, N32 Ingress Gateway) • application • chart • configVersion • container • engVersion • microservice • mktgVersion • namespace • pod • releaseVersion • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision • vendor "Oracle"

5.1.2.5 oc_configserver_reachability

Table 5-6 oc_configserver_reachability

Metric Details	Gauge metric to peg the reachability of config server
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Gauge
Dimensions	• app (Plmn Ingress Gateway, N32 Ingress Gateway) • application • chart • container • endpoint • engVersion • microservice • mktgVersion • namespace • pod • releaseVersion • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision • vendor "Oracle"

5.1.2.6 service_resource_overload_level

Table 5-7   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

5.1.3 CN32F Common Metrics

5.1.3.1 ocsepp_cn32f_jetty_request_stat_metrics_total

Table 5-8   ocsepp_cn32f_jetty_request_stat_metrics_total

Metric Details	This metric will be pegged for every event occurred when a request is sent to CN32F
Microservice	Consumer N32f, Producer 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 5-9   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 5-9 (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 5-10 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 5-11 ocsepp_cn32f_requests_failure_total

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 5-11 (Cont.) ocsepp_cn32f_requests_failure_total

Dimensions	• app(consumer) • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • vendor "Oracle"

5.1.3.5 ocsepp_cn32f_response_failure_total

Table 5-12 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
Dimensions	• app(consumer) • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • vendor "Oracle"

5.1.3.6 ocsepp_cn32f_requests_total

Table 5-13 ocsepp_cn32f_requests_total

Metric Details	Total number of requests received from NF. Condition:When a request is received on InboundInterface of cn32f.
Microservice	Consumer N32f
Type	Counter

Table 5-13 (Cont.) ocsepp_cn32f_requests_total

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) • vendor "Oracle"

5.1.3.7 ocsepp_cn32f_response_total

Table 5-14 ocsepp_cn32f_response_total

Field	Details
Metric Details	Total number of response received from remote SEPP. Condition: When a response is received on 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) • http_status • vendor "Oracle"

5.1.3.8 ocsepp_cn32f_latency_seconds_count

Table 5-15 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
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 "Oracle"

5.1.3.9 ocsepp_cn32f_latency_seconds_max

Table 5-16 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 • vendor "Oracle"

5.1.3.10 ocsepp_cn32f_latency_seconds_sum

Table 5-17 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 "Oracle"

5.1.3.11 ocsepp_cn32f_outgoing_connections

Table 5-18 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 5-19 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

5.1.4 PN32F Common Metrics

5.1.4.1 ocsepp_pn32f_requests_total

Table 5-20 ocsepp_pn32f_requests_total

Metric Details	Number of requests received from Peer SEPP. Condition: When a request reaches pn32f from peer SEPP.
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 "Oracle"

5.1.4.2 ocsepp_pn32f_requests_failure_total

Table 5-21 ocsepp_pn32f_requests_failure_total

Metric Details	Number of requests transmitted to NF. Condition: When a request transmits a message to a NF.
Microservice	Producer N32f
Type	Counter

Table 5-21 (Cont.) ocsepp_pn32f_requests_failure_total

Dimensions	• app(producer) • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • vendor

5.1.4.3 ocsepp_pn32f_responses_total

Table 5-22 ocsepp_pn32f_responses_total

Metric Details	Number of responses received from Egress Gateway. Condition: When a response reaches pn32f from Egress Gateway.
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 (201 CREATED, 404 NOT_FOUND) • vendor "Oracle"

5.1.4.4 ocsepp_pn32f_responses_failure_total

Table 5-23 ocsepp_pn32f_responses_failure_total

Metric Details	Number of responses transmitted to Consumer SEPP (cSEPP). Condition: When a response transmits a message to cSEPP.
Microservice	Producer N32f
Type	Counter

Table 5-23 (Cont.) ocsepp_pn32f_responses_failure_total

Dimensions	• app (producer) • chart • container • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • http_status • vendor

5.1.4.5 ocsepp_pn32f_latency_seconds_count

Table 5-24 ocsepp_pn32f_latency_seconds_count

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 • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • nf_type • vendor

5.1.4.6 ocsepp_pn32f_latency_seconds_sum

Table 5-25 ocsepp_pn32f_latency_seconds_sum

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 5-25 (Cont.) ocsepp_pn32f_latency_seconds_sum

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

5.1.4.7 ocsepp_pn32f_latency_seconds_max

Table 5-26 ocsepp_pn32f_latency_seconds_max

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 • nfInstanceId • peer_fqdn • peer_domain • plmn_id • direction • message_type • namespace • remote_sepp_name • targetUrl • app • container • pod • release

5.1.4.8 ocsepp_pn32f_connection_failure_total

Table 5-27 ocsepp_pn32f_connection_failure_total

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, Producer N32f
Type	Counter

Table 5-27 (Cont.) ocsepp_pn32f_connection_failure_total

Dimensions	• host • port • direction • instanceIdentifier • errorReason • errorOriginator

5.1.4.9 ocsepp_pn32f_jetty_request_stat_metrics_total

Table 5-28 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 5-29 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
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 5-30 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 5-31 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 5-32 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.
Microservice	Producer and Consumer N32c
Type	Counter

Table 5-32 (Cont.) ocsepp_n32c_handshake_failure_attempts_total

Field	Details
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 5-33 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 5-34 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

Table 5-34 (Cont.) ocsepp_cn32c_handshake_requests_total

Field	Details
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 5-35 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 5-36 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.

Table 5-36 (Cont.) ocsepp_cn32c_handshake_initiation_req_total

Field	Details
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 5-37 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 5-38 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

Table 5-38 (Cont.) ocsepp_cn32c_handshake_delete_req_total

Field	Details
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 5-39 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 5-40 ocsepp_pn32c_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.

Table 5-40 (Cont.) ocsepp_pn32c_handshake_response_total

Field	Details
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

5.1.6 5G SBI Message Mediation Support Metrics

5.1.6.1 ocsepp_cn32f_mediation_requests_total

Table 5-41 ocsepp_cn32f_mediation_requests_total

Metric Details	Metric is common for both CN32F & 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	• peer_fqdn • peer_domain • plmn_id • statusCode • direction • method • NfType • NfServiceType • vendor • nfInstanceId • ruleApplied • requestType

5.1.6.2 ocsepp_n32f_mediation_not_applied_total

Table 5-42 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
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 "Oracle"

5.1.6.3 ocsepp_cn32f_mediation_response_total

Table 5-43 ocsepp_cn32f_mediation_response_total

Metric Details	Number of requests in which CN32F service of SEPP get Response from Mediation Service.
Microservice	Consumer N32f
Type	Counter

Table 5-43 (Cont.) ocsepp_cn32f_mediation_response_total

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 "Oracle"

5.1.6.4 ocsepp_cn32f_mediation_response_failure

Table 5-44 ocsepp_cn32f_mediation_response_failure

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

5.1.6.5 ocsepp_pn32f_mediation_response_total

Table 5-45 ocsepp_pn32f_mediation_response_total

Metric Details	Number of requests in which PN32F service of SEPP get Response from Mediation Service.
Microservice	Producer N32f

Table 5-45 (Cont.) ocsepp_pn32f_mediation_response_total

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 "Oracle"

5.1.6.6 ocsepp_pn32f_mediation_response_failure

Table 5-46 ocsepp_pn32f_mediation_response_failure

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 "Oracle"

5.1.6.7 ocsepp_cn32f_mediation_latency_seconds_count

Table 5-47 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
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 "Oracle"

5.1.6.8 ocsepp_cn32f_mediation_latency_seconds_max

Table 5-48 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 "Oracle"

5.1.6.9 ocsepp_cn32f_mediation_latency_seconds_sum

Table 5-49 ocsepp_cn32f_mediation_latency_seconds_sum

Metric Details	Total time taken for processing a message (from sending a message to receiving the response).
Microservice	Producer 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 "Oracle"

5.1.6.10 ocsepp_pn32f_mediation_latency_seconds_count

Table 5-50 ocsepp_pn32f_mediation_latency_seconds_count

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

5.1.6.11 ocsepp_pn32f_mediation_latency_seconds_max

Table 5-51 ocsepp_pn32f_mediation_latency_seconds_max

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

5.1.6.12 ocsepp_pn32f_mediation_latency_seconds_sum

Table 5-52 ocsepp_pn32f_mediation_latency_seconds_sum

Metric Details	This metrics is used to display the average of 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

5.1.6.13 ocsepp_pn32f_mediation_latency_seconds_sum

Table 5-53 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
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

5.1.6.14 ocsepp_n32f_mediation_requests_total

Table 5-54 ocsepp_n32f_mediation_requests_total

Field	Details
Metric Details	Number of requests in which Trigger Rule Applied at SEPP end for Mediation, based on configuration. Separation happens based on "app" tag.
Microservice	Consumer N32F, Producer N32F
Type	Counter
Dimensions	• app (Consumer, Producer) • chart • container • direction • http_method • namespace • nf_instance_id • NfServiceType • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • sourceRss • http_status • vendor "Oracle"

5.1.7 Hosted SEPP Metrics

5.1.7.1 ocsepp_allowed_p_rss_routing_failure_total

Table 5-55 ocsepp_allowed_p_rss_routing_failure_total

Metric Details	Number of requests failing due to Hosted SEPP failure.
Microservice	Consumer N32F
Type	Counter
Dimensions	• app (Consumer, Producer) • chart • container • http_error_message • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • sourceRss • http_status • vendor "Oracle"

5.1.8 Message Copy Metrics

5.1.8.1 oc_ingressgateway_msgcopy_requests_total

Table 5-56 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

5.1.8.2 oc_ingressgateway_msgcopy_responses_total

Table 5-57 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

5.1.8.3 oc_ingressgateway_dd_unreachable

Table 5-58 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

5.1.8.4 oc_egressgateway_msgcopy_requests_total

Table 5-59 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

5.1.8.5 oc_egressgateway_msgcopy_responses_total

Table 5-60 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

5.1.8.6 oc_egressgateway_dd_unreachable

Table 5-61 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

5.1.9 SOR Metrics

5.1.9.1 ocsepp_pn32f_sor_requests_total

Table 5-62 ocsepp_pn32f_sor_requests_total

Metric Details	Number of requests sent to SOR
Microservice	Producer N32f
Type	Counter

Table 5-62 (Cont.) ocsepp_pn32f_sor_requests_total

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 "Oracle"

5.1.9.2 ocsepp_pn32f_sor_responses_total

Table 5-63 ocsepp_pn32f_sor_responses_total

Metric Details	Number of 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 • vendor "Oracle"

5.1.9.3 ocsepp_pn32f_sor_retry_to_producer_requests_total

Table 5-64 ocsepp_pn32f_sor_retry_to_producer_requests_total

Metric Details	Number of requests sent to Producer based on 3gpp header
Microservice	Producer N32f
Type	Counter

Table 5-64 (Cont.) ocsepp_pn32f_sor_retry_to_producer_requests_total

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 • vendor "Oracle"

5.1.9.4 ocsepp_pn32f_sor_back_to_consumer_responses_total

Table 5-65 ocsepp_pn32f_sor_back_to_consumer_responses_total

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 • vendor "Oracle"

5.1.9.5 ocsepp_pn32f_sor_failure_total

Table 5-66 ocsepp_pn32f_sor_failure_total

Metric Details	Number of 4xx or 5xx responses received from SOR
Microservice	Producer N32f
Type	Counter

Table 5-66 (Cont.) ocsepp_pn32f_sor_failure_total

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 • vendor "Oracle"

5.1.9.6 ocsepp_pn32f_sor_timeout_failure_total

Table 5-67 ocsepp_pn32f_sor_timeout_failure_total

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 • vendor "Oracle"

5.1.10 Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set Metrics

5.1.10.1 oc_ingressgateway_rss_ratelimit_total

Table 5-68 oc_ingressgateway_rss_ratelimit_total

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	• app (PLMN Ingress Gateway, N32 Ingress Gateway) • chart • container • ErrorOriginator • nf_instance_id • http_method • namespace • peer_plmn_id • pod • remote_sepp_set_name • Scheme • Status
Metric filter	Status = accepted

5.1.10.2 oc_ingressgateway_rss_ratelimit_total

Table 5-69 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

5.1.10.3 ocsepp_configmgr_routefailure_total

Table 5-70 ocsepp_configmgr_routefailure_total

Metric Details	Metric pegged due to route sync issue in SEPP.
Microservice	Config Manager
Type	Counter

Table 5-70 (Cont.) ocsepp_configmgr_routefailure_total

Dimensions	• errorCode • app

5.1.10.4 oc_ingressgateway_rss_ratelimit_total

Table 5-71 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	• app (PLMN Ingress Gateway, N32 Ingress Gateway) • chart • container • ErrorOriginator • nf_instance_id • http_method • namespace • peer_plmn_id • pod • remote_sepp_set_name • Scheme • Status
Metric filter	Status = dropped

5.1.11 Topology Hiding Metrics

5.1.11.1 ocsepp_topology_latency_seconds_count

Table 5-72 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

Table 5-72 (Cont.) ocsepp_topology_latency_seconds_count

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_type • vendor "Oracle"

5.1.11.2 ocsepp_topology_latency_seconds_max

Table 5-73 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
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 • vendor "Oracle"

5.1.11.3 ocsepp_topology_latency_seconds_sum

Table 5-74 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 • vendor "Oracle"

5.1.11.4 ocsepp_topology_header_success_total

Table 5-75 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 "Oracle"

5.1.11.5 ocsepp_topology_header_failure_total

Table 5-76 ocsepp_topology_header_failure_total

Field	Details
Metric Details	Count of headers for which topology hiding and recovery failed

Table 5-76 (Cont.) ocsepp_topology_header_failure_total

Field	Details
Microservice	Consumer N32f, Producer N32f
Type	Counter
Dimensions	• app (consumer, producer) • chart • container • header • namespace • nf_instance_id • peer_domain • peer_fqdn • peer_plmn_id • pod • remote_sepp_name • vendor "Oracle"

5.1.11.6 ocsepp_topology_body_success_total

Table 5-77 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 "Oracle"

5.1.11.7 ocsepp_topology_body_failure_total

Table 5-78 ocsepp_topology_body_failure_total

Metric Details	Count of body for which topology hiding and recovery failed
Microservice	Consumer N32f, Producer N32f
Type	Counter

Table 5-78 (Cont.) ocsepp_topology_body_failure_total

Dimensions	- request_path - app (Consumer, Producer) - chart - container - http_method - namespace - nf_instance_id - peer_domain - peer_fqdn - peer_plmn_id - pod - http_status - vendor "Oracle"

5.1.11.8 ocsepp_topology_success_total

Table 5-79 ocsepp_topology_success_total

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 - sourceRss - vendor "Oracle"

5.1.11.9 ocsepp_topology_invalid_header_regex_configured_total

Table 5-80 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 - nfInstanceId - error_message

5.1.11.10 ocsepp_topology_header_regex_not_configured_total

Table 5-81 ocsepp_topology_header_regex_not_configured_total

Field	Details
Metric Details	If header regex pattern is not configured, this metric will be pegged.
Microservice	Consumer N32f, Producer N32f
Type	Counter
Dimensions	• vendor • nfInstanceId • error_message

5.1.12 Cat 0 - SBI Message Schema Validation Metrics

5.1.12.1 ocsepp_message_validation_applied_total

Table 5-82 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 "Oracle"

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.

.

5.1.12.2 ocsepp_message_validation_on_body_failure_total

Table 5-83 ocsepp_message_validation_on_body_failure_total

Metric Details	Number of requests in which message validation failed on body at SEPP end.
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 "Oracle" • status_code

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.

.

5.1.12.3 ocsepp_message_validation_on_header_failure_total

Table 5-84 ocsepp_message_validation_on_header_failure_total

Metric Details	Number of requests for which message validation failed on query parameters at SEPP end.
Microservice	Producer N32f, Consumer N32F
Type	Counter

Table 5-84 (Cont.) ocsepp_message_validation_on_header_failure_total

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 "Oracle" • status_code

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 Cat 1 - Service API Validation Metrics

5.1.13.1 ocsepp_security_service_api_failure_total

Table 5-85 ocsepp_security_service_api_failure_total

Field	Details
Metric Details	Metric are 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

Table 5-85 (Cont.) ocsepp_security_service_api_failure_total

Field	Details
Dimensions	- app (Consumer, Producer) - chart - container - http_method - namespace - nf_instance_id - peer_fqdn - peer_plmn_id - pod - remote_sepp_name - request_path - http_status - vendor "Oracle" - nf_service_type - nf_type

5.1.14 Cat 2 - Network ID Validation Metrics

5.1.14.1 ocsepp_network_id_validation_body_failure_total

Table 5-86 ocsepp_network_id_validation_body_failure_total

Metric Details	Number of request for which Network ID body 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 - request_path - sepp_type - vendor "Oracle"

5.1.14.2 ocsepp_network_id_validation_header_failure_total

Table 5-87 ocsepp_network_id_validation_header_failure_total

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 • request_path • sepp_type • vendor "Oracle"

5.1.15 Cat 3 - Previous Location Check Metrics

5.1.15.1 ocsepp_previous_location_exception_failure_total

Table 5-88 ocsepp_previous_location_exception_failure_total

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 • request_path • vendor "Oracle"

5.1.15.2 ocsepp_previous_location_validation_success_total

Table 5-89 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 • vendor "Oracle"

5.1.15.3 ocsepp_previous_location_validation_failure_total

Table 5-90 ocsepp_previous_location_validation_failure_total

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 • request_path • vendor "Oracle"

5.1.15.4 ocsepp_previous_location_validation_requests_total

Table 5-91 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 • vendor "Oracle"

5.1.15.5 ocsepp_pn32f_notification_total

Table 5-92 ocsepp_pn32f_notification_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	NA

5.1.16 Rate Limiting for Egress Roaming Signaling per PLMN Metrics

5.1.16.1 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-93 oc_ingressgateway_plmn_egress_ratelimit_total

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

Table 5-93 (Cont.) oc_ingressgateway_plmn_egress_ratelimit_total

Dimensions	- Status - PLMN - Scheme - Egress Rate Limit List - InstanceIdentifier - ErrorOriginator
Status	ERL_MATCH_TOKEN_AVAILABLE_FWD

5.1.16.2 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-94 oc_ingressgateway_plmn_egress_ratelimit_total

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 - Scheme - Egress Rate Limit List - InstanceIdentifier - ErrorOriginator
Status	ERL_MATCH_NO_TOKEN_LOW_PRI_REJECT

5.1.16.3 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-95 oc_ingressgateway_plmn_egress_ratelimit_total

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 - Scheme - Egress Rate Limit List - InstanceIdentifier - ErrorOriginator
Status	ERL_MATCH_NO_TOKEN_HIGH_PRI_FWD

5.1.16.4 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-96 oc_ingressgateway_plmn_egress_ratelimit_total

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.
Microservice	Ingress Gateway
Type	Counter
Dimensions	• Status • PLMN • Scheme • Egress Rate Limit List • InstanceIdentifier • ErrorOriginator
Status	ERROR_UNABLE_TO_EXTRACT_PLMN_FWD

5.1.16.5 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-97 oc_ingressgateway_plmn_egress_ratelimit_total

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 • Scheme • Egress Rate Limit List • InstanceIdentifier • ErrorOriginator
Status	ERL_NO_MATCH_FWD

5.1.16.6 oc_ingressgateway_plmn_egress_ratelimit_total

Table 5-98 oc_ingressgateway_plmn_egress_ratelimit_total

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 5-98 (Cont.) oc_ingressgateway_plmn_egress_ratelimit_total

Dimensions	- Status - PLMN - Scheme - Egress Rate Limit List - InstanceIdentifier - ErrorOriginator
Status	ERROR_INTERNAL_FWD

5.1.17 Config Manager Metrics

5.1.17.1 ocsepp_configmgr_rpp_config_failure_total

Table 5-99 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

5.1.17.2 ocsepp_configmgr_rpp_validation_failure_total

Table 5-100 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
Dimensions	- app - http_error_message - http_status - vendor - chart - pod - namespace - container

5.1.17.3 ocsepp_configmgr_routeupdate_total

Table 5-101 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 • vendor • chart • pod • namespace • container

5.1.18 Support for TLS 1.3 Metrics

5.1.18.1 oc_ingressgateway_incoming_tls_connections

Table 5-102 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

5.1.18.2 oc_egressgateway_outgoing_tls_connections

Table 5-103 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 5-103 (Cont.) oc_egressgateway_outgoing_tls_connections

Field	Details
Dimensions	• NegotiatedTLSVersion • Host • Direction • InstanceIdentifier

5.1.18.3 security_cert_x509_expiration_seconds

Table 5-104 security_cert_x509_expiration_seconds

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 • endpoint • container • namespace • pod • serialNumber • subject • CN (CommonName) • O (Organization) • L (Locality) • S (State or ProvinceName) • C (CountryName)

5.1.19 Ingress and Egress Gateway Metrics

Table 5-105 Ingress and Egress Gateway Dimensions

Dimension	Details
NFType	Name of the NF Type in path. For Eg: Path is /n**xxx-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 /n**xxx-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)	Servivce profile Id of the backend svc

Table 5-105 (Cont.) Ingress and Egress Gateway Dimensions

Dimension	Details
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
Scheme	Http protocol scheme
Path	Path predicate that matched the current request
ClientCertIdentity	Cerificate 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

Table 5-105 (Cont.) Ingress and Egress Gateway Dimensions

Dimension	Details
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
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.,

5.1.19.1 Ingress Gateway Metrics

This section provides information about the Ingress Gateway metrics used in SEPP.

5.1.19.1.1 oc_ingressgateway_http_requests_total

Table 5-106 oc_ingressgateway_http_requests_total

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
Dimensions	• Method • NFType • NFServiceType • Host • HttpVersion • Scheme • Route_path • InstanceIdentifier • ClientCertIdentity app

5.1.19.1.2 oc_ingressgateway_http_responses_total

Table 5-107 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

5.1.19.1.3 oc_ingressgateway_request_latency_seconds

Table 5-108 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.
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Histogram
Dimensions	• quantile • InstanceIdentifier • Route_path • Method

5.1.19.1.4 oc_ingressgateway_request_latency_seconds_count

Table 5-109 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

5.1.19.1.5 oc_ingressgateway_request_latency_seconds_sum

Table 5-110 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

5.1.19.1.6 oc_configclient_request_total

Table 5-111 oc_configclient_request_total

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	• app (Plmn Ingress Gateway, N32 Ingress Gateway) • application • chart • configVersion • container • engVersion • microservice • mktgVersion • namespace • pod • releaseVersion • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision • vendor "Oracle"

5.1.19.1.7 oc_configclient_response_total

Table 5-112 oc_configclient_response_total

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	• app (Plmn Ingress Gateway, N32 Ingress Gateway) • application • chart • configVersion • container • engVersion • microservice • mktgVersion • namespace • pod • releaseVersion • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision • vendor "Oracle"

5.1.19.1.8 oc_configserver_reachability

Table 5-113 oc_configserver_reachability

Metric Details	Gauge metric to peg the reachability of config server
Microservice	Plmn Ingress Gateway, N32 Ingress Gateway
Type	Gauge
Dimensions	• app (Plmn Ingress Gateway, N32 Ingress Gateway) • application • chart • container • endpoint • engVersion • microservice • mktgVersion • namespace • pod • releaseVersion • security_istio_io_tlsMode • service_istio_io_canonical_name • service_istio_io_canonical_revision • vendor "Oracle"

5.1.19.1.9 oc_ingressgateway_incoming_connections

Table 5-114 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

5.1.19.1.10 oc_ingressgateway_outgoing_connections

Table 5-115 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
Dimensions	• host • direction • instanceIdentifier

5.1.19.1.11 oc_ingressgateway_connection_failure_total

Table 5-116 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

5.1.19.1.12 oc_ingressgateway_global_ratelimit_total

Table 5-117 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

5.1.19.1.13 oc_ingressgateway_request_content_metrics_total

Table 5-118 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 • InstanceIden tifier

5.1.19.1.14 oc_ingressgateway_request_processing_latency_seconds

Table 5-119 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

5.1.19.1.15 oc_ingressgateway_route_overloadcontrol_total

Table 5-120 oc_ingressgateway_route_overloadcontrol_total

Metric Details	When overload is enabled , this metric is pegged for every incoming request and describes whether request is accepted or discarded.
Microservice	N32 Ingress Gateway
Type	Counter
Dimensions	• namespace • app • DiscardAction • Status

5.1.19.2 Egress Gateway Metrics

This section provides information about the Egress Gateway metrics used in SEPP.

5.1.19.2.1 oc_egressgateway_http_requests_total

Table 5-121 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 egressOut.
Microservice	Plmn Egress GatewayN32 Egress Gateway
Type	Counter

Table 5-121 (Cont.) oc_egressgateway_http_requests_total

Field	Details
Dimensions	- Method - NFType - NFServiceType - Host - HttpVersion - Scheme - Proxy - InstanceIdentifier - Direction

5.1.19.2.2 oc_egressgateway_http_responses_total

Table 5-122 oc_egressgateway_http_responses_total

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 GatewayN32 Egress Gateway
Type	Counter
Dimensions	- Status - Method - NFType - NFServiceType - Host - HttpVersion - Scheme - Proxy - InstanceIdentifier - Direction - BlacklistedFqdn

5.1.19.2.3 oc_egressgateway_incoming_connections

Table 5-123 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

5.1.19.2.4 oc_egressgateway_outgoing_connections

Table 5-124 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

5.1.19.2.5 oc_egressgateway_connection_failure_total

Table 5-125 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
Dimensions	• host • port • direction • instanceIdentifier • errorReason • errorOriginator

5.1.19.2.6 oc_egressgateway_sbiRouting_http_requests_total

Table 5-126 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 EGW.
Microservice	Plmn Egress GatewayN32 Egress Gateway
Type	Counter
Dimensions	• Sbi_Fqdn • Reroute_Path • Response_Code (This would be populated as blank for requests) • Attempt • HttpVersion • Scheme • InstanceIdentifier

5.1.19.2.7 oc_egressgateway_sbiRouting_http_responses_total

Table 5-127 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
Dimensions	• Sbi_Fqdn • Reroute_Path • Status • Attempt • HttpVersion • Scheme • InstanceIdentifier • ErrorOriginator

5.1.19.2.8 oc_egressgateway_server_latency_seconds

Table 5-128 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

5.1.19.2.9 oc_fqdn_alternate_route_total

Table 5-129 oc_fqdn_alternate_route_total

Field	Details
Metric Details	Tracks the number of registration, deregistration and GET calls received for a given scheme and FQDN. **Note:** Registration does not reflect active registration numbers. It captured number of registration requests received.
Microservice	Egress Gateway
Type	Counter
Dimensions	type: Register/Deregister/GET binding_value: <scheme>+<FQDN>

5.1.19.2.10 oc_dns_srv_lookup_total

Table 5-130 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	Egress Gateway
Type	Counter
Dimensions	binding_value: <scheme>+<FQDN>

5.1.19.2.11 oc_alternate_route_resultset

Table 5-131 oc_alternate_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	Egress Gateway
Type	Gauge
Dimensions	binding_value: <scheme>+<FQDN>

5.1.19.2.12 oc_configclient_request_total

Table 5-132 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	Egress 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.

5.1.19.2.13 oc_configclient_response_total

Table 5-133 oc_configclient_response_total

Field	Details
Metric Details	This metric is pegged whenever a response is received from the server to client.

Table 5-133 (Cont.) oc_configclient_response_total

Field	Details
Microservice	Egress Gateway
Type	Counter
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.

5.2 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 5-134 cn32c Handshake Requests Per Remote SEPP

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 5-135 cn32c Handshake Success Rate

KPI Detail	Measures the cn32c handshake success rate.
Metric Used for KPI	(sum(ocsepp_cn32c_handshake_response_total{namespace=~"$Namespace",responseCode="200 OK"})/sum(ocsepp_cn32c_handshake_requests_total{namespace=~"$Namespace"}))*100
Service Operation	n32c handshake success rate
Response Code	200 OK

5.2.1.3 cn32c Handshake Response Per Remote SEPP

Table 5-136 cn32c Handshake Response Per Remote SEPP

KPI Detail	Measures the cn32c handshake response per remote SEPP.

Table 5-136 (Cont.) 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 5-137 cn32c Handshake Failure Per Remote SEPP

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 5-138 pn32c Handshake Requests Total Per Remote SEPP

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 5-139 pn32c Handshake Response Total Per Remote SEPP

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 5-140 pn32c Handshake Success rate

KPI Detail	Measures the pn32c handshake success rate.

Table 5-140 (Cont.) pn32c Handshake Success rate

Metric Used for KPI	(sum(ocsepp_pn32c_handshake_response_total{namespace=~"$Namespace",responseCode="200 OK"})/sum(ocsepp_pn32c_handshake_requests_total{namespace=~"$Namespace"}))*100
Service Operation	n32c Handshake
Response Code	200

5.2.1.8 pn32c Handshake Failure Per Remote SEPP

Table 5-141 pn32c Handshake Failure Per Remote SEPP

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 5-142 Memory Usage per POD

KPI Detail	Measures the memory usage per POD
Metric Used for KPI	sum(container_memory_usage_bytes{namespace=~"$Namespace",image!=""}/(1024*1024*1024)) by (pod)
Service Operation	NA
Response Code	NA

5.2.2.2 CPU Usage per POD

Table 5-143 CPU Usage per POD

KPI Detail	Measures the CPU usage per POD
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 5-144   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 5-145   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 5-146   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 5-147   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 5-148   IGW Processing Time (ms)

Field	Details
KPI Detail	Measures the IGW Processing Time
Metric Used for KPI	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)

5.2.2.8 PercentageDiscard

Table 5-149 PercentageDiscard

KPI Detail	Measures the number of Discard requests for Percentage based scheme
Metric Used for KPI	oc_ingressgateway_route_overloadcontrol_total sum(irate(oc_ingressgateway_route_overloadcontrol_total{DiscardAction="PercentageBased",Status="DISCARDED",namespace=$NAMESPACE}[2m]))

5.2.2.9 PriorityDiscard

Table 5-150 PriorityDiscard

KPI Detail	Measures the number of Discard requests for Priority based scheme
Metric Used for KPI	oc_ingressgateway_route_overloadcontrol_total sum(irate(oc_ingressgateway_route_overloadcontrol_total{DiscardAction="PriorityBased",Status="DISCARDED",namespace=NAMESPACE}[2m]))

.

5.2.3 CN32F Common KPIs

5.2.3.1 cn32f Routing Success Rate

Table 5-151 cn32f Routing Success Rate

KPI Detail	Measures the cn32f routing success rate.
Metric Used for KPI	(sum(ocsepp_cn32f_response_total{namespace=~"$Namespace"})/ sum(ocsepp_cn32f_requests_total{namespace=~"$Namespace"}))*100
Service Operation	n32f message forward
Response Code	All

5.2.3.2 Total cn32f Requests

Table 5-152 Total cn32f Requests

KPI Detail	Measures the cn32f requests rate per remote SEPP.
Metric Used for KPI	sum((ocsepp_cn32f_requests_total{namespace=~"$Namespace", direction="egress"}))by(PEER_DOMAIN, PEER_FQDN, PLMN_ID)
Service Operation	n32f message forward
Response Code	All

5.2.3.3 cn32f Processing Time (ms)

Table 5-153 cn32f Processing Time (ms)

KPI Detail	Measures the cn32f processing time (ms)
Metric Used for KPI	sum(irate(ocsepp_cn32f_latency_seconds_sum{namespace=~"$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name) / sum(irate(ocsepp_cn32f_latency_seconds_count{namespace=~"$Namespace"}[2m])) by(peer_fqdn,remote_sepp_name)
Service Operation	n32f message forward
Response Code	All

5.2.3.4 Total cn32f Responses

Table 5-154 Total cn32f Responses

KPI Detail	Measures the cn32f response rate per remote SEPP
Metric Used for KPI	sum((ocsepp_cn32f_response_total{namespace=~"$Namespace", direction="egress"})) by(PEER_DOMAIN, PEER_FQDN, PLMN_ID)
Service Operation	n32f message forward
Response Code	All

5.2.3.5 cn32f Failures

Table 5-155 cn32f Failures

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 5-156 Total pn32f Requests

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 5-157 Total pn32f Responses

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 5-158 pn32f Processing Time (ms)

KPI Detail	Measures the pn32f processing time in milli seconds
Metric Used for KPI	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)
Service Operation	n32f message forward
Response Code	All

5.2.4.4 pn32f Failures

Table 5-159 pn32f Failures

KPI Detail	Measures the pn32f request failures in total
Metric Used for KPI	sum(ocsepp_pn32f_requests_failure_total{namespace=~"$Namespace"}) by (PEER_DOMAIN, PEER_FQDN, PLMN_ID)
Service Operation	n32f message forward
Response Code	4xx and 5xx

5.2.4.5 pn32f Routing Success Rate

Table 5-160 pn32f Routing Success Rate

KPI Detail	Measures the pn32f routing success rate
Metric Used for KPI	(sum(ocsepp_pn32f_responses_total{namespace=~"$Namespace"})/sum(ocsepp_pn32f_requests_total{namespace=~"$Namespace"}))*100
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 5-161 PLMN IGW Global Rate limit Traffic Rejected

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

5.2.5.2 N32 IGW Global Rate limit Traffic Rejected

Table 5-162 N32 IGW Global Rate limit Traffic Rejected

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 5-163 CN32F Topology Egress Request Processing Time(ms)

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 5-164 CN32F Topology Ingress Response Processing Time

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)

Table 5-164 (Cont.) CN32F Topology Ingress Response Processing Time

Service Operation	n32f message forward
Response Code	NA

5.2.6.3 CN32F Topology Hiding Success

Table 5-165 CN32F Topology Hiding Success

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 5-166 N32F N32F Topology Success by headers

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 5-167 CN32F Topology Hiding Missing Regex Configuration

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 5-168 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 5-169 PN32F Topology Ingress Request Processing Time(ms)

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 5-170 PN32F Topology Egress Response Processing Time(ms)

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 5-171 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 5-172 PN32F Topology Hiding Missing Regex Configuration

Field	Details
KPI Detail	Measures the pn32f topology hiding missing Regex configuration
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

Table 5-172 (Cont.) PN32F Topology Hiding Missing Regex Configuration

Field	Details
Response Code	NA

5.2.6.11 PN32F Topology Hiding Success

Table 5-173 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 5-174 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 5-175 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 5-176 Mediation Requests Counters - N32 Ingress Request

Field	Details
KPI Detail	Measures the Mediation Requests Counters for N32 Ingress Request

Table 5-176 (Cont.) Mediation Requests Counters - N32 Ingress Request

Field	Details
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 5-177 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 5-178 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 5-179 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 5-180 Mediation Response Counters - N32 Ingress Request

Field	Details
KPI Detail	Measures the Mediation Response Counters for N32 EIngress Request
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 5-181 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 5-182 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 5-183 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 5-184 Mediation Response Time At PN32F

Field	Details
KPI Detail	Measures the Mediation Response Time at PN32F
Metric Used for KPI	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)

5.2.7.12 Mediation Response Time At CN32F

Table 5-185 Mediation Response Time At CN32F

Field	Details
KPI Detail	Measures the Mediation Response Time at CN32F
Metric Used for KPI	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)

5.2.8 Ingress Gateway Message Copy KPIs

5.2.8.1 Total Requests Data sent towards DD for Ingress Gateway

Table 5-186 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	sum(irate(oc_egressgateway_msgcopy_requests_total{namespace="$Namespace",type="req"}[2m])) by(app)

5.2.8.2 Total Ack received from DD for Requests for Ingress Gateway

Table 5-187 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	sum(irate(oc_egressgateway_msgcopy_requests_total{namespace="$Namespace",type="ack"}[2m])) by(app)

5.2.9 Egress Gateway Message Copy KPIs

5.2.9.1 Total Requests Data sent towards DD for Egress Gateway

Table 5-188 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	sum(irate(oc_egressgateway_msgcopy_requests_total{namespace="$Namespace",type="req"}[2m])) by(app)

5.2.9.2 Total Ack received from DD for Requests for Egress Gateway

Table 5-189 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	sum(irate(oc_egressgateway_msgcopy_requests_total{namespace="$Namespace",type="ack"}[2m])) by(app)

5.2.10 Hosted SEPP KPIs

5.2.10.1 CN32F Allowed P-RSS Validation Failure Count

Table 5-190 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	sum(ocsepp_allowed_p_rss_routing_failure_total{app="cn32f-svc", namespace=~"$Namespace"}) by (app)

5.2.10.2 PN32F Allowed P-RSS Validation Failure Count

Table 5-191 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	sum(ocsepp_allowed_p_rss_routing_failure_total{app="pn32f-svc", namespace=~"$Namespace"}) by (app)

5.2.11 SoR KPIs

5.2.11.1 Pn32f to SoR Request count total

Table 5-192 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 5-193 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 5-194 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 5-195 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 5-196 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 5-197 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 5-198 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 5-199 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 5-200 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 5-201 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 5-202 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 5-203 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 5-204 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 5-205 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 5-206 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 5-207 Average Number of Messages Accepted for a Particular PLMN

Field	Details
KPI Detail	Measures the average number of messages accepted 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_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 5-208 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_ratelimit_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 5-209 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_ratelimit_total{namespace=“namespace”}[2m])) by (Status)

5.2.14.5 Average Number of Messages Rejected per PLMN

Table 5-210 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_ratelimit_total{namespace="namespace", Status="ERL_MATCH_NO_TOKEN_LOW_PRI_REJECT"}[2m])) by (PLMN_ID)

5.2.14.6 Average Number of Messages Accepted per PLMN

Table 5-211 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_FWD|ERL_MATCH_NO_TOKEN_HIGH_PRI_FWD"}[2m])) by (PLMN_ID)

5.3 SEPP Alerts

This section provides information about the SEPP alerts and their configuration.

Note

For CNE1.8.4 or earlier versions:

- namespace: {{$labels.kubernetes_namespace}}
- podname: {{$labels.kubernetes_pod_name}}

For CNE 1.9.x or later versions:

- namespace: {{$labels.namespace}}
- podname: {{$labels.pod}}

5.3.1 System Level Alerts

5.3.1.1 SEPPPodMemoryUsageAlert

Table 5-212 SEPPPodMemoryUsageAlert

<table>
<tr><td>Trigger Condition</td><td>Pod memory usage is above the threshold (70%)</td></tr>
<tr><td>Severity</td><td>Warning</td></tr>
<tr><td>Alert details provided</td><td><b>Summary</b>
<code>'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}:</code>
<code>Memory usage is {{ $value | printf "%.2f" }} which is above 70% (current value is: {{ $value }})'</code>
<b>Expression:</b>
<code>(sum by(namespace,container)(container_memory_usage_bytes{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.*"}) / sum by(namespace,container)(container_spec_memory_limit_bytes{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.*"})) * 100 >= 70</code></td></tr>
<tr><td>OID</td><td>1.3.6.1.4.1.323.5.3.46.1.2.4003</td></tr>
<tr><td>Metric Used</td><td>container_memory_usage_bytes
<b>Note:</b> 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.</td></tr>
</table>

Table 5-212 (Cont.) SEPPPodMemoryUsageAlert

Resolution	The alert gets cleared when the memory utilization falls below the critical threshold. **Note:** The threshold is configurable in the *SeppAlertrules.yaml* file. If guidance is required, contact My Oracle Support.

5.3.1.2 SEPPPodCpuUsageAlert

Table 5-213 SEPPPodCpuUsageAlert

Field	Details
Trigger Condition	Pod CPU usage is above the threshold (70%)
Severity	Warning

Alert details provided	**Summary**
	`'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 }})'`
	Expression:
	`(sum by (namespace,container) (rate(container_cpu_usage_seconds_total{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.*"}[2m]))) / (sum by (container, namespace) (kube_pod_container_resource_limits{resource="cpu",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.*"})) * 100 >= 70`

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 *SeppAlertrules.yaml* file. If guidance is required, contact My Oracle Support.

5.3.2 Application Level Alerts

5.3.2.1 Common Alerts

5.3.2.1.1 SEPPN32fRoutingFailure

Table 5-214   SEPPN32fRoutingFailure

Field	Details
Trigger Condition	N32f service not able to forward message
Severity	Info

Alert details provided	**Summary**
	`namespace: {{ $labels.namespace}}, podname:`
	`{{ $labels.pod}}, timestamp: {{ with query`
	`      "time()" }}{{ . | first | value | humanizeTimestamp }}`
	`{{ end }}`
	Expression:
	`idelta(ocsepp_cn32f_requests_failure_total[2m]) > 0 or`
	`(ocsepp_cn32f_requests_failure_total`
	`      unless ocsepp_cn32f_requests_failure_total offset 2m)`

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 5-215   SEPPConfigMgrRouteFailureAlert

Trigger Condition	When routing failure occurs while posting remote SEPP or roaming partner set, this alert will be raised.
Severity	Major

Table 5-215 (Cont.) SEPPConfigMgrRouteFailureAlert

Alert Details Provided	**Summary**
	`namespace: {{$labels.kubernetes_namespace}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Route Failure has occurred because {{ $labels.errorReason }}`
	Expression
	`sum(increase(ocsepp_configmgr_routefailure_total{app="config-mgr-svc"}[5m]) >0 or (ocsepp_configmgr_routefailure_total{app="config-mgr-svc"} unless ocsepp_configmgr_routefailure_total{app="config-mgr-svc"} offset 5m )) by (namespace,errorCode) > 0`

OID	1.3.6.1.4.1.323.5.3.46.1.2.4026
Metric Name	Metric ocsepp_configmgr_routefailure_total
Resolution	The alert is cleared if no new failures are observed in 5 minutes window.

5.3.2.1.3 EgressSbiErrorRateAbove1Percent

Table 5-216 EgressSbiErrorRateAbove1Percent

Trigger Condition	Sbi Transaction Error Rate exceeded configured threshold
Severity	Major

Alert details provided	**Summary**
	`"Sbi Transaction Error Rate detected above 1 Percent of Total Sbi Transactions"`
	Expression
	`sum(rate(oc_egressgateway_sbiRouting_http_responses_total{Status!~"2.*"}[24h])) by (app,pod, namespace) / sum(rate(oc_egressgateway_sbiRouting_http_responses_total[24h])) by (app,pod, namespace) *100 >= 1`

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 24 hour time period. Metric will be cleared when the error rate will be below 1%

5.3.2.2 Handshake Alerts

5.3.2.2.1 SEPPCn32cHandshakeFailureAlert

Table 5-217 SEPPCn32cHandshakeFailureAlert

Trigger Condition	Handshake procedure has failed on Consumer SEPP
Severity	Major

Alert details provided	**Summary**
	`'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}:`
	` Handshake procedure has failed on Consumer side because {{ $labels.reason }}'`
	Expression:
	`sum(increase(ocsepp_n32c_handshake_failure_attempts_total{app="cn32c-svc"}[5m])`
	`    >0 or (ocsepp_n32c_handshake_failure_attempts_total{app="cn32c-svc"}  unless`
	`    ocsepp_n32c_handshake_failure_attempts_total{app="cn32c-svc"}  offset 5m )) by`
	`    (namespace,remote_sepp_name,nfinstanceid,peer_fqdn,app) > 0`

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=cn32-svc

Table 5-217 (Cont.) SEPPCn32cHandshakeFailureAlert

<table>
<tr>
<td>Resolution 1</td>
<td>

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 *404 –route not found* or *Route not found*, follow the recovery steps:

1. 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
   ```

2. Run the following command to navigate to the pod:

   `$ kubectl exec –it <config-mgr-pod name> –n <namespace> bash`

   Example:

   `$ kubectl exec -it ocsepp-release-config-mgr-svc-79c95d4b9d-8stk7 -n csepp bash`

</td>
</tr>
</table>

Table 5-217 (Cont.) SEPPCn32cHandshakeFailureAlert

	3. Run the command to get the existing route details present on N32 Egress Gateway:
	`curl -X GET http://<config-manager-service-name>:9090/sepp/nf-common-component/v1/egw/n32/routesconfiguration`
	Example:
	`curl -X GET http://ocsepp-release-config-mgr-svc:9090/sepp/nf-common-component/v1/egw/n32/routesconfiguration`
	4. If this output is **null**, add the configuration details in `config-mgr-svc` deployment.
	For more information about the configuration details, see the *Deployment Configuration for Config-mgr-svc* section in *Oracle Communications Cloud Native Core Security Edge Protection Proxy Installation Guide.*
	5. 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.
	6. 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: 1. Disable the Remote SEPP. 2. Delete the Remote SEPP. 3. Update and reinitiate Handshake.

5.3.2.2.2 SEPPPn32cHandshakeFailureAlert

Table 5-218 SEPPPn32cHandshakeFailureAlert

Trigger Condition	Handshake procedure has failed on Producer sepp
Severity	Major

Table 5-218 (Cont.) SEPPPn32cHandshakeFailureAlert

Alert details provided	**Summary**
	`'namespace: {{$labels.namespace}}, podname: {{$labels.pod}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}:`
	`Handshake procedure has failed on Producer side because {{ $labels.error_msg }}'`
	Expression:
	`sum(increase(ocsepp_n32c_handshake_failure_attempts_total{app="pn32c-svc"}[5m])`
	`    >0 or`
	`(ocsepp_n32c_handshake_failure_attempts_total{app="pn32c-svc"}  unless`
	`    ocsepp_n32c_handshake_failure_attempts_total{app="pn32c-svc"}  offset 5m )) by`
	`    (namespace,remote_sepp_name,nfinstanceid,peer_fqdn,app)`
	`> 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=pn32-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 5-219 SEPPUpgradeStartedAlert

Trigger Condition	Rest API trigger at start of Upgrade
Severity	NA
Alert details provided	applicationname alertname servicename releasename namespace oid severity vendor sourcerelease targetrelease
OID	1.3.6.1.4.1.323.5.3.46.1.2.8001
Metric Used	NA

Table 5-219 (Cont.) SEPPUpgradeStartedAlert

Resolution	If a success alert is generated then start and failure alerts will be cleared.

5.3.2.3.2 SEPPUpgradeFailedAlert

Table 5-220 SEPPUpgradeFailedAlert

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
Resolution	If a success alert is generated then start and failure alerts will be cleared.

5.3.2.3.3 SEPPUpgradeSuccessfulAlert

Table 5-221 SEPPUpgradeSuccessfulAlert

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	If a success alert is generated then start and failure alerts will be cleared.

5.3.2.4 Rollback Alerts

5.3.2.4.1 SEPPRollbackStartedAlert

Table 5-222 SEPPRollbackStartedAlert

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 5-223 SEPPRollbackFailedAlert

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
Resolution	If a success alert is generated then start and failure alerts will be cleared.

5.3.2.4.3 SEPPRollbackSuccessfulAlert

Table 5-224 SEPPRollbackSuccessfulAlert

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 5-225 IngressGlobalMessageDropAbovePointOnePercent

Trigger Condition	Ingress Global Message Drop Rate detected greater than or equal to 0.1 Percent of Total Transactions.
Severity	Warning

Alert details provided	**Summary**
	`"Ingress Global Message Drop Rate detected above 0.1 Percent of Total Transactions"`
	Expression
	`sum(rate(oc_ingressgateway_global_ratelimit_total{Status="dropped"}[5m])) by`
	`      (namespace)/`
	`sum(rate(oc_ingressgateway_global_ratelimit_total[5m])) by (namespace) *100 >= 0.1 <`
	`      1`

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%.

5.3.2.5.2 IngressGlobalMessageDropAbove1Percent

Table 5-226   IngressGlobalMessageDropAbove1Percent

Trigger Condition	Ingress Global Message Drop Rate detected greater than or equal to 1 Percent of Total Transactions.
Severity	Warning

Alert details provided	**Summary**
	`"Ingress Global Message Drop Rate detected above 1 Percent of Total Transactions"`
	Expression
	`sum(rate(oc_ingressgateway_global_ratelimit_total{Status="dropped"}[5m])) by`
	`      (namespace)/`
	`sum(rate(oc_ingressgateway_global_ratelimit_total[5m])) by`
	`(namespace) *100 >= 1 <`
	`      10`

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%.

5.3.2.5.3 IngressGlobalMessageDropAbove10Percent

Table 5-227   IngressGlobalMessageDropAbove10Percent

Trigger Condition	Ingress Global Message Drop Rate detected greater than or equal to 10 Percent of Total Transactions
Severity	Minor

Alert details provided	**Summary**
	`"Ingress Global Message Drop Rate detected above 10 Percent of Total Transactions"`
	Expression
	`sum(rate(oc_ingressgateway_global_ratelimit_total{Status="dropped"}[5m])) by`
	`      (namespace)/`
	`sum(rate(oc_ingressgateway_global_ratelimit_total[5m])) by`
	`(namespace) *100 >= 10 <`
	`      25`

OID	1.3.6.1.4.1.323.5.3.46.1.2.7004
Metric Used	oc_ingressgateway_global_ratelimit_total

Table 5-227 (Cont.) IngressGlobalMessageDropAbove10Percent

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% .

5.3.2.5.4 IngressGlobalMessageDropAbove25Percent

Table 5-228 IngressGlobalMessageDropAbove25Percent

Trigger Condition	Ingress Global Message Drop Rate detected greater than or equal to 25 Percent of Total Transactions
Severity	Major

Alert details provided	**Summary**
	`"Ingress Global Message Drop Rate detected above 25 Percent of Total Transactions"`
	Expression
	`sum(rate(oc_ingressgateway_global_ratelimit_total{Status="dropped"}[5m])) by`
	`      (namespace)/`
	`sum(rate(oc_ingressgateway_global_ratelimit_total[5m])) by (namespace) *100 >= 25 <`
	`      50`

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%.

5.3.2.5.5 IngressGlobalMessageDropAbove50Percent

Table 5-229 IngressGlobalMessageDropAbove50Percent

Trigger Condition	Ingress Global Message Drop Rate detected greater than or equal to 50 Percent of Total Transactions
Severity	Critical

Table 5-229 (Cont.) IngressGlobalMessageDropAbove50Percent

Alert details provided	**Summary**
	`"Ingress Global Message Drop Rate detected above 50 Percent of Total Transactions"`
	Expression
	`sum(rate(oc_ingressgateway_global_ratelimit_total{Status="dropped"}[5m])) by (namespace)/ sum(rate(oc_ingressgateway_global_ratelimit_total[5m])) by (namespace) *100 >= 50`

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%.

5.3.2.6 Topology Hiding Alerts

5.3.2.6.1 SEPPN32fTopologyOperationFailureAlert

Table 5-230 SEPPN32fTopologyOperationFailureAlert

Field	Details
Trigger Condition	Topology Hiding or Recovery Failure exceeded configured threshold (1%)
Severity	Major

Alert details provided	**Summary**
	`"Topology hiding/recovery operation failres reached more than configured threshold"`
	Expression
	`delta(ocsepp_topology_header_failure_total[2m])>0 or (ocsepp_topology_header_failure_total unless ocsepp_topology_header_failure_total offset 2m)`

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

Table 5-230 (Cont.) SEPPN32fTopologyOperationFailureAlert

Field	Details
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 5-231 SEPPN32fTopologyBodyOperationFailureAlert

Field	Details
Trigger Condition	Topology Operation failed and exceeds defined threshold
Severity	Major

Alert details provided	**Summary**
	`"Topology Hiding/Recovery Operation failures reached more than configured`
	`    threshold"`
	Expression:
	`delta(ocsepp_topology_body_failure_total[2m])>0 or (ocsepp_topology_body_failure_total unless ocsepp_topology_body_failure_total 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 5-232 SEPPCN32fMediationFailure

Trigger Condition	Mediation processing Failure
Severity	Info

Alert details provided	**Summary**
	`"Mediation processing Failure"`
	Expression:
	`increase(ocsepp_cn32f_mediation_response_failure{status_code!`
	`="504`
	`      GATEWAY_TIMEOUT"}[10m]) > 0`

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 5-233 SEPPCN32fMediationUnreachable

Trigger Condition	Mediation service is not accessible
Severity	Critical

Alert details provided	**Summary**
	`"Mediation service is not accessible"`
	Expression:
	`increase(ocsepp_cn32f_mediation_response_failure`
	`{status_code="504`
	`      GATEWAY_TIMEOUT"}[10m]) > 0`

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 5-234 SEPPPN32fMediationFailure

Trigger Condition	Mediation processing Failure
Severity	Info

Alert details provided	**Summary**
	`"Mediation processing Failure"`
	Expression:
	`increase(ocsepp_pn32f_mediation_response_failure {status_code!="504 GATEWAY_TIMEOUT"}[10m]) > 0`

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 5-235 SEPPPN32fMediationUnreachable

Trigger Condition	Mediation service is not accessible
Severity	Critical

Alert details provided	**Summary**
	`"Mediation service is not accessible"`
	Expression:
	`increase(ocsepp_pn32f_mediation_response_failure {status_code="504 GATEWAY_TIMEOUT"}[10m]) > 0`

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 SEPPServiceOverload65Percent

Table 5-236 SEPPServiceOverload65Percent

Trigger Condition	CPU memory of pn32f-svc more than 65%
Severity	Warning

Alert details provided	**Summary**
	`Backend service is in overload with load level > 65%`
	Expression
	`service_resource_overload_level == 1`

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 60%.

5.3.2.8.2 SEPPServiceOverload70Percent

Table 5-237 SEPPServiceOverload70Percent

Trigger Condition	CPU memory of pn32f-svc more than 70%
Severity	Minor

Alert details provided	**Summary**
	`Backend service is in overload with load level > 70%`
	Expression
	`service_resource_overload_level == 2`

OID	1.3.6.1.4.1.323.5.3.46.1.2.7008
Metric Used	service_resource_overload_level
Resolution	The alert will be cleared when CPU Memory for backend-svc goes below 70%

5.3.2.8.3 SEPPServiceOverload80Percent

Table 5-238 SEPPServiceOverload80Percent

Trigger Condition	CPU memory of pn32f-svc more than 80%
Severity	Major

Table 5-238 (Cont.) SEPPServiceOverload80Percent

Alert details provided	**Summary**
	`Backend service is in overload with load level > 80%`
	Expression
	`service_resource_overload_level == 3`

OID	1.3.6.1.4.1.323.5.3.46.1.2.7009
Metric Used	service_resource_overload_level
Resolution	The alert will be cleared when CPU Memory for backend-svc goes below 80%

5.3.2.8.4 SEPPServiceOverload90Percent

Table 5-239 SEPPServiceOverload90Percent

Trigger Condition	CPU memory of pn32f-svc more than 90%
Severity	Critical

Alert details provided	**Summary**
	`Backend service is in overload with load level > 90%`
	Expression
	`service_resource_overload_level == 4`

OID	1.3.6.1.4.1.323.5.3.46.1.2.7010
Metric Used	service_resource_overload_level
Resolution	The alert will be cleared when CPU Memory for backend-svc goes below 90%

5.3.2.9 Hosted SEPP Alerts

5.3.2.9.1 SEPPPn32fHSRoutingFailureAlert

Table 5-240 SEPPPn32fHSRoutingFailureAlert

Trigger Condition	When the routing failure rate at Pn32f service is greater than 20 percentage.
Severity	Major
Alert details provided	Allowed P-RSS Validation failure at Roaming Hub **Expression** ((sum by(namespace, app, nfInstanceId, pod) (ocsepp_allowed_p_rss_routing_failure_total)) / (sum by(namespace, app, nfInstanceId, pod) (ocsepp_pn32f_requests_total))) > 0.2
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 5-240 (Cont.) SEPPPn32fHSRoutingFailureAlert

Resolution	The alert gets automatically cleared when the failure rate at pn32f microservice goes below 20 percent.

5.3.2.9.2 SEPPCn32fHSRoutingFailureAlertMinor

Table 5-241 SEPPCn32fHSRoutingFailureAlertMinor

Field	Details
Trigger Condition	When the routing failure rate at Cn32f service is greater than 50 percentage.
Severity	Minor
Alert details provided	Allowed P-RSS Validation failure at Roaming Hub for Consumer SEPP. **Expression** ((sum by(namespace, app, nfInstanceId, pod, sourceRss) (ocsepp_allowed_p_rss_routing_failure_total)) / (sum by(namespace, app, nfInstanceId, pod, sourceRss) (ocsepp_cn32f_requests_total))) > 0.5
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.

5.3.2.9.3 SEPPCn32fHSRoutingFailureAlertMajor

Table 5-242 SEPPCn32fHSRoutingFailureAlertMajor

Field	Details
Trigger Condition	When the routing failure rate at Cn32f service is greater than 60 percentage.
Severity	Major
Alert details provided	Allowed P-RSS Validation failure at Roaming Hub for Consumer SEPP. **Expression** ((sum by(namespace, app, nfInstanceId, pod, sourceRss) (ocsepp_allowed_p_rss_routing_failure_total)) / (sum by(namespace, app, nfInstanceId, pod, sourceRss) (ocsepp_cn32f_requests_total))) > 0.6
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.

5.3.2.9.4 SEPPCn32fHSRoutingFailureAlertCritical

Table 5-243 SEPCn32fHSRoutingFailureAlertCritical

Field	Details
Trigger Condition	When the routing failure rate at Cn32f service is greater than 65 percentage.
Severity	Critical
Alert details provided	Allowed P-RSS Validation failure at Roaming Hub for Consumer SEPP. **Expression** ((sum by(namespace, app, nfInstanceId, pod, sourceRss) (ocsepp_allowed_p_rss_routing_failure_total)) / (sum by(namespace, app, nfInstanceId, pod, sourceRss) (ocsepp_cn32f_requests_total))) > 0.65
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.

5.3.2.9.5 SEPPCn32fHSRoutingFailureAlertWarning

Table 5-244 SEPCn32fHSRoutingFailureAlertWarning

Field	Details
Trigger Condition	When the routing failure rate at Cn32f service is greater than 25 percentage.
Severity	Warning
Alert details provided	Allowed P-RSS Validation failure at Roaming Hub for Consumer SEPP. **Expression** ((sum by(namespace, app, nfInstanceId, pod, sourceRss) (ocsepp_allowed_p_rss_routing_failure_total)) / (sum by(namespace, app, nfInstanceId, pod, sourceRss) (ocsepp_cn32f_requests_total))) > 0.25
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.

5.3.2.10 SEPP Message Feed Alerts

5.3.2.10.1 DDUnreachableFromN32IGW

Table 5-245 DDUnreachableFromN32IGW

Trigger Condition	This alarm is raised when Data Director is not reachable from N32 Ingress Gateway.
Severity	major

Table 5-245 (Cont.) DDUnreachableFromN32IGW

Alert details provided	**Summary** (oc_ingressgateway_dd_unreachable{app="n32-ingress-gateway"} == 1)
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.

5.3.2.10.2 DDUnreachableFromPLMNIGW

Table 5-246 DDUnreachableFromPLMNIGW

Trigger Condition	This alarm is raised when Data Director is not reachable from PLMN Ingress Gateway.
Severity	major
Alert details provided	**Summary** (oc_ingressgateway_dd_unreachable{app="n32-ingress-gateway"} == 1)
OID	1.3.6.1.4.1.323.5.3.46.1.2.4019
Metric Used	oc_ingressgateway_dd_unreachable
Resolution	Alert gets cleared automatically when the connection with Data Director is established.

5.3.2.10.3 DDUnreachableFromN32EGW

Table 5-247 DDUnreachableFromN32EGW

Trigger Condition	This alarm is raised when Data Director is not reachable from N32 Egress Gateway.
Severity	major
Alert details provided	**Summary** (oc_egressgateway_dd_unreachable{app="n32-egress-gateway"} == 1)
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.

5.3.2.10.4 DDUnreachableFromPLMNEGW

Table 5-248 DDUnreachableFromPLMNEGW

Trigger Condition	This alarm is raised when Data Director is not reachable from PLMN Egress Gateway.
Severity	major
Alert details provided	**Summary** (oc_egressgateway_dd_unreachable{app="plmn-egress-gateway"} == 1)
OID	1.3.6.1.4.1.323.5.3.46.1.2.4021
Metric Used	oc_egressgateway_dd_unreachable

Table 5-248 (Cont.) DDUnreachableFromPLMNEGW

Resolution	Alert gets cleared automatically when the connection with Data Director is established.

5.3.2.11 Steering of Roaming (SOR) Alerts

5.3.2.11.1 SEPPPn32fSORFailureAlertPercent30to40

Table 5-249 SEPPPn32fSORFailureAlertPercent30to40

Field	Details
Trigger Condition	30% to 40% of SOR traffic results in failure.
Severity	Minor
Alert details provided	Summary: 'namespace: {{$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}' Expression: sum(rate(ocsepp_pn32f_sor_failure_total[2m]))by(namespace,nf_instance_id,app)/ sum(rate(ocsepp_pn32f_sor_requests_total[2m]))by(namespace,nf_instance_id,app)>=0.3 and sum(rate(ocsepp_pn32f_sor_failure_total[2m]))by(namespace,nf_instance_id,app)/ sum(rate(ocsepp_pn32f_sor_requests_total[2m]))by(namespace,nf_instance_id,app)<0.4
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 metricocsepp_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 metricocsepp_pn32f_sor_failure_total.

5.3.2.11.2 SEPPPn32fSORFailureAlertPercent40to50

Table 5-250 SEPPPn32fSORFailureAlertPercent40to50

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: sum(rate(ocsepp_pn32f_sor_failure_total[2m]))by(namespace,nf_instance_id,app)/ sum(rate(ocsepp_pn32f_sor_requests_total[2m]))by(namespace,nf_instance_id,app)>=0.4 and sum(rate(ocsepp_pn32f_sor_failure_total[2m]))by(namespace,nf_instance_id,app)/ sum(rate(ocsepp_pn32f_sor_requests_total[2m]))by(namespace,nf_instance_id,app)<0.5
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 SEPPPn32fSORFailureAlertPercentAbove50

Table 5-251 SEPPPn32fSORFailureAlertPercentAbove50

Field	Details
Trigger Condition	50% of SOR traffic results in failure
Severity	Critical

Table 5-251 (Cont.) SEPPPn32fSORFailureAlertPercentAbove50

Field	Details
Alert details provided	Summary: 'namespace: {{$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}' Expression: sum(rate(ocsepp_pn32f_sor_failure_total[2m]))by(namespace,nf_instance_id,app)/ sum(rate(ocsepp_pn32f_sor_requests_total[2m]))by(namespace,nf_instance_id,app)>=0.5
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 SEPPPn32fSORTimeoutFailureAlert

Table 5-252 SEPPPn32fSORTimeoutFailureAlert

Field	Details
Trigger Condition	Increase of more than five timeout errors in last two minutes for SOR.
Severity	critical

Table 5-252 (Cont.) SEPPPn32fSORTimeoutFailureAlert

Field	Details
Alert details provided	Summary: 'namespace: {{$labels.namespace}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}' Expression: idelta(ocsepp_pn32f_sor_timeout_failure_total[2m]) > 5 or (ocsepp_pn32f_sor_timeout_failure_total unless ocsepp_pn32f_sor_timeout_failure_total offset 2m)
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 Global Rate Limiting on Ingress Gateway of SEPP Alerts

5.3.2.12.1 Ingress RSS Rate Limit per RSS Message Drop Above Point one Percent Alert

Table 5-253 Ingress RSS Rate Limit per RSS Message Drop Above Point one Percent Alert

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:** sum(rate(oc_ingressgateway_rss_ratelimit_total{Status="dropped"}[5m])) by (Remote_SEPP_Set,namespace)/ sum(rate(oc_ingressgateway_rss_ratelimit_total[5m])) by (Remote_SEPP_Set,namespace) *100 >= 0.1 < 10
OID	1.3.6.1.4.1.323.5.3.46.1.2.7011
Metric Name	oc_ingressgateway_rss_ratelimit_total

Table 5-253 (Cont.) Ingress RSS Rate Limit per RSS Message Drop Above Point one Percent Alert

Resolution	The alerts gets cleared when the count goes down.

5.3.2.12.2 Ingress RSS Rate Limit per RSS Message Drop Above 10 Percent Alert

Table 5-254 Ingress RSS Rate Limit per RSS Message Drop Above 10 Percent Alert

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.
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:** sum(rate(oc_ingressgateway_rss_ratelimit_total{Status="dropped"}[5m])) by (Remote_SEPP_Set,namespace)/ sum(rate(oc_ingressgateway_rss_ratelimit_total[5m])) by (Remote_SEPP_Set,namespace) *100 >= 10 < 25
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 count goes down.

5.3.2.12.3 Ingress RSS Rate Limit per RSS Message Drop Above 25 Percent Alert

Table 5-255 Ingress RSS Rate Limit per RSS Message Drop Above 25 Percent Alert:

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
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:** sum(rate(oc_ingressgateway_rss_ratelimit_total{Status="dropped"}[5m])) by (Remote_SEPP_Set,namespace)/ sum(rate(oc_ingressgateway_rss_ratelimit_total[5m])) by (Remote_SEPP_Set,namespace) *100 >= 25 < 50
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 count goes down.

5.3.2.12.4 Ingress RSS Rate Limit per RSS Message Drop Above 50 Percent Alert

Table 5-256 Ingress RSS Rate Limit per RSS Message Drop Above 50 Percent Alert

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
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:** sum(rate(oc_ingressgateway_rss_ratelimit_total{Status="dropped"}[5m])) by (Remote_SEPP_Set,namespace)/ sum(rate(oc_ingressgateway_rss_ratelimit_total[5m])) by (Remote_SEPP_Set,namespace) *100 >= 50
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 count goes down.

5.3.2.12.5 Ingress RSS Rate Limit Message Drop Above Point one Percent Alert

Table 5-257 Ingress RSS Rate Limit Message Drop Above Point one Percent Alert

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
Alert Details Provided	**Summary:** Ingress RSS Based Rate Limiting Message Drop Rate detected above 0.1 Percent of Total Transaction **Expression:** sum(rate(oc_ingressgateway_rss_ratelimit_total{Status="dropped"}[5m])) by (namespace)/ sum(rate(oc_ingressgateway_rss_ratelimit_total[5m])) by (namespace) *100 >= 0.1 < 1
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 count goes down.

5.3.2.12.6 Ingress RSS Rate Limit Message Drop Above one Percent Alert

Table 5-258 Ingress RSS Rate Limit Message Drop Above one Percent Alert:

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 5-258 (Cont.) Ingress RSS Rate Limit Message Drop Above one Percent Alert:

Alert Details Provided	**Summary:** Ingress RSS Based Rate Limiting Message Drop Rate detected above 1 Percent of Total Transactions **Expression:** sum(rate(oc_ingressgateway_rss_ratelimit_total{Status="dropped"}[5m])) by (namespace)/ sum(rate(oc_ingressgateway_rss_ratelimit_total[5m])) by (namespace) *100 >= 1 < 10
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 count goes down.

5.3.2.12.7 Ingress RSS Rate Limit Message Drop Above 10 Percent Alert

Table 5-259 Ingress RSS Rate Limit Message Drop Above 10 Percent Alert

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
Alert Details Provided	**Summary:** Ingress RSS Based Rate Limiting Message Drop Rate detected above 10 Percent of Total Transactions. **Expression:** sum(rate(oc_ingressgateway_rss_ratelimit_total{Status="dropped"}[5m])) by (Remote_SEPP_Set,namespace)/ sum(rate(oc_ingressgateway_rss_ratelimit_total[5m])) by (Remote_SEPP_Set,namespace) *100 >= 10 < 25
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 count goes down.

5.3.2.12.8 Ingress RSS Rate Limit Message Drop Above 25 Percent Alert

Table 5-260 Ingress RSS Rate Limit Message Drop Above 25 Percent Alert

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
Alert Details Provided	**Summary:** Ingress RSS Based Rate Limiting Message Drop Rate detected above 25 Percent of Total Transactions **Expression:** sum(rate(oc_ingressgateway_rss_ratelimit_total{Status="dropped"}[5m])) by (Remote_SEPP_Set,namespace)/ sum(rate(oc_ingressgateway_rss_ratelimit_total[5m])) by (Remote_SEPP_Set,namespace) *100 >= 25 < 50

Table 5-260 (Cont.) Ingress RSS Rate Limit Message Drop Above 25 Percent Alert

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 count goes down.

5.3.2.12.9 Ingress RSS Rate Limit Message Drop Above 50 Percent Alert

Table 5-261 Ingress RSS Rate Limit Message Drop Above 50 Percent Alert

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 andcorresponding alert will be raised.
Severity	Critical
Alert Details Provided	**Summary:** Ingress RSS Based Rate Limiting Message Drop Rate detected above 50 Percent of Total Transactions **Expression:** sum(rate(oc_ingressgateway_rss_ratelimit_total{Status="dropped"}[5m])) by (Remote_SEPP_Set,namespace)/ sum(rate(oc_ingressgateway_rss_ratelimit_total[5m])) by (Remote_SEPP_Set,namespace) *100 >= 50
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 count goes down.

5.3.2.13 Cat-0 SBI Message Schema Validation Alerts

5.3.2.13.1 SEPPN32fMessageValidationOnHeaderFailureMinorAlert

Table 5-262 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
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[2m])) by (app, pod, namespace, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total[2m])) by (app, pod, namespace, nf_instance_id))*100 >= 40 < 60

Table 5-262 (Cont.) SEPPN32fMessageValidationOnHeaderFailureMinorAlert

Field	Details
OID	1.3.6.1.4.1.323.5.3.46.1.2.4026
Metric Used	ocsepp_message_validation_on_header_failure_total
Resolution	The alerts gets cleared when the count is not between 40 to 60.

5.3.2.13.2 SEPPN32fMessageValidationOnHeaderFailureMajorAlert

Table 5-263 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
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[2m])) by (app, pod, namespace, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total[2m])) by (app, pod, namespace, nf_instance_id))*100 >= 60 < 80
OID	1.3.6.1.4.1.323.5.3.46.1.2.4027
Metric Name	ocsepp_message_validation_on_header_failure_total

Table 5-263 (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 *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Troubleshooting Guide*. 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 5-264 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 5-264 (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[2m])) by (app, pod, namespace, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total[2m])) by (app, pod, namespace, 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
Resolution	The alerts gets cleared when the count is not between 80 to 100.

5.3.2.13.4 SEPPN32fMessageValidationOnBodyFailureMinorAlert

Table 5-265 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
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[2m])) by (app, pod, namespace, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total[2m])) by (app, pod, namespace, 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
Resolution	The alerts gets cleared when the count is not between 60 to 100.

5.3.2.13.5 SEPPN32fMessageValidationOnBodyFailureMajorAlert

Table 5-266 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
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[2m])) by (app, pod, namespace, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total[2m])) by (app, pod, namespace, nf_instance_id))*100 >= 60 < 80
OID	1.3.6.1.4.1.323.5.3.46.1.2.4030
Metric Name	ocsepp_message_validation_on_body_failure_total
Resolution	The alerts gets cleared when the count is not between 80 to 100.

5.3.2.13.6 SEPPN32fMessageValidationOnBodyFailureCriticalAlert

Table 5-267 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
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[2m])) by (app, pod, namespace, nf_instance_id) / sum(rate(ocsepp_message_validation_applied_total[2m])) by (app, pod, namespace, 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
Resolution	The alerts gets cleared when the count is not between 80 to 100.

5.3.2.14 Cat-1 Service API Validation Alerts

5.3.2.14.1 SEPPN32fServiceApiValidationFailureAlert

Table 5-268 SEPPN32fServiceApiValidationFailureAlert

Trigger Condition	Service API not in allowed list
Severity	Major

Alert details provided	**Summary**
	`N32f : Service API not in allowed list`
	Expression:
	`delta(ocsepp_security_service_api_failure_total[2m]) > 0`

OID	1.3.6.1.4.1.323.5.3.46.1.2.4005
Metric Used	ocsepp_security_service_api_failure_total
Resolution 1	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.
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: 1. Disable the Remote SEPP. 2. Delete the Remote SEPP. 3. Update and reinitiate Handshake.

5.3.2.15 Cat-2 Network ID Validation Alerts

5.3.2.15.1 SEPPN32fNetworkIDValidationHeaderFailureAlert

Table 5-269 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

Table 5-269 (Cont.) SEPPN32fNetworkIDValidationHeaderFailureAlert

Field	Details
Alert details provided	**Summary:** 'namespace: {{ $labels.namespace}}, timestamp: {{ with query "time()" }}{{ . | first | value | humanizeTimestamp }}{{ end }}: Network ID Validation has failed because {{ $labels.cause }}' **Expression:** sum(increase(ocsepp_network_id_validation_header_failure_total[2m]) >0 or (ocsepp_network_id_validation_header_failure_total unless ocsepp_network_id_validation_header_failure_total offset 2m)) by (namespace, remote_sepp_name, nf_instance_id, peer_fqdn, plmn_identifier, app, resource_uri, pod) > 0
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.

5.3.2.15.2 SEPPN32fNetworkIDValidationBodyIEFailureAlert

Table 5-270 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:** sum(increase(ocsepp_network_id_validation_body_failure_total[2m]) >0 or (ocsepp_network_id_validation_body_failure_total unless ocsepp_network_id_validation_body_failure_total offset 2m)) by (namespace, remote_sepp_name, nf_instance_id, peer_fqdn, plmn_identifier, app, resource_uri, pod) > 0
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.

5.3.2.16 Cat-3 Previous Location Check Alerts

5.3.2.16.1 SEPPPn32fPreviousLocationCheckValidationFailureAlertPercent30to40

Table 5-271 SEPPPn32fPreviousLocationCheckValidationFailureAlertPercent30to40

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

Table 5-271 (Cont.) SEPPPn32fPreviousLocationCheckValidationFailureAlertPercent30to40

Alert Details Provided	**Summary**
	`Previous location check validation failure detected between 30 to 40 Percent of Total Transactions`
	Expression
	`sum(rate(ocsepp_previous_location_validation_failure_total[2m]))by(namespace)/ sum(rate(ocsepp_previous_location_validation_requests_total[2m]))by(namespace)>=0.3`
	`and`
	`sum(rate(ocsepp_previous_location_validation_failure_total[2m]))by(namespace)/ sum(rate(ocsepp_previous_location_validation_requests_total[2m]))by(namespace)<0.4`

OID	1.3.6.1.4.1.323.5.3.46.1.2.4032
Metric Name	ocsepp_previous_location_validation_failure_total
Resolution	The alerts gets cleared when the previous location check validation failure error does not lie between 30 to 40 percent of total transactions.

5.3.2.16.2 SEPPPn32fPreviousLocationCheckValidationFailureAlertPercent40to50

Table 5-272 SEPPPn32fPreviousLocationCheckValidationFailureAlertPercent40to50

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

Alert Details Provided	**Summary**
	`Previous location check validation failure detected between 40 to 50 Percent of Total Transactions`
	Expression
	`sum(rate(ocsepp_previous_location_validation_failure_total[2m]))by(namespace)/ sum(rate(ocsepp_previous_location_validation_requests_total[2m]))by(namespace)>=0.4`
	`and`
	`sum(rate(ocsepp_previous_location_validation_failure_total[2m]))by(namespace)/ sum(rate(ocsepp_previous_location_validation_requests_total[2m]))by(namespace)<0.5`

OID	1.3.6.1.4.1.323.5.3.46.1.2.4033
Metric Name	ocsepp_previous_location_validation_failure_total
Resolution	The alerts gets cleared when the previous location check validation failure error does not lie between 40 to 50 percent of total transactions.

5.3.2.16.3 SEPPPn32fPreviousLocationCheckValidationFailureAlertPercentAbove50

Table 5-273 SEPPPn32fPreviousLocationCheckValidationFailureAlertPercentAbove50

Trigger Condition	When previous location check validation failure error is detected above 50 Percent of Total Transactions , this alert will be raised.
Severity	Critical

Alert Details Provided	**Summary**
	`Previous location check validation failure detected above 50 Percent of Total Transactions`
	Expression
	`sum(rate(ocsepp_previous_location_validation_failure_total[2m]))by(namespace)/ sum(rate(ocsepp_previous_location_validation_requests_total[2m]))by(namespace)>=0.5"`

OID	1.3.6.1.4.1.323.5.3.46.1.2.4034
Metric Name	ocsepp_previous_location_validation_failure_total
Resolution	The alerts gets cleared when the previous location check validation failure error does not lie above 50 percent of total transactions.

5.3.2.16.4 SEPPPn32fPreviousLocationCheckExceptionFailureAlertPercent30to40

Table 5-274 SEPPPn32fPreviousLocationCheckExceptionFailureAlertPercent30to40

Trigger Condition	When previous location check exception failure is detected between 30 to 40 Percent of Total Transactions , this alert will be raised.
Severity	Minor

Alert Details Provided	**Summary**
	`Previous location check exception failure detected between 30 to 40 Percent of Total Transactions`
	Expression
	`sum(rate(ocsepp_previous_location_exception_failure_total[2m]))by(namespace)/ sum(rate(ocsepp_previous_location_validation_requests_total[2m]))by(namespace)>=0.3 and sum(rate(ocsepp_previous_location_exception_failure_total[2m]))by(namespace)/ sum(rate(ocsepp_previous_location_validation_requests_total[2m]))by(namespace)<0.4`

OID	1.3.6.1.4.1.323.5.3.46.1.2.4035
Metric Name	ocsepp_previous_location_exception_failure_total

Table 5-274 (Cont.) SEPPPn32fPreviousLocationCheckExceptionFailureAlertPercent30to40

Resolution	The alerts gets cleared when the previous location check exception failure does not lie between 30 to 40 percent of total transactions.

5.3.2.16.5 SEPPPn32fPreviousLocationCheckExceptionFailureAlertPercent40to50

Table 5-275 SEPPPn32fPreviousLocationCheckExceptionFailureAlertPercent40to50

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

Alert Details Provided	**Summary**
	`Previous location check exception failure detected between 40 to 50 Percent of Total Transactions`
	Expression
	`sum(rate(ocsepp_previous_location_exception_failure_total[2m]))by(namespace)/ sum(rate(ocsepp_previous_location_validation_requests_total[2m]))by(namespace)>=0.4`
	`and`
	`sum(rate(ocsepp_previous_location_exception_failure_total[2m]))by(namespace)/ sum(rate(ocsepp_previous_location_validation_requests_total[2m]))by(namespace)<0.5`

OID	1.3.6.1.4.1.323.5.3.46.1.2.4036
Metric Name	ocsepp_previous_location_exception_failure_total
Resolution	The alerts gets cleared when the previous location check exception failure error does not lie between 40 to 50 percent of total transactions.

5.3.2.16.6 SEPPPn32fPreviousLocationCheckExceptionFailureAlertPercentAbove50

Table 5-276 SEPPPn32fPreviousLocationCheckExceptionFailureAlertPercentAbove50

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 5-276 (Cont.) SEPPPn32fPreviousLocationCheckExceptionFailureAlertPercentAbove50

Alert Details Provided	**Summary**
	`Previous location check exception failure detected above 50 Percent of Total Transactions`
	Expression
	`sum(rate(ocsepp_previous_location_exception_failure_total[2m]))by(namespace)/ sum(rate(ocsepp_previous_location_validation_requests_total[2m]))by(namespace)>=0.5`

OID	1.3.6.1.4.1.323.5.3.46.1.2.4037
Metric Name	ocsepp_previous_location_exception_failure_total
Resolution	The alerts gets cleared when the previous location check exception failure error does not lie above 50 percent of total transactions.

5.3.2.17 Rate Limiting for Egress Roaming Signaling per PLMN Alerts

5.3.2.17.1 Egress Request Rate Limit per PLMN Message Drop Above 10 Percent Alert

Table 5-277 Egress Request Rate Limit per PLMN Message Drop Above 10 Percent Alert

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
	`sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total{Status="ERL_MATCH_NO_TOKEN_LOW_PRI_REJECT"}[5m])) by (EgressRateLimitList,PLMN_ID,namespace)/ sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total[5m])) by (EgressRateLimitList,PLMN_ID,namespace) *100 >= 10 < 25`

OID	1.3.6.1.4.1.323.5.3.46.1.2.4039
Metric Name	oc_ingressgateway_plmn_egress_ratelimit_total
Resolution	The alerts gets cleared when the count goes down.

5.3.2.17.2 Egress Request Rate Limit per PLMN Message Drop Above 25 Percent Alert

Table 5-278 Egress Request Rate Limit per PLMN Message Drop Above 25 Percent Alert

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, oc_ingressgateway_plmn_egress_ratelimit_total metric will be pegged and corresponding alert will be raised.
Severity	Major

Alert Details Provided	**Summary**
	`Egress Rate Limiting Request Drop Rate detected per PLMN above 25 Percent of Total Transactions`
	Expression
	`sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total{Status="ERL_MATCH_NO_TOKEN_LOW_PRI_REJECT"}[5m])) by (EgressRateLimitList,PLMN_ID,namespace)/ sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total[5m])) by (EgressRateLimitList,PLMN_ID,namespace) *100 >= 10 < 25`

OID	1.3.6.1.4.1.323.5.3.46.1.2.4040
Metric Name	oc_ingressgateway_plmn_egress_ratelimit_total
Resolution	The alerts gets cleared when the count goes down.

5.3.2.17.3 Egress Request Rate Limit per PLMN Message Drop Above 50 Percent Alert

Table 5-279 Egress Request Rate Limit per PLMN Message Drop Above 50 Percent Alert

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 5-279 (Cont.) Egress Request Rate Limit per PLMN Message Drop Above 50 Percent Alert

Alert Details Provided	**Summary**
	`Egress Rate Limiting Request Drop Rate detected per PLMN above 50 Percent of Total Transactions`
	Expression
	`sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total{Status="ERL_MATCH_NO_TOKEN_LOW_PRI_REJECT"}[5m])) by (EgressRateLimitList,PLMN_ID,namespace)/ sum(rate(oc_ingressgateway_plmn_egress_ratelimit_total[5m])) by (EgressRateLimitList,PLMN_ID,namespace) *100 >= 50`

OID	1.3.6.1.4.1.323.5.3.46.1.2.4041
Metric Name	oc_ingressgateway_plmn_egress_ratelimit_total
Resolution	The alerts gets cleared when the count goes down.

5.3.2.18 Separate Port Configurations for N32c and N32f on the Egress Routes Alerts

5.3.2.18.1 EgressInterfaceConnectionFailure

Table 5-280 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
Alert Details Provided	**Summary:** Egress connection failure on the interface **Expression:** sum(increase(oc_egressgateway_connection_failure_total{app="n32-egress-gateway"}[5m])) by (namespace,app,Host,Port) >0
OID	1.3.6.1.4.1.323.5.3.46.1.2.4042
Metric Name	oc_egressgateway_connection_failure_total
Resolution	If the destination host and port are reachable, then the alert will be cleared.

5.4 SEPP Alert Configuration

This section describes the Measurement based Alert rules configuration for SEPP. The Alert Manager uses the Prometheus measurements values as reported by microservices in conditions under alert rules to trigger alerts.

5.4.1 Configuring SEPP Alerts in OCI

The following procedure describes how to configure the SEPP alerts for OCI. The OCI supports metric expressions written in MQL (Metric Query Language) and thus, requires a new SEPP alert file for configuring alerts in OCI observability platform.

The following are the steps:

1. Run the following command to extract the .zip file:

   ```
   unzip ocsepp_oci_alertrules_<version>.zip
   ```

   The `ocsepp_oci` and `ocsepp_oci_resources` folders are available in the zip file.

   > **Note**
   >
   > The zip file is available in the Scripts folder of CSAR package.

2. Open the `ocsepp_oci` folder, in the `notifications.tf file`, update the parameter `endpoint` with the email id of the user.
3. Open the `ocsepp_oci_resources` folder, in the `notifications.tf file`, update the parameter `endpoint` with the email id of the user.
4. Log in to the OCI Console.

   > **Note**
   >
   > For more details about logging in to the OCI, refer to Signing In to the OCI Console.

5. Open the navigation menu and select **Developer Services**. The **Developer Services** window appears on the right pane.
6. Under the **Developer Services**, select **Resource Manager**.
7. Under **Resource Manager**, select **Stacks**. The **Stacks** window appears.
8. Click **Create Stack**.
9. Select the default **My Configuration** radio button.
10. Under Stack configuration, select the folder radio button and upload the `ocsepp_oci` folder.
11. Enter the **Name** and **Description** and select the **compartment**.
12. Select the latest Terraform version from the **Terraform version** drop-down.
13. Click **Next**. The **Edit Stack** screen appears.
14. Enter the required inputs to create the SEPP alerts or alarms and click **Save** and **Run Apply.**
15. Verify that the alarms are created in the Alarm Definitions screen (**OCI Console> Observability & Management> Monitoring>Alarm Definitions**) provided.
    The required inputs are:
    - **Alarms Configuration**

- **Compartment Name** - Choose name of compartment from the drop-down
- **Metric namespace** - Metric namespace that the user provided while deploying OCI Adaptors.
- **Topic Name** - Any user configurable name. Must contain fewer than 256 characters. Only alphanumeric characters plus hyphens (-) and underscores (_) are allowed.
- **Message Format** - Keep it as **ONS_OPTIMIZED**. (This is pre-populated)
- **Alarm is_enabled** - Keep it as **True**. (This is pre-populated)

16. The steps 6 to 15 must be repeated for uploading the `ocsepp_oci_resources` folder. Here, **Metric namespace** will be pre-populated.

For more details, see *Oracle Communications Cloud Native Core, OCI Adaptor Deployment Guide*.

5.4.2 Configuring SEPP Alerts for OCCNE 1.8.x and Previous Versions

The following procedure describes how to configure the SEPP alerts for OCCNE version 1.8.x and previous versions:

1. Run the following command to find the config map to configure alerts in the Prometheus server:

   ```
   kubectl get configmap -n <Namespace>
   ```

   where, <Namespace> is the prometheus server namespace used in helm install command.
2. Run the following command to take backup of current config map of prometheus server:

   ```
   kubectl get configmaps <NAME>-server -o yaml -n <Namespace> > /tmp/
   tempConfig.yaml
   ```

   where, <Namespace> is the prometheus server namespace used in helm install command. For example, assuming chart name is "prometheus-alert", so "_NAME_-server" becomes "prometheus-alert-server", run the following command to find the config map:

   ```
   kubectl get configmaps prometheus-alert-server -o yaml -n prometheus-
   alert2 > /tmp/tempConfig.yaml
   ```

3. Run the following command to check if alertssepp is present in the tempConfig.yaml file:

   ```
   cat /tmp/t_mapConfig.yaml  | grep alertssepp
   ```

4. Run the following command to delete the alertssepp entry from the t_mapConfig.yaml file, if the alertssepp is present :

   ```
   sed -i '/etc\/config\/alertssepp/d' /tmp/t_mapConfig.yaml
   ```

5. Run the following command to add the alertssepp entry in the t_mapConfig.yaml file, if the alertssepp is not present :

   ```
   sed -i '/rule_files:/a\    \- /etc/config/alertssepp'  /tmp/
   t_mapConfig.yaml
   ```

6. Run the following command to reload the config map with the modifed file:

   ```
   kubectl replace configmap <Name> -f /tmp/t_mapConfig.yaml
   ```

7. Run the following command to add seppAlertRules.yaml file into prometheus config map under filename of SEPP alert file :

   ```
   kubectl patch configmap <Name> -n <Namespace> --type merge --patch
   "$(cat <PATH>/seppAlertRules.yaml)"
   ```

8. Restart prometheus-server pod.
9. Verify the alerts in prometheus GUI.

> **Note**
>
> ```
> Prometheus takes about 20 seconds to apply the updated Config
>     map.
> ```

5.4.3 Configuring SEPP Alerts for OCCNE 1.9.x and Higher Versions

The following procedure describes how to configure the SEPP alerts for OCCNE 1.9.x and higher versions:

1. Run the following command to apply the Prometheus rules Custom Resource Definition (CRD):

   ```
   kubectl apply -f <file_name> -n <sepp namespace>
   ```

   Where,

   - <file_name> is the SEPP alerts file
   - <sepp namespace> is the SEPP namespace

   Example:

   ```
   $ kubectl apply -f ocsepp_alerting_rules_promha.yaml -n seppsvc
   ```

2. Run the following command to check if SEPP alert file is added to Prometheus rules:

   ```
   $ kubectl get prometheusrules --namespace <namespace>
   ```

   Example:

   ```
   $ kubectl get prometheusrules --namespace seppsvc
   ```

3. Log in to Prometheus GUI and verify the alerts section.

   > **Note**
   >
   > The Prometheus server takes an updated config map that is automatically reloaded after approximately 60 seconds. Refresh the Prometheus GUI to confirm that the SEPP alerts have been reloaded.

6

SEPP Response Codes

Following are the SEPP Response Codes:

- **N32c Handshake Response Codes**
- **N32f Message Forwarding Response Codes**

N32c Handshake Response Codes

Table 6-1 N32c Handshake Response Codes

Request	Response Code	ResponseBody	Reference	Use Case	Comment/ Remarks
Capability Exchange	200 OK	SecNegotiateRspBody	TS29573_N32_Handshake.yaml	Handshake success	Handshake Success
	400 Bad Request	Problem Details	TS29573_N32_Handshake.yaml	1. Invalid RequestBody 2. 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_Handshake.yaml	When Content Length header is not present in request.	
	413 Payload Too Large	Problem Details	TS29573_N32_Handshake.yaml	When request entity is larger than limits defined by server.	
	415 Unsupported Media Type	Problem Details	TS29573_N32_Handshake.yaml	When payload format is not in json. When "Content-Type:application/json" is not present.	
	429 Too Many Requests	Problem Details	TS29573_N32_Handshake.yaml	Too many requests in a given amount of time ("rate limiting").	

Table 6-1 (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 6-2 N32f Message Forwarding Response Codes

Pod Name	Response Code	ResponseBody	Use Case	Reference	Comment
cN32f-Sepp	200 OK	Success Body	Success	TS29573_JOSEProtectedMessageForwarding.yaml	
	404 Not Found	Problem Details	N32f Context Not Found	3gpp Spec 29500	
	401 Unauthorised	Problem Details	Context Not Established	3gpp Spec 29500	
	401 Unauthorised	Problem Details	Security Capability apart from TLS	3gpp Spec 29500	

Table 6-2 (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_JOSEProtectedMessageForwarding.yaml	
	503 Service Unavailable	Problem Details	Service Unavailable	TS29573_JOSEProtectedMessageForwarding.yaml	
pN32f-Sepp	400 Bad Request	Problem Details	3GPP-target-sbi-apiroot/ destination producer NF is absent	TS29573_JOSEProtectedMessageForwarding.yaml	
	503 Service Unavailable	Problem Details	Service Unavailable	TS29573_JOSEProtectedMessageForwarding.yaml	

Note

All the exceptions/error response codes sent by the N32-EGW at consumer sand PLMN-EGW at producer will be same and we will be adding a problem statement which contains the reason why it occured.

A.1 5G CNC Mediation Service RuleBook

This section describes about the 5G CNC Mediation Service RuleBook.

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())
11	count(String key);	Integer	Returns the count of total numbers of headers with the key.	Request(headers.count("Accept"))

S.no	API	Return type	Description	Rule Example
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 3 Body Rule APIs

S.no	API	Return type	Description	Example
1	get(String path)	Object	Returns the value present at that path.	body.get("$.serviceName")
2	getAll(String path)	List<Object>	Returns the list of objects present at the path given.	body.getAll("$.serviceName")
3	has(String path);	boolean	Checks if the key is present at that path or not.	body.has("$.serviceName")
4	has(String path, Object value)	boolean	Checks if the key has the value present at that path or not.	body.get("$.serviceName","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("$","key","value")
8	add(String path, T value)	boolean	Adds the key embedded in the path with value given if element is not present.	req.body.add("$.ipEndPoints","value")
9	set(String path, T value)	boolean	Updates the key embedded in the path with value given if element is present.	rsp.body.add("$.ipEndPoints","value")
10	del(String path, T value)	boolean	Deletes the element at that path with the value present. Not supported.	req.body.del("$.path","value")

Table 3 (Cont.) Body Rule APIs

S.no	API	Return type	Description	Example
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":"10.75.24
3.203"},
{"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.

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
56ba21b45d936238aebeb7
```

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:

   ```
   kubectl 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("$","nfId","ccc5ccbb-5bb9-465f-9ace-0faf08cb4223")
       req.body.del("$.supiOrSuci")
    end
```

3. Run the following command to update the configmap with the updated rules:

```
kubectl replace ocsepp-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/notifycallback"))
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.getHttpMessageFactImpl().forwardPath=IWFConsts.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.getHttpMessageFactImpl().forwardPath=IWFConsts.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.getHttpMessageFactImpl().forwardPath=IWFConsts.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.getHttpMessageFactImpl().forwardPath=IWFConsts.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.2 Default SEPP Service APIs

The following are the list of default SEPP Service APIs supported by SEPP features:

Table 4 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 4 (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	/nnssf-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/{polAssoId}	3GPP Release Version: 16.14.0 (29.518) Specification #: 29.525 Release 16.12.0
15	GET	/npcf-ue-policy-control/v1/policies/{polAssoId}	GSMA Version: FS.36 v2.2 3GPP Release Version: 16.14.0 (29.518)
16	POST	/npcf-ue-policy-control/v1/policies/{polAssoId}/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 4 (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 4 (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/{ueId}/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/{ueId}/registrations/amf-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
43	PATCH	/nudm-uecm/v1/{ueId}/registrations/amf-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
44	PUT	/nudm-uecm/v1/{ueId}/registrations/amf-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
45	GET	/nudm-uecm/v1/{ueId}/registrations/amf-non-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
46	PATCH	/nudm-uecm/v1/{ueId}/registrations/amf-non-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
47	PUT	/nudm-uecm/v1/{ueId}/registrations/amf-non-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
48	DELETE	/nudm-uecm/v1/{ueId}/registrations/smf-registrations/{pduSessionId}	3GPP Release Version: 16.14.0 (29503-ge0)
49	PUT	/nudm-uecm/v1/{ueId}/registrations/smf-registrations/{pduSessionId}	3GPP Release Version: 16.14.0 (29503-ge0)
50	DELETE	/nudm-uecm/v1/{ueId}/registrations/smsf-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
51	GET	/nudm-uecm/v1/{ueId}/registrations/smsf-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)

Table 4 (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/{ueId}/registrations/smsf-3gpp-access	3GPP Release Version: 16.14.0 (29.518) Specification #: 29.503 Release 16.14.0
53	DELETE	/nudm-uecm/v1/{ueId}/registrations/smsf-non-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
54	GET	/nudm-uecm/v1/{ueId}/registrations/smsf-non-3gpp-access	3GPP Release Version: 16.14.0 (29503-ge0)
55	PUT	/nudm-uecm/v1/{ueId}/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-nsselection/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.