Oracle® Communications

Network Analytics Data Director Outbound Interface Specification Guide

Release 25.2.200
G48540-01
December 2025

Oracle Communications Network Analytics Data Director Outbound Interface Specification Guide, Release 25.2.200

G48540-01

Contents

Preface

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

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

Conventions

The following text conventions are used in this document:

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

My Oracle Support (MOS)

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:

- 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**.
- For Hardware, Networking and Solaris Operating System Support, select **3**.

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 terminology used in the document.

Table Acronyms

Acronym	Description
3GPP	3rd Generation Partnership Project 3GPP is the standard body for wireless communications
5G	Fifth Generation 5G is the fifth-generation technology standard for broadband cellular networks
ACL	Access Control List
BSF	Oracle Communications Cloud Native Core, Binding Support Function
CNC	Cloud Native Core CNC is a market-leading core network solution utilizing Cloud Native principles and architecture to deliver Service Agility, Innovation, Efficiency, and Adaptability for 4G and 5G network functions including an optional on-premises Cloud Native Environment
C-NF	Consumer Network Function
CSP	Communication Service Provider
CSV	Comma-separated Values, each field in the record is separated by a comma.
HA	High Availability High-availability infrastructure is configured to deliver quality performance and handle different loads and failures with minimal or zero downtime
HTTP	Hypertext Transfer Protocol HTTP is an application-layer protocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers, but it can also be used for other purposes
HTTP2	Hypertext Transfer Protocol version 2 HTTP2 is a major revision of the HTTP network protocol used by the World Wide Web
JSON	Java Script Object Notation JSON is a language-independent, text-based data format that can represent objects, arrays, and scalar data
L3L4	Layer 3 and Layer4 as in OSI layers
mTLS	Mutual Transport Layer Security mTLS authentication ensures that traffic is both secure and trusted in both directions between a client and server. It allows requests that do not log in with an identity provider (like IoT devices) to demonstrate that they can reach a given resource

Table (Cont.) Acronyms

Acronym	Description
MVP	Minimum Viable Product
NF	Network Function
NRF	Network Repository Function or Network Function Repository Function NRF is a key component of the 5G Service Based Architecture. It maintains an updated repository of all the NFs available in the operator's network along with the services provided by each of the NFs in the 5G core that is expected to be instantiated, scaled, and terminated with minimal to no manual intervention
NWDAF	Oracle Communications Networks Data Analytics Function
PCAP	PCAP files are a common format for storing packet captures. A PCAP file includes an exact copy of every byte of every packet as seen on the network, including OSI layers 2-7.
PCF	Oracle Communications Cloud Native Core, Policy Control Function
P-NF	Producer Network Function
SBI	Service Based Interface SBI is the term given to the API based communication that can take place between two NFs
SCP	Oracle Communications Cloud Native Core, Service Communication Proxy SCP helps operators to efficiently secure and manage their 5G network by providing routing control, resiliency, and observability to the core network. It leverages IT service mesh (ISTIO) and adds critical capabilities to make it 5G-aware, thereby addressing many of the challenges caused by the new service-based architecture (SBA) in the 5G core
SEPP	Oracle Communications Cloud Native Core, Security Edge Protection Proxy SEPP as a 5G node is a non-transparent proxy that sits at the perimeter of the Public Land Mobile Network (PLMN) network and enables secured communication between inter-PLMN network messages. It 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
SFTP	Secured File transport service. This will be used by the Data Director export service to transfer the files to the 3rd party server securely.
TCP	Transmission Control Protocol TCP is a connection-oriented protocol used by applications on networked hosts to connect to one another and to exchange streams of data in a reliable and in-order manner

Table (Cont.) Acronyms

Acronym	Description
TLS	Transport Layer Security TLS and its now-deprecated predecessor, Secure Sockets Layer, are cryptographic protocols designed to provide communications security over a computer network
xDR	Extended Data Record or eXtended Detail Record

What's New in This Guide

This section introduces the documentation updates for Release *25.2.2xx* in *Oracle Communications Network Analytics Data Director Outbound Interface Specification Guide*.

Release 25.2.200 - G48540-01, December 2025

There are no updates to this document in this reelase.

1 Introduction

This document provides information on the Data Director Outbound Interface specifications required by customers to use Oracle's SBI Application-Level Traffic Feed solution.

1.1 Overview

5G SBI Application-Level Traffic Feed Solution is a common pre-integrated, on demand, and automated solution that is applicable across all NFs, independent of the underlying infrastructure to mirror the 5G SBI message flows towards analytics or third-party tools.

The solution has no specific dependencies, but it provides clear insights into direct NF-to-NF communications. In addition, it maintains security while mirroring the required data and provides all necessary data through standardized interfaces to third-party consumers.

1.2 References

For more information on OCNADD, refer to the following documents:

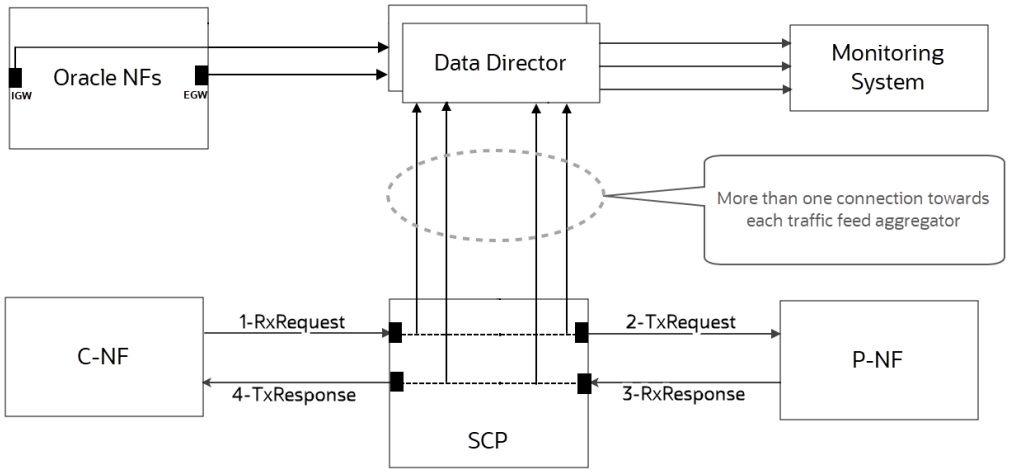
- *Oracle Communications Network Analytics Data Director User Guide*
- *Oracle Communications Network Analytics Data Director Troubleshooting Guide*
- *Oracle Communications Network Analytics Data Director Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core cnDBTier Installation,Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Network Analytics Suite Security Guide*
- *Oracle Communications Network Analytics Data Director Benchmarking Guide*

2
Architecture

This chapter covers the Oracle's 5G SBI Application-Level Traffic Feed solution that demonstrates SBI traffic feed going from Oracle 5G NFs to Oracle Data Director (OCNADD), acting as a traffic feed aggregator.

Following is a high-level block diagram showing the traffic feed from Oracle NFs:

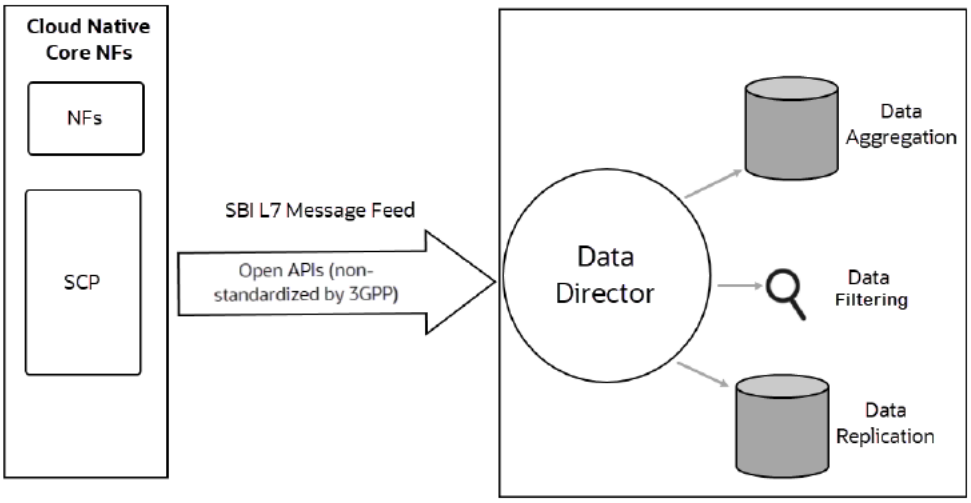
Figure 2-1 Traffic Feed from Oracle NFs



The message mirroring takes place through four endpoints covering RxRequest, TxRequest, RxResponse, and TxResponse with respect to SCP and through two endpoints. Namely, Ingress and Egress Gateways with respect to Oracle NFs. All transactions are mirrored for both SCP and Oracle NFs.

Following is a high-level block diagram depicting the solution overview:

Figure 2-2 Solution Overview



Data Director is a solution within the Network Analytics suite, which addresses the 5G traffic feed aggregation and data enrichment. It assimilates the data required to statistical prediction.

Data Director provides the following key features:

- Correlation and xDR
  - Correlation feature generates xDRs from network transactions, enabling insights.
  - xDRs aid in network troubleshooting, tracing scenarios across NFs, and KPI generation.
  - Enhances network visibility and observability through KPIs and threshold alerts.
  - Enables network efficiency reports via intuitive dashboards.
  - xDRs facilitate advanced descriptive and predictive network analytics.
  - Supports integration with network analytics frameworks like NWDAF or Insight Engine.
- Aggregation
  - Collects and aggregates the network traffic from multiple NFs, for example, SCP, SEPP, BSF, PCF, and NRF.
  - Provides the aggregated traffic feed to one or many third-party monitoring tools.
- Filtering
  - Filtering is supported for selected metadata and header attributes in OCNADD.
  - Delivers only relevant traffic, such as traffic matching specific consumer-id and/or service-name, to the third party tool.
    For more information, see "Data Filtering" and "Data Filters List" sections in *Oracle Communications Network Analytics Data Director User Guide*.
- Replication
  - Feeds multiple third-party systems with the collected feed, for example, to the monitoring, troubleshooting, and security tool.
- Secure Transport (TLS)

  - Provides the data delivery to third-party tools securely.
- Synthetic Packet Generation
  - Synthetic packet generation in Data Director converts incoming JSON data to network transfer wire format.
  - Supports L3L4 mapping on DD that allows end user to map L3 and L4 information with desired metadata/L7 layer parameters
  - Transfers converted packets securely to 3rd party monitoring probes.
  - Third party probes utilize these synthetic packets for internal monitoring applications.
  - Eliminates the necessity for additional applications by vendors to handle JSON data.
  - Saves critical compute resources and reduces associated costs for vendors.
- High Availability
  - Data Director is implemented as a Kubernetes service.
  - Multiple Kafka connections from NFs and Data Director are established to stream the ingress data from NFs.
  - Each instance of the Data Director supports two HTTP2 endpoints of the third-party monitoring tools.
  - The number of connections depends on the amount of throughput required.
- Network Visibility & KPIs
  - Data Director as a data broker holds network data for generating reports and KPIs.
  - Reports and KPIs via dashboards offer insights and operational support.
  - Generates KPIs for network utilization and load, enhancing visibility.
  - xDRs from correlated messages used for intuitive network efficiency reports.
- Data Export Service
  - Export xDRs in CSV format and/or associated messages in PCAP format.
  - Option to apply filter criteria on xDR records before data export to file storage.
  - Exported xDR records can be used to:
    * Provide deep insights into the customer network, supporting network troubleshooting, revenue assurance, and advanced analytics.
    * Troubleshoot network issues and trace scenarios across multiple NFs.
    * Generate Key Performance Indicators (KPIs) for network utilization and load.
    * Enhance network observability through traceability of records.
- Metadata Framework
  - Metadata framework on allows user to configure and generate additional metadata on DD.
  - It is used in message processing without deep inspection at the application level.
  - Applications use metadata for enrichment of other messages, filtering messages, and correlating transactions.
  - Enriched data through metadata enhances network troubleshooting capabilities.
- Data Trace

- Data Trace feature provides the capability to visualize trace of records in the DD (Data Director) UI.
- Traces can be represented by a list of transactions, calls, or sessions.
- The generated trace of records offers deep insights and visibility into the customer network.
- Provides deep insights into the customer network, supporting network troubleshooting, revenue assurance, and advanced analytics.

3
Data Director Configuration

This chapter lists the Data Director Configuration requirements on Oracle Communications Cloud Native Environment.

3.1 Requirements

Before you begin with the procedure for setting up Data Director in Cloud Native Core, ensure that the following requirements are met:

- The metadata from NFs
- The third-party target that receives the packets
- Optional TLS config for For HTTP2 and Synthetic TCP Feeds
- The HTTP standards require a response for every message sent. Oracle will provide a configuration option to ignore the response for HTTP2 Feed.
- The message acquisition point is configurable (**ingress or egreess, or both**) at the NF level.
- Kafka consumer Feed is enabled only when OCNADD services are on TLS.
- Create ACL user prior to creating Kafka feeds.

> **Note**
>
> With the HTTP2 ignore the response option enabled the OCNADD considers message transfer as successful as soon as data is sent to 3rd party monitoring consumers. OCNADD does not wait for 200 OK response to consider the message transfer as successful. Message retransmission is not attempted. However, for maintaining the connection status to 3rd party monitoring App endpoints, OCNADD still expects response for each post request sent to 3rd party monitoring App.

3.2 Outbound Protocols

Data Director currently supports three Egress Feed types, over which the 5G SBI messages and metadata added by the NFs are forwarded to third-party consumers:

- **HTTP2 Feed:** The HTTP2 Feed is used for monitoring purposes. It employs the HTTP/2 protocol, utilizing JSON as the application layer serialization protocol. Additionally, there is an option to implement TLS for security protection at the transport layer.
- **Synthetic Feed:** The Synthetic Feed operates through a TCP connection, enabling the transmission of synthetic packets. These packets contain comprehensive L2 to L7 information, complete with synthesized layers and necessary information. For added security at the transport layer, optional TLS is available.
- **Kafka Consumer Feed:** The Kafka Consumer Feed allows 3rd-party consumers to retrieve the 5G SBI messages and metadata introduced by the NFs in the form of JSON

documents, using the Kafka consumer API. To enhance security during transmission, TLS is employed at the transport layer.

3.2.1 Metadata

The following table lists the metadata that are part of the available metadata from SCP:

> **Note**
>
> For more information on each metadata component, see Data Stream Contents.

Table 3-1 Format based on 3GPP

Metadata	Information
correlation-id	This is a unique identifier in the message for correlation within a single transaction. • If an intermediate Oracle NF like SCP or SEPP sees a correlation-id custom header in the message, then it forwards the header without any modification. • Oracle NFs add the correlation-id custom header in the responses.
consumer-id	The **5G NF Instance ID** of the NF that originated the received message. **Conditions:** 1. **For SCP-initiated requests and response messages** (e.g., delegated Discovery and OAuth access token requests): • The `consumer-id` metadata is always present and matches the value of `feed-source->nf-instance-id` metadata. • **Note:** Since this corresponds to SCP's own instance ID, there is no dependency on the user-header to retrieve the `nf-instance-id` information. 2. **For Consumer NF-initiated messages**: • The `consumer-id` metadata depends on the presence of the `User-Agent` header in the received service request. • **Recommended User-Agent header format:**`<NF Type>-<NF Instance ID> <NF FQDN>`
producer-id	This is a 5G NF Instance ID of the destination NF. • Oracle SCP can find the destination NF instance Id using the authority in the service request and learning from the NRF. • Other Oracle NFs may not be able to find NF instance id of destination in be able to put destination FQDN

Table 3-1 (Cont.) Format based on 3GPP

Metadata	Information
consumer-fqdn	The **FQDN** of the NF that originated the received message. **Conditions:** 1. **For SCP-initiated requests and response messages** (e.g., delegated Discovery and OAuth access token requests): • The `consumer-fqdn` metadata is always present and matches the value of the `feed-source->nf-fqdn` metadata. • **Note:** Since this corresponds to SCP's own FQDN, there is no dependency on the user-header to retrieve the FQDN information. 2. **For Consumer NF-initiated messages**: • The `consumer-fqdn` metadata depends on the presence of the `User-Agent` header in the received service request. • **Recommended User-Agent header format:**`<NF Type>-<NF Instance ID> <NF FQDN>`
producer-fqdn	This is an FQDN of the destination NF. It depends on the presence of FQDN in the authority of service request.
hop-by-hop-id	Oracle NFs can add Hop-by-Hop id to identify a request and response pair to the next node. This is required in addition to correlation-id for uniquely identifying the request-response pair in case of re-routing. **hop-by-hop-id format is as shown below** **RxRequest/TxResponse:** • If the consumer is provided in the request: – **Format**: `<First 30 characters of consumerFqdn>_<Last 30 characters of worker-pod-instance-Id>` • If the consumer is not provided: – **Format**: `NA_< Last 30 characters of worker-pod- instanceId>` **TxRequest/RxResponse:** • **Format**: `< Last 30 characters of worker-pod-instance-Id>_< First 30 characters of producerFqdn>_Suffix` – **Suffix**: An incrementing integer that increases with each routing hop.

Table 3-1 (Cont.) Format based on 3GPP

Metadata	Information
reroute-cause	Indicate the re-route cause. Contains one of the following: • Circuit breaking: Flag to indicate that a message is an alternate attempt due to circuit breaking functionality at the SCP • Outlier detection: Flag to indicate that a message is an alternate attempt due to outlier detection functionality at the SCP • Egress-rate-limit: Flag to indicate that a message is an alternate attempt due to egress rate limiting functionality at the SCP • producer-nf-congestion: Flag to indicate that a message is an alternate attempt due to producer NF congestion • Error received • Timeout • Not Available
timestamp	This is a timestamp (in nanoseconds) at the traffic feed trigger point when the message is received or forwarded by the SCP. It is an epoch time.
message-direction	This is a parameter to indicate whether a message is ingress to or egress from NF. It can be indicated by putting the traffic feed trigger point name. • RxRequest • TxRequest • RxResponse TxResponse
feed-source	Source of this traffic feed. This contains the key-value of different identity of the node sending the traffic feed. • Feed-source : – nf-type = SCP – nf-fqdn = SCP's FQDN – nf-instance-id = SCP's NF instance id – pod-instance-id = SCP-worker's pod instance id
dd-ingress-timestamp	This is a timestamp (in nanoseconds) at Data Director when the message is received from NF's traffic feed and written to Data Director. It is an epoch time. Data Director uses this timestamp for calculating the end-to-end Data Director latency for the feed. **Note**: For HTTP2 feeds, end-to-end Data Director latency includes the time taken by the HTTP application on the feed consumer side to acknowledge the HTTP2 messages. For TCP/ Synthetic feeds, end-to-end Data Director latency includes the time taken by the feed consumer node to acknowledge the TCP packets.
source-ip	The origination IP address of the Message./ responses. It's applicable to SEPP.

Table 3-1 (Cont.) Format based on 3GPP

Metadata	Information
destination-ip	The destination IP address of the message/response. It's applicable to SEPP.
source-port	The port on which the message/response was received. It's applicable to SEPP.
destination-port	The port on which the message/response is delivered. It's applicable to SEPP.

Data Director Metadata Attributes

The following metadata attributes can be optionally generated on DD using DD metadata framework:

- `path`
- `user-agent`
- `method`
- `consumer-via`
- `ingress-authority`
- `supi`
- `previous-hop`
- `egress-authority`

`path`

Description: The path and query parts of the target URI. It is present in the HEADERS frame.

Feed Source Mapping – Priority Mapping Rule:

Table 3-2 Feed Source Mapping – Priority Mapping Rule

NF Type	Message Direction (source of attribute population)	Rule Name	Location
SCP/SEPP	First occurrence of RxRequest	`:path`	header-list
NRF/PCF/BSF	First occurrence of RxRequest or TxRequest		

Example:

```
/nausf-auth/v1/ue-authentications/reg-helm-charts-ausfauth-6bf59-kx.34/5g-aka-confirmation
```

`user-agent`

Description: The User Agent identifies which equipment made the request. It is present in the HEADERS frame.

Feed Source Mapping – Priority Mapping Rule:

Table 3-3 Feed Source Mapping – Priority Mapping Rule

NF Type	Message Direction (source of attribute population)	Rule Name	Location
SCP/SEPP	First occurrence of RxRequest	`user-agent`	header-list
NRF/PCF/BSF	First occurrence of RxRequest or TxRequest		

Example:

```
UDM-26740918-e9cd-0205-aada-71a76214d33c udm12.oracle.com
```

`method`

Description: Represents the type of request for the transaction. It is present in the HEADERS frame.

Feed Source Mapping – Priority Mapping Rule:

Table 3-4 Feed Source Mapping – Priority Mapping Rule

NF Type	Message Direction (source of attribute population)	Rule Name	Location
SCP/SEPP	First occurrence of RxRequest	`:method`	header-list
NRF/PCF/BSF	First occurrence of RxRequest or TxRequest		

Value: `POST`, `PUT`, `DELETE`, `PATCH`

`consumer-via`

Description: Contains a branch unique in space and time, identifying the transaction with the next hop.

Feed Source Mapping – Priority Mapping Rule:

Table 3-5 Feed Source Mapping – Priority Mapping Rule

NF Type	Message Direction (source of attribute population)	Rule Name	Location
SCP/SEPP	First occurrence of RxRequest	`via`	header-list
NRF/PCF/BSF	First occurrence of RxRequest or TxRequest		

Note: In case of an array of `via` in a message, the last occurrence from the list will be used.

Example:

```
SCP-scp1.5gc.mnc001.mcc208.3gppnetwork.org
```

`ingress-authority`

Description: Node's local IP/FQDN on the ingress side.

Feed Source Mapping – Priority Mapping Rule:

Table 3-6 Feed Source Mapping – Priority Mapping Rule

NF Type	Message Direction (source of attribute population)	Rule Name	Location
SCP/SEPP	Last occurrence of RxRequest	`:authority`	header-list
NRF/PCF/BSF	Last occurrence of RxRequest or TxRequest		

Example:

```
172.19.100.5:9443
```

`supi`

Description: Represents the subscription identifier. Pattern: `^(imsi-[0-9]{5,15}|nai-.+|gci-.+|gli-.+|.+)$`

Feed Source Mapping – Priority Mapping Rule:

Table 3-7 Feed Source Mapping – Priority Mapping Rule

NF Type	Message Direction (source of attribute population)	Rule Name	Location
SCP/SEPP	Last occurrence of RxRequest	`:path`	header-list
NRF/PCF/BSF	Last occurrence of RxRequest or TxRequest	`3gpp-Sbi-Discovery-supi`	header-list

Example:

```
imsi-208014489186000
```

`previous-hop`

Description: Represents a portion of the network path between the source NF and the destination NF.

Feed Source Mapping – Priority Mapping Rule:

Table 3-8 Feed Source Mapping – Priority Mapping Rule

NF Type	Message Direction (source of attribute population)
SCP/SEPP	Last occurrence of RxRequest
NRF/PCF/BSF	Last occurrence of RxRequest or TxRequest

Table 3-9 Priority Mapping Rule

Rule Name	Location
via	header-list
3gpp-Sbi-NF-Peer-Info	header-list
3gpp-Sbi-Discovery-requester-nf-instance-fqdn	header-list
3gpp-Sbi-Discovery-requester-nf-instance-id	header-list
consumer-fqdn	metadata-list
user-agent	header-list

Format of `previous-hop` value in `dd-metadata-list`:

Table 3-10 Format of previous-hop value in dd-metadata-list

Default Priority Order	DD Metadata Attribute Name	DD Metadata Value Format
1	via	via_<value>
2	3gpp-Sbi-NF-Peer-Info	nf-info_<value>
3	3gpp-Sbi-Discovery-requester-nf-instance-fqdn	nf-fqdn_<value>
4	3gpp-Sbi-Discovery-requester-nf-instance-id	nf-id_<value>
5	consumer-fqdn	con-fqdn_<value>
6	user-agent	usr-agnt_<value>

- The priority rule order can be changed in the `dd-metadata` configuration.
- A prefix (short name of the attribute) will be added before the value to identify the source attribute.
- In L3L4 mapping, Filter, and other processing features using `dd-metadata`, the prefix will be removed from the value before applying `previous-hop` conditions.

Example (populated from `via`):

```
previous-hop: "via_SCP-scp1.5gc.mnc001.mcc208.3gppnetwork.org"
```

`egress-authority`

Description: Node's local IP/FQDN on the egress side.

Feed Source Mapping – Priority Mapping Rule:

Table 3-11 Feed Source Mapping – Priority Mapping Rule

NF Type	Message Direction (source of attribute population)	Rule Name	Location
SCP/SEPP	Last occurrence of TxRequest	`:authority`	header-list
NRF/PCF/BSF	Last occurrence of RxRequest or TxRequest		

Example:

```
172.19.100.5:9443
```

> **Note**
>
> `egress-authority` is supported from release `25.2.100` release. It shall not be populated in the RxRequest message's `dd-metadata-list` header for SCP/SEPP.

3.2.2 Data Director Message

The 5G SBI message that is received or forwarded contains the following components:

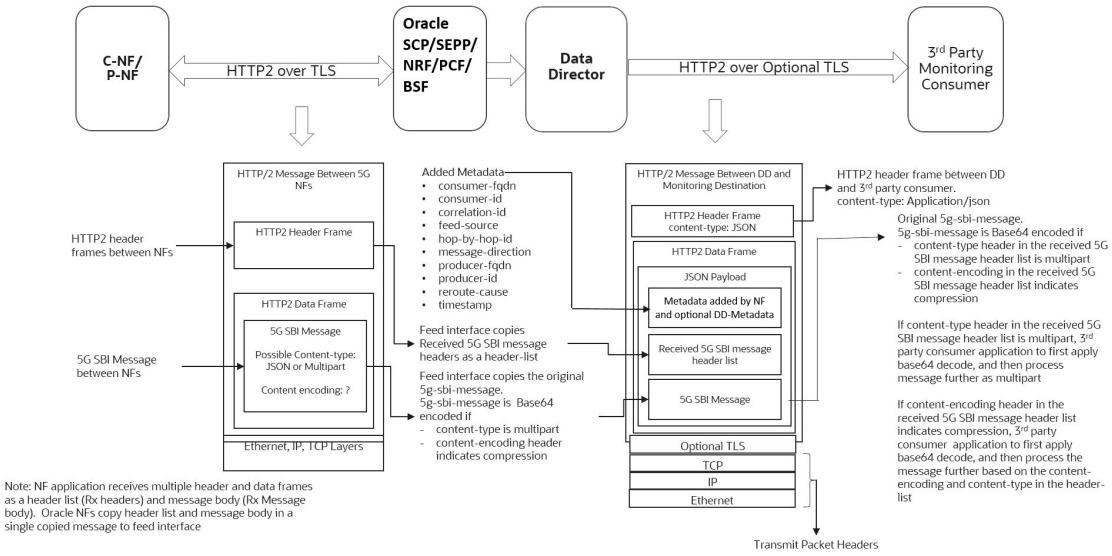
- HTTP2: HTTP2 Headers - All received HTTP2 standard and 3gpp defined headers.
- Received Data Director Message Payload

3.2.2.1 Data Director Message Format

The Data Director supports the following message formats:

- HTTP2 Message Format
- Synthetic Packet Message Format
- Kafka Consumer Egress Feed Message Format

HTTP2 Message Format



Data Director supports HTTP2 feed for forwarding from Data Director to third-party monitoring consumer applications. 5G monitoring data is forwarded to third-party monitoring consumer using HTTP2 POST requests. The following components are delivered as JSON payload in the HTTP2 data frames:

- Original received 5G SBI message headers as a header-list.
- Original received 5g sbi data as 5g-sbi-message
- Metadata-list added by NF

The 5g-sbi-message forwarded to third-party consumer application is Base64 encoded if:

- The content-type header in the received 5G SBI message header list is multipart

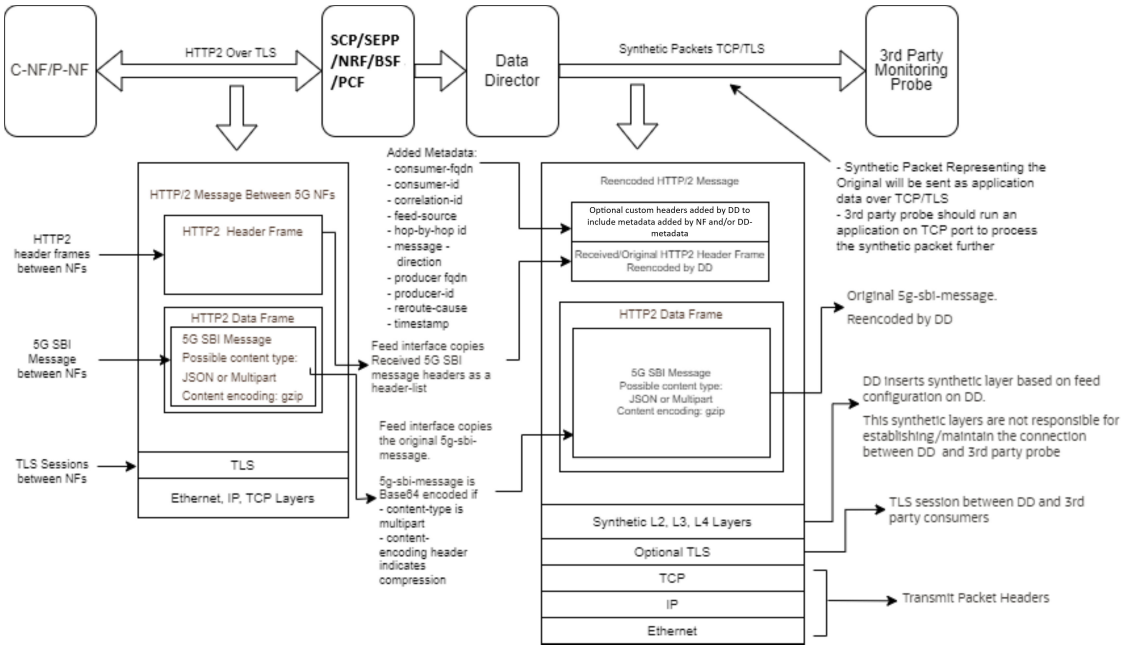
Or

- The content-encoding in the received 5G SBI message header list indicates compression

If the "content-type" header in the received 5G SBI message header list is labeled as "multipart," the third-party consumer application performs an initial base64 decode. Subsequently, the application proceeds to process the message as multipart content.

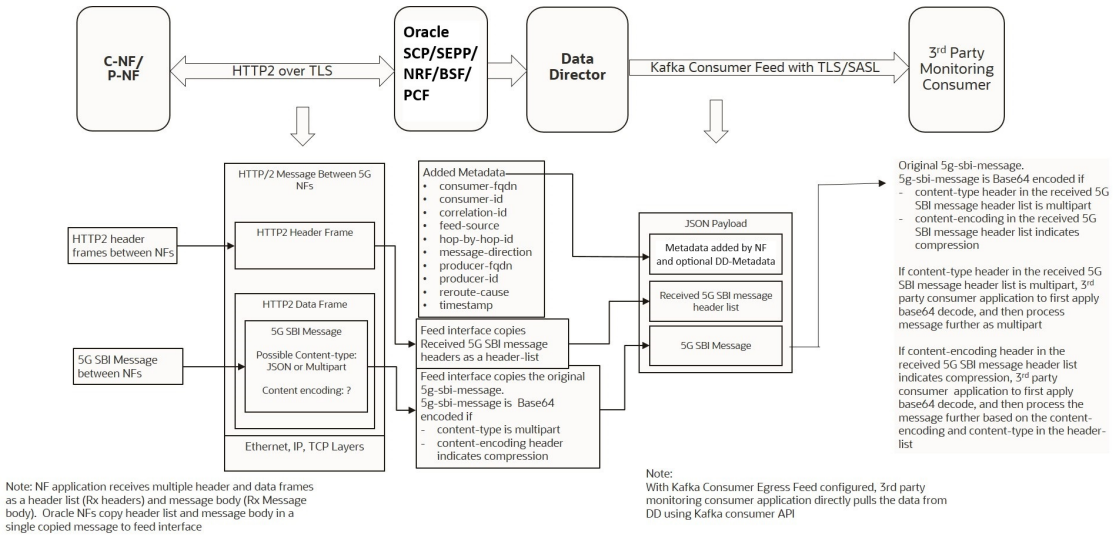
When the "content-encoding" header in the received 5G SBI message header list shows compression, the third-party consumer application first applies base64 decode. Then, it processes the message further based on the content-encoding and content-type in the header list.

Synthetic Packet Message Format



OCNADD converts incoming JSON data into network transfer wire format and sends the converted packets to the third-party monitoring applications in a secure manner. The third-party probe feeds the synthetic packets to the internal monitoring applications. The feature helps third-party vendors to eliminate the need of creating additional applications to receive JSON data and converting the data into probe compatible format, thereby saving critical compute resources and associated costs.

Kafka Consumer Egress Feed Message Format



OCNADD supports the external Kafka consumer applications using the external Kafka Feeds. This enables third-party consumer applications to directly consume data from the Data Director Kafka topic, eliminating the need for any egress adapter.

Clients need to be authenticated through either SASL or SSL (mTLS) for authorization by Kafka. As a result, enabling external Kafka feed support requires specific settings to be activated within the Kafka broker. This ensures mandatory authentication of Kafka clients by the Kafka service.

OCNADD only allows authorized and authenticated third-party applications to use the Data Director Kafka service. Application authorization is handled using the KAFKA ACL (Access Control List) functionality. Access control for the external feed is established during Kafka feed creation. Presently, third-party applications are exclusively allowed to READ from a specific topic using a designated consumer group.

3.2.2.2 Third-Party Feed Format

Third-Party HTTP2 Feed Format

A third-party HTTP2 feed contains the following components:

Figure 3-1 Third-Party HTTP2 Feed Format

Transmit Packet Header				Data Portion		
		Optional		JSON		
IPv4 Header	TCP Header	TLS	HTTP/2 Hdr	Mirror 5G SBI Message	Received 5G SBI Message Headers	Metadata Added by Mirror Feed Source

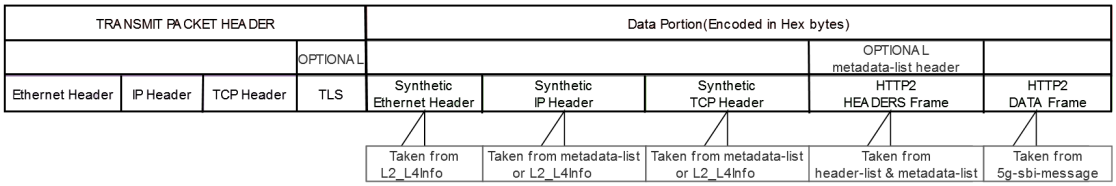
Following TLS options are supported:

- TLSv1.2 (minimum) with oracle approved TLS Ciphers
- TLSv1.2 with Static Key Cipher support (TLS_RSA_WITH_AES_128_GCM_SHA256)

- No TLS (H2C)

Third-Party Synthetic Feed Format

A third-party synthetic feed contains the following components:



3.2.2.3 Example for the JSON Data

This section shows example for the JSON Data in the following two scenarios:

- Example for JSON Data Without Optional dd-metadata-list
- Example for JSON Data With Optional dd-metadata-list

Example for JSON Data Without Optional dd-metadata-list

Following is an example for the JSON data without optional dd-metadata-list:

```
{
        "version": "1.0.0",
        "metadata-list": {
            "timestamp": 1675675430883500606,
            "correlation-id": "029d2736-1111-92c8",
            "consumer-id": "b159694e-8138-4826-bde2-ed6d82571b26",
            "producer-id": "adb514c8-b9fa-450a-bda2-4bd73140b974",
            "consumer-fqdn": "AUSF.d5g.oracle.com",
            "producer-fqdn": "UDM.d5g.oracle.com",
            "message-direction": "TxRequest",
            "feed-source": {
                "nf-type": "SCP",
                "nf-instance-id": "02043348-a5a4-41bf-84c2-945881b22ab2",
                "nf-fqdn": "SCP.d5g.oracle.com"
            }
        },
        "header-list": {
            ":scheme": "http",
            ":method": "PUT",
            ":authority": "10.100.101.100:443",
            ":path": "/nnrf-nfm/v1/nf-instances/029d2736-1111-4c48-92c8",
            "x-http2-scheme": "http",
            "accept": "application/json",
            "content-type": "application/json",
            "user-agent": "AUSF-029d2736-1111-4c48-92c8",
            "via": "UDM.d5g.oracle.com",
            "content-length": "1200"
        },
        "5g-sbi-message": {
            "nfInstanceId": "029d2736-1111-4c48-92c8-{{.nfID}}",
```

```
                    "nfType": "AUSF",
                    "nfStatus": "REGISTERED",
                    "plmnList": [
                        {
                            "mcc": "130",
                            "mnc": "52"
                        }
                    ],
                    "fqdn": "AUSF.d5g.oracle.com",
                    "interPlmnFqdn": "AUSF-d5g.oracle.com",
                    "ipv4Addresses": [
                        "192.168.2.100"
                    ],
                    "ipv6Addresses": [
                        "2001:0db8:85a3:0000:0000:8a2e:0370:7334"
                    ],
                    "capacity": 2000,
                    "load": 0,
                    "locality": "US_East",
                    "ausfInfo": {
                        "groupId": "ausfgroup",
                        "supiRanges": [
                            {
                                "start": "987654000000000",
                                "end": "987654000049999"
                            }
                        ]
                    },
                    "nfServices": [
                        {
                            "serviceInstanceId": "029d2736-1112-4c48-92c8-{{.nfID}}",
                            "serviceName": "nausf-auth",
                            "versions": [
                                {
                                    "apiVersionInUri": "v1",
                                    "apiFullVersion": "1.0.0",
                                    "expiry": "2018-12-03T18:55:08.871+0000"
                                }
                            ],
                            "nfServiceStatus": "REGISTERED",
                            "scheme": "http",
                            "fqdn": "AUSF.d5g.oracle.com",
                            "interPlmnFqdn": "AUSF-d5g.oracle.com",
                            "apiPrefix": "",
                            "defaultNotificationSubscriptions": [
                                {
                                    "notificationType": "LOCATION_NOTIFICATION",
                                    "callbackUri": "http://somehost.oracle.com/
callback-uri"
                                }
                            ],
                            "allowedPlmns": [
                                {
                                    "mcc": "130",
                                    "mnc": "52"
                                }
```

```
                    ],
                    "allowedNfTypes": [
                        "NRF",
                        "AMF",
                        "UDR",
                        "AUSF",
                        "NSSF",
                        "UDM"
                    ],
                    "capacity": 500,
                    "load": 0,
                    "supportedFeatures": "2"
                }
            ]
        }
    }
```

Example for JSON Data With Optional dd-metadata-list

Following is an example for the JSON data with optional dd-metadata-list:

```
{
        "version": "1.0.0",
        "metadata-list": {
            "timestamp": 1675675430883500606,
            "correlation-id": "029d2736-1111-92c8",
            "consumer-id": "b159694e-8138-4826-bde2-ed6d82571b26",
            "producer-id": "adb514c8-b9fa-450a-bda2-4bd73140b974",
            "consumer-fqdn": "AUSF.d5g.oracle.com",
            "producer-fqdn": "UDM.d5g.oracle.com",
            "message-direction": "RxRequest",
            "feed-source": {
                "nf-type": "SCP",
                "nf-instance-id": "02043348-a5a4-41bf-84c2-945881b22ab2",
                "nf-fqdn": "SCP.d5g.oracle.com"
            }
        },
        "header-list": {
            ":scheme": "http",
            ":method": "PUT",
            ":authority": "10.100.101.100:443",
            ":path": "/nnrf-nfm/v1/nf-instances/029d2736-1111-4c48-92c8",
            "x-http2-scheme": "http",
            "accept": "application/json",
            "content-type": "application/json",
            "user-agent": "AUSF-029d2736-1111-4c48-92c8",
            "via": "UDM.d5g.oracle.com",
            "content-length": "1200"
        },
        "dd-metadata-list": {
            "method": "PUT",
            "ingress-authority": "10.100.101.100:443",
            "path": "/nnrf-nfm/v1/nf-instances/029d2736-1111-4c48-92c8",
            "user-agent": "AUSF-029d2736-1111-4c48-92c8",
            "consumer-via": "UDM.d5g.oracle.com"
```

```
                },
                "5g-sbi-message": {
                    "nfInstanceId": "029d2736-1111-4c48-92c8-{{.nfID}}",
                    "nfType": "AUSF",
                    "nfStatus": "REGISTERED",
                    "plmnList": [
                        {
                            "mcc": "130",
                            "mnc": "52"
                        }
                    ],
                    "fqdn": "AUSF.d5g.oracle.com",
                    "interPlmnFqdn": "AUSF-d5g.oracle.com",
                    "ipv4Addresses": [
                        "192.168.2.100"
                    ],
                    "ipv6Addresses": [
                        "2001:0db8:85a3:0000:0000:8a2e:0370:7334"
                    ],
                    "capacity": 2000,
                    "load": 0,
                    "locality": "US_East",
                    "ausfInfo": {
                        "groupId": "ausfgroup",
                        "supiRanges": [
                            {
                                "start": "987654000000000",
                                "end": "987654000049999"
                            }
                        ]
                    },
                    "nfServices": [
                        {
                            "serviceInstanceId": "029d2736-1112-4c48-92c8-{{.nfID}}",
                            "serviceName": "nausf-auth",
                            "versions": [
                                {
                                    "apiVersionInUri": "v1",
                                    "apiFullVersion": "1.0.0",
                                    "expiry": "2018-12-03T18:55:08.871+0000"
                                }
                            ],
                            "nfServiceStatus": "REGISTERED",
                            "scheme": "http",
                            "fqdn": "AUSF.d5g.oracle.com",
                            "interPlmnFqdn": "AUSF-d5g.oracle.com",
                            "apiPrefix": "",
                            "defaultNotificationSubscriptions": [
                                {
                                    "notificationType": "LOCATION_NOTIFICATION",
                                    "callbackUri": "http://somehost.oracle.com/
callback-uri"
                                }
                            ],
                            "allowedPlmns": [
                                {
```

```
                        "mcc": "130",
                        "mnc": "52"
                    }
                ],
                "allowedNfTypes": [
                    "NRF",
                    "AMF",
                    "UDR",
                    "AUSF",
                    "NSSF",
                    "UDM"
                ],
                "capacity": 500,
                "load": 0,
                "supportedFeatures": "2"
            }
        ]
    }
}
```

3.3 Inbound Protocols for Non-Oracle NFs

Data Director currently supports HTTP2 Ingress Feed type, over which it received the 5GSBI messages and metadata added by the Non-Oracle NFs.

Non-Oracle NFs should be sending mirrored copy of actual HTTP2 request or response message (HTTP Header + Body) in the body of HTTP2 messages using POST method. The metadata fields from Non-Oracle NFs can be present either in "MESSAGE_HEADER" or "MESSAGE_BODY".

3.3.1 Data Transformation & Metadata Mapping

The message transformation functionality will allow data conversion and mapping from Non-Oracle NF to Oracle NF data which will be consumed by DD internal services for data processing. The conversion framework will provide capabilities to map the following metadata fields with OCNADD for processing.

The metadata fields from Non-Oracle NFs can be present either in "MESSAGE_HEADER" (as custom headers) or "MESSAGE_BODY". Based on the value of the parameter "metadataLocation" while creating configuration, the ingress adapter will take the attributes and perform the transformation of these fields to the Oracle Data Director format. If metadata is present in message body, then additional fields are required to be configured.

Metadata Mapping

Table 3-12 Metadata Mapping

Oracle Attribute Name	Ingress Attribute Name	Presence	Static Value (Default)	Description
correlation-id	<Ingress-attribute-name>	M	NA	Correlation id is mandatory to correlate all mirrored request and response messages of a transaction. If custom correlation id is not provided DD will attempt to retrieve this from 3gpp-Sbi-Correlation-Info header if available. It must be present in either of the two attributes.
timestamp	<Ingress-attribute-name>	M	NA	This property defines the timestamp of the request when it is initiated.
message-direction	<Ingress Attribute name(list)>	M	NA	It consists of both the messages direction (ingress or egress) and the message type (Request or Response). The non-Oracle feeds may send messages direction and message type in different custom headers. Oracle ingress adapter will combine both and map it to the supported OracleNfFeedDto.
consumer-fqdn	<Ingress Attribute name>	O	NA	The consumer fqdn will be mapped with the received value of configured ingress attribute name in custom headers. If the value is not present, then it will be skipped.
consumer-id	<Ingress Attribute name>	O	NA	The consumer id will be mapped with the received value of configured ingress attribute name in custom headers. If the value is not present, then it will be skipped.
hop-by-hop-id	<Ingress Attribute name>	O	NA	The hop by hop id will be mapped with the received value of configured ingress attribute name in custom headers. If the value is not present, then it will be skipped.

Table 3-12 (Cont.) Metadata Mapping

Oracle Attribute Name	Ingress Attribute Name	Presence	Static Value (Default)	Description
producer-fqdn	<Ingress Attribute name>	O	NA	The producer fqdn will be mapped with the received value of configured ingress attribute name in custom headers. If the value is not present, then it will be skipped.
producer-id	<Ingress Attribute name>	O	NA	The producer id will be mapped with the received value of configured ingress attribute name in custom headers. If the value is not present, then it will be skipped.
reroute-cause	<Ingress Attribute name>	O	NA	The reroute cause will be mapped with the received value of configured ingress attribute name in custom headers. If the value is not present, then it will be skipped.
feed-source-nf-type	<Ingress Attribute name>, Use feed-source Host Address mapping	M	<default-nf-type>	The "*nf type*" for OracleNfFeedDto will be mapped from the ingress attribute name which is provided during feed creation. However, if attribute name is not present in the custom headers, then feed-source host IP address will be taken from "custom-forward-for" or "x-forwarded-for" header if present and a look up will be performed from feed source host address mapping to get the nf-type. If the x-forwarded-for header is not present then Source IP of the request will be used. If the source IP is not present in the feed source host address map, then default value will be used for mapping. However, it is recommended to use default value when only one NF is producing data.

Table 3-12 (Cont.) Metadata Mapping

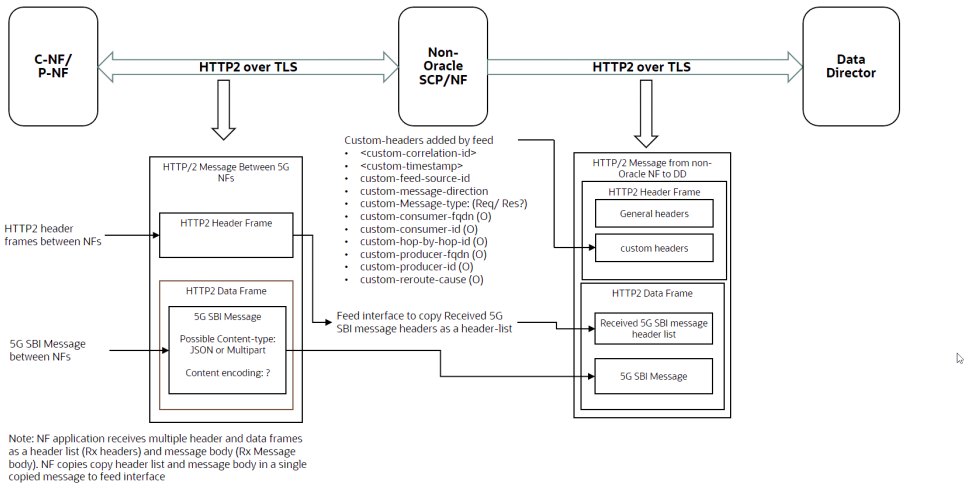
Oracle Attribute Name	Ingress Attribute Name	Presence	Static Value (Default)	Description
feed-source-nf-instance-id	<Ingress Attribute name>, Use feed-source Host Address mapping	C	<default-nf-instance-id>	The "*nf instance id*" for OracleNfFeedDto will be mapped from the ingress attribute name which is provided during feed creation. However, if attribute name is not present in the custom headers, then feed-source host IP address will be taken from "custom-forward-for" or "x-forwarded-for" header if present and a look up will be performed from feed source host address mapping to get the nf-instance-id. If the x-forwarded-for header is not present then Source IP of the request will be used. If the source IP is not present in the feed source host address map, then default value will be used for mapping. However, it is recommended to use default value when only one NF is producing data.
feed-source-nf-fqdn	<Ingress Attribute name>, Use feed-source Host Address mapping	C	<default-nf-fqdn>	The "*nf instance fqdn"* for OracleNfFeedDto will be mapped from the ingress attribute name which is provided during feed creation. However, if attribute name is not present in the custom headers, then feed-source host IP address will be taken from "custom-forward-for" or "x-forwarded-for" header if present and a look up will be performed from feed source host address mapping to get the nf-fqdn. If the x-forwarded-for header is not present then Source IP of the request will be used. If the source IP is not present in the feed source host address map, then default value will be used for mapping. However, it is recommended to use default value when only one NF is producing data.

Table 3-12 (Cont.) Metadata Mapping

Oracle Attribute Name	Ingress Attribute Name	Presence	Static Value (Default)	Description
feed-source-nf-pod-instance-id	<Ingress Attribute name>	O	<default-nf-pod-instance-id>	The "*nf pod instance id*" for OracleNfFeedDto will be mapped from the ingress attribute name which is provided during feed creation. However, if attribute name is not present in the custom headers, then default value will be used for mapping. It is recommended to use default value when only one NF is producing data.

3.3.2 Ingress Message Format for Non-Oracle NFs

The following diagram explains the format of the Ingress messages for non-Oracle network functions (NFs).



3.4 xDR

xDRs generated by correlation services are stored in corresponding xDR topic as JSON data. External application acting as Kafka consumer can subscribe to the xDR Kafka topic to read the xDR data. For more information on mandatory and optional xRD contents, see *Oracle Communications Network Analytics Data Director User Guide*.

3.4.1 xDR Format

Following is an example for xDR when includeMessageWithxDR option is set to none.

```
[
  {
    "version": "1.0.0",
    "beginTime": "2023-01-23T07:03:36.311Z",
    "endTime": "2023-01-23T07:03:36.311Z",
    "configurationName": "corr-test-2",
    "xdrStatus": "SUDR",
    "path": "/nudm-uecm/v1/imsi-208014489186000/registrations/smf-
registrations/1",
    "supi": "imsi-208014489186000",
    "methodType": "PUT",
    "producerNfType": "SCP",
    "consumerFqdn": "SMF.5g.oracle.com",
    "producerFqdn": "UDM.5g.oracle.com",
    "contentType": "application/json",
    "ueId": "imsi-208014489186000",
    "pduSessionId": 1,
    "smfInstanceId": "8e81-4010-a4a0-30324ce870b2",
    "snssai": "{\u0022sst\u0022:1,\u0022sd\u0022:\u0022000001\u0022}",
    "pcfInstanceId": "8e81-4010-a4a0-30324823334"
  }
]
```

> **Note**
>
> `includeMessageWithxDR` option allows user to select whether original feed message will be included with xDR or not and If included, which part of message to be included.

Below examples capture xDRs with includeMessageWithxDR

- `includeMessageWithxDR` is set to DATA

```
[
    // XDR
    {
        "version": "1.0.0",
        "configurationName": "cap4c",
        "beginTime": "2023-06-26T14:40:29.950313200",
        "endTime": "2023-06-26T14:40:29.950313200",
        "xdrStatus": "SUDR",
        ….
    },
    // MESSAGE 1
    {
        "5g-sbi-message":
        {
            No change in data(
                if incoming data to DD is nested, same will be transferred)
```

```
        }
    },
    // END OF MESSAGE 1
    // MESSAGE 2
    {
        "5g-sbi-message":
        {
            No change in data(
                if incoming data to DD is nested, same will be transferred)
        }
    }
    // END OF MESSAGE 2
]
```

- `includeMessageWithxDR` is set to HEADERS_DATA

```
[
  // XDR
  {
   "version": "1.0.0",
    "configurationName" : "cap4c",
    "beginTime" : "2023-06-26T14:40:29.950313200",
    "endTime" : "2023-06-26T14:40:29.950313200",
    "xdrStatus" : "SUDR",
          ….
  },
  // MESSAGE 1
 {
  "header-list":
     {
      No change in data
     },
   "5g-sbi-message":
     {
      No change in data
     }
  }
  // END OF MESSAGE 1
]
```

- `includeMessageWithxDR` is set to METADATA_HEADERS_DATA

```
[
 // XDR
  {
    "version": "1.0.0",
    "configurationName" : "cap4c",
    "beginTime" : "2023-06-26T14:40:29.950313200",
    "endTime" : "2023-06-26T14:40:29.950313200",
    "xdrStatus" : "SUDR",
     ….
  },
  // MESSAGE1
  {
    "metadata-list":{No change in format},
    "header-list":{No change in format},
```

```
    "5g-sbi-message":{No change in format}
  },
  // END OF MESSAGE 1
  // MESSAGE2
  {
    "metadata-list":{No change in format},
    "header-list":{No change in format},
    "5g-sbi-message":{No change in format}
  }
  // END OF MESSAGE 2
]
```

> **Note**
>
> The format of message would be same that is received in OCNADD from Oracle NFs.

4

Third-party Tool Configuration

Customers need to configure respective HTTP2 or TCP endpoints of third-party tools in Data Director. The connection status is managed at the TCP stack level.

4.1 For HTTP2 and Synthetic TCP Feed

Customers need to configure respective HTTP2/TCP endpoints of third-party tools in Data Director. The connection status is managed at the TCP stack level. For more information, see *Oracle Communications Network Analytics Data Director User Guide.*

4.1.1 Multiple IP Addresses

Each instance of the Data Director supports two HTTP2/TCP endpoints of the third-party monitoring tools.

4.2 For Kafka Consumer Feed

Customers need to configure ACL User for Kafka feed on Data Director to authorize the third-party application. Then the Kafka consumer on external application can be configured to subscribe to respective Kafka topic and stream data. For more information, see *Oracle Communications Network Analytics Data Director User Guide*

5
Data Stream Contents

5.1 Traffic Feed Contents

The 5G SBI feed message that is forwarded third-party contains the following components:

- Received 5G SBI Message Headers.
- Received 5G SBI Message Body
- Metadata added by traffic feed.

Following are the metadata added by traffic feed:

correlation-id

This is a unique identifier in the message for correlation within a single transaction.

- If an intermediate Oracle NF like SCP or SEPP sees a correlation-id custom header in the message, then it forwards the header without any modification.
- Oracle NFs add the correlation-id custom header in the responses.

The actual correlation-id custom header name is confirmed during the implementation.

consumer-id

This is a 5G NF Instance ID of the NF originating the received message.

- Depends on the presence of the User-Agent header in the received service request
- Recommended User-Agent header format: User-Agent:<NF Type>-<NF Instance ID> <NF FQDN>

producer-id

This is a 5G NF Instance ID of the destination NF.

- Oracle SCP can find the destination NF instance Id using the authority in the service request and learning from the NRF.
- Other Oracle NFs may not be able to find NF instance id of destination in be able to put destination FQDN.

consumer-fqdn

This is a FQDN of the network function originating the received message.

- Depends on the presence of User-Agent header in the received service request
- Recommended User-Agent header format: User-Agent:<NF Type>-<NF Instance ID> <NF FQDN>

producer-fqdn

This is an FQDN of the destination NF.

Depends on the presence of FQDN in the authority of service request.

hop-by-hop-id

Oracle NFs can add Hop-by-Hop id to identify a request and response pair to the next node.

This is required in addition to correlation-id for uniquely identifying the request-response pair in case of re-routing.

re-route cause

Indicates the re-route cause. Contains one of the following:

- Circuit breaking: Flag to indicate that a message is an alternate attempt due to circuit breaking functionality at the SCP.
- Outlier detection: Flag to indicate that a message is an alternate attempt due to outlier detection functionality at the SCP.
- Egress-rate-limit: Flag to indicate that a message is an alternate attempt due to egress rate limiting functionality at the SCP.
- producer-nf-congestion: Flag to indicate that a message is an alternate attempt due to producer NF congestion.
- Error received
- Timeout
- Not Available

timestamp

This is a timestamp (in nanoseconds) at the traffic feed trigger point when the message is received or forwarded by the SCP. It is an epoch time.

message-direction

This is a parameter to indicate whether a message is ingress to or egress from NF.

It can be indicated by putting the traffic feed trigger point name.

- RxRequest
- TxRequest
- RxResponse
- TxResponse

feed-source

Source of this traffic feed. This contains the identity of the node sending the traffic feed.

Feed-source:

- nf-type = NF Type
- nf-fqdn = NF's FQDN
- nf-instance-id = NF instance id
- pod-instance-id = pod instance id

Data Director makes reasonable attempts to deliver packets in the same sequence as received from each pod (SCP Worker pod, NRF, SEPP, PCF or BSF API Gateway pod).

Due to the parallel nature of sending packets across multiple pods within CNE and IP routing, reception in order at the monitoring system cannot be guaranteed. Note that within a single transaction, request and answer follow the same path and processed by the same pod, therefore, there is no need to follow the packet order across multiple pods.

5.2 xDR Contents

For details on mandatory and optional xDR content, see *Oracle Communications Network Analytics Data Director User Guide*.

6
Data Export Feature

Data Export Service provides the capability to export xDRs in CSV format and/or associated messages in PCAP format, which can be represented by a list of transactions, calls, or sessions. The generated export records can provide deep insights and visibility into the customer network and can be useful in features such as:

- Network troubleshooting
- Revenue assurance
- Advanced analytics & observability

Network troubleshooting is one of the key features of the monitoring solution, and the correlation capability will help Data Director to provide applications and utilities to perform troubleshooting of failing network scenarios, trace network scenarios across multiple NFs, and generate KPIs to provide network utilization and load. This feature is an enabler for network visibility and observability through the trace of records.

6.1 Prerequisites

This section lists the prerequisites for data export configuration.

1. Create SFTP test server
2. Create SFTP credential
3. Create stored procedure and events (managed internally by the Export Service)

> **Note**
>
> For steps to delete stored procedures, see "Extended xDR Storage" section in the *Oracle Communications Cloud Native Configuration Console User Guide*.

6.2 Configuration Parameters

For more information about the configuration parameters, see "Export Configuration" section in the "Configuring OCNADD" chapter of the *Oracle Communications Network Analytics Data Director User Guide*.

6.3 Data Export Format

The Data Export feature provides conversion of JSON data into two formats:

CSV Export

The CSV Export feature supports exporting xDR records and their messages from JSON format to CSV format.

JSON xDR:


```
{
    "version": "2.0.0",
    "beginTime": "2023-10-04T05:39:24.228Z",
    "endTime": "2023-10-04T05:39:27.728Z",
    "configurationName": "feed1",
    "xdrStatus": "COMPLETE",
    "totalPduCount": 4,
    "totalLength": 7030,
    "transactionTime": 3500,
    "userAgent": "Go-http-client/2.0",
    "path": "/nudm-uecm/v1/234150999999999/registrations/amf-3gpp-access",
    "pei": "990000862471854",
    "methodType": "PUT",
    "statusCode": "503",
    "consumerVia": "2.0 SEPP-sepp1.inter.oracle.com",
    "producerVia": "2.0 SEPP-sepp2.inter.oracle.com",
    "feedSourceNfFqdn": "SEPP",
    "feedSourceNfId": "9faf1bbc-6e4a-4454-a507-aef01a101a06",
    "consumerId": "b159694e-8138-4826-bde2-ed6d82571b26",
    "producerId": "adb514c8-b9fa-450a-bda2-4bd73140b974",
    "producerFqdn": "udmsonu.5gc.mnc555.mcc444.3gppnetwork.org:5815",
    "contentType": "application/json"
}
```


CSV xDR:


```
version,beginTime,endTime,configurationName,xdrStatus,totalPduCount,totalLengt
h,transactionTime,userAgent,path,pei,methodType,statusCode,consumerVia,produce
rVia,feedSourceNfFqdn,feedSourceNfId,consumerId,producerId,producerFqdn,conten
tType
2.0.0,2023-10-04T05:39:24.228Z,2023-10-04T05:39:27.728Z,feed1,COMPLETE,4,7030,
3500,Go-http-client/2.0,/nudm-uecm/v1/234150999999999/registrations/amf-3gpp-
access,990000862471854,PUT,503,2.0 SEPP-sepp1.inter.oracle.com,2.0 SEPP-
sepp2.inter.oracle.com,SEPP,9faf1bbc-6e4a-4454-a507-
aef01a101a06,b159694e-8138-4826-bde2-ed6d82571b26,adb514c8-b9fa-450a-
bda2-4bd73140b974,udmsonu.5gc.mnc555.mcc444.3gppnetwork.org:5815,application/
json
```


PCAP Export

The PCAP export feature is specifically designed for exporting messages of xDR records in JSON format to PCAP format.

7

High Availability for Feed

Data Director will support High Availability as per the requirements of customers.

Messages will be available for up to six hours in case of site connectivity issues. Increased redundancy or message caching will require additional resources. Customers will be provided with an option to configure the message caching for up to six hours.

> **Note**
>
> Current Data Director software release assumes underlying data storage provides data redundancy.

In case of recovery after failure, the data streaming will resume from where it got stopped. If the failure duration is more than the retention duration (based on the HA configuration), the streaming will resume from the oldest available data stored in Data Director.

8
Error Handling

The error handling is maintained by 5G Core NFs. Data Director is an aggregator of 5G Core NF Feed and streams it towards the third-party tools. However, Data Director can create more than one copy of messages, so the message loss is mitigated. Also, Data Director caches the messages for up to six hours and restarts the stream once connectivity is restored with the third-party system.

> **Note**
>
> Current Data Director software release assumes underlying data storage provides data redundancy.