ETSI TS 129 122 V15.1.0 (2018-10)

TECHNICAL SPECIFICATION

Universal Mobile Telecommunications System (UMTS);
LTE;
5G;
T8 reference point for Northbound APIs
(3GPP TS 29.122 version 15.1.0 Release 15)

Reference

RTS/TSGC-0329122vf10

Keywords

5G,LTE,UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

Y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present specification describes the protocol for the T8 reference point between the SCEF and the SCS/AS. The T8 reference point and the related stage 2 functional requirements are defined in 3GPP TS 23.682 [2].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.682: "Architecture enhancements to facilitate communications with packet data networks and applications".

[3] 3GPP TS 23.032: "Universal Geographical Area Description (GAD)".

[4] IETF RFC 5246, "The Transport Layer Security (TLS) Protocol Version 1.2".

[5] IETF RFC 7159: "The JavaScript Object Notation (JSON) Data Interchange Format".

[6] Hypertext Transfer Protocol (HTTP) Status Code Registry at IANA, http://www.iana.org/assignments/http-status-codes

[7] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

[8] IETF RFC 7807: "Problem Details for HTTP APIs".

[9] 3GPP TS 29.154: "Service capability exposure functionality over Nt reference point".

[10] 3GPP TS 29.214: "Policy and Charging Control over Rx reference point".

[11] 3GPP TS 29.336: "Home Subscriber Server (HSS) diameter interfaces for interworking with packet data networks and applications".

[12] 3GPP TS 29.128: "Mobility Management Entity (MME) and Serving GPRS Support Node (SGSN) interfaces for interworking with packet data networks and applications".

[13] 3GPP TS 29.201: "Representational State Transfer (REST) reference point between Application Function (AF) and Protocol Converter (PC)".

[14] 3GPP TS 23.003: "Numbering, addressing and identification".

[15] IETF RFC 3339: "Date and Time on the Internet: Timestamps".

[16] IETF RFC 7230: "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing".

[17] IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".

[18] IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests".

[19] IETF RFC 7233: "Hypertext Transfer Protocol (HTTP/1.1): Range Requests".

[20] IETF RFC 7234: "Hypertext Transfer Protocol (HTTP/1.1): Caching".

[21] IETF RFC 7235: "Hypertext Transfer Protocol (HTTP/1.1): Authentication".

[22] IETF RFC 7540: "Hypertext Transfer Protocol Version 2 (HTTP/2)".

[23] 3GPP TS 29.155: "Traffic steering control; Representational state transfer (REST) over St reference point".

[24] 3GPP TS 29.368: "Tsp interface protocol between the MTC Interworking Function (MTC-IWF) and Service Capability Server (SCS)".

[25] 3GPP TS 29.337: "Diameter-based T4 interface for communications with packet data networks and applications".

[26] 3GPP TS 29.250: "Nu reference point between SCEF and PFDF for sponsored data connectivity".

[27] Open API Initiative, "OpenAPI 3.0.0 Specification", https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.0.md.

[28] IETF RFC 1166: "Internet Numbers".

[29] IETF RFC 5952: "A recommendation for Ipv6 address text representation".

[30] 3GPP TS 29.153: "Service capability exposure functionality over Ns reference point".

[31] 3GPP TS 24.250: "Protocol for Reliable Data Service between UE and SCEF; Stage 3".

[32] IETF RFC 6455: "The Websocket Protocol".

[33] 3GPP TS 29.272: "Mobility Management Entity (MME) and Serving GPRS Support Node (SGSN) related interfaces based on Diameter protocol".

[34] 3GPP TS 29.338: "Diameter based protocols to support Short Message Service (SMS) capable Mobile Management Entities (MMEs)".

[35] 3GPP TS 33.187: "Security aspects of Machine-Type Communications (MTC) and other mobile data applications communications enhancements".

[36] 3GPP TS 29.468: "Group Communication System Enablers for LTE (GCSE_LTE);MB2 Reference Point;Stage 3".

[37] 3GPP TS 29.116: "Presentational state transfer over xMB reference point between Content Provider and BM-SC".

[38] IETF RFC 5789: "PATCH method for HTTP".

[39] IETF RFC 7386: "JSON Merge Patch".

[40] IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format".

[41] YAML (10/2009): "YAML Ain't Markup Language (YAML™) Version 1.2", http://www.yaml.org/spec/1.2/spec.html.

[42] 3GPP TS 29.572: "5G System; Location Management Services; Stage 3".

[43] 3GPP TS 23.040: "Technical realization of the Short Message Service (SMS)".

[44] 3GPP TS 29.500: "5G System; Technical Realization of Service Based Architecture; Stage 3".

[45] 3GPP TS 29.571: "5G System; Common Data Types for Service Based Interfaces Stage 3".

[46] IETF RFC 6733: "Diameter Base Protocol".

[47] 3GPP TS 23.222: "Common API Framework for 3GPP Northbound APIs; Stage 2".

[48] 3GPP TS 29.222: "Common API Framework for 3GPP Northbound APIs; Stage 3".

[49] 3GPP TS 29.501: "5G System; Principles and Guidelines for Services Definition; Stage 3".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

AF	Application Function
AS	Application Server
ASP	Application Service Provider
BDT	Background Data Transfer
CAPIF	Common API Framework
CP	Communication Pattern
DDN	Downlink Data Notification
DL	Downlink
eNB	Evolved Node B
GMD	Group Message Delivery
IWK-SCEF	Interworking SCEF
JSON	JavaScript Object Notation
MIME	Multipurpose Internet Mail Extensions
MT	Mobile Terminated
MTC	Machine Type Communications
NIDD	Non-IP Data Delivery
NP	Network Parameter
PCRF	Policy and Charging Rule Function
PDN	Packet Data Network
PFD	Packet Flow Description
PFDF	Packet Flow Description Function
RCAF	RAN Congestion Awareness Function
REST	Representational State Transfer
SCEF	Service Capability Exposure Function
SCS	Services Capability Server
TAI	Tracking Area Identity
TLTRI	T8 Long Term Transaction Reference ID
YAML	YAML Ain't Markup Language

4 T8 reference point

4.1 Overview

The T8 reference point is between the SCS/AS and the SCEF. It specifies APIs that allow the SCS/AS to access the services and capabilities provided by 3GPP network entities and securely exposed by the SCEF.

This document also specifies the procedures triggered at the SCEF by API requests from the SCS/AS and by event notifications received from 3GPP network entities.

The stage 2 level requirements and signalling flows for the T8 reference point are defined in 3GPP TS 23.682 [2].

The T8 reference point supports the following procedures:

- Monitoring Procedures

- Procedures for resource management of Background Data Transfer
- Procedures for changing the chargeable party
- Procedures for Non-IP Data Delivery
- Procedures for Device Triggering
- Procedures for Group Message Delivery
- Procedures for Reporting of Network Status
- Procedures for Communication Pattern Parameters Provisioning
- Procedures for PFD Management
- Procedures for Enhanced Coverage Restriction Control
- Procedures for Network Parameter Configuration
- Procedures for setting up an AS session with required QoS
- Procedures for MSISDN-less Mobile Originated SMS

4.2 Reference model

The T8 reference point resides between the SCEF and the SCS/AS as depicted in figure 4.2.1. The overall SCEF architecture is depicted in subclause 4.2 of 3GPP TS 23.682 [2].

NOTE: The SCS/AS can be provided by the third party.

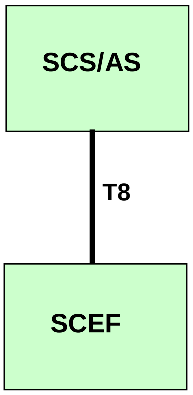

Figure 4.2.1: T8 reference model

4.3 Functional elements

4.3.1 SCEF

The SCEF is a functional element, which provides a means to securely expose the services and capabilities provided by 3GPP network interfaces. The SCEF provides access to network capabilities through homogenous application programming interfaces.

Individual instances of SCEF may vary depending on what service capabilities are exposed and what API features are supported.

The SCEF shall protect the other PLMN entities (e.g. HSS, MME) from requests exceeding the permission arranged in the SLA with the third-party service provider.

When needed, the SCEF supports mapping between information exchanged with SCS/AS (e.g. geographical identifiers) and information exchanged with internal PLMN functions (e.g. cell-Id, eNB-Id, TAI, MBMS SAI etc.). This mapping is assumed to be provided by the SCEF based on local configuration data.

4.3.2 SCS/AS

The SCS is the entity, which connects MTC application servers to the 3GPP network to enable them to communicate through specific 3GPP defined services with UEs used for MTC and with the SCEF in the HPLMN. The SCS offers capabilities for use by one or multiple MTC application servers. The MTC applications in the external network are hosted on one or more Ass.

An SCS/AS can get services from multiple SCEFs, and an SCEF can provide service to multiple SCS/AS.

The SCS is controlled by the operator of the HPLMN or by a MTC Service Provider.

The AS can be controlled by a 3rd party.

4.4 Procedures over T8 reference point

4.4.1 Introduction

All procedures that operate across the T8 reference point, as specified in 3GPP TS 23.682 [2], are specified in the following subclauses.

4.4.2 Monitoring Procedures

4.4.2.1 General

The procedures are used to perform event monitoring fucntion via the T8 inteface, which include:

- Monitoring event configuration as specified in subclause 4.4.2.2;
- Monitoring event report as specified in subclause 4.4.2.3; and
- Network initiated notification of monitoring event cancellation as specified in subclause 4.4.2.4.

4.4.2.2 Monitoring Events Configuration

4.4.2.2.1 General

In order to subscribe a new monitoring event configuration, the SCS/AS shall send an HTTP POST message to the SCEF to the resource "Monitoring Event Subscriptions". The body of the HTTP POST message shall include:

- SCS/AS Identifier;
- Monitoring Type;
- Notification Destination Address; and
- one of External Identifier or MSISDN or External Group Identifier, the External Identifier or MSISDN indicates the subscription for an individual UE and the External Group Identifier indicates a group of UEs.
- In addition, the HTTP POST request may include:
  - Maximum Number of Reports;
  - Monitoring Duration indicated by the property "monitorExpireTime";
  - Group Reporting Guard Time.

If the Subscription_modification feature is supported, the SCS/AS may send an HTTP PUT message, in order to update an existing monitoring event subscription, the SCS/AS may send an HTTP PUT message to the resource "Individual Monitoring Event Subscription" replacing all properties in the existing configuration.

For one-time monitoring request, the SCS/AS shall include the Maximum Number of Reports with the value be set to 1, and shall not include the Monitoring Duration in the HTTP request message sent to the SCEF.

Upon receipt of the HTTP POST or PUT request message, if the SCS/AS is authorized to perform the request, the SCEF shall check whether the parameters (e.g. Maximum Number of Reports, Monitoring Duration, Maximum

Latency, Maximum Response Time, Suggested number of downlink packets) in the HTTP request body are within the range defined by operator policies, if one or more of these parameters are not within the range, the SCEF shall:

- either reject the request message by sending an HTTP response to the SCS/AS with a status code set to 403 Forbidden and indicate which parameters are out of the range in the "invalidParams" attribute of the "ProblemDetails" structure; or
- modify the parameters which are not within the range by selecting different values which are in the range.

For individual UE configuration request, the SCEF shall also check whether the Idle Status Indication is included for UE reachability event, if the Idle Status Indication is received in the request but not supported by the network, the SCEF shall reject the request message by sending an HTTP response to the SCS/AS with a status code set to 403 Forbidden.

After validation, the SCEF shall store the parameters and

- may assign an SCEF Reference ID related to the created monitoring event subscription resource; and based on operator policies, shall
  - map the accuracy into permissible granularity for location reporting event;
  - map the location area into a list of cells, eNodeB(s) and/or RAI(s)/TAI(s) and derive the corresponding MME(s)/SGSN(s), for number of UEs present in a geographic area event.

In order to delete a previous active configured monitoring event subscription at the SCEF, the SCS/AS shall send an HTTP DELETE message to the SCEF to the resource "Individual Monitoring Event Subscription" which is received in the response to the request that has created the monitoring events subscription resource. The SCEF shall detemine the SCEF Reference ID related to the active monitoring subscription resource.

4.4.2.2.2 Monitoring Events Configuration via HSS

4.4.2.2.2.1 General

The following monitoring events: Loss of connectivity, UE reachability, Location Reporting, Change of IMSI-IMEI(SV) Association, Roaming Status, Communication Failure and Availability after DNN Failure are applicable for the monitoring event configuration via HSS for an individual UE or a group of Ues.

Only one-time reporting is supported if the "reachabilityType" attribute sets to "SMS" for the event UE reachability, if the "locationType" attribute sets to "LAST_KNOWN_LOCATION" for the event Location Reporting in the monitoring event request.

4.4.2.2.2.2 Configuration Request for an individual UE

Upon receipt of the request from the SCS/AS, if the configuration request from the SCS/AS is for an individual UE, the SCEF shall interact with the HSS via S6t as specified in 3GPP TS 29.336 [11].

Upon receipt of the successful response from the HSS,

- if it is a one-time monitoring request and the monitoring event report is received, the SCEF shall delete the associated configuration, send the HTTP POST response message to the SCS/AS with a "200 OK" status code and including the received monitoring event report.
- otherwise, the SCEF shall,
  - for HTTP POST request, create a resource "Individual Monitoring Event Subscription" addressed by the URI that contains the SCS/AS identifier and an SCEF-created subscription identifier, and send an HTTP POST response to the SCS/AS with a "201 Created" status code, the final suggested configuration parameter(s) (if modified), the indication(s) for the discarded parameter(s) (if discarded), the monitoring event report if received and a location header field containing the URI for the created resource.
  - for HTTP PUT request, update the active resource "Individual Monitoring Event Subscription" addressed by the URI and send an HTTP response to the SCS/AS with a "200 OK" status code, the final suggested configuration parameter(s) (if modified), the indication(s) for the discarded parameter(s) (if discarded) and the monitoring event report if received.

- for HTTP DELETE request, delete the active resource "Individual Monitoring Event Subscription" addressed by the URI and send an HTTP response to the SCS/AS with a "204 No Content" status code, or a "200 OK" status code and including the monitoring event report if received.

If the SCEF receives a response with an error code from the HSS, the SCEF shall not create, update or delete the resource and shall respond to the SCS/AS with a status code set to "500 Internal Server Error".

4.4.2.2.2.3 Configuration Request for a group of Ues

Upon receipt of the request from the SCS/AS, if the External Group Identifier is included in the configuration request from the SCS/AS, then the monitoring configuration is for a group of Ues. The SCEF shall interact with the HSS via S6t as specified in 3GPP TS 29.336 [11].

Upon receipt of the successful response indicating that group processing is in progress from the HSS before beginning the processing of individual Ues, the SCEF shall,

- for HTTP POST request, create a resource "Individual Monitoring Event Subscription" addressed by a URI that contains the SCS/AS identity and an SCEF-created subscription identifier, store the number of UEs received in the response message from the HSS within the resource and send an HTTP POST response to the SCS/AS including a location header field containing the URI for the created resource and "201 Created" status code to acknowledge the SCS/AS of the successful group processing request.

- for HTTP PUT request, update the active resource "Individual Monitoring Event Subscription" addressed by the requested URL, and send "200 OK" status code to acknowledge the SCS/AS of the successful group processing request.

- for HTTP DELETE request, delete the active resource "Individual Monitoring Event Subscription" addressed by the requested URI and send an HTTP response to the SCS/AS with "204 No Content" status code.

If the SCEF receives a response with an error code from the HSS, the SCEF shall not create, update or delete the resource and shall respond to the SCS/AS with a status code set to "500 Internal Server Error".

Upon receipt of the processing result of the individual UEs from the HSS, the SCEF shall perform as follows:

- if no Group Reporting Guard Time is received, the SCEF shall send an HTTP POST request message to the SCS/AS including a reference to the related monitoring subscription, a list of configuration failure result if received for the group members, and the "monitoringEventReports" attribute including a list of monitoring event reports if received for the group members;

- otherwise, the SCEF shall accumulate all of the configuration results and/or monitoring event reports received from the HSS for the group members until the Group Reporting Guard Time expires. Then the SCEF shall send an HTTP POST request message to the SCS/AS including a reference to the related monitoring subscription, and a list of configuration failure result if received for the group members, and the "monitoringEventReports" attribute including a list of monitoring event reports at the Group Reporting Guard Time.

The SCS/AS shall send an HTTP POST response to acknowledge the SCEF about the handling result of the received HTTP POST request.

4.4.2.2.3 Monitoring Events Configuration directly via MME/SGSN

The monitoring event Number of UEs in a geographic area is applicable for the monitoring event configuration via MME/SGSN. Only one-time reporting is supported for this event with the value of Maximum Number of Reports indicated by "maximumNumberOfReports" sets to 1.

Upon receipt of the HTTP POST request from the SCS/AS, the SCEF shall

- resolve the location area to the involved SGSN(s)/MME(s) by local configuration;

- interact with the HSS via the S6t interface as specified in 3GPP TS 29.336 [11] if the External Group ID is included; and

- interact with the SGSN(s)/MME(s) via the T6a/b inteface as specified in 3GPP TS 29.128 [12].

After collecting responses from the SGSN(s)/MME(s), if the SCEF does not receive any successful response from the involved SGSN(s)/MME(s), the SCEF shall respond to the SCS/AS with a status code set to 500 Internal Server Error; otherwise the SCEF should send 200 OK status code to acknowledge the SCS/AS with one aggregated report in the requested area by including the total count of number of UEs in the "ueCount" attribute and the External Identifier(s) (if available) or the MSISDN(s) (if available) associated with External Group ID.

NOTE: It is possible that the number of UEs does not reflect the actual number of UEs in the designated area (e.g. some SGSN(s)/MME(s) do not respond successfully). The SCEF still provides the result to the SCS/AS if at least one SGSN/MME returns successful response.

4.4.2.2.4 Monitoring Events Configuration via PCRF

4.4.2.2.4.1 General

The following monitoring events: the location reporting event and communication failure event are applicable for the monitoring event configuration via PCRF for an individual UE.

If monitoring event configuration via PCRF is used for a subscription resource, the Subscription_modification feature cannot be supported.

Only the location reporting event is applicabe for the monitoring event configuration via PCRF for a group of Ues.

Only one-time reporting is supported for the monitoring event configuration via PCRF.

HTTP PUT is not supported for the monitoring event configuration via PCRF. If it is received, the SCEF shall reject the HTTP PUT message with 403 Forbidden during monitoring.

4.4.2.2.4.2 Configuration Request for an individual UE

Upon receipt of the HTTP POST request from the SCS/AS, if the configuration request from the SCS/AS is for an individual UE, the SCEF shall:

- interact with the PCRF via the Rx inteface by using existing AF session or establishing a new AF session as specified in 3GPP TS 29.214 [10];

NOTE 1: The SCEF can derive the service information over the Rx interface based on SCS/AS ID for communication failure event.

- after receiving the AAA message from the PCRF, create a resource which represents the monitoring event configuration, addressed by a URI that contains the SCS/AS identifier and an SCEF-created subscription identifier; and
- send a corresponding status code to acknowledge the SCS/AS of the successful processing request in the HTTP response message.

Then the SCEF shall wait for the reporting from the PCRF as specified in 3GPP TS 29.214 [10].

NOTE 2: Different events can be reported in different messages according to 3GPP TS 29.214 [10], e.g. STR/RAR for communication failure.

During configuration resource deletion, the SCEF shall also terminate the AF session if it was established and used only for event monitoring.

4.4.2.2.4.3 Configuration Request for a group of Ues

Upon receipt of the HTTP POST request from the SCS/AS, if the configuration request from the SCS/AS is for a group of Ues, the SCEF shall:

- interact with all PCRFs in the same PLMN via Nta application of Nt inteface as specified in 3GPP TS 29.154 [9];
- after collecting ECA message from all PCRFs, create a resource which represents the monitoring event configuration, addressed by a URI that contains the SCS/AS identifier and an SCEF-created subscription identifier; and
- send a corresponding status code to acknowledge the SCS/AS of the successful processing request in the HTTP response message.

Then the SCEF shall wait for the reporting from the PCRF(s) as specified in 3GPP TS 29.154 [9].

4.4.2.3 Reporting of Monitoring Event Procedure

Upon receipt of the Monitoring Event Report from the HSS or the MME/SGSN as defined in subclause 5.6.3 or subclause 5.6.8 of 3GPP TS 23.682 [2], from the PCRF as defined in subclause 5.6.5 or from the IWK-SCEF as defined in subclause 5.6.8 of 3GPP TS 23.682 [2], the SCEF shall determine the monitoring event subscription associated with the corresponding Monitoring Event Report.

If the monitoring event subscription refers to a Monitoring Event Configuration for a single UE or the monitoring event subscription refers to a group-based Monitoring Event configuration, and if no Group Reporting Guard Time was set, then the SCEF shall send an HTTP POST message including the link to the SCEF-created subscription resource and the received Monitoring Event Report to the identified destination. If the monitoring event subscription refers to a group-based Monitoring Event Configuration, and if Group Reporting Guard Time was provided during the Monitoring Event configuration procedure, then the SCEF shall accumulate all of the received Monitoring Event reports for the group of UEs until the Group Reporting Guard Time expires or the monitoring duration indicated by the property "monitorExpireTime" is reached.

Upon expiration of Group Reporting Guard Time or upon reach of the monitoring duration, the SCEF shall send an HTTP POST message to the identified destination including the link to the SCEF-created subscription resource and a list of accumulated Monitoring Event Reports for each UE identified by either External Identifier or MSISDN. The destination URL of the HTTP POST is provided by the SCS/AS during the Monitoring Event Configuration procedure.

If the monitoring event subscription refers to a one-time monitoring request or a continuous monitoring request but the maximum number of reports is reached, the SCEF shall determine the reporting is completed, and then delete the corresponding subscription resource "Individual Monitoring Event Subscription". In addition, the SCEF shall interact with the HSS to delete the event configuration if the event configuration was performed via the HSS but event reports were performed via the SGSN/MME. The SCEF determines that the reporting for a group is completed by comparing the total number of received reports with the number of UEs for the group (received from the HSS during event configuration for a group of UEs) multiplied by the maximum number of reports.

When the monitoring duration indicated by the property "monitorExpireTime" is reached, the SCEF shall delete the related event subscription and event configuration locally, the SCS/AS shall no longer address the corresponding resource "Individual Monitoring Event Subscription".

4.4.2.4 Network-initiated Explicit Monitoring Event Deletion Procedure

Upon receipt of an identifier for the event to be deleted from the HSS as defined in 3GPP TS 29.336 [11], the SCEF shall determine the subscription identifier associated with the indicated active monitoring subscription. Then the SCEF shall delete the related resource "Individual Monitoring Event Subscription", send an HTTP POST message including the subscription identifier and a cancellation indication to the identified destination. The cancellation indication shall set to "true" indicating to cancel the configured monitoring subscription. The destination URL of the HTTP POST is provided by the SCS/AS during the Monitoring Event Configuration procedure.

NOTE: The above procedure can be triggered from the HSS due to parameter overwritten by Network Parameter Configuration.

4.4.3 Procedures for resource management of Background Data Transfer

These procedures are used by an SCS/AS to perform the resource management of background data transfer (BDT) to a set of Ues, i.e. the SCS/AS requests a time window and related conditions from the SCEF via the T8 interface.

In order to create a resource for the background data transfer policy, the SCS/AS shall send an HTTP POST message to the SCEF for the "BDT Subscription" resource to negotiate the transfer policy. The body of the HTTP POST message shall include SCS/AS Identifier, Volume per UE (total volume for both DL and UL or separate volume for DL and/or UL), Number of Ues, Desired Time Window and optionally a location area information. After receiving the HTTP POST message, if the SCS/AS is authorized, the SCEF shall map the SCS/AS Identifier to ASP Identifier and negotiate the transfer policy with the PCRF as defined in 3GPP TS 29.154 [9]. After receiving the response including the determined transfer policies from the PCRF, the SCEF shall create a resource "Individual BDT Subscription" which represents the BDT subscription, addressed by a URI that contains the SCS/AS identifier and an SCEF-created subscription identifier, and shall respond to the SCS/AS with a 201 Created message, including a Location header field containing the URI for the created resource and a message body, which may also include Reference ID and a set of transfer policies. The SCS/AS shall use the URI received in the Location header in subsequent requests to the SCEF to refer to this background data transfer subscription. If the SCEF receives a response with an error code from the PCRF, the SCEF shall not create the resource and shall respond to the SCS/AS with 500 Internal Server Error.

The SCS/AS may also send an HTTP PUT message to the SCEF for the "Individual BDT Subscription" resource to request starting an update for negotiation of background data transfer policy. The body of the HTTP PUT message shall include data as described in the POST message. After receiving such request, if the SCS/AS is authorized, the SCEF shall negotiate the transfer policy with the PCRF as defined in 3GPP TS 29.154 [9]. After receiving the response including the determined transfer policies from the PCRF, the SCEF shall send an HTTP response to the SCS/AS with a corresponding status code and may include Reference ID and a set of transfer policies. If the SCEF receives a response with an error code from the PCRF, the SCEF shall not update the resource and shall respond to the SCS/AS with 500 Internal Server Error.

NOTE 1: The SCEF starts a new BDT policy negotiation in the Nt interface by sending the request to the PCRF without the previously associated BDT Reference ID.

If more than one policy is included in the HTTP response, the SCS/AS shall send an HTTP PATCH message to inform the SCEF for the "Individual BDT Subscription" resource of the transfer policy selected by the SCS/AS. After receiving the HTTP PATCH message, the SCEF shall send an HTTP response to the SCS/AS with a corresponding status code and interact with the PCRF as defined in 3GPP TS 29.154 [9]. If the SCEF identifies any error (e.g. selected policy is not within the set of transfer policies), the SCEF shall not update the resource and shall respond to the SCS/AS with 500 Internal Server Error.

The SCS/AS may also send an HTTP DELETE message to the SCEF for the "Individual BDT Subscription" resource requesting to remove an individual resource identified by the URI received in the response to the request that has created resource a URI. After receiving such request, the SCEF shall delete the resource and send an HTTP response to the SCS/AS with a corresponding status code.

NOTE 2: The SCEF can also remove the resource when the last window end time in transfer policies expires.

4.4.4 Procedures for changing the chargeable party at session set up or during the session

This procedure is used by an SCS/AS to either request to sponsor the traffic from the beginning or to request becoming the chargeable party at a later point in time via the T8 interface.

When setting up the connection between the AS and UE via the SCEF, the SCS/AS shall send an HTTP POST request to the SCEF for the "Chargeable Party Transactions" resource requesting to become the chargeable party for the session to be set up. The body of the HTTP POST message shall include SCS/AS Identifier, UE IP address, Flow description, Sponsor ID, ASP ID, Sponsoring Status and may include time period and/or traffic volume used for sponsoring. The SCS/AS may also request to activate a previously selected policy of background data transfer by including Reference ID in the body of the HTTP POST message.

After receiving the HTTP POST message, if the authorization performed by the SCEF is successful, the SCEF shall act as an AF to interact with the PCRF via the Rx interface as defined in 3GPP TS 29.214 [10] or 3GPP TS 29.201 [13] to trigger a PCRF initiated IP-CAN Session Modification. The SCEF may map the SCS/AS Identifier to AF Application Identifier and may request to be notified about the traffic plane status (e.g. USAGE_REPORT).

After receiving a successful response from the PCRF, the SCEF shall create a resource "Individual Chargeable Party Transaction", which represents the chargeable party transaction, addressed by a URI that contains the SCS/AS identity and an SCEF-created transaction identifier, and shall respond to the SCS/AS with a 201 Created message, including a Location header field containing the URI for the created resource. The SCS/AS shall use the URI received in the Location header in subsequent requests to the SCEF to refer to this chargeable party transaction. If the SCEF receives a response with an error code from the PCRF, the SCEF shall not create the resource and respond to the SCS/AS with a status code set to 500 Internal Server Error.

In order to update the sponsoring status of an established AS session, the SCS/AS shall send an HTTP PATCH message to the SCEF for the "Individual Chargeable Party Transaction" resource requesting to change Sponsoring Status. When receiving the HTTP PATCH message, the SCEF shall make the change and interact with the PCRF to modify the Rx session as defined in 3GPP TS 29.214 [10] or 3GPP TS 29.201 [13]. After receiving the response with successful result code from the PCRF, the SCEF shall send an HTTP response to the SCS/AS with a 200 OKstatus code and the result in the body of the HTTP response. The accumulated usage received from the PCRF shall be included if the SCS/AS requested to disable the sponsoring. If the SCEF receives a response with an error code from the PCRF, the SCEF shall not update the resource and respond to the SCS/AS with a status code set to 500 Internal Server Error.

If the SCEF receives a traffic plane notification (e.g. the usage threshold is reached or transmission resource lost), or if the SCEF gets informed that the Rx session is terminated (e.g. due to a release of PDN connection), the SCEF shall send an HTTP POST message including the notified event (e.g. session terminated) and the accumulated usage to the SCS/AS. The SCS/AS shall respond with an HTTP response to confirm the received notification.

In order to remove the established AS session, the SCS/AS shall send an HTTP DELETE message to the SCEF for the "Individual Chargeable Party Transaction" resource. After receiving the HTTP DELETE message, the SCEF shall remove all properties of the resource and interact with the PCRF to terminate the Rx session (as defined in 3GPP TS 29.214 [10] or 3GPP TS 29.201 [13]). After receiving the response from the PCRF, the SCEF shall send an HTTP response to the SCS/AS with a 204 No Content status code and the accumulated usage.

4.4.5 Procedures for Non-IP Data Delivery

4.4.5.1 General

This procedure is used by an SCS/AS to support the Non-IP Data Delivery (NIDD) via SCEF. It performs the NIDD configuration via the T8 interface. It also includes the mobile terminated (MT) and mobile originated (MO) communication with UEs via the T8 interface. It also includes the group message delivery via MT NIDD via the T8 interface.

Error handling for the procedures in this subclause shall be handled based on subclause 5.2.6.

4.4.5.2 NIDD Configuration

4.4.5.2.1 NIDD Configuration for a single UE

For a NIDD configuration creation, the SCS/AS shall send an HTTP POST message to the SCEF for the "NIDD configurations" resource. The body of the HTTP POST message shall include External Identifier or MSISDN, SCS/AS Identifier and may include NIDD Duration, PDN Connection Establishment Option and Reliable Data Service Configuration. In addition, the SCS/AS may send non-IP data and its associated parameters (e.g. Priority) as described in subsclause 4.4.5.3.1 in the NIDD configuration creation request. The Reliable Data Service Configuration includes port numbers on UE and SCEF that are used to identify specific applications for data transfer between UE and SCS/AS and an indication if reliable data service acknowledgement is enabled or not.

Upon receipt of the HTTP POST request from the SCS/AS to create a NIDD configuration, the SCEF shall check whether the SCS/AS is authenticated and authorized to create NIDD configuration, and also authorize the NIDD configuration. If authorization is successful, the SCEF shall interact with the HSS via S6t as specified in 3GPP TS 29.336 [11]. Upon receipt of the successful response from the HSS, the SCEF shall store the UE identity (IMSI and External Identifier or MSISDN) which is associated with the External Identifier or MSISDN and create a resource "Individual NIDD configuration", which represents the NIDD configuration, addressed by a URI that contains the SCS/AS identity and an SCEF-created NIDD configuration identifier, and shall respond to the SCS/AS with a 201 Created message, including a Location header field containing the URI for the created resource. The body of the response message shall include Maximum Packet Size and may include Reliable Data Service Indication. When the SCS/AS receives the URI in the Location header, it shall use this URI in subsequent requests to the SCEF to refer to this NIDD configuration.

For a NIDD configuration modification, the SCS/AS shall send an HTTP PATCH message to the SCEF for the "Individual NIDD configuration" resource, using the URI received in the response to the request that has created the NIDD configuration resource. Upon receipt of the HTTP PATCH request from the SCS/AS to update the parameters of the NIDD configuration, the SCEF shall check whether the SCS/AS is authenticated and authorized to update NIDD configuration. If the authorization is successful, the SCEF shall verify that the resource to be modified already exists as identified by the URI. If the NIDD configuration resource is found, the SCEF shall update the NIDD configuration as requested. Upon successful update of the requested NIDD configuration including the interaction with the HSS via S6t as specified in 3GPP TS 29.336 [11], the SCEF shall respond to the SCS/AS with a 200 OK success message indicating that the NIDD configuration resource was successfully updated.

If the SCS/AS includes downlink non-IP data together with the NIDD configuration creation, the SCEF shall also create an "Individual NIDD downlink data delivery" sub-resource before responding the SCS/AS. After sending the HTTP response to NIDD configuration request, the SCEF shall perform the procedure for individual MT NIDD as described in subclause 4.4.5.3.1.

NOTE: Any further interaction with the SCS/AS for the piggybacked individual MT NIDD is performed by the notification of NIDD downlink data delivery status.

For a NIDD configuration cancellation, the SCS/AS shall send an HTTP DELETE message to the SCEF for the "Individual NIDD configuration" resource, using the URI received in the response to the request that has created the

NIDD configuration resource. Upon receipt of the HTTP DELETE message from the SCS/AS, the SCEF shall check whether the SCS/AS is authenticated and authorized to delete NIDD configuration. If the authorization is successful, the SCEF shall verify that the NIDD configuration resource identified by the URI already exists. If the configuration resource exists, the SCEF shall delete the requested configuration, and perform related NIDD procedure to EPC network elements if applicable. Upon successful deletion of requested NIDD configuration including the interaction with the HSS via S6t as specified in 3GPP TS 29.336 [11], the SCEF shall respond to the SCS/AS with a 200 OK success message indicating that the NIDD configuration was successfully cancelled. As an alternative to the 200 OK success message, the SCEF may send a 204 No Content success message without any message content to the SCS/AS.

When the NIDD Duration expires, the SCEF may remove the associated NIDD configuration resource and all individual downlink data delivery resources under such NIDD configuration.

4.4.5.2.2 NIDD Configuration for a group of UEs

The NIDD configuration procedure for a single UE as described in subclause 4.4.5.2.1 shall be applicable for a group of UEs with the following differences:

- The External Group Identifier shall be included in the POST request instead of MSISDN or External Identifier.
- After receiving the response message from the HSS, the SCEF shall store the list of UE identifiers (IMSI and External Identifier or MSISDN) which are associated with the External Group Identifier.
- The downlink non-IP data is not supported to be handled together with the NIDD configuration.

4.4.5.3 Mobile Terminated NIDD procedure

4.4.5.3.1 Mobile Terminated NIDD for a single UE

If the SCS/AS needs to perform a downlink non-IP data delivery for a single UE, the SCS/AS shall send an HTTP POST message to the SCEF identifying an existing NIDD configuration resource. The body of the HTTP POST message shall include External Identifier or MSISDN and non-IP data and may include PDN Connection Establishment Option, Reliable Data Service Configuration, Maximum Latency and Priority. The Reliable Data Service Configuration includes port numbers on UE and SCEF that are used to identify a specific application for data transfer between UE and SCS/AS and an indication if reliable data service acknowledgement is enabled or not.

Upon receipt of a HTTP POST request from the SCS/AS for a downlink data delivery for a single UE, the SCEF shall verify the NIDD configuration resource already exists based on the URI passed, the SCEF shall also check whether the SCS/AS is authorised to send NIDD requests and whether the non-IP packet size is larger than the Maximum Packet Size that was provided to the SCS/AS during NIDD Configuration. If any of those checks fails, the SCEF shall respond a 403 Forbidden response with a cause value indicating the reason for the failure condition. If all checks are successful, the SCEF shall determine the EPS Bearer Context based on the APN associated with the NIDD configuration and the User Identity. If the SCEF EPS bearer context is not found in the SCEF, depending on PDN Connection Establishment Option received in the POST request or from NIDD configuration, the SCEF may:

- reject the request with an error message to the SCS/AS;
- send a Device Trigger to the UE as described in subclause 4.4.6 without buffering the non-IP data and respond the SCS/AS with a 200 OK response; or
- buffer the non-IP data and create the "Individual NIDD downlink data delivery" sub-resource, then send a 201 Created response to the SCS/AS. The response message also includes an indication of whether the Device Trigger procedure (as described in subclause 4.4.6) was performed by the SCEF and Requested Re-Transmission time to indicate the SCS/AS when the UE is expected to be reachable. A Location header shall be included in the response message that provides the URI of the resource identifying this individual downlink data delivery. The SCS/AS shall use the URI received in the Location header in subsequent requests to the SCEF to refer to this individual downlink data delivery for possible replacement or cancellation. The non-IP data shall be delivered when the non-IP PDN connection is established.

If the SCEF EPS bearer context is found in the SCEF, the SCEF shall check if the SCS/AS has exceeded the quota or rate of data submission considering the number of existing buffered non-IP data and restriction in APN and serving PLMN rate control. If quota or rate limit is reached, the SCEF shall respond the SCS/AS with a 403 Forbidden response and a cause value indicating the reason for the failure condition. If the check is passed, the SCEF shall continue the downlink non-IP data delivery procedure as the defined 3GPP TS 29.128 [12].

Based on the delivery result from the MME/SGSN, the SCEF may:

- send a 200 OK response to the HTTP POST request indicating the downlink non-IP data delivery is successful along with the acknowledge information;
- send a 500 Internal Server Error and a cause value indicating the reason for the delivery failure; or
- buffer the non-IP data and and create the "Individual NIDD downlink data delivery" sub-resource, then send a 201 Created response to the SCS/AS if the MME/SGSN indicates UE is temporary not reachable.

If the MT_NIDD_modification_cancellation feature is supported and the SCS/AS decides to replace the pending downlink data delivery in the SCEF, the SCS/AS shall send an HTTP PUT message to the SCEF, using the URI received in the response to the request that has created the individual downlink data delivery resource. The External Identifier or MSISDN shall remain unchanged from previous values. Upon receipt of the HTTP PUT request from the SCS/AS, the SCEF shall check whether a pending non-IP data exists with the same URI (i.e. resource exists). If it is found, the SCEF shall perform the downlink data delivery with the new non-IP data and continue the same procedure as described for the POST request; otherwise the SCEF shall respond with cause value indicating replacement failure. If the buffered data is already delivered, the SCEF shall respond with 404 Not Found and include a cause value indicating replacement failure.

If the MT_NIDD_modification_cancellation feature is supported and the SCS/AS decides to cancel the pending downlink data delivery in the SCEF, the SCS/AS shall send an HTTP DELETE message to the SCEF, using the URI received in the response to the request that has created the individual downlink data delivery resource. Upon receipt of the HTTP DELETE request from the SCS/AS, the SCEF shall check whether a pending request exists with the same URI. If such non-IP data has not been delivered, the SCEF shall remove the individual downlink data delivery resource and respond with an HTTP 204 No Content response; otherwise the SCEF shall respond with 404 Not Found (i.e. data already delivered) or 409 Conflict (i.e. data delivery ongoing) response, and include a cause value indicating cancellation failure.

If a pending non-IP data is delivered by the SCEF (e.g. due to non-IP PDN connection establishment), and the SCEF gets the delivery result from the MME/SGSN, the SCEF shall send an HTTP POST message to notify the delivery result for the pending non-IP data. Upon receipt of the request, the SCS/AS shall acknowledge the notification with an HTTP 200 OK or 204 No Content response.

4.4.5.3.2 Mobile Terminated NIDD for a group of UEs

If the SCS/AS needs to perform a downlink non-IP data delivery to a group of UEs and if both the SCS/AS and the SCEF support GroupMesageDelivery feature as defined in sublcause 5.6.4, the SCS/AS shall send an HTTP POST request message to the SCEF for the "NIDD downlink data deliveries" resource, identifying an existing NIDD configuration resource as parent resource. The body of the HTTP POST request message shall include the External Group Identifier and the non-IP data, and may include Reliable Data Service Configuration, PDN Connection Establishment Option and Maximum Latency.

Upon receipt of such an HTTP POST request from the SCS/AS requesting the group message delivery, the SCEF checks whether the SCS/AS is authorised to send NIDD requests and whether the non-IP packet size is larger than the Maximum Packet Size that was provided to the SCS/AS during NIDD Configuration. If any of those checks fails, the SCEF shall respond a HTTP response with a cause value indicating the reason for the failure condition. If all checks are successful, the SCEF shall create an "Individual NIDD downlink data delivery" resource and sends a 201 Created response to the SCS/AS to acknowledge acceptance of the HTTP POST request.

Then for each authorized External Identifier associated to the External Group Identifier which is retrieved from the HSS during preceding NIDD configuration procedure (as specified in subclause 4.4.5.2.2), the SCEF shall determine the EPS Bearer Context based on the APN associated with the NIDD configuration and the User Identity and continue the procedure as described for MT NIDD for a single UE in subclause 4.4.5.3.1 without sending downlink data delivery status notification for any individual UE to the SCS/AS.

At the end of buffering (duration determined by the Maximum Latency or local policy) or after processing data delivery for all UEs in the group, the SCEF shall send an HTTP POST message to SCS/AS to indicate the aggregated result of data delivery of each UE. The body of the HTTP POST request message shall include MSISDN or External Identifier, Retransmission Time (optional) and delivery result for each UE. Upon receipt of the request, the SCS/AS shall acknowledge the request with an HTTP 200 OK or 204 No Content response.

The MT_NIDD_modification_cancellation feature is not supported for the group message delivery via NIDD. If a PUT or DELETE request is received for the "Individual NIDD downlink data delivery" resource which was created for a group of UEs, the SCEF shall reject the message with 403 Forbidden response.

4.4.5.4 Mobile Originated NIDD procedure

When the SCEF receives the non-IP data from MME/SGSN (or IWK-SCEF) as defined in 3GPP TS 29.128 [12], and finds an SCEF EPS bearer context and the associated NIDD configuration, the SCEF shall determine the SCS/AS by the corresponding NIDD configuration, and send an HTTP POST request to the SCS/AS identified by the Notification Destination Address received in the NIDD configuration to notify the uplink non-IP data. The body of the HTTP POST message shall include External Identifier or MSISDN, non-IP data, NIDD configuration identifier, Reliable Data Service Configuration (if available). The Reliable Data Service Configuration includes port numbers on UE and SCEF that are used to identify a specific application for data transfer between UE and SCS/AS and an indication if reliable data service acknowledgement is enabled or not.

Upon receipt of the request, if the SCS/AS knows the NIDD configuration identified by the NIDD configuration identifier, the SCS/AS shall acknowledge a 200 OK or 204 No Content message to the SCEF.

4.4.5.5 NIDD Authorisation Update procedure

When the SCEF receives a NIDD Authorisation Update Request message from HSS to update a user's NIDD authorisation as defined in 3GPP TS 29.336 [11], the SCEF shall determine the SCS/AS with the corresponding NIDD Configuration, and send an HTTP POST message to the SCS/AS to notify it of the NIDD Authorisation Update. The body of the HTTP POST message shall include External Identifier or MSISDN, NIDD configuration identifier and the NIDD configuration status.

Upon receipt of the request, if the SCS/AS knows the corresponding NIDD configuration, then the SCS/AS shall acknowledge the request with an HTTP 200 OK or 204 No Content response.

If the NIDD configuration is revoked by the HSS within the received NIDD Authorisation Update Request, the SCEF shall release the corresponding T6a/b PDN connection as specified in 3GPP TS 29.128 [12]. In this case, the SCEF shall reject any subsequent MT NIDD deliveries with a 403 Forbidden response. Or 404 Not Found is returned, if the SCEF locally removed the associated NIDD configuration resource when the configuration was revoked.

4.4.6 Procedures for Device Triggering

The procedures are used by the SCS/AS to deliver the device trigger via T8 interface.

In order to create a new device trigger, the SCS/AS shall send an HTTP POST message to the SCEF for the "Device Triggering Transactions" resource. The body of the HTTP POST message shall include the External Identifier or MSISDN, validity period, priority, Application Port ID and trigger payload.

Upon receipt of the corresponding HTTP POST message, the SCEF shall check if the SCS/AS is authorised to send a trigger request and if the SCS/AS has exceeded its quota or rate of trigger submission. The SCEF shall also resolve the External Identifier or MSISDN to IMSI and retrieve the "Routing Information" from HSS for the triggering delivery. If the authorisation check fails, or if the quota or rate of trigger submission was exceeded, or if there is no valid subscription information or if the "Routing Information" cannot be found, then the SCEF shall reject the request with an error message to the SCS/AS. Otherwise, the SCEF shall perform the device trigger procedure over Tsp as defined in 3GPP TS 29.368 [24] and T4 as defined in 3GPP TS 29.337 [25]. Upon completion of this procedure, the SCEF shall create a resource "Individual Device Triggering Transaction" which represents the triggering transaction, addressed by a URI that contains the SCS/AS identity and an SCEF-created transaction identifier, and shall respond to the SCS/AS with a 201 Created message, including the trigger and a Location header field containing the URI for the created resource. The SCS/AS shall use the URI received in the Location header in subsequent requests to the SCEF to refer to this device triggering transaction.

In order to replace an existing device trigger, the SCS/AS shall send an HTTP PUT message to the SCEF for the "Individual Device Triggering Transaction" resource, using the URI received in the response to the request that has created the device triggering transaction resource. The body of the HTTP PUT message shall include External Identifier or MSISDN, validity period, priority, Application Port ID and trigger payload.

After receiving the corresponding HTTP PUT message from the SCS/AS, the SCEF shall check if the SCS/AS is authorised to replace an existing device trigger and if the SCS/AS has not exceeded its quota or rate of trigger submission. If any of these checks fail, then the SCEF shall reject the message with an error. Otherwise, the SCEF shall replace the device triggering with the SMS-SC by performing the device trigger replace procedure over Tsp as defined in 3GPP TS 29.368 [24] and T4 as defined in 3GPP TS 29.337 [25]. Upon completion of this procedure, the SCEF shall send an HTTP response to the SCS/AS to indicate trigger replace success or failure.

In order to recall an existing device trigger, the SCS/AS shall send an HTTP DELETE message to the SCEF for the "Individual Device Triggering Transaction" resource, using the URI received in the response to the request that has created the device triggering transaction resource.

After receiving the corresponding HTTP DELETE message from the SCS/AS, the SCEF shall check if the SCS/AS is authorised to send a recall trigger request and if the SCS/AS has not exceeded its quota or rate of trigger submission. The SCEF shall also check if the device triggering transaction resource referenced by the URI exists. If any of these checks fail, then the SCEF shall reject the message with an error. Otherwise, the SCEF shall recall the device triggering with the SMS-SC by performing the device trigger replace procedure over Tsp as defined in 3GPP TS 29.368 [24] and T4 as defined in 3GPP TS 29.337 [25]. Upon completion of this procedure, the SCEF shall send an HTTP response to the SCS/AS to indicate trigger recall success or failure.

When it receives the Message Delivery Report from the SMS/SC, the SCEF shall send an HTTP POST message to the SCS/AS to report the trigger delivery result. The body of the HTTP POST message shall include the identifier if the transaction and cause. The SCS/AS shall respond with an HTTP 200 OK or 204 No Content response.

4.4.7 Procedures for Group Message Delivery

4.4.7.1 General

This procedure is used by an SCS/AS to deliver a payload to a group of Ues. Two methods of Group Message Delivery via the T8 are specified:

- Group Message Delivery via MBMS which is intended to efficiently distribute the same content to the members of a group that are located in a particular geographical area when MBMS is used. This method further includes two varieties:
  - MB2 is used as southbound interface;
  - xMB is used as southbound interface.
- Group Message Delivery via unicast MT NIDD for UEs which are part of the same External Group Identifier.

NOTE: Group Message Delivery via MT NIDD is defined in subclause 4.4.5.3.2.

Error handling for the procedures in the subsequent subclauses shall be handled based on subclause 5.2.6.

4.4.7.2 Group Message Delivery via MBMS

4.4.7.2.1 General

This procedure is used by an SCS/AS to deliver a payload to a group of UEs via the T8 interface. The SCEF use the Group Message Delivery via MBMS to efficiently distribute the same content to the members of a group that are located in a particular geographical area when MBMS is used.

The procedure of Group message Delivery via MBMS and MB2 used as southbound interface is described in subcaluse 4.4.7.2.2 and the procedure of Group message Delivery via MBMS and xMB used as southbound interface is described in subcaluse 4.4.7.2.3.

4.4.7.2.2 Group Message Delivery via MBMS by MB2

4.4.7.2.2.1 TMGI Allocation

If the SCS/AS acts as a GCS AS in the application level and if there is no assigned TMGI for an External Group Identifier, the SCS/AS shall send an HTTP message to the SCEF to the resource "TMGI Allocation". The body of the HTTP POST request message shall include the External Group Identifier. The SCS/AS may also include the location information in the body.

Upon receipt of the HTTP POST request from the SCS/AS to allocate a TMGI, the SCEF shall check whether the SCS/AS is authorized to request TMGI allocation. If authorization is successful, the SCEF shall initiate TMGI allocation by the BM-SC as defined in subclause 5.2.1 of 3GPP TS 29.468 [36]. Upon successful allocation of a TMGI, the SCEF shall create the resource which represents the TMGI allocation, addressed by a URI that contains the SCS identity and TMGI, and shall respond to the SCS/AS with a 201 Created message including the TMGI and the TMGI expiration.

In order to renew the TMGI, the SCS/AS shall send an HTTP PUT or PATCH message to the SCEF to the resource "Individual TMGI Allocation". Upon receipt of the HTTP PUT or PATCH request from the SCS/AS to renew TMGI, the SCEF shall initiate TMGI renewal by the BM-SC as defined in subclause 5.2.1 of 3GPP TS 29.468 [36]. Upon successful renewal of a TMGI, the SCEF shall update the resource and respond to the SCS/AS by sending an HTTP response with 200 OK including the TMGI and the TMGI expiration.

If the SCEF receives the response with an error code from the BM-SC for the allocation or renewal of TMGI, the SCEF shall not create or update the resource and shall respond to the SCS/AS with a status code set to 500 Internal Server Error.

Upon the TMGI expired, the SCEF may delete the resource of the TMGI locally.

4.4.7.2.2.2 TMGI Deallocation

In order to deallocate the TMGI, the SCS/AS shall send an HTTP DELETE message to the SCEF to the resource "Individual TMGI Allocation". Upon receipt of the HTTP DELETE request from the SCS/AS to deallocate the TMGI, the SCEF shall initiate TMGI deallocation by the BM-SC as defined in subclause 5.2.2 of 3GPP TS 29.468 [36]. Upon successful deallocation of a TMGI, the SCEF shall delete the resource "Individual TMGI Allocation" together with all sub-resouces "GMD via MBMS by MB2" if available,and shall respond to the SCS/AS by sending an HTTP response with 204 No Content.

4.4.7.2.2.3 Creation of group message delivery

If the SCS/AS acts as a GCS AS in the application level and if the SCS/AS has an assigned TMGI for the External Group Identifier, in order to perform the group message delivery, the SCS/AS shall sends an HTTP POST request message to the SCEF to the resource "GMD via MBMS by MB2". The body of the HTTP POST request message shall include the External Group Identifier. The SCS/AS may also include the Group Message Payload, the location information and a Message Delivery Start Time in the body.

The SCS/AS may also send an HTTP POST message to the SCEF directly to the resource "TMGI Allocation" without previously requesting TMGI allocation as defined in subclause 4.4.7.2.2. The SCEF shall create the resource "Individual TMGI Allocation" and perform the procedure as define in subclause 4.4.7.2.2, and shall also create resource "GMD via MBMS by MB2" and perform the procedure as mentioned in this subcaluse for MBMS bearer creation.

Upon receipt of the HTTP POST request from the SCS/AS to deliver the group message, the SCEF shall check whether the SCS/AS is authorized to send a group message request. It also checks to see if the Message Delivery Start Time does not start after the TMGI expiration. If authorization is successful, the SCEF shall initiate the Active MBMS Bearer procedure as defined in subclause 5.3.2 of 3GPP TS 29.468 [36] with the difference that the SCEF acts as a GCS AS. The SCEF shall include the location information based on the local configuration if the location information is not provided in the HTTP POST request message.

Upon successful activation of MBMS bearer, the SCEF shall create resource which represents "Individual GMD via MBMS by MB2", addressed by a URI that contains Transaction Id allocated by the SCEF and respond to the SCS/AS by sending an HTTP response with a 201 Created status code, including a Location header field containing the URI for the created resource. When the SCS/AS receives the URI in the Location header, it shall use this URI in subsequent requests to the SCEF to refer to this active MBMS bearer. If the Group Message Payload was not include in the HTTP POST above, the HTTP response sent from the SCEF shall also include the SCEF message delivery Ipv4 address or Ipv6 address and port number.

If the SCEF receives the response with an error code from the BM-SC for the activation of MBMS bearer, the SCEF shall not create the resource and shall respond to the SCS/AS with a status code set to 500 Internal Server Error.

If the Group Message Payload was included the HTTP POST above, the SCEF shall deliver to BM-SC the Group Message Payload(s) as defined in 3GPP TS 29.468 [36] at Message Delivery Start Time.

If the Group Message Payload was not include in the HTTP POST above, the SCEF shall transfer the contents received from the SCS/AS to the BM-CS at or after the requested Group Message Start Time, but before the TMGI Expiration time. In this case, when the SCEF detects the group message delivery was triggered successful, the SCEF shall send an HTTP POST request message to the SCS/AS.

NOTE: If Group Message Payload was included, then at Message Delivery Start Time, the SCEF delivers to BM-SC the Group Message Payload(s) to corresponding to MB2-U IP address and port number associated with respective TMGI.

4.4.7.2.2.4 Modification of previous submitted group message delivery

If the SCS/AS determines that modification of previous accepted Group Message Delivery Request is required, the SCS/AS shall send an HTTP PATCH or HTTP PUT request message to the SCEF to the resource "Individual GMD via MBMS by MB2". The body of the HTTP PATCH request message shall include the Message Delivery Start Time. The SCS/AS may also include the External Group Identifier, the Group Message Payload and the location information in the body. The body of the HTTP PUT request message shall include the information as the information provided in the HTTP POST in subclause 4.4.7.2.2.2.3. The body of the HTTP PATCH request message shall include the information defined in the data type of GMDViaMBMSByMb2Patch as defined in subclause 5.8.2.1.1.6.

Upon receipt of the HTTP PATCH or HTTP PUT request from the SCS/AS to modify the previous group message delivery subscription, the SCEF shall check whether the SCS/AS is authenticated and authorized to modify the submitted group message delivery. If the authorization is successful, the SCEF shall initiate the Modify MBMS Bearer procedure as defined in subclause 5.3.4 of 3GPP TS 29.468 [36] with the difference that the SCEF acts as a GCS AS. The SCEF shall include the location information based on the local configuration if the location information is not provided in the HTTP PATCH or HTTP PUT request message.

Upon successful modification of MBMS bearer, the SCEF shall update the resource and respond to the SCS/AS with a 200 success message indicating that previous group message delivery subscription is successfully updated.

If the SCEF receives the response with an error code from the BM-SC for the modification of MBMS bearer, the SCEF shall not update the resource and shall respond to the SCS/AS with a status code set to 500 Internal Server Error.

4.4.7.2.2.5 Cancellation of previous submitted group message delivery

If the SCS/AS determines that deletion of previous accepted Group Message Delivery Request is required, the SCS/AS shall send an HTTP DELETE request message to the SCEF.

Upon receipt of the HTTP DELETE request from the SCS/AS to delete the previous group message delivery, the SCEF shall check whether the SCS/AS is authenticated and authorized to delete an existing group message delivery subscription. If the authorization is successful, the SCEF shall initiate the Delete MBMS Bearer procedure as defined in subclause 5.3.3 of 3GPP TS 29.468 [36] with the difference that the SCEF acts as a GCS AS.

Upon successful deletion of MBMS bearer, the SCEF shall respond to the SCS/AS with a message indicating that submitted group message delivery is successfully deleted.

4.4.7.2.3 Group message Delivery via MBMS by xMB

4.4.7.2.3.1 Service Creation

If the SCS/AS acts as a content provider in the application level and if there is no assigned Service ID for an External Group Identifier, the SCS/AS shall send an HTTP POST message to the SCEF to the resource "xMB Services". The body of the HTTP POST request message shall include the External Group Identifier.

Upon receipt of the HTTP POST request from the SCS/AS to create a service, the SCEF shall check whether the SCS/AS is authorized to request service creation. If authorization is successful, the SCEF shall initiate service creation by the BM-SC as defined in subclause 5.2.1.2.2 of 3GPP TS 29.116 [37]. Upon successful service creation, the SCEF shall create the resource which represents the service creation, addressed by a URI that contains the SCS identity and Service Id, and shall respond to the SCS/AS with a 201 Created message which may include the service announcement information.

If the SCEF receives the response with an error status code from the BM-SC for the service creation, the SCEF shall not create or update the resource and shall respond to the SCS/AS with a status code set to 500 Internal Server Error.

4.4.7.2.3.2 Service Deletion

In order to delete the service, the SCS/AS shall send an HTTP DELETE message to the SCEF to the resource "Individual xMB Service". Upon receipt of the HTTP DELETE request from the SCS/AS to delete the service, the SCEF shall initiate service deletion by the BM-SC as defined in subclause 5.2.1.2.4 of 3GPP TS 29.116 [37]. Upon successful deletion of a service, the SCEF shall delete the resource "Individual xMB Service" together with all sub-resouces "GMD via MBMS by xMB" if available, and shall respond to the SCS/AS by sending an HTTP response with 204 No Content.

4.4.7.2.3.3 Creation of group message delivery

If the SCS/AS acts as a content provider in the application level, the SCS/AS may sends an HTTP POST request message to the SCEF to the resource "GMD via MBMS by xMB". The body of the HTTP POST request message shall include the External Group Identifier. The SCS/AS may also include the Group Message Payload, the location information, a Message Delivery Start Time and Message Delivery Stop Time in the body.

Upon receipt of the HTTP POST request from the SCS/AS to deliver the group message, the SCEF shall check whether the SCS/AS is authorized to send a group message request. It also checks to see if the Message Delivery Start Time doesn't start after the Message Delivery Stop Time. If authorization is successful, the SCEF shall initiate the Create Session procedure as defined in subclause 4.4.5.2 of 3GPP TS 29.116 [37] and the Update Session procedure as defined in subclause 4.4.5.3 of 3GPP TS 29.116 [37] with the difference that the SCEF acts as a Content Provider, Session Start is set accordiong to the Message Delivery Start Time and the Session Stop is set according to the Message Delivery Stop Time. The SCEF shall include the location information based on the local configuration if the location information is not provided and include the session type set to "Files" in the HTTP POST request message.

Upon successful activation of MBMS bearer, the SCEF shall create resource which represents "Individual GMD via MBMS by xMB ", addressed by a URI that contains Transaction Id allocated by the SCEF and respond to the SCS/AS by sending an HTTP response with a 201 Created status code, including a Location header field containing the URI for the created resource. When the SCS/AS receives the URI in the Location header, it shall use this URI in subsequent requests to the SCEF to refer to this active MBMS bearer. If the Group Message Payload was not included in the HTTP POST above, the HTTP response sent from the SCEF shall also include the SCEF message delivery Ipv4 address or Ipv6 address and port number.

If the SCEF receives the response with an error code from the BM-SC for the activation of MBMS bearer, the SCEF shall not create the resource and shall respond to the SCS/AS with a status code set to 500 Internal Server Error.

If the Group Message Payload was included the HTTP POST above, the SCEF shall deliver to BM-SC the Group Message Payload(s) as defined in 3GPP TS 29.468 [36] at Message Delivery Start Time.

If the Group Message Payload was not included in the HTTP POST above, the SCEF shall transfer the contents received from the SCS/AS to the BM-CS at or after the requested Message Delivery Start Time, but before the Message Delivery Stop Time. In this case, when the SCEF detects the group message delivery was triggered successful, the SCEF shall send an HTTP POST request message to the SCS/AS.

4.4.7.2.3.4 Modification of previous submitted group message delivery

If the SCS/AS determines that modification of previous accepted Group Message Delivery Request is required, the SCS/AS shall send an HTTP PATCH or HTTP PUT request message to the SCEF to the resource "Individual GMD via MBMS by xMB ". The body of the HTTP PATCH request message shall include the Message Delivery Start Time and Message Delivery Stop Time. The SCS/AS may also include the External Group Identifier, the Group Message Payload and the location information in the body. The body of the HTTP PUT request message shall include the information as the information provided in the HTTP POST in subclause 4.4.7.2.3.3. The body of the HTTP PATCH request message shall include the information defined in the data type of GMDViaMBMSByxMBPatch as defined in subclause 5.8.3.1.1.4.

Upon receipt of the HTTP PATCH or HTTP PUT request from the SCS/AS to modify the previous group message delivery subscription, the SCEF shall check whether the SCS/AS is authenticated and authorized to modify the submitted group message delivery. If the authorization is successful, the SCEF shall initiate the Update Session procedure as defined in subclause 4.4.5.3 of 3GPP TS 29.116 [37] with the difference that the SCEF acts as a Content Provider, Session Start is set accordiong to the Message Delivery Start Time and the Session Stop is set according to the Message Delivery Stop Time. The SCEF shall include the location information based on the local configuration if the location information is not provided in the HTTP PATCH or HTTP PUT request message.

Upon successful modification of MBMS bearer, the SCEF shall respond to the SCS/AS with a 200 success message indicating that previous group message delivery subscription is successfully updated.

If the SCEF receives the response with an error code from the BM-SC for the modification of MBMS bearer, the SCEF shall not update the resource and shall respond to the SCS/AS with a status code set to 500 Internal Server Error.

4.4.7.2.3.5 Cancellation of previous submitted group message delivery

If the SCS/AS determines that deletion of previous accepted Group Message Delivery Request is required, the SCS/AS shall send an HTTP DELETE request message to the SCEF.

Upon receipt of the HTTP DELETE request from the SCS/AS to delete the previous group message delivery, the SCEF shall check whether the SCS/AS is authenticated and authorized to delete an existing group message delivery subscription. If the authorization is successful, the SCEF shall initiate the Delete Session procedure as defined in subclause 4.4.5.4 of 3GPP TS 29.116 [37] with the difference that the SCEF acts as a Content Provider.

Upon successful deletion of MBMS bearer, the SCEF shall respond to the SCS/AS with a message indicating that submitted group message delivery is successfully deleted.

4.4.8 Procedures for Reporting of Network Status

4.4.8.1 General

These procedures are used by an SCS/AS to perform reporting of network status via the T8 interface in one time or continuous reporting cases. The SCEF uses the reporting procedures based on the network status information from one or more RCAF(s). These procedures can also be used by the SCS/AS to indicate the removal of a previously subscribed reporting request.

4.4.8.2 Network Status Reporting Subscription

In order to create a new subscription to request for notification about the network status, the SCS/AS shall send an HTTP POST message to the SCEF for the "Network Status Reporting Subscriptions" resource. The body of HTTP POST message shall include a Notification destination address and Location area, and may include time duration and threshold (s).

Upon receiving the HTTP POST message from the SCS/AS, the SCEF shall check:

- if the SCS/AS is authorized to perform the request. If not the SCEF shall respond to the SCS/AS with a status code set to 401 Unauthorized.
- if the SCS/AS has exceeded its quota or rate of submitting requests. If so the SCEF shall respond to the SCS/AS with a status code set to 403 Forbidden.

After the SCEF authorized the HTTP request message, the SCEF shall create a resource "Individual Network Status Reporting Subscription" which represents the subscription, addressed by a URI that contains the SCS/AS identity and an SCEF-created subscription identifier, and shall respond to the SCS/AS with a 201 Created message, including a Location header field containing the URI for the created resource, to acknowledge to the SCS/AS the successful subscription. The SCS/AS shall use the URI received in the Location header in subsequent requests to the SCEF to refer to this network status reporting subscription. Then, the SCEF shall perform request of network status reporting procedure with the RCAF over Ns interface as defined in 3GPP TS 29.153 [30].

In order to update an existing subscription of continuous network status reporting, the SCS/AS shall send an HTTP PUT message to the SCEF for the "Individual Network Status Reporting Subscription" resource, using the URI received in the response to the request that has created the network status reporting subscription resource. After receiving the HTTP PUT message, the SCEF shall send an HTTP response to the SCS/AS with a 200 OKstatus code and include the result in the body of the HTTP response. Then, the SCEF shall make the change and interact with the RCAF as defined in 3GPP TS 29.153 [30].

NOTE: In order to update an existing subscription, the SCEF needs to send a cancellation to the previously associated RCAF(s) to remove the related SCEF instructions and then send a new request with updated parameters.

In order to remove an existing subscription of continuous network status reporting, the SCS/AS shall send an HTTP DELETE message to the SCEF for the "Individual Network Status Reporting Subscription" resource, using the URI received in the response to the request that has created the network status reporting subscription resource. Upon receipt of the HTTP DELETE message, the SCEF shall send an HTTP response to the SCS/AS with a 204 No Content status code. Then, the SCEF shall interact with the RCAF to terminate the continuous reporting of network status as defined in 3GPP TS 29.153 [30].

4.4.8.3 Network Status Reporting Notification

After receiving reports from all the involved RCAF(s) as defined in 3GPP TS 29.153 [30], the SCEF shall send an HTTP POST message to the SCS/AS using the identified destination URL, which is provided by the SCS/AS during the network status reporting subscription. The body of HTTP POST message shall include the NSI.

4.4.9 Procedures for Communication Pattern Parameters Provisioning

One or more set of CP parameters may be provisioned by the SCS/AS for a single UE or a group of Ues.

In order to create resources for one or more CP parameter set(s), the SCS/AS shall send an HTTP POST message to the SCEF for the "CP provisioning Subscription" resource, including one or more new provisioned CP parameter set(s). The body of HTTP POST message shall include External Identifier or MSISDN for a single UE or External Group ID for a group of Ues, SCS/AS Identifier and one or more set of CP information associated with CP parameter set Id(s). If the ExpectedUMT_5G feature is supported, the expected UE moving trajectory within the "expectedUmts" attribute shall also be included.

After receiving the HTTP POST message, the SCEF shall check if the SCS/AS is authorised. The SCEF may also check if the number of CP parameter sets(s) reaches the limitation based on operator policy or configuration.

After validation, the SCEF shall for each received CP parameter set Id, assign an SCEF Reference ID which may be derived from the CP parameter set Id, and send Update CP Parameter Request message to the HSS for delivering the CP parameter set(s) as specified in 3GPP TS 29.336 [11].

After receiving successful result from the HSS, the SCEF shall create a resource "Individual CP Provisioning Subscription" and all sub-resources "Individual CP set Provisioning" which represents a CP parameter set. The SCEF shall respond to the SCS/AS with a 201 Created including Location header field containing the URI for the created subscription resource "Individual CP Provisioning Subscription" and the sub-resource(s) "Individual CP set Provisioning" each within the "self" attribute in the "cpParameterSet" attribute; otherwise, the SCEF shall not create any resource and shall respond to the SCS/AS with a status code set to 500 Internal Server Error..

In order to add new CP parameter set(s), update and/or remove the existing CP parameter set(s) for one or more CP parameter set Id(s), the SCS/AS may send an HTTP PUT message to the SCEF for the "Individual CP Provisioning Subscription" resource requesting to add new CP parameter set(s) by creating new resource(s), change some created properties (e.g. Validity Time) of the existing resource(s), and/or remove some or entire properties of the existing resource(s). After receiving the HTTP PUT message, the SCEF shall send the CP parameter changes to the HSS as specified in 3GPP TS 29.336 [11]. After receiving the response from the HSS with a successful code, the SCEF shall update the resource(s) and send an HTTP response to the SCS/AS with a 200 OK status code and include a list of CP parameter set Id(s) and result(s) in the body of the HTTP response. Otherwise, the SCEF shall not update the resource and shall send an HTTP response to the SCS/AS with a status code set to 500 Internal Server Error.

The SCS/AS may send a HTTP PUT message to the SCEF for the "Individual CP set Provisioning" resource requesting to replace an individual resource identified by the CP parameter set Id. The body of the HTTP PUT message shall include set of CP information. After receiving such request, the SCEF shall interact with the HSS as specified in 3GPP TS 29.336 [11]. After receiving the response from the HSS with a successful code, the SCEF shall update the resource and send an HTTP response to the SCS/AS with a 200 OK status code; otherwise, the SCEF shall not update the resource and shall send an HTTP response to the SCS/AS with a status code set to 500 Internal Server Error.

The SCS/AS may send an HTTP DELETE message to the SCEF requesting to delete an individual CP set resource "Individual CP set Provisioning". After receiving such request, the SCEF shall determine the SCEF Reference ID for Deletion associated with the CP parameter set Id, and interact with the HSS as specified in 3GPP TS 29.336 [11]. After receiving the response from the HSS, the SCEF shall delete the addressed resource and send an HTTP response to the SCS/AS with a 204 No Content status code.

The SCS/AS may send an HTTP DELETE message to the SCEF requesting to delete an individual subscription resource "Individual CP Provisioning Subscription". After receiving such request, the SCEF shall determine the SCEF Reference ID (s) for Deletion associated with the CP parameter set Id(s) and interact with the HSS as specified in 3GPP TS 29.336 [11]. After receiving the response from the HSS, the SCEF shall delete the addressed resource and its sub-resources addressed by "Individual CP set Provisioning" and send an HTTP response to the SCS/AS with a 204 No Content status code.

4.4.10 Procedures for PFD Management

The PFDs associated with application identifier(s) may be created, updated or removed by the third party SCS/AS as defined in 3GPP TS 23.682 [2].

In order to create PFDs resources for one or more external Application Identifier(s), the SCS/AS shall send an HTTP POST message to the request URL of the resource "PFD Management Transactions" including one or more set of PFDs for external Application Identifier(s). The body of the HTTP POST message shall include external Application Identifier(s) and PFDs associated with its PFD Identifier(s), an Allowed Delay may be included for the external Application Identifier(s) as well.

After receiving the HTTP POST message, if the SCS/AS is authorized, the SCEF shall create an "Individual PFD Management Transaction" resource for the request and one or more "Individual Application PFD Management" sub-resource(s) for each external Application identifier, and shall provision the PFDs to the PFDF as defined in 3GPP TS 29.250 [26]. When receiving the response from the PFDF, the SCEF shall send an HTTP response to the SCS/AS with a corresponding status code. The SCEF shall also include PFD report(s) with a list of external Application Identifier(s) and result(s) in the body of the HTTP response if some application(s) are not provisioned successfully (i.e. the PFDF returns failure and/or the SCEF does not accept the PFD provisioning (e.g. one or more external Application Identifiers in the request are already present in existing transactions)).

In order to update the PFDs for an existing individual transaction, the SCS/AS shall send an HTTP PUT message to URL of the resource "Individual PFD Management Transaction" including one or more set of PFDs for external Application Identifier(s). After receiving the HTTP PUT message, the SCEF shall make the change and send the change to the PFDF (i.e. add/update/remove PFDs) as defined in 3GPP TS 29.250 [26]. After receiving the response from the PFDF, the SCEF shall send an HTTP response to the SCS/AS with a corresponding status code. The SCEF shall also include PFD report(s) with a list of external Application Identifier(s) and result(s) in the body of the HTTP response if some application(s) are not provisioned successfully (i.e. the PFDF returns failure and/or the SCEF does not accept the PFD provisioning (e.g. one or more external Application Identifiers in the request are already present in existing transactions)).

NOTE 1: When the PUT for "Individual PFD Management Transaction" is received in the SCEF, SCEF can use partial update or full update towards the PFDF.

In order to remove the PFDs for an existing individual transaction, the SCS/AS shall send an HTTP DELETE message to the URL of the resource "Individual PFD Management Transaction". After receiving such request, the SCEF shall delete the "Individual PFD Management Transaction" resource and its "Individual Application PFD Management" sub-resouce(s), and shall interact with the PFDF as defined in 3GPP TS 29.250 [26]. After receiving the response from the PFDF, the SCEF shall send an HTTP response to the SCS/AS with a corresponding status code.

For the POST message to the resource "PFD Management Transactions" or the PUT message to the resource "Individual PFD Management Transaction", if the provisioning of all application(s) fails (i.e. the PFDF returns failure and/or the SCEF does not accept the PFD provisioning),, the SCEF shall respond with 500 status code, and include the attribute "pfdReports" with the corresponding failure code as specified in table 5.11.2.2.3-1 and the external Application Identifier(s) for which the provisioning has failed.

In order to update the PFDs for an existing external Application Identifier, the SCS/AS shall send an HTTP PUT message to the resource "Individual Application PFD Management" to update the full set of PFDs of an existing resource. After receiving the HTTP PUT message, the SCEF shall make the change and send the change to the PFDF (i.e. add/update/remove PFDs) as defined in 3GPP TS 29.250 [26]. After receiving the response from the PFDF, the SCEF shall send an HTTP response to the SCS/AS with a corresponding status code. The SCEF shall also include PFD report(s) with a list of external Application Identifier(s) and result(s) in the body of the HTTP response if the PFDF indicates that the PFDs of application(s) cannot be provisioned successfully.

NOTE 2: When the PUT for "Individual Application PFD Management" is received in the SCEF, SCEF can use partial update or full update towards the PFDF.

In order to update the PFDs for an existing external Application Identifier, the SCS/AS may also send an HTTP PATCH message to URL of the resource "Individual Application PFD Management" to partially update PFDs. After receiving the HTTP PATCH message, the SCEF shall make the change and send the change to the PFDF (i.e. add/update/remove PFDs) as defined in 3GPP TS 29.250 [26]. After receiving the response from the PFDF, the SCEF shall send an HTTP response to the SCS/AS with a corresponding status code.

In order to remove the PFDs for an existing individual application, the SCS/AS shall send an HTTP DELETE message to the resource "Individual Application PFD Management". After receiving such request, the SCEF shall delete the resource and interact with the PFDF as defined in 3GPP TS 29.250 [26]. After receiving the response from the PFDF, the SCEF shall send an HTTP response to the SCS/AS with a corresponding status code.

For the PUT/PATCH message to the resource "Individual Application PFD Management", if the provisioning of the application fails (i.e. the PFDF returns failure or the SCEF does not accept the PFD provisioning), the SCEF shall reject the request with a corresponding status code, and include the attribute "pfdReports" with the corresponding failure code as specified in table 5.11.2.2.3-1 and the external Application Identifier for which the provisioning has failed.

4.4.11 Procedures for Enhanced Coverage Restriction Control

The procedures are used by an SCS/AS to query the status of, or to configure the enhanced coverage restriction for a UE via the T8 interface as defined in 3GPP TS 23.682 [2].

In order to query the current status of enhanced coverage restriction, the SCS/AS shall send an HTTP POST message to the SCEF using the query custom operation as defined in subclause 5.12.13.2. The body of the HTTP POST message shall include External Identifier or MSISDN.

In order to configure the enhanced coverage restriction, the SCS/AS shall send an HTTP POST message to the SCEF using the configure custom operation as defined in subclause 5.12.13.3. The body of the HTTP POST message shall include External Identifier or MSISDN and the Enhanced Coverage Restriction setting (i.e. allowed-PLMN-List or restricted-PLMN-List).

Upon receiving the HTTP POST message from the SCS/AS, the SCEF shall check:

- if the SCS/AS is authorized to perform the request. If not the SCEF shall respond to the SCS/AS with a status code set to 401 Unauthorized.
- if the request is malformed. If it is malformed, the SCEF shall respond to the SCS/AS with a status code set to 400 Bad Request.
- if the SCS/AS has exceeded its quota or rate of submitting requests. If so the SCEF shall respond to the SCS/AS with a status code set to 403 Forbidden.

The SCEF shall send a Configuration Information Request to the HSS to query or configure the setting of Enhanced Coverage Restriction as defined in 3GPP TS 29.336 [11].

Upon receipt of the response from the HSS, the SCEF shall send an HTTP response to the SCS/AS with a 200 OK message for query custom operation or a 204 No Content for configure custom operation. For the case of query custom operation, the SCEF shall include the Enhanced Coverage Restriction Data from HSS into the HTTP response.

If the SCEF receives a response with an error code from the HSS, the SCEF shall respond to the SCS/AS with a status code set to 500 Internal Server Error.

4.4.12 Procedures for Network Parameter Configuration

4.4.12.1 General

The procedures are used by an SCS/AS to request that the network consider setting the suggested network parameter values which can influence certain aspects of UE/network behaviour. The procedures are applicable for an individual UE or a group of Ues.

In order to create a new network parameter configuration to configure suggested network parameters, the SCS/AS shall send an HTTP POST request message to the SCEF to the resource "NP Configurations". The body of the HTTP request message shall include External Identifier or MSISDN or External Group Identifier, SCS/AS Identifier, and may include Maximum Latency, Maximum Response Time and Suggested Number of Downlink Packets and Group Reporting Guard Time, wherein, the External Identifier or MSISDN indicates the configuration for an individual UE and the External Group Identifier indicates for a group of Ues. If the External Group Identifier is included, the SCS/AS shall provide the Notification Destination Address in the request.

NOTE: The Notification Delivery as described in subclause 5.2.5 is not supported for individual UE configuration case.

In order to update an existing Network Parameter Configuration, the SCS/AS may send an HTTP PUT message to the resource "Individual NP Configuration" requesting the SCEF to replace all properties in the existing resource.

The SCS/AS may also use an HTTP PATCH message to request to change some properties in the existing resource.

Upon receipt of the HTTP POST, PUT or PATCH message, if the SCS/AS is authorized to perform the request, the SCEF shall check whether the Maximum Latency, Maximum Response Time and/or Suggested Number of Downlink Packets in the HTTP request body are within the range defined by operator policies, if one or more of these parameters are not within the range, the SCEF shall:

- either reject the request message by sending an HTTP response to the SCS/AS with a status code set to "403 Forbidden" and indicate which parameters are out of the range in the "invalidParams" attribute of the "ProblemDetails" structure; or

- modify the parameters which are not within the range by selecting different values which are in the range.

After validation, the SCEF shall perform the Network Parameter Configuration as described in subclause 4.4.12.2 for an individual UE or in subclause 4.4.12.3 for a group of Ues.

In order to delete an existing Network Parameter Configuration at the SCEF, the SCS/AS shall send an HTTP DELETE message to the corresponding resource "Individual NP Configuration" at the SCEF. The SCEF shall determine the SCEF Reference ID for deletion and interact with the HSS via S6t as defined in 3GPP TS 29.336 [11]. Upon receipt of the response from the HSS, the SCEF shall delete active resource "Individual NP Configuration" addressed by the URI and send an HTTP response to the SCS/AS with a "204 No Content" status code.

4.4.12.2 Configuration Request for an individual UE

If the configuration request from the SCS/AS is for an individual UE, the SCEF shall send the Configuration Information Request command to the HSS via S6t as defined in 3GPP TS 29.336 [11].

Upon receipt of the response from the HSS, the SCEF shall,

- for the HTTP POST message, create a new resource "Individual NP Configuration" addressed by a URI that contains the SCS/AS identifier and an SCEF-created configuration identifier, and send an HTTP POST response to the SCS/AS with "201 Created" status code, the final suggested configuration parameter(s) (if modified), the indication(s) for the discarded parameter(s) (if discarded), and a location header field containing the URI for the created resource.

- for the HTTP PUT or PATCH message, update the active resource "Individual NP Configuration", and send an HTTP response to the SCS/AS with "200 OK" status code, the final suggested network parameter(s) (if modified), the indication(s) for the discarded parameter(s) (if discarded).

If the SCEF receives a response with an error code from the HSS, the SCEF shall not create or update the resource and shall respond to the SCS/AS with a status code set to "500 Internal Server Error".

4.4.12.3 Configuration Request for a group of Ues

If the configuration request from the SCS/AS is for a group of Ues, the SCS/AS shall provide the Notification Destination Address, the SCEF shall send the Configuration Information Request command to the HSS via S6t as defined in 3GPP TS 29.336 [11].

Upon receipt of the successful response indicating that group processing is in progress from the HSS before beginning the processing of individual Ues, the SCEF shall,

- for the HTTP POST message, create a resource "Individual NP Configuration" addressed by a URI that contains the SCS/AS identity and an SCEF-created configuration identifier. The SCEF shall send an HTTP POST response to the SCS/AS including a location header field containing the URI for the created resource and a "201 Created" status code to acknowledge the SCS/AS of the successful group processing request.

- for the HTTP PUT or PATCH message, update the resource "Individual NP Configuration" addressed by the requested URL, and shall send "200 OK" status code to acknowledge the SCS/AS of the successful group processing request in the HTTP response message.

If the SCEF receives a response with an error code from the HSS, the SCEF shall not create or update the resource and shall respond to the SCS/AS with a status code set to 500 Internal Server Error.

Upon receipt of the processing result of the individual UEs from the HSS, the SCEF shall perform as follows:

- if no Group Reporting Guard Time is received, the SCEF shall send an HTTP POST request message with a reference to the related network parameter configuration and a list of processing result for the group members to the SCS/AS;
- otherwise, the SCEF shall accumulate all of the configuration results received from the HSS for the group members until the Group Reporting Guard Time expires, and send an HTTP POST request message with a reference to the related network parameter configuration, and a list of processing result for the group members received at the Group Reporting Guard Time.

The SCS/AS shall send an HTTP response to acknowledge the SCEF about the handling result of the received request.

4.4.13 Procedures for setting up an AS session with required QoS

This procedure is used to set up an AS session with required QoS for the service as defined in 3GPP TS 23.682 [2].

For initial AS session creation, the SCS/AS shall send an HTTP POST message to the SCEF for the "AS Session with Required QoS Subscriptions" resource. The body of HTTP POST message shall include SCS/AS Identifier, UE IP address, Flow description, QoS reference and notification destination address. And it may also include time period and/or traffic volume for sponsored data connectivity purpose.

After receiving the HTTP POST message, the SCEF shall authorize the request and may check if the total number of requested QoS reference has exceeded the limit for the SCS/AS. If the authorization is successful, the SCEF shall map the SCS/AS Identifier to AF Application Identifier, and if required, map the SCS/AS Identifier to ASP Identity and Sponsor Identity.

NOTE 1: Before the QoS reference is mapped to Rx parameters, the SCEF can perform a mapping from the name space of the 3rd party SCS/AS to the name space of the operator.

NOTE 2: The QoS reference referring to pre-defined QoS information in the SCEF can be mapped to media component descriptions (e.g. bandwidth, media type) according to SLA.

If the authorization performed by the SCEF is successful, then the SCEF shall act as an AF to interact with the PCRF via the Rx interface as defined in 3GPP TS 29.214 [10] or 3GPP TS 29.201 [13] and trigger a PCRF initiated IP-CAN Session Modification. The SCEF shall also request to be notified about the transmission resource status, i.e. INDICATION_OF_SUCCESSFUL_RESOURCES_ALLOCATION, INDICATION_OF_RELEASE_OF_BEARER, INDICATION_OF_FAILED_RESOURCES_ALLOCATION, and optionally INDICATION_OF_LOSS_OF_BEARER, INDICATION_OF_RECOVERY_OF_BEARER and USAGE_REPORT.

The SCEF, after receiving the AAA message over the Rx interface from the PCRF with successful result code, shall create a resource "Individual AS Session with Required QoS Subscription" which represents AS session, addressed by a URI that contains the SCS/AS identity and an SCEF-created AS session identifier, and shall respond to the SCS/AS with a 201 Created message, including the result in the body of the HTTP response and a Location header field containing the URI for the created resource. The SCS/AS shall use the URI received in the Location header in subsequent requests to the SCEF to refer to this AS session. Otherwise, the SCEF shall send an HTTP response to the SCS/AS with a corresponding status code and include the result in the body of the HTTP response. If the SCEF receives a response with an error code from the PCRF, the SCEF shall not create the resource and respond to the SCS/AS with a status code set to 500 Internal Server Error.

In order to update the established AS session, the SCS/AS may send an HTTP PUT message to the SCEF for the "Individual AS Session with Required QoS Subscription" resource requesting to replace all properties in the existing resource, addressed by the URI received in the response to the request that has created the resource. The UE IP address shall remain unchanged from previously provided values. After receiving such message, the SCEF shall make the change and interact with the PCRF to modify the Rx session (as defined in 3GPP TS 29.214 [10] or 3GPP TS 29.201 [13]). After receiving the response with successful result code from the PCRF, the SCEF shall replace all properties of the existing resource, send an HTTP response to the SCS/AS with a corresponding status code, and include the result in the body of the HTTP response. If the SCEF receives a response with an error code from the PCRF, the SCEF shall not update the resource and respond to the SCS/AS with a status code set to 500 Internal Server Error.

The SCS/AS may also send an HTTP PATCH message to the SCEF for the "Individual AS Session with Required QoS Subscription" resource requesting to change some created properties (e.g. Flow Description). After receiving the HTTP PATCH message, the SCEF shall make the change and interact with the PCRF to modify the Rx session (as defined in 3GPP TS 29.214 [10] or 3GPP TS 29.201 [13]). After receiving the response from the PCRF, the SCEF shall send an HTTP response to the SCS/AS with a corresponding status code and include the result in the body of the HTTP response.

If the SCEF receives a traffic plane notification (e.g. the usage threshold is reached or transmission resource lost), or if the SCEF gets informed that the Rx session is terminated (e.g. due to a release of PDN connection), the SCEF shall send an HTTP POST message including the notified event (e.g. session terminated) and the accumulated usage (if received from the PCRF) to the callback URI "notificationUri" provided by the SCS/AS during the creation of individual AS Session with Required QoS Subscription. The SCS/AS shall respond with an HTTP response to confirm the received notification.

In order to remove the established AS session, the SCS/AS shall send an HTTP DELETE message to the SCEF for the "Individual AS Session with Required QoS Subscription" resource. After receiving the HTTP DELETE message, the SCEF shall remove all properties and interact with the PCRF to terminate the Rx session (as defined in 3GPP TS 29.214 [10] or 3GPP TS 29.201 [13]). After receiving the response from the PCRF, the SCEF shall send an HTTP response to the SCS/AS with a corresponding status code and include the accumulated usage (if received from the PCRF).

4.4.14 Procedures for MSISDN-less Mobile Originated SMS

4.4.14.1 General

The procedures are used by the SCEF to send the MSISDN-less MO-SMS to the SCS/AS via T8 interface.

4.4.14.2 Delivery of MSISDN-less MO SMS

If the SCEF receives an MSISDN-less MO-SMS via T4 including an destination SME address (long/short code of the SCS/AS), the SCEF will use the IMSI of the UE and application port ID received over T4 to query the HSS/HLR for an external ID, and the SCEF shall then determine the notification destination URL of an SCS/AS based on configured information on the mapping of SME addresses to destination URLs. The SCEF shall send to the determined destination URL an HTTP POST request that shall include an MsisdnLessMoSmsNotification data type with:

- the short message transfer protocol data unit as received on the T4 interface.
- the Application Port as received on the T4 interface, and
- the external identifier of the UE that send the SMS, as received from the HSS/HLR.

NOTE: The Notification Delivery using Websocket (see subclause 5.2.5.4) and the Notification Test Event (see subclause 5.2.5.3) are not supported for the present API.

5 T8 APIs

5.1 Introduction

The T8 APIs are a set of APIs defining the related procedures and resources for the interaction between the SCEF and the SCS/AS.

5.2 Information applicable to several APIs

5.2.1 Data Types

5.2.1.1 Introduction

This clause defines structured data types, simple data types and enumerations that are applicable to several APIs defined in the present specification and can be referenced from data structures defined in the subsequent clauses. In addition, data types that are defined in OpenAPI 3.0.0 Specification [27] can also be referenced from data structures defined in the subsequent clauses.

NOTE: As a convention, data types in the present specification are written with an upper-case letter in the beginning. Parameters are written with a lower-case letter in the beginning. As an exception, data types that are also defined in OpenAPI 3.0.0 Specification [27] can use a lower-case case letter in the beginning for consistency.

5.2.1.2 Referenced structured data types

5.2.1.2.1 Type: SponsorInformation

This type represents a sponsor information. It shall comply with the provisions defined in table 5.2.1.2.1-1.

Table 5.2.1.2.1-1: Definition of the SponsorInformation data type

Attribute name	Data type	Cardinality	Description
sponsorId	string	1	It indicates Sponsor ID.
aspId	string	1	It indicates Application Service Provider ID.

5.2.1.2.2 Type: UsageThreshold

This type represents a usage threshold. It shall comply with the provisions defined in table 5.2.1.2.2-1.

Only one of DownlinkVolume, UplinkVolume and TotalVolume shall be provided. If the server supports both duration and volume, then the first threshold that has been reached will apply.

Table 5.2.1.2.2-1: Definition of the UsageThreshold data type

Attribute name	Data type	Cardinality	Description
duration	DurationSec	0..1	Indicates the length of time in seconds
totalVolume	Volume	0..1	Total data octets for both downlink and uplink
downlinkVolume	Volume	0..1	Downlink data octets
uplinkVolume	Volume	0..1	Uplink data octets

5.2.1.2.3 Type: TimeWindow

This type represents a start time and a stop time of a time window. It shall comply with the provisions defined in table 5.2.1.2.3-1.

Table 5.2.1.2.3-1: Definition of the TimeWindow data type

Attribute name	Data type	Cardinality	Description
startTime	DateTime	1	Indicates the absolute start time of the time window
stopTime	DateTime	1	Indicates the absolute stop time of the time window

5.2.1.2.4 Type: Acknowledgement

This type represents a successful acknowledgement for a notification.

Table 5.2.1.2.4-1: Definition of the Acknowledgement data type

Attribute name	Data type	Cardinality	Description
details	string	1	A human-readable explanation specific to this successful acknowledgement

5.2.1.2.5 Type: NotificationData

This type represents the parameters which shall be notify the SCS/AS for bearer level event(s).

Table 5.2.1.2.5-1: Definition of the NotificationData data type

Attribute name	Data type	Cardinality	Description
transaction	Link	1	Link to the transaction resource to which this notification is related.
eventReports	array(EventReport)	1..N	Contains the reported event and applicable information

5.2.1.2.6 Type: EventReport

This type represents an event report. It shall comply with the provisions defined in table 5.2.1.2.6-1.

Table 5.2.1.2.6-1: Definition of the EventReport data type

Attribute name	Data type	Cardinality	Description
event	Event	1	Indicates the event reported by the SCEF.
accumulatedUsage	AccumulatedUsage	0..1	Contains the applicable information corresponding to the event.
flowIds	array(integer)	0..N	Identifies the IP flows that were sent during event subscription

5.2.1.2.7 Type: AccumulatedUsage

This type represents an accumulated usage. It shall comply with the provisions defined in table 5.2.1.2.7-1.

Table 5.2.1.2.7-1: Definition of the AccumulatedUsage data type

Attribute name	Data type	Cardinality	Description
duration	DurationSec	0..1	Indicates the length of time in seconds
totalVolume	Volume	0..1	Total data octets for both downlink and uplink
downlinkVolume	Volume	0..1	Downlink data octets
uplinkVolume	Volume	0..1	Uplink data octets

5.2.1.2.8 Type: FlowInfo

This type represents flow information. It shall comply with the provisions defined in table 5.2.1.2.8-1.

Table 5.2.1.2.8-1: Definition of the FlowInfo data type

Attribute name	Data type	Cardinality	Description
flowId	integer	1	Indicates the IP flow.
flowDescriptions	array(string)	0..2	Indicates the packet filters of the IP flow. Refer to subclause 5.3.8 of 3GPP TS 29.214 [10] for encoding. It shall contain UL and/or DL IP flow description.

5.2.1.2.9 Type: TestNotification

This type represents a notification that can be sent to test whether a chosen notification mechanism works. It shall be supported if the feature "Notification_test_event", as defined for APIs that use notifications, is supported.

Table 5.2.1.2.9-1: Definition of the TestNotification data type

Attribute name	Data type	Cardinality	Description
subscription	Link	1	Link of the subscription resource to which the notification is related.

5.2.1.2.10 Type: WebsockNotifConfig

This type represents configuration for the delivery of notifications over Websockets. It shall be supported if the feature "Notification_websocket", as defined for APIs that use notifications, is supported.

Table 5.2.1.2.10-1: Definition of the WebsockNotifConfig data type

Attribute name	Data type	Cardinality	Description
websocketUri	Link	0..1	Set by the SCEF to indicate to the SCS/AS the Websocket URI to be used for delivering notifications. (NOTE 1)
requestWebsocketUri	boolean	0..1	Set by the SCS/AS to indicate that the Websocket delivery is requested. (NOTE 2)
NOTE 1: A Websocket URI should use the scheme "wss" (Websocket Secure) for encrypted delivery and may use the scheme "ws" (Websocket) for unencrypted delivery. If the WebsockNotifConfig data type is used in an HTTP response, this attribute shall be present. If the WebsockNotifConfig data type is used in an HTTP request, this attribute shall not be set by the SCS/AS in a request to create a resource, and shall not be modified by the SCS/AS in a request to modify a resource. NOTE 2: In a request to create or update a resource, this attribute shall be set to true by the SCS/AS to request the SCEF to provide a Websocket URI for the delivery of notifications, and shall be absent otherwise. In any HTTP response, this attribute shall retain the value that was provided upon resource creation or update.			

5.2.1.2.11 Type: LocationArea

This data type represents the user location area which is sent from the SCS/AS to the SCEF.

Table 5.2.1.2.11-1: Definition of the LocationArea data Type

Attribute name	Data type	Cardinality	Description
cellId	array(string)	0..N	Indicates a Cell Global Identification of the user which identifies the cell the UE is registered.
enodeBId	array(string)	0..N	Indicates an eNodeB in which the UE is currently located.
routingAreaId	array(string)	0..N	Identifies a Routing Area Identity of the user where the UE is located.
trackingAreaId	array(string)	0..N	Identifies a Tracking Area Identity of the user where the UE is located.
geographicArea	array(GeographicArea)	0..N	Identifies a geographic area of the user where the UE is located.
civicAddress	array(CivicAddress)	0..N	Identifies a civic address of the user where the UE is located.

5.2.1.2.12 Type: ProblemDetails

Table 5.2.1.2.12-1: Definition of the ProblemDetails data type

Attribute name	Data type	Cardinality	Description
type	Uri	0..1	A URI reference according to IETF RFC 3986 [6] that identifies the problem type.
Title	string	0..1	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem.
Status	integer	0..1	The HTTP status code for this occurrence of the problem.
Detail	string	0..1	A human-readable explanation specific to this occurrence of the problem.
Instance	Uri	0..1	A URI reference that identifies the specific occurrence of the problem.
cause	string	0..1	A machine-readable application error cause specific to this occurrence of the problem This IE should be present and provide application-related error information, if available.
invalidParams	array(InvalidParam)	0..N	Description of invalid parameters, for a request rejected due to invalid parameters.
NOTE 1: See IETF RFC 7807 [8] for detailed information and guidance for each attribute. NOTE 2: Additional attributes may be defined per API.			

5.2.1.2.13 Type: InvalidParam

Table 5.2.1.2.13-1: Definition of the InvalidParam data type

Attribute name	Data type	Cardinality	Description
param	string	1	Attribute's name encoded as a JSON Pointer, or header's name.
reason	string	0..1	A human-readable reason, e.g. "must be a positive integer".

5.2.1.2.14 Type: PlmnId

Table 5.2.1.2.14-1: Definition of the PlmnId data type

Attribute name	Data type	Cardinality	Description
mcc	Mcc	1	Mobile Country Code
mnc	Mnc	1	Mobile Network Code

5.2.1.2.15 Type: ConfigResult

This type represents one configuration processing result for the group members.

Table 5.2.1.2.15-1: Definition of the ConfigResult data type

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
externalIds	array(ExternalId)	0..N	Each element indicates an external identifier of the UE. (NOTE 2)	
msisdns	array(Msisdn)	0..N	Each element identifies the MS internal PSTN/ISDN number allocated for the UE (NOTE 2)	
resultReason	ResultReason	1	Identifies the configuration failure reason for the group members.	
NOTE 1: Properties marked with a feature as defined in subclause 5.13.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features. NOTE 2: Either "externalId" or "msisdn" shall be included for a group member.				

5.2.1.2.16 Type: UsageThresholdRm

This type represents a usage threshold which is defined in subclause 5.2.1.2.2 but defined with "nullable: true" property so it can be removed in "JSON Merge Patch", as defined in IETF RFC 7386 [39]. It shall comply with the provisions defined in table 5.2.1.2.16-1.

Only one of "downlinkVolume", "uplinkVolume" and "totalVolume" shall be provided. Duration and volume are also removable in "JSON Merge Patch". If the server supports both duration and volume, then the first threshold that has been reached will apply.

Table 5.2.1.2.16-1: Definition of the UsageThresholdRm data type

Attribute name	Data type	Cardinality	Description
duration	DurationSecRm	0..1	Indicates the length of time in seconds
totalVolume	VolumeRm	0..1	Total data octets for both downlink and uplink
downlinkVolume	VolumeRm	0..1	Downlink data octets
uplinkVolume	VolumeRm	0..1	Uplink data octets

5.2.1.3 Referenced Simple data types and enumerations

5.2.1.3.1 Introduction

This clause defines simple data types and enumerations that are referenced from data structures.

5.2.1.3.2 Simple data types

The reused datatypes defined in OpenAPI Specification [27] listed in table 5.2.1.3.2-1 and the simple data types defined in table 5.2.1.3.2-2 apply to several T8 APIs.

Table 5.2.1.3.2-1: Reused OpenAPI data types

Type name	Description
boolean	As defined in OpenAPI Specification [27], i.e. either value "true" or value "false" as defined in IETF RFC 7159 [5].
Integer	As defined in OpenAPI Specification [27].
Number	As defined in OpenAPI Specification [27].
String	As defined in OpenAPI Specification [27].
NOTE: Data type names defined in OpenAPI Specification [27] do not follow the convention to start with capital letters otherwise used in this specification.	

Table 5.2.1.3.2-2: Simple data types applicable to several APIs

Type name	Description
Bandwidth	integer indicating a bandwidth in bits per second.
BdtReferenceId	string identifying a BDT Reference ID as defined in subclause 5.3.3 of 3GPP TS 29.154 [9].
Binary	string with format "binary" as defined in OpenAPI Specification [27].
Bytes	String with format "byte" as defined in OpenAPI Specification [27], i.e, base64-encoded characters,
DayOfWeek	integer between and including 1 and 7 denoting a weekday. "1" shall indicate "Monday", and the subsequent weekdays shall be indicated with the next higher numbers. "7" shall indicate "Sunday".
DateTime	string with format "date-time" as defined in OpenAPI Specification [27].
DateTimeRm	string with format "date-time" as defined in OpenAPI [27] with "nullable: true" property.
DurationSec	Unsigned integer identifying a period of time in units of seconds.
DurationSecRm	Unsigned integer identifying a period of time in units of seconds with "nullable: true" property.
DurationSecRo	Unsigned integer identifying a period of time in units of seconds with "readonly: true" property.
DurationMin	Unsigned integer identifying a period of time in units of minutes.
ExternalId	string containing a local identifier followed by "@" and a domain identifier. Both the local identifier and the domain identifier shall be encoded as strings that do not contain any "@" characters. See Clause 4.6.2 of 3GPP TS 23.682 [2] for more information.
ExternalGroupId	string containing a local identifier followed by "@" and a domain identifier. Both the local identifier and the domain identifier shall be encoded as strings that do not contain any "@" characters. See Clauses 4.6.2 and 4.6.3 of 3GPP TS 23.682 [2] for more information.
Ipv4Addr	string identifying a Ipv4 address formatted in the "dotted decimal" notation as defined in IETF RFC 1166 [28].
Ipv6Addr	string identifying a Ipv6 address formatted according to clause 4 in IETF RFC 5952 [29]. The mixed Ipv4 Ipv6 notation according to clause 5 of IETF RFC 5952 [29] shall not be used.
Link	string formatted according to IETF RFC 3986 [7] identifying a referenced resource.
Mcc	String encoding a Mobile Country Code part of the PLMN, comprising 3 digits, as defined in 3GPP TS 38.413 [11].
Mnc	String encoding a Mobile Network Code part of the PLMN, comprising 2 or 3 digits, as defined in 3GPP TS 38.413 [11].
Msisdn	string formatted according to subclause 3.3 of 3GPP TS 23.003 [14] that describes an MSISDN.
Port	Unsigned integer with valid values between 0 and 65535.
ResourceId	string chosen by the SCEF to serve as identifier in a resource URI.
ScsAsId	string that identifies an SCS/AS.
TimeOfDay	String with format "partial-time" or "full-time" as defined in subclause 5.6 of IETF RFC 3339 [15]. Examples: "20:15:00", "20:15:00-08:00" (for 8 hours behind UTC).
Uri	string providing an URI formatted according to IETF RFC 3986 [7].
Volume	Unsigned integer identifying a volume in units of bytes.
VolumeRm	Unsigned integer identifying a volume in units of bytes with "nullable: true" property.

5.2.1.3.3 Enumeration: Event

The enumeration Event represents event reported by the SCEF.

Table 5.2.1.3.3-1: Enumeration Event

Enumeration value	Description
SESSION_TERMINATION	Indicates that Rx session is terminated.
LOSS_OF_BEARER	Indicates a loss of a bearer.
RECOVERY_OF_BEARER	Indicates a recovery of a bearer.
RELEASE_OF_BEARER	Indicates a release of a bearer.
USAGE_REPORT	Indicates the usage report event.

5.2.1.3.4 Enumeration: ResultReason

The enumeration ResultReason represents a failure result reason.

Table 5.2.1.3.4-1: Enumeration ResultReason

Enumeration value	Description
ROAMING_NOT_ALLOWED	Identifies the configuration parameters are not allowed by roaming agreement.
OTHER_REASON	Identifies the configuration parameters are not configured due to other reason.

5.2.1.4 Conventions for documenting structured data types

The structured data types shall represent an object (see IETF RFC 8259 [40]). The structured data types shall contain attributes that are simple data types, structured data types, arrays (see below), maps (see below) or enumerations.

An array (see IETF RFC 8259 [40]) shall represent a list of values without keys and with significance in the order of sequence. All values shall be of the same type.

A map shall represent an object (see IETF RFC 8259 [40]) with a list of key-value pairs (with no significance in the order of sequence), where all keys are of type string and shall be unique identifiers assigned by the application rather than by the schema, and where all values shall be of the same type.

NOTE: Maps are supported by the OpenAPI specification [27] as described at https://swagger.io/docs/specification/data-models/dictionaries/. Maps can enable a faster lookup of elements identified by some key in huge data structures compared to arrays that contain the key within the elements. Maps can also be used instead of arrays to modify individual elements when modification instructions of the PATCH method are compliant to IETF RFC 7396 [39].

Each structured data type shall be specified in a separate subclause as illustrated in table 5.2.1.4-1.

Table 5.2.1.4-1: Definition of type <Data type>

Attribute name	Data type	Cardinality	Description
<*attribute name*>	"<*type*>" or "array*(<type*>)" or "map*(<type*>)"	"0..1", "1", or "*M..N*",	<only if applicable>

Attribute name: Name of attributes that belong to the specified data type.

Data type: Data type of the attribute. If the data type is indicated as "<*type*>", the attribute shall be of data type <*type*>. If the data type is indicated as "array(<*type*>)", the attribute shall be an array (see IETF RFC 8259 [40]) that contains elements of data type <*type*>. If the data type is indicated as "map (<*type*>)", the attribute shall be an object (see IETF RFC 8259 [40]) encoding a map (see OpenAPI specification [27]) that contains as values elements of data type <*type*>. <*type*> can either be "integer", "number", "string" or "boolean" (as defined in the OpenAPI specification [27]), or a data type defined in a 3GPP specification.

Cardinality: Defines the allowed number of occurrence of data type <type>. A cardinality of "M..N", is only allowed for data types "array(<type>)" and "map(<type>)" and indicates the number of elements within the array or map; the values M and N can either be the characters "M" and "N", respectively, or integer numbers with M being greater than or equal 0, and N being greater than 1 and M, For data type "<type>", the cardinality shall be set to "0..1" or "1". A lower boundary of "0" for the cardinality also indicates that the attribute is optional. A lower boundary of "0" for the cardinality indicates that the attribute is optional.

Description: Describes the meaning and use of the attribute and may contain normative statements.

5.2.2 Usage of HTTP

5.2.2.1 General

For T8 APIs, support of HTTP/1.1 (IETF RFC 7230 [16], IETF RFC 7231 [17], IETF RFC 7232 [18], IETF RFC 7233 [19], IETF RFC 7234 [20] and IETF RFC 7235 [21]) over TLS (IETF RFC 5246 [4]) is mandatory and support of HTTP/2 (IETF RFC 7540 [22]) over TLS (IETF RFC 5246 [4]) is recommended. An SCS/AS desiring to use HTTP/2 shall use the HTTP upgrade mechanism to negotiate applicable HTTP version as described in IETF RFC 7540 [22].

5.2.2.2 Usage of the HTTP PATCH method

The HTTP PATCH method, as defined in IETF RFC 5789 [38], allows for a partial update of previously sent data, e.g. resources. For a complete replacement of previously sent data, the HTTP PUT method is used. It is defined separately for each resource whether the HTTP PUT and/or the HTTP PATCH are applicable.

If the HTTP PATCH method is used, the JSON bodies within the PATCH request shall be encoded according to "JSON Merge Patch", as defined in IETF RFC 7386 [39].

NOTE: JSON merge patch is not able to modify individual elements of an array. This is not required in the present release.

5.2.3 Content type

The bodies of HTTP request and successful HTTP responses shall be encoded in JSON format (see IETF RFC 7159 [5]).

The MIME media type that shall be used within the related Content-Type header field is "application/json", as defined in IETF RFC 7159 [5].

NOTE: This release only supports the content type JSON.

5.2.4 URI structure

All resource URIs of T8 APIs should have the following root structure:

{apiRoot}/{apiName}/{apiVersion}/

"apiRoot" is configured by means outside the scope of the present document. "apiName" and "apiVersion" shall be set dependent on the API, as defined in the corresponding subclauses below. All resource URIs in the subclauses below are defined relative to the above root URI.

NOTE 1: The "apiVersion" will only be increased if the new API version contains not backward compatible changes. Otherwise, the supported feature mechanism defined in subclause 5.2.7 can instead be used to negotiate extensions.

NOTE 2: A different root structure can be used when the Resource URI is preconfigured in the SCS/AS.

The root structure may be followed by "apiSpecificSuffixes" that are dependent on the API and are defined separately for each API where they apply:

{apiRoot}/{apiName}/{apiVersion}/{apiSpecificSuffixes}

5.2.5 Notifications

5.2.5.1 General

The SCEF and SCS/AS shall support the delivery of Notifications using a separate HTTP connection towards an address as assigned the SCS/AS described in subclause 5.2.5.2.

An SCEF and SCS/AS may support testing a notification connection as described in subclause 5.2.5.3. An SCEF and SCS/AS may support the delivery of Notification using Websocket (IETF RFC 6455 [32]) as described in subclause 5.2.5.4.

5.2.5.2 Notification Delivery using a separate HTTP connection

If a delivery of notifications is required for an API, the SCS/AS shall provide a URI in the "notificationDestination" attribute defined in the data types that are passed in a request to create a resource that represents a subscription to notifications, designating where to send HTTP Notifications to the SCEF unless it is specified for that API that a preconfigured destination address is used.

The SCS/AS may provide the same "notificationDestination" for several subscriptions, and the SCEF should then use the same HTTP connection to deliver related notifications.

The SCEF shall take the role of the HTTP client on the HTTP connection for the delivery of Notifications. Subclause 5.2.2 shall also apply for this HTTP connection with the exception that an SCEF (rather than an SCS/AS) desiring to use HTTP/2 shall use the HTTP upgrade mechanism to negotiate applicable HTTP version.

5.2.5.3 Notification Test Event

If the optional "Notification_test_event" feature is supported, the SCS/AS may test whether notifications can be received by subscribing to the notification of a test event by providing a "requestTestNotification" attribute set to "true" in the HTTP request to create or update a subscription for notifications. In any other HTTP request or response, this attribute shall retain the value that was provided upon subscription resource creation.

Upon receiving the "requestTestNotification" attribute as part of a subscription creation or update request, the SCEF shall send immediately after establishing the notification delivery mechanism a test notification containing a body formatted according to the "TestNotification" data type as defined in subclause 5.2.1.2.12. If the SCS/AS does not receive the test notification within a configured time, the SCS/AS knows that the notification delivery with the selected method is not possible and may take corrective actions.

5.2.5.4 Notification Delivery using Websocket

The procedures in the present subclause only apply if SCS/AS and SCEF support the "Notification_websocket" featuture. If the feature "Notification_websocket", is supported, the feature "Notification_test_event" shall also be supported.

If a delivery of notifications is required for an API and the SCS/AS does not know from previous interactions with the SCEF whether delivery of notifications over a separate HTTP connection works, the SCS/AS should initially request the SCEF to try to establish a separate HTTP connection for notification delivery according to subclause 5.2.5.2 by providing a URI to the SCEF designating where to send HTTP Notifications, and shall also subscribe to the notification of a test event as in to subclause 5.2.5.3.

If the SCS/AS does not receive the requested notification of the test event during a configured period after the subscription, the SCS/AS may configure the subscription to request the SCEF to provide a URI for an HTTP connection to upgrade to Websocket, setting the "requestWebsocketUri" attribute to "true" as specified in subclause 5.2.1.2.13. The SCS/AS may also request the SCEF to provide a URI in a new subscription creation request, and should in this case terminate the original subscription.

NOTE 1: If the SCS/AS has requested the delivery of notifications to a separate entity, it needs to be informed by that separate entity about the receipt of the test notification. That communication between the separate entity and the SCS/AS is out of scope of the present document.

When the SCEF receives a subscription creation or update request to use Websockets to deliver notifications (i.e. with the "requestWebsocketUri" attribute set to "true"), it shall assign a Websocket URI where to receive a Websocket connection establishment and shall provide this URI in the "websocketUri" attribute in the response as defined in subclause 5.1.2.1.13. Once such a Websocket URI has been assigned for a particular subscription resource, subsequent update requests to this resource that ask for the assignment of a new Websocket URI for that subscription shall be rejected by the SCEF.

Upon the reception of the Websocket URI from the SCEF in the "websocketUri" attribute as specified in subclause 5.2.1.2.13-1 in the subscription creation or subscription update response, the SCS/AS or a separate entity that is intended to receive the notification shall establish an HTTP connection towards that URI and shall upgrade that connection to the Websocket protocol (IETF RFC 6455 [32]) using the HTTP upgrade mechanism defined in IETF RFC 7230 [16].

NOTE 2: For delivery of Notifications to a separate entity, the SCS/AS needs to provide the Websocket URI to that separate entity. That communication between the SCS/AS and the separate entity is out of scope of the present document.

The following framing of the request and response shall be used when delivering a notification or acknowledging its delivery through Websockets.

NOTE 3: The framing is aligned as much as possible with HTTP delivery in order to simplify implementations.

To deliver a notification towards the SCS/AS, the SCEF shall embed the following structure in a separate Websocket data frame with 0x2 (Binary) opcode in the following order:

1) The string "3GPP-WS-Notif-Seq:", followed by a blank, followed by a four-byte sequence number, encoded as decimal number in ASCII, followed by CRLF

2) The following HTTP headers in any order, with the syntax and semantics as defined in IETF RFC 7230 [16] and IETF RFC 7231 [17]: Content-Type (mandatory), Content-Encoding (optional), Content-Length (mandatory). Every HTTP header line shall be ended by CRLF.

3) CRLF to end the headers section.

4) The payload body of the notification, as defined in the individual APIs.

NOTE 4: The payload body is the same as the one that would be used if delivering the notification as defined in subclause 5.2.5.3.

To acknowledge the reception of a notification message towards the SCEF, the SCS/AS shall embed the following structure in a separate Websocket data frame with 0x2 (Binary) opcode in the following order:

1) The string "3GPP-WS-Notif-Seq:", followed by a blank, followed by the four-byte sequence number of the notification to be confirmed, encoded as decimal number in ASCII, followed by CRLF.

2) The HTTP status code (e.g. 204) and status message (e.g. No Content) as defined for HTTP delivery of the notification in the individual APIs, separated by a single blank character, and ended by CRLF.

3) Conditionally, as defined in IETF RFC 7230 [16] and IETF RFC 7231 [17], the following HTTP headers in any order: Content-Type, Content-Encoding, and Content-Length. Every HTTP header line shall be ended by CRLF.

4) CRLF to end the headers section.5) The payload body of the response, if applicable based on the status code and the HTTP headers, as defined in IETF RFC 7230 [16] and IETF RFC 7231 [17].

NOTE 5: The status code, the status message and the payload body (if applicable), are the same as if delivering the notification as defined in subclause 5.2.5.3.

Use of CRLF is defined in IETF RFC 7230 [16].

The SCEF need not wait for the confirmation of each notification before delivering the next notification. The SCEF shall determine whether a notification has been delivered successfully by correlating the sent notification with the received acknowledgement by checking the sequence numbers of both for equality. The SCEF may re-send a notification, using the same sequence number, if it has not received an acknowledgement with a matching sequence number after a configurable time-out. The SCS/AS shall consider notifications with the same sequence number that arrive within a configurable time interval as duplicates.

The SCS/AS should send periodic Websocket "PING" frames to keep the connection alive.

NOTE 6: the TCP layer will handle a possible fragmentation and reassembly of large messages.

The security related clause 6 shall also apply for the HTTP connection that is upgraded to Websocket.

5.2.6 Error handling

Table 5.2.6-1 lists response bodies that are applicable to all APIs and as responses for all requests in the present specification unless otherwise specified.

Table 5.2.6-1: Response bodies supported for responses to all requests.

	Data type	Cardinality	Response Codes (NOTE 1)	Remarks (NOTE 2)	Applied Methods
Response body	ProblemDetails	0..1	400 Bad Request	Incorrect parameters were passed in the request. More information may be provided in the "detail" attribute of the "ProblemDetails" structure.	GET, POST PUT, PATCH, DELETE
	ProblemDetails	0..1	401 Unauthorized	The client is not authorized as described in IETF RFC 7235 [21]. More information may be provided in the "detail" attribute of the "ProblemDetails" structure.	GET, POST, PUT, PATCH, DELETE
	ProblemDetails	0..1	403 Forbidden	This represents the case when the server is able to understand the request but unable to fulfil the request due to errors (e.g. the requested parameters are out of range). More information may be provided in the "detail" attribute and/or the "invalidParams" attribute of the "ProblemDetails" structure. (NOTE 3)	GET, POST, PUT, PATCH, DELETE
	ProblemDetails	0..1	404 Not Found	The resource URI was incorrect, for instance because of a wrong "scsAsId" field. More information may be provided in the "detail" attribute of the "ProblemDetails" structure.	GET, PUT, PATCH, DELETE
	ProblemDetails	0..1	409 Conflict	The operation is not allowed given the current status of the resource. This response shall be used when the creation of a resource with an identifier is requested and a resource with the same such identifiers already exists in the server and has not been deleted; or the change cannot be made due to conflict with current state of the resource. More information may be provided in the "detail" attribute of the "ProblemDetails" structure.	POST, PUT, PATCH
	ProblemDetails	0..1	411 Length Required	The 411 (Length Required) status code indicates that the server refuses to accept the request without a Content-Length header field. More information may be provided in the "detail" attribute of the "ProblemDetails" structure.	POST, PUT, PATCH
	ProblemDetails	0..1	412 Precondition Failed	Can be used according to IETF RFC 7232 [18]. Also used for the supported feature mechanism as described in subclause 5.2.7. More information may be provided in the "detail" attribute of the "ProblemDetails" structure.	GET, POST, PUT, PATCH, DELETE
	ProblemDetails	0..1	413 Payload Too Large	If the received HTTP request contains payload body larger than the server is able to process, the NF shall reject the HTTP request with the HTTP status code "413 Payload Too Large". More information may be provided in the "detail" attribute of the "ProblemDetails" structure.	POST, PUT, PATCH
	ProblemDetails	0..1	500 Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request. More information may be provided in the "detail" attribute of the "ProblemDetails" structure.	GET, POST, PUT, PATCH, DELETE

	ProblemDetails	0..1	503 Service Unavailable	The server is unable to handle the request. More information may be provided in the "detail" attribute of the "ProblemDetails" structure.	GET, POST, PUT, PATCH, DELETE
NOTE 1: In addition to the above response codes, the SCEF can also send other valid HTTP response codes, if applicable. The list of all valid HTTP response codes can be found in HTTP Status Code Registry at IANA [6]. NOTE 2: The MIME media type that shall be used within the related Content-Type header field is "application/problem+json", as defined in IETF RFC 7807 [8]. NOTE 3: The information about which provided parameters are out of range shall be provided in the "invalidParams" attribute of the "ProblemDetails" structure for the API of network parameter configuration.					

5.2.7 Feature negotiation

The procedures in subclause 6.6.2 of 3GPP TS 29.500 [44] shall be applicable for the APIs defined in the present specification.

The supported features are negotiated seperately for each API. For each of the APIs defined in the present specification, the applicable list of features is contained in the related API definition.

5.2.8 HTTP custom headers

5.2.8.1 General

This subclause lists reused HTTP custom headers and defines any new HTTP custom headers introduced by this specification.

5.2.8.2 Reused HTTP custom headers

Table 5.2.8.2-1: Reused HTTP custom headers

Name	Reference	Description

5.2.9 Conventions for Open API specification files

5.2.9.1 General

T8 Open API specification files shall comply with the OpenAPI specification [27] and with the present subclause.

Each API shall be described in one Open API specification file. In addition, 3GPP specifications may contain Open API specification file with common data types.

For the purpose of referencing (see subclause 5.2.9.6), it is assumed that each Open API specification file contained in a 3GPP specification is stored as separate physical, that all Open API specification files are stored in the same directory on the local server, and that the files are named according to the conventions in subclause 5.2.9.6.

5.2.9.2 Formatting of OpenAPI files

The following guidelines shall be used when documenting OpenAPI files:

- OpenAPI specifications shall be documented using YAML format (see YAML 1.2 [41]). For specific restrictions on the usage of YAML in OpenAPI, see OpenAPI 3.0.0 Specification [27].
- The style used for the specification shall be "PL" (Programming Language).
- The different scopes in the YAML data structures representing collections (objects, arrays…) shall use an indentation of two white spaces.
- Comments may be added by following the standard YAML syntax ("#").

5.2.9.3 Structured data types

The OpenAPI file shall contain a definition in the components/schemas section defining a schema with the name of the structured data type as key.

The schema shall contain:

- "type: object";
- If any attributes in the structured data type are marked as mandatory via a minimum cardinality greater than "0", a "required" keyword listing those attributes;
- A "properties" keyword containing for each attribute in the structured data type an entry with the attribute name as key and;

  1. if the data type is "<*type*>"

     a. if the data type of the attribute is "string", "number", "integer", or "boolean", a type definition using that data type as value ("type: <*data type*>"); or

     b. otherwise a reference to the data type schema for the data type <*data type*> of the attribute, i.e. "$ref: '#/components/schemas/<*data type*>'" if that data type schema is contained in the same OpenAPI file and "$ref: '<*filename*>#/components/schemas/<*data type*>'" if that data type schema is contained in file <*filename*> in the same directory on the same server.

  2. if the data type is "array*(<type>*)":

     a. a type definition "type: array";

     b. an "items:" definition containing

        i). if the data type of the attribute is "string", "number", "integer", or "boolean", a type definition using that data type as value ("type: <*data type*>"); or

        ii). Otherwise a reference to the data type schema for the data type <*data type*> of the attribute, i.e. "$ref: '#/components/schemas/<*data type*>'" if that data type schema is contained in the same OpenAPI file and "$ref: '<*filename*>#/components/schemas/<*data type*>'" if that data type schema is contained in file <*filename*> in the same directory on the same server.

     c. if the cardinality contained an integer value <*m*> as lower boundary, "minItems: <*m*>"; and

     d. if the cardinality contained an integer value <*n*> as upper boundary, "maxItems: <*n*>";

  3. if the data type is "map*(<type>*)";

     a. a type definition "type: object"; and

     b. an "additionalProperties:" definition containing

        i). if the data type of the attribute is "string", "number", "integer", or "boolean", a type definition using that data type as value ("type: <*data type*>"); or

        ii). Otherwise a reference to the data type schema for the data type <*data type*> of the attribute, i.e. "$ref: '#/components/schemas/<*data type*>'" if that data type schema is contained in the same OpenAPI file and "$ref: '<*filename*>#/components/schemas/<*data type*>'" if that data type schema is contained in file <*filename*> in the same directory on the same server.

     c. if the cardinality contained an integer value <*m*> as lower boundary, "min Properties: <*m*>"; and

     d. if the cardinality contained an integer value <*n*> as upper boundary, "max Properties: <*n*>";

  4. "description: <*description*>", where <*description*> is the description of the attribute in the table defining the structured data type.

NOTE: An omission of the "minProperties", and "maxProperties" keywords indicates that no lower or upper boundaries respectively, for the number of properties in an object are defined. An omission of the "minItems", and "maxItems" keywords indicates that no lower or upper boundaries, respectively, for the number of items in an array are defined.

Example:

Table 5.2.9.3-1: Definition of type ExampleStructuredType

Attribute name	Data type	Cardinality	Description
exSimple	ExSimple	1	exSimple attribute description
exArrayElements	array(string)	0..10	exArrayElements attribute description
exMapElements	map(ExStructure)	1..N	exMapElements attribute description

The data structure in table 5.2.9.3-1 is described in an OpenAPI file as follows:

```
components:
  schemas:
    ExampleStructuredType:
      type: object
      required:
        - exSimple
        - exMapElements
      properties:
        exSimple:
          $ref: '#/components/schemas/ExSimple'
          description: exSimple attribute description
        exArrayElements:
          type: array
          items:
            type: string
          minItems: 0
          maxItems: 10
          description: exArrayElements attribute description
        exMapElements:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/ExStructure'
          minProperties: 1
          description: exMapElements attribute description
```

5.2.9.4 Info

```
The Open API specification file of an API shall contain an "info" field with the title and version
as defined in subclause 4.3 of 3GPP TS 29.501 [49].
```

5.2.9.5 Servers

As defined in subclause 5.2.4, the base URI of an API consists of **{apiRoot}/{apiName}/{apiVersion}**. It shall be encoded in the corresponding Open API specification file as "servers" field with **{apiRoot}** as variable.

Example:

```
servers:
  - url: '{apiRoot}/3gpp-yyyy/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
```

5.2.9.6 References to other 3GPP-defined Open API specification files

For the purpose of referencing, it shall be assumed that each Open API specification file contained in a 3GPP specification is stored as separate physical file, that all Open API specification files are stored in the same directory on the local server, and that the files are named according to the following convention: The file name shall consist of (in the order below):

- the 3GPP specification number in the format "Tsxxyyy";
- an "_" character;
- if the OpenAPI specification file contains an API definition, the API name as defined for corresponding base URL parts (see subclause 4.4) of that API.

- if the OpenAPI specification file contains a definition of CommonData, the string "CommonData"; and
- the string ".yaml".

Examples:

Reference to Data Type "Xxx" defined in the same file

```
$ref: '#/components/schemas/Xxx'
```

Reference to Data Type "Xxx" defined as Common Data in 3GPP TS 29.122:

```
$ref: 'TS29122_CommonData.yaml#/components/schemas/Xxx'
```

Reference to Data Type "Xxx" defined within API "Nxxx_Yyy" in 3GPP TS ab.cde:

```
$ref: 'Tsabcde_Nxxx_Yyy.yaml#/components/schemas/Xxx'
```

5.2.9.7 Server-initiated communication

If an API contains notifications as described in subclause 5.2.5, it should be described as "callback" in Open API specification files.

Example:

```
paths:
  /subscriptions:
    post:
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                callbackUrl: # Callback URL
                  type: string
                  format: uri
      responses:
        '201':
          description: Success
      callbacks:
        myNotification: # arbitrary name
          '{$request.body#/callbackUrl}': # refers The callback URL in the POST
            post:
              requestBody: # Contents of the callback message
                required: true
                content:
                  application/json:
                    schema:
                      $ref: '#/components/schemas/NotificationBody'
              responses: # Expected responses to the callback message
                '200':
                  description: xxx
```

5.2.9.8 Describing the body of HTTP PATCH requests

In the OpenAPI file, the content field key of the Request Body Object shall contain "application/merge-patch+json". The content field value is a Media Type Object identifying the applicable patch body Schema Object. The patch body Schema Object may contain structured data types derived from the data types used in the schema to describe a complete representation of the resource in such a manner that attributes that are allowed to be modified are listed in the "properties" validation keyword.

NOTE 1: A derived structured data type is beneficial if the data types used to describe a complete representation of the resource contains mandatory attributes, if attributes are allowed to be removed by the PATCH operation, or if a checking by the OpenAPI tooling that only allowed modifications are done via the "additionalProperties: false" keyword is desired. It also provides a clear description in the OpenAPI file to developers which modifications need to be supported.

As an alternative, the data types used in the schema to describe a complete representation of the resource may be used if any attributes that are allowed to be removed are marked as "nullable: true" in that schema.

Any attributes that are allowed to be removed shall be marked as "nullable: true" in the patch body Schema Object.

The "additionalProperties: false" keyword may be set.

NOTE 2: The "additionalProperties: false" keyword enables the OpenAPI tooling to check that only allowed modifications are done. Extensions of the object in future releases are still possible under the assumption that the supported features mechanism is used to negotiate the usage of any new attribute prior to the PATCH invocation. If new optional attributes are expected to be introduced without corresponding supported feature or if PATCH can be used as first operation in an API, the usage of the "additionalProperties: false" keyword is not appropriate.

5.2.9.9 Error Responses

As described in subclause 5.2.6, T8 APIs use valid HTTP response codes as error codes in HTTP responses and may include a "ProblemDetails" data structure specified in subclause 5.2.1.2.12 or an application-specific data structure.

Subclause 5.2.6 specifies HTTP status code per HTTP method. OpenAPI files should include at least the status codes in that table.

For the purpose of referencing, HTTP error responses with "ProblemDetails" data structure are specified as part of the CommonData OpenAPI file in Annex A.2.

Example:

In the example below, the 400, and 500 and default error response descriptions are referenced.

```
Paths:
  /users:
    get:
      responses:
        '200':
          content:
            application/json
              schema:
                $ref: '#/components/schemas/ExampleGetBody'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
```

The following definitions provided in Annex A.2 are used in that example:

```
components:
  responses:
    '400':
      description: Bad request
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '500':
      description: Internal Server Error
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    default:
      description: Generic Error
```

5.2.9.10 Enumerations

For enumerations, the OpenAPI file shall contain a definition in the components/schemas section defining a schema with the name of the enumeration as key.

The schema

- shall contain the "anyOf" keyword listing as alternatives:

1. the "type: string" keyword and the "enum" keyword with a list of all defined values for the enumeration; and
2. the "type: string" keyword and the "description" keyword with a description stating that the string is only provided for extensibility and is not used to encode contents defined in the present version of the specification. And

- may contain a description listing the defined values of the enumeration together with explanations of those values.

NOTE: The "enum" keyword restricts the permissible values of the string to the enumerated ones. This can lead to extensibility problems when new values need to be introduced.

Example:

Table 5.2.9.10-1: Enumeration ExampleEnumeration

Enumeration value	Description	Applicability
One	Value One description	
Two	Value Two description	

The data structure in table 5.2.9.10-1 is described in an OpenAPI file as follows:

```
components:
  schemas:
    ExampleEnumeration:
      anyOf:
      - type: string
        enum:
          - One
          - Two
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      Description: >
        Possible values are
        - One: Value One description
        - Two: Value Two description
```

5.2.9.11 Read only attribute

Each OpenAPI specification should include "readOnly: true" for those attributes that are only provided by the SCEF in the HTTP response message to prevent the SCS/AS from provisioning those attributes which is not expected, if the write and read operations (e.g. POST request and response) share the same data type which contains those attributes.

Example:

```
    NiddStatus:
      anyOf:
      - type: string
        enum:
          - ACTIVE
          - TERMINATED_UE_NOT_AUTHORIZED
          - TERMINATED
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      Description: >
        Possible values are
        - ACTIVE: The NIDD configuration is active.
        - TERMINATED_UE_NOT_AUTHORIZED: The NIDD configuration was terminated because the UE´s
authorisation was revoked.
        - TERMINATED: The NIDD configuration was terminated.
      readOnly: true
```

5.3 MonitoringEvent API

5.3.1 Overview

The MonitoringEvent API is a RESTful API that allows the SCS/AS to subscribe to notifications about specific events in 3GPP networks. It also allows the SCEF to report the event by sending notifications to the authorised users when the corresponding event is detected. The API also allows the SCEF to indicate the removal of a previously configured monitoring request. The MonitoringEvent API defines a set of data models, resources and the related procedures for the creation and management of the monitoring event subscription. The corresponding JSON schema for the representation of the resources and operations defined by the MonitoringEvent API is provided in its complete form in Annex A.3.

5.3.2 Data model

5.3.2.1 Resource data types

5.3.2.1.1 Introduction

This clause defines data structures to be used in resource representations, including subscription resources.

Table 5.3.2.1.1-1 specifies data types re-used by the MonitoringEvent API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the MonitoringEvent API.

Table 5.3.2.1.1-1: MonitoringEvent API re-used Data Types

Data type	Reference	Comments
GeographicArea	3GPP TS 29.572 [42]	Identifies the geographical information of the user(s).
CivicAddress	3GPP TS 29.572 [42]	Identifies the civic address information of the user(s).
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.3.4-1.

5.3.2.1.2 Type: MonitoringEventSubscription

This type represents a subscription to monitoring an event. The same structure is used in the subscription request and subscription response.

Table 5.3.2.1.2-1: Definition of type MonitoringEventSubscription

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 3)
self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses.	
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
externalId	ExternalId	0..1	Identifies a user as defined in Clause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 1)	(NOTE 5)
msisdn	Msisdn	0..1	Identifies the MS internal PSTN/ISDN number allocated for a UE. (NOTE 1)	(NOTE 5)
externalGroupId	ExternalGroupId	0..1	Identifies a user group as defined in Clause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 1)	
ipv4Addr	Ipv4Addr	0..1	Identifies the Ipv4 address. (NOTE 1)	Location_notification, Communication_failure_notification
ipv6Addr	Ipv6Addr	0..1	Identifies the Ipv6 address. (NOTE 1)	Location_notification, Communication_failure_notification
notificationDestination	Link	1	An URI of a notification destination that T8 message shall be delivered to.	
requestTestNotification	boolean	0..1	Set to true by the SCS/AS to request the SCEF to send a test notification as defined in subclause 5.2.5.3. Set to false or omitted otherwise.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 5.2.5.4.	Notification_websocket
monitoringType	MonitoringType	1	Enumeration of monitoring type. Refer to clause 5.3.2.4.3.	
maximumNumberOfReports	integer	0..1	Identifies the maximum number of event reports to be generated by the HSS, MME/SGSN as specified in subclause 5.6.0 of 3GPP TS 23.682 [2]. (NOTE 2)	
monitorExpireTime	DateTime	0..1	Identifies the absolute time at which the related monitoring event request is considered to expire, as specified in subclause 5.6.0 of 3GPP TS 23.682 [2]. (NOTE 2)	
groupReportGuardTime	DurationSec	0..1	Identifies the time for which the SCEF can aggregate the monitoring event reports detected by the Ues in a group and report them together to the SCS/AS, as specified in subclause 5.6.0 of 3GPP TS 23.682 [2].	
maximumDetectionTime	DurationSec	0..1	If "monitoring-Type" is "LOSS_OF_CONNECTIVITY", this parameter may be included to identify the maximum period of time after which the UE is considered to be unreachable.	Loss_of_connectivity_notification
reachabilityType	ReachabilityType	0..1	If "monitoring-Type" is "UE_REACHABILITY", this parameter shall be included to identify whether the request is for "Reachability for SMS" or "Reachability for Data".	Ue-reachability_notification

maximumLatency	DurationSec	0..1	If "monitoring-Type" is "UE_REACHABILITY", this parameter may be included to identify the maximum delay acceptable for downlink data transfers.	Ue-reachability_notification
maximumResponseTime	DurationSec	0..1	If "monitoring-Type" is "UE_REACHABILITY", this parameter may be included to identify the length of time for which the UE stays reachable to allow the SCS/AS to reliably deliver the required downlink data.	Ue-reachability_notification
suggestedNumberOfDlPackets	integer	0..1	If "monitoringType" is "UE_REACHABILITY", this parameter may be included to identify the number of packets that the serving gateway shall buffer in case that the UE is not reachable.	Ue-reachability-notification
idleStatusIndication	boolean	0..1	If "monitoringType" is set to "UE_REACHABILITY" or "AVAILABILITY_AFTER_DDN_FAILURE", this parameter may be included to indicate the notification of when a UE, for which PSM is enabled, transitions into idle mode. - "true": indicate enabling of notification - "false": indicate no need to notify Default: "false".	Ue-reachability_notification, Availability_after_DDN_failure_notification
locationType	LocationType	0..1	If "monitoringType" is "LOCATION_REPORTING" or "NUMBER_OF_UES_IN_AN_AREA", this parameter shall be included to identify whether the request is for Current Location or Last known Location. (NOTE 4)	Location_notification, Number_of_UEs_in_an_area_notification
accuracy	Accuracy	0..1	If "monitoring-Type" is "LOCATION_REPORTING", this parameter may be included to identify the desired level of accuracy of the requested location information, as described in subclause 4.9.2 of 3GPP TS 23.682 [2].	Location_notification
minimumReportInterval	DurationSec	0..1	If "monitoring-Type" is "LOCATION_REPORTING", this parameter may be included to identify a minimum time interval between Location Reporting notifications.	Location_notification
associationType	AssociationType	0..1	If "monitoring-Type" is "CHANGE_OF_IMSI_IMEI_ASSOCIATION", this parameter shall be included to identify whether the change of IMSI-IMEI or IMSI-IMEISV association shall be detected.	Change_of_IMSI_IMEI_association_notification
plmnIndication	boolean	0..1	If "monitoring-Type" is "ROAMING_STATUS", this parameter may be included to indicate the notification of UE's Serving PLMN ID. - "true": The value shall be used to indicate enabling of notification; - "false": The value shall be used to indicate disabling of notification. Default: "false".	Roaming_status_notification

locationArea	LocationArea	0..1	If "monitoring-Type" is "NUMBER_OF_UES_IN_AN_AREA", this parameter shall be included to indicate the area within which the SCS/AS requests the number of Ues.	Number_of_UEs_in_an_area_notification
monitoringEventReport	MonitoringEventReport	0..1	Identifies a monitoring event report which is sent from the SCEF to the SCS/AS.	
NOTE 1: One of the properties "externalId", "msisdn", "ipv4Addr", "ipv6Addr" or "externalGroupId" shall be included. "ipv4Addr" or "ipv6Addr" is required for monitoring via the PCRF for an individual UE. NOTE 2: Inclusion of either "maximumNumberOfReports" (with a value higher than 1) or "monitorExpireTime" makes the Monitoring Request a Continuous Monitoring Request, where the SCEF sends Notifications until either the maximum number of reports or the monitoring duration indicated by the property "monitorExpireTime" is exceeded. The "maximumNumberOfReports" with a value 1 makes the Monitoring Request a One-time Monitoring Request. At least one of "maximumNumberOfReports" or "monitorExpireTime" shall be provided. NOTE 3: Properties marked with a feature as defined in subclause 5.3.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features. NOTE 4: In this release, for feature "Number_of_Ues_in_an_area_notification", locationType shall be set to "LAST_KNOWN_LOCATION". NOTE 5: The property does not apply for the feature "Number_of_Ues_in_an_area_notification".				

5.3.2.2 Notification data types

5.3.2.2.1 Introduction

This clause defines data structures to be used in notifications.

5.3.2.2.2 Type: MonitoringNotification

This data type represents a monitoring notification which is sent from the SCEF to the SCS/AS.

Table 5.3.2.2.2-1: Definition of type MonitoringNotification

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
subscription	Link	1	Link to the subscription resource to which this notification is related.	
configResults	array(ConfigResult)	0..N	Each element identifies a notification of grouping configuration result.	
monitoringEventReports	array(MonitoringEventReport)	0..N	Each element identifies a monitoring event report.	
Cancelind	boolean	0..1	Indicates whether to request to cancel the corresponding monitoring subscription. Set to false or omitted otherwise.	
NOTE: Properties marked with a feature as defined in subclause 5.3.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.				

5.3.2.3 Referenced structured data types

5.3.2.3.1 Introduction

This clause defines structured data types that are referenced from data structures defined in the previous clauses.

5.3.2.3.2 Type: MonitoringEventReport

This data type represents a monitoring event notification which is sent from the SCEF to the SCS/AS.

Table 5.3.2.3.2-1: Definition of type MonitoringEventReport

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
imeiChange	AssociationType	0..1	If "monitoring-Type" is "CHANGE_OF_IMSI_IMEI_ASSOCIATION", this parameter shall be included to identify the event of change of IMSI-IMEI or IMSI-IMEISV association is detected. Refer to 3GPP TS 29.336 [11] Subclause 8.4.22.	Change_of_IMSI_IMEI_association_notification
externalIds	array(ExternalId)	0..N	External identifier (NOTE 2)	
idleStatusInfo	IdleStatusInfo	0..1	If "idleStatusIndication" in the "MonitoringEventSubscription"sets to "true", this parameter shall be included to indicate the information when the UE transitions into idle mode.	Ue-reachability_notification, Availability_after_DDN_failure_notification
locationInfo	LocationInfo	0..1	If "monitoringType" is "LOCATION_REPORTING", this parameter shall be included to indicate the user location related information.	Location_notification
lossOfConnectReason	integer	0..1	If "monitoring-Type" is "LOSS_OF_CONNECTIVITY", this parameter shall be included if available to identify the reason why loss of connectivity is reported. Refer to 3GPP TS 29.336 [11] Subclause 8.4.58.	Loss_of_connectivity_notification
maxUEAvailabilityTime	DateTime	0..1	If "monitoring-Type" is "UE_REACHABILITY", this parameter may be included to identify the timestamp until which a UE using a power saving mechanism is expected to be reachable for SM delivery. Refer to Subclause 5.3.3.22 of 3GPP TS 29.338 [34].	Ue-reachability_notification
msisdns	array(Msisdn)	0..N	Identifies the MS internal PSTN/ISDN number (NOTE 2)	
monitoringType	MonitoringType	1	Identifies the type of monitoring type as defined in clause 5.3.2.4.3.	
uePerLocationReport	UePerLocationReport	0..1	If "monitoringType" is "NUMBER_OF_UES_IN_AN_AREA", this parameter shall be included to indicate the number of Ues found at the location.	Number_of_UEs_in_an_area_notification
plmnId	PlmnId	0..1	If "monitoringType" is "ROAMING_STATUS" and "plmnIIndication" in the "MonitoringEventSubscription" sets to "true", this parameter shall be included to indicate the UE's serving PLMN.	Roaming_status_notification
reachabilityType	ReachabilityType	0..1	If "monitoring-Type" is "UE_REACHABILITY", this parameter shall be included to identify the reachability of the UE. Refer to 3GPP TS 29.336 [11] Subclause 8.4.20.	Ue-reachability_notification

roamingStatus	boolean	0..1	If "monitoringType" is "ROAMING_STATUS", this parameter shall be set to "true" if the UE is on roaming status. Set to false or omitted otherwise.	Roaming_status_notification
failureCause	FailureCause	0..1	If "monitoringType" is "COMMUNICATION_FAILURE", this parameter shall be included to indicate the reason of communication failure.	Communication_failure_notification
NOTE 1: Properties marked with a feature as defined in subclause 5.3.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features. NOTE 2: Identifies the user(s) for which the event occurred. Can be omitted if the event occurred for all users identified by the surrounding structure.				

5.3.2.3.3 Type: IdleStatusInfo

This data type represents the information when the UE transitions into idle mode. It is sent from the SCEF to the SCS/AS.

Table 5.3.2.3.3-1: Definition of type IdleStatusInfo

Attribute name	Data type	Cardinality	Description
activeTime	DurationSec	0..1	Identifies the active time granted to the UE. It shall be present if the idle status indication is requested by the SCS/AS with "idleStatusIndication" in the "monitoringEventSubscription" sets to "true".
edrxCycleLength	DurationSec	0..1	Identifies the power saving interval. It shall be present if the idle status indication is requested by the SCS/AS with "idleStatusIndication" in the "monitoringEventSubscription" sets to "true".
suggestedNumberOfDlPackets	integer	0..1	Identifies the number of packets shall be buffered in the serving gateway. It shall be present if the idle status indication is requested by the SCS/AS with "idleStatusIndication" in the "monitoringEventSubscription" sets to "true".
idleStatusTimestamp	DateTime	0..1	Identifies the timestamp at which the UE transitions into idle mode. It shall be present if the idle status indication is requested by the SCS/AS with "idleStatusIndication" in the "monitoringEventSubscription" sets to "true".
periodicAUTimer	DurationSec	0..1	Identifies the subscribed periodic RAU/TAU timer value. It shall be present if the idle status indication is requested by the SCS/AS with "idleStatusIndication" in the "monitoringEventSubscription" sets to "true".

5.3.2.3.4 Type: UePerLocationReport

This data type represents the number of UEs found at the indicated location information. It is sent from the SCEF to the SCS/AS.

Table 5.3.2.3.4-1: Definition of type UePerLocationReport

Attribute name	Data type	Cardinality	Description
ueCount	integer	1	Identifies the number of Ues.
externalIds	array(ExternalId)	0..N	Each element uniquely identifies a user. (NOTE)
msisdns	array(Msisdn)	0..N	Each element identifies the MS internal PSTN/ISDN number allocated for a UE. (NOTE)
NOTE: The property shall be included if received from the MME(s)/SGSN(s).			

5.3.2.3.5 Type: LocationInfo

This data type represents the user location information which is sent from the SCEF to the SCS/AS.

Table 5.3.2.3.5-1: Definition of LocationInfo data Type

Attribute name	Data type	Cardinality	Description
ageOfLocationInfo	DurationMin	0..1	Indicates the elapsed time since the last network contact of the UE. Refer to 3GPP TS 29.272 [33].
cellId	string	0..1	Indicates the Cell Global Identification of the user which identifies the cell the UE is registered.
enodeBId	string	0..1	Indicates the eNodeB in which the UE is currently located.
routingAreaId	string	0..1	Identifies the Routing Area Identity of the user where the UE is located.
trackingAreaId	string	0..1	Identifies the Tracking Area Identity of the user where the UE is located.
plmmId	string	0..1	Identifies the PLMN Identity of the user where the UE is located.
twanId	string	0..1	Identifies the TWAN Identity of the user where the UE is located.

5.3.2.3.6 Type: FailureCause

This data type represents the reason of communication failure. It shall comply with the provisions defined in table 5.3.2.3.6-1.

Table 5.3.2.3.6-1: Definition of type FailureCause

Attribute name	Data type	Cardinality	Description
bssgpCause	integer	0..1	Identifies a non-transparent copy of the BSSGP cause code. Refer to 3GPP TS 29.128 [12].
causeType	integer	0..1	Identify the type of the S1AP-Cause. Refer to 3GPP TS 29.128 [12].
gmmCause	integer	0..1	Identifies a non-transparent copy of the GMM cause code. Refer to 3GPP TS 29.128 [12].
ranapCause	integer	0..1	Identifies a non-transparent copy of the RANAP cause code. Refer to 3GPP TS 29.128 [12].
ranNasCause	string	0..1	Indicates RAN and/or NAS release cause code information, TWAN release cause code information or untrusted WLAN release cause code information. Refer to 3GPP TS 29.214 [10]. (NOTE)
s1ApCause	integer	0..1	Identifies a non-transparent copy of the S1AP cause code. Refer to 3GPP TS 29.128 [12].
smCause	integer	0..1	Identifies a non-transparent copy of the SM cause code. Refer to 3GPP TS 29.128 [12].
NOTE: If this property is provided in the "FailureCause" type, then other properties shall not be provided.			

5.3.2.4 Referenced simple data types and enumerations

5.3.2.4.1 Introduction

This clause defines simple data types and enumerations that are referenced from data structures defined in the previous clauses. In addition, data types and enumerations defined in subclause 5.2.1 can be referenced.

5.3.2.4.2 Simple data types

The simple data types defined in table 5.3.2.4.2-1 shall be supported.

Table 5.3.2.4.2-1: Simple data types

Type name	Description

5.3.2.4.3 Enumeration: MonitoringType

The enumeration MonitoringType represents a monitoring event type. It shall comply with the provisions defined in table 5.3.2.4.3-1.

Table 5.3.2.4.3-1: Enumeration MonitoringType

Enumeration value	Description	Applicability (NOTE 1)
LOSS_OF_CONNECTIVITY	The SCS/AS requests to be notified when the 3GPP network detects that the UE is no longer reachable for signalling or user plane communication	Loss_of_connectivity_notification
UE_REACHABILITY	The SCS/AS requests to be notified when the UE becomes reachable for sending either SMS or downlink data to the UE	Ue-reachability_notification
LOCATION_REPORTING	The SCS/AS requests to be notified of the current location or the last known location of the UE	Location_notification
CHANGE_OF_IMSI_IMEI_ASSOCIATION	The SCS/AS requests to be notified when the association of an ME (IMEI(SV)) that uses a specific subscription (IMSI) is changed	Change_of_IMSI_IMEI_association_notification
ROAMING_STATUS	The SCS/AS queries the UE's current roaming status and requests to get notified when the status changes	Roaming_status_notification
COMMUNICATION_FAILURE	The SCS/AS requests to be notified of communication failure events	Communication_failure_notification
AVAILABILITY_AFTER_DDN_FAILURE	The SCS/AS requests to be notified when the UE has become available after a DDN failure	Availability_after_DDN_failure_notification
NUMBER_OF_UES_IN_AN_AREA	The SCS/AS requests to be notified the number of Ues in a given geographic area	Number_of_Ues_in_an_area_notification
NOTE 1: Properties marked with a feature as defined in subclause 5.3.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features. NOTE 2: More monitoring types can be added in the future based on stage 2.		

5.3.2.4.4 Enumeration: ReachabilityType

The enumeration ReachabilityType represents a reachability type. It shall comply with the provisions defined in table 5.3.2.4.4-1. If reachabilityType set to "SMS", the monitoring event request from SCS/AS shall be only for one-time monitoring request.

Table 5.3.2.4.4-1: Enumeration ReachabilityType

Enumeration value	Description	Applicability (NOTE)
SMS	The SCS/AS requests to be notified when the UE becomes reachable for sending SMS to the UE	Ue-reachability_notification
DATA	The SCS/AS requests to be notified when the UE becomes reachable for sending downlink data to the UE	Ue-reachability_notification
NOTE: Properties marked with a feature as defined in subclause 5.3.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.		

5.3.2.4.5 Enumeration: LocationType

The enumeration LocationType represents a location type. It shall comply with the provisions defined in table 5.3.2.4.5-1. If locationType set to "LAST_KNOWN_LOCATION", the monitoring event request from SCS/AS shall be only for one-time monitoring request.

Table 5.3.2.4.5-1: Enumeration LocationType

Enumeration value	Description	Applicability (NOTE)
CURRENT_LOCATION	The SCS/AS requests to be notified for current location	Location_notification
LAST_KNOWN_LOCATION	The SCS/AS requests to be notified for last known location	Location_notification, Number_of_Ues_in_an_area_notification
NOTE: Properties marked with a feature as defined in subclause 5.3.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.		

5.3.2.4.6 Enumeration: AssociationType

The enumeration AssociationType represents an IMEI or IMEISV to IMSI association. It shall comply with the provisions defined in table 5.3.2.4.6-1.

Table 5.3.2.4.6-1: Enumeration AssociationType

Enumeration value	Description	Applicability (NOTE)
IMEI	The value shall be used when the change of IMSI-IMEI association shall be detected	Change_of_IMSI_IMEI_association_notification
IMEISV	The value shall be used when the change of IMSI-IMEISV association shall be detected	Change_of_IMSI_IMEI_association_notification
NOTE: Properties marked with a feature as defined in subclause 5.3.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.		

5.3.2.4.7 Enumeration: Accuracy

The enumeration Accuracy represents a desired granularity of accuracy of the requested location information. It shall comply with the provisions defined in table 5.3.2.4.7-1.

Table 5.3.2.4.7-1: Enumeration Accuracy

Enumeration value	Description
CGI_ECGI	The SCS/AS requests to be notified at cell level location accuracy.
ENODEB	The SCS/AS requests to be notified at eNodeB level location accuracy.
TA_RA	The SCS/AS requests to be notified at TA/RA level location accuracy.
PLMN	The SCS/AS requests to be notified at PLMN level location accuracy.
TWAN_ID	The SCS/AS requests to be notified at TWAN identifier level location accuracy.

5.3.3 Resource structure

5.3.3.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-monitoring-event/v1/

"apiRoot" is set as described in subclause 5.2.4. "apiName" shall be set to "3gpp-monitoring-event" and "apiVersion" shall be set to "v1" for the current version defined in the present document. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.3.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
Monitoring Event Subscriptions	3gpp-monitoring-event/v1/{scsAsId}/subscriptions/	GET	Read all subscriptions for a given SCS/AS
		POST	Create a new subscription to monitoring an event
Individual Monitoring Event Subscription	3gpp-monitoring-event/v1/{scsAsId}/subscriptions/{subscriptionId}	PUT	Modify an existing subscription of monitoring event
		GET	Read a subscription of monitoring event
		DELETE	Delete a subscription of monitoring event
Monitoring Notification	{notificationDestination}	POST	Sent from the SCEF to the SCS/AS about the notification of grouping configuration result, detected monitoring event report(s) or notify the SCS/AS to cancel a monitoring subscription

5.3.3.2 Resource: Monitoring Event Subscriptions

5.3.3.2.1 Introduction

This resource allows an SCS/AS to read all of the active monitoring event subscriptions or create a new monitoring event subscription for the SCS/AS at the SCEF.

5.3.3.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-monitoring-event/v1/{scsAsId}/subscriptions/**

This resource shall support the resource URI variables defined in table 5.3.3.2.2-1.

Table 5.3.3.2.2-1: Resource URI variables for resource "Monitoring Event Subscriptions"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.3.3.2.3 Resource methods

5.3.3.2.3.1 GET

The GET method allows to read all active subscriptions for a given SCS/AS. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.3.3.2.3.1-1 and table 5.3.3.2.3.1-2.

Table 5.3.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.3.3.2.3.1-2: Data structures supported by the GET request/response by the resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	none			
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	array(MonitoringEventSubscription)	0..N	200 OK	The subscription information for the SCS/AS in the request URI are returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.3.3.2.3.2 PUT

This HTTP method is not supported for the resource.

5.3.3.2.3.3 PATCH

This HTTP method is not supported for the resource.

5.3.3.2.3.4 POST

The POST method creates a new subscription resource to monitor an event for a given SCS/AS. The SCS/AS shall initiate the HTTP POST request message and the SCEF shall respond to the message. The SCS/AS may provide the TLTRI in the message body. The SCEF shall construct the URI of the created resource.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.3.3.2.3.4-1 and table 5.3.3.2.3.4-2.

Table 5.3.3.2.3.4-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.3.3.2.3.4-2: Data structures supported by the POST request/response by the resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	MonitoringEventSubscription	1	Parameters to register a subscription to notifications about monitoring event with the SCEF.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	MonitoringEventSubscription	1	201 Created	The subscription was created successfully. The URI of the created resource shall be returned in the "Location" HTTP header.
	MonitoringEventReport	1	200 OK	The operation is successfully, and corresponding monitoring event report is included.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.3.3.2.3.5 DELETE

This HTTP method is not supported for the resource.

5.3.3.3 Resource: Individual Monitoring Event Subscription

5.3.3.3.1 Introduction

This resource allows an SCS/AS to read, update or delete an active monitoring event subscription at the SCEF.

5.3.3.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp-monitoring-event/v1/{scsAsId}/subscriptions/{subscriptionId}**

This resource shall support the resource URI variables defined in table 5.2.3.2.2-1.

Table 5.3.3.3.2-1: Resource URI variables for resource "Individual Monitoring Event Subscription"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
subscriptionId	Identifier of the subscription resource of type string. The subscriptionId corresponds to the stage 2 TLTRI.

5.3.3.3.3 Resource methods

5.3.3.3.3.1 GET

The GET method allows to read an active subscription resource to obtain details of the subscription. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.3.3.3.3.1-1 and table 5.3.3.3.3.1-2.

Table 5.3.3.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.3.3.3.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	MonitoringEventSubscription	1	200 OK	The subscription information related to the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.3.3.3.3.2 PUT

The PUT method modifies an existing subscription resource to update the subscription. The SCS/AS shall initiate the HTTP PUT request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.3.3.3.3.2-1 and table 5.3.3.3.3.2-2.

Table 5.3.3.3.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.3.3.3.3.2-2: Data structures supported by the PUT request/response by the resource

	Data type	Cardinality	Response codes	Remarks
Request body	MonitoringEventSubscription	1		Parameters to register a subscription to notifications about monitoring event with the SCEF.
Response body	Data type	Cardinality	Response codes	Remarks
	MonitoringEventSubscription	1	200 OK	The subscription was updated successfully.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.3.3.3.3.3 PATCH

This HTTP method is not supported for the resource.

5.3.3.3.3.4 POST

This HTTP method is not supported for the resource.

5.3.3.3.3.5 DELETE

The DELETE method deletes the related resource and terminates the related monitoring subscription. The SCS/AS shall initiate the HTTP DELETE request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.3.3.3.3.5-1 and table 5.3.3.3.3.5-2.

Table 5.3.3.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.3.3.3.3.5-2: Data structures supported by the DELETE request/response by the resource

	Data type	Cardinality	Response codes	Remarks
Request body	none			
Response body	Data type	Cardinality	Response codes	Remarks
	none		204 No Content	The subscription was terminated successfully.
	Array(MonitoringEventReport)	1..N	200 OK	The subscription was terminated successfully, the monitoring event report(s) shall be included if received.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.3.3.4 Monitoring Notification

5.3.3.4.1 Introduction

This resource allows the SCEF to send notification about grouping configuration result, monitoring event report(s) or a monitoring subscription cancellation to the SCS/AS.

5.3.3.4.2 Resource definition

Resource URI: **{**notificationDestination**}**

This resource shall support the resource URI variables defined in table 5.3.3.4.2-1.

Table 5.3.3.4.2-1: Resource URI variables for resource "Monitoring Event Notification"

Name	Definition
notificationDestination	Callback reference provided by the SCS/AS during creation of the monitoring event subscription.

5.3.3.4.3 Resource methods

5.3.3.4.3.1 Notification via POST

The HTTP POST method reports the notification for a monitoring subscription. The SCEF shall initiate the HTTP POST request message and the SCS/AS shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.3.3.4.3.1-1 and table 5.3.3.4.3.1-2.

Table 5.3.3.4.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.3.3.4.3.1-2: Data structures supported by the POST request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>MonitoringNotification</td><td>1</td><td colspan="2">The monitoring notification provided by the SCEF.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>none</td><td></td><td>204 No Content</td><td>The monitoring notification is received successfully.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCS/AS can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.3.3.4.3.2 Notification via Websocket

If supported by both SCS/AS and SCEF and successfully negotiated, the MonitoringNotification may alternatively be delivered through the Websocket mechanism as defined in subclause 5.2.5.4.

5.3.4 Used Features

The table below defines the features applicable to the MonitoringEvent API. Those features are negotiated as described in subclause 5.2.7.

Table 5.3.4-1: Features used by MonitoringEvent API

Feature Number	Feature	Description
1	Loss_of_connectivity_notification	The SCS/AS is notified when the 3GPP network detects that the UE is no longer reachable for signalling or user plane communication
2	Ue-reachability_notification	The SCS/AS is notified when the UE becomes reachable for sending either SMS or downlink data to the UE
3	Location_notification	The SCS/AS is notified of the current location or the last known location of the UE
4	Change_of_IMSI_IMEI_association_notification	The SCS/AS is notified when the association of an ME (IMEI(SV)) that uses a specific subscription (IMSI) is changed
5	Roaming_status_notification	The SCS/AS is notified when the UE's roaming status changes
6	Communication_failure_notification	The SCS/AS is notified of communication failure events
7	Availability_after_DDN_failure_notification	The SCS/AS is notified when the UE has become available after a DDN failure
8	Number_of_Ues_in_an_area_notification	The SCS/AS is notified the number of Ues present in a given geographic area
9	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 5.2.5.4. This feature requires that the Notification_test_event featute is also supported.
10	Notification_test_event	The testing of notification connection is supported according to subclause 5.2.5.3.
11	Subscription_modification	Modifications of an individual subscription resource.
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.4 ResourceManagementOfBdt API

5.4.1 Overview

The ResourceManagementOfBdt API is a RESTful API that allows the SCS/AS to request background data transfer related conditions for a set of Ues. The ResourceManagementOfBdt API defines a set of data models, resources and the related procedures for the creation and management of the background data transfer request. The corresponding JSON schema for the representation of the resources and operations defined by the ResourceManagementOfBdt API is provided in its complete form in Annex A.4.

5.4.2 Data model

5.4.2.1 Resource data types

5.4.2.1.1 Introduction

This clause defines data structures to be used in resource representations.

Table 5.4.2.1.1-1 specifies data types re-used by the ResourceManagementOfBdt API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the ResourceManagementOfBdt API.

Table 5.4.2.1.1-1: ResourceManagementOfBdt API re-used Data Types

Data type	Reference	Comments
GeographicArea	3GPP TS 29.572 [42]	Identifies the geographical information of the user(s).
CivicAddress	3GPP TS 29.572 [42]	Identifies the civic address information of the user(s).
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.4.4-1.

5.4.2.1.2 Type: Bdt

This type represents a BDT subscription. The same structure is used in the subscription request and subscription response.

Table 5.4.2.1.2-1: Definition of type Bdt

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses that include an object of Bdt type	
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
volumePerUE	UsageThreshold	1	Identifies the data volume expected to be transferred per UE.	
numberOfUEs	integer	1	Identifies the number of Ues.	
desiredTimeWindow	TimeWindow	1	Identifies the time interval.	
locationArea	LocationArea	0..1	Identifies the area within which the SCS/AS requests the number of UE.	
referenceId	BdtReferenceId	0..1	Identifies a selected policy of background data transfer.	
transferPolicies	array(TransferPolicy)	0..N	Identifies an offered transfer policy.	
selectedPolicy	integer	0..1	Identity of the selected background data transfer policy. Shall not be present in initial message exchange, can be provided by NF service consumer in a subsequent message exchange.	
NOTE: Properties marked with a feature as defined in subclause 5.4.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.4.2.1.3 Type: BdtPatch

This type represents a BDT request for the service provided by the SCS/AS to the SCEF via T8 interface. The structure is used for PATCH request.

Table 5.4.2.1.3-1: Definition of type BdtPatch

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
selectedPolicy	integer	1	Identity of the selected background data transfer policy.	
NOTE: Properties marked with a feature as defined in subclause 5.4.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.4.2.2 Referenced structured data types

5.4.2.2.1 Introduction

This clause defines structured data types that are referenced from data structures defined in the previous clauses.

5.4.2.2.2 Type: TransferPolicy

This data type represents an offered transfer policy sent from the SCEF to the SCS/AS, or a selected transfer policy sent from the SCS/AS to the SCEF.

Table 5.4.2.2.2-1: Definition of type TransferPolicy

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
bdtPolicyId	integer	1	Identifier for the transfer policy	
maxUplinkBandwidth	Bandwidth	0..1	Indicates the maximum aggregated bitrate in the uplink authorized by the PCRF.	
maxDownlinkBandwidth	Bandwidth	0..1	Indicates the maximum aggregated bitrate in the downlink authorized by the PCRF.	
ratingGroup	integer	1	Indicates the rating group during the time window.	
timeWindow	TimeWindow	1	Indicates the recommended time window of the transfer policy	
NOTE: Properties marked with a feature as defined in subclause 5.4.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.4.3 Resource structure

5.4.3.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-bdt/v1/

"apiRoot" is set as described in subclause 5.2.4. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.4.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
BDT Subscription	3gpp-bdt/v1/{scsAsId}/subscriptions	GET	Read all active background data transfer subscription resources for a given SCS/AS
		POST	Create a new background data transfer subscription resource
Individual BDT Subscription	3gpp-bdt/v1/{scsAsId}/subscriptions/{subscriptionId}	PATCH	Modify a background data transfer subscription resource to select one of the transfer policies offered by the SCEF
		PUT	Update a background data transfer subscription resource for negotiation of background data transfer policy
		GET	Read a background data transfer subscription resource
		DELETE	Delete a background data transfer resources

5.4.3.2 Resource: BDT Subscriptions

5.4.3.2.1 Introduction

This resource allows the SCS/AS to read all active long-term transactions related to BDT resource management.

5.4.3.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-bdt/v1/{scsAsId}/subscriptions/**

This resource shall support the resource URI variables defined in table 5.4.3.2.2-1.

Table 5.4.3.2.2-1: Resource URI variables for resource "BDT Subscriptions"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.4.3.2.3 Resource methods

5.4.3.2.3.1 GET

The GET method allows to read all active resources for a given SCS/AS. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.4.3.2.3.1-1 and table 5.4.3.2.3.1-2.

Table 5.4.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.4.3.2.3.1-2: Data structures supported by the GET request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>array(Bdt)</td><td>0..N</td><td>200 OK</td><td>The resource information for the SCS/AS in the request URI are returned.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.4.3.2.3.2 PUT

This HTTP method is not supported for the resource.

5.4.3.2.3.3 PATCH

This HTTP method is not supported for the resource.

5.4.3.2.3.4 POST

The POST method creates a new background data transfer subscription resource for a given SCS/AS. The SCS/AS shall initiate the HTTP POST request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.4.3.2.3.4-1 and table 5.4.3.2.3.4-2.

Table 5.4.3.2.3.4-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.4.3.2.3.4-2: Data structures supported by the POST request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>Bdt</td><td>1</td><td colspan="2">Parameters to register a subscription to request background data transfer related information with the SCEF.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>Bdt</td><td>1</td><td>201 Created</td><td>The resource was created successfully.
The URI of the created resource shall be returned in the "Location" HTTP header.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.4.3.2.3.5 DELETE

This HTTP method is not supported for the resource.

5.4.3.3 Resource: Individual BDT Subscription

5.4.3.3.1 Introduction

This resource allows the SCS/AS to manage resources for BDT using a long-term transaction.

5.4.3.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp-bdt/v1/{scsAsId}/subscriptions/{subscriptionId}**

This resource shall support the resource URI variables defined in table 5.4.3.3.2-1.

Table 5.4.3.3.2-1: Resource URI variables for resource "Individual BDT Subscription"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
subscriptionId	Identifier of the subscription resource of type string. The subscriptionId corresponds to the stage 2 TLTRI.

5.4.3.3.3 Resource methods

5.4.3.3.3.1 GET

The GET method allows to read an individual BDT subscription resource to obtain details of an active resource BDT subscription. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.4.3.3.3.1-1 and table 5.4.3.3.3.1-2.

Table 5.4.3.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.4.3.3.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	Bdt	1	200 OK	The resource information related to the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.4.3.3.3.2 PUT

The PUT method allows the SCS/AS to modify an existing subscription resource completely. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.4.3.3.3.2-1.

Table 5.4.3.3.3.2-1: Data structures supported by the PUT request/response by the resource

Request body	Data type	Cardinality	Remarks	
	Bdt	1	The SCS/AS requests to update the BDT policy subscription.	
Response body	Data type	Cardinality	Response codes	Remarks
	Bdt	1	200 OK	The subscription was modified successfully. The SCEF shall return an updated subscription in the response payload body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.4.3.3.3.3 PATCH

The PATCH method allows the SCS/AS to modify an existing subscription resource, in order to notify the SCEF about the selected transfer policy. The SCS/AS shall initiate the HTTP PATCH message and the SCEF shall respond to the message.

This method shall support request and response data structures, and response codes, as specified in the table 5.4.3.3.3.3-1.

Table 5.4.3.3.3.3-1: Data structures supported by the PATCH request/response by the resource

Request body	Data type	Cardinality	Remarks	
	BdtPatch	1	Background data transfer policy selected by the SCS/AS.	
Response body	Data type	Cardinality	Response codes	Remarks
	Bdt	1	200 OK	The resource was modified successfully. The SCEF shall return an updated subscription in the response payload body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.4.3.3.3.4 POST

This HTTP method is not supported for the resource.

5.4.3.3.3.5 DELETE

The DELETE method deletes the resource and terminates the BDT subscription. The SCS/AS shall initiate the HTTP DELETE message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.4.3.3.3.5-1 and table 5.4.3.3.3.5-2.

Table 5.4.3.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.4.3.3.3.5-2: Data structures supported by the DELETE request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>none</td><td></td><td>204 No Content</td><td>The resource was terminated successfully.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.4.4 Used Features

The table below defines the features applicable to the ResourceManagementOfBdt API. Those features are negotiated as described in subclause 5.2.7.

Table 5.4.4-1: Features used by ResourceManagementOfBdt API

<table>
<tr><th>Feature Number</th><th>Feature</th><th>Description</th></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3">Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification".
Description: A clear textual description of the feature.</td></tr>
</table>

5.5 ChargeableParty API

5.5.1 Overview

The ChargeableParty API is a RESTful API that allows the SCS/AS to either request to sponsor the traffic from the beginning or to request becoming the chargeable party at a later point in time via the T8 interface. The ChargeableParty API defines a set of data models, resources and the related procedures for the creation and management of the AS sessions with chargeable party change. The corresponding JSON schema for the representation of the resources and operations defined by the Chargeable API is provided in its complete form in Annex A.5.

5.5.2 Data model

5.5.2.1 Resource data types

5.5.2.1.1 Introduction

This clause defines data structures to be used in resource representations.

Table 5.5.2.1.1-1 specifies data types re-used by the ChargeableParty API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the ChargeableParty API.

Table 5.5.2.1.1-1: ChargeableParty API re-used Data Types

Data type	Reference	Comments
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.5.4-1.

5.5.2.1.2 Type: ChargeableParty

This type represents the configuration of a chargeable party. The same structure is used in the configuration request and configuration response.

Table 5.5.2.1.2-1: Definition of type ChargeableParty

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses that include an object of ChargeableParty type.	
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
notificationDestination	Link	1	Contains the URL to receive the notification of bearer level event(s) from the SCEF.	
requestTestNotification	boolean	0..1	Set to true by the SCS/AS to request the SCEF to send a test notification as defined in subclause 5.2.5.3. Set to false or omitted otherwise.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 5.2.5.4.	Notification_websocket
ipv4Addr	Ipv4Addr	0..1	Identifies the Ipv4 address. (NOTE 2)	
ipv6Addr	Ipv6Addr	0..1	Identifies the Ipv6 address. (NOTE 2)	
flowInfo	array(FlowInfo)	1..N	Describes the application flows.	
sponsorInformation	SponsorInformation	1	Describes the sponsor information such as who is sponsoring the traffic.	
sponsoringEnabled	boolean	1	Indicates sponsoring status.	
referenceId	BdtReferenceId	0..1	The reference ID for a previously selected policy of background data transfer.	
usageThreshold	UsageThreshold	0..1	Time period and/or traffic volume.	
NOTE 1: Properties marked with a feature as defined in subclause 5.5.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features. NOTE 2: Either ipv4 or ipv6 address shall be provided.				

5.5.2.1.3 Type: ChargeablePartyPatch

This type represents the configuration of a chargeable party. The structure is used for PATCH request.

Table 5.5.2.1.3-1: Definition of type ChargeablePartyPatch

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
flowInfo	array(FlowInfo)	0..N	Describes the application flows.	
sponsoringEnabled	boolean	0..1	Indicates sponsoring status.	
referenceId	BdtReferenceId	0..1	The reference ID for a previously selected policy of background data transfer.	
usageThreshold	UsageThresholdRm	0..1	Time period and/or traffic volume.	
NOTE: Properties marked with a feature as defined in subclause 5.5.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.				

5.5.3 Resource structure

5.5.3.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-chargeable-party/v1/

"apiRoot" is set as described in subclause 5.2.4. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.5.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
Chargeable Party Transactions	3gpp-chargeable-party/v1/{scsAsId}/transactions	GET	Read all chargeable party transaction resources for a given SCS/AS
		POST	Create a new chargeable party transaction resource
Individual Chargeable Party Transaction	3gpp-chargeable-party/v1/{scsAsId}/transactions/{transactionId}	GET	Read a chargeable party transaction resource
		PATCH	Activate or Deactivate sponsoring by a chargeable party.
		DELETE	Delete an existing chargeable party transaction resource
Event Notification	{notificationUrl}	POST	Notify the bearer level event(s) from the SCEF to the SCS/AS

5.5.3.2 Resource: Chargeable Party Transactions

5.5.3.2.1 Introduction

This resource allows the SCS/AS to read all active long-term transactions related to setting a chargeable party and create individual long-term transactions.

5.5.3.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-chargeable-party/v1/{scsAsId}/transactions/**

This resource shall support the resource URI variables defined in table 5.5.3.2.2-1.

Table 5.5.3.2.2-1: Resource URI variables for resource "Chargeable Party Transactions"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.5.3.2.3 Resource methods

5.5.3.2.3.1 GET

The GET method allows to read all active chargeable party transactions for a given SCS/AS. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.5.3.2.3.1-1 and table 5.5.3.2.3.1-2.

Table 5.5.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.5.3.2.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	array(ChargeableParty)	0..N	200 OK	The chargeable party transactions information for the SCS/AS in the request URI are returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.5.3.2.3.2 PUT

This HTTP method is not supported for the resource.

5.5.3.2.3.3 PATCH

This HTTP method is not supported for the resource.

5.5.3.2.3.4 POST

The POST method creates a new chargeable party transaction resource for a given SCS/AS. The SCS/AS shall initiate the HTTP POST request message and the SCEF shall respond to the message. The SCEF shall construct the URI of the created resource.

This method shall support request and response data structures, and response codes, as specified in the table 5.5.3.2.3.4-1.

Table 5.5.3.2.3.4-1: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality	Remarks	
	ChargeableParty	1	Parameters to create a chargeable party transaction with the SCEF.	
Response body	Data type	Cardinality	Response codes	Remarks
	ChargeableParty	1	201 Created	The transaction was created successfully. The URI of the created resource shall be returned in the "Location" HTTP header.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.5.3.2.3.5 DELETE

This HTTP method is not supported for the resource.

5.5.3.3 Resource: Individual Chargeable Party Transaction

5.5.3.3.1 Introduction

This resource allows the SCS/AS to configure a chargeable party for some application flows using a long-term transaction.

5.5.3.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp-chargeable-party/v1/{scsAsId}/transactions/{transactionId}**

This resource shall support the resource URI variables defined in table 5.5.3.3.2-1.

Table 5.5.3.3.2-1: Resource URI variables for resource "Individual Chargeable Party Transaction"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
transactionId	Identifier of the transaction of type string. The transactionId corresponds to the stage 2 TLTRI.

5.5.3.3.3 Resource methods

5.5.3.3.3.1 GET

The GET method allows to read a transaction resource to obtain details of an active. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.5.3.3.3.1-1 and table 5.5.3.3.3.1-2.

Table 5.5.3.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.5.3.3.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	ChargeableParty	1	200 OK	The chargeable party transactions information related to the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.5.3.3.3.2 PUT

This HTTP method is not supported for the resource.

5.5.3.3.3.3 PATCH

The PATCH method allows to change the sponsoring status of an active chargeable party transaction. It also allows to activate a background data tranfer policy. The SCS/AS shall initiate the HTTP PATCH request message and the SCEF shall respond to the message.This method shall support request and response data structures, and response codes, as specified in the table 5.5.3.3.3.3-1.

Table 5.5.3.3.3.3-1: Data structures supported by the PATCH request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>ChargeablePartyPatch</td><td>1</td><td colspan="2">Sponsor status change, usage threshold change and/or background data transfer policy activation.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>ChargeableParty</td><td>1</td><td>200 OK</td><td>The chargeable party transaction resource was modified successfully.
The SCEF shall return a representation of the updated chargeable party transaction resource in the response payload body.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.5.3.3.3.4 POST

This HTTP method is not supported for the resource.

5.5.3.3.3.5 DELETE

The DELETE method allows to delete an active chargeable party transaction resource and to terminate the related chargeable party transaction. The SCS/AS shall initiate the HTTP DELETE request message and the SCEF shall respond to the message.

This method shall support request and response data structures, and response codes, as specified in the table 5.5.3.3.3.5-1.

Table 5.5.3.3.3.5-1.: Data structures supported by the DELETE request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>none</td><td></td><td>204 No Content</td><td>The subscription was deleted successfully.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.5.3.4 Event Notification

5.5.3.4.1 Introduction

This resource allows the SCEF to notify the SCS/AS of the bearer level event(s).

5.5.3.4.2 Resource definition

Resource URI: **{notificationUrl}**

This resource shall support the resource URI variables defined in table 5.5.3.4.2-1.

Table 5.5.3.4.2-1: Resource URI variables for resource "Event Notification"

Name	Definition
notificationUrl	Reference provided by the SCS/AS when the SCS/AS requests to sponsor the traffic from the beginning or to become the chargeable party at a later point.

5.5.3.4.3 Resource methods

5.5.3.4.3.1 Notification via HTTP POST

The POST method allows to notify SCS/AS of the bearer level event(s) by the SCEF and the SCS/AS shall respond to the message.

This method shall support request and response data structures, and response codes, as specified in the table 5.5.3.4.3.1-1.

Table 5.5.3.4.3.1-1.: Data structures supported by the POST request/response by the resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	NotificationData	1	Representation of the bearer level notification.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	n/a	1	200 OK	This case represents a successful notification of bearer level event(s).
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.5.3.4.3.2 Notification via Websocket

If supported by both SCS/AS and SCEF and successfully negotiated, the NiddConfigurationStatusNotification may alternatively be delivered through the Websocket mechanism as defined in subclause 5.2.5.4.

5.5.4 Used Features

The table below defines the features applicable to the ChargeableParty API. Those features are negotiated as described in subclause 5.2.7.

Table 5.5.4-1: Features used by ChargeableParty API

Feature Number	Feature	Description
1	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 5.2.5.4. This feature requires that the Notification_test_event feature is also supported.
2	Notification_test_event	The testing of notification connection is supported according to subclause 5.2.5.3.
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.6 NIDD API

5.6.1 Overview

The NIDD API is a RESTful API that allows the SCS/AS to send non-IP data to the UE or receive non-IP data from the UE. The NIDD API defines a set of data models, resources and the related procedures for the non-IP data transfer. The corresponding JSON schema for the representation of the resources and operations defined by the NIDD API is provided in its complete form in Annex A.6.

5.6.2 Data model

5.6.2.1 Resource data types

5.6.2.1.1 Introduction

This clause defines data structures to be used in resource representations, including subscription resources.

Table 5.6.2.1.1-1 specifies data types re-used by the NIDD API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the NIDD API.

Table 5.5.2.1.1-1: NIDD API re-used Data Types

Data type	Reference	Comments
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.6.4-1.

5.6.2.1.2 Type: NiddConfiguration

This type represents the configuration for NIDD. The same structure is used in the configuration request and configuration response.

Table 5.6.2.1.2-1: Definition of type NiddConfiguration

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses	
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
externalId	ExternalId	0..1	Each element uniquely identifies a user as defined in subclause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 2)	
msisdn	Msisdn	0..1	Each element identifies the MS internal PSTN/ISDN number allocated for a UE. (NOTE 2)	
externalGroupId	ExternalGroupId	0..1	Identifies a user group as defined in subclause 4.6.3 of 3GPP TS 23.682 [2]. (NOTE 2)	GroupMessageDelivery
duration	DateTime	0..1	Identifies the absolute time at which the related NIDD Configuration request is considered to expire, as specified in subclause 5.13.2 of 3GPP TS 23.682 [2]. When omitted in the request, it indicates the configuration is requested to be valid forever by the SCS/AS. When omitted in the response, it indicates the configuration is set to valid forever by the SCEF.	
reliableDataService	boolean	0..1	The reliable data service (as defined in subclause 4.5.15.3 of 3GPP TS 23.682 [2]) to indicate if a reliable data service acknowledgment is enabled or not.	
rdsPorts	array(RdsPort)	0..N	Indicates the port configuration that is used for reliable data transfer between specific applications using RDS (as defined in subclause 5.2.4 and 5.2.5 of 3GPP TS 24.250 [31]).	
pdnEstablishmentOption	PdnEstablishmentOptions	0..1	Indicate what the SCEF should do if the UE has not established the PDN connection and MT non-IP data needs to be sent. (wait for the UE to establish the PDN connection, respond with an error cause, or send a device trigger; see step 2 of the MT NIDD Procedure in clause 5.13.3 of 3GPP TS 23.682 [2]) The SCEF will use the value as the default preference from the SCS/AS when handling all MT non-IP packets associated with the NIDD connection.	
notificationDestination	Link	1	An URI of a notification destination that T8 message shall be delivered to.	
requestTestNotification	boolean	0..1	Set to true by the SCS/AS to request the SCEF to send a test notification as defined in subclause 5.2.5.3. Set to false or omitted otherwise.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 5.2.5.4.	Notification_websocket

maximumPacketSize	integer	0..1	The Maximum Packet Size is the maximum NIDD packet size that was transferred to the UE by the SCEF in the PCO, see subclause 4.5.14.1 of 3GPP TS 23.682 [2]. If no maximum packet size was provided to the UE by the SCEF, the SCEF sends a default configured max packet size to SCS/AS. Unit: bit.	
niddDownlinkDataTransfers	array(NiddDownlinkDataTransfer)	0..N	The downlink data deliveries that needed to be executed by the SCEF. The cardinality of the property shall be 0..1 in the request and 0..N in the response (i.e. response may contain multiple buffered MT NIDD). For GroupMessageDelivery feature, this property is only applicable for the configuration response to GET request.	
Status	NiddStatus	0..1	May be supplied by the SCEF	
NOTE 1: Properties marked with a feature as defined in subclause 5.6.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features unless stated otherwise. NOTE 2: One of the properties "externalIds", "msisdns" or "externalGroupId" shall be included.				

5.6.2.1.3 Type: NiddDownlinkDataTransfer

This type represents received NIDD downlink data from the SCS/AS.

Table 5.6.2.1.3-1: Definition of type NiddDownlinkDataTransfer

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
externalId	ExternalId	0..1	Each element uniquely identifies a user as defined in subclause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 2)	
externalGroupId	ExternalGroupId	0..1	Identifies a user group as defined in subclause 4.6.3 of 3GPP TS 23.682 [2]. (NOTE 2)	GroupMessageDelivery
msisdn	Msisdn	0..1	Each element identifies the MS internal PSTN/ISDN number allocated for a UE. (NOTE 2)	
Self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses that include an object of NiddDownlinkDataTransfer type	
data	Binary	1	The non-IP data that needed to be delivered to UE from the SCS/AS.	
reliableDataService	boolean	0..1	The reliable data service (as defined in subclause 4.5.15.3 of 3GPP TS 23.682 [2]) to indicate if a reliable data service acknowledgment is enabled or not.	
rdsPort	RdsPort	0..1	Indicates the port configuration that is used for reliable data transfer between specific applications using RDS (as defined in subclause 5.2.4 and 5.2.5 of 3GPP TS 24.250 [31]).	
maximumLatency	DurationSec	0..1	It is used to indicate maximum delay acceptable for downlink data and may be used to configure the buffer duration; a Maximum Latency of 0 indicates that buffering is not allowed. If not provided, the SCEF determines the acceptable delay based on local polices.	
Priority	integer	0..1	It is used to indicate the priority of the non-IP data packet relative to other non-IP data packets. For GroupMessageDelivery feature, this property is not applicable.	
pdnEstablishmentOption	PdnEstablishmentOptions	0..1	Indicate what the SCEF should do if the UE has not established the PDN connection and MT non-IP data needs to be sent (wait for the UE to establish the PDN connection, respond with an error cause, or send a device trigger; see step 2 of the MT NIDD Procedure in subclause 5.13.3 of 3GPP TS 23.682 [2]) If PDN Connection Establishment Option is not provided with the non-IP packet, the SCEF uses the PDN Connection Establishment Option that was provided during NIDD Configuration to decide how to handle the absence of a PDN connection.	
deliveryStatus	DeliveryStatus	0..1	Indicates the MT NIDD delivery status.	

requestedRetransmissionTime	DateTime	0..1	Identifies the absolute time at which the SCEF is expected to retransmit the non-IP data when the deliveryStatus indicates that the non-IP data is buffered in the SCEF. If no buffering indication is returned in the deliveryStatus, it identifies the absolute time at which the UE will be reachable. This parameter may be supplied by the SCEF for delivery status "FAILURE_TEMORARILY_NOT_REACHABLE" or "BUFFERING_TEMORARILY_NOT_REACHABLE"	
NOTE 1: Properties marked with a feature as defined in subclause 5.6.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features unless stated otherwise. NOTE 2: One of the properties "externalId", "msisdn" or "externalGroupId" shall be included.				

5.6.2.1.4 Type: NiddUplinkDataNotification

This type represents NIDD uplink data to be notified to the SCS/AS.

Table 5.6.2.1.4-1: Definition of type NiddUplinkDataNotification

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
niddConfiguration	Link	1	Link to the NIDD configuration resource to which this notification is related.	
externalId	ExternalId	0..1	Each element uniquely identifies a user as defined in Clause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 2)	
msisdn	Msisdn	0..1	Each element identifies the MS internal PSTN/ISDN number allocated for a UE. (NOTE 2)	
data	Binary	1	The non IP data that needed to be delivered from the UE to the SCS/AS.	
reliableDataService	boolean	0..1	Indicates whether the reliable data service is enabled.	
rdsPort	RdsPort	0..1	Indicates the port configuration that is used for reliable data transfer between specific applications using RDS (as defined in clause 5.2.4 and 5.2.5 of 3GPP TS 24.250 [31]).	
NOTE 1: Properties marked with a feature as defined in subclause 5.6.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features. NOTE 2: One of the properties "externalIds" or "msisdns" shall be included.				

5.6.2.1.5 Type: NiddDownlinkDataDeliveryStatusNotification

This type represents the delivery status for a specific NIDD downlink data delivery.

Table 5.6.2.1.5-1: Definition of type NiddDownlinkDataDeliveryStatusNotification

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
niddDownlinkDataTransfer	Link	1	Link to the NIDD downlink data transfer resource to which this notification is related.	
deliveryStatus	DeliveryStatus	1	Indicates the MT NIDD delivery status.	
requestedRetransmissionTime	DateTime	0..1	Identifies the absolute time at which the UE will be reachable. This parameter may be supplied by the SCEF for delivery status "FAILURE_TEMORARILY_NOT_REACHABLE"	
NOTE: Properties marked with a feature as defined in subclause 5.6.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.6.2.1.6 Type: NiddConfigurationStatusNotifiation

This type represents an NIDD configuration status notification.

Table 5.6.2.1.6-1: Definition of type NiddConfigurationStatusNotification

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
niddConfiguration	Link	1	Link to the NIDD configuration resource to which this notification is related.	
externalId	ExternalId	0..1	Each element uniquely identifies a user as defined in Clause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 2)	
msisdn	Msisdn	0..1	Each element identifies the MS internal PSTN/ISDN number allocated for a UE. (NOTE 2)	
status	NiddStatus	1	Indicates the NIDD configuration status.	
NOTE 1: Properties marked with a feature as defined in subclause 5.6.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features. NOTE 2: One of the properties "externalId" or "msisdn" shall be included.				

5.6.2.1.7 Type: NiddConfigurationPatch

This type represents an NIDD configuration used in PATCH.

Table 5.6.2.1.7-1: Definition of type NiddConfigurationPatch

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
duration	DateTimeRm	0..1	Identifies the absolute time at which the related NIDD Configuration request is considered to expire, as specified in subclause 5.13.2 of 3GPP TS 23.682 [2]. When set to null in the PATCH request, it indicates the configuration is requested to be valid forever by the SCS/AS. When omitted in the response, it indicates the configuration is set to valid forever by the SCEF.	
reliableDataService	boolean	0..1	The reliable data service (as defined in clause 4.5.15.3 of 3GPP TS 23.682 [2]) to indicate if a reliable data service acknowledgment is enabled or not.	
rdsPorts	array(RdsPort)	0..N	Indicates the port configuration that is used for reliable data transfer between specific applications using RDS (as defined in clause 5.2.4 and 5.2.5 of 3GPP TS 24.250 [31]).	
pdnEstablishmentOption	PdnEstablishmentOptionsRm	0..1	Indicate what the SCEF should do if the UE has not established the PDN connection and MT non-IP data needs to be sent. (wait for the UE to establish the PDN connection, respond with an error cause, or send a device trigger; see step 2 of the MT NIDD Procedure in clause 5.13.3 of 3GPP TS 23.682 [2]) The SCEF will use the value as the default preference from the SCS/AS when handling all MT non-IP packets associated with the NIDD connection.	
NOTE: Properties marked with a feature as defined in subclause 5.6.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.6.2.1.8 Type: GmdNiddDownlinkDataDeliveryStatusNotification

This type represents the delivery status for a specific group NIDD downlink data delivery.

Table 5.6.2.1.8-1: Definition of type GmdNiddDownlinkDataDeliveryNotification

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
niddDownlinkDataTransfer	Link	1	Link to the NIDD downlink data transfer resource to which this notification is related.	
gmdResults	array(GmdResult)	1..N	Indicates the group message delivery result.	GroupMessageDelivery
NOTE: Properties marked with a feature as defined in subclause 5.6.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.6.2.2 Referenced structured data types

5.6.2.2.1 Introduction

This clause defines structured data types that are referenced from data structures defined in the previous clauses.

5.6.2.2.2 Type: RdsPort

This type represents the port configuration for Reliable Data Transfer. It shall comply with the provisions defined in table 5.6.2.2.2-1.

Table 5.6.2.2.2-1: Definition of RdsPort data Type

Attribute name	Data type	Cardinality	Description
portUE	Port	1	Indicates the port number on UE that is used for reliable data transfer with a specific application on UE using RDS (as defined in clause 5.2.4 and 5.2.5 of 3GPP TS 24.250 [31]).
portSCEF	Port	1	Indicates the port number on SCEF that is used for reliable data transfer with a specific application on SCEF using RDS (as defined in clause 5.2.4 and 5.2.5 of 3GPP TS 24.250 [31]).

5.6.2.2.3 Type: GmdResult

This type represents the group message delivery result. It shall comply with the provisions defined in table 5.6.2.2.3-1.

Table 5.6.2.2.3-1: Definition of GmdResult data Type

Attribute name	Data type	Cardinality	Description
externalId	ExternalId	0..1	Each element uniquely identifies a user as defined in sublause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE)
msisdn	Msisdn	0..1	Each element identifies the MS internal PSTN/ISDN number allocated for a UE. (NOTE)
deliveryStatus	DeliveryStatus	1	Indicates the MT NIDD delivery status.
requestedRetransmissionTime	DateTime	0..1	Identifies the absolute time at which the UE will be reachable. This parameter may be supplied by the SCEF for delivery status "FAILURE_TEMORARILY_NOT_REACHABLE"
NOTE: One of the properties "externalId" or "msisdn" shall be included.			

5.6.2.3 Referenced simple data types and enumerations

5.6.2.3.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses. In addition, data types and enumerations defined in subclause 5.2.1 can be referenced.

5.6.2.3.2 Simple data types

The simple data types defined in table 5.6.2.3.2-1 shall be supported.

Table 5.6.2.3.2-1: Simple data types

Type name	Description

5.6.2.3.3 Enumeration: PdnEstablishmentOptions

The enumeration PdnEstablishmentOptions represents PDN establishment options that describe the network behaviour when there is no PDN connection towards the addressed UE.

Table 5.6.2.3.3-1: Enumeration PdnEstablishmentOptions

Enumeration value	Description	Applicability (NOTE)
WAIT_FOR_UE	wait for the UE to establish the PDN connection	
INDICATE_ERROR	respond with an error cause	
SEND_TRIGGER	send a device trigger	
NOTE: Properties marked with a feature as defined in subclause 5.6.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.		

5.6.2.3.4 Enumeration: DeliveryStatus

The enumeration DeliveryStatus represents the status of a downlink NIDD data delivery resource.

Table 5.6.2.3.4-1: Enumeration DeliveryStatus

Enumeration value	Description	Applicability (NOTE 1)
SUCCESS	Success but details not provided	
SUCCESS_NEXT_HOP_ACKNOWLEDGED	Successful delivery to the next hop with acknowledgment.	
SUCCESS_NEXT_HOP_UNACKNOWLEDGED	Successful delivery to the next hop without acknowledgment	
SUCCESS_ACKNOWLEDGED	Reliable delivery was acknowledged by the UE	
SUCCESS_UNACKNOWLEDGED	Reliable delivery was not acknowledged by the UE	
TRIGGERING	The SCEF is triggering the device and buffering the data.	
BUFFERING	The SCEF is buffering the data due to no PDN connection established.	
BUFFERING_TEMPORARILY_NOT_REACHABLE	The SCEF has been informed that the UE is temporarily not reachable but is buffering the data	
SENDING	The SCEF has forwarded the data, but they may be stored elsewhere. (NOTE 2)	
STOPPED_BUT_TRIGGERING	The SCEF is triggering the device but did not buffer the data. The SCS AS may resubmit the data	
FAILURE	Delivery failure but details not provided	
FAILURE_QUOTA_EXCEEDED	Not enough quota for the MT NIDD	
FAILURE_RATE_EXCEEDED	MT NIDD sending rate is exceeded.	
FAILURE_DATA TOO_LARGE	The non-IP data size is larger than "maximumPacketSize" of the NIDD configuration.	
FAILURE_TEMORARILY_NOT_REACHABLE	The SCEF has aborted the delivery because the UE is temporarily not reachable. The SCEF may in addition indicate a requested re-submission time for the data.	
FAILURE_NEXT_HOP	Unsuccessful delivery to the next hop.	
FAILURE_TIMEOUT	Unsuccessful delivery due to timeout.	
NOTE 1: Properties marked with a feature as defined in subclause 5.6.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features. NOTE 2: This value may be included by the SCEF to indicate the failure reason for 409 Conflict.		

5.6.2.3.5 Enumeration: NiddStatus

The enumeration NiddStatus represents the status of a NIDD configuration.

Table 5.6.2.3.5-1: Enumeration NiddStatus

Enumeration value	Description	Applicability (NOTE)
ACTIVE	The NIDD configuration is active.	
TERMINATED_UE_NOT_AUTHORIZED	The NIDD configuration was terminated because the UE´s authorisation was revoked.	
TERMINATED	The NIDD configuration was terminated.	
NOTE: Properties marked with a feature as defined in subclause 5.6.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.		

5.6.2.3.6 Enumeration: PdnEstablishmentOptionsRm

The enumeration PdnEstablishmentOptionsRm represents PDN establishment options that describe the network behaviour when there is no PDN connection towards the addressed UE. It is defined in subclause 5.6.2.3.3 but with "nullable: true" property so it can be removed in "JSON Merge Patch", as defined in IETF RFC 7386 [39].

Table 5.6.2.3.6-1: Enumeration PdnEstablishmentOptionsRm

Enumeration value	Description	Applicability (NOTE)
WAIT_FOR_UE	wait for the UE to establish the PDN connection	
INDICATE_ERROR	respond with an error cause	
SEND_TRIGGER	send a device trigger	
NOTE: Properties marked with a feature as defined in subclause 5.6.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.		

5.6.3 Resource structure

5.6.3.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-nidd/v1/

"apiRoot" is set as described in subclause 5.2.4. "apiName" shall be set to "3gpp-nidd" and "apiVersion" shall be set to "v1" for the version defined in the present document. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.6.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	HTTP initiator	Meaning
NIDD configurations	3gpp-nidd/v1/{scsAsId}/configurations	GET	SCS/AS	Read all NIDD configuration resources for a given SCS/AS
		POST	SCS/AS	Create a new NIDD configuration resource.
Individual NIDD configuration	3gpp-nidd/v1/{scsAsId}/configurations/{configurationId}	PATCH	SCS/AS	Modify an existing NIDD configuration resource
		GET	SCS/AS	Read an NIDD configuration resource
		DELETE	SCS/AS	Delete an existing NIDD configuration resource
NIDD downlink data deliveries	3gpp-nidd/v1/{scsAsId}/configurations/{configurationId}/downlink-data-deliveries	GET	SCS/AS	Read all pending NIDD downlink data delivery resources related to a particular NIDD configuration resource.
		POST	SCS/AS	Create an NIDD downlink data delivery resource related to a particular NIDD configuration resource.
Individual NIDD downlink data delivery	3gpp-nidd/v1/{scsAsId}/configurations/{configurationId}/downlink-data-deliveries/{downlinkDataDeliveryId}	PUT	SCS/AS	Replace an NIDD downlink data delivery resource.
		DELETE	SCS/AS	Delete an NIDD downlink data delivery resource.
		GET	SCS/AS	Read pending NIDD downlink data delivery resource
NIDD Configuration Update Notification	{notification_uri}	POST	SCEF	Send notifications about the status of an NIDD configuration to the SCS/AS.
NIDD Downlink Data Delivery Status Notification	{notification_uri}	POST	SCEF	Report a specific NIDD downlink data delivery result to the SCS/AS.
NIDD Uplink Data Notification	{notification_uri}	POST	SCEF	Send an uplink non-IP data notification from the SCEF to the SCS/AS.

5.6.3.2 Resource: NIDD Configurations

5.6.3.2.1 Introduction

This resource allows the SCS/AS to create an NIDD configuration at the SCEF, and read all NIDD configurations in the SCEF.

5.6.3.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-nidd/v1/{scsAsId}/configurations/**

This resource shall support the resource URI variables defined in table 5.6.3.2.2-1.

Table 5.6.3.2.2-1: Resource URI variables for resource "NIDD Configurations"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.6.3.2.3 Resource methods

5.6.3.2.3.1 GET

The GET method allows to read all active NIDD configurations for a given SCS/AS. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.6.3.2.3.1-1 and table 5.6.3.2.3.1-2.

Table 5.6.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.6.3.2.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	array(NiddConfiguration)	0..N	200 OK	The configuration information for the SCS/AS in the request URI are returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.6.3.2.3.2 PUT

This HTTP method is not supported for the resource.

5.6.3.2.3.3 PATCH

This HTTP method is not supported for the resource.

5.6.3.2.3.4 POST

To create a NIDD configuration, the SCS/AS shall use the HTTP POST method on the "configurations" collection resource as follows:

- the body of the message is encoded in JSON format with the data structure defined in table 5.6.2.1.2-1.

The possible response messages from the SCEF, depending on whether the POST request is successful or unsuccessful, are shown in Table 5.6.3.2.3.4-1.

Table 5.6.3.2.3.4-1: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality	Remarks	
	NiddConfiguration	1	Parameters to create and authorize a NIDD configuration with the SCEF.	
Response body	Data type	Cardinality	Response codes	Remarks
	NiddConfiguration	1	201 Created	The NIDD configuration was created successfully. The SCEF shall return a data structure of type "NiddConfiguration" in the response payload body. The URI of the created resource shall be returned in the "Location" HTTP header.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1				

5.6.3.2.3.5 DELETE

This HTTP method is not supported for the resource.

5.6.3.3 Resource: Individual NIDD Configuration

5.6.3.3.1 Introduction

This resource allows the SCS/AS to query/update/cancel the specific NIDD configuration at the SCEF.

5.6.3.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp-nidd/v1/{scsAsId}/configurations/{configurationId}**

This resource shall support the resource URI variables defined in table 5.6.3.3.2-1.

Table 5.6.3.3.2-1: Resource URI variables for resource "Individual NIDD Configuration"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
configurationId	Identifier of the configuration of type string. The transactionId corresponds to the stage 2 TLTRI.

5.6.3.3.3 Resource methods

5.6.3.3.3.1 GET

The GET method allows to read a NIDD configuration resource to obtain details of an active configuration. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.6.3.3.3.1-1 and table 5.6.3.3.3.1-2.

Table 5.6.3.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.6.3.3.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	None			
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	NiddConfiguration	1	200 OK	The configuration information related to the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.6.3.3.3.2 PUT

This HTTP method is not supported for the resource.

5.6.3.3.3.3 PATCH

Assuming that a NIDD configuration has been created using the HTTP POST method described in subclause 5.6.3.2.3.4, partial updating of its properties can be performed by the SCS/AS by using the HTTP PATCH method on the "configuration" instance resource as follows:

- the body of the message is encoded in JSON format with the data structure defined in table 5.6.2.1.2-1

The possible response messages from the SCEF, depending on whether the PATCH request is successful or unsuccessful, are shown in Table 5.6.3.3.3.3-1.

Table 5.6.3.3.3.3-1: Data structures supported by the PATCH request/response by the resource

Request body	Data type	Cardinality	Remarks	
	NiddConfigurationPatch	1	Parameters to update a NIDD configuration with the SCEF.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	NiddConfiguration	1	200 OK	The NIDD configuration was modified successfully. The SCEF shall return an updated data structure of type "NiddConfiguration" in the response payload body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.6.3.3.3.4 POST

This HTTP method is not supported for the resource.

5.6.3.3.3.5 DELETE

To cancel a NIDD configuration, the SCS/AS shall use the HTTP DELETE method on the individual "NIDD configuration" resource which is indicated by the URI in the Location header of the HTTP POST response:

The possible response messages from the SCEF, depending on whether the DELETE request is successful or unsuccessful, are shown in Table 5.6.3.3.3.5-1.

Table 5.6.3.3.3.5-1.: Data structures supported by the DELETE request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none		.	
Response body	Data type	Cardinality	Response codes	Remarks
	NiddConfiguration	1	200 OK	The NIDD configuration was cancelled successfully. The SCEF shall return a full representation of the deleted resource including a data structure of type "NiddConfiguration" with a "TERMINATE" status in the response body.
	None		204 No Content	The NIDD configuration was cancelled successfully. The response body shall be empty.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.6.3.4 Resource: NIDD downlink data deliveries

5.6.3.4.1 Introduction

This resource allows the SCS/AS to read all pending NIDD downlink data delivery resources for a particular NIDD configuration resource.

This resource also allows the SCS/AS to create an NIDD downlink data delivery.

This resource is applicable for a single UE and a group of UEs NIDD MT delivery.

5.6.3.4.2 Resource definition

Resource URI: **{apiRoot}/3gpp-nidd/v1/{scsAsId}/configurations/{configurationId}/downlink-data-deliveries**

This resource shall support the resource URI variables defined in table 5.6.3.4.2-1.

Table 5.6.3.4.2-1: Resource URI variables for resource "NIDD Downlink Data Deliveries"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
configurationId	Identifier of the configuration of type string.

5.6.3.4.3 Resource methods

5.6.3.4.3.1 GET

The GET method allows to read all pending NIDD downlink data deliveries for a given SCS/AS and NIDD configuration. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.6.3.4.3.1-1 and table 5.6.3.4.3.1-2.

Table 5.6.3.4.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.6.3.4.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	None			
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	array(NiddDownlinkData Transfer)	0..N	200 OK	All pending NIDD downlink data deliveries for the SCS/AS and NIDD configuration in the request URI are returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.6.3.4.3.2 PUT

This HTTP method is not supported for the resource.

5.6.3.4.3.3 PATCH

This HTTP method is not supported for the resource.

5.6.3.4.3.4 POST

To deliver the downlink non-IP data, the SCS/AS shall use the HTTP POST method on the "NIDD downlink data deliveries" resource with the body of the message is encoded in JSON format with the data structure defined in table 5.6.2.1.3-1.

The possible response messages from the SCEF, depending on whether the POST request is successful or unsuccessful, are shown in Table 5.6.3.4.3.4-1.

Table 5.6.3.4.3.4-1: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality	Remarks	
	NiddDownlinkDataTransfer	1	The parameters and non-IP data for the MT delivery.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	NiddDownlinkDataTransfer	1	200 OK	The NIDD downlink data delivery was successful. The SCEF shall return a data structure of type "NiddDownlinkDataTransfer" in the response payload body.
	NiddDownlinkDataTransfer	1	201 Created	The NIDD downlink data delivery request was accepted by the SCEF, the NIDD will be performed later. The SCEF shall return a data structure of type "NiddDownlinkDataTransfer" in the response payload body, and shall return the URI of the resource representing the downlink data transfer in the "Location" header.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1. When applicable, the "cause" attribute in "ProblemDetails" data type (as defined in subclause 5.2.1.2.12) includes value defined for "DeliveryStatus" data type in subclause 5.6.2.3.4.				

5.6.3.4.3.5 DELETE

This HTTP method is not supported for the resource.

5.6.3.5 Resource: Individual NIDD downlink data delivery

5.6.3.5.1 Introduction

For NIDD MT delivery for a single UE, this resource allows the SCS/AS to read a pending NIDD downlink delivery resource, or to replace or cancel an NIDD downlink data delivery resource related to an NIDD configuration resource.

For NIDD MT delivery for a group of UEs, this resource allows the SCS/AS to read a pending NIDD downlink delivery resource related to a NIDD configuration resource.

5.6.3.5.2 Resource definition

Resource URI: **{apiRoot}/3gpp-nidd/v1/{scsAsId}/configurations/{configurationId}/downlink-data-deliveries/{downlinkDataDeliveryId}**

This resource shall support the resource URI variables defined in table 5.6.3.5.2-1.

Table 5.6.3.4.2-1: Resource URI variables for resource "Individual NIDD Downlink Data Delivery"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
configurationId	Identifier of the configuration of type Identifier.
downlinkDataDeliveryId	Identifier of the downlink data delivery of type string.

5.6.3.5.3 Resource methods

5.6.3.5.3.1 GET

The GET method allows to read a NIDD downlink data delivery resource to obtain details. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.6.3.5.3.1-1 and table 5.6.3.5.3.1-2.

Table 5.6.3.5.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.6.3.5.3.1-2: Data structures supported by the GET request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>NiddDownlinkDataTransfer</td><td>1</td><td>200 OK</td><td>Individual NIDD downlink data delivery resource is returned.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.6.3.5.3.2 PUT

To replace the buffered downlink non-IP data, the SCS/AS shall use the HTTP PUT method on the"Individual NIDD downlink data delivery" resource with the body of the message is encoded in JSON format with the data structure defined in table 5.6.2.1.3-1.

The possible response messages from the SCEF, depending on whether the PUT request is successful or unsuccessful, are shown in Table 5.6.3.5.3.2-1.

Table 5.6.3.5.3.2-1: Data structures supported by the PUT request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>NiddDownlinkDataTransfer</td><td>1</td><td colspan="2">The parameters and non-IP data for the MT delivery.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>NiddDownlinkDataTransfer</td><td>1</td><td>200 OK</td><td>The NIDD downlink data delivery was accepted by the SCEF, the NIDD will be performed later.
The SCEF shall return a data structure of type "NiddDownlinkDataTransfer" in the response payload body.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1. When applicable, the "cause" attribute in "ProblemDetails" data type (as defined in subclause 5.2.1.2.12) includes value defined for "DeliveryStatus" data type in subclause 5.6.2.3.4.</td></tr>
</table>

5.6.3.5.3.3 PATCH

This HTTP method is not supported for the resource.

5.6.3.5.3.4 POST

This HTTP method is not supported for the resource.

5.6.3.5.3.5 DELETE

To cancel a NIDD downlink data delivery, the SCS/AS shall use the HTTP DELETE method on the "Individual NIDD downlink data delivery" resource which is indicated by the URI in the Location header of the HTTP POST response:

The possible response messages from the SCEF, depending on whether the DELETE request is successful or unsuccessful, are shown in table 5.6.3.3.3.5-1.

Table 5.6.3.3.3.5-1.: Data structures supported by the DELETE request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2">.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>none</td><td></td><td>204 No Content</td><td>The NIDD downlink data delivery was cancelled successfully.
The response body shall be empty.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1. When applicable, the "cause" attribute in "ProblemDetails" data type (as defined in subclause 5.2.1.2.12) includes value defined for "DeliveryStatus" data type in subclause 5.6.2.3.4.</td></tr>
</table>

5.6.3.6 NIDD Configuration Update Notification

5.6.3.6.1 Introduction

This resource allows the SCEF to send notifications about the status of an NIDD configuration to the SCS/AS.

5.6.3.6.2 Resource definition

Resource URI: {notification_uri}

This resource shall support the resource URI variables defined in table 5.6.3.6.2-1.

Table 5.6.3.6.2-1: Resource URI variables for resource "NIDD Configuration Update Notification"

Name	Definition
notification_uri	A URI indicating the notification destination where T8 notification requests shall be delivered. This URI shall be provided within the "notificationDestination" attribute in the NiddConfiguration type.

5.6.3.6.3 Resource methods

5.6.3.6.3.1 Notification via HTTP POST

To report the status of the NIDD configuration to the SCS/AS, the SCEF shall use the HTTP POST method on the notification point as follows:

- the body of the message is encoded in JSON format with the data structure defined in table 5.6.2.1.6-1.

The possible response messages from the SCS/AS, depending on whether the POST request is successful or unsuccessful, are shown in Table 5.6.3.6.3.1-1.

Table 5.6.3.6.3.1-1: Data structures supported by the POST request/response by the resource

	Data type	Cardinality	Remarks	
Request body	NiddConfigurationStatusNotification	1	The NIDD configuration status notification.	
Response body	Data type	Cardinality	Response codes	Remarks
	Acknowledgement	1	200 OK	The successful acknowledgement of the notification with a body.
	(None)		204 No Content	The successful acknowledgement of the notification without a body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.6.3.6.4 Notification via Websocket

If supported by both SCS/AS and SCEF and successfully negotiated, the NiddConfigurationStatusNotification may alternatively be delivered through the Websocket mechanism as defined in subclause 5.2.5.4.

5.6.3.7 NIDD Downlink Data Delivery Status Notification

5.6.3.7.1 Introduction

This resource allows the SCEF to send notifications about the status of downlink NIDD data delivery to the SCS/AS. This resource is applicable for a single UE and a group of UEs NIDD MT delivery.

5.6.3.7.2 Resource definition

Resource URI: {notification_uri}

This resource shall support the resource URI variables defined in table 5.6.3.7.2-1.

Table 5.6.3.7.2-1: Resource URI variables for resource "NIDD Downlink Data Delivery Status Notification"

Name	Definition
notification_uri	A URI indicating the notification destination URI where T8 notification requests shall be delivered. This URI shall be provided within the "notificationDestination" attribute in the NiddConfiguration type.

5.6.3.7.3 Resource methods

5.6.3.7.3.1 Notification via HTTP POST

To report the delivery status of the downlink non-IP data delivery, the SCEF shall use the HTTP POST method on the notification endpoint

with the body of the message encoded in JSON format with the data structure defined in table 5.6.2.1.5-1 for a single UE or table 5.6.2.1.5-2 for a group of UEs.

The possible response messages from the SCS/AS, depending on whether the POST request is successful or unsuccessful, are shown in Table 5.6.3.7.3.1-1.

Table 5.6.3.7.3.1-1: Data structures supported by the POST request/response by the resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	NiddDownlinkDataDeliveryStatusNotification	1	The Down link data delivery status notification for a single UE.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	Acknowledgement	1	200 OK	The successful acknowledgement of the notification.
	(None)		204 No Content	The successful acknowledgement of the notification without a body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

Table 5.6.3.7.3.1-2: Data structures supported by the POST request/response by the resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	GmdNiddDownlinkDataDeliveryStatusNotification	1	The Down link data delivery status notification for a group of UEs.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	Acknowledgement	1	200 OK	The successful acknowledgement of the notification.
	(None)		204 No Content	The successful acknowledgement of the notification without a body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.6.3.7.4 Notification via Websocket

If supported by both SCS/AS and SCEF and successfully negotiated, the NiddDownlinkDataDeliveryStatusNotification may alternatively be delivered through the Websocket mechanism as defined in subclause 5.2.5.4.

5.6.3.8 NIDD Uplink Data Notification

5.6.3.8.1 Introduction

This resource allows the SCEF to send notifications about received NIDD uplink data.

5.6.3.8.2 Resource definition

Resource URI: **{notification_uri}**

This resource shall support the resource URI variables defined in table 5.6.3.8.2-1.

Table 5.6.3.8.2-1: Resource URI variables for resource "NIDD Uplink Data Notification"

Name	Definition
notification_uri	A URI indicating the notification destination URI where T8 notification requests shall be delivered. This URI shall be provided within the "notificationDestination" attribute in the NiddConfiguration type.

5.6.3.8.3 Resource methods

5.6.3.8.3.1 Notification via HTTP POST

To send the uplink non-IP data to the SCS/AS, the SCEF shall use the HTTP POST method on the notification endpoint in SCS/AS as follows:

- the body of the message is encoded in JSON format with the data structure defined in table 5.6.2.1.4-1.

The possible response messages from the SCS/AS, depending on whether the POST request is successful or unsuccessful, are shown in Table 5.6.3.8.3.1-1.

Table 5.6.3.8.3.1-1: Data structures supported by the POST request/response by the resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	NiddUplinkDataNotification	1	The parameters and non-IP data for the NIDD uplink non-IP data notification.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	Acknowledgement	1	200 OK	The successful acknowledgement of the uplink data notification
	(None)		204 No Content	The successful acknowledgement of the notification without a body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.6.3.8.4 Notification via Websocket

If supported by both SCS/AS and SCEF and successfully negotiated, the NiddUplinkDataNotification may alternatively be delivered through the Websocket mechanism as defined in subclause 5.2.5.4.

5.6.4 Used Features

The table below defines the features applicable to the NIDD API. Those features are negotiated as described in subclause 5.2.7.

Table 5.6.4-1: Features used by NIDD API

Feature Number	Feature	Description
1	GroupMessageDelivery	This feature indicates the support of group message delivery via MT NIDD as defined in subclause 5.5.3 of 3GPP TS 23.682 [2].
2	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 5.2.5.4. This feature requires that the Notification_test_event featute is also supported.
3	Notification_test_event	The testing of notification connection is supported according to subclause 5.2.5.3.
4	MT_NIDD_modification_cancellation	Modification and cancellation of an individual MT NIDD resource.
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.7 DeviceTriggering API

5.7.1 Overview

The DeviceTriggering API is a RESTful API that allows the SCS/AS to deliver specific device trigger to the SCEF; it allows the SCS/AS to replace or recall the pending device trigger via the SCEF. If the corresponding device trigger delivery report is received by the SCEF, it also allows the SCEF to indicate the trigger delivery result to the SCS/AS.

The DeviceTriggering API defines a set of data models, resources and the related procedure for the creation and management of the device triggering. The corresponding JSON schema for the representation of the resources and operations defined by the DeviceTriggering API is provided in its complete form in Annex A.7.

5.7.2 Data model

5.7.2.1 Resource data types

5.7.2.1.1 Introduction

This clause defines data structures to be used in resource representations, including subscription resources.

Table 5.7.2.1.1-1 specifies data types re-used by the DeviceTriggering API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the DeviceTriggering API.

Table 5.7.2.1.1-1: DeviceTriggering API re-used Data Types

Data type	Reference	Comments
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.7.4-1.

5.7.2.1.2 Type: DeviceTriggering

This type represents device triggering request. The same structure is used in the request and response.

Table 5.7.2.1.2-1: Definition of type DeviceTriggering

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses that include an object of DeviceTriggering type	
externalId	ExternalId	0..1	Uniquely identifies a user as defined in Clause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 2)	
msisdn	Msisdn	0..1	Identifies the MS internal PSTN/ISDN number allocated for a UE. (NOTE 2)	
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
validityPeriod	DurationSec	1	The validity time in seconds for the specific action requested.	
priority	Priority	1	Identifies the priority of the device trigger.	
applicationPortId	Port	1	This is used to uniquely identify the triggering application addressed in the device.	
triggerPayload	Binary	1	The device triggering payload.	
notificationDestination	Link	1	A URI indicating the notification destination for T8 notifications.	
requestTestNotification	boolean	0..1	Set to true by the SCS/AS to request the SCEF to send a test notification as defined in subclause 5.2.5.3. Set to false or omitted otherwise.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 5.2.5.4.	Notification_websocket
deliveryResult	DeliveryResult	0..1	The delivery result shall be included in the HTTP responses that indicate the delivery status of the device triggering.	
NOTE 1: Properties marked with a feature as defined in subclause 5.7.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features. NOTE 2: One of the properties "externalId" or "msisdn" shall be included.				

5.7.2.1.3 Type: DeviceTriggeringDeliveryReportNotification

This type represents device triggering delivery report notification.

Table 5.7.2.1.4-1: Definition of type DeviceTriggeringDeliveryReportNotification

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
transaction	Link	1	Link to the related device triggering transaction resource to which this notification is related.	
Result	DeliveryResult	1	OK, unknown or diverse failures	
NOTE: Properties marked with a feature as defined in subclause 5.3.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.7.2.2 Referenced simple data types and enumerations

5.7.2.2.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses. In addition, data types and enumerations defined in subclause 5.2.1 can be referenced.

5.7.2.2.2 Simple data types

The simple data types defined in table 5.7.2.2.2-1 shall be supported.

Table 5.7.2.2.2-1: Simple data types

Type name	Description

5.7.2.2.3 Enumeration: DeliveryResult

The enumeration Delivery Results represents the result of the delivery of a device triggering request

Table 5.7.2.2.3-1: Enumeration DeliveryResult

Enumeration value	Description	Applicability (NOTE)
SUCCESS	The SCEF includes this value in a device triggering notification. The value indicates that the device action request was successfully completed.	
UNKNOWN	The SCEF includes this value in a device triggering notification. The value indicates any unspecified errors.	
FAILURE	The SCEF includes this value in a device triggering notification. The value indicates that this trigger encountered a delivery error and is deemed permanently undeliverable.	
TRIGGERED	The SCEF includes this value in the response for a successful device triggering request. The value indicates that device triggering request is accepted by the SCEF.	
EXPIRED	The SCEF includes this value in a device triggering notification. The value indicates that the validity period expired before the trigger could be delivered.	
UNCONFIRMED	The SCEF includes this value in a device triggering notification. The value indicates that the delivery of the device action request is not confirmed.	
REPLACED	The SCEF includes this value in the response for a successful device triggering replace request. The value indicates that the device triggering replace request is accepted by the SCEF.	
TERMINATE	The SCEF includes this value in the response for a successful device triggering cancellation request. The value indicates that the delivery of the device action request is terminated by the SCS/AS.	
NOTE: Properties marked with a feature as defined in subclause 5.7.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.		

5.7.2.2.4 Enumeration: Priority

The enumeration Priority represents the priority indication for a trigger payload.

Table 5.7.2.2.4-1: Enumeration Priority

Enumeration value	Description	Applicability (NOTE)
NO_PRIORITY	This value indicates that the device trigger has no priority.	
PRIORITY	This value indicates that the device trigger has priority.	
NOTE: Properties marked with a feature as defined in subclause 5.7.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.		

5.7.3 Resource structure

5.7.3.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-device-triggering/v1/

"apiRoot" is set as described in subclause 5.2.4. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.7.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	HTTP initiator	Meaning
Device Triggering Transactions	3gpp-device-triggering/v1/{scsAsId}/transactions	GET	SCS/AS	Read all active device triggering transaction resources for a given SCS/AS
		POST	SCS/AS	Create a new device triggering transaction resource
Individual Device Triggering Transaction	3gpp-device-triggering/v1/{scsAsId}/transactions/{transactionId}	PUT	SCS/AS	Replace an existing device triggering transaction resource and the corresponding device trigger request
		GET	SCS/AS	Read a device triggering transaction resource
		DELETE	SCS/AS	Delete an existing device triggering transaction resource and cancel the device triggering
Device Triggering Delivery Report Notification	{notification_uri}	POST	SCEF	Report a device triggering delivery report to SCS/AS.

5.7.3.2 Resource: Device Triggering Transactions

5.7.3.2.1 Introduction

This resource allows the SCS/AS to read all active resources related to device triggering, and create a resource for a device triggering transaction with the SCEF.

5.7.3.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-device-triggering/v1/{scsAsId}/transactions/**

This resource shall support the resource URI variables defined in table 5.7.3.2.2-1.

Table 5.7.3.2.2-1: Resource URI variables for resource "Device Triggering Transactions"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.7.3.2.3 Resource methods

5.7.3.2.3.1 GET

The GET method allows to read all active device triggering transactions for a given SCS/AS. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in table 5.7.3.2.3.1-1 and table 5.7.3.2.3.1-2.

Table 5.7.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.7.3.2.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	array(DeviceTriggering)	0..N	200 OK	The device triggering transactions information for the SCS/AS in the request URI are returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.7.3.2.3.2 PUT

This HTTP method is not supported for the resource.

5.7.3.2.3.3 PATCH

This HTTP method is not supported for the resource.

5.7.3.2.3.4 POST

To create a long-term transaction for a device triggering, the SCS/AS shall use the HTTP POST method on the "transactions" collection resource as follows:

- the body of the message is encoded in JSON format with the data structure defined in table 5.7.2.1.2-1.

The possible response messages from the SCEF, depending on whether the POST request is successful or unsuccessful, are shown in Table 5.7.3.2.3.4-1.

Table 5.7.3.2.3.4-1: Data structures supported by the POST request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>DeviceTriggering</td><td>1</td><td colspan="2">Parameters to request a device triggering delivery.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>DeviceTriggering</td><td>1</td><td>201 Created</td><td>The long term transaction for the device triggering was created successfully.
The SCEF shall return a data structure of type "DeviceTriggering" in the response payload body.
The URI of the created resource shall be returned in the "Location" HTTP header</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1</td></tr>
</table>

5.7.3.2.3.5 DELETE

This HTTP method is not supported for the resource.

5.7.3.3 Resource: Individual Device Triggering Transaction

5.7.3.3.1 Introduction

This resource allows the SCS/AS to operate a specific pending device triggering by using a long-term transaction.

5.7.3.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp-device-triggering/v1/{scsAsId}/transactions/{transactionId}**

This resource shall support the resource URI variables defined in table 5.7.3.3.2-1.

Table 5.7.3.3.2-1: Resource URI variables for resource "Individual Device Triggering Transaction"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
transactionId	Identifier of the transaction resource of type string. The transactionId corresponds to the stage 2 TLTRI.

5.7.3.3.3 Resource methods

5.7.3.3.3.1 GET

The GET method allows to read an individual device triggering transaction resource to obtain details of an active transaction. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.7.3.3.3.1-1 and table 5.7.3.3.3.1-2.

Table 5.7.3.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.7.3.3.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	DeviceTriggering	1	200 OK	The device triggering transaction information related to the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.7.3.3.3.2 PUT

A pending device triggering delivery can be replaced by the SCS/AS. To replace the pending device triggering, the SCS/AS shall use the HTTP PUT method on the "transaction" instance resource as follows:

- the body of the message is encoded in JSON format with the data structure defined in table 5.7.2.1.2-1

The properties "msisdn" or "externalId" shall remain unchanged from previously provided value.

The possible response messages from the SCEF, depending on whether the PUT request is successful or unsuccessful, are shown in Table 5.7.3.3.3.2-1.

Table 5.7.3.3.3.2-1: Data structures supported by the PUT request/response by the resource

Request body	Data type	Cardinality	Remarks	
	DeviceTriggering	1	Parameters to replace a device triggering with the SCEF.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	DeviceTriggering	1	200 OK	The device triggering was modified successfully. The SCEF shall return an updated data structure of type "DeviceTriggering" with the "DeliveryResult" field in the response payload body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.7.3.3.3.3 PATCH

This HTTP method is not supported for the resource.

5.7.3.3.3.4 POST

This HTTP method is not supported for the resource.

5.7.3.3.3.5 DELETE

To cancel an ongoing device triggering delivery, the SCS/AS shall use the HTTP DELETE method on the individual "transaction" resource which is indicated by the URI in the Location header of the HTTP POST response:

The possible response messages from the SCEF, depending on whether the DELETE request is successful or unsuccessful, are shown in Table 5.7.3.3.3.5-1.

Table 5.7.3.3.3.5-1: Data structures supported by the DELETE request/response by the resource

	Data type	Cardinality	Response codes	Remarks
Request body	none			.
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	DeviceTriggering	1	200 OK	The Device Triggering delivery was cancelled successfully. The SCEF shall return a data structure of type "DeviceTriggering" with a "TERMINATE" status in the response body.
	None		204 No Content	The Device Triggering was cancelled successfully. The SCEF shall not return a response payload.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.7.3.4 Device Triggering Delivery Report Notification

5.7.3.4.1 Introduction

This resource allows the SCEF to send notifications about device triggering delivery report events to the SCS/AS.

5.7.3.4.2 Resource definition

Resource URI: {notification_uri}

This resource shall support the resource URI variables defined in table 5.7.3.4.2-1.

Table 5.7.3.4.2-1: Resource URI variables for resource "Device Triggering Delivery Report Notification"

Name	Definition
notification_uri	A URI indicating the notification destination URI where T8 notification requests shall be delivered. This URI can be provided within the "notificationDestination" attribute in the DeviceTriggering type. If this attribute is not provided within the DeviceTriggering type, then a preconfigured notification URI shall be used by the SCEF.

5.7.3.4.3 Resource methods

5.7.3.4.3.1 Notification via HTTP POST

To report the delivery status of the device trigging delivery, the SCEF shall use the HTTP POST method on the notification endpoint as follows:

- the body of the message is encoded in JSON format with the data structure defined in table 5.7.2.1.4-1.

The possible response messages from the SCS/AS, depending on whether the POST request is successful or unsuccessful, are shown in Table 5.7.3.4.3.1-1.

Table 5.7.3.4.3.1-1: Data structures supported by the POST request/response by the resource

	Data type	Cardinality	Response codes	Remarks
Request body	Data type	Cardinality	Remarks	
	DeviceTriggeringDelivery ReportNotification	1	The Device Triggering delivery report.	
Response body	Data type	Cardinality	Response codes	Remarks
	Acknowledgement	1	200 OK	The successful acknowledgement of the notification.
	(None)		204 No Content	The successful acknowledgement of the notification without a body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.7.3.4.3.2 Notification via Websocket

If supported by both SCS/AS and SCEF and successfully negotiated, the DeviceTriggeringDeliveryReportNotification may alternatively be delivered through the Websocket mechanism as defined in subclause 5.2.5.4.

5.7.4 Used Features

The table below defines the features applicable to the DeviceTriggering API. Those features are negotiated as described in subclause 5.2.7.

Table 5.7.4-1: Features used by DeviceTriggering API

Feature Number	Feature	Description
1	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 5.2.5.4. This feature requires that the Notification_test_event featute is also supported.
2	Notification_test_event	The testing of notification connection is supported according to subclause 5.2.5.3.
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.8 GMD via MBMS related APIs

5.8.1 Overview

There are two Group Message Delivery via MBMS related APIs defined:

- GMDviaMBMSbyMB2 API;
- GMDviaMBMSbyxMB API.

Both APIs are RESTful APIs that allow the SCS/AS to deliver the group message to the SCEF. They define a set of data models, resources and the related procedures for the creation and management of the group message delivery. The corresponding JSON schema for the representation of the resources and operations defined by the GMDviaMBMSbyMB2 API and GMDviaMBMSbyxMB API are provided in its complete form in Annex A.8.1 and Annex A.8.2, respectively.

5.8.2 GMDviaMBMSbyMB2 API

5.8.2.1 Data model

5.8.2.1.1 Resource data types

5.8.2.1.1.1 Introduction

This clause defines data structures to be used in resource representations.

Table 5.8.2.1.1.1-1 specifies data types re-used by the GMDviaMBMSbyMB2 API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the GMDviaMBMSbyMB2 API.

Table 5.8.2.1.1.1-1: GMDviaMBMSbyMB2 API re-used Data Types

Data type	Reference	Comments
GeographicArea	3GPP TS 29.572 [42]	Identifies the geographical information of the user(s).
CivicAddress	3GPP TS 29.572 [42]	Identifies the civic address information of the user(s).
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.8.2.3-1.

5.8.2.1.1.2 Type: TMGIAllocation

This type represents TMGI Allocation request. The same structure is used in the request and response.

Table 5.8.2.1.1.2-1: Definition of type TMGIAllocation

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses that include an object of TMGIAllocation type	
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
externalGroupId	ExternalGroupId	0..1	Identifies a user group as defined in subclause 4.6.2 of 3GPP TS 23.682 [2].	
mbmsLocArea	MbmsLocArea	0..1	Restricts the distribution of the group message.	
Tmgi	string	0..1	Identifies a particular MBMS bearer service.	
tmgiExpiration	DateTime	0..1	Identifies the absolute time at which the TMGI is considered to expire.	
NOTE: Properties marked with a feature as defined in subclause 5.8.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.8.2.1.1.3 Type: GMDViaMBMSByMb2

This type represents the group message delivery via MBMS by MB2.

Table 5.8.2.1.1.3-1: Definition of type GMDViaMBMSByMb2

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses that include an object of GroupMessageDeliveryViaMBMS type	
notificationDestination	Link	1	A URI indicating the notification destination where T8 notification requests shall be delivered	
requestTestNotification	boolean	0..1	Set to true by the SCS/AS to request the SCEF to send a test notification as defined in subclause 5.2.5.3. Set to false or omitted otherwise.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 5.2.5.4.	Notification_websocket
externalGroupId	ExternalGroupId	0..1	Identifies a user group as defined in subclause 4.6.2 of 3GPP TS 23.682 [2].	
mbmsLocArea	MbmsLocArea	0..1	Restricts the distribution of the group message.	
messageDeliveryStartTime	DateTime	0..1	Identifies the absolute time at which the SCS/As starts to distribute the data. If absent, it indicates the message shall be sent immediately.	
groupMessagePayload	Binary	0..1	Indicates the payload the SCS/AS intends to deliver to the Ues.	
scefMessageDeliveryIPv4	Ipv4Addr	0..1	Indicates the Ipv4 address where the SCEF wants to receive the data.	
scefMessageDeliveryIPv6	Ipv6Addr	0..1	Indicates the Ipv6 address where the SCEF wants to receive the data.	
scefMessageDelivery Port	integer	0..1	Indicates the port number where the SCEF wants to receive the data.	
NOTE: Properties marked with a feature as defined in subclause 5.8.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.8.2.1.1.4 Type: GMDByMb2Notification

This type represents the group message delivery notification.

Table 5.8.2.1.1.4-1: Definition of type GMDByMb2Notification

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
transaction	Link	1	Link to the transaction resource to which this notification is related.	
Tmgi	string	0..1	Identifies the TMGI.	
deliveryTriggerStatus	boolean	0..1	Indicates whether delivery of group message payload corresponding to the TMGI was successful or not	
NOTE: Properties marked with a feature as defined in subclause 5.8.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.8.2.1.1.5 Type: TMGIAllocationPatch

This type represents TMGI Allocation request. The structure is used for PATCH request.

Table 5.8.2.1.1.5-1: Definition of type TMGIAllocationPatch

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
externalGroupId	ExternalGroupId	0..1	Identifies a user group as defined in subclause 4.6.2 of 3GPP TS 23.682 [2].	
mbmsLocArea	MbmsLocArea	0..1	Restricts the distribution of the group message.	
NOTE: Properties marked with a feature as defined in subclause 5.8.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.8.2.1.1.6 Type: GMDViaMBMSByMb2Patch

This type represents group message delivery via MBMS request by MB2. The structure is used for PATCH request.

Table 5.8.2.1.1.6-1: Definition of the GMDViaMBMSByMb2Patch data type

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
externalGroupId	ExternalGroupId	0..1	Identifies a user group as defined in subclause 4.6.2 of 3GPP TS 23.682 [2].	
mbmsLocArea	MbmsLocArea	0..1	Restricts the distribution of the group message.	
messageDeliveryStartTime	DateTime	0..1	Identifies the absolute time at which the SCS/As starts to distribute the data. If absent, it indicates the message shall be sent immediately.	
groupMessagePayload	Binary	0..1	Indicates the payload the SCS/AS intends to deliver to the Ues.	
NOTE: Properties marked with a feature as defined in subclause 5.8.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.8.2.1.1.7 Type: MbmsLocArea

This data type represents the user location area which is sent from the SCS/AS to the SCEF by group message delivery via MBMS request.

Table 5.8.2.1.1.7-1: Definition of the MbmsLocArea data type

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
cellId	array(string)	0..N	Indicates a Cell Global Identification of the user which identifies the cell the UE is registered.	
enodeBId	array(string)	0..N	Indicates an eNodeB in which the UE is currently located.	
geographicArea	array(GeographicArea)	0..N	Identifies a geographic area of the user where the UE is located.	
mbmsServiceAreaId	array(string)	0..N	Identifies an MBMS Service Area Identity of the user where the UE is located.	
civicAddress	array(CivicAddress)	0..N	Identifies a civic address of the user where the UE is located.	
NOTE: Properties marked with a feature as defined in subclause 5.8.2.3 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.				

5.8.2.2 Resource structure

5.8.2.2.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-group-message-delivery-mb2 /v1/

"apiRoot" is set as described in subclause 5.2.4. "apiName" shall be set to "3gpp-group-message-delivery-mb2" and "apiVersion" shall be set to "v1" for the version defined in the present document. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.8.2.2.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	HTTP initiator	Meaning
TMGI Allocation	3gpp-group-message-delivery-mb2/v1/{scsAsId} /tmgi-allocation	GET	SCS/AS	Read all active TMGI resources for a given SCS/AS
		POST	SCS/AS	Create a new TMGI resource for a given SCS/AS

Individual TMGI Allocation	3gpp-group-message-delivery-mb2/v1/{scsAsId}
	/tmgi-allocation/{tmgi}
		PUT	SCS/AS	Replace an existing TMGI resource for a given SCS/AS and TMGI

		PATCH	SCS/AS	Modify an existing TMGI resource for a given SCS/AS and TMGI
		GET	SCS/AS	Read a TMGI allocation resource for a given SCS/AS and a TMGI
		DELETE	SCS/AS	Deallocate an existing TMGI resource for agiven SCS/AS and TMGI
GMD via MBMS by MB2	3gpp-group-message-delivery-mb2/v1/{scsAsId} /tmgi-allocation/{tmgi}/delivery-via-mbms	GET	SCS/AS	Read all group message delivery resources for a given SCS/AS and TMGI.
		POST	SCS/AS	Create a group message delivery resource for given SCS/AS and TMGI selected by the SCS/AS when MB2 is used as a southbound interface.

Individual GMD via MBMS by MB2	3gpp-group-message-delivery-mb2/v1/{scsAsId}
	/tmgi-allocation/{tmgi}/delivery-via-mbms/{transactionId}
		PUT	SCS/AS	Replace a group message delivery resource

		PATCH	SCS/AS	Modify a group message delivery resource.
		GET	SCS/AS	Read a group message delivery resource.
		DELETE	SCS/AS	Delete a group message delivery resource.
GMD via MBMS by MB2 Notification	{notificationUrl}	POST	SCEF	Report a specific group message delivery result to the SCS/AS for a given transaction Id selected by the SCEF.

5.8.2.2.2 Resource: TMGI Allocation

5.8.2.2.2.1 Introduction

This resource allows the SCS/AS to read all active TMGI or request a TMGI allocation.

5.8.2.2.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-group-message-delivery-mb2/v1/{scsAsId}/tmgi-allocation/**

This resource shall support the resource URI variables defined in table 5.8.2.2.2.2-1.

Table 5.8.2.2.2.2-1: Resource URI variables for resource "TMGI Allocation"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.8.2.2.2.3 Resource methods

5.8.2.2.2.3.1 GET

The GET method read all TMGI Allocation resource for a given SCS/AS. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.2.2.2.3.1-1.

Table 5.8.2.2.2.3.1-1: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	array(TMGIAllocation)	0..N	200 OK	The TMGI allocation for the SCS/AS in the request URI are returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1				

5.8.2.2.2.3.2 PUT

This HTTP method is not supported for the resource.

5.8.2.2.2.3.3 PATCH

This HTTP method is not supported for the resource.

5.8.2.2.2.3.4 POST

The POST method creates a new TMGI Allocation resource for a given SCS/AS. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.2.2.2.3.4-1.

Table 5.8.2.2.2.3.4-1: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality	Remarks	
	TMGIAllocation	1	Parameters to create a TMGI allocation resource	
Response body	Data type	Cardinality	Response codes	Remarks
	TMGIAllocation	1	201 Created	The creation of a TMGI allocation was created successfully. The SCEF shall return a data structure of type "TMGI Allocation" in the response payload body. On success, the HTTP response shall include a "Location" HTTP header that points to the created resource URI identified by the ScsAsId and the TMGI
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1				

5.8.2.2.2.3.5 DELETE

This HTTP method is not supported for the resource.

5.8.2.2.3 Resource: Individual TMGI Allocation

5.8.2.2.3.1 Introduction

This resource allows the SCS/AS to read an active TMGI or renew an TMGI.

5.8.2.2.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp-group-message-delivery-mb2/v1/{scsAsId}/tmgi-allocation/{tmgi}**

This resource shall support the resource URI variables defined in table 5.8.2.2.3.2-1.

Table 5.8.2.2.3.2-1: Resource URI variables for resource "Individual TMGI Allocation"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
Tmgi	TMGI.

5.8.2.2.3.3 Resource methods

5.8.2.2.3.3.1 GET

The GET method reads a TMGI Allocation resource for a given SCS/AS. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.2.2.3.3.1-1.

Table 5.8.2..2.3.3.1-1: Data structures supported by the GET request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>TMGIAllocation</td><td>1</td><td>200 OK</td><td>A TMGI allocation for the SCS/AS in the request URI is returned.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1</td></tr>
</table>

5.8.2.2.3.3.2 PUT

The PUT method replaces a TMGI Allocation resource for a given SCS/AS to renew an TMGI. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.2.2.3.3.2-1.

Table 5.8.2.2.3.3.2-1: Data structures supported by the PUT request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>TMGIAllocation</td><td>1</td><td colspan="2">Parameters to replace a TMGI resource</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>TMGIAllocation</td><td>1</td><td>200 OK</td><td>The replace of a TMGI allocation was created successfully.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1</td></tr>
</table>

5.8.2.2.3.3.3 PATCH

The PATCH method modifies a TMGI Allocation resource for a given SCS/AS to renew an expired TMGI. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.2.2.3.3.3-1.

Table 5.8.2.2.3.3.3-1: Data structures supported by the PATCH request/response by the resource

	Data type	Cardinality	Remarks	
Request body	TMGIAllocationPatch	1	Parameters to modify a TMGI resource	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	TMGIAllocation	1	200 OK	The modification of a TMGI allocation was created successfully.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1				

5.8.2.2.3.3.4 POST

This HTTP method is not supported for the resource.

5.8.2.2.3.3.5 DELETE

The DELETE method deletes a TMGI Allocation resource for a given SCS/AS to deallocate a TMGI. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.2.2.3.3.5-1.

Table 5.8.2.2.3.3.5-1: Data structures supported by the DELETE request/response by the resource

	Data type	Cardinality	Remarks	
Request body	none			
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	none		200 OK	The TMGI allocation resource was removed successfully.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1				

5.8.2.2.4 Resource: GMD via MBMS by MB2

5.8.2.2.4.1 Introduction

This resource allows the SCS/AS to read all active group message delivery resources or create a group message delivery when the MB2 is used as a southbound interface.

5.8.2.2.4.2 Resource definition

Resource URI: **{apiRoot}/3gpp-group-message-delivery-mb2/v1/{scsAsId}/tmgi-allocation/{tmgi}/delivery-via-mbms/**

This resource shall support the resource URI variables defined in table 5.8.2.2.4.2-1.

Table 5.8.2.2.4.2-1: Resource URI variables for resource "GMD via MBMS by MB2"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
Tmgi	TMGI

5.8.2.2.4.3 Resource methods

5.8.2.2.4.3.1 GET

The GET method reads all group message delivery via MBMS resources for a given SCS/AS and a TMGI. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.2.2.4.3.1-1.

Table 5.8.2.2.4.3.1-1: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	array(GMDViaMBMSByMb2)	0..N	200 OK	The TMGI allocation for the SCS/AS in the request URI are returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1				

5.8.2.2.4.3.2 PUT

This HTTP method is not supported for the resource.

5.8.2.2.4.3.3 PATCH

This HTTP method is not supported for the resource.

5.8.2.2.4.3.4 POST

The POST method creates a new group message delivery via MBMS resource for a given SCS/AS and TMGI selected by the SCS/AS. It is initiated by the SCS/AS and answered by the SCEF. The SCEF shall construct the URI of the created resource using that URI.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.2.2.4.3.4-1.

Table 5.8.2.2.4.3.4-1: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality	Remarks	
	GMDViaMBMSByMb2	1	Parameters to create and authorize a group message delivery via MBMS with the SCEF.	
Response body	Data type	Cardinality	Response codes	Remarks
	GMDViaMBMSByMb2	1	201 Created	The creation of a group message delivery was created successfully. The SCEF shall return a data structure of type "GMDViaMBMSByMb2" in the response payload body. On success, the HTTP response shall include a "Location" HTTP header that points to the created resource URI identified by the ScsAsId and the Transaction Id.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1				

5.8.2.2.4.3.5 DELETE

This HTTP method is not supported for the resource.

5.8.2.2.5 Resource: Individual GMD via MBMS by MB2

5.8.2.2.5.1 Introduction

This resource allows the SCS/AS to modify or delete a group message delivery via MBMS by MB2 resource.

5.8.2.2.5.2 Resource definition

Resource URI: **{apiRoot}/3gpp-group-message-delivery-mb2/v1/{scsAsId}/tmgi-allocation/{tmgi}/delivery-via-mbms/{transactionId}**

This resource shall support the resource URI variables defined in table 5.8.2.2.5.2-1.

Table 5.8.2.2.5.2-1: Resource URI variables for resource "Individual GMD via MBMS by MB2"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
transactionId	transactionId selected by the SCEF. The transactionId corresponds to the stage 2 TLTRI.

5.8.2.2.5.3 Resource methods

5.8.2.2.5.3.1 GET

The GET method reads a group message delivery via MBMS by MB2 resource for a given SCS/AS, a TMGI and a transactionId. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.2.2.5.3.1-1.

Table 5.8.2.2.5.3.1-1: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	GMDViaMBMSbyMb2	0..1	200 OK	The group message delivery resource for the SCS/AS in the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1				

5.8.2.2.5.3.2 PUT

Assuming that a group message delivery has been created using the HTTP POST method described in subclause 5.8.2.2.4.3.4, replace of its properties can be performed by the SCS/AS by using the HTTP PUT method on the "delivery_via_mbms" instance resource as follows:

- the body of the message is encoded in JSON format with the data structure defined in table 5.8.2.2.5.3.2-1.

The content body of the group message delivery via MBMS update request shall contain updated full representation of the group message delivery resource. Only the properties "locationinfo", "accuracy", "messageDeliverystarttime" and "groupMessagepayload" can be modified.

The possible response messages from the SCEF, depending on whether the PUT request is successful or unsuccessful, are shown in Table 5.8.2.2.5.3.2-1.

Table 5.8.2.2.5.3.2-1: Data structures supported by the PUT request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>GMDViaMBMSByMb2</td><td>1</td><td colspan="2">Parameters to replace group message delivery resource with the SCEF.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>GMDViaMBMSByMb2</td><td>1</td><td>200 OK</td><td>The group message delivery was modified successfully.
The SCEF shall return an updated data structure of type "GMDViaMBMSByMb2" in the response payload body.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.8.2.2.5.3.3 PATCH

Assuming that a group message delivery has been created using the HTTP POST method described in subclause 5.8.2.2.4.3.4, partial updating of its properties can be performed by the SCS/AS by using the HTTP PATCH method on the "delivery-via-mbms" instance resource.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.2.2.5.3.3-1.

Table 5.8.2.2.5.3.3-1: Data structures supported by the PATCH request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>GMDViaMBMSByMb2Patch</td><td>1</td><td colspan="2">Parameters to partially update a group message delivery with the SCEF.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>GMDViaMBMSByMb2</td><td>1</td><td>200 OK</td><td>The group message delivery was modified successfully.
The SCEF shall return an updated data structure of type "GMDViaMBMSByMb2" in the response payload body.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.8.2.2.5.3.4 POST

This HTTP method is not supported for the resource.

5.8.2.2.5.3.5 DELETE

To cancel a group message delivery, the SCS/AS shall use the HTTP DELETE method on the individual "delivery_via_mbms" resource which is indicated by the URI in the Location header of the HTTP POST response:

The possible response messages from the SCEF, depending on whether the DELETE request is successful or unsuccessful, are shown in Table 5.8.2.2.5.3.5-1.

Table 5.8.2.2.5.3.5-1.: Data structures supported by the DELETE request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	none		204 No Content	The group message delivery subscription was cancelled successfully.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.8.2.2.6 Resource: GMD via MBMS by MB2 Notification

5.8.2.2.6.1 Introduction

This resource allows the SCEF report the delivery trigger status to the SCS/AS to indicate whether group message delivery was triggered successful.

5.8.2.2.6.2 Resource definition

Resource URI: {notificationUrl}

This resource shall support the resource URI variables defined in table 5.8.2.2.6.2-1.

Table 5.8.2.2.6.2-1: Resource URI variables for resource "GMD via MBMS by MB2 Notification"

Name	Definition
notificationUrl	A URI indicating the notification destination where T8 notification requests shall be delivered. This URI can be provided within the field "notificationUrl" in the GMDViaMBMSByMb2 type. If this field is not provided with the GMDViaMBMSByMb2, then preconfigured notification URI shall be used by the SCEF.

5.8.2.2.6.3 Resource methods

5.8.2.2.6.3.1 Notification via POST

To report the status of the delivery trigger status to the SCS/AS, the SCEF shall use the HTTP POST method on the notification point as follows:

- the body of the message is encoded in JSON format with the data structure defined in table 5.8.2.1.1.4-1.

The possible response messages from the SCS/AS, depending on whether the POST request is successful or unsuccessful, are shown in Table 5.8.2.2.6.3.1-1.

Table 5.8.2.2.6.3.1-1: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality	Remarks	
	GMDByMb2Notification	1	The delivery status notification.	
Response body	Data type	Cardinality	Response codes	Remarks
	Acknowledgement	1	200 OK	The successful acknowledgement of the notification with a body.
	(None)		204 No Content	The successful acknowledgement of the notification without a body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.8.2.2.6.3.2 Notification via Websocket

If supported by both SCS/AS and SCEF and successfully negotiated, the Group Message Delivery via MBMS Notification may alternatively be delivered through the Websocket mechanism as defined in subclause 5.2.5.4.

5.8.2.3 Used Features

The table below defines the features applicable to the GMDviaMBMSbyMB2 API. Those features are negotiated as described in subclause 5.2.7.

Table 5.8.2.3-1: Features used by GMDviaMBMSbyMB2 API

Feature Number	Feature	Description
1	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 5.2.5.4. This feature requires that the Notification_test_event featute is also supported.
2	Notification_test_event	The testing of notification connection is supported according to subclause 5.2.5.3.
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.8.3 GMDviaMBMSbyxMB API

5.8.3.1 Data model

5.8.3.1.1 Resource data types

5.8.3.1.1.1 Introduction

This clause defines data structures to be used in resource representations.

Table 5.8.3.1.1.1-1 specifies data types re-used by the GMDviaMBMSbyxMB API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the GMDviaMBMSbyxMB API.

Table 5.8.3.1.1.1-1: GMDviaMBMSbyxMB API re-used Data Types

Data type	Reference	Comments
GeographicArea	3GPP TS 29.572 [42]	Identifies the geographical information of the user(s).
CivicAddress	3GPP TS 29.572 [42]	Identifies the civic address information of the user(s).
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.8.3.3-1.

5.8.3.1.1.2 Type: ServiceCreation

This type represents the service resource.

Table 5.8.3.1.1.2-1: Definition of the ServiceCreation data type

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses that include an object of ServiceCreation type	
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
externalGroupId	ExternalGroupId	1	Identifies a user group as defined in subclause 4.6.2 of 3GPP TS 23.682 [2] supplied by the SCS/AS.	
userServiceId	string	0..1	Identifies the MBMS User Service supplied by the SCEF.	
serviceClass	string	0..1	The service class that service belongs to supplied by the SCEF.	
serviceLanguages	array(string)	0..1	List of language of the service content supplied by the SCEF.	
serviceNames	array(string)	0..1	List of Service Names supplied by the SCEF	
receiveOnlyMode	boolean	0..1	When set to 'true', the Content Provider indicates that the service is a Receive Only Mode service. This parameter is supplied by the SCEF	
serviceAnnouncementMode	string	0..1	Enumeration of Service Announcement Mode supplied by the SCEF.	
NOTE: Properties marked with a feature as defined in subclause 5.8.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.8.3.1.1.3 Type: GMDViaMBMSByxMB

This type represents the group message delivery via MBMS by xMB.

Table 5.8.3.1.1.3-1: Definition of type GMDViaMBMSByxMB

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses that include an object of GroupMessageDeliveryViaMBMS type	
notificationDestination	Link	1	A URI indicating the notification destination where T8 notification requests shall be delivered	
requestTestNotification	boolean	0..1	Set to true by the SCS/AS to request the SCEF to send a test notification as defined in subclause 5.2.5.3. Set to false or omitted otherwise.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 5.2.5.4.	Notification_websocket
mbmsLocArea	MbmsLocArea	0..1	Restricts the distribution of the group message.	
messageDeliveryStartTime	DateTime	0..1	Identifies the absolute time at which the SCS/As starts to distribute the data. If absent, it indicates the message shall be sent immediately.	
messageDeliveryStopTime	DateTime	0..1	Identifies the absolute time at which the SCS/As is expected to stop distributing the data. If absent, configuration time will be used.	
groupMessagePayload	Binary	0..1	Indicates the payload the SCS/AS intends to deliver to the Ues.	
scefMessageDeliveryIPv4	Ipv4Addr	0..1	Indicates the Ipv4 address where the SCEF wants to receive the data.	
scefMessageDeliveryIPv6	Ipv6Addr	0..1	Indicates the Ipv6 address where the SCEF wants to receive the data.	
scefMessageDeliveryPort	integer	0..1	Indicates the port number where the SCEF wants to receive the data.	
NOTE: Properties marked with a feature as defined in subclause 5.8.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.8.3.1.1.4 Type: GMDByxMBNotification

This type represents the group message delivery notification.

Table 5.8.3.1.1.4-1: Definition of type GMDByxMbNotification

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
transaction	Link	1	Link to the transaction resource to which this notification is related.	
deliveryTriggerStatus	boolean	0..1	Indicates whether delivery of group message payload was successful or not	
NOTE: Properties marked with a feature as defined in subclause 5.8.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.8.3.1.1.5 Type: GMDViaMBMSByxMBPatch

This type represents group message delivery via MBMS request. The structure is used for PATCH request.

Table 5.8.3.1.1.5-1: Definition of the GMDViaMBMSByxMBPatch data type

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
mbmsLocArea	MbmsLocArea	0..1	Restricts the distribution of the group message.	
messageDeliveryStartTime	DateTime	0..1	Identifies the absolute time at which the SCS/As starts to distribute the data. If absent, it indicates the message shall be sent immediately.	
messageDeliveryStopTime	DateTime	0..1	Identifies the absolute time at which the SCS/As is expected to stop distributing the data. If absent, the configuration value will be used.	
groupMessagePayload	Binary	0..1	Indicates the payload the SCS/AS intends to deliver to the Ues.	
NOTE: Properties marked with a feature as defined in subclause 5.8.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.8.3.1.1.6 Type: MbmsLocArea

This data type represents the user location area which is sent from the SCS/AS to the SCEF by group message delivery via MBMS request.

Table 5.8.3.1.1.6-1: Definition of the MbmsLocArea data type

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
cellId	array(string)	0..N	Indicates a Cell Global Identification of the user which identifies the cell the UE is registered.	
enodeBId	array(string)	0..N	Indicates an eNodeB in which the UE is currently located.	
geographicArea	array(GeographicArea)	0..N	Identifies a geographic area of the user where the UE is located.	
mbmsServiceAreaId	array(string)	0..N	Identifies an MBMS Service Area Identity of the user where the UE is located.	
civicAddress	array(CivicAddress)	0..N	Identifies a civic address of the user where the UE is located.	
NOTE: Properties marked with a feature as defined in subclause 5.8.2.3 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.				

5.8.3.2 Resource structure

5.8.3.2.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-group-message-delivery-xmb /v1/

"apiRoot" is set as described in subclause 5.2.4. "apiName" shall be set to "3gpp-group-message-delivery-xmb" and "apiVersion" shall be set to "v1" for the version defined in the present document. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.8.3.2.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	HTTP initiator	Meaning
xMB Services	3gpp-group-message-delivery-xmb/v1/{scsAsId} /services/	POST	SCS/AS	Create a service when xMB is used as a southbound interface.
		GET	SCS/AS	Read all active service resource for a given SCS/AS.

Individual xMB Service	3gpp-group-message-delivery-xmb/v1/{scsAsId}
	/services/{serviceId}
		GET	SCS/AS	Read an active service resource for a given SCS/AS and Service Id.

		DELETE	SCS/AS	Delete an existing service resource for agiven SCS/AS and Service Id.
GMD via MBMS by xMB	3gpp-group-message-delivery-xmb/v1/{scsAsId} /services/{serviceId}/delivery-via-mbms	GET	SCS/AS	Read all group message delivery resources for a given SCS/AS and Service Id.
		POST	SCS/AS	Create a group message delivery resource for given SCS/AS and Service Id when xMB is used as a southbound interface.

Individual GMD via MBMS by xMB	3gpp-group-message-delivery-xmb/v1/{scsAsId}
	/services/{serviceId}/delivery-via-mbms/{transactionId}
		PUT	SCS/AS	Replace a group message delivery resource

		PATCH	SCS/AS	Modify a group message delivery resource.
		GET	SCS/AS	Read a group message delivery resource.
		DELETE	SCS/AS	Delete a group message delivery resource.
GMD via MBMS by xMB Notification	{notificationUrl}	POST	SCEF	Report a specific group message delivery result to the SCS/AS for a given Transaction Id selected by the SCEF.

5.8.3.2.2 Resource: xMB Services

5.8.3.2.2.1 Introduction

This resource allows the SCS/AS to create service resource when the xMB is used as a southbound interface.

5.8.3.2.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-group-message-delivery-xmb/v1/{scsAsId}/services**

This resource shall support the resource URI variables defined in table 5.8.3.2.2.2-1.

Table 5.8.3.2.2.2-1: Resource URI variables for resource "GMD via MBMS by xMB"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.8.3.2.2.3 Resource methods

5.8.3.2.2.3.1 GET

The GET method allows the SCS/AS read all active service resources for a given SCS/AS. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.3.2.2.3.1-1.

Table 5.8.3.2.2.3.1-1: Data structures supported by the POST request/response by the resource

	Data type	Cardinality		Remarks
Request body	none.			
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	array(ServiceCreation)	0..N	200 OK	The service resource for the SCS/AS in the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1				

5.8.3.2.2.3.2 PUT

This HTTP method is not supported for the resource.

5.8.3.2.2.3.3 PATCH

This HTTP method is not supported for the resource.

5.8.3.2.2.3.4 POST

The POST method creates a new service resource for a given SCS/AS. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.3.2.2.3.4-1.

Table 5.8.3.2.2.3.4-1: Data structures supported by the POST request/response by the resource

	Data type	Cardinality		Remarks
Request body	ServiceCreation	1		Parameters to create and authorize a service with the SCEF.
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	ServiceCreation	1	201 Created	The resource of a service was created successfully. The SCEF shall return a data structure of type "ServiceCreation" in the response payload body. On success, the HTTP response shall include a "Location" HTTP header that points to the created resource URI identified by the ScsAsId and the ServiceId.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1				

5.8.3.2.2.3.5 DELETE

This HTTP method is not supported for the resource.

5.8.3.2.3 Resource: Individual xMB Service

5.8.3.2.3.1 Introduction

This resource allows the SCS/AS to delete a service resource.

5.8.3.2.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp_device_triggering_xmb/v1/{scsAsId}/services/{serviceId}**

This resource shall support the resource URI variables defined in table 5.8.3.2.3.2-1.

Table 5.8.3.2.3.2-1: Resource URI variables for resource "Individual Service Creation"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
serviceId	Identifier of the service.

5.8.3.2.3.3 Resource methods

5.8.3.2.3.3.1 GET

The GET method reads a active service resource for a given SCS/AS and a service Id. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.3.2.3.3.1-1.

Table 5.8.3.2.3.3.1-1: Data structures supported by the GET request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>ServiceCreation</td><td>0..1</td><td>200 OK</td><td>The service resource for the SCS/AS and Service Id in the request URI is returned.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1</td></tr>
</table>

5.8.3.2.3.3.2 PUT

This HTTP method is not supported for the resource.

5.8.3.2.3.3.3 PATCH

This HTTP method is not supported for the resource.

5.8.3.2.3.3.4 POST

This HTTP method is not supported for the resource.

5.8.3.2.3.3.5 DELETE

To delete a service resource, the SCS/AS shall use the HTTP DELETE method on the "ServiceCreation" resource which is indicated by the URI in the Location header of the HTTP POST response:

The possible response messages from the SCEF, depending on whether the DELETE request is successful or unsuccessful, are shown in Table 5.8.3.2.3.3.5-1.

Table 5.8.3.2.3.3.5-1.: Data structures supported by the DELETE request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>none</td><td></td><td>200 OK</td><td>The group message delivery subscription was cancelled successfully.</td></tr>
</table>

5.8.3.2.4 Resource: GMD via MBMS by xMB

5.8.3.2.4.1 Introduction

This resource allows the SCS/AS to create a group message delivery when the xMB is used as a southbound interface.

5.8.3.2.4.2 Resource definition

Resource URI: **{apiRoot}/3gpp-group-message-delivery-xmb/v1/{scsAsId}/services/{serviceId}/delivery-via-mbms**

This resource shall support the resource URI variables defined in table 5.8.3.2.4.2-1.

Table 5.8.3.2.4.2-1: Resource URI variables for resource "GMD via MBMS by xMB"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
ServiceId	Identifier of the service

5.8.3.2.4.3 Resource methods

5.8.3.2.4.3.1 GET

This HTTP method is not supported for the resource.

5.8.3.2.4.3.2 PUT

This HTTP method is not supported for the resource.

5.8.3.2.4.3.3 PATCH

This HTTP method is not supported for the resource.

5.8.3.2.4.3.4 POST

The POST method creates a new group message delivery via MBMS resource for a given SCS/AS and service Id selected by the SCS/AS. It is initiated by the SCS/AS and answered by the SCEF. This method shall support request and response data structures, and response codes, as specified in the table 5.8.3.2.4.3.4-1.

Table 5.8.3.2.4.3.4-1: Data structures supported by the POST request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>GMDViaMBMSByxMB</td><td>1</td><td colspan="2">Parameters to create and authorize a group message delivery via MBMS with the SCEF.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>GMDViaMBMSByxMB</td><td>1</td><td>201 Created</td><td>The creation of a group message delivery was created successfully.
The SCEF shall return a data structure of type "GMDViaMBMSByxMB" in the response payload body.
On success, the HTTP response shall include a "Location" HTTP header that points to the created resource URI identified by the ScsAsId and the Transaction Id</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1</td></tr>
</table>

5.8.3.2.4.3.5 DELETE

This HTTP method is not supported for the resource.

5.8.3.2.5 Resource: Individual GMD via MBMS by xMB

5.8.3.2.5.1 Introduction

This resource allows the SCS/AS to modify or delete a group message delivery via MBMS resource.

5.8.3.2.5.2 Resource definition

Resource URI: **{apiRoot}/3gpp-device-triggering-xmb/v1/{scsAsId}/services/{serviceId}/delivery-via-mbms/{transactionId}**

This resource shall support the resource URI variables defined in table 5.8.3.2.5.2-1.

Table 5.8.3.2.5.2-1: Resource URI variables for resource "Individual GMD via MBMS by xMB"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
serviceIdId	Identifier of the service selected by the SCEF. The transactionId corresponds to the stage 2 TLTRI.

5.8.3.2.5.3 Resource methods

5.8.3.2.5.3.1 GET

The GET method reads a group message delivery via MBMS resource for a given SCS/AS, a service Id and a transactionId. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.3.2.3.3.1-1.

Table 5.8.3.2.5.3.1-1: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	GMDViaMBMSByxMB	0..1	200 OK	The group message delivery resource for the SCS/AS in the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1				

5.8.3.2.5.3.2 PUT

Assuming that a group message delivery has been created using the HTTP POST method described in subclause 5.8.3.2.4.3.4, replace of its properties can be performed by the SCS/AS by using the HTTP PUT method on the "delivery-via-mbms" instance resource as follows:

- the body of the message is encoded in JSON format with the data structure defined in table 5.8.3.1.1.3-1.

The content body of the group message delivery via MBMS update request shall contain updated full representation of the group message delivery resource. Only the properties "locationinfo", "accuracy", "messageDeliverystarttime", "messageDeliverystoptime" and "groupMessagepayload" can be modified.

The possible response messages from the SCEF, depending on whether the PUT request is successful or unsuccessful, are shown in Table 5.8.3.2.5.3.2-1.

Table 5.8.3.2.5.3.2-1: Data structures supported by the PUT request/response by the resource

	Data type	Cardinality	Remarks	
Request body	GMDViaMBMSByxMB	1	Parameters to replace group message delivery resource with the SCEF.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	GMDViaMBMSByxMB	1	200 OK	The group message delivery was modified successfully. The SCEF shall return an updated data structure of type "GMDViaMBMSByxMB" in the response payload body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.8.3.2.5.3.3 PATCH

Assuming that a group message delivery has been created using the HTTP POST method described in subclause 5.8.3.2.4.3.4, partial updating of its properties can be performed by the SCS/AS by using the HTTP PATCH method on the "delivery-via-mbms" instance resource.

This method shall support request and response data structures, and response codes, as specified in the table 5.8.3.2.5.3.3-1.

Table 5.8.3.2.5.3.3-1: Data structures supported by the PATCH request/response by the resource

	Data type	Cardinality	Remarks	
Request body	GMDViaMBMSByxMB Patch	1	Parameters to partially update a group message delivery with the SCEF.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	GMDViaMBMSByxMB	1	200 OK	The group message delivery was modified successfully. The SCEF shall return an updated data structure of type "GMDViaMBMSByxMB" in the response payload body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.8.3.2.5.3.4 POST

This HTTP method is not supported for the resource.

5.8.3.2.5.3.5 DELETE

To cancel a group message delivery, the SCS/AS shall use the HTTP DELETE method on the individual "delivery_via_mbms" resource which is indicated by the URI in the Location header of the HTTP POST response:

The possible response messages from the SCEF, depending on whether the DELETE request is successful or unsuccessful, are shown in Table 5.8.3.2.5.3.5-1.

Table 5.8.3.2.5.3.5-1.: Data structures supported by the DELETE request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	none		200 OK	The group message delivery subscription was cancelled successfully.

5.8.3.2.6 Resource: GMD via MBMS by xMB Notification

5.8.3.2.6.1 Introduction

This resource allows the SCEF report the delivery trigger status to the SCS/AS to indicate whether group message delivery was triggered successful.

5.8.3.2.6.2 Resource definition

Resource URI: {notificationUrl}

This resource shall support the resource URI variables defined in table 5.8.3.2.6.2-1.

Table 5.8.3.2.6.2-1: Resource URI variables for resource "Group Message Delivery via MBMS Notification"

Name	Definition
notificationUrl	A URI indicating the notification destination where T8 notification requests shall be delivered. This URI can be provided within the field "notificationUrl" in the GMDViaMBMSByxMB type. If this field is not provided with the GMDViaMBMSByxMB, then preconfigured notification URI shall be used by the SCEF.

5.8.3.2.6.3 Resource methods

5.8.3.2.6.3.1 Notification via POST

To report the status of the delivery trigger status to the SCS/AS, the SCEF shall use the HTTP POST method on the notification point as follows:

- the body of the message is encoded in JSON format with the data structure defined in table 5.8.3.1.1.4-1.

The possible response messages from the SCS/AS, depending on whether the POST request is successful or unsuccessful, are shown in Table 5.8.3.2.6.3.1-1.

Table 5.8.3.2.6.3.1-1: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality	Remarks	
	GMDByxMBNotification	1	The delivery status notification.	
Response body	Data type	Cardinality	Response codes	Remarks
	Acknowledgement	1	200 OK	The successful acknowledgement of the notification with a body.
	(None)		204 No Content	The successful acknowledgement of the notification without a body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.8.3.2.6.3.2 Notification via Websocket

If supported by both SCS/AS and SCEF and successfully negotiated, the Group Message Delivery via MBMS Notification may alternatively be delivered through the Websocket mechanism as defined in subclause 5.2.5.4.

5.8.3.3 Used Features

The table below defines the features applicable to the GMDviaMBMSbyxMB API. Those features are negotiated as described in subclause 5.2.7.

Table 5.8.3.3-1: Features used by GMDviaMBMSbyxMB API

Feature Number	Feature	Description
1	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 5.2.5.4. This feature requires that the Notification_test_event featute is also supported.
2	Notification_test_event	The testing of notification connection is supported according to subclause 5.2.5.3.
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.9 ReportingNetworkStatus API

5.9.1 Overview

The ReportingNetworkStatus API is a RESTful API that allows the SCS/AS to be one-time or continuous notified of the network status in a geographic area. The ReportingNetworkStatus API defines a set of data models, resources and the related procedures for the creation and management of the network status reporting request. The corresponding JSON schema for the representation of the resources and operations defined by the ReportingNetworkStatus API is provided in its complete form in Annex A.9.

5.9.2 Data model

5.9.2.1 Resource data types

5.9.2.1.1 Introduction

This clause defines data structures to be used in resource representations.

Table 5.9.2.1.1-1 specifies data types re-used by the ReportingNetworkStatus API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the ReportingNetworkStatus API.

Table 5.9.2.1.1-1: ReportingNetworkStatus API re-used Data Types

Data type	Reference	Comments
GeographicArea	3GPP TS 29.572 [42]	Identifies the geographical information of the user(s).
CivicAddress	3GPP TS 29.572 [42]	Identifies the civic address information of the user(s).
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.9.4-1.

5.9.2.1.2 Type: NetworkStatusReportingSubscription

This type represents the subscription of reporting the network status. The same structure is used in the subscription request and subscription response.

Table 5.9.2.1.2-1: Definition of type NetworkStatusReportingSubscription

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses that include an object of NetworkStatusReportingSubscription type.	
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
notificationDestination	Link	1	A URI indicating the notification destination where T8 notification requests shall be delivered	
requestTestNotification	boolean	0..1	Set to true by the SCS/AS to request the SCEF to send a test notification as defined in subclause 5.2.5.3. Set to false or omitted otherwise.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 5.2.5.4.	Notification_websocket
locationArea	LocationArea	1	Identifies a location area. It can be either a list of cell IDs, or a list of Tracking Areas, or civic addresses, or a geographic area, or a combination of any of the above.	
timeDuration	DateTme	0..1	Identifies the time for which a continuous reporting is requested. Shall not be provided for one time reporting case.	
thresholdValues	array(CongestionValue)	0..N	Identifies a list of congestion level(s) with exact value that the SCS/AS requests to be informed of when reached. (NOTE 2)	
thresholdTypes	array(CongestionType)	0..N	Identifies a list of congestion level(s) with abstracted value that the SCS/AS requests to be informed of when reached. (NOTE 2)	
NOTE 1: Properties marked with a feature as defined in subclause 5.9.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features. NOTE 2: thresholdValue and thresholdType shall be mutually exclusive.				

5.9.2.2 Notification data types

5.9.2.2.1 Introduction

This clause defines data structures to be used in notifications.

5.9.2.2.2 Type: NetworkStatusReportingNotification

This data type represents a network status reporting notification which is sent from the SCEF to the SCS/AS.

Table 5.9.2.2.2-1: Definition of type NetworkStatusReportingNotification

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
subscription	Link	1	Link to the subscription resource to which this notification is related.	
nsiValue	CongestionValue	0..1	Network Status Indicator based on exact value for congestion status received from RCAF(s). (NOTE 2)	
nsiType	CongestionType	0..1	Network Status Indicator based on abstracted value for congestion status. (NOTE 2)	
NOTE 1: Properties marked with a feature as defined in subclause 5.9.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features. NOTE 2: nsiValue and nsiType shall be mutually exclusive.				

5.9.2.3 Referenced simple data types and enumerations

5.9.2.3.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses. In addition, data types and enumerations defined in subclause 5.2.1 can be referenced.

5.9.2.3.2 Simple data types

The simple data types defined in table 5.9.2.3.2-1 shall be supported.

Table 5.9.2.3.2-1: Simple data types

Type name	Description
CongestionValue	Unsigned integer with valid values between 0 and 31. The value 0 indicates that there is no congestion. The value 1 is the lowest congestion level and value 31 is the highest congestion level.

5.9.2.3.3 Enumeration: CongestionType

The enumeration CongestionType represents abstracted values for congestion status.

Table 5.9.2.3.3-1: Enumeration CongestionType

Enumeration value	Description	Applicability (NOTE)
HIGH	The congestion status is high.	
MEDIUM	The congestion status is medium.	
LOW	The congestion status is low.	
NOTE: Properties marked with a feature as defined in subclause 5.9.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.		

5.9.3 Resource structure

5.9.3.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-net-stat-report/v1/

"apiRoot" is set as described in subclause 5.2.4. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.9.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
Network Status Reporting Subscriptions	3gpp-net-stat-report/v1/{scsAsId}/subscriptions	GET	Read all network status reporting subscription resources for a given SCS/AS.
		POST	Create a new network status reporting subscription resource.
Individual Network Status Reporting subscription	3gpp-net-stat-report/v1/{scsAsId}/subscriptions/{subscriptionId}	GET	Read a network status reporting subscription resource.
		PUT	Modify an existing continuous network status reporting subscription resource.
		DELETE	Delete an existing continuous network status reporting subscription resource.
Network Status Reporting Notification	{notificationDestination}	POST	Report a detected network status for a subscription from the SCEF to the SCS/AS

5.9.3.2 Resource: Network Status Reporting Subscriptions

5.9.3.2.1 Introduction

This resource allows the SCS/AS to read all active long-term subscriptions related to a network status reporting.

5.9.3.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-net-stat-report/v1/{scsAsId}/subscriptions/**

This resource shall support the resource URI variables defined in table 5.9.3.2.2-1.

Table 5.9.3.2.2-1: Resource URI variables for resource "Network Status Reporting Subscriptions"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.9.3.2.3 Resource methods

5.9.3.2.3.1 GET

The GET method allows to read all active network status reporting subscriptions for a given SCS/AS. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.9.3.2.3.1-1 and table 5.9.3.2.3.1-2.

Table 5.9.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.9.3.2.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	array(NetworkStatusReportingSubscription)	0..N	200 OK	The information about the network status reporting subscriptions related to the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.9.3.2.3.2 PUT

This HTTP method is not supported for the resource.

5.9.3.2.3.3 PATCH

This HTTP method is not supported for the resource.

5.9.3.2.3.4 POST

The POST method creates a new network status reporting subscription resource for a given SCS/AS. The SCS/AS shall initiate the HTTP POST request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.9.3.2.3.4-1 and table 5.9.3.2.3.4-2.

Table 5.9.3.2.3.4-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.9.3.2.3.4-2: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality	Remarks	
	NetworkStatusReporting Subscription	1	Parameters to register a subscription to request notifications about network status information report with the SCEF.	
Response body	Data type	Cardinality	Response codes	Remarks
	NetworkStatusReporting Subscription	1	201 Created	The subscription was created successfully. The URI of the created resource shall be returned in the "Location" HTTP header.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.9.3.2.3.5 DELETE

This HTTP method is not supported for the resource.

5.9.3.3 Resource: Individual Network Status Reporting Subscription

5.9.3.3.1 Introduction

This resource allows the SCS/AS to request for being notified about the network status using a long-term subscription.

5.9.3.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp-net-stat-report/v1/{scsAsId}/subscriptions/{subscriptionId}**

This resource shall support the resource URI variables defined in table 5.9.3.3.2-1.

Table 5.9.3.3.2-1: Resource URI variables for resource "Individual Network Status Reporting Subscription"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
subscriptionId	Identifier of the subscription resource of type string. The subscriptionId corresponds to the stage 2 TLTRI.

5.9.3.3.3 Resource methods

5.9.3.3.3.1 GET

The GET method allows to read an active network status reporting subscription resource. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.9.3.3.3.1-1 and table 5.9.3.3.3.1-2.

Table 5.9.3.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.9.3.3.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	NetworkStatusReporting Subscription	1	200 OK	The subscription information related to the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.9.3.3.3.2 PUT

The PUT method modifies an existing subscription resource to update a subscription. The SCS/AS shall initiate the HTTP PUT request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.9.3.3.3.2-1 and table 5.9.3.3.3.2-2.

Table 5.9.3.3.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.9.3.3.3.2-2: Data structures supported by the PUT request/response by the resource

Request body	Data type	Cardinality	Remarks	
	NetworkStatusReporting Subscription	1	Parameters to register a subscription to request notifications about network status information report with the SCEF.	
Response body	Data type	Cardinality	Response codes	Remarks
	NetworkStatusReporting Subscription	1	200 OK	The subscription was updated successfully.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.9.3.3.3.3 PATCH

This HTTP method is not supported for the resource.

5.9.3.3.3.4 POST

This HTTP method is not supported for the resource.

5.9.3.3.3.5 DELETE

The DELETE method deletes the resource and terminates the related network status reporting subscription. The SCS/AS shall initiate the HTTP DELETE request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.9.3.3.3.5-1 and table 5.9.3.3.3.5-2.

Table 5.9.3.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.9.3.3.3.5-2: Data structures supported by the DELETE request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>none</td><td></td><td>204 No Content</td><td>The subscription was terminated successfully.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.9.3.4 Network Status Reporting Notification

5.9.3.4.1 Introduction

This resource allows the SCEF to send notifications about the detected network status to the SCS/AS.

5.9.3.4.2 Resource definition

Resource URI: **{**notificationDestination**}**

This resource shall support the resource URI variables defined in table 5.9.3.4.2-1.

Table 5.9.3.4.2-1: Resource URI variables for resource "Network Status Reporting Notification"

Name	Definition
notificationDestination	Callback reference provided by the SCS/AS during creation of the network status reporting subscription.

5.9.3.4.3 Resource methods

5.9.3.4.3.1 Notification via POST

The HTTP POST method reports the detected network status for a network status subscription. The SCEF shall initiate the HTTP POST request message and the SCS/AS shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.9.3.4.3.1-1 and table 5.9.3.4.3.1-2.

Table 5.9.3.4.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.9.3.4.3.1-2: Data structures supported by the POST request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>NetworkStatusReporting Notification</td><td>1</td><td colspan="2">The network status reporting notification provided by the SCEF.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>none</td><td></td><td>204 No Content</td><td>The network status reporting notification is received successfully.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCS/AS can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.9.3.4.3.2 Notification via Websocket

If supported by both SCS/AS and SCEF and successfully negotiated, the NetworkStatusReportingNotification may alternatively be delivered through the Websocket mechanism as defined in subclause 5.2.5.4.

5.9.4 Used Features

The table below defines the features applicable to the ReportingNetworkStatus API. Those features are negotiated as described in subclause 5.2.7.

Table 5.9.4-1: Features used by ReportingNetworkStatus API

Feature Number	Feature	Description
1	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 5.2.5.4. This feature requires that the Notification_test_event feature is also supported.
2	Notification_test_event	The testing of notifications connections is supported according to subclause 5.2.5.3.
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.10 CpProvisioning API

5.10.1 Overview

The CpProvisioning API is a RESTful API that allows the SCS/AS to add, change or delete the communication pattern parameter sets of the UE. The CpProvisioning API defines a set of data models, resources and the related procedures for the creation and management of the resources for communication pattern parameter provisioning. The corresponding JSON schema for the representation of the resources and operations defined by the CpProvisioning API is provided in its complete form in Annex A.10.

5.10.2 Data model

5.10.2.1 Resource data types

5.10.2.1.1 Introduction

This clause defines data structures to be used in resource representations.

Table 5.10.2.1.1-1 specifies data types re-used by the CpProvisioning API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the CpProvisioning API.

Table 5.10.2.1.1-1: CpProvisioning API re-used Data Types

Data type	Reference	Comments
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.10.4-1.

5.10.2.1.2 Type: CpInfo

This type represents the resources for communication pattern parameter provisioning. The same structure is used in the subscription request and subscription response.

Table 5.10.2.1.2-1: Definition of type CpInfo

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 2)
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
externalId	ExternalId	0..1	Each element uniquely identifies a user as defined in Clause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 1)	
msisdn	Msisdn	0..1	Each element identifies the MS internal PSTN/ISDN number allocated for a UE. (NOTE 1)	
externalGroupId	ExternalGroupId	0..1	Identifies a user group as defined in Clause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 1)	
cpParameterSet	map(CpParameterSet)	1..N	Identifies a set of CP parameter information that may be part of this CpInfo structure.	
NOTE 1: One of the properties "externalId", "msisdn" or "externalGroupId" shall be included. NOTE 2: Properties marked with a feature as defined in subclause 5.10.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.10.2.2 Referenced structured data types

5.10.2.2.1 Introduction

This clause defines structured data types that are referenced from data structures defined in the previous clauses.

5.10.2.2.2 Type: CpParameterSet

This data type represents an offered communication pattern parameter set sent from the SCC/AS to the SCEF.

Table 5.10.2.2.2-1: Definition of type CpParameterSet

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
setId	string	1	SCS/AS-chosen correlator provided by the SCS/AS in the request to create a resource fo CP parameter set(s).	
self	Link	0..1	Link to the resources. This parameter shall be supplied by the SCEF in HTTP responses that include an object of CpParameterSet type	
validityTime	DateTime	0..1	Identifies when the CP parameter set expires and shall be deleted. If absent, it indicates that there is no expiration time for this CP parameter set.	
periodicCommunicationIndicator	CommunicationIndicator	0..1	Identifies whether UE communicates periodically or on demand.	
communicationDurationTime	DurationSec	0..1	Identifies duration time of periodic communication	
periodicTime	DurationSec	0..1	Identifies interval time of periodic communication	
scheduledCommunicationTime	ScheduledCommunicationTime	0..1	Identifies time zone and day of the week when the UE is available for communication.	
stationaryIndication	StationaryIndication	0..1	Identifies whether the UE is stationary or mobile.	
expectedUmts	array(LocationArea)	0..N	Identifies the UE's expected geographical movement. The attribute is only applicable in 5G. (NOTE 2)	ExpectedUMT_5G
NOTE 1: Properties marked with a feature as defined in subclause 5.10.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features. NOTE 2: The first instance of the attribute represents the start of the location, and the last one represents the stop of the location.				

5.10.2.2.3 Type: ScheduledCommunicationTime

This data type represents an offered scheduled communication time.

Table 5.10.2.2.3-1: Definition of type ScheduledCommunicationTime

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
daysOfWeek	array(DayOfWeek)	0..6	Identifies the day(s) of the week. If absent, it indicates every day of the week.	
timeOfDayStart	TimeOfDay	0..1	Identifies the start time of the day.	
timeOfDayEnd	TimeOfDay	0..1	Identifies the end time of the day.	
NOTE: Properties marked with a feature as defined in subclause 5.10.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.				

5.10.2.3 Referenced simple data types and enumerations

5.10.2.3.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses. In addition, data types and enumerations defined in subclause 5.2.1 can be referenced.

5.10.2.3.2 Simple data types

The simple data types defined in table 5.10.2.3.2-1 shall be supported.

Table 5.10.2.3.2-1: Simple data types

Type name	Description

5.10.2.3.3 Enumeration: CommunicationIndicator

Table 5.10.2.3.3-1: Enumeration CommunicationIndicator

Enumeration value	Description	Applicability (NOTE)
PERIODICALLY	Identifies the UE communicates periodically	
ON_DEMAND	Identifies the UE communicates on demand	
NOTE: Properties marked with a feature as defined in subclause 5.10.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.		

5.10.2.3.4 Enumeration: StationaryIndication

Table 5.10.2.3.4-1: Enumeration StationaryIndication

Enumeration value	Description	Applicability (NOTE)
STATIONARY	Identifies the UE is stationary	
MOBILE	Identifies the UE is mobile	
NOTE: Properties marked with a feature as defined in subclause 5.10.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.		

5.10.3 Resource structure

5.10.3.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-cp-parameter-provisioning/v1/

"apiRoot" is set as described in subclause 5.2.4. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.10.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
CP provisioning Subscription	3gpp-cp-parameter-provisioning/v1/{scsAsId}/subscriptions	GET	Read all active CP parameter provisioning subscription resources for a given SCS/AS
		POST	Create new subscription resources of provisioning CP parameter set(s)
Individual CP Provisioning Subscription	3gpp-cp-parameter-provisioning/v1/{scsAsId}/subscriptions/{subscriptionId}	GET	Read a CP parameter provisioning subscription resource
		PUT	Modify a CP parameter provisioning subscription resource
		DELETE	Delete a CP parameter provisioning subscription resource
Individual CP set Provisioning	3gpp-cp-parameter-provisioning/v1/{scsAsId}/subscriptions/{subscriptionId}/cpSets/{setId} (NOTE)	PUT	Update CP at individual CP set(s) level associated a CP parameter set Id.
		GET	Read CP at individual CP set(s) level associated a CP parameter set Id.
		DELETE	Delete CP at individual CP set(s) level associated a CP parameter set Id.
NOTE: This setId as a resource identifier is not necessarily identical as the CP parameter set Id received from the SCS/AS.			

5.10.3.2 Resource: CP Provisioning Subscriptions

5.10.3.2.1 Introduction

This resource allows the SCS/AS to read all active long-term transactions related to CP parameter provisioning resource management.

5.10.3.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-cp-parameter-provisioning/v1/{scsAsId}/subscriptions/**

This resource shall support the resource URI variables defined in table 5.10.3.2.2-1.

Table 5.10.3.2.2-1: Resource URI variables for resource "CP Provisioning Subscriptions"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.10.3.2.3 Resource methods

5.10.3.2.3.1 GET

The GET method allows to read all active subscriptions for a given SCS/AS. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.10.3.2.3.1-1 and table 5.10.3.2.3.1-2.

Table 5.10.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.10.3.2.3.1-2: Data structures supported by the GET request/response by the resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	none			
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	array(CpInfo)	0..N	200 OK	The subscription information related to the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.10.3.2.3.2 PUT

This HTTP method is not supported for the resource.

5.10.3.2.3.3 PATCH

This HTTP method is not supported for the resource.

5.10.3.2.3.4 POST

The POST method allows to create subscription for a given SCS/AS. The SCS/AS shall initiate the HTTP POST request message and the SCEF shall respond to the message.

This method shall support request and response data structures, and response codes, as specified in the table 5.10.3.2.3.4-1.

Table 5.10.3.2.3.4-1: Data structures supported by the POST request/response by the resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	CpInfo	1	Change information in CP parameter set.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	CpInfo	1	201 Created	The subscription was created successfully. The SCEF shall return the created subscription in the response payload body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.10.3.2.3.5 DELETE

This HTTP method is not supported for the resource.

5.10.3.3 Resource: Individual CP Provisioning Subscription

5.10.3.3.1 Introduction

This resource allows the SCS/AS to manage resources for CP parameter provisioning using a subscription Id.

5.10.3.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp-cp-parameter-provisioning/v1/{scsAsId}/subscriptions/{subscriptionId}**

This resource shall support the resource URI variables defined in table 5.10.3.3.2-1.

Table 5.10.3.3.2-1: Resource URI variables for resource "Individual CP Provisioning Subscription"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
subscriptionId	Identifier of the subscription resource of type string. The subscriptionId corresponds to the stage 2 TLTRI.

5.10.3.3.3 Resource methods

5.10.3.3.3.1 GET

The GET method allows to read the subscription for a given SCS/AS and subscription Id. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.10.3.3.3.1-1 and table 5.10.3.3.3.1-2.

Table 5.10.3.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.10.3.3.3.1-2: Data structures supported by the GET request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>CpInfo</td><td>1</td><td>200 OK</td><td>The subscription information related to the request URI is returned.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.10.3.3.3.2 PUT

The PUT method allows to change (add/update/remove) one ore more CP parameter set(s) resource(s). The SCS/AS shall initiate the HTTP PUT request message and the SCEF shall respond to the message.

This method shall support request and response data structures, and response codes, as specified in the table 5.14.3.3.3.2-1.

Table 5.10.3.3.3.2-1: Data structures supported by the PUT request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>CpInfo</td><td>1</td><td colspan="2">Change information in CP parameter set(s).</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>CpInfo</td><td>1</td><td>200 OK</td><td>The subscription was modified successfully.
The SCEF shall return an updated subscription in the response payload body.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.10.3.3.3.3 PATCH

This HTTP method is not supported for the resource.

5.10.3.3.3.4 POST

This HTTP method is not supported for the resource.

5.10.3.3.3.5 DELETE

The DELETE method allows to remove an active subscription. The SCS/AS shall initiate the HTTP DELETE request message and the SCEF shall respond to the message.

This method shall support request and response data structures, and response codes, as specified in the table 5.10.3.3.3.5-1.

Table 5.10.3.3.3.5-1.: Data structures supported by the DELETE request/response by the resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	none			
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	none		204 No Content	The subscription was deleted successfully. The payload body shall be empty.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.10.3.4 Resource: Individual CP Set Provisioning

5.10.3.4.1 Introduction

This resource allows the SCS/AS to manage resources for CP parameter set provisioning associated with a CP parameter set Id.

5.10.3.4.2 Resource definition

Resource URI: **{apiRoot}/3gpp-cp-parameter-provisioning/v1/{scsAsId}/subscriptions/{subscriptionId}/cpSets/{setId}**

This resource shall support the resource URI variables defined in table 5.10.3.3.2-1.

Table 5.10.3.3.2-1: Resource URI variables for resource "Individual CP Set Provisioning"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
subscriptionId	Identifier of the subscription resource of type string.
setId	Identifier of the CP parameter set

5.10.3.4.3 Resource methods

5.10.3.4.3.1 GET

The GET method allows to read a CP parameter set resource. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.10.3.4.3.1-1 and table 5.10.3.4.3.1-2.

Table 5.10.3.4.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.10.3.4.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	CpParameterSet	1	200 OK	The subscription information related to the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.10.3.4.3.2 PUT

The PUT method allows to update a CP parameter set resource. The SCS/AS shall initiate the HTTP PUT request message and the SCEF shall respond to the message.

This method shall support request and response data structures, and response codes, as specified in the table 5.10.3.4.3.2-1.

Table 5.10.3.4.3.2-1: Data structures supported by the PUT request/response by the resource

Request body	Data type	Cardinality	Remarks	
	CpParameterSet	1	Change information in CP parameter set.	
Response body	Data type	Cardinality	Response codes	Remarks
	CpParameterSet	1	200 OK	The CP parameter set resource was modified successfully. The SCEF shall return an updated CP parameter set resource in the response payload body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.10.3.4.3.3 PATCH

This HTTP method is not supported for the resource.

5.10.3.4.3.4 POST

This HTTP method is not supported for the resource.

5.10.3.4.3.5 DELETE

The DELETE method allows to remove an active subscription. The SCS/AS shall initiate the HTTP DELETE request message and the SCEF shall respond to the message.

This method shall support request and response data structures, and response codes, as specified in the table 5.10.3.4.3.5-1.

Table 5.10.3.4.3.5-1.: Data structures supported by the DELETE request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	none		204 No Content	The subscription was deleted successfully. The payload body shall be empty.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.10.4 Used Features

The table below defines the features applicable to the CpProvisioning API. Those features are negotiated as described in subclause 5.2.7.

Table 5.10.4-1: Features used by CpProvisioning API

Feature Number	Feature	Description
1	ExpectedUMT_5G	The UE's expected geographical movement is supported. This feature shall be supported in 5G.
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.11 PfdManagement API

5.11.1 Overview

The PfdManagement API allows the SCS/AS to manage the PFDs via the SCEF. The PfdManagement API defines a set of data models, resources and the related procedures for the creation and management of the PFD management request. The corresponding JSON schema for the representation of the resources and operations defined by the PfdManagement API is provided in its complete form in Annex A.11.

5.11.2 Data model

5.11.2.1 Resource data types

5.11.2.1.1 Introduction

This clause defines data structures to be used in resource representations.

Table 5.11.2.1.1-1 specifies data types re-used by the PfdManagement API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the PfdManagement API.

Table 5.11.2.1.1-1: PfdManagement API re-used Data Types

Data type	Reference	Comments
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.11.4-1.

5.11.2.1.2 Type: PfdManagement

This type represents a PFD management resource for a PFD management request.

Table 5.11.2.1.2-1: Definition of type PfdManagement

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
pfdDatas	map(PfdData)	0..N	Each element uniquely identifies the PFDs for an external application identifier. Each element is identified in the map via an external application identifier as key. The response shall include successfully provisioned PFD data of application(s).	
pfdReports	map(PfdReport)	0..N	Supplied by the SCEF and contains the external application identifiers for which PFD(s) are not added or modified successfully. The failure reason is also included. Each element provides the related information for one or more external application identifier(s) and is identified in the map via the failure identifier as key.	
NOTE: Properties marked with a feature as defined in subclause 5.11.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.11.2.1.3 Type: PfdData

This type represents a PFD request to add, update or remove PFD(s) for one external application identifier provided by the SCS/AS to the SCEF via T8 interface.

Table 5.11.2.1.3-1: Definition of type PfdData

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
externalAppId	string	1	Each element uniquely external application identifier (NOTE 2)	
self	Link	0..1	Link to the resource. This parameter shall be supplied by the SCEF in HTTP responses that include an object of PfdData type	
pfds	map(Pfd)	1..N	Contains the PFDs of the external application identifier. Each PFD is identified in the map via a key containing the PFD identifier.	
allowedDelay	DurationSecRm	0..1	Indicates that the list of PFDs in this request should be deployed within the time interval indicated by the Allowed Delay	
cachingTime	DurationSecRo	0..1	SCEF supplied property, inclusion of this property means the allowed delayed cannot be satisfied, i.e. it is smaller than the caching time, but the PFD data is still stored.	
NOTE 1: Properties marked with a feature as defined in subclause 5.11.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features. NOTE 2: An externalAppId can only belong to one "individual PFD Management Transaction" resource.				

5.11.2.1.4 Type: Pfd

This data type represents a PFD for an external Application Identifier.

Table 5.11.2.1.4-1: Definition of type Pfd

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
pfdId	string	1	Identifies a PFD of an application identifier.	
flowDescriptions	array(string)	0..N	Represents a 3-tuple with protocol, server ip and server port for UL/DL application traffic. The content of the string has the same encoding as the IPFilterRule AVP value as defined in IETF RFC 6733 [46]. (NOTE 2)	
urls	array(string)	0..N	Indicates a URL or a regular expression which is used to match the significant parts of the URL. (NOTE 2)	
domainNames	array(string)	0..N	Indicates an FQDN or a regular expression as a domain name matching criteria. (NOTE 2)	
NOTE 1: Properties marked with a feature as defined in subclause 5.11.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features. NOTE 2: One of the properties "flowDescriptions", "urls" or "domainNames" shall be included.				

5.11.2.1.5 Type: PfdReport

This type represents a PFD report to indicate the external application identifier(s) which PFD(s) are not added or modified successfully and corresponding failure reason.

Table 5.11.2.1.5-1: Definition of type PfdReport

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
externalAppId	array(string)	1..N	Identifies the external application identifier(s) which PFD(s) are not added or modified successfully	
failureCode	FailureCode	1	Identifies the failure reason	
cachingTime	DurationSec	0..1	It shall be included when the allowed delayed cannot be satisfied, i.e. it is smaller than the caching time configured in fetching PFD.	
NOTE: Properties marked with a feature as defined in subclause 5.11.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.11.2.2 Referenced simple data types and enumerations

5.11.2.2.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses. In addition, data types and enumerations defined in subclause 5.2.1 can be referenced.

5.11.2.2.2 Simple data types

The simple data types defined in table 5.11.2.2.2-1 shall be supported.

Table 5.11.2.2.2-1: Simple data types

Type name	Description

5.11.2.2.3 Enumeration: FailureCode

The enumeration FailureCode represents the failure reason of the PFD management.

Table 5.11.2.2.3-1: Enumeration FailureCode

Enumeration value	Description	Applicability (NOTE)
MALFUNCTION	This value indicates that something functions wrongly in PFD provisioning or the PFD provisioning does not function at all.	
RESOURCE_LIMITATION	This value indicates there is resource limitation for PFD storage.	
SHORT_DELAY	This value indicates that the allowed delay is too short and PFD(s) are not stored.	
APP_ID_DUPLICATED	The received external application identifier(s) are already provisioned.	
OTHER_REASON	Other reason unspecified.	
NOTE: Properties marked with a feature as defined in subclause 5.11.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.		

5.11.3 Resource structure

5.11.3.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-pfd-management/v1/

"apiRoot" is set as described in subclause 5.2.4. "apiName" shall be set to "3gpp-pfd-management" and "apiVersion" shall be set to "v1" for the version defined in the present document. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.11.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
PFD Management Transactions	3gpp-pfd-management/v1/{scsAsId}/transactions/	GET	Read all PFDs for a given SCS/AS
		POST	Create PFDs for a given SCS/AS and one or more external Application Identifier(s)
Individual PFD Management Transaction	3gpp-pfd-management/v1/{scsAsId}/transactions/{transactionId}	GET	Read all PFDs for a given SCS/AS and a transaction for one or more external Application Identifier(s)
		PUT	Update PFDs for a given SCS/AS and a transaction for one or more external Application Identifier(s)
		DELETE	Delete PFDs for a given SCS/AS and a transaction for one or more external Application Identifier(s)
Individual Application PFD Management	3gpp-pfd-management/v1/{scsAsId}/transactions/{transactionId}/applications/{appId} (NOTE)	PUT	Update PFDs at individual application level
		PATCH	Update PFDs at individual application level
		GET	Read PFDs at individual application level
		DELETE	Delete PFDs at individual application level
NOTE: The appId as the resource identifier is not necessarily be identical as the external application identifier received from the SCS/AS.			

5.11.3.2 Resource: PFD Management Transactions

5.11.3.2.1 Introduction

This resource allows an SCS/AS to read all PFDs for a given SCS/AS or create PFDs for a given SCS/AS with one or more external Application Identifier(s).

5.11.3.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-pfd-management/v1/{scsAsId}/transactions/**

This resource shall support the resource URI variables defined in table 5.11.3.2.2-1.

Table 5.11.3.2.2-1: Resource URI variables for resource "PFD Management Transactions"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.11.3.2.3 Resource methods

5.11.3.2.3.1 GET

The GET method allows to read all active PFDs for a given SCS/AS. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support the request and response data structures, and response codes, as specified in the table 5.11.3.2.3.1-1.

Table 5.11.3.2.3.1-1: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	map(PfdManagement)	0..N	200 OK	All transactions including the PFDs for the SCS/AS in the request URI are returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.11.3.2.3.2 PUT

This HTTP method is not supported for the resource.

5.11.3.2.3.3 POST

The POST method creates new PFDs resource(s) for a given SCS/AS with one or more external Application Identifier provided by the SCS/AS. It is initiated by the SCS/AS and answered by the SCEF. The SCS/AS shall provide the external Application Identifier in the message body and upon receipt of the HTTP POST message, the SCEF shall generate the resource "Individual PFD Management Transaction" and also the sub-resource(s) "Individual Application PFD Management", the SCEF shall send these resource URI in the HTTP response to the SCS/AS.

This method shall support the request and response data structures, and response codes, as specified in the table 5.11.3.2.3.3-1.

Table 5.11.3.2.3.3-1: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality	Remarks	
	PfdManagement	1		
Response body	Data type	Cardinality	Response codes	Remarks
	PfdManagement	1	201 Created	The PFDs resource was created successfully. PfdReport may be included to provide detailed failure information for some applications.
	Array(PfdReport)	1..N	500 Internal Server Error	The PFDs for all applications were not created successfully. PfdReport is included with detailed information.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.11.3.2.3.4 PATCH

This HTTP method is not supported for the resource.

5.11.3.2.3.5 DELETE

To remove all PFDs for a given SCS/AS, the SCS/AS shall use the HTTP DELETE method on the "PFD Management Transactions" resource.

The possible response messages from the SCEF, depending on whether the DELETE request is successful or unsuccessful, are shown in Table 5.11.3.2.3.5-1.

Table 5.11.3.2.3.5-1: Data structures supported by the DELETE request/response by the resource

Request body	Data type	Cardinality	Remarks
	none		

Response body	Data type	Cardinality	Response codes	Remarks
	none		204 No Content	All PFDs were removed successfully. The SCEF shall not return a response payload.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.11.3.3 Resource: Individual PFD Management Transaction

5.11.3.3.1 Introduction

This resource allows an SCS/AS to read, or update or delete PFDs for a given SCS/AS and a transaction Id for one or more application identifier(s) at the SCEF.

5.11.3.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp-pfd-management/v1/{scsAsId}/transactions/{transactionId}**

This resource shall support the resource URI variables defined in table 5.11.3.3.2-1.

Table 5.11.3.3.2-1: Resource URI variables for resource "Individual PFD Management Transaction"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
transactionId	Identifier of the transaction of type string. The transactionId corresponds to the stage 2 TLTRI.

5.11.3.3.3 Resource methods

5.11.3.3.3.1 GET

The GET method allows to read all PFDs for a given SCS/AS and a transaction Id generated by the SCEF. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.11.3.3.3.1-1.

Table 5.11.3.3.3.1-1: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	PfdManagement	1	200 OK	The PFDs for the SCS/AS and the transaction Id for one or more application identifier(s) in the request URI are returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.11.3.3.3.2 PUT

The PUT method modifies the PFDs for a given SCS/AS and an existing transaction Id generated by the SCEF. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support the request and response data structures, and response codes, as specified in the table 5.11.3.3.3.2-1.

Table 5.11.3.3.3.2-1: Data structures supported by the PUT request/response by the resource

	Data type	Cardinality	Response codes	Remarks
Request body	PfdManagement	1		update of PFD(s) for an existing transaction Id.
Response body	Data type	Cardinality	Response codes	Remarks
	PfdManagement	1	200 OK	The PFDs were updated successfully. PfdReport may be included to provide detailed failure information for some applications.
	Array(PfdReport)	1..N	500 Internal Server Error	The PFDs for all applications were not updated successfully. PfdReport is included with detailed information.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.11.3.3.3.3 PATCH

This HTTP method is not supported for the resource.

5.11.3.3.3.4 POST

This HTTP method is not supported for the resource.

5.11.3.3.3.5 DELETE

The DELETE method deletes the PFDs for a given SCS/AS and an transaction Id generated by the SCEF. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.11.3.3.3.5-1 and table 5.11.3.3.3.5-2.

Table 5.11.3.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
N/A			

Table 5.11.3.3.3.5-2: Data structures supported by the DELETE request/response by the resource

	Data type	Cardinality	Response codes	Remarks
Request body	none			
Response body	Data type	Cardinality	Response codes	Remarks
	none		204 No Content	The PFDs for an existing transaction Id were removed successfully.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.11.3.4 Resource: Individual Application PFD Management

5.11.3.4.1 Introduction

This resource allows an SCS/AS to read, update or remove the PFDs for a given SCS/AS and an external Application Identifier at the SCEF.

5.11.3.4.2 Resource definition

Resource URI: **{apiRoot}/3gpp-pfd-management/v1/{scsAsId}/transactions/{transactionId}/applications/{appId}**

This resource shall support the resource URI variables defined in table 5.11.3.4.2-1.

Table 5.11.3.3.4-1: Resource URI variables for resource "Individual Application PFD Management"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
transactionId	Identifier of the transaction of type string.
appId	External Application Identifier of type string.

5.11.3.4.3 Resource methods

5.11.3.4.3.1 GET

The GET method allows to read all PFDs at individual application level. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support request and response data structures, and response codes, as specified in the table 5.11.3.4.3.1-1.

Table 5.11.3.4.3.1-1: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	PfdData	1	200 OK	The PFDs at individual application level in the request URI are returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.11.3.4.3.2 PUT

The PUT method modifies the PFDs at individual application level. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support the request and response data structures, and response codes, as specified in the table 5.11.3.4.3.2-1.

Table 5.11.3.4.3.2-1: Data structures supported by the PUT request/response by the resource

Request body	Data type	Cardinality	Remarks	
	PfdData	1	Update of PFD(s) for an existing external application identifier.	
Response body	Data type	Cardinality	Response codes	Remarks
	PfdData	1	200 OK	The PFDs for an existing external application identifier were updated successfully.
	PfdReport	1	403 Forbidden	The PFDs for the application were not updated successfully, applicable for error SHORT_DELAY and SHORT_DELAY_STORED in table 5.11.2.2.3-1.
	PfdReport	1	409 Conflict	The PFDs for the application were not updated successfully, applicable for error APP_ID_DUPLICATED in table 5.11.2.2.3-1.
	PfdReport	1	500 Internal Server Error	The PFDs for the application were not updated successfully, applicable for other errors in table 5.11.2.2.3-1.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.11.3.4.3.3 PATCH

The PATCH method modifies the PFDs at individual application level. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support the request and response data structures, and response codes, as specified in the table 5.11.3.3.3.3-1.

Table 5.11.3.3.3.3-1: Data structures supported by the PATCH request/response by the resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	PfdData	1	Update of PFD(s) for an existing external application identifier.	
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	PfdData	1	200 OK	The PFDs for an existing external application identifier were updated successfully.
	PfdReport	1	403 Forbidden	The PFDs for the application were not updated successfully, applicable for error SHORT_DELAY and SHORT_DELAY_STORED in table 5.11.2.2.3-1.
	PfdReport	1	409 Conflict	The PFDs for the application were not updated successfully, applicable for error APP_ID_DUPLICATED in table 5.11.2.2.3-1.
	PfdReport	1	500 Internal Server Error	The PFDs for the application were not updated successfully, applicable for other errors in table 5.11.2.2.3-1.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.11.3.4.3.4 POST

This HTTP method is not supported for the resource.

5.11.3.4.3.5 DELETE

The DELETE method deletes all the PFDs at individual application level. It is initiated by the SCS/AS and answered by the SCEF.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.11.3.4.3.5-1 and table 5.11.3.4.3.5-2.

Table 5.11.3.4.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
N/A			

Table 5.11.3.4.3.5-2: Data structures supported by the DELETE request/response by the resource

Request body	**Data type**	**Cardinality**	**Remarks**	
	none			
Response body	**Data type**	**Cardinality**	**Response codes**	**Remarks**
	none		204 No Content	The PFDs were removed successfully.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.11.4 Used Features

The table below defines the features applicable to the PfdManagement API. Those features are negotiated as described in subclause 5.2.7.

Table 5.11.4-1: Features used by PfdManagement API

Feature Number	Feature	Description
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.12 ECRControl API

5.12.1 Overview

The ECRControl API is a custom API (RPC interaction) that allows the SCS/AS to query or configure the enhanced converage restriction over 3GPP networks. The ECRControl API defines a set of data models and related custom operation procedures for the enhanced coverage restriction control request. The corresponding JSON schema for the representation of the operations defined by the ECRControl API is provided in its complete form in Annex A.12.

5.12.2 Data model

5.12.2.1 Data types

5.12.2.1.1 Introduction

This clause defines data structures to be used in the request and response.

Table 5.12.2.1.1-1 specifies data types re-used by the ECRControl API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the ECRControl API.

Table 5.12.2.1.1-1: ECRControl API re-used Data Types

Data type	Reference	Comments
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.12.4-1.

5.12.2.1.2 Type: ECRControl

This type represents the Enhanced Coverage Restriction control request. The structure is used only for request.

Table 5.12.2.1.2-1: Definition of type ECRControl

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
supportedFeatures	SupportedFeatures	1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7.	
externalId	ExternalId	0..1	Identifies a user as defined in Clause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 2)	
msisdn	Msisdn	0..1	Identifies the MS internal PSTN/ISDN number allocated for a UE. (NOTE 2)	
restrictedPlmnIds	array(PlmnId)	0..N	Indicates a complete list (and possibly empty) of serving PLMNs where Enhanced Coverage shall be restricted. This attribute shall not be present for the query custom operation. (NOTE 3)	
allowedPlmnIds	array(PlmnId)	0..N	Indicates a complete list (and possibly empty) of serving PLMNs where Enhanced Coverage shall be allowed. This attribute shall not be present for the query custom operation. (NOTE 3)	
NOTE 1: Properties marked with a feature as defined in subclause 5.4.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features. NOTE 2: One of the properties "externalId" or "msisdn" shall be included. NOTE 3: "restrictedPlmnIds" and "allowedPlmnIds" shall be mutually exclusive.				

5.12.2.1.3 Type: ECRData

This data type represents the current visited PLMN (if any) and the current settings of enhanced coverage restriction. The structure is used only for response.

Table 5.12.2.1.3-1: Definition of type ECRData

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
supportedFeatures	SupportedFeatures	1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7.	
visitedPlmnId	PlmnId	0..1	Indicates the current visited PLMN.	
restrictedPlmnIds	array(PlmnId)	0..N	Indicates a complete list (and possibly empty) of serving PLMNs where Enhanced Coverage shall be restricted. (NOTE 2)	
allowedPlmnIds	array(PlmnId)	0..N	Indicates a complete list (and possibly empty) of serving PLMNs where Enhanced Coverage shall be allowed. (NOTE 2)	
NOTE 1: Properties marked with a feature as defined in subclause 5.4.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features. NOTE 2: "restrictedPlmnIds" and "allowedPlmnIds" shall be mutually exclusive.				

5.12.3 Custom Operations without associated resources

5.12.3.1 Overview

Custom operations used for this API are summarized in table 5.12.3.1-1. "apiRoot" is set as described in subclause 5.2.4.

Table 5.12.3.1-1: Custom operations without associated resources

Custom operation URI	Mapped HTTP method	Description
{apiRoot}/3gpp-ecr-control/v1/query	POST	Query the status of enhanced converage restriction for a UE
{apiRoot}/3gpp-ecr-control/v1/configure	POST	Configure the enhanced converage restriction for a UE

5.12.3.2 Operation: query

5.12.3.2.1 Description

This custom operation allows an SCS/AS to query the current status of enhanced converage restriction for a UE via the T8 interface as defined in 3GPP TS 23.682 [2].

5.12.3.2.2 Operation Definition

This operation shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.12.3.2.2-1 and table 5.12.3.2.2-2.

Table 5.12.3.2.2-1: URI query parameters supported by the POST on this operation

Name	Data type	Cardinality	Remarks
none specified			

Table 5.12.3.2.2-2: Data structures supported by the POST request/response on this operation

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>ECRControl</td><td>1</td><td colspan="2">Parameters to query the current status of Enhanced Coverage Restriction.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>ECRData</td><td>1</td><td>200 OK</td><td>The requested information was returned successfully.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.12.3.3 Operation: configure

5.12.3.3.1 Description

This custom operation allows an SCS/AS to configure the current setting of enhanced converage restriction for a UE via the T8 interface as defined in 3GPP TS 23.682 [2].

5.12.3.3.2 Operation Definition

This operation shall support the request data structures specified in table 5.12.3.3.2-1 and the response data structure and response codes specified in table 5.12.3.3.2-2.

Table 5.12.3.3.2-1: URI query parameters supported by the POST on this operation

Name	Data type	Cardinality	Remarks
none specified			

Table 5.12.3.3.2-2: Data structures supported by the POST request/response on this operation

Request body	Data type	Cardinality	Remarks	
	ECRControl	1	Parameters to configure the setting of Enhanced Coverage Restriction.	
Response body	Data type	Cardinality	Response codes	Remarks
	none		204 No Content	The Enhanced Coverage Restriction setting was configured successfully. The response body shall be empty.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.12.4 Used Features

The table below defines the features applicable to the ECRControl API. Those features are negotiated as described in subclause 5.2.7.

Table 5.12.4-1: Features used by ECRControl API

Feature Number	Feature	Description
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.13 NpConfiguration API

5.13.1 Overview

The NpConfiguration API is a RESTful API that allows the SCS/AS to send suggested network parameters to influence certain aspects of UE/network behaviour such as the UE's PSM, extended idle mode DRX, and extended buffering configurations. The NpConfiguration API defines a set of data models, resources and the related procedures for the creation and management of the network parameter configuration. The corresponding JSON schema for the representation of the resources and operations defined by the NpConfiguration API is provided in its complete form in Annex A.13.

5.13.2 Data model

5.13.2.1 Resource data types

5.13.2.1.1 Introduction

This clause defines data structures to be used in resource representations.

Table 5.13.2.1.1-1 specifies data types re-used by the NetworkParameterConfiguration API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the NetworkParameterConfiguration API.

Table 5.13.2.1.1-1: NetworkParameterConfiguration API re-used Data Types

Data type	Reference	Comments
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.13.4-1.

5.13.2.1.2 Type: NpConfiguration

This type represents a configuration of network parameters. The same structure is used in the configuration request and response.

Table 5.13.2.1.2-1: Definition of type NpConfiguration

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 2)
self	Link	0..1	Link to this resource. This parameter shall be supplied by the SCEF in HTTP responses that include an object of NpConfiguration type	
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
externalId	ExternalId	0..1	Identifies a user as defined in Clause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 1)	
msisdn	Msisdn	0..1	Identifies the MS internal PSTN/ISDN number allocated for a UE. (NOTE 1)	
externalGroupId	ExternalGroupId	0..1	Identifies a user group as defined in Clause 4.6.2 of 3GPP TS 23.682 [2]. (NOTE 1)	
maximumLatency	DurationSec	0..1	This parameter may be included to identify the maximum delay acceptable for downlink data transfers.	
maximumResponseTime	DurationSec	0..1	This parameter may be included to identify the length of time for which the UE stays reachable to allow the SCS/AS to reliably deliver the required downlink data.	
suggestedNumberOfDlPackets	integer	0..1	This parameter may be included to identify the number of packets that the serving gateway shall buffer in case that the UE is not reachable.	
groupReportingGuardTime	DurationSec	0..1	Identifies the time for which the SCEF can aggregate the reports detected by the Ues in a group and report them together to the SCS/AS, as specified in subclause 5.6.0 of 3GPP TS 23.682 [2].	
notificationDestination	Link	0..1	A URI indicating the notification destination where T8 notification requests shall be delivered. The attribute shall be provided if the attribute "externalGroupId" is provided.	
requestTestNotification	boolean	0..1	Set to true by the SCS/AS to request the SCEF to send a test notification as defined in subclause 5.2.5.3. Set to false or omitted otherwise. The attribute may only be provided if the attribute "externalGroupId" is provided.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 5.2.5.4. The attribute may only be provided if the attribute "externalGroupId" is provided.	Notification_websocket
NOTE 1: Only one of the properties "externalId", "msisdn" or "externalGroupId" shall be included. NOTE 2: Properties marked with a feature as defined in subclause 5.13.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.13.2.1.3 Type: NpConfigurationPatch

This type represents a configuration of network parameters provided by the SCS/AS to the SCEF. The structure is used for HTTP PATCH request.

Table 5.13.2.1.3-1: Definition of type NpConfigurationPatch

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
maximumLatency	DurationSecRm	0..1	This parameter may be included to identify the maximum delay acceptable for downlink data transfers.	
maximumResponseTime	DurationSecRm	0..1	This parameter may be included to identify the length of time for which the UE stays reachable to allow the SCS/AS to reliably deliver the required downlink data.	
suggestedNumberOfDlPackets	integer	0..1	This parameter may be included to identify the number of packets that the serving gateway shall buffer in case that the UE is not reachable.	
groupReportGuardTime	DurationSecRm	0..1	Identifies the time for which the SCEF can aggregate the reports detected by the Ues in a group and report them together to the SCS/AS, as specified in subclause 5.6.0 of 3GPP TS 23.682 [2].	
NOTE: Properties marked with a feature as defined in subclause 5.13.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.13.2.1.4 Type: ConfigurationNotification

This type represents a configuration result notification.

Table 5.13.2.1.4-1: Definition of the ConfigurationNotification data type

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
configuration	Link	1	Link to the configuration resource to which this notification is related.	
configResults	array(ConfigResult)	0..N	Each element identifies a notification of grouping configuration result.	
NOTE: Properties marked with a feature as defined in subclause 5.13.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.				

5.13.3 Resource structure

5.13.3.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-network-parameter-configuration/v1/

"apiRoot" is set as described in subclause 5.2.4. "apiName" shall be set to "3gpp-network-parameter-configuration" and "apiVersion" shall be set to "v1" for the version defined in the present document. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.13.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
NP Configurations	3gpp-network-parameter-configuration /v1/{scsAsId}/configurations/	GET	Read all NP configurations for a given SCS/AS
		POST	Create a new np configuration
Individual NP Configuration	3gpp-network-parameter-configuration /v1/{scsAsId}/configurations/{configurationId}	PUT	Replace all of the properties in an existing np configuration
		PATCH	Modify some properties in an existing np configuration
		GET	Read an existing NP configuration
		DELETE	Delete a NP configuration
Configuration Notification	{notificationDestination}	POST	Report a grouping configuration result from the SCEF to the SCS/AS

5.13.3.2 Resource: NP Configurations

5.13.3.2.1 Introduction

This resource allows an SCS/AS to read all active network parameter configurations or create a new configuration to configure network parameters.

5.13.3.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-network-parameter-configuration /v1/{scsAsId}/configurations/**

This resource shall support the resource URI variables defined in table 5.13.3.2.2-1.

Table 5.13.3.2.2-1: Resource URI variables for resource "NP Configurations"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.13.3.2.3 Resource methods

5.13.3.2.3.1 GET

The GET method allows to read all active configurations indicated by the resource URI as defined in subclause 5.13.3.2.2. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.13.3.2.3.1-1 and table 5.13.3.2.3.1-2.

Table 5.13.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.13.3.2.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality		Remarks
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	array(NpConfiguration)	0..N	200 OK	The NP configuration information related to the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.13.3.2.3.2 PUT

This HTTP method is not supported for the resource.

5.13.3.2.3.3 PATCH

This HTTP method is not supported for the resource.

5.13.3.2.3.4 POST

The POST method creates a new configuration resource for a given SCS/AS. The SCS/AS shall initiate the HTTP POST request message and the SCEF shall respond to the message. The SCEF shall construct the URI of the created resource.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.13.3.2.3.4-1 and table 5.13.3.2.3.4-2.

Table 5.13.3.2.3.4-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.13.3.2.3.4-2: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality		Remarks
	NpConfiguration	1		Parameters to create a new configuration to configure network parameters with the SCEF.
Response body	Data type	Cardinality	Response codes	Remarks
	NpConfiguration	1	201 Created	The configuration was created successfully. The URI of the created resource shall be returned in the "Location" HTTP header.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.13.3.2.3.5 DELETE

This HTTP method is not supported for the resource.

5.13.3.3 Resource: Individual NP Configuration

5.13.3.3.1 Introduction

This resource allows an SCS/AS to query, update and delete a network parameter configuration indicated by the resource URI as defined in subclause 5.13.3.3.2.

5.13.3.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp-network-parameter-configuration/v1/{scsAsId}/configurations/{configurationId}**

This resource shall support the resource URI variables defined in table 5.13.3.3.2-1.

Table 5.13.3.2.2-1: Resource URI variables for resource "Individual NP Configuration"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
configurationId	Identifier of the configuration resource of type string.

5.13.3.3.3 Resource methods

5.13.3.3.3.1 GET

The GET method allows to read an active configuration indicated by the resource URI as defined in subclause 5.13.3.3.2. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.13.3.3.3.1-1 and table 5.13.3.3.3.1-2.

Table 5.13.3.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks

Table 5.13.3.3.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	NpConfiguration	1	200 OK	The configuration information related to the request URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.13.3.3.3.2 PUT

The PUT method modifies an existing configuration indicated by the resource URI as defined in subclause 5.13.3.3.2. The SCS/AS shall initiate the HTTP PUT request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.13.3.3.3.2-1 and table 5.13.3.3.3.2-2.

Table 5.13.3.3.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.13.3.3.3.2-2: Data structures supported by the PUT request/response by the resource

Request body	Data type	Cardinality	Remarks	
	NpConfiguration	1	Update of network parameter(s) for an existing Configuration.	
Response body	Data type	Cardinality	Response codes	Remarks
	NpConfiguration	1	200 OK	The resource was updated successfully. The SCEF shall return an updated Configuration information in the response.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.13.3.3.3.3 PATCH

The PATCH method shall be used to update some properties in an existing configuration indicated by the Resource URI as defined in subclause 5.13.3.3.2. The SCS/AS shall initiate the HTTP PATCH request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.13.3.3.3.3-1 and table 5.13.3.3.3.3-2.

Table 5.13.3.3.3.3-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.13.3.3.3.3-2: Data structures supported by the PUT request/response by the resource

Request body	Data type	Cardinality	Remarks	
	NpConfigurationPatch	1	Partial update an existing network parameter configuration.	
Response body	Data type	Cardinality	Response codes	Remarks
	NpConfiguration	1	200 OK	The configuration was updated successfully. The SCEF shall return an updated configuration information in the response.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.13.3.3.3.4 POST

This HTTP method is not supported for the resource.

5.13.3.3.3.5 DELETE

The DELETE method deletes an existing configuration resource "Individual NP Configuration". The SCS/AS shall initiate the HTTP DELETE request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.13.3.3.3.5-1 and table 5.13.3.3.3.5-2.

Table 5.13.3.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
N/A			

Table 5.13.3.3.3.5-2: Data structures supported by the DELETE request/response on the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="3">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>none</td><td></td><td>204 No Content</td><td>The configuration was terminated successfully.
The response body shall be empty.</td></tr>
<tr><td>Array(ConfigResult)</td><td>1..N</td><td>200 OK</td><td>The configuration was terminated successfully, the configuration failure information for group members shall be included if received.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.13.3.4 Configuration Notification

5.13.3.4.1 Introduction

This resource allows the SCEF to send notifications about grouping configuration result to the SCS/AS.

5.13.3.4.2 Resource definition

Resource URI: **{**notificationDestination**}**

This resource shall support the resource URI variables defined in table 5.13.3.4.2-1.

Table 5.13.3.4.2-1: Resource URI variables for resource "Configuration Notification"

Name	Definition
notificationDestination	Callback reference provided by the SCS/AS during creation of the NP configuration subscription.

5.13.3.4.3 Resource methods

5.13.3.4.3.1 Notification via POST

The HTTP POST method reports the grouping configuration results for a NP configuration subscription. The SCEF shall initiate the HTTP POST request message and the SCS/AS shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.13.3.4.3.1-1 and table 5.13.3.4.3.1-2.

Table 5.13.3.4.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.13.3.4.3.1-2: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality	Remarks	
	ConfiguationNotification	1	The grouping configuration result notification provided by the SCEF.	
Response body	Data type	Cardinality	Response codes	Remarks
	none		204 No Content	The notification is received successfully.
NOTE: In addition to the above response codes, the SCS/AS can also send the HTTP response codes in table 5.2.6-1.				

5.13.3.4.3.2 Notification via Websocket

If supported by both SCS/AS and SCEF and successfully negotiated, the ConfigurationNotification may alternatively be delivered through the Websocket mechanism as defined in subclause 5.2.5.4.

5.13.4 Used Features

The table below defines the features applicable to the NpConfiguration API. Those features are negotiated as described in subclause 5.2.7.

Table 5.13.4-1: Features used by NpConfiguration API

Feature Number	Feature	Description
1	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 5.2.5.4. This feature requires that the Notification_test_event featute is also supported.
2	Notification_test_event	The testing of notification connection is supported according to subclause 5.2.5.3.
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.14 AsSessionWithQoS API

5.14.1 Overview

The AsSessionWithQoS API is a RESTful API that allows the SCS/AS to set up a session with SCEF with required QoS based on the application and service requirement. The AsSessionWithQoS API defines a set of data models, resources and the related procedures for the creation and management of the AS sessions with required QoS. The corresponding JSON schema for the representation of the resources and operations defined by the AsSessionWithQoS API is provided in its complete form in Annex A.14.

5.14.2 Data model

5.14.2.1 Resource data types

5.14.2.1.1 Introduction

This clause defines data structures to be used in resource representations, including subscription resources.

Table 5.14.2.1.1-1 specifies data types re-used by the AsSessionWithQoS API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the AsSessionWithQoS API.

Table 5.14.2.1.1-1: AsSessionWithQoS API re-used Data Types

Data type	Reference	Comments
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.14.4-1.

5.14.2.1.2 Type: AsSessionWithQoSSubscription

This type represents an AS session request with specific QoS for the service provided by the SCS/AS to the SCEF via T8 interface. The structure is used for subscription request and response.

Table 5.14.2.1.2-1: Definition of type AsSessionWithQoSSubscription

Attribute name	Data type	Cardinality	Description	Applicability (NOTE 1)
supportedFeatures	SupportedFeatures	0..1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7. This attribute shall be provided in the POST request and in the response of successful resource creation.	
notificationDestination	Link	1	Contains the URL to receive the notification bearer level event(s) from the SCEF.	
flowInfo	array(FlowInfo)	1..N	Describe the data flow which requires QoS.	
qosReference	string	0..1	Identifies a pre-defined QoS information	
ueIpv4Addr	Ipv4Addr	0..1	The Ipv4 address of the UE. (NOTE 2)	
ueIpv6Addr	Ipv6Addr	0..1	The Ipv6 address of the UE. (NOTE 2)	
usageThreshold	UsageThreshold	0..1	Time period and/or traffic volume in which the QoS is to be applied.	
sponsorInfo	SponsorInformation	0..1	Indicates a sponsor information	
requestTestNotification	boolean	0..1	Set to true by the SCS/AS to request the SCEF to send a test notification as defined in subclause 5.2.5.3. Set to false or omitted otherwise.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 5.2.5.4.	Notification_websocket
NOTE 1: Properties marked with a feature as defined in subclause 5.14.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features. NOTE 2: The property "ueIpv4Addr" or "ueIpv6Addr" shall be included.				

5.14.2.1.3 Type: AsSessionWithQoSSubscriptionPatch

This type represents an AS session request with specific QoS for the service provided by the SCS/AS to the SCEF via T8 interface. The structure is used for PATCH request.

Table 5.14.2.1.3-1: Definition of type AsSessionWithQoSSubscriptionPatch

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
flowInfo	array(FlowInfo)	0..N	Describe the data flow which requires QoS.	
qosReference	string	0..1	Pre-defined QoS reference	
usageThreshold	UsageThresholdRm	0..1	Time period and/or traffic volume in which the QoS is to be applied.	
NOTE: Properties marked with a feature as defined in subclause 5.14.4 are applicable as described in subclause 5.2.7. If no feature are indicated, the related property applies for all the features.				

5.14.3 Resource structure

5.14.3.1 General

All resource URIs of this API should have the following root:

{apiRoot}/3gpp-as-session-with-qos/v1/

"apiRoot" is set as described in subclause 5.2.4. "apiName" shall be set to "3gpp-as-session-with-qos" and "apiVersion" shall be set to "v1" for the version defined in the present document. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.14.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
AS Session with Required QoS Subscriptions	3gpp-as-session-with-qos/v1/{scsAsId}/subscriptions/	GET	Get all subscription resources for a given SCS/AS.
		POST	Create a new AS session.
Individual AS Session with Required QoS Subscription	3gpp-as-session-with-qos/v1/{scsAsId}/subscriptions/{subscriptionId}	GET	Read a subscription resource for a given SCS/AS and a subscription Id.
		PUT	Modify a subscription resource for a given SCS/AS and a subscription Id.
		PATCH	Modify a subscription resource for a given SCS/AS and a subscription Id.
		DELETE	Delete a subscription resource for a given SCS/AS and a subscription Id.
Event Notification	{notificationUri}	POST	Notify the bearer level event(s) from the SCEF to the SCS/AS

5.14.3.2 Resource: AS Session with Required QoS subscriptions

5.14.3.2.1 Introduction

This resource allows the SCS/AS to read all active AS session with required QoS subscription resources, or create a new subscription resource for the SCS/AS.

5.14.3.2.2 Resource definition

Resource URI: **{apiRoot}/3gpp-as-session-with-qos /v1/{scsAsId}/subscriptions/**

This resource shall support the resource URI variables defined in table 5.14.3.2.2-1.

Table 5.14.3.2.2-1: Resource URI variables for resource "AS Session with Required QoS Subscriptions"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.

5.14.3.2.3 Resource methods

5.14.3.2.3.1 GET

The GET method allows to read all active subscriptions for a given SCS/AS. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.14.3.2.3.1-1 and table 5.14.3.2.3.1-2.

Table 5.14.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks

Table 5.14.3.2.3.1-2: Data structures supported by the GET request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>none</td><td></td><td colspan="2"></td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>array(AsSessionWithQoSSubscription)</td><td>0..N</td><td>200 OK</td><td>The subscription information related to the request URI is returned.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.14.3.2.3.2 PUT

This HTTP method is not supported for the resource.

5.14.3.2.3.3 PATCH

This HTTP method is not supported for the resource.

5.14.3.2.3.4 POST

The POST method creates a new subscription resource for a given SCS/AS. The SCS/AS shall initiate the HTTP POST request message and the SCEF shall respond to the message. The SCEF shall construct the URI of the created resource using that URI.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.14.3.2.3.4-1 and table 5.14.3.2.3.4-2.

Table 5.14.3.2.3.4-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks

Table 5.14.3.2.3.4-2: Data structures supported by the POST request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>AsSessionWithQoSSubscription</td><td>1</td><td colspan="2">Parameters to create a subscription for an AS session with required QoS for the service requirement.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>AsSessionWithQoSSubscription</td><td>1</td><td>201 Created</td><td>The subscription was created successfully.
The URI of the created resource shall be returned in the "Location" HTTP header.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.14.3.2.3.5 DELETE

This HTTP method is not supported for the resource.

5.14.3.3 Resource: Individual AS Session with Required QoS Subscription

5.14.3.3.1 Introduction

This resource allows an SCS/AS to query, update and delete an AS session with required QoS subscription.

5.14.3.3.2 Resource definition

Resource URI: **{apiRoot}/3gpp-as-session-with-qos /v1/{scsAsId}/subscriptions/{subscriptionId}**

This resource shall support the resource URI variables defined in table 5.14.3.3.2-1.

Table 5.14.3.2.2-1: Resource URI variables for resource "Individual AS Session with Required QoS Subscription"

Name	Definition
apiRoot	See clause 5.2.4.
scsAsId	Identifier of the SCS/AS of type ScsAsId.
subscriptionId	Identifier of the subscription resource of type string. The subscriptionId corresponds to the stage 2 TLTRI.

5.14.3.3.3 Resource methods

5.14.3.3.3.1 GET

The GET method allows to read a subscription resource. The SCS/AS shall initiate the HTTP GET request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.14.3.3.3.1-1 and table 5.14.3.3.3.1-2.

Table 5.14.3.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks

Table 5.14.3.3.3.1-2: Data structures supported by the GET request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	AsSessionWithQoSSubscription	1	200 OK	The subscription information related to the resource URI is returned.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.14.3.3.3.2 PUT

The PUT method allows changing the service information of an active subscription. The properties "ueIpv4Addr" or "ueIpv6Addr" shall remain unchanged from previously provided value.

This method shall support request and response data structures, and response codes, as specified in the table 5.14.3.3.3.2-1.

Table 5.14.3.3.3.2-1: Data structures supported by the PUT request/response by the resource

Request body	Data type	Cardinality	Remarks	
	AsSessionWithQoSSubscription	1	Set up AS session with required QoS.	
Response body	Data type	Cardinality	Response codes	Remarks
	AsSessionWithQoSSubscription	1	200 OK	The subscription was modified successfully. The SCEF shall return an updated subscription in the response payload body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.14.3.3.3.3 PATCH

The PATCH method allows to change the service information of an active subscription.

This method shall support request and response data structures, and response codes, as specified in the table 5.14.3.3.3.3-1.

Table 5.14.3.3.3.3-1: Data structures supported by the PATCH request/response by the resource

Request body	Data type	Cardinality	Remarks	
	AsSessionWithQoSSubscriptionPatch	1	Partial update of an AS session with required QoS.	
Response body	Data type	Cardinality	Response codes	Remarks
	AsSessionWithQoSSubscription	1	200 OK	The subscription was modified successfully. The SCEF shall return an updated subscription in the response payload body.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.14.3.3.3.4 POST

This HTTP method is not supported for the resource.

5.14.3.3.3.5 DELETE

The DELETE method deletes the AsSessionWithQoSSubscription resource and terminates the related subscription. The SCS/AS shall initiate the HTTP DELETE request message and the SCEF shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.14.3.3.3.5-1 and table 5.14.3.3.3.5-2.

Table 5.14.3.3.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks

Table 5.14.3.3.3.5-2: Data structures supported by the DELETE request/response by the resource

Request body	Data type	Cardinality	Remarks	
	none			
Response body	Data type	Cardinality	Response codes	Remarks
	none		204 No Content	The subscription was terminated successfully. The response body shall be empty.
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.14.3.4 Event Notification

5.14.3.4.1 Introduction

This resource allows the SCEF to notify the SCS/AS of the bearer level event(s).

5.14.3.4.2 Resource definition

Resource URI: **{notificationUrl}**

This resource shall support the resource URI variables defined in table 5.14.3.4.2-1.

Table 5.14.3.4.2-1: Resource URI variables for resource "Event Notification"

Name	Definition
notificationUri	Reference provided by the SCS/AS when the SCS/AS requests to sponsor the traffic from the beginning or to become the chargeable party at a later point.

5.14.3.4.3 Resource methods

5.14.3.4.3.1 Notification via POST

The POST method allows to notify SCS/AS of the bearer level event(s) by the SCEF and the SCS/AS shall respond to the message.

This method shall support request and response data structures, and response codes, as specified in the table 5.14.3.4.3.1-1.

Table 5.14.3.4.3.1-1.: Data structures supported by the POST request/response by the resource

Request body	Data type	Cardinality	Remarks	
	NotificationData	1	Representation of the bearer level notification.	
Response body	Data type	Cardinality	Response codes	Remarks
	none	1	200 OK	This case represents a successful notification of bearer level event(s).
NOTE: In addition to the above response codes, the SCEF can also send the HTTP response codes in table 5.2.6-1.				

5.14.3.4.3.2 Notification via Websocket

If supported by both SCS/AS and SCEF and successfully negotiated, the NotificationData may alternatively be delivered through the Websocket mechanism as defined in subclause 5.2.5.4.

5.14.4 Used Features

The table below defines the features applicable to the AsSessionWithQoS API. Those features are negotiated as described in subclause 5.2.7.

Table 5.14.4-1: Features used by AsSessionWithQoS API

Feature Number	Feature	Description
1	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 5.2.5.4. This feature requires that the Notification_test_event featute is also supported.
2	Notification_test_event	The testing of notifications connections is supported according to subclause 5.2.5.3.
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

5.15 MsisdnLessMoSms API

5.15.1 Overview

The MsisdnLessMoSms API allows the delivery of MSISDN-less mobile originated SMSs from the SCEF to the SCS/AS. The corresponding JSON schema for the representation of the resources and operations defined by the MsisdnLessMoSms API is provided in its complete form in Annex A.15.

5.15.2 Data model

5.15.2.1 Notification data types

5.15.2.1.1 Introduction

This clause defines data structures to be used in notifications.

Table 5.15.2.1.1-1 specifies data types re-used by the MsisdnLessMoSms API from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the MsisdnLessMoSms API.

Table 5.15.2.1.1-1: MsisdnLessMoSms API re-used Data Types

Data type	Reference	Comments
SupportedFeatures	3GPP TS 29.571 [45]	Used to negotiate the applicability of the optional features defined in table 5.15.4-1.

5.15.2.1.2 Type: MsisdnLessMoSmsNotification

This data type represents a MSISDN-less MO SMS, which is sent from the SCEF to the SCS/AS.

Table 5.15.2.1.2-1: Definition of type MsisdnLessMoSmsNotification

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
supportedFeatures	SupportedFeatures	1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7.	
sms	Bytes	1	The MSISDN-less MO SMS containing a short message transfer protocol data unit (TPDU) which is defined in 3GPP TS 23.040 [43] and represents the user data field carried by the short message service relay sub-layer protocol modiefied.	
externalId	string	1	External identifier has the form username@realm.	
ApplicationPort	integer	1	Unsigned integer used to uniquely identify the triggering application addressed in the device, see subclause 9.2.3.24.4 in TS 23.040 [43] for further details.	
NOTE: Properties marked with a feature as defined in subclause 5.15.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.				

5.15.2.1.3 Type: MsisdnLessMoSmsNotificationReply

This data type represents a reply to an MSISDN-less MO SMS notification and is sent from the SCS/AS to the SCEF.

Table 5.15.2.1.3-1: Definition of type MsisdnLessMoSmsNotificationReply

Attribute name	Data type	Cardinality	Description	Applicability (NOTE)
supportedFeatures	SupportedFeatures	1	Used to negotiate the supported optional features of the API as described in subclause 5.2.7.	
NOTE: Properties marked with a feature as defined in subclause 5.3.4 are applicable as described in subclause 5.2.7. If no features are indicated, the related property applies for all the features.				

5.15.3 Resource structure

5.15.3.1 General

All resource URIs of this API should have the following root:

{notificationDestination}

"{notificationDestination}" is determined based on preconfigured information in the SCEF as described in subclause 4.4.14.2. All resource URIs in the subclauses below are defined relative to the above root URI.

The following resources and HTTP methods are supported for this API:

Table 5.15.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
MSISDN-less MO SMS Notification	{notificationDestination}	POST	Deliver a received MSIDN-less MO SMS from the SCEF to the SCS/AS

5.15.3.2 MSISDN-less MO SMS Notification

5.15.3.2.1 Introduction

This resource allows the SCEF to deliver a received MSIDN-less MO SMS to the SCS/AS.

5.15.3.2.2 Resource definition

Resource URI: **{**notificationDestination**}**

This resource shall support the resource URI variables defined in table 5.15.3.2.2-1.

Table 5.15.3.2.2-1: Resource URI variables for resource "MSISDN-less MO SMS Notification"

Name	Definition
notificationDestination	A URI indicating the notification destination where T8 notification requests shall be delivered to. This URI shall be preconfigured in the SCEF.

5.15.3.2.3 Resource methods

5.15.3.2.3.1 Notification via POST

The HTTP POST method delivers a received MSISDN-less MO SMS. The SCEF shall initiate the HTTP POST request message and the SCS/AS shall respond to the message.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the table 5.15.3.2.3.1-1 and table 5.15.3.2.3.1-2.

Table 5.15.3.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	Cardinality	Remarks
none specified			

Table 5.15.3.3.3.1-2: Data structures supported by the POST request/response by the resource

<table>
<tr><th rowspan="2">Request body</th><th>Data type</th><th>Cardinality</th><th colspan="2">Remarks</th></tr>
<tr><td>MsisdnLessMoSmsNotification</td><td>1</td><td colspan="2">The MSISDN-less MO SMS.</td></tr>
<tr><th rowspan="2">Response body</th><th>Data type</th><th>Cardinality</th><th>Response codes</th><th>Remarks</th></tr>
<tr><td>MsisdnLessMoSmsNotificationReply</td><td></td><td>200 OK</td><td>The MSISDN-less MO SMS is received successfully.</td></tr>
<tr><td colspan="5">NOTE: In addition to the above response codes, the SCS/AS can also send the HTTP response codes in table 5.2.6-1.</td></tr>
</table>

5.15.4 Used Features

The table below defines the features applicable to the MsisdnLessMoSms API. Those features are negotiated as described in subclause 5.2.7.

Table 5.15.4-1: Features used by MsisdnLessMoSms API

Feature Number	Feature	Description
Feature: A short name that can be used to refer to the bit and to the feature, e.g. "Notification". Description: A clear textual description of the feature.		

6 Security

TLS (IETF RFC 5246 [4]) shall be used to support the security communication between the SCEF and the SCS/AS over T8 as defined in subclause 5.5 of 3GPP TS 33.187 [35].

7 Using Common API Framework

When CAPIF is used with SCEF, SCEF shall support the following as defined in 3GPP TS 29.222 [48]:

- the API exposing function and related APIs over CAPIF-2/2e and CAPIF-3 reference points;
- the API publishing function and related APIs over CAPIF-4 reference point;
- the API management function and related APIs over CAPIF-5 reference point; and
- at least one of the the security methods for authentication and authorization, and related security mechanisms.

In a centralized deployment as defined in 3GPP TS 23.222 [47], where the CAPIF core function and API provider domain functions are co-located, the interactions between the CAPIF core function and API provider domain functions may be independent of CAPIF-3, CAPIF-4 and CAPIF-5 reference points.

Annex A (normative): OpenAPI representation for the APIs defined in the present document

A.1 General

This Annex is based on the OpenAPI 3.0.0 specification [27] and provides corresponding representations of all APIs defined in the present specification.

NOTE: An OpenAPIs representation embeds JSON Schema representations of HTTP message bodies.

A.2 Data Types applicable to several APIs

```
openapi: 3.0.0
info:
  title: TS 29.122 Common Data Types
  version: "1.PreR15.1.0"
paths: {}
components:
  schemas:
    SponsorInformation:
      type: object
      properties:
        sponsorId:
          type: string
          description: It indicates Sponsor ID.
        aspId:
          type: string
          description: It indicates Application Service Provider ID.
      required:
        - sponsorId
        - aspId
    UsageThreshold:
      type: object
      properties:
        duration:
          $ref: '#/components/schemas/DurationSec'
        totalVolume:
          $ref: '#/components/schemas/Volume'
        downlinkVolume:
          $ref: '#/components/schemas/Volume'
        uplinkVolume:
          $ref: '#/components/schemas/Volume'
    UsageThresholdRm:
      type: object
      properties:
        duration:
          $ref: '#/components/schemas/DurationSecRm'
        totalVolume:
          $ref: '#/components/schemas/VolumeRm'
        downlinkVolume:
          $ref: '#/components/schemas/VolumeRm'
        uplinkVolume:
          $ref: '#/components/schemas/VolumeRm'
      nullable: true
    TimeWindow:
      type: object
      properties:
        startTime:
          $ref: '#/components/schemas/DateTime'
        stopTime:
          $ref: '#/components/schemas/DateTime'
      required:
        - startTime
        - stopTime
    Acknowledgement:
      type: object
      properties:
        details:
          type: string
          description: A human-readable explanation specific to this successful acknowledgement
      required:
        - details
    NotificationData:
      type: object
      properties:
        transaction:
          $ref: '#/components/schemas/Link'
        eventReports:
          type: array
          items:
            $ref: '#/components/schemas/EventReport'
          minItems: 1
          description: Contains the reported event and applicable information
      required:
        - transaction
        - eventReports
    EventReport:
      type: object
      properties:
        event:
          $ref: '#/components/schemas/Event'
        accumulatedUsage:
          $ref: '#/components/schemas/AccumulatedUsage'
        flowIds:
          type: array
          items:
            type: integer
          minItems: 0
          description: Identifies the IP flows that were sent during event subscription
      required:
        - event
    AccumulatedUsage:
      type: object
      properties:
        duration:
          $ref: '#/components/schemas/DurationSec'
        totalVolume:
          $ref: '#/components/schemas/Volume'
        downlinkVolume:
          $ref: '#/components/schemas/Volume'
        uplinkVolume:
          $ref: '#/components/schemas/Volume'
    FlowInfo:
      type: object
      properties:
        flowId:
          type: integer
          description: Indicates the IP flow.
        flowDescriptions:
          type: array
          items:
            type: string
          description: Indicates the packet filters of the IP flow. Refer to subclause 5.3.8 of
3GPP TS 29.214 [10] for encoding. It shall contain UL and/or DL IP flow description.
          minItems: 0
          maxItems: 2
      required:
        - flowId
    TestNotification:
      type: object
      properties:
        subscription:
          $ref: '#/components/schemas/Link'
      required:
        - subscription
    WebsockNotifConfig:
      type: object
      properties:
        websocketUri:
          $ref: '#/components/schemas/Link'
        requestWebsocketUri:
          type: boolean
          description: Set by the SCS/AS to indicate that the Websocket delivery is requested.
    LocationArea:
      type: object
      properties:
        cellId:
          type: array
          items:
            type: string
          minItems: 0
          description: Indicates a Cell Global Identification of the user which identifies the cell
the UE is registered.
        enodeBId:
          type: array
          items:
            type: string
          minItems: 0
          description: Indicates an eNodeB in which the UE is currently located.
        routingAreaId:
          type: array
          items:
            type: string
          minItems: 0
          description: Identifies an Routing Area Identity of the user where the UE is located.
        trackingAreaId:
          type: array
          items:
            type: string
          minItems: 0
          description: Identifies a Tracking Area Identity of the user where the UE is located.
        geographicArea:
          type: array
          items:
            $ref: 'TS29572_Nlmf_Location.yaml#/components/schemas/GeographicArea'
          minItems: 0
          description: Identifies a geographic area of the user where the UE is located.
        civicAddress:
          type: array
          items:
            $ref: 'TS29572_Nlmf_Location.yaml#/components/schemas/CivicAddress'
          minItems: 0
          description: Identifies a civic address of the user where the UE is located.
    ProblemDetails:
      type: object
      properties:
        type:
          $ref: '#/components/schemas/Uri'
        title:
          type: string
          description: A short, human-readable summary of the problem type. It should not change
from occurrence to occurrence of the problem.
        Status:
          type: integer
          description: The HTTP status code for this occurrence of the problem.
        Detail:
          type: string
          description: A human-readable explanation specific to this occurrence of the problem.
        Instance:
          $ref: '#/components/schemas/Uri'
        cause:
          type: string
          description: A machine-readable application error cause specific to this occurrence of the
problem. This IE should be present and provide application-related error information, if available.
        invalidParams:
          type: array
          items:
            $ref: '#/components/schemas/InvalidParam'
          minItems: 0
          description: Description of invalid parameters, for a request rejected due to invalid
parameters.
    InvalidParam:
      type: object
      properties:
        param:
          type: string
          description: Attribute's name encoded as a JSON Pointer, or header's name.
        reason:
          type: string
          description: A human-readable reason, e.g. "must be a positive integer".
      required:
        - param
    PlmnId:
      type: object
      properties:
        mcc:
          $ref: '#/components/schemas/Mcc'
        mnc:
          $ref: '#/components/schemas/Mnc'
      required:
        - mcc
        - mnc
    ConfigResult:
      type: object
      properties:
        externalIds:
          type: array
          items:
            $ref: '#/components/schemas/ExternalId'
          minItems: 0
          description: Each element indicates an external identifier of the UE.
        Msisdns:
          type: array
          items:
            $ref: '#/components/schemas/Msisdn'
          minItems: 0
          description: Each element identifies the MS internal PSTN/ISDN number allocated for the
UE.
        resultReason:
          $ref: '#/components/schemas/ResultReason'
      required:
        - resultReason
    Bandwidth:
      type: integer
      description: integer indicating a bandwidth in bits per second.
    BdtReferenceId:
      type: string
      description: string identifying a BDT Reference ID as defined in subclause 5.3.3 of
3GPP TS 29.154 [9].
    Binary:
      type: string
      description: string with format "binary" as defined in OpenAPI Specification [27].
    Bytes:
      type: string
      description: String with format "byte" as defined in OpenAPI Specification [27], i.e, base64-
encoded characters.
    DayOfWeek:
      type: integer
      description: integer between and including 1 and 7 denoting a weekday. 1 shall indicate
Monday, and the subsequent weekdays shall be indicated with the next higher numbers. 7 shall
indicate Sunday.
    DateTime:
      type: string
      description: string with format "date-time" as defined in OpenAPI [27].
    DateTimeRm:
      type: string
      description: string with format "date-time" as defined in OpenAPI [27] with "nullable=true"
property.
      nullable: true
    DurationSec:
      type: integer
      description: Unsigned integer identifying a period of time in units of seconds.
    DurationSecRm:
      type: integer
      description: Unsigned integer identifying a period of time in units of seconds with
"nullable=true" property.
      nullable: true
    DurationSecRo:
      type: integer
      description: Unsigned integer identifying a period of time in units of seconds with
"readonly=true" property.
      readOnly: true
    DurationMin:
      type: integer
      description: Unsigned integer identifying a period of time in units of minutes.
    ExternalId:
      type: string
      description: string containing a local identifier followed by "@" and a domain identifier.
Both the local identifier and the domain identifier shall be encoded as strings that do not contain
any "@" characters. See Clause 4.6.2 of 3GPP TS 23.682 [2] for more information.
    ExternalGroupId:
      type: string
      description: string containing a local identifier followed by "@" and a domain identifier. 
Both the local identifier and the domain identifier shall be encoded as strings that do not contain 
any "@" characters. See Clauses 4.6.2 and 4.6.3 of 3GPP TS 23.682 [2] for more information.
    Ipv4Addr:
      type: string
      description: string identifying a Ipv4 address formatted in the "dotted decimal" notation as 
defined in IETF RFC 1166 [28].
    Ipv6Addr:
      type: string
      description: string identifying a Ipv6 address formatted according to clause 4 in 
IETF RFC 5952 [29]. The mixed Ipv4 Ipv6 notation according to clause 5 of IETF RFC 5952 [29] shall 
not be used.
    Link:
      type: string
      description: string formatted according to IETF RFC 3986 [7] identifying a referenced 
resource.
    Mcc:
      type: string
      description: String encoding a Mobile Country Code part of the PLMN, comprising 3 digits, as 
defined in 3GPP TS 38.413 [11].
    Mnc:
      type: string
      description: String encoding a Mobile Network Code part of the PLMN, comprising 2 or 3 digits, 
as defined in 3GPP TS 38.413 [11].
    Msisdn:
      type: string
      description: string formatted according to subclause 3.3 of 3GPP TS 23.003 [14] that describes 
an MSISDN.
    Port:
      type: integer
      description: Unsigned integer with valid values between 0 and 65535.
      minimum: 0
      maximum: 65535
    ResourceId:
      type: string
      description: string chosen by the SCEF to serve as identifier in a resource URI.
    ScsAsId:
      type: string
      description: string that identifies an SCS/AS.
    TimeOfDay:
      type: string
      format: TimeOfDay
      description: String with format partial-time or full-time as defined in subclause 5.6 of IETF 
RFC 3339. Examples, 20:15:00, 20:15:00-08:00 (for 8 hours behind UTC).
    Uri:
      type: string
      description: string providing an URI formatted according to IETF RFC 3986 [7].
    Volume:
      type: integer
      description: Unsigned integer identifying a volume in units of bytes.
    VolumeRm:
      type: integer
      description: Unsigned integer identifying a volume in units of bytes with "nullable=true" 
property.
      nullable: true
    Event:
      anyOf:
      - type: string
        enum:
          - SESSION_TERMINATION
          - LOSS_OF_BEARER
          - RECOVERY_OF_BEARER
          - RELEASE_OF_BEARER
          - USAGE_REPORT
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - SESSION_TERMINATION: Indicates that Rx session is terminated.
        - LOSS_OF_BEARER : Indicates a loss of a bearer.
        - RECOVERY_OF_BEARER: Indicates a recovery of a bearer.
        - RELEASE_OF_BEARER: Indicates a release of a bearer.
        - USAGE_REPORT: Indicates the usage report event.
    ResultReason:
      anyOf:
```

```
      - type: string
        enum:
          - ROAMING_NOT_ALLOWED
          - OTHER_REASON
      - type: string
        description: >
          This string provides a failure reason.
      description: >
        Possible values are
        - ROAMING_NOT_ALLOWED: Identifies the configuration parameters are not allowed by roaming
agreement.
        - OTHER_REASON: Identifies the configuration parameters are not configured due to other
reason.
#
# HTTP responses
#
  responses:
    '400':
      description: Bad request
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '401':
      description: Unauthorized
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '403':
      description: Forbidden
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '404':
      description: Not Found
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '409':
      description: Conflict
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '411':
      description: Length Required
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '412':
      description: Precondition Failed
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '413':
      description: Payload Too Large
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '414':
      description: URI Too Long
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '500':
      description: Internal Server Error
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '503':
      description: Service Unavailable
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    default:
      description: Generic Error
```

A.3 MonitoringEvent API

```
openapi: 3.0.0
info:
  title: 3gpp-monitoring-event
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-monitoring-event/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /{scsAsId}/subscriptions:
    get:
      summary: read all of the active subscriptions for the SCS/AS
      tags:
        - MonitoringEvent API SCS/AS level GET Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (Successful get all of the active subscriptions for the SCS/AS)
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/MonitoringEventSubscription'
                minItems: 0
                description: Monitoring event subscriptions
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

    post:
      summary: Creates a new subscription resource for monitoring event notification
      tags:
        - MonitoringEvent API Subscription level POST Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
      requestBody:
        description: Subscription for notification about monitoring event
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/MonitoringEventSubscription'
      callbacks:
        notificationDestination:
          '{request.body#/notificationDestination}':
            post:
              requestBody:  # contents of the callback message
                required: true
                content:
                  application/json:
                    schema:
                      $ref: '#/components/schemas/MonitoringNotification'
              responses:
                '204':
                  description: No Content (successful notification)
                '400':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/400'
                '401':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/401'
                '403':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/403'
                '409':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/409'
                '411':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/411'
                '412':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/412'
                '413':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/413'
                '500':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/500'
                '503':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/503'
                default:
                  $ref: 'TS29122_CommonData.yaml#/components/responses/default'
      responses:
        '201':
          description: Created (Successful creation of subscription)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/MonitoringEventSubscription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /{scsAsId}/subscriptions/{subscriptionId}:
    get:
      summary: read an active subscriptions for the SCS/AS and the subscription Id
      tags:
        - MonitoringEvent API Subscription level GET Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Identifier of the subscription resource
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (Successful get the active subscription)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/MonitoringEventSubscription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

    put:
      summary: Updates/replaces an existing subscription resource
      tags:
        - MonitoringEvent API subscription level PUT Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Identifier of the subscription resource
          required: true
          schema:
            type: string
      requestBody:
        description: Parameters to update/replace the existing subscription
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/MonitoringEventSubscription'
      responses:
        '200':
          description: OK (Successful update of the subscription)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/MonitoringEventSubscription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

    delete:
      summary: Deletes an already existing monitoring event subscription
      tags:
        - MonitoringEvent API Subscription level DELETE Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Identifier of the subscription resource
          required: true
          schema:
            type: string
      responses:
        '204':
          description: No Content (Successful deletion of the existing subscription)
        '200':
          description: OK (Successful deletion of the existing subscription)
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/MonitoringEventReport'
                minItems: 1
                description: The subscription was terminated successfully, the monitoring event 
report(s) shall be included if received.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    MonitoringEventSubscription:
      type: object
      properties:
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        externalId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalId'
        msisdn:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Msisdn'
        externalGroupId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalGroupId'
        ipv4Addr:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv4Addr'
        ipv6Addr :
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv6Addr'
        notificationDestination:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        requestTestNotification:
          type: boolean
          description: Set to true by the SCS/AS to request the SCEF to send a test notification as 
defined in subclause 5.2.5.3. Set to false or omitted otherwise.
        websockNotifConfig:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
        monitoringType:
          $ref: '#/components/schemas/MonitoringType'
        maximumNumberOfReports:
          type: integer
          description: Identifies the maximum number of event reports to be generated by the HSS, 
MME/SGSN as specified in subclause 5.6.0 of 3GPP TS 23.682 [2].
        monitorExpireTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        groupReporGuardTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        maximumDetectionTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        reachabilityType:
          $ref: '#/components/schemas/ReachabilityType'
        maximumLatency:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        maximumResponseTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        suggestedNumberOfDlPackets:
          type: integer
          description: If "monitoringType" is "UE_REACHABILITY", this parameter may be included to 
identify the number of packets that the serving gateway shall buffer in case that the UE is not 
reachable.
        idleStatusIndication:
          type: boolean
          description: If "monitoringType" is set to "UE_REACHABILITY" or 
"AVAILABILITY_AFTER_DDN_FAILURE", this parameter may be included to indicate the notification of 
when a UE, for which PSM is enabled, transitions into idle mode. -  "true"  indicate enabling of 
notification -  "false"  indicate no need to notify Default  "false".
        locationType:
          $ref: '#/components/schemas/LocationType'
        accuracy:
          $ref: '#/components/schemas/Accuracy'
        minimumReportInterval:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        associationType:
          $ref: '#/components/schemas/AssociationType'
        plmnIndication:
          type: boolean
          description: If "monitoring-Type" is "ROAMING_STATUS", this parameter may be included to 
indicate the notification of UE's Serving PLMN ID. -    "true"  The value shall be used to indicate 
enabling of notification; - "false"  The value shall be used to indicate disabling of notification. 
Default  "false".
        locationArea:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/LocationArea'
        monitoringEventReport:
          $ref: '#/components/schemas/MonitoringEventReport'
      required:
        - notificationDestination
        - monitoringType
    MonitoringNotification:
      type: object
      properties:
        subscription:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        configResults:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/ConfigResult'
          minItems: 1
          description: Each element identifies a notification of grouping configuration result.
        monitoringEventReports:
          type: array
          items:
            $ref: '#/components/schemas/MonitoringEventReport'
          minItems: 0
          description: Monitoring event reports.
        cancelInd:
          type: boolean
          description: Indicates whether to request to cancel the corresponding monitoring 
subscription. Set to false or omitted otherwise.
      required:
        - subscription
    MonitoringEventReport:
      type: object
      properties:
        imeiChange:
          $ref: '#/components/schemas/AssociationType'
        externalIds:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalId'
          minItems: 0
          description: External identifier
        idleStatusInfo:
          $ref: '#/components/schemas/IdleStatusInfo'
        locationInfo:
          $ref: '#/components/schemas/LocationInfo'
        lossOfConnectReason:
          type: integer
          description: If "monitoring-Type" is "LOSS_OF_CONNECTIVITY", this parameter shall be
included if available to identify the reason why loss of connectivity is reported. Refer to
3GPP TS 29.336 [11] Subclause 8.4.58.
        maxUEAvailabilityTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        msisdns:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/Msisdn'
          minItems: 0
          description: Identifies the MS internal PSTN/ISDN number
        monitoringType:
          $ref: '#/components/schemas/MonitoringType'
        uePerLocationReport:
          $ref: '#/components/schemas/UePerLocationReport'
        plmnId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/PlmnId'
        reachabilityType:
          $ref: '#/components/schemas/ReachabilityType'
        roamingStatus:
          type: string
        failureCause:
          $ref: '#/components/schemas/FailureCause'
      required:
        - monitoringType
    IdleStatusInfo:
      type: object
      properties:
        activeTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        edrxCycleLength:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        suggestedNumberOfDlPackets:
          type: integer
          description: Identifies the number of packets shall be buffered in the serving gateway. It
shall be present if the idle status indication is requested by the SCS/AS with
"idleStatusIndication" in the "monitoringEventSubscription" sets to "true".
        idleStatusTimestamp:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        periodicAUTimer:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
    UePerLocationReport:
      type: object
      properties:
        ueCount:
          type: integer
          description: Identifies the number of Ues.
        externalIds:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalId'
          minItems: 0
          description: Each element uniquely identifies a user.
        Msisdns:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/Msisdn'
          minItems: 0
          description: Each element identifies the MS internal PSTN/ISDN number allocated for a UE.
      required:
        - ueCount
    LocationInfo:
      type: object
      properties:
        ageOfLocationInfo:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationMin'
        cellId:
          type: string
          description: Indicates the Cell Global Identification of the user which identifies the
cell the UE is registered.
        enodeBId:
          type: string
          description: Indicates the eNodeB in which the UE is currently located.
        routingAreaId:
          type: string
          description: Identifies the Routing Area Identity of the user where the UE is located.
        trackingAreaId:
          type: string
          description: Identifies the Tracking Area Identity of the user where the UE is located.
        plmmId:
          type: string
          description: Identifies the PLMN Identity of the user where the UE is located.
        twanId:
          type: string
          description: Identifies the TWAN Identity of the user where the UE is located.
    FailureCause:
      type: object
      properties:
        bssgpCause:
          type: integer
          description: Identifies a non-transparent copy of the BSSGP cause code. Refer to
3GPP TS 29.128 [12].
        causeType:
          type: integer
          description: Identify the type of the S1AP-Cause. Refer to 3GPP TS 29.128 [12].
        gmmCause:
          type: integer
          description: Identifies a non-transparent copy of the GMM cause code. Refer to
3GPP TS 29.128 [12].
        ranapCause:
          type: integer
          description: Identifies a non-transparent copy of the RANAP cause code. Refer to
3GPP TS 29.128 [12].
        ranNasCause:
          type: string
          description: Indicates RAN and/or NAS release cause code information, TWAN release cause
code information or untrusted WLAN release cause code information. Refer to 3GPP TS 29.214 [10].
        S1ApCause:
          type: integer
          description: Identifies a non-transparent copy of the S1AP cause code. Refer to
3GPP TS 29.128 [12].
        smCause:
          type: integer
          description: Identifies a non-transparent copy of the SM cause code. Refer to
3GPP TS 29.128 [12].
    MonitoringType:
      anyOf:
      - type: string
        enum:
          - LOSS_OF_CONNECTIVITY
          - UE_REACHABILITY
          - LOCATION_REPORTING
          - CHANGE_OF_IMSI_IMEI_ASSOCIATION
          - ROAMING_STATUS
          - COMMUNICATION_FAILURE
          - AVAILABILITY_AFTER_DDN_FAILURE
          - NUMBER_OF_UES_IN_AN_AREA
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - LOSS_OF_CONNECTIVITY: The SCS/AS requests to be notified when the 3GPP network detects
that the UE is no longer reachable for signalling or user plane communication
        - UE_REACHABILITY: The SCS/AS requests to be notified when the UE becomes reachable for
sending either SMS or downlink data to the UE
        - LOCATION_REPORTING: The SCS/AS requests to be notified of the current location or the last
known location of the UE
        - CHANGE_OF_IMSI_IMEI_ASSOCIATION: The SCS/AS requests to be notified when the association
of an ME (IMEI(SV)) that uses a specific subscription (IMSI) is changed
        - ROAMING_STATUS: The SCS/AS queries the UE's current roaming status and requests to get
notified when the status changes
        - COMMUNICATION_FAILURE: The SCS/AS requests to be notified of communication failure events
        - AVAILABILITY_AFTER_DDN_FAILURE: The SCS/AS requests to be notified when the UE has become
available after a DDN failure
        - NUMBER_OF_UES_IN_AN_AREA: The SCS/AS requests to be notified the number of Ues in a given
geographic area
    ReachabilityType:
      anyOf:
      - type: string
        enum:
          - SMS
          - DATA
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - SMS : The SCS/AS requests to be notified when the UE becomes reachable for sending SMS to
the UE
        - DATA: The SCS/AS requests to be notified when the UE becomes reachable for sending
downlink data to the UE
    LocationType:
      anyOf:
      - type: string
        enum:
          - CURRENT_LOCATION
          - LAST_KNOWN_LOCATION
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - CURRENT_LOCATION: The SCS/AS requests to be notified for current location
        - LAST_KNOWN_LOCATION: The SCS/AS requests to be notified for last known location
    AssociationType:
      anyOf:
      - type: string
        enum:
          - IMEI
          - IMEISV
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - IMEI: The value shall be used when the change of IMSI-IMEI association shall be detected
        - IMEISV: The value shall be used when the change of IMSI-IMEISV association shall be
detected
    Accuracy:
      anyOf:
      - type: string
        enum:
          - CGI_ECGI
          - ENODEB
          - TA_RA
          - PLMN
          - TWAN_ID
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - CGI_ECGI: The SCS/AS requests to be notified at cell level location accuracy.
        - ENODEB: The SCS/AS requests to be notified at eNodeB level location accuracy.
        - TA_RA: The SCS/AS requests to be notified at TA/RA level location accuracy.
        - PLMN: The SCS/AS requests to be notified at PLMN level location accuracy.
        - TWAN_ID: The SCS/AS requests to be notified at TWAN identifier level location accuracy.
```

A.4 ResourceManagementOfBdt API

```
openapi: 3.0.0
info:
  title: 3gpp-bdt
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-bdt/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /{scsAsId}/subscriptions:
    parameters:
      - name: scsAsId
        description: String identifying the SCS/AS.
        in: path
        required: true
        schema:
          type: string
    get:
      responses:
        '200':
          description: all BDT policy subscriptions.
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Bdt'
                minItems: 0
                description: individual BDT policy subscription.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    post:
      requestBody:
        description: Contains the data to create a BDT subscription.
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Bdt'
      responses:
        '201':
          description: Background data transfer policies offered to the SCS/AS.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Bdt'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
  /{scsAsId}/subscriptions/{subscriptionId}:
    parameters:
      - name: scsAsId
        description: String identifying the SCS/AS.
        in: path
        required: true
        schema:
          type: string
      - name: subscriptionId
        description: String identifying the individual BDT policy resource in the SCEF.
        in: path
        required: true
        schema:
          type: string
    get:
      responses:
        '200':
          description: Background data transfer policies offered to and selected by the SCEF.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Bdt'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    patch:
      requestBody:
        description: Contains information to be performed on the Bdt data structure to select a 
transfer policy.
        required: true
        content:
          application/merge-patch+json:
            schema:
              $ref: '#/components/schemas/BdtPatch'
      responses:
        '200':
          description: The Individual BDT Policy resource is modified with a selected policy and a 
representation of that resource is returned.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Bdt'
        '204':
          description: The Individual BDT Policy resource is modified with a selected policy.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    delete:
      responses:
        '204':
          description: The Individual BDT Policy resource is deleted.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    Bdt:
      type: object
      properties:
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        volumePerUE:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/UsageThreshold'
        numberOfUEs:
          type: integer
          description: Identifies the number of Ues.
        desiredTimeWindow:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/TimeWindow'
        locationArea:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/LocationArea'
        referenceId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/BdtReferenceId'
        transferPolicies:
          type: array
          items:
            $ref: '#/components/schemas/TransferPolicy'
          minItems: 0
          description: Identifies an offered transfer policy.
          readOnly: true
        selectedPolicy:
          type: integer
          description: Identity of the selected background data transfer policy. Shall not be
present in initial message exchange, can be provided by NF service consumer in a subsequent message
exchange.
      required:
        - volumePerUE
        - numberOfUEs
        - desiredTimeWindow
    BdtPatch:
      type: object
      properties:
        selectedPolicy:
          type: integer
          description: Identity of the selected background data transfer policy.
      required:
        - selectedPolicy
    TransferPolicy:
      type: object
      properties:
        bdtPolicyId:
          type: integer
          description: Identifier for the transfer policy
        maxUplinkBandwidth:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Bandwidth'
        maxDownlinkBandwidth:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Bandwidth'
        ratingGroup:
          type: integer
          description: Indicates the rating group during the time window.
        timeWindow:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/TimeWindow'
      required:
        - bdtPolicyId
        - ratingGroup
        - timeWindow
```

A.5 ChargeableParty API

```
openapi: 3.0.0
info:
  title: 3gpp-chargeable-party
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-chargeable-party/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /{scsAsId}/transactions:
    get:
      summary: Read all chargeable party transaction resources for a given SCS/AS
      tags:
        - Chargeable Party Transaction Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (successful query of Chargeable Party resource)
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/ChargeableParty'
                minItems: 0
                description: individual BDT policy subscription.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

    post:
      summary:  Create a new chargeable party transaction resource
      tags:
        - Chargeable Party Transaction Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
```

```
          type: string
      requestBody:
        description: representation of the Chargeable Party resource to be Created in the SCEF
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ChargeableParty'
      callbacks:
        eventNotification:
          '{$request.body#/notificationUri}':
            post:
              requestBody:  # contents of the callback message
                required: true
                content:
                  application/json:
                    schema:
                      $ref: 'TS29122_CommonData.yaml#/components/schemas//notificationData'
              responses:
                '200':
                  description: OK (The successful acknowledgement of the notification with a body)
                '400':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/400'
                '401':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/401'
                '403':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/403'
                '409':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/409'
                '411':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/411'
                '412':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/412'
                '413':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/413'
                '500':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/500'
                '503':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/503'
                default:
                  $ref: 'TS29122_CommonData.yaml#/components/responses/default'
      responses:
        '201':
          description: successful creation of a chargeable party resource
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ChargeableParty'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /{scsAsId}/transactions/{transactionId}:
    get:
      summary: read a chargeable party resource for a given SCS/AS and a transaction Id
      tags:
        - Individual chargeable party resource Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: transactionId
          in: path
          description: Identifier of transaction
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (successful query of a chargeable party resource)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ChargeableParty'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

    patch:
      summary:  Updates a existing chargeable party resource for a given SCS/AS and transaction Id.
      tags:
        - Individual chargeable party resource Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: transactionId
          in: path
          description: Identifier of transaction
          required: true
          schema:
            type: string
      requestBody:
        description: representation of the chargeable party resource to be udpated in the SCEF
        required: true
        content:
          application/merge-patch+json:
            schema:
              $ref: '#/components/schemas/ChargeablePartyPatch'
      responses:
        '200':
          description: successful update of a chargeable party resource
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ChargeableParty'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

    delete:
      summary:  deletes a chargeable party resource for a given SCS/AS and a transcation Id.
      tags:
        - Individual chargeable party resource Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: transactionId
          in: path
          description: Identifier of transaction
          required: true
          schema:
            type: string
      responses:
        '200':
          description: successful deletion of an resouce of chargeable party
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    ChargeableParty:
      type: object
      properties:
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        notificationDestination:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        requestTestNotification:
          type: boolean
          description: Set to true by the SCS/AS to request the SCEF to send a test notification as
defined in subclause 5.2.5.3. Set to false or omitted otherwise.
        websockNotifConfig:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
        ipv4Addr:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv4Addr'
        ipv6Addr :
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv6Addr'
        flowInfo:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/FlowInfo'
          minItems: 1
          description: Describes the application flows.
        sponsorInformation:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/SponsorInformation'
        sponsoringEnabled:
          type: boolean
          description: Indicates sponsoring status.
        referenceId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/BdtReferenceId'
        usageThreshold:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/UsageThreshold'
      required:
        - notificationDestination
        - flowInfo
        - sponsorInformation
        - sponsoringEnabled
    ChargeablePartyPatch:
      type: object
      properties:
        flowInfo:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/FlowInfo'
          minItems: 0
          description: Describes the application flows.
        sponsoringEnabled:
          type: boolean
          description: Indicates sponsoring status.
        referenceId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/BdtReferenceId'
        usageThreshold:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/UsageThresholdRm'
```

A.6 NIDD API

```
openapi: 3.0.0
info:
  title: 3gpp-nidd
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-nidd/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /{scsAsId}/configurations:
    parameters:
      - name: scsAsId
        description: String identifying the SCS/AS.
        in: path
        required: true
        schema:
          type: string
    get:
      responses:
        '200':
          description: all NIDD configurations.
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/NiddConfiguration'
                minItems: 0
                description: individual NIDD configuration.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    post:
      requestBody:
        description: Contains the data to create a NIDD configuration.
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/NiddConfiguration'
      responses:
        '201':
          description: NIDD configuration is successfully created.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NiddConfiguration'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
      callbacks:
        niddNotifications:
          '{$request.body#/notificationDestination}':
            post:
              requestBody:
                description: Notification for NIDD configuration status, MO NIDD, MT NIDD delivery 
report.
                content:
                  application/json:
                    schema:
                      oneOf:
                      - $ref: '#/components/schemas/NiddConfigurationStatusNotification'
                      - $ref: '#/components/schemas/NiddUplinkDataNotification'
                      - $ref: '#/components/schemas/NiddDownlinkDataDeliveryStatusNotification'
                      - $ref: '#/components/schemas/GmdNiddDownlinkDataDeliveryNotification'
              responses:
                '204':
                  description: Expected response to a successful callback processing without a body
                '200':
                  description: Expected response to a successful callback processing with a body
                  content:
                    application/json:
                      schema:
                        $ref: 'TS29122_CommonData.yaml#/components/schemas/Acknowledgement'
                '400':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/400'
                '401':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/401'
                '403':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/403'
                '409':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/409'
                '411':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/411'
                '412':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/412'
                '413':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/413'
                '500':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/500'
                '503':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/503'
                default:
                  $ref: 'TS29122_CommonData.yaml#/components/responses/default'
  /{scsAsId}/configurations/{configurationId}:
    parameters:
      - name: scsAsId
        description: String identifying the SCS/AS.
        in: path
        required: true
        schema:
          type: string
      - name: configurationId
        description: String identifying the individual NIDD configuration resource in the SCEF.
        in: path
        required: true
        schema:
          type: string
    get:
      responses:
        '200':
          description: The individual NIDD configuration is successfully retrieved.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NiddConfiguration'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    patch:
      requestBody:
        description: Contains information to be applied to the individual NIDD configuration.
        required: true
        content:
          application/merge-patch+json:
            schema:
              $ref: '#/components/schemas/NiddConfigurationPatch'
      responses:
        '200':
          description: The Individual NIDD configuration is modified successfully and a 
representation of that resource is returned.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NiddConfiguration'
        '204':
          description: The Individual NIDD configuration is modified successfully.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    delete:
      responses:
        '204':
          description: The Individual NIDD configuration is deleted.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /{scsAsId}/configurations/{configurationId}/downlink-data-deliveries:
    parameters:
      - name: scsAsId
        description: String identifying the SCS/AS.
        in: path
        required: true
        schema:
          type: string
      - name: configurationId
        description: String identifying the individual NIDD configuration resource in the SCEF.
        in: path
        required: true
        schema:
          type: string
    get:
      responses:
        '200':
          description: all NIDD downlink data deliveries.
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/NiddDownlinkDataTransfer'
                minItems: 0
                description: individual NIDD downlink data delivery.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    post:
      requestBody:
        description: Contains the data to create a NIDD downlink data delivery.
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/NiddDownlinkDataTransfer'
      responses:
        '200':
          description: NIDD downlink data delivery is successful.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NiddDownlinkDataTransfer'
        '201':
          description: NIDD downlink data delivery is pending.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NiddDownlinkDataTransfer'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
  /{scsAsId}/configurations/{configurationId}/downlink-data-deliveries/{downlinkDataDeliveryId}:
    parameters:
      - name: scsAsId
        description: String identifying the SCS/AS.
        in: path
        required: true
        schema:
          type: string
      - name: configurationId
        description: String identifying the individual NIDD configuration resource in the SCEF.
        in: path
        required: true
        schema:
          type: string
      - name: downlinkDataDeliveryId
        description: String identifying the individual NIDD downlink data delivery in the SCEF.
        in: path
        required: true
        schema:
          type: string
    get:
      responses:
        '200':
          description: The individual NIDD downlink data delivery is successfully retrieved.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NiddDownlinkDataTransfer'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    put:
      requestBody:
        description: Contains information to be applied to the individual NIDD downlink data
delivery.
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/NiddDownlinkDataTransfer'
      responses:
        '200':
          description: The pending NIDD downlink data is replaced sucessfully but delivery is
pending.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NiddDownlinkDataTransfer'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    delete:
      responses:
        '204':
          description: The pending NIDD downlink data is deleted.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    NiddConfiguration:
      type: object
      properties:
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        externalId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalId'
        msisdn:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Msisdn'
        externalGroupId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalGroupId'
        duration:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        reliableDataService:
          type: boolean
          description: The reliable data service (as defined in subclause 4.5.15.3 of
3GPP TS 23.682 [2]) to indicate if a reliable data service acknowledgment is enabled or not.
        rdsPorts:
          type: array
          items:
            $ref: '#/components/schemas/RdsPort'
          minItems: 0
          description: Indicates the port configuration that is used for reliable data transfer 
between specific applications using RDS (as defined in subclause 5.2.4 and 5.2.5 of 
3GPP TS 24.250 [31]).
        pdnEstablishmentOption:
          $ref: '#/components/schemas/PdnEstablishmentOptions'
        notificationDestination:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        requestTestNotification:
          type: boolean
          description: Set to true by the SCS/AS to request the SCEF to send a test notification as 
defined in subclause 5.2.5.3. Set to false or omitted otherwise.
        websockNotifConfig:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
        maximumPacketSize:
          type: integer
          description: The Maximum Packet Size is the maximum NIDD packet size that was transferred 
to the UE by the SCEF in the PCO, see subclause 4.5.14.1 of 3GPP TS 23.682 [2]. If no maximum packet 
size was provided to the UE by the SCEF, the SCEF sends a default configured max packet size to 
SCS/AS. Unit  bit.
          readOnly: true
        niddDownlinkDataTransfers:
          type: array
          items:
            $ref: '#/components/schemas/NiddDownlinkDataTransfer'
          minItems: 0
          description: The downlink data deliveries that needed to be executed by the SCEF. The 
cardinality of the property shall be 0..1 in the request and 0..N in the response (i.e. response may 
contain multiple buffered MT NIDD).
        status:
          $ref: '#/components/schemas/NiddStatus'
      required:
        - notificationDestination
    NiddDownlinkDataTransfer:
      type: object
      properties:
        externalId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalId'
        externalGroupId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalGroupId'
        msisdn:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Msisdn'
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        data:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Binary'
        reliableDataService:
          type: boolean
          description: The reliable data service (as defined in subclause 4.5.15.3 of 
3GPP TS 23.682 [2]) to indicate if a reliable data service acknowledgment is enabled or not.
        rdsPort:
          $ref: '#/components/schemas/RdsPort'
        maximumLatency:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        priority:
          type: integer
          description: It is used to indicate the priority of the non-IP data packet relative to 
other non-IP data packets.
        pdnEstablishmentOption:
          $ref: '#/components/schemas/PdnEstablishmentOptions'
        deliveryStatus:
          $ref: '#/components/schemas/DeliveryStatus'
        requestedRetransmissionTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
      required:
        - data
    NiddUplinkDataNotification:
      type: object
      properties:
        niddConfiguration:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        externalId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalId'
        msisdn:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Msisdn'
        data:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Binary'
        reliableDataService:
          type: boolean
          description: Indicates whether the reliable data service is enabled.
        rdsPort:
          $ref: '#/components/schemas/RdsPort'
      required:
        - niddConfiguration
        - data
    NiddDownlinkDataDeliveryStatusNotification:
      type: object
      properties:
        niddDownlinkDataTransfer:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        deliveryStatus:
          $ref: '#/components/schemas/DeliveryStatus'
        requestedRetransmissionTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
      required:
        - niddDownlinkDataTransfer
        - deliveryStatus
    NiddConfigurationStatusNotification:
      type: object
      properties:
        niddConfiguration:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        externalId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalId'
        msisdn:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Msisdn'
        status:
          $ref: '#/components/schemas/NiddStatus'
      required:
        - niddConfiguration
        - status
    GmdNiddDownlinkDataDeliveryNotification:
      type: object
      properties:
        niddDownlinkDataTransfer:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        gmdResults:
          type: array
          items:
            $ref: '#/components/schemas/GmdResult'
          minItems: 1
          description: Indicates the group message delivery result.
      required:
        - niddDownlinkDataTransfer
        - gmdResults
    RdsPort:
      type: object
      properties:
        portUE:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Port'
        portSCEF:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Port'
      required:
        - portUE
        - portSCEF
    GmdResult:
      type: object
      properties:
        deliveryStatus:
          $ref: '#/components/schemas/DeliveryStatus'
        requestedRetransmissionTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
      required:
        - deliveryStatus
    PdnEstablishmentOptions:
      anyOf:
      - type: string
        enum:
          - WAIT_FOR_UE
          - INDICATE_ERROR
          - SEND_TRIGGER
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
```

```
        Possible values are
        - WAIT_FOR_UE: wait for the UE to establish the PDN connection
        - INDICATE_ERROR: respond with an error cause
        - SEND_TRIGGER: send a device trigger
    PdnEstablishmentOptionsRm:
      anyOf:
      - type: string
        enum:
          - WAIT_FOR_UE
          - INDICATE_ERROR
          - SEND_TRIGGER
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - WAIT_FOR_UE: wait for the UE to establish the PDN connection
        - INDICATE_ERROR: respond with an error cause
        - SEND_TRIGGER: send a device trigger
      nullable: true
    DeliveryStatus:
      anyOf:
      - type: string
        enum:
          - SUCCESS
          - SUCCESS_NEXT_HOP_ACKNOWLEDGED
          - SUCCESS_NEXT_HOP_UNACKNOWLEDGED
          - SUCCESS_ACKNOWLEDGED
          - SUCCESS_UNACKNOWLEDGED
          - TRIGGERING
          - BUFFERING
          - BUFFERING_TEMPORARILY_NOT_REACHABLE
          - SENDING
          - STOPPED_BUT_TRIGGERING
          - FAILURE
          - FAILURE_QUOTA_EXCEEDED
          - FAILURE_RATE_EXCEEDED
          - FAILURE_DATA_TOO_LARGE
          - FAILURE_TEMORARILY_NOT_REACHABLE
          - FAILURE_NEXT_HOP
          - FAILURE_TIMEOUT
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - SUCCESS: Success but details not provided
        - SUCCESS_NEXT_HOP_ACKNOWLEDGED: Successful delivery to the next hop with acknowledgment.
        - SUCCESS_NEXT_HOP_UNACKNOWLEDGED: Successful delivery to the next hop without 
acknowledgment
        - SUCCESS_ACKNOWLEDGED: Reliable delivery was acknowledged by the UE
        - SUCCESS_UNACKNOWLEDGED: Reliable delivery was not acknowledged by the UE
        - TRIGGERING: The SCEF is triggering the device and buffering the data.
        - BUFFERING: The SCEF is buffering the data due to no PDN connection established.
        - BUFFERING_TEMPORARILY_NOT_REACHABLE: The SCEF has been informed that the UE is temporarily 
not reachable but is buffering the data
        - SENDING: The SCEF has forwarded the data, but they may be stored elsewhere
        - STOPPED_BUT_TRIGGERING: The SCEF is triggering the device but did not buffer the data. The 
SCS AS may resubmit the data
        - FAILURE: Delivery failure but details not provided
        - FAILURE_QUOTA_EXCEEDED: Not enough quota for the MT NIDD
        - FAILURE_RATE_EXCEEDED: MT NIDD sending rate is exceeded.
        - FAILURE_DATA_TOO_LARGE: The non-IP data size is larger than "maximumPacketSize" of the 
NIDD configuration.
        - FAILURE_TEMORARILY_NOT_REACHABLE: The SCEF has aborted the delivery because the UE is 
temporarily not reachable. The SCEF may in addition indicate a requested re-submission time for the 
data.
        - FAILURE_NEXT_HOP: Unsuccessful delivery to the next hop.
        - FAILURE_TIMEOUT: Unsuccessful delivery due to timeout.
      readOnly: true
    NiddStatus:
      anyOf:
      - type: string
        enum:
          - ACTIVE
          - TERMINATED_UE_NOT_AUTHORIZED
          - TERMINATED
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - ACTIVE: The NIDD configuration is active.
        - TERMINATED_UE_NOT_AUTHORIZED: The NIDD configuration was terminated because the UE´s 
authorisation was revoked.
        - TERMINATED: The NIDD configuration was terminated.
      readOnly: true
    NiddConfigurationPatch:
      type: object
      properties:
        duration:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTimeRm'
        reliableDataService:
          type: boolean
          description: The reliable data service (as defined in subclause 4.5.15.3 of 
3GPP TS 23.682 [2]) to indicate if a reliable data service acknowledgment is enabled or not.
          nullable: true
        rdsPorts:
          type: array
          items:
            $ref: '#/components/schemas/RdsPort'
          minItems: 0
          description: Indicates the port configuration that is used for reliable data transfer 
between specific applications using RDS (as defined in subclause 5.2.4 and 5.2.5 of 
3GPP TS 24.250 [31]).
        pdnEstablishmentOption:
          $ref: '#/components/schemas/PdnEstablishmentOptionsRm'
```

A.7 DeviceTriggering API

```
openapi: 3.0.0
info:
  title: 3gpp-device-triggering
  version: "1.PreR15.1.0"
servers:
  - url: 'apiRoot}/3gpp-device-triggering/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /{scsAsId}/transactions:
    get:
      summary: read all active device triggering transactions for a given SCS/AS.
      tags:
        - Device Triggering API SCS/AS level GET Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/ScsASId'
      responses:
        '200':
          description: OK (Successful get all of the active device triggering transactions for the 
SCS/AS)
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/DeviceTriggering'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    post:
      summary: Create a long-term transaction for a device triggering.
      tags:
        - DeviceTriggering API Transaction level POST Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
           $ref: 'TS29122_CommonData.yaml#/components/schemas/ScsASId'
      requestBody:
        description: Parameters to request a device triggering delivery.
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/DeviceTriggering'
      callbacks:
        notificationDestination:
          '{request.body#/notificationDestination}':
            post:
              requestBody:  # contents of the callback message
                required: true
                content:
                  application/json:
                    schema:
                      $ref: '#/components/schemas/DeviceTriggeringDeliveryReportNotification'
              responses:
                '200':
                  description: OK (successful notification)
                  content:
                   application/json:
                    schema:
                     $ref: 'TS29122_CommonData.yaml#/components/Acknowledgement'
                '204':
                  description: No Content (successful notification)
                '400':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/400'
                '401':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/401'
                '403':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/403'
                '409':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/409'
                '411':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/411'
                '412':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/412'
                '413':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/413'
                '500':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/500'
                '503':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/503'
                default:
                  $ref: 'TS29122_CommonData.yaml#/components/responses/default'
      responses:
        '201':
          description: Created (Successful creation of subscription)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/DeviceTriggering'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
  /{scsAsId}/transactions/{transactionId}:
    get:
      summary: Read a device triggering transaction resource.
      tags:
        - DeviceTriggering API Transaction level GET Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
           $ref: 'TS29122_CommonData.yaml#/components/schemas/ScsASId'
        - name: transactionId
          in: path
          description: Identifier of the transaction resource
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (Successful get the active subscription)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/DeviceTriggering'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    put:
      summary: Replace an existing device triggering transaction resource and the corresponding
device trigger request.
      tags:
        - DeviceTriggering API transaction level PUT Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
           $ref: 'TS29122_CommonData.yaml#/components/schemas/ScsASId'
        - name: transactionId
          in: path
          description: Identifier of the transaction resource
          required: true
          schema:
            type: string
      requestBody:
        description: Parameters to update/replace the existing device triggering
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/DeviceTriggering'
      responses:
        '200':
          description: OK (Successful update of the device triggering)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/DeviceTriggering'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    delete:
      summary: Deletes an already existing device triggering transaction.
      tags:
        - DeviceTriggering API Transaction level DELETE Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
           $ref: 'TS29122_CommonData.yaml#/components/schemas/ScsASId'
        - name: transactionId
          in: path
          description: Identifier of the transaction resource
          required: true
          schema:
            type: string
      responses:
        '204':
          description: No Content (Successful deletion of the existing subscription)
        '200':
          description: OK (Successful deletion of the existing subscription)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/DeviceTriggering'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    DeviceTriggering:
      type: object
      properties:
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        externalId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalId'
        msisdn:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Msisdn'
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        validityPeriod:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        priority:
          $ref: '#/components/schemas/Priority'
        applicationPortId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Port'
        triggerPayload:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Binary'
        notificationDestination:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        requestTestNotification:
          type: boolean
          description: Set to true by the SCS/AS to request the SCEF to send a test notification as
defined in subclause 5.2.5.3. Set to false or omitted otherwise.
        websockNotifConfig:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
        deliveryResult:
          $ref: '#/components/schemas/DeliveryResult'
      required:
        - validityPeriod
        - priority
        - applicationPortId
        - triggerPayload
        - notificationDestination
    DeviceTriggeringDeliveryReportNotification:
      type: object
      properties:
        transaction:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        result:
          $ref: '#/components/schemas/DeliveryResult'
      required:
        - transaction
        - result
    DeliveryResult:
      anyOf:
      - type: string
        enum:
          - SUCCESS
          - UNKNOWN
          - FAILURE
          - TRIGGERED
          - EXPIRED
          - UNCONFIRMED
          - REPLACED
          - TERMINATE
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - SUCCESS: This value indicates that the device action request was successfully completed.
        - UNKNOWN: This value indicates any unspecified errors.
        - FAILURE: This value indicates that this trigger encountered a delivery error and is deemed
permanently undeliverable.
        - TRIGGERED: This value indicates that device triggering request is accepted by the SCEF.
        - EXPIRED: This value indicates that the validity period expired before the trigger could be
delivered.
        - UNCONFIRMED: This value indicates that the delivery of the device action request is not
confirmed.
        - REPLACED: This value indicates that the device triggering replace request is accepted by
the SCEF.
        - TERMINATE: This value indicates that the delivery of the device action request is
terminated by the SCS/AS.
      readOnly: true
    Priority:
      anyOf:
      - type: string
        enum:
          - NO_PRIORITY
          - PRIORITY
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - NO_PRIORITY: This value indicates that the device trigger has no priority.
        - PRIORITY: This value indicates that the device trigger has priority.
```

A.8 GMDViaMBMS APIs

A.8.1 GMDviaMBMSbyMB2 API

```
openapi: 3.0.0
info:
  title: GMDviaMBMSbyMB2
  description: API for Group Message Delivery via MBMS by MB2
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-group-message-delivery-mb2/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /{scsAsId}/tmgi-allocation:
    get:
      summary: read all TMGI Allocation resource for a given SCS/AS
      tags:
        - TMGI Allocatin Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (successful query of TMGI Allocation resource)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/TMGIAllocation'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    post:
      summary:  creates a new TMGI Allocation resource for a given SCS/AS
      tags:
        - TMGI Allocatin Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
      requestBody:
        description: representation of the TMGI Allocation to be created in the SCEF
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/TMGIAllocation'
      responses:
        '201':
          description: successful creation of an TMGI Allocation
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/TMGIAllocation'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /{scsAsId}/tmgi-allocation/{tmgi}:
    get:
      summary: read a TMGI Allocation resource for a given SCS/AS and a TMGI
      tags:
        - Individual TMGI Allocatin Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: tmgi
          in: path
          description: TMGI
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (successful query of TMGI Allocation resource)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/TMGIAllocation'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    put:
      summary:  Updates an existing TMGI Allocation resource for a given SCS/AS and a TMGI
      tags:
        - Individual TMGI Allocatin Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: tmgi
          in: path
          description: TMGI
          required: true
          schema:
            type: string
      requestBody:
        description: representation of the TMGI Allocation to be updated in the SCEF
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/TMGIAllocation'
      responses:
        '200':
          description: successful creation of an TMGI Allocation
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/TMGIAllocation'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    patch:
      summary:  Updates an existing TMGI Allocation resource for a given SCS/AS and a TMGI
      tags:
        - Individual TMGI Allocatin Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: tmgi
          in: path
          description: TMGI
          required: true
          schema:
            type: string
      requestBody:
        description: representation of the TMGI Allocation to be updated in the SCEF
        required: true
        content:
```

```
        application/merge-patch+json:
          schema:
            $ref: '#/components/schemas/TMGIAllocation'
    responses:
      '200':
        description: successful creation of an TMGI Allocation
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/TMGIAllocation'
      '400':
        $ref: 'TS29122_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29122_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29122_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29122_CommonData.yaml#/components/responses/404'
      '409':
        $ref: 'TS29122_CommonData.yaml#/components/responses/409'
      '411':
        $ref: 'TS29122_CommonData.yaml#/components/responses/411'
      '412':
        $ref: 'TS29122_CommonData.yaml#/components/responses/412'
      '413':
        $ref: 'TS29122_CommonData.yaml#/components/responses/413'
      '500':
        $ref: 'TS29122_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29122_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29122_CommonData.yaml#/components/responses/default'


  delete:
    summary:  deletes an existing TMGI Allocation resource for a given SCS/AS and a TMGI
    tags:
      - Individual TMGI Allocatin Operation
    parameters:
      - name: scsAsId
        in: path
        description: Identifier of SCS/AS
        required: true
        schema:
          type: string
      - name: tmgi
        in: path
        description: TMGI
        required: true
        schema:
          type: string
    responses:
      '200':
        description: successful deletion of an TMGI Allocation
      '400':
        $ref: 'TS29122_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29122_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29122_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29122_CommonData.yaml#/components/responses/404'
      '412':
        $ref: 'TS29122_CommonData.yaml#/components/responses/412'
      '500':
        $ref: 'TS29122_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29122_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29122_CommonData.yaml#/components/responses/default'

/{scsAsId}/tmgi-allocation/{tmgi}/delivery-via-mbms/:
  get:
    summary: read all group message delivery via MBMS resource for a given SCS/AS and a TMGI
    tags:
      - Delivery via MBMS Operation
    parameters:
      - name: scsAsId
```

```
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: tmgi
          in: path
          description: TMGI
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (successful query of Delivery via MBMS resource)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/GMDViaMBMSByMb2'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    post:
      summary:  Creates a new delivery via MBMS for a given SCS/AS and a TMGI
      tags:
        - Delivery via MBMS Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: tmgi
          in: path
          description: TMGI
          required: true
          schema:
            type: string
      requestBody:
        description: representation of the GMD via MBMS by MB2 resource to be Created in the SCEF
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/GMDViaMBMSByMb2'
      callbacks:
        gMDByMb2Notification:
          '{$request.body#/notificationUri}':
            post:
              requestBody:  # contents of the callback message
                required: true
                content:
                  application/json:
                    schema:
                      $ref: '#/components/schemas/GMDByMb2Notification'
              responses:
                '200':
                  description: OK (The successful acknowledgement of the notification with a body)
                  content:
                    application/json:
                      schema:
                        $ref: 'TS29122_CommonData.yaml#/components/schemas/Acknowledgement'
                '204':
```

```
              description: successful notification
            '400':
              $ref: 'TS29122_CommonData.yaml#/components/responses/400'
            '401':
              $ref: 'TS29122_CommonData.yaml#/components/responses/401'
            '403':
              $ref: 'TS29122_CommonData.yaml#/components/responses/403'
            '409':
              $ref: 'TS29122_CommonData.yaml#/components/responses/409'
            '411':
              $ref: 'TS29122_CommonData.yaml#/components/responses/411'
            '412':
              $ref: 'TS29122_CommonData.yaml#/components/responses/412'
            '413':
              $ref: 'TS29122_CommonData.yaml#/components/responses/413'
            '500':
              $ref: 'TS29122_CommonData.yaml#/components/responses/500'
            '503':
              $ref: 'TS29122_CommonData.yaml#/components/responses/503'
            default:
              $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    responses:
      '201':
        description: successful creation of an GMD via MBMS by MB2 resource
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/GMDViaMBMSByMb2'
      '400':
        $ref: 'TS29122_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29122_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29122_CommonData.yaml#/components/responses/403'
      '409':
        $ref: 'TS29122_CommonData.yaml#/components/responses/409'
      '411':
        $ref: 'TS29122_CommonData.yaml#/components/responses/411'
      '412':
        $ref: 'TS29122_CommonData.yaml#/components/responses/412'
      '413':
        $ref: 'TS29122_CommonData.yaml#/components/responses/413'
      '500':
        $ref: 'TS29122_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29122_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29122_CommonData.yaml#/components/responses/default'

/{scsAsId}/tmgi-allocation/{tmgi}/delivery-via-mbms/{transactionId}:
  get:
    summary: read all group message delivery via MBMS resource for a given SCS/AS and a TMGI
    tags:
      - Individual Delivery via MBMS resource Operation
    parameters:
      - name: scsAsId
        in: path
        description: Identifier of SCS/AS
        required: true
        schema:
          type: string
      - name: tmgi
        in: path
        description: TMGI
        required: true
        schema:
          type: string
      - name: transactionId
        in: path
        description: Identifier of transaction
        required: true
        schema:
          type: string
    responses:
      '200':
        description: OK (successful query of an Delivery via MBMS resource)
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/GMDViaMBMSByMb2'
      '400':
        $ref: 'TS29122_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29122_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29122_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29122_CommonData.yaml#/components/responses/404'
      '412':
        $ref: 'TS29122_CommonData.yaml#/components/responses/412'
      '500':
        $ref: 'TS29122_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29122_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29122_CommonData.yaml#/components/responses/default'


  put:
    summary:  Updates a existing delivery via MBMS for a given SCS/AS, a TMGI and transaction Id.
    tags:
      - Individual Delivery via MBMS resource Operation
    parameters:
      - name: scsAsId
        in: path
        description: Identifier of SCS/AS
        required: true
        schema:
          type: string
      - name: tmgi
        in: path
        description: TMGI
        required: true
        schema:
          type: string
      - name: transactionId
        in: path
        description: Identifier of transaction
        required: true
        schema:
          type: string
    requestBody:
      description: representation of the GMD via MBMS by MB2 resource to be udpated in the SCEF
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/GMDViaMBMSByMb2'
    responses:
      '201':
        description: successful creation of an GMD via MBMS by MB2 resource
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/GMDViaMBMSByMb2'
      '400':
        $ref: 'TS29122_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29122_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29122_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29122_CommonData.yaml#/components/responses/404'
      '409':
        $ref: 'TS29122_CommonData.yaml#/components/responses/409'
      '411':
        $ref: 'TS29122_CommonData.yaml#/components/responses/411'
      '412':
        $ref: 'TS29122_CommonData.yaml#/components/responses/412'
      '413':
        $ref: 'TS29122_CommonData.yaml#/components/responses/413'
      '500':
        $ref: 'TS29122_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29122_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29122_CommonData.yaml#/components/responses/default'


  patch:
    summary:  Updates a existing delivery via MBMS for a given SCS/AS, a TMGI and transaction Id.
    tags:
      - Individual Delivery via MBMS resource Operation
    parameters:
      - name: scsAsId
        in: path
        description: Identifier of SCS/AS
        required: true
        schema:
          type: string
      - name: tmgi
        in: path
        description: TMGI
        required: true
        schema:
          type: string
      - name: transactionId
        in: path
        description: Identifier of transaction
        required: true
        schema:
          type: string
    requestBody:
      description: representation of the GMD via MBMS by MB2 resource to be udpated in the SCEF
      required: true
      content:
        application/merge-patch+json:
          schema:
            $ref: '#/components/schemas/GMDViaMBMSByMb2Patch'
    responses:
      '201':
        description: successful creation of an GMD via MBMS by MB2 resource
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/GMDViaMBMSByMb2'
      '400':
        $ref: 'TS29122_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29122_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29122_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29122_CommonData.yaml#/components/responses/404'
      '409':
        $ref: 'TS29122_CommonData.yaml#/components/responses/409'
      '411':
        $ref: 'TS29122_CommonData.yaml#/components/responses/411'
      '412':
        $ref: 'TS29122_CommonData.yaml#/components/responses/412'
      '413':
        $ref: 'TS29122_CommonData.yaml#/components/responses/413'
      '500':
        $ref: 'TS29122_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29122_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29122_CommonData.yaml#/components/responses/default'


  delete:
    summary:  deletes a delivery via MBMS resource for a given SCS/AS, a TMGI and a transcation
Id.
    tags:
      - Individual Delivery via MBMS resource Operation
    parameters:
      - name: scsAsId
        in: path
        description: Identifier of SCS/AS
        required: true
        schema:
          type: string
      - name: tmgi
        in: path
        description: TMGI
        required: true
        schema:
          type: string
      - name: transactionId
        in: path
        description: Identifier of transaction
        required: true
        schema:
          type: string
    responses:
      '200':
        description: successful deletion of an resouce of deliery via MBMS
      '400':
        $ref: 'TS29122_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29122_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29122_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29122_CommonData.yaml#/components/responses/404'
      '412':
        $ref: 'TS29122_CommonData.yaml#/components/responses/412'
      '500':
        $ref: 'TS29122_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29122_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
schemas:
  TMGIAllocation:
    type: object
    properties:
      self:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
      supportedFeatures:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      externalGroupId:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalGroupId'
      mbmsLocArea:
        $ref: '#/components/schemas/MbmsLocArea'
      tmgi:
        type: string
        description: Identifies a particular MBMS bearer service.
      tmgiExpiration:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
  GMDViaMBMSByMb2:
    type: object
    properties:
      self:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
      notificationDestination:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
      requestTestNotification:
        type: boolean
        description: Set to true by the SCS/AS to request the SCEF to send a test notification as
defined in subclause 5.2.5.3. Set to false or omitted otherwise.
      websockNotifConfig:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
      externalGroupId:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalGroupId'
      mbmsLocArea:
        $ref: '#/components/schemas/MbmsLocArea'
      messageDeliveryStartTime:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
      groupMessagePayload:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/Binary'
      scefMessageDeliveryIPv4:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv4Addr'
      scefMessageDeliveryIPv6:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv6Addr'
      scefMessageDeliveryPort:
        type: integer
        description: Indicates the port number where the SCEF wants to receive the data.
    required:
      - notificationDestination
  GMDByMb2Notification:
    type: object
    properties:
      transaction:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
      tltrId:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/TltrId'
      tmgi:
        type: string
        description: Identifies the TMGI.
      deliveryTriggerStatus:
        type: boolean
        description: Indicates whether delivery of group message payload corresponding to the TMGI
was successful or not
    required:
      - transaction
  TMGIAllocationPatch:
    type: object
    properties:
      externalGroupId:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalGroupId'
      mbmsLocArea:
        $ref: '#/components/schemas/MbmsLocArea'
  GMDViaMBMSByMb2Patch:
    type: object
    properties:
      externalGroupId:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalGroupId'
      mbmsLocArea:
        $ref: '#/components/schemas/MbmsLocArea'
      messageDeliveryStartTime:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
      groupMessagePayload:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/Binary'
  MbmsLocArea:
    type: object
    properties:
      cellId:
        type: array
        items:
          type: string
        minItems: 0
        description: Indicates a Cell Global Identification of the user which identifies the cell
the UE is registered.
      enodeBId:
        type: array
        items:
          type: string
        minItems: 0
        description: Indicates an eNodeB in which the UE is currently located.
      geographicArea:
        type: array
        items:
          $ref: 'TS29572_Nlmf_Location.yaml#/components/schemas/GeographicArea'
        minItems: 0
        description: Identifies a geographic area of the user where the UE is located.
      mbmsServiceAreaId:
        type: array
        items:
          type: string
        minItems: 0
        description: Identifies an MBMS Service Area Identity of the user where the UE is located.
      civicAddress:
        type: array
        items:
          $ref: 'TS29572_Nlmf_Location.yaml#/components/schemas/CivicAddress'
        minItems: 0
        description: Identifies a civic address of the user where the UE is located.
```

A.8.2 GMDviaMBMSbyxMB API

```
openapi: 3.0.0
info:
  title: GMDviaMBMSbyxMB
  description: API for Group Message Delivery via MBMS by xMB
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-group-message-delivery-xmb/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /{scsAsId}/services/:
    get:
      summary: read all service resources for a given SCS/AS
      tags:
        - Service Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (successful query of service creation resource)
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/ServiceCreation'
                minItems: 0
                description: The service resource for the SCS/AS in the request URI is returned.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    post:
      summary: creates a new service creation resource for a given SCS/AS
      tags:
        - Service Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
      requestBody:
        description: representation of the service to be created in the SCEF
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ServiceCreation'
      responses:
        '201':
          description: successful creation of a service
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ServiceCreation'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /{scsAsId}/services/{serviceId}:
    get:
      summary: read a service resource for a given SCS/AS and a Service Id
      tags:
        - Individual Service Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: serviceId
          in: path
          description: Service Id
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (successful query of service resource)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ServiceCreation'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    delete:
      summary: deletes an existing service resource for a given SCS/AS and a service id
      tags:
        - Individual service Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: serviceId
          in: path
          description: Service Id
          required: true
          schema:
            type: string
      responses:
        '200':
          description: successful deletion of a service resource
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /{scsAsId}/services/{serviceId}/delivery-via-mbms/:
    get:
      summary: read all group message delivery via MBMS resource for a given SCS/AS and a service id
      tags:
        - Delivery via MBMS Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: serviceId
          in: path
          description: Service Id
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (successful query of Delivery via MBMS resource)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/GMDViaMBMSByxMB'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    post:
      summary:  Creates a new delivery via MBMS for a given SCS/AS and a service Id
      tags:
        - Delivery via MBMS Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: serviceId
          in: path
          description: Service Id
          required: true
          schema:
            type: string
      requestBody:
        description: representation of the GMD via MBMS by xMB resource to be Created in the SCEF
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/GMDViaMBMSByxMB'
      callbacks:
        gMDByxMBNotification:
          '{$request.body#/notificationUri}':
            post:
              requestBody:  # contents of the callback message
                required: true
                content:
                  application/json:
                    schema:
                      $ref: '#/components/schemas/GMDByxMBNotification'
              responses:
                '200':
                  description: OK (The successful acknowledgement of the notification with a body)
                  content:
                    application/json:
                      schema:
                        $ref: 'TS29122_CommonData.yaml#/components/schemas/Acknowledgement'
                '204':
                  description: successful notification
                '400':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/400'
                '401':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/401'
                '403':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/403'
                '409':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/409'
                '411':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/411'
                '412':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/412'
                '413':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/413'
                '500':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/500'
                '503':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/503'
                default:
                  $ref: 'TS29122_CommonData.yaml#/components/responses/default'
      responses:
        '201':
          description: successful creation of an GMD via MBMS by xMB resource
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/GMDViaMBMSByxMB'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /{scsAsId}/services/{serviceId}/delivery-via-mbms/{transactionId}:
    get:
      summary: read all group message delivery via MBMS resource for a given SCS/AS and a service Id
      tags:
        - Individual Delivery via MBMS resource Operation
      parameters:
        - name: scsAsId
```

```
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: serviceId
          in: path
          description: Service Id
          required: true
          schema:
            type: string
        - name: transactionId
          in: path
          description: Identifier of transaction
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (successful query of an Delivery via MBMS resource)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/GMDViaMBMSByxMB'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    put:
      summary:  Updates an existing delivery via MBMS for a given SCS/AS, a service Id and 
transaction Id.
      tags:
        - Individual Delivery via MBMS resource Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: serviceId
          in: path
          description: Service Id
          required: true
          schema:
            type: string
        - name: transactionId
          in: path
          description: Identifier of transaction
          required: true
          schema:
            type: string
      requestBody:
        description: representation of the GMD via MBMS by xMB resource to be udpated in the SCEF
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/GMDViaMBMSByxMB'
      responses:
        '201':
          description: successful creation of an GMD via MBMS by xMB resource
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/GMDViaMBMSByxMB'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    patch:
      summary:  Updates an existing delivery via MBMS for a given SCS/AS, a service Id and 
transaction Id.
      tags:
        - Individual Delivery via MBMS resource Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: serviceId
          in: path
          description: Service Id
          required: true
          schema:
            type: string
        - name: transactionId
          in: path
          description: Identifier of transaction
          required: true
          schema:
            type: string
      requestBody:
        description: representation of the GMD via MBMS by xMB resource to be udpated in the SCEF
        required: true
        content:
          application/merge-patch+json:
            schema:
              $ref: '#/components/schemas/GMDViaMBMSByxMBPatch'
      responses:
        '201':
          description: successful creation of an GMD via MBMS by xMB resource
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/GMDViaMBMSByxMB'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    delete:
      summary:  deletes a delivery via MBMS resource for a given SCS/AS, a service Id and a 
transcation Id.
      tags:
        - Individual Delivery via MBMS resource Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of SCS/AS
          required: true
          schema:
            type: string
        - name: serviceId
          in: path
          description: Service Id
          required: true
          schema:
            type: string
        - name: transactionId
          in: path
          description: Identifier of transaction
          required: true
          schema:
            type: string
      responses:
        '200':
          description: successful deletion of an resouce of deliery via MBMS
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

components:
  schemas:
    ServiceCreation:
      type: object
      properties:
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        externalGroupId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalGroupId'
        userServiceId:
          type: string
          description: Identifies the MBMS User Service supplied by the SCEF.
        serviceClass:
          type: string
          description: The service class that service belongs to supplied by the SCEF.
        serviceLanguages:
          type: array
          items:
            type: string
          minItems: 0
          description: List of language of the service content supplied by the SCEF.
        serviceNames:
          type: array
          items:
            type: string
          minItems: 0
          description: List of Service Names supplied by the SCEF.
        receiveOnlyMode:
          type: boolean
          description: When set to 'true', the Content Provider indicates that the service is a 
Receive Only Mode service. This parameter is supplied by the SCEF.
        serviceAnnouncementMode:
          type: string
          description: Enumeration of Service Announcement Mode supplied by the SCEF.
    GMDViaMBMSByxMB:
      type: object
      properties:
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        notificationDestination:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        requestTestNotification:
          type: boolean
          description: Set to true by the SCS/AS to request the SCEF to send a test notification as 
defined in subclause 5.2.5.3. Set to false or omitted otherwise.
        websockNotifConfig:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
        mbmsLocArea:
          $ref: '#/components/schemas/MbmsLocArea'
        messageDeliveryStartTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        messageDeliveryStopTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        groupMessagePayload:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Binary'
        scefMessageDeliveryIPv4:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv4Addr'
        scefMessageDeliveryIPv6:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv6Addr'
        scefMessageDeliveryPort:
          type: integer
          description: Indicates the port number where the SCEF wants to receive the data.
      required:
        - scsAsId
        - notificationDestination
    GMDByxMBNotification:
      type: object
      properties:
        transaction:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        deliveryTriggerStatus:
          type: boolean
          description: Indicates whether delivery of group message payload was successful or not
      required:
        - transaction
    GMDViaMBMSByxMBPatch:
      type: object
      properties:
        mbmsLocArea:
          $ref: '#/components/schemas/MbmsLocArea'
        messageDeliveryStartTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        messageDeliveryStopTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        groupMessagePayload:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Binary'
    MbmsLocArea:
      type: object
      properties:
        cellId:
          type: array
          items:
            type: string
          minItems: 0
          description: Indicates a Cell Global Identification of the user which identifies the cell 
the UE is registered.
        enodeBId:
          type: array
          items:
            type: string
          minItems: 0
          description: Indicates an eNodeB in which the UE is currently located.
        geographicArea:
          type: array
          items:
            $ref: 'TS29572_Nlmf_Location.yaml#/components/schemas/GeographicArea'
          minItems: 0
          description: Identifies a geographic area of the user where the UE is located.
        mbmsServiceAreaId:
          type: array
          items:
            type: string
          minItems: 0
          description: Identifies an MBMS Service Area Identity of the user where the UE is located.
        civicAddress:
          type: array
          items:
            $ref: 'TS29572_Nlmf_Location.yaml#/components/schemas/CivicAddress'
          minItems: 0
          description: Identifies a civic address of the user where the UE is located.
```

A.9 ReportingNetworkStatus API

```
openapi: 3.0.0
info:
  title: 3gpp-network-status-reporting
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-net-stat-report/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /{scsAsId}/subscriptions:
    parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/ScsASId'
    get:
      summary: Read all network status reporting subscription resources for a given SCS/AS.
      responses:
        '200':
          description: The requested information was returned successfully.
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/NetworkStatusReportingSubscription'
                minItems: 0
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

    post:
      summary: Create a new network status reporting subscription resource.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/NetworkStatusReportingSubscription'
      callbacks:
        notificationDestination:
          '{request.body#/notificationDestination}':
            post:
              requestBody:  # contents of the callback message
                required: true
                content:
                  application/json:
                    schema:
                      $ref: '#/components/schemas/NetworkStatusReportingNotification'
              responses:
                '204':
                  description: No Content (successful notification)
                '400':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/400'
                '401':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/401'
                '403':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/403'
                '409':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/409'
                '411':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/411'
                '412':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/412'
                '413':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/413'
                '500':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/500'
                '503':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/503'
                default:
                  $ref: 'TS29122_CommonData.yaml#/components/responses/default'
      responses:
        '201':
          description: The subscription was created successfully. The URI of the created resource
shall be returned in the "Location" HTTP header.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NetworkStatusReportingSubscription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /{scsAsId}/subscriptions/{subscriptionId}:
    parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/ScsASId'
        - name: subscriptionId
          in: path
          description: Identifier of the subscription resource of type string
          required: true
          schema:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/subscriptionId'
    get:
      summary: Read an active network status reporting subscription resource.
      responses:
        '200':
          description: The requested information was returned successfully.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NetworkStatusReportingSubscription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    put:
      summary: Modify an existing subscription resource to update a subscription.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/NetworkStatusReportingSubscription'
      responses:
        '200':
          description: The subscription was updated successfully.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NetworkStatusReportingSubscription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    delete:
      summary: Delete an existing continuous network status reporting subscription resource.
      responses:
        '204':
          description: The subscription was updated successfully.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    NetworkStatusReportingSubscription:
      type: object
      properties:
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        notificationDestination:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        requestTestNotification:
          type: boolean
          description: Set to true by the SCS/AS to request the SCEF to send a test notification as
defined in subclause 5.2.5.3. Set to false or omitted otherwise.
        websockNotifConfig:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
        locationArea:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/LocationArea'
        timeDuration:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        thresholdValues:
          type: array
          items:
            $ref: '#/components/schemas/CongestionValue'
          minItems: 0
          description: Identifies a list of congestion level(s) with exact value that the SCS/AS
requests to be informed of when reached.
        thresholdTypes:
          type: array
          items:
            $ref: '#/components/schemas/CongestionType'
          minItems: 0
          description: Identifies a list of congestion level(s) with abstracted value that the
SCS/AS requests to be informed of when reached.
      required:
        - notificationDestination
        - locationArea
    NetworkStatusReportingNotification:
      type: object
      properties:
        subscription:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        nsiValue:
          $ref: '#/components/schemas/CongestionValue'
        nsiType:
          $ref: '#/components/schemas/CongestionType'
      required:
        - subscription
    CongestionValue:
      type: integer
      description: Unsigned integer with valid values between 0 and 31. The value 0 indicates that
there is no congestion. The value 1 is the lowest congestion level and value 31 is the highest
congestion level.
    CongestionType:
      anyOf:
      - type: string
        enum:
          - HIGH
          - MEDIUM
          - LOW
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - HIGH: The congestion status is high.
        - MEDIUM: The congestion status is medium.
        - LOW: The congestion status is low.
```

A.10 CpProvisioning API

```
openapi: 3.0.0
info:
  title: 3gpp-cp-parameter-provisioning
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-cp-parameter-provisioning/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /{scsAsId}/subscriptions:
    get:
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS as defined in subclause subclause 5.2.4 of 3GPP TS 
29.122.
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK. The subscription information related to the request URI is returned.
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/CpInfo'
                minItems: 0
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    post:
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS as defined in subclause subclause 5.2.4 of 3GPP TS 
29.122.
          required: true
          schema:
            type: string
      requestBody:
        description: create new subscriptions for a given SCS/AS and the provisioning CP parameter 
sets.
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/CpInfo'
      responses:
        '201':
          description: Created. The subscription was created successfully. The SCEF shall return the 
created subscription in the response payload body.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/CpInfo'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
  /{scsAsId}/subscriptions/{subscriptionId}:
    get:
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS as defined in subclause subclause 5.2.4 of 3GPP TS
29.122.
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Subscription ID
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK. The subscription information related to the request URI is returned.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/CpInfo
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    put:
      requestBody:
        description: Modify a CP parameter provisioning subscription resource.
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/CpInfo'
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS as defined in subclause subclause 5.2.4 of 3GPP TS
29.122.
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Subscription ID
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK. The subscription was modified successfully. The SCEF shall return an
updated subscription in the response payload body.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/CpInfo'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    delete:
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS as defined in subclause subclause 5.2.4 of 3GPP TS
29.122.
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Subscription ID
          required: true
          schema:
            type: string
      responses:
        '204':
          description: No Content. The subscription was deleted successfully. The payload body shall
be empty.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
  /{scsAsId}/subscriptions/{subscriptionId}/cpSets/{setId}:
    get:
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS as defined in subclause subclause 5.2.4 of 3GPP TS
29.122.
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Subscription ID
          required: true
          schema:
            type: string
        - name: setId
          in: path
          description: Identifier of the CP parameter set
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK. The subscription information related to the request URI is returned.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/CpParameterSet
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    put:
      requestBody:
        description: Change information for a CP parameter set.
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/CpParameterSet'
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS as defined in subclause subclause 5.2.4 of 3GPP TS
29.122.
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Subscription ID
          required: true
          schema:
            type: string
        - name: setId
          in: path
          description: Identifier of the CP parameter set
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK. The CP parameter set resource was modified successfully. The SCEF shall
return an updated CP parameter set resource in the response payload body.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/CpParameterSet'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    delete:
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS as defined in subclause subclause 5.2.4 of 3GPP TS
29.122.
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Subscription ID
          required: true
          schema:
            type: string
        - name: setId
          in: path
          description: Identifier of the CP parameter set
          required: true
          schema:
            type: string
      responses:
        '204':
          description: No Content. The subscription was deleted successfully. The payload body shall
be empty.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    CpInfo:
      type: object
      properties:
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        externalId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalId'
        msisdn:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Msisdn'
        externalGroupId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalGroupId'
        cpParameterSet:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/CpParameterSet'
          minProperties: 1
          description: Identifies a set of CP parameter information that may be part of this CpInfo
structure.
      required:
        - cpParameterSet
    CpParameterSet:
      type: object
      properties:
        setId:
          type: string
          description: SCS/AS-chosen correlator provided by the SCS/AS in the request to create a 
resource fo CP parameter set(s).
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        validityTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        periodicCommunicationIndicator:
          $ref: '#/components/schemas/CommunicationIndicator'
        communicationDurationTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        periodicTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        scheduledCommunicationTime:
          $ref: '#/components/schemas/ScheduledCommunicationTime'
        stationaryIndication:
          $ref: '#/components/schemas/StationaryIndication'
        expectedUmts:
          type: array
          items:
            $ref: ' TS29122_CommonData.yaml#/components/schemas/LocationArea'
          minItems: 0
          description: Identifies the UE's expected geographical movement. The attribute is only 
applicable in 5G.
      required:
        - setId
    ScheduledCommunicationTime:
      type: object
      properties:
        daysOfWeek:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/DayOfWeek'
          minItems: 0
          maxItems: 6
          description: Identifies the day(s) of the week. If absent, it indicates every day of the 
week.
        timeOfDayStart:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/TimeOfDay'
        timeOfDayEnd:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/TimeOfDay'
    CommunicationIndicator:
      anyOf:
      - type: string
        enum:
          - PERIODICALLY
          - ON_DEMAND
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - PERIODICALLY: Identifies the UE communicates periodically
        - ON_DEMAND: Identifies the UE communicates on demand
    StationaryIndication:
      anyOf:
      - type: string
        enum:
          - STATIONARY
          - MOBILE
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - STATIONARY: Identifies the UE is stationary
        - MOBILE: Identifies the UE is mobile
```

A.11 PfdManagement API

```
openapi: 3.0.0
info:
  title: 3gpp-pfd-management
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-pfd-management/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /{scsAsId}/transactions:
    parameters:
      - name: scsAsId
        in: path
        description: Identifier of the SCS/AS as defined in subclause subclause 5.2.4 of 3GPP TS 
29.122.
        required: true
        schema:
          type: string
    get:
      responses:
        '200':
          description: OK. All transactions related to the request URI are returned.
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/PfdManagement'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    post:
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/PfdManagement'
        description: Create a new transaction for PFD management.
      responses:
        '201':
          description: Created. The transaction was created successfully. The SCEF shall return the 
created transaction in the response payload body. PfdReport may be included to provide detailed 
failure information for some applications.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdManagement'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          description: The PFDs for all applications were not created successfully. PfdReport is 
included with detailed information.
```

```
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/PfdReport'
                minItems: 1
            application/problem+json:
              schema:
                $ref: 'TS29122_CommonData.yaml#/components/schemas/ProblemDetails'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
  /{scsAsId}/transactions/{transactionId}:
    parameters:
      - name: scsAsId
        in: path
        description: Identifier of the SCS/AS as defined in subclause subclause 5.2.4 of 3GPP TS
29.122.
        required: true
        schema:
          type: string
      - name: transactionId
        in: path
        description: Transaction ID
        required: true
        schema:
          type: string
    get:
      responses:
        '200':
          description: OK. The transaction information related to the request URI is returned.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdManagement'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    put:
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/PfdManagement'
        description: Change information in PFD management transaction.
      responses:
        '200':
          description: OK. The transaction was modified successfully. The SCEF shall return an
updated transaction in the response payload body.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdManagement'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          description: The PFDs for all applications were not updated successfully. PfdReport is
included with detailed information.
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/PfdReport'
                minItems: 1
            application/problem+json:
              schema:
                $ref: 'TS29122_CommonData.yaml#/components/schemas/ProblemDetails'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    delete:
      responses:
        '204':
          description: No Content. The transaction was deleted successfully. The payload body shall
be empty.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
  /{scsAsId}/transactions/{transactionId}/applications/{appId}:
    parameters:
      - name: scsAsId
        in: path
        description: Identifier of the SCS/AS as defined in subclause subclause 5.2.4 of 3GPP TS
29.122.
        required: true
        schema:
          type: string
      - name: transactionId
        in: path
        description: Transaction ID
        required: true
        schema:
          type: string
      - name: appId
        in: path
        description: Identifier of the application
        required: true
        schema:
          type: string
    get:
      responses:
        '200':
          description: OK. The application information related to the request URI is returned.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdData'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    put:
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/PfdData'
        description: Change information in application.
      responses:
        '200':
          description: OK. The application resource was modified successfully. The SCEF shall return 
an updated application resource in the response payload body.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdData'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          description: The PFDs for the application were not updated successfully.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdReport'
            application/problem+json:
              schema:
                $ref: 'TS29122_CommonData.yaml#/components/schemas/ProblemDetails'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          description: The PFDs for the application were not updated successfully.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdReport'
            application/problem+json:
              schema:
                $ref: 'TS29122_CommonData.yaml#/components/schemas/ProblemDetails'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          description: The PFDs for the application were not updated successfully.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdReport'
            application/problem+json:
              schema:
                $ref: 'TS29122_CommonData.yaml#/components/schemas/ProblemDetails'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    patch:
      requestBody:
        required: true
        content:
          application/merge-patch+json:
            schema:
              $ref: '#/components/schemas/PfdData'
        description: Change information in PFD management transaction.
      responses:
        '200':
          description: OK. The transaction was modified successfully. The SCEF shall return an 
updated transaction in the response payload body.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdData'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          description: The PFDs for the application were not updated successfully.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdReport'
            application/problem+json:
              schema:
                $ref: 'TS29122_CommonData.yaml#/components/schemas/ProblemDetails'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          description: The PFDs for the application were not updated successfully.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdReport'
            application/problem+json:
              schema:
                $ref: 'TS29122_CommonData.yaml#/components/schemas/ProblemDetails'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          description: The PFDs for the application were not updated successfully.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PfdReport'
            application/problem+json:
              schema:
                $ref: 'TS29122_CommonData.yaml#/components/schemas/ProblemDetails'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    delete:
      responses:
        '204':
          description: No Content. The application was deleted successfully. The payload body shall 
be empty.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    PfdManagement:
      type: object
      properties:
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        pfdDatas:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/PfdData'
          minProperties: 0
          description: Each element uniquely identifies the PFDs for an external application 
identifier. Each element is identified in the map via an external application identifier as key. The 
response shall include successfully provisioned PFD data of application(s).
        pfdReports:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/PfdReport'
          minProperties: 0
          description: Supplied by the SCEF and contains the external application identifiers for 
which PFD(s) are not added or modified successfully. The failure reason is also included. Each 
element provides the related information for one or more external application identifier(s) and is 
identified in the map via the failure identifier as key.
          readOnly: true
    PfdData:
      type: object
      properties:
        externalAppId:
          type: string
          description: Each element uniquely external application identifier
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        pfds:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/Pfd'
          description: Contains the PFDs of the external application identifier. Each PFD is 
identified in the map via a key containing the PFD identifier.
        allowedDelay:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSecRm'
        cachingTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSecRo'
      required:
        - externalAppId
        - pfds
    Pfd:
      type: object
      properties:
        pfdId:
          type: string
          description: Identifies a PDF of an application identifier.
        flowDescriptions:
          type: array
          items:
            type: string
          minItems: 0
          description: Represents a 3-tuple with protocol, server ip and server port for UL/DL 
application traffic. The content of the string has the same encoding as the IPFilterRule AVP value 
as defined in IETF RFC 6733 [46].
        urls:
          type: array
          items:
            type: string
          minItems: 0
          description: Indicates a URL or a regular expression which is used to match the 
significant parts of the URL.
        domainNames:
          type: array
          items:
            type: string
          minItems: 0
          description: Indicates an FQDN or a regular expression as a domain name matching criteria.
      required:
        - pfdId
    PfdReport:
      type: object
      properties:
        externalAppId:
          type: array
          items:
            type: string
          minItems: 1
          description: Identifies the external application identifier(s) which PFD(s) are not added 
or modified successfully
        failureCode:
          $ref: '#/components/schemas/FailureCode'
        cachingTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
      required:
        - externalAppId
        - failureCode
    FailureCode:
      anyOf:
      - type: string
        enum:
          - MALFUNCTION
          - RESOURCE_LIMITATION
          - SHORT_DELAY
          - APP_ID_DUPLICATED
          - OTHER_REASON
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - MALFUNCTION: This value indicates that something functions wrongly in PFD provisioning or 
the PFD provisioning does not function at all.
        - RESOURCE_LIMITATION: This value indicates there is resource limitation for PFD storage.
        - SHORT_DELAY: This value indicates that the allowed delay is too short and PFD(s) are not 
stored.
        - APP_ID_DUPLICATED: The received external application identifier(s) are already 
provisioned.
        - OTHER_REASON: Other reason unspecified.
```

A.12 ECRControl API

```
openapi: 3.0.0
info:
  title: 3gpp-ecr-control
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-ecr-control/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /query:
    post:
      summary: Query the status of enhanced converage restriction for a UE.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ECRControl'
      responses:
        '200':
          description: The requested information was returned successfully.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ECRData'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /configure:
    post:
      summary: Configure the enhanced converage restriction for a UE.
      requestBody:
         required: true
         content:
          application/json:
            schema:
              $ref: '#/components/schemas/ECRControl'
      responses:
        '204':
          description: The Enhanced Coverage Restriction setting was configured successfully. The 
response body shall be empty.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    ECRControl:
      type: object
      properties:
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        externalId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalId'
        msisdn:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Msisdn'
        restrictedPlmnIds:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/PlmnId'
          minItems: 0
          description: Indicates a complete list (and possibly empty) of serving PLMNs where 
Enhanced Coverage shall be restricted. This attribute shall not be present for the query custom 
operation.
        allowedPlmnIds:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/PlmnId'
          minItems: 0
          description: Indicates a complete list (and possibly empty) of serving PLMNs where 
Enhanced Coverage shall be allowed. This attribute shall not be present for the query custom 
operation.
      required:
        - supportedFeatures
    ECRData:
      type: object
      properties:
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        visitedPlmnId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/PlmnId'
        restrictedPlmnIds:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/PlmnId'
          minItems: 0
          description: Indicates a complete list (and possibly empty) of serving PLMNs where
Enhanced Coverage shall be restricted.
        allowedPlmnIds:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/PlmnId'
          minItems: 0
          description: Indicates a complete list (and possibly empty) of serving PLMNs where
Enhanced Coverage shall be allowed.
      required:
        - supportedFeatures
```

A.13 NpConfiguration API

```
openapi: 3.0.0
info:
  title: 3gpp-network-parameter-configuration
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-network-parameter-configuration/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause of 3GPP TS 29.122.
paths:
  /{scsAsId}/configurations:
    get:
      summary: read all of the active configurations for the SCS/AS
      tags:
        - NpConfiguration API SCS/AS level GET Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (Successful get all of the active NpConfigurations for the SCS/AS)
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/NpConfiguration'
                minItems: 0
                description: Network Parameter configurations
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    post:
      summary: Creates a new configuration resource for network parameter configuration
      tags:
        - NpConfiguration API Configuration level POST Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
      requestBody:
        description: new configuration creation
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/NpConfiguration'
      callbacks:
        notificationDestination:
          '{request.body#/notificationDestination}':
            post:
              requestBody:  # contents of the callback message
                required: true
                content:
                  application/json:
                    schema:
                      $ref: '#/components/schemas/ConfigurationNotification'
              responses:
                '204':
                  description: No Content (successful notification)
                '400':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/400'
                '401':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/401'
                '403':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/403'
                '409':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/409'
                '411':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/411'
                '412':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/412'
                '413':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/413'
                '500':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/500'
                '503':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/503'
                default:
                  $ref: 'TS29122_CommonData.yaml#/components/responses/default'

      responses:
        '201':
          description: Created (Successful creation of configuration)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NpConfiguration'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /{scsAsId}/configurations/{configurationId}:
    get:
      summary: read an active configuration for the SCS/AS and the configuration Id
      tags:
        - NpConfiguration API Configuration level GET Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
        - name: configurationId
          in: path
          description: Identifier of the configuration resource
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (Successful get the active configuration)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NpConfiguration'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    put:
      summary: Updates/replaces an existing configuration resource
      tags:
        - NpConfiguration API Configuration level PUT Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
        - name: configurationId
          in: path
          description: Identifier of the configuration resource
          required: true
          schema:
            type: string
      requestBody:
        description: Parameters to update/replace the existing configuration
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/NpConfiguration'
      responses:
        '200':
          description: OK (Successful deletion of the existing configuration)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NpConfiguration'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    patch:
      summary: Updates/replaces an existing configuration resource
      tags:
        - NpConfiguration API Configuration level PATCH Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
        - name: configurationId
          in: path
          description: Identifier of the configuration resource
          required: true
          schema:
            type: string
      requestBody:
        required: true
        content:
          application/merge-patch+json:
            schema:
              $ref: '#/components/schemas/NpConfigurationPatch'
      responses:
        '200':
          description: OK. The configuration was modified successfully.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NpConfiguration'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    delete:
      summary: Deletes an already existing configuration
      tags:
        - NpConfiguration API Configuration level DELETE Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
        - name: configurationId
          in: path
          description: Identifier of the configuration resource
          required: true
          schema:
            type: string
      responses:
        '204':
          description: No Content (Successful deletion of the existing configuration)
        '200':
          description: OK. (Successful deletion of the existing configuration)
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: 'TS29122_CommonData.yaml#/components/schemas/ConfigResult'
                minItems: 1
                description: The configuration was terminated successfully, the configuration
failure information for group members shall be included if received.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    NpConfiguration:
      type: object
      properties:
        self:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        externalId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalId'
        msisdn:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Msisdn'
        externalGroupId:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/ExternalGroupId'
        maximumLatency:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        maximumResponseTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
        suggestedNumberOfDlPackets:
          type: integer
          description: This parameter may be included to identify the number of packets that the
serving gateway shall buffer in case that the UE is not reachable.
        groupReportingGuardTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSec'
    NpConfigurationPatch:
      type: object
      properties:
        maximumLatency:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSecRm'
        maximumResponseTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSecRm'
        suggestedNumberOfDlPackets:
          type: integer
          description: This parameter may be included to identify the number of packets that the
serving gateway shall buffer in case that the UE is not reachable.
          nullable: true
        groupReportGuardTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DurationSecRm'
        notificationDestination:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        requestTestNotification:
          type: boolean
          description: Set to true by the SCS/AS to request the SCEF to send a test notification as 
defined in subclause 5.2.5.3. Set to false or omitted otherwise.
        websockNotifConfig:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
    ConfigurationNotification:
      type: object
      properties:
        configuration:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        configResults:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/ConfigResult'
          minItems: 1
          description: The grouping configuration result notification provided by the SCEF.
      required:
        - configuration
```

A.14 AsSessionWithQoS API

```
openapi: 3.0.0
info:
  title: 3gpp-as-session-with-qos
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}/3gpp-as-session-with-qos/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /{scsAsId}/subscriptions:
    get:
      summary: read all of the active subscriptions for the SCS/AS
      tags:
        - AsSessionWithQoS API SCS/AS level GET Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK.
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/AsSessionWithQoSSubscription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

    post:
      summary: Creates a new subscription resource
      tags:
        - AsSessionWithQoS API Subscription level POST Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
      requestBody:
        description: Request to create a new subscription resource
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/AsSessionWithQoSSubscription'
      callbacks:
        notificationDestination:
          '{request.body#/notificationDestination}':
            post:
              requestBody:  # contents of the callback message
                required: true
                content:
                  application/json:
                    schema:
                      $ref: 'TS29122_CommonData.yaml#/components/schemas//notificationData'
              responses:
                '204':
                  description: No Content (successful notification)
                '400':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/400'
                '401':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/401'
                '403':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/403'
                '409':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/409'
                '411':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/411'
                '412':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/412'
                '413':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/413'
                '500':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/500'
                '503':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/503'
                default:
                  $ref: 'TS29122_CommonData.yaml#/components/responses/default'
      responses:
        '201':
          description: Created (Successful creation of subscription)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/AsSessionWithQoSSubscription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /{scsAsId}/subscriptions/{subscriptionId}:
    get:
      summary: read an active subscriptions for the SCS/AS and the subscription Id
      tags:
        - AsSessionWithQoS API Subscription level GET Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Identifier of the subscription resource
          required: true
          schema:
            type: string
      responses:
        '200':
          description: OK (Successful get the active subscription)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/AsSessionWithQoSSubscription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    put:
      summary: Updates/replaces an existing subscription resource
      tags:
        - AsSessionWithQoS API subscription level PUT Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Identifier of the subscription resource
          required: true
          schema:
            type: string
      requestBody:
        description: Parameters to update/replace the existing subscription
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/AsSessionWithQoSSubscription'
      responses:
        '200':
          description: OK (Successful update of the subscription)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/AsSessionWithQoSSubscription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    patch:
      summary: Updates/replaces an existing subscription resource
      tags:
        - AsSessionWithQoS API subscription level PATCH Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Identifier of the subscription resource
          required: true
          schema:
            type: string
      requestBody:
        required: true
        content:
          application/merge-patch+json:
            schema:
              $ref: '#/components/schemas/AsSessionWithQoSSubscriptionPatch'
      responses:
        '200':
          description: OK. The subscription was modified successfully.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/AsSessionWithQoSSubscription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'


    delete:
      summary: Deletes an already existing subscription
      tags:
        - AsSessionWithQoS API Subscription level DELETE Operation
      parameters:
        - name: scsAsId
          in: path
          description: Identifier of the SCS/AS
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Identifier of the subscription resource
          required: true
          schema:
            type: string
      responses:
        '204':
          description: No Content (Successful deletion of the existing subscription)
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    AsSessionWithQoSSubscription:
      type: object
      properties:
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        notificationDestination:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Link'
        flowInfo:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/FlowInfo'
          minItems: 1
          description: Describe the data flow which requires QoS.
        qosReference:
          type: string
          description: Identifies a pre-defined QoS information
        ueIpv4Addr:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv4Addr'
        ueIpv6Addr:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv6Addr'
        usageThreshold:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/UsageThreshold'
        sponsorInfo:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/SponsorInformation'
        requestTestNotification:
          type: boolean
          description: Set to true by the SCS/AS to request the SCEF to send a test notification as
defined in subclause 5.2.5.3. Set to false or omitted otherwise.
        websockNotifConfig:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
      required:
        - notificationDestination
        - flowInfo
    AsSessionWithQoSSubscriptionPatch:
      type: object
      properties:
        flowInfos:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/FlowInfos'
          minItems: 0
          description: Describe the data flow which requires QoS.
        qosReference:
          type: string
          description: Pre-defined QoS reference
        usageThreshold:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/UsageThresholdRm'
```

A.15 MsisdnLessMoSms API

```
openapi: 3.0.0
info:
  title: 3gpp-msisdn-less-mo-sms
  version: "1.PreR15.1.0"
servers:
  - url: '{apiRoot}'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 5.2.4 of 3GPP TS 29.122.
paths:
  /:
    post:
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/MsisdnLessMoSmsNotification'
      responses:
        '200':
          description: Success
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/MsisdnLessMoSmsNotificationReply'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

components:
  schemas:
    MsisdnLessMoSmsNotification:
      type: object
      properties:
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        sms:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Bytes'
        externalId:
          type: string
          description: External identifier has the form username@realm.
        ApplicationPort:
          type: integer
          description: Unsigned integer used to uniquely identify the triggering application
addressed in the device, see subclause 9.2.3.24.4 in TS 23.040 [43] for further details.
      required:
        - supportedFeatures
        - sms
        - externalId
        - ApplicationPort
    MsisdnLessMoSmsNotificationReply:
      type: object
      properties:
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      required:
        - supportedFeatures
```

Annex B (informative): Change history

Change history							
Date	**Meeting**	**TDoc.**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New**
2017-04	CT3#89					TS skeleton of T8 reference point stage 3. Inclusion of C3-172217, C3-172242, C3-172249 and editorial change from Rapporteur.	0.0.0
2017-05	CT3#90					Inclusion of C3-173256, C3-173257, C3-173306, C3-173318, C3-173323, C3-173329 and editorial change from Rapporteur.	0.1.0
2017-09	CT3#91					Inclusion of C3-174368, C3-174279, C3-174079, C3-174352, C3-174328, C3-174329, C3-174374, C3-174375, C3-174243, C3-174333, C3-174334, C3-174376, C3-174281, C3-174377, C3-174356, C3-174280, C3-174358, C3-174359, C3-174379, C3-174337, C3-174361, C3-174294, C3-174362, C3-174385, C3-174299 and editorial change from Rapporteur.	0.2.0
2017-10	CT3#92					Inclusion of C3-175068, C3-175365, C3-175370, C3-175301, C3-175227, C3-175228, C3-175359, C3-175230, C3-175231, C3-175337, C3-175247, C3-175290, C3-175360 and editorial changes from Rapporteur.	0.3.0
2017-12	CT3#93					Inclusion of C3-176329, C3-176218, C3-176356, C3-176375, C3-176357, C3-176334, C3-176378, C3-176067, C3-176335, C3-176073, C3-176337, C3-176380, C3-176390, C3-176362, C3-176339, C3-176340, C3-176233, C3-176089, C3-176267, C3-176383, C3-176365, C3-176366, C3-176272, C3-176273, C3-176274, C3-176391, C3-176368, C3-176369, C3-176385, C3-176279 and editorial changes from Rapporteur.	0.4.0
2018-01	CT3#94					Inclusion of C3-180349, C3-180329, C3-180222, C3-180078, C3-180285, C3-180081, C3-180330, C3-180331, C3-180084, C3-180332, C3-180333, C3-180290, C3-180291, C3-180089, C3-180234, C3-180334, C3-180294, C3-180237, C3-180295, C3-180296, C3-180297, C3-180337, C3-180357, C3-180136, C3-180298 and editorial changes from Rapporteur.	0.5.0
2018-02	CT3#95					Inclusion of C3-181271, C3-181324, C3-181340, C3-181273, C3-181333, C3-181334, C3-181276, C3-181325 and editorial changes from Rapporteur.	0.6.0
2018-03	CT#79	CP-181053				TS sent to plenary for information	1.0.0
2018-04	CT3#96					Inclusion of C3-182464, C3-182017, C3-182444, C3-182019, C3-182486, C3-182487, C3-182344, C3-182074, C3-182373, C3-182446, C3-182291, C3-182292, C3-182488, C3-182508, C3-182345, C3-182094, C3-182295, C3-182346, C3-182096, C3-182347, C3-182348, C3-182026, C3-182349, C3-182098, C3-182350, C3-182099, C3-182100, C3-182209, C3-182210, C3-182101, C3-182102, C3-182455, C3-182506, C3-182353, C3-182491, C3-182355, C3-182447, C3-182507, C3-182359, C3-182106, C3-182107, C3-182108, C3-182492, C3-182109, C3-182110, C3-182480, C3-182362, C3-182448, C3-182449, C3-182450, C3-182425, C3-182365, C3-182112, C3-182113, C3-182035, C3-182367, C3-182115, C3-182368, C3-182493, C3-182494, C3-182509, C3-182372 and editorial changes from Rapporteur.	1.1.0

2018-05	CT3#97					Inclusion of C3-183028, C3-183029, C3-183030, C3-183474, C3-183051, C3-183475, C3-183476, C3-183054, C3-183055, C3-183056, C3-183478, C3-183479, C3-183336, C3-183337, C3-183832, C3-183340, C3-183609, C3-183611, C3-183612, C3-183833, C3-183614, C3-183834, C3-183883, C3-183616, C3-183343, C3-183088, C3-183617, C3-183618, C3-183741, C3-183625, C3-183626, C3-183122, C3-183123, C3-183124, C3-183635, C3-183792, C3-183634, C3-183619, C3-183786, C3-183082, C3-183083, C3-183084, C3-183622, C3-183835, C3-183624, C3-183627, C3-183075, C3-183737, C3-183836, C3-183419, C3-183630, C3-183066, C3-183347, C3-183861, C3-183788, C3-183862, C3-183789, C3-183637, C3-183837, C3-183863, C3-183639, C3-183864, C3-183067, C3-183068, C3-183069, C3-183641, C3-183059, C3-183071, C3-183606, C3-183779 and editorial changes from Rapporteur.	1.2.0
2018-06	CT#80	CP-181038				TS sent to plenary for approval	2.0.0
2018-06	CT#80	CP-181038				TS approved by plenary	15.0.0
2018-09	CT#81	CP-182022	0001	2	F	MonitoringEvent API OpenAPI schema	15.1.0
2018-09	CT#81	CP-182022	0002	2	F	ChargeableParty API OpenAPI schema	15.1.0
2018-09	CT#81	CP-182022	0003	2	F	GMDviaMBMSbyMB2 API OpenAPI schema	15.1.0
2018-09	CT#81	CP-182022	0004	2	F	GMDviaMBMSbyxMB API OpenAPI schema	15.1.0
2018-09	CT#81	CP-182022	0005	2	F	ReportingNetworkStatus API OpenAPI schema	15.1.0
2018-09	CT#81	CP-182022	0006	2	F	ECRControl API OpenAPI schema	15.1.0
2018-09	CT#81	CP-182022	0007	2	F	NpConfiguration API OpenAPI schema	15.1.0
2018-09	CT#81	CP-182022	0008	2	F	AsSessionWithQoS API OpenAPI schema	15.1.0
2018-09	CT#81	CP-182022	0009		F	Update the Error Handling	15.1.0
2018-09	CT#81	CP-182022	0010	2	F	CpProvisioning OpenAPI schema	15.1.0
2018-09	CT#81	CP-182022	0011	2	F	Remove the monitoring configuration for a group	15.1.0
2018-09	CT#81	CP-182022	0012	1	F	Resource description in Procedure subclauses	15.1.0
2018-09	CT#81	CP-182022	0013	1	F	Monitoring event API Update	15.1.0
2018-09	CT#81	CP-182022	0014	2	F	BDT API Update	15.1.0
2018-09	CT#81	CP-182022	0015		F	Network Parameter Configuration API Update	15.1.0
2018-09	CT#81	CP-182022	0016	1	F	Complete the common openAPI definition	15.1.0
2018-09	CT#81	CP-182022	0017	2	F	Complete the openAPI definition for BDT API	15.1.0
2018-09	CT#81	CP-182022	0018		F	NIDD API cleanup	15.1.0
2018-09	CT#81	CP-182022	0019	2	F	Complete the openAPI definition for NIDD API	15.1.0
2018-09	CT#81	CP-182022	0020	1	F	One-time monitoring	15.1.0
2018-09	CT#81	CP-182178	0021	3	F	PATCH data type in ChargeableParty API	15.1.0

2018-09	CT#81	CP-182022	0022	1	F	Procedure correction of PFD management error handling	15.1.0
2018-09	CT#81	CP-182022	0023	2	F	Complete the openAPI definition for PFD API	15.1.0
2018-09	CT#81	CP-182022	0024	1	F	Essential Corrections on TS 29.122	15.1.0
2018-09	CT#81	CP-182022	0025	2	F	Allignment of error codes with TS 29.500	15.1.0
2018-09	CT#81	CP-182022	0026	2	F	OpenAPI updates for Device Triggering	15.1.0
2018-09	CT#81	CP-182022	0027	1	F	OpenAPI updates for MSISDN-less Mobile Originated SMS	15.1.0
2018-09	CT#81	CP-182022	0028	2	F	Monitoring event report	15.1.0
2018-09	CT#81	CP-182022	0030		F	Add cause in ProblemDetails data type	15.1.0
2018-09	CT#81	CP-182022	0031	1	F	Remove format keyword for 3GPP defined data type	15.1.0
2018-09	CT#81	CP-182022	0032		F	Correct supported features	15.1.0
2018-09	CT#81	CP-182022	0033		F	Correct ConfigResult data type for Monitoring API	15.1.0
2018-09	CT#81	CP-182022	0034		F	Correct reachability type in monitoring API	15.1.0
2018-09	CT#81	CP-182022	0035		F	Add HSS interaction for NIDD configuration	15.1.0
2018-09	CT#81	CP-182022	0036		F	Correct NiddConfigurationPatch data type	15.1.0
2018-09	CT#81	CP-182179	0037	2	F	Removable attribute definition for NIDD configuration	15.1.0
2018-09	CT#81	CP-182180	0038	2	F	Correct MT NIDD procedure	15.1.0
2018-09	CT#81	CP-182022	0039	1	F	Correct group NIDD procedure	15.1.0
2018-09	CT#81	CP-182022	0040		F	Device Trigger cleanup	15.1.0
2018-09	CT#81	CP-182022	0041		F	Removable attribute definition for PFD management	15.1.0
2018-09	CT#81	CP-182181	0042	2	F	Correct PFD error code	15.1.0
2018-09	CT#81	CP-182022	0043		F	Removable attribute definition for NP configuration	15.1.0
2018-09	CT#81	CP-182022	0044		F	Correct ConfigResult data type for NP Provisioning API	15.1.0
2018-09	CT#81	CP-182022	0045		F	Removable attribute definition for AS required QoS	15.1.0
2018-09	CT#81	CP-182022	0047		F	Version numbering update	15.1.0
2018-09	CT#81	CP-182022	0048	1	F	Removal of external docs field	15.1.0
2018-09	CT#81	CP-182203	0049	-	F	paths property in A.2	15.1.0

History

Document history		
V15.0.0	July 2018	Publication
V15.1.0	October 2018	Publication