**3GPP TSG-CT3 Meeting #112e *C3-205084***

**Electronic Meerting, November 4th - November 13th**

|  |
| --- |
| *CR-Form-v12.1* |
| **CHANGE REQUEST** |
|  |
|  | **29.122** | **CR** | **293** | **rev** | **-** | **Current version:** | **16.7.0** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network |  | Core Network | **X** |

|  |
| --- |
|  |
| ***Title:***  | Adding Support for Indicating Serialization Format in RDS |
|  |  |
| ***Source to WG:*** | Convida Wireless LLC, Intel |
| ***Source to TSG:*** | CT3 |
|  |  |
| ***Work item code:*** | RDSSI  |  | ***Date:*** | 11-12-2020 |
|  |  |  |  |  |
| ***Category:*** | B |  | ***Release:*** | Rel-17 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)Rel-12 (Release 12)**Rel-13 (Release 13)Rel-14 (Release 14)Rel-15 (Release 15)Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18)* |
|  |  |
| ***Reason for change:*** | As part of the RDSSI (SP-190446) work item, SA2 added support to the Reliable Data Service for indicating the serialization format of the data that will be sent in a NIDD session. This feature was added to the SCEF in TS 23.682 (clauses 4.5.14.3 and 5.13.2) and the NEF in TS 23.501 (clause 5.31.6) and TS 23.502 (clause 4.25.3).The corresponding CT1 CR is C1-206475. |
|  |  |
| ***Summary of change:*** | Updated the Individual ManagePort Configuration so that the SCS can indicate what serialization format(s) it supports on the port and so that the SCEF can indicate what serialization format has been configured/negotiated for the port. |
|  |  |
| ***Consequences if not approved:*** | Serialization format indications would not be supported and stage-3 specification will be inconsistent with stage-2. |
|  |  |
| ***Clauses affected:*** | 4.4.5.6.1, 5.6.2.1.9, 5.6.2.2.5, 5.6.2.3.x (new), 5.6.5.3, 5.6.4, A.6 |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **X** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **X** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** | This CR introduces backward compatible feature into the OpenAPI file to NIDD API |
|  |  |
| ***This CR's revision history:*** |  |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* First Change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##### 4.4.5.6.1 Port Reservation and Release

As part of the Port Management configuration, operations to reserve a combination of port numbers, release a combination of port numbers, query the list of port numbers that are reserved and notification of reservation of a port number may be performed, if the Rds\_dynamic\_port feature is supported.

Indication of the supported serializtion formats by the SCS/AS, query of the supported and configured serializtion formats by the SCS/AS, and notification of the supported and configured serializtion formats by the SCS/AS may be performed if the Rds\_serialization\_format feature is supported.

If the SCS/AS needs to reserve port numbers and associate them with an application, 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 NIDD configuration resource and the specific part of "/rds-ports/{portId}" as described in subclause 5.6.3.9.2. The SCS/AS may use this operation to reserve port numbers on the UE and SCEF and associate them with an application. The SCS/AS may also use this operation to indicate the serialization formats that are supported by the SCS/AS on the port. Upon receipt of the HTTP PUT request from the SCS/AS,

- if the "skipUeInquiry" is set to "false" and if the "individual ManagePort Configuration" resource already exists in the same NIDD configuration, the SCEF shall respond with 403 Forbidden response with a cause value "PORT\_NOT\_FREE" in the "cause" attribute of the "ProblemDetails" structure; otherwise, the SCEF shall interact with the UE via the SGSN/MME to reserve the port and optionally configure the serialization format by using RDS protocol as specified in 3GPP TS 24.250 [31] and return 202 Accept to the SCS/AS if successful response is received from the SGSN/MME. Then if the SCEF receives successful UE response, the SCEF shall create the "individual ManagePort Configuration" resource and notify the SCS/AS with the reserved port and configured serialization format as specified in subclause 4.4.5.6.2, the SCEF shall also mark the resource is created by the SCS/AS; otherwise, the SCEF shall notify the SCS/AS about the currently reserved ports as specified in subclause 4.4.5.6.2.

- if the "skipUeInquiry" is set to "true" and if the requested SCEF port already exists in an NIDD configuration within the same APN, the SCEF shall respond with 403 Forbidden response with a cause value "PORT\_NOT\_FREE" in the "cause" attribute of the "ProblemDetails" structure; otherwise, the SCEF shall create the "individual ManagePort Configuration" resource and send an HTTP 201 Created response to the SCS/AS, the SCEF shall also mark the resource is created by the SCS/AS and notify the UE by using RDS protocol as specified in 3GPP TS 24.250 [31].

If the SCS/AS needs to release port numbers associated with an application, 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 ManagePort Configuration" resource. Upon receipt of of the HTTP DELETE request from the SCS/AS, if the "individual ManagePort Configuration" resource does not exist in the same NIDD configuration, the SCEF shall respond with 404 Not Found with a cause value "PORT\_NOT\_ASSOC\_WITH\_APP" in the "cause" attribute of the "ProblemDetails" structure; otherwise if the the "individual ManagePort Configuration" resource was created by the SCS/AS and

- if the "skipUeInquiry" is set to "false", the SCEF shall interact with the UE via the SGSN/MME to release the port by using RDS protocol as specified in 3GPP TS 24.250 [31] and return 202 Accept to the SCS/AS if successful response is received from the SGSN/MME. Then upon receipt of the UE response, the SCEF shall notify the SCS/AS with the currently reserved ports as specified in subclause 4.4.5.6.2.

- if the "skipUeInquiry" is set to "true", the SCEF shall delete the individual ManagePort Configuration resource and respond with an HTTP 204 No Content response to the SCS/AS. The SCEF shall also notify the UE by using RDS protocol as specified in 3GPP TS 24.250 [31].

If the HTTP DELETE request is received for the "Individual ManagePort Configuration" resource which was created by the UE, the SCEF shall reject the message with 403 Forbidden response with a cause value "OPERATION\_PROHIBITED" in the "cause" attribute of the "ProblemDetails" structure.

If the "skipUeInquiry" is set to "false" and the SCEF is not able to interact with the UE because:

- PDN connection is not established, the SCEF shall reject the HTTP PUT/DELETE request with a 500 Internal Server Error response with a cause value "NO\_PDN\_CONNECTION";

- UE is not reachable, the SCEF shall reject the HTTP PUT/DELETE request with a 500 Internal Server Error response with a cause value "TEMPORARILY\_NOT\_REACHABLE". The response may include a Requested Re-Transmission time to indicate the SCS/AS when the UE is expected to be reachable so that the SCS/AS may prepare any re-configuration for the RDS port; or

- the interaction with the SGSN/MME is not successful, the SCEF shall reject the HTTP PUT/DELETE request with a 500 Internal Server Error and a proper cause value indicating the reason for the delivery failure.

If the SCS/AS needs to read the port numbers and serialization formats that are asocigted with an application, the SCS/AS shall send an HTTP GET message to the SCEF, using the URI received in the response to the request that has created the NIDD configuration resource and the specific part of "/rds-ports/{portId}" as described in subclause 5.6.3.9.2.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Next Change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##### 5.6.2.1.9 Type: ManagePort

This type represents the configuration for RDS dynamic port management which is applicable for Rds\_dynamic\_port feature.

Table 5.6.2.1.9-1: Definition of type ManagePort

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | Cardinality | Description | Applicability |
| self | Link | 0..1 | Link to the resource "Individual ManagePort Configuration". This parameter shall be supplied by the SCEF in HTTP PUT/GET responses and in the notification POST request. |  |
| applicationId | string | 1 | Indicates the application that is associated with port configuration specified above (as defined in subclause 5.4.2.6 of 3GPP TS 24.250 [31]). |  |
| manageEntity | ManageEntity | 0..1 | Indicates the "Individual ManagePort Configuration" resource is created by which function entity (UE or the AS). This attribute is supplied by the SCEF. |  |
| skipUeInquiry | boolean | 0..1 | Indicate whether to skip UE inquiry. Set to "true" if the SCS/AS decides to reserve port on the SCEF without confirming with the UE; otherwise set to "false". Default value is "false" if omitted. |  |
| supportedFormats | array(SerializationFormat) | 0..N | Indicates the serialization format(s) that are supported by the SCS/AS on the associated RDS port. This attribute is supplied by the SCS/AS. | Rds\_serialization\_format |
| configuredFormat | SerializationFormat | 0..1 | Indicates the serialization format that has been configured/negotiated for the RDS port. This attribute is supplied by the SCEF.(NOTE) | Rds\_serialization\_format |
| NOTE: The same serialization format is used in the MO and MT directions. |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Next Change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##### 5.6.2.2.5 Type: RdsDownlinkDataDeliveryFailure

This type represents the failure delivery result for RDS. It shall comply with the provisions defined in table 5.6.2.2.5-1.

Table 5.6.2.2.5-1: Definition of RdsDownlinkDataDeliveryFailure data Type

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Data type | Cardinality | Description |
| requestedRetransmissionTime | DateTime | 0..1 | Identifies the absolute time at which the UE will be reachable.This parameter may be supplied by the SCEF for error indicating "TEMPORARILY\_NOT\_REACHABLE" |
| supportedUeFormats | array(SerializationFormat) | 0..N | Indicates the serialization format(s) that are supported by the UE on the associated RDS port. This parameter may be supplied by the SCEF for error indicating " SERIALIZATION\_FORMAT\_NOT\_SUPPORTED" |
| NOTE: This data type also contains all the properties defined for ProblemDetails data type in subclause 5.2.1.2.12. |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Next Change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##### 5.6.2.3.x Enumeration: SerializationFormat

The enumeration SerializationFormat represents a serialization format associated with an RDS port which is applicable for the Rds\_serialization\_format feature.

Table 5.6.2.3.x-1: Enumeration SerializationFormat

|  |  |  |
| --- | --- | --- |
| Enumeration value | Description | Applicability (NOTE) |
| CBOR | The CBOR Serialzition format |  |
| JSON | The JSON Serialzition format |  |
| XML | The XML Serialzition format |  |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Next Change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 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. |
| 5 | Rds\_port\_verification | This feature indicates the support of RDS port verification in the MO/MT NIDD delivery. |
| 6 | Rds\_dynamic\_port | This feature indicates the support of RDS dynamic port management. |
| 7 | Rds\_serialization\_format | This feature indicates the support of RDS Serialization Format reservation, notification, and query. This feature requires that the Rds\_dynamic\_port is also supported. |
| 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. |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Next Change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### 5.6.5.3 Application Errors

The application errors defined for the NIDD API are listed in table 5.6.5.3-1. The SCEF shall include in the HTTP status code a "ProblemDetails" data structure with the "cause" attribute indicating the application error as listed in table 5.6.5.3-1.

Table 5.6.5.3-1: Application errors

|  |  |  |  |
| --- | --- | --- | --- |
| Application Error | HTTP status code | Description | Applicability |
| QUOTA\_EXCEEDED | 403 Forbidden | Not enough quota for the MT NIDD |  |
| DATA\_TOO\_LARGE | 403 Forbidden | The non-IP data size is larger than "maximumPacketSize" of the NIDD configuration. |  |
| RDS\_PORT\_UNKNOWN | 403 Forbidden | The SCEF does not know the RDS port numbers in the MT NIDD, such port numbers does not match with the configured port numbers. | Rds\_port\_verification |
| OPERATION\_PROHIBITED | 403 Forbidden | Indicates the operation is prohibitted. |  |
| PORT\_NOT\_FREE | 403 Forbidden | Port is not free as it is already associated with an application | Rds\_dynamic\_port |
| ALREADY\_DELIVERED | 404 Not Found | The SCEF has already delivered the buffered data. |  |
| PORT\_NOT\_ASSOC\_WITH\_APP | 404 Not Found | Port is free and is not associated with any application. | Rds\_dynamic\_port |
| SENDING | 409 Conflict | The SCEF is already in sending the buffered non-IP data. |  |
| TRIGGERED | 500 Internal Server Error | The SCEF triggered the device but did not buffer the data. The SCS AS may resubmit the data |  |
| TEMPORARILY\_NOT\_REACHABLE | 500 Internal Server Error | 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. |  |
| NEXT\_HOP | 500 Internal Server Error | Unsuccessful delivery to the next hop. |  |
| TIMEOUT | 500 Internal Server Error | Unsuccessful delivery due to timeout. |  |
| NO\_PDN\_CONNECTION | 500 Internal Server Error | The SCEF cannot proceed since there is no PDN connection. |  |
| SERIALIZATION\_FORMAT\_NOT\_SUPPORTED | 500 Internal Server Error | The SCEF was not able to configure a Serialization Format for the port | Rds\_serialization\_format |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Next Change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## A.6 NIDD API

openapi: 3.0.0

info:

 title: 3gpp**-**nidd

 version: 1.1.0

 description: |

 API for non IP data delivery.

 © 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

 All rights reserved.

externalDocs:

 description: 3GPP TS 29.122 V16.6.0 T8 reference point for Northbound APIs

 url: 'http://www.3gpp.org/ftp/Specs/archive/29\_series/29.122/'

security:

 - {}

 - oAuth2ClientCredentials: []

servers:

 - url: '{apiRoot}/3gpp**-**nidd/v1'

 variables:

 apiRoot:

 default: https://example.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'

 '406':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/406'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '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'

 headers:

 Location:

 description: 'Contains the URI of the newly created resource'

 required: true

 schema:

 type: string

 '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'

 '411':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/411'

 '413':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/413'

 '415':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/415'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '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.

 required: true

 content:

 application/json:

 schema:

 oneOf:

 - $ref: '#/components/schemas/NiddConfigurationStatusNotification'

 - $ref: '#/components/schemas/NiddUplinkDataNotification'

 - $ref: '#/components/schemas/NiddDownlinkDataDeliveryStatusNotification'

 - $ref: '#/components/schemas/GmdNiddDownlinkDataDeliveryNotification'

 - $ref: '#/components/schemas/ManagePortNotification'

 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'

 '404':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/404'

 '411':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/411'

 '413':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/413'

 '415':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/415'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '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'

 '406':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/406'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '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'

 '411':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/411'

 '413':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/413'

 '415':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/415'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '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'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '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'

 '406':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/406'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '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'

 headers:

 Location:

 description: 'Contains the URI of the newly created resource'

 required: true

 schema:

 type: string

 '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'

 '411':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/411'

 '413':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/413'

 '415':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/415'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '500':

 description: The NIDD downlink data delivery request was not successful.

 content:

 application/json:

 schema:

 $ref: '#/components/schemas/NiddDownlinkDataDeliveryFailure'

 '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'

 '406':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/406'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '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'

 '413':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/413'

 '415':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/415'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '500':

 description: The NIDD downlink data replacement request was not successful.

 content:

 application/json:

 schema:

 $ref: '#/components/schemas/NiddDownlinkDataDeliveryFailure'

 '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'

 '409':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/409'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '500':

 description: The NIDD downlink data cancellation request was not successful.

 content:

 application/json:

 schema:

 $ref: '#/components/schemas/NiddDownlinkDataDeliveryFailure'

 '503':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/503'

 default:

 $ref: 'TS29122\_CommonData.yaml#/components/responses/default'

 /{scsAsId}/configurations/{configurationId}/rds-ports:

 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 ManagePort configurations.

 content:

 application/json:

 schema:

 type: array

 items:

 $ref: '#/components/schemas/ManagePort'

 minItems: 0

 description: individual ManagePort 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'

 '406':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/406'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '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}/rds-ports/{portId}:

 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: portId

 description: The UE port number.

 in: path

 required: true

 schema:

 type: string

 pattern: '^(ue([0-9]|(1[0-5]))-ef([0-9]|(1[0-5])))$'

 get:

 responses:

 '200':

 description: The individual ManagePort configuration is successfully retrieved.

 content:

 application/json:

 schema:

 $ref: '#/components/schemas/ManagePort'

 '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'

 '406':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/406'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '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 ManagePort configuration.

 required: true

 content:

 application/json:

 schema:

 $ref: '#/components/schemas/ManagePort'

 responses:

 '201':

 description: The individual ManagePort configuration is created.

 content:

 application/json:

 schema:

 $ref: '#/components/schemas/ManagePort'

 headers:

 Location:

 description: 'Contains the URI of the newly created resource'

 required: true

 schema:

 type: string

 '202':

 description: The request is accepted and under processing.

 '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'

 '413':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/413'

 '415':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/415'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '500':

 description: The request was not successful.

 content:

 application/problem+json:

 schema:

 $ref: '#/components/schemas/RdsDownlinkDataDeliveryFailure'

 '503':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/503'

 default:

 $ref: 'TS29122\_CommonData.yaml#/components/responses/default'

 delete:

 responses:

 '202':

 description: The request is accepted and under processing.

 '204':

 description: The individual ManagePort 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'

 '409':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/409'

 '429':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/429'

 '500':

 description: The request was not successful.

 content:

 application/problem+json:

 schema:

 $ref: '#/components/schemas/RdsDownlinkDataDeliveryFailure'

 '503':

 $ref: 'TS29122\_CommonData.yaml#/components/responses/503'

 default:

 $ref: 'TS29122\_CommonData.yaml#/components/responses/default'

components:

 securitySchemes:

 oAuth2ClientCredentials:

 type: oauth2

 flows:

 clientCredentials:

 tokenUrl: '{tokenUrl}'

 scopes: {}

 schemas:

 NiddConfiguration:

 type: object

 properties:

 self:

 $ref: 'TS29122\_CommonData.yaml#/components/schemas/Link'

 supportedFeatures:

 $ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

 mtcProviderId:

 type: string

 description: Identifies the MTC Service Provider and/or MTC Application.

 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) to indicate if a reliable data service acknowledgment is enabled or not.

 rdsPorts:

 type: array

 items:

 $ref: '#/components/schemas/RdsPort'

 minItems: 1

 description: Indicates the static 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).

 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

 minimum: 1

 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. 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: 1

 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

 oneOf:

 - required: [externalId]

 - required: [msisdn]

 - required: [externalGroupId]

 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/Bytes'

 reliableDataService:

 type: boolean

 description: The reliable data service (as defined in subclause 4.5.15.3 of 3GPP TS 23.682) 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

 oneOf:

 - required: [externalId]

 - required: [msisdn]

 - required: [externalGroupId]

 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/Bytes'

 reliableDataService:

 type: boolean

 description: Indicates whether the reliable data service is enabled.

 rdsPort:

 $ref: '#/components/schemas/RdsPort'

 required:

 - niddConfiguration

 - data

 oneOf:

 - required: [externalId]

 - required: [msisdn]

 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'

 rdsCapIndication:

 type: boolean

 description: It indicates whether the network capability for the reliable data service is enabled or not.

 rdsPort:

 $ref: '#/components/schemas/RdsPort'

 required:

 - niddConfiguration

 - status

 oneOf:

 - required: [externalId]

 - required: [msisdn]

 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:

 externalId:

 $ref: 'TS29122\_CommonData.yaml#/components/schemas/ExternalId'

 msisdn:

 $ref: 'TS29122\_CommonData.yaml#/components/schemas/Msisdn'

 deliveryStatus:

 $ref: '#/components/schemas/DeliveryStatus'

 requestedRetransmissionTime:

 $ref: 'TS29122\_CommonData.yaml#/components/schemas/DateTime'

 required:

 - deliveryStatus

 oneOf:

 - required: [externalId]

 - required: [msisdn]

 NiddDownlinkDataDeliveryFailure:

 type: object

 properties:

 problemDetail:

 $ref: 'TS29122\_CommonData.yaml#/components/schemas/ProblemDetails'

 requestedRetransmissionTime:

 $ref: 'TS29122\_CommonData.yaml#/components/schemas/DateTime'

 required:

 - problemDetail

 ManagePort:

 type: object

 properties:

 self:

 $ref: 'TS29122\_CommonData.yaml#/components/schemas/Link'

 appId:

 type: string

 description: Identifies the application.

 manageEntity:

 $ref: '#/components/schemas/ManageEntity'

 skipUeInquiry:

 type: boolean

 description: Indicate whether to skip UE inquiry.

 supportedFormats:

 type: array

 items:

 $ref: '#/components/schemas/SerializationFormat'

 minItems: 1

 description: Indicates the serialization format(s) that are supported by the SCS/AS on the associated RDS port.

 configuredFormat:

 $ref: '#/components/schemas/SerializationFormat'

 required:

 - appId

 ManagePortNotification:

 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'

 managedPorts:

 type: array

 items:

 $ref: '#/components/schemas/ManagePort'

 minItems: 1

 description: Indicates the reserved RDS port configuration information.

 required:

 - niddConfiguration

 oneOf:

 - required: [externalId]

 - required: [msisdn]

 RdsDownlinkDataDeliveryFailure:

 allOf:

 - $ref: 'TS29122\_CommonData.yaml#/components/schemas/ProblemDetails'

 - type: object

 properties:

 requestedRetransmissionTime:

 $ref: 'TS29122\_CommonData.yaml#/components/schemas/DateTime'

 supportedUeFormats:

 type: array

 items:

 $ref: '#/components/schemas/SerializationFormat'

 minItems: 1

 description: Indicates the serialization format(s) that are supported by the UE on the associated RDS port.

 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:

 - $ref: '#/components/schemas/PdnEstablishmentOptions'

 - $ref: 'TS29571\_CommonData.yaml#/components/schemas/NullValue'

 DeliveryStatus:

 anyOf:

 - type: string

 enum:

 - SUCCESS

 - SUCCESS\_NEXT\_HOP\_ACKNOWLEDGED

 - SUCCESS\_NEXT\_HOP\_UNACKNOWLEDGED

 - SUCCESS\_ACKNOWLEDGED

 - SUCCESS\_UNACKNOWLEDGED

 - TRIGGERED

 - BUFFERING

 - BUFFERING\_TEMPORARILY\_NOT\_REACHABLE

 - SENDING

 - FAILURE

 - FAILURE\_RDS\_DISABLED

 - FAILURE\_NEXT\_HOP

 - FAILURE\_TIMEOUT

 - FAILURE\_TEMPORARILY\_NOT\_REACHABLE

 - 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

 - TRIGGERED: The SCEF triggered the device and is 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

 - FAILURE: Delivery failure but details not provided

 - FAILURE\_RDS\_DISABLED: RDS was disabled

 - FAILURE\_NEXT\_HOP: Unsuccessful delivery to the next hop.

 - FAILURE\_TIMEOUT: Unsuccessful delivery due to timeout.

 - FAILURE\_TEMPORARILY\_NOT\_REACHABLE: The SCEF has been informed that the UE is temporarily not reachable without buffering the data.

 readOnly: true

 NiddStatus:

 anyOf:

 - type: string

 enum:

 - ACTIVE

 - TERMINATED\_UE\_NOT\_AUTHORIZED

 - TERMINATED

 - RDS\_PORT\_UNKNOWN

 - 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.

 - RDS\_PORT\_UNKNOWN: The RDS port is unknown.

 readOnly: true

 ManageEntity:

 anyOf:

 - type: string

 enum:

 - UE

 - AS

 - 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

 - UE: Representing the UE.

 - AS: Representing the Application Server.

 readOnly: true

 SerializationFormat:

 anyOf:

 - type: string

 enum:

 - CBOR

 - JSON

 - XML

 - 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

 - CBOR: The CBOR Serialzition format

 - JSON: The JSON Serialzition format

 - XML: The XML Serialzition format

 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) to indicate if a reliable data service acknowledgment is enabled or not.

 nullable: true

 rdsPorts:

 type: array

 items:

 $ref: '#/components/schemas/RdsPort'

 minItems: 1

 description: Indicates the static 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).

 pdnEstablishmentOption:

 $ref: '#/components/schemas/PdnEstablishmentOptionsRm'