ETSI TS 129 250 V19.0.0 (2025-10)

TECHNICAL SPECIFICATION

LTE;
Nu reference point between SCEF and PFDF
for sponsored data connectivity
(3GPP TS 29.250 version 19.0.0 Release 19)

Reference
RTS/TSGC-0329250vj00

Keywords
LTE

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Legal Notice

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. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found at 3GPP to ETSI numbering cross-referencing.

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 document provides the stage 3 specification of the Nu reference point. The functional requirements and the stage 2 specifications of the Nu reference point are specified in 3GPP TS 23.682 [2]. The Nu reference point lies between the Packet Flow Description Function (PFDF) and the Service Capability Exposure Function (SCEF).

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 23.203: "Policy and charging control architecture".

[4] 3GPP TS 29.213: "Policy and Charging Control signalling flows and QoS parameter mapping".

[5] 3GPP TS 33.210: "3G security; Network Domain Security (NDS); IP network layer security".

[6] IETF RFC 2818: "HTTP Over TLS".

[7] IETF RFC 793: "Transmission Control Protocol".

[8] Void.

[9] 3GPP TS 29.251: "Gw and Gwn reference points for sponsored data connectivity".

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

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

[12] IETF draft-newton-json-content-rules-09: "A Language for Rules Describing JSON Content".

NOTE: This individual draft will not further progress in IETF. It is available from the following link: https://www.ietf.org/archive/id/draft-newton-json-content-rules-09.txt.

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

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

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

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

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

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

[19] 3GPP TS 29.274: "Evolved GPRS Tunnelling Protocol for EPS (GTPv2)".

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].

Packet Flow Description (PFD): A set of information enabling the detection of application traffic provided by a 3rd party service provider (from 3GPP TS 23.203 [3]).

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 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].

JSON	JavaScript Object Notation
PCEF	Policy and Charging Enforcement Function
PFD	Packet Flow Description
PFDF	Packet Flow Description Function
SCEF	Service Capability Exposure Function
TDF	Traffic Detection Function

4 Nu reference point

4.1 Overview

The Nu reference point is located between the Packet Flow Description Function (PFDF) and the Service Capability Exposure Function (SCEF). The Nu reference point is used for provisioning of PFDs from the SCEF to the PFDF and reporting the result of the PFD Management from the PFDF to the SCEF.

The stage 2 level requirements for the Nu reference point are defined in 3GPP TS 23.682 [2].

4.2 Nu reference model

The Nu reference point is defined between the SCEF and the PFDF. The relationships between the different functional entities involved are depicted in figure 4.2.1. The overall PCC architecture is depicted in clause 3a of 3GPP TS 29.213 [4].

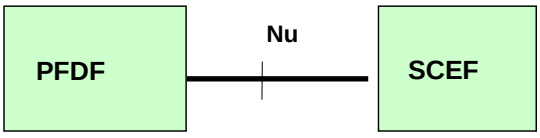

Figure 4.2.1: Nu reference model

4.3 Functional elements

4.3.1 PFDF

The PFDF (Packet Flow Description Function) is a functional element which receives and manages the PFDs associated to application identifier (s) from the SCEF via the Nu reference point.

The PFDF provisions PFDs for the corresponding application identifier (s) to the PCEF/TDF as defined in 3GPP TS 23.203 [3] and 3GPP TS 29.251 [9].

4.3.2 SCEF

The SCEF (Service Capability Exposure Function) is a functional element which provides means to securely expose the services and capabilities provided by the 3GPP network interfaces.

The SCEF shall support the management of PFDs provided by the 3rd party SCS/AS. The SCEF may provision the PFDs to the PFDF via the Nu reference point.

4.4 Procedures over Nu reference point

4.4.1 Management of PFD

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

If the SCEF receives one or more sets of PFDs for external application identifier (s) provisioned by the third party SCS/AS, which is authorized to perform the management of PFDs based on operator policies, the SCEF shall:

- If the external application identifier(s) is different from the application identifier(s) known at the PFDF, translate the external application identifier(s) to the application identifier(s) known at the PFDF; and
- may check if the allowed delay satisfies the required SLA against the minimum allowed delay as defined in 3GPP TS 23.682 [2]; and
- send an HTTP POST message to the PFDF including the provisioned PFD changes for the the application identifier (s) within the body of the HTTP POST as described in clause 5.3.5.2.

NOTE 1: It is up to operator configuration whether to use different external application identifiers that require a mapping to application identifiers known at the PFDF. The external application identifier can be the same as the application identifier known at the PFDF.

Upon receipt of the HTTP request for the provisioning operation from the SCEF, the PFDF shall perform the following steps:

- If an allowed delay is received for an application identifier, for Pull mode as defined in 3GPP TS 29.251 [9], the PFDF shall compare the allowed delay with the configured caching time which is:
  - a caching time value configured for that application identifier; or
  - the default caching time value if no caching time value is configured for that application identifier.
- Then if the PFDF cannot ensure the PCEF/TDF will pull the PFDs in time (i.e. allowed delay is shorter than the caching time), the PFDF shall within the HTTP response send a failure reason and that caching time value used in the comparison and may still store (create/update/remove) the PFDs for this application identifier.

NOTE 2: In the Combination mode as defined in 3GPP TS 29.251 [9], the PFDF can check the received allowed delay against the caching time but will always store (create/update/remove) the PFDs.

- In the Pull mode as defined in 3GPP TS 29.251 [9], for the application identifier(s) without the need to send failure reason; or in the Push or Combination mode as defined in 3GPP TS 29.251 [9], for received application identifier(s), the PFDF shall:

- delete all the PFD(s) for the application identifier(s) where the removal-flag is also provided and set to true;
- update the existing PFD(s) if a new PFD(s) with the same PFD identifier(s) is received , add new PFD(s) if the new PFD(s) with a new PFD identifier(s) is received, and/or delete an existing PFD(s) if the same PFD identifier(s) without any content is received, where the partial-flag is also provided and set to true;
- remove existing PFD(s) (if available) and install the new PFD(s) for the corresponding PFD identifier(s) whereno flag is provided;
- acknowledge the HTTP POST message by sending a corresponding HTTP response with the appropriate status code as defined in clause 5.3.2. If the POST operation was successful for at least one application identifier, the PFDF shall respond with an HTTP 200 OK status code.

4.4.2 PFD management notification

In the Push mode or Combination mode as defined in 3GPP TS 29.251 [9], if the PFDs are provisioned to at least one of the known PCEFs/TDFs (but not all) within the allowed delay (i.e. the provisioned PFDs can not be enforced successfully in some PCEF/TDFs known on the PFDF), the PFDF may notify the SCEF about the failed PFD provisioning with the HTTP POST message using the failure reason "PARTIAL_FAILURE" as defined in Table 5.4.7.1-1. In this case, the PFDF may include location area(s) of the PCEF/TDF(s) which can not enforce the provisioned PFD(s) within the field "user-plane-location-area" of the corresponding instance of the PFD report(s). If the PFDs are provisioned to none of the known PCEFs/TDFs within the allowed delay, the PFDF shall notify the SCEF about the failed PFD provisioning with the HTTP POST message using appropriate failure reason as defined in clause 6.4.6.3 of 3GPP TS 29.251 [9].

When receiving the HTTP POST message, the SCEF shall respond with an HTTP response message.

5 Nu protocol

5.1 Introduction

The following layers of the protocol stack for the Nu reference point between the SCEF and the PFDF are described in clauses:

- IETF RFC 793 [7] provides the communication service at the transport layer.
- An optional communication security layer can be added between the transport and the application delivery layer (see clause 6).
- The application delivery layer provides the transport of the specific application communication data using IETF RFC 7230 [13], IETF RFC 7231 [14], IETF RFC 7232 [15], IETF RFC 7233 [16], IETF RFC 7234 [17] and IETF RFC 7235 [18].
- The specific application communication layer constitutes the transport of the JSON content type.

Figure 5.1.1 illustrates the protocol stack of the RESTful Nu reference point.

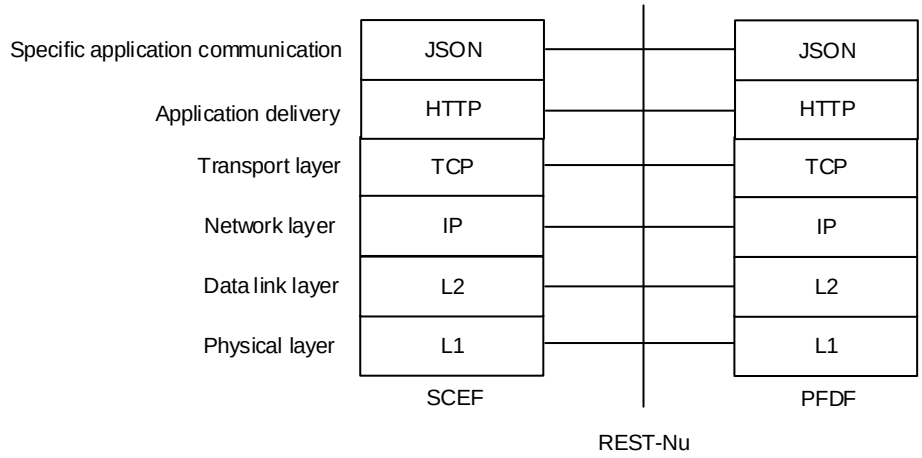

Figure 5.1.1: Protocol stack of the RESTful Nu reference point

5.2 Transport layer

HTTP is layered over TCP, which provides a reliable transport.

For provisioning of PFDs from the SCEF to the PFDF, the SCEF acts as an HTTP client and the PFDF acts as an HTTP server. As a result, the SCEF shall initiate a TCP connection with the PFDF.

5.3 Application delivery layer

5.3.1 General

The application delivery layer shall use RESTful HTTP.

The application delivery layer provides provisioning of the PFDs by the SCEF.

If the SCEF needs to provision PFDs for a set of application identifier(s) (creation/update/deletion) to the PFDF, the SCEF shall send an HTTP POST message.

5.3.2 HTTP status codes

The HTTP status codes for the REST-based Nu interface are specified in the IETF RFC 7231 [14].

5.3.3 Methods

Methods indicate to the server what action has to be performed. Every HTTP request message has a method.

The HTTP POST method is used by the SCEF to provision PFDs for a set of application identifiers. The request URI defines the address responsible for the management of the PFDs provisioning as a controller resource.

The HTTP POST method is also used by the PFDF to inform the SCEF for PFD management notification.

Every HTTP request results in a response message that comes back with a status code and further information in its body, if required. The HTTP request initiator waits for this response before initiating a further request.

5.3.4 Resources and URI design

The URI design shall be based on the structure defined in IETF RFC 3986 [10]:

scheme ":" hier-part ["?" query] ["#" fragment]
hier-part = "//" authority path-abempty
/ path-absolute
/ path-rootless
/ path-empty

The scheme may be HTTP or HTTPS for the Nu interface. Within a scheme the definition of names shall follow the rules of HTTP URIs. Host and port are the main parts of the authority. The path element identifies the resources.

For the Nu interface, the following required parts of the URI shall be used as follows:

scheme: The application delivery layer protocol "http" or "https".

authority: It includes the server address and optionally a port as follows: host [":" port]

path-absolute: The path-absolute should have the following ABNF: "/" mainapp "/" mainresource. In this release:

"mainapp" is "nuapplication".

"mainresource" is "provisioning".

The PFDs management (associating/disassociating PFDs with application identifiers) in PFDF is a controller resource that is responsible for processing requests that provisioning a set of changes for more than one set of PFDs for corresponding application identifiers atomically.

An example of the URI to identify the controller resource is http://pfdfserver.example.com/nuapplication/provisioning.

NOTE: A different path can be used when the Resource URI is preconfigured in the SCEF.

5.3.5 HTTP request/response formats

5.3.5.1 General

The PFDs provisioning procedure is performed through HTTP transactions consisting of a request initiated by the SCEF and answered by the PFDF.

Table 5.3.5.1-1 summarizes the content of the requests and responses. More detailed information is specified in the corresponding clauses as indicated in the table.

Table 5.3.5.1-1: Nu requests/response summary table

Method	Resource URI's path	Clause Defined	Request body	Initiator	Response body
POST	/nuapplication/provisioning (NOTE 1)	5.3.5.2	Content-Type: application/json The SCEF shall include PFDs content associated with application identifier(s) using the schema defined in Annex A.	SCEF	Successful response: The PFDF may include informational data in the body of the response in Annex A.
POST	{scef-notification-uri} or /nuapplication/notification (NOTE 2)	5.3.5.3	Content-Type: application/json The PFDF shall include notification result and application identifier(s) using the schema defined in Annex A.	PFDF	
NOTE 1: A different path from /nuapplication/provisioning may be used when it is configured in the SCEF. In that case the "path" part set in the different methods should use the configured one. NOTE 2: A different path from /nuapplication/notification may be used when it is configured in the PFDF and no scef-notification-uri is received in the provisioning POST. In that case the "path" part set in the different methods should use the configured one.					

5.3.5.2 POST /nuapplication/provisioning

The provisioning of the PFDs shall be performed by the SCEF by using the POST method as follows:

- The request URI formatted as defined in clause 5.3.4 with the "path" part set to: /nuapplication/provisioning.
- The Content-Type header field set to "application/json"
- The body of the message encoded in JSON format as defined in Annex A. The body shall include
  - for the PFD(s) creation for a new application identifier, a new application identifier and its full list of PFD(s) to be created;
  - for the PFD(s) full update for an existing application identifier, the existing application identifier and its new full list of PFD(s)
  - for the PFD(s) partial update for an existing application identifier, the existing application identifier, partial update indication and
    - new PFD(s) with new PFD identifier(s) to add new PFD(s),
    - new PFD(s) with existing PFD identifier(s) to update existing PFD(s), and/or
    - the existing PFD identifier(s) without any content to remove the existing PFD(s);
  - for the PFD(s) removal for an existing application identifier, the existing application identifier and the removal indication.

Upon receipt of the HTTP POST request, the PFDF shall respond to the SCEF indicating whether the provisioning was successful or not using one of the HTTP status codes as defined in clause 5.3.2. If the provisioning was accepted, the PFDF shall respond with an HTTP 200 OK status code if no resource is created, or an HTTP 201 Created status code if one or more resources are created. If the allowed delay is too short according to the criteria in clause 4.4.1, the PFDF shall respond with an HTTP 200 OK status code and additional information in the body of the response indicating failure reason "too short allowed delay" and the value of caching time as defined in Annex A. If the provisioning was rejected, the PFDF shall indicate the reason using an appropriate HTTP status code as defined in clause 5.3.2 and optionally additional information in the body of the response as defined in Annex A.

Below is an example of an HTTP POST and a corresponding successful response:

```
POST /nuapplication/provisioning HTTP/1.1
Host: pfdfserver.example.com
Content-Type: application/json
Content-Length: …
[
  {
    "application-identifier":"test-application-1",
    "removal-flag":true
  },
  {
    "application-identifier":"test-application-2",
    "allowed-delay":600,
    "pfds":[
     {
        "pfd-identifier":"pfd1",
        "flow-descriptions":[
         "permit in ip from 10.68.28.39 80 to any"
        ]
      },
      {
        "pfd-identifier":"pfd2",
        "urls":[
         "^http://test.example.com(/\\S*)?$"
        ]
      }
    ]
  },
  {
    "application-identifier":"test-application-3",
    "partial-flag":true,
    "pfds":[
     {
        "pfd-identifier":"pfd3",
        "urls":[
          "^http://test.example2.net(/\\S*)?$"
```

```
          ]
        },
        {
          "pfd-identifier":"pfd4"
        }
    ]
  }
]
```

Here is an example of a successful response:

```
HTTP/1.1 200 OK
Date: Mon, 7 May 2012 16:00:00 GMT
Server: pfdfserver.example.com
Content-Type: application/json
{
  "success-message": "Notification was processed successfully.",
}
```

5.3.5.3 POST /nuapplication/notification

The notification of the PFD management shall be performed by the PFDF by using the POST method as follows:

- The request URI formatted as defined in clause 5.3.4 with the "path" part set to: /nuapplication/notification.
- The Content-Type header field set to "application/json".
- The body of the message encoded in JSON format as defined in Annex A. The body shall include the pfd-report.

Upon receipt of the HTTP POST request, the SCEF shall respond to the PFDF with HTTP 200 OK status code.

Below is an example of an HTTP POST and a corresponding response:

```
POST /nuapplication/notification HTTP/1.1
Host: scefserver.example.com
Content-Type: application/json
Content-Length: …
{
  "notification-pfd-reports": [
    {
      "application-ids": ["test-application-1", "test-application-2", "test-application-3"],
      "pfd-failure-code":"PARTIAL_FAILURE",
      "user-plane-location-area": {
              "cell-ids" : ["46000045BD6007", "46000045BD6008"],
```

```
            "tracking-area-ids" : ["46000063F8", "46000063F9"]
        }
    }
 ]
}
```


Here is an example of a response:

HTTP/1.1 200 OK

Date: Mon, 7 May 2018 16:00:00 GMT

Server: scefserver.example.com

Content-Type: application/json

5.3.6 Feature negotiation

5.3.6.1 General

The REST based Nu interface needs to provide a mechanism to advertise required and optional features supported by both the SCEF and PFDF for interoperability reasons as the functionality of the REST Nu based interface is augmented.

Feature negotiation shall take place during the first interaction (i.e. HTTP POST in provisioning) between the PFDF and the SCEF. The client shall include in the HTTP request the set of supported features as follows:

- if a feature is required for the proper operation of the application, it shall be included within the 3gpp-Required-Features header;
- if a feature is optional for the proper operation of the application, it shall be included within the 3gpp-Optional-Features header.

The server shall include, within the 3gpp-Accepted-Features header in the HTTP response, the set of features it supports in common with the client.

If the server does not support any of the required features advertised by the client within the 3gpp-Required-Features header, the server shall reject the HTTP request with an HTTP 412 Precondition Failed status code and shall include the commonly supported features with the client within the 3gpp-Accepted-Features.

If the server requires certain features to be supported that are not advertised by the client, the server shall reject the HTTP request with an HTTP 412 Precondition Failed status code and shall include the commonly supported features with the client within the 3gpp-Accepted-Features and the required features in the 3gpp-required-features.

If the SCEF and the PFDF successfully negotiate supported features, the list of commonly supported features shall be applicable for the lifetime of the application. Features that are not advertised as supported shall not be used.

The sender may send information that is related to the supported features. Any unrecognized information shall be ignored by the receiver.

The table below defines the features applicable to the Nu interface.

Table 5.3.6.1-1: Features used in Nu Interface

Feature	M/O	Description
DomainNameProtocol	O	This feature supports the additional protocol matching condition for the domain name in PFD data.
PfdMgmtNotification	O	This feature supports the PFD management notification.
Feature: A short name for the feature to which the M/O and description pertain. M/O: Indication on whether the implementation of the feature is mandatory ("M") or optional ("O") in this 3GPP Release. Description: Textual description of the feature.		

NOTE: The base functionality for the Nu interface is defined in the Release-14 version of this specification and a feature is an extension of that functionality. The negotiation of supported features allows interworking between the endpoints of the Nu interface whereby each entity may support all, some, or none of the features that the Nu application can support defined in this specification. Features are defined so that they are independent of each other. Any introduced feature is explicitly defined in this specification.

Since a post Release-16 version of SCEF does not send any supported feature information, the interworking PFDF supporting Release-16 version does not need to send any information related to supported feature and both sides shall behave as specified in the Release-14 version of this specification.

A post Release-16 version of PFDF may receive supported feature information from a Release-16 version of SCEF, those information will be ignored by the PFDF and both sides shall behave as specified in the Release-14 version of this specification.

5.3.6.2 HTTP custom headers

This clause defines any new HTTP custom headers introduced by this specification.

5.3.6.2.1 3gpp-Optional-Features

This header is used by the client to advertise the optional features that are supported by the client.

The encoding of the header follows the ABNF as defined in IETF RFC 7230 [13].

3gpp-Optional-Features = "3gpp-Optional-Features" ":" 1#token

An example is: 3gpp-Optional-Features: feature1, feature2

5.3.6.2.2 3gpp-Required-Features

This header is used by the client to announce the mandatory features that must be supported in the server.

This header is also used by the server to indicate the missing features that must be supported in the client.

The encoding of the header follows the ABNF as defined in IETF RFC 7230 [13].

3gpp-Required-Features = "3gpp-Required-Features" ":" 1#token

An example is: 3gpp-Required-Features: feature1, feature2

5.3.6.2.3 3gpp-Accepted-Features

The header is used by the server to confirm the commonly supported set of features with the client.

The encoding of the header follows the ABNF as defined in IETF RFC 7230 [13].

3gpp-Accepted-Features = "3gpp-Accepted-Features" ":" 1#token

An example is: 3gpp-Accepted-Features: feature1, feature2

5.4 Specific application communication

5.4.1 General

Specific application communication represents the presentation of application data structures by transforming data into the form that the application accepts. It establishes the context between application-layer entities.

NOTE: This release only supports the content type JSON.

5.4.2 Content type

The body of HTTP messages shall be in JSON format. The content of the JSON text is defined in clause 5.4.3 and Annex A.

The MIME media type that shall be used within the Content-Type header field is "application/json" as defined in IETF RFC 7159 [11].

5.4.3 JSON provisioning fields

5.4.3.1 General

Table 5.4.3.1-1 describes the JSON provisioning fields used within the body of the HTTP messages representing the PFDs information associated with an application identifier. The table includes the information about the name of the field and the type of the fields.

Table 5.4.3.1-1: Nu Provisioning JSON fields

Field Name	Clause defined	JSON Value Type (NOTE 1)	JCR Type (NOTE 2)	Applicability (NOTE 4)
application-identifier	3GPP TS 29.251 [9]	string	string	
allowed-delay	3GPP TS 29.251 [9]	number	uint64	
pfds	3GPP TS 29.251 [9]	array	array	
pfd-identifier	3GPP TS 29.251 [9]	string	string	
flow-descriptions	3GPP TS 29.251 [9]	array	array	
urls	3GPP TS 29.251 [9]	array	array	
domain-names	3GPP TS 29.251 [9]	array	array	
removal-flag (NOTE 3)	3GPP TS 29.251 [9]	boolean	boolean	
partial-flag (NOTE 3)	3GPP TS 29.251 [9]	boolean	boolean	
dn-protocol	3GPP TS 29.251 [9]	string	string	DomainNameProtocol
scef-notification-uri	5.4.3.2	string	string	PfdMgmtNotification
NOTE 1: The basic JSON value types are defined in IETF RFC 7159 [11]. NOTE 2: The JCR types are defined in IETF draft-newton-json-content-rules [12]. NOTE 3: Only one of the removal-flag and the partial-flag for the application identifier shall be set to true. NOTE 4: Fields marked with a supported feature are applicable as described in clause 5.3.6.				

5.4.3.2 scef-notification-uri

The scef-notification-uri is of type string indicating the SCEF sent URI for receiving notifications.

5.4.4 Void

5.4.5 JSON errors and informational response fields

5.4.5.1 General

Table 5.4.4.1-1 describes the JSON fields defined for the errors and informational responses including their types and the field names.

Table 5.4.5.1-1: JSON fields for errors and informational response

Field Name	Clause defined	JSON Value Type (NOTE 1)	JCR Type (NOTE 2)	Applicability (NOTE 5)
errors	3GPP TS 29.251 [9] (NOTE 4)	array	array	
error-type	3GPP TS 29.251 [9] (NOTE 4)	string	"application" "interface" "server" "other" (NOTE 3)	
error-message	3GPP TS 29.251 [9]	string	string	
error-tag	3GPP TS 29.251 [9]	string	string	
error-path	3GPP TS 29.251 [9]	string	string	
error-info	3GPP TS 29.251 [9]	object	object	
success-message	3GPP TS 29.251 [9]	string	string	
success-path	3GPP TS 29.251 [9]	string	string	
success-info	3GPP TS 29.251 [9]	object	object	

NOTE 1: The basic JSON value types are defined in IETF RFC 7159 [11].
NOTE 2: The JCR types are defined in IETF draft-newton-json-content-rules [
NOTE 3: The quoted strings for a string type.
NOTE 4: The error is sent from the PFDF to the SCEF.
NOTE 5: Fields marked with a supported feature are applicable as described i

5.4.6 JSON report fields

5.4.6.1 General

Table 5.4.6.1-1 describes the JSON fields defined for the report information objects which are included in the error-info field.

Table 5.4.6.1-1: JSON fields for Report from the PFDF

Field Name	Clause defined	JSON Value Type (NOTE 1)	JCR Type (NOTE 2)	Applicability (NOTE 4)
pfd-reports	5.4.6.2	array	array	
application-ids	5.4.6.3	array	array	
pfd-failure-code	3GPP TS 29.251 [9] (NOTE 3)	string	string	
caching-time	3GPP TS 29.251 [9]	number	uint64	

NOTE 1: The basic JSON value types are defined in IETF RFC 7159 [11].
NOTE 2: The JCR types are defined in IETF draft-newton-json-content-rules [1
NOTE 3: The additional failure reasons for Nu are "TOO_SHORT_ALLOWED_
NOTE 4: Fields marked with a supported feature are applicable as described i

5.4.6.2 pfd-reports

The pfd-reports field is of type array and it contains a list of pfds reports.

The pdf-reports field can be used in an HTTP response to report failures in the installation/modification of PFDs. In this case, it shall be provided as a sub field of the error-info field.

Multiple pfd report instances shall be provided within the pfd-reports field if different pfd-failure-code values are applicable within the same HTTP response.

caching-time field is included if the pfd-failure-code is set to the value "TOO_SHORT_ALLOWED_DELAY".

A report instance shall contain application-ids and pfd-failure-code fields.

The JCR format for the pfd-reports is:

$pfd-reports = "pfd-reports" : [

{ $application-ids,

$pfd-failure-code,

$caching-time,

} +

]

5.4.6.3 application-ids

The application-ids field is of type array of string, and contains one or more application identifiers to which the PFDs belong.

5.4.7 JSON notification fields

5.4.7.1 General

Table 5.4.7.1-1 describes the JSON notification fields used within the body of the HTTP messages. The table includes the information about the name of the field and the type of the fields.

Table 5.4.7.1-1: Nu Notification JSON fields

Field Name	Clause defined	JSON Value Type (NOTE 1)	JCR Type (NOTE 2)	Applicability (NOTE 3)	
notification-pfd-reports	5.4.7.2	array	array	PfdMgmtNotification	
application-ids	5.4.6.3	array	array	PfdM	
pfd-failure-code	3GPP TS 29.251 [9] (NOTE 4)	string	string	PfdM	
user-plane-location-area	5.4.7.3	object	object	PfdM	
cell-ids	5.4.7.4	array	array	PfdM	
enodeb-ids	5.4.7.5	array	array	PfdM	
extended-enodeb-ids	5.4.7.6	array	array	PfdM	
routing-area-ids	5.4.7.7	array	array	PfdM	
tracking-area-ids	5.4.7.8	array	array	PfdM	

NOTE 1: The basic JSON value types are defined in IETF RFC 7159 [11].
NOTE 2: The JCR types are defined in IETF draft-newton-json-content-rules [12].
NOTE 3: Fields marked with a supported feature are applicable as described in clause 5.3.6.
NOTE 4: The additional failure reasons for Nu are "PARTIAL_FAILURE".

5.4.7.2 notification-pfd-reports

The notification-pfd-reports field is of type array and it contains a list of pfds reports.

The notification-pdf-reports field can be used in an HTTP POST to notify the failures in the installation/modification of PFDs. In this case, it shall be included in the body of the meessage.

Multiple pfd report instances shall be provided within the notification-pfd-reports field if different pfd-failure-code values are applicable within the same HTTP response.

user-plane-location-area field is included if the pfd-failure-code is set to the value "PARTIAL_FAILURE".

A report instance shall contain application-ids and pfd-failure-code fields.

The JCR format for the notification-pfd-reports is:

```
$notification-pfd-reports = "notification-pfd-reports" : [
  {    $application-ids,
       $pfd-failure-code,
       $user-plane-location-area ?
    } +
]
```

5.4.7.3 user-plane-location-area

The user-plane-location-areas field is of type object and it contains location areas of the user plane(s) which can not enforce the PFD(s) successfully.

The JCR format for the user-plane-location-area is:

```
$user-plane-location-area =  // : {
                                $cell-ids ?,
                                $enodeb-ids ?,
                                $extended-enodeb-ids ?,
                                $routing-area-ids ?,
                                $tracking-area-id ?
                                }
```

5.4.7.4 cell-ids

The cell-ids is of type string array and it contains a list of Cell Global Identities served by the user plane(s) which can not enforce the PFD(s) successfully. The content of the string has the same encoding as defined in clause 8.21.1 of 3GPP TS 29.274 [19].

5.4.7.5 enodeb-ids

The enodeb-ids is of type string array and it contains a list of eNodeB identities served by the user plane(s) which can not enforce the PFD(s) successfully. The content of the string has the same encoding as defined in clause 8.21.7 of 3GPP TS 29.274 [19].

5.4.7.6 extended-enodeb-ids

The enodeb-ids is of type string array and it contains a list of eNodeB identities served by the user plane(s) which can not enforce the PFD(s) successfully. The content of the string has the same encoding as defined in clause 8.21.8 of 3GPP TS 29.274 [19].

5.4.7.7 routing-area-ids

The routing-area-ids is of type string array and it contains a list of Routing Area Identities served by the user plane(s) which can not enforce the PFD(s) successfully. The content of the string has the same encoding as defined in clause 8.21.3 of 3GPP TS 29.274 [19].

5.4.7.8 tracking-area-ids

The tracking-area-ids is of type string array and it contains a list of Tracking Area Identities served by the user plane(s) which can not enforce the PFD(s) successfully. The content of the string has the same encoding as defined in clause 8.21.4 of 3GPP TS 29.274 [19].

5.5 PFDF discovery

The PFDF URI may be pre-configured on the SCEF.

The SCEF may select the PFDF by this configuration.

5.6 SCEF discovery

The SCEF notification URI may be pre-configured on the PFDF.

The SCEF notification URI may also be sent to the PFDF in the PFD provisioning.

The PFDF may select the SCEF by the pre-configuration or use the notification URI received in the PFD provisioning.

6 Secure communication

Either the NDS/IP network layer security defined in 3GPP TS 33.210 [5] or HTTP over TLS as defined in IETF RFC 2818 [6] should be used to secure communication over the REST based Nu interface.

Annex A (informative): JSON Schema

A.1 Provisioning schema

This clause defines the JSON schema for the body of HTTP request providing the provisioned PFDs. The schema is based on IETF draft-newton-json-content-rules [12] and is defined below:

```
# jcr-version 0.7

# ruleset-id 3gpp.nuapplication.provisioning

# import 3gpp.gwapplication.pfds as pfds


; JCR representing the PFDs provisioning data

$provisioning-root = @{root}{

  $application-identifier,

  $removal-flag ?,

  $partial-flag ?,

  $allowed-delay ?,

  $pfds ?,

  $scef-notification-uri ?

}


; An array list of the PFDs for multiple application identifiers

$pfds-array-root = @{root} [ $provisioning-root * ]


; The detected application traffic identifier for the PFDs

$application-identifier = "application-identifier" : string


; The allowed delay time for the PFDs deployment

$allowed-delay = "allowed-delay" : uint64


; The PFDs associated with the same application identifier

$pfds = "pfds" : [ $pfd * ]

; The PFD content

$pfd = {

  $pfd-identifier,

  ( $flow-descriptions | $urls | $domain-names | // : any) ?,
```

```
  $pfds.dn-protocol ?
}

; The PFD identifier
$pfd-identifier = "pfd-identifier" : string

; The flow descriptions
$flow-descriptions = "flow-descriptions" : [ string + ]

; The url matching expressions
$urls = "urls" :  [ string + ]

; The domain name match criteria
$domain-names = "domain-names" : [ string + ]

; A flag indicates whether this is a removal or not
$removal-flag = "removal-flag" : boolean

; A flag indicates whether this is a partial update or not
$partial-flag = "partial-flag" : boolean

; The URI in the SCEF for receiving notifications from the PFDF
$scef-notification-uri = "scef-notification-uri" : string
```

A.2 Error and Informational response schema

This clause defines the JSON schema for the body of HTTP responses in case of errors or success. The schema is based on IETF draft-newton-json-content-rules [12] and is defined below:

```
# jcr-version 0.7
# ruleset-id 3gpp.nuapplication.info

; A JCR for the error/successful response body
; Errors information
$errors-root = @{root} { $errors }

; Success information
```

```
 $success-root = @{root} {
    $success-message,
    $success-path ?,
    $success-info ?
  }

; Resource fields definitions
; The list of errors returned in responses sent by the PCEF/TDF
$errors = "errors" : [
  {
     $error-type,
     $error-message,
     $error-tag ?,
     $error-path ?,
     $error-info ?
    } +
]

; The error type for an error. It can be one of 'application', 'interface', 'server' and 'other'.
$error-type = "error-type" : ( "application" | "interface" | "server" | "other" )

; The error text message
$error-message = "error-message" : string

; The error tag for a specific error
$error-tag = "error-tag" : string

; A JSON pointer path to the error resource
$error-path = "error-path" : string

; Any additional information for the error
$error-info = "error-info" : {
  $pfd-reports ?,
  // : any *
}
```

```
; Report fields definitions
; The list of pfd reports sent to the SCEF
$pfd-reports = "pfd-reports" : [
  {
      $application-ids,
      $pfd-failure-code,
      $caching-time
    } +
]
; The application identifiers for the PFDs
$application-ids = "application-ids" : [string+]

; The string format for the pfd failure code
$pfd-failure-code =: (
    "MALFUNCTION" |
    "RESOURCES_LIMITATION" |
    "TOO_SHORT_ALLOWED_DELAY" |
    "PARTIAL_FAILURE" |
    "OTHER_REASON"
)

; The caching time for the PFDs
$caching-time = "caching-time" : uint64

; The successful text message
$success-message = "success-message" : string

; A JSON pointer path to the success resource
$success-path = "success-path" : string

; Any additional information for the success.
$success-info = "success-info" : {  // : any *}
```

A.3 PFD management notification schema

This clause defines the JSON schema for the body of HTTP request providing the notification of PFD management. The schema is based on IETF draft-newton-json-content-rules [12] and is defined below:

```
# jcr-version 0.7

# ruleset-id 3gpp.nuapplication.notification


; A JCR for the notification body

$notification-root = @{root} {

   $notification-pfd-reports

  }

; Report fields definitions

; The list of pfd reports sent to the SCEF

$notification-pfd-reports = "notification- pfd-reports" : [

  {

      $application-ids,

      $pfd-failure-code,

      $user-plane-location-area ?

   } +

]


; The application identifiers for the PFDs

$application-ids = "application-ids" : [string+]


; The string format for the pfd failure code

$pfd-failure-code =: (

   "MALFUNCTION" |

   "RESOURCES_LIMITATION" |

   "PARTIAL_FAILURE" |

   "OTHER_REASON"

)


; The location area of user planes which can not enforce the PFDs successfully

$user-plane-location-area =  {

     $cell-ids ?,

     $enodeb-ids ?,
```

```
    $extended-enodeb-ids ?,
    $routing-area-ids ?,
    $tracking-area-ids ?
}

; The cell IDs served by the user planes which can not enforce the PFDs successfully
$cell-ids = "cell-ids" : [ string + ]

; The eNodeB IDs served by the user planes which can not enforce the PFDs successfully
$enodeb-ids = "enodeb-ids" : [ string + ]

; The extended eNodeB IDs served by the user planes which can not enforce the PFDs successfully
$extended-enodeb-ids = "extended-enodeb-ids" : [ string + ]

; The Routing Area IDs served by the user planes which can not enforce the PFDs successfully
$routing-area-ids = "routing-area-ids" : [ string + ]

; The Tracking Area IDs served by the user planes which can not enforce the PFDs successfully
$tracking-area-ids = "tracking-area-ids" : [ string + ]
```

Annex B (informative): Call Flows

B.1 General

This annex describes the procedures for the interactions between the PFDF and the SCEF.

B.2 Provisioning of PFDs

This clause describes the signalling flow for the Provisioning of PFDs.

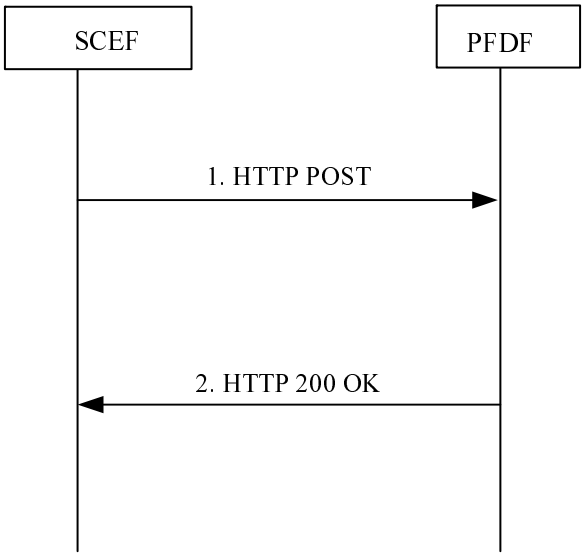

Figure B.2.1: Provisioning of PFDs

1. The SCEF sends the HTTP POST to the PFDF to indicate the creation, modification or deletion of PFDs for one or more application identifier(s) including the parameters defined in clause 5.3.5.2.
2. The PFDF sends the HTTP 200 OK response to the SCEF including the parameters defined in clause 5.3.5.2.

Annex C (informative): Change history

Date	TSG #	TSG Doc.	CR	Rev	Cat	Subject/Comment	New
2016-10						TS skeleton of Nu reference point stage 3.	0.0.0
2016-10						Inclusion of C3-163247, C3-163248, C3-163322 and editorial change from Rapporteur.	0.1.0
2016-12						Inclusion of C3-164069, C3-164227 and editorial change from Rapporteur.	0.2.0
2017-01						Inclusion of C3-170053,C3-17 0054, C3-170056 and editorial change from Rapporteur.	0.3.0
2017-02						Inclusion of C3-171187, C3-171324, and editorial change from Rapporteur.	0.4.0
2017-04						Inclusion of C3-172121, C3-172193, C3-172278, and editorial change from Rapporteur.	0.5.0
2017-05						Inclusion of C3-173200, C3-173202, C3-173332, C3-173333 and editorial change from Rapporteur.	0.6.0
2017-06	CT#76	CP-171145				TS sent to plenary for information and approval	1.0.0
2017-06	CT#76	CP-171145				TS approved at plenary	14.0.0
2017-09	CT#77	CP-172047	0001	2	F	Caching time over the Nu interface	14.1.0
2017-09	CT#77	CP-172047	0002	2	F	Clarification of partial update	14.1.0
2017-09	CT#77	CP-172047	0003	1	F	Correct the example of the PFD provisioning	14.1.0
2017-09	CT#77	CP-172047	0004	1	F	Update the reference of HTTP 1.1	14.1.0
2017-09	CT#77	CP-172047	0007	-	F	PFD handling for Push or Combination mode in PFDF	14.1.0
2017-12	CT#78	CP-173100	0008	-	F	JSON example correction	14.2.0
2018-06	CT#80					Automatic upgrade from previous Release	15.0.0
2019-03	CT#83	CP-190131	0010	2	B	Supported feature negotiation	16.0.0
2019-03	CT#83	CP-190133	0012	2	B	PFD extension	16.0.0
2019-06	CT#84	CP-191097	0015	-	A	Reference update: draft-newton-json-content-rules	16.1.0
2019-06	CT#84	CP-191100	0016	-	F	Encoding of custom header fields	16.1.0
2019-06	CT#84	CP-191102	0018	1	B	PFD management notification	16.1.0
2019-12	CT#86	CP-193220	0019	3	B	PFD partial failure notification	16.2.0
2022-03	SA#95e	-	-	-		Update to Rel-17 version (MCC)	17.0.0
2024-03	SA#103	-	-	-		Update to Rel-18 version (MCC)	18.0.0
2025-09	SA#109	-	-	-		Update to Rel-19 version (MCC)	**19.0.0**

History

Document history		
V19.0.0	October 2025	Publication