

# ETSI TS 129 250 V14.3.0 (2019-07)



**LTE;**  
**Nu reference point between SCEF and PFDF for sponsored**  
**data connectivity**  
**(3GPP TS 29.250 version 14.3.0 Release 14)**



---

**Reference**RTS/TSGC-0329250ve30

---

**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 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° 7803/88

---

**Important notice**

---

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at [www.etsi.org/deliver](http://www.etsi.org/deliver).

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommitteeSupportStaff.aspx>

---

**Copyright Notification**

---

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2019.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.

**3GPP™** and **LTE™** 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.

---

## Intellectual Property Rights

### Essential patents

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

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

### Trademarks

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

---

## 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 under <http://webapp.etsi.org/key/queryform.asp>.

---

## Modal verbs terminology

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

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

# Contents

|   |           |
|---|-----------|
| Intellectual Property Rights .....                        | 2         |
| Legal Notice .....  | 2         |
| Modal verbs terminology.....                              | 2         |
| Foreword.....   | 5         |
| 1 Scope .....   | 6         |
| 2 References .....  | 6         |
| 3 Definitions and abbreviations.....                      | 7         |
| 3.1 Definitions .....                                     | 7         |
| 3.2 Abbreviations .....                                   | 7         |
| 4 Nu reference point .....                                | 7         |
| 4.1 Overview .....  | 7         |
| 4.2 Nu reference model .....                              | 7         |
| 4.3 Functional elements.....                              | 8         |
| 4.3.1 PFD.....  | 8         |
| 4.3.2 SCEF.....   | 8         |
| 4.4 Procedures over Nu reference point .....              | 8         |
| 4.4.1 Management of PFD .....                             | 8         |
| 5 Nu protocol.....  | 9         |
| 5.1 Introduction .....                                    | 9         |
| 5.2 Transport layer .....                                 | 9         |
| 5.3 Application delivery layer .....                      | 10        |
| 5.3.1 General.....  | 10        |
| 5.3.2 HTTP status codes .....                             | 10        |
| 5.3.3 Methods .....                                       | 10        |
| 5.3.4 Resources and URI design.....                       | 10        |
| 5.3.5 HTTP request/response formats.....                  | 11        |
| 5.3.5.1 General .....                                     | 11        |
| 5.3.5.2 POST /nuapplication/provisioning .....            | 11        |
| 5.4 Specific application communication .....              | 13        |
| 5.4.1 General.....  | 13        |
| 5.4.2 Content type.....                                   | 13        |
| 5.4.3 JSON provisioning fields.....                       | 13        |
| 5.4.3.1 General .....                                     | 13        |
| 5.4.4 Void .....  | 14        |
| 5.4.5 JSON errors and informational response fields ..... | 14        |
| 5.4.5.1 General .....                                     | 14        |
| 5.4.6 JSON report fields .....                            | 14        |
| 5.4.6.1 General .....                                     | 14        |
| 5.4.6.2 pfd-reports.....                                  | 15        |
| 5.4.6.3 application-ids.....                              | 15        |
| 5.5 PFD discovery.....                                    | 15        |
| 6 Secure communication .....                              | 15        |
| <b>Annex A (informative): JSON Schema.....</b>            | <b>16</b> |
| A.1 Provisioning schema.....                              | 16        |
| A.2 Error and Informational response schema.....          | 17        |
| <b>Annex B (informative): Call Flows .....</b>            | <b>20</b> |
| B.1 General .....   | 20        |
| B.2 Provisioning of PFDs .....                            | 20        |

**Annex C (informative):**    **Change history** .....21  
History .....22

---

# Foreword

This Technical Specification has been produced by the 3<sup>rd</sup> 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".

---

## 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 3<sup>rd</sup> 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 subclause 3a of 3GPP TS 29.213 [4].

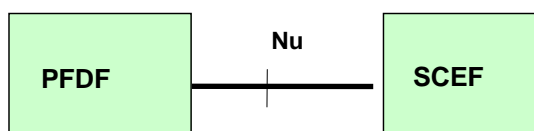


Figure 4.2.1: Nu reference model



## 4.3 Functional elements

### 4.3.1 PFD

The PFD (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 PFD 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 3<sup>rd</sup> party SCS/AS. The SCEF may provision the PFDs to the PFD 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 PFD via the Nu reference point 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 PFD, translate the external application identifier(s) to the application identifier(s) known at the PFD; 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 PFD including the provisioned PFD changes for the the application identifier (s) within the body of the HTTP POST as described in subclause 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 PFD. The external application identifier can be the same as the application identifier known at the PFD.

Upon receipt of the HTTP request for the provisioning operation from the SCEF, the PFD 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 PFD 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 PFD cannot ensure the PCEF/TDF will pull the PFDs in time (i.e. allowed delay is shorter than the caching time), the PFD 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 PFD 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 PFD 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) where no flag is provided;
- acknowledge the HTTP POST message by sending a corresponding HTTP response with the appropriate status code as defined in subclause 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.

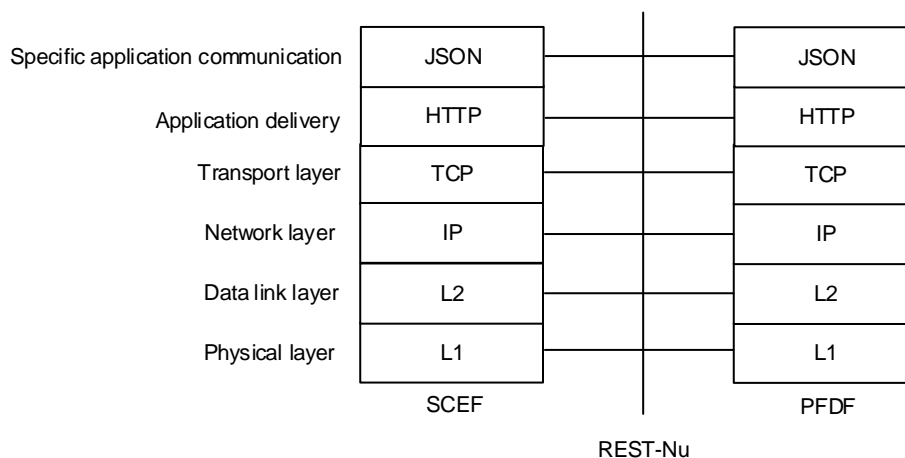
## 5 Nu protocol

### 5.1 Introduction

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

- 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 subclause 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. 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://pfdserver.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 subclauses 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<br>(NOTE 1) | 5.3.5.2        | Content-Type:<br>application/json<br>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. |
| 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. |   |                |   |           |  |

### 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 subclause 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 subclause 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 subclause 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 subclause 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: pfdserver.example.com

Content-Type: application/json

Content-Length: ...

```
[
  {
    "application-identifier":"test-application-1",
    "removal-flag":true
  },
  {
    "application-identifier":"test-application-2",
    "allowed-delay":600,
    "pfd":[
      {
        "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*)?$"
        ]
      }
    ]
  },
  {
    "application-identifier":"test-application-3",
    "partial-flag":true,
    "pfd":[
      {
        "pfd-identifier":"pfd3",
        "urls":[
          "^http://test.example2.net(/\\S*)?$"
        ]
      }
    ]
  }
]
```

```
]
},
{
  "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: pfdserver.example.com

Content-Type: application/json

```
{
  "success-message": "Notification was processed successfully.",
}
```

## 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 subclause 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) |
|--|--------------------|--------------------------|-------------------|
| 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           |
| 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. |                    |                          |                   |

## 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      | Subclause defined           | JSON Value Type (NOTE 1) | JCR Type (NOTE 2)   |
|-----------------|-----------------------------|--------------------------|---|
| errors          | 3GPP TS 29.251 [9] (NOTE 4) | array                    | array   |
| error-type      | 3GPP TS 29.251 [9] (NOTE 4) | string                   | "application"<br>"interface"<br>"server"<br>"other"<br>(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 [12].  
NOTE 3: The quoted strings for a string type.  
NOTE 4: The error is sent from the PFDf to the SCEF.

## 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       | Subclause defined           | JSON Value Type (NOTE 1) | JCR Type (NOTE 2) |
|------------------|-----------------------------|--------------------------|-------------------|
| pdf-reports      | 5.4.6.2                     | array                    | array             |
| application-ids  | 5.4.6.3                     | array                    | array             |
| pdf-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 [12]  
NOTE 3: The additional failure reason for Nu is "TOO\_SHORT\_ALLOWED\_DELAY".

### 5.4.6.2 pdf-reports

The pdf-reports field is of type array and it contains a list of pdfs 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 pdf report instances shall be provided within the pdf-reports field if different pdf-failure-code values are applicable within the same HTTP response.

caching-time field is included if the pdf-failure-code is set to the value "TOO\_SHORT\_ALLOWED\_DELAY".

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

The JCR format for the pdf-reports is:

```
$pdf-reports = "pdf-reports" : [
```

```
  {
    $application-ids,
    $pdf-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.

## 5.5 PFDF discovery

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

The SCEF may select the PFDF by this configuration.

---

## 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 subclause 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
```

```
; JCR representing the PFDs provisioning data
```

```
$provisioning-root = @{root}{
```

```
  $ application-identifier,
```

```
  $removal-flag ?,
```

```
  $partial-flag ?,
```

```
  $allowed-delay ?,
```

```
  $pfd ?
```

```
}
```

```
; An array list of the PFDs for multiple application identifiers
```

```
$pfd-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
```

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

```
; The PFD content
```

```
$pfd = {
```

```
  $pfd-identifier,
```

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

```
}
```

; 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
```

---

## A.2 Error and Informational response schema

This subclause 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" |

"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 \* }

---

## Annex B (informative): Call Flows

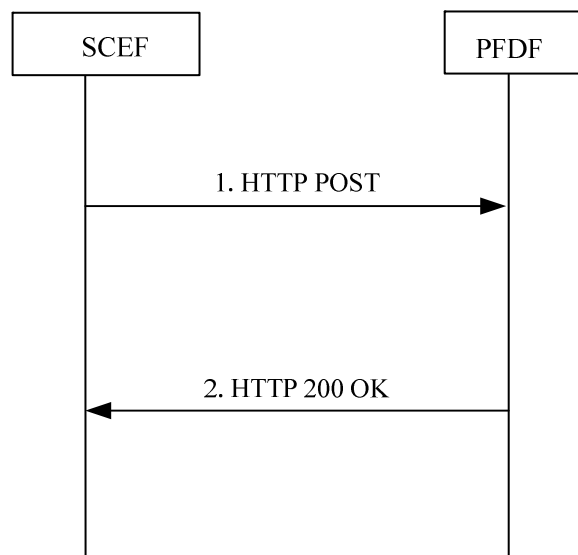
### B.1 General

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

---

### B.2 Provisioning of PFDFs

This subclause describes the signalling flow for the Provisioning of PFDFs.



**Figure B.2.1: Provisioning of PFDFs**

1. The SCEF sends the HTTP POST to the PFDF to indicate the creation, modification or deletion of PFDFs for one or more application identifier(s) including the parameters defined in subclause 5.3.5.2.
2. The PFDF sends the HTTP 200 OK response to the SCEF including the parameters defined in subclause 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 |
| 2019-06 | CT#84 | CP-191097 | 0013 | -   | F   | Reference update: draft-newton-json-content-rules   | 14.3.0 |

---

# History

| <b>Document history</b> |              |             |
|-------------------------|--------------|-------------|
| V14.0.0                 | July 2017    | Publication |
| V14.1.0                 | October 2017 | Publication |
| V14.2.0                 | January 2018 | Publication |
| V14.3.0                 | July 2019    | Publication |
|                         |              |             |