ETSI TS 129 251 V14.2.0 (2018-01)

LTE;
Gw and Gwn reference point for sponsored data connectivity
(3GPP TS 29.251 version 14.2.0 Release 14)

Reference
RTS/TSGC-0329251ve20

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

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

Foreword

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

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

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

Modal verbs terminology

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

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

Contents

Foreword

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

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

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document provides the stage 3 specification of the Gw and Gwn reference points. The functional requirements and the stage 2 specifications of the Gw and Gwn reference points are specified in 3GPP TS 23.203 [2]. The Gw reference point lies between the Packet Flow Description Function (PFDF) and the Policy and Charging Enforcement Function (PCEF). The Gwn reference point lies between the Packet Flow Description Function (PFDF) and the Traffic Detection Function (TDF).

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 23.203: “Policy and charging control architecture”.

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

[4] 3GPP TS 23.682: "Architecture enhancements to facilitate communications with packet data networks and applications".

[5] 3GPP TS 29.212: "Policy and Charging Control (PCC); Reference points".

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

[7] Void.

[8] 3GPP TS 33.210: "3G security; Network Domain Security (NDS); IP network layer security".

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

[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-08: "A Language for Rules Describing JSON Content".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

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

[14] 3GPP TS 29.250: "Nu reference point between SCEF and PFDF for sponsored data connectivity".

[15] 3GPP TS 29.155: "Traffic steering control; Representational state transfer (REST) over St reference point".

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

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

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

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

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

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

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 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 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 [2]).

Pull mode: A mode used between the PCEF/TDF and the PFDF where the PFDs are sent by the PFDF at reception of an HTTP request from the PCEF/TDF.

Push mode: A mode used between the PCEF/TDF and the PFDF where the PFDs and/or notification of PFD deletion are sent by the PFDF in an HTTP request to the PCEF/TDF.

Combination mode: A mode used between the PCEF/TDF and the PFDF where both pull and push exist. This mode allows the PFDF to send the PFDs at reception of an HTTP request from the PCEF/TDF and provision PFD changes (creation/update/deletion) and/or notification of PFD creation/update/deletion in the HTTP request to the PCEF/TDF.

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 Gw reference point

4.1 Overview

The Gw reference point is located between the Packet Flow Description Function (PFDF) and the Policy and Charging Enforcement Function (PCEF). The Gw reference point is used for provisioning and removal of PFDs from the PFDF to the PCEF and reporting the handling result of PFDs from the PCEF to the PFDF.

The stage 2 level requirements for the Gw reference point are defined in 3GPP TS 23.203 [2].

4.2 Gw reference model

The Gw reference point is defined between the PFDF and the PCEF. 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 [3].

Figure 4.2.1: Gw reference model

4.3 Functional elements

4.3.1 PFDF

The PFDF (Packet Flow Description Function) is a functional element which stores PFDs associated with application identifier (s) and transfers them to the PCEF via Gw interface to enable the PCEF to perform accurate application detection when the PFDs are managed by a 3rd party service provider.

The PFDF receives PFDs for the corresponding application identifier (s) from the SCEF as defined in 3GPP TS 23.682 [4]and 3GPP TS 29.250 [14].

4.3.2 PCEF

The PCEF (Policy and Charging Enforcement Function) funtionality defined in subclause 4.4.2 of 3GPP TS 29.212 [5] shall be applied. In addition, the PCEF shall support the management of PFDs provisioned by the PFDF. The application detection filter may be extended with the PFDs provided by the PFDF. The new PFDs provided by the PFDF replace the existing ones in the PCEF.

4.4 Procedures over Gw reference point

When the PFDF is deployed and the management of PFDs is supported by the PCEF, the mode of PFDs management shall be consistently configured in one PLMN, it may be:

- Pull mode only, or
- Push mode only, or
- Combination mode.

4.4.1 Request for PFDs ("Pull mode" and "Combination mode")

At the time a PCC Rule with an application identifier for which PFDs are not available is activated or provisioned, or when the caching timer for an application identifier elapses and a PCC Rule for the application identifier is still active, the PCEF shall request all PFDs for the application identifier from the PFDF.

The PCEF shall send an HTTP GET message to the PFDF to retrieve the PFDs for one or more application identifier(s) as defined in subclauses 6.3.3.2, 6.3.3.3 and 6.3.3.4.

Upon receipt of the HTTP request for the pull operation, the PFDF shall respond to the PCEF indicating whether the pull operation was successful or not by using one of the HTTP status codes; if the pull operation is successful, the PFDF :

- shall provide the PFDs of the specified application identifier(s) from the request within the body of the HTTP response; and
- if there are caching time value(s) configured for any of the specified application identifier(s), shall set the caching-time to the caching time value(s) for those application identifier(s) in the body of the HTTP response.

The PCEF shall install, update or remove the received PFD(s) for the application identifier(s).

The PCEF shall also for each application identifier start a caching timer with a value

- according to the received caching time value for that application identifier, if such a value is received for that application identifier; or

- otherwise with a default caching time value configured in the PCEF.

NOTE 1: The PCEF(s) and the PFDF(s) within an operator network are configured with the same default caching time value to be applied for all application identifiers.

NOTE 2: In the combination mode, the HTTP POST message from the PFDF does not impact the running caching timer of an application identifier.

When the PCEF removes the last PCC rule that refers to the corresponding application identifier, or when the caching timer expires and no PCC rule refers to the application identifier, the PCEF may remove the PFD(s) related with the application identifier.

4.4.2 Provisioning of PFDs ("Push mode"" and "Combination mode")

The PFDF may create, update or remove one or multiple PFDs associated with application identifier(s) to the PCEF. When the request for creation, update, or removal for an application identifier(s) is received from the SCEF, for the Push mode, the PFDF shall for each PCEF it serves:

- immediately send to the PCEF an HTTP POST message including the provisioned changes of one or more application identifiers as specified in subclause 6.3.3.5, or

- wait for a period shorter than the allowed delay (e.g.to aggregate all the PFDs for several application) , then send to the PCEF an HTTP POST message including the provisioned changes.

For the Combination mode, the PFDF may for each PCEF it serves wait for a period shorter than the allowed delay (e.g. to aggregate all the PFDs for several application) and, if no HTTP GET request for that application identifier is received during the waiting time, then send an HTTP POST message including the provisioned changes and/or notification of PFD creation/update for that application identifier as specified in subclause 6.3.3.5.

NOTE 1: In the combination mode, the PFDF can check the received allowed delay against the caching time as specified in subclause 4.4.1 of 3GPP TS 29.250 [14], in order to utilize the pull procedure.

When the PCEF receives the HTTP POST message, the PCEF shall

- for an application identifier(s) where no flag is provided, remove the all existing all PFD(s) (if available) and install all the new provided PDF(s).

- for an application identifier(s) where the removal-flag is also provided and set to true, delete the existing PFD(s);

- for an application identifier(s) where partial-flag is also provided and set to true

  - install a new PFD(s) if the new PFD(s) with a new PFD identifier(s) is received,

  - update an existing PFD(s) if a new PFD(s) with the same PFD identifier(s) is recevied, and/or

  - delete an existing PFD(s) if the same PFD identifier(s) without any content is received;

- acknowledge the HTTP POST message by sending a corresponding HTTP response with the appropriate status code (200 OK for success) as described in subsclause 6.3.4; and

- retrieve the PFDs for the corresponding application identifier(s) as defined in subclause 4.4.1 either immediately if the allowed-delay is not provided, or within the provided allowed-delay for the application identifier(s) where the notification-flag is included and set to true.

NOTE 2: It depends on the implementation whether the PCEF initiates separate pull requests to retrieve the PFDs for different application identifier(s) within the allowed-delay. The decision to send PFD, and/or notification of PFD creation/update/deletion for certain application identifier on the PFDF is based on the operator policy.

NOTE 3: In the push mode, the PFDF can be configured with the list of PCEFs per application identifier(s).

NOTE 4: The caching timer is not applicable for the PCEF/TDF in Push mode.

5 Gwn reference point

5.1 Overview

The Gwn reference point is located between the Packet Flow Description Function (PFDF) and the Traffic Detection Function (TDF). The Gwn reference point enables transport of PFDs from the PFDF to the TDF for a particular Application Identifier or for a set of Application Identifiers.

The Gwn reference point supports the following functions:

- Creation, updating and removal of individual or the whole set of PFDs from the PFDF to the TDF.
- Confirmation of creation, updating and removal of PFDs from the TDF to the PFDF.

NOTE: The interaction between the PFDF and the TDF is not related to any IP-CAN session.

The stage 2 level requirements for the Gwn reference point are defined in 3GPP TS 23.203 [2].

5.2 Gwn reference model

The Gwn reference point is defined between the PFDF and the TDF. The relationships between the different functional entities involved are depicted in figure 5.2.1. The overall PCC architecture is depicted in subclause 3a of 3GPP TS 29.213 [3].

Figure 5.2.1: Gwn reference model

5.3 Functional elements

5.3.1 PFDF

Subclause 4.3.1 is applicable with the clarification that the TDF replaces the PCEF and Gwn interface replaces Gw interface.

5.3.2 TDF

The TDF (Traffic Detection Function) funtionality defined in subclause 4b.4.2 of 3GPP TS 29.212 [5] shall be applied. In addition, the TDF shall support the management of PFDs provisioned by the PFDF.The application detection filter may be extended with the PFDs provided by the PFDF.

5.4 Procedures over Gwn reference point

The procedures in subclause 4.4 are applicable with the clarification that the TDF replaces the PCEF in those procedures.

6 Gw/Gwn protocol

6.1 Introduction

The following layers of the protocol stack for the Gw/Gwn reference point between PCEF/TDF and PFDF are described in subclauses:

- TCP as defined in IETF RFC 793 [6] 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 7).
- The application delivery layer provides the transport of the specific application communication data using HTTP as defined in IETF RFC 7230 [16], IETF RFC 7231 [17], IETF RFC 7232 [18], IETF RFC 7233 [19], IETF RFC 7234 [20] and IETF RFC 7235 [21].
- The specific application communication layer constitutes the transport of the JSON content type.

Figure 6.1.1 illustrates the protocol stack of the RESTful Gw/Gwn reference point.

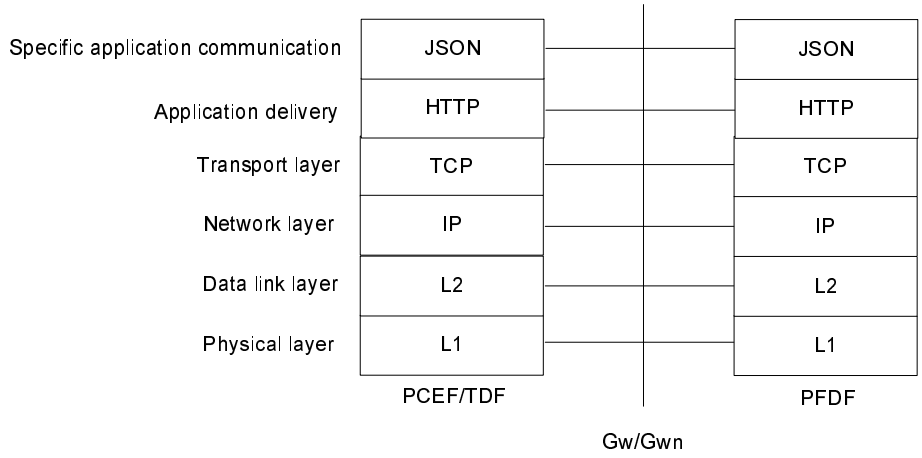

Figure 6.1.1: Protocol stack of the Gw/Gwn reference point

6.2 Transport layer

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

If the "Pull mode" is deployed, then the PCEF/TDF shall act as an HTTP client and the PFDF shall act as an HTTP server. In this case, the PCEF/TDF shall initiate a TCP connection with the PFDF.

If the "Push mode" is deployed, then the PFDF shall act as an HTTP client and the PCEF/TDF shall act as an HTTP server. In this case, the PFDF shall initiate a TCP connection with the PCEF/TDF.

If the "Combination mode" is deployed, then

- for pull request, the HTTP roles and TCP session establishment as described above for the "Pull mode" apply.
- for push request (with or without notification), the HTTP roles and TCP session establishment as described above for the "Push mode" apply.

6.3 Application delivery layer

The application delivery layer shall use RESTful HTTP.

The application delivery layer provides the following services:

- Retrieval of the PFDs from the PFDF
- Provisioning of the PFDs by the PFDF
- Notification sent from PFDF to the PCEF/TDF to indicate PFD changes related to application identifiers being created, updated or deleted

In order to retrieve the PFD for a specific application identifier, the PFDF shall send an HTTP GET message including the application identifier as a path element of the request URI as specified in subclause 6.3.3.2.

In order to retrieve the PFDs for a set of application identifiers, the PCEF/TDF shall send an HTTP GET message to the PFDF including the application identifier(s) as the query parameters in the URI as specified in subclause 6.3.3.3.

In order to retrieve all PFDs, the PCEF/TDF may omit the '?' and {query parameters} in the HTTP GET message's URI as specified in subclause 6.3.3.4.

If the PFDF needs to provision PFDs for a set of application identifier(s) (creation/update/deletion) to the PCEF/TDF, the PFDF shall send an HTTP POST message, which includes the provisioned changes and/or the notifications of impacted application identifiers as described in subclause 6.3.3.5.

6.3.1 Methods

Methods indicate to the server what action has to be performed. Every HTTP request message has a method. The following HTTP methods can be used:

- GET: Used by the PCEF/TDF to retrieve the PFDs information for a set of application identifier(s). The server returns the PFDs representation within the body of the response.
- POST: Used by the PFDF to provision or notify PFDs changes for a set of application identifiers(s). 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.

6.3.2 Resources and URI design

6.3.2.1 General

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 Gw/Gwn 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 Gw/Gwn 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 ["/" resourcepath]. In this release:

- "mainapp" is "gwapplication".
- "mainresource" is defined in subclauses 6.3.2.2 and 6.3.2.3.
- "resourcepath" contains the path to identify the PFDs resource for a specific application identifier.

6.3.2.2 URI design for Pull mode

In pull mode, the PFDs in PFDF is a readable RESTful resource:

- "mainresource" is "pfds".
- "resourcepath" varies and is defined in subclauses 6.3.3.2, 6.3.3.3 and 6.3.3.4.

An example of the URI to identify the RESTful resource is: http://pfdfserver.example.com/gwapplication/pfds.

NOTE: A different resource path can be used when the Resource URI is preconfigured in the PCEF/TDF.

6.3.2.3 URI design for Push mode

In push mode, the PFDs management (associating/disassociating PFDs with application identifiers) in PCEF/TDF is a controller resource that is responsible for processing push requests that involve provisioning and/or notifying a set of changes for more than one set of PFDs for corresponding application identifiers atomically.

- "mainresource" shall be "provisioning".
- "resourcepath" is not applicable in push mode.

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

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

6.3.3 HTTP request/response formats

6.3.3.1 General

The PFDs pull procedure is performed through HTTP transactions consisting of a request initiated by the PCEF/TDF and its corresponding response provided by the PFDF. While the PFDs push procedure is initiated by the PFDF and answered by the PCEF/TDF

Table 6.3.3.1-1 summarizes the content of the requests and responses. More detailed information is specified in the corresponding subclauses as indicated in the table.

Table 6.3.3.1-1: Gw/Gwn requests/response summary table

Method	Resource URI's path	Clause Defined	Request body	Initiator	Response body
GET	/gwapplication/pfds/{application-identifier} (NOTE 1)	6.3.3.2	None	PCEF /TDF	Successful response: The PFDF shall include the representation of the corresponding PDFs of the specified application identifier in the body of the response as per Annex A.1
GET	/gwapplication/pfds?{query-parameters} (NOTE 1)	6.3.3.3	None	PCEF /TDF	Successful response: The PFDF shall include the representation of the corresponding PDFs for a set of application identifier(s) in the body of the response as per Annex A.1
GET	/gwapplication/pfds (NOTE 1)	6.3.3.4	None	PCEF /TDF	Successful response: The PFDF shall include the representation of the corresponding PDFs for all application identifier(s) in the body of the response as per Annex A.1
POST	/gwapplication/provisioning (NOTE 2)	6.3.3.5	Content-Type: application/json The PFDF may include PFDs content and/or notification associated with application identifier(s) using the schema defined in Annex A.2.	PFDF	Successful response: The PCEF/TDF may include informational data in the body of the response as per Annex A.3
NOTE 1: A different path from /gwapplication/pfds/ may be used when it is configured in the PCEF/TDF. In that case the "path" part set in the different methods should use the configured one. NOTE 2: A different path from /gwapplication/provisioning may be used when it is configured in the PFDF. In that case the "path" part set in the different methods should use the configured one.					

6.3.3.2 GET /gwapplication/pfds/{application-identifier}

To retrieve the PFDs for a specific application identifier, the PCEF/TDF shall send an HTTP GET request to the PFDF as follows:

- the request URI formatted as defined in subclause 6.3.2 with the "path" part set to: /gwapplication/pfds/{application-identifier}, where the application-identifier is the application id with which the PFDs are associated.

Upon receipt of the HTTP GET, the PFDF shall respond to the PCEF/TDF indicating whether the querying of the resource was successful or not using one of the HTTP status codes as defined in subclause 6.3.4. If the resource exists, the PFDF shall respond with an HTTP 200 OK status code and include the PFDs representation state within the body of the response as defined in Annex A.1. If no resource was found, the PFDF shall respond with an HTTP 404 not found. The PFDF shall also include the Content-Type header field set to "application/json".

If the PCEF/TDF receives the HTTP 404 not found status code, the PCEF/TDF shall remove all the PFD(s) of the queried application identifier.

Below is an example of a corresponding HTTP GET:

GET /gwapplication/pfds/test-application-1 HTTP/1.1

Host: pfdfserver.example.com

Here is an example of a successful response :

HTTP/1.1 200 OK

Server: pfdfserver.example.com

Content-Type: application/json

Content-Length: …


```
{
  "application-identifier":"test-application-1",

  "caching-time":200000,
  "pfds":[
   {

      "pfd-identifier":"pfd1",
      "flow-descriptions":[
        "permit in ip from 10.68.28.39 80 to any",
        "permit out ip from any to 10.68.28.39 80"
      ]
    },
    {
      "pfd-identifier":"pfd2",
      "urls":[
        "^http://test.example.com(/\\S*)?$"
      ]
    }
  ]
}
```


6.3.3.3 GET /gwapplication/pfds?{query-parameters }

To retrieve the PFDs for a set of application identifier(s), the PCEF/TDF may send an HTTP GET request to the PFDF as follows:

- The request URI formatted as defined in subclause 6.3.2 with the "path" part set to: /gwapplication/pfds?{query-parameters}, where the query-parameters contains one or multiple application identifier(s) delimited by comma with a parameter name "application-identifiers". As an example: "application-identifiers=id1,id2,id3"
- Any "=" and "," for any application identifier in the query parameters shall be encoded as "%3D" and "%2C" in the URI.

Upon receipt of the HTTP request, the PFDF shall respond to the PCEF/TDF indicating whether the querying of the resources was successful or not using one of the HTTP status codes as defined in subclause 6.3.4. If at least one resource exists, the PFDF shall respond with an HTTP 200 OK status code and include a list of the PFDs associated with the found application identifier(s) within the body of the response as defined in Annex A.1. If the PFDs associated with a queried application-identifier are not provided in the response, it implicitly means that the corresponding application-identifier does not exist at the PFDF. If no resource was found, the PFDF shall respond with an HTTP 404 not found. The PFDF shall also include the Content-Type header field set to "application/json".

If the PCEF/TDF receives the 200 OK but one or more queried application identifier(s) is not provided in the response, the PCEF/TDF shall remove the PFD(s) of the queried application identifier(s) which is not included in the response.

If the PCEF/TDF receives the HTTP 404 not found status code, the PCEF/TDF shall remove all the PFD(s) of the queried application identifier(s).

Below is an example of this HTTP GET on two sets of PFDs query. But only one is found in the PFDF.

GET /gwapplication/pfds?application-identifiers=test-application-1,test-application-2 HTTP/1.1

Host: pfdfserver.example.com

Here is an example of a successful response :

HTTP/1.1 200 OK

Server: pfdfserver.example.com

Content-Type: application/json

Content-Length: …

```
[
  {
    "application-identifier":"test-application-1",
    "caching-time":200000,
    "pfds":[
      {
         "pfd-identifier":"pfd1",
        "flow-descriptions":[
          "permit in ip from 10.68.28.39 80 to any",
          "permit out ip from any to 10.68.28.39 80"
        ]
      },
      {
        "pfd-identifier":"pfd2",
        "urls":[
          "^http://test.example.com(/\\S*)?$"
        ]
      }
    ]
```

```
  }
]
```

6.3.3.4 GET /gwapplication/pfds

To retrieve PFDs associated with all application identifiers, the PCEF/TDF may send an HTTP GET request to the PFDF as follows:

- The request URI formatted as defined in subclause 6.3.2 with the "path" part set to: /gwapplication/pfds.

Upon receipt of the HTTP request, the PFDF shall respond to the PCEF/TDF indicating whether the querying of the resources was successful or not using one of the HTTP status codes as defined in subclause 6.3.4. If at least one resource exists, the PFDF shall respond with an HTTP 200 OK status code and include a list of the PFDs associated with the found application identifier(s) within the body of the response as defined in Annex A.1. If no resource was found, the PFDF shall respond with an HTTP 404 not found. The PFDF shall also include the Content-Type header field set to "application/json".

If the PCEF/TDF receives the 200 OK but one or more application identifier(s) which exists at the PCEF/TDF is not provided in the response, the PCEF/TDF shall remove the PFD(s) of the application identifier(s) which is not included in the response.

If the PCEF/TDF receives the HTTP 404 not found status code, the PCEF/TDF shall remove all the PFD(s) of all the identifier(s) which exist at the PCEF/TDF.

Below is an example of this HTTP GET.

GET /gwapplication/pfds HTTP/1.1

Host: pfdfserver.example.com

Here is an example of a successful response :

HTTP/1.1 200 OK

Server: pfdfserver.example.com

Content-Type: application/json

Content-Length: …

```
[
  {
    "application-identifier":"test-application-1",
    "caching-time":200000,
    "pfds":[
     {

       "pfd-identifier":"pfd1",
       "flow-descriptions":[
         "permit in ip from 10.68.28.39 80 to any",
         "permit out ip from any to 10.68.28.39 80"
       ]
```

```
      },
      {
        "pfd-identifier":"pfd2",
        "urls":[
          "^http://test.example.com(/\\S*)?$"
        ]
      }
    ]
  }
]
```

6.3.3.5 POST /gwapplication/provisioning

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

- The request URI formatted as defined in subclause 6.3.2 with the "path" part set to: /gwapplication/provisioning.
- The Content-Type header field set to "application/json"
- The body of the message encoded in JSON format as defined in Annex A.2. 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 install the new PFD(s ,
    - new PFD(s) with existing PFD identifier(s) to update the existing PFD(s), and/or
    - the existing PFD identifier(s) without any content to remove the PFD(s);
  - for the PFD(s) removal for an existing application identifier, the existing application identifier and the removal indication;
  - for the PFD(s) notification, the notification indication and an optional allowed delay time for the PCEF/TDF to pull the corresponding PFD(s) within the allowed delay time.

Upon receipt of the HTTP POST, the PCEF/TDF shall respond to the PFDF indicating whether the provisioning was successful or not using one of the HTTP status codes as defined in subclause 6.3.4. If the provisioning was accepted, the PCEF/TDF 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 provisioning was rejected, the PCEF/TDFshall indicate the reason using an appropriate HTTP status code for as defined in subclause 6.3.4 and optionally additional information in the body of the response as defined in Annex A.3. If the pfd-reports is included, the PCEF/TDF shall identify the failed PFDs by including the application-identifier and specify the failed reason code by including a pfd-failure-code field.

Depending on the value of the pfd-failure-code field, the PFDF may provision the PFDs again.

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

POST /gwapplication/provisioning HTTP/1.1

```
Host: pcefserver.example.com
Content-Type: application/json
Content-Length: …
[
  {
    "application-identifier":"test-application-1",
    "notification-flag":true,
    "allowed-delay":600
  },
  {
    "application-identifier":"test-application-2",
    "removal-flag":true
  },
  {
    "application-identifier":"test-application-3",
    "pfds":[
     {
        "pfd-identifier":"pfd1",
        "flow-descriptions":[
          "permit in ip from 10.68.28.39 80 to any"
        ]
      }
     ]
  },
  {
    "application-identifier":"test-application-4",
    "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, 23 Apr 2012 17:10:00 GMT
Server: pcefserver.example.com
Content-Type: application/json
{
  "success-message": "Notification was processed successfully.",
}
```


6.3.4 HTTP status codes

HTTP status codes are used as defined in IETF RFC 7231 [17]

6.3.5 Feature negotiation

6.3.5.1 General

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

Feature negotiation shall take place during the first interaction (either HTTP POST in push or HTTP GET in pull) between the PFDF and the PCEF/TDF. 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 PCEF/TDF 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 Gw interface.

Table 6.3.5.1-1: Features used in Gw Interface

Feature	M/O	Description
PartialUpdate	O	The PFDF can use this feature for partial update in PFD push.
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.		

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

Table 6.3.5.1-2: Features used in Gwn Interface

Feature	M/O	Description
PartialUpdate	O	The PFDF can use this feature for partial update in PFD push.
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 Gw/Gwn 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 Gw/Gwn interface whereby each entity may support all, some, or none of the features that the Gw/Gwn 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.

6.3.5.2 HTTP custom headers

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

6.3.5.2.1 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 [7].

3gpp-Optional-Features = "3gpp-Optional-Features" ":" 1#token
An example is: 3gpp-Optional-Features: feature1, feature2

6.3.5.2.2 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 [7].

3gpp-Required-Features = "3gpp-Required-Features" ":" 1#token
An example is: 3gpp-Required-Features: feature1, feature2

6.3.5.2.3 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 [7].

3gpp-Accepted-Features = "3gpp-Accepted-Features" ":" 1#token
An example is: 3gpp-Accepted-Features: feature1, feature2

6.3.5.3 Precedence for HTTP custom headers

The server shall evaluate the HTTP custom headers first if other conditional headers (e.g. If-None-Match, as defined in IETF RFC 7232 [18]) are also present.

6.4 Specific application communication

6.4.1 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

6.4.2 Content type

The HTTP POST requests in pull mode shall be an encoded string which is defined in subclause 6.3.3.4.

The HTTP responses in pull mode as well as the HTTP messages in push mode shall be in JSON format. The content of the JSON text is defined in subclause 6.4.3, 6.4.4 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].

6.4.3 JSON PFDs fields

6.4.3.1 General

Table 6.4.3.1.1 describes the JSON 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 6.4.3.1.1: Gw/Gwn PFD JSON fields

Field Name	Clause defined	JSON Value Type (NOTE 1)	JCR Type (NOTE 2)	Applicability (NOTE 3)
application-identifier	6.4.3.2	string	string	
caching-time	6.4.3.4	number	uint64	
pfds	6.4.3.5	array	array	
pfd-identifier	6.4.3.6	string	string	
flow-descriptions	6.4.3.7	array	array	
urls	6.4.3.8	array	array	
domain-names	6.4.3.9	array	array	
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 subclause 6.3.5.				

6.4.3.2 application-identifier

The application-identifier field is of type string, and contains the application identifier to which the PFDs belong.

6.4.3.3 void

6.4.3.4 caching-time

The caching-time field is of type uint64. It's a caching time value in seconds. When the caching timer elapses, the PCEF/TDF shall re-fetch the PFDs associated with the corresponding application-identifier from the PFDF. If the caching-time is:

- omitted, the PCEF/TDF shall use the preconfigured default caching time value to control how long the PFDs valid; or
- set to zero, then the PFDs are valid until explicitly deleted by the PFDF.

NOTE: The value zero of caching-time is only valid in the Combination mode and can be used as an indication of infinite PFD validity. It can be used to avoid unnecessary PFD refreshment for an application, in such case, the PFDs cannot be deleted at the expiration of the caching timer if it was not stopped at reception of caching-time set to zero value.

6.4.3.5 pfds

The pfds field is of type array. It contains the PFDs that belong to a specific application identifier.

The following defines the content of the PFD:

The pfd-identifier field shall be included within the PFD, and shall be unique within the context of an application identifier.

One of the flow-descriptions field, the urls field, the domain-names field or a custom filed shall be present for the application traffic detection.

The custom field is type of any with an arbitrary field name. Any primitive type, array, or object defined in IETF RFC 7159 [11] can be set to this field. It provides the extensions for the proprietary application traffic detection mechanisms, based on the agreement between the ASP and the mobile operator.

Below are the JCR for the PFD:

```
$pfd = {
  $pfd-identifier,
  ($flow-descriptions | $urls | $domain-names | // : any) ?
}
```

6.4.3.6 pfd-identifier

The pfd-identifier is of type string. It uniquely identifies a PFD within its associated application identifier.

6.4.3.7 flow-descriptions

The flow-descriptions is of type string array. The content of the string has the same encoding as the IPFilterRule AVP value as defined in IETF RFC 6733 [13]. It represents a 3-tuple with protocol, server ip and server port for UL/DL application traffic.

6.4.3.8 urls

The urls is type of string array. The content of the string shall be a URL or a regular expression which is used to match the significant parts of the URL.

6.4.3.9 domain-names

The domain-names is type of string array. The content of the string shall be an FQDN or a regular expression as a domain name matching criteria.

6.4.4 JSON provisioning fields

6.4.4.1 General

In addition to subclause 6.4.3, Table 6.4.4.1.1 describes the extra JSON fields used within the body of the HTTP messages representing the PFDs provisioning information in push mode. The table includes the information about the name of the field and the type of the fields.

Table 6.4.4.1.1: Gw/Gwn Provisioning JSON fields

Field Name	Clause defined	JSON Value Type (NOTE 1)	JCR Type (NOTE 2)	Applicability (NOTE 3)
notification-flag (NOTE 4)	6.4.4.2	boolean	boolean	
removal-flag (NOTE 4)	6.4.4.3	boolean	boolean	
allowed-delay (NOTE 4)	6.4.4.4	number	uint64	
partial-flag (NOTE 4)	6.4.4.5	boolean	boolean	PartialUpdate
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 subclause 6.3.5. NOTE 4: Only one of the notification-flag, removal-flag and the partial-flag for the application identifier shall be set to true.				

6.4.4.2 notification-flag

The notification-flag is of type boolean.

If the value of the flag is true, then it indicates a PFDs notification, and the corresponding PFDs information for the application identifier is not provided in the JSON body. Instead, the PCEF/TDF shall reload it with the application identifier via the pull procedure.

6.4.4.3 removal-flag

The removal-flag is of type boolean.

If the value of the flag is true, then it indicates the PFDs and its associated application identifier have been removed in the PFDF. The PCEF/TDF shall unbind all PFDs associated with that application identifier.

6.4.4.4 allowed-delay

The allowed-delay field is of type uint64. It contains a time interval in seconds. The PFDs shall be deployed within this time interval. If the allowed-delay is omitted, or if it is set to zero, then the PFDs shall be immediately deployed.

6.4.4.5 partial-flag

The partial-flag is of type boolean.

If the value of the flag is true, then it indicates some PFDs for the application identifier(s) have been created/updated/deleted in the PFDF. The PCEF/TDF shall update existing PFDs or add new PFDs for the PFD identifier(s) with PFDs, or unbind all PFDs associated with the PFD identifier(s) without any PFDs content.

6.4.5 JSON errors and informational response fields

6.4.5.1 General

Table 6.4.5.1.1 describes the JSON fields defined for the errors and informational responsesincluding their types and the filed names.

Table 6.4.5.1.1: JSON fields for errors and informational response

Field Name	Subclause defined	JSON Value Type (NOTE 1)	JCR Type (NOTE 2)
errors	3GPP TS 29.155 [15] (NOTE 4)	array	array
error-type	3GPP TS 29.155 [15] (NOTE 4)	string	"application" "interface" "server" "other" (NOTE 3)
error-message	3GPP TS 29.155 [15]	string	string
error-tag	6.4.5.2	string	string
error-path	3GPP TS 29.155 [15]	string	string
error-info	3GPP TS 29.155 [15]	object	object
success-message	3GPP TS 29.155 [15]	string	string
success-path	3GPP TS 29.155 [15]	string	string
success-info	3GPP TS 29.155 [15]	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 PCEF/TDF to the PFDF.			

6.4.5.2 error-tag

The error-tag field is of type string. It defines a tag for a particular error.

PFD_EVENT

This error-tag shall be used when the PFDs cannot be installed/modified at the PCEF, pfd-reports shall be provided in the error-info as described in subclause 6.4.6.

6.4.6 JSON report fields

6.4.6.1 General

Table 6.4.6.1.1 describes the JSON fields defined for the report information objects which are included in the error-info field.

Table 6.4.6.1.1: JSON fields for Report from the PCEF

Field Name	Subclause defined	JSON Value Type (NOTE 1)	JCR Type (NOTE 2)
pdf-reports	6.4.6.2	array	array
application-ids	3GPP TS 29.250 [14]	array	array
pfd-failure-code	6.4.6.3	string	string
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].			

6.4.6.2 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 the Push mode. 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.

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

The JCR format for the pfd-reports is:

```
$pfd-reports = "pfd-reports" : [
  {     $application-ids,
       $pfd-failure-code
    } +
]
```

6.4.6.3 pfd-failure-code

The pfd-failure-code is of type string.

The following values are defined in this release:

MALFUNCTION

RESOURCES_LIMITATION

OTHER_REASON

6.5 The discovery of PCEF/TDF and PFDF

6.5.1 PCEF/TDF discovery

For “Push mode”, the PFDF requires a method to discover the PCEF(s)/TDF(s).

A list of PCEF/TDF URIs may be pre-configured on the PFDF.

The PFDF may select the PCEF/TDF by this configuration.

NOTE: Other methods to discover the PCEF(s)/TDF(s) are implementation specific.

6.5.2 PFDF discovery

For “Pull mode”, the PCEF/TDF requires a method to discover the PFDF.

The PFDF URI may be pre-configured on the PCEF/TDF.

The PCEF/TDF may select the PFDF by this configuration.

NOTE 1: Other methods to discover the PFDF are implementation specific.

NOTE 2: If multiple PFDFs are deployed within one PLMN, coordination between the PFDFs is needed to avoid duplicated provisioning, e.g. to avoid that the PCEF/TDFs served by different PFDFs are overlapping

7 Secure communication

Either the NDS/IP network layer security defined in 3GPP TS 33.210 [8] or HTTP over TLS as defined in IETF RFC 2818 [9] should be used to secure communication over the REST based Gw and Gwn interfaces.

Annex A (normative): JSON Schema

A.1 PFDs schema

This subclause defines the JSON schema for the body of HTTP responses providing the state of one or more PFDs resources in pull mode. The schema is based on IETF draft-newton-json-content-rules [12] and is defined below:

```
# jcr-version 0.7

# ruleset-id 3gpp.gwapplication.pfds


; JCR based on draft v7 representing the pfds root resource

$pfds-root = @{root}{

  $application-identifier,

   $caching-time ?,

  $pfds

}


; An array list of the PFDs for multiple application identifiers

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




; The caching timer for the PFDs

$caching-time = "caching-time" : uint64


; The PFDs associated with the same application identifier

$pfds = "pfds" : [ $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.2 Provisioning schema

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

```
# jcr-version 0.7

# ruleset-id 3gpp.gwapplication.provisioning

# import 3gpp.gwapplication.pfds as pfds


; JCR based on draft v7 representing the PFDs provisioning data

$provisioning-root = @{root}{
  $pfds.application-identifier,
  $notification-flag ?,
  $removal-flag ?,
  $partial-flag ?,
  $allowed-delay ?,
  $pfds.pfds ?
}


; An array list of the PFDs provisioning/notification for multiple application identifiers

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


; A flag indicates whether this is a notification or a provisioning

$notification-flag = "notification-flag" : boolean


; 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 allowed delay time for the PFDs deployment
$allowed-delay = "allowed-delay" : uint64
```

A.3 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.gwapplication.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 PFDF
$pfd-reports = "pfd-reports" : [
  {     $application-ids,
       $pfd-failure-code
    } +
]

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

; 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 *}

; application identifiers
$application-ids = "application-ids" : [ string +]
```

Annex B (informative): Call Flows

B.1 General

This annex describes the procedures for the interactions between the PFDF and the PCEF/TDF.

B.2 Request for PFDs (“Pull mode”)

This subclause describes the signalling flow for the request for PFDs.

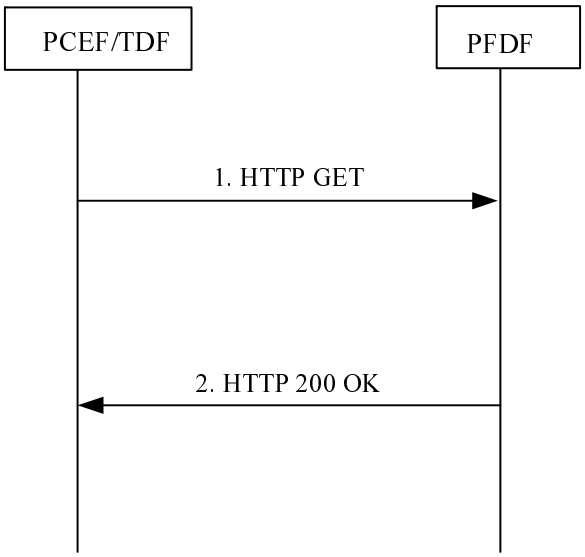

Figure B.2.1: Request for PFDs

1. The PCEF/TDF sends the HTTP GET to the PFDF to request the PFDs including the parameters defined in subclause 6.3.3.2, 6.3.3.3 or 6.3.3.4.
2. The PFDF sends the HTTP 200 OK response to the PCEF/TDF including the parameters defined in subclause 6.3.3.2, 6.3.3.3 or 6.3.3.4.

B.3 Provisioning of PFDs (“Push mode”)

This subclause describes the signalling flow for the provisioning of PFDs.

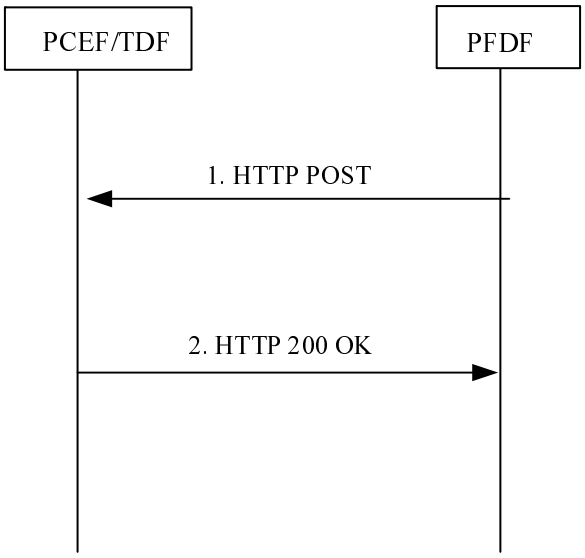

Figure B.3.1: Provisioning of the PFDs

1. The PFDF sends the HTTP POST to the PCEF/TDF to provision the PFDs of one or more application identifiers including the parameters defined in subclause 6.3.3.5.
2. The PCEF/TDF sends the HTTP 200 OK response to the PFDF including the parameters defined in subclause 6.3.3.5.

B.4 Provisioning of PFDs by Push with Notifications

This subclause describes the signalling flow for the provisioning of PFDs by push with notifications.

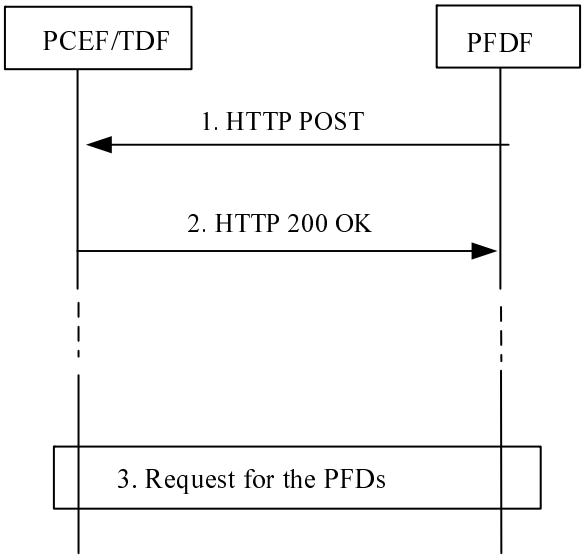

Figure B.4.1: Provisioning of the PFDs by push with notification

1. The PFDF sends the HTTP POST to the PCEF/TDF to provision the notification of creation/update for one or more application identifiers including the parameters defined in subclause 6.3.3.5.
2. The PCEF/TDF sends the HTTP 200 OK response to the PFDF including the parameters defined in subclause 6.3.3.5.
3. Within the allowed-delay provided in the step1, the PCEF/TDF initiates the procedure defined in Annex B.2 to request the PFDs for the application identifiers.

NOTE: The step 3 can be executed multiple times if the PCEF/TDF initiates separate pull requests to retrieve the PFDs for different application identifier(s).

Annex C (informative): Change history

Date	TSG #	TSG Doc.	CR	Rev	Cat	Subject/Comment	New
2016-10						TS skeleton of Gw and Gwn reference points stage 3.	0.0.0
2016-10						Inclusion of C3-163212, C3-163213, C3-163304, C3-163323, C3-163326, C3-163327 and editorial change from Rapporteur.	0.1.0
2016-11						Inclusion of C3-164174, C3-164177, C3-164231 and editorial change from Rapporteur.	0.2.0
2017-01						Inclusion of C3-170055, C3-170060, C3-170061, C3-170063, C3-170065 and editorial change from Rapporteur.	0.3.0
2017-02						Inclusion of C3-171274, C3-171304, C3-171307, C3-171308, C3-171319, and editorial change from Rapporteur.	0.4.0
2017-03	CT#75	CP-170101				TS sent for Information to Plenary	1.0.0
2017-04						Inclusion of C3-172137, C3-172195, C3-172197, C3-172237, C3-172238, C3-172246, and editorial change from Rapporteur.	1.1.0
2017-05						Inclusion of C3-173120, C3-173154, C3-173196, C3-173198, C3-173201, C3-173288, C3-173289, C3-173290, C3-173295, C3-173297, C3-173320, C3-173322, C3-173331, and editorial change from Rapporteur.	1.2.0
2017-06	CT#76	CP-171144				TS sent for approval to plenary.	2.0.0
2017-06	CT#76	CP-171144				TS approved at plenary.	14.0.0
2017-09	CT#77	CP-172047	0001	2	F	Clarification of partial update.	14.1.0
2017-09	CT#77	CP-172047	0002	3	F	Claification of combination mode.	14.1.0
2017-09	CT#77	CP-172047	0005	2	F	Clarification of PCEF's behaviour in the pull mode.	14.1.0
2017-09	CT#77	CP-172047	0006	1	F	Update the reference of HTTP 1.1.	14.1.0
2017-09	CT#77	CP-172047	0007	1	F	Caching time correction.	14.1.0
2017-09	CT#77	CP-172047	0011	1	F	PFD error report correction.	14.1.0
2017-09	CT#77	CP-172047	0012	1	F	Correction for PFD provisioning notification.	14.1.0
2017-09	CT#77	CP-172047	0014	1	F	Reference and URI design correction.	14.1.0
2017-09	CT#77	CP-172047	0015	1	F	Correction to PFD removal.	14.1.0
2017-12	CT#78	CP-173100	0016	-	F	Correction to PFD removal.	14.2.0
2017-12	CT#78	CP-173100	0017	-	F	Precedence for 3GPP custom headers.	14.2.0

History

Document history		
V14.0.0	July 2017	Publication
V14.1.0	October 2017	Publication
V14.2.0	January 2018	Publication