ETSI TS 129 500 V15.6.0 (2020-07)

TECHNICAL SPECIFICATION

5G;
5G System;
Technical Realization of Service Based Architecture;
Stage 3
(3GPP TS 29.500 version 15.6.0 Release 15)

Reference

RTS/TSGC-0429500vf60

Keywords

5G

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.

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

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

Copyright Notification

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

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

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 specifies the technical realization of the 5GC Service Based Architecture, protocols supported over the Service Based Interfaces, and the functionalities supported in the Service Based Architecture.

The service requirements for the 5G system are defined in 3GPP TS 22.261 [2]. The system architecture requirements are defined in 3GPP TS 23.501 [3] and the procedures and flows in 3GPP TS 23.502 [4].

The design principles and documentation guidelines for 5GC SBI APIs are specified in 3GPP TS 29.501 [5].

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 22.261: "Service requirements for the 5G system; Stage 1".

[3] 3GPP TS 23.501: "System Architecture for the 5G System; Stage 2".

[4] 3GPP TS 23.502: "Procedures for the 5G System; Stage 2".

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

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

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

[8] 3GPP TS 29.510: "5G System; Network Function Repository Services; Stage 3".

[9] OpenAPI: "OpenAPI 3.0.0 Specification", https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.0.md.

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

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

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

[13] 3GPP TS 29.571: "5G System; Common Data Types for Service Based Interfaces Stage 3".

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

[15] 3GPP TS 23.003: "Numbering, addressing and identification".

[16] IETF RFC 5681: "TCP Congestion Control".

[17] 3GPP TS 33.501: "Security Architecture and Procedures for 5G System".

[18] IANA: "SMI Network Management Private Enterprise Codes", http://www.iana.org/assignments/enterprise-numbers.

[19] IETF RFC 7944: "Diameter Routing Message Priority".

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

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

[22] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[23] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".

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

[25] IETF RFC 7516: "JSON Web Encryption (JWE)".

[26] IETF RFC 7515: "JSON Web Signature (JWS)".

[27] 3GPP TS 29.573: "5G System: Public Land Mobile Network (PLMN) Interconnection; Stage 3".

[28] 3GPP TS 29.502: "5G System; Session Management Services; Stage 3".

[29] 3GPP TS 29.503: "5G System; Unified Data Management Services; Stage 3".

[30] Void.

[31] 3GPP TS 29.518: "5G System; Access and Mobility Management Services; Stage 3".

[32] 3GPP TS 29.531: "5G System; Network Slice Selection Services; Stage 3".

[33] IETF RFC 7694: "Hypertext Transfer Protocol (HTTP) Client-Initiated Content-Encoding".

[34] IETF RFC 1952: "GZIP file format specification version 4.3".

[35] 3GPP TS 29.525: "5G System; UE Policy Control Service; Stage 3".

3 Definitions and abbreviations

3.1 Definitions

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

3.2 Abbreviations

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

HTTP	Hypertext Transfer Protocol
TCP	Transmission Control Protocol
SMP	SBI Message Priority

4 Service Based Architecture Overview

4.1 NF Services

3GPP TS 23.501 [3] defines the 5G System Architecture as a Service Based Architecture, i.e. a system architecture in which the system functionality is achieved by a set of NFs providing services to other authorized NFs to access their services.

Control Plane (CP) Network Functions in the 5G System architecture shall be based on the service based architecture.

A NF service is one type of capability exposed by a NF (NF Service Producer) to other authorized NF (NF Service Consumer) through a service based interface. A NF service may support one or more NF service operation(s). See clause 7 of 3GPP TS 23.501 [3].

Network Functions may offer different functionalities and thus different NF services. Each of the NF services offered by a Network Function shall be self-contained, acted upon and managed independently from other NF services offered by the same Network Function (e.g. for scaling, healing).

4.2 Service Based Interfaces

A service based interface represents how the set of services is provided or exposed by a given NF. This is the interface where the NF service operations are invoked.

The following Control Plane interfaces within the 5G Core Network specified in 3GPP TS 23.501 [3] are defined as service based interfaces:

- Namf, Nsmf, Nudm, Nnrf, Nnssf, Nausf, Nnef, Nsmsf, Nudr, Npcf, N5g-eir, Nlmf, Nnwdaf.

4.3 NF Service Framework

4.3.1 General

The Service Based Architecture shall support the NF Service Framework that enable the use of NF services as specified in clause 7.1 of 3GPP TS 23.501 [3].

The NF Service Framework includes the following mechanisms:

- NF service registration and de-registration: to make the NRF aware of the available NF instances and supported services (see clause 7.1.5 of 3GPP TS 23.501 [3]);
- NF service discovery: to enable a NF Service Consumer to discover NF Service Producer instance(s) which provide the expected NF service(s) (see clause 7.1.3 of 3GPP TS 23.501 [3]);
- NF service authorization: to ensure the NF Service Consumer is authorized to access the NF service provided by the NF Service Producer (see clause 7.1.4 of 3GPP TS 23.501 [3]).

The corresponding stage 3 procedures are defined in 3GPP TS 29.510 [8].

4.3.2 NF Service Advertisement URI

When invoking a service operation of a NF Service Producer that use HTTP methods with a message body (i.e PUT, POST and PATCH), the NF Service Consumer may provide NF Service Advertisement URL(s) in the service operation request, based on operator policy, if it expects that the NF Service Producer may subsequently consume NF service(s) which the NF Service Consumer can provide (as a NF Service Producer).

When receiving NF Service Advertisement URI(s) in a service operation request, the NF Service Producer may store and use the Service Advertisement URL(s) to discover NF services produced by the NF Service Consumer in subsequent procedures, based on operator policy.

The NF Service Advertisement URI identifies the nfInstance resource(s) in the NRF which are registered by NF Service Producer(s).

An example of NF Service Advertisement URI could be represented as:

"{apiRoot}/nnrf-disc/nf-instances?nfInstanceId={nfInstanceId}".

NOTE: The NF Service Advertisement URI can be used e.g. when different NRFs are deployed in the PLMN.

When applicable, the NF Service Advertisement URI(s) shall be carried in HTTP message body.

5 Protocols Over Service Based Interfaces

5.1 Protocol Stack Overview

The protocol stack for the service based interfaces is shown on Figure 5.1-1.

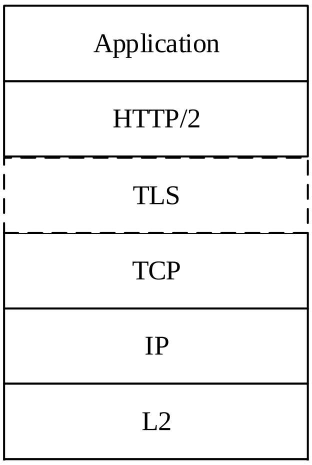

Figure 5.1-1: SBI Protocol Stack

The service based interfaces use HTTP/2 protocol (see clause 5.2) with JSON (see clause 5.4) as the application layer serialization protocol. For the security protection at the transport layer, all 3GPP NFs shall support TLS and TLS shall be used within a PLMN if network security is not provided by other means, as specified in 3GPP TS 33.501 [17].

5.2 HTTP/2 Protocol

5.2.1 General

HTTP/2 as described in IETF RFC 7540 [7] shall be used in Service based interface.

5.2.2 HTTP standard headers

5.2.2.1 General

This clause lists the HTTP standard headers that shall be supported on SBI, other HTTP standard headers defined in IETF RFCs may be supported by NF.

5.2.2.2 Mandatory to support HTTP standard headers

The HTTP request standard headers and the HTTP response standard headers that shall be supported on SBI are defined in Table 5.2.2.2-1 and in Table 5.2.2.2-2 respectively. Mandatory to support HTTP standard headers does not mean all the HTTP requests and responses carry the identified request and response headers respectively. It only means it is mandatory to support the processing of the identified headers in request and response message.

Table 5.2.2.2-1: Mandatory to support HTTP request standard headers

Name	Reference	Description
Accept	IETF RFC 7231 [11]	This header is used to specify response media types that are acceptable.
Accept-Encoding	IETF RFC 7231 [11]	This header may be used to indicate what response content-encodings (e.g gzip) are acceptable in the response.
Content-Length	IETF RFC 7230 [12]	This header is used to provide the anticipated size, as a decimal number of octets, for a potential payload body.
Content-Type	IETF RFC 7231 [11]	This header is used to indicate the media type of the associated representation.
Content-Encoding	IETF RFC 7231 [11]	This header may be used in some requests to indicate the content encodings (e.g gzip) applied to the resource representation beyond those inherent in the media type.
User-Agent	IETF RFC 7231 [11]	This header shall be mainly used to identify the NF type of the HTTP/2 client. The pattern of the content should start with the value of NF type (e.g. udm, see NOTE 1) and followed by a "-" and any other specific information if needed afterwards.
Cache-Control	IETF RFC 7234 [20]	This header may be used in some HTTP/2 requests to provide the HTTP cache-control directives that the client is willing to accept from the server.
If-Modified-Since	IETF RFC 7232 [24]	This header may be used in a conditional GET request, for server revalidation. This is used in conjunction with the Last-Modified server response header, to fetch content only if the content has been modified from the cached version.
If-None-Match	IETF RFC 7232 [24]	This header may be used in a conditional GET request. This is used in conjunction with the ETag server response header, to fetch content only if the tag value of the resource on the server differs from the tag value in the If-None-Match header.
If-Match	IETF RFC 7232 [24]	This header may be used in a conditional POST or PUT or DELETE or PATCH request. This is used in conjunction with the ETag server response header, to update / delete content only if the tag value of the resource on the server matches the tag value in the If-Match header.
Via	IETF RFC 7230 [12]	This header shall be inserted by HTTP proxies.
Authorization	IETF RFC 7235 [21]	This header shall be used if OAuth 2.0 based access authorization with "Client Credentials" grant type is used as specified in clause 13.4.1 of 3GPP TS 33.501 [17], clause 7 of IETF RFC 6749 [22] and IETF RFC 6750 [23].
NOTE 1: The value of NF type in the User-Agent header shall comply with the enumeration value of Table 6.1.6.3.3-1 in 3GPP TS 29.510 [8].		

Table 5.2.2.2-2: Mandatory to support HTTP response standard headers

Name	Reference	Description
Content-Length	IETF RFC 7230 [12]	This header may be used to provide the anticipated size, as a decimal number of octets, for a potential payload body.
Content-Type	IETF RFC 7231 [11]	This header shall be used to indicate the media type of the associated representation.
Location	IETF RFC 7231 [11]	This header may be used in some responses to refer to a specific resource in relation to the response.
Retry-After	IETF RFC 7231 [11]	This header may be used in some responses to indicate how long the user agent ought to wait before making a follow-up request.
Content-Encoding	IETF RFC 7231 [11]	This header may be used in some responses to indicate to the HTTP/2 client the content encodings (e.g gzip) applied to the resource representation beyond those inherent in the media type.
Cache-Control	IETF RFC 7234 [20]	This header may be used in some responses (e.g. NRF responses to queries) to provide HTTP response cache control directives. The cache directives "no-cache", "no-store", "max-age" and "must-revalidate" values shall be supported.
Age	IETF RFC 7234 [20]	This header may be inserted by HTTP proxies when returning a cached response. The "Age" header field conveys the sender's estimate of the amount of time since the response was generated or successfully validated at the origin server. The presence of an Age header field implies that the response was not generated or validated by the origin server for this request.
Last-Modified	IETF RFC 7232 [24]	This header may be sent to allow for conditional GET with the If-Modified-Since header.
ETag	IETF RFC 7232 [24]	This header may be sent to allow for conditional GET with the If-If-None-Match header or a conditional POST / PUT / PATCH / DELETE with the If-Match header.
Via	IETF RFC 7230 [12]	This header shall be inserted by HTTP proxies.
Allow	IETF RFC 7231 [11]	This header field shall be used to indicate methods supported by the target resource.
WWW-Authenticate	IETF RFC 7235 [21]	This header field shall be included when an NF service producer rejects a request with a "401 Unauthorized" status code (e.g when a request is sent without an OAuth 2.0 access token or with an invalid OAuth 2.0 access token).
Accept-Encoding	IETF RFC 7694 [33]	See clause 6.9 for the use of this header.

5.2.3 HTTP custom headers

5.2.3.1 General

The list of custom HTTP headers applicable to 3GPP Service Based NFs are specified below.

5.2.3.2 Mandatory to support custom headers

5.2.3.2.1 General

The 3GPP NF Services shall support the HTTP custom headers specified in Table 5.2.3.2-1 below. A description of each custom header and the normative requirements on when to include them are also provided in Table 5.2.3.2-1.

Table 5.2.3.2-1: Mandatory HTTP custom headers

Name	Reference	Description
3gpp-Sbi-Message-Priority	Clause 5.2.3.2.1	This header is used to specify the HTTP/2 message priority for 3GPP service based interfaces. This header shall be included in HTTP/2 messages when a priority for the message needs to be conveyed (e.g HTTP/2 messages related to Multimedia Priority Sessions).
3gpp-Sbi-Callback	Clause 5.2.3.2.3	This header is used to indicate if a HTTP/2 message is a callback (e.g notification). This header shall be included in HTTP POST messages for callbacks towards NF service consumer(s) in another PLMN via the SEPP (See 3GPP TS 29.573 [27]).

5.2.3.2.2 3gpp-Sbi-Message-Priority

The header contains the HTTP/2 message priority value.

The encoding of the header follows the ABNF as defined in IETF RFC 7230 [12].

3gpp-Sbi-Message-Priority = "3gpp-Sbi-Message-Priority" ":" (DIGIT / %x31-32 DIGIT / "3" %x30-31)

A message with 3gpp-Sbi-Message-Priority "0" has the highest priority.

An example is: 3gpp-Sbi-Message-Priority: 10.

5.2.3.2.3 3gpp-Sbi-Callback

The header contains the type of notification. The value for the notificaition type is a string used identifing a particular type of callback (e.g a notification, typically the name of the notify service operation).

The encoding of the header follows the ABNF as defined in IETF RFC 7230 [12].

3gpp-Sbi-Notification header field = "3gpp-Sbi-Callback" ":" OWS cbtype

cbtype = 1*cbchar

cbchar = "-" / "_" / DIGIT / ALPHA

An example is: 3gpp-Sbi-Callback: Nnrf_NFManagement_NFStatusNotify.

The list of valid values for this header is specified in Annex B.

5.2.4 HTTP error handling

HTTP/2 connection error and stream error shall be supported as specified in clause 5.4 of IETF RFC 7540 [7].

Guidelines for error responses to the invocation of APIs of NF services are specified in clause 4.8 of 3GPP TS 29.501 [3]. API specific error responses are specified in the respective technical specifications.

5.2.5 HTTP/2 server push

HTTP/2 Server Push as specified in clause 8.2 of IETF RFC 7540 [7] may be supported and may be used by a NF Service Producer to proactively push resources to a NF Service Consumer, see clause 4.9.5 of 3GPP TS 29.501 [5].

A NF Service Consumer may choose to disable HTTP/2 Server Push by setting SETTINGS_ENABLE_PUSH to 0, as specified in clause 8.2 of IETF RFC 7540 [7].

5.2.6 HTTP/2 connection management

The HTTP request / response exchange mechanism as specified in clause 8.1 of IETF RFC 7540 [7] shall be supported between the 3GPP NFs. An HTTP/2 endpoint shall support establishing multiple HTTP/2 connections (at least two) towards a peer HTTP/2 endpoint. The peer HTTP/2 endpoint is identified by host and port pair where the host is derived from the target URI (see clause 6.1.1).

NOTE 1: HTTP/2 connection redundancy allows transporting messages through diverse IP paths and improve 5GC resiliency.

As per clause 8.1 of IETF RFC 7540 [7] a HTTP request / response exchange fully consumes a single stream. When the HTTP/2 Stream IDs on a given HTTP/2connection is exhausted, an HTTP/2 endpoint, shall establish another HTTP/2connection towards that peer HTTP/2 endpoints.

NOTE 2: As per IETF RFC 7540 [7], a stream ID once closed cannot be reused on the same HTTP/2 connection.

The 3GPP NF shall take care to avoid simultaneous stream ID exhaustion on all the available HTTP/2 connections towards each peer.

An NF acting as an HTTP/2 client shall support testing whether a connection is still active by sending a PING frame as specified in clause 6.7 of IETF RFC 7540 [7]. When and how often a PING frame may be sent is implementation specific but shall be configurable by operator policy.

A PING frame shall not be sent more often than every 60 s on each path.

5.2.7 HTTP status codes

5.2.7.1 General

This clause describes the HTTP status codes usage on SBI.

HTTP status codes are carried in ":status" pseudo header field in HTTP/2, as defined in clause 8.1.2.4 in IETF RFC 7540 [7].

Table 5.2.7.1-1 specifies HTTP status codes per HTTP method which shall be supported on SBI. Support of an HTTP status code shall be:

- mandatory, which is marked in table as "M". This means that all 3GPP NFs shall support the processing of the specific HTTP status code for the specific HTTP method, when received in a HTTP response message. In such cases the 3GPP NF shall also support the handling of the "ProblemDetails" JSON object with the Content-Type header field set to the value "application/problem+json" for HTTP status codes 4xx and 5xx, if the corresponding API definition in the related technical specification does not specify another response body for the corresponding status code;
- service specific, which is marked in table as "SS" and means that the requirement to process the HTTP status code depends on the definition of the specific API; or
- not applicable, which is marked in table as "N/A". This means that the specific HTTP status code shall not be used for the specific HTTP method within the 3GPP NFs.

Table 5.2.7.1-1: HTTP status code supported on SBI

HTTP status code	HTTP method					
	DELETE	GET	PATCH	POST	PUT	OPTIONS
100 Continue	N/A	N/A	N/A	N/A	N/A	N/A
200 OK (NOTE 1, NOTE 2)	SS	M	SS	SS	SS	M
201 Created	N/A	N/A	N/A	SS	SS	N/A
202 Accepted	SS	N/A	SS	SS	SS	N/A
204 No Content (NOTE 2)	M	N/A	SS	SS	SS	SS
300 Multiple Choices	N/A	N/A	N/A	N/A	N/A	N/A
303 See Other	SS	SS	N/A	SS	SS	N/A
307 Temporary Redirect	SS	SS	SS	SS	SS	SS
308 Permanent Redirect	SS	SS	SS	SS	SS	SS
400 Bad Request	M	M	M	M	M	M
401 Unauthorized	M	M	M	M	M	M
403 Forbidden	M	M	M	M	M	M
404 Not Found	M	M	M	M	M	M
405 Method Not Allowed	SS	SS	SS	SS	SS	SS
406 Not Acceptable	N/A	M	N/A	N/A	N/A	SS
408 Request Timeout	SS	SS	SS	SS	SS	SS
409 Conflict	N/A	N/A	SS	SS	SS	N/A
410 Gone	SS	SS	SS	SS	SS	SS
411 Length Required	N/A	N/A	M	M	M	SS
412 Precondition Failed	SS	SS	SS	SS	SS	N/A
413 Payload Too Large	N/A	N/A	M	M	M	SS
414 URI Too Long	N/A	SS (NOTE 3)	N/A	N/A	SS	N/A
415 Unsupported Media Type	N/A	N/A	M	M	M	SS
429 Too Many Requests	M	M	M	M	M	M
500 Internal Server Error	M	M	M	M	M	M
501 Not Implemented	SS	SS	SS	SS	SS	SS
503 Service Unavailable	M	M	M	M	M	M
504 Gateway Timeout	SS	SS	SS	SS	SS	SS
NOTE 1: "200 OK" response used on SBI shall contain body. NOTE 2: If the NF acting as an HTTP Client receives 2xx response code not appearing in table, the NF shall treat the received 2xx response: - as "204 No Content" if 2xx response does not contain body; and - as "200 OK" if 2xx response contains body. NOTE 3: If GET method includes any query parameter, the NF acting as an HTTP Client shall support "414 URI Too Long" status code.						

5.2.7.2 NF as HTTP Server

A NF acting as an HTTP server shall be able to generate HTTP status codes specified in clause 5.2.7.1 per indicated HTTP method.

A request using an HTTP method which is not supported by any resource of a given 5GC SBI API shall be rejected with the HTTP status code "501 Not Implemented".

NOTE 1: In this case, the NF does not need to include in the HTTP response the "cause" attribute indicating corresponding error since the HTTP status code "501 Not Implemented" itself provides enough information of the error, i.e. the NF does not recognize the HTTP method.

If the specified target resource does not exist, the NF shall reject the HTTP method with the HTTP status code "404 Not Found".

If the NF supports the HTTP method for several resources in the API, but not for the target resource of a given HTTP request, the NF shall reject the request with the HTTP status code "405 Method Not Allowed" and shall include in the response an Allow header field containing the supported method(s) for that resource.

NOTE 2: In this case, the NF does not need to include in the HTTP response the "cause" attribute indicating corresponding error since the HTTP status code "405 Method Not Allowed" itself provides enough information of the error and hence the Allow header field lists HTTP method(s) supported by the target resource.

If a received HTTP request contains unknown IEs, i.e. Information Elements within the JSON body, the NF may discard such IEs and shall process the rest of the request message, unless the schema definition of the received message prohibits the presence of additional IEs or constrains their types. There are cases (e.g. Nnrf_NFManagement API) where the receiver of certain HTTP requests needs to process unknown IEs (e.g. to store in NRF an NF Profile containing vendor-specific attributes, and send them in NFDiscovery results).

If a received HTTP request contains IEs or query parameters not compliant with the schema defined in the corresponding OpenAPI specification, the NF should reject the request with the appropriate error code, e.g. "400 Bad Request (INVALID_MSG_FORMAT)", even when the failed IEs are defined as optional by the schema.

If a received HTTP PATCH request contains a body with modification instruction(s) for unknown attribute(s) in addition to modification instruction(s) for known attribute(s), the NF shall discard those modification instruction(s) for unknown attribute(s).

If the NF supports the HTTP method by a target resource but the NF cannot successfully fulfil the received request, the following requirements apply.

A NF as HTTP Server should map application errors to the most similar 3xx/4xx/5xx HTTP status code specified in table 5.2.7.1-1. If no such code is applicable, it should use "400 Bad Request" status code for errors caused by client side or "500 Server Internal Error" status code for errors caused on server side.

If the received HTTP request contains unsupported payload format, the NF shall reject the HTTP request with the HTTP status code "415 Unsupported Media Type". If the HTTP PATCH method is rejected, the NF shall include the Accept-Patch header field set to the value of supported patch document media types for a target resource i.e. to "application/merge-patch+json" if the NF supports "JSON Merge Patch" and to "application/json-patch+json" if the NF supports "JSON Patch". If the received HTTP PATCH request contains both "JSON Merge Patch" and "JSON Patch" documents and the NF supports only one of them, the NF shall ignore unsupported patch document.

NOTE 3: The format problem might be due to the request's indicated Content-Type or Content-Encoding header fields, or as a result of inspecting the payload body directly.

If the received HTTP request contains payload body larger than the NF is able to process, the NF shall reject the HTTP request with the HTTP status code "413 Payload Too Large".

If the result of the received HTTP POST request used for a resource creation would be equivalent to the existing resource, the NF shall reject the HTTP request with the HTTP status code "303 See Other" and shall include in the HTTP response a Location header field set to the URI of the existing resource.

Protocol and application errors common to several 5GC SBI API specifications for which the NF shall include in the HTTP response a payload body ("ProblemDetails" data structure or application specific error data structure) with the "cause" attribute indicating corresponding error are listed in table 5.2.7.2-1.

Table 5.2.7.2-1: Protocol and application errors common to several 5GC SBI API specifications

Protocol or application Error	HTTP status code	Description
INVALID_API	400 Bad Request	The HTTP request contains an unsupported API name or API version in the URI.
INVALID_MSG_FORMAT	400 Bad Request	The HTTP request has an invalid format.
INVALID_QUERY_PARAM	400 Bad Request	The HTTP request contains an unsupported query parameter in the URI. (NOTE 1)
MANDATORY_QUERY_PARAM_INCORRECT	400 Bad Request	A mandatory query parameter, or a conditional query parameter but mandatory required, for an HTTP method was received in the URI with semantically incorrect value. (NOTE 1)
OPTIONAL_QUERY_PARAM_INCORRECT	400 Bad Request	An optional query parameter for an HTTP method was received in the URI with a semantically incorrect value that prevents successful processing of the service request. (NOTE 1)
MANDATORY_QUERY_PARAM_MISSING	400 Bad Request	Query parameter which is defined as mandatory, or as conditional but mandatory required, for an HTTP method is not included in the URI of the request. (NOTE 1)
MANDATORY_IE_INCORRECT	400 Bad Request	A mandatory IE or conditional IE in data structure, but mandatory required, for an HTTP method was received with a semantically incorrect value. (NOTE 1)
OPTIONAL_IE_INCORRECT	400 Bad Request	An optional IE in data structure for an HTTP method was received with a semantically incorrect value that prevents successful processing of the service request. (NOTE 1)
MANDATORY_IE_MISSING	400 Bad Request	IE which is defined as mandatory or as conditional in data structure, but mandatory required, for an HTTP method is not included in the payload body of the request. (NOTE 1)
UNSPECIFIED_MSG_FAILURE	400 Bad Request	The request is rejected due to unspecified client error. (NOTE 2)
MODIFICATION_NOT_ALLOWED	403 Forbidden	The request is rejected because the contained modification instructions attempt to modify IE which is not allowed to be modified.
SUBSCRIPTION_NOT_FOUND	404 Not Found	The request for modification or deletion of subscription is rejected because the subscription is not found in the NF.
RESOURCE_URI_STRUCTURE_NOT_FOUND	404 Not Found	The request is rejected because a fixed part after the first variable part of an "apiSpecificResourceUriPart" (as defined in clause 4.4.1 of 3GPP TS 29.501 [5]) is not found in the NF. This fixed part of the URI may represent a sub-resource collection (e.g. contexts, subscriptions, policies) or a custom operation. (NOTE 5)
INCORRECT_LENGTH	411 Length Required	The request is rejected due to incorrect value of a Content-length header field.
NF_CONGESTION_RISK	429 Too Many Requests	The request is rejected due to excessive traffic which, if continued over time, may lead to (or may increase) an overload situation.
INSUFFICIENT_RESOURCES	500 Internal Server Error	The request is rejected due to insufficient resources.
UNSPECIFIED_NF_FAILURE	500 Internal Server Error	The request is rejected due to unspecified reason at the NF. (NOTE 3)
SYSTEM_FAILURE	500 Internal Server Error	The request is rejected due to generic error condition in the NF.
NF_CONGESTION	503 Service Unavailable	The NF experiences congestion and performs overload control, which does not allow the request to be processed. (NOTE 4)

NOTE 1: "invalidParams" attribute shall be included in the "ProblemDetails" data structure indicating unsupported, missing or incorrect IE(s) or query parameter(s).
NOTE 2: This application error indicates error in the HTTP request and there is no other application error value that can be used instead.
NOTE 3: This application error indicates error condition in the NF and there is no other application error value that can be used instead.
NOTE 4: If the reason for rejection is a temporary overload, the NF may include in the response a Retry-After header field to indicate how long the service is expected to be unavailable.
NOTE 5: If the request is rejected because of an error in an URI before the first variable part of an "apiSpecificResourceUriPart", the "404 Not Found" HTTP status code may be sent without "ProblemDetails" data structure indicating protocol or application error.

5.2.7.3 NF as HTTP Client

Besides the HTTP Status Codes defined in the API specification, a NF as HTTP client should support handling of 1xx, 3xx, 4xx and 5xx HTTP Status Codes specified in table 5.2.7.1-1, following the client behaviour in corresponding IETF RFC where the received HTTP Status Code is defined.

When receiving a not recommended or not recognized 1xx, 3xx, 4xx or 5xx HTTP Status Code, a NF as HTTP client should treat it as x00 status code of the class, as described in clause 6 of IETF RFC 7231 [11].

If 100, 200/204, 300, 400 or 500 response code is not defined by the API specification, the client may follow guidelines below:

a) For 1xx (Informational):

   1) Discard the response and wait for final response.

b) For 2xx (Successful):

   1) Consider the service operation is successful if no mandatory information is expected from the response payload in subsequent procedure.

   2) If mandatory information is expected from response payload in subsequent procedure, parse the payload following description in clause 6.2.1 of IETF RFC 7231 [11]. If parse is successful and mandatory information is extracted, continue with subsequent procedure.

   3) Otherwise, consider service operation has failure and start failure handling.

c) For 3xx (Redirection):

   1) Retry the request towards the directed resource referred in the Location header, using same request method.

d) For 4xx (Client Error):

   1) Validate the request message and make correction before resending. Otherwise, stop process and go to error handling procedure.

e) For 5xx (Server Error):

   1) Stop process and go to error handling process.

The handling of unknown, unexpected or erroneous HTTP request message IEs shall provide for the forward compatibility of the HTTP APIs used for the service based interfaces. Therefore, the sending HTTP entity shall be able to safely include in a message a new optional IE. Such an IE may also have a new type. A receiving HTTP entity shall behave as specified in clause 5.2.7.2.

5.2.8 HTTP/2 request retries

All NF services expose APIs across the service based interfaces and the APIs operate on resources. Invocation of an API though a HTTP method may result in the change of state of a resource depending of the request type. When a HTTP/2 client sends a request and it does not receive a response or it experiences a delay, it does not guarantee that the HTTP/2 request has not been processed by the HTTP/2 server.

A HTTP/2 client may retry the same request that uses an idempotent method any time (see IETF RFC 7231 [11] clause 4.2.2).

Retrying a non-idempotent HTTP/2 request on the same resource before a response for the previous request is received may lead to state changes on the resource with unspecified behaviour. HTTP conditional requests, as specified in IETF RFC 7232 [24] may be used to avoid such situations.

An NF acting as an HTTP/2 client may decide to retry non-idempotent request if it matches one of the conditions set out in clause 8.1.4 of IETF RFC 7540 [7]. API specific mechanisms as specified in respective technical specifications may be used for reconciling the state of resources, if the retry is attempted through a new TCP connection after a TCP connection failure.

The number of retry shall be limited. A client should always prefer to retry requests to an alternative server if the initial server is overloaded. In case of general overload situation where all possible servers are overloaded retry mechanisms should be disabled automatically.

5.2.9 Handling of unsupported query parameters

Unless specified otherwise for an API, a NF Service Producer that receives an HTTP request with one or more unsupported (i.e. not comprehended) query parameters shall:

a) for safe HTTP methods (e.g. HTTP GET request):

  - ignore the unsupported query parameters and respond to the request based on the rest of the request (e.g. other supported query parameters); or

  - reject the HTTP request as specified below for non-safe HTTP methods, e.g. based on other query parameters in the request or based on a response becoming very large;

b) for non-safe HTTP methods:

  - reject the request with a 400 Bad Request including a ProblemDetails IE with:

    - the cause attribute set to INVALID_QUERY_PARAM;

    - the invalidParams attribute indicating the unsupported query parameters;

    - the supportedFeatures attribute listing the features supported by the NF Service Producer, if any, set as specified for HTTP responses in clause 6.6.2.

5.3 Transport Protocol

The Transmission Control Protocol as described in IETF RFC 793 [6] shall be used as transport protocol as required by HTTP/2 (see IETF RFC 7540 [7]).

NOTE: When using TCP as the transport protocol, an HTTP/2 connection is mapped to a TCP connection.

If a Network Function does not register any port number to the NRF then it shall be prepared to receive connections on default port numbers, i.e. TCP port 80 for "http" URIs and TCP port 443 for "https" URIs as specified in IETF RFC 7540 [7].

5.4 Serialization Protocol

The JavaScript Object Notation (JSON) format as described in IETF RFC 8259 [10] shall be used as serialization protocol.

For transmitting large parts of opaque binary data along with JSON format, multipart messages shall be supported using:

- A multipart/related media type;

- 3gpp vendor specific content subtype; and

- Cross-referencing from the JSON payload using the Content-ID field.

Use of multipart messages is documented in specific specifications.

5.5 Interface Definition Language

OpenAPI Specification [9] shall be used as Interface Definition Language (IDL) of SBI.

6 General Functionalities in Service Based Architecture

6.1 Routing Mechanisms

6.1.1 General

For HTTP message routing between Network Functions, the message routing mechanism as specified in clause 5 of IETF RFC 7230 [12] is almost followed with some differences due to the adoption of HTTP/2 and to some 5G system specificities.

NOTE: The term "inbound" are defined in clause 2.3 of IETF RFC 7230 [12]. It describes a directional requirement in relation to the request route: "inbound" means toward the origin server.

6.1.2 Identifying a target resource

The target resource is identified by a target URI (e.g. a Resource URI, a Custom operation URI or a Callback URI as defined in clause 4.4 of 3GPP TS 29.501 [5]).

6.1.3 Connecting inbound

If the request is not satisfied by a local cache, then the client shall connect to an authority server for the target resource or to a proxy.

If a proxy is applicable for the target URI, the client connects inbound by establishing (or reusing) a connection to that proxy as defined in clause 5.2 of IETF RFC 7230 [12]. For connecting inbound to an authority not in the same PLMN, the client connects to the Security Edge Protection Proxy.

If no proxy is applicable, then the client connects directly to an authority server for the target resource as defined in IETF RFC 7230 [12].

6.1.4 Pseudo-header setting

6.1.4.1 General

Once an inbound connection is obtained, the client sends a request message over the wire. The message starts with a HEADERS frame containing the Pseudo-Header Fields identifying the request target. The ":method" pseudo-header is always present.

When sending a request directly to an origin server or to a proxy, other than a CONNECT or server-wide OPTIONS request, a client shall include the below pseudo-headers:

- ":scheme".

- ":authority".

- "path" includes the path and query components of the target URI. The path includes the optional deployment-specific string of the Resource URI or Custom operation URI "apiRoot" part.

When sending a CONNECT request to a proxy, a client shall include the ":authority" pseudo-header. The ":scheme" and ":path" ones shall be absent.

When sending a server-wide OPTIONS request to an origin server or to a proxy, a client shall include the below pseudo-headers:

- ":scheme".
- ":authority".
- "path" set with the value "*".

6.1.4.2 Routing within a PLMN

For HTTP/2 request messages where the target URI authority component designates an origin server in the same PLMN as the client, the ":authority" HTTP/2 pseudo-header field shall be set to:

":authority" = uri-host [":" port] as specified in clause 8.1.2.3 of IETF RFC 7540 [7], excluding the [userinfo "@"] information as specified in clause 3.2 of IETF RFC 3986 [14].

Where the uri-host shall be:

- FQDN of the target NF service; or
- IP address of the target NF service

The FQDN of the target NF service need not contain the PLMN identifier.

6.1.4.3 Routing across PLMN

In order to reach the correct target NF service in the right PLMN and for HTTP/2 request messages where the target URI authority component designates an origin server not in the same PLMN as the client, the ":authority" HTTP/2 pseudo-header shall contain the FQDN including the PLMN ID.

The ":authority" pseudo-header field in the HTTP/2 request message shall be set to:

":authority" = uri-host [":" port] as specified in clause 8.1.2.3 of IETF RFC 7540 [7], excluding the [userinfo "@"] information as specified in clause 3.2 of IETF RFC 3986 [14].

Where the uri-host shall be:

- FQDN of the target NF service or the FQDN (authority) part of a callback URI or a specified link relation

The FQDN of the target NF service or the FQDN (authority) part of a callback URI or a specified link relation shall contain the PLMN identifier.

The format of the FQDN of target NF service is specified in clause 28.5 of 3GPP TS 23.003 [15].

SEPP on the HTTP/2 client side shall form the telescopic FQDN, as specified in 3GPP TS 23.003 [15], for the following cases:

- FQDN of the target NF service in HPLMN is modified into a telescopic FQDN by the SEPP in the VPLMN;
- FQDN of the target NF service in VPLMN is modified into a telescopic FQDN by the SEPP in the HPLMN;
- FQDN (authority) part of callback URI of NF service resources in VPLMN is modified into a telescopic FQDN by the SEPP in the HPLMN;
- FQDN (authority) part of callback URI of NF service resources in HPLMN is modified into a telescopic FQDN by the SEPP in the VPLMN;
- FQDN (authority) part of link relation URI of NF service resources in VPLMN is modified into a telescopic FQDN by the SEPP in the HPLMN;
- FQDN (authority) part of link relation URI of NF service resources in HPLMN is modified into a telescopic FQDN by the SEPP in the VPLMN.

6.1.5 Host header

Clients that generate HTTP/2 requests shall use the ":authority" pseudo-header field instead of the Host header field.

6.1.6 Message forwarding

An HTTP/2 proxy shall use the ":authority" pseudo-header field to connect inbound to the origin server or another proxy if the request cannot be satisfied by the proxy cache.

An HTTP/2 proxy may also use other headers and/or payload content to connect inbound to the origin server or another proxy if the request cannot be satisfied by the proxy cache.

6.2 Server-Initiated Communication

The Subscribe-Notify service operations shall be supported between NFs as specified in clause 7.1.2 of 3GPP TS 23.501 [3].

Subscribe-Notify service operations require bidirectional communication between the NFs when the server needs to initiate communication with the client.

Subscribe-Notify service operations shall be supported with two TCP connections, one per direction, as follows:

- NF service consumer acts as an HTTP client and NF service producer acts as an HTTP server when NF service consumer subscribes to NF service producer's notifications;

- NF service producer acts as an HTTP client and NF service consumer acts as an HTTP server when NF service producer delivers notifications to NF service consumer.

6.3 Load Control

Load control for the service based interfaces, based on the current dynamic load on an NF and/or NF service instance, is not specified in this release of this specification.

During NF discovery procedures (see clause 4.17.4 and 4.17.5 of 3GPP TS 23.502 [4]), the NRF may provide the NF instance and/or the NF service instance information with the NF capacity information advertised during NF Service Registration and/or NF Service Update procedures (see clause 4.17.1 and 4.17.2 of 3GPP TS 23.502 [4] and clause 6.2.6 of 3GPP TS 23.501 [3]). The NF service consumer that is discovering the NF service producer, may use the NF capacity information to select the appropriate NF instance as specified in 3GPP TS 29.510 [8].

6.4 Overload Control

6.4.1 General

Service Based Interfaces use HTTP/2 over TCP for communication between the NF Services. TCP provides transport level congestion control mechanisms as specified in IETF RFC 5681 [16], which may be used for congestion control between two TCP endpoints (i.e., hop by hop). HTTP/2 also provides flow control mechanisms and limitation of stream concurrency, as specified in IETF RFC 7540 [7], that may be configured for connection level congestion control.

In addition to TCP and HTTP/2 congestion control mechanisms, end to end overload control shall be supported per NF service / API according to the below defined principles.

An NF Service Producer may mitigate a potential overload status by sending the NF Service Consumer the following HTTP status codes as a response to requests received during, or close to reaching, an overload situation:

- 503 Service Unavailable;

- 429 Too Many Requests; or

- 307 Temporary Redirect

The first 2 status codes (503 and 429) are intended to inform the NF Service Consumer that the server cannot handle the current received traffic rate, so it shall abate the traffic sent to the NF Service Producer by throttling part of this traffic locally at the NF Service Consumer, or diverting it to an alternative destination (another NF Service Producer where an alternative resource exists) that is not overloaded. If possible, traffic diversion shall always be preferred to throttling; the result of the throttling is a permanent rejection of the transaction.

If the client needs to abate a certain part of the available traffic, it shall do it based on the determined priority of each message.

Depending on regional/national requirements and network operator policy, requests related to priority traffic and emergency shall be the last to be throttled by the client, and shall be exempted from throttling due to overload control up to the point where the required traffic reduction cannot be achieved without throttling the priority requests.

The last status code (307) is intended to inform the NF Service Consumer about the availability of other endpoints where the service offered by the NF Service Producer is available, so the NF Service Consumer does not need to discard traffic locally.

6.4.2 HTTP Status Code "503 Service Unavailable"

This status code should be sent when the NF Service Producer undergoes an overload situation, and it needs to reject HTTP requests. The NF Service Producer may include detailed information about its status in the ProblemDetails JSON element, in the HTTP response body. Also, the HTTP header field "Retry-After" may be added in the response to convey an estimated time (in number of seconds) for the recovery of the service.

As for all 5xx status codes, this indicates a server-related issue (not limited to a specific client, or HTTP method), and it indicates that the server is incapable of performing the request.

Upon receipt of a "503 Service Unavailable" status code, the NF Service Consumer shall monitor the amount of rejected and timed-out traffic, in comparison to the accepted traffic by the NF Service Producer, and it shall abate (by divertion or throttling) the traffic sent to the NF Service Producer in such a way that the rate between accepted and rejected traffic improves with time, and eventually reaches a situation where the server accepts all requests once the overload status ceases at the server. The mechanism to achieve this is implementation-specific; Annex A contains a description of an example algorithm based on "adaptive throttling" of the traffic sent by the NF Service Consumer towards an NF Service Producer.

6.4.3 HTTP Status Code "429 Too Many Requests"

This status code may be sent, if supported by the server, when the NF Service Producer detects that a given NF Service Consumer is sending excessive traffic which, if continued over time, may lead to (or may increase) an overload situation in the NF Service Producer.

How the NF Service Producer detects that the incoming traffic comes from a same NF Service Consumer, and therefore subject to a given traffic rate limit, is out of the scope of this specification. The HTTP header field "Retry-After" may be added in the response to indicate how long the NF Service Consumer has to wait before making a new request.

As for all 4xx status codes, this indicates a client-related issue (not limited to a specific HTTP method), and it indicates that the client seems to be misbehaving.

6.4.4 HTTP Status Code "307 Temporary Redirect"

This status code should be sent when the NF Service Producer decides to redirect HTTP requests to another less loaded server, or HTTP/2 end point, to offload some part of the incoming traffic, with the goal to avoid entering (or to mitigate) an overload situation. The NF Service Producer shall not use it if it does not know the load status of the alternative server.

How the NF Service Producer becomes aware of the load levels of other servers or HTTP/2 end points is deployment-specific, and out of the scope of this specification. The URI for the temporary redirection shall be given by the Location header field of the response.

As for all 3xx status codes (redirection), this indicates a client-related action; the client shall be responsible of the detection of infinite redirection loops.

6.5 Support of Stateless NFs

6.5.1 General

A NF may become stateless by decoupling the "compute" resource and "storage" resource as specified in clause 4.1 of 3GPP TS 23.501 [3].

6.5.2 Stateless AMFs

6.5.2.1 General

AMF may become stateless by storing the UE related information in the UDSF. Procedures for AMF planned removal or the AMF auto-recovery are specified in clauses 5.21.2.2 and 5.21.2.3 of 3GPP TS 23.501 [3].

6.5.2.2 AMF as service consumer

1. When the AMF subscribes to notifications from another NF Service Producer, the AMF shall provide its GUAMI to the NF Service Producer to enable the latter to discover AMFs within the AMF set, or information about a backup AMF, in addition to Callback URI in the subscription resource.

   The AMF may also provide the name of the AMF service to which these notifications are to be sent (this service shall be one of the service produced by the AMF and registered in the NRF or a custom service registered in the NRF for the purpose of receiving these notifications);

NOTE 1: Providing an AMF service name allows the NF Service Producer to find the endpoint address to deliver the notifications (see bullet 2). The provided AMF service might not use itself the information received in these notifications.

2. A NF service producer may also use the Nnrf_NFDiscovery service to discover AMFs within an AMF set or backup AMF.

   If the AMF provided the name of its service (see bullet 1), the NF Service Producer shall look up for the same AMF service from the AMFs within the AMF set or from backup AMF, and use endpoint addresses (i.e. IP addresses, transport and port information, or FQDN) of that service to send notifications (see bullet 6). Otherwise, the notifications shall be sent to an endpoint address registered in the NF Profile of the AMF.

NOTE 2: The AMF can register different endpoint addresses in the NRF for different services.

3. An NF service producer may subscribe to GUAMI changes using the AMFStatusChange service operation of the Namf_Communication service.

4. An NF service producer may become aware of AMF changes (at the time of the AMF change or subsequently when sending signalling to the AMF) via Namf_Communication service AMFStatusChange Notifications, via HTTP Error response from the old or a wrongly selected new AMF, via link level failures (e.g. no response from the AMF), or via a notification from the NRF that the AMF has deregistered. The HTTP error response may be a 3xx redirect response pointing to a new AMF.

NOTE 3:. AMFs are identified by GUAMIs. A GUAMI can point to an individual AMF or to some or all AMFs within an AMF set. If a GUAMI points to several AMFs, and the UE is served by one of those, all those AMFs can immediately handle communication for that service, and the NF service producer does not need to be aware which of those AMFs is serving a UE.

5. When becoming aware of an AMF change, and the new AMF is not known, the NF service producer shall select an AMF within the AMF set or the possibly earlier received backup AMF.

6. When becoming aware of an AMF change, the NF service producer shall exchange the authority part of the Notification URI with new AMF information and shall use that URI in subsequent communication.

7. Each AMF within the AMF set shall be prepared to receive notifications from the NF service producer, by either handling the notification to the Notification URI constructed according to bullet 6 with the own address as authority part, or by replying with an HTTP 3xx redirect pointing to a new AMF, or by replying with another HTTP error.

8. The NF service producer shall be prepared to receive updates to resources of the related service from any AMF within the set.

9. If the UE moves to an AMF from a different AMF Set, or to an AMF from the same AMF set that does not support handling the notification as specified in bullet 7, the new AMF should update peer NFs with the new callback URI for the notification.

6.5.2.3 AMF as service producer

1. When AMF receives request to establish a service, it may provide information about a backup AMF in a suitable resource.

2. NF service consumer may also use the Nnrf_NFDiscovery service to discover AMFs within an AMF set.

3. An NF service consumer may subscribe to GUAMI changes using the AMFStatusChange service operation of the Namf_Communication service.

4. An NF service consumer may become aware of AMF changes (at the time of the AMF change or subsequently when sending signalling to the AMF) via Namf_Communication service AMFStatusChange Notifications, via Error response from the old or a wrongly selected new AMF, via link level failures (e.g. no response from the AMF), or via a notification from the NRF that the AMF has deregistered. The HTTP error response may be a 3xx redirect response pointing to a new AMF.

NOTE. AMFs are identified by GUAMIs. A GUAMI can point to an individual AMF or to some or all AMFs within an AMF set. If a GUAMI points to several AMFs, and the UE is served by one of those, all those AMFs can immediately handle communication for that service, and the NF service consumer does not need to be aware which of those AMFs is serving a UE.

5. When becoming aware of an AMF change, and the new AMF is not known, the NF service consumer shall select an AMF within the AMF set or the possibly earlier received backup AMF.

6. When becoming aware of an AMF change, the NF service consumer shall exchange the authority part of resource URIs with new AMF information and shall use that URI in subsequent communication.

7. Each AMF within the AMF set shall be prepared to receive updates for resources from the NF service consumer, by either handling the updates to the resource URIs constructed according to step 6 with the own address as authority part, or by replying with an HTTP 3xx redirect pointing to a new AMF, or by replying with another HTTP error.

8. For a service that includes notifications from the AMF, the NF service consumer shall be prepared to receive for the that service notifications from any AMF within the set.

NOTE: If the UE moves to an AMF from a different AMF Set, or to an AMF from the same AMF set that does not support handling the updates as specified in bullet 7, but mechanisms exist to transfer information related to the resource to the AMF, service specific mechanism can exist.to notify the NF service consumer about the resource at the AMF. For instance, for the Namf_EventExposure service, information and an event subscription is transferred to the new AMF in such a manner and the new AMF will then report an event-change event.

6.6 Extensibility Mechanisms

6.6.1 General

This clause describes the extensibility mechanisms supported in the Service-Based Architecture in 3GPP 5GC, such as feature negotiation, vendor-specific extensions, etc.

6.6.2 Feature negotiation

A versioning of services in the request URI shall be supported by 3GPP 5G APIs, but version upgrades shall only be applied for non-backward compatible changes or the introduction of new mandatory features.

The following mechanism to negotiate applicable optional features shall be used by 5G APIs. This supported feature mechanism shall be applied separately for each API.

For any API that defines resources, suitable resources associated to or representing the NF Service Consumer (e.g. a top-level resource or a sub-resource representing the NF Service Consumer) shall be identified in each API to support the negotiation of the applicable optional features between the NF Service Consumer and NF Service Producer for this resource. Each such resource for a 5G API shall contain an attribute (e.g. "supportedFeatures") of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] containing a bitmask to indicate supported features. The features and their positions in that bitmask are defined separately for each API.

The HTTP client acting as NF service consumer shall include the attribute of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] in the HTTP PUT or POST requests to create the resource associated to or representing the NF Service Consumer of 5G API. This attribute indicates which of the optional features defined for the corresponding service are supported by the HTTP client. The HTTP server shall determine the supported features for the corresponding resource by comparing the supported features indicated by the client with the supported features the HTTP server supports. Features that are supported both by the client and the server are supported for that resource. The HTTP server shall include the attribute of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] indicating those features in the representation of the resource it returns to the HTTP client in the HTTP response confirming the creation of the resource.

The HTTP client acting as NF service consumer may include a query parameter (e.g. "supported-features") of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] in HTTP GET requests to fetch resource(s) associated to the NF Service Consumer of 5G API. This query parameter indicates which of the optional features defined for the corresponding service are supported by the HTTP client. The HTTP server shall determine the supported features for the corresponding resource(s) by comparing the supported features indicated by the client with the supported features the HTTP server supports. Features that are supported both by the client and the server are supported for the resource(s); attributes or enumerated values that are only of relevance to a feature unsupported by the requested resource(s) should be ommitted from the representation sent in the response. The HTTP Server shall include the attribute of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] indicating those features in the HTTP GET response, if supported by the API definition.

The supported features for a resource associated to or representing the NF Service Consumer shall also be applicable to all subordinate resources of that resource, and for all custom operations related to any of those resources. If any of those resources is used for the subscription to notifications (see clause 4.6.2 of 3GPP TS 29.501 [5]), the supported features shall also apply to those notifications.

Attributes used for the representation of a resource, particular values in enumerated data types, and/or procedural description can be marked to relate to a particular supported feature. Such attributes shall not be mandated in data structures. Such attributes or enumerated values shall only be sent and such procedures shall only be applied if the corresponding feature is supported.

Unknown attributes and values shall be ignored by the receiving entity. Unsupported query parameters shall be handled as specified in clause 5.2.9.

NOTE: The sender may send such information before the supported features for a resource have been determined.

For an API that does not define any resources, only custom operations without associated resources or notifications without subscription will be used. For such APIs, if a feature negotiation is desired, the request and response bodies of a suitable custom operation or notification need to be defined in such a manner that an attribute of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] is included. The client invoking that custom operation or notification shall indicate its supported features for that API within the corresponding HTTP request. The data structures to be included in the HTTP request as defined for that API, shall include the attribute of the SupportedFeatures data type defined in 3GPP TS 29.571 [13], preferably in the outermost data structure for cases where the body contains a complex structure with several layers of JSON objects. The server shall determine the supported features by comparing the supported features indicated by the client with its own supported features. Features that are supported both by the client and the server are supported for subsequent custom operations and notifications of that API. The server shall include the attribute of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] indicating those features in the successful response to the custom operation or notification. The data structures to be included in the HTTP response as defined for that API, shall include the attribute of the SupportedFeatures data type defined in 3GPP TS 29.571 [13], preferably in the outermost data structure for cases where the body contains a complex structure with several layers of JSON objects. The client and server shall only use those supported features in subsequent communication of that API between each other until the supported feature negotiation performed as part of that communication yields a new result.

Additionally, a NF instance should register all the features it supports to the NRF, to enable NF Service Consumers to discover NF Service Producers supporting specific features.

6.6.3 Vendor-specific extensions

Information elements sent on the 3GPP 5GC APIs should be extensible with vendor-specific data. The definition of JSON data structures using OpenAPI as Interface Definition Language (see OpenAPI Specification [9]) allows to extend by default any JSON object with additional member elements, as long as no specific directives are included in the schema definition preventing such extension (e.g., by setting "additionalProperties" to false).

NOTE 1: The only JSON data types that can be extended, by defining additional members, are JSON objects; simple data types (and arrays of items of simple data types) cannot be extended in this way.

However, in order to avoid duplication of member names inside a same object (see 3GPP TS 29.501 [5], clause 5.2.4.2, for the requirement of uniqueness of member names in JSON objects), it is necessary to comply with a naming scheme for vendor-specific data elements, to avoid clashing names between vendors.

Vendor-specific member names in JSON objects shall be named in the following manner:

```
"vendorSpecific-nnnnnn": {
  ...
}
```

where the value "nnnnnn" is a fixed 6-digit string, using the IANA-assigned "SMI Network Management Private Enterprise Codes" [18] value associated to a given vendor, and padding with leading digits "0" to complete a 6-digit value.

NOTE 2: The content (value) of those vendor-specific member elements, and their usage, is not to be defined by any of the 3GPP Technical Specifications. Also, the type of value assigned to these members is not defined by 3GPP, and therefore, they can be any of the types allowed in the JSON specification: objects, arrays, or simple types (string, number, Boolean, etc.). However, to allow future extensibility of these values, it is recommended that they are defined as objects.

EXAMPLE: The vendor-specific member name for vendor "3GPP" would be:

```
"vendorSpecific-010415": {
  ...
}
```

6.6.4 Extensibility for Query parameters

New query parameters may be defined after the OpenAPI freeze of the first 3GPP release that contains an API.

A new feature should be defined in the API for any query parameter added in a new version of the API or for any new functionality resulting in defining new query parameter(s). A single feature may be defined, if appropriate, when adding multiple query parameters in a new version of the API.

Prior to using such a query parameter in an HTTP request, the NF Service Consumer should determine, if possible, whether the query parameter is supported by the NF Service Producer, using the feature negotiation mechanism specified in clause 6.6.2.

NOTE 1: Not doing so could result in the NF Service Producer rejecting the request if it does not support the query parameters, see clause 5.2.9.

NOTE 2: A NF Service Consumer can discover the features (and therefore the query parameters) supported by a NF Service Producer using the NRF, if the latter has registered the features it supports to the NRF.

If the NF Service Consumer includes the query parameter (e.g. "supported-features") of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] in an HTTP GET request (see clause 6.6.2), the NF Service Producer shall include the attribute (e.g. "supportedFeatures") of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] in the HTTP GET response, set as defined for HTTP responses in clause 6.6.2, if supported by the API definition.

NOTE 3: This allows a NF Service Consumer to discover the features (and therefore the query parameters) supported by the NF Service Producer when the first interaction with the NF Service Producer is an HTTP GET request and the service was not discovered via the NRF, e.g. for a NF Discovery Request sent to the NRF.

NOTE 4: Some APIs are designed to allow returning the attribute of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] in the HTTP GET response, regardless of whether the query parameter of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] is present in the request.

If a NF Service Consumer uses such a query parameter in an HTTP GET request without prior knowledge of whether it is supported by the NF Service Producer, the NF Service Consumer shall be prepared to receive a successful response that may not match all the query parameters sent in the request, and to act accordingly. The NF Service Consumer may use the attribute of the SupportedFeatures data type defined in 3GPP TS 29.571 [13] returned by the NF Service Producer in the HTTP GET response, if available, to determine the features (and thus query parameters) not supported by the NF Service Producer.

When defining new query parameters in a new version of an API, it needs to be checked that the addition of the query parameter does not cause backward compatibility problems with NF Service Producers complying with an earlier version of the API, e.g. if the query parameter is ignored in a HTTP GET request. Otherwise, it needs to be ascertained that the NF Service Consumer does not use such a query parameter without prior knowledge that it is supported by the NF Service Producer.

6.7 Security Mechanisms

6.7.1 General

The security mechanisms for service based interfaces are specified in clause 13 of 3GPP TS 33.501 [17].

Security Protection Edge Proxy (SEPP), as specified in 3GPP TS 33.501 [17], shall be used between service based interfaces across PLMNs. The NFs in a PLMN shall use the SEPP as a HTTP/2 proxy for the HTTP/2 messages that carry ":authority" pseudo header with a uri-host formatted as specified in clause 6.1.4.3.

6.7.2 Transport layer security protection of messages

As specified in clause 13.1 of 3GPP TS 33.501 [17], TLS shall be used for the security protection of messages at the transport layer for the service based interfaces if network security is not provided by other means.

6.7.3 Authorization of NF service access

As specified in clause 13.4.1 of 3GPP TS 33.501 [17] OAuth 2.0 (see IETF RFC 6749 [22]) may be used for authorization of NF service access. All NFs and the NRF shall support the OAuth 2.0 authorization framework with "Client Credentials" grant type as specified in clause 4.4 of IETF RFC 6749 [22] , except that there is no "Authorization" HTTP request header in the access token request.

The NRF shall act as the Authorization Server providing "Bearer" access tokens (IETF RFC 6750 [23]) to the NF service consumers to access the services provided by the NF service providers.

If an NF service (i.e API) receives an OAuth 2.0 access token in the "Authorization" HTTP request header field, the NF service shall validate the access token, its expiry and its access scope before allowing access to the requested resource, as specified in clause 7 of IETF RFC 6749 [22].

An NF service consumer shall support OAuth 2.0.

For request/response semantics service operations and for the subscribe and unsubscribe operations of subscribe/notify semantics service operations, an NF service consumer may use OAuth 2.0 for the authorization of the API access, based on local configuration.

When Oauth2 authorization is used, the NF service consumer shall use the token received from NRF as a "Bearer" token and include it in the Authorization header of the HTTP service requests, as described in IETF RFC 6750 [23] clause 2.1.

An NF service producer shall decide to accept or reject an API request without the OAuth2.0 access token in the "Authorization" HTTP request header field, based on local configuration.

If an NF service producer rejects an API request without the OAuth2.0 access token or an API request with an invalid OAuth2.0 access token, it shall return the HTTP "401 Unauthorized" status code together with the "WWW-Authenticate" header as specified in IETF RFC 7235 [21], where:

- the scheme for challenge in the "WWW-Authenticate" header shall be set to "Bearer" (IETF RFC 6750 [23]).
- the "realm" attribute shall be set to the URI of the service (i.e API URI) for which the access failed, in the case of request / response service operations.
- the "error" attribute shall be set to "invalid_token", as described in IETF RFC 6750 [23], if the request contained a token which was deemed as invalid by the NF service producer (e.g. it is expired, malformed...); if the request did not contain a token, the "error" attribute shall not be included.

For the notify operation of subscribe/notify semantics service operations, in this release of this specification OAuth 2.0 access token is not used.

When an NF service consumer receives a "401 Unauthorized" status code with a "WWW-Authenticate" header containing "Bearer" as the scheme for challenge it shall not repeat the same request without an OAuth2.0 access token or with an access token that has been already used. The NF service consumer may repeat the same request with a new OAuth 2.0 access token.

NOTE: If a NF service producer accepts a request without the OAuth 2.0 access token, based on local policy, it is assumed that such accesses are allowed based on trust relationships and hence full access to the resource as it would have been otherwise allowed, is provided.

6.7.4 Application layer security across PLMN

6.7.4.1 General

HTTP/2 messages sent across the PLMN between the SEPPs shall follow the application layer security procedures specified in clause 13.2 of 3GPP TS 33.501 [17].

6.7.4.2 N32 Procedures

As specified in clause 13.2 of 3GPP TS 33.501 [17], the following procedures shall be supported across N32

- Capability Negotiation Request and Response;
- Parameter Exchange Request and Response;
- forwarding of the JOSE (see IETF RFC 7516 [25] and IETF RFC 7515 [26]) protected messages over N32.

Based on the capability negotiation and parameters exchanged between the SEPPs, the service based interface messages sent across N32 interface shall be subjected to JOSE based protection (see IETF RFC 7516 [25] and IETF RFC 7515 [26]) as specified in clause 13.2.4 of 3GPP TS 33.501 [17].

3GPP TS 29.573 [27] specifies protocol for the exchange of the messages described above over N32, the format of the JOSE (see IETF RFC 7516 [25] and IETF RFC 7515 [26]) protected messages and the procedure for forwarding of the JOSE protected messages over N32.

6.8 SBI Message Priority Mechanism

6.8.1 General

The primary usage of SBI Message Priority (SMP) is to provide guidance to 5GC NF acting as HTTP/2 clients or servers and HTTP/2 proxies when making throttling decisions related to overload control. The priority information may also be used for routing in proxies. Eventually a server may use the priority information to process higher-priority requests before lower-priority requests.

The SMP mechanism defined in this clause uses the "3gpp-Sbi-Message-Priority" custom HTTP header defined in clause 5.2.3.2.1 to set and carry the message priority between the client and the server.

NOTE 1: The custom HTTP header enforces the message priority end to end between the client and the server through one or more proxies.

The SMP mechanism should also use the stream priority mechanism specified in IETF RFC 7540 [7] clause 5.3.

NOTE 2: The stream priority enforces the message priority at the HTTP/2 connection level not end to end.

HTTP/2 clients, servers and proxies implementing SBIs shall support the custom HTTP header and should support the stream priority.

6.8.2 Message level priority

A client, proxy and server shall use the "3gpp-Sbi-Message-Priority" value (see clause 5.2.3.2.1) when setting or evaluating the priority of a message.

The client shall assign the request priority by adding the "3gpp-Sbi-Message-Priority" custom HTTP header (see clause 5.2.3.2.1) to the message and setting its value.

If the "3gpp-Sbi-Message-Priority" custom HTTP header is not present in a response message then the HTTP nodes shall use the priority indicated in the "3gpp-Sbi-Message-Priority" of the associated request message.

If the server wants to assign a different priority to the response message than the request one then the server shall assign the response priority by adding the "3gpp-Sbi-Message-Priority" custom HTTP header to the message and setting its value.

6.8.3 Stream priority

The purpose of HTTP/2 stream priority is to allow an endpoint to prioritize streams for transmitting frames when there is limited capacity for sending and to express how it would prefer its peer to allocate resources when managing concurrent streams. Setting the stream priority ensures a priority treatment to a message between the two endpoints of an HTTP/2 connection.

The stream priority applies to all frames in both directions. If it is not changed until the stream is closed then all frames of the request and response messages will have the same priority. A client assigns a priority for a request and the correspondent response by including dependency and Weight information in the HEADERS frame that opens the stream carrying the message.

The stream dependency shall be set to 0.

If the stream priority is used then the stream priority Weight is mapped from the custom HTTP header. The mapping algorithm shall respect the ordering of the priority. If message 1 has a priority of "x" and message 2 has a priority of "y" where "x" is lower than "y" then the Weight of the stream carrying the message 1 shall be higher than the Weight of the stream carrying the message 2.

If the server wants to change the priority of the response, it shall send a PRIORITY frame after the stream state became "half-closed (remote)" or shall send the priority information in the HEADERS frame.

6.8.4 Recommendations when defining SBI Message Priorities

The recommendations provided in this clause are compliant with clause 10 of IETF RFC 7944 [19]. They have been adapted to 5G services and Service Based Architecture.

The priorities defined for all messages across all SBIs used in an HTTP/2 administrative domain must be defined in a consistent and coordinated fashion, taking the default priority (see below for default priority values) into account.

The following are some guidelines to be considered when defining the SMPs to be used in SBA networks that support HTTP/2 nodes handling multiple services.

- As with any prioritization scheme, it is possible for higher-priority messages to block lower-priority messages from ever being handled. In 5GC, this will often result in the messages being retried. This may result in more traffic than the network would have handled without use of the SMP mechanism.

One potential guideline to prevent unwanted starving of lower-priority messages is to have higher-priority messages represent a relatively small portion of messages handled by the 5GC under normal scenarios. Multimedia Priority Service (see 3GPP TS 23.501 [5] clause 5.16.5) and Mission Critical Service (see 3GPP TS 23.501 [5] clause 5.16.6) typically generate little traffic compared to the total traffic of a 5GC.

Multimedia Priority Service (MPS) or Mission Critical Service (MCX) requires the blocking of lower-priority services.

- When setting priorities for Multimedia Priority Services, Mission Critical Services or Emergency calls, it is important to use the same priority values across all APIs and services exposed by the 5GC. For instance, if it is defined that the MPS priority level of [1; n] shall be assigned the priority of [k; k+n-1] in the same order then it shall be the same on all SBIs.

- Messages related to MPS, MCX and Emergency calls may be ranked according to their priority (e.g., based on ARP priority level, 5QI priority level, MPS Priority) based on regional/national regulatory and operator policies when it is known by the application sending the message. Otherwise MPS and MCX should have higher priorities than Emergency calls. Emergency call related messages should have higher priority than the rest of the messages.

NOTE: In some situations (e.g. REGISTRATION or SERVICE REQUEST procedure); it is possible to identify that the message belongs to a procedure of a high priority user without knowing the identity of the priority service. In such a case, all the messages sent over an SBI of these high priority procedures should be given the same SBI message priority.

- Requests without the "3gpp-Sbi-Message-Priority" header shall be assigned the default priority value of "24".

- Streams without priority shall be assigned a Stream Dependency of 0x0 and the default Weight of 16.

- When defining priorities of the messages of a service it is needed to follow the same rules independently of the application, the SBI and the service.

  - When there is a series of request/response required to complete a procedure, it is appropriate to mark request/response occurrences that occur later in the series at a higher priority than those that occur early in the series.

  - The requests that establish new sessions should have a lower priority than the ones that update or end a session.

  - The requests that will result on deregistering users if they failed (authentication vector retrieval, update location…) shall have a higher priority than the ones of a non-registered user.

  - Request/response of optional procedure and delay tolerant services should have lower priority than those of mandatory procedures.

6.8.5 HTTP/2 client behaviour

The client sending a request shall determine its required priority according to 6.8.4. It shall include a "3gpp-Sbi-Message-Priority" header (see clause 5.2.3.2.1) indicating the required priority level in the request and shall prioritise the requests according to the required priority level. If the client also uses the stream priority at the HTTP/2 connection level then it shall map the header value into a Weight and include it in the HEADERS of the request message.

When the client receives a response with the "3gpp-Sbi-Message-Priority" header, it shall prioritise the received response according to the priority level received, otherwise according to the priority level of the corresponding request. This includes determining the order in which responses are handled and resources that are applied to the handling of the responses. The client may use the stream priority to determine how to prioritize the response at the HTTP/2 connection level.

6.8.6 HTTP/2 server behaviour

The server should use the "3gpp-Sbi-Message-Priority" header (see clause 5.2.3.2.1) and may use the stream priority information to determine how to handle the request. This includes determining the order in which requests are handled and resources that are applied to the handling of the request.

Servers should use "3gpp-Sbi-Message-Priority" value when making overload throttling decisions.

Servers should use stream priority information when making overload throttling decisions at the connection level.

When the priority of the response message needs to have a different value than the request, a server shall include a "3gpp-Sbi-Message-Priority" header in the response message which value is set to the response required priority level.

If a server has included "3gpp-Sbi-Message-Priority" header in the response message it may also set the stream priority as described in IETF RFC 7540 [7], via priority information in the HEADERS frame or in a PRIORITY frame. In both cases the priority Weight value shall be mapped from the 3gpp-Sbi-Message-Priority" header value. When sending the priority information with a PRIORITY frame the server shall send it before sending the HEADERS frame of the response message. A server shall not send a PRIORITY frame after the HEADER one.

6.8.7 HTTP/2 proxy behaviour

A proxy should forward request and response without removing the "3gpp-Sbi-Message-Priority" header or changing its value.

While done only in exceptional circumstances, a proxy may modify priority information when relaying request and response by changing the "3gpp-Sbi-Message-Priority" value. For example, a SEPP may modify the priority set by a roaming partner.

Proxies should use the request priority information (respectively response priority information) according to the "3gpp-Sbi-Message-Priority" value and may use the stream priority Weight value when making overload throttling decisions to a request (respectively a response).

Proxies may use the priority information according to the "3gpp-Sbi-Message-Priority" value and may use the stream priority Weight value when relaying a request or a response messages. This includes the selection of routes (only for the requests) and the ordering of messages relayed.

6.8.8 DSCP marking of messages

A client, proxy or server may prioritize traffic at IP level by placing messages into different traffic classes and marking them with an appropriate Differentiated Services Code Point (DSCP).

Multiple HTTP/2 connections between two HTTP/2 end points are necessary: one per DSCP value. All messages sent over a connection are assigned the same traffic class and receive the same DSCP marking. The "3gpp-Sbi-Message-Priority" value shall be considered in the selection of the appropriate connection to send the message.

6.9 Discovering the communication options supported by a target resource

6.9.1 General

The OPTIONS method, as described in clause 4.3.7 of IETF RFC 7231 [11], may be used by a NF Service Consumer to determine the communication options supported by a NF Service Producer for a target resource.

Clause 6.9.2 describes example communication options that may be discovered using the OPTIONS method.

6.9.2 Discoverable communication options

6.9.2.2 Content-encodings supported in HTTP requests

Certain service operations may result in large HTTP request payloads, e.g. to register NF profiles in the NRF (see 3GPP TS 29.510 [8]) or to update the NSSF with the available S-NSSAIs supported by Tracking Areas (see 3GPP TS 29.531 [32]). Gzip coding (see IETF RFC 1952 [34]) may be supported to optimally reduce the payload size of HTTP requests in this case.

A NF Service Consumer may determine the content-encodings supported by the NF Service Producer in HTTP requests targeting a particular resource by:

- sending an HTTP OPTIONS request targeting the resource of the NF Service Producer; and/or
- receiving an "Accept-Encoding" header in HTTP responses from the NF Service Producer for requests targeting the resource.

A NF Service Producer that receives an HTTP OPTIONS request for a target resource shall include an "Accept-Encoding" header in the HTTP 200 OK response (see IETF RFC 7694 [33]), if specific content-encodings, e.g. Gzip coding (e.g. see IETF RFC 1952 [34]) are supported in HTTP requests targeting the resource.

A NF Service Producer that receives an HTTP request with a content-encoding that it does not support shall reject the request with the status code "415 Unsupported Media Type" and include an "Accept-Encoding" header in the response indicating the supported encodings in HTTP requests, as described in clause 3 of IETF RFC 7694 [33].

A NF Service Producer may include an "Accept-Encoding" header in the HTTP 2xx response for requests other than HTTP OPTIONS if specific content-encodings, e.g. Gzip coding (e.g. see IETF RFC 1952 [34]), are supported in HTTP requests targeting the resource, to optimize future interactions, e.g. when the request payload was big enough to justify use of a compression coding but the client did not do so.

Annex A (informative): Internal NF Routing of HTTP Requests

The internal details of the architecture of a Network Function instance are out of the scope of 3GPP and are entirely implementation-specific. This annex describes how an instance of an NF Service Producer can route internally HTTP requests received on a given Service-Based Interface.

Figure A-1 illustrates an example component architecture where incoming HTTP requests are received and processed in a component named as "Ingress Proxy" module and route them to the appropriate computing resource in the NF.

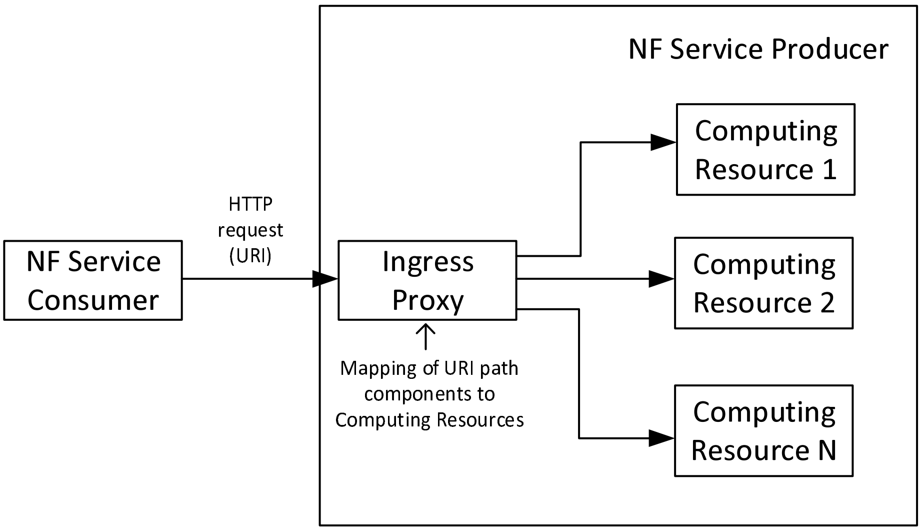

Figure A-1: Internal message routing inside NF Service Producer

The Ingress Proxy may parse any of the different components in the HTTP request, but typically it may parse the path of the URI (i.e. the :path pseudo-header in the HTTP/2 request).

The path component of the URI contains the service name of the requested SBA service, so frequently the routing is done based on this component.

It is also frequent to inspect other components of the path (i.e. path segments), to do a more fine-grained routing and direct requests done on a specific HTTP resource(s) towards a given computing resource(s).

It can be noted that the path components used to determine the target computing resource typically do not need to be statically defined but are frequently defined in terms of "variables", or placeholders, similarly to how they are defined in the OpenAPI specification language (a mechanism usually known as "path templating"). See: https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.0.md#path-templating

Annex B (informative): Client-side Adaptive Throttling for Overload Control

This clause contains an example algorithm to make an NF Service Consumer adjust the traffic rate sent to an NF Service Producer based on the number of received "rejects" of HTTP requests with a status code "503 Service Unavailable", or requests that have timed-out and the response was never received. This algorithm is described in the book "Betsy Beyer, et al; Google: Site Reliability Engineering" (https://landing.google.com/sre/book.html), chapter 21, "Handling Overload".

NOTE: The reference link provided to the book can change and hence the name of the book is expected to be used for referring to the latest edition.

Each client (NF Service Consumer) keeps track of the following counters during a certain time window:

- Requests: The number of requests that the client (NF Service Consumer) needs to handle. Under normal operation (no overload), all these requests are sent to the server (NF Service Producer). Under an overload situation, part of these requests are locally rejected by the client (and not sent to the server), and the rest of the requests are sent to the server.

- Accepts: The number of requests accepted by the server (i.e., requests for which a response has been effectively received at the client, with a status code other than "503 Service Unavailable").

When there is no server overload, these values are equal.

When there is an overload status in the server, the rate between "Accepts" and "Requests" decreases progressively. When this rate falls below a certain point (given by an algorithm parameter named "K"), the client shall start dropping some requests locally and not send them to the server.

The local rejection of requests can be done by calculating a "Client request rejection probability", as:

$$\max(0, \frac{\text{requests} - K \times \text{accepts}}{\text{requests} + 1})$$

So, for example, assuming that the K parameter is set at 1.5:

- if the server accepts >67% of the traffic, and rejects <33% of the traffic, the client does not take any throttling action, and keeps sending to the server all the traffic it has available for processing

- if, during a first time-window, the server accepts, e.g., only 60% of the requests, and rejects 40% due to overload, the application of this algorithm implies that the client must drop locally 10% of the requests (probabilistically), and only send to the server the remainder 90% of its traffic.

- if, during a second time-window, the client keeps the same amount of available traffic to handle, but the server continues rejecting requests with same rate as before (40%) of the received requests, the application of the algorithm again, results in increasing the drop rate to 14.5%, and sending to the server only 85.5% of the available traffic.

The value of the parameter K, along with the size of the time window during which the total number of "requests" and "accepts" is accounted for, has a fundamental role on how the algorithm behaves. If K is higher, the algorithm is more "permissive", and the client does not start dropping requests locally until the rejection rate is higher (e.g., >50%, for K = 2); if K is lower, the algorithm is more "aggressive", and the client starts dropping requests sooner (e.g., K = 1.1 implies to start dropping requests as soon as the server rejects >10% of the requests).

Annex B (normative): 3gpp-Sbi-Callback Types

This annex specifies the list of allowed 3GPP SBI callback type values for the 3gpp-Sbi-Callback HTTP custom header specified in clause 5.2.3.2.3. Only the callbacks that are invoked across PLMN are specified.

Table B-1: Values for 3gpp-Sbi-Callback Custom HTTP Header

Value for 3gpp-Sbi-Callback Custom HTTP Header	Reference
"Nsmf_PDUSession_Update"	3GPP TS 29.502 [28], Clause 5.2.2.8.3.2, 5.2.2.8.3.3, 5.2.2.8.3.4 and 5.2.2.8.3.5.
"Nsmf_PDUSession_StatusNotify"	3GPP TS 29.502 [28], Clause 5.2.2.10.
"Nudm_SDM_Notification"	3GPP TS 29.503 [29], Clause 6.1.5.2
"Nudm_UECM_DeregistrationNotification"	3GPP TS 29.503 [29], Clause 6.2.5.2
"Nudm_UECM_PCSCFRestorationNotification"	3GPP TS 29.503 [29], Clause 6.2.5.3
"Nnrf_NFManagement_NFStatusNotify"	3GPP TS 29.510 [8], Clause 6.1.5.2.
"Namf_EventExposure_Notify"	3GPP TS 29.518 [31], Clause 6.2.5.2.
"Npcf_UEPolicyControl_UpdateNotify"	3GPP TS 29.525 [35], Clauses 4.2.4, 5.5.2 and 5.5.3.
"Nnssf_NSSAIAvailability_Notification"	3GPP TS 29.531 [32], Clause 6.2.5.2

Annex C (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2017-10	CT4#80	C4-175246				TR skeleton	0.1.0
2017-10	CT4#80	C4-175390				Implementation of pCRs agreed at CT4#80.	0.2.0
2017-12	CT4#81	C4-176433				Implementation of pCRs agreed at CT4#81.	0.3.0
2018-01	CT4#82	C4-181387				Implementation of pCRs agreed at CT4#82.	0.4.0
2018-03	CT4#83	C4-182430				Implementation of pCRs agreed at CT4#83.	0.5.0
2018-03	CT#79	CP-180028				Presented for information	1.0.0
2018-04	CT4#84	C4-183512				Implementation of pCRs agreed at CT4#84.	1.1.0
2018-05	CT4#85	C4-184617				Implementation of pCRs agreed at CT4#85. The following pCRs are implemented. C4-184589, C4-184580, C4-184347, C4-184590, C4-184338, C4-184591, C4-184349, C4-184490, C4-184350, C4-184579, C4-184577 and C4-184498.	1.2.0
2018-06	CT#80	CP-181098				Presented for approval	2.0.0
2018-06	CT#80					Apprvoed in CT#80	15.0.0
2018-09	CT#81	CP-182053	0001	4	F	OAuth2.0 Clarifications	15.1.0
2018-09	CT#81	CP-182053	0002	2	B	Specifying N32 Aspects	15.1.0
2018-09	CT#81	CP-182053	0003	1	F	Determination of SBI message priorities	15.1.0
2018-09	CT#81	CP-182053	0004	5	F	Stateless AMF support	15.1.0
2018-09	CT#81	CP-182053	0005		F	Support of status code "501 Not implemented"	15.1.0
2018-09	CT#81	CP-182053	0006	2	B	Default port number	15.1.0
2018-12	CT#82	CP-183011	0009	3	F	Keep-alive on idle HTTP connections	15.2.0
2018-12	CT#82	CP-183011	0010	1	F	Stream Concurrency for overload control	15.2.0
2018-12	CT#82	CP-183011	0011	1	F	Update of missing status code 429	15.2.0
2018-12	CT#82	CP-183011	0012	1	F	Correction of the entity upon which content encoding is performed	15.2.0
2018-12	CT#82	CP-183011	0013	2	F	Custom header for notifications	15.2.0
2018-12	CT#82	CP-183011	0014	3	F	Routing across PLMN	15.2.0
2018-12	CT#82	CP-183011	0015		F	HTTP status code "406 Not Acceptable"	15.2.0
2018-12	CT#82	CP-183011	0016	1	F	Support of HTTP status code "414 URI Too Long"	15.2.0
2018-12	CT#82	CP-183011	0018		F	HTTP status code "414 URI Too Long" on PUT method	15.2.0
2018-12	CT#82	CP-183011	0020	1	F	Correction of Stream Priority in HTTP/2 Server Behaviour	15.2.0
2018-12	CT#82	CP-183194	0022	2	F	Change 403 to mandatory and clarify conditional headers	15.2.0
2018-12						Change history annex number corected	15.2.1
2019-03	CT#83	CP-190016	0023	1	F	Extensibility mechanism for Query parameters	15.3.0
2019-03	CT#83	CP-190016	0024	1	F	Bearer Tokens	15.3.0
2019-03	CT#83	CP-190016	0025	1	F	Handling of Incorrect IEs	15.3.0
2019-03	CT#83	CP-190016	0026	2	F	Clarification on Handling of Incorrect Optional IEs	15.3.0
2019-03	CT#83	CP-190016	0027		F	Status Codes	15.3.0
2019-06	CT#84	CP-191027	0030	1	F	Content-encodings supported in HTTP requests	15.4.0
2019-06	CT#84	CP-191027	0031	3	F	Missing Application Error Codes	15.4.0
2019-06	CT#84	CP-191027	0032	2	F	Correction on Feature Negotiation	15.4.0
2019-06	CT#84	CP-191027	0037	1	F	Allowed values of 3gpp-Sbi-Callback header field	15.4.0
2019-06	CT#84	CP-191027	0038	1	F	Adding the Control Plane interfaces that support service based interface	15.4.0
2019-09	CT#85	CP-192100	0050	1	F	Informative description of internal NF routing of HTTP messages	15.5.0
2020-06	CT#88e	CP-201017	0129	1	F	WWW-Authenticate Header	15.6.0

History

Document history		
V15.0.0	July 2018	Publication
V15.1.0	October 2018	Publication
V15.2.1	April 2019	Publication
V15.3.0	July 2019	Publication
V15.4.0	July 2019	Publication
V15.5.0	September 2019	Publication
V15.6.0	July 2020	Publication