ETSI TS 129 510 V16.4.0 (2020-07)

5G;
5G System;
Network function repository services;
Stage 3
(3GPP TS 29.510 version 16.4.0 Release 16)

Reference
RTS/TSGC-0429510vg40

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

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.

In the present document, modal verbs have the following meanings:

shall indicates a mandatory requirement to do something

shall not indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

should indicates a recommendation to do something

should not indicates a recommendation not to do something

may indicates permission to do something

need not indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

can indicates that something is possible

cannot indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

will indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

will not indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

might indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

might not indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

is (or any other verb in the indicative mood) indicates a statement of fact

is not (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

1 Scope

The present document specifies the stage 3 protocol and data model for the Nnrf Service Based Interface. It provides stage 3 protocol definitions and message flows, and specifies the API for each service offered by the NRF.

The 5G System stage 2 architecture and procedures are specified in 3GPP TS 23.501 [2] and 3GPP TS 23.502 [3].

The Technical Realization of the Service Based Architecture and the Principles and Guidelines for Services Definition are specified in 3GPP TS 29.500 [4] and 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 23.501: "System Architecture for the 5G System; Stage 2".

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

[4] 3GPP TS 29.500: "5G System; Technical Realization of Service Based Architecture; Stage 3".

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

[6] 3GPP TS 29.518: "5G System; Access and Mobility Management Services; Stage 3".

[7] 3GPP TS 29.571: "5G System; Common Data Types for Service Based Interfaces; Stage 3".

[8] ECMA-262: "ECMAScript® Language Specification", https://www.ecma-international.org/ecma-262/5.1/.

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

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

[11] IETF RFC 7807: "Problem Details for HTTP APIs".

[12] 3GPP TS 23.003: "Numbering, Addressing and Identification".

[13] IETF RFC 6902: "JavaScript Object Notation (JSON) Patch".

[14] IETF RFC 6901: "JavaScript Object Notation (JSON) Pointer".

[15] 3GPP TS 33.501: "Security architecture and procedures for 5G system".

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

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

[18] IETF RFC 4122: "A Universally Unique IDentifier (UUID) URN Namespace".

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

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

[21] 3GPP TS 29.244: "Interface between the Control Plane and the User Plane Nodes; Stage 3".

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

[23] IETF RFC 2782: "A DNS RR for specifying the location of services (DNS SRV)".

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

[25] IETF RFC 7519: "JSON Web Token (JWT)".

[26] W3C HTML 4.01 Specification, https://www.w3.org/TR/2018/SPSD-html401-20180327/.

[27] 3GPP TS 23.527: "5G System; Restoration Procedures; Stage 2".

[28] 3GPP TS 29.513: "5G System; Policy and Charging Control signalling flows and QoS parameter mapping; Stage 3".

[29] 3GPP TS 38.413: "NG-RAN; NG Application Protocol (NGAP)".

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

[31] 3GPP TR 21.900: "Technical Specification Group working methods".

[32] 3GPP TS 29.520: "5G System; Network Data Analytics Services; Stage 3".

[33] 3GPP TS 29.572: "5G System; Location Management Services; Stage 3".

[34] 3GPP TS 23.288: "Architecture enhancements for 5G System (5GS) to support network data analytics services".

[35] 3GPP TS 29.517: "Application Function Event Exposure Service".

[36] 3GPP TS 29.503: "Unified Data Management Services".

[37] 3GPP TS 29.336: "Home Subscriber Server (HSS) diameter interfaces for interworking with packet data networks and applications".

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

[39] Semantic Versioning Specification: https://semver.org

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

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

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].

5GC	5G Core Network
CHF	Charging Function
IPUPS	Inter-PLMN User Plane Security
NF	Network Function
NRF	NF Repository Function
NWDAF	Network Data Analytics Function
PFD	Packet Flow Description
SNPN	Stand-alone Non-Public Network
TNGF	Trusted Non-3GPP Gateway Function
TWIF	Trusted WLAN Interworking Function
W-AGF	Wireline Access Gateway Function

4 Overview

The Network Function (NF) Repository Function (NRF) is the network entity in the 5G Core Network (5GC) supporting the following functionality:

- Maintains the NF profile of available NF instances and their supported services;
- Maintains the SCP profile of available SCP instances;
- Allows other NF or SCP instances to subscribe to, and get notified about, the registration in NRF of new NF instances of a given type. It also allows SCP instances to subscribe to, and get notified about, the registration in NRF of new SCP instances;
- Supports service discovery function. It receives NF Discovery Requests from NF or SCP instances, and provides the information of the available NF instances fulfilling certain criteria (e.g., supporting a given service);
- Support SCP discovery function. It receives NF Discovery Requests for SCP profiles from other SCP instances, and provides the information of the available SCP instances fulfilling certain criteria (e.g., serving a given NF set).

Editor's Note: It is FFS whether the SCPp serving an NFp may be discovered by an NFc not configured with any SCPc, based on operator policies, e.g. in deployments where not all NFc are configured with an SCPc and where it would be desired to reach an NFp through an SCPp.

Figures 4-1 shows the reference architecture for the 5GC, with focus on the NRF:

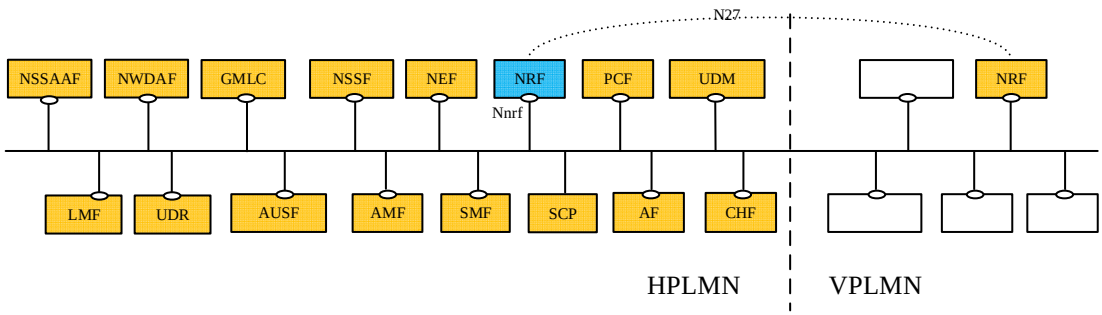

Figure 4-1: 5G System architecture

For the sake of clarity, the NRF is never depicted in reference point representation figures, given that the NRF interacts with every other NF in the 5GC. As an exception, in the roaming case, the reference point between the vNRF and the

hNRF is named as N27. The reference point name of N27 is used only for representation purposes, but its functionality is included in the services offered by the Nnrf Service-Based Interface.

5 Services Offered by the NRF

5.1 Introduction

The NRF offers to other NFs the following services:

- Nnrf_NFManagement
- Nnrf_NFDiscovery
- Nnrf_AccessToken (OAuth2 Authorization)
- Nnrf_Bootstrapping

Table 5.1-1 summarizes the corresponding APIs defined for this specification.

Table 5.1-1: API Descriptions

Service Name	Clause	Description	OpenAPI Specification File	apiName	Annex
Nnrf_NFManagement	6.1	NRF NFManagement Service	TS29510_Nnrf_NFManagement.yaml	nnrf-nfm	A.2
Nnrf_NFDiscovery	6.2	NRF NFDiscovery Service	TS29510_Nnrf_NFDiscovery.yaml	nnrf-disc	A.3
Nnrf_AccessToken	6.3	NRF OAuth2 Authorization	TS29510_Nnrf_AccessToken.yaml		A.4
Nnrf_Bootstrapping	6.4	NRF Bootstrapping	TS29510_Nnrf_Bootstrapping.yaml		A.5

5.2 Nnrf_NFManagement Service

5.2.1 Service Description

The Nnrf_NFManagement service allows an NF or an SCP Instance in the serving PLMN to register, update or deregister its profile in the NRF.

The Nnrf_NFManagement service also allows an NRF Instance to register, update or deregister its profile in another NRF in the same PLMN.

NOTE: Alternatively, other means such as OA&M can also be used to register, update or deregister NRF profile in another NRF.

It also allows an NF or an SCP to subscribe to be notified of registration, deregistration and profile changes of NF Instances, along with their potential NF services. It also enables an SCP to subscribe to be notified of registration, deregistration and profile changes of other SCP instances.

The NF profile consists of general parameters of the NF Instance, and also the parameters of the different NF Service Instances exposed by the NF Instance, if applicable.

The PLMN of the NRF may comprise one or multiple PLMN IDs (i.e. MCC and MNC). An NRF configured with multiple PLMN IDs shall support registering, updating and deregistering the profile of Network Function Instances from any of these PLMN IDs.

The Nnrf_NFManagement service also allows retrieving a list of NF and SCP Instances currently registered in the NRF or the NF Profile of a given NF Instance.

The Nnrf_NFManagement service also allows checking whether the registered NFs and SCPs are operative.

5.2.2 Service Operations

5.2.2.1 Introduction

The services operations defined for the Nnrf_NFManagement service are as follows:

- NFRegister: It allows an NF or SCP Instance to register its profile in the NRF; it includes the registration of the general parameters of the NF or SCP Instance, together with the list of potential services exposed by the NF Instance. This service operation is not allowed to be invoked from an NRF in a different PLMN.

- NFUpdate: It allows an NF or SCP Instance to replace, or update partially, the parameters of its profile (including the parameters of the associated services, if any) in the NRF; it also allows to add or delete individual services offered by the NF Instance. This service operation is not allowed to be invoked from an NRF in a different PLMN.

- NFDeregister: It allows an NF or SCP Instance to deregister its profile in the NRF, including the services offered by the NF Instance, if any. This service operation is not allowed to be invoked from an NRF in a different PLMN.

- NFStatusSubscribe: It allows an NF or SCP Instance to subscribe to changes on the status of NF Instances registered in NRF. It also allows an SCP Instance to subscribe to changes on the status of other SCP Instances registered in NRF. This service operation can be invoked by an NF Instance in a different PLMN (via the local NRF in that PLMN). It cannot be invoked by an SCP instance in a different PLMN.

- NFStatusNotify: It allows the NRF to notify subscribed NF or SCP Instances of changes on the status of NF Instances. It also allows the NRF to notify subscribed SCP Instances of changes on the status of SCP Instances. This service operation can be invoked directly between the NRF and an NF Instance in a different PLMN (without involvement of the local NRF in that PLMN). It cannot be invoked between the NRF and an SCP instance in a different PLMN.

- NFStatusUnsubscribe: It allows an NF or SCP Instance to unsubscribe to changes on the status of NF Instances registered in NRF. It also allows an SCP Instance to unsubscribe to changes on the status of other SCP Instances registered in NRF. This service operation can be invoked by an NF Instance in a different PLMN (via the local NRF in that PLMN). It cannot be invoked by an SCP instance in a different PLMN.

NOTE 1: The "change of status" of the NFStatus service operations can imply a request to be notified of newly registered NF or SCP Instances in NRF, or to be notified of profile changes of a specific NF or SCP Instance, or to be notified of the deregistration of an NF or SCP Instance.

NOTE 2: An NRF instance can also use the NFRegister, NFUpdate or NFDeregister service operations or OA&M system to register, update or deregister its profile in another NRF in the same PLMN.

- NFListRetrieval: It allows retrieving a list of NFs and SCPs currently registered in the NRF. This service operation is not allowed to be invoked from an NRF in a different PLMN.

- NFProfileRetrieval: It allows retrieving the profile of a given NF or SCP instance. This service operation is not allowed to be invoked from an NRF in a different PLMN.

The NFStatusSubscribe / NFstatusNotify / NFStatusUnsubscribe operations can be invoked by an NF Service Consumer (i.e., "source NF") or SCP requesting to be notified about events (registration, deregistration, profile change) related to an NF instance (i.e., "target NF") located in the same PLMN, or in a different PLMN. An SCP can also invoke these operations to be notified about events (registration, deregistration, profile change) related to an SCP instance located in the same PLMN.

In the description of these operations in clauses 5.2.2.5, 5.2.2.6 and 5.2.2.7, when the NF instances are located in the same PLMN, both source NF and target NF are said to be located in the "Serving PLMN" but, in the general case, the functionality is not restricted to the PLMN that is serving a given UE, and it shall be applicable as well to any scenario in which source NF and target NFs belong to the same PLMN.

When source NF and target NF are located in different PLMNs, the source NF is said to be in the "Serving PLMN", and the target NF (and the NRF where such NF is registered) is said to be in the "Home PLMN", similarly to the scenarios described in 3GPP TS 23.502 [3], but the functionality shall be equally applicable to any scenario between any pair of PLMNs (e.g. with the source NF in the Home PLMN and the target NF in the Serving PLMN).

The SCP is treated by the Nnrf_NFManagement service in the same way as NFs. Specifically, the SCP is designated with a specific NF type and NF Instance ID. However, the SCP does not support services. Accordingly, references to "NF" or "NF Profile" in the description of the service operations in the following clauses also apply to an SCP.

5.2.2.2 NFRegister

5.2.2.2.1 General

This service operation is used:

- to register an NF in the NRF by providing the NF profile of the requesting NF to the NRF, and the NRF marks the requesting NF as available to be discovered by other NFs;
- to register services associated to an existing NF Instance;
- to register NRF information in another NRF, and this information is used for forwarding or redirecting service discovery request.

5.2.2.2.2 NF (other than NRF) registration to NRF

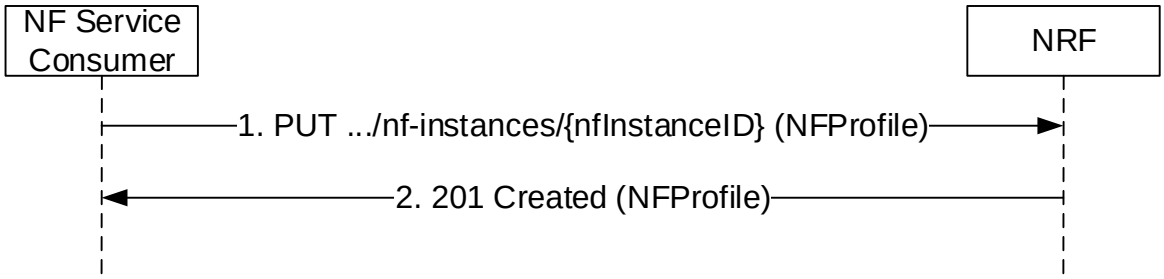

Figure 5.2.2.2.2-1: NF Instance Registration

1. The NF Service Consumer shall send a PUT request to the resource URI representing the NF Instance. The URI is determined by the NF Instance. The variable {nfInstanceID} represents an identifier, provided by the NF Service Consumer, that shall be globally unique inside the PLMN of the NRF where the NF is being registered. The format of the NF Instance ID shall be a Universally Unique Identifier (UUID) version 4, as described in IETF RFC 4122 [18].

   EXAMPLE: UUID version 4: "4947a69a-f61b-4bc1-b9da-47c9c5d14b64"

   The payload body of the PUT request shall contain a representation of the NF Instance to be created.

2. On success, "201 Created" shall be returned, the payload body of the PUT response shall contain the representation of the created resource and the "Location" header shall contain the URI of the created resource. Additionally, the NRF returns a "heart-beat timer" containing the number of seconds expected between two consecutive heart-beat messages from an NF Instance to the NRF (see clause 5.2.2.3.2). The representation of the created resource may be a complete NF Profile or a NF Profile just including the mandatory attributes of the NF Profile and the attributes which the NRF added or changed (see Annex B).

   If the registration of the NF instance fails at the NRF due to errors in the encoding of the NFProfile JSON object, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the registration of the NF instance fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

The NRF shall allow the registration of a Network Function instance with any of the NF types described in clause 6.1.6.3.3, and it shall also allow registration of Network Function instances with custom NF types (e.g., NF type values not defined by 3GPP, or NF type values not defined by this API version).

NOTE 1: When registering a custom NF in NRF, it is recommended to use a NF type name that prevents collisions with other custom NF type names, or with NF types defined in the future by 3GPP. E.g., prefixing the custom NF type name with the string "CUSTOM_".

During the registration of a Network Function instance with a custom NF type, the NF instance may provide NF-specific data (in the "customInfo" attribute), that shall be stored by the NRF as part of the NF profile of the NF instance.

The NRF shall accept the registration of NF Instances containing Vendor-Specific attributes (see 3GPP TS 29.500 [4], clause 6.6.3), and therefore, it shall accept NF Profiles containing attributes whose type may be unknown to the NRF, and those attributes shall be stored as part of the NF's profile data in NRF.

Before an NF Instance registers its NF Profile in NRF, the NF Instance should check the capabilities of the NRF by issuing an OPTIONS request to the "nf-instances" resource (see clause 6.1.3.2.3.2). The NRF may indicate in the response capabilities such as the support of receiving compressed payloads in the HTTP PUT request used for registration of the NF Profile, or support of specific attributes of the NF Profile.

NOTE 2: A Rel-16 NF needs to register the list of NF Service Instances in the "nfServices" array attribute towards an NRF not supporting the Service-Map feature (i.e. a Rel-15 NRF).

5.2.2.2.3 NRF registration to another NRF

The procedure specified in clause 5.2.2.2.2 applies. Additionally:

a) the registering NRF shall set the nfType to "NRF" in the nfProfile;

b) the registering NRF shall set the nfService to contain "nnrf-disc", "nnrf-nfm" and optionally "nnrf-oauth2" in the nfProfile;

c) the registering NRF may include nrfInfo which contains the information of udrInfo, udmInfo, ausfInfo, amfInfo, smfInfo, upfInfo, pcfInfo, bsfInfo, nefInfo, chfInfo, pcscfInfo, lmfInfo, gmlcInfo and nfInfo in the nfProfile locally configured in the NRF or the NRF received during registration of other NFs, this means the registering NRF is able to provide service for discovery of NFs subject to that information;

d) if the NRF receives an NF registration with the nfType set to "NRF", the NRF shall use the information contained in the nfProfile to target the registering NRF when forwarding or redirecting NF service discovery request.

5.2.2.3 NFUpdate

5.2.2.3.1 General

This service operation updates the profile of a Network Function previously registered in the NRF by providing the updated NF profile of the requesting NF to the NRF. The update operation may apply to the whole profile of the NF (complete replacement of the existing profile by a new profile), or it may apply only to a subset of the parameters of the profile (including adding/deleting/replacing services to the NF profile).

To perform a complete replacement of the NF Profile of a given NF Instance, the NF Service Consumer shall issue an HTTP PUT request, as shown in Figure 5.2.2.3.1-1:

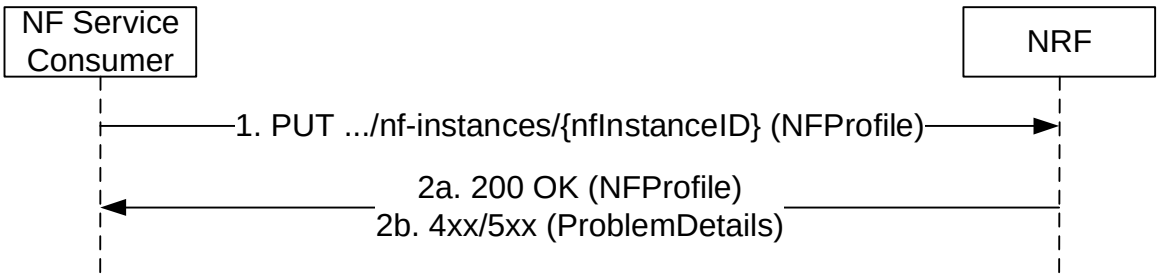

Figure 5.2.2.3.1-1: NF Profile Complete Replacement

1. The NF Service Consumer shall send a PUT request to the resource URI representing the NF Instance. The payload body of the PUT request shall contain a representation of the NF Instance to be completely replaced in the NRF.

2a. On success, "200 OK" shall be returned, the payload body of the PUT response shall contain the representation of the replaced resource. The representation of the replaced resource may be a complete NF Profile or a NF Profile just including the mandatory attributes of the NF Profile and the attributes which the NRF added or changed (see Annex B).

2b. If the update of the NF instance fails at the NRF due to errors in the encoding of the NFProfile JSON object, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the update of the NF instance fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

To perform a partial update of the NF Profile of a given NF Instance, the NF Service Consumer shall issue an HTTP PATCH request, as shown in Figure 5.2.2.3.1-2. This partial update shall be used to add/delete/replace individual parameters of the NF Instance, and also to add/delete/replace any of the services (and their parameters) offered by the NF Instance.

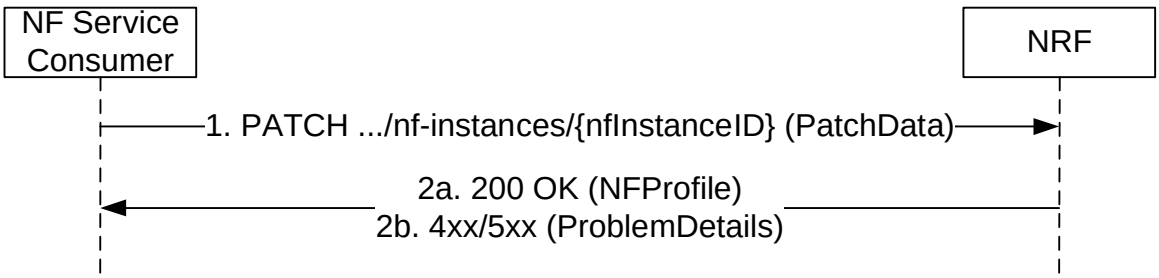

Figure 5.2.2.3.1-2: NF Profile Partial Update

1. The NF Service Consumer shall send a PATCH request to the resource URI representing the NF Instance. The payload body of the PATCH request shall contain the list of operations (add/delete/replace) to be applied to the NF Profile of the NF Instance; these operations may be directed to individual parameters of the NF Profile or to the list of services (and their parameters) offered by the NF Instances. In order to leave the NF Profile in a consistent state, all the operations specified by the PATCH request body shall be executed atomically.

2a. On success, "200 OK" shall be returned, the payload body of the PATCH response shall contain the representation of the replaced resource.

2b. If the NF Instance, identified by the "nfInstanceID", is not found in the list of registered NF Instances in the NRF's database, the NRF shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

The NRF shall allow updating Vendor-Specific attributes (see 3GPP TS 29.500 [4], clause 6.6.3) that may exist in the NF Profile of a registered NF Instance.

5.2.2.3.2 NF Heart-Beat

Each NF that has previously registered in NRF shall contact the NRF periodically (heart-beat), by invoking the NFUpdate service operation, in order to show that the NF is still operative.

The time interval at which the NRF shall be contacted is deployment-specific, and it is returned by the NRF to the NF Service Consumer as a result of a successful registration.

When the NRF detects that a given NF has not updated its profile for a configurable amount of time (longer than the heart-beat interval), the NRF changes the status of the NF to SUSPENDED and considers that the NF and its services can no longer be discovered by other NFs via the NFDiscovery service. The NRF notifies NFs subscribed to receiving notifications of changes of the NF Profile that the NF status has been changed to SUSPENDED.

If the NRF modifies the heart-beat interval value of a given NF instance currently registered (e.g. as a result of an OA&M operation), it shall return the new value to the registered NF in the response of the next periodic heart-beat interaction received from that NF and, until then, the NRF shall apply the heart-beat check procedure according to the original interval value.

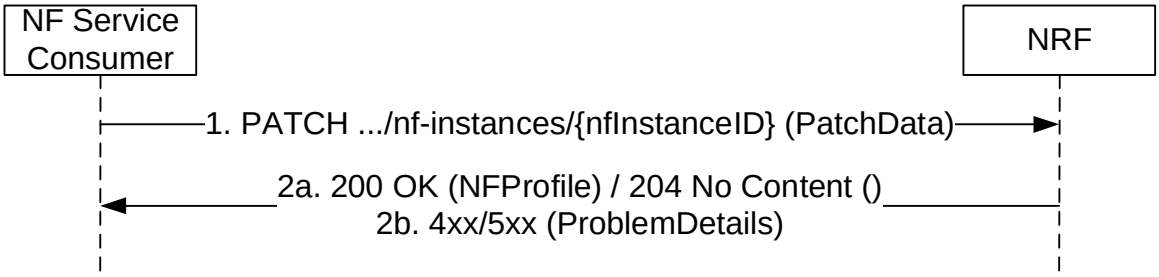

Figure 5.2.2.3.2-1: NF Heart-Beat

1. The NF Service Consumer shall send a PATCH request to the resource URI representing the NF Instance. The payload body of the PATCH request shall contain a "replace" operation on the "nfStatus" attribute of the NF Profile of the NF Instance, and set it to the value "REGISTERED" or "UNDISCOVERABLE".

   In addition, the NF Service Consumer may also provide the load information of the NF, and/or the load information of the NF associated NF services. The provision of such load information may be limited by this NF via appropriate configuration (e.g. granularity threshold) in order to avoid notifying minor load changes.

2a. On success, the NRF should return "204 No Content"; the NRF may also answer with "200 OK" along with the full NF Profile, e.g. in cases where the NRF determines that the NF Profile has changed significantly since the last heart-beat, and wants to send the new profile to the NF Service Consumer (note that this alternative has bigger signalling overhead).

2b. If the NF Instance, identified by the "nfInstanceID", is not found in the list of registered NF Instances in the NRF's database, the NRF shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

EXAMPLE:

```
PATCH .../nf-instances/4947a69a-f61b-4bc1-b9da-47c9c5d14b64
Content-Type: application/json-patch+json

[
  { "op": "replace", "path": "/nfStatus", "value": "REGISTERED" },
  { "op": "replace", "path": "/load", "value": 50 }
]


HTTP/2 204 No Content
Content-Location: .../nf-instances/4947a69a-f61b-4bc1-b9da-47c9c5d14b64
```

5.2.2.4 NFDeregister

5.2.2.4.1 General

This service operation removes the profile of a Network Function previously registered in the NRF.

It is executed by deleting a given resource identified by a "NF Instance ID". The operation is invoked by issuing a DELETE request on the URI representing the specific NF Instance.

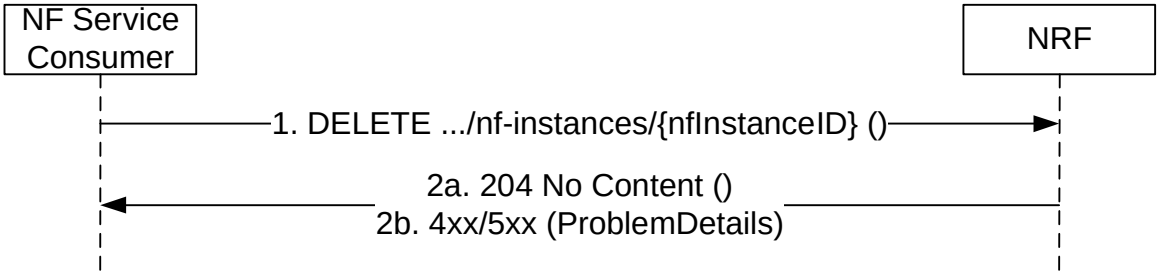

Figure 5.2.2.4.1-1: NF Instance Deregistration

1. The NF Service Consumer shall send a DELETE request to the resource URI representing the NF Instance. The request body shall be empty.

2a. On success, "204 No Content" shall be returned. The response body shall be empty.

2b. If the NF Instance, identified by the "nfInstanceID", is not found in the list of registered NF Instances in the NRF's database, the NRF shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

5.2.2.5 NFStatusSubscribe

5.2.2.5.1 General

This service operation is used to:

- create a subscription so an NF Service Consumer can request to be notified when NF Instances of a given set, following certain filter criteria are registered/deregistered in NRF or when their profile is modified;
- create a subscription to a specific NF Instance so an NF Service Consumer can request to be notified when the profile of such NF Instance is modified or when the NF Instance is deregistered from NRF.

5.2.2.5.2 Subscription to NF Instances in the same PLMN

The subscription to notifications on NF Instances is executed creating a new individual resource under the collection resource "subscriptions". The operation is invoked by issuing a POST request on the URI representing the "subscriptions" resource.

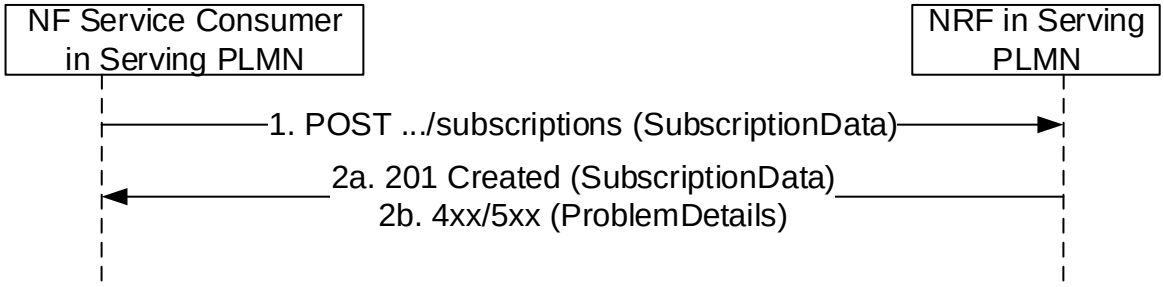

Figure 5.2.2.5.2-1: Subscription to NF Instances in the same PLMN

1. The NF Service Consumer shall send a POST request to the resource URI representing the "subscriptions" collection resource.

   The request body shall include the data indicating the type of notifications that the NF Service Consumer is interested in receiving; it also contains a callback URI, where the NF Service Consumer shall be prepared to receive the actual notification from the NRF (see NFStatusNotify operation in 5.2.2.6) and it may contain a validity time, suggested by the NF Service Consumer, representing the time span during which the subscription is desired to be kept active.

   The subscription request may also include additional parameters indicating the list of attributes (including Vendor-Specific attributes, see 3GPP TS 29.500 [4], clause 6.6.3) in the NF Profile to be monitored (or to be excluded from monitoring), in order to determine whether a notification from NRF should be sent, or not, when any of those attributes is changed in the profile.

   The NF Service Consumer may request the creation of a subscription to a specific NF Instance, or to a set of NF Instances, where the set is determined according to different criteria specified in the request body, in the "subscrCond" attribute of the "SubscriptionData" object type (see clause 6.1.6.2.16).

   The subscription shall be authorized, or rejected, by the NRF by checking the "reqNfType" and "reqNfFqdn" input attributes in the subscription request body (along with the contents of any optional Oauth2 access token provided in the API request) against the list of authorization attributes in the NF Profile of the target NF Instance to be monitored.

   When the subscription request is for a set of NFs, the authorization attributes of the NF Instances in the set may differ, resulting in positive authorization of the subscription for only a part of the NF Instances in the set; in that case, the subscription to the set of NFs may be accepted by the NRF, but the NF Instances in the set that are not authorized for the NF Service Consumer that requested the subscription, shall not result in triggering any notification event from the NRF to the NF Service Consumer.

2a. On success, "201 Created" shall be returned. The response shall contain the data related to the created subscription, including the validity time, as determined by the NRF, after which the subscription becomes invalid. Once the subscription expires, if the NF Service Consumer wants to keep receiving status notifications, it shall create a new subscription in the NRF.

2b. If the creation of the subscription fails at the NRF due to errors in the SubscriptionData JSON object in the request body, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

   If the creation of the subscription fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

5.2.2.5.3 Subscription to NF Instances in a different PLMN

The subscription to notifications on NF Instances in a different PLMN is done by creating a resource under the collection resource "subscriptions", in the NRF of the Home PLMN.

For that, step 1 in clause 5.2.2.5.2 is executed (send a POST request to the NRF in the Serving PLMN); this request shall include the identity of the PLMN of the home NRF in the SubscriptionData parameter in the request body.

Then, steps 1-2 in Figure 5.2.2.5.3-1 are executed, between the NRF in the Serving PLMN and the NRF in the Home PLMN. In this step, the presence of the PLMN ID in the SubscriptionData parameter is not required. The NRF in the Home PLMN returns a subscriptionID identifying the created subscription.

Finally, step 2 in clause 5.2.2.5.2 is executed; a new subscriptionID shall be generated by the NRF in the Serving PLMN as indicated in step 2 of Figure 5.2.2.5.3-1, and shall be sent to the NF Service Consumer in the Serving PLMN.

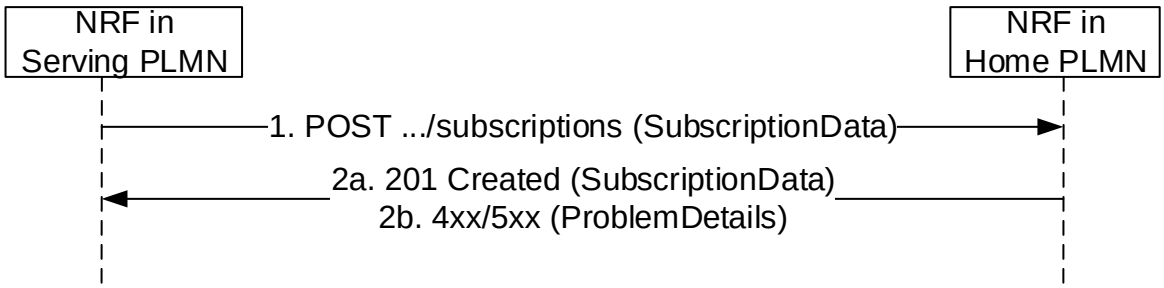

Figure 5.2.2.5.3-1: Subscription to NF Instances in a different PLMN

1. The NRF in Serving PLMN shall send a POST request to the resource URI in the NRF in Home PLMN representing the "subscriptions" collection resource. The request body shall include the SubscriptionData as received by the NRF in Serving PLMN from the NF Service Consumer in the Serving PLMN (see 5.2.2.5.2), containing the data about the type of notifications that the NF Service Consumer is interested in receiving and the callback URI where the NF Service Consumer shall be prepared to receive the notifications from the NRF (see NFStatusNotify operation in 5.2.2.6).

2a. On success, "201 Created" shall be returned. The NRF in Serving PLMN should not keep state for this created subscription and shall send to the NF Service Consumer in Serving PLMN (step 2 in 5.2.2.5.2) a subscriptionID that shall consist on the following structure: <MCC>+<MNC>+"-"+<OriginalSubscriptionID>

EXAMPLE: If the NRF in a Home PLMN (where MCC = 123, and MNC=456) creates a subscription with value "subs987654", the subscriptionID that the NRF in Serving PLMN would send to the NF Service Consumer in Serving PLMN is: "123456-subs987654"

The URI in the Location header that the NRF in Serving PLMN returns to the NF Service Consumer in Serving PLMN shall contain a <subscriptionId> modified as described above and, if it is as an absolute URI, an apiRoot pointing to the address of the NRF in Serving PLMN.

2b. If the creation of the subscription fails at the NRF due to errors in the SubscriptionData JSON object in the request body, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the creation of the subscription fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

5.2.2.5.4 Subscription to NF Instances with intermediate forwarding NRF

When multiple NRFs are deployed in one PLMN, an NF Instance can subscribe to changes of NF Instances registered in an NRF to which it is not directly interacting. The subscription message is forwarded by an intermediate NRF to which the subscribing NF instance is directly interacting.

For that, step 1 in clause 5.2.2.5.2 is executed (send a POST request to the NRF-1 in the Serving PLMN); this request shall include the SubscriptionData parameter in the request body.

Then, steps 1-4 in Figure 5.2.2.5.4-1 are executed between NF Service Consumer in Serving PLMN, NRF-1 in Serving PLMN and NRF-2 in Serving PLMN. In thest steps, NRF-1 sends the subscription request to a pre-configured NRF-2. NRF-2 requests corresponding NRF (e.g. the NF Service Producer registered NRF) and returns a subscriptionID identifying the created subscription and this subscriptionID is sent to the NF Service Consumer via NRF-1.

Finally, step 2 in clause 5.2.2.5.2 is executed; the subscriptionID shall be sent to the NF Service Consumer.

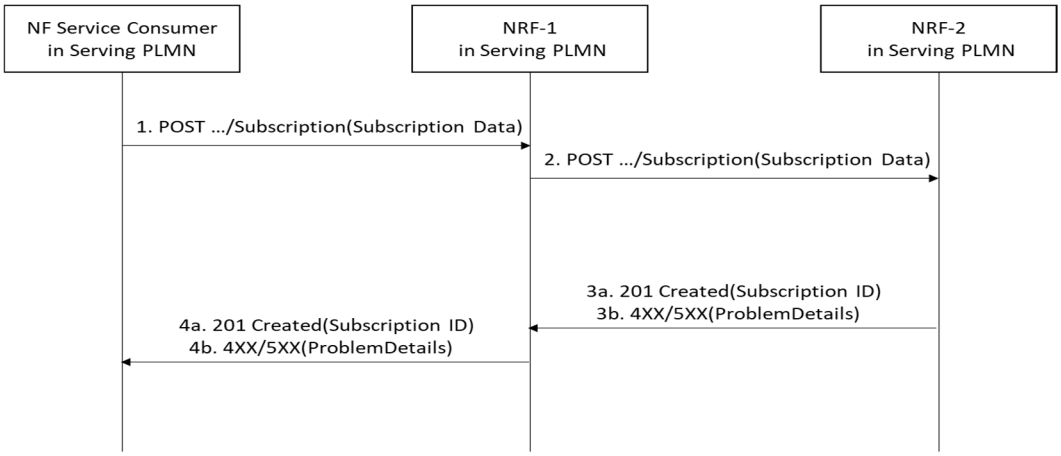

Figure 5.2.2.5.4-1: Subscription with intermediate forwarding NRF

1. NRF-1 receives a subscription request but does not have the information to fulfil the request. Then NRF-1 sends the subscription request to a pre-configured NRF-2.

2. Upon receiving a subscription request, based on the SubscriptionData contained in the subscription request (e.g.NF type) and locally stored information (see clause 5.2.2.2.3), NRF-2 shall identify the next hop NRF and forward the subscription request to that NRF (i.e. NF Service Producer registered NRF).

3a. On success, "201 Created" shall be returned by NRF-2.

3b. If the creation of the subscription fails, the NRF-2 shall return "4XX/5XX" response.

4a. NRF-1 forwards the success response to NF Service Consumer. The payload body of the POST response shall contain the representation describing the status of the request and the "Location" header shall be present and shall contain the URI of the created resource. The authority and/or deployment-specific string of the apiRoot of the created resource URI may differ from the authority and/or deployment-specific string of the apiRoot of the request URI received in the POST request.

4b. NRF-1 forwards the error response to NF Service Consumer.

5.2.2.5.5 Subscription to NF Instances with intermediate redirecting NRF

When multiple NRFs are deployed in one PLMN, an NF Instance can subscribe to changes of NF Instances registered in another NRF. The subscription message is redirected by a third NRF.

For that, step 1 in clause 5.2.2.5.2 is executed (send a POST request to the NRF-1 in the Serving PLMN); this request shall include the SubscriptionData parameter in the request body.

Then, steps 2-5 in Figure 5.2.2.5.5-1 are executed between NRF-1, NRF-2 and NRF-3.

Finally, step 2 in clause 5.2.2.5.2 is executed; the subscriptionID shall be sent to the NF Service Consumer.

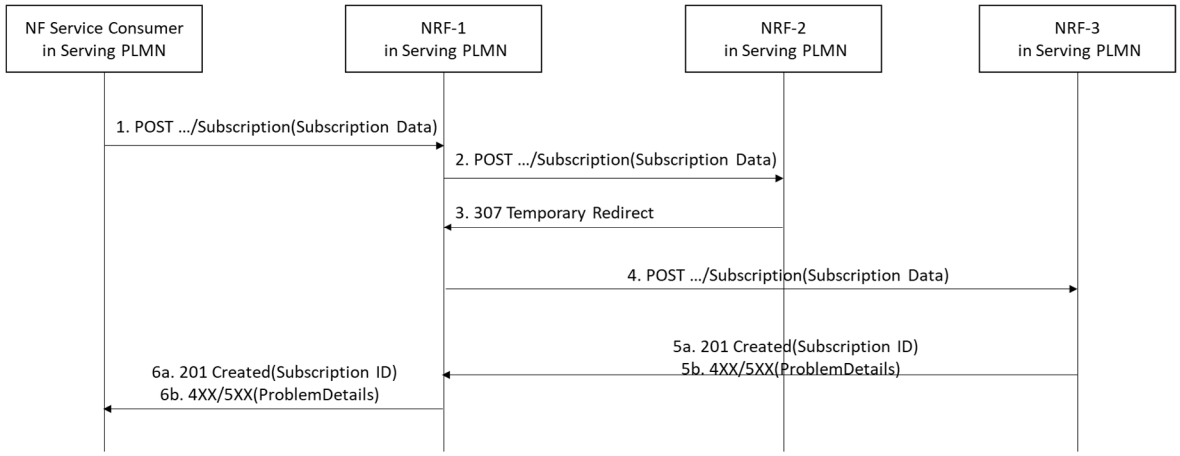

Figure 5.2.2.5.5-1: Subscription to NF Instances with intermediate redirecting NRF

1. NF Service Consumer send a subscription request to NRF-1.
2. NRF-1 receives a subscription request but does not have the information to fulfil the request. Then NRF-1 sends the subscription request to a pre-configured NRF-2.
3. Upon receiving a subscription request, based on the SubscriptionData contained in the subscription request (e.g.NF type) and locally stored information (see clause 5.2.2.2.3), NRF-2 shall identify the next hop NRF, and redirect the subscription request by returning HTTP 307 Temporary Redirect response.

   The 307 Temporary Redirect response shall contain a Location header field, the host part of the URI in the Location header field represents NRF-3.
4. Upon receiving 307 Temporary Redirect response, NRF-1 sends the subscription request to NRF-3 by using the URI contained in the Location header field of the 307 Temporary Redirect response.

5a. On success, "201 Created" shall be returned by NRF-3.

5b. If the creation of the subscription fails at the NRF-3, the NRF-3 shall return "4XX/5XX" response.

6a. On success, "201 Created" shall be forwarded to NF Service Consumer via NRF-1. The payload body of the POST response shall contain the representation describing the status of the request and the "Location" header shall be present and shall contain the URI of the created resource. The authority and/or deployment-specific string of the apiRoot of the created resource URI may differ from the authority and/or deployment-specific string of the apiRoot of the request URI received in the POST request.

6b. If the creation of the subscription fails, "4XX/5XX" shall be forwarded to NF Service Consumer via NRF-1.

5.2.2.5.6 Update of Subscription to NF Instances

The subscription to notifications on NF Instances may be updated to refresh the validity time, when this time is about to expire. The NF Service Consumer may request a new validity time to the NRF, and the NRF shall answer with the new assigned validity time, if the operation is successful.

This operation is executed by updating the resource identified by "subscriptionID". It is invoked by issuing an HTTP PATCH request on the URI representing the individual resource received in the Location header field of the "201 Created" response received during a successful subscription (see clause 5.2.2.5).

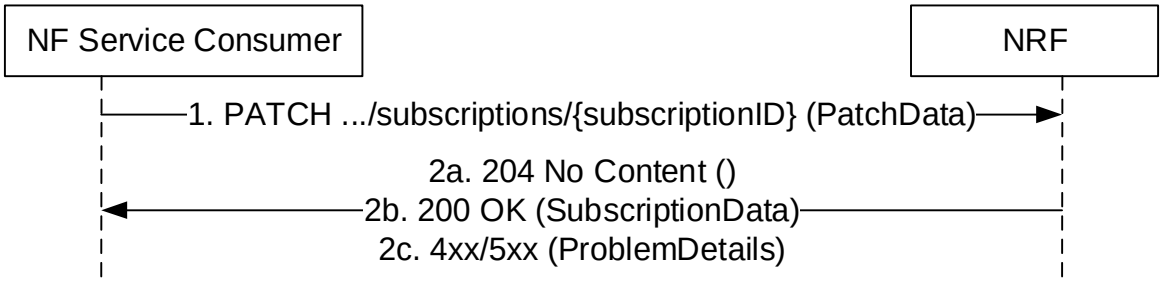

Figure 5.2.2.5.6-1: Subscription to NF Instances in the same PLMN

1. The NF Service Consumer shall send a PATCH request to the resource URI identifying the individual subscription resource. The payload body of the PATCH request shall contain a "replace" operation on the "validityTime" attribute of the SubscriptionData structure and shall contain a new suggested value for it; no other attribute of the resource shall be updated as part of this operation.

2a. On success, if the NRF accepts the extension of the lifetime of the subscription, and it accepts the requested value for the "validityTime" attribute, a response with status code "204 No Content" shall be returned.

2b. On success, if the NRF accepts the extension of the lifetime of the subscription, but it assigns a validity time different than the value suggested by the NF Service Consumer, a "200 OK" response code shall be returned. The response shall contain the new resource representation of the "subscription" resource, which includes the new validity time, as determined by the NRF, after which the subscription becomes invalid.

2c. If the update of the subscription fails at the NRF due to errors in the JSON Patch object in the request body, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the update of the subscription fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

EXAMPLE:

```
PATCH .../subscriptions/2a58bf47
Content-Type: application/json-patch+json

[
  { "op": "replace", "path": "/validityTime", "value": "2018-12-30T23:20:50Z" },
]


HTTP/2 204 No Content
```

5.2.2.5.7 Update of Subscription to NF Instances in a different PLMN

The update of subscription in a different PLMN is done by updating a subscription resource identified by a "subscriptionID".

For that, step 1 in clause 5.2.2.5.6 is executed (send a PATCH request to the NRF in the Serving PLMN); this request shall include the identity of the PLMN of the home NRF (MCC/MNC values) as a leading prefix of the susbcriptionID.

Then, steps 1-2 in Figure 5.2.2.5.7-1 are executed, between the NRF in the Serving PLMN and the NRF in the Home PLMN. In this step, the subscriptionID sent to the NRF in the Home PLMN shall not contain the identity of the PLMN (i.e., it shall be the same subscriptionID value as originally generated by the NRF in the Home PLMN). The NRF in the Home PLMN returns a status code with the result of the operation.

Finally, step 2 in clause 5.2.2.5.7-2 is executed; a status code is returned to the NF Service Consumer in Serving PLMN in accordance to the result received from NRF in the Home PLMN.

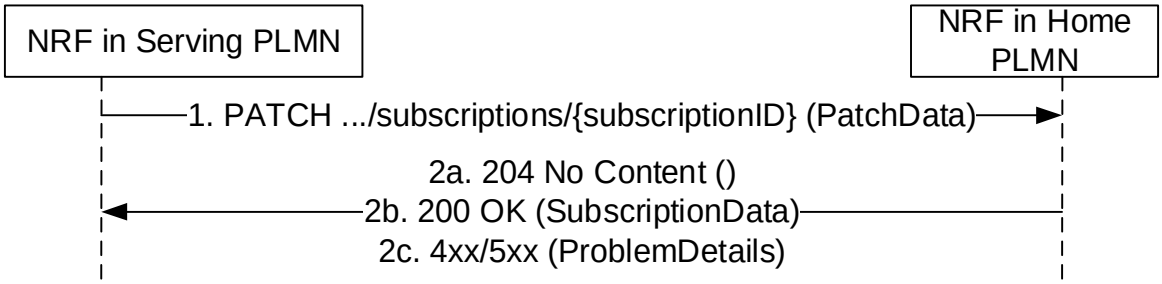

Figure 5.2.2.5.7-1: Update of Subscription to NF Instances in a different PLMN

1. The NRF in Serving PLMN shall send a PATCH request to the resource URI representing the individual subscription. The payload body of the PATCH request shall contain a "replace" operation on the "validityTime" attribute of the SubscriptionData structure and shall contain a new suggested value for it;

2a. On success, if the NRF in the Home PLMN accepts the extension of the lifetime of the subscription, and it accepts the requested value for the "validityTime" attribute, a response with status code "204 No Content" shall be returned.

2b. On success, if the NRF in the Home PLMN accepts the extension of the lifetime of the subscription, but it assigns a validity time different than the value suggested by the NF Service Consumer, a "200 OK" response code shall be returned. The response shall contain the new resource representation of the "subscription" resource, which includes the new validity time, as determined by the NRF in the Home PLMN, after which the subscription becomes invalid.

2c. If the update of the subscription fails at the NRF due to errors in the JSON Patch object in the request body, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the update of the subscription fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

5.2.2.6 NFStatusNotify

5.2.2.6.1 General

This service operation notifies each NF Service Consumer that was previously subscribed to receiving notifications of registration/deregistration of NF Instances, or notifications of changes of the NF profile of a given NF Instance. The notification is sent to a callback URI that each NF Service Consumer provided during the subscription (see NFStatusSubscribe operation in 5.2.2.5).

5.2.2.6.2 Notification from NRF in the same PLMN

The operation is invoked by issuing a POST request to each callback URI of the different subscribed NF Instances.

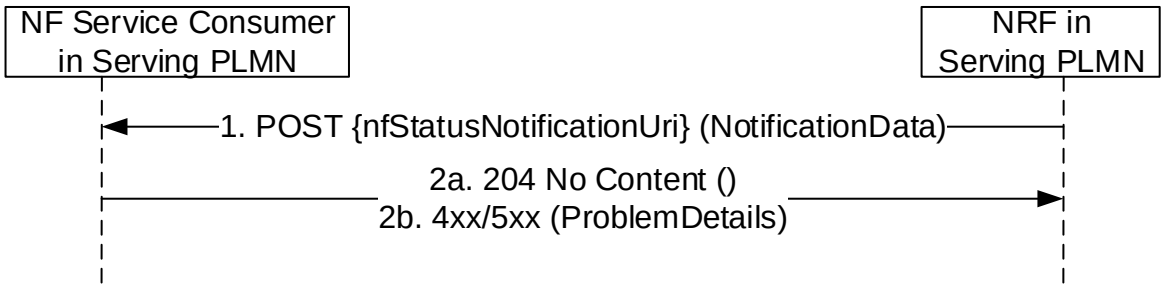

Figure 5.2.2.6.2-1: Notification from NRF in the same PLMN

1. The NRF shall send a POST request to the callback URI.

For notifications of newly registered NF Instances, the request body shall include the data associated to the newly registered NF, and its services, according to the criteria indicated by the NF Service Consumer during the subscription operation. These data shall contain the NFInstanceID of the NF Instance, an indication of the event

being notified ("registration"), and the new profile data (including, among others, the services offered by the NF Instance).

For notifications of changes of the profile of a NF Instance, the request body shall include the NFInstancceID of the NF Instance whose profile was changed, an indication of the event being notified ("profile change"), and the new profile data.

For notifications of deregistration of the NF Instance from NRF, the request body shall include the NFInstanceID of the deregistered NF Instance, and an indication of the event being notified ("deregistration").

When an NF Service Consumer subscribes to a set of NFs (using the different subscription conditions specified in clause 6.1.6.2.35), a change in the profile of the monitored NF Instance may result in such NF becoming a part of the NF set, or stops becoming a part of it (e.g., an NF Service Consumer subscribing to all NFs offering a given NF Service, and then, a certain NF Instance changes its profile by adding or removing an NF Service of its NF Profile); in such case, the NRF shall use the "NF_PROFILE_CHANGED" event type in the notification. Similarly, a change of the status (i.e. the "nfStatus" attribute of the NF Profile) shall result into the NRF to send notifications to subscribing NFs with event type set to "NF_PROFILE_CHANGED".

When an NF Service Consumer subscribes to a set of NFs, using the subscription conditions specified in clause 6.1.6.2.35, in case of a change of profile(s) of NFs potentially related to those subscription conditions, the NRF shall send notification to subscribing NF Service Consumer(s) to those NFs no longer matching the subscription conditions, and to subscribing NF Service Consumer(s) to NFs that start matching the subscription conditions. In that case, the NRF indicates in the notification data whether the notification is due to the NF Instance to newly start or stop matching the subscription condition (i.e. based on the presence of the "conditionEvent" attribute of the NotificationData).

The notification of changes of the profile may be done by the NRF either by sending the entire new NF Profile, or by indicating a number of "delta" changes (see clause 6.1.6.2.17) from an existing NF Profile that might have been previously received by the NF Service Consumer during an NFDiscovery search operation (see clause 5.3.2.2). If the NF Service Consumer receives "delta" changes related to an NF Service Instance (other than adding a new NF Service Instance) that had not been previously discovered, those changes shall be ignored by the NF Service Consumer, but any other "delta" changes related to NF Service Instances previously discovered or adding a new NF Service Instance shall be applied.

2a. On success, "204 No content" shall be returned by the NF Service Consumer.

2b. If the NF Service Consumer does not consider the "nfStatusNotificationUri" as a valid notification URI (e.g., because the URI does not belong to any of the existing subscriptions created by the NF Service Consumer in the NRF), the NF Service Consumer shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

5.2.2.6.3 Notification from NRF in a different PLMN

The operation is invoked by issuing a POST request to each callback URI of the different subscribed NF Instances.

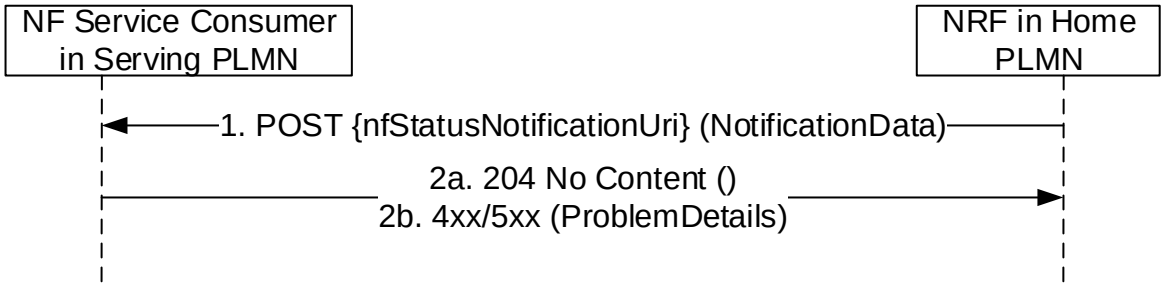

Figure 5.2.2.6.3-1: Notification from NRF in a different PLMN

Steps 1 and 2 are identical to steps 1 and 2 in Figure 5.2.2.6.2-1.

It should be noted that the POST request shall be sent directly from the NRF in Home PLMN to the NF Service Consumer in Serving PLMN, without involvement of the NRF in Serving PLMN.

5.2.2.6.4 Notification for subscription via intermediate NRF

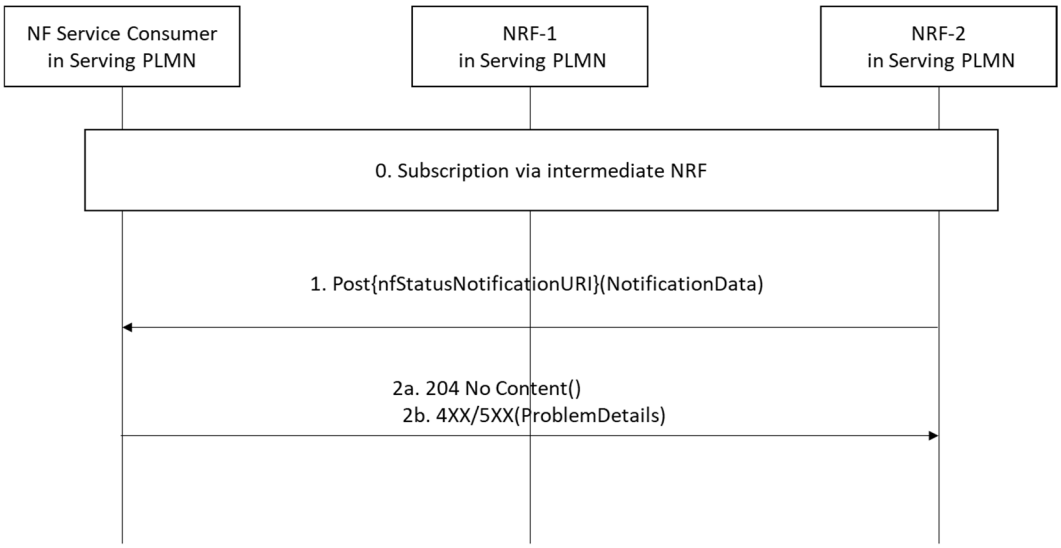

Figure 5.2.2.6.4-1: Notification for subscription via intermediate NRF

Step 0 is the NF Service Consumer creates a subscription to NRF-2 via intermediate NRF.

Steps 1 and 2 are identical to steps 1 and 2 in Figure 5.2.2.6.2-1.

The POST request shall be sent directly from NRF-2 to the NF Service Consumer without involvement of NRF-1.

5.2.2.7 NFStatusUnSubscribe

5.2.2.7.1 General

This service operation removes an existing subscription to notifications.

5.2.2.7.2 Subscription removal in the same PLMN

It is executed by deleting a given resource identified by a "subscriptionID". The operation is invoked by issuing a DELETE request on the URI representing the specific subscription received in the Location header field of the "201 Created" response received during a successful subscription (see clause 5.2.2.5).

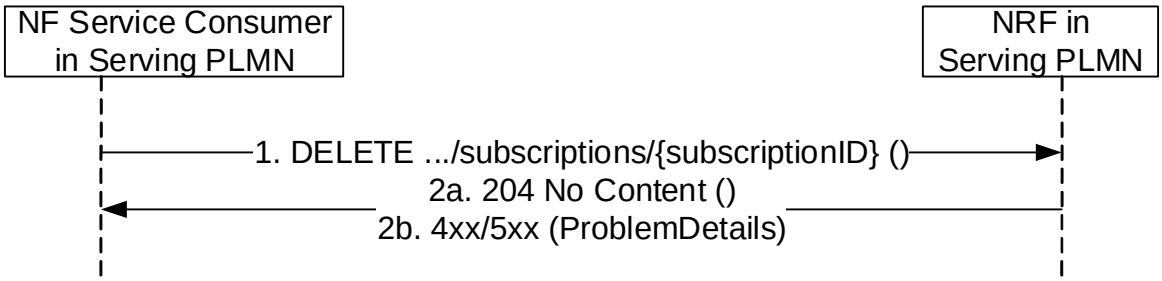

Figure 5.2.2.7.2-1: Subscription removal in the same PLMN

1. The NF Service Consumer shall send a DELETE request to the resource URI representing the individual subscription. The request body shall be empty.

2a. On success, "204 No Content" shall be returned. The response body shall be empty.

2b. If the subscription, identified by the "subscriptionID", is not found in the list of active subscriptions in the NRF's database, the NRF shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

5.2.2.7.3 Subscription removal in a different PLMN

The subscription removal in a different PLMN is done by deleting a resource identified by a "subscriptionID", in the NRF of the Home PLMN.

For that, step 1 in clause 5.2.2.7.2 is executed (send a DELETE request to the NRF in the Serving PLMN); this request shall include the identity of the PLMN of the home NRF (MCC/MNC values) as a leading prefix of the subscriptionID (see clause 5.2.2.5.3).

Then, steps 1-2 in Figure 5.2.2.7.3-1 are executed, between the NRF in the Serving PLMN and the NRF in the Home PLMN. In this step, the subscriptionID sent to the NRF in the Home PLMN shall not contain the identity of the PLMN (i.e., it shall be the same subscriptionID value as originally generated by the NRF in the Home PLMN). The NRF in the Home PLMN returns a status code with the result of the operation.

Finally, step 2 in clause 5.2.2.7.2 is executed; a status code is returned from the NRF in serving PLMN to the NF Service Consumer in Serving PLMN in accordance to the result received from NRF in Home PLMN.

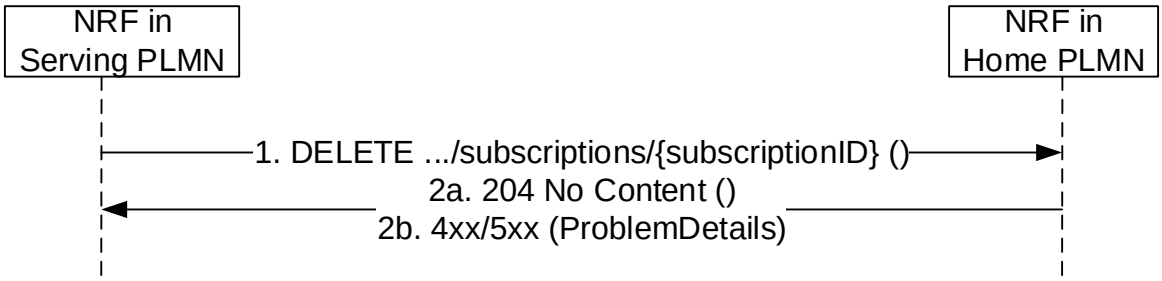

Figure 5.2.2.7.3-1: Subscription removal in a different PLMN

1. The NF Service Consumer shall send a DELETE request to the resource URI representing the individual subscription. The request body shall be empty.

2a. On success, "204 No Content" shall be returned. The response body shall be empty.

2b. If the subscription, identified by the "subscriptionID", is not found in the list of active subscriptions in the NRF's database, the NRF shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

5.2.2.8 NFListRetrieval

5.2.2.8.1 General

This service operation allows the retrieval of a list of NF Instances that are currently registered in NRF. The operation may apply to the whole set of registered NF instances or only to a subset of the NF instances, based on a given NF type and/or maximum number of NF instances to be returned.

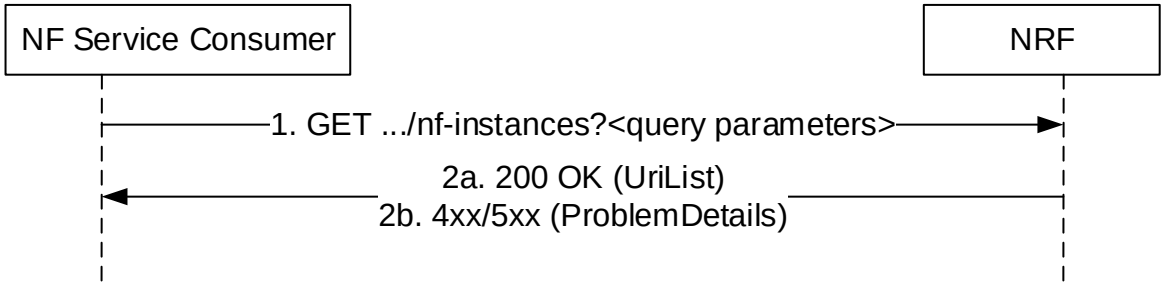

Figure 5.2.2.8.1-1: NF instance list retrieval

1. The NF Service Consumer shall send an HTTP GET request to the resource URI "nf-instances" collection resource. The optional input filter criteria for the retrieval request shall be included in query parameters.

2a. On success, "200 OK" shall be returned. The response body shall contain the URI (conforming to the resource URI structure as described in clause 5.2.2.9.1) of each registered NF in the NRF that satisfy the retrieval filter criteria (e.g., all NF instances of the same NF type), or an empty list if there are no NFs to return in the query result (e.g., because there are no registered NFs in the NRF, or because there are no matching NFs of the type specified in the "nf-type" query parameter, currently registered in the NRF).

2b. If the NF Service Consumer is not allowed to retrieve the registered NF instances, the NRF shall return "403 Forbidden" status code.

If the NF Instance list retrieval fails at the NRF due to errors in the input data in the URI query parameters, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the discovery request fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

5.2.2.9 NFProfileRetrieval

5.2.2.9.1 General

This service operation allows the retrieval of the NF profile of a given NF instance currently registered in NRF.

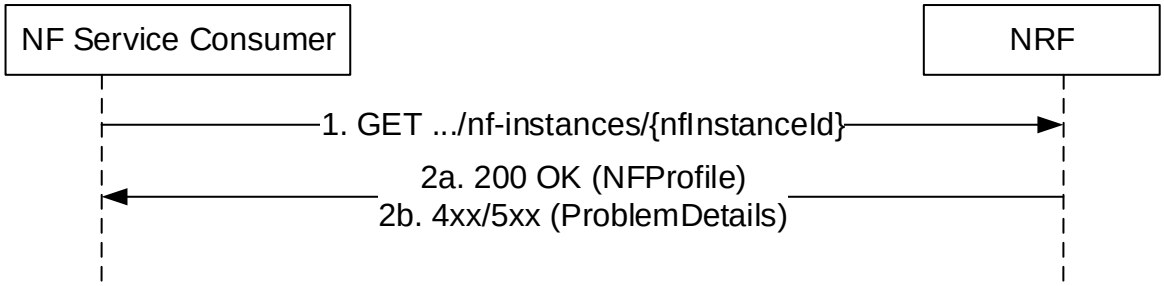

Figure 5.2.2.9.1-1: NF profile retrieval

1. The NF Service Consumer shall send an HTTP GET request to the resource URI "nf-instances/{nfInstanceId}".

2a. On success, "200 OK" shall be returned. The response body shall contain the NF profile of the NF instance identified in the request.

2b. If the NF Service Consumer is not allowed to retrieve the NF profile of this specific registered NF instance, the NRF shall return "403 Forbidden" status code.

If the NF Profile retrieval fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

5.3 Nnrf_NFDiscovery Service

5.3.1 Service Description

The Nnrf_NFDiscovery service allows a NF or SCP Instance to discover other NF Instances with the potential services they offer, by querying the local NRF.

The Nnrf_NFDiscovery service also allows to a SCP discover other SCP instances.

It also allows an NRF in a PLMN to re-issue a discovery request towards an NRF in another PLMN (e.g., the HPLMN of a certain UE).

5.3.2 Service Operations

5.3.2.1 Introduction

The service operations defined for the Nnrf_NFDiscovery service are as follows:

- NFDiscover: It provides to the NF service consumer or SCP the profile (including IP address(es) or FQDN) of the NF Instance(s) or NF Service(s) matching certain input criteria. It also provides to the SCP the profile (including IP address(es) or FQDN) of the SCP Instance(s) matching certain input criteria.

The NFDiscover operation can be invoked by an NF Service Consumer (i.e., "source NF") or SCP requesting to discover NF instances (i.e., "target NFs") located in the same PLMN, or in a different PLMN. It can also be invoked by an SCP requesting to discover SCP instances located in the same PLMN.

In the description of these operations in clause 5.3.2.2, when the NF instances are located in the same PLMN, both source NF and target NFs are said to be located in the "Serving PLMN" but, in the general case, the functionality is not restricted to the PLMN that is serving a given UE, and it shall be applicable as well to any scenario in which source NF and target NFs belong to the same PLMN.

When source NF and target NFs are located in different PLMNs, the source NF is said to be in the "Serving PLMN", and the target NFs (and the NRF where they are registered) are said to be in the "Home PLMN", similarly to the scenarios described in 3GPP TS 23.502 [3], but the functionality shall be equally applicable to any scenario between any pair of PLMNs (e.g. with the source NF in the Home PLMN and the target NF in the Serving PLMN).

The SCP is treated by the Nnrf_NFDiscovery service in the same way as NFs. Specifically, the SCP is designated with a specific NF type and NF Instance ID. However, the SCP does not support services. Accordingly, references to "NF" or "NF Profile" in the description of the service operations in the following clauses also apply to an SCP.

5.3.2.2 NFDiscover

5.3.2.2.1 General

This service operation discovers the set of NF Instances (and their associated NF Service Instances), represented by their NF Profile, that are currently registered in NRF and satisfy a number of input query parameters.

Before a service consumer invokes this service operation, it shall consider if it is possible to reuse the results from a previous searching (service discovery).

The service consumer should reuse the previous result if input query parameters in the new service discovery request are the same as used for the previous search and the validity period of the result is not expired.

The service consumer may consider reusing the previous result if the attributes as required for the new query is also part of NF profile of the candidates NFs from a previous query. In such case, when the results of a previous query are reused, the service consumer need consider that the results, e.g. in terms of the number of discovered NFs, can be different than the potential results obtained after performing a new query.

5.3.2.2.2 Service Discovery in the same PLMN

This service operation is executed by querying the "nf-instances" resource. The request is sent to an NRF in the same PLMN of the NF Service Consumer.

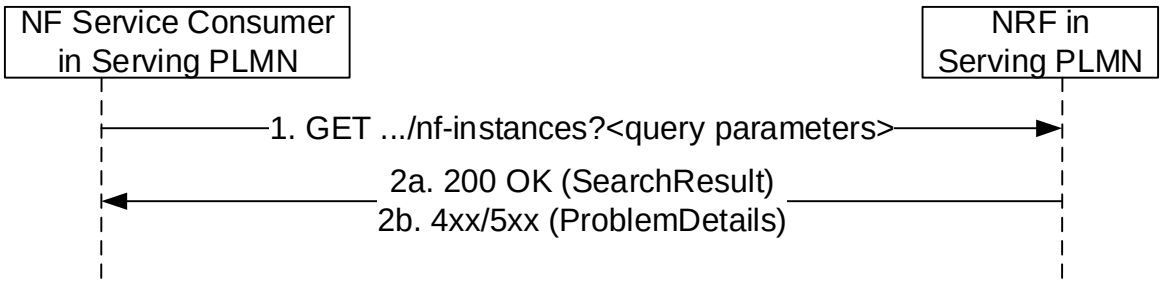

Figure 5.3.2.2.2-1: Service Discovery Request in the same PLMN

1. The NF Service Consumer shall send an HTTP GET request to the resource URI "nf-instances" collection resource. The input filter criteria for the discovery request shall be included in query parameters.

2a. On success, "200 OK" shall be returned. The response body shall contain a validity period, during which the search result can be cached by the NF Service Consumer, and an array of NF Profile objects, that satisfy the search filter criteria (e.g., all NF Instances offering a certain NF Service name).

2b. If the NF Service Consumer is not allowed to discover the NF services for the requested NF type provided in the query parameters, the NRF shall return "403 Forbidden" response.

If the discovery request fails at the NRF due to errors in the input data in the URI query parameters, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the discovery request fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

The NF Profile objects returned in a successful result shall contain generic data of each NF Instance, applicable to any NF type, and it may also contain NF-specific data, for those NF Instances belonging to a specific type (e.g., the attribute "udrInfo" is typically present in the NF Profile when the type of the NF Instance takes the value "UDR"). In addition, the attribute "customInfo", may be present in the NF Profile for those NF Instances with custom NF types.

For those NF Instances, the "customInfo" attribute shall be returned by NRF, if available, as part of the NF Profiles returned in the discovery response.

The NRF shall also include, in the returned NF Profile objects, the Vendor-Specific attributes (see 3GPP TS 29.500 [4], clause 6.6.3) that may have been provided by the registered NF Instances.

5.3.2.2.3 Service Discovery in a different PLMN

The service discovery in a different PLMN is done by querying the "nf-instances" resource in the NRF of the Home PLMN.

For that, step 1 in clause 5.3.2.2.2 is executed (send a GET request to the NRF in the Serving PLMN); this request shall include the identity of the PLMN of the home NRF in a query parameter of the URI.

Then, steps 1-2 in Figure 5.3.2.2.3-1 are executed, between the NRF in the Serving PLMN and the NRF in the Home PLMN. In this step, the presence of the PLMN ID of the Home NRF in the query parameter of the URI is not required. The NRF in the Home PLMN returns a status code with the result of the operation. The NRF in the Serving PLMN shall be configured with:

- a telescopic FQDN (see 3GPP TS 23.003 [12] and 3GPP TS 29.500 [4]) of the NRF in the Home PLMN, if TLS protection between the NRF and the SEPP in the serving PLMN relies on using telescopic FQDN; or

NOTE: This is required for the NRF in the serving PLMN to route the NF discovery request to the NRF in the HPLMN through a SEPP in the serving PLMN and the SEPP to terminate the TLS connection with a wildcard certificate.

- with the SEPP FQDN (or the FQDN of the SCP if the communication between the NRF and the SEPP goes through an SCP), if TLS protection between the NRF and the SEPP in the serving PLMN relies on using the 3gpp-Sbi-Target-apiRoot header.

See clause 6.1.4.3 of 3GPP TS 29.500 [4].

Finally, step 2 in clause 5.3.2.2.2 is executed; a status code is returned to the NF Service Consumer in Serving PLMN in accordance to the result received from NRF in Home PLMN.

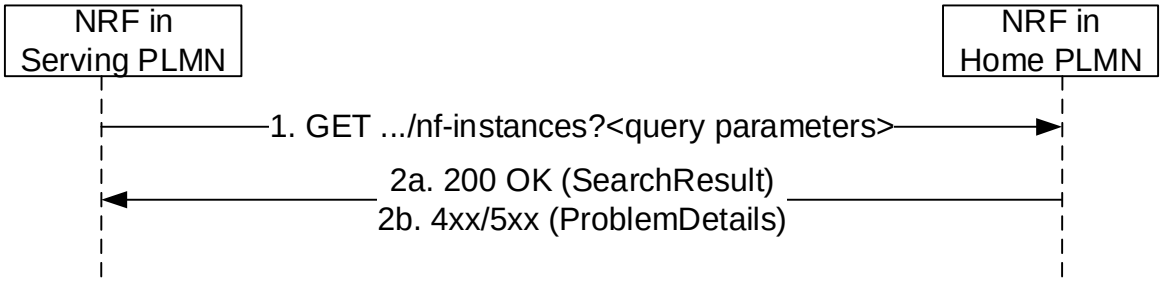

Figure 5.3.2.2.3-1: Service Discovery in a different PLMN

Steps 1 and 2 are similar to steps 1 and 2 in Figure 5.3.2.2.2-1, where the originator of the service invocation is the NRF in Serving PLMN, and the recipient of the service invocation is the NRF in the Home PLMN.

5.3.2.2.4 Service Discovery with intermediate redirecting NRF

When multiple NRFs are deployed in one PLMN, one NRF may query the "nf-instances" resource in a different NRF so as to fulfil the service discovery request from a NF service consumer. The query between these two NRFs is redirected by a third NRF.

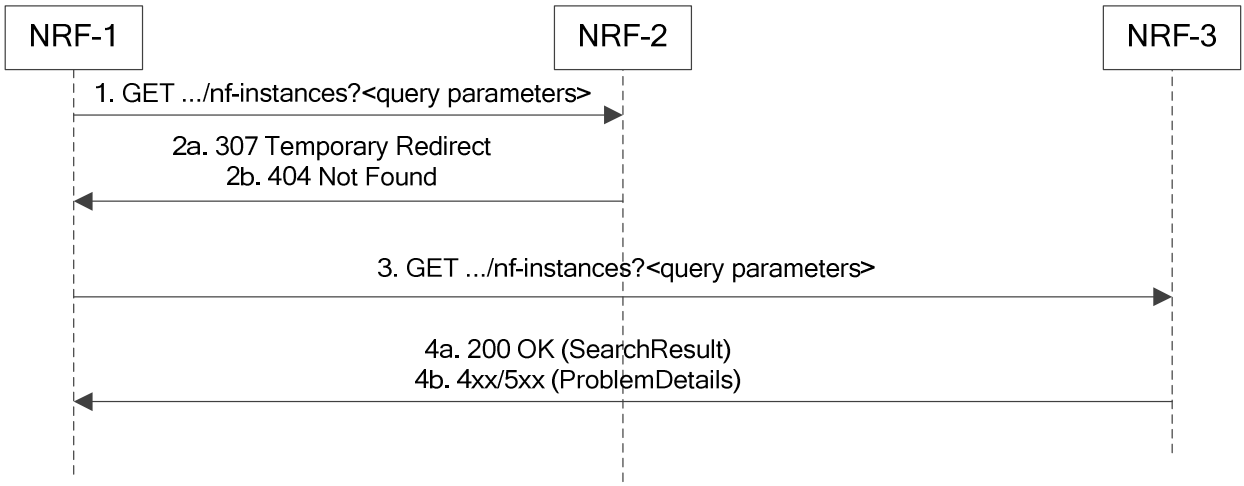

Figure 5.3.2.2.4-1: Service Discovery with intermediate redirecting NRF

1. NRF-1 receives a service discovery request but does not have the information to fulfil the request. Then NRF-1 sends the service discovery request to a pre-configured NRF-2.

2a. Upon receiving a service discovery request, based on the information contained in the service discovery request (e.g. the "supi" query parameter in the URI) and locally stored information NRF-2 shall identify the next hop NRF (see clause 5.2.2.2.3), and redirect the service discovery request by returning HTTP 307 Temporary Redirect response. The locally stored information in NRF-2 may:

   a) be preconfigured; or

   b) registered by other NRFs (see clause 5.2.2.2.3).

   The 307 Temporary Redirect response shall contain a Location header field, the host part of the URI in the Location header field represents NRF-3.

2b. if NRF-2 does not have enough information to redirect the service discovery request, then it responds with 404 Not Found, and the rest of the steps are omitted.

3. Upon receiving 307 Temporary Redirect response, NRF-1 sends the service discovery request to NRF-3 by using the URI contained in the Location header field of the 307 Temporary Redirect response.

4a. Upon success, NRF-3 returns the search result.

4b. If the NF Service Consumer is not allowed to discover the NF services for the requested NF type provided in the query parameters, the NRF shall return "403 Forbidden" response.

If the discovery request fails at the NRF due to errors in the input data in the URI query parameters, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the discovery request fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

5.3.2.2.5 Service Discovery with intermediate forwarding NRF

When multiple NRFs are deployed in one PLMN, one NRF may query the "nf-instances" resource in a different NRF so as to fulfil the service discovery request from a NF service consumer. The query between these two NRFs is forwarded by a third NRF.

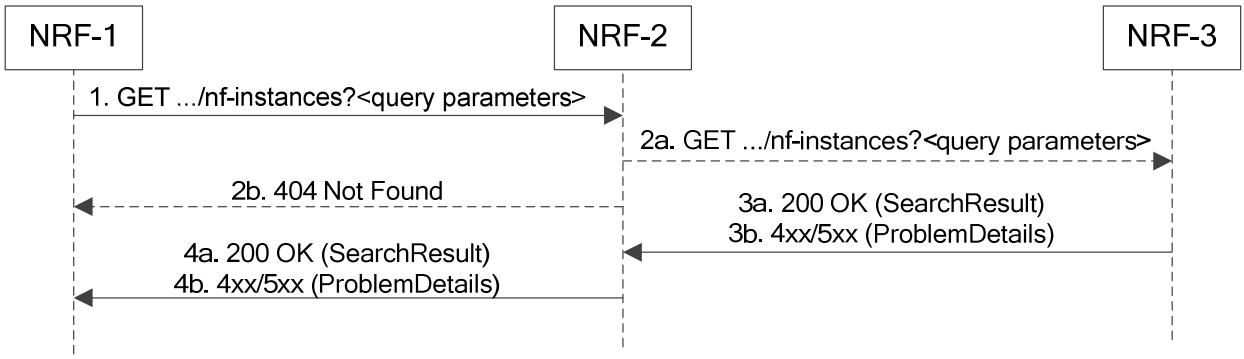

Figure 5.3.2.2.5-1: Service Discovery with intermediate forwarding NRF

1. NRF-1 receives a service discovery request but does not have the information to fulfil the request. Then NRF-1 sends the service discovery request to a pre-configured NRF-2.

2a. Upon receiving a service discovery request, based on the information contained in the service discovery request (e.g. the "supi" query parameter in the URI) and locally stored information, NRF-2 shall identify the next hop NRF (see clause 5.2.2.2.3), and forward the service discovery request to that NRF (i.e. NRF-3 in this example) similarly to steps 1 and 2 in Figure 5.3.2.2.2-1 where the originator of the service invocation is NRF-2 and the recipient of the service invocation is NRF-3. The locally stored information in NRF-2 may:

a) be preconfigured; or

b) registered by other NRFs (see clause 5.2.2.2.3).

2b. if NRF-2 does not have enough information to forward the service discovery request, then it responds with 404 Not Found, and the rest of the steps are omitted.

3a. Upon success, NRF-3 returns the search result.

3b. If the NF Service Consumer is not allowed to discover the NF services for the requested NF type provided in the query parameters, the NRF shall return "403 Forbidden" response.

If the discovery request fails at the NRF due to errors in the input data in the URI query parameters, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the discovery request fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

4a. NRF-2 forwards the success response to NRF-1.

4b. NRF-2 forwards the error response to NRF-1.

NOTE: It is not assumed that there can only be two NRF hierarchies, i.e. the NRF-3 can go on to forward the service discovery request to another NRF.

5.4 Nnrf_AccessToken Service

5.4.1 Service Description

The NRF offers an Nnrf_AccessToken service (used for OAuth2 authorization, see IETF RFC 6749 [16]), following the "Client Credentials" authorization grant, as specified in 3GPP TS 33.501 [15]. It exposes a "Token Endpoint" where the Access Token Request service can be requested by NF Service Consumers.

5.4.2 Service Operations

5.4.2.1 Introduction

The services operations defined for the Nnrf_AccessToken service are as follows:

- Access Token Request (i.e. Nnrf_AccessToken_Get)

5.4.2.2 Get (Access Token Request)

5.4.2.2.1 General

This service operation is used by an NF Service Consumer to request an OAuth2 access token from the authorization server (NRF).

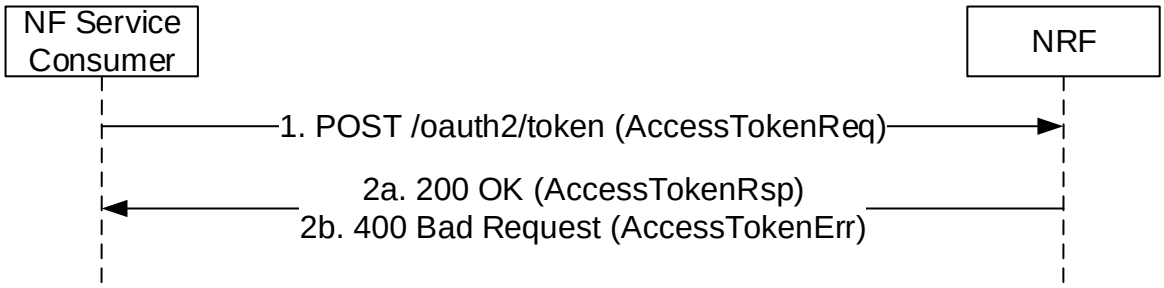

Figure 5.4.2.2.1-1: Access Token Request

1. The NF Service Consumer shall send a POST request to the "Token Endpoint", as described in IETF RFC 6749 [16], clause 3.2. The "Token Endpoint" URI shall be:

   {nrfApiRoot}/oauth2/token

   where {nrfApiRoot} represents the concatenation of the "scheme" and "authority" components of the NRF, as defined in IETF RFC 3986 [17].

   The OAuth 2.0 Access Token Request includes in the body of the HTTP POST request shall contain:

   - An OAuth2 grant type set to "client_credentials";
   - The "scope" parameter indicating the names of the NF Services that the NF Service Consumer is trying to access (i.e., the expected NF service names);
   - The NF Instance Id of the the NF Service Consumer requesting the OAuth2.0 access token;

- NF type of the NF Service Consumer, if this is an access token request not for a specific NF Service Producer;
- NF type of the expected NF Service Producer, if this is an access token request not for a specific NF Service Producer;
- The NF Instance Id of the expected NF Service Producer, if this is an access token request for a specific NF Service Producer;
- Home and Serving PLMN IDs, if this is an access token request for use in roaming scenarios (see clause 13.4.1.2 of 3GPP TS 33.501 [15]).

The request may additionally contain:

- the NF Set ID of the expected NF service producer instances, if this is an access token request not for a specific NF Service Producer.

The NF Service Consumer shall use TLS for mutual authentication with the NRF in order to access this endpoint, if the PLMN uses protection at the transport layer. Otherwise the NF Service Consumer shall use NDS or physical security to mutually authenticate with the NRF as specified in clause 13.3.1 of 3GPP TS 33.501 [15].

2. On success, "200 OK" shall be returned, the payload body of the POST response shall contain the requested access token and the token type set to value "Bearer". The response in addition:

   - should contain the expiration time for the token as indicated in IETF RFC 6749 [16] unless the expiration time of the token is made available by other means (e.g. deployment-specific documentation); and
   - shall contain the NF service name(s) of the requested NF service producer(s), if it is different from the scope included in the access token request (see IETF RFC 6749 [16]).

   The access token shall be a JSON Web Token (JWT) as specified in IETF RFC 7519 [25]. The access token returned by the NRF shall include the claims encoded as a JSON object as specified in clause 6.3.5.2.4 and then digitally signed using JWS as specified in IETF RFC 7515 [24] and in clause 13.4.1 of 3GPP TS 33.501 [15].

   The digitally signed access token shall be converted to the JWS Compact Serialization encoding as a string as specified in clause 7.1 of IETF RFC 7515 [24].

   If the access token request fails at the NRF, the NRF shall return "400 Bad Request" status code, including in the response payload a JSON object that provides details about the specific error that occurred.

5.4.2.2.2 Access Token request with intermediate forwarding NRF

When multiple NRFs are deployed in one PLMN, one NRF may request an OAuth2 access token to a different NRF so as to fulfil the Access Token Request from a NF service consumer. The acces token request between these two NRFs is forwarded by a third NRF in this case.

For this, step 1 in clause 5.4.2.2.1 is executed (send a POST request to NRF-1 in the Serving PLMN); this request shall include the OAuth 2.0 Access Token Request in the request body.

Then, steps 1-4 in Figure 5.4.2.2.2-1 hereinafter are executed between NRF-1in Serving PLMN, NRF-2 in Serving PLMN and NRF-3 in Serving PLMN.

Finally, step 2 in clause 5.4.2.2.1 is executed, the Access Token Response containing the requested access token, the token type and additional attributes shall be sent to the NF Service Consumer.

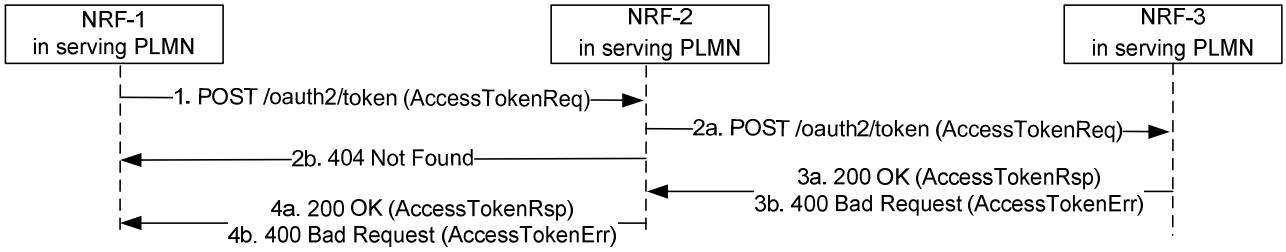

Figure 5.4.2.2.2-1: Access Token Request with intermediate forwarding NRF

1. NRF-1 receives an Access token request but does not have the information to fulfil the request. Then NRF-1 sends the Access token request to a pre-configured NRF-2.

2a. Upon reception of the Access token request and based on the information contained in the Acces token request and locally stored information, NRF-2 shall identify the next hop NRF (see clause 5.2.2.2.3), and forward the Access token request to that NRF (i.e. NRF-3 in this example) by replacing the originator of the service invocation with NRF-2, and the recipient of the service invocation with NRF-3. The locally stored information in NRF-2 may:

   a) be preconfigured; or

   b) registered by other NRFs (see clause 5.2.2.2.3).

2b. if NRF-2 does not have enough information to forward the Access token request, then it responds with 404 Not Found, and the rest of the steps are omitted.

3a. Upon success, NRF-3 shall return a "200 OK" status code, including in the response payload the Access token response containing the requested access token, the token type and additional attributes.

3b. Upon failure, NRF-3 shall return "400 Bad Request" status code, including in the response payload a JSON object that provides details about the specific error(s) that occurred.

4a. NRF-2 forwards the success response to NRF-1.

4b. NRF-2 forwards the error response to NRF-2.

NOTE: It is not assumed that there can only be two NRF hierarchies, i.e. the NRF-3 can go on and forward the Access token request request to another NRF.

5.4.2.2.3 Access Token request with intermediate redirecting NRF

When multiple NRFs are deployed in one PLMN, one NRF may request an OAuth2 access token to a different NRF so as to fulfil the Access Token Request from a NF service consumer. The acces token request between these two NRFs is redirected by a third NRF in this case.

For this, step 1 in clause 5.4.2.2.1 is executed (send a POST request to NRF-1 in the Serving PLMN); this request shall include the OAuth 2.0 Access Token Request in the request body

Then, steps 1-4 in Figure 5.4.2.2.3-1 hereinafter are executed between NRF-1in Serving PLMN, NRF-2 in Serving PLMN and NRF-3 in Serving PLMN.

Finally, step 2 in clause 5.4.2.2.1 is executed, the Access token response containing the requested access token, the token type and additional attributes shall be sent to the NF Service Consumer.

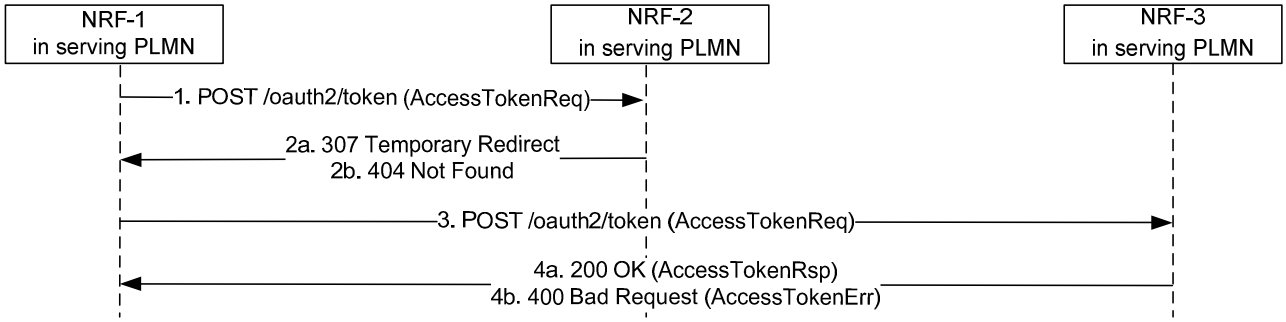

Figure 5.4.2.2.3-1: Access Token Request with intermediate redirecting NRF

1. NRF-1 receives an Access token request but does not have the information to fulfil the request. Then NRF-1 sends the Access token request to a pre-configured NRF-2.

2a. Upon reception of the Access token requestand based on the information contained in the Acces token request and locally stored information, NRF-2 shall identify the next hop NRF (see clause 5.2.2.2.3), and redirect the Access token request by returning HTTP "307 Temporary Redirect" response. The locally stored information in NRF-2 may:

   a) be preconfigured; or

   b) registered by other NRFs (see clause 5.2.2.2.3).

   The "307 Temporary Redirect" response shall contain a Location header field, the host part of the URI in the Location header field represents NRF-3.

2b. if NRF-2 does not have enough information to forward the Access token request, then it responds with "404 Not Found", and the rest of the steps are omitted.

3. Upon reception of "307 Temporary Redirect" response, NRF-1 sends the Acces token request to NRF-3 by using the URI contained in the Location header field of the "307 Temporary Redirect" response.

4a. Upon success, NRF-3shall return a "200 OK" status code including in the response payload the Access token response containing the requested access token, the token type and additional attributes.

4b. Upon failure, the NRF-3 shall return "400 Bad Request" status code, including in the response payload a JSON object that provides details about the specific error(s) that occurred.

5.5 Nnrf_Bootstrapping Service

5.5.1 Service Description

The NRF offers a Nnrf_Bootstrapping service to let NF Service Consumers of the NRF know about the services endpoints it supports, by using a version-independent URI endpoint that does not need to be discovered by using a Discovery service.

This service shall be used in inter-PLMN scenarios where the NRF in a PLMN-A needs to invoke services from an NRF in PLMN-B, when there is no pre-configured information indicating the version of the services deployed in PLMN-B.

This service may also be used in intra-PLMN scenarios, to avoid configuring statically in the different NFs information about the service versions deployed in the NRF to be used by those NFs.

5.5.2 Service Operations

5.5.2.1 Introduction

The services operations defined for the Nnrf_Bootstrapping service are as follows:

- Nnrf_Bootstrapping_Get

5.5.2.2 Get

5.5.2.2.1 General

This service operation is used by an NF Service Consumer to request bootstrapping information from the NRF.

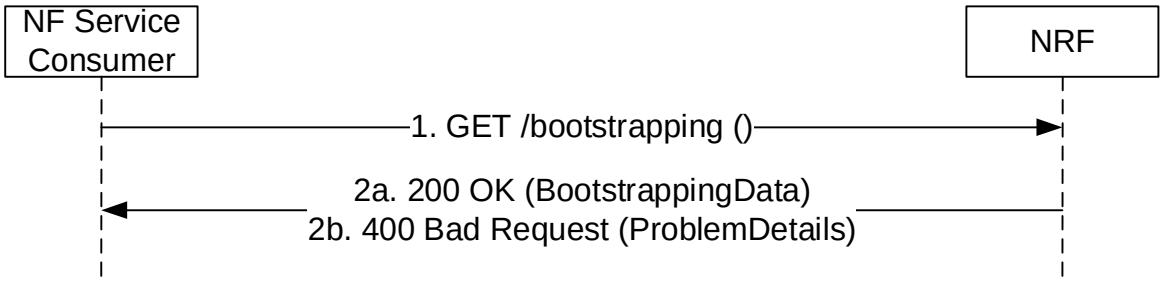

Figure 5.5.2.2.1-1: Bootstrapping Request

1. The NF Service Consumer shall send a GET request to the "Bootstrapping Endpoint", as described in 3GPP TS 23.003 [12], clause x.y. The "Bootstrapping Endpoint" URI shall be:

   {nrfApiRoot}/boostrapping

   where {nrfApiRoot} represents the concatenation of the "scheme" and "authority" components of the NRF, as defined in IETF RFC 3986 [17].

2. On success, "200 OK" shall be returned, the payload body of the GET response shall contain the requested bootstrapping information.

EXAMPLE:

```
GET https://nrf.example.com/bootstrapping
Accept: application/3gppHal+json


HTTP/2 200 OK
Content-Type: application/3gppHal+json

{
  "status": "OPERATIVE",
  "_links": {
    "self": {
      "href": "https://nrf.example.com/bootstrapping"
    },
    "manage": {
      "href": "https://nrf.example.com/nnrf-nfm/v1/nf-instances"
    },
    "subscribe": {
      "href": "https://nrf.example.com/nnrf-nfm/v1/subscriptions"
    },
    "discover": {
      "href": "https://nrf.example.com/nnrf-disc/v1/nf-instances"
    },
    "authorize": {
      "href": "https://nrf.example.com/oauth2/token"
    }
  }
}
```

6 API Definitions

6.1 Nnrf_NFManagement Service API

6.1.1 API URI

URIs of this API shall have the following root:

{apiRoot}/{apiName}/{apiVersion}/

where "apiRoot" is defined in clause 4.4.1 of 3GPP TS 29.501 [5], the "apiName" shall be set to "nnrf-nfm" and the "apiVersion" shall be set to "v1" for the current version of this specification.

6.1.2 Usage of HTTP

6.1.2.1 General

HTTP/2, as defined in IETF RFC 7540 [9], shall be used as specified in clause 5 of 3GPP TS 29.500 [4].

HTTP/2 shall be transported as specified in clause 5.3 of 3GPP TS 29.500 [4].

HTTP messages and bodies for the Nnrf_NFManagement service shall comply with the OpenAPI [10] specification contained in Annex A.

6.1.2.2 HTTP standard headers

6.1.2.2.1 General

The mandatory standard HTTP headers as specified in clause 5.2.2.2 of 3GPP TS 29.500 [4] shall be supported.

6.1.2.2.2 Content type

The following content types shall be supported:

- JSON, as defined in IETF RFC 8259 [22], shall be used as content type of the HTTP bodies specified in the present specification as indicated in clause 5.4 of 3GPP TS 29.500 [4].
- The Problem Details JSON Object (IETF RFC 7807 [11]). The use of the Problem Details JSON object in a HTTP response body shall be signalled by the content type "application/problem+json".
- JSON Patch (IETF RFC 6902 [13]). The use of the JSON Patch format in a HTTP request body shall be signalled by the content type "application/json-patch+json".
- The 3GPP hypermedia format as defined in 3GPP TS 29.501 [5]. The use of the 3GPP hypermedia format in a HTTP response body shall be signalled by the content type "application/3gppHal+json".

6.1.2.2.3 Accept-Encoding

The NRF should support gzip coding (see IETF RFC 1952 [30]) in HTTP requests and responses and indicate so in the Accept-Encoding header, as described in clause 6.9 of 3GPP TS 29.500 [4].

NF Service Consumers of the NFManagement API should support gzip coding in HTTP requests and responses and they should support gzip coding in the reception of notification requests sent by the NRF.

6.1.2.3 HTTP custom headers

6.1.2.3.1 General

In this release of this specification, no custom headers specific to the Nnrf_NFManagement service are defined. For 3GPP specific HTTP custom headers used across all service-based interfaces, see clause 5.2.3 of 3GPP TS 29.500 [4].

6.1.3 Resources

6.1.3.1 Overview

The structure of the Resource URIs of the NFManagement service is shown in figure 6.1.3.1-1.

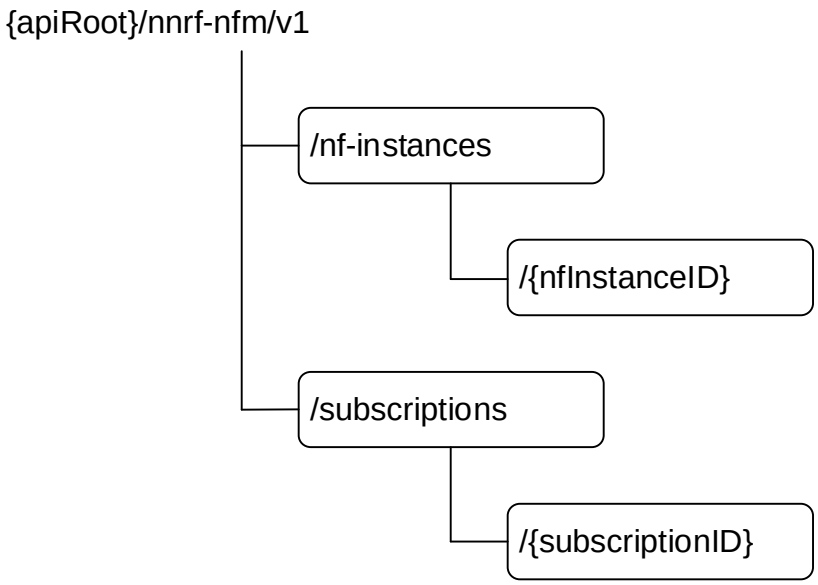

Figure 6.1.3.1-1: Resource URI structure of the NFManagement API

Table 6.1.3.1-1 provides an overview of the resources and applicable HTTP methods.

Table 6.1.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
nf-instances (Store)	/nf-instances	GET	Read a collection of NF Instances.
		OPTIONS	Discover the communication options supported by the NRF for this resource.
nf-instance (Document)	/nf-instances/{nfInstanceID}	GET	Read the profile of a given NF Instance.
		PUT	Register in NRF a new NF Instance, or replace the profile of an existing NF Instance, by providing an NF profile.
		PATCH	Modify the NF profile of an existing NF Instance.
		DELETE	Deregister from NRF a given NF Instance.
subscriptions (Collection)	/subscriptions	POST	Creates a new subscription in NRF to newly registered NF Instances.
subscription (Document)	/subscriptions/{subscriptionID}	PATCH	Updates an existing subscription in NRF.
		DELETE	Deletes an existing subscription from NRF.
Notification Callback	{nfStatusNotificationUri}	POST	Notify about newly created NF Instances, or about changes of the profile of a given NF Instance.

6.1.3.2 Resource: nf-instances (Store)

6.1.3.2.1 Description

This resource represents a collection of the different NF instances registered in the NRF.

This resource is modelled as the Store resource archetype (see clause C.3 of 3GPP TS 29.501 [5]).

6.1.3.2.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-nfm/v1/nf-instances**

This resource shall support the resource URI variables defined in table 6.1.3.2.2-1.

Table 6.1.3.2.2-1: Resource URI variables for this resource

Name	Data type	Definition
apiRoot	string	See clause 6.1.1

6.1.3.2.3 Resource Standard Methods

6.1.3.2.3.1 GET

This method retrieves a list of all NF instances currently registered in the NRF. This method shall support the URI query parameters specified in table 6.1.3.2.3.1-1.

Table 6.1.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
nf-type	NFType	O	0..1	The type of NF to restrict the list of returned NF Instances.
limit	integer	O	0..1	Maximum number of items to be returned in this query.

This method shall support the request data structures specified in table 6.1.3.2.3.1-2 and the response data structures and response codes specified in table 6.1.3.2.3.1-3.

Table 6.1.3.2.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.1.3.2.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
UriList	M	1	200 OK	The response body contains a "_links" object containing the URI of each registered NF in the NRF, or an empty object if there are no NFs to return in the query result (e.g., because there are no registered NFs in the NRF, or because there are no matching NFs of the type specified in the "nf-type" query parameter, currently registered in the NRF).
NOTE: The mandatory HTTP error status codes for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

6.1.3.2.3.2 OPTIONS

This method queries the communication options supported by the NRF (see clause 6.9 of 3GPP TS 29.500 [4]). This method shall support the URI query parameters specified in table 6.1.3.2.3.2-1.

Table 6.1.3.2.3.2-1: URI query parameters supported by the OPTIONS method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 6.1.3.2.3.2-2 and the response data structures and response codes specified in table 6.1.3.2.3.2-3.

Table 6.1.3.2.3.2-2: Data structures supported by the OPTIONS Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.1.3.2.3.2-3: Data structures supported by the OPTIONS Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	
OptionsResponse	M	1	200 OK	
ProblemDetails	O	0..1	405 Method Not Allowed	
ProblemDetails	O	0..1	501 Not Implemented	
NOTE: The mandatory HTTP error status codes for the OPTIONS method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

Table 6.1.3.2.3.2-4: Headers supported by the 200 Response Code on this resource

Name	Data type	P	Cardinality	Description
Accept-Encoding	string	O	0..1	Accept-Encoding, described in IETF RFC 7694 [41]

6.1.3.2.4 Resource Custom Operations

There are no resource custom operations for the Nnrf_NFManagement service in this release of the specification.

6.1.3.3 Resource: nf-instance (Document)

6.1.3.3.1 Description

This resource represents a single NF instance.

6.1.3.3.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-nfm/v1/nf-instances/{nfInstanceID}**

This resource shall support the resource URI variables defined in table 6.1.3.3.2-1.

Table 6.1.3.3.2-1: Resource URI variables for this resource

Name	Data type	Definition
apiRoot	string	See clause 6.1.1
nfInstanceID	NfInstanceId	Represents a specific NF Instance

6.1.3.3.3 Resource Standard Methods

6.1.3.3.3.1 GET

This method retrieves the NF Profile of a given NF instance.

This method shall support the URI query parameters specified in table 6.1.3.3.3.1-1.

Table 6.1.3.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 6.1.3.3.3.1-2 and the response data structures and response codes specified in table 6.1.3.3.3.1-3.

Table 6.1.3.3.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.1.3.3.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
NFProfile	M	1	200 OK	The response body contains the profile of a given NF Instance.
NOTE: The mandatory HTTP error status codes for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

6.1.3.3.3.2 PUT

This method registers a new NF instance in the NRF, or replaces completely an existing NF instance.

This method shall support the URI query parameters specified in table 6.1.3.3.3.2-1.

Table 6.1.3.3.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 6.1.3.3.3.2-2 and the response data structures and response codes specified in table 6.1.3.3.3.2-3.

Table 6.1.3.3.3.2-2: Data structures supported by the PUT Request Body on this resource

Data type	P	Cardinality	Description
NFProfile	M	1	Profile of the NF Instance to be registered, or completely replaced, in NRF.

Table 6.1.3.3.3.2-3: Data structures supported by the PUT Response Body on this resource

Data type	P	Cardinality	Response codes	Description
NFProfile	M	1	200 OK	This case represents the successful replacement of an existing NF Instance profile. Upon success, a response body is returned containing the replaced profile of the NF Instance.
NFProfile	M	1	201 Created	This case represents the successful registration of a new NF Instance. Upon success, a response body is returned containing the newly created NF Instance profile; also, the HTTP response shall include a "Location" HTTP header that contains the resource URI of the created NF Instance.
NOTE: The mandatory HTTP error status codes for the PUT method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

Table 6.1.3.3.3.2-4: Headers supported by the PUT method on this resource

Name	Data type	P	Cardinality	Description
Content-Encoding	string	O	0..1	Content-Encoding, described in IETF RFC 7231 [40]

Table 6.1.3.3.3.2-5: Headers supported by the 200 Response Code on this resource

Name	Data type	P	Cardinality	Description
Accept-Encoding	string	O	0..1	Accept-Encoding, described in IETF RFC 7694 [41]

Table 6.1.3.3.3.2-6: Headers supported by the 201 Response Code on this resource

Name	Data type	P	Cardinality	Description
Location	string	M	1	Contains the URI of the newly created resource, according to the structure: {apiRoot}/nnrf-nfm/v1/nf-instances/{nfInstanceId}
Accept-Encoding	string	O	0..1	Accept-Encoding, described in IETF RFC 7694 [41]

6.1.3.3.3 PATCH

This method updates partially the profile of a given NF instance.

This method shall support the URI query parameters specified in table 6.1.3.3.3.3-1.

Table 6.1.3.3.3.3-1: URI query parameters supported by the PATCH method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 6.1.3.3.3.3-2 and the response data structures and response codes specified in table 6.1.3.3.3.3-3.

Table 6.1.3.3.3.3-2: Data structures supported by the PATCH Request Body on this resource

Data type	P	Cardinality	Description
array(PatchItem)	M	1	It contains the list of changes to be made to the profile of the NF Instance, according to the JSON PATCH format specified in IETF RFC 6902 [13].

Table 6.1.3.3.3.3-3: Data structures supported by the PATCH Response Body on this resource

Data type	P	Cardinality	Response codes	Description
NFProfile	M	1	200 OK	Upon success, a response body is returned containing the updated profile of the NF Instance.
n/a			204 No Content	Successful response sent when there is no need to provide a full updated profile of the NF Instance (e.g., in the Heart-Beat operation response described in clause 5.2.2.3.2).
NOTE: The mandatory HTTP error status codes for the PATCH method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

6.1.3.3.4 DELETE

This method deregisters an existing NF instance from the NRF.

This method shall support the URI query parameters specified in table 6.1.3.3.3.4-1.

Table 6.1.3.3.3.4-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 6.1.3.3.3.4-2 and the response data structures and response codes specified in table 6.1.3.3.3.4-3.

Table 6.1.3.3.3.4-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.1.3.3.3.4-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	
NOTE: The mandatory HTTP error status codes for the DELETE method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

6.1.3.4 Resource: subscriptions (Collection)

6.1.3.4.1 Description

This resource represents a collection of subscriptions of NF Instances to newly registered NF Instances.

6.1.3.4.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-nfm/v1/subscriptions**

This resource shall support the resource URI variables defined in table 6.1.3.4.2-1.

Table 6.1.3.4.2-1: Resource URI variables for this resource

Name	Data type	Definition
apiRoot	string	See clause 6.1.1

6.1.3.4.3 Resource Standard Methods

6.1.3.4.3.1 POST

This method creates a new subscription. This method shall support the URI query parameters specified in table 6.1.3.4.3.1-1.

Table 6.1.3.4.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 6.1.3.4.3.1-2 and the response data structures and response codes specified in table 6.1.3.4.3.1-3.

Table 6.1.3.4.3.1-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
SubscriptionData	M	1	The request body contains the input parameters for the subscription. These parameters include, e.g.: - Target NF type - Target Service Name - Callback URI of the Requester NF

Table 6.1.3.4.3.1-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
SubscriptionData	M	1	201 Created	This case represents the successful creation of a subscription. Upon success, the HTTP response shall include a "Location" HTTP header that contains the resource URI of the created resource.
NOTE: The mandatory HTTP error status codes for the PUT method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

Table 6.1.3.4.3.1-4: Headers supported by the 201 Response Code on this resource

Name	Data type	P	Cardinality	Description
Location	string	M	1	Contains the URI of the newly created resource, according to the structure: {apiRoot}/nnrf-nfm/v1/subscriptions/{subscriptionId}

6.1.3.5 Resource: subscription (Document)

6.1.3.5.1 Description

This resource represents an individual subscription of a given NF Instance to newly registered NF Instances.

6.1.3.5.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-nfm/v1/subscriptions/{subscriptionID}**

This resource shall support the resource URI variables defined in table 6.1.3.5.2-1.

Table 6.1.3.5.2-1: Resource URI variables for this resource

Name	Data type	Definition
apiRoot	string	See clause 6.1.1
subscriptionID	string	Represents a specific subscription

6.1.3.5.3 Resource Standard Methods

6.1.3.5.3.1 DELETE

This method terminates an existing subscription. This method shall support the URI query parameters specified in table 6.1.3.5.3.1-1.

Table 6.1.3.5.3.1-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 6.1.3.5.3.1-2 and the response data structures and response codes specified in table 6.1.3.5.3.1-3.

Table 6.1.3.5.3.1-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.1.3.5.3.1-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	
NOTE: The mandatory HTTP error status codes for the PUT method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

6.1.3.5.3.2 PATCH

This method updates an existing subscription. This method shall support the URI query parameters specified in table 6.1.3.5.3.2-1.

Table 6.1.3.5.3.2-1: URI query parameters supported by the PATCH method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 6.1.3.5.3.2-2 and the response data structures and response codes specified in table 6.1.3.5.3.2-3.

Table 6.1.3.5.3.2-2: Data structures supported by the PATCH Request Body on this resource

Data type	P	Cardinality	Description
array(PatchItem)	M	1..N	It contains the list of changes to be made to the profile of the NF Instance, according to the JSON PATCH format specified in IETF RFC 6902 [13].

Table 6.1.3.5.3.2-3: Data structures supported by the PATCH Response Body on this resource

Data type	P	Cardinality	Response codes	Description
SubscriptionData	M	1	200 OK	
n/a			204 No Content	

6.1.4 Custom Operations without associated resources

There are no custom operations defined without any associated resources for the Nnrf_NFManagement service in this release of the specification.

6.1.5 Notifications

6.1.5.1 General

This clause specifies the notifications provided by the Nnrf_NFManagement service.

The delivery of notifications shall be supported as specified in clause 6.2 of 3GPP TS 29.500 [4] for Server-initiated communication.

Table 6.1.5.1-1: Notifications overview

Notification	Resource URI	HTTP method or custom operation	Description (service operation)
NF Instance Status Notification	{nfStatusNotificationUri} (NF Service Consumer provided callback reference)	POST	Notify about registrations / deregistrations or profile changes of NF Instances

6.1.5.2 NF Instance Status Notification

6.1.5.2.1 Description

The NF Service Consumer provides a callback URI for getting notified about NF Instances status events, the NRF shall notify the NF Service Consumer, when the conditions specified in the subscription are met.

6.1.5.2.2 Notification Definition

The POST method shall be used for NF Instance Status notification and the URI shall be the callback reference provided by the NF Service Consumer during the subscription to this notification.

Resource URI: **{nfStatusNotificationUri}**

Support of URI query parameters is specified in table 6.1.5.2.2-1.

Table 6.1.5.2.2-1: URI query parameters supported by the POST method

Name	Data type	P	Cardinality	Description
n/a				

Support of request data structures is specified in table 6.1.5.2.2-2, and support of response data structures and response codes is specified in table 6.1.5.2-3.

Table 6.1.5.2.2-2: Data structures supported by the POST Request Body

Data type	P	Cardinality	Description
NotificationData	M	1	Representation of the NF Instance status notification.

Table 6.1.5.2.2-3: Data structures supported by the POST Response Body

Data type	P	Cardinality	Response codes	Description
N/A			204 No Content	This case represents a successful notification of the NF Instance status event.
NOTE: The mandatory HTTP error status codes for the PUT method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

6.1.6 Data Model

6.1.6.1 General

This clause specifies the application data model supported by the API.

Table 6.1.6.1-1 specifies the data types defined for the Nnrf_NFManagement service-based interface protocol.

Table 6.1.6.1-1: Nnrf_NFManagement specific Data Types

Data type	Clause defined	Description
NFProfile	6.1.6.2.2	Information of an NF Instance registered in the NRF.
NFService	6.1.6.2.3	Information of a given NF Service Instance; it is part of the NFProfile of an NF Instance.
DefaultNotificationSubscription	6.1.6.2.4	Data structure for specifying the notifications the NF service subscribes by default along with callback URI.
IpEndPoint	6.1.6.2.5	IP addressing information of a given NFService; it consists on, e.g. IP address, TCP port, transport protocol...
UdrInfo	6.1.6.2.6	Information of an UDR NF Instance.
UdmInfo	6.1.6.2.7	Information of an UDM NF Instance.
AusfInfo	6.1.6.2.8	Information of an AUSF NF Instance.
SupiRange	6.1.6.2.9	A range of SUPIs (subscriber identities), either based on a numeric range, or based on regular-expression matching.
IdentityRange	6.1.6.2.10	A range of GPSIs (subscriber identities), either based on a numeric range, or based on regular-expression matching.
AmfInfo	6.1.6.2.11	Information of an AMF NF Instance.
SmfInfo	6.1.6.2.12	Information of an SMF NF Instance.
UpfInfo	6.1.6.2.13	Information of an UPF NF Instance.
SnssaiUpfInfoItem	6.1.6.2.14	Set of parameters supported by UPF for a given S-NSSAI.
DnnUpfInfoItem	6.1.6.2.15	Set of parameters supported by UPF for a given DNN.
SubscriptionData	6.1.6.2.16	Information of a subscription to notifications to NRF events, included in subscription requests and responses.
NotificationData	6.1.6.2.17	Data sent in notifications from NRF to subscribed NF Instances.
NFServiceVersion	6.1.6.2.19	Contains the version details of an NF service.
PcfInfo	6.1.6.2.20	Information of a PCF NF Instance.
BsfInfo	6.1.6.2.21	Information of a BSF NF Instance.
Ipv4AddressRange	6.1.6.2.22	Range of IPv4 addresses.
Ipv6PrefixRange	6.1.6.2.23	Range of IPv6 prefixes.
InterfaceUpfInfoItem	6.1.6.2.24	Information of a given IP interface of an UPF.
UriList	6.1.6.2.25	Set of URIs following 3GPP hypermedia format (containing a "_links" attribute).
N2InterfaceAmfInfo	6.1.6.2.26	AMF N2 interface information
TaiRange	6.1.6.2.27	Range of TAIs (Tracking Area Identities).
TacRange	6.1.6.2.28	Range of TACs (Tracking Area Codes).
SnssaiSmfInfoItem	6.1.6.2.29	Set of parameters supported by SMF for a given S-NSSAI.
DnnSmfInfoItem	6.1.6.2.30	Set of parameters supported by SMF for a given DNN.
NrfInfo	6.1.6.2.31	Information of an NRF NF Instance, used in hierarchical NRF deployments.
ChfInfo	6.1.6.2.32	Information of a CHF NF Instance.
ChfServiceInfo	6.1.6.2.33	Information of primary and secondary CHF services.
PlmnRange	6.1.6.2.34	Range of PLMN IDs.
SubscrCond	6.1.6.2.35	Condition to determine the set of NFs to monitor under a certain subscription in NRF.
NfInstanceIdCond	6.1.6.2.36	Subscription to a given NF Instance Id.
NfTypeCond	6.1.6.2.37	Subscription to a set of NFs based on their NF Type.
ServiceNameCond	6.1.6.2.38	Subscription to a set of NFs based on their support for a given Service Name.
AmfCond	6.1.6.2.39	Subscription to a set of AMFs, based on AMF Set Id and/or AMF Region Id.
GuamiListCond	6.1.6.2.40	Subscription to a set of AMFs, based on their GUAMIs.
NetworkSliceCond	6.1.6.2.41	Subscription to a set of NFs, based on the slices (S-NSSAI and NSI) they support .
NfGroupCond	6.1.6.2.42	Subscription to a set of NFs based on their Group Id.
NotifCondition	6.1.6.2.43	Condition (list of attributes in the NF Profile) to determine whether a notification must be sent by NRF.
PlmnSnssai	6.1.6.2.44	List of network slices (S-NSSAIs) for a given PLMN ID.
NwdafInfo	6.1.6.2.45	Information of a NWDAF NF Instance.
LmfInfo	6.1.6.2.46	Information of an LMF NF Instance.
GmlcInfo	6.1.6.2.47	Information of a GMLC NF Instance.
NefInfo	6.1.6.2.48	Information of an NEF NF Instance.
PfdData	6.1.6.2.49	List of Application IDs and/or AF IDs managed by a given NEF Instance.
AfEventExposureData	6.1.6.2.50	AF Event Exposure data managed by a given NEF Instance.
WAgfInfo	6.1.6.2.51	Information of the W-AGF endpoints.

TngfInfo	6.1.6.2.52	Information of the TNGF endpoints.
PcscfInfo	6.1.6.2.53	Information of a P-CSCF NF Instance.
NfSetCond	6.1.6.2.54	Subscription to a set of NFs based on their Set Id.
NfServiceSetCond	6.1.6.2.55	Subscription to a set of NFs based on their Service Set Id.
NfInfo	6.1.6.2.56	Information of a generic NF Instance.
HssInfo	6.1.6.2.57	Information of an HSS NF Instance.
ImsiRange	6.1.6.2.58	A range of IMSIs (subscriber identities), either based on a numeric range, or based on regular-expression matching.
InternalGroupIdRange	6.1.6.2.59	A range of Group IDs (internal group identities), either based on a numeric range, or based on regular-expression matching.
UpfCond	6.1.6.2.60	Subscription to a set of NF Instances (UPFs), able to serve a certain service area (i.e. SMF serving area or TAI list).
TwifInfo	6.1.6.2.61	Addressing information (IP addresses, FQDN) of the TWIF.
VendorSpecificFeature	6.1.6.2.62	Information about a vendor-specific feature
UdsfInfo	6.1.6.2.63	Information related to UDSF
ScpInfo	6.1.6.2.65	Information of an SCP Instance
ScpDomainInfo	6.1.6.2.66	SCP domain information
ScpDomainCond	6.1.6.2.67	Subscription to an SCP domain
OptionsResponse	6.1.6.2.68	Communication options of the NRF
Fqdn	6.1.6.3.2	Fully Qualified Domain Name.
NefId	6.1.6.3.2	Identity of the NEF.
VendorId	6.1.6.3.2	Vendor ID of the NF Service instance (Private Enterprise Number assigned by IANA)
NFType	6.1.6.3.3	NF types known to NRF.
NotificationType	6.1.6.3.4	Types of notifications used in Default Notification URIs in the NF Profile of an NF Instance.
TransportProtocol	6.1.6.3.5	Types of transport protocol used in a given IP endpoint of an NF Service Instance.
NotificationEventType	6.1.6.3.6	Types of events sent in notifications from NRF to subscribed NF Instances.
NFStatus	6.1.6.3.7	Status of a given NF Instance stored in NRF.
DataSetId	6.1.6.3.8	Types of data sets stored in UDR.
UPInterfaceType	6.1.6.3.9	Types of User-Plane interfaces of the UPF.
ServiceName	6.1.6.3.11	Service names known to NRF.
NFServiceStatus	6.1.6.3.12	Status of a given NF Service Instance of an NF Instance stored in NRF.
AnNodeType	6.1.6.3.13	Access Network Node Type (gNB, ng-eNB...).

Table 6.1.6.1-2 specifies data types re-used by the Nnrf_NFManagement service-based interface protocol from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the Nnrf_NFManagement service-based interface.

Table 6.1.6.1-2: Nnrf_NFManagement re-used Data Types

Data type	Reference	Comments
N1MessageClass	3GPP 29.518 [6]	The N1 message type
N2InformationClass	3GPP 29.518 [6]	The N2 information type
IPv4Addr	3GPP TS 29.571 [7]	
IPv6Addr	3GPP TS 29.571 [7]	
IPv6Prefix	3GPP TS 29.571 [7]	
Uri	3GPP TS 29.571 [7]	
Dnn	3GPP TS 29.571 [7]	
SupportedFeatures	3GPP TS 29.571 [7]	
Snssai	3GPP TS 29.571 [7]	
PlmnId	3GPP TS 29.571 [7]	
Guami	3GPP TS 29.571 [7]	
Tai	3GPP TS 29.571 [7]	
NfInstanceId	3GPP TS 29.571 [7]	
LinksValueSchema	3GPP TS 29.571 [7]	3GPP Hypermedia link
UriScheme	3GPP TS 29.571 [7]	
AmfName	3GPP TS 29.571 [7]	
DateTime	3GPP TS 29.571 [7]	
Dnai	3GPP TS 29.571 [7]	
ChangeItem	3GPP TS 29.571 [7]	
DiameterIdentity	3GPP TS 29.571 [7]	
AccessType	3GPP TS 29.571 [7]	
NfGroupId	3GPP TS 29.571 [7]	Network Function Group Id
AmfRegionId	3GPP TS 29.571 [7]	
AmfSetId	3GPP TS 29.571 [7]	
PduSessionType	3GPP TS 29.571 [7]	
AtsssCapability	3GPP TS 29.571 [7]	Capability to support procedures related to Access Traffic Steering, Switching, Splitting.
Nid	3GPP TS 29.571 [7]	
PlmnIdNid	3GPP TS 29.571 [7]	
NfSetId	3GPP TS 29.571 [7]	NF Set ID (see clause 28.12 of 3GPP TS 23.003 [12])
NfServiceSetId	3GPP TS 29.571 [7]	NF Service Set ID (see clause 28.13 of 3GPP TS 23.003 [12])
GroupId	3GPP TS 29.571 [7]	Internal Group Identifier
RatType	3GPP TS 29.571 [7]	RAT Type
EventId	3GPP TS 29.520 [32]	Defined in Nnwdaf_AnalyticsInfo API.
NwdafEvent	3GPP TS 29.520 [32]	Defined in Nnwdaf_EventsSubscription API.
ExternalClientType	3GPP TS 29.572 [33]	
LMFIdentification	3GPP TS 29.572 [33]	LMF Identification
AfEvent	3GPP TS 29.517 [35]	Defined in Naf_EventExposure API

6.1.6.2 Structured data types

6.1.6.2.1 Introduction

This clause defines the structures to be used in resource representations.

6.1.6.2.2 Type: NFProfile

Table 6.1.6.2.2-1: Definition of type NFProfile

Attribute name	Data type	P	Cardinality	Description
nfInstanceId	NfInstanceId	M	1	Unique identity of the NF Instance.
nfType	NFType	M	1	Type of Network Function
nfStatus	NFStatus	M	1	Status of the NF Instance (NOTE 5)
nfInstanceName	string	O	0..1	Human readable name of the NF Instance
heartBeatTimer	integer	C	0..1	Time in seconds expected between 2 consecutive heart-beat messages from an NF Instance to the NRF. It may be included in the registration request. When present in the request it shall contain the heartbeat time proposed by the NF service consumer. It shall be included in responses from NRF to registration requests (PUT) or in NF profile updates (PUT or PATCH). If the proposed heartbeat time is acceptable by the NRF based on the local configuration, it shall use the same value as in the registration request; otherwise the NRF shall override the value using a preconfigured value.
plmnList	array(PlmnId)	C	1..N	PLMN(s) of the Network Function (NOTE 7). This IE shall be present if this information is available for the NF. If not provided, PLMN ID(s) of the PLMN of the NRF are assumed for the NF.
snpnList	array(PlmnIdNid)	C	1..N	SNPN(s) of the Network Function. This IE shall be present if the NF pertains to one or more SNPNs.
sNssais	array(ExtSnssai)	O	1..N	S-NSSAIs of the Network Function. If not provided, the NF can serve any S-NSSAI. When present this IE represents the list of S-NSSAIs supported in all the PLMNs listed in the plmnList IE. If the sNSSAIs attribute is provided in at least one NF Service, the S-NSSAIs supported by the NF Profile shall be the set or a superset of the S-NSSAIs of the NFService(s).
perPlmnSnssaiList	array(PlmnSnssai)	O	1..N	This IE may be included when the list of S-NSSAIs supported by the NF for each PLMN it is supporting is different. When present, this IE shall include the S-NSSAIs supported by the Network Function for each PLMN supported by the Network Function. When present, this IE shall override sNssais IE. (NOTE 9) If the perPlmnSnssaiList attribute is provided in at least one NF Service, the S-NSSAIs supported per PLMN in the NF Profile shall be the set or a superset of the perPlmnSnssaiList of the NFService(s).
nsiList	array(string)	O	1..N	NSI identities of the Network Function. If not provided, the NF can serve any NSI.
fqdn	Fqdn	C	0..1	FQDN of the Network Function (NOTE 1) (NOTE 2). For AMF, the FQDN registered with the NRF shall be that of the AMF Name (see 3GPP 23.003 [12] clause 28.3.2.5).
interPlmnFqdn	Fqdn	C	0..1	If the NF needs to be discoverable by other NFs in a different PLMN, then an FQDN that is used for inter-PLMN routing as specified in 3GPP 23.003 [12] shall be registered with the NRF (NOTE 8). A change of this attribute shall result in triggering a "NF_PROFILE_CHANGED" notification from NRF towards subscribing NFs located in a different PLMN, but the new value shall be notified as a change of the "fqdn" attribute.
ipv4Addresses	array(Ipv4Addr)	C	1..N	IPv4 address(es) of the Network Function (NOTE 1) (NOTE 2)
ipv6Addresses	array(Ipv6Addr)	C	1..N	IPv6 address(es) of the Network Function (NOTE 1) (NOTE 2)

allowedPlmns	array(PlmnId)	O	1..N	PLMNs allowed to access the NF instance. If not provided, any PLMN is allowed to access the NF. A change of this attribute shall not trigger a "NF_PROFILE_CHANGED" notification from NRF, and this attribute shall not be included in profile change notifications to subscribed NFs.
allowedSnpns	array(PlmnIdNid)	O	1..N	SNPNs allowed to access the NF instance. If this attribute is present in the NFService and in the NF profile, the attribute from the NFService shall prevail. The absence of this attribute in both the NFService and in the NF profile indicates that no SNPN, other than the SNPN(s) registered in the snpnList attribute of the NF Profile, is allowed to access the service instance. A change of this attribute shall not trigger a "NF_PROFILE_CHANGED" notification from NRF, and this attribute shall not be included in profile change notifications to subscribed NFs.
allowedNfTypes	array(NFType)	O	1..N	Type of the NFs allowed to access the NF instance. If not provided, any NF type is allowed to access the NF. A change of this attribute shall not trigger a "NF_PROFILE_CHANGED" notification from NRF, and this attribute shall not be included in profile change notifications to subscribed NFs.
allowedNfDomains	array(string)	O	1..N	Pattern (regular expression according to the ECMA-262 dialect [8]) representing the NF domain names within the PLMN of the NRF allowed to access the NF instance. If not provided, any NF domain is allowed to access the NF. A change of this attribute shall not trigger a "NF_PROFILE_CHANGED" notification from NRF, and this attribute shall not be included in profile change notifications to subscribed NFs.
allowedNssais	array(ExtSnssai)	O	1..N	S-NSSAI of the allowed slices to access the NF instance. If not provided, any slice is allowed to access the NF. A change of this attribute shall not trigger a "NF_PROFILE_CHANGED" notification from NRF, and this attribute shall not be included in profile change notifications to subscribed NFs.
priority	integer	O	0..1	Priority (relative to other NFs of the same type) within the range 0 to 65535, to be used for NF selection; lower values indicate a higher priority. Priority may or may not be present in the nfServiceList parameters, xxxInfo parameters and in this attribute. Priority in the nfServiceList has precedence over the priority in this attribute, which has precedence over the priority in xxxInfo parameter. (NOTE 4). The NRF may overwrite the received priority value when exposing an NFProfile with the Nnrf_NFDiscovery service.

capacity	integer	O	0..1	Static capacity information within the range 0 to 65535, expressed as a weight relative to other NF instances of the same type; if capacity is also present in the nfServiceList parameters, those will have precedence over this value. (NOTE 4).
load	integer	O	0..1	Dynamic load information, within the range 0 to 100, indicates the current load percentage of the NF.
loadTimeStamp	DateTime	O	0..1	It indicates the point in time in which the latest load information (sent by the NF in the "load" attribute of the NF Profile) was generated at the NF Instance. If the NF did not provide a timestamp, the NRF should set it to the instant when the NRF received the message where the NF provided the latest load information.
locality	string	O	0..1	Operator defined information about the location of the NF instance (e.g. geographic location, data center) (NOTE 3)
udrInfo	UdrInfo	O	0..1	Specific data for the UDR (ranges of SUPI, group ID …)
udrInfoList	map(UdrInfo)	O	1..N	Multiple entries of UdrInfo. This attribute provides additional information to the udrInfo. udrInfoList may be present even if the udrInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
udmInfo	UdmInfo	O	0..1	Specific data for the UDM (ranges of SUPI, group ID…)
udmInfoList	map(UdmInfo)	O	1..N	Multiple entries of UdmInfo. This attribute provides additional information to the udmInfo. udmInfoList may be present even if the udmInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
ausfInfo	AusfInfo	O	0..1	Specific data for the AUSF (ranges of SUPI, group ID…)
ausfInfoList	map(AusfInfo)	O	1..N	Multiple entries of AusfInfo. This attribute provides additional information to the ausfInfo. ausfInfoList may be present even if the ausfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
amfInfo	AmfInfo	O	0..1	Specific data for the AMF (AMF Set ID, …)
amfInfoList	map(AmfInfo)	O	1..N	Multiple entries of AmfInfo. This attribute provides additional information to the amfInfo. amfInfoList may be present even if the amfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
smfInfo	SmfInfo	O	0..1	Specific data for the SMF (DNN's, …). (NOTE 12)
smfInfoList	map(SmfInfo)	O	1..N	Multiple entries of SmfInfo. This attribute provides additional information to the smfInfo. smfInfoList may be present even if the smfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters. (NOTE 12)
upfInfo	UpfInfo	O	0..1	Specific data for the UPF (S-NSSAI, DNN, SMF serving area, interface…)
upfInfoList	map(UpfInfo)	O	1..N	Multiple entries of UpfInfo. This attribute provides additional information to the upfInfo. upfInfoList may be present even if the upfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
pcfInfo	PcfInfo	O	0..1	Specific data for the PCF

pcfInfoList	map(PcfInfo)	O	1..N	Multiple entries of PcfInfo. This attribute provides additional information to the pcfInfo. pcfInfoList may be present even if the pcfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
bsfInfo	BsfInfo	O	0..1	Specific data for the BSF
bsfInfoList	map(BsfInfo)	O	1..N	Multiple entries of BsfInfo. This attribute provides additional information to the bsfInfo. bsfInfoList may be present even if the bsfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
chfInfo	ChfInfo	O	0..1	Specific data for the CHF
chfInfoList	map(ChfInfo)	O	1..N	Multiple entries of ChfInfo. This attribute provides additional information to the chfInfo. chfInfoList may be present even if the chfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
nefInfo	NefInfo	O	0..1	Specific data for the NEF
nrfInfo	NrfInfo	O	0..1	Specific data for the NRF
udsfInfo	UdsfInfo	O	0..1	Specific data for the UDSF
udsfInfoList	map(UdsfInfo)	O	1..N	Multiple entries of udsfInfo. This attribute provides additional information to the udsfInfo. udsfInfoExt may be present even if the udsfInfo is absent.
nwdafInfo	NwdafInfo	O	0..1	Specific data for the NWDAF.
pcscfInfoList	map(PcscfInfo)	O	1..N	Specific data for the P-CSCF. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters. (NOTE 11)
hssInfoList	map(HssInfo)	O	1..N	Specific data for the HSS. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
customInfo	object	O	0..1	Specific data for custom Network Functions
recoveryTime	DateTime	O	0..1	Timestamp when the NF was (re)started (NOTE 5) (NOTE 6)
nfServicePersistence	boolean	O	0..1	- true: If present, and set to true, it indicates that the different service instances of a same NF Service in this NF instance, supporting a same API version, are capable to persist their resource state in shared storage and therefore these resources are available after a new NF service instance supporting the same API version is selected by a NF Service Consumer (see 3GPP 23.527 [27]). - false (default): Otherwise, it indicates that the NF Service Instances of a same NF Service are not capable to share resource state inside the NF Instance.
nfServices	array(NFService)	O	1..N	List of NF Service Instances. It shall include the services produced by the NF that can be discovered by other NFs, if any. This attribute is deprecated; the attribute "nfServiceList" should be used instead.
nfServiceList	map(NFService)	O	1..N	Map of NF Service Instances, where the "serviceInstanceId" attribute of the NFService object shall be used as the key of the map. It shall include the services produced by the NF that can be discovered by other NFs, if any.

nfProfileChangesSupportInd	boolean	O	0..1	NF Profile Changes Support Indicator. See Annex B. This IE may be present in the NFRegister or NFUpdate (NF Profile Complete Replacement) request and shall be absent in the response. true: the NF Service Consumer supports receiving NF Profile Changes in the response. false (default): the NF Service Consumer does not support receiving NF Profile Changes in the response. Write-Only: true
nfProfileChangesInd	boolean	O	0..1	NF Profile Changes Indicator. See Annex B. This IE shall be absent in the request to the NRF and may be included by the NRF in NFRegister or NFUpdate (NF Profile Complete Replacement) response. true: the NF Profile contains NF Profile changes. false (default): complete NF Profile. Read-Only: true
defaultNotificationSubscriptions	array(DefaultNotificationSubscription)	O	1..N	Notification endpoints for different notification types. (NOTE 10)
lmfInfo	LmfInfo	O	0..1	Specific data for the LMF
gmlcInfo	GmlcInfo	O	0..1	Specific data for the GMLC
nfSetIdList	array(NfSetId)	C	1..N	NF Set ID defined in clause 28.12 of 3GPP TS 23.003 [12]. At most one NF Set ID shall be indicated per PLMN of the NF. This information shall be present if available.
servingScope	array(string)	O	1..N	The served area(s) of the NF instance. The absence of this attribute does not imply that the NF instance can serve every area in the PLMN. (NOTE 13)
lcHSupportInd	boolean	O	0..1	This IE indicates whether the NF supports Load Control based on LCI Header (see clause 6.3 of 3GPP TS 29.500 [4]). - true: the NF supports the feature. - false (default): the NF does not support the feature.
olcHSupportInd	boolean	O	0..1	This IE indicates whether the NF supports Overload Control based on OCI Header (see clause 6.4 of 3GPP TS 29.500 [4]). - true: the NF supports the feature. - false (default): the NF does not support the feature.
nfSetRecoveryTimeList	map(DateTime)	O	1..N	Map of recovery time, where the key of the map is the *NfSetId* of NF Set(s) that the NF instance belongs to. When present, the value of each entry of the map shall be the recovery time of the NF Set indicated by the key.
serviceSetRecoveryTimeList	map(DateTime)	O	1..N	Map of recovery time, where the key of the map is the *NfServiceSetId* of the NF Service Set(s) configured in the NF instance. When present, the value of each entry of the map shall be the recovery time of the NF Service Set indicated by the key.

scpDomains	array(string)	O	1..N	List of SCP domains the NF belongs to. (NOTE 14)
scpInfo	ScpInfo	O	0..1	Specific data for the SCP

NOTE 1: At least one of the addressing parameters (fqdn, ipv4address or ipv6adress) shall be included in the NF Profile. If the NF supports the NF services with "https" URI scheme (i.e use of TLS is mandatory), then the FQDN shall be provided in the NF Profile or the NF Service profile (see clause 6.1.6.2.3). See NOTE 1 of Table 6.1.6.2.3-1 for the use of these parameters. If multiple ipv4 addresses and/or ipv6 addresses are included in the NF Profile, the NF Service Consumer of the discovery service shall select one of these addresses randomly, unless operator defined local policy of IP address selection, in order to avoid overload for a specific ipv4 address and/or ipv6 address.
NOTE 2: If the type of Network Function is UPF, the addressing information is for the UPF N4 interface.
NOTE 3: A requester NF may use this information to select a NF instance (e.g. a NF instance preferably located in the same data center).
NOTE 4: The capacity and priority parameters, if present, are used for NF selection and load balancing. The priority and capacity attributes shall be used for NF selection in the same way that priority and weight are used for server selection as defined in IETF RFC 2782 [23].
NOTE 5: The NRF shall notify NFs subscribed to receiving notifications of changes of the NF profile, if the NF recoveryTime or the nfStatus is changed. See clause 6.2 of 3GPP 23.527 [27].
NOTE 6: A requester NF may consider that all the resources created in the NF before the NF recovery time have been lost. This may be used to detect a restart of a NF and to trigger appropriate actions, e.g. release local resources. See clause 6.2 of 3GPP 23.527 [27].
NOTE 7: A NF may register multiple PLMN IDs in its profile within a PLMN comprising multiple PLMN IDs. If so, all the attributes of the NF Profile shall apply to each PLMN ID registered in the plmnList. As an exception, attributes including a PLMN ID, e.g. IMSI-based SUPI ranges, TAIs and GUAMIs, are specific to one PLMN ID and the NF may register in its profile multiple occurrences of such attributes for different PLMN IDs (e.g. the UDM may register in its profile SUPI ranges for different PLMN IDs).
NOTE 8: Other NFs are in a different PLMN if they belong to none of the PLMN ID(s) configured for the PLMN of the NRF.
NOTE 9: This is for the use case where an NF (e.g. AMF) supports multiple PLMNs and the slices supported in each PLMN are different. See clause 9.2.6.2 of 3GPP TS 38.413 [29].
NOTE 10: If notification endpoints are present both in the profile of the NF instance (NFProfile) and in some of its NF Services (NFService) for a same notification type, the notification endpoint(s) of the NF Services shall be used for this notification type.
NOTE 11: The absence of the pcscfInfoList attribute in a P-CSCF profile indicates that the P-CSCF can be selected for any DNN and Access Type.
NOTE 12: The absence of both the smfInfo and smfInfoList attributes in an SMF profile indicates that the SMF can be selected for any S-NSSAI, DNN, TAI and access type.
NOTE 13: The servingScope attribute may indicate geographical areas, It may be used e.g. to discover and select NFs in centralized Data Centers that are expected to serve users located in specific region(s) or province(s). It may also be used to reduce the large configuration of TAIs in the NF instances.
NOTE 14: If an NF includes this information in its profile, this indicates that the services produced by this NF should be accessed preferably via an SCP from an SCP domain to which the NF belongs to.

Editor's note: Whether an NF can register the SCP domains it pertains to in its NF profile (e.g. to simplify SCP profile configuration and discovery, and to allow an SCPc to determine whether NFp is reachable through an SCPp) is pending confirmation from SA2.

6.1.6.2.3 Type: NFService

Table 6.1.6.2.3-1: Definition of type NFService

Attribute name	Data type	P	Cardinality	Description
serviceInstanceId	string	M	1	Unique ID of the service instance within a given NF Instance
serviceName	ServiceName	M	1	Name of the service instance (e.g. "nudm-sdm")
versions	array(NFService Version)	M	1..N	The API versions supported by the NF Service and if available, the corresponding retirement date of the NF Service. The different array elements shall have distinct unique values for "apiVersionInUri", and consequently, the values of "apiFullVersion" shall have a unique first digit version number.
scheme	UriScheme	M	1	URI scheme (e.g. "http", "https")
nfServiceStatus	NFServiceStatus	M	1	Status of the NF Service Instance (NOTE 3)
fqdn	Fqdn	O	0..1	FQDN of the NF Service Instance (NOTE 1) (NOTE 8)
interPlmnFqdn	Fqdn	O	0..1	If the NF service needs to be discoverable by other NFs in a different PLMN, then an FQDN that is used for inter PLMN routing as specified in 3GPP 23.003 [12] may be registered with the NRF (NOTE 1) (NOTE 6). A change of this attribute shall result in triggering a "NF_PROFILE_CHANGED" notification from NRF towards subscribing NFs located in a different PLMN, but the new value shall be notified as a change of the "fqdn" attribute.
ipEndPoints	array(IpEndPoint)	O	1..N	IP address(es) and port information of the Network Function (including IPv4 and/or IPv6 address) where the service is listening for incoming service requests (NOTE 1) (NOTE 7).
apiPrefix	string	O	0..1	Optional path segment(s) used to construct the {apiRoot} variable of the different API URIs, as described in 3GPP 29.501 [5], clause 4.4.1
defaultNotificationSubscriptions	array(DefaultNotificationSubscription)	O	1..N	Notification endpoints for different notification types.
allowedPlmns	array(PlmnId)	O	1..N	PLMNs allowed to access the service instance (NOTE 5). The absence of this attribute indicates that any PLMN is allowed to access the service instance. When included, the allowedPlmns attribute needs not include the PLMN ID(s) registered in the plmnList attribute of the NF Profile, i.e. the PLMN ID(s) registered in the NF Profile shall be considered to be allowed to access the service instance. A change of this attribute shall not trigger a "NF_PROFILE_CHANGED" notification from NRF, and this attribute shall not be included in profile change notifications to subscribed NFs.

allowedSnpns	array(PlmnIdNid)	O	1..N	SNPNs allowed to access the service instance. If this attribute is present in the NFService and in the NF profile, the attribute from the NFService shall prevail. The absence of this attribute in both the NFService and in the NF profile indicates that no SNPN, other than the SNPN(s) registered in the snpnList attribute of the NF Profile, is allowed to access the service instance. When included, the allowedSnpns attribute needs not include the PLMN ID/NID(s) registered in the snpnList attribute of the NF Profile, i.e. the SNPNs registered in the NF Profile shall be considered to be allowed to access the service instance. A change of this attribute shall not trigger a "NF_PROFILE_CHANGED" notification from NRF, and this attribute shall not be included in profile change notifications to subscribed NFs.
allowedNfTypes	array(NFType)	O	1..N	Type of the NFs allowed to access the service instance (NOTE 5). The absence of this attribute indicates that any NF type is allowed to access the service instance. A change of this attribute shall not trigger a "NF_PROFILE_CHANGED" notification from NRF, and this attribute shall not be included in profile change notifications to subscribed NFs.
allowedNfDomains	array(string)	O	1..N	Pattern (regular expression according to the ECMA-262 dialect [8]) representing the NF domain names within the PLMN of the NRF allowed to access the service instance (NOTE 5). The absence of this attribute indicates that any NF domain is allowed to access the service instance. A change of this attribute shall not trigger a "NF_PROFILE_CHANGED" notification from NRF, and this attribute shall not be included in profile change notifications to subscribed NFs.
allowedNssais	array(ExtSnssai)	O	1..N	S-NSSAI of the allowed slices to access the service instance (NOTE 5). The absence of this attribute indicates that any slice is allowed to access the service instance. A change of this attribute shall not trigger a "NF_PROFILE_CHANGED" notification from NRF, and this attribute shall not be included in profile change notifications to subscribed NFs.
allowedOperationsPerNfType	map(array(string))	O	1..N	Map of allowed operations on resources for each type of NF; the key of the map is the NF Type, and the value is an array of scopes. The scopes shall be any of those defined in the API that defines the current service (identified by the "serviceName" attribute). (NOTE 11)

allowedOperationsPerNfInstance	map(array(string))	O	1..N	Map of allowed operations on resources for a given NF Instance; the key of the map is the NF Instance Id, and the value is an array of scopes. The scopes shall be any of those defined in the API that defines the current service (identified by the "serviceName" attribute). (NOTE 11)
priority	integer	O	0..1	Priority (relative to other services of the same type) in the range of 0-65535, to be used for NF Service selection; lower values indicate a higher priority. (NOTE 2). The NRF may overwrite the received priority value when exposing an NFProfile with the Nnrf_NFDiscovery service.
capacity	integer	O	0..1	Static capacity information in the range of 0-65535, expressed as a weight relative to other services of the same type. (NOTE 2).
load	integer	O	0..1	Dynamic load information, ranged from 0 to 100, indicates the current load percentage of the NF Service.
loadTimeStamp	DateTime	O	0..1	It indicates the point in time in which the latest load information (sent by the NF in the "load" attribute of the NF Service) was generated at the NF Service Instance. If the NF did not provide a timestamp, the NRF should set it to the instant when the NRF received the message where the NF provided the latest load information.
recoveryTime	DateTime	O	0..1	Timestamp when the NF service was (re)started (NOTE 3) (NOTE 4)
chfServiceInfo	ChfServiceInfo	O	0..1	Specific data for a CHF service instance
supportedFeatures	SupportedFeatures	O	0..1	Supported Features of the NF Service instance
nfServiceSetIdList	array(NfServiceSetId)	C	1..N	NF Service Set ID (see clause 28.11 of 3GPP TS 23.003 [12]) At most one NF Service Set ID shall be indicated per PLMN of the NF. This information shall be present if available.
sNssais	array(ExtSnssai)	O	1..N	S-NSSAIs of the NF Service. This may be a subset of the S-NSSAIs supported by the NF (see sNssais attribute in NFProfile). When present, this IE shall represent the list of S-NSSAIs supported by the NF Service in all the PLMNs listed in the plmnList IE and it shall prevail over the list of S-NSSAIs supported by the NF instance.
perPlmnSnssaiList	array(PlmnSnssai)	O	1..N	S-NSSAIs of the NF Service per PLMN. This may be a subset of the S-NSSAIs supported per PLMN by the NF (see perPlmnSnssaiList attribute in NFProfile). This IE may be included when the list of S-NSSAIs supported by the NF Service for each PLMN it is supporting is different. When present, this IE shall include the S-NSSAIs supported by the NF Service for each PLMN and it shall prevail over the list of S-NSSAIs supported per PLMN by the NF instance. When present, this IE shall override the sNssais IE. (NOTE 9)
vendorId	VendorId	O	0..1	Vendor ID of the NF Service instance, according to the IANA-assigned "SMI Network Management Private Enterprise Codes" [38].

supportedVendorSpecificFeatures	map(array(VendorSpecificFeature))	O	1..N	Map of Vendor-Specific features, where the key of the map is the IANA-assigned "SMI Network Management Private Enterprise Codes" [38]. The value of each entry of the map shall be a list (array) of VendorSpecificFeature objects. (NOTE 10)
oauth2Required	boolean	O	0..1	It indicates whether the NF Service Instance requires Oauth2-based authorization. Absence of this IE means that the NF Service Producer has not provided any indication about its usage of Oauth2 for authorization.

NOTE 1: The NF Service Consumer will construct the API URIs of the service using:
- for intra-PLMN signalling: the FQDN and IP addresses related attributes present in the NF Service Profile, if any, otherwise the FQDN and IP addresses related attributes present in the NF Profile.
- for inter-PLMN signalling: the interPlmnFqdn present in the NF Service Profile, if any, otherwise the interPlmnFqdn present in the NF Profile.
See Table 6.2.6.2.4-1.

NOTE 2: The capacity and priority parameters, if present, are used for NF selection and load balancing. The priority and capacity attributes shall be used for NF selection in the same way that priority and weight are used for server selection as defined in IETF RFC 2782 [23].

NOTE 3: The NRF shall notify NFs subscribed to receiving notifications of changes of the NF profile, if the recoveryTime or the nfServiceStatus is changed. See clause 6.2 of 3GPP 23.527 [27].

NOTE 4: A requester NF subscribed to NF status changes may consider that all the resources created in the NF service before the NF service recovery time have been lost. This may be used to detect a restart of a NF service and to trigger appropriate actions, e.g. release local resources. See clause 6.2 of 3GPP 23.527 [27].

NOTE 5: If this attribute is present in the NFService and in the NF profile, the attribute from the NFService shall prevail. The absence of this attribute in the NFService and in the NFProfile indicates that there is no corresponding restriction to access the service instance. If this attribute is absent in the NF Service, but it is present in the NF Profile, the attribute from the NF Profile shall be applied.

NOTE 6: Other NFs are in a different PLMN if they belong to none of the PLMN ID(s) configured for the PLMN of the NRF.

NOTE 7: If multiple ipv4 addresses and/or ipv6 addresses are included in the NF Service, the NF Service Consumer of the discovery service shall select one of these addresses randomly, unless operator defined local policy of IP address selection, in order to avoid overload for a specific ipv4 address and/or ipv6 address.

NOTE 8: If the URI scheme registered for the NF service is "https" then FQDN shall be provided in the NF Service profile or in NF Profile (see clause 6.1.6.2.2).

NOTE 9: This is for the use case where an NF (e.g. AMF) supports multiple PLMNs and the slices supported in each PLMN are different. See clause 9.2.6.2 of 3GPP TS 38.413 [29].

NOTE 10: When present, this attribute allows the NF Service Consumer to determine which vendor-specific extensions are supported in a given NF Service Producer in order to include, or not, the vendor-specific attributes (see 3GPP TS 29.500 [4] clause 6.6.3) required for a given feature in subsequent service requests towards a certain service instance of the NF Service Producer.

NOTE 11: These attributes are used in order to determine whether a given resource/operation-level scope shall be granted to an NF Service Consumer that requested an Oauth2 access token with a specific scope; the NRF shall only grant such scope in the access token, if the scope is present in either "allowedOperationsPerNfType", for the specific NF type of the NF Service Consumer, or in "allowedOperationsPerNfInstance", for the specific instance ID of the NF Service Consumer.

6.1.6.2.4 Type: DefaultNotificationSubscription

Table 6.1.6.2.4-1: Definition of type DefaultNotificationSubscription

Attribute name	Data type	P	Cardinality	Description
notificationType	NotificationType	M	1	Type of notification for which the corresponding callback URI is provided.
callbackUri	Uri	M	1	This attribute contains a default notification endpoint to be used by a NF Service Producer towards an NF Service Consumer that has not registered explicitly a callback URI in the NF Service Producer (e.g. as a result of an implicit subscription).
n1MessageClass	N1MessageClass	C	0..1	If the notification type is N1_MESSAGES, this IE shall be present and shall identify the class of N1 messages to be notified.
n2InformationClass	N2InformationClass	C	0..1	If the notification type is N2_INFORMATION, this IE shall be present and shall identify the class of N2 information to be notified.
versions	array(string)	O	1..N	API versions (e.g. "v1") supported for the default notification type.

6.1.6.2.5 Type: IpEndPoint

Table 6.1.6.2.5-1: Definition of type IpEndPoint

Attribute name	Data type	P	Cardinality	Description
ipv4Address	Ipv4Addr	C	0..1	IPv4 address (NOTE 1)
ipv6Address	Ipv6Addr	C	0..1	IPv6 address (NOTE 1)
transport	TransportProtocol	O	0..1	Transport protocol
port	integer	O	0..1	Port number (NOTE 2) Minimum: 0 Maximum: 65535
NOTE 1: At most one occurrence of either ipv4Address or ipv6Address shall be included in this data structure. NOTE 2: If the port number is absent from the ipEndPoints attribute, the NF service consumer shall use the default HTTP port number, i.e. TCP port 80 for "http" URIs or TCP port 443 for "https" URIs as specified in IETF RFC 7540 [9] when invoking the service.				

6.1.6.2.6 Type: UdrInfo

Table 6.1.6.2.6-1: Definition of type UdrInfo

Attribute name	Data type	P	Cardinality	Description
groupId	NfGroupId	O	0..1	Identity of the UDR group that is served by the UDR instance. If not provided, the UDR instance does not pertain to any UDR group.
supiRanges	array(SupiRange)	O	1..N	List of ranges of SUPI's whose profile data is available in the UDR instance (NOTE 1)
gpsiRanges	array(IdentityRange)	O	1..N	List of ranges of GPSIs whose profile data is available in the UDR instance (NOTE 1)
externalGroupIdentifiersRanges	array(IdentityRange)	O	1..N	List of ranges of external groups whose profile data is available in the UDR instance (NOTE 1)
supportedDataSets	array(DataSetId)	O	1..N	List of supported data sets in the UDR instance. If not provided, the UDR supports all data sets.
NOTE 1: If none of these parameters is provided, the UDR can serve any external group and any SUPI or GPSI.				

6.1.6.2.7 Type: UdmInfo

Table 6.1.6.2.7-1: Definition of type UdmInfo

Attribute name	Data type	P	Cardinality	Description
groupId	NfGroupId	O	0..1	Identity of the UDM group that is served by the UDM instance. If not provided, the UDM instance does not pertain to any UDM group.
supiRanges	array(SupiRange)	O	1..N	List of ranges of SUPIs whose profile data is available in the UDM instance (NOTE 1)
gpsiRanges	array(IdentityRange)	O	1..N	List of ranges of GPSIs whose profile data is available in the UDM instance (NOTE 1)
externalGroupIdentifiersRanges	array(IdentityRange)	O	1..N	List of ranges of external groups whose profile data is available in the UDM instance (NOTE 1)
routingIndicators	array(string)	O	1..N	List of Routing Indicator information that allows to route network signalling with SUCI (see 3GPP 23.003 [12]) to the UDM instance. If not provided, the UDM can serve any Routing Indicator. Pattern: '^[0-9]{1,4}$'
internalGroupIdentifiersRanges	array(InternalGroupIdRange)	O	1..N	List of ranges of Internal Group Identifiers whose profile data is available in the UDM instance. If not provided, it does not imply that the UDM supports all internal groups.
NOTE 1: If none of these parameters is provided, the UDM can serve any external group and any SUPI or GPSI.				

6.1.6.2.8 Type: AusfInfo

Table 6.1.6.2.8-1: Definition of type AusfInfo

Attribute name	Data type	P	Cardinality	Description
groupId	NfGroupId	O	0..1	Identity of the AUSF group. If not provided, the AUSF instance does not pertain to any AUSF group.
supiRanges	array(SupiRange)	O	1..N	List of ranges of SUPIs that can be served by the AUSF instance. If not provided, the AUSF can serve any SUPI.
routingIndicators	array(string)	O	1..N	List of Routing Indicator information that allows to route network signalling with SUCI (see 3GPP 23.003 [12]) to the AUSF instance. If not provided, the AUSF can serve any Routing Indicator. Pattern: '^[0-9]{1,4}$'

6.1.6.2.9 Type: SupiRange

Table 6.1.6.2.9-1: Definition of type SupiRange

Attribute name	Data type	P	Cardinality	Description
start	string	O	0..1	First value identifying the start of a SUPI range, to be used when the range of SUPI's can be represented as a numeric range (e.g., IMSI ranges). This string shall consist only of digits. Pattern: "^[0-9]+$"
end	string	O	0..1	Last value identifying the end of a SUPI range, to be used when the range of SUPI's can be represented as a numeric range (e.g. IMSI ranges). This string shall consist only of digits. Pattern: "^[0-9]+$"
pattern	string	O	0..1	Pattern (regular expression according to the ECMA-262 dialect [8]) representing the set of SUPI's belonging to this range. A SUPI value is considered part of the range if and only if the SUPI string fully matches the regular expression.
NOTE: Either the start and end attributes, or the pattern attribute, shall be present.				

EXAMPLE 1: IMSI range. From: 123 45 6789040000 To: 123 45 6789059999 (i.e., 20,000 IMSI numbers)
JSON: { "start": "123456789040000", "end": "123456789059999" }

EXAMPLE 2: IMSI range. From: 123 45 6789040000 To: 123 45 6789049999 (i.e., 10,000 IMSI numbers)
JSON: { "pattern": "^imsi-12345678904[0-9]{4}$" }, or
JSON: { "start": "123456789040000", "end": "123456789049999" }

EXAMPLE 3: NAI range. "smartmeter-{factoryID}@company.com" where "*{factoryID}*" can be any string.
JSON: { "pattern": "^nai-smartmeter-.+@company\.com$" }

6.1.6.2.10 Type: IdentityRange

Table 6.1.6.2.10-1: Definition of type IdentityRange

Attribute name	Data type	P	Cardinality	Description
start	string	O	0..1	First value identifying the start of an identity range, to be used when the range of identities can be represented as a numeric range (e.g., MSISDN ranges). This string shall consist only of digits. Pattern: "^[0-9]+$"
end	string	O	0..1	Last value identifying the end of an identity range, to be used when the range of identities can be represented as a numeric range (e.g. MSISDN ranges). This string shall consist only of digits. Pattern: "^[0-9]+$"
pattern	string	O	0..1	Pattern (regular expression according to the ECMA-262 dialect [8]) representing the set of identities belonging to this range. An identity value is considered part of the range if and only if the identity string fully matches the regular expression. To be used when identity is External Identifier or External Group Identifier or MSISDN.
NOTE: Either the start and end attributes, or the pattern attribute, shall be present.				

6.1.6.2.11 Type: AmfInfo

Table 6.1.6.2.11-1: Definition of type AmfInfo

Attribute name	Data type	P	Cardinality	Description
amfRegionId	AmfRegionId	M	1	AMF region identifier
amfSetId	AmfSetId	M	1	AMF set identifier.
guamiList	array(Guami)	M	1..N	List of supported GUAMIs
taiList	array(Tai)	O	1..N	The list of TAIs the AMF can serve. It may contain the non-3GPP access TAI. The absence of this attribute and the taiRangeList attribute indicate that the AMF can be selected for any TAI in the serving network.
taiRangeList	array(TaiRange)	O	1..N	The range of TAIs the AMF can serve. The absence of this attribute and the taiList attribute indicate that the AMF can be selected for any TAI in the serving network.
backupInfoAmfFailure	array(Guami)	O	1..N	List of GUAMIs for which the AMF acts as a backup for AMF failure
backupInfoAmfRemoval	array(Guami)	O	1..N	List of GUAMIs for which the AMF acts as a backup for planned AMF removal
n2InterfaceAmfInfo	N2InterfaceAmfInfo	O	0..1	N2 interface information of the AMF. This information needs not be sent in NF Discovery responses. It may be used by the NRF to update the DNS for AMF discovery by the 5G Access Network. The procedures for updating the DNS are out of scope of this specification.

The "backupInfoAmfFailure" attribute and "backupInfoAmfRemoval" attribute indicates the GUAMIs for which the AMF can act as Backup, when the serving AMF has failed or under planned removal.

EXAMPLE:

When AMF-A, AMF-B and AMF-C registered their NF profiles for PLMN (e.g. MCC = 234, MNC = 15) as following:

AMF-A NF Profile:

```
{
  "amfInfo": {
    "guamiList": [{ "plmnId": { "mcc": "234", "mnc": "15" }, "amfId": "000001"}],
    "backupInfoAmfFailure": [{ "plmnId": { "mcc": "234", "mnc": "15" }, "amfId": "000003"}]
  }
}
```

AMF-B NF Profile:

```
{
  "amfInfo": {
    "guamiList": [{ "plmnId": { "mcc": "234", "mnc": "15" }, "amfId": "000002"}],
    "backupInfoAmfRemoval": [{ "plmnId": { "mcc": "234", "mnc": "15" }, "amfId": "000003"}]
  }
}
```

AMF-C NF Profile:

```
{
  "amfInfo": {
    "guamiList": [{ "plmnId": { "mcc": "234", "mnc": "15" }, "amfId": "000003"}]
  }
}
```

When one NF consumer queries NRF with a GUAMI served by AMF-C (i.e. `{"plmnId":{"mcc":"234","mnc":"15"},"amfId":"000003"}`), then

- if the NRF detects the AMF-C has failed, e.g. using heartbeat, the NRF shall return AMF-A instance as backup AMF; or

- if the NRF detects AMF-C has entered planned removal, i.e. received a de-registration request from AMF-C, the NRF shall return AMF-B instance as backup AMF.

6.1.6.2.12 Type: SmfInfo

Table 6.1.6.2.12-1: Definition of type SmfInfo

Attribute name	Data type	P	Cardinality	Description
sNssaiSmfInfoList	array(SnssaiSmfInfoItem)	M	1..N	List of parameters supported by the SMF per S-NSSAI (NOTE 1).
taiList	array(Tai)	O	1..N	The list of TAIs the SMF can serve. It may contain the non-3GPP access TAI. The absence of this attribute and the taiRangeList attribute indicate that the SMF can be selected for any TAI in the serving network.
taiRangeList	array(TaiRange)	O	1..N	The range of TAIs the SMF can serve. It may contain the non-3GPP access' TAI. The absence of this attribute and the taiList attribute indicate that the SMF can be selected for any TAI in the serving network.
pgwFqdn	Fqdn	O	0..1	The FQDN of the PGW if the SMF is a combined SMF/PGW-C.
accessType	array(AccessType)	C	1..2	If included, this IE shall contain the access type (3GPP_ACCESS and/or NON_3GPP_ACCESS) supported by the SMF. If not included, it shall be assumed the both access types are supported.
priority	integer	O	0..1	Priority (relative to other NFs of the same type) in the range of 0-65535, to be used for NF selection for a service request matching the attributes of the SmfInfo; lower values indicate a higher priority. See the precedence rules in the description of the priority attribute in NFProfile, if Priority is also present in the nfServiceList parameters or in NFProfile. The NRF may overwrite the received priority value when exposing an NFProfile with the Nnrf_NFDiscovery service. (NOTE 2)
NOTE 1: If this S-NSSAIs is present in the SmfInfo and in the NFprofile, the S-NSSAIs from the SmfInfo shall prevail. NOTE 2: An SMF profile may e.g. contain multiple SmfInfo entries, with each entry containing a different list of TAIs and a different priority, to differentiate the priority to select the SMF based on the user location. The priority in SmfInfo has the least precedence, i.e. it applies between SMFs or SMF Services with the same priority.				

6.1.6.2.13 Type: UpfInfo

Table 6.1.6.2.13-1: Definition of type UpfInfo

Attribute name	Data type	P	Cardinality	Description
sNssaiUpfInfoList	array(SnssaiUpfInfoItem)	M	1..N	List of parameters supported by the UPF per S-NSSAI (NOTE 1)
smfServingArea	array(string)	O	1..N	The SMF service area(s) the UPF can serve. If not provided, the UPF can serve any SMF service area.
interfaceUpfInfoList	array(InterfaceUpfInfoItem)	O	1..N	List of User Plane interfaces configured on the UPF. When this IE is provided in the NF Discovery response, the NF Service Consumer (e.g. SMF) may use this information for UPF selection.
iwkEpsInd	boolean	O	0..1	Indicates whether interworking with EPS is supported by the UPF. true: Supported false (default): Not Supported
pduSessionTypes	array(PduSessionType)	O	1..N	List of PDU session type(s) supported by the UPF. The absence of this attribute indicates that the UPF can be selected for any PDU session type.
atsssCapability	AtsssCapability	C	0..1	If present, this IE shall indicate the ATSSS capability of the UPF. If not present, the UPF shall be regarded with no ATSSS capability.
ueIpAddrInd	boolean	O	0..1	Indicates whether the UPF supports allocating UE IP addresses/prefixes. true: supported false (default): not supported
taiList	array(Tai)	O	1..N	The list of TAIs the UPF can serve. It may contain the non-3GPP access TAI. If not provided, the UPF can serve the whole SMF service area defined by the smfServingArea attribute.
wAgfInfo	WAgfInfo	C	0..1	If present, this IE shall indicate that the UPF is collocated with W-AGF. If not present, the UPF is not collocated with W-AGF.
tngfInfo	TngfInfo	C	0..1	If present, this IE shall indicate that the UPF is collocated with TNGF. If not present, the UPF is not collocated with TNGF.
twifInfo	TwifInfo	C	0..1	If present, this IE shall indicate that the UPF is collocated with TWIF. If not present, the UPF is not collocated with TWIF.
priority	integer	O	0..1	Priority (relative to other NFs of the same type) in the range of 0-65535, to be used for NF selection for a service request matching the attributes of the UpfInfo; lower values indicate a higher priority. See the precedence rules in the description of the priority attribute in NFProfile, if Priority is also present in NFProfile. The NRF may overwrite the received priority value when exposing an NFProfile with the Nnrf_NFDiscovery service. (NOTE 2)
redundantGtpu	boolean	O	0..1	Indicates whether the UPF supports redundant GTP-U path. true: supported false (default): not supported
ipups	boolean	O	0..1	Indicates whether the UPF is configured for IPUPS. (NOTE 3) true: the UPF is configured for IPUPS. false (default): the UPF is not configured for IPUPS.

dataForwarding	boolean	O	0..1	Indicates whether the UPF is configured for data forwarding. (NOTE 4) When present, this IE shall be set as following: - true: the UPF is configured for data forwarding - false (default): the UPF is not configured for data forwarding If the UPF is configured for data forwarding, it shall support UP network interface with type "DATA_FORWARDING".
NOTE 1: If this S-NSSAIs is present in the UpfInfo and in the NFprofile, the S-NSSAIs from the UpfInfo shall prevail. NOTE 2: An UPF profile may e.g. contain multiple UpfInfo entries, with each entry containing a different list of TAIs and a different priority, to differentiate the priority to select the UPF based on the user location. The priority in UpfInfo has the least precedence, i.e. it applies between UPFs with the same priority. NOTE 3: Any UPF can support the IPUPS functionality. In network deployments where specific UPFs are used to provide IPUPS, UPFs configured for providing IPUPS services shall be selected to provide IPUPS. NOTE 4: Based on operator policies, if dedicated UPFs are preferred to be used for indirect data forwarding during handover scenarios, when setting up the indirect data forwarding tunnel, the SMF should preferably select a UPF configured for data forwarding and use the network instance indicated in the Network Instance ID associated to the DATA_FORWARDING interface type in the interfaceUpfInfoList attribute.				

6.1.6.2.14 Type: SnssaiUpfInfoItem

Table 6.1.6.2.14-1: Definition of type SnssaiUpfInfoItem

Attribute name	Data type	P	Cardinality	Description
sNssai	Snssai	M	1	Supported S-NSSAI
dnnUpfInfoList	array(DnnUpfInfoItem)	M	1..N	List of parameters supported by the UPF per DNN
redundantTransport	boolean	O	0..1	Indicates whether the UPF supports redundant transport path on the transport layer in the corresponding network slice. true: supported false (default): not supported

6.1.6.2.15 Type: DnnUpfInfoItem

Table 6.1.6.2.15-1: Definition of type DnnUpfInfoItem

Attribute name	Data type	P	Cardinality	Description
dnn	Dnn	M	1	Supported DNN. The DNN shall contain the Network Identifier and it may additionally contain an Operator Identifier. If the Operator Identifier is not included, the DNN is supported for all the PLMNs in the plmnList of the NF Profile.
dnaiList	array(Dnai)	O	1..N	List of Data network access identifiers supported by the UPF for this DNN. The absence of this attribute indicates that the UPF can be selected for this DNN for any DNAI.
pduSessionTypes	array(PduSessionType)	O	1..N	List of PDU session type(s) supported by the UPF for a specific DNN. The absence of this attribute indicates that the UPF can be selected for this DNN for any PDU session type supported by the UPF (see clause 6.1.6.2.13).
ipv4AddressRanges	array(Ipv4AddressRange)	O	1..N	List of ranges of IPv4 addresses handled by UPF. (NOTE 1)
ipv6PrefixRanges	array(Ipv6PrefixRange)	O	1..N	List of ranges of IPv6 prefixes handled by the UPF. (NOTE 1)
NOTE 1: The list of ranges of IPv4/v6 address may be used by the SMF to select a UPF which supports a UE static IP address received in user subscription.				

6.1.6.2.16 Type: SubscriptionData

Table 6.1.6.2.16-1: Definition of type SubscriptionData

Attribute name	Data type	P	Cardinality	Description
nfStatusNotificationUri	Uri	M	1	Callback URI where the NF Service Consumer will receive the notifications from NRF.
reqNfInstanceId	NfInstanceId	O	0..1	If present, this IE shall contain the NF instance id of the NF service consumer.
subscrCond	SubscrCond	O	0..1	If present, this attributed shall contain the conditions identifying the set of NF Instances whose status is requested to be monitored. If this attribute is not present, it means that the NF Service Consumer requests a subscription to all NFs in the NRF (NOTE 1).
subscriptionId	string	C	0..1	Subscription ID for the newly created resource. This parameter shall be absent in the request to the NRF and shall be included by NRF in the response to the subscription creation request. Read-Only: true Pattern: "^([0-9]{5,6}-)?[^-]+$"
validityTime	DateTime	C	0..1	Time instant after which the subscription becomes invalid. This parameter may be sent by the client, as a hint to the server, but it shall be always sent back by the server (regardless of the presence of the attribute in the request) in the response to the subscription creation request.
reqNotifEvents	array(NotificationEventType)	O	1..N	If present, this attribute shall contain the list of event types that the NF Service Consumer is interested in receiving. If this attribute is not present, it means that notifications for all event types are requested.
reqNfType	NFType	O	0..1	If included, this IE shall contain the NF type of the NF Service Consumer that is requesting the creation of the subscription. The NRF shall use it for authorizing the request, in the same way as the "requester-nf-type" is used in the NF Discovery service (see Table 6.2.3.2.3.1-1). When the subscription is for a set of NF Instances, the subscription may be accepted by NRF, but it shall only generate notifications from NF Instances whose authorization parameters allow the NF Service Consumer to access their services (NOTE 2).
reqNfFqdn	Fqdn	O	0..1	This IE may be present for a subscription request within the same PLMN as the NRF. If included, this IE shall contain the FQDN of the NF Service Consumer that is requesting the creation of the subscription. The NRF shall use it for authorizing the request, in the same way as the "requester-nf-instance-fqdn" is used in the NF Discovery service (see Table 6.2.3.2.3.1-1). This IE shall be ignored by the NRF if it is received from a requester NF belonging to a different PLMN. When the subscription is for a set of NF Instances, the subscription may be accepted by NRF, but it shall only generate notifications from NF Instances whose authorization parameters allow the NF Service Consumer to access their services (NOTE 2).

reqSnssais	array(Snssai)	O	0..1	If included, this IE shall contain the list of S-NSSAIs of the NF Service Consumer that is requesting the creation of the subscription. If this IE is included in a subscription request in a different PLMN, the requester NF shall provide S-NSSAI values of the target PLMN, that correspond to the S-NSSAI values of the requester NF. The NRF shall use it for authorizing the request, in the same way as the "requester-snssais" is used in the NF Discovery service (see Table 6.2.3.2.3.1-1). When the subscription is for a set of NF Instances, the subscription may be accepted by NRF, but it shall only generate notifications from NF Instances whose authorization parameters allow the NF Service Consumer to access their services (NOTE 2).
plmnId	PlmnId	O	0..1	If present, this attribute contains the target PLMN ID of the NF Instance(s) whose status is requested to be monitored.
nid	Nid	O	0..1	If present, this attribute contains the target NID that, together with the plmnId attribute, identifies the SNPN of the NF Instance(s) whose status is requested to be monitored.
notifCondition	NotifCondition	O	0..1	If present, this attribute contains the conditions that trigger a notification from NRF; this attribute shall only be present if the NF Service Consumer has subscribed to changes on the NF Profile (i.e., reqNotifEvents contains the value "NF_PROFILE_CHANGED", or reqNotifEvents attribute is absent) (NOTE 3). If this attribute is absent, it means that the NF Service Consumer does not indicate any restriction, or condition, on which attributes of the NF Profile shall trigger a notification from NRF. (NOTE 5).
reqPlmnList	array(PlmnId)	C	1..N	This IE shall be included when subscribing to NF services in a different PLMN. When included, this IE shall contain the PLMN ID(s) of the requester NF. (NOTE 2)
reqSnpnList	array(PlmnIdNid)	C	1..N	This IE shall be included when the subscribing NF belongs to one or several SNPNs and it subscribes to NF services of a specific SNPN. When included, this IE shall contain the SNPN ID(s) of the requester NF. When the subscription is for a set of NF Instances, the subscription may be accepted by NRF, but it shall only generate notifications from NF Instances whose authorization parameters allow the NF Service Consumer to access their services. (NOTE 2)
servingScope	array(string)	O	1..N	If present, this attribute indicates the target served area(s) of the NF instance(s) whose status is required to be monitored. (NOTE 4)
requesterFeatures	SupportedFeatures	C	0..1	Nnrf_NFManagement features supported by the NF Service Consumer that is invoking the Nnrf_NFManagement service. See clause 6.1.9. This IE shall be included if at least one feature is supported by the NF Service Consumer. Write-Only: true (NOTE 6)

nrfSupportedFeatures	SupportedFeatures	C	0..1	Features supported by the NRF in the Nnrf_NFManagement service. See clause 6.1.9. This IE shall be included if at least one feature is supported by the NRF. Read-Only: true

NOTE 1: The "subscription to all NFs" may be quite demanding in terms of resources in NRF and also in terms of network traffic of the resulting notifications, so it should be authorized by NRF under very strict policies (e.g. only to a specific requesting NF, as indicated by reqNfType and reqNfFqdn attributes).

NOTE 2: The authorization parameters in NF Profile are those used by NRF to determine whether a given NF Instance / NF Service Instance can be discovered by an NF Service Consumer in order to consume its offered services (e.g. "allowedNfTypes", "allowedNfDomains", etc.). Based on operator's policies, a subscription request not including the requester's information necessary to validate the authorization parameters in NF Profiles may be rejected or may be accepted but with only generating notifications from NF Instances whose authorization parameters allow any NF Service Consumer to access their services.

NOTE 3: The subscription to load changes may be quite demanding in terms of network traffic of the resulting notifications, thus it may be limited by the NRF via appropriate configuration (e.g. granularity threshold)

NOTE 4: An NF instance may explicitly indicate the served areas in the NF profile when registered to NRF. When this IE is present, the NRF shall only monitor the NF instance(s) indicating at least one of the served areas in the list. If an NF instance has not indicated any served area in its NF profile, it shall not be monitored.

NOTE 5: If the attributes to be monitored or excluded from monitoring, included as part of the "notifCondition" attribute, refer to a specific element of an array (e.g. they refer to a specifc array index of the "nfServices" attribute of the NFProfile), the NRF shall apply the same condition to all elements of the same array.

NOTE 6: If the NF Service Consumer that issued the subscription request indicated support for the "Service-Map" feature, the NRF shall send notifications of profile changes (see clause 6.1.6.2.17) affecting the list of NF Service Instances, as modifications of specific attributes of the "nfServiceList" map. Otherwise, the NRF shall send those notifications as a complete replacement of the "nfServices" array attribute.

6.1.6.2.17 Type: NotificationData

Table 6.1.6.2.17-1: Definition of type NotificationData

Attribute name	Data type	P	Cardinality	Description
event	NotificationEventType	M	1	Notification type. It shall take the values "NF_REGISTERED", "NF_DEREGISTERED" or "NF_PROFILE_CHANGED".
nfInstanceUri	Uri	M	1	Uri of the NF Instance (see clause 6.1.3.3.2) associated to the notification event.
nfProfile	NFProfile	C	0..1	New NF Profile or Updated NF Profile; it shall be present when the notification type is "NF_REGISTERED" and it may be present when the notification type is "NF_PROFILE_CHANGED". (NOTE 3)
profileChanges	array(ChangeItem)	C	1..N	List of changes on the profile of the NF Instance associated to the notification event; it may be present when the notification type is "NF_PROFILE_CHANGED" (see NOTE 1, NOTE 2).
conditionEvent	ConditionEventType	C	0..1	Type of event indicating wether a change of NF Profile results in that the NF Instance starts or stops being part of a given set of NF Instances, as indicated in the subscription condition (see attribute "subscrCond" in clause 6.1.6.2.16). It can take the value "NF_ADDED" (if the NF Instance starts being part of a given set) or "NF_REMOVED" (if the NF Instance stops being part of a given set). (NOTE 3)

NOTE 1: If "event" attribute takes the value "NF_PROFILE_CHANGED", then either "nfProfile" or "profileChanges" attributes shall be present, but not both.
NOTE 2: The NRF shall notify about NF Profile changes affecting attributes of type "array" only as a complete replacement of the whole array (i.e. it shall not notify about changes of individual array elements).
NOTE 3: When a change in an NF Profile results in an NF to start being part of a given set, the NRF shall indicate such condition by including the "conditionEvent" attribute with value "NF_ADDED", and it shall include in the notification the "nfProfile" attribute with the full NF Profile of the NF Instance; the "profileChanges" attribute shall not be included.
When a change in an NFProfile results in an NF to stop being part of a given set, the NRF shall indicate such condition by including the "conditionEvent" attribute with value "NF_REMOVED", and both attributes "nfProfile" and "profileChanges" shall be absent.

EXAMPLE: Notification payload sent from NRF when an NF Instance has changed its profile by updating the value of the "recoveryTime" attribute of its NF Profile, and updated any attribute of any of its NF Service Instances:

```
{
  "event": "NF_PROFILE_CHANGED",
  "nfInstanceUri": ".../nf-instances/4947a69a-f61b-4bc1-b9da-47c9c5d14b64",
  "profileChanges": [
    {
      "op": "REPLACE",
      "path": "/recoveryTime",
      "newValue": "2018-12-30T23:20:50Z"
    },
    {
      "op": "REPLACE",
      "path": "/nfServices",
      "newValue": [ ...new array content... ]
    }
  ]
}
```

6.1.6.2.18 Void

6.1.6.2.19 Type: NFServiceVersion

Table 6.1.6.2.19-1: Definition of type NFServiceVersion

Attribute name	Data type	P	Cardinality	Description
apiVersionInUri	string	M	1	Version of the service instance to be used in the URI for accessing the API (e.g. "v1").
apiFullVersion	string	M	1	Full version number of the API as specified in clause 4.3.1 of 3GPP 29.501 [5].
expiry	DateTime	O	0..1	Expiry date and time of the NF service. This represents the planned retirement date as specified in clause 4.3.1.5 of 3GPP 29.501 [5].

6.1.6.2.20 Type: PcfInfo

Table 6.1.6.2.20-1: Definition of type PcfInfo

Attribute name	Data type	P	Cardinality	Description
groupId	NfGroupId	O	0..1	Identity of the PCF group that is served by the PCF instance. If not provided, the PCF instance does not pertain to any PCF group.
dnnList	array(Dnn)	O	1..N	DNNs supported by the PCF. The DNN shall contain the Network Identifier and it may additionally contain an Operator Identifier. If the Operator Identifier is not included, the DNN is supported for all the PLMNs in the plmnList of the NF Profile. If not provided, the PCF can serve any DNN.
supiRanges	array(SupiRange)	O	1..N	List of ranges of SUPIs that can be served by the PCF instance. If not provided, the PCF can serve any SUPI.
gpsiRanges	array(IdentityRange)	O	1..N	List of ranges of GPSIs that can be served by the PCF instance. If not provided, the PCF can serve any GPSI.
rxDiamHost	DiameterIdentity	C	0..1	This IE shall be present if the PCF supports Rx interface. When present, this IE shall indicate the Diameter host of the Rx interface for the PCF.
rxDiamRealm	DiameterIdentity	C	0..1	This IE shall be present if the PCF supports Rx interface. When present, this IE shall indicate the Diameter realm of the Rx interface for the PCF.
v2xSupportInd	boolean	O	0..1	Indicates whether V2X Policy/Parameter provisioning is supported by the PCF. true: Supported false (default): Not Supported

6.1.6.2.21 Type: BsfInfo

Table 6.1.6.2.21-1: Definition of type BsfInfo

Attribute name	Data type	P	Cardinality	Description
ipv4AddressRanges	array(Ipv4AddressRange)	O	1..N	List of ranges of IPv4 addresses handled by BSF. If not provided, the BSF can serve any IPv4 address.
dnnList	array(Dnn)	O	1..N	List of DNNs handled by the BSF. The DNN shall contain the Network Identifier and it may additionally contain an Operator Identifier. If the Operator Identifier is not included, the DNN is supported for all the PLMNs in the plmnList of the NF Profile. If not provided, the BSF can serve any DNN.
ipDomainList	array(string)	O	1..N	List of IPv4 address domains, as described in clause 6.2 of 3GPP 29.513 [28], handled by the BSF. If not provided, the BSF can serve any IP domain.
ipv6PrefixRanges	array(Ipv6PrefixRange)	O	1..N	List of ranges of IPv6 prefixes handled by the BSF. If not provided, the BSF can serve any IPv6 prefix.

6.1.6.2.22 Type: Ipv4AddressRange

Table 6.1.6.2.22-1: Definition of type IPv4AddressRange

Attribute name	Data type	P	Cardinality	Description
start	Ipv4Addr	M	1	First value identifying the start of an IPv4 address range
end	Ipv4Addr	M	1	Last value identifying the end of an IPv4 address range

6.1.6.2.23 Type: Ipv6PrefixRange

Table 6.1.6.2.23-1: Definition of type IPv6PrefixRange

Attribute name	Data type	P	Cardinality	Description
start	Ipv6Prefix	M	1	First value identifying the start of an IPv6 prefix range
end	Ipv6Prefix	M	1	Last value identifying the end of an IPv6 prefix range

6.1.6.2.24 Type: InterfaceUpfInfoItem

Table 6.1.6.2.24-1: Definition of type InterfaceUpfInfoItem

Attribute name	Data type	P	Cardinality	Description
interfaceType	UPInterfaceType	M	1	User Plane interface type
ipv4EndpointAddresses	array(Ipv4Addr)	C	1..N	Available endpoint IPv4 address(es) of the User Plane interface (NOTE 1) (NOTE 2)
ipv6EndpointAddresses	array(Ipv6Addr)	C	1..N	Available endpoint IPv6 address(es) of the User Plane interface (NOTE 1) (NOTE 2)
endpointFqdn	Fqdn	C	0..1	FQDN of available endpoint of the User Plane interface (NOTE 1) (NOTE 2)
networkInstance	string	O	0..1	Network Instance (See 3GPP 29.244 [21]) associated to the User Plane interface
NOTE 1: At least one of the addressing parameters (ipv4address, ipv6adress or endpointFqdn) shall be included in the InterfaceUpfInfoItem. NOTE 2: When interfaceType is "DATA_FORWARDING", the SMF shall ignore these IEs. The UPF shall register a dummy FQDN or IP address for interfaceType "DATA_FORWARDING" (for backward compatibility reason).				

6.1.6.2.25 Type: UriList

Table 6.1.6.2.25-1: Definition of type UriList

Attribute name	Data type	P	Cardinality	Description
_links	map(LinksValueSchema)	O	1..N	See clause 4.9.4 of 3GPP TS 29.501 [5] for the description of the members.

6.1.6.2.26 Type: N2InterfaceAmfInfo

Table 6.1.6.2.26-1: Definition of type N2InterfaceAmfInfo

Attribute name	Data type	P	Cardinality	Description
ipv4EndpointAddress	array(Ipv4Addr)	C	1..N	Available AMF endpoint IPv4 address(es) for N2 (see NOTE 1)
ipv6EndpointAddress	array(Ipv6Addr)	C	1..N	Available AMF endpoint IPv6 address(es) for N2 (see NOTE 1)
amfName	AmfName	O	0..1	AMF Name
NOTE 1: At least one of the addressing parameters (ipv4address or ipv6adress) shall be included.				

6.1.6.2.27 Type: TaiRange

Table 6.1.6.2.27-1: Definition of type TaiRange

Attribute name	Data type	P	Cardinality	Description
plmnId	PlmnId	M	1	PLMN ID related to the TacRange.
tacRangeList	array(TacRange)	M	1..N	The range of the TACs
nid	Nid	O	0..1	NID related to the TacRange, for an SNPN

6.1.6.2.28 Type: TacRange

Table 6.1.6.2.28-1: Definition of type TacRange

Attribute name	Data type	P	Cardinality	Description
start	string	O	0..1	First value identifying the start of a TAC range, to be used when the range of TAC's can be represented as a hexadecimal range (e.g., TAC ranges). 3-octet string identifying a tracking area code, each character in the string shall take a value of "0" to "9" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the TAC shall appear first in the string, and the character representing the 4 least significant bit of the TAC shall appear last in the string. Pattern: "^([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6})$"
end	string	O	0..1	Last value identifying the end of a TAC range, to be used when the range of TAC's can be represented as a hexadecimal range (e.g. TAC ranges). 3-octet string identifying a tracking area code, each character in the string shall take a value of "0" to "9" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the TAC shall appear first in the string, and the character representing the 4 least significant bit of the TAC shall appear last in the string. Pattern: "^([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6})$"
pattern	string	O	0..1	Pattern (regular expression according to the ECMA-262 dialect [8]) representing the set of TAC's belonging to this range. A TAC value is considered part of the range if and only if the TAC string fully matches the regular expression.
NOTE: Either the start and end attributes, or the pattern attribute, shall be present.				

EXAMPLE 1: TAC range. From: 543000 To: 5433E7 (i.e., 1000 TAC numbers)
JSON: { "start": "543000", "end": "5433E7" }

EXAMPLE 2: TAC range. From: 54E000 To: 54EFFF (i.e., 4096 TAC numbers)
JSON: { "pattern": "^54E[0-9a-fA-F]{3}$" }, or
JSON: { "start": "54E000", "end": "54EFFF" }

6.1.6.2.29 Type: SnssaiSmfInfoItem

Table 6.1.6.2.29-1: Definition of type SnssaiSmfInfoItem

Attribute name	Data type	P	Cardinality	Description
sNssai	Snssai	M	1	Supported S-NSSAI
dnnSmfInfoList	array(DnnSmfInfoItem)	M	1..N	List of parameters supported by the SMF per DNN

6.1.6.2.30 Type: DnnSmfInfoItem

Table 6.1.6.2.30-1: Definition of type DnnSmfInfoItem

Attribute name	Data type	P	Cardinality	Description
dnn	Dnn	M	1	Supported DNN (NOTE). The DNN shall contain the Network Identifier and it may additionally contain an Operator Identifier. If the Operator Identifier is not included, the DNN is supported for all the PLMNs in the plmnList of the NF Profile.
NOTE: For a SMF which only supports the I-SMF related functionalities, the dnn attribute may be an invalid DNN according to operator's local policy.				

6.1.6.2.31 Type: NrfInfo

Table 6.1.6.2.31-1: Definition of type NrfInfo

Attribute name	Data type	P	Cardinality	Description
servedUdrInfo	map(UdrInfo)	O	1..N	This attribute contains all the udrInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the udrInfo belongs to.
servedUdrInfoList	map(map(UdrInfo))	O	1..N	This attribute contains the udrInfoList attribute locally configured in the NRF or that the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedUdmInfo	map(UdmInfo)	O	1..N	This attribute contains all the udmInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the udmInfo belongs to.
servedUdmInfoList	map(map(UdmInfo))	O	1..N	This attribute contains the udmInfoList attribute locally configured in the NRF or that the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedAusfInfo	map(AusfInfo)	O	1..N	This attribute contains all the ausfInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the ausfInfo belongs to.
servedAusfInfoList	map(map(AusfInfo))	O	1..N	This attribute contains the ausfInfoList attribute locally configured in the NRF or that the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedAmfInfo	map(AmfInfo)	O	1..N	This attribute contains all the amfInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the amfInfo belongs to.
servedAmfInfoList	map(map(AmfInfo))	O	1..N	This attribute contains the amfInfoList attribute locally configured in the NRF or that the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedSmfInfo	map(SmfInfo)	O	1..N	This attribute contains all the smfInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the smfInfo belongs to.
servedSmfInfoList	map(map(SmfInfo))	O	1..N	This attribute contains the smfInfoList attribute locally configured in the NRF or that the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedUpfInfo	map(UpfInfo)	O	1..N	This attribute contains all the upfInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the upfInfo belongs to.
servedUpfInfoList	map(map(UpfInfo))	O	1..N	This attribute contains the upfInfoList attribute locally configured in the NRF or that the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedPcfInfo	map(PcfInfo)	O	1..N	This attribute contains all the pcfInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the pcfInfo belongs to.
servedPcfInfoList	map(map(PcfInfo))	O	1..N	This attribute contains the pcfInfoList attribute locally configured in the NRF or that the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedBsfInfo	map(BsfInfo)	O	1..N	This attribute contains all the bsfInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the bsfInfo belongs to.

servedBsfInfoList	map(map(BsfInfo))	O	1..N	This attribute contains the bsfInfoList attribute locally configured in the NRF or that the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedChfInfo	map(ChfInfo)	O	1..N	This attribute contains all the chfInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the chfInfo belongs to.
servedChfInfoList	map(map(ChfInfo))	O	1..N	This attribute contains the chfInfoList attribute locally configured in the NRF or that the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedNefInfo	map(NefInfo)	O	1..N	This attribute contains all the nefInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the nefInfo belongs to.
servedNwdafInfo	map(NwdafInfo)	O	1..N	This attribute contains all the nwdafInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the nwdafInfo belongs to.
servedPcscfInfoList	map(map(PcscfInfo))	O	1..N	This attribute contains all the pcscfInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedGmlcInfo	map(GmlcInfo)	O	1..N	This attribute contains all the gmlcInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the gmlcInfo belongs to.
servedLmfInfo	map(LmfInfo)	O	1..N	This attribute contains all the lmfInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of which the lmfInfo belongs to.
servedNfInfo	map(NfInfo)	O	1..N	This attribute contains information of other NFs without corresponding NF type specific Info extensions locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId of the NF.
servedHssInfoList	map(map(HssInfo))	O	1..N	This attribute contains all the hssInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedUdsfInfo	map(UdsfInfo)	O	1..N	This attribute contains all the udsfInfo attributes locally configured in the NRF or the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedUdsfInfoList	map(map(UdsfInfo))	O	1..N	This attribute contains the udsfInfoList attribute locally configured in the NRF or that the NRF received during NF registration. The key of the map is the nfInstanceId to which the map entry belongs to.
servedScpInfoList	map(ScpInfo)	O	1..N	This attribute contains the scpInfo attribute locally configured in the NRF or that the NRF received during SCP registration. The key of the map is the nfInstanceId to which the scpInfo belongs to.
NOTE: The absence of these parameters means the NRF is able to serve any NF discovery request.				

6.1.6.2.32 Type: ChfInfo

Table 6.1.6.2.32-1: Definition of type ChfInfo

Attribute name	Data type	P	Cardinality	Description
supiRangeList	array(SupiRange)	O	1..N	List of ranges of SUPIs that can be served by the CHF instance. If not provided, the CHF can serve any SUPI.
gpsiRangeList	array(IdentityRange)	O	1..N	List of ranges of GPSI that can be served by the CHF instance. If not provided, the CHF can serve any GPSI.
plmnRangeList	array(PlmnRange)	O	1..N	List of ranges of PLMNs (including the PLMN IDs of the CHF instance) that can be served by the CHF instance. If not provided, the CHF can serve any PLMN.
groupId	NfGroupId	O	0..1	Identity of the CHF group that is served by the CHF instance. If not provided, the CHF instance does not pertain to any CHF group.

6.1.6.2.33 Type: ChfServiceInfo

Table 6.1.6.2.33-1: Definition of type ChfServiceInfo

Attribute name	Data type	P	Cardinality	Description
primaryChfServiceInstance	string	C	0..1	This IE shall be present if the CHF service instance serves as a secondary CHF instance of another primary CHF service instance. When present, it shall be set to the serviceInstanceId of the primary CHF service instance. This IE shall be absent if the secondaryChfServiceInstance is present.
secondaryChfServiceInstance	string	C	0..1	This IE shall be present if the CHF service instance serves as a primary CHF instance of another secondary CHF service instance. When present, it shall be set to the serviceInstanceId of the secondary CHF service instance. This IE shall be absent if the primaryChfServiceInstance is present.

6.1.6.2.34 Type: PlmnRange

Table 6.1.6.2.34-1: Definition of type PlmnRange

Attribute name	Data type	P	Cardinality	Description
start	string	O	0..1	First value identifying the start of a PLMN range. The string shall be encoded as follows: <MCC><MNC> Pattern: '^[0-9]{3}[0-9]{2,3}$'
end	string	O	0..1	Last value identifying the end of a PLMN range. The string shall be encoded as follows: <MCC><MNC> Pattern: '^[0-9]{3}[0-9]{2,3}$'
pattern	string	O	0..1	Pattern (regular expression according to the ECMA-262 dialect [8]) representing the set of PLMNs belonging to this range. A PLMN value is considered part of the range if and only if the PLMN string (formatted as <MCC><MNC>) fully matches the regular expression.
NOTE: Either the start and end attributes, or the pattern attribute, shall be present.				

EXAMPLE 1: PLMN range. MCC 123, any MNC
JSON: { "start": "12300", "end": "123999" }

EXAMPLE 2: PLMN range. MCC 123, MNC within range 45 to 49
JSON: { "pattern": "^1234[5-9]$" }, or
JSON: { "start": "12345", "end": "12349" }

EXAMPLE 3: PLMN range. MCC within range 123 to 257, any MNC
JSON: { "start": "12300", "end": "257999" }

6.1.6.2.35 Type: SubscrCond

Table 6.1.6.2.35-1: Definition of type SubscrCond as a list of mutually exclusive alternatives

Data type	Cardinality	Description
NfInstanceIdCond	1	Subscription to a given NF Instance
NfInstanceIdListCond	1	Subscription to a list of NF Instances
NfTypeCond	1	Subscription to a set of NF Instances, identified by their NF Type
ServiceNameCond	1	Subscription to a set of NF Instances that offer a certain service name
AmfCond	1	Subscription to a set of NF Instances (AMFs), belonging to a certain AMF Set and/or belonging to a certain AMF Region.
GuamiListCond	1	Subscription to a set of NF Instances (AMFs), identified by their Guamis.
NetworkSliceCond	1	Subscription to a set of NF Instances, identified by S-NSSAI(s) and NSI ID(s).
NfGroupCond	1	Subscription to a set of NF Instances, identified by a NF (UDM, AUSF or UDR) Group Identity.
NfSetCond	1	Subscription to a set of NF Instances belonging to a certain NF Set.
NfServiceSetCond	1	Subscription to a set of NF Service Instances belonging to a certain NF Service Set.
UpfCond	1	Subscription to a set of NF Instances (UPFs), able to serve a certain service area (i.e. SMF serving area or TAI list).
ScpDomainCond	1	Subscription to a set of NF or SCP instances belonging to certain SCP domains.

6.1.6.2.36 Type: NfInstanceIdCond

Table 6.1.6.2.36-1: Definition of type NfInstanceIdCond

Attribute name	Data type	P	Cardinality	Description
nfInstanceId	NfInstanceId	M	1	NF Instance ID of the NF Instance whose status is requested to be monitored.

6.1.6.2.37 Type: NfTypeCond

Table 6.1.6.2.37-1: Definition of type NfTypeCond

Attribute name	Data type	P	Cardinality	Description
nfType	NFType	M	1	NF type of the NF Instances whose status is requested to be monitored.
NOTE: This type shall not contain the attribute "nfGroupId", to avoid that this type has a matching definition with "NfGroupCond" type.				

6.1.6.2.38 Type: ServiceNameCond

Table 6.1.6.2.38-1: Definition of type ServiceNameCond

Attribute name	Data type	P	Cardinality	Description
serviceName	ServiceName	M	1	Service name offered by the NF Instances whose status is requested to be monitored.

6.1.6.2.39 Type: AmfCond

Table 6.1.6.2.39-1: Definition of type AmfCond

Attribute name	Data type	P	Cardinality	Description
amfSetId	AmfSetId	C	1	AMF Set ID of the NF Instances (AMF) whose status is requested to be monitored.
amfRegionId	AmfRegionId	C	1	AMF Region ID of the NF Instances (AMF) whose status is requested to be monitored.
NOTE 1: At least amfSetId or amfRegionId shall be present; if both the amfRegionId and amfSetId attributes are present in the SubscriptionData, this indicates a subscription for notifications satisfying both attributes (i.e. notifications for NFs from that amfRegionId and amfSetId). NOTE 2: The PLMN ID (or PLMN ID and NID) of the AMF Region and AMF Set of the NF Instances (AMF) whose status is requested to be monitored may be indicated in the plmnId attribute (or plmnid and nid attributes) in the SubscriptionData.				

6.1.6.2.40 Type: GuamiListCond

Table 6.1.6.2.40-1: Definition of type GuamiListCond

Attribute name	Data type	P	Cardinality	Description
guamiList	array(Guami)	M	1..N	Guamis of the NF Instances (AMFs) whose status is requested to be monitored.

6.1.6.2.41 Type: NetworkSliceCond

Table 6.1.6.2.41-1: Definition of type NetworkSliceCond

Attribute name	Data type	P	Cardinality	Description
snssaiList	array(Snssai)	M	1..N	S -NSSAIs of the NF Instances whose status is requested to be monitored.
nsiList	array(string)	O	1..N	NSI IDs of the NF Instances whose status is requested to be monitored.

6.1.6.2.42 Type: NfGroupCond

Table 6.1.6.2.42-1: Definition of type NfGroupCond

Attribute name	Data type	P	Cardinality	Description
nfType	string	M	1	NF type (UDM, AUSF, PCF, UDR or CHF) of the NF Instances whose status is requested to be monitored.
nfGroupId	NfGroupId	M	1	Group ID of the NF Instances whose status is requested to be monitored.

6.1.6.2.43 Type: NotifCondition

Table 6.1.6.2.43-1: Definition of type NotifCondition

Attribute name	Data type	P	Cardinality	Description
monitoredAttributes	array(string)	C	1..N	List of JSON Pointers of attributes in the NF Profile. If this attribute is present, the NRF shall send notification only for changes in the attributes included in this list (see NOTE 1).
unmonitoredAttributes	array(string)	C	1..N	List of JSON Pointers of attributes in the NF Profile. If this attribute is present, the NRF shall send notification for changes on any attribute, except for those included in this list (see NOTE 1).
NOTE 1: Attributes "monitoredAttributes" and "unmonitoredAttributes" shall not be included simultaneously				

EXAMPLE 1: The following JSON object would represent a monitoring condition where the client requests to be notified of all changes on the NF Profile, except "load" attribute.

```
{
  "unmonitoredAttributes": [ "/load" ]
}
```

EXAMPLE 2: The following JSON object would represent a monitoring condition where the client requests to be notified only of changes on attribute "nfStatus":

```
{
  "monitoredAttributes": [ "/nfStatus" ]
}
```

6.1.6.2.44 Type: PlmnSnssai

Table 6.1.6.2.44-1: Definition of type PlmnSnssai

Attribute name	Data type	P	Cardinality	Description
plmnId	PlmnId	M	1	PLMN ID for which list of supported S-NSSAI(s) is provided.
sNssaiList	array(ExtSnssai)	M	1..N	The specific list of S-NSSAIs supported by the given PLMN or SNPN.
nid	Nid	O	0..1	NID for which list of supported S-NSSAI(s) is provided.

6.1.6.2.45 Type: NwdafInfo

Table 6.1.6.2.45-1: Definition of type NwdafInfo

Attribute name	Data type	P	Cardinality	Description
eventIds	array(EventId)	C	1..N	EventId(s) supported by the Nnwdaf_AnalyticsInfo service, if none are provided the NWDAF can serve any eventId.
nwdafEvents	array(NwdafEvent)	C	1..N	Event(s) supported by the Nnwdaf_EventsSubscription service, if none are provided the NWDAF can serve any nwdafEvent.
taiList	array(Tai)	O	1..N	The list of TAIs the NWDAF can serve. It may contain the non-3GPP access TAI. The absence of this attribute and the taiRangeList attribute indicate that the NWDAF can be selected for any TAI in the serving network.
taiRangeList	array(TaiRange)	O	1..N	The range of TAIs the NWDAF can serve. The absence of this attribute and the taiList attribute indicate that the NWDAF can be selected for any TAI in the serving network.

6.1.6.2.46 Type: LmfInfo

Table 6.1.6.2.46-1: Definition of type LmfInfo

Attribute name	Data type	P	Cardinality	Description
servingClientTypes	array(ExternalClientType)	C	1..N	This IE shall be present if the LMF is dedicated to serve the listed external client type(s), e.g. emergency client. The NRF should only include this LMF instance to NF discovery with "client-type" query parameter indicating one of the external client types in the list. Absence of this IE means the LMF is not dedicated to serve specific client types.
lmfId	LMFIdentification	C	0..1	When present, this ID shall indicate the LMF identification.
servingAccessTypes	array(AccessType)	C	1..N	If included, this IE shall contain the access type (i.e. 3GPP_ACCESS and/or NON_3GPP_ACCESS) supported by the LMF. If not included, it shall be assumed that all access types are supported.
servingAnNodeTypes	array(AnNodeType)	C	1..N	If included, this IE shall contain the AN node type (i.e. gNB or NG-eNB) supported by the LMF. If not included, it shall be assumed that all AN node types are supported.
servingRatTypes	array(RatType)	C	1..N	If included, this IE shall contain the RAT type (e.g. 5G NR or eLTE) supported by the LMF. If not included, it shall be assumed that all RAT types are supported.

6.1.6.2.47 Type: GmlcInfo

Table 6.1.6.2.47-1: Definition of type GmlcInfo

Attribute name	Data type	P	Cardinality	Description
servingClientTypes	array(ExternalClientType)	C	1..N	This IE shall be present if the GMLC is dedicated to serve the listed external client type(s), e.g. emergency client. The NRF should only include this GMLC instance to NF discovery with "client-type" query parameter indicating one of the external client types in the list. Absence of this IE means the GMLC is not dedicated to serve specific client types.

6.1.6.2.48 Type: NefInfo

Table 6.1.6.2.48-1: Definition of type NefInfo

Attribute name	Data type	P	Cardinality	Description
nefId	NefId	C	0..1	This IE shall be present and contain the NEF ID of the NEF if NIDD service is supported.
pfdData	PfdData	O	0..1	PFD data. The NRF shall return the NEF profiles that have at least one nnef-pfdmanagement service matching the application identifiers and/or application function identifiers in the corresponding identifier list. If not included, the NRF shall return all the application identifiers and/or application function identifiers registered in the NEF profile.
afEeData	AfEventExposure Data	O	0..1	The AF provided event exposure data. The NEF registers such information in the NRF on behalf of the AF.
gpsiRanges	array(IdentityRange)	O	1..N	Range(s) of External Identifiers
externalGroupIdentifiers Ranges	array(IdentityRange)	O	1..N	Range(s) of External Group Identifiers
servedFqdnList	array(string)	O	1..N	Pattern (regular expression according to the ECMA-262 dialect [8]) representing the Domain names served by the NEF

6.1.6.2.49 Type: PfdData

Table 6.1.6.2.49-1: Definition of type PfdData

Attribute name	Data type	P	Cardinality	Description
appIds	array(string)	O	1..N	List of internal application identifiers of the managed PFDs.
afIds	array(string)	O	1..N	List of application function identifiers of the managed PFDs.

6.1.6.2.50 Type: AfEventExposureData

Table 6.1.6.2.50-1: Definition of type AfEventExposureData

Attribute name	Data type	P	Cardinality	Description
afEvents	array(AfEvent)	M	1..N	AF Event(s) exposed by the NEF after registration of the AF(s) at the NEF.
afIds	array(string)	O	1..N	Associated AF identifications to the AfEvents. The absence of this attribute indicate that the NEF can be selected for any AF.
appIds	array(string)	O	1..N	The list of Application ID(s) the AF(s) connected to the NEF supports. The absence of this attribute indicate that the NEF can be selected for any Application.

6.1.6.2.51 Type: WAgfInfo

Table 6.1.6.2.51-1: Definition of type WAgfInfo

Attribute name	Data type	P	Cardinality	Description
ipv4EndpointAddresses	array(Ipv4Addr)	C	1..N	Available endpoint IPv4 address(es) of the N3 terminations (NOTE 1).
ipv6EndpointAddresses	array(Ipv6Addr)	C	1..N	Available endpoint IPv6 address(es) of the N3 terminations (NOTE 1).

endpointFqdn	Fqdn	C	0..1	Available endpoint FQDN of the N3 terminations (NOTE 1).
NOTE 1: At least one of the addressing parameters (ipv4address, ipv6adress or endpointFqdn) shall be included in the WAgfInfo.				

6.1.6.2.52 Type: TngfInfo

Table 6.1.6.2.52-1: Definition of type TngfInfo

Attribute name	Data type	P	Cardinality	Description
ipv4EndpointAddresses	array(Ipv4Addr)	C	1..N	Available endpoint IPv4 address(es) of the N3 terminations (NOTE 1).
ipv6EndpointAddresses	array(Ipv6Addr)	C	1..N	Available endpoint IPv6 address(es) of the N3 terminations (NOTE 1).
endpointFqdn	Fqdn	C	0..1	Available endpoint FQDN of the N3 terminations (NOTE 1).
NOTE 1: At least one of the addressing parameters (ipv4address, ipv6adress or endpointFqdn) shall be included in the TngfInfo.				

6.1.6.2.53 Type: PcscfInfo

Table 6.1.6.2.53-1: Definition of type PcscfInfo

Attribute name	Data type	P	Cardinality	Description
accessType	array(AccessType)	C	1..N	If included, this IE shall contain the access type (3GPP_ACCESS and/or NON_3GPP_ACCESS) supported by the P-CSCF. If not included, it shall be assumed that all access types are supported.
dnnList	array(Dnn)	O	1..N	DNNs supported by the P-CSCF. The DNN shall contain the Network Identifier and it may additionally contain an Operator Identifier. If the Operator Identifier is not included, the DNN is supported for all the PLMNs in the plmnList of the NF Profile. If not provided, the P-CSCF can serve any DNN.

6.1.6.2.54 Type: NfSetCond

Table 6.1.6.2.54-1: Definition of type NfSetCond

Attribute name	Data type	P	Cardinality	Description
nfSetId	NfSetId	M	1	NF Set ID (see clause 28.12 of 3GPP TS 23.003 [12]) of NF Instances whose status is requested to be monitored.

6.1.6.2.55 Type: NfServiceSetCond

Table 6.1.6.2.55-1: Definition of type NfServiceSetCond

Attribute name	Data type	P	Cardinality	Description
nfServiceSetId	NfServiceSetId	M	1	NF Service Set ID (see clause 28.11 of 3GPP TS 23.003 [12]) of NF service instances whose status is requested to be monitored.

6.1.6.2.56 Type: NfInfo

Table 6.1.6.2.56-1: Definition of type NfInfo

Attribute name	Data type	P	Cardinality	Description
nfType	NFType	M	1	This IE shall indicate the type of the NF.

6.1.6.2.57 Type: HssInfo

Table 6.1.6.2.57-1: Definition of type HssInfo

Attribute name	Data type	P	Cardinality	Description
groupId	NfGroupId	O	0..1	Identity of the HSS group that is served by the HSS instance. If not provided, the HSS instance does not pertain to any HSS group.
imsiRanges	array(ImsiRange)	O	1..N	List of ranges of IMSIs whose profile data is available in the HSS instance (NOTE 1)
NOTE 1: If none of these parameters is provided, the HSS can serve any group and any IMSI.				

6.1.6.2.58 Type: ImsiRange

Table 6.1.6.2.58-1: Definition of type ImsiRange

Attribute name	Data type	P	Cardinality	Description
start	string	O	0..1	First value identifying the start of a IMSI range. Pattern: "^[0-9]+$"
end	string	O	0..1	Last value identifying the end of a IMSI range. Pattern: "^[0-9]+$"
pattern	string	O	0..1	Pattern (regular expression according to the ECMA-262 dialect [8]) representing the set of IMSIs belonging to this range. An IMSI value is considered part of the range if and only if the IMSI string fully matches the regular expression.
NOTE: Either the start and end attributes, or the pattern attribute, shall be present.				

6.1.6.2.59 Type: InternalGroupIdRange

Table 6.1.6.2.59-1: Definition of type InternalGroupIdRange

Attribute name	Data type	P	Cardinality	Description
start	GroupId	O	0..1	First value identifying the start of an identity range, to be used when the range of identities can be represented as a consecutive numeric range.
end	GroupId	O	0..1	Last value identifying the end of an identity range, to be used when the range of identities can be represented as a consecutive numeric range.
pattern	string	O	0..1	Pattern (regular expression according to the ECMA-262 dialect [8]) representing the set of identities belonging to this range. An identity value is considered part of the range if and only if the identity string fully matches the regular expression.
NOTE: Either the start and end attributes, or the pattern attribute, shall be present.				

6.1.6.2.60 Type: UpfCond

Table 6.1.6.2.60-1: Definition of type UpfCond

Attribute name	Data type	P	Cardinality	Description
smfServingArea	array(string)	C	1..N	SMF service area(s) of the UPF whose status is requested to be monitored. This IE shall be present if the monitored granularity is SMF service area(s).
taiList	array(Tai)	C	1..N	TAI(s) of the UPF whose status is requested to be monitored. This IE shall be present if the monitored granularity is TAI list.

6.1.6.2.61 Type: TwifInfo

Table 6.1.6.2.61-1: Definition of type TwifInfo

Attribute name	Data type	P	Cardinality	Description
ipv4EndpointAddresses	array(Ipv4Addr)	C	1..N	Available endpoint IPv4 address(es) of the N3 terminations (NOTE 1)
ipv6EndpointAddresses	array(Ipv6Addr)	C	1..N	Available endpoint IPv6 address(es) of the N3 terminations (NOTE 1)
endpointFqdn	Fqdn	C	0..1	Available endpoint FQDN of the N3 terminations (NOTE 1)
NOTE 1: At least one of the addressing parameters (ipv4address, ipv6adress or endpointFqdn) shall be included in the TwifInfo.				

6.1.6.2.62 Type: VendorSpecificFeature

Table 6.1.6.2.62-1: Definition of type VendorSpecificFeature

Attribute name	Data type	P	Cardinality	Description
featureName	string	M	1	String representing a proprietary feature specific to a given vendor. It is recommended that the case convention for these strings is the same as for enumerated data types (i.e. UPPER_WITH_UNDERSCORE; see 3GPP TS 29.501 [5], clause 5.1.1).
featureVersion	string	M	1	String representing the version of the feature. It is recommended that the versioning system follows the Semantic Versioning Specification [39].

6.1.6.2.63 Type: UdsfInfo

Table 6.1.6.2.63-1: Definition of type UdsfInfo

Attribute name	Data type	P	Cardinality	Description
groupId	NfGroupId	O	0..1	Identity of the UDSF group that is served by the UDSF instance. If not provided, the UDSF instance does not pertain to any UDSF group.
supiRanges	array(SupiRange)	O	1..N	List of ranges of SUPIs whose profile data is available in the UDSF instance (NOTE 1)
NOTE 1: If this parameter is not provided, then the UDSF can serve any SUPI range.				

6.1.6.2.64 Type: NfInstanceListCond

Table 6.1.6.2.64-1: Definition of type NfInstanceListCond

Attribute name	Data type	P	Cardinality	Description
nfInstanceIdList	array(NfInstanceId)	C	1..N	A list of NF Instances whose status is requested to be monitored.

6.1.6.2.65 Type: ScpInfo

Table 6.1.6.2.65-1: Definition of type ScpInfo

Attribute name	Data type	P	Cardinality	Description
scpDomainInfoList	map(ScpDomainInfo)	O	1..N	SCP domain specific information. The key of the map shall be the string identifying an SCP domain.
addressDomains	array(string)	O	1..N	Pattern (regular expression according to the ECMA-262 dialect [8]) representing the address domain names reachable through the SCP.
ipv4Addresses	array(Ipv4Addr)	O	1..N	List of IPv4 addresses reachable through the SCP.
ipv6Prefixes	array(Ipv6Prefix)	O	1..N	List of IPv6 prefixes reachable through the SCP.
ipv4AddrRanges	array(Ipv4AddressRange)	O	1..N	List of IPv4 addresses ranges reachable through the SCP.
ipv6PrefixRanges	array(Ipv6PrefixRange)	O	1..N	List of IPv6 prefixes ranges reachable through the SCP.
servedNfSetIdList	array(NfSetId)	O	1..N	List of NF set ID of NFs served by the SCP.
servedNfTypeList	array(NFType)	O	1..N	List of NF types served by the SCP.
remotePlmnList	array(PlmnId)	O	1..N	List of remote PLMNs reachable through the SCP.

Editor's note: Whether an SCP profile may contain the served NF IDs and served NF Types is pending clarification from SA2.

6.1.6.2.66 Type: ScpDomainInfo

Table 6.1.6.2.66-1: Definition of type ScpDomainInfo

Attribute name	Data type	P	Cardinality	Description
scpFqdn	Fqdn	C	0..1	FQDN of the SCP (NOTE)
scpIpEndPoints	array(IpEndPoint)	C	1..N	IP address(es) and port information of the SCP (NOTE)
NOTE: If any of these attributes is present for a given SCP domain, it shall apply instead of the attributes fqdn, Ipv4Addresses and Ipv4Addresses within the NFProfile data type for the corresponding SCP Domain. If none of these attributes is present for a given SCP domain, the attributes fqdn, Ipv4Addresses, and Ipv4Addresses within the NFProfile data type shall apply for the corresponding SCP Domain.				

6.1.6.2.67 Type: ScpDomainCond

Table 6.1.6.2.67-1: Definition of type ScpDomainCond

Attribute name	Data type	P	Cardinality	Description
scpDomains	array(string)	M	1..N	SCP domains of NF or SCP instances whose status is requested to be monitored.

6.1.6.2.68 Type: OptionsResponse

Table 6.1.6.2.68-1: Definition of type OptionsResponse

Attribute name	Data type	P	Cardinality	Description
supportedFeatures	SupportedFeatures	C	0..1	Supported features of the NRF, for the nf-instances store resource. See clause 6.1.9. This IE shall be included if at least one Nnrf_NFManagement feature is supported by the NRF.

6.1.6.3 Simple data types and enumerations

6.1.6.3.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

6.1.6.3.2 Simple data types

The simple data types defined in table 6.1.6.3.2-1 shall be supported.

Table 6.1.6.3.2-1: Simple data types

Type Name	Type Definition	Description
Fqdn	string	FQDN (Fully Qualified Domain Name)
NefId	string	The NEF ID as specified in clause 4.25.2 of 3GPP TS 23.502 [3]. For combined SCEF+NEF, the NEF ID shall contain the SCEF ID encoded as specified in clause 8.4.5 of 3GPP TS 29.336 [37].
VendorId	string	Vendor ID, according to the IANA-assigned "SMI Network Management Private Enterprise Codes" [38]. It shall be formatted as a fixed 6-digit string, padding with leading digits "0" to complete a 6-digit length. Pattern: "^[0-9]{6}$"

6.1.6.3.3 Enumeration: NFType

The enumeration NFType represents the different types of Network Functions or Network Entities that can be found in the 5GC.

Table 6.1.6.3.3-1: Enumeration NFType

Enumeration value	Description
"NRF"	Network Function: NRF
"UDM"	Network Function: UDM
"AMF"	Network Function: AMF
"SMF"	Network Function: SMF
"AUSF"	Network Function: AUSF
"NEF"	Network Function: NEF
"PCF"	Network Function: PCF
"SMSF"	Network Function: SMSF
"NSSF"	Network Function: NSSF
"UDR"	Network Function: UDR
"LMF"	Network Function: LMF
"GMLC"	Network Function: GMLC
"5G_EIR"	Network Function: 5G-EIR
"SEPP"	Network Entity: SEPP
"UPF"	Network Function: UPF
"N3IWF"	Network Function and Entity: N3IWF
"AF"	Network Function: AF
"UDSF"	Network Function: UDSF
"BSF"	Network Function: BSF
"CHF"	Network Function: CHF
"NWDAF"	Network Function: NWDAF
"PCSCF"	Network Function: P-CSCF
"CBCF"	Network Function: CBCF
"UCMF"	Network Function: UCMF
"HSS"	Network Function: HSS
"SOR_AF"	Network Function: SOR-AF
"SPAF"	Network Function: SP-AF
"MME"	Network Function: MME
"SCSAS"	Network Function: SCS/AS
"SCEF"	Network Function: SCEF
"SCP"	Network Entity: SCP
"NSSAAF"	Network Function: NSSAAF

6.1.6.3.4 Enumeration: NotificationType

Table 6.1.6.3.4-1: Enumeration NotificationType

Enumeration value	Description
"N1_MESSAGES"	Notification of N1 messages
"N2_INFORMATION"	Notification of N2 information
"LOCATION_NOTIFICATION"	Notification of Location Information by AMF towards NF Service Consumers (e.g GMLC)
"DATA_REMOVAL_NOTIFICATION"	Notification of Data Removal by UDR (e.g., removal of UE registration data upon subscription withdrawal)
"DATA_CHANGE_NOTIFICATION"	Notification of Data Changes by UDR
"LOCATION_UPDATE_NOTIFICATION"	Notification of UE Location Information Update by GMLC towards NF Service Consumers (e.g. NEF), during MO_LR procedure.
"NSSAA_REAUTH_NOTIFICATION"	Re-authentication notification for slice-specific authentication and authorization by AAA-P/S towards NF Service Consumers (e.g. AMF)
"NSSAA_REVOC_NOTIFICATION"	Revocation notification for slice-specific authentication and authorization by AAA-P/S towards NF Service Consumers (e.g. AMF)

6.1.6.3.5 Enumeration: TransportProtocol

Table 6.1.6.3.5-1: Enumeration TransportProtocol

Enumeration value	Description
"TCP"	Transport protocol: TCP

6.1.6.3.6 Enumeration: NotificationEventType

Table 6.1.6.3.6-1: Enumeration NotificationEventType

Enumeration value	Description
"NF_REGISTERED"	The NF Instance has been registered in NRF
"NF_DEREGISTERED"	The NF Instance has been deregistered from NRF
"NF_PROFILE_CHANGED"	The profile of the NF Instance has been modified

6.1.6.3.7 Enumeration: NFStatus

Table 6.1.6.3.7-1: Enumeration NFStatus

Enumeration value	Description
"REGISTERED"	The NF Instance is registered in NRF and can be discovered by other NFs.
"SUSPENDED"	The NF Instance is registered in NRF but it is not operative and cannot be discovered by other NFs. This status may result from a NF Heart-Beat failure (see clause 5.2.2.3.2) or a NF failure and may trigger restoration procedures (see clause 6.2 of 3GPP 23.527 [27]).
"UNDISCOVERABLE"	The NF instance is registered in NRF, is operative but cannot be discovered by other NFs. This status may be set by the NF e.g. in shutting down scenarios where the NF is still able to process requests for existing resources or sessions but cannot accept new resource creation or session establishment.

6.1.6.3.8 Enumeration: DataSetId

The enumeration DataSetId represents the different types of data sets supported by an UDR instance.

Table 6.1.6.3.8-1: Enumeration DataSetId

Enumeration value	Description
"SUBSCRIPTION"	Data set: Subscription data
"POLICY"	Data set: Policy data
"EXPOSURE"	Data set: Structured data for exposure
"APPLICATION"	Data set: Application data

6.1.6.3.9 Enumeration: UPInterfaceType

Table 6.1.6.3.9-1: Enumeration UPInterfaceType

Enumeration value	Description
"N3"	User Plane Interface: N3
"N6"	User Plane Interface: N6
"N9"	User Plane Interface: N9
"DATA_FORWARDING"	User Plane Interface for indirect data forwarding. (NOTE 1)
NOTE 1: This interface type may be used when a dedicated network instance is deployed for data forwarding.	

6.1.6.3.10 Relation Types

6.1.6.3.10.1 General

This clause describes the possible relation types defined within NRF API. See clause 4.7.5.2 of 3GPP TS 29.501 [5] for the description of the relation types.

Table 6.1.6.3.10.1-1: supported registered relation types

Relation Name
self
item

6.1.6.3.11 Enumeration: ServiceName

Table 6.1.6.3.11-1: Enumeration ServiceName

Enumeration value	Description
"nnrf-nfm"	Nnrf_NFManagement Service offered by the NRF
"nnrf-disc"	Nnrf_NFDiscovery Service offered by the NRF
"nnrf-oauth2"	Nnrf_AccessToken Service offered by the NRF
"nudm-sdm"	Nudm_SubscriberDataManagement Service offered by the UDM
"nudm-uecm"	Nudm_UEContextManagement Service offered by the UDM
"nudm-ueau"	Nudm_UEAuthentication Service offered by the UDM
"nudm-ee"	Nudm_EventExposure Service offered by the UDM
"nudm-pp"	Nudm_ParameterProvision Service offered by the UDM
"nudm-niddau"	Nudm_NIDDAuthorization Service offered by the UDM
"nudm-mt"	Nudm_MT Service offered by the UDM
"namf-comm"	Namf_Communication Service offered by the AMF
"namf-evts"	Namf_EventExposure Service offered by the AMF
"namf-mt"	Namf_MT Service offered by the AMF
"namf-loc"	Namf_Location Service offered by the AMF
"nsmf-pdusession"	Nsmf_PDUSession Service offered by the SMF
"nsmf-event-exposure"	Nsmf_EventExposure Service offered by the SMF
"nsmf-nidd"	Nsmf_NIDD Service offered by the SMF
"nausf-auth"	Nausf_UEAuthentication Service offered by the AUSF
"nausf-sorprotection"	Nausf_SoRProtection Service offered by the AUSF
"nausf-upuprotection"	Nausf_UPUProtection Service offered by the AUSF
"nnef-pfdmanagement"	Nnef_PFDManagement offered by the NEF
"nnef-smcontext"	Nnef_SMContext Service offered by the NEF
"nnef-eventexposure"	Nnef_EventExposure Service offered by the NEF
"npcf-am-policy-control"	Npcf_AMPolicyControl Service offered by the PCF
"npcf-smpolicycontrol"	Npcf_SMPolicyControl Service offered by the PCF
"npcf-policyauthorization"	Npcf_PolicyAuthorization Service offered by the PCF
"npcf-bdtpolicycontrol"	Npcf_BDTPolicyControl Service offered by the PCF
"npcf-eventexposure"	Npcf_EventExposure Service offered by the PCF
"npcf-ue-policy-control"	Npcf_UEPolicyControl Service offered by the PCF
"nsmsf-sms"	Nsmsf_SMService Service offered by the SMSF
"nnssf-nsselection"	Nnssf_NSSelection Service offered by the NSSF
"nnssf-nssaiavailability"	Nnssf_NSSAIAvailability Service offered by the NSSF
"nudr-dr"	Nudr_DataRepository Service offered by the UDR
"nudr-group-id-map"	Nudr_GroupIDmap Service offered by the UDR
"nlmf-loc"	Nlmf_Location Service offered by the LMF
"n5g-eir-eic"	N5g-eir_EquipmentIdentityCheck Service offered by the 5G-EIR
"nbsf-management"	Nbsf_Management Service offered by the BSF
"nchf-spendinglimitcontrol"	Nchf_SpendingLimitControl Service offered by the CHF
"nchf-convergedcharging"	Nchf_Converged_Charging Service offered by the CHF
"nchf-offlineonlycharging"	Nchf_OfflineOnlyCharging Service offered by the CHF
"nnwdaf-eventssubscription"	Nnwdaf_EventsSubscription Service offered by the NWDAF
"nnwdaf-analyticsinfo"	Nnwdaf_AnalyticsInfo Service offered by the NWDAF
"ngmlc-loc"	Ngmlc_Location Service offered by GMLC
"nucmf-provisioning"	Nucmf_Provisioning Service offered by UCMF
"nucmf-uecapabilitymanagement"	Nucmf_UECapabilityManagement Service offered by UCMF
"nhss-sdm"	Nhss_SubscriberDataManagement Service offered by the HSS
"nhss-uecm"	Nhss_UEContextManagement Service offered by the HSS
"nhss-ueau"	Nhss_UEAuthentication Service offered by the HSS
"nhss-ee"	Nhss_EventExposure Service offered by the HSS
"nhss-ims-sdm"	Nhss_imsSubscriberDataManagement Service offered by the HSS
"nhss-ims-uecm"	Nhss_imsUEContextManagement Service offered by the HSS
"nhss-ims-ueau"	Nhss_imsUEAuthentication Service offered by the HSS
"nsepp-telescopic"	Nsepp_Telescopic_FQDN_Mapping Service offered by the SEPP
"nsoraf-sor"	Nsoraf_SteeringOfRoaming Service offered by the SOR-AF
"nspaf-secured-packed"	Nspaf_SecuredPacket Service offered by the SP-AF
"nudsf-dr"	Nudsf Data Repository service offered by the UDSF.
"nnssaaf-nssaa"	Nnssaaf_NSSAA service offered by the NSSAAF.
NOTE: The services defined in this table are those defined by 3GPP NFs in 5GC; however, in order to support custom services offered by standard and custom NFs, the NRF shall also accept the registration of NF Services with other service names.	

6.1.6.3.12 Enumeration: NFServiceStatus

Table 6.1.6.3.12-1: Enumeration NFServiceStatus

Enumeration value	Description
"REGISTERED"	The NF Service Instance is registered in NRF and can be discovered by other NFs.
"SUSPENDED"	The NF Service Instance is registered in NRF but it is not operative and cannot be discovered by other NFs. This status may result from a NF Service failure and may trigger restoration procedures (see clause 6.2 of 3GPP 23.527 [27]).
"UNDISCOVERABLE"	The NF Service instance is registered in NRF, is operative but cannot be discovered by other NFs. This status may be set by the NF e.g. in shutting down scenarios where the NF service is still able to process requests for existing resources or sessions but cannot accept new resource creation or session establishment.
NOTE: An NF service cannot be discovered by other NFs if the NF status is set to "SUSPENDED" or "UNDISCOVERABLE", regardless of the NF service status.	

6.1.6.3.13 Enumeration: AnNodeType

Table 6.1.6.3.13-1: Enumeration AnNodeType

Enumeration value	Description
"GNB"	gNB
"NG_ENB"	NG-eNB

6.1.6.3.14 Enumeration: ConditionEventType

Table 6.1.6.3.14-1: Enumeration ConditionEventType

Enumeration value	Description
"NF_ADDED"	The NF Instance notified by NRF starts being part of a condition for a subscription on a set of NFs.
"NF_REMOVED"	The NF Instance notified by NRF stops being part of a condition for a subscription on a set of NFs.

6.1.7 Error Handling

6.1.7.1 General

HTTP error handling shall be supported as specified in clause 5.2.4 of 3GPP TS 29.500 [4].

6.1.7.2 Protocol Errors

Protocol errors handling shall be supported as specified in clause 5.2.7 of 3GPP TS 29.500 [4].

6.1.7.3 Application Errors

The application errors defined for the Nnrf_NFManagement service are listed in Table 6.1.7.3-1.

Table 6.1.7.3-1: Application errors

Application Error	HTTP status code	Description

6.1.8 Security

As indicated in 3GPP TS 33.501 [15], the access to the Nnrf_NFManagement API may be authorized by means of the OAuth2 protocol (see IETF RFC 6749 [16]), using the "Client Credentials" authorization grant, where the NRF plays the role of the authorization server.

If Oauth2 authorization is used, an NF Service Consumer, prior to consuming services offered by the Nnrf_NFManagement API, shall obtain a "token" from the authorization server, by invoking the Access Token Request service, as described in clause 5.4.2.2.

NOTE: When multiple NRFs are deployed in a network, the NRF used as authorization server is the same NRF where the Nnrf_NFManagement service is invoked by the NF Service Producer.

The Nnrf_NFManagement API defines scopes for OAuth2 authorization as specified in 3GPP TS 33.501 [15]; it defines a single scope consisting on the name of the service (i.e., "nnrf-nfm"), and it does not define any additional scopes at resource or operation level.

6.1.9 Features supported by the NFManagement service

The syntax of the supportedFeatures attribute is defined in clause 5.2.2 of 3GPP TS 29.571 [7].

The following features are defined for the Nnrf_NFManagement service.

Table 6.1.9-1: Features of supportedFeatures attribute used by Nnrf_NFManagement service

Feature Number	Feature	M/O	Description
1	Service-Map	M	Support of defining in the profile of the NF Instance the list of NF Service Instances based on a map type (i.e. support of the "nfServiceList" attribute in NFProfile).
Feature number: The order number of the feature within the supportedFeatures attribute (starting with 1). Feature: A short name that can be used to refer to the bit and to the feature. M/O: Defines if the implementation of the feature is mandatory ("M") or optional ("O"). Description: A clear textual description of the feature.			

6.2 Nnrf_NFDiscovery Service API

6.2.1 API URI

URIs of this API shall have the following root:

{apiRoot}/{apiName}/{apiVersion}/

where "apiRoot" is defined in clause 4.4.1 of 3GPP TS 29.501 [5], the "apiName" shall be set to "nnrf-disc" and the "apiVersion" shall be set to "v1" for the current version of this specification.

6.2.2 Usage of HTTP

6.2.2.1 General

HTTP/2, as defined in IETF RFC 7540 [9], shall be used as specified in clause 5 of 3GPP TS 29.500 [4].

HTTP/2 shall be transported as specified in clause 5.3 of 3GPP TS 29.500 [4].

HTTP messages and bodies for the Nnrf_NFDiscovery service shall comply with the OpenAPI [10] specification contained in Annex A.

6.2.2.2 HTTP standard headers

6.2.2.2.1 General

The mandatory standard HTTP headers as specified in clause 5.2.2.2 of 3GPP TS 29.500 [4] shall be supported.

6.2.2.2.2 Content type

The following content types shall be supported:

- The JSON format (IETF RFC 8259 [22]). The use of the JSON format shall be signalled by the content type "application/json". See also clause 5.4 of 3GPP TS 29.500 [4].
- The Problem Details JSON Object (IETF RFC 7807 [11]). The use of the Problem Details JSON object in a HTTP response body shall be signalled by the content type "application/problem+json".

6.2.2.2.3 Cache-Control

A "Cache-Control" header should be included in HTTP responses, as described in IETF RFC 7234 [20], clause 5.2. It shall contain a "max-age" value, indicating the amount of time in seconds after which the received response is considered stale; this value shall be the same as the content of the "validityPeriod" element described in clause 6.2.6.2.2.

6.2.2.2.4 ETag

An "ETag" (entity-tag) header should be included in HTTP responses, as described in IETF RFC 7232 [19], clause 2.3. It shall contain a server-generated strong validator, that allows further matching of this value (included in subsequent client requests) with a given resource representation stored in the server or in a cache.

6.2.2.2.5 If-None-Match

An NF Service Consumer should issue conditional GET request towards NRF, by including an If-None-Match header in HTTP requests, as described in IETF RFC 7232 [19], clause 3.2, containing one or several entity tags received in previous responses for the same resource.

6.2.2.3 HTTP custom headers

6.2.2.3.1 General

In this release of this specification, no custom headers specific to the Nnrf_NFDiscovery service are defined. For 3GPP specific HTTP custom headers used across all service-based interfaces, see clause 5.2.3 of 3GPP TS 29.500 [4].

6.2.3 Resources

6.2.3.1 Overview

The structure of the Resource URIs of the NFDiscovery service is shown in figure 6.2.3.1-1.

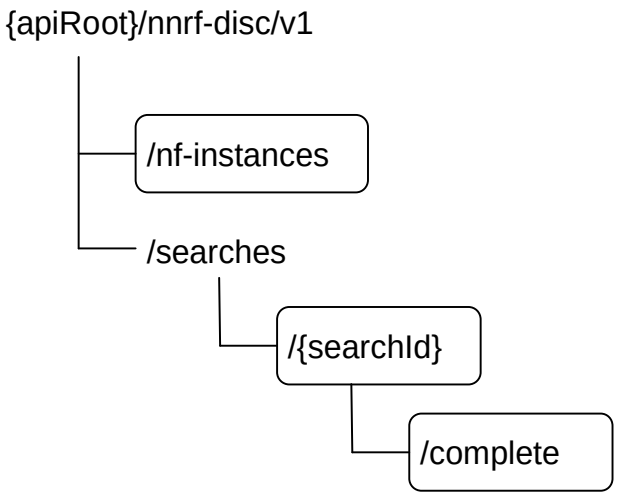

Figure 6.2.3.1-1: Resource URI structure of the NFDiscovery API

Table 6.2.3.1-1 provides an overview of the resources and applicable HTTP methods.

Table 6.2.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
nf-instances (Store)	/nf-instances	GET	Retrieve a collection of NF Instances according to certain filter criteria.
Stored Search (Document)	/searches/{searchId}	GET	Retrieve a collection of NF Instances, previously stored by NRF as a consequence of a prior search result.
Complete Stored Search (Document)	/searches/{searchId}/complete	GET	Retrieve a collection of NF Instances, previously stored by NRF as a consequence of a prior search result, without applying any client restriction on the number of instances (e.g. "limit" or "max-payload-size" query parameters).

6.2.3.2 Resource: nf-instances (Store)

6.2.3.2.1 Description

This resource represents a collection of the different NF instances registered in the NRF.

This resource is modelled as the Store resource archetype (see clause C.3 of 3GPP TS 29.501 [5]).

6.2.3.2.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-disc/v1/nf-instances**

This resource shall support the resource URI variables defined in table 6.1.3.2.2-1.

Table 6.2.3.2.2-1: Resource URI variables for this resource

Name	Data type	Definition
apiRoot	string	See clause 6.1.1

6.2.3.2.3 Resource Standard Methods

6.2.3.2.3.1 GET

This operation retrieves a list of NF Instances, and their offered services, currently registered in the NRF, satisfying a number of filter criteria, such as those NF Instances offering a certain service name, or those NF Instances of a given NF type (e.g., AMF).

Table 6.2.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description	Applicability
target-nf-type	NFType	M	1	This IE shall contain the NF type of the NF Service Producer being discovered.	
requester-nf-type	NFType	M	1	This IE shall contain the NF type of the NF Service Consumer that is invoking the Nnrf_NFDiscovery service.	
requester-nf-instance-id	NfInstanceId	O	0..1	If included, this IE shall contain the NF instance id of the NF service consumer.	Query-Params-Ext2
service-names	array(ServiceName)	O	1..N	If included, this IE shall contain an array of service names for which the NRF is queried to provide the list of NF profiles. The NRF shall return the NF profiles that have at least one NF service matching the NF service names in this list. The NF service names returned by the NRF shall be an interclause of the NF service names requested and the NF service names registered in the NF profile. If not included, the NRF shall return all the NF service names registered in the NF profile. Contains unique items.	
requester-nf-instance-fqdn	Fqdn	O	0..1	This IE may be present for an NF discovery request within the same PLMN as the NRF. If included, this IE shall contain the FQDN of the NF Service Consumer that is invoking the Nnrf_NFDiscovery service. The NRF shall use this to return only those NF profiles that include at least one NF service containing an entry in the "allowedNfDomains" list (see clause 6.1.6.2.3) that matches the domain of the requester NF. This IE shall be ignored by the NRF if it is received from a requester NF belonging to a different PLMN. (NOTE 12)	
target-plmn-list	array(PlmnId)	C	1..N	This IE shall be included when NF services in a different PLMN, or NF services of specific PLMN ID(s) in a same PLMN comprising multiple PLMN IDs, need to be discovered. When included, this IE shall contain the PLMN ID of the target NF. If more than one PLMN ID is included, NFs from any PLMN ID present in the list matches the query parameter. For inter-PLMN service discovery, at most 1 PLMN ID shall be included in the list; it shall be included in the service discovery from the NF in the source PLMN sent to the NRF in the same PLMN, while it may be absent in the service discovery request sent from the source NRF to the target NRF. In such case, if the NRF receives more than 1 PLMN ID, it shall only consider the first element of the array, and ignore the rest.	
requester-plmn-list	array(PlmnId)	C	1..N	This IE shall be included when NF services in a different PLMN need to be discovered. When included, this IE shall contain the PLMN ID(s) of the requester NF. (NOTE 12)	
requester-snpn-list	array(PlmnIdNid)	C	1..N	This IE shall be included when the NF service consumer belongs to one or several SNPNs, and NF services of a specific SNPN need to be discovered. When present, this IE shall contain the SNPN ID(s) of the requester NF. The NRF shall use this to return only those NF profiles of NF Instances allowing to be discovered from the SNPNs identified by this IE, according to the "allowedSnpns" list in the NF Profile and NF Service (see clauses 6.1.6.2.2 and 6.1.6.2.3).	Query-Params-Ext2
target-nf-instance-id	NfInstanceId	O	0..1	Identity of the NF instance being discovered.	
target-nf-fqdn	Fqdn	O	0..1	FQDN of the target NF instance being discovered.	
hnrf-uri	Uri	C	0..1	If included, this IE shall contain the API URI of the NFDiscovery Service (see clause 6.2.1) of the home NRF. It shall be included if the NF Service Consumer has previously received such API URI to be used for service discovery (e.g., from the NSSF in the home PLMN).	

snssais	array(Snssai)	O	1..N	If included, this IE shall contain the list of S-NSSAIs that are served by the NF (Service) Instances being discovered. The NRF shall return those NF profiles/NF services of NF (Service) Instances that have at least one of the S-NSSAIs in this list. The S-NSSAIs included in the NF profiles/NF services of NF (Service) Instances returned by the NRF shall be an interclause of the S-NSSAIs requested and the S-NSSAIs supported by those NF (Service) Instances. (NOTE 10)	
requester-snssais	array(Snssai)	O	1..N	If included, this IE shall contain the list of S-NSSAI of the requester NF. If this IE is included in a service discovery in a different PLMN, the requester NF shall provide S-NSSAI values of the target PLMN, that correspond to the S-NSSAI values of the requester NF. The NRF shall use this to return only those NF profiles of NF Instances allowing to be discovered from at least one network slice identified by this IE, according to the "allowedNssais" list in the NF Profile and NF Service (see clause 6.1.6.2.2 and 6.1.6.2.3). (NOTE 12)	
plmn-specific-snssai-list	array(PlmnSnssai)	O	1..N	If included, this IE shall contain the list of S-NSSAI that are served by the NF service being discovered for the corresponding PLMN provided. The NRF shall use this to identify the NF services that have registered their support for the S-NSSAIs for the corresponding PLMN given. The NRF shall return the NF profiles that have at least one per PLMN S-NSSAI entry matching the PLMN specific S-NSSAIs provided in this list. The per PLMN list of S-NSSAIs included in the NF profile returned by the NRF shall be an interclause of the list requested and the list registered in the NF profile. (NOTE 10).	
nsi-list	array(string)	O	1..N	If included, this IE shall contain the list of NSI IDs that are served by the services being discovered.	
dnn	Dnn	O	0..1	If included, this IE shall contain the DNN for which NF services serving that DNN is discovered. DNN may be included if the target NF type is e.g. "BSF", "SMF", "PCF", "PCSCF" or "UPF". The DNN shall contain the Network Identifier and it may additionally contain an Operator Identifier. (NOTE 11). If the Snssai(s) are also included, the NF services serving the DNN shall be available in the network slice(s) identified by the Snssai(s).	
smf-serving-area	string	O	0..1	If included, this IE shall contain the serving area of the SMF. It may be included if the target NF type is "UPF".	
tai	Tai	O	0..1	Tracking Area Identity.	
amf-region-id	AmfRegionId	O	0..1	AMF Region Identity.	
amf-set-id	AmfSetId	O	0..1	AMF Set Identity.	
guami	Guami	O	0..1	Guami used to search for an appropriate AMF. (NOTE 1)	
supi	Supi	O	0..1	If included, this IE shall contain the SUPI of the requester UE to search for an appropriate NF. SUPI may be included if the target NF type is e.g. "PCF", "CHF", "AUSF", "UDM" or "UDR".	
ue-ipv4-address	Ipv4Addr	O	0..1	The IPv4 address of the UE for which a BSF needs to be discovered.	
ip-domain	string	O	0..1	The IPv4 address domain of the UE for which a BSF needs to be discovered.	
ue-ipv6-prefix	Ipv6Prefix	O	0..1	The IPv6 prefix of the UE for which a BSF needs to be discovered.	
pgw-ind	boolean	O	0..1	When present, this IE indicates whether a combined SMF/PGW-C or a standalone SMF needs to be discovered. true: A combined SMF/PGW-C is requested to be discovered; false: A standalone SMF is requested to be discovered. (See NOTE 2)	
pgw	Fqdn	O	0..1	If included, this IE shall contain the PGW FQDN which is received by the AMF from the MME to find the combined SMF/PGW.	
gpsi	Gpsi	O	0..1	If included, this IE shall contain the GPSI of the requester UE to search for an appropriate NF. GPSI may be included if the target NF type is "CHF", "PCF", "UDM" or "UDR".	

external-group-identity	ExtGroupId	O	0..1	If included, this IE shall contain the external group identifier of the requester UE to search for an appropriate NF. This may be included if the target NF type is "UDM" or "UDR".	
pfd-data	PfdData	O	0..1	When present, this IE shall contain the application identifiers and/or application function identifiers in PFD management. This may be included if the target NF type is "NEF".	Query-Params-Ext2
data-set	DataSetId	O	0..1	Indicates the data set to be supported by the NF to be discovered. May be included if the target NF type is "UDR".	
routing-indicator	string	O	0..1	Routing Indicator information that allows to route network signalling with SUCI (see 3GPP 23.003 [12]) to an AUSF and UDM instance capable to serve the subscriber. May be included if the target NF type is "AUSF" or "UDM". Pattern: "^[0-9]{1,4}$"	
group-id-list	array(NfGroupId)	O	1..N	Identity of the group(s) of the NFs of the target NF type to be discovered. May be included if the target NF type is "UDR", "UDM", "HSS", "PCF", "AUSF" or "CHF".	
dnai-list	array(Dnai)	O	1..N	If included, this IE shall contain the Data network access identifiers. It may be included if the target NF type is "UPF".	
upf-iwk-eps-ind	boolean	O	0..1	When present, this IE indicates whether a UPF supporting interworking with EPS needs to be discovered. true: A UPF supporting interworking with EPS is requested to be discovered; false: A UPF not supporting interworking with EPS is requested to be discovered. (NOTE 3)	
chf-supported-plmn	PlmnId	O	0..1	If included, this IE shall contain the PLMN ID that a CHF supports (i.e., in the PlmnRange of ChfInfo attribute in the NFProfile). This IE may be included when the target NF type is "CHF".	
preferred-locality	string	O	0..1	Preferred target NF location (e.g. geographic location, data center). When present, the NRF shall prefer NF profiles with a locality attribute that matches the preferred-locality. The NRF may return additional NFs in the response not matching the preferred target NF location, e.g. if no NF profile is found matching the preferred target NF location. The NRF should set a lower priority for any additional NFs on the response not matching the preferred target NF location than those matching the preferred target NF location. (NOTE 6)	
access-type	AccessType	C	0..1	If included, this IE shall contain the Access type which is required to be supported by the target Network Function (i.e. SMF).	
supported-features	SupportedFeatures	O	0..1	List of features required to be supported by the target Network Function. This IE may be present only if the service-names attribute is present and if it contains a single service-name, or if the target Network Function does not support any service. It shall be ignored by the NRF otherwise. (NOTE 4)	
required-features	array(SupportedFeatures)	O	1..N	List of features required to be supported by the target Network Function, as defined by the supportedFeatures attribute in NFService (see clauses 6.1.6.2.3 and 6.2.6.2.4). This IE may be present only if the service-names attribute is present. When present, the required-features attribute shall contain as many entries as the number of entries in the service-names attribute. The n^{th} entry in the required-features attribute shall correspond to the n^{th} entry in the service-names attribute. An entry corresponding to a service for which no specific feature is required shall be encoded as "0".	Query-Params-Ext1
complex-query	ComplexQuery	O	0..1	This query parameter is used to override the default logical relationship of query parameters.	Complex-Query
limit	integer	O	0..1	Maximum number of NFProfiles to be returned in the response. Minimum: 1	Query-Params-Ext1

max-payload-size	integer	O	0..1	Maximum payload size (before compression, if any) of the response, expressed in kilo octets. When present, the NRF shall limit the number of NF profiles returned in the response such as to not exceed the maximum payload size indicated in the request. Default: 124. Maximum: 2000 (i.e. 2 Mo).	Query-Params-Ext1
max-payload-size-ext	integer	O	0..1	Maximum payload size (before compression, if any) of the response, expressed in kilo octets. When present, the NRF shall limit the number of NF profiles returned in the response such as to not exceed the maximum payload size indicated in the request. This query parameter is used when the consumer supports payload size bigger than 2 million octets. Default: 124	Query-Params-Ext2
pdu-session-types	array(PduSessionType)	O	1..N	List of the PDU session type (s) requested to be supported by the target Network Function (i.e UPF).	Query-Params-Ext1
event-id-list	array(EventId)	O	1..N	If present, this attribute shall contain the list of events requested to be supported by the Nnwdaf AnalyticsInfo Service, the NRF shall return NF which support all the requested events.	Query-Param-Analytics
nwdaf-event-list	array(NwdafEvent)	O	1..N	If present, this attribute shall contain the list of events requested to be supported by the Nnwdaf_EventsSubscription service, the NRF shall return NF which support all the requested events.	Query-Param-Analytics
atsss-capability	AtsssCapability	O	0..1	When present, this IE indicates the ATSSS capability of the target UPF needs to be supported.	MAPDU
upf-ue-ip-addr-ind	boolean	O	0..1	When present, this IE indicates whether a UPF supporting allocating UE IP addresses/prefixes needs to be discovered. true: a UPF supporting UE IP addresses/prefixes allocation is requested to be discovered; false: a UPF not supporting UE IP addresses/prefixes allocation is requested to be discovered.	Query-Params-Ext2
client-type	ExternalClientType	O	0..1	When present, this IE indicates that NF(s) dedicatedly serving the specified Client Type needs to be discovered. This IE may be included when target NF Type is "LMF" and "GMLC". If no NF profile is found dedicately serving the requested client type, the NRF may return NF(s) not dedicatedly serving the request client type in the response.	Query-Params-Ext2
lmf-id	LMFIdentification	O	0..1	When present, this IE shall contain LMF identification to be discovered.This may be included if the target NF type is "LMF".	Query-Params-Ext2
an-node-type	AnNodeType	O	0..1	If included, this IE shall contain the AN Node type which is required to be supported by the target Network Function (i.e. LMF).	Query-Params-Ext2
rat-type	RatType	O	0..1	If included, this IE shall contain the RAT type which is required to be supported by the target Network Function (i.e. LMF).	Query-Params-Ext2
target-snpn	PlmnIdNid	C	0..1	This IE shall be included when NF services of a specific SNPN need to be discovered. When included, this IE shall contain the PLMN ID and NID of the target NF.	Query-Params-Ext2
af-ee-data	AfEventExposureData	O	0..1	When present, this shall contain the application events, and optionally application function identifiers, application identifiers of the AF(s). This may be included if the target NF type is "NEF".	Query-Params-Ext2
w-agf-info	WAgfInfo	O	0..1	If included, this IE shall contain the W-AGF identifiers of N3 terminations which is received by the SMF to find the combined W-AGF/UPF.	Query-Params-Ext2
tngf-info	TngfInfo	O	0..1	If included, this IE shall contain the TNGF identifiers of N3 terminations which is received by the SMF to find the combined TNGF/UPF.	Query-Params-Ext2
twif-info	TwifInfo	O	0..1	If included, this IE shall contain the TWIF identifiers of N3 terminations which is received by the SMF to find the combined TWIF/UPF.	Query-Params-Ext2
target-nf-set-id	NfSetId	O	0..1	When present, this IE shall contain the target NF Set ID (as defined in clause 28.12 of 3GPP TS 23.003 [12]) of the NF instances being discovered.	Query-Params-Ext2

target-nf-service-set-id	NfServiceSetId	O	0..1	When present, this IE shall contain the target NF Service Set ID (as defined in clause 28.13 of 3GPP TS 23.003 [12]) of the NF service instances being discovered.	Query-Params-Ext2
preferred-tai	Tai	O	0..1	When present, the NRF shall prefer NF profiles that can serve the TAI, or the NRF shall return NF profiles not matching the TAI if no NF profile is found matching the TAI. (NOTE 5)	Query-Params-Ext2
nef-id	NefId	O	0..1	When present, this IE shall contain the NEF ID of the NEF to be discovered. This may be included if the target NF type is "NEF". (NOTE 7)	Query-Params-Ext2
preferred-nf-instances	array(NfInstanceId)	O	1..N	When present, this IE shall contain a list of preferred candidate NF instance IDs. (NOTE 8)	Query-Params-Ext2
notification-type	NotificationType	O	0..1	If included, this IE shall contain the notification type of default notification subscriptions that shall be registered in the NFProfile or NFService of the NF Instances being discovered. The NF profiles returned by the NRF shall contain all the registered default notification subscriptions, including the one corresponding to the notification-type parameter. (NOTE 9)	Query-Params-Ext2
serving-scope	array(string)	O	1..N	If present, this attribute shall contain the list of areas that can be served by the NF instances to be discovered. The NRF shall return NF profiles of NFs which can serve all the areas requested in this query parameter.	Query-Params-Ext2
imsi	string	O	0..1	If included, this IE shall contain the IMSI of the requester UE to search for an appropriate NF. IMSI may be included if the target NF type is "HSS". pattern: "[0-9]{5,15}"	Query-Params-Ext2
internal-group-identity	GroupId	O	0..1	If included, this IE shall contain the internal group identifier of the UE to search for an appropriate NF. This may be included if the target NF type is "UDM"	Query-Params-Ext2

preferred-api-versions	map(string)	O	1..N	When present, this IE indicates the preferred API version of the services that are supported by the target NF instances. The key of the map is the ServiceName (see clause 6.1.6.3.11) for which the preferred API version is indicated. Each element carries the API Version Indication for the service indicated by the key. An API Version Indication is a string formatted as {operator}+{API Version}. The following operators shall be supported: "=" match a version equals to the version value indicated. ">" match any version greater than the version value indicated ">=" match any version greater than or equal to the version value indicated "<" match any version less than the version value indicated "<=" match any version less than or equal to the version value indicated "^" match any version compatible with the version indicated, i.e. any version with the same major version as the version indicated. Precedence between versions is identified by comparing the Major, Minor, and Patch version fields numerically, from left to right. If no operator or an unknown operator is provided in API Version Indication, "=" operator is applied. Example of API Version Indication: Case1: "=1.2.4.operator-ext" or "1.2.4.operator-ext" means matching the service with API version "1.2.4.operator-ext" Case2: ">1.2.4" means matching the service with API versions greater than "1.2.4" Case3: "^2.3.0" or "^2" means matching the service with all API versions with major version "2".	Query-Params-Ext2
v2x-support-ind	boolean	O	0..1	When present, this IE indicates whether a PCF supporting V2X Policy/Parameter provisioning needs to be discovered. true: a PCF supporting V2X Policy/Parameter provisioning is requested to be discovered; false: a PCF not supporting V2X Policy/Parameter provisioning is requested to be discovered.	Query-Params-Ext2
redundant-gtpu	boolean	O	0..1	When present, this IE indicates whether a UPF supporting redundant GTP-U path needs to be discovered. true: a UPF supporting redundant GTP-U path is requested to be discovered; false: a UPF not supporting redundant GTP-U path is requested to be discovered.	Query-Params-Ext2
redundant-transport	boolean	O	0..1	When present, this IE indicates whether a UPF supporting redundant transport path on the transport layer in the corresponding network slice needs to be discovered. true: a UPF supporting redundant transport path on the transport layer is requested to be discovered; false: a UPF not supporting redundant transport path on the transport layer is requested to be discovered. If the Snssai(s) are also included, the UPF supporting redundant transport path on the transport layer shall be available in the network slice(s) identified by the Snssai(s).	Query-Params-Ext2

ipups	boolean	O	0..1	When present, this IE indicates whether a UPF which is configured for IPUPS is requested to be discovered. true: a UPF which is configured for IPUPS is requested to be discovered; false: a UPF which is not configured for IPUPS is requested to be discovered.	Query-Params-Ext2
scp-domain-list	array(string)	O	1..N	When present, this IE shall contain the SCP domain(s) the target NF or SCP belongs to. The NRF shall return NF or SCP profiles that belong to all the SCP domains provided in this list.	Query-Params-Ext2
address-domain	Fqdn	O	0..1	If included, this IE shall contain the address domain that shall be reachable through the SCP. This IE may be included when the target NF type is "SCP".	Query-Params-Ext2
ipv4-addr	Ipv4Addr	O	0..1	If included, this IE shall contain the IPv4 address that shall be reachable through the SCP. This IE may be included when the target NF type is "SCP".	Query-Params-Ext2
ipv6-prefix	Ipv6Prefix	O	0..1	If included, this IE shall contain the IPv6 prefix that shall be reachable through the SCP. This IE may be included when the target NF type is "SCP".	Query-Params-Ext2
served-nf-set-id	NfSetId	O	0..1	When present, this IE shall contain the NF Set ID that shall be reachable through the SCP. This IE may be included when the target NF type is "SCP".	Query-Params-Ext2
served-nf-type	NFType	O	0..1	When present, this IE shall contain the NF type of the NFs that shall be reachable through the SCP. This IE may be included when the target NF type is "SCP".	Query-Params-Ext2
remote-plmn-id	PlmnId	O	0..1	If included, this IE shall contain the remote PLMN ID that shall be reachable through the SCP. This IE may be included when the target NF type is "SCP".	Query-Params-Ext2
data-forwarding	boolean	O	0..1	This may be included if the target NF type is "UPF". (NOTE 13) When present, the IE indicates whether UPF(s) configured for data forwarding needs to be discovered. true: UPF(s) configured for data forwarding is requested to be discovered; false: UPF(s) not configured for data forwarding is requested to be discovered.	Query-Params-Ext2
preferred-full-plmn	boolean	O	0..1	When present, the NRF shall prefer NF profile(s) that can serve the full PLMN (i.e. can serve any TAI in the PLMN), or the NRF shall return other NF profiles if no NF profile serving the full PLMN is found: - true: NF instance(s) serving the full PLMN is preferred; - false: NF instance(s) serving the full PLMN is not preferred. (NOTE 14)	Query-Params-Ext2
requester-features	SupportedFeatures	C	0..1	Nnrf_NFDiscovery features supported by the NF Service Consumer that is invoking the Nnrf_NFDiscovery service. This IE shall be included if at least one feature is supported by the NF Service Consumer.	

NOTE 1: If this parameter is present and no AMF supporting the requested GUAMI is available due to AMF Failure or planned AMF removal, the NRF shall return in the response AMF instances acting as a backup for AMF failure or planned AMF removal respectively for this GUAMI (see clause 6.1.6.2.11). The NRF can detect if an AMF has failed, using the Heartbeat procedure. The NRF will receive a de-registration request from an AMF performing a planned removal.
NOTE 2: If the combined SMF/PGW-C is requested to be discovered, the NRF shall return in the response the SMF instances registered with the SmfInfo containing pgwFqdn.
NOTE 3: If a UPF supporting interworking with EPS is requested to be discovered, the NRF shall return in the response the UPF instances registered with the upfInfo containing iwkEpsInd set to true.
NOTE 4: This attribute has a different semantic than what is defined in clause 6.6.2 of 3GPP TS 29.500 [4], i.e. it is not used to signal optional features of the Nnrf_NFDiscovery Service API supported by the requester NF.
NOTE 5: The AMF may perform the SMF discovery based on the dnn, snssais and preferred-tai during a PDU session establishment procedure, and the NRF shall return the SMF profiles matching all if possible, or the SMF profiles only matching dnn and snssais. If the SMF profiles only matching dnn and snssais are returned, the AMF shall insert an I-SMF. An SMF may also perform a UPF discovery using this parameter.
NOTE 6: The SMF may select the P-CSCF close to the UPF by setting the preferred-locality to the value of the locality of the UPF.
NOTE 7: During EPS to 5GS idle mobility procedure, the NF service consumer (i.e. SMF) discovers the anchor NEF for NIDD using the SCEF ID received from EPS as the value of the NEF ID, as specified in clause 4.11.1.3.3 of 3GPP TS 23.502 [3].
NOTE 8: The service consumer may include a list of preferred-nf-instance-ids in the query. If so, the NRF shall first check if the NF profiles of the preferred NF instances match the other query parameters, and if so, then the NRF shall return the corresponding NF profiles; otherwise, the NRF shall return a list of candidate NF profiles matching the query parameters other than the preferred-nf-instance-ids. For example, the target AMF may set this query parameter to the SMF Instance ID and I-SMF Instance ID during an inter AMF mobility procedure to select an I-SMF.
NOTE 9: This parameter may be used by the SCP (with other query parameters) to discover and select a NF service consumer with a default notification subscription supporting the notication type of a notification request (see clause 6.10.3.x of 3GPP TS 29.500 [4]).
NOTE 10: An S-NSSAI value used in discovery request query parameters shall be considered as matching the S-NSSAI value in the NF Profile or NF Service of a given NF Instance if both the SST and SD components are identical (i.e. an S-NSSAI value where SD is absent, shall not be considered as matching an S-NSSAI where SD is present, regardless if SST is equal in both).
NOTE 11: The dnn query parameter shall be considered as matching a DNN attribute in the NF Profile of a given NF Instance if:
- both contain the same Network Identifier and Operator Identifier;
- both contain the same Network Identifier and none contains an Operator Identifier;
- the dnn query parameter contains the Network Identifier only, the DNN value in the NF Profile contains both the Network Identifier and Operator Identifier, and both contain the same Network Identifier; or
- the dnn query parameter contains both the Network Identifier and Operator Identifier, the DNN value in the NF Profile contains the Network Identifier only, both contain the same Network Identifier and the Operator Identifier matches one PLMN of the NF (i.e. plmnList of the NF Profile).

NOTE 12: Based on operator's policies, a discovery request not including the requester's information necessary to validate the authorization parameters in NF Profiles may be rejected or accepted but with only returning in the discovery response NF Instances whose authorization parameters allow any NF Service Consumer to access their services. The authorization parameters in NF Profile are those used by NRF to determine whether a given NF Instance / NF Service Instance can be discovered by an NF Service Consumer in order to consume its offered services (e.g. "allowedNfTypes", "allowedNfDomains", etc.).
NOTE 13: Different UPF instances for data forwarding may be configured in the network e.g. for different serving areas. The SMF may use this query parameter together with others (like SMF Serving Area or TAI) in discovery to select the UPF candidate for data forwarding.
NOTE 14: For HR roaming, if the V-PLMN requires Deployments Topologies with specific SMF Service Areas (DTSSA) but no H-SMF can be selected supporting V-SMF change, AMF may use this query parameter to select a V-SMF serving the full VPLMN if available.

The default logical relationship among the query parameters is logical "AND", i.e. all the provided query parameters shall be matched, with the exception of the "preferred-locality" or the "preferred-nf-instances" query (see Table 6.2.3.2.3.1-1).

The NRF may support the Complex query expression as defined in 3GPP TS 29.501 [5] for the NF Discovery service. If the "complexQuery" query parameter is included, then the logical relationship among the query parameters contained in "complexQuery" query parameter is as defined in 3GPP TS 29.571 [7].

A NRF not supporting Complex query expression shall reject a NF service discovery request including a complexQuery parameter, with a ProblemDetails IE including the cause attribute set to INVALID_QUERY_PARAM and the invalidParams attribute indicating the complexQuery parameter.

This method shall support the request data structures specified in table 6.1.3.2.3.1-2 and the response data structures and response codes specified in table 6.1.3.2.3.1-3.

Table 6.2.3.2.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.2.3.2.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
SearchResult	M	1	200 OK	The response body contains the result of the search over the list of registered NF Instances.
n/a			307 Temporary Redirect	The response shall be used when the intermediate NRF redirects the service discovery request. The NRF shall include in this response a Location header field containing a URI pointing to the resource located on the redirect target NRF.
ProblemDetails	O	0..1	400 Bad Request	The response body contains the error reason of the request message. If the query parameter used to match the authorization parameter is required but not provided in the NF discovery request, the "cause" attribute shall be set to "MANDATORY_QUERY_PARAM_MISSING", and the missing query parameter shall be indicated.
ProblemDetails	O	0..1	403 Forbidden	This response shall be returned if the NF Service Consumer is not allowed to discover the NF Service(s) being queried.
ProblemDetails	O	0..1	404 Not Found	This response shall be returned if the requested resource URI is not found in the server. It may also be sent in hierarchical NRF deployments when the NRF needs to forward/redirect the request to another NRF but lacks information in the request to do so; similarly, the NRF shall return this response code when it is received from the upstream NRF.
ProblemDetails	O	0..1	500 Internal Server Error	The response body contains the error reason of the request message.

Table 6.2.3.2.3.1-4: Headers supported by the GET method on this endpoint

Name	Data type	P	Cardinality	Description
If-None-Match	string	C	0..1	Validator for conditional requests, as described in IETF RFC 7232 [19], clause 3.2

Table 6.2.3.2.3.1-5: Headers supported by the 200 Response Code on this endpoint

Name	Data type	P	Cardinality	Description
Cache-Control	string	C	0..1	Cache-Control containing max-age, described in IETF RFC 7234 [20], clause 5.2
ETag	string	C	0..1	Entity Tag containing a strong validator, described in IETF RFC 7232 [19], clause 2.3

Table 6.2.3.2.3.1-6: Headers supported by the 307 Response Code on this endpoint

Name	Data type	P	Cardinality	Description
Location	string	M	1	The URI pointing to the resource located on the redirect target NRF

Table 6.2.3.2.3.1-7: Links supported by the 200 Response Code on this endpoint

Name	Resource name	HTTP method or custom operation	Parameters table	Description
search	Stored Search (Document)	GET	6.2.3.2.3.1-8	The 'searchId' parameter returned in the response can be used as the 'searchId' parameter in the GET request to '/searches/{searchId}'
completeSearch	Complete Stored Search (Document)	GET	6.2.3.2.3.1-9	The 'searchId' parameter returned in the response can be used as the 'searchId' parameter in the GET request to '/searches/{searchId}/complete'

6.2.3.2.4 Resource Custom Operations

There are no resource custom operations for the Nnrf_NFDiscovery service in this release of the specification.

6.2.3.3 Resource: Stored Search (Document)

6.2.3.3.1 Description

This resource represents a search result (i.e. a number of discovered NF Instances), stored by NRF as a consequence of a prior search result.

This resource is modelled as the Document resource archetype (see clause C.3 of 3GPP TS 29.501 [5]).

6.2.3.3.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-disc/v1/searches/{searchId}**

This resource shall support the resource URI variables defined in table 6.2.3.3.2-1.

Table 6.2.3.3.2-1: Resource URI variables for this resource

Name	Data type	Definition
apiRoot	string	See clause 6.1.1
searchId	string	Identifier of a stored search result, returned by NRF to the NF Consumer in the original response to the NF Discovery GET operation (see clause 6.2.6.2.2).

6.2.3.3.2.1 GET

This method retrieves the NF Instances corresponding to a given stored search result.

This method shall support the URI query parameters specified in table 6.2.3.3.2.1-1.

Table 6.2.3.3.2.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 6.2.3.3.2.1-2 and the response data structures and response codes specified in table 6.2.3.3.2.1-3.

Table 6.2.3.3.2.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.2.3.3.2.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
StoredSearchResult	M	1	200 OK	The response body contains the NF Instances corresponding to a given stored search result.
NOTE: The mandatory HTTP error status codes for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

6.2.3.4 Resource: Complete Stored Search (Document)

6.2.3.4.1 Description

This resource represents a complete search result (i.e. a number of discovered NF Instances), stored by NRF as a consequence of a prior search result, but without applying any client restrictions in terms of the number of instances to be returned (i.e. "limit" or "max-payload-size" query parameters).

This resource is modelled as the Document resource archetype (see clause C.3 of 3GPP TS 29.501 [5]).

6.2.3.4.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-disc/v1/searches/{searchId}/complete**

This resource shall support the resource URI variables defined in table 6.2.3.4.2-1.

Table 6.2.3.4.2-1: Resource URI variables for this resource

Name	Data type	Definition
apiRoot	string	See clause 6.1.1
searchId	string	Identifier of a stored search result, returned by NRF to the NF Consumer in the original response to the NF Discovery GET operation (see clause 6.2.6.2.2).

6.2.3.4.2.1 GET

This method retrieves the NF Instances corresponding to a given stored search result.

This method shall support the URI query parameters specified in table 6.2.3.4.2.1-1.

Table 6.2.3.4.2.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 6.2.3.4.2.1-2 and the response data structures and response codes specified in table 6.2.3.4.3.1-3.

Table 6.2.3.4.2.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.2.3.4.2.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
StoredSearchResult	M	1	200 OK	The response body contains the NF Instances corresponding to a given stored search result, but without applying any client restrictions in terms of the number of instances to be returned (i.e. "limit" or "max-payload-size" query parameters).
NOTE: The mandatory HTTP error status codes for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

6.2.4 Custom Operations without associated resources

There are no custom operations defined without any associated resources for the Nnrf_NFDiscovery service in this release of this specification.

6.2.5 Notifications

There are no notifications defined for the Nnrf_NFDiscovery service in this release of the specification.

6.2.6 Data Model

6.2.6.1 General

This clause specifies the application data model supported by the API.

Table 6.2.6.1-1 specifies the data types defined for the Nnrf service based interface protocol.

Table 6.2.6.1-1: Nnrf_NFDiscovery specific Data Types

Data type	Clause defined	Description
SearchResult	6.2.6.2.2	Contains the list of NF Profiles returned in a Discovery response.
NFProfile	6.2.6.2.3	Information of an NF Instance discovered by the NRF.
NFService	6.2.6.2.4	Information of a given NF Service Instance; it is part of the NFProfile of an NF Instance discovered by the NRF.
StoredSearchResult	6.2.6.2.5	Contains a complete search result (i.e. a number of discovered NF Instances), stored by NRF as a consequence of a prior search result.
PreferredSearch	6.2.6.2.6	Contains information on whether the returned NFProfiles match the preferred query parameters.

Table 6.2.6.1-2 specifies data types re-used by the Nnrf_NFDiscovery service-based interface protocol from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the Nnrf_NFDiscovery service-based interface.

Table 6.2.6.1-2: Nnrf_NFDiscovery re-used Data Types

Data type	Reference	Comments
Snssai	3GPP TS 29.571 [7]	
PlmnId	3GPP TS 29.571 [7]	
Dnn	3GPP TS 29.571 [7]	
Tai	3GPP TS 29.571 [7]	
SupportedFeatures	3GPP TS 29.571 [7]	
NfInstanceId	3GPP TS 29.571 [7]	
Uri	3GPP TS 29.571 [7]	
Gpsi	3GPP TS 29.571 [7]	
GroupId	3GPP TS 29.571 [7]	
Guami	3GPP TS 29.571 [7]	
IPv4Addr	3GPP TS 29.571 [7]	
IPv6Addr	3GPP TS 29.571 [7]	
UriScheme	3GPP TS 29.571 [7]	
Dnai	3GPP TS 29.571 [7]	
NfGroupId	3GPP TS 29.571 [7]	Identifier of a NF Group
PduSessionType	3GPP TS 29.571 [7]	
AtsssCapability	3GPP TS 29.571 [7]	
PlmnIdNid	3GPP TS 29.571 [7]	
NfSetId	3GPP TS 29.571 [7]	
NfServiceSetId	3GPP TS 29.571 [7]	
ExtSnssai	3GPP TS 29.571 [7]	
EventId	3GPP TS 29.520 [32]	Defined in Nnwdaf_AnalyticsInfo API.
NwdafEvent	3GPP TS 29.520 [32]	Defined in Nnwdaf_EventsSubscription API.
ExtGroupId	3GPP TS 29.503 [36]	
ExternalClientType	3GPP TS 29.572 [33]	
DefaultNotificationSubscription	3GPP TS 29.510	See clause 6.1.6.2.4
IPEndPoint	3GPP TS 29.510	See clause 6.1.6.2.5
NFType	3GPP TS 29.510	See clause 6.1.6.3.3
UdrInfo	3GPP TS 29.510	See clause 6.1.6.2.6
UdmInfo	3GPP TS 29.510	See clause 6.1.6.2.7
AusfInfo	3GPP TS 29.510	See clause 6.1.6.2.8
SupiRange	3GPP TS 29.510	See clause 6.1.6.2.9
AmfInfo	3GPP TS 29.510	See clause 6.1.6.2.11
SmfInfo	3GPP TS 29.510	See clause 6.1.6.2.12
UpfInfo	3GPP TS 29.510	See clause 6.1.6.2.13
PcfInfo	3GPP TS 29.510	See clause 6.1.6.2.20
BsfInfo	3GPP TS 29.510	See clause 6.1.6.2.21
ChfInfo	3GPP TS 29.510	See clause 6.1.6.2.32
ChfServiceInfo	3GPP TS 29.510	See clause 6.1.6.2.33
NFServiceVersion	3GPP TS 29.510	See clause 6.1.6.2.19
PlmnSnssai	3GPP TS 29.510	See clause 6.1.6.2.44
NwdafInfo	3GPP TS 29.510	See clause 6.1.6.2.45
NFStatus	3GPP TS 29.510	See clause 6.1.6.3.7
DataSetId	3GPP TS 29.510	See clause 6.1.6.3.8
ServiceName	3GPP TS 29.510	See clause 6.1.6.3.11
NFServiceStatus	3GPP TS 29.510	See clause 6.1.6.3.12
LmfInfo	3GPP TS 29.510	See clause 6.1.6.2.46
GmlcInfo	3GPP TS 29.510	See clause 6.1.6.2.47
NefInfo	3GPP TS 29.510	See clause 6.1.6.2.48
PfdData	3GPP TS 29.510	See clause 6.1.6.2.49
AfEventExposureData	3GPP TS 29.510	See clause 6.1.6.2.50
PcscfInfo	3GPP TS 29.510	See clause 6.1.6.2.53
HssInfo	3GPP TS 29.510	See clause 6.1.6.2.57
ImsiRange	3GPP TS 29.510	See clause 6.1.6.2.58
VendorSpecificFeature	3GPP TS 29.510	See clause 6.1.6.2.62
ScpInfo	3GPP TS 29.510	See clause 6.1.6.2.65
NefId	3GPP TS 29.510	See clause 6.1.6.3
VendorId	3GPP TS 29.510	See clause 6.1.6.3
AnNodeType	3GPP TS 29.510	See clause 6.1.6.3.13

6.2.6.2 Structured data types

6.2.6.2.1 Introduction

This clause defines the structures to be used in resource representations.

6.2.6.2.2 Type: SearchResult

Table 6.2.6.2.2-1: Definition of type SearchResult

Attribute name	Data type	P	Cardinality	Description
validityPeriod	integer	M	1	It shall contain the time in seconds during which the discovery result is considered valid and can be cached by the NF Service Consumer. This value shall be the same as the value contained in the "max-age" parameter of the "Cache-Control" header field sent in the HTTP response.
nfInstances	array(NFProfile)	M	0..N	It shall contain an array of NF Instance profiles, matching the search criteria indicated by the query parameters of the discovery request. An empty array means there is no NF instance that can match the search criteria.
searchId	string	O	0..1	This IE may be present if the NRF stores the result of the current service discovery response in a given URL (server-side caching), to make it available in the future to NF Service Consumers without having to compute the whole search process again.
numNfInstComplete	Uint32	O	0..1	This IE may be present when the total number of NF Instances found by NRF, as the result of the service discovery process, is higher than the actual number of NF Instances included in the attribute nfInstances of the SearchResult object. This may happen due to the NF Service Consumer including in the discovery request parameters such as "limit" or "max-payload-size".
preferredSearch	PreferredSearch	C	0..1	This IE shall be present to indicate whether the returned NFProfiles match the preferred query parameters, if the discovery request contain any of the query parameter defined in the PreferredSearch data type.
nrfSupportedFeatures	SupportedFeatures	C	0..1	Features supported by the NRF for the NFDiscovery service (see clause 6.2.9). This IE should be present if the NRF supports at least one feature.

6.2.6.2.3 Type: NFProfile

Table 6.2.6.2.3-1: Definition of type NFProfile

Attribute name	Data type	P	Cardinality	Description
nfInstanceId	NfInstanceId	M	1	Unique identity of the NF Instance.
nfType	NFType	M	1	Type of Network Function
nfStatus	NFStatus	M	1	Status of the NF Instance
nfInstanceName	string	O	0..1	Human readable name of the NF Instance
plmnList	array(PlmnId)	C	1..N	PLMN(s) of the Network Function (NOTE 5). This IE shall be present if this information is available for the NF. If this information was not provided by the NF during registration, the NRF should return the list of PLMN ID(s) of the PLMN of the NRF. If this IE is absent in the response, PLMN ID(s) of the PLMN of the NRF are assumed for the NF.
sNssais	array(ExtSnssai)	O	1..N	S-NSSAIs of the Network Function. If not provided, the NF can serve any S-NSSAI. If the sNSSAIs attribute is provided in at least one NF Service, the sNssais attribute in the NF Profile shall be present and be the set or a superset of the sNSSAIs of the NFService(s).
perPlmnSnssaiList	array(PlmnSnssai)	O	1..N	The per-PLMN list of S-NSSAI(s) supported by the Network Function. If the perPlmnSnssaiList attribute is provided in at least one NF Service, the perPlmnSnssaiList attribute in the NF Profile shall be present and be the set or a superset of the perPlmnSnssaiList of the NFService(s).
nsiList	array(string)	O	1..N	List of NSIs of the Network Function. If not provided, the NF can serve any NSI.
fqdn	Fqdn	C	0..1	FQDN of the Network Function (NOTE 1, NOTE 3)
ipv4Addresses	array(Ipv4Addr)	C	1..N	IPv4 address(es) of the Network Function (NOTE 1)
ipv6Addresses	array(Ipv6Addr)	C	1..N	IPv6 address(es) of the Network Function (NOTE 1)
capacity	integer	O	0..1	Static capacity information within the range 0 to 65535, expressed as a weight relative to other NF instances of the same type; if capacity is also present in the nfServiceList parameters, those will have precedence over this value. (See NOTE 2)
load	integer	O	0..1	Latest known load information of the NF within the range 0 to 100 in percentage (See NOTE 4)
loadTimeStamp	DateTime	O	0..1	It indicates the point in time in which the latest load information of the NF Instance was sent from the NF to the NRF.
locality	string	O	0..1	Operator defined information about the location of the NF instance (e.g. geographic location, data center)
priority	integer	O	0..1	Priority (relative to other NFs of the same type) within the range 0 to 65535, to be used for NF selection; lower values indicate a higher priority. Priority may or may not be present in the nfServiceList parameters, xxxInfo parameters and in this attribute. Priority in the nfServiceList has precedence over the priority in this attribute, which has precedence over the priority in xxxInfo parameter. (NOTE 2)
udrInfo	UdrInfo	O	0..1	Specific data for the UDR (ranges of SUPI, …)
udrInfoList	map(UdrInfo)	O	1..N	Multiple entries of UdrInfo. This attribute provides additional information to the udrInfo. udrInfoList may be present even if the udrInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
udmInfo	UdmInfo	O	0..1	Specific data for the UDM
udmInfoList	map(UdmInfo)	O	1..N	Multiple entries of UdmInfo. This attribute provides additional information to the udmInfo. udmInfoList may be present even if the udmInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.

ausfInfo	AusfInfo	O	0..1	Specific data for the AUSF
ausfInfoList	map(AusfInfo)	O	1..N	Multiple entries of AusfInfo. This attribute provides additional information to the ausfInfo. ausfInfoExt may be present even if the ausfInfo is absent.
amfInfo	AmfInfo	O	0..1	Specific data for the AMF (AMF Set ID, …)
amfInfoList	map(AmfInfo)	O	1..N	Multiple entries of AmfInfo. This attribute provides additional information to the amfInfo. amfInfoList may be present even if the amfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
smfInfo	SmfInfo	O	0..1	Specific data for the SMF (DNN's, …). (NOTE 8)
smfInfoList	map(SmfInfo)	O	1..N	Multiple entries of SmfInfo. This attribute provides additional information to the smfInfo. smfInfoList may be present even if the smfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters. (NOTE 8)
upfInfo	UpfInfo	O	0..1	Specific data for the UPF (S-NSSAI, DNN, SMF serving area, …)
upfInfoList	map(UpfInfo)	O	1..N	Multiple entries of UpfInfo. This attribute provides additional information to the upfInfo. upfInfoList may be present even if the upfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
pcfInfo	PcfInfo	O	0..1	Specific data for the PCF
pcfInfoList	map(PcfInfo)	O	1..N	Multiple entries of PcfInfo. This attribute provides additional information to the pcfInfo. pcfInfoList may be present even if the pcfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
bsfInfo	BsfInfo	O	0..1	Specific data for the BSF
bsfInfoList	map(BsfInfo)	O	1..N	Multiple entries of BsfInfo. This attribute provides additional information to the bsfInfo. bsfInfoList may be present even if the bsfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
chfInfo	ChfInfo	O	0..1	Specific data for the CHF
chfInfoList	map(ChfInfo)	O	1..N	Multiple entries of ChfInfo. This attribute provides additional information to the chfInfo. chfInfoList may be present even if the chfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
udsfInfo	UdsfInfo	O	0..1	Specific data for the UDSF
udsfInfoList	map(UdsfInfo)	O	1..N	Multiple entries of udsfInfo. This attribute provides additional information to the udsfInfo. udsfInfoList may be present even if the udsfInfo is absent. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.
nefInfo	NefInfo	O	0..1	Specific data for the NEF
nwdafInfo	NwdafInfo	O	0..1	Specific data for the NWDAF
pcscfInfoList	map(PcscfInfo)	O	1..N	Specific data for the P-CSCF. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters. (NOTE 7)
hssInfoList	map(HssInfo)	O	1..N	Specific data for the HSS. The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [22], with a maximum of 32 characters.

customInfo	object	O	0..1	Specific data for custom Network Functions
recoveryTime	DateTime	O	0..1	Timestamp when the NF was (re)started
nfServicePersistence	boolean	O	0..1	- true: If present, and set to true, it indicates that the different service instances of a same NF Service in the NF instance, supporting a same API version, are capable to persist their resource state in shared storage and therefore these resources are available after a new NF service instance supporting the same API version is selected by a NF Service Consumer (see 3GPP 23.527 [27]). - false (default): Otherwise, it indicates that the NF Service Instances of a same NF Service are not capable to share resource state inside the NF Instance.
nfServices	array(NFService)	O	1..N	List of NF Service Instances. (NOTE 10) This attribute is deprecated; the attribute "nfServiceList" should be used instead.
nfServiceList	map(NFService)	O	1..N	Map of NF Service Instances, where the "serviceInstanceId" attribute of the NFService object shall be used as the key of the map. (NOTE 10)
defaultNotificationSubscriptions	array(DefaultNotificationSubscription)	O	1..N	Notification endpoints for different notification types. (NOTE 6)
lmfInfo	LmfInfo	O	0..1	Specific data for the LMF
gmlcInfo	GmlcInfo	O	0..1	Specific data for the GMLC
snpnList	array(PlmnIdNid)	C	1..N	SNPN(s) of the Network Function. This IE shall be present if the NF pertains to one or more SNPNs.
nfSetIdList	array(NfSetId)	C	1..N	NF Set ID defined in clause 28.12 of 3GPP TS 23.003 [12]. At most one NF Set ID shall be indicated per PLMN of the NF. This information shall be present if available.
servingScope	array(string)	O	1..N	The served area(s) of the NF instance. The absence of this attribute does not imply the NF instance can serve every area.
lcHSupportInd	boolean	O	0..1	This IE indicates whether the NF supports Load Control based on LCI Header (see clause 6.3 of 3GPP TS 29.500 [4]). - true: the NF supports the feature. - false (default): the NF does not support the feature.
olcHSupportInd	boolean	O	0..1	This IE indicates whether the NF supports Overload Control based on OCI Header (see clause 6.4 of 3GPP TS 29.500 [4]). - true: the NF supports the feature. - false (default): the NF does not support the feature.
nfSetRecoveryTimeList	map(DateTime)	O	1..N	Map of recovery time, where the key of the map is the NfSetId of NF Set(s) that the NF instance belongs to. When present, the value of each entry of the map shall be the recovery time of the NF Set indicated by the key.
serviceSetRecoveryTimeList	map(DateTime)	O	1..N	Map of recovery time, where the key of the map is the NfServiceSetId of the NF Service Set(s) configured in the NF instance. When present, the value of each entry of the map shall be the recovery time of the NF Service Set indicated by the key.
scpDomains	array(string)	O	1..N	List of SCP domains the NF belongs to. (NOTE 9)

scpInfo	ScpInfo	O	0..1	Specific data for the SCP
NOTE 1: At least one of the addressing parameters (fqdn, ipv4address or ipv6adress) shall be included in the NF Profile. See NOTE 1 of Table 6.2.6.2.4-1 for the use of these parameters. If multiple ipv4 addresses and/or ipv6 addresses are included in the NF Profile, the NF Service Consumer shall select one of these addresses randomly, unless operator defined local policy of IP address selection, in order to avoid overload for a specific ipv4 address and/or ipv6 address.				
NOTE 2: The capacity and priority parameters, if present, are used for NF selection and load balancing. The priority and capacity attributes shall be used for NF selection in the same way that priority and weight are used for server selection as defined in IETF RFC 2782 [23].				
NOTE 3: If the requester-plmn in the query parameter is different from the PLMN of the discovered NF, then the fqdn attribute value shall contain the interPlmnFqdn value registered by the NF during NF registration (see clause 6.1.6.2.2). The requester-plmn is different from the PLMN of the discovered NF if it belongs to none of the PLMN ID(s) configured for the PLMN of the NRF.				
NOTE 4: The usage of the load parameter by the NF service consumer is implementation specific, e.g. be used for NF selection and load balancing, together with other parameters.				
NOTE 5: An NF may register multiple PLMN IDs in its profile within a PLMN comprising multiple PLMN IDs. If so, all the attributes of the NF Profile shall apply to each PLMN ID registered in the plmnList. As an exception, attributes including a PLMN ID, e.g. IMSI-based SUPI ranges, TAIs and GUAMIs, are specific to one PLMN ID and the NF may register in its profile multiple occurrences of such attributes for different PLMN IDs (e.g. the UDM may register in its profile SUPI ranges for different PLMN IDs).				
NOTE 6: If notification endpoints are present both in the profile of the NF instance (NFProfile) and in some of its NF Services (NFService) for a same notification type, the notification endpoint(s) of the NF Services shall be used for this notification type.				
NOTE 7: The absence of the pcscfInfoList attribute in a P-CSCF profile indicates that the P-CSCF can be selected for any DNN and Access Type.				
NOTE 8: The absence of both the smfInfo and smfInfoList attributes in an SMF profile indicates that the SMF can be selected for any S-NSSAI, DNN, TAI and access type.				
NOTE 9: If an NF includes this information in its profile, this indicates that the services produced by this NF should be accessed preferably via an SCP from an SCP domain to which the NF belongs to.				
NOTE 10: If the NF Service Consumer that issued the discovery request indicated support for the "Service-Map" feature, the NRF shall return in the discovery response the list of NF Service Instances in the "nfServiceList" map attribute. Otherwise, the NRF shall return the list of NF Service Instances in the "nfServices" array attribute.				

6.2.6.2.4 Type: NFService

Table 6.2.6.2.4-1: Definition of type NFService

Attribute name	Data type	P	Cardinality	Description
serviceInstanceId	string	M	1	Unique ID of the service instance within a given NF Instance
serviceName	ServiceName	M	1	Name of the service instance (e.g. "udm-sdm")
versions	array(NFServiceVersion)	M	1..N	The API versions supported by the NF Service and if available, the corresponding retirement date of the NF Service. The different array elements shall have distinct unique values for "apiVersionInUri", and consequently, the values of "apiFullVersion" shall have a unique first digit version number.
scheme	UriScheme	M	1	URI scheme (e.g. "http", "https")
nfServiceStatus	NFServiceStatus	M	1	Status of the NF Service Instance
fqdn	Fqdn	O	0..1	FQDN of the NF Service Instance (see NOTE 1, NOTE 3)
ipEndPoints	array(IpEndPoint)	O	1..N	IP address(es) and port information of the Network Function (including IPv4 and/or IPv6 address) where the service is listening for incoming service requests (see NOTE 1, NOTE 5, NOTE 6)
apiPrefix	string	O	0..1	Optional path segment(s) used to construct the {apiRoot} variable of the different API URIs, as described in 3GPP 29.501 [5], clause 4.4.1 (optional deployment-specific string that starts with a "/" character)
defaultNotificationSubscriptions	array(DefaultNotificationSubscription)	O	1..N	Notification endpoints for different notification types.
capacity	integer	O	0..1	Static capacity information within the range 0 to 65535, expressed as a weight relative to other services of the same type. (See NOTE 2)
load	integer	O	0..1	Latest known load information of the NF Service, within the range 0 to 100 in percentage. (See NOTE 4)
loadTimeStamp	DateTime	O	0..1	It indicates the point in time in which the latest load information of the NF Service Instance was sent from the NF to the NRF.
priority	integer	O	0..1	Priority (relative to other services of the same type) within the range 0 to 65535, to be used for NF Service selection; lower values indicate a higher priority. (See NOTE 2)
recoveryTime	DateTime	O	0..1	Timestamp when the NF service was (re)started
chfServiceInfo	ChfServiceInfo	O	0..1	Specific data for the CHF service instance
supportedFeatures	SupportedFeatures	O	0..1	Supported Features of the NF Service instance
nfServiceSetIdList	array(NfServiceSetId)	C	1..N	NF Service Set ID (see clause 28.11 of 3GPP TS 23.003 [12]) At most one NF Service Set ID shall be indicated per PLMN of the NF. This information shall be present if available.
sNssais	array(ExtSnssai)	O	1..N	S-NSSAIs of the NF Service. This may be a subset of the S-NSSAIs supported by the NF (see sNssais attribute in NFProfile). When present, this IE represents the list of S-NSSAIs supported by the NF Service in all the PLMNs listed in the plmnList IE.
perPlmnSnssaiList	array(PlmnSnssai)	O	1..N	S-NSSAIs of the NF Service per PLMN. This may be a subset of the S-NSSAIs supported per PLMN by the NF (see perPlmnSnssaiList attribute in NFProfile). This IE may be included when the list of S-NSSAIs supported by the NF Service for each PLMN it is supporting is different. When present, this IE shall include the S-NSSAIs supported by the NF Service for each PLMN. When present, this IE shall override the sNssais IE.
vendorId	VendorId	O	0..1	Vendor ID of the NF Service instance, according to the IANA-assigned "SMI Network Management Private Enterprise Codes" [38].

supportedVendorSpecificFeatures	map(array(VendorSpecificFeature)	O	1..N	Map of Vendor-Specific features, where the key of the map is the IANA-assigned "SMI Network Management Private Enterprise Codes" [38]. The value of each entry of the map shall be a list (array) of VendorSpecificFeature objects. (NOTE 7)
oauth2Required	boolean	O	0..1	It indicates whether the NF Instance requires Oauth2-based authorization. Absence of this IE means that the NF Service Producer has not provided any indication about its usage of Oauth2 for authorization.
allowedOperationsPerNfType	map(array(string))	O	1..N	Map of allowed operations on resources for each type of NF; the key of the map is the NF Type, and the value is an array of scopes. The scopes shall be any of those defined in the API that defines the current service (identified by the "serviceName" attribute). (NOTE 8)
allowedOperationsPerNfInstance	map(array(string))	O	1..N	Map of allowed operations on resources for a given NF Instance; the key of the map is the NF Instance Id, and the value is an array of scopes. The scopes shall be any of those defined in the API that defines the current service (identified by the "serviceName" attribute). (NOTE 8)

NOTE 1: The NF Service Consumer shall construct the API URIs of the service using:
- for intra-PLMN signalling: the FQDN and IP addresses related attributes present in the NF Service Profile, if any, otherwise the FQDN and IP addresses related attributes present in the NF Profile.
- for inter-PLMN signalling: the FQDN present in the NF Service Profile, if any, otherwise the FQDN present in the NF Profile (see NOTE 3).

NOTE 2: The capacity and priority parameters, if present, are used for service selection and load balancing. The priority and capacity attributes shall be used for NF selection in the same way that priority and weight are used for server selection as defined in IETF RFC 2782 [23].

NOTE 3: If the requester-plmn in the query parameter is different from the PLMN of the discovered NF Service, then the fqdn attribute value, if included shall contain the interPlmnFqdn value registered by the NF Service during NF registration (see clause 6.1.6.2.3). The requester-plmn is different from the PLMN of the discovered NF Service if it belongs to none of the PLMN ID(s) configured for the PLMN of the NRF.

NOTE 4: The usage of the load parameter by the NF service consumer is implementation specific, e.g. be used for NF service selection and load balancing, together with other parameters.

NOTE 5: If the ipEndPoints attribute is absent in the NF Service and NF Profile, the NF service consumer shall use the fqdn attribute value for DNS query and if the NF service consumer does not receive a port number during the DNS query it shall use the default HTTP port number, i.e. TCP port 80 for "http" URIs or TCP port 443 for "https" URIs as specified in IETF RFC 7540 [9] when invoking the service.

NOTE 6: If multiple ipv4 addresses and/or ipv6 addresses are included in the NF Service, the NF Service Consumer shall select one of these addresses randomly, unless operator defined local policy of IP address selection, in order to avoid overload for a specific ipv4 address and/or ipv6 address.

NOTE 7: When present, this attribute allows the NF Service Consumer to determine which vendor-specific extensions are supported in a given NF Service Producer in order to include, or not, the vendor-specific attributes (see 3GPP TS 29.500 [4] clause 6.6.3) required for a given feature in subsequent service requests towards a certain service instance of the NF Service Producer.

NOTE 8: These attributes are used by the NF Service Consumer in order to discover the additional scopes (resource/operation-level scopes) that might be required to invoke a certain service operation, based on the authorization information registered in NRF by the NF Service Producer in its NF profile.

6.2.6.2.5 Type: StoredSearchResult

Table 6.2.6.2.5-1: Definition of type StoredSearchResult

Attribute name	Data type	P	Cardinality	Description
nfInstances	array(NFProfile)	M	0..N	An array of NF Instances corresponding to a given stored search result.

6.2.6.2.6 Type: PreferredSearch

Table 6.2.6.2.6-1: Definition of type PreferredSearch

Attribute name	Data type	P	Cardinality	Description
preferredTaiMatchInd	boolean	C	0..1	Indicates whether the returned NFProfiles match the query parameter preferred-tai. true: Match false (default): Not Match
preferredFullPlmnMatchInd	boolean	O	0..1	Indicates whether the returned NFProfiles match the query parameter preferred-full-plmn. true: Match false (default): Not Match

6.2.6.3 Simple data types and enumerations

6.2.6.3.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

6.2.6.3.2 Simple data types

The simple data types defined in table 6.2.6.3.2-1 shall be supported.

Table 6.2.6.3.2-1: Simple data types

Type Name	Type Definition	Description

6.2.7 Error Handling

6.2.7.1 General

HTTP error handling shall be supported as specified in clause 5.2.4 of 3GPP TS 29.500 [4].

6.2.7.2 Protocol Errors

Protocol errors handling shall be supported as specified in clause 5.2.7 of 3GPP TS 29.500 [4].

6.2.7.3 Application Errors

The application errors defined for the Nnrf_NFDiscovery service are listed in Table 6.2.7.3-1.

Table 6.2.7.3-1: Application errors

Application Error	HTTP status code	Description

6.2.8 Security

As indicated in 3GPP TS 33.501 [15], the access to the Nnrf_NFDiscovery API may be authorized by means of the OAuth2 protocol (see IETF RFC 6749 [16]), using the "Client Credentials" authorization grant, where the NRF plays the role of the authorization server.

If Oauth2 authorization is used, an NF Service Consumer, prior to consuming services offered by the Nnrf_NFDiscovery API, shall obtain a "token" from the authorization server, by invoking the Access Token Request service, as described in clause 5.4.2.2.

NOTE: When multiple NRFs are deployed in a network, the NRF used as authorization server is the same NRF where the Nnrf_NFDiscovery service is invoked by the NF Service Consumer.

The Nnrf_NFDiscovery API defines scopes for OAuth2 authorization as specified in 3GPP TS 33.501 [15]; it defines a single scope consisting on the name of the service (i.e., "nnrf-disc"), and it does not define any additional scopes at resource or operation level.

6.2.9 Features supported by the NFDiscovery service

The syntax of the supportedFeatures attribute is defined in clause 5.2.2 of 3GPP TS 29.571 [7].

The following features are defined for the Nnrf_NFDiscovery service.

Table 6.2.9-1: Features of supportedFeatures attribute used by Nnrf_NFDiscovery service

Feature Number	Feature	M/O	Description
1	Complex-Query	O	Support of Complex Query expression (see clause 6.2.3.2.3.1)
2	Query-Params-Ext1	O	Support of the following query parameters: - limit - max-payload-size - required-features - pdu-session-types
3	Query-Param-Analytics	O	Support of the query parameters for Analytics identifier: - event-id-list - nwdaf-event-list
4	MAPDU	O	This feature indicates whether the NRF supports selection of UPF with ATSSS capability.
5	Query-Params-Ext2	O	Support of the following query parameters: - requester-nf-instance-id - upf-ue-ip-addr-ind - pfd-data - target-snpn - af-ee-data - w-agf-info - tngf-info - twif-info - target-nf-set-id - target-nf-service-set-id - preferred-tai - nef-id - preferred-nf-instances - notification-type - serving-scope - internal-group-identity - preferred-api-versions - v2x-support-ind - redundant-gtpu - redundant-transport - lmf-id - an-node-type - rat-type - ipups - scp-domain-list - address-domain - ipv4-addr - ipv6-prefix - served-nf-set-id - served-nf-type - remote-plmn-id - data-forwarding - preferred-full-plmn - requester-snpn-list - max-payload-size-ext
6	Service-Map	M	This feature indicates whether it is supported to identify the list of NF Service Instances as a map (i.e. the "nfServiceList" attribute of NFProfile is supported).
Feature number: The order number of the feature within the supportedFeatures attribute (starting with 1). Feature: A short name that can be used to refer to the bit and to the feature. M/O: Defines if the implementation of the feature is mandatory ("M") or optional ("O"). Description: A clear textual description of the feature.			

6.3 Nnrf_AccessToken Service API

6.3.1 General

This API reuses the API endpoints and input / output parameters specified in IETF RFC 6749 [16] as a custom operation without resources. Hence this clause does not follow the 3GPP API specification guidelines described in 3GPP TS 29.501 [5].

6.3.2 API URI

URIs of this API shall have the following root:

{nrfApiRoot}/oauth2/

where {nrfApiRoot} represents the concatenation of the "scheme" and "authority" components of the NRF, as defined in IETF RFC 3986 [17].

6.3.3 Usage of HTTP

6.3.3.1 General

HTTP/2, as defined in IETF RFC 7540 [9], shall be used as specified in clause 5 of 3GPP TS 29.500 [4].

HTTP/2 shall be transported as specified in clause 5.3 of 3GPP TS 29.500 [4].

HTTP messages and bodies this API shall comply with the OpenAPI [10] specification contained in Annex A.

6.3.3.2 HTTP standard headers

6.3.3.2.1 General

The HTTP headers as specified in clause 4.4 of IETF RFC 6749 [16] shall be supported, with the exception that there shall not be "Authorization" HTTP request header in the access token request.

6.3.3.2.2 Content type

The following content types shall be supported:

- The x-www-form-urlencoded format (see clause 17.13.4 of W3C HTML 4.01 Specification [26]). The use of the x-www-form-urlencoded format shall be signalled by the content type "application/x-www-form-urlencoded".
- The JSON format (IETF RFC 8259 [22]). The use of the JSON format shall be signalled by the content type "application/json". See also clause 5.4 of 3GPP TS 29.500 [4].

6.3.3.3 HTTP custom headers

6.3.3.3.1 General

In this release of this specification, no custom headers specific to the OAuth2.0 Authorization Service API are defined. For 3GPP specific HTTP custom headers used across all service-based interfaces, see clause 5.2.3 of 3GPP TS 29.500 [4].

6.3.4 Custom Operations without associated resources

6.3.4.1 Overview

The /token endpoint as specified in IETF RFC 6749 [16] shall be supported. The "token endpoint" URI shall be:

{nrfApiRoot}/oauth2/token

where {nrfApiRoot} is defined in clause 6.3.2.

Table 6.3.4.1-1 provides an overview of the endpoints and applicable HTTP methods.

Table 6.3.4.1-1: Custom operations without associated resources

Operation Name	Custom operation URI	Mapped HTTP method	Description
Get (Access Token Request)	/oauth2/token	POST	Access token request for obtaining OAuth2.0 access token. This operation maps to Nnrf_AccessToken_Get service operation.

6.3.4.2 Operation: Get (Access Token Request)

6.3.4.2.1 Description

This custom operation represents the process for issuing the OAuth2.0 access token.

6.3.4.2.2 Operation Definition

This operation returns an OAuth 2.0 access token based on the input parameters provided. This custom operation shall use the HTTP POST method.

This method shall support the request data structures specified in table 6.3.4.2.2-1 and the response data structures and response codes specified in table 6.3.4.2.2-2. The data structure used for the POST request body shall be using x-www-form-urlencoded format as specified in clause 17.13.4 of W3C HTML 4.01 Specification [26].

Table 6.3.4.2.2-1: Data structures supported by the POST Request Body on this endpoint

Data type	P	Cardinality	Description
AccessTokenReq	M	1	This IE shall contain the request information for the access token request. Content-Type: "application/x-www-form-urlencoded"

Table 6.3.4.2.2-2: Data structures supported by the POST Response Body on this endpoint

Data type	P	Cardinality	Response codes	Description
AccessTokenRsp	M	1	200 OK	This IE shall contain the access token response information.
AccessTokenErr	M	1	400 Bad Request	See IETF RFC 6749 [16] clause 5.2.

Table 6.3.4.2.2-3: Headers supported by the 200 Response Code on this endpoint

Name	Data type	P	Cardinality	Description
Cache-Control	string	M	1	Enum: "no-store"
Pragma	string	M	1	Enum: "no-cache"

Table 6.3.4.2.2-4: Headers supported by the 400 Response Code on this endpoint

Name	Data type	P	Cardinality	Description
Cache-Control	string	M	1	Enum: "no-store"
Pragma	string	M	1	Enum: "no-cache"

6.3.5 Data Model

6.3.5.1 General

This clause specifies the application data model supported by the API.

Table 6.3.5.1-1 specifies the data types defined for the OAuth 2.0 Authorization Service API. The AccessTokenReq data structure shall be converted to the content type "application/x-www-form-urlencoded" when the OAuth 2.0 Access Token Request is invoked.

Table 6.3.5.1-1: OAuth 2.0 Authorization service specific Data Types

Data type	Clause defined	Description
AccessTokenReq	6.3.5.2.2	Contains information related to the access token request.
AccessTokenRsp	6.3.5.2.3	Contains information related to the access token response.
AccessTokenClaims	6.3.5.2.4	The claims data structure for the access token.
Audience	6.3.5.4.1	Contains the audience claim of the access token.

Table 6.3.5.1-2 specifies data types re-used by the OAuth 2.0 Authorization service from other specifications, including a reference to their respective specifications and when needed, a short description of their use.

Table 6.3.5.1-2: OAuth 2.0 Authorization service re-used Data Types

Data type	Reference	Comments
NfInstanceId	3GPP TS 29.571 [7]	
PlmnId	3GPP TS 29.571 [7]	PLMN ID
NFType	3GPP TS 29.510	See clause 6.1.6.3.3
Snssai	3GPP TS 29.571 [7]	
NfSetId	3GPP TS 29.571 [7]	NF Set ID (see clause 28.12 of 3GPP TS 23.003 [12])

6.3.5.2 Structured data types

6.3.5.2.1 Introduction

This clause defines the structures to be used in the APIs.

6.3.5.2.2 Type: AccessTokenReq

Table 6.3.5.2.2-1: Definition of type AccessTokenReq

Attribute name	Data type	P	Cardinality	Description
grant_type	string	M	1	This IE shall contain the grant type as "client_credentials". Enum: "client_credentials"
nfInstanceId	NfInstanceId	M	1	This IE shall contain the NF instance id of the NF service consumer.
nfType	NFType	C	0..1	This IE shall be included when the access token request is for an NF type and not for a specific NF / NF service instance. When present, this IE shall contain the NF type of the NF service consumer. (NOTE 3)
targetNfType	NFType	C	0..1	This IE shall be included when the access token request is for an NF type and not for a specific NF / NF service instance. When present, this IE shall contain the NF type of the NF service producer.
scope	string	M	1	This IE shall contain the scopes requested by the NF service consumer. The scopes shall consist of a list of NF service name(s) of the NF service producer(s) or resource/operation-level scopes defined by each service API, separated by whitespaces, as described in IETF RFC 6749 [16], clause 3.3. The service name(s) included in this attribute shall be any of the services defined in the ServiceName enumerated type (see clause 6.1.6.3.11). The resource/operation-level scopes shall be any of those defined in the "securitySchemes" clause of each service API. pattern: '^([a-zA-Z0-9_:-]+)([a-zA-Z0-9_:-]+)*$' See NOTE 2.
targetNfInstanceId	NfInstanceId	C	0..1	This IE shall be included, if available and if it is an access token request for a specific NF Service Producer. When present this IE shall contain the NF Instance ID of the specific NF Service Producer for which the access token is requested.
requesterPlmn	PlmnId	C	0..1	This IE shall be included when the NF service consumer in one PLMN requests a service access authorization for an NF service producer from a different PLMN. When present, this IE shall contain the PLMN ID of the requester NF service consumer. (NOTE 3) (NOTE 4)
requesterPlmnList	array(PlmnId)	C	2..N	This IE shall be included when the NF service consumer serving a PLMN, with more than one PLMN ID, requests a service access authorization for an NF service producer from a different PLMN. When present, this IE shall contain the PLMN IDs of the requester NF service consumer. (NOTE 4)
requesterSnssaiList	array(Snssai)	O	1..N	When present, this IE shall contain the list of S-NSSAIs of the requester NF service consumer. This may be used by the NRF to validate that the requester NF service consumer is allowed to access the target NF Service Producer. (NOTE 3)
requesterFqdn	Fqdn	O	0..1	When present, this IE shall contain the FQDN of the requester NF Service Consumer. This may be used by the NRF to validate that the requester NF service consumer is allowed to access the target NF Service Producer. (NOTE 3)

requesterSnpnList	array(PlmnIdNid)	O	1..N	When present, this IE shall contain the list of SNPNs the requester NF service consumer belongs to. This may be used by the NRF to validate that the requester NF service consumer is allowed to access the target NF Service Producer. (NOTE 3)
targetPlmn	PlmnId	C	0..1	This IE shall be included when the NF service consumer in one PLMN requests a service access authorization for an NF service producer from a different PLMN. When present, this IE shall contain the PLMN ID of the target PLMN (i.e., PLMN ID of the NF service producer).
targetSnssaiList	array(Snssai)	O	1..N	This IE may be included during an access token request for an NF type and not for a specific NF / NF service instance. When present, this IE shall contain the list of S-NSSAIs of the NF Service Producer.
targetNsiList	array(string)	O	1..N	This IE may be included during an access token request for an NF type and not for a specific NF / NF service instance. When present, this IE shall contain the list of NSIs of the NF Service Producer.
targetNfSetId	NfSetId	O	0..1	This IE may be included during an access token request for an NF type and not for a specific NF / NF service instance. When present, this IE shall contain the NF Set ID of the NF Service Producer.
targetNfServiceSetId	NfServiceSetId	O	0..1	This IE may be included during an access token request for a specific NF / NF service instance. When present, this IE shall contain the NF Service Set ID of the NF Service Producer. This may be used by the NRF to validate that the requester NF service consumer is allowed to access the target NF service instance. (NOTE 3)

NOTE 1: This data structure shall not be treated as a JSON object. It shall be treated as a key, value pair data structure to be encoded using x-www-form-urlencoded format as specified in clause 17.13.4 of W3C HTML 4.01 Specification [26].

NOTE 2: Though scope attribute is optional as per IETF RFC 6749 [16], it is mandatory for 3GPP as per 3GPP TS 33.501 [15].

NOTE 3: An access token request should be rejected if the requester NF is not allowed to access the target NF based on the authorization parameters in the NF profile of the target NF. The authorization parameters in NF Profile are those used by NRF to determine whether a given NF Instance / NF Service Instance can be discovered by an NF Service Consumer in order to consume its offered services (e.g. "allowedNfTypes", "allowedNfDomains", etc.). Based on operator's policies, an access token request not including the requester's information necessary to validate the authorization parameters in the target NF Profile may be rejected.

NOTE 4: When the NF service consumer is serving a PLMN consisting of one PLMN ID, the attribute "requesterPlmn" shall be used; otherwise, if the NF service consumer is serving a PLMN consisting of more than one PLMN ID, the attribute "requesterPlmnList" shall be used.

6.3.5.2.3 Type: AccessTokenRsp

Table 6.3.5.2.3-1: Definition of type AccessTokenRsp

Attribute name	Data type	P	Cardinality	Description
access_token	string	M	1	This IE shall contain JWS Compact Serialized representation of the JWS signed JSON object containing AccessTokenClaims (see clause 6.3.5.2.4).
token_type	string	M	1	This IE shall contain the token type, set to value "Bearer". Enum: "Bearer"
expires_in	integer	C	0..1	This IE when present shall contain the number of seconds after which the access token is considered to be expired. As indicated in IETF RFC 6749 [16], this attribute should be included, unless the expiration time of the token is made available by other means (e.g. deployment-specific documentation).
scope	string	C	0..1	This IE when present shall contain the scopes granted to the NF service consumer. The scopes shall consist of a list of NF service name(s) of the NF service producer(s) or resource/operation-level scopes defined by each service API, separated by whitespaces, as described in IETF RFC 6749 [16], clause 3.3. The service name(s) included in this attribute shall be any of the services defined in the ServiceName enumerated type (see clause 6.1.6.3.11). The resource/operation-level scopes shall be any of those defined in the "securitySchemes" clause of each service API. As indicated in IETF RFC 6749 [16], this attribute shall be present if it is different than the scope included in the access token request; if it is the same as the requested scope, this attribute may be absent. pattern: '^([a-zA-Z0-9_:-]+)([a-zA-Z0-9_:-]+)*$'

6.3.5.2.4 Type: AccessTokenClaims

Table 6.3.5.2.4-1: Definition of type AccessTokenClaims

Attribute name	Data type	P	Cardinality	Description
iss	NfInstanceId	M	1	This IE shall contain NF instance id of the NRF. , corresponding to the standard "Issuer" claim described in IETF RFC 7519 [25], clause 4.1.1
sub	NfInstanceId	M	1	This IE shall contain the NF instance ID of the NF service consumer, corresponding to the standard "Subject" claim described in IETF RFC 7519 [25], clause 4.1.2.
aud	Audience	M	1	This IE shall contain the NF service producer's NF instance ID(s) (if the exact NF instance(s) of the NF service producer is known) or the NF type of NF service producers for which the claim is applicable, corresponding to the standard "Audience" claim described in IETF RFC 7519 [25], clause 4.1.3.
scope	string	M	1	This IE shall contain the name of the NF services and the resource/operation-level scopes for which the access_token is authorized for use; this claim corresponds to a private claim, as described in IETF RFC 7519 [25], clause 4.3. pattern: '^([a-zA-Z0-9_:-]+)([a-zA-Z0-9_:-]+)*$'
exp	integer	M	1	This IE shall contain the expiration time after which the access_token is considered to be expired, corresponding to the standard "Expiration Time" claim described in IETF RFC 7519 [25], clause 4.1.4.
consumerPlmnId	PlmnId	C	0..1	This IE shall be included if the NRF supports providing PLMN ID of the NF service consumer in the access token claims, to be interpreted for subject (sub IE), as specified in clause 13.4.1.2 of 3GPP TS 33.501 [15]. If an NF service producer that receives this IE in the token included in the authorization header does not understand this IE, it shall be ignored.
producerPlmnId	PlmnId	C	0..1	This IE shall be included if the NRF supports providing PLMN ID of the NF service producer in the access token claims, to be interpretted for audience (aud IE), as specified in clause 13.4.1.2 of 3GPP TS 33.501 [15]. If an NF service producer that receives this IE in the token included in the authorization header does not understand this IE, it shall be ignored.
producerSnssaiList	array(Snssai)	O	1..N	This IE may be included if the NRF supports providing list of S-NSSAIs of the NF service producer in the access token claims. If an NF service producer that receives this IE in the token included in the authorization header does not understand this IE, it shall be ignored.
producerNsiList	array(string)	O	1..N	This IE may be included if the NRF supports providing list of NSIs of the NF service producer in the access token claims. If an NF service producer that receives this IE in the token included in the authorization header does not understand this IE, it shall be ignored.
producerNfSetId	NfSetId	O	0..1	This IE may be included if the NRF supports providing NF Set ID of the NF service producer in the access token claims and if the audience contains an NF type. When present, it shall indicate the NF Set ID of the NF service producer instances for which the claim is applicable. If an NF service producer that receives this IE in the token included in the authorization header does not understand this IE, it shall be ignored.

6.3.5.3 Simple data types and enumerations

6.3.5.3.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

6.3.5.3.2 Simple data types

There are no specific simple data types defined in this version of this API. For the re-used data types from other specifications see clause 6.3.5.1

6.3.5.3.3 Enumeration: GrantType

Table 6.3.5.3.3-1: Enumeration GrantType

Enumeration value	Description
"client_credentials"	Represents the Client Credentials grant type.

6.3.5.4 Data types describing alternative data types or combinations of data types

6.3.5.4.1 Type: Audience

Table 6.3.5.4.1-1: Definition of type Audience as a list of "non-exclusive alternatives"

Data type	Cardinality	Description	Applicability
NFType	1	NF type	
array(NfInstanceId)	1..N	Array of NF Instance Ids	

6.4 Nnrf_Bootstrapping Service API

6.4.1 API URI

URIs of this API shall have the following root:

{nrfApiRoot}

where {nrfApiRoot} represents the concatenation of the "scheme" and "authority" components of the NRF, as defined in IETF RFC 3986 [17].

6.4.2 Usage of HTTP

6.4.2.1 General

HTTP/2, as defined in IETF RFC 7540 [9], shall be used as specified in clause 5 of 3GPP TS 29.500 [4].

HTTP/2 shall be transported as specified in clause 5.3 of 3GPP TS 29.500 [4].

HTTP messages and bodies this API shall comply with the OpenAPI [10] specification contained in Annex A.

6.4.2.2 HTTP standard headers

6.4.2.2.1 General

The HTTP headers as specified in clause 4.4 of IETF RFC 6749 [16] shall be supported, with the exception that there shall not be "Authorization" HTTP request header in the access token request.

6.4.2.2.2 Content type

The following content types shall be supported:

- The JSON format (IETF RFC 8259 [22]). The use of the JSON format shall be signalled by the content type "application/json". See also clause 5.4 of 3GPP TS 29.500 [4].
- The Problem Details JSON Object (IETF RFC 7807 [11]). The use of the Problem Details JSON object in a HTTP response body shall be signalled by the content type "application/problem+json".
- The 3GPP hypermedia format as defined in 3GPP TS 29.501 [5]. The use of the 3GPP hypermedia format in a HTTP response body shall be signalled by the content type "application/3gppHal+json".

6.4.2.3 HTTP custom headers

6.4.2.3.1 General

In this release of this specification, no custom headers specific to the Nnrf_Bootstrapping Service API are defined. For 3GPP specific HTTP custom headers used across all service-based interfaces, see clause 5.2.3 of 3GPP TS 29.500 [4].

6.4.3 Resources

6.4.3.1 Overview

The structure of the Resource URIs of the Nnrf_Bootstrapping service is shown in figure 6.4.3.1-1.

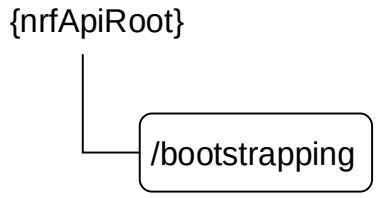

Figure 6.4.3.1-1: Resource URI structure of the Nnrf_Bootstrapping API

Table 6.4.3.1-1 provides an overview of the resources and applicable HTTP methods.

Table 6.4.3.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
Bootstrapping (Document)	{nrfApiRoot}/bootstrapping	GET	Retrieve a collection of links pointing to other services exposed by NRF.

6.4.3.2 Resource: Bootstrapping (Document)

6.4.3.2.1 Description

This resource represents a collection of links pointing to other services exposed by NRF.

This resource is modelled as the Document resource archetype (see clause C.3 of 3GPP TS 29.501 [5]).

6.4.3.2.2 Resource Definition

Resource URI: **{nrfApiRoot}/bootstrapping**

This resource shall support the resource URI variables defined in table 6.4.3.2.2-1.

Table 6.4.3.2.2-1: Resource URI variables for this resource

Name	Definition
nrfApiRoot	See clause 6.4.1

6.4.3.2.3 Resource Standard Methods

6.4.3.2.3.1 GET

This method retrieves a list of links pointing to other services exposed by NRF. This method shall support the URI query parameters specified in table 6.4.3.2.3.1-1.

Table 6.4.3.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
n/a	n/a			

This method shall support the request data structures specified in table 6.4.3.2.3.1-2 and the response data structures and response codes specified in table 6.4.3.2.3.1-3.

Table 6.4.3.2.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.4.3.2.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
BootstrappingInfo	M	1	200 OK	The response body contains a "_links" object containing the URI of each service exposed by the NRF.
NOTE: The mandatory HTTP error status codes for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] other than those specified in the table above also apply, with a ProblemDetails data type (see clause 5.2.7 of 3GPP TS 29.500 [4]).				

6.4.4 Custom Operations without associated resources

There are no custom operations defined without any associated resources for the Nnrf_Bootstrapping service in this release of the specification.

6.4.5 Notifications

There are no notifications defined for the Nnrf_Bootstrapping service in this release of the specification.

6.4.6 Data Model

6.4.6.1 General

This clause specifies the application data model supported by the API.

Table 6.4.6.1-1 specifies the data types defined for the Nnrf_Bootstrapping service-based interface protocol.

Table 6.4.6.1-1: Nnrf_Bootstrapping specific Data Types

Data type	Clause defined	Description
BootstrappingInfo	6.4.6.2.2	Information returned by NRF in the bootstrapping response message.
Status	6.4.6.3.2	Overal status of the NRF.

Table 6.4.6.1-2 specifies data types re-used by the Nnrf_Bootstrapping service-based interface protocol from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the Nnrf service-based interface.

Table 6.4.6.1-2: Nnrf_Bootstrapping re-used Data Types

Data type	Reference	Comments
LinksValueSchema	3GPP TS 29.571 [7]	3GPP Hypermedia link
ProblemDetails	3GPP TS 29.571 [7]	

6.4.6.2 Structured data types

6.4.6.2.1 Introduction

This clause defines the structures to be used in resource representations.

6.4.6.2.2 Type: BootstrappingInfo

Table 6.4.6.2.2-1: Definition of type BootstrappingInfo

Attribute name	Data type	P	Cardinality	Description
status	Status	O	0..1	Status of the NRF (operative, non-operative, ...) The NRF shall be considered as operative if this attribute is absent.
_links	map(LinksValueSchema)	M	1..N	Map of LinksValueSchema objects, where the keys are the link relations, as described in Table 6.4.6.3.3.1-1, and the values are objects containing an "href" attribute, whose value is an absolute URI corresponding to each link relation.

6.4.6.3 Simple data types and enumerations

6.4.6.3.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

6.4.6.3.2 Enumeration: Status

Table 6.4.6.3.2-1: Enumeration Status

Enumeration value	Description
"OPERATIVE"	The NRF is operative
"NON_OPERATIVE"	The NRF is not operative

6.4.6.3.3 Relation Types

6.4.6.3.3.1 General

This clause describes the possible relation types defined within NRF API. See clause 4.7.5.2 of 3GPP TS 29.501 [5] for the description of the relation types.

Table 6.4.6.3.3.1-1: supported registered relation types

Relation Name	Description
self	The "href" attribute of the object associated to this relation type contains the URI of the same resource returned in the response body (i.e. the "bootstrapping" resource).
manage	The "href" attribute of the object associated to this relation type contains the URI of the resource used in the Nnrf_NFManagement API to register/deregister/update NF Instances profiles in the NRF (i.e. the "nf-instances" store resource). (NOTE)
subscribe	The "href" attribute of the object associated to this relation type contains the URI of the resource used in the Nnrf_NFManagement API to manage subscriptions to the NRF (i.e. the "subscriptions" collection resource). (NOTE)
discover	The "href" attribute of the object associated to this relation type contains the URI of the resource used in the Nnrf_NFDiscovery API (i.e. the "nf-instances" collection resource).
authorize	The "href" attribute of the object associated to this relation type contains the URI of the Oauth2 Access Token Request endpoint, used to request authorization to other APIs in the 5G Core Network.
NOTE: The URIs of the "manage" and "subscribe" "href" attributes shall have the same apiRoot (i.e. authority and prefix) since these service operations belong to the same service.	

Annex A (normative): OpenAPI specification

A.1 General

This Annex specifies the formal definition of the Nnrf Service API(s). It consists of OpenAPI 3.0.0 specifications, in YAML format.

This Annex takes precedence when being discrepant to other parts of the specification with respect to the encoding of information elements and methods within the API(s).

NOTE: The semantics and procedures, as well as conditions, e.g. for the applicability and allowed combinations of attributes or values, not expressed in the OpenAPI definitions but defined in other parts of the specification also apply.

Informative copies of the OpenAPI specification files contained in this 3GPP Technical Specification are available on a Git-based repository hosted in ETSI Forge, that uses the GitLab software version control system (see 3GPP TS 29.501 [5] clause 5.3.1 and 3GPP TR 21.900 [31] clause 5B).

A.2 Nnrf_NFManagement API

```
openapi: 3.0.0

info:
  version: '1.1.0'
  title: 'NRF NFManagement Service'
  description: |
    NRF NFManagement Service.
    © 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).
    All rights reserved.

externalDocs:
  description: 3GPP TS 29.510 V16.4.0; 5G System; Network Function Repository Services; Stage 3
  url: 'http://www.3gpp.org/ftp/Specs/archive/29_series/29.510/'

servers:
  - url: '{apiRoot}/nnrf-nfm/v1'
    variables:
      apiRoot:
        default: https://example.com
        description: apiRoot as defined in clause 4.4 of 3GPP TS 29.501

security:
  - {}
  - oAuth2ClientCredentials:
      - nnrf-nfm

paths:
  /nf-instances:
    get:
      summary: Retrieves a collection of NF Instances
      operationId: GetNFInstances
      tags:
        - NF Instances (Store)
      parameters:
        - name: nf-type
          in: query
          description: Type of NF
          required: false
          schema:
            $ref: '#/components/schemas/NFType'
        - name: limit
          in: query
          description: How many items to return at one time
          required: false
          schema:
            type: integer
      responses:
        '200':
          description: Expected response to a valid request
          content:
```

```
            application/3gppHal+json:
              schema:
                type: object
                properties:
                  _links:
                    type: object
                    description: 'List of the URI of NF instances. It has two members whose names
are item and self. The item one contains an array of URIs.'
                    additionalProperties:
                      $ref: 'TS29571_CommonData.yaml#/components/schemas/LinksValueSchema'
                minProperties: 1
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '406':
          $ref: 'TS29571_CommonData.yaml#/components/responses/406'
        '411':
          $ref: 'TS29571_CommonData.yaml#/components/responses/411'
        '413':
          $ref: 'TS29571_CommonData.yaml#/components/responses/413'
        '415':
          $ref: 'TS29571_CommonData.yaml#/components/responses/415'
        '429':
          $ref: 'TS29571_CommonData.yaml#/components/responses/429'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '501':
          $ref: 'TS29571_CommonData.yaml#/components/responses/501'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
    options:
      summary: Discover communication options supported by NRF for NF Instances
      operationId: OptionsNFInstances
      tags:
        - NF Instances (Store)
      responses:
        '200':
          description: OK
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/OptionsResponse'
          headers:
            Accept-Encoding:
              description: Accept-Encoding, described in IETF RFC 7694
              schema:
                type: string
        '204':
          description: No Content
          headers:
            Accept-Encoding:
              description: Accept-Encoding, described in IETF RFC 7694
              schema:
                type: string
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '405':
          $ref: 'TS29571_CommonData.yaml#/components/responses/405'
        '429':
          $ref: 'TS29571_CommonData.yaml#/components/responses/429'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '501':
          $ref: 'TS29571_CommonData.yaml#/components/responses/501'
        '503':
```

```
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'
/nf-instances/{nfInstanceID}:
  get:
    summary: Read the profile of a given NF Instance
    operationId: GetNFInstance
    tags:
      - NF Instance ID (Document)
    parameters:
      - name: nfInstanceID
        in: path
        description: Unique ID of the NF Instance
        required: true
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
    responses:
      '200':
        description: Expected response to a valid request
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/NFProfile'
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '406':
        $ref: 'TS29571_CommonData.yaml#/components/responses/406'
      '411':
        $ref: 'TS29571_CommonData.yaml#/components/responses/411'
      '413':
        $ref: 'TS29571_CommonData.yaml#/components/responses/413'
      '415':
        $ref: 'TS29571_CommonData.yaml#/components/responses/415'
      '429':
        $ref: 'TS29571_CommonData.yaml#/components/responses/429'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '501':
        $ref: 'TS29571_CommonData.yaml#/components/responses/501'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'
  put:
    summary: Register a new NF Instance
    operationId: RegisterNFInstance
    tags:
      - NF Instance ID (Document)
    parameters:
      - name: nfInstanceID
        in: path
        required: true
        description: Unique ID of the NF Instance to register
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
      - name: Content-Encoding
        in: header
        description: Content-Encoding, described in IETF RFC 7231
        schema:
          type: string
      - name: Accept-Encoding
        in: header
        description: Accept-Encoding, described in IETF RFC 7231
        schema:
          type: string
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/NFProfile'
      required: true
    responses:
```

```
        '200':
          description: OK (Profile Replacement)
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NFProfile'
          headers:
            Accept-Encoding:
              description: Accept-Encoding, described in IETF RFC 7694
              schema:
                type: string
            Content-Encoding:
              description: Content-Encoding, described in IETF RFC 7231
              schema:
                type: string
        '201':
          description: Expected response to a valid request
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NFProfile'
          headers:
            Location:
              description: 'Contains the URI of the newly created resource, according to the
structure: {apiRoot}/nnrf-nfm/v1/nf-instances/{nfInstanceId}'
              required: true
              schema:
                type: string
            Accept-Encoding:
              description: Accept-Encoding, described in IETF RFC 7694
              schema:
                type: string
            Content-Encoding:
              description: Content-Encoding, described in IETF RFC 7231
              schema:
                type: string
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '411':
          $ref: 'TS29571_CommonData.yaml#/components/responses/411'
        '413':
          $ref: 'TS29571_CommonData.yaml#/components/responses/413'
        '415':
          $ref: 'TS29571_CommonData.yaml#/components/responses/415'
        '429':
          $ref: 'TS29571_CommonData.yaml#/components/responses/429'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '501':
          $ref: 'TS29571_CommonData.yaml#/components/responses/501'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
    patch:
      summary: Update NF Instance profile
      operationId: UpdateNFInstance
      tags:
        - NF Instance ID (Document)
      parameters:
        - name: nfInstanceID
          in: path
          required: true
          description: Unique ID of the NF Instance to update
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
        - name: Content-Encoding
          in: header
          description: Content-Encoding, described in IETF RFC 7231
          schema:
            type: string
        - name: Accept-Encoding
```

```
        in: header
        description: Accept-Encoding, described in IETF RFC 7231
        schema:
          type: string
      requestBody:
        content:
          application/json-patch+json:
            schema:
              type: array
              items:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/PatchItem'
              minItems: 1
        required: true
      responses:
        '200':
          description: Expected response to a valid request
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NFProfile'
          headers:
            Accept-Encoding:
              description: Accept-Encoding, described in IETF RFC 7694
              schema:
                type: string
            Content-Encoding:
              description: Content-Encoding, described in IETF RFC 7231
              schema:
                type: string
        '204':
          description: Expected response with empty body
          headers:
            Accept-Encoding:
              description: Accept-Encoding, described in IETF RFC 7694
              schema:
                type: string
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '411':
          $ref: 'TS29571_CommonData.yaml#/components/responses/411'
        '413':
          $ref: 'TS29571_CommonData.yaml#/components/responses/413'
        '415':
          $ref: 'TS29571_CommonData.yaml#/components/responses/415'
        '429':
          $ref: 'TS29571_CommonData.yaml#/components/responses/429'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '501':
          $ref: 'TS29571_CommonData.yaml#/components/responses/501'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
    delete:
      summary: Deregisters a given NF Instance
      operationId: DeregisterNFInstance
      tags:
        - NF Instance ID (Document)
      parameters:
        - name: nfInstanceID
          in: path
          required: true
          description: Unique ID of the NF Instance to deregister
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
      responses:
        '204':
          description: Expected response to a successful deregistration
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
```

```
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '411':
          $ref: 'TS29571_CommonData.yaml#/components/responses/411'
        '429':
          $ref: 'TS29571_CommonData.yaml#/components/responses/429'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '501':
          $ref: 'TS29571_CommonData.yaml#/components/responses/501'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
  /subscriptions:
    post:
      summary: Create a new subscription
      operationId: CreateSubscription
      tags:
        - Subscriptions (Collection)
      parameters:
        - name: Content-Encoding
          in: header
          description: Content-Encoding, described in IETF RFC 7231
          schema:
            type: string
        - name: Accept-Encoding
          in: header
          description: Accept-Encoding, described in IETF RFC 7231
          schema:
            type: string
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SubscriptionData'
        required: true
      responses:
        '201':
          description: Expected response to a valid request
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/SubscriptionData'
          headers:
            Location:
              description: 'Contains the URI of the newly created resource, according to the
structure: {apiRoot}/nnrf-nfm/v1/subscriptions/{subscriptionId}'
              required: true
              schema:
                type: string
            Accept-Encoding:
              description: Accept-Encoding, described in IETF RFC 7694
              schema:
                type: string
            Content-Encoding:
              description: Content-Encoding, described in IETF RFC 7231
              schema:
                type: string
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '411':
          $ref: 'TS29571_CommonData.yaml#/components/responses/411'
        '413':
          $ref: 'TS29571_CommonData.yaml#/components/responses/413'
        '415':
          $ref: 'TS29571_CommonData.yaml#/components/responses/415'
        '429':
          $ref: 'TS29571_CommonData.yaml#/components/responses/429'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
```

```
        '501':
          $ref: 'TS29571_CommonData.yaml#/components/responses/501'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
      callbacks:
        onNFStatusEvent:
          '{$request.body#/nfStatusNotificationUri}':
            post:
              parameters:
                - name: Content-Encoding
                  in: header
                  description: Content-Encoding, described in IETF RFC 7231
                  schema:
                    type: string
              requestBody:
                description: Notification Payload
                content:
                  application/json:
                    schema:
                      $ref: '#/components/schemas/NotificationData'
              responses:
                '204':
                  description: Expected response to a successful callback processing
                  headers:
                    Accept-Encoding:
                      description: Accept-Encoding, described in IETF RFC 7694
                      schema:
                        type: string
                '400':
                  $ref: 'TS29571_CommonData.yaml#/components/responses/400'
                '401':
                  $ref: 'TS29571_CommonData.yaml#/components/responses/401'
                '403':
                  $ref: 'TS29571_CommonData.yaml#/components/responses/403'
                '404':
                  $ref: 'TS29571_CommonData.yaml#/components/responses/404'
                '411':
                  $ref: 'TS29571_CommonData.yaml#/components/responses/411'
                '413':
                  $ref: 'TS29571_CommonData.yaml#/components/responses/413'
                '415':
                  $ref: 'TS29571_CommonData.yaml#/components/responses/415'
                '429':
                  $ref: 'TS29571_CommonData.yaml#/components/responses/429'
                '500':
                  $ref: 'TS29571_CommonData.yaml#/components/responses/500'
                '501':
                  $ref: 'TS29571_CommonData.yaml#/components/responses/501'
                '503':
                  $ref: 'TS29571_CommonData.yaml#/components/responses/503'
                default:
                  $ref: 'TS29571_CommonData.yaml#/components/responses/default'
  /subscriptions/{subscriptionID}:
    patch:
      summary: Updates a subscription
      operationId: UpdateSubscription
      tags:
        - Subscription ID (Document)
      parameters:
        - name: subscriptionID
          in: path
          required: true
          description: Unique ID of the subscription to update
          schema:
            type: string
            pattern: '^([0-9]{5,6}-)?[^-]+$'
        - name: Content-Encoding
          in: header
          description: Content-Encoding, described in IETF RFC 7231
          schema:
            type: string
        - name: Accept-Encoding
          in: header
          description: Accept-Encoding, described in IETF RFC 7231
          schema:
            type: string
```

```
    requestBody:
      content:
        application/json-patch+json:
          schema:
            type: array
            items:
              $ref: 'TS29571_CommonData.yaml#/components/schemas/PatchItem'
      required: true
    responses:
      '200':
        description: Expected response to a valid request
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SubscriptionData'
        headers:
          Accept-Encoding:
            description: Accept-Encoding, described in IETF RFC 7694
            schema:
              type: string
          Content-Encoding:
            description: Content-Encoding, described in IETF RFC 7231
            schema:
              type: string
      '204':
        description: No Content
        headers:
          Accept-Encoding:
            description: Accept-Encoding, described in IETF RFC 7694
            schema:
              type: string
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '411':
        $ref: 'TS29571_CommonData.yaml#/components/responses/411'
      '413':
        $ref: 'TS29571_CommonData.yaml#/components/responses/413'
      '415':
        $ref: 'TS29571_CommonData.yaml#/components/responses/415'
      '429':
        $ref: 'TS29571_CommonData.yaml#/components/responses/429'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '501':
        $ref: 'TS29571_CommonData.yaml#/components/responses/501'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'
  delete:
    summary: Deletes a subscription
    operationId: RemoveSubscription
    tags:
      - Subscription ID (Document)
    parameters:
      - name: subscriptionID
        in: path
        required: true
        description: Unique ID of the subscription to remove
        schema:
          type: string
          pattern: '^([0-9]{5,6}-)?[^-]+$'
    responses:
      '204':
        description: Expected response to a successful subscription removal
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '411':
```

```
          $ref: 'TS29571_CommonData.yaml#/components/responses/411'
        '413':
          $ref: 'TS29571_CommonData.yaml#/components/responses/413'
        '415':
          $ref: 'TS29571_CommonData.yaml#/components/responses/415'
        '429':
          $ref: 'TS29571_CommonData.yaml#/components/responses/429'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '501':
          $ref: 'TS29571_CommonData.yaml#/components/responses/501'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
components:
  securitySchemes:
    oAuth2ClientCredentials:
      type: oauth2
      flows:
        clientCredentials:
          tokenUrl: '/oauth2/token'
          scopes:
            nnrf-nfm: Access to the Nnrf_NFManagement API
  schemas:
    NFProfile:
      description: Information of an NF Instance registered in the NRF
      type: object
      required:
        - nfInstanceId
        - nfType
        - nfStatus
      anyOf:
        - required: [ fqdn ]
        - required: [ ipv4Addresses ]
        - required: [ ipv6Addresses ]
      properties:
        nfInstanceId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
        nfInstanceName:
          type: string
        nfType:
          $ref: '#/components/schemas/NFType'
        nfStatus:
          $ref: '#/components/schemas/NFStatus'
        heartBeatTimer:
          type: integer
        plmnList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
          minItems: 1
        snpnList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnIdNid'
          minItems: 1
        sNssais:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/ExtSnssai'
          minItems: 1
        perPlmnSnssaiList:
          type: array
          items:
            $ref: '#/components/schemas/PlmnSnssai'
          minItems: 1
        nsiList:
          type: array
          items:
            type: string
          minItems: 1
        fqdn:
          $ref: '#/components/schemas/Fqdn'
        interPlmnFqdn:
          $ref: '#/components/schemas/Fqdn'
        ipv4Addresses:
          type: array
```

```
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
      minItems: 1
    ipv6Addresses:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Addr'
      minItems: 1
    allowedPlmns:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
      minItems: 1
    allowedSnpns:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnIdNid'
      minItems: 1
    allowedNfTypes:
      type: array
      items:
        $ref: '#/components/schemas/NFType'
      minItems: 1
    allowedNfDomains:
      type: array
      items:
        type: string
      minItems: 1
    allowedNssais:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/ExtSnssai'
      minItems: 1
    priority:
      type: integer
      minimum: 0
      maximum: 65535
    capacity:
      type: integer
      minimum: 0
      maximum: 65535
    load:
      type: integer
      minimum: 0
      maximum: 100
    loadTimeStamp:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
    locality:
      type: string
    udrInfo:
      $ref: '#/components/schemas/UdrInfo'
    udrInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/UdrInfo'
      minProperties: 1
    udmInfo:
      $ref: '#/components/schemas/UdmInfo'
    udmInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/UdmInfo'
      minProperties: 1
    ausfInfo:
      $ref: '#/components/schemas/AusfInfo'
    ausfInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/AusfInfo'
      minProperties: 1
    amfInfo:
      $ref: '#/components/schemas/AmfInfo'
    amfInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/AmfInfo'
      minProperties: 1
    smfInfo:
```

```
      $ref: '#/components/schemas/SmfInfo'
    smfInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/SmfInfo'
      minProperties: 1
    upfInfo:
      $ref: '#/components/schemas/UpfInfo'
    upfInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/UpfInfo'
      minProperties: 1
    pcfInfo:
      $ref: '#/components/schemas/PcfInfo'
    pcfInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/PcfInfo'
      minProperties: 1
    bsfInfo:
      $ref: '#/components/schemas/BsfInfo'
    bsfInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/BsfInfo'
      minProperties: 1
    chfInfo:
      $ref: '#/components/schemas/ChfInfo'
    chfInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/ChfInfo'
      minProperties: 1
    nefInfo:
      $ref: '#/components/schemas/NefInfo'
    nrfInfo:
      $ref: '#/components/schemas/NrfInfo'
    udsfInfo:
      $ref: '#/components/schemas/UdsfInfo'
    udsfInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/UdsfInfo'
      minProperties: 1
    nwdafInfo:
      $ref: '#/components/schemas/NwdafInfo'
    pcscfInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/PcscfInfo'
      minProperties: 1
    hssInfoList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/HssInfo'
      minProperties: 1
    customInfo:
      type: object
    recoveryTime:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
    nfServicePersistence:
      type: boolean
      default: false
    nfServices:
      deprecated: true
      type: array
      items:
        $ref: '#/components/schemas/NFService'
      minItems: 1
    nfServiceList:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/NFService'
      minProperties: 1
    nfProfileChangesSupportInd:
      type: boolean
      default: false
```

```
        writeOnly: true
      nfProfileChangesInd:
        type: boolean
        default: false
        readOnly: true
      defaultNotificationSubscriptions:
        type: array
        items:
          $ref: '#/components/schemas/DefaultNotificationSubscription'
      lmfInfo:
        $ref: '#/components/schemas/LmfInfo'
      gmlcInfo:
        $ref: '#/components/schemas/GmlcInfo'
      nfSetIdList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfSetId'
        minItems: 1
      servingScope:
        type: array
        items:
          type: string
        minItems: 1
      lcHSupportInd:
        type: boolean
        default: false
      olcHSupportInd:
        type: boolean
        default: false
      nfSetRecoveryTimeList:
        type: object
        additionalProperties:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
        minProperties: 1
      serviceSetRecoveryTimeList:
        type: object
        additionalProperties:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
        minProperties: 1
      scpDomains:
        type: array
        items:
          type: string
        minItems: 1
      scpInfo:
        $ref: '#/components/schemas/ScpInfo'
  NFService:
    description: Information of a given NF Service Instance; it is part of the NFProfile of an NF
Instance
    type: object
    required:
      - serviceInstanceId
      - serviceName
      - versions
      - scheme
      - nfServiceStatus
    properties:
      serviceInstanceId:
        type: string
      serviceName:
        $ref: '#/components/schemas/ServiceName'
      versions:
        type: array
        items:
          $ref: '#/components/schemas/NFServiceVersion'
        minItems: 1
      scheme:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/UriScheme'
      nfServiceStatus:
        $ref: '#/components/schemas/NFServiceStatus'
      fqdn:
        $ref: '#/components/schemas/Fqdn'
      interPlmnFqdn:
        $ref: '#/components/schemas/Fqdn'
      ipEndPoints:
        type: array
        items:
          $ref: '#/components/schemas/IpEndPoint'
```

```
      minItems: 1
    apiPrefix:
      type: string
    defaultNotificationSubscriptions:
      type: array
      items:
        $ref: '#/components/schemas/DefaultNotificationSubscription'
      minItems: 1
    allowedPlmns:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
      minItems: 1
    allowedSnpns:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnIdNid'
      minItems: 1
    allowedNfTypes:
      type: array
      items:
        $ref: '#/components/schemas/NFType'
      minItems: 1
    allowedNfDomains:
      type: array
      items:
        type: string
      minItems: 1
    allowedNssais:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/ExtSnssai'
      minItems: 1
    allowedOperationsPerNfType:
      type: object
      additionalProperties:
        type: array
        items:
          type: string
        minItems: 1
    allowedOperationsPerNfInstance:
      type: object
      additionalProperties:
        type: array
        items:
          type: string
        minItems: 1
    priority:
      type: integer
      minimum: 0
      maximum: 65535
    capacity:
      type: integer
      minimum: 0
      maximum: 65535
    load:
      type: integer
      minimum: 0
      maximum: 100
    loadTimeStamp:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
    recoveryTime:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
    chfServiceInfo:
      $ref: '#/components/schemas/ChfServiceInfo'
    supportedFeatures:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    nfServiceSetIdList:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/NfServiceSetId'
      minItems: 1
    sNssais:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/ExtSnssai'
      minItems: 1
    perPlmnSnssaiList:
```

```
          type: array
          items:
            $ref: '#/components/schemas/PlmnSnssai'
          minItems: 1
        vendorId:
          $ref: '#/components/schemas/VendorId'
        supportedVendorSpecificFeatures:
          type: object
          additionalProperties:
            type: array
            items:
              $ref: '#/components/schemas/VendorSpecificFeature'
          minProperties: 1
        oauth2Required:
          type: boolean
    NFType:
      description: NF types known to NRF
      anyOf:
        - type: string
          enum:
            - NRF
            - UDM
            - AMF
            - SMF
            - AUSF
            - NEF
            - PCF
            - SMSF
            - NSSF
            - UDR
            - LMF
            - GMLC
            - 5G_EIR
            - SEPP
            - UPF
            - N3IWF
            - AF
            - UDSF
            - BSF
            - CHF
            - NWDAF
            - PCSCF
            - CBCF
            - HSS
            - UCMF
            - SOR_AF
            - SPAF
            - MME
            - SCSAS
            - SCEF
            - SCP
            - NSSAAF
        - type: string
    Fqdn:
      description: Fully Qualified Domain Name
      type: string
    NefId:
      description: Identity of the NEF
      type: string
    IpEndPoint:
      description: IP addressing information of a given NFService; it consists on, e.g. IP address,
TCP port, transport protocol...
      type: object
      properties:
        ipv4Address:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
        ipv6Address:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Addr'
        transport:
          $ref: '#/components/schemas/TransportProtocol'
        port:
          type: integer
          minimum: 0
          maximum: 65535
    SubscriptionData:
      description: Information of a subscription to notifications to NRF events, included in
subscription requests and responses
      type: object
```

```
    required:
      - nfStatusNotificationUri
      - subscriptionId
    properties:
      nfStatusNotificationUri:
        type: string
      reqNfInstanceId:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
      subscrCond:
        oneOf:
          - $ref: '#/components/schemas/NfInstanceIdCond'
          - $ref: '#/components/schemas/NfInstanceIdListCond'
          - $ref: '#/components/schemas/NfTypeCond'
          - $ref: '#/components/schemas/ServiceNameCond'
          - $ref: '#/components/schemas/AmfCond'
          - $ref: '#/components/schemas/GuamiListCond'
          - $ref: '#/components/schemas/NetworkSliceCond'
          - $ref: '#/components/schemas/NfGroupCond'
          - $ref: '#/components/schemas/NfSetCond'
          - $ref: '#/components/schemas/NfServiceSetCond'
          - $ref: '#/components/schemas/UpfCond'
          - $ref: '#/components/schemas/ScpDomainCond'
      subscriptionId:
        type: string
        pattern: '^([0-9]{5,6}-)?[^-]+$'
        readOnly: true
      validityTime:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
      reqNotifEvents:
        type: array
        items:
          $ref: '#/components/schemas/NotificationEventType'
        minItems: 1
      plmnId:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
      nid:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Nid'
      notifCondition:
         $ref: '#/components/schemas/NotifCondition'
      reqNfType:
        $ref: '#/components/schemas/NFType'
      reqNfFqdn:
        $ref: '#/components/schemas/Fqdn'
      reqSnssais:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Snssai'
        minItems: 1
      reqPlmnList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
        minItems: 1
      reqSnpnList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnIdNid'
        minItems: 1
      servingScope:
        type: array
        items:
          type: string
        minItems: 1
      requesterFeatures:
        writeOnly: true
        allOf:
          - $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      nrfSupportedFeatures:
        readOnly: true
        allOf:
          - $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
  NfInstanceIdCond:
    description: Subscription to a given NF Instance Id
    type: object
    required:
      - nfInstanceId
    properties:
      nfInstanceId:
```

```
      $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
NfInstanceIdListCond:
  description: Subscription to a list of NF Instances
  type: object
  required:
    - nfInstanceIdList
  properties:
    nfInstanceIdList:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
      minItems: 1
NfTypeCond:
  description: Subscription to a set of NFs based on their NF Type
  type: object
  required:
    - nfType
  not:
    required: [ nfGroupId ]
  properties:
    nfType:
      $ref: '#/components/schemas/NFType'
ServiceNameCond:
  description: Subscription to a set of NFs based on their support for a given Service Name
  type: object
  required:
    - serviceName
  properties:
    serviceName:
      $ref: '#/components/schemas/ServiceName'
AmfCond:
  description: Subscription to a set of AMFs, based on AMF Set Id and/or AMF Region Id
  type: object
  anyOf:
    - required: [ amfSetId ]
    - required: [ amfRegionId ]
  properties:
    amfSetId:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/AmfSetId'
    amfRegionId:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/AmfRegionId'
GuamiListCond:
  description: Subscription to a set of AMFs, based on their GUAMIs
  type: object
  required:
    - guamiList
  properties:
    guamiList:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Guami'
NetworkSliceCond:
  description: Subscription to a set of NFs, based on the slices (S-NSSAI and NSI) they support
  type: object
  required:
    - snssaiList
  properties:
    snssaiList:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Snssai'
    nsiList:
      type: array
      items:
        type: string
NfGroupCond:
  description: Subscription to a set of NFs based on their Group Id
  type: object
  required:
    - nfType
    - nfGroupId
  properties:
    nfType:
      type: string
      enum:
        - UDM
        - AUSF
        - UDR
```

```
            - PCF
            - CHF
        nfGroupId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfGroupId'
    NotifCondition:
      description: Condition (list of attributes in the NF Profile) to determine whether a
notification must be sent by NRF
      type: object
      not:
        required: [ monitoredAttributes, unmonitoredAttributes ]
      properties:
        monitoredAttributes:
          type: array
          items:
            type: string
          minItems: 1
        unmonitoredAttributes:
          type: array
          items:
            type: string
          minItems: 1
    UdrInfo:
      description: Information of an UDR NF Instance
      type: object
      properties:
        groupId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfGroupId'
        supiRanges:
          type: array
          items:
            $ref: '#/components/schemas/SupiRange'
          minItems: 1
        gpsiRanges:
          type: array
          items:
            $ref: '#/components/schemas/IdentityRange'
          minItems: 1
        externalGroupIdentifiersRanges:
          type: array
          items:
            $ref: '#/components/schemas/IdentityRange'
          minItems: 1
        supportedDataSets:
          type: array
          items:
            $ref: '#/components/schemas/DataSetId'
          minItems: 1
    SupiRange:
      description: A range of SUPIs (subscriber identities), either based on a numeric range, or
based on regular-expression matching
      type: object
      properties:
        start:
          type: string
          pattern: '^[0-9]+$'
        end:
          type: string
          pattern: '^[0-9]+$'
        pattern:
          type: string
    IdentityRange:
      description: A range of GPSIs (subscriber identities), either based on a numeric range, or
based on regular-expression matching
      type: object
      properties:
        start:
          type: string
          pattern: '^[0-9]+$'
        end:
          type: string
          pattern: '^[0-9]+$'
        pattern:
          type: string
    InternalGroupIdRange:
      description: A range of Group IDs (internal group identities), either based on a numeric
range, or based on regular-expression matching
      type: object
      properties:
```

```
      start:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/GroupId'
      end:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/GroupId'
      pattern:
        type: string
  DataSetId:
    description: Types of data sets stored in UDR
    anyOf:
      - type: string
        enum:
          - SUBSCRIPTION
          - POLICY
          - EXPOSURE
          - APPLICATION
      - type: string
  UdmInfo:
    description: Information of an UDM NF Instance
    type: object
    properties:
      groupId:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/NfGroupId'
      supiRanges:
        type: array
        items:
          $ref: '#/components/schemas/SupiRange'
        minItems: 1
      gpsiRanges:
        type: array
        items:
          $ref: '#/components/schemas/IdentityRange'
        minItems: 1
      externalGroupIdentifiersRanges:
        type: array
        items:
          $ref: '#/components/schemas/IdentityRange'
        minItems: 1
      routingIndicators:
        type: array
        items:
          type: string
          pattern: '^[0-9]{1,4}$'
        minItems: 1
      internalGroupIdentifiersRanges:
        type: array
        items:
          $ref: '#/components/schemas/InternalGroupIdRange'
        minItems: 1
  AusfInfo:
    description: Information of an AUSF NF Instance
    type: object
    properties:
      groupId:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/NfGroupId'
      supiRanges:
        type: array
        items:
          $ref: '#/components/schemas/SupiRange'
        minItems: 1
      routingIndicators:
        type: array
        items:
          type: string
          pattern: '^[0-9]{1,4}$'
        minItems: 1
  AmfInfo:
    description: Information of an AMF NF Instance
    type: object
    required:
      - amfSetId
      - amfRegionId
      - guamiList
    properties:
      amfSetId:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/AmfSetId'
      amfRegionId:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/AmfRegionId'
      guamiList:
```

```
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Guami'
        minItems: 1
      taiList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Tai'
        minItems: 1
      taiRangeList:
        type: array
        items:
          $ref: '#/components/schemas/TaiRange'
        minItems: 1
      backupInfoAmfFailure:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Guami'
        minItems: 1
      backupInfoAmfRemoval:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Guami'
        minItems: 1
      n2InterfaceAmfInfo:
        $ref: '#/components/schemas/N2InterfaceAmfInfo'
  SmfInfo:
    description: Information of an SMF NF Instance
    type: object
    required:
      - sNssaiSmfInfoList
    properties:
      sNssaiSmfInfoList:
        type: array
        items:
          $ref: '#/components/schemas/SnssaiSmfInfoItem'
        minItems: 1
      taiList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Tai'
        minItems: 1
      taiRangeList:
        type: array
        items:
          $ref: '#/components/schemas/TaiRange'
        minItems: 1
      pgwFqdn:
        $ref: '#/components/schemas/Fqdn'
      accessType:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/AccessType'
        minItems: 1
      priority:
        type: integer
        minimum: 0
        maximum: 65535
  SnssaiSmfInfoItem:
    description: Set of parameters supported by SMF for a given S-NSSAI
    type: object
    required:
      - sNssai
      - dnnSmfInfoList
    properties:
      sNssai:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Snssai'
      dnnSmfInfoList:
        type: array
        items:
          $ref: '#/components/schemas/DnnSmfInfoItem'
        minItems: 1
  DnnSmfInfoItem:
    description: Set of parameters supported by SMF for a given DNN
    type: object
    required:
      - dnn
    properties:
```

```
      dnn:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Dnn'
  UpfInfo:
    description: Information of an UPF NF Instance
    type: object
    required:
      - sNssaiUpfInfoList
    properties:
      sNssaiUpfInfoList:
        type: array
        items:
          $ref: '#/components/schemas/SnssaiUpfInfoItem'
        minItems: 1
      smfServingArea:
        type: array
        items:
          type: string
        minItems: 1
      interfaceUpfInfoList:
        type: array
        items:
          $ref: '#/components/schemas/InterfaceUpfInfoItem'
        minItems: 1
      iwkEpsInd:
        type: boolean
        default: false
      pduSessionTypes:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/PduSessionType'
        minItems: 1
      atsssCapability:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/AtsssCapability'
      ueIpAddrInd:
        type: boolean
        default: false
      taiList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Tai'
        minItems: 1
      wAgfInfo:
        $ref: '#/components/schemas/WAgfInfo'
      tngfInfo:
        $ref: '#/components/schemas/TngfInfo'
      twifInfo:
        $ref: '#/components/schemas/TwifInfo'
      priority:
        type: integer
        minimum: 0
        maximum: 65535
      redundantGtpu:
        type: boolean
        default: false
      ipups:
        type: boolean
        default: false
      dataForwarding:
        type: boolean
        default: false
  SnssaiUpfInfoItem:
    description: Set of parameters supported by UPF for a given S-NSSAI
    type: object
    required:
      - sNssai
      - dnnUpfInfoList
    properties:
      sNssai:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Snssai'
      dnnUpfInfoList:
        type: array
        items:
          $ref: '#/components/schemas/DnnUpfInfoItem'
        minItems: 1
      redundantTransport:
        type: boolean
        default: false
  DnnUpfInfoItem:
```

```
    description: Set of parameters supported by UPF for a given DNN
    type: object
    required:
      - dnn
    properties:
      dnn:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Dnn'
      dnaiList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Dnai'
        minItems: 1
      pduSessionTypes:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/PduSessionType'
        minItems: 1
      ipv4AddressRanges:
        type: array
        items:
          $ref: '#/components/schemas/Ipv4AddressRange'
        minItems: 1
      ipv6PrefixRanges:
        type: array
        items:
          $ref: '#/components/schemas/Ipv6PrefixRange'
        minItems: 1
  InterfaceUpfInfoItem:
    description: Information of a given IP interface of an UPF
    type: object
    required:
      - interfaceType
    properties:
      interfaceType:
        $ref: '#/components/schemas/UPInterfaceType'
      ipv4EndpointAddresses:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
        minItems: 1
      ipv6EndpointAddresses:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Addr'
        minItems: 1
      endpointFqdn:
          $ref: '#/components/schemas/Fqdn'
      networkInstance:
        type: string
  UPInterfaceType:
    description: Types of User-Plane interfaces of the UPF
    anyOf:
      - type: string
        enum:
          - N3
          - N6
          - N9
          - DATA_FORWARDING
      - type: string
  WAgfInfo:
    description: Information of the W-AGF end-points
    type: object
    properties:
      ipv4EndpointAddresses:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
        minItems: 1
      ipv6EndpointAddresses:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Addr'
        minItems: 1
      endpointFqdn:
        $ref: '#/components/schemas/Fqdn'
  TngfInfo:
    description: Infomation of the TNGF endpoints
    type: object
```

```
    properties:
      ipv4EndpointAddresses:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
        minItems: 1
      ipv6EndpointAddresses:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Addr'
        minItems: 1
      endpointFqdn:
        $ref: '#/components/schemas/Fqdn'
  PcfInfo:
    description: Information of a PCF NF Instance
    type: object
    properties:
      groupId:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/NfGroupId'
      dnnList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Dnn'
        minItems: 1
      supiRanges:
        type: array
        items:
          $ref: '#/components/schemas/SupiRange'
        minItems: 1
      gpsiRanges:
        type: array
        items:
          $ref: '#/components/schemas/IdentityRange'
        minItems: 1
      rxDiamHost:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/DiameterIdentity'
      rxDiamRealm:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/DiameterIdentity'
      v2xSupportInd:
        type: boolean
        default: false
  BsfInfo:
    description: Information of a BSF NF Instance
    type: object
    properties:
      dnnList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Dnn'
        minItems: 1
      ipDomainList:
        type: array
        items:
          type: string
        minItems: 1
      ipv4AddressRanges:
        type: array
        items:
          $ref: '#/components/schemas/Ipv4AddressRange'
        minItems: 1
      ipv6PrefixRanges:
        type: array
        items:
          $ref: '#/components/schemas/Ipv6PrefixRange'
        minItems: 1
  ChfInfo:
    description: Information of a CHF NF Instance
    type: object
    properties:
      supiRangeList:
        type: array
        items:
          $ref: '#/components/schemas/SupiRange'
        minItems: 1
      gpsiRangeList:
        type: array
        items:
          $ref: '#/components/schemas/IdentityRange'
```

```
          minItems: 1
        plmnRangeList:
          type: array
          items:
            $ref: '#/components/schemas/PlmnRange'
          minItems: 1
        groupId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfGroupId'
    Ipv4AddressRange:
      description: Range of IPv4 addresses
      type: object
      properties:
        start:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
        end:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
    Ipv6PrefixRange:
      description: Range of IPv6 prefixes
      type: object
      properties:
        start:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Prefix'
        end:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Prefix'
    DefaultNotificationSubscription:
      description: Data structure for specifying the notifications the NF service subscribes by
default along with callback URI
      type: object
      required:
        - notificationType
        - callbackUri
      properties:
        notificationType:
          $ref: '#/components/schemas/NotificationType'
        callbackUri:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
        n1MessageClass:
          $ref: 'TS29518_Namf_Communication.yaml#/components/schemas/N1MessageClass'
        n2InformationClass:
          $ref: 'TS29518_Namf_Communication.yaml#/components/schemas/N2InformationClass'
        versions:
          type: array
          items:
            type: string
          minItems: 1
    NfSetCond:
      description: Subscription to a set of NFs based on their Set Id
      type: object
      required:
        - nfSetId
      properties:
        nfSetId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfSetId'
    NfServiceSetCond:
      description: Subscription to a set of NFs based on their Service Set Id
      type: object
      required:
        - nfServiceSetId
      properties:
        nfServiceSetId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfServiceSetId'
    UpfCond:
      description: Subscription to a set of NF Instances (UPFs), able to serve a certain service
area (i.e. SMF serving area or TAI list)
      type: object
      properties:
        smfServingArea:
          type: array
          items:
            type: string
          minItems: 1
        taiList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Tai'
          minItems: 1
    NotificationType:
```

```
      description: Types of notifications used in Default Notification URIs in the NF Profile of an
NF Instance
      anyOf:
        - type: string
          enum:
            - N1_MESSAGES
            - N2_INFORMATION
            - LOCATION_NOTIFICATION
            - DATA_REMOVAL_NOTIFICATION
            - DATA_CHANGE_NOTIFICATION
            - LOCATION_UPDATE_NOTIFICATION
            - NSSAA_REAUTH_NOTIFICATION
            - NSSAA_REVOC_NOTIFICATION
        - type: string
    TransportProtocol:
      description: Types of transport protocol used in a given IP endpoint of an NF Service Instance
      anyOf:
        - type: string
          enum:
            - TCP
        - type: string
    NotificationEventType:
      description: Types of events sent in notifications from NRF to subscribed NF Instances
      anyOf:
        - type: string
          enum:
            - NF_REGISTERED
            - NF_DEREGISTERED
            - NF_PROFILE_CHANGED
        - type: string
    NotificationData:
      description: Data sent in notifications from NRF to subscribed NF Instances
      type: object
      required:
        - event
        - nfInstanceUri
      allOf:
        #
        # Condition: If 'event' takes value 'NF_PROFILE_CHANGED',
        # then either 'nfProfile' or 'profileChanges' (but not both) must be present
        #
        - anyOf:
          - not:
              properties:
                event:
                  type: string
                  enum:
                    - NF_PROFILE_CHANGED
          - oneOf:
              - required: [ nfProfile ]
              - required: [ profileChanges ]
        #
        # Condition: If 'event' takes value 'NF_REGISTERED',
        # then 'nfProfile' must be present
        #
        - anyOf:
          - not:
              properties:
                event:
                  type: string
                  enum:
                    - NF_REGISTERED
          - required: [ nfProfile ]
      properties:
        event:
          $ref: '#/components/schemas/NotificationEventType'
        nfInstanceUri:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
        nfProfile:
          allOf:
            - $ref: '#/components/schemas/NFProfile'
            - not:
                required: [ interPlmnFqdn ]
            - not:
                required: [ allowedPlmns ]
            - not:
                required: [ allowedSnpns ]
            - not:
```

```
            required: [ allowedNfTypes ]
          - not:
            required: [ allowedNfDomains ]
          - not:
            required: [ allowedNssais ]
          - properties:
            nfServices:
              type: array
              items:
                allOf:
                  - $ref: '#/components/schemas/NFService'
                  - not:
                      required: [ interPlmnFqdn ]
                  - not:
                      required: [ allowedPlmns ]
                  - not:
                      required: [ allowedSnpns ]
                  - not:
                      required: [ allowedNfTypes ]
                  - not:
                      required: [ allowedNfDomains ]
                  - not:
                      required: [ allowedNssais ]
    profileChanges:
      type: array
      items:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/ChangeItem'
      minItems: 1
    conditionEvent:
      $ref: '#/components/schemas/ConditionEventType'
NFStatus:
  description: Status of a given NF Instance stored in NRF
  anyOf:
    - type: string
      enum:
        - REGISTERED
        - SUSPENDED
        - UNDISCOVERABLE
    - type: string
NFServiceVersion:
  description: Contains the version details of an NF service
  type: object
  required:
    - apiVersionInUri
    - apiFullVersion
  properties:
    apiVersionInUri:
      type: string
    apiFullVersion:
      type: string
    expiry:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
ServiceName:
  description: Service names known to NRF
  anyOf:
    - type: string
      enum:
        - nnrf-nfm
        - nnrf-disc
        - nnrf-oauth2
        - nudm-sdm
        - nudm-uecm
        - nudm-ueau
        - nudm-ee
        - nudm-pp
        - nudm-niddau
        - nudm-mt
        - namf-comm
        - namf-evts
        - namf-mt
        - namf-loc
        - nsmf-pdusession
        - nsmf-event-exposure
        - nsmf-nidd
        - nausf-auth
        - nausf-sorprotection
        - nausf-upuprotection
        - nnef-pfdmanagement
```

```
          - nnef-smcontext
          - nnef-eventexposure
          - npcf-am-policy-control
          - npcf-smpolicycontrol
          - npcf-policyauthorization
          - npcf-bdtpolicycontrol
          - npcf-eventexposure
          - npcf-ue-policy-control
          - nsmsf-sms
          - nnssf-nsselection
          - nnssf-nssaiavailability
          - nudr-dr
          - nudr-group-id-map
          - nlmf-loc
          - n5g-eir-eic
          - nbsf-management
          - nchf-spendinglimitcontrol
          - nchf-convergedcharging
          - nchf-offlineonlycharging
          - nnwdaf-eventssubscription
          - nnwdaf-analyticsinfo
          - ngmlc-loc
          - nucmf-provisioning
          - nucmf-uecapabilitymanagement
          - nhss-sdm
          - nhss-uecm
          - nhss-ueau
          - nhss-ee
          - nhss-ims-sdm
          - nhss-ims-uecm
          - nhss-ims-ueau
          - nsepp-telescopic
          - nsoraf-sor
          - nspaf-secured-packet
          - nudsf-dr
          - nnssaaf-nssaa
      - type: string
  N2InterfaceAmfInfo:
    description: AMF N2 interface information
    type: object
    properties:
      ipv4EndpointAddress:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
        minItems: 1
      ipv6EndpointAddress:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Addr'
        minItems: 1
      amfName:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/AmfName'
  NFServiceStatus:
    description: Status of a given NF Service Instance of an NF Instance stored in NRF
    anyOf:
      - type: string
        enum:
          - REGISTERED
          - SUSPENDED
          - UNDISCOVERABLE
      - type: string
  TaiRange:
    description: Range of TAIs (Tracking Area Identities)
    type: object
    required:
      - plmnId
      - tacRangeList
    properties:
      plmnId:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
      tacRangeList:
        type: array
        items:
          $ref: '#/components/schemas/TacRange'
        minItems: 1
      nid:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Nid'
```

```
    TacRange:
      description: Range of TACs (Tracking Area Codes)
      type: object
      properties:
        start:
          type: string
          pattern: '^([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6})$'
        end:
          type: string
          pattern: '^([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6})$'
        pattern:
          type: string
    ChfServiceInfo:
      description: Information of primary and secondary CHF services
      type: object
      not:
        required: [ primaryChfServiceInstance, secondaryChfServiceInstance ]
      properties:
        primaryChfServiceInstance:
          type: string
        secondaryChfServiceInstance:
          type: string
    PlmnRange:
      description: Range of PLMN IDs
      type: object
      properties:
        start:
          type: string
          pattern: '^[0-9]{3}[0-9]{2,3}$'
        end:
          type: string
          pattern: '^[0-9]{3}[0-9]{2,3}$'
        pattern:
          type: string
    NrfInfo:
      description: Information of an NRF NF Instance, used in hierarchical NRF deployments
      type: object
      properties:
        servedUdrInfo:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/UdrInfo'
          minProperties: 1
        servedUdrInfoList:
          type: object
          additionalProperties:
            type: object
            additionalProperties:
              $ref: '#/components/schemas/UdrInfo'
            minProperties: 1
          minProperties: 1
        servedUdmInfo:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/UdmInfo'
          minProperties: 1
        servedUdmInfoList:
          type: object
          additionalProperties:
            type: object
            additionalProperties:
              $ref: '#/components/schemas/UdmInfo'
            minProperties: 1
          minProperties: 1
        servedAusfInfo:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/AusfInfo'
          minProperties: 1
        servedAusfInfoList:
          type: object
          additionalProperties:
            type: object
            additionalProperties:
              $ref: '#/components/schemas/AusfInfo'
            minProperties: 1
          minProperties: 1
        servedAmfInfo:
```

```
      type: object
      additionalProperties:
        $ref: '#/components/schemas/AmfInfo'
      minProperties: 1
    servedAmfInfoList:
      type: object
      additionalProperties:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/AmfInfo'
        minProperties: 1
      minProperties: 1
    servedSmfInfo:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/SmfInfo'
      minProperties: 1
    servedSmfInfoList:
      type: object
      additionalProperties:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/SmfInfo'
        minProperties: 1
      minProperties: 1
    servedUpfInfo:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/UpfInfo'
      minProperties: 1
    servedUpfInfoList:
      type: object
      additionalProperties:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/UpfInfo'
        minProperties: 1
      minProperties: 1
    servedPcfInfo:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/PcfInfo'
      minProperties: 1
    servedPcfInfoList:
      type: object
      additionalProperties:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/PcfInfo'
        minProperties: 1
      minProperties: 1
    servedBsfInfo:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/BsfInfo'
      minProperties: 1
    servedBsfInfoList:
      type: object
      additionalProperties:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/BsfInfo'
        minProperties: 1
      minProperties: 1
    servedChfInfo:
      type: object
      additionalProperties:
        $ref: '#/components/schemas/ChfInfo'
      minProperties: 1
    servedChfInfoList:
      type: object
      additionalProperties:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/ChfInfo'
        minProperties: 1
      minProperties: 1
    servedNefInfo:
```

```
        type: object
        additionalProperties:
          $ref: '#/components/schemas/NefInfo'
        minProperties: 1
      servedNwdafInfo:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/NwdafInfo'
        minProperties: 1
      servedPcscfInfoList:
        type: object
        additionalProperties:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/PcscfInfo'
          minProperties: 1
        minProperties: 1
      servedGmlcInfo:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/GmlcInfo'
        minProperties: 1
      servedLmfInfo:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/LmfInfo'
        minProperties: 1
      servedNfInfo:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/NfInfo'
        minProperties: 1
      servedHssInfoList:
        type: object
        additionalProperties:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/HssInfo'
          minProperties: 1
        minProperties: 1
      servedUdsfInfo:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/UdsfInfo'
        minProperties: 1
      servedUdsfInfoList:
        type: object
        additionalProperties:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/UdsfInfo'
          minProperties: 1
        minProperties: 1
      servedScpInfoList:
        type: object
        additionalProperties:
          $ref: '#/components/schemas/ScpInfo'
        minProperties: 1
  PlmnSnssai:
    description: List of network slices (S-NSSAIs) for a given PLMN ID
    type: object
    required:
      - plmnId
      - sNssaiList
    properties:
      plmnId:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
      sNssaiList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/ExtSnssai'
        minItems: 1
      nid:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Nid'
  NefInfo:
    description: Information of an NEF NF Instance
    type: object
    properties:
```

```
      nefId:
        $ref: '#/components/schemas/NefId'
      pfdData:
        $ref: '#/components/schemas/PfdData'
      afEeData:
        $ref: '#/components/schemas/AfEventExposureData'
      gpsiRanges:
        type: array
        items:
          $ref: '#/components/schemas/IdentityRange'
        minItems: 1
      externalGroupIdentifiersRanges:
        type: array
        items:
          $ref: '#/components/schemas/IdentityRange'
        minItems: 1
      servedFqdnList:
        type: array
        items:
          type: string
        minItems: 1
  PfdData:
    description: List of Application IDs and/or AF IDs managed by a given NEF Instance
    type: object
    properties:
      appIds:
        type: array
        items:
          type: string
        minItems: 1
      afIds:
        type: array
        items:
          type: string
        minItems: 1
  NwdafInfo:
    description: Information of a NWDAF NF Instance
    type: object
    properties:
      eventIds:
        type: array
        items:
          $ref: 'TS29520_Nnwdaf_AnalyticsInfo.yaml#/components/schemas/EventId'
        minItems: 1
      nwdafEvents:
        type: array
        items:
          $ref: 'TS29520_Nnwdaf_EventsSubscription.yaml#/components/schemas/NwdafEvent'
        minItems: 1
      taiList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Tai'
        minItems: 1
      taiRangeList:
        type: array
        items:
          $ref: '#/components/schemas/TaiRange'
        minItems: 1
  LmfInfo:
    description: Information of an LMF NF Instance
    type: object
    properties:
      servingClientTypes:
        type: array
        items:
          $ref: 'TS29572_Nlmf_Location.yaml#/components/schemas/ExternalClientType'
        minItems: 1
      lmfId:
        $ref: 'TS29572_Nlmf_Location.yaml#/components/schemas/LMFIdentification'
      servingAccessTypes:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/AccessType'
        minItems: 1
      servingAnNodeTypes:
        type: array
        items:
```

```
          $ref: '#/components/schemas/AnNodeType'
        minItems: 1
      servingRatTypes:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/RatType'
        minItems: 1
  GmlcInfo:
    description: Information of a GMLC NF Instance
    type: object
    properties:
      servingClientTypes:
        type: array
        items:
          $ref: 'TS29572_Nlmf_Location.yaml#/components/schemas/ExternalClientType'
        minItems: 1
  AfEventExposureData:
    description: AF Event Exposure data managed by a given NEF Instance
    type: object
    required:
      - afEvents
    properties:
      afEvents:
        type: array
        items:
          $ref: 'TS29517_Naf_EventExposure.yaml#/components/schemas/AfEvent'
        minItems: 1
      afIds:
        type: array
        items:
          type: string
        minItems: 1
      appIds:
        type: array
        items:
          type: string
        minItems: 1
  PcscfInfo:
    description: Information of a P-CSCF NF Instance
    type: object
    properties:
      accessType:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/AccessType'
        minItems: 1
      dnnList:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Dnn'
        minItems: 1
  NfInfo:
    description: Information of a generic NF Instance
    type: object
    properties:
      nfType:
        $ref: '#/components/schemas/NFType'
  HssInfo:
    description: Information of an HSS NF Instance
    type: object
    properties:
      groupId:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/NfGroupId'
      imsRanges:
        type: array
        items:
          $ref: '#/components/schemas/ImsiRange'
        minItems: 1
  ImsiRange:
    description: A range of IMSIs (subscriber identities), either based on a numeric range, or
based on regular-expression matching
    type: object
    properties:
      start:
        type: string
        pattern: '^[0-9]+$'
      end:
        type: string
```

```
        pattern: '^[0-9]+$'
      pattern:
        type: string
  TwifInfo:
    description: Addressing information (IP addresses, FQDN) of the TWIF
    type: object
    properties:
      ipv4EndpointAddresses:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
        minItems: 1
      ipv6EndpointAddresses:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Addr'
        minItems: 1
      endpointFqdn:
        $ref: '#/components/schemas/Fqdn'
  VendorId:
    description: Vendor ID of the NF Service instance (Private Enterprise Number assigned by IANA)
    type: string
    pattern: '^[0-9]{6}$'
  VendorSpecificFeature:
    description: Information about a vendor-specific feature
    type: object
    required:
      - featureName
      - featureVersion
    properties:
      featureName:
        type: string
      featureVersion:
        type: string
  AnNodeType:
    description: Access Network Node Type (gNB, ng-eNB...)
    anyOf:
      - type: string
        enum:
          - GNB
          - NG_ENB
      - type: string
  UdsfInfo:
    description: Information related to UDSF
    type: object
    properties:
      groupId:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/NfGroupId'
      supiRanges:
        type: array
        items:
          $ref: '#/components/schemas/SupiRange'
        minItems: 1
  ScpInfo:
    description: Information of an SCP Instance
    type: object
    properties:
      scpDomainInfoList:
        type: object
        additionalProperties:
          items:
            $ref: '#/components/schemas/ScpDomainInfo'
        minProperties: 1
      addressDomains:
        type: array
        items:
          type: string
        minItems: 1
      ipv4Addresses:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
        minItems: 1
      ipv6Prefixes:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Prefix'
        minItems: 1
```

```
        ipv4AddrRanges:
          type: array
          items:
            $ref: '#/components/schemas/Ipv4AddressRange'
          minItems: 1
        ipv6PrefixRanges:
          type: array
          items:
            $ref: '#/components/schemas/Ipv6PrefixRange'
          minItems: 1
        servedNfSetIdList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/NfSetId'
          minItems: 1
        servedNfTypeList:
          type: array
          items:
            $ref: '#/components/schemas/NFType'
          minItems: 1
        remotePlmnList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
          minItems: 1

    ScpDomainInfo:
      description: SCP Domain specific information
      type: object
      properties:
        scpFqdn:
          $ref: '#/components/schemas/Fqdn'
        scpIpEndPoints:
          type: array
          items:
            $ref: '#/components/schemas/IpEndPoint'
          minItems: 1

    ScpDomainCond:
      type: object
      properties:
        scpDomains:
          type: array
          items:
            type: string
          minItems: 1

    OptionsResponse:
      type: object
      properties:
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'

    ConditionEventType:
      description: Indicates whether a notification is due to the NF Instance to start or stop being
part of a condition for a subscription to a set of NFs
      anyOf:
        - type: string
          enum:
            - NF_ADDED
            - NF_REMOVED
        - type: string
```

A.3 Nnrf_NFDiscovery API

```
openapi: 3.0.0

info:
  version: '1.1.0'
  title: 'NRF NFDiscovery Service'
  description: |
    NRF NFDiscovery Service.
    © 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).
    All rights reserved.

externalDocs:
```

```
  description: 3GPP TS 29.510 V16.4.0; 5G System; Network Function Repository Services; Stage 3
  url: 'http://www.3gpp.org/ftp/Specs/archive/29_series/29.510/'

servers:
  - url: '{apiRoot}/nnrf-disc/v1'
    variables:
      apiRoot:
        default: https://example.com
        description: apiRoot as defined in clause 4.4 of 3GPP TS 29.501

security:
  - {}
  - oAuth2ClientCredentials:
      - nnrf-disc

paths:
  /nf-instances:
    get:
      summary: Search a collection of NF Instances
      operationId: SearchNFInstances
      tags:
        - NF Instances (Store)
      parameters:
        - name: Accept-Encoding
          in: header
          description: Accept-Encoding, described in IETF RFC 7231
          schema:
            type: string
        - name: target-nf-type
          in: query
          description: Type of the target NF
          required: true
          schema:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NFType'
        - name: requester-nf-type
          in: query
          description: Type of the requester NF
          required: true
          schema:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NFType'
        - name: requester-nf-instance-id
          in: query
          description: NfInstanceId of the requester NF
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
        - name: service-names
          in: query
          description: Names of the services offered by the NF
          schema:
            type: array
            items:
              $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/ServiceName'
            minItems: 1
            uniqueItems: true
          style: form
          explode: false
        - name: requester-nf-instance-fqdn
          in: query
          description: FQDN of the requester NF
          schema:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/Fqdn'
        - name: target-plmn-list
          in: query
          description: Id of the PLMN of the target NF
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
                minItems: 1
        - name: requester-plmn-list
          in: query
          description: Id of the PLMN where the NF issuing the Discovery request is located
          content:
            application/json:
              schema:
                type: array
```

```
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
        minItems: 1
- name: target-nf-instance-id
  in: query
  description: Identity of the NF instance being discovered
  schema:
    $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
- name: target-nf-fqdn
  in: query
  description: FQDN of the NF instance being discovered
  schema:
    $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/Fqdn'
- name: hnrf-uri
  in: query
  description: Uri of the home NRF
  schema:
    $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
- name: snssais
  in: query
  description: Slice info of the target NF
  content:
    application/json:
      schema:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Snssai'
        minItems: 1
- name: requester-snssais
  in: query
  description: Slice info of the requester NF
  content:
    application/json:
      schema:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Snssai'
        minItems: 1
- name: plmn-specific-snssai-list
  in: query
  description: PLMN specific Slice info of the target NF
  content:
    application/json:
      schema:
        type: array
        items:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/PlmnSnssai'
        minItems: 1
- name: dnn
  in: query
  description: Dnn supported by the BSF, SMF or UPF
  schema:
    $ref: 'TS29571_CommonData.yaml#/components/schemas/Dnn'
- name: nsi-list
  in: query
  description: NSI IDs that are served by the services being discovered
  schema:
    type: array
    items:
      type: string
    minItems: 1
  style: form
  explode: false
- name: smf-serving-area
  in: query
  schema:
    type: string
- name: tai
  in: query
  description: Tracking Area Identity
  content:
    application/json:
      schema:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Tai'
- name: amf-region-id
  in: query
  description: AMF Region Identity
  schema:
```

```
      $ref: 'TS29571_CommonData.yaml#/components/schemas/AmfRegionId'
  - name: amf-set-id
    in: query
    description: AMF Set Identity
    schema:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/AmfSetId'
  - name: guami
    in: query
    description: Guami used to search for an appropriate AMF
    content:
      application/json:
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Guami'
  - name: supi
    in: query
    description: SUPI of the user
    schema:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/Supi'
  - name: ue-ipv4-address
    in: query
    description: IPv4 address of the UE
    schema:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
  - name: ip-domain
    in: query
    description: IP domain of the UE, which supported by BSF
    schema:
      type: string
  - name: ue-ipv6-prefix
    in: query
    description: IPv6 prefix of the UE
    schema:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Prefix'
  - name: pgw-ind
    in: query
    description: Combined PGW-C and SMF or a standalone SMF
    schema:
      type: boolean
  - name: pgw
    in: query
    description: PGW FQDN of a combined PGW-C and SMF
    schema:
      $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/Fqdn'
  - name: gpsi
    in: query
    description: GPSI of the user
    schema:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/Gpsi'
  - name: external-group-identity
    in: query
    description: external group identifier of the user
    schema:
      $ref: 'TS29503_Nudm_SDM.yaml#/components/schemas/ExtGroupId'
  - name: internal-group-identity
    in: query
    description: internal group identifier of the user
    schema:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/GroupId'
  - name: pfd-data
    in: query
    description: PFD data
    content:
      application/json:
        schema:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/PfdData'
  - name: data-set
    in: query
    description: data set supported by the NF
    schema:
      $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/DataSetId'
  - name: routing-indicator
    in: query
    description: routing indicator in SUCI
    schema:
      type: string
      pattern: '^[0-9]{1,4}$'
  - name: group-id-list
    in: query
```

```
          description: Group IDs of the NFs being discovered
          schema:
            type: array
            items:
              $ref: 'TS29571_CommonData.yaml#/components/schemas/NfGroupId'
            minItems: 1
          style: form
          explode: false
        - name: dnai-list
          in: query
          description: Data network access identifiers of the NFs being discovered
          schema:
            type: array
            items:
              $ref: 'TS29571_CommonData.yaml#/components/schemas/Dnai'
            minItems: 1
          style: form
          explode: false
        - name: pdu-session-types
          in: query
          description: list of PDU Session Type required to be supported by the target NF
          schema:
            type: array
            items:
              $ref: 'TS29571_CommonData.yaml#/components/schemas/PduSessionType'
            minItems: 1
          style: form
          explode: false
        - name: event-id-list
          in: query
          description: Analytics event(s) requested to be supported by the Nnwdaf_AnalyticsInfo
service
          schema:
            type: array
            items:
              $ref: 'TS29520_Nnwdaf_AnalyticsInfo.yaml#/components/schemas/EventId'
            minItems: 1
          style: form
          explode: false
        - name: nwdaf-event-list
          in: query
          description: Analytics event(s) requested to be supported by the Nnwdaf_EventsSubscription
service.
          schema:
            type: array
            items:
              $ref: 'TS29520_Nnwdaf_EventsSubscription.yaml#/components/schemas/NwdafEvent'
            minItems: 1
          style: form
          explode: false
        - name: supported-features
          in: query
          description: Features required to be supported by the target NF
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        - name: upf-iwk-eps-ind
          in: query
          description: UPF supporting interworking with EPS or not
          schema:
            type: boolean
        - name: chf-supported-plmn
          in: query
          description: PLMN ID supported by a CHF
          content:
            application/json:
              schema:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
        - name: preferred-locality
          in: query
          description: preferred target NF location
          schema:
            type: string
        - name: access-type
          in: query
          description: AccessType supported by the target NF
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/AccessType'
        - name: limit
```

```
          in: query
          description: Maximum number of NFProfiles to return in the response
          required: false
          schema:
            type: integer
            minimum: 1
        - name: required-features
          in: query
          description: Features required to be supported by the target NF
          schema:
            type: array
            items:
              $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
            minItems: 1
          style: form
          explode: false
        - name: complex-query
          in: query
          description: the complex query condition expression
          content:
            application/json:
              schema:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/ComplexQuery'
        - name: max-payload-size
          in: query
          description: Maximum payload size of the response expressed in kilo octets
          required: false
          schema:
            type: integer
            maximum: 2000
            default: 124
        - name: max-payload-size-ext
          in: query
          description: Extended query for maximum payload size of the response expressed in kilo
octets
          required: false
          schema:
            type: integer
            default: 124
        - name: atsss-capability
          in: query
          description: ATSSS Capability
          content:
            application/json:
              schema:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/AtsssCapability'
        - name: upf-ue-ip-addr-ind
          in: query
          description: UPF supporting allocating UE IP addresses/prefixes
          schema:
            type: boolean
        - name: client-type
          in: query
          description: Requested client type served by the NF
          content:
            application/json:
              schema:
                $ref: 'TS29572_Nlmf_Location.yaml#/components/schemas/ExternalClientType'
        - name: lmf-id
          in: query
          description: LMF identification to be discovered
          content:
            application/json:
              schema:
                $ref: 'TS29572_Nlmf_Location.yaml#/components/schemas/LMFIdentification'
        - name: an-node-type
          in: query
          description: Requested AN node type served by the NF
          content:
            application/json:
              schema:
                $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/AnNodeType'
        - name: rat-type
          in: query
          description: Requested RAT type served by the NF
          content:
            application/json:
              schema:
```

```
        $ref: 'TS29571_CommonData.yaml#/components/schemas/RatType'
    - name: preferred-tai
      in: query
      description: preferred Tracking Area Identity
      content:
        application/json:
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Tai'
    - name: preferred-nf-instances
      in: query
      description: preferred NF Instances
      schema:
        type: array
        items:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
        minItems: 1
      style: form
      explode: false
    - name: If-None-Match
      in: header
      description: Validator for conditional requests, as described in IETF RFC 7232, 3.2
      schema:
        type: string
    - name: target-snpn
      in: query
      description: Target SNPN Identity
      content:
        application/json:
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnIdNid'
    - name: requester-snpn-list
      in: query
      description: SNPN ID(s) of the NF instance issuing the Discovery request
      content:
        application/json:
          schema:
            type: array
            items:
              $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnIdNid'
            minItems: 1
    - name: af-ee-data
      in: query
      description: NEF exposured by the AF
      content:
        application/json:
          schema:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/AfEventExposureData'
    - name: w-agf-info
      in: query
      description: UPF collocated with W-AGF
      content:
        application/json:
          schema:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/WAgfInfo'
    - name: tngf-info
      in: query
      description: UPF collocated with TNGF
      content:
        application/json:
          schema:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/TngfInfo'
    - name: twif-info
      in: query
      description: UPF collocated with TWIF
      content:
        application/json:
          schema:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/TwifInfo'
    - name: target-nf-set-id
      in: query
      description: Target NF Set ID
      schema:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/NfSetId'
    - name: target-nf-service-set-id
      in: query
      description: Target NF Service Set ID
      schema:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/NfServiceSetId'
```

```
    - name: nef-id
      in: query
      description: NEF ID
      schema:
        $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NefId'
    - name: notification-type
      in: query
      description: Notification Type
      schema:
        $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NotificationType'
    - name: serving-scope
      in: query
      description: areas that can be served by the target NF
      schema:
        type: array
        items:
          type: string
        minItems: 1
      style: form
      explode: false
    - name: imsi
      in: query
      description: IMSI of the requester UE to search for an appropriate NF (e.g. HSS)
      schema:
        type: string
    - name: preferred-api-versions
      in: query
      description: Preferred API version of the services to be discovered
      content:
        application/json:
          schema:
            type: object
            additionalProperties:
              type: string
            minProperties: 1
    - name: v2x-support-ind
      in: query
      description: PCF supports V2X
      schema:
        type: boolean
    - name: redundant-gtpu
      in: query
      description: UPF supports redundant gtp-u to be discovered
      schema:
        type: boolean
    - name: redundant-transport
      in: query
      description: UPF supports redundant transport path to be discovered
      schema:
        type: boolean
    - name: ipups
      in: query
      description: UPF which is configured for IPUPS functionality to be discovered
      schema:
        type: boolean
    - name: scp-domain-list
      in: query
      description: SCP domains the target SCP belongs to
      schema:
        type: array
        items:
          type: string
        minItems: 1
      style: form
      explode: false
    - name: address-domain
      in: query
      description: Address domain reachable through the SCP
      schema:
        $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/Fqdn'
    - name: ipv4-addr
      in: query
      description: IPv4 address reachable through the SCP
      schema:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
    - name: ipv6-prefix
      in: query
      description: IPv6 prefix reachable through the SCP
```

```
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Prefix'
        - name: served-nf-set-id
          in: query
          description: NF Set ID served by the SCP
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/NfSetId'
        - name: served-nf-type
          in: query
          description: NF type of NFs served by the SCP
          schema:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NFType'
        - name: remote-plmn-id
          in: query
          description: Id of the PLMN reachable through the SCP
          content:
            application/json:
              schema:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
        - name: data-forwarding
          in: query
          description: UPF Instance(s) configured for data forwarding are requested
          schema:
            type: boolean
        - name: preferred-full-plmn
          in: query
          description: NF Instance(s) serving the full PLMN are preferred
          schema:
            type: boolean
        - name: requester-features
          in: query
          description: Features supported by the NF Service Consumer that is invoking the
Nnrf_NFDiscovery service
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      responses:
        '200':
          description: Expected response to a valid request
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/SearchResult'
          links:
            search:
              operationId: RetrieveStoredSearch
              parameters:
                searchId: $response.body#/searchId
              description: >
                The 'searchId' parameter returned in the response can be used as the
                'searchId' parameter in the GET request to '/searches/{searchId}'
            completeSearch:
              operationId: RetrieveCompleteSearch
              parameters:
                searchId: $response.body#/searchId
              description: >
                The 'searchId' parameter returned in the response can be used as the
                'searchId' parameter in the GET request to '/searches/{searchId}/complete'
          headers:
            Cache-Control:
              description: Cache-Control containing max-age, described in IETF RFC 7234, 5.2
              schema:
                type: string
            ETag:
              description: Entity Tag containing a strong validator, described in IETF RFC 7232, 2.3
              schema:
                type: string
            Content-Encoding:
              description: Content-Encoding, described in IETF RFC 7231
              schema:
                type: string
        '307':
          description: Temporary Redirect
          headers:
            Location:
              description: 'The URI pointing to the resource located on the redirect target NRF'
              required: true
              schema:
                type: string
```

```
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '406':
          $ref: 'TS29571_CommonData.yaml#/components/responses/406'
        '411':
          $ref: 'TS29571_CommonData.yaml#/components/responses/411'
        '413':
          $ref: 'TS29571_CommonData.yaml#/components/responses/413'
        '415':
          $ref: 'TS29571_CommonData.yaml#/components/responses/415'
        '429':
          $ref: 'TS29571_CommonData.yaml#/components/responses/429'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '501':
          $ref: 'TS29571_CommonData.yaml#/components/responses/501'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'

  /searches/{searchId}:
    get:
      operationId: RetrieveStoredSearch
      tags:
        - Stored Search (Document)
      parameters:
        - $ref: '#/components/parameters/searchId'
        - name: Accept-Encoding
          in: header
          description: Accept-Encoding, described in IETF RFC 7231
          schema:
            type: string
      responses:
        '200':
          $ref: '#/components/responses/200'

  /searches/{searchId}/complete:
    get:
      operationId: RetrieveCompleteSearch
      tags:
        - Complete Stored Search (Document)
      parameters:
        - $ref: '#/components/parameters/searchId'
        - name: Accept-Encoding
          in: header
          description: Accept-Encoding, described in IETF RFC 7231
          schema:
            type: string
      responses:
        '200':
          $ref: '#/components/responses/200'

components:
  securitySchemes:
    oAuth2ClientCredentials:
      type: oauth2
      flows:
        clientCredentials:
          tokenUrl: '/oauth2/token'
          scopes:
            nnrf-disc: Access to the Nnrf_NFDiscovery API
  parameters:
    searchId:
      name: searchId
      in: path
      description: Id of a stored search
      required: true
      schema:
        type: string
  responses:
    '200':
```

```
      description: Expected response to a valid request
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/StoredSearchResult'
      headers:
        Cache-Control:
          description: Cache-Control containing max-age, described in IETF RFC 7234, 5.2
          schema:
            type: string
        ETag:
          description: Entity Tag containing a strong validator, described in IETF RFC 7232, 2.3
          schema:
            type: string
        Content-Encoding:
          description: Content-Encoding, described in IETF RFC 7231
          schema:
            type: string
  schemas:
    SearchResult:
      description: Contains the list of NF Profiles returned in a Discovery response
      type: object
      required:
        - nfInstances
      properties:
        validityPeriod:
          type: integer
        nfInstances:
          type: array
          items:
            $ref: '#/components/schemas/NFProfile'
        searchId:
          type: string
        numNfInstComplete:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uint32'
        preferredSearch:
          $ref: '#/components/schemas/PreferredSearch'
        nrfSupportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    StoredSearchResult:
      description: Contains a complete search result (i.e. a number of discovered NF Instances),
stored by NRF as a consequence of a prior search result
      type: object
      required:
        - nfInstances
      properties:
        nfInstances:
          type: array
          items:
            $ref: '#/components/schemas/NFProfile'
    NFProfile:
      description: Information of an NF Instance discovered by the NRF
      type: object
      required:
        - nfInstanceId
        - nfType
        - nfStatus
      properties:
        nfInstanceId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
        nfInstanceName:
          type: string
        nfType:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NFType'
        nfStatus:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NFStatus'
        plmnList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
          minItems: 1
        sNssais:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/ExtSnssai'
          minItems: 1
        perPlmnSnssaiList:
          type: array
```

```
    items:
      $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/PlmnSnssai'
    minItems: 1
  nsiList:
    type: array
    items:
      type: string
    minItems: 1
  fqdn:
    $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/Fqdn'
  ipv4Addresses:
    type: array
    items:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv4Addr'
    minItems: 1
  ipv6Addresses:
    type: array
    items:
      $ref: 'TS29571_CommonData.yaml#/components/schemas/Ipv6Addr'
    minItems: 1
  capacity:
    type: integer
    minimum: 0
    maximum: 65535
  load:
    type: integer
    minimum: 0
    maximum: 100
  loadTimeStamp:
    $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
  locality:
    type: string
  priority:
    type: integer
    minimum: 0
    maximum: 65535
  udrInfo:
    $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/UdrInfo'
  udrInfoList:
    type: object
    additionalProperties:
      $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/UdrInfo'
    minProperties: 1
  udmInfo:
    $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/UdmInfo'
  udmInfoList:
    type: object
    additionalProperties:
      $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/UdmInfo'
    minProperties: 1
  ausfInfo:
    $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/AusfInfo'
  ausfInfoList:
    type: object
    additionalProperties:
      $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/AusfInfo'
    minProperties: 1
  amfInfo:
    $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/AmfInfo'
  amfInfoList:
    type: object
    additionalProperties:
      $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/AmfInfo'
    minProperties: 1
  smfInfo:
    $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/SmfInfo'
  smfInfoList:
    type: object
    additionalProperties:
      $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/SmfInfo'
    minProperties: 1
  upfInfo:
    $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/UpfInfo'
  upfInfoList:
    type: object
    additionalProperties:
      $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/UpfInfo'
    minProperties: 1
```

```
        pcfInfo:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/PcfInfo'
        pcfInfoList:
          type: object
          additionalProperties:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/PcfInfo'
          minProperties: 1
        bsfInfo:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/BsfInfo'
        bsfInfoList:
          type: object
          additionalProperties:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/BsfInfo'
          minProperties: 1
        chfInfo:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/ChfInfo'
        chfInfoList:
          type: object
          additionalProperties:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/ChfInfo'
          minProperties: 1
        udsfInfo:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/UdsfInfo'
        udsfInfoList:
          type: object
          additionalProperties:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/UdsfInfo'
          minProperties: 1
        nwdafInfo:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NwdafInfo'
        nefInfo:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NefInfo'
        pcscfInfoList:
          type: object
          additionalProperties:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/PcscfInfo'
          minProperties: 1
        hssInfoList:
          type: object
          additionalProperties:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/HssInfo'
          minProperties: 1
        customInfo:
          type: object
        recoveryTime:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
        nfServicePersistence:
          type: boolean
          default: false
        nfServices:
          deprecated: true
          type: array
          items:
            $ref: '#/components/schemas/NFService'
          minItems: 1
        nfServiceList:
          type: object
          additionalProperties:
            $ref: '#/components/schemas/NFService'
          minProperties: 1
        defaultNotificationSubscriptions:
          type: array
          items:
            $ref:
'TS29510_Nnrf_NFManagement.yaml#/components/schemas/DefaultNotificationSubscription'
        lmfInfo:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/LmfInfo'
        gmlcInfo:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/GmlcInfo'
        snpnList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnIdNid'
          minItems: 1
        nfSetIdList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/NfSetId'
```

```
          minItems: 1
        servingScope:
          type: array
          items:
            type: string
          minItems: 1
        lcHSupportInd:
          type: boolean
          default: false
        olcHSupportInd:
          type: boolean
          default: false
        nfSetRecoveryTimeList:
          type: object
          additionalProperties:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
          minProperties: 1
        serviceSetRecoveryTimeList:
          type: object
          additionalProperties:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
          minProperties: 1
        scpDomains:
          type: array
          items:
            type: string
          minItems: 1
        scpInfo:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/ScpInfo'
    NFService:
      description: Information of a given NF Service Instance; it is part of the NFProfile of an NF
Instance discovered by the NRF
      type: object
      required:
        - serviceInstanceId
        - serviceName
        - versions
        - scheme
        - nfServiceStatus
      properties:
        serviceInstanceId:
          type: string
        serviceName:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/ServiceName'
        versions:
          type: array
          items:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NFServiceVersion'
          minItems: 1
        scheme:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/UriScheme'
        nfServiceStatus:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NFServiceStatus'
        fqdn:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/Fqdn'
        ipEndPoints:
          type: array
          items:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/IpEndPoint'
          minItems: 1
        apiPrefix:
          type: string
        defaultNotificationSubscriptions:
          type: array
          items:
            $ref:
'TS29510_Nnrf_NFManagement.yaml#/components/schemas/DefaultNotificationSubscription'
          minItems: 1
        capacity:
          type: integer
          minimum: 0
          maximum: 65535
        load:
          type: integer
          minimum: 0
          maximum: 100
        loadTimeStamp:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
```

```
        priority:
          type: integer
          minimum: 0
          maximum: 65535
        recoveryTime:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
        chfServiceInfo:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/ChfServiceInfo'
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        nfServiceSetIdList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/NfServiceSetId'
          minItems: 1
        sNssais:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/ExtSnssai'
          minItems: 1
        perPlmnSnssaiList:
          type: array
          items:
            $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/PlmnSnssai'
          minItems: 1
        vendorId:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/VendorId'
        supportedVendorSpecificFeatures:
          type: object
          additionalProperties:
            type: array
            items:
              $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/VendorSpecificFeature'
          minProperties: 1
        oauth2Required:
          type: boolean
        allowedOperationsPerNfType:
          type: object
          additionalProperties:
            type: array
            items:
              type: string
            minItems: 1
        allowedOperationsPerNfInstance:
          type: object
          additionalProperties:
            type: array
            items:
              type: string
            minItems: 1
    PreferredSearch:
      description: Contains information on whether the returned NFProfiles match the preferred query
parameters
      type: object
      properties:
        preferredTaiMatchInd:
          type: boolean
          default: false
        preferredFullPlmnMatchInd:
          type: boolean
          default: false
```

A.4 Nnrf_AccessToken API (NRF OAuth2 Authorization)

```
openapi: 3.0.0

info:
  version: '1.1.0'
  title: 'NRF OAuth2'
  description: |
    NRF OAuth2 Authorization.
    © 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).
    All rights reserved.

externalDocs:
  description: 3GPP TS 29.510 V16.4.0; 5G System; Network Function Repository Services; Stage 3
```

```
  url: 'http://www.3gpp.org/ftp/Specs/archive/29_series/29.510/'

paths:
  /oauth2/token:
    post:
      summary: Access Token Request
      operationId: AccessTokenRequest
      tags:
        - Access Token Request
      parameters:
        - name: Content-Encoding
          in: header
          description: Content-Encoding, described in IETF RFC 7231
          schema:
            type: string
        - name: Accept-Encoding
          in: header
          description: Accept-Encoding, described in IETF RFC 7231
          schema:
            type: string
      requestBody:
        content:
          application/x-www-form-urlencoded:
            schema:
              $ref: '#/components/schemas/AccessTokenReq'
        required: true
      responses:
        '200':
          description: Successful Access Token Request
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/AccessTokenRsp'
          headers:
            Cache-Control:
              $ref: '#/components/headers/cache-control'
            Pragma:
              $ref: '#/components/headers/pragma'
            Accept-Encoding:
              description: Accept-Encoding, described in IETF RFC 7694
              schema:
                type: string
            Content-Encoding:
              description: Content-Encoding, described in IETF RFC 7231
              schema:
                type: string
        '307':
          description: Temporary Redirect
        '400':
          description: Error in the Access Token Request
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/AccessTokenErr'
          headers:
            Cache-Control:
              $ref: '#/components/headers/cache-control'
            Pragma:
              $ref: '#/components/headers/pragma'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '411':
          $ref: 'TS29571_CommonData.yaml#/components/responses/411'
        '413':
          $ref: 'TS29571_CommonData.yaml#/components/responses/413'
        '415':
          $ref: 'TS29571_CommonData.yaml#/components/responses/415'
        '429':
          $ref: 'TS29571_CommonData.yaml#/components/responses/429'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '501':
          $ref: 'TS29571_CommonData.yaml#/components/responses/501'
        '503':
```

```
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
components:
  headers:
    cache-control:
      required: true
      schema:
        type: string
        enum:
          - no-store
    pragma:
      required: true
      schema:
        type: string
        enum:
          - no-cache
  schemas:
    AccessTokenReq:
      description: Contains information related to the access token request
      type: object
      required:
        - grant_type
        - nfInstanceId
        - scope
      properties:
        grant_type:
          type: string
          enum:
            - client_credentials
        nfInstanceId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
        nfType:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NFType'
        targetNfType:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NFType'
        scope:
          type: string
          pattern: '^([a-zA-Z0-9_:-]+)( [a-zA-Z0-9_:-]+)*$'
        targetNfInstanceId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
        requesterPlmn:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
        requesterPlmnList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
          minItems: 2
        requesterSnssaiList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Snssai'
          minItems: 1
        requesterFqdn:
          $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/Fqdn'
        requesterSnpnList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnIdNid'
          minItems: 1
        targetPlmn:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
        targetSnssaiList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Snssai'
          minItems: 1
        targetNsiList:
          type: array
          items:
            type: string
          minItems: 1
        targetNfSetId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfSetId'
        targetNfServiceSetId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfServiceSetId'
    AccessTokenRsp:
      description: Contains information related to the access token response
```

```
      type: object
      required:
        - access_token
        - token_type
      properties:
        access_token:
          type: string
          description: JWS Compact Serialized representation of JWS signed JSON object
(AccessTokenClaims)
        token_type:
          type: string
          enum:
            - Bearer
        expires_in:
          type: integer
        scope:
          type: string
          pattern: '^([a-zA-Z0-9_:-]+)( [a-zA-Z0-9_:-]+)*$'
    AccessTokenClaims:
      description: The claims data structure for the access token
      type: object
      required:
        - iss
        - sub
        - aud
        - scope
        - exp
      properties:
        iss:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
        sub:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
        aud:
          anyOf:
            - $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/NFType'
            - type: array
              items:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
              minItems: 1
        scope:
          type: string
          pattern: '^([a-zA-Z0-9_:-]+)( [a-zA-Z0-9_:-]+)*$'
        exp:
          type: integer
        consumerPlmnId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
        producerPlmnId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/PlmnId'
        producerSnssaiList:
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Snssai'
          minItems: 1
        producerNsiList:
          type: array
          items:
            type: string
          minItems: 1
        producerNfSetId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfSetId'
    AccessTokenErr:
      description: Error returned in the access token response message
      type: object
      required:
        - error
      properties:
        error:
          type: string
          enum:
            - invalid_request
            - invalid_client
            - invalid_grant
            - unauthorized_client
            - unsupported_grant_type
            - invalid_scope
        error_description:
          type: string
        error_uri:
```

```
          type: string
```

A.5 Nnrf_Bootstrapping API

```
openapi: 3.0.0

info:
  version: '1.0.0'
  title: 'NRF Bootstrapping'
  description: |
    NRF Bootstrapping.
    © 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).
    All rights reserved.

externalDocs:
  description: 3GPP TS 29.510 V16.4.0; 5G System; Network Function Repository Services; Stage 3
  url: 'http://www.3gpp.org/ftp/Specs/archive/29_series/29.510/'

paths:
  /bootstrapping:
    get:
      summary: Bootstrapping Info Request
      operationId: BootstrappingInfoRequest
      tags:
        - Bootstrapping Request
      responses:
        '200':
          description: Successful Bootstrapping Request
          content:
            application/3gppHal+json:
              schema:
                $ref: '#/components/schemas/BootstrappingInfo'
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'

components:
  schemas:
    BootstrappingInfo:
      description: Information returned by NRF in the bootstrapping response message
      type: object
      required:
        - _links
      properties:
        status:
          $ref: '#/components/schemas/Status'
        _links:
          type: object
          description: 'Map of link objects where the keys are the link relations defined in 3GPP TS
29.510 clause 6.4.6.3.3'
          additionalProperties:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/LinksValueSchema'
          minProperties: 1
    Status:
      description: Overal status of the NRF
      anyOf:
        - type: string
          enum:
            - OPERATIVE
            - NON_OPERATIVE
        - type: string
```

Annex B (normative): NF Profile changes in NFRegister and NFUpdate (NF Profile Complete Replacement) responses

B.1 General

In the NFRegister and NFUpdate (NF Profile Complete Replacement) service operations, a NF Service Consumer may indicate to the NRF that it supports receiving NF Profile changes in the response from the NRF, by including the nfProfileChangesSupportInd attribute set to "true" in the NFProfile it registers to or replaces in the NRF.

The NRF may return NF Profile changes, instead of the complete NF Profile, in NFRegister or NFUpdate (NF Profile Complete Replacement) responses, if the NF Service Consumer indicated corresponding support in the request. When doing so, the NRF shall include in the NF Profile returned in the response:

- attributes that are mandatory to include in the NF Profile; if an optional IE is included (e.g. nfServices), attributes that are mandatory to include in this optional IE (e.g. serviceInstanceId) shall also be included;
- optional or conditional IEs that have been changed or added by the NRF; and
- the nfProfileChangesInd IE set to "true", indicating that the returned profile contains NF profile changes.

EXAMPLE 1: The NRF does not change the NF Profile received in the request.

The NRF response contains a NFProfile with just the following IEs:

- nfInstanceId, nfType, nfStatus; and
- nfProfileChangesInd IE set to "true".

EXAMPLE 2: The NRF modifies or adds the heartbeatTimer attribute to the NF Profile received in the request.

The NRF response contains a NFProfile with just the following IEs:

- nfInstanceId, nfType, nfStatus;
- heartbeatTimer with NRF chosen value;
- nfProfileChangesInd IE set to "true".

Annex C (informative): Change history

Date	Meeting	TDoc.	CR	Rev	Cat	Subject/Comment	New
2017-10	CT4#80	C4-175271				Initial draft	0.1.0
2017-10	CT4#80	C4-175395				Incorporation of agreed pCRs from CT4#80: C4-175109, C4-175272, C4-175274, C4-175363	0.2.0
2017-12	CT4#81	C4-176438				Incorporation of agreed pCRs from CT4#81: C4-176184, C4-176278, C4-176280, C4-176281, C4-176282	0.3.0
2018-01	CT4#82	C4-181392				Incorporation of agreed pCRs from CT4#82: C4-181348, C4-181351	0.4.0
2018-03	CT4#83	C4-182435				Incorporation of agreed pCRs from CT4#83: C4-182098, C4-182327, C4-182328, C4-182365, C4-182413	0.5.0
2018-04	CT4#84	C4-183517				Incorporation of agreed pCRs from CT4#84: C4-183450, C4-183451, C4-183452, C4-183487, C4-183488, C4-183490, C4-183491	0.6.0
2018-05	CT4#85	C4-184625				Incorporation of agreed pCRs from CT4#85: C4-184207, C4-184208, C4-184280, C4-184466, C4-184469, C4-184478, C4-184517, C4-184519, C4-184545, C4-184595, C4-184596, C4-184597, C4-184600, C4-184615, C4-184616, C4-184626	0.7.0
2018-06	CT#80	CP-181105				Presented for information and approval	1.0.0
2018-06	CT#80					Approved in CT#80.	15.0.0
2018-09	CT#81	CP-182012	0001	2	F	Implementing the Indirect Delivery method for the GET method to retrieve NF instances	15.1.0
2018-09	CT#81	CP-182093	0003	3	F	Defining the range of the priority and capacity attributes and aligning their usage with SRV RFC 2782	15.1.0
2018-09	CT#81	CP-182060	0004	-	F	Corrections to descriptions, references and SUPI parameter in Discovery Request	15.1.0
2018-09	CT#81	CP-182047	0006	2	F	SubscriptionData	15.1.0
2018-09	CT#81	CP-182045	0008	2	F	Error Cases	15.1.0
2018-09	CT#81	CP-182060	0009	2	F	Heart Beat Procedure	15.1.0
2018-09	CT#81	CP-182060	0010	1	B	Vendor-Specific NF Types	15.1.0
2018-09	CT#81	CP-182044	0011	3	F	Presence condition of service discovery query parameters	15.1.0
2018-09	CT#81	CP-182060	0012	4	F	Description of Inter-PLMN scenarios	15.1.0
2018-09	CT#81	CP-182060	0013	1	F	NF Service Versions	15.1.0
2018-09	CT#81	CP-182060	0014	1	B	Custom Headers	15.1.0
2018-09	CT#81	CP-182060	0015	1	F	Overall Clean-up	15.1.0
2018-09	CT#81	CP-182060	0016	-	F	Formatting of query parameters	15.1.0
2018-09	CT#81	CP-182060	0017	-	F	Editorial corrections	15.1.0
2018-09	CT#81	CP-182060	0018	2	F	Backup AMF	15.1.0
2018-09	CT#81	CP-182060	0020	1	B	NF Service Names	15.1.0
2018-09	CT#81	CP-182060	0023	-	F	CHF as service consumer	15.1.0
2018-09	CT#81	CP-182060	0024	3	B	Hierarchical NF discovery in recursion mode	15.1.0
2018-09	CT#81	CP-182060	0025	2	B	Hierarchical NF discovery in iteration mode	15.1.0
2018-09	CT#81	CP-182060	0026	-	F	Correction of Allowed NF Domains	15.1.0
2018-09	CT#81	CP-182060	0027	-	F	Correction of BsfInfo data type	15.1.0
2018-09	CT#81	CP-182161	0028	1	F	IPv6 Prefix for NF / NF Service Address	15.1.0
2018-09	CT#81	CP-182060	0030	1	B	NF Set Id	15.1.0
2018-09	CT#81	CP-182060	0031	1	F	URI Scheme	15.1.0
2018-09	CT#81	CP-182060	0032	2	B	NRF service registration	15.1.0
2018-09	CT#81	CP-182060	0034	2	F	Discovery of combined SMF and PGW-C	15.1.0
2018-09	CT#81	CP-182163	0035	3	F	Support TAI Range for AMF/SMF and SUPI Range for PCF	15.1.0
2018-09	CT#81	CP-182060	0036	1	F	SUPI Range for PCF	15.1.0
2018-09	CT#81	CP-182164	0037	3	F	Scope for OAuth 2.0 Access Token Request	15.1.0
2018-09	CT#81	CP-182060	0039	1	F	Corrections to NotificationData and "supi" parameter in Discovery Request	15.1.0
2018-09	CT#81	CP-182060	0040	1	F	Group ID in Discovery Request	15.1.0
2018-09	CT#81	CP-182060	0041	1	F	Registering multiple Routing Indicators	15.1.0
2018-09	CT#81	CP-182060	0045	-	F	Description of Structured data types	15.1.0
2018-09	CT#81	CP-182060	0046	1	F	Service names in Discovery Request	15.1.0
2018-09	CT#81	CP-182060	0047	1	F	Resource structure presentation	15.1.0
2018-09	CT#81	CP-182060	0048	-	B	Default Notifications for UDM	15.1.0
2018-09	CT#81	CP-182060	0049	-	F	Cell ID in Discovery Request	15.1.0
2018-09	CT#81	CP-182046	0050	2	F	NRF Subscription Data	15.1.0
2018-09	CT#81	CP-182060	0051	1	F	AMF Discovery by 5G-AN	15.1.0
2018-09	CT#81	CP-182060	0052	1	F	Detecting NF Failure and Restart using the NRF	15.1.0
2018-09	CT#81	CP-182060	0053	2	B	NRF Subscription Lifespan	15.1.0
2018-09	CT#81	CP-182060	0054	1	F	NRF servers clause in OpenAPI	15.1.0
2018-09	CT#81	CP-182060	0056	2	F	Default port number	15.1.0
2018-09	CT#81	CP-182162	0057	1	F	AMF Discovery Based on AMF Name	15.1.0
2018-09	CT#81	CP-182060	0058	-	F	API Version Update	15.1.0
2018-12	CT#82	CP-183018	0060	4	F	Heartbeat Timer	15.2.0
2018-12	CT#82	CP-183018	0061	1	F	Location Header	15.2.0
2018-12	CT#82	CP-183018	0062	2	F	NF Profile Addressing Parameters	15.2.0
2018-12	CT#82	CP-183018	0063	1	F	NRF Notifications	15.2.0
2018-12	CT#82	CP-183018	0064	-	F	Oauth2 Corrections	15.2.0
2018-12	CT#82	CP-183018	0065	1	F	Regular Expression Patterns	15.2.0
2018-12	CT#82	CP-183183	0066	5	F	Subscription Data	15.2.0

2018-12	CT#82	CP-183147	0067	2	F	UPF selection based on DNAI	15.2.0
2018-12	CT#82	CP-183018	0068	5	F	CHF registration and selection	15.2.0
2018-12	CT#82	CP-183018	0069	1	F	Clarify the NRF management functionality in the case of hierarchical NRFs	15.2.0
2018-12	CT#82	CP-183149	0070	5	F	OAuth2.0 Service Alignments and Corrections	15.2.0
2018-12	CT#82	CP-183150	0071	1	F	HTTP Basic Authentication For OAuth2.0 Access Token Request	15.2.0
2018-12	CT#82	CP-183018	0072	1	F	Multiple PLMNs support	15.2.0
2018-12	CT#82	CP-183018	0075	2	F	NFService attribute in NFProfile	15.2.0
2018-12	CT#82	CP-183018	0076	1	F	Corrections of ServiceName enumeration	15.2.0
2018-12	CT#82	CP-183018	0077	4	F	Indicating support of EPS interworking in UPF Profile	15.2.0
2018-12	CT#82	CP-183018	0079	2	F	Cardinality	15.2.0
2018-12	CT#82	CP-183018	0081	-	F	APIRoot Clarification	15.2.0
2018-12	CT#82	CP-183018	0082	2	F	Clarification on the reuse of the previous search results	15.2.0
2018-12	CT#82	CP-183018	0083	1	F	NF profile detail for hierarchical NRF	15.2.0
2018-12	CT#82	CP-183235	0084	3	F	Complex query	15.2.0
2018-12	CT#82	CP-183152	0087	1	F	SMF discovery based on S-NSSAI and DNN	15.2.0
2018-12	CT#82	CP-183153	0088	2	F	CHF discovery based on GPSI and SUPI	15.2.0
2018-12	CT#82	CP-183146	0089	3	F	Add access type in SMF selection	15.2.0
2018-12	CT#82	CP-183018	0090	2	F	Hierarchical subscription with intermediate forwarding NRF	15.2.0
2018-12	CT#82	CP-183018	0091	2	F	Hierarchical subscription with intermediate redirecting NRF	15.2.0
2018-12	CT#82	CP-183018	0093	1	F	Notifications for subscriptions via intermediate NRF	15.2.0
2018-12	CT#82	CP-183018	0096	1	F	DNN and IP Domain in BSF Info	15.2.0
2018-12	CT#82	CP-183018	0097	1	F	PCF Information	15.2.0
2018-12	CT#82	CP-183018	0100	-	F	NF Service FQDN	15.2.0
2018-12	CT#82	CP-183018	0101	-	F	NRF Corrections	15.2.0
2018-12	CT#82	CP-183018	0102	1	F	Notification Data	15.2.0
2018-12	CT#82	CP-183181	0103	2	F	NRF Oauth Scopes	15.2.0
2018-12	CT#82	CP-183182	0104	1	F	NRF Subscription Handling	15.2.0
2018-12	CT#82	CP-183018	0105	-	F	NF Profile Change Notification	15.2.0
2018-12	CT#82	CP-183171	0107	-	F	UDM Group ID	15.2.0
2018-12	CT#82	CP-183018	0108	2	F	Preferred target NF Location in Discovery Request	15.2.0
2018-12	CT#82	CP-183018	0109	1	F	Telescopic FQDN for HNRF	15.2.0
2018-12	CT#82	CP-183184	0112	1	F	Description of NF instances/NF profile retrieval	15.2.0
2018-12	CT#82	CP-183018	0113	-	F	Content of the Subscription to notification response	15.2.0
2018-12	CT#82	CP-183018	0115	-	F	Adding new services in ServiceName enumeration	15.2.0
2018-12	CT#82	CP-183018	0116	-	F	NF Profile Service Instances	15.2.0
2018-12	CT#82	CP-183018	0117	-	F	API Version	15.2.0
2018-12	CT#82	CP-183180	0118	1	F	ExternalDocs Update	15.2.0
2019-03	CT#83	CP-190023	0119	1	F	AmfRegionId and AmfSetId	15.3.0
2019-03	CT#83	CP-190023	0120	1	F	Interpretation of absence of IEs in NF Profile	15.3.0
2019-03	CT#83	CP-190023	0121	1	F	Usage of FQDN and IP address related attributes from NF / NF Service profiles	15.3.0
2019-03	CT#83	CP-190023	0122	1	F	AMF Region and AMF Set in PLMNs supporting multiple PLMN Ids	15.3.0
2019-03	CT#83	CP-190023	0123	1	F	Encoding of GUAMI query parameter in NFDiscover Request	15.3.0
2019-03	CT#83	CP-190023	0124	1	F	Status for operative NF (service) not discoverable by other NFs	15.3.0
2019-03	CT#83	CP-190023	0126	1	F	Limiting the number of NFProfiles returned in NFDiscover response	15.3.0
2019-03	CT#83	CP-190023	0127	2	F	Maximum payload size of NFDiscover Response	15.3.0
2019-03	CT#83	CP-190155	0128	2	F	NF Profile Changes in NF Register / NFUpdate Response	15.3.0
2019-03	CT#83	CP-190023	0129	1	F	supported-features query parameter of NFDiscover Request	15.3.0
2019-03	CT#83	CP-190023	0130	1	F	OpenAPI Corrections	15.3.0
2019-03	CT#83	CP-190023	0132	1	F	Oauth2 Token Claims	15.3.0
2019-03	CT#83	CP-190023	0133	1	F	Oauth2 Token Type	15.3.0
2019-03	CT#83	CP-190023	0134	1	F	Authorization Attributes of NF Profile	15.3.0
2019-03	CT#83	CP-190023	0135	2	F	Features of NF Discovery service	15.3.0
2019-03	CT#83	CP-190023	0136	-	F	Subscription Authorization for Sets of NFs	15.3.0
2019-03	CT#83	CP-190023	0137	1	F	S-NSSAI per PLMN	15.3.0
2019-03	CT#83	CP-190059	0138	4	F	UPF selection based on PDUSessionType	15.3.0
2019-03	CT#83	CP-190163	0139	2	F	Service Names in URI Query Parameters	15.3.0
2019-03	CT#83	CP-190023	0140	1	F	GMLC URI for Namf_Location EventNotify	15.3.0
2019-03	CT#83	CP-190023	0141	1	F	Corrections on complex query	15.3.0
2019-03	CT#83	CP-190023	0142	1	F	NRF Notifications	15.3.0
2019-03	CT#83	CP-190023	0143	1	F	NRF Heart-Beat	15.3.0
2019-03	CT#83	CP-190023	0144	-	F	Addition of new Service Name	15.3.0
2019-03	CT#83	CP-190023	0145	-	F	API version update	15.3.0
2019-06	CT#84	CP-191034	0146	3	F	PLMN ID in Access Token Claims	15.4.0
2019-06	CT#84	CP-191034	0147	1	F	Content encodings supported in HTTP requests	15.4.0
2019-06	CT#84	CP-191034	0156	3	F	Correct the condition of the FQDN parameter of NFProfile and NFService	15.4.0
2019-06	CT#84	CP-191034	0159	1	F	NRF Service Description	15.4.0
2019-06	CT#84	CP-191034	0161	1	F	Slice Info in NRF	15.4.0
2019-06	CT#84	CP-191034	0162	1	F	Subscription Conditions	15.4.0
2019-06	CT#84	CP-191034	0163	1	F	Vendor-Specific IEs in NF Profile	15.4.0
2019-06	CT#84	CP-191034	0165	2	F	Target PLMN List in Inter-PLMN Service Discovery	15.4.0

2019-06	CT#84	CP-191034	0167	2	F	Storage of OpenAPI specification files	15.4.0
2019-06	CT#84	CP-191034	0170	-	F	Corrections on NFStatusUnSubscribe operation to take into account multiple NRFs	15.4.0
2019-06	CT#84	CP-191034	0171	1	F	Corrections on UpdateSubscription operation to take into account multiple NRFs	15.4.0
2019-06	CT#84	CP-191034	0174	4	F	Corrections on Nnrf_AccessToken Service for multiple NRFs	15.4.0
2019-06	CT#84	CP-191034	0176	-	F	LowerCamel Correction in Data Structures	15.4.0
2019-06	CT#84	CP-191034	0177	-	F	Removal of Basic Authentication	15.4.0
2019-06	CT#84	CP-191034	0178	1	F	Location header in redirect response	15.4.0
2019-06	CT#84	CP-191034	0185	2	F	Add HTTP error codes in 29.510	15.4.0
2019-09	CT#85	CP-192108	0192	4	F	Add selection mechanism for multiple IP addresses in NFProfile	15.4.0
2019-06	CT#84	CP-191034	0194	2	F	Add retrieval of the NF profile using the URI	15.4.0
2019-06	CT#84	CP-191034	0195	1	F	Add the update of subscription in a different PLMN	15.4.0
2019-06	CT#84	CP-191034	0198	1	F	PLMN-IDs in Discovery Response	15.4.0
2019-06	CT#84	CP-191034	0202	-	F	Copyright Note in YAML files	15.4.0
2019-06	CT#84	CP-191034	0206	-	F	3GPP TS 29.510 API version update	15.4.0
2019-06	CT#84	CP-191052	0148	7	B	NWDAF Discovery and Selection	16.0.0
2019-06	CT#84	CP-191057	0149	4	B	Multiple entries of pcfInfo	16.0.0
2019-06	CT#84	CP-191057	0150	3	B	Multiple entries of bsfInfo	16.0.0
2019-06	CT#84	CP-191057	0151	3	B	Multiple entries of smfInfo	16.0.0
2019-06	CT#84	CP-191051	0154	5	B	ATSSS Capability for UPF Selection	16.0.0
2019-06	CT#84	CP-191057	0175	1	B	GPSI range in pcfInfo	16.0.0
2019-06	CT#84	CP-191057	0180	2	B	Multiple entries of xxxInfo (generalized)	16.0.0
2019-06	CT#84	CP-191057	0186	2	F	Add the name of NF Instance	16.0.0
2019-06	CT#84	CP-191057	0187	3	B	Add requester nfInstanceId parameter in NFStatusSubscribe and NFDiscovery operations	16.0.0
2019-06	CT#84	CP-191034	0196	2	F	Correct The subscription notification procedure under the exception case	16.0.0
2019-06	CT#84	CP-191057	0199	1	B	PCF Group ID	16.0.0
2019-06	CT#84	CP-191057	0200	1	B	Number of NF Instances	16.0.0
2019-06	CT#84	CP-191050	0201	1	B	NIDDAU Service Name	16.0.0
2019-06	CT#84	CP-191054	0204	1	B	UE IP address allocation by UPF	16.0.0
2019-06	CT#84	CP-191048	0205	-	B	3GPP TS 29.510 API version update	16.0.0
2019-09	CT#85	CP-192033	0207	3	C	CBCF as Network Function	16.1.0
2019-09	CT#85	CP-192127	0208	1	F	callbackUri the same as nfStatusNotificationUri	16.1.0
2019-09	CT#85	CP-192193	0210	1	B	Extensions for I-SMF and I-UPF selection	16.1.0
2019-09	CT#85	CP-192194	0211	2	B	NF Set and NF Service Set	16.1.0
2019-09	CT#85	CP-192034	0212	2	B	Update NRF descriptions to support AF Available Data Registration as described in TS23.288	16.1.0
2019-09	CT#85	CP-192035	0213	3	B	SMF Selection	16.1.0
2019-09	CT#85	CP-192107	0215	-	A	Expiration Time of AccessTokenClaims	16.1.0
2019-09	CT#85	CP-192109	0217	1	F	Requester PLMN ID in SubscriptionData	16.1.0
2019-09	CT#85	CP-192127	0218	-	F	Correct the conditions of the information included in the access token request	16.1.0
2019-09	CT#85	CP-192107	0222	-	A	Slice Information in Access Token Claims	16.1.0
2019-09	CT#85	CP-192130	0223	1	B	UPF collocated with W-AGF	16.1.0
2019-09	CT#85	CP-192127	0224	2	F	URI in Location header for subscription to NF Instances in a different PLMN	16.1.0
2019-09	CT#85	CP-192249	0226	5	A	Support of Static IP Address	16.1.0
2019-09	CT#85	CP-192133	0228	1	B	Network Identifier for Stand-alone Non-Public Networks	16.1.0
2019-09	CT#85	CP-192127	0229	1	F	SMF profile without the smfInfo attribute	16.1.0
2019-09	CT#85	CP-192107	0231	-	A	Authorization Attributes in NF Service	16.1.0
2019-09	CT#85	CP-192123	0232	-	B	Handling of authorization parameters	16.1.0
2019-09	CT#85	CP-192136	0233	1	B	P-CSCF Discovery	16.1.0
2019-09	CT#85	CP-192123	0235	-	B	GPSI support for PCF Query	16.1.0
2019-09	CT#85	CP-192123	0236	-	B	LMF and GMLC Info	16.1.0
2019-09	CT#85	CP-192123	0238	1	B	NEF discovery information for PFD	16.1.0
2019-09	CT#85	CP-192135	0239	-	B	Services invoked by NWDAF	16.1.0
2019-09	CT#85	CP-192127	0240	1	F	Regulation of load update notifications	16.1.0
2019-09	CT#85	CP-192120	0242	-	F	3GPP TS 29.510 API version update	16.1.0
2019-10						Corrupted references fixed	16.1.1
2019-12	CT#86	CP-193063	0244	-	B	Missing NFs information in NrfInfo	16.2.0
2019-12	CT#86	CP-193059	0245	2	B	P-CSCF Discovery based on preferred-locality	16.2.0
2019-12	CT#86	CP-193049	0247	2	B	NEF ID	16.2.0
2019-12	CT#86	CP-193042	0248	-	F	ExternalDocs Clause	16.2.0
2019-12	CT#86	CP-193036	0250	3	F	Support of Static IP Address	16.2.0
2019-12	CT#86	CP-193063	0251	3	B	NRF Bootstrapping	16.2.0
2019-12	CT#86	CP-193056	0252	3	B	I-SMF selection in a mobility procedure	16.2.0
2019-12	CT#86	CP-193283	0253	3	B	UCMF Registration in NRF	16.2.0
2019-12	CT#86	CP-193057	0254	1	B	Handling of default notification subscriptions with Delegated Discovery	16.2.0
2019-12	CT#86	CP-193063	0255	1	B	Support of different priorities in SMF/UPF profiles for different TAIs	16.2.0
2019-12	CT#86	CP-193031	0256	1	A	External Group ID	16.2.0
2019-12	CT#86	CP-193053	0257	2	B	Internal Group Identifier in UdmInfo	16.2.0
2019-12	CT#86	CP-193054	0258	1	B	HSS Service Discovery	16.2.0

2019-12	CT#86	CP-193063	0259	1	B	NF serving scope	16.2.0
2019-12	CT#86	CP-193031	0260	1	A	S-NSSAI Discovery Parameter	16.2.0
2019-12	CT#86	CP-193063	0261	1	F	NF discovery based on DNN	16.2.0
2019-12	CT#86	CP-193063	0262	1	F	S-NSSAI for SMF or UPF selection	16.2.0
2019-12	CT#86	CP-193055	0264	2	B	GMLC Location service	16.2.0
2019-12	CT#86	CP-193063	0265	-	F	Error code 404 for Hierarchical NRFs	16.2.0
2019-12	CT#86	CP-193063	0266	3	B	Preferred API Version	16.2.0
2019-12	CT#86	CP-193042	0267	3	F	Clarification on Backup AMF Info	16.2.0
2019-12	CT#86	CP-193055	0269	-	B	Location Update Notification Default Subscription	16.2.0
2019-12	CT#86	CP-193044	0273	-	F	3GPP TS 29.510 API version update	16.2.0
2020-03	CT#87	CP-200025	0274	1	B	3GPP Rel-16 LOLC implications on Nnrf service	16.3.0
2020-03	CT#87	CP-200039	0275	2	F	Add Corresponding API descriptions in clause 5.1	16.3.0
2020-03	CT#87	CP-200016	0277	3	F	Service Discovery in a different PLMN using 3gpp-Sbi-Target-apiRoot	16.3.0
2020-03	CT#87	CP-200134	0278	3	B	Data Type Descriptions	16.3.0
2020-03	CT#87	CP-200020	0279	1	D	Editorial corrections in clause headings	16.3.0
2020-03	CT#87	CP-200020	0280	2	B	Service Names	16.3.0
2020-03	CT#87	CP-200029	0281	1	B	SoR Application Function	16.3.0
2020-03	CT#87	CP-200035	0282	3	B	N3 terminations of TWIF for UPF selection	16.3.0
2020-03	CT#87	CP-200044	0283	1	F	Correcting relevant typing errors	16.3.0
2020-03	CT#87	CP-200016	0284	3	B	CHF Group ID	16.3.0
2020-03	CT#87	CP-200101	0285	4	B	S-NSSAIs of an NF Service	16.3.0
2020-03	CT#87	CP-200021	0286	2	F	NFtype enumeration values for MME, SCEF and SCS/AS	16.3.0
2020-03	CT#87	CP-200020	0287	3	F	DNN encoding in NRF APIs	16.3.0
2020-03	CT#87	CP-200047	0288	2	F	Content type of Access Token Request	16.3.0
2020-03	CT#87	CP-200020	0289	2	F	Registering the AccessToken service in another NRF	16.3.0
2020-03	CT#87	CP-200039	0290	3	D	Editorial corrections	16.3.0
2020-03	CT#87	CP-200039	0291	2	F	Correction - formatting consistency	16.3.0
2020-03	CT#87	CP-200020	0293	1	B	29510 CR optionality of ProblemDetails	16.3.0
2020-03	CT#87	CP-200016	0294	1	F	Wrong reference	16.3.0
2020-03	CT#87	CP-200020	0295	1	B	Subscription Condition for UPF	16.3.0
2020-03	CT#87	CP-200026	0297	1	B	OTAF	16.3.0
2020-03	CT#87	CP-200017	0298	1	B	Supported DNN of the I-SMF	16.3.0
2020-03	CT#87	CP-200036	0299	-	B	PCF selection for V2X	16.3.0
2020-03	CT#87	CP-200023	0300	2	B	UPF selection for redundant transmission	16.3.0
2020-03	CT#87	CP-200016	0302	1	B	Service access authorization of a NF Set	16.3.0
2020-03	CT#87	CP-200028	0307	3	B	UDSF registration with NRF	16.3.0
2020-03	CT#87	CP-200016	0308	-	F	API versions supported for default notification subscriptions	16.3.0
2020-03	CT#87	CP-200044	0309	-	F	NF Discovery with intermediate forwarding NRF	16.3.0
2020-03	CT#87	CP-200020	0310	-	F	Modifications in the NRF service APIs for the support of compression	16.3.0
2020-03	CT#87	CP-200020	0311	1	B	Vendor ID in NF Profile	16.3.0
2020-03	CT#87	CP-200177	0312	3	B	LMF selection	16.3.0
2020-03	CT#87	CP-200020	0313	1	B	Load Time Stamp	16.3.0
2020-03	CT#87	CP-200020	0314	1	B	Security Settings	16.3.0
2020-03	CT#87	CP-200016	0315	1	B	NFType for SCP	16.3.0
2020-03	CT#87	CP-200092	0316	-	F	3GPP TS 29.510 API Version Update	16.3.0
2020-07	CT#88	CP-201050	0317	1	B	Support of IPUPS Functionality	16.4.0
2020-07	CT#88	CP-201034	0318	1	F	Authorization parameters in roaming scenarios	16.4.0
2020-07	CT#88	CP-201034	0319	1	F	Missing attributes in NrfInfo data type	16.4.0
2020-07	CT#88	CP-201034	0320	-	F	Slice Differentiator Ranges and Wildcard	16.4.0
2020-07	CT#88	CP-201034	0321	1	F	Undiscoverable NF service	16.4.0
2020-07	CT#88	CP-201057	0322	2	F	Supported Headers and Links Tables	16.4.0
2020-07	CT#88	CP-201030	0323	1	B	Recovery Time for NF Service Set and NF Set	16.4.0
2020-07	CT#88	CP-201022	0324	1	A	Requester-snssais	16.4.0
2020-07	CT#88	CP-201034	0325	2	B	Resource-Level Authorization	16.4.0
2020-07	CT#88	CP-201034	0326	-	B	Data type descriptions	16.4.0
2020-07	CT#88	CP-201034	0327	-	B	Storage of YAML files in ETSI Forge	16.4.0
2020-07	CT#88	CP-201034	0331	1	B	Subscription Condition for a List of NF Instances	16.4.0
2020-07	CT#88	CP-201034	0334	-	B	Serving Scope for NF Subscriptions	16.4.0
2020-07	CT#88	CP-201046	0335	1	B	SMF NIDD Service	16.4.0
2020-07	CT#88	CP-201057	0336	1	F	Datatype column in Resource URI variables Table	16.4.0
2020-07	CT#88	CP-201041	0338	2	F	ServiceName nudsf-dr missing from yaml	16.4.0
2020-07	CT#88	CP-201030	0341	1	F	Presence condition of Set IDs	16.4.0
2020-07	CT#88	CP-201047	0342	1	B	AMF Callback URIs for NSSAA	16.4.0
2020-07	CT#88	CP-201047	0343	1	B	Introduce NSSAAF	16.4.0
2020-07	CT#88	CP-201201	0345	2	F	SCP profile registration and discovery	16.4.0
2020-07	CT#88	CP-201034	0347	1	F	Bootstrapping API	16.4.0
2020-07	CT#88	CP-201034	0348	1	F	Defining xxxInfoExt data types as maps	16.4.0
2020-07	CT#88	CP-201034	0349	-	F	Requester's information in Access Token Request	16.4.0
2020-07	CT#88	CP-201034	0350	-	F	Discovery or subscription requests with missing requester's information	16.4.0
2020-07	CT#88	CP-201030	0354	1	F	Supplement to NefInfo	16.4.0
2020-07	CT#88	CP-201070	0355	1	B	UPF for Data Forwarding	16.4.0

2020-07	CT#88	CP-201031	0356	1	B	V-SMF Selection for Serving Full PLMN	16.4.0
2020-07	CT#88	CP-201045	0357	-	F	NRF Additional Authorization Parameters	16.4.0
2020-07	CT#88	CP-201183	0359	2	F	NRF Notifications	16.4.0
2020-07	CT#88	CP-201034	0360	-	F	NF Services Map	16.4.0
2020-07	CT#88	CP-201034	0361	1	B	NRF Notification Data	16.4.0
2020-07	CT#88	CP-201033	0364	-	B	Service Name for Nhss_EE	16.4.0
2020-07	CT#88	CP-201034	0365	1	F	Maximum payload size	16.4.0
2020-07	CT#88	CP-201073	0368	-	F	3GPP TS 29.510 API Version Update	16.4.0

History

Document history		
V16.4.0	July 2020	Publication