|  |  |
| --- | --- |
| 3GPP TR 29.831 V18.0.0 (2023-09) | |
| Technical Report | |
| 3rd Generation Partnership Project;  Technical Specification Group Core Network and Terminals;  Study on NRF API enhancements to avoid signalling and storing of redundant data  (Release 18) | |
|  | |
|  |  |
|  | |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| ***Copyright Notification***  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2023, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 5

1 Scope 6

2 References 6

3 Definitions of terms, symbols and abbreviations 6

3.1 Terms 6

3.2 Symbols 7

3.3 Abbreviations 7

4 Baseline 7

4.1 General 7

4.2 Data Storage Overhead 7

4.3 Signalling Overhead 7

4.4 Processing Overhead 7

5 Key Issues 8

5.1 Key Issue #1: Avoid Duplicate Configuration, Storage and Transmission of Shareable Data 8

5.1.1 Description of the use case 8

5.1.2 Key issue definition 8

5.2 Key Issue #2: Signaling overhead of NF discovery responses between PLMNs 9

5.2.1 Description of the use case 9

5.2.2 Key Issue definition 9

6 Solutions 9

6.1 Solution#1: Avoid Duplicate Configuration and Transmission of Shareable Data 9

6.1.1 Description 9

6.1.2 Impacts on services, entities and interfaces 11

6.1.3 Pros 11

6.1.4 Cons 11

6.2 Solution#2: Support of NF (Service) Set Profiles 11

6.2.1 Description 11

6.2.1.1 General 11

6.2.1.2 NFManagement API Resources 12

6.2.1.3 NF-Instance Registration 13

6.2.1.4 NF Discovery 15

6.2.1.4.1 Alternative 1 15

6.2.1.4.2 Alternative 2 16

6.2.1.5 NF Set profile subscription 18

6.2.1.6 NF Set profile modification / deletion 19

6.2.2 Impacts on services, entities and interfaces 20

6.3 Solution #3: Download Large Volume Data from NRF 20

6.3.1 Description 20

6.3.2 Impacts on services, entities and interfaces 21

6.3.3 Pros 21

6.3.4 Cons 21

6.4 Solution #4: Enhancement of inter NRF interaction to optimize shareable data transmission 21

6.4.1 Description 21

6.4.2 Impacts on services, entities and interfaces 22

6.4.3 Pros 23

6.4.4 Cons 23

6.5 Solution #5: Optimization of shareable data transmission for NF set level shared data 23

6.5.1 Description 23

6.5.2 Impacts on services, entities and interfaces 24

6.5.3 Pros 24

6.5.4 Cons 24

6.6 Solution #6: NRF Policy Control on Information Exposure for different PLMNs 24

6.6.1 Description 24

6.6.2 Impacts on services, entities and interfaces 25

6.6.3 Pros 26

6.6.4 Cons 26

6.7 Solution#7: List of Shareable Profile Items 26

6.7.1 Description 26

6.7.2 Impacts on services, entities and interfaces 27

6.7.3 Pros 27

6.7.4 Cons 27

6.8 Solution #8: Single Subscription of NF Profile Change from an NF Set 27

6.8.1 Description 27

6.8.2 Impacts on services, entities and interfaces 28

6.8.3 Pros 28

6.8.4 Cons 28

6.9 Solution #9: Single NF Profile Change Notification to an NF Set 28

6.9.1 Description 28

6.9.2 Impacts on services, entities and interfaces 29

6.9.3 Pros 29

6.9.4 Cons 29

6.10 Solution #10: Simplifying the NF service discovery response of inter-PLMN 29

6.10.1 Description 29

6.10.2 Impacts on services, entities and interfaces 30

6.10.3 Pros 31

6.10.4 Cons 31

6.11 Solution 11: Avoid large configuration of UE Routing Indicator 31

6.11.1 Description 31

6.11.2 Impacts on services, entities and interfaces 31

6.11.3 Pros 31

6.11.4 Cons 31

6.12 Solution #12: Support for NF Set level shared data 32

6.12.1 Description 32

6.12.2 Impacts on services, entities and interfaces 34

6.12.3 Pros 34

6.12.4 Cons 34

7 Evaluations 34

7.1 Evaluation of Solutions for Key Issue#1 34

7.1.1 Synchronization of Shareable Data 34

7.1.1.1 Configured Shareable Data 34

7.1.1.2 Discovered Shareable Data 35

7.1.1.3 Change of Shareable Data 35

7.1.2 Storage requirements 35

7.1.2.1 Storage of configured data 35

7.1.2.2 Storage of discovered data 35

7.1.3 Discovery procedure 36

7.1.4 Which data should be shareable? 36

7.1.5 New NF services operations 36

7.1.6 Applicability Scope 36

7.1.7 Backward compatibility 37

7.1.8 Comparison 37

7.2 Evaluation of Solutions for Key Issue#2 38

8 Conclusions 39

8.1 Conclusion of Solutions for Key Issue#1 39

8.2 Conclusion of Solutions for Key Issue#2 39

Annex A (informative): Change history 40

# Foreword

This Technical Report 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 identifies deficiencies of the NRF APIs as specified in 3GPP TS 29.510 [2] and studies potential optimization solutions. The document will focus on the following aspects:

-avoid data storage overhead in the NRF due to many NFs (especially NFs from the same NF set) registering with identical information,

-avoid signalling overhead (both in terms of payload size and number of HTTP requests) due to discovery responses containing lots of duplicate information and due to the number of data change notification sent when duplicated data are changed one by one.

-avoid processing overhead (both at NRF and its consumers) resulting e.g. from signalling overhead.

# 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 29.510: "Network Function Repository Services; Stage 3".

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

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms 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].

void

## 3.2 Symbols

For the purposes of the present document, the following symbols apply:

void

## 3.3 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].

void

# 4 Baseline

## 4.1 General

In order to assist mobile network’s needs of service governance (e.g. topology selection, signalling routing), the NRF API system is designed to support mechanisms including service registration, discovery, subscription, and notification. However, in current NRF framework, duplicated network-level attribute values and UE ranges are separately maintained between different NFs, and transmissions of full NF profiles are repeatedly performed, resulting in overhead of data storage, signalling, and processing. See Section 4.2, 4.3, and 4.4 for details.

## 4.2 Data Storage Overhead

Firstly, in the mobile network, to achieve load balancing and high availability, it is common that multiple NFs serving the same geographical region (e.g., NFs within a same NF Set) are assigned by O&M with identical network-level service attribute values (e.g. TAI-list, DNN). In this case, the NFs’ registration to a same NRF introduces redundant storage of many partially or fully duplicated NF profiles.

Secondly, current NRF API system maintains UE identifier ranges in NFs, although the attribute of the ranges is a global concept rather than per-NF’s. In large-scale network, the attribute can contain millions of UEs and achieve several million bytes, resulting in NFs’ storage overhead. Moreover, when these NFs perform registration, multiple copies of the same ranges will occur simultaneously in the NRF’s storage.

## 4.3 Signalling Overhead

Data size is the first reason for signalling overhead. The current NRF API system supports only single-layer query view. During registration, discovery and notification processes, full NF profiles with huge size have to be submitted or retrieved at one time. Take signalling routing in some large-scale network as an example, huge full information is provided, containing millions of UE ranges, leading to huge-size (typically several million bytes) frame overhead. Moreover, as stated in Section 4.2, these huge NF profiles can contain a lot of duplicated data, which further makes the overhead unnecessary.

Transmission range and frequency is another two reasons for signalling overhead. In current NRF framework, even the tiny change of a subscribed network element can trigger large-scale notifications. To distribute the full NF profile to all related subscribers simultaneously, the NRF can easily trigger a push storm and become a performance bottleneck.

## 4.4 Processing Overhead

The receiving, parsing, and handling of redundant information as stated in Section 4.2 and 4.3 apparently increase processing overhead. Moreover, when network elements are pulled up in batches and concurrent registration or discovery happens, the intensive transmission of many full NF profiles will make not only the NRF but also NFs suffer from performance fluctuations.

# 5 Key Issues

## 5.1 Key Issue #1: Avoid Duplicate Configuration, Storage and Transmission of Shareable Data

### 5.1.1 Description of the use case

In existing NF profile stored in the NRF, operator may configure same data part of the NF profile with same value to multiple NFs, as per network management requirement.

For example, multiple NFs (e.g. AUSF/UDM/PCF/BSF/CHF) may be configured to serve same range of UEs and thus be configured with same UE identifier ranges (e.g. SUPI/GPSI/External ID ranges) in each NF profile. These UE identifier ranges configured in the NF profile may have large volume, and operator needs to configure such large volume data to those indicated multiple NFs (e.g. same type of NFs in a NF Set, or even different type of NFs).

As another example, NFs of an NF set share typically a significant set of common data. Common information include the authorization parameters (“allowedxxx parameters”), the served PLMN or SNPN, the supported S-NSSAIs, SCP information, part or all of NF specific data such as amfInfo (e.g. list of TAIs), udminfo (e.g. subscriber identity ranges), smfinfo (e.g. all S-NSSAIs/DNNs configuration supported by the SMF set).

Such duplicate data (e.g. UE identifier ranges or NF instance profiles) introduces complexity to the NRF API and OAM system (e.g. to manage the configuring, updating, and downloading of duplicate data (e.g. UE identifier ranges or NF instance profiles). Meanwhile, this duplicate data (e.g. UE identifier ranges or NF instance profiles) reduces the signaling efficiency of NRF API, especially when the duplicate data (e.g. UE identifier ranges or NF instance profiles) have large volume e.g. upper to x million octets.

### 5.1.2 Key issue definition

This key issue will study the following aspects:

- Which data part in the NF profile can be organized as shared data and used by multiple NFs and whether such categorization needs to be standardized?

- How to configure/register shared data (e.g. UE identifier ranges) in the NRF so that multiple NF profiles can share the same data (e.g. UE identifier ranges) and avoid duplicate storage/configuration/registration;

- How to optimize download of shared data (e.g. UE identifier ranges) to the requesting NF during NF discovery procedure or NF profile change notification procedure;

- How to handle the download shared data (e.g. UE identifier ranges) with large data volume, e.g. upper to x million octets (extremely upper to 16 million octets exceeding the maximum data size of JSON object).

- How to leverage the NF (Service) Set concept to avoid duplicate configuration and storage of NF profiles within the NFs belonging to the same NF set, and to avoid duplicate transmission of NF profiles for NFs belonging to the same NF set.

- How the shared data can be obtained by the NRF in case the shared data corresponding to the shared-data ID is not available in the NRF.

- How to avoid sending multiple NF profile change notifications from/to NF-Instances which may or may not be within an NF-Set, when a common configuration parameter changes in the NF-Profile.

## 5.2 Key Issue #2: Signaling overhead of NF discovery responses between PLMNs

### 5.2.1 Description of the use case

In the NF discovery response, NRF will return the NF profiles matching the search criteria indicated by the query parameters of the discovery request, e.g. all the list of ranges of SUPIs whose profile data is available in the UDM instance, etc.

Take SUPI as an example.

The AMF carries SUPI as one of the main query parameters in the NF discovery request, and initiates NF discovery to the NRF to obtain the information of the UDMs where the SUPI profile is stored. If the query parameters match successfully, the NRF will return the correct UDMs, and will carry all the list of ranges of SUPIs whose profile data is available in the UDM instance. That results in large response messages, even exceeding 2 million octets in some network deployments.

The AMF completely stores the data in the response message in the cache, therefore other SUPIs can be searched locally in the subsequent service discovery procedure, avoiding excessive signaling interaction with the NRF.

However, if the AMF and NRF belong to different PLMNs, the ranges of SUPIs in the cache of AMF might not work. Because there are relatively few users of different SUPI accessing the network from different PLMN. Additionally, the storage costs of saving NF profiles of PLMNs around the world would be high.

In another scenario, when AMF in VPLMN queries SMF in HPLMN, NRF will include all “taiList” and “taiRangeList” served by the H-SMF in the NF discovery response message. However, these TAIs are useless for the AMF in VPLMN.

Consequently, these large response messages between different PLMNs might cause unnecessary signaling overhead and data storage overhead.

It’s recommended considering how to simplify these numerous parameters in NF discovery response of roaming scenarios, e.g., plmnList, sNssais, perPlmnSnssaiList, nsiList, supiRanges, gpsiRanges, routingIndicators, externalGroupIdentifiersRanges, internalGroupIdentifiersRanges, suciInfos, taiList, taiRangeList, dnnList, etc.

### 5.2.2 Key Issue definition

This key issue will study the following aspects:

- Which parameters of NF discovery responses between PLMNs can be simplified and how to simplify them.

- How the NRF processes requests from different PLMNs differently, e.g. AMF of some PLMNs might need the whole supiRanges UDM served.

- How the NF processes simplified response from NRF.

# 6 Solutions

## 6.1 Solution#1: Avoid Duplicate Configuration and Transmission of Shareable Data

### 6.1.1 Description

The solution is to address the Key Issue #1: Avoid Duplicate Configuration and Transmission of Shareable Data.

Duplicate configuration may exist in various data types (e.g. UE identifier ranges) among different NF (or NF service) profiles. Possible shareable data in NF (or NF service) profile are listed in clause 6.7.

Editor’s Note: It is FFS that whether and which data types need to be defined as standardized shared data types.

To avoid duplicate configuration and transmission of shareable data (e.g. UE identifier ranges), the shareable data can be organized as shared data which is separately stored outside of the NF profile. One NF profile using the shared data (e.g. UE identifier ranges) should carry a reference (i.e. shared data identifier) pointing to that shared data. An NF may be associate with multiple shared data with different types, thus multiple instance of shared data identifiers may be registered in one NF profile.

The following enhancements to data structure of the NRF API are proposed:

- Define individual data structures for each type of shared data;

- For example, define a SharedIdRanges data structure, which carries the identifier of the SharedIdRanges and a list of IMSI/MSISDN/SUPI/GPSI/External-Group-ID/Internal-Group-ID ranges, etc;

- Define a SharedDataIdentifier data type as identifier of each type of shared data;

- For example, the SharedIdRanges data structure includes an "sharedDataId" attribute of type SharedDataIdentifier, to indicate the unique identifier of that SharedIdRanges;

- Add attribute in the NF profile to refer to the separately stored shared data;

- For example, add "sharedDataReferences" attribute in the NFProfile of an NF to contain a list of shared data identifiers to multiple types of shared data, and within the "sharedDataReferences" container a "sharedIdRangeRef" attribute refers to one SharedIdRanges. Or, add one attribute "sharedIdRangeRef" directly in the AusfInfo/UdmInfo/PcfInfo/BsfInfo/ChfInfo/etc. to indicate the reference to a SharedIdRanges;

The following enhancements to resource tree of the NRF API are proposed:

- Define individual resources in the resource tree of the NRF API, to provide creation/update/deletion/retrieval operation to each type of shared data;

- For example, a separate resource "/shared-id-ranges" is defined to represent the collection of SharedIdRages. A POST operation to the "/shared-id-ranges" resource will create an instance of SharedIdRanges. A GET operation to the "/shared-id-ranges/{sharedDataRef}" will retrieve the content of the SharedIdRanges identified by the {sharedDataRef}.

- A parent resource "/shared-data" may be used to umbrella all kinds of shared data types. In this case, the resource "/shared-data/shared-id-ranges" represents the collection of SharedIdRanges.

To configure the shared data in the NRF, the following methods should be used:

- An operator should arrange those shareable data (e.g. UE identifier ranges) to shared data in well-organized manner, and register those shared data to the NRF;

- The registration of shared data to the NRF may be done by the operator management system, or by an operator granted NF (e.g. a special UDR). In the latter case, the granted NF invokes POST method to the resource representing the collection of that type of shared data (e.g. to the resource "/shared-id-ranges") to register the shared data.

- An NF can register a reference (i.e. the shared data identifier) to one shared data in its own NF profile, if the NF is configured to share some data with other NFs and is aware of the shared data identifier. Multiple shared data identifiers may be registered in the NF profile to refer to different type of shared data;

To download NF profiles with shared data from the NRF, the following methods should be used:

- A request NF should indicate its support for the SharedData feature to the NRF, in the NF discovery procedure or subscription to NF profile change notification.

- During NF discovery or NF profile change notification procedure, the NRF and the request NF behaves as the following:

- For a request NF supporting the SharedData feature, the NRF only indicates the shared data identifier(s) referring to the shared data (e.g. shared data identifier to a SharedIdRanges) in the NF profile, when sending NF profile the request NF.

- For a request NF not supporting SharedData feature, the NRF shall translate the shared data identifier included in the NF profile to the content of existing attributes in the NF profile, when sending NF profile to the request NF.

- When receiving NF profile with shared data identifier(s), the request NF shall separately invoke GET operation to the NRF resource representing the shared data to retrieval the shared data, if not yet get the shared data.

### 6.1.2 Impacts on services, entities and interfaces

NRF:

- extend the existing NRF API to separately organize the shared data, and provide individual operations to the resource representing the shared UE identifier ranges.

NF:

- invoke separate retrieval to the resource representing the shared data, after receiving NF profile indicating a reference to the shared data from the NRF.

### 6.1.3 Pros

This solution provides an efficient way to configure the shared data to multiple NF profiles, and avoid duplicate transmission of same data configured in multiple NF profiles.

### 6.1.4 Cons

The NRF needs to differentiate the handling to a request NF supporting or not supporting SharedData feature.

## 6.2 Solution#2: Support of NF (Service) Set Profiles

### 6.2.1 Description

#### 6.2.1.1 General

This solution is to address Key Issue #1 with a focus on leveraging the NF (Service) Set concept to avoid duplicate configuration, storage and transmission of NF profiles for NFs belonging to the same NF set.

This solution proposes optional optimizations to the Nnrf\_NFManagement and Nnrf\_NFDiscovery APIs to be controlled by a new feature "NF Set Profile Support".  
Consumers of the Nnrf\_NFManagement service should discover the NRF's support of the new feature "NF Set Profile Support" using the HTTP OPTION method as already defined in 3GPP TS 29.510 [2] clause 5.2.2.2.2.  
Consumers of the Nnrf\_NFDiscovery service shall indicate in discovery request messages whether they support the new feature "NF Set Profile Support".

This solution proposes two alternative options with regard to NF Set profile provisioning at the NRF:

A) NF Set profiles are registered at the NRF by any NF instance that belongs to the corresponding NF set.

B) NF Set profiles are configured at the NRF by OAM.

It is a PLMN operator choice to deploy option A) or B).

This solution proposes two alternatives with regard to NF discovery:

1) The NF Set profile is conveyed from NRF to the consumer within the discovery response message that conveys NF instance profiles which contain the corresponding NF Set Profile ID.

2) The NF Set profile is separately retrieved by the NRF consumer when needed.

#### 6.2.1.2 NFManagement API Resources

This solution proposes to extend the NFManagement API with new resources "nf-set" allowing NF consumers to retrieve (applicable only for alternative 2, see clause 6.2.1.1) NF Profile Set information from the NRF. In addition (only for option A, see clause 6.2.1.1) the new resource "nf-set" allows NF consumers (i.e. NF instances belonging to an NF Set) to register NF Set profile information (i.e. profile information common to all NF instances belonging to the set) at the NRF using the PUT method. Modification of the resource's representation using PATCH and deletion of the resource using DELETE will also be supported.

Proposed modifications to the URI structure of the NFManagement API are shown in red in figure 6.2.1.2-1:



Figure 6.2.1.2-1

Proposed modifications to the Resource and Methods overview are shown in red in table 6.2.1.2-1:

Table 6.2.1.2-1

|  |  |  |  |
| --- | --- | --- | --- |
| 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. |
| nf-set  (Document) | /nf-sets/{nfSetID} | GET only for alternative 2 | Read the profile of a given NF Set. |
| PUT only for option A | Register in NRF a new NF Set, or replace the profile of an existing NF Set, by providing an NF set profile. |
| PATCH only for option A | Modify the NF set profile of an existing NF Set. |
| DELETE only for option A | Deregister from NRF a given NF Set. |
| 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. |

#### 6.2.1.3 NF-Instance Registration

For option A, the NF-Set profile can be registered (is so needed) by any NF instance from the NF Set during NF-Instance registration as shown in figure 6.2.1.3-1. (It is assumed that all NF instances belonging to the same NF Set share some common data e.g. using UDSF. The NF-Set profile would be part of these common data.)

For option B, NF-Set profiles are configured at the NRF by OAM and need not be conveyed from the registering NF instance to the NRF.



Figure 6.2.1.3-1

1. NF instance of a particular NF set sends a PUT request to register the NF instance profile at the NRF by creating a resource with a URI representing the NF Instance ID. The variable nfInstanceID represents the identifier of the registering NF instance. The registering NF instance registers only its NF Set ID and NF instance specific attributes, and not the entire NFSet profile.

2. On receipt of PUT request, if the NRF determines that there is no corresponding NF Set profile already registered (option A) or configured (option B) for the NF Set ID in the NRF, it replies with an error message 400 Bad Request (Problem details: missing NF Set Profile).

3. However, this step and also steps 3, 4 and 5 are skipped if on receipt of PUT request in step 1, NRF determines that there is already a registered or configured NF Set profile for the NF Set ID of the registering NF instance, e.g. in case another NF instance belonging to the same NF set already registered to the NRF or the NF set profile was configured by OAM at the NRF i.e. as per the steps explained in 3 and 4.

4. In case NF instance receives error message in step 2, with details ‘missing NF Set Profile’,  
for option A: NF instance sends a PUT request to the NRF to register the NF Set Profile, by creating a resource representing the NF set profile with a URI identifying the NF set id e.g. …/nf-sets/{nfSetID} (NF Set Profile). For example, this may be the case if this is the first NF instance in this NF set registering to the NRF.  
for option B: NF instance reports an error to OAM, and the NF instance should retry registration with its full NF instance profile, i.e. not making use of the "NF Set Profile Support" feature.

5. On success, NRF replies with "201 Created". The payload body of the PUT response contains the representation of the created resource (NF set profile) and the "Location" header contains the URI of the created resource.

Same as 1.

6. On success, NRF replies with "201 Created". The payload body of the PUT response contains the representation of the created resource (NF profile) and the "Location" header contains the URI of the created resource.

7. Other NF instance(s) of the particular NF set sends a PUT request to register its NF instance, as defined in step 1.

8. On success, in response to the above step 7, NF instance(s) of the particular NF set receives NRF replies with "201 Created".

#### 6.2.1.4 NF Discovery

##### 6.2.1.4.1 Alternative 1

NF service consumer indicates "Set Profile support" in the NF discovery request in order for NRF to provide optimal response, including the NF Set profile. Figure 6.2.1.4.1-1 shows an example call flow.



Figure 6.2.1.4.1-1: NF Discovery Request by NF Service Consumer supporting NF Set Profile

1. The NF Service Consumer sends GET request to the resource URI "nf-instances" collection resource. The input filter criteria for the discovery request are included in query parameters. It provides “Set Profile Support” indicating to the NRF that it supports the "NF Set Profile Support" feature.

2. Based on the received input of “NF Set Profile Support”, the NRF determines that the NF service consumer supports NF set profiles and that requested NFInstance(s) belong to a particular NF Set e.g. NF Set 1.

3. On success, "200 OK" is returned. In its response, the NRF includes an array of NF Profile objects, that satisfy the search filter criteria e.g., all NF Instances offering a certain NF Service name, and additionally includes the NF Set Profile data. NRF does not provide NF Set profile attributes in each of the NF Instance Profile, but instead it indicates NF Set ID in each NF Instance profile.

##### 6.2.1.4.2 Alternative 2

NF service consumer indicates "Set Profile support" in the NF discovery request in order for NRF to provide optimal response, but not including the NF Set profile.

Figure 6.2.1.4.2-1 shows an example call flow, where the NRF does not return the NF Set Profile data in the NF Discovery Response. If the NF consumer does not already have the NF Set Profile corresponding to the NF Set ID indicated in the returned NF profiles, the NF service consumer separately requests to get it from NRF.



Figure 6.2.1.4.2-1: NF Discovery Request by NF Service Consumer supporting NF Set Profile (variant)

1. Same as in Figure 6.2.1.4.1-1.

2. Same as in Figure 6.2.1.4.1-1.

3. On success, "200 OK" is returned. In its response, the NRF includes an array of NF Profile objects, that satisfy the search filter criteria e.g., all NF Instances offering a certain NF Service name. NRF does not provide NF Set profile attributes in each of the NF Instance Profile, but instead it indicates NF Set ID in each NF Instance profile. Here (i.e. in this alternative approach) the NRF does not return the NF set profile (as it might already be available to the NRF consumer from previous requests).

4. On receipt of NRF response in step 3, NF Service consumer determines that it does not already have any stored information about NF Set 1 ID.

5. NF Service consumer sends a request to get NF Set profile corresponding to NF Set 1 ID. As a further optimization retrieval of the NF Set profile and subscription to profile change notifications may be done with a single request (POST).

6. On success, "200 OK" is returned, along with NF Set 1 profile

#### 6.2.1.5 NF Set profile subscription

After having received the NF Set profile (see step 3 in figure 6.2.1.4-1 or step 6 in figure 6.2.1.4-2) the NF service consumer may want to subscribe to NF Set Profile Change Notifications, see figure 6.2.1.5-1.



figure 6.2.1.5-1

1. NF Service Consumer subscribes by sending a POST request to the resource URI representing the "subscriptions" collection resource, with subscription data identifying the NF status changes to be notified, e.g. identifying the need to notify NF status changes affecting the NF service producers of a given NF set.

2. On receipt of the request with NF Set profile support indication in step 1, NRF determines that NF service consumer supports NF set profiles and stores the subscription (including the indication that the subscribing NF supports NF set profiles).

3. On success, "201 Created" is returned to the NF Consumer.

Note: NF Set Profile retrieval and subscription to change notifications can be optimized using a single request message (POST). See clause 6.2.1.4 step 5.

#### 6.2.1.6 NF Set profile modification / deletion

Figure 6.2.1.6-1 describes the procedure to modify/delete the NF set profile of NF instances already registered in the NRF (only applicable for option A, see clause 6.2.1.1). Modification is done by providing the updated NF set profile to the NRF. The update operation may apply to the whole NF set profile i.e. complete replacement of the existing profile by a new profile using PUT, or it may apply only to a subset of the parameters of the NF set profile i.e. including adding/deleting/replacing attributes of the registered NF set profile using PATCH. Deletion is done by using DELETE.

The procedure in figure 6.2.1.6-1 also enables the NRF to notify subscribed NF consumers supporting NF Set profiles of changes (modification or deletion) done to an NF set profile (applicable to both options A and B).



figure 6.2.1.6-1

1. A NF instance from Set 1 performs an update / deletion of the NF Set Profile by sending a PATCH or PUT or DELETE request to the resource URI representing the NF set profile. For update, it provides the changes corresponding to the NF Set Profile.

2. On success, "200 OK" or "204 No Content" is returned.

3. Triggered by steps 1-2 (for option A) or by OAM provisioning (for option B) the NRF determines that it needs to send NFStatusNotify to subscribed NF Service Consumer(s) as shown in figure 6.2.1.5-1.

4. NRF sends NFStatusNotify to subscribed NF Service Consumer(s) (see clause 6.2.1.5) with changes to NF set profile data with update attributes / delete indication.

### 6.2.2 Impacts on services, entities and interfaces

NRF:

- Extend the Nnrf\_NFManagement and Nnrf\_NFDiscovery APIs with the new feature "NF Set Profile Support"

- Support a new resource "nf-set" and offer methods to NRF consumers to read and subscribe (only alternative 2), and (only option A) to register, modify, and delete the new resource.

- Support sending of change notifications to consumers having subscribed

- Mediate between supporting NF consumers registering (and modifying) NF SET profiles and non supporting NF consumers discovering (and subscribing) registered profiles.

NRF consumer:

- Support the new feature when consuming Nnrf\_NFManagement or Nnrf\_NFDiscovery services

- Register /Update / Delete NF Set Profile at the NRF when needed (only option A).

- Retrieve and subscribe to NF Set profiles

OAM procedures:

- Configuration impacts to enable provisioning of NF Set profiles at NF instances.

- Configuration impacts to enable provisioning of NF Set profiles at the NRF (option B only).

## 6.3 Solution #3: Download Large Volume Data from NRF

### 6.3.1 Description

The solution is to address the Key Issue #1: Avoid Duplicate Configuration and Transmission of Shareable Data.

Some special sub-item in the NF profile may be configured with a huge content list. A typical example is the IMSI/SUPI list configured in the AUSF/UDM profiles. If the IMSI/SUPI list cannot be easily organized as reduced IMSI/SUPI range list, the size of the whole list may occupy a large volume of storage and may reaches or even exceeds the maximum size of JSON object (see clause 6.2 of 3GPP TS 29.501). It is the operator’s responsibility to determine what content size is regarded as large volume based on its policy.

A possible way to solve this issue is to use pagination mechanism for downloading such NF profile sub-item with huge content list. Currently, the pagination mechanism has already been introduced in TS29.510 to retrieve part of the entire NF profiles from one type of NF instances (see clause 6.1.3.2.3). Similar mechanism can be utilized to retrieve a specific sub-item (e.g. IMSI/SUPI range list) of a NF profile, if that NF profile sub-item has large volume of content.

To support the pagination downloading mechanism for NF profile sub-items, following enhancements shall be considered:

- 1) When sending NF Discovery request to the NRF, the requesting NF shall:

- (a) indicate its support of paginated data retrieval feature;

- 2) On receiving NF Discovery request from an NF, the NRF shall:

- (a) check if the requesting NF supports paginated data retrieval feature;

- (b) check if at least one NF profile sub-item has large volume of content, perform (c)-(d);

- (c) keep the detected NF profile sub-item with empty content;

- (d) indicate a special pagination reference for retrieving the content of the NF profile sub-item;

- For example, the NRF includes a paginatedDataReference IE in the NF profile. Within the paginatedDataReference IE, it carries: (a) the original resource URI of the NF profile sub-item, (b) a special paginated-data-retrieval URL used to retrieve one page of data of that NF profile inner item, (c) total pages of the NF profile sub-item, (d) page size;

- An example of the paginated-data-retrieval URL:

{apiRoot}/nnrf-nfm/v1/paginated-data-retrieval/{nf-instance-id}/{nf-profile-sub-item}

- 3) On receiving NF Discovery response from the NRF, the requesting NF shall:

- (a) store the NF profile returned by the NRF;

- (b) check if pagination reference is present in the response, perform bullet (c)-(d);

- (c) initiate an separate HTTP GET to the paginated-data-retrieval URL, indicating the target page number;

- An example of HTTP GET request:

GET {apiRoot}/nnrf-nfm/v1/paginated-data-retrieval/{nf-instance-id}/{nf-profile-sub-item}?page-number=1;

- (d) continue (h) to retrieve all pages from the NRF, and assemble all pages to get the entire content of the indicated NF profile sub-item.

### 6.3.2 Impacts on services, entities and interfaces

NRF:

- extend the existing NRF API to support paginated data retrieval.

NF:

- indicate its support of paginated data retrieval feature to the NRF;

- invoke paginated data retrieval to the indicated paginated-data-retrieval URL to retrieve one page of content of a specific NF profile sub-item.

### 6.3.3 Pros

This solution provides an efficient way to download one piece of content of an NF profile sub-item, in the case that the NF profile sub-item has large volume of content.

### 6.3.4 Cons

The NRF needs to differentiate the handling to a requesting NF supporting or not supporting paginated data retrieval feature.

## 6.4 Solution #4: Enhancement of inter NRF interaction to optimize shareable data transmission

### 6.4.1 Description

The solution is to address the Key Issue #1: Avoid Duplicate Configuration and Transmission of Shareable Data.

Duplicate configuration may exist in various data types (e.g. UE identifier ranges) among different NF (or NF service) profiles. Possible shareable data in NF (or NF service) profile are listed in clause 6.7.

Editor’s Note: It is FFS that whether and which data types need to be defined as standardized shared data types.

To avoid repeat transmission of shareable data within the NF profile, it is proposed to classify these shareable data as different set of shared data, identified by shared-data ID, in the same way adopted in subscription data retrieval. Within the NRF interaction related procedures (e.g. NF registration, NF discovery), the shared-data ID will be provided instead of shared data.

As NRF is deployed in a hierarchical architecture, e.g. PLMN level, shared-slice level, and slice-specific level, to configure the shared data on lower level NRF might be not suitable for the fact that NFs may register its profile in one of multiple NRF instances, and to configure the shared data on all NRF instances as global unique information might cause a waste of storage resource. In this solution, it is proposed to configure the shared on a higher level NRF instance, and the corresponding NRF instance which stores this shared data will be provided simultaneously together with the shared-data ID when needed.

The procedure of shared-data ID provisioning would be as below:

- The mapping correlation between shared-data ID and the corresponding shared data is provisioned on a higher level NRF instance.

- Besides, for the NF instances which holds these shared data, the shared-data ID and the NRF instance ID which maintains the mapping correlation is provisioned during the initialization process.

The procedure of NF registration procedure would be as below:

- The NF registers its NF profile to the NRF, in which the NF profile contains one or more shared-data ID(s) and the NRF instance ID which maintains the shared data corresponds to the shared-data ID.

- If the NRF determines that the shared data corresponds to the shared-data ID is not available yet, it initiates the retrieval of shared data towards the NRF instance identified by the NRF instance ID of the shared-data ID.

- The NRF which receives the NF registration request needs to subscribe on the change notification of shared data towards the NRF instance which maintains the shared data.

The procedure of NF discovery procedure would be as below:

- Upon receiving the NF discovery request from a request NF which indicates its support for SharedData feature, the NRF can respond to the request NF with the NF profile which includes the shared-data ID and the corresponding NRF instance ID which maintains the related shared data.

- Upon receiving the NF discovery request from a request NF which does not indicate its support for SharedData feature, the NRF shall translate the shared-data ID within the NF profile into the corresponding shared data, and respond the NF profile to the request NF.

- When the request NF receives NF profile contains shared-data ID, and the corresponding shared data is not available yet, it initiates the retrieval and subscription of shared data towards the NRF instance identified by the NRF instance ID of the shared-data ID, or alternatively, towards the current NRF instance.

As an alternative, if the shared-data ID is unique at PLMN granularity, the mapping between the shared-data ID and corresponding NRF instance ID can be registered outside of the NF profile, and the NRF doesn’t need to provide the NRF instance ID corresponding to the shared-data ID, as the request NF can request and subscribe to the shared data by providing the uniquely assigned shared-data ID.

### 6.4.2 Impacts on services, entities and interfaces

NRF:

- retrieves the shared-data corresponds to the received shared-data ID from the NRF instance identified by the NRF instance ID.

- responds the shared data according to the supported feature of the request NF.

NF:

- registers its NF profile to the NRF, which may include shared-data ID and the corresponding NRF instance ID.

- retrieves the shared-data corresponds to the received shared-data ID from the NRF instance identified by the NRF instance ID or from the current NRF instance.

### 6.4.3 Pros

This solution provided an efficient way to optimize the signalling load during NRF interaction, and allows the shared data being provisioned in a distributed way.

### 6.4.4 Cons

Shared-data ID is tightly bound to the corresponding NRF instance ID which maintains the shared data, and needs to be transmitted together.

## 6.5 Solution #5: Optimization of shareable data transmission for NF set level shared data

### 6.5.1 Description

The solution is to address the Key Issue #1: Avoid Duplicate Configuration and Transmission of Shareable Data.

Duplicate configuration may exist in various data types (e.g. UE identifier ranges) among different NF (or NF service) profiles. Possible shareable data in NF (or NF service) profile are listed in clause 6.7.

Editor’s Note: It is FFS that whether and which data types need to be defined as standardized shared data types.

According to the definition in TS 23.501[3], NF Set is a group of interchangeable NF instances of the same type, supporting the same services and the same network slice(s), and can have access to the same context data even geographically distributed. Under this precondition, it can be assumed that NF profile of different NF instances within the same NF set tend to converge, and would only differ with some NF instance specific data types, e.g. address info, serving area. To optimize the transmission of shared data for an NF set, it is proposed that the NRF can fetch the shared data from the register NF upon receiving the NF set level shared data.

For set level shared data transmission, most of the data type within the NF profile can be identified as a one or more shared-data ID(s), and the shared-data ID can be provided together with the NF Set ID to the NRF during the NF registration procedure, and further to the request NF by the NRF during the NF discovery procedure.

The procedure of shared-data ID provisioning would be as below:

- The mapping correlation between shared-data ID(s) and the corresponding shared data is provisioned on all the NF instances within the same NF Set.

The procedure of NF registration procedure would be as below:

- The NF registers its NF profile to the NRF, in which the NF profile contains one or more shared-data ID(s), the shared-data ID can uniquely identify the shared data stored within any of the NF instance in the same NF set.

- In case that the NRF determines that the shared data corresponds to the shared-data ID is not available yet, it initiates the retrieval and subscription of shared data towards the register NF by providing the shared-data ID.

The procedure of NF discovery procedure would be as below:

- Upon receiving the NF discovery request from a request NF which indicates its support for SharedData feature, the NRF can respond to the request NF the target NF profile(s). In case that the target NF(s) includes NF one or more NF instance(s) within the same NF set, the mapping between shared-data ID and the corresponding shared data will be provided within the response message.

- Upon receiving the NF discovery request from a request NF which does not indicate its support for SharedData feature, the NRF shall translate the shared-data ID within the NF profile into the corresponding shared data, and respond the NF profile to the request NF.

- When the request NF receives NF profile contains shared-data ID, it initiates the subscription of shared data change event towards the current NRF instance, or alternatively, the NRF creates the subscription in an implicitly way and notifies when shared data change event occurs.

When the NF which is subscribed by the NRF on shared data change event deregisters from the NRF, the NRF can select another NF instance within the same NF set and subscribe on the shared data change event to the NF instance.

### 6.5.2 Impacts on services, entities and interfaces

NRF:

- retrieves the shared-data corresponds to the received shared-data ID from the register NF when needed.

- responds the shared data according to the supported feature of the request NF.

- subscribe and re-subscribe on shared data change event to the register NF or other NF instance within the same NF Set.

NF:

- registers its NF profile to the NRF, which may include shared-data ID, and support to handle the shared data retrieval and subscription of shared data change event initiated by the NRF.

- receives the shared-data ID and the corresponding shared data within the NF discovery procedure.

### 6.5.3 Pros

This solution introduced shared data mechanism for the NF instances within the same NF Set, to avoid the repeated transmission of the shared data related to these NF instances.

### 6.5.4 Cons

This solution is designed for NF set level, and may be inefficient for NF instances from different NF set scenario.

## 6.6 Solution #6: NRF Policy Control on Information Exposure for different PLMNs

### 6.6.1 Description

The solution is to address the Key Issue #2: Signaling overhead of NF discovery responses between PLMNs.

The NRF may be configured by means of OAM with sets of policy rules per PLMN or per groups of PLMNs (e.g. one policy rule set for all foreign PLMNs, one policy rule set for equivalent PLMNs,…). These rules define what parameters of a registered NF profile can be exposed to the PLMN. Also these rules can define the way how to modify/reduce registered profile parameters for exposure to the PLMN in order to reduce the message length and required storage capacity at the consumer.

The PLMN specific rules may take additional information into account (e.g. the NF type of the registered NF) when deciding to remove parameters from the discovery response message.

The policy rules shall be limited to cases where impacts to NF consumers are avoided, i.e. NF consumers need not be aware of the fact that NF profiles in the discovery response message have been modified/reduced.

E.g.: If the registered SmfInfo contains a list of TAIs or TAI ranges, the policy rule for all foreign PLMNs could require to remove TAIs and TAI ranges from SMF discovery responses sent to a foreign PLMN.

The PLMN-specific policy rules shall apply to NF discovery response messages and to profile change notification messages.

Figure 6.6.1-1 shows an example flow.



Figure 6.6.1-1

1-2. PLMN X -specific policy rules are configured at the NRF.

3-4. NF producer registers its profile at the NRF.

5. NRF receives a discovery request from PLMN X.

6. NRF applies the configured policy rules for PLMN X to the registered NF profile.

7. NRF sends a discovery response containing the registered NF profile with PLMN X -specific policy rules applied.

8-9. NRF receives a subscribe request from PLMN X.

10-11. NF producer modifies its NF profile

12. NRF applies the configured policy rules for PLMN X to the modified NF profile.

13-14. If the modified profile with policy rules applied differs from the original profile with policy rules applied: NRF notifies the consumer.

### 6.6.2 Impacts on services, entities and interfaces

OAM:

- Configure the NRF with policy rules per PLMN.

NRF:

- Apply configured policy rules to NF discovery response messages and profile change notification messages.

NF consumer:

- None.

### 6.6.3 Pros

This solution provides an efficient way to shorten NF Discovery response messages and Profile change notification messages sent to different PLMNs. Also the amount of Profile change notification messages will be reduced.

NF consumers are not impacted.

### 6.6.4 Cons

Depending on the complexity of OAM configured policy rules, additional NRF storage and processing resources to store and apply the rules are needed. However, as inter-PLMN discovery traffic represents a rather low contribution to the overall NF discovery traffic, the need for additional processing resources may be limited.

## 6.7 Solution#7: List of Shareable Profile Items

### 6.7.1 Description

This solution is to address the Key Issue #1: Avoid Duplicate Configuration and Transmission of Shareable Data. Specially, this solution lists those NF profile items which are possibly shared among different NF (or NF service) profiles which may belong to same or different NF set.

Table 6.7.1-1 lists NF profile items that might be shared by multiple NFs (or NF services), either across different NF sets or within same NF set.

Table 6.7.1-1: Possible shareable NF profile items

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Shareable NF profile items** | **Shared by NFs across different NF sets** | **Shared by NFs within same NF set** |
| 01 | List of supported PLMN (e.g. allowed PLMNs) | O (Optionally shared) | M (Mandatorily shared) |
| 02 | List of supported features | O | M |
| 03 | UE identifier ranges (e.g. SUPI ranges, GPSI ranges, IMSI ranges, IMEI ranges, MSISDN ranges, external/internal group identifier ranges, etc.) | O | M |
| 04 | Routing indicator, SUCI info (e.g. configured in AusfInfo); | O | M |
| 05 | List of supported serving area (e.g. TAI list or TAI range list) | O | M |
| 06 | List of supported NF services | O | M |
| 07 | List of allowed NF types | O | M |
| 08 | List of allowed NF domains | O | M |
| 09 | List of supported S-NSSAI | O | M |
| 10 | List of supported DNN | O | M |
| 11 | List of supported DNAI | O | M |
| 12 | NF/NFS FQDN (used for intra-PLMN or inter-PLMN) | N/A | O |
| 13 | Service endpoint info (HTTP scheme, port number) | O | O |
| 14 | List of locality | O | O |
| 15 | List of serving scope | O | O |
| 16 | List of SCP domain | O | O |
| 17 | List of default notification callback URL | N/A | O |

If above sharable profile items are shared by NFs within the same NF set, solution #2 provides candidate methods to configure and store the shared NF set profile without need of definitions for new data type for shareable profile items.

If above shareable profile items are shared by NFs not belonging to the same NF Set, new data types for those shareable data might be defined. For example, a variant type might be defined as a container to carry different shareable profile items, with an enumeration attribute (e.g. contentType) indicating the exact type of inner profile item.

### 6.7.2 Impacts on services, entities and interfaces

NRF, NF:

- May need to support new data types defined for shareable data.

### 6.7.3 Pros

If a common variant data type is defined for various shareable data, it is no need to define separate data types for each shareable profile items.

### 6.7.4 Cons

No special cons foreseen.

## 6.8 Solution #8: Single Subscription of NF Profile Change from an NF Set

### 6.8.1 Description

The solution is to address the Key Issue #1: Avoid Duplicate Configuration and Transmission of Shareable Data.

Multiple NFs in an NF Set may subscribe the NF profile changes of the same NF. As per existing procedure, the NRF creates individual subscription for each requesting NF. Once the NF profile of the target NF changes, the NRF sends separate notifications to each NF that previously subscribed the NF profile change notification. This mechanism can be simply implemented but it creates unnecessary notifications towards NFs within an NF Set.

However, as NFs within an NF Set may share the runtime context (e.g. cached NF profiles, subscriptions to NF profile changes, etc.) by the assistance of UDSF, it is possible for those NFs to only create one instance of subscription for the indicated purpose in the NRF.

The following enhancements can be considered for this solution:

- 1) When one NF wants to subscribe NF profile change for a specific NF, the requesting NF shall first check if such subscription context exists in the UDSF:

- 1.1) if exist, it means such subscription to the NRF has already been created by another NF in the NF Set, thus it is no need to trigger duplicate subscription to the NRF. The requesting NF may register itself to the subscription context in the UDSF so as to be notified later.

- 1.2) if not exist, the requesting NF sends subscription request to the NRF, and stores the subscription context (e.g. subscribed NF Instance ID, requesting NF ID, etc.) in the UDSF.

- 2) When detecting NF profile changes for a specific NF, the NRF initiates NF profile change notification to the NF(s) that subscribed the event, without any special enhancement to the notification procedure.

- 3) On receiving NF profile change notification from the NRF, the receiving NF shall store the updated NF profile of the indicated NF in the UDSF. Implementation specific method may be used by the UDSF to make other NFs in the NF Set be aware of the updated NF profile.

### 6.8.2 Impacts on services, entities and interfaces

NF:

- Enhancements to NF internal logic to avoid sending duplicate subscriptions to the NRF to subscribe NF profile change for a specific NF;

NRF:

- No special enhancement.

### 6.8.3 Pros

This solution doesn't require any enhancement to the NRF interface and NRF logic.

### 6.8.4 Cons

This solution relies on the NF internal enhancement and implementation of UDSF, and may not always be available. The complexity of UDSF implementation may further decrease the usability of the solution.

## 6.9 Solution #9: Single NF Profile Change Notification to an NF Set

### 6.9.1 Description

The solution is to address the Key Issue #1: Avoid Duplicate Configuration and Transmission of Shareable Data.

Multiple NFs in an NF Set may subscribe the NF profile changes of the same NF. As per existing procedure, the NRF creates individual subscription for each requesting NF. Once the NF profile of the target NF changes, the NRF sends separate notifications to each NF that previously subscribed the NF profile change notification. This mechanism can be simply implemented but it creates unnecessary notifications towards NFs within an NF Set.

However, it is possible for the NRF to aggregate separate subscriptions to NF profile change of a specific NF from multiple NFs within an NF Set, and thus only sends one instance of NF profile change notification to multiple NFs within an NF Set.

The following enhancements can be considered for the solution:

- 1) When one NF subscribes NF profile change of a specific NF to the NRF, the requesting NF shall indicate its support of sharing NF profile change notification within NF Set in the request message;

- certain methods (if available) might be used by the requesting NF to avoid sending duplicate NF profile change subscription to the NRF, e.g. as addressed by solution #8;

- 2) On receiving subscription to a specific NF profile, the NRF shall:

- 2.1) check if the requesting NF supports sharing NF profile change notification within NF Set;

- if the requesting NF does not have the expected capability, the NRF handles the subscription request as per existing procedure;

- if the requesting NF has the expected capability, continue step 2.2);

- 2.2) check if the NF profile change subscription for a specific NF Set exists in the NRF;

- if not exist, create the subscription at per NF Set level, and store the following information in the subscription context: NF Instance ID of the subscribed NF, NF Set ID, NF Instance ID of the requesting NF, callback URI of the requesting NF;

- if exist, store the following information in the existing subscription context: NF Instance ID of the requesting NF, callback URI of the requesting NF.

- 3) When NF profile change of a specific NF is detected, the NRF shall:

- 3.1) check if the stored NF profile change subscription is at NF Set level, i.e. multiple NFs in an NF Set have subscribed the NF profile change notification:

- if not, handle the NF profile change notification as per existing procedure;

- if yes, continue step 3.2);

- 3.2) handle the NF profile change notification at NF set level, i.e. select one target NF from the requesting NFs belonging to same NF Set (i.e. from the request NF list stored in the subscription context), and send NF profile change notification to that selected NF;

- 4) When one NF receives NF profile change notification from the NRF, it shall store the updated NF profile, and synchronize the received NF profile update with other NFs in the same NF Set.

### 6.9.2 Impacts on services, entities and interfaces

NF:

- indicate the support of sharing NF profile change notification within NF Set to the NRF;

NRF:

- enhance the NRF interface and logic, so as to detect the subscription from an NF with support of sharing NF profile change notification within NF Set and handle the subscription and notification accordingly.

### 6.9.3 Pros

With the enhancement, the NF profile change notifications to multiple NFs within an NF Set are reduced.

### 6.9.4 Cons

The NRF needs to differentiate the handling to a requesting NF with or with support of sharing NF profile change notification within NF Set.

## 6.10 Solution #10: Simplifying the NF service discovery response of inter-PLMN

### 6.10.1 Description

The solution is to address the Key Issue #2: Signaling Overhead of NF Discovery Responses between PLMNs.

According to the definition in 3GPP TS 29.510 [2], NRF returns all value of attributes of array or map type in NF Profile in NF discovery response, however, in most inter-PLMN scenarios, NF consumer does not need to obtain all attribute values. For example, all the list of ranges of SUPIs whose profile data is available in the UDM instance is stored in the supiRanges of the NF profile, and NF consumers in other PLMNs need only a few of them.

Consequently, in this solution, the inter-PLMN NF discovery response message will only contain value that match the NF discovery request parameters, not all value. NF consumer could process the simplified response as normal without any additional change, as the matched values fully meet its requirements in this service procedure. Furthermore, NF consumer could process the simplified response more appropriately with identifying it.

To support the solution, following enhancements shall be considered:



Figure 6.10.1.1: inter-PLMN NF service discovery procedure with partial information

1) The NF service consumer sends the NF discovery request to the NRF when query fail in the cache;

2) The intermediate NRF (NRF1) forwards the NF discovery request to the NRF in home network (NRF2);

3) On receiving NF Discovery request, the target NRF only returns the value matching the query parameters in the response, and carried an indication.

An example of simplifying response message:

The supiRanges value of the NF1 profile registered on NRF is:

{{"start": "123456789040000", "end": "123456789049999"}

{"start": "123456789050000", "end": "123456789059999"}

{"start": "123456789060000", "end": "123456789069999"}}

In the NF discovery request from NF2, the value of the query parameter SUPI is “123456789041234”, and other query parameters also match NF1. Then NF discovery response returned to NF2 contains only “{"start": "123456789040000", "end": "123456789049999"}”.

4) On receiving NF Discovery response from the NRF the consumer could cache it as normal. Provided that the consumer could handle the indication that indicates a simplified response, it could process the response more appropriately, e.g., updating values of the response to cache and not deleting other values of same NF producer already exist in cache, so as to cache more useful data and saving more signaling overhead.

### 6.10.2 Impacts on services, entities and interfaces

NRF:

- Simplify the NF discovery response message.

NF:

- Process the simplified NF discovery response message.

### 6.10.3 Pros

This solution introduces a simplification mechanism for inter-PLMN NF discovery response, to avoid the unnecessary signalling overhead and data storage overhead in inter-PLMN scenario.

### 6.10.4 Cons

The number of inter-PLMN NF discovery messages might increase a little.

## 6.11 Solution 11: Avoid large configuration of UE Routing Indicator

### 6.11.1 Description

The solution is to address the Key Issue #1: Avoid Duplicate Configuration and Transmission of Shareable Data.

Several NF types, such as UDM, AUSF, AAnF…, may contain in their NFProfile info a (potentially very large) set of Routing Indicators (RID). When a candidate NF instance of any of those types needs to be discovered by a consumer, it includes the UE RID in the NRF Discovery request, so the NRF returns only those NF instances containing such RID in their profiles.

This is a quite similar scenario to the discovery of candidate producer instances based on SUPI, and the problem of provisioning a large set of SUPI ranges in the NF Profile of the NF instances themselves. For this problem, the approach of configuring the mapping between SUPI and NF Group ID was addressed by moving the configuration of SUPI ranges and NF Group IDs to the NRF and letting the NRF to obtain the NF Group ID for a given SUPI received in the discovery request and include the NF instances of the corresponding NF Group ID in the discovery response.

If the NRF has no stored configuration mapping identity sets/ranges to NF Group ID locally, the NRF can retrieve the NF Group ID corresponding to a specific subscriber identifier from the UDR using the Nudr\_GroupIDmap\_Query service operation.

A similar approach can be proposed for the RIDs: the configuration of mapping between RID and NF Group ID can be moved out from the producer NFProfile (and NRF), and can be done at the NRF either based on local configuration or by interacting with UDR. It should be noted that all NF instances deployed in the same NF Group will define support for the same list of Routing Indicators and multiple Routing Indicators can be supported within the same NF Group.

### 6.11.2 Impacts on services, entities and interfaces

NRF:

- add logic to determine the NF Group ID for a given RID, either based on local configuration or by interacting with UDR.

UDR:

- enhance the Nudr\_GroupIDmap API to support receiving the RID as input for the mapping.

### 6.11.3 Pros

This solution provides a very simple enhancements to relieve the configuration of large sets of RID in NF producers' NFProfile and lowering the size of these profiles.

### 6.11.4 Cons

None identified.

## 6.12 Solution #12: Support for NF Set level shared data

### 6.12.1 Description

The solution is to address the Key Issue #1: Avoid Duplicate Configuration and Transmission of Shareable Data.

According to the definition in TS 23.501[3], NF Set is a group of interchangeable NF instances of the same type, supporting the same services and the same network slice(s), and can have access to the same context data even geographically distributed. Under this precondition, it can be assumed that NF profiles of different NF instances within the same NF Set tend to converge, and would only differ with some NF instance specific data types, e.g. address info, NF Instance info, etc.

To optimize the storage and transmission of shared data for an NF Set, the following issues needs to be addressed in this solution:

a) How the NRF receives the shared data for a specific NF Set.

b) How the NRF can be notified of the updated shared data for the NF Set.

c) How the NF discovery procedure can be optimized to avoid the transmission of multiple copies of shared data for the NF Set.

Evaluation of Issue a) and issue b) heavily depends on the endpoint in which the shared data is provisioned, e.g. the NF instances, or the NRF. If the shared data for the NF Set is provisioned only in the NF instance, the mapping information between NF Set ID and corresponding shared data in the NRF needs to be maintained by the NF instance(s), thus implicit and explicit subscription may be involved as specified in Option A of Solution #2 and Solution #5 respectively, which may cause complicated interaction between NRF and NF instance to guarantee the consistency of the shared data for the NF Set.

In consideration of the fact that the NF instances of the same NF Set is usually served by a specific NRF instance or specific pair of NRF instances, it would be much easier to enable the direct provisioning and updating of shared data for the NF Set by the OAM. In this case, transmission of shared data for the NF Set is not needed during the NF registration procedure.

For issue c), either one step or two step retrieval can be adopted to avoid the transmission of shared data for the NF Set. As the NF discovery is triggered to discovery the target NF instance or target set of NF instance(s), it means the NF Consumer have no NF profile context for the NF Producer within the NF Set, not to mention the shared data for the NF Set, one step retrieval is proposed to be adopted to avoid additional interaction.

Similar with the updating on URI structure of the NFManagement API in Solution #2, resource nf-sets is proposed to be managed independent of the NF profile within the resource nf-instances. As the resource nf-sets is provisioned and updated by the OAM directly, only HTTP GET method triggered by NF Consumer or other NRF instance is needed for this resource.



Figure 6.12.1.2-1 Resource Structure for Nnrf\_NFManagement service

The procedure of shared data for NF Set provisioning and updating would be as below:

- The mapping correlation between NF Set ID and the corresponding shared data for the NF Set is provisioned and updated by the OAM, based on the serving relationship between NRF instance(s) and NF Set(s), which is outside the 3GPP scope.

NOTE 1: OAM is responsible for the provisioning of shared data for the NF Set into the corresponding NRF instance(s) which serves the NF Set.

The procedure of NF registration procedure would be as below:

- The NF registers its NF profile to the NRF, which contains the NF Set ID it belongs to. As the shared data for the NF Set is already provisioned in the NRF, no shared data is to be contained within the registered NF profile.

- Upon receiving the NF registration request, the NRF correlates the registered NF instance with the stored nf-sets by using NF Set ID.

The procedure of NF discovery procedure would be as below:

- Upon receiving the NF discovery request from a requester NF which indicates its support for NFSetSharedData feature, the NRF responds to the requester NF the registered NF profile(s) of the target NF instance(s), together with the shared data for the NF Set(s) corresponding to the target NF instance(s).

- Upon receiving the NF discovery request from a requester NF which does not indicate its support for NFSetSharedData feature, the NRF merges the registered NF profile with the shared data of the NF Set for each target NF instance, and respond the merged NF profile(s) to the requester NF.

- When the requester NF receives the additional shared data for the NF Set(s), it correlates the shared data for the NF Set(s) with the received NF profile(s), and regards the shared data for the NF Set applicable for every NF instance of the NF Set when interacting with the NF Producer. It initiates the subscription of change event for the shared data of NF Set(s) and NF profile(s) towards the NRF respectively if not triggered yet.

NOTE 2: OAM is responsible for avoiding duplicate provisioning for a specific attribute in both NF profile registered by the NF Producer and the shared data for the NF Set.

### 6.12.2 Impacts on services, entities and interfaces

NRF:

- Correlates the registered NF profile and stored shared data for the corresponding NF Set, and responds the NF profile and shared data for the NF set to the requester NF in a proper way, based on the supported feature of the requester NF.

NF Producer:

- registers its NF profile to the NRF, which includes NF Set ID it belongs to, while shared data for the NF set is not included.

NF Consumer:

- receives the NF profile for the target NF instance(s) and the corresponding shared data for the NF Set, regards the shared data for the NF Set applicable for each NF instance within the NF Set.

### 6.12.3 Pros

No complicated explicit and implicit coordination for shared data of the NF set is needed between NF Producer and the NRF.

### 6.12.4 Cons

This solution is designed for NF Set level, and may be inefficient for NF instances from different NF Set scenario.

# 7 Evaluations

## 7.1 Evaluation of Solutions for Key Issue#1

### 7.1.1 Synchronization of Shareable Data

#### 7.1.1.1 Configured Shareable Data

NF configuration data (configured by means of OAM at the NF/NF set) have an impact on the NF's profile data stored at the NRF. Profile data stored at the NRF are either statically configured by means of OAM (in sync with the data configured at the NF/NF set), or dynamically registered by the NF (avoiding synchronization issues).

Similarly for shared configuration data and their IDs, which are configured by means of OAM at all sharing NFs/NF sets: The corresponding shared data / set profile and IDs needs to be stored at the NRF.

Solution #1 allows data sharing among NFs of different types or of same type from different sets. This adds the need for synchronized configuration. In addition, solution #1 proposes to configure shared data and their IDs at all NRFs, adding even more complexity to ensure overall synchronized configuration.

The synchronization issue between NRFs is addressed by solution #4 which proposes a higher level NRF to store the shared data, at the cost of higher NRF to NRF traffic, to retrieve the shared data when needed.

Similarly, solution #1 proposes as an option to sync NRFs by means of an operator granted NF (e.g. special UDR).

Still synchronization between NFs, between NF sets and between NF and NRF remains an issue adding complexity to network configuration.

Solution #2 (option A) and solution #5 propose to configure/provision shareable data (NF set profiles) and their IDs (NF set ID) at a single place (the NF set), avoiding all kinds of synchronization issues.

Solution #2 (option B) proposes to configure/provision shareable data (NF set profiles) and their IDs (NF set IDs) at two places (a single NRF set and a single NF set) allowing to minimize synchronization issues.

Solution #8 and #9 proposes similar way to avoid duplicate subscription and change notification, via the UDSF and NRF determination. However, the basis of the solution might be inaccurate, as the NF will only determine to trigger the subscription when no shared data can be found locally, thus no duplication subscription will be received by the NRF, and the NRF does not need to avoid duplicate notification to the NFs within the same NF set.

Solution #11 proposes to delegate the configuration of a certain type of shareable data (Routing Indicators) to the NRF or UDR and let the NRF find out the mapping between RID and NF Group ID based on local configuration or via an existing API with the UDR. This approach is quite similar to the existing mechanism for handling SUPI ranges, and in that sense, it does not bring any additional issue related to configuration, discovery or change of such shareable data, compared to the existing mechanism already in place for mapping between SUPI and NF Group ID.

Solution #12 proposes to configure the shared data in NRF(s) which serves the NF Set, thus no synchronization on update of shared data is required between NF(s) and NRF(s).

#### 7.1.1.2 Discovered Shareable Data

Once shareable data have been discovered by NFs during the NF discovery procedure, the data are subject to change. Solutions #1, #2 and #5 propose similar subscribe/notify mechanisms ensuring that discovered shared data are kept in sync.

#### 7.1.1.3 Change of Shareable Data

When the shareable data is configured in NF instance or NRF, the operator may update the correlation between shared-data ID with shareable data. The NRF should be aware of the change event.

Editor's note: The solution evaluation for Change of Shareable Data is FFS.

### 7.1.2 Storage requirements

#### 7.1.2.1 Storage of configured data

Storing/configuring multiple copies of the same configured data (i.e. redundant data) at an NF is an issue addressed by solutions #1, #2, #4 and #5. However, storing requirements for a single copy of shared data differ. It is beneficial to store/configure shared data at a single place in the network.

Solution #1 proposes to configure/store shared data at all or multiple NRFs /NRF set in the PLMN.

Solution #4 proposes to configure/store shared data at a single higher level NRF /NRF set.

Solution #2 option A proposes to configure/store shared data (NF set profiles) at a single NF set.

Solution #2 option B proposes to configure/store shared data (NF set profiles) at a single NRF/NRF set.

Solution #5 proposes to configure/store shared data at a single NF set.

Solution #8 and #9 proposes to configure/store shared data at a single NF set.

Solution #11 proposes to configure Routing Indicator to NF Group ID mapping at the NRF or in the UDR, rather than at the NFProfile of the producer (UDM, AUSF, AAnF).

Solution #12 proposes to configure shared data within the NRF(s) serving the NF set.

#### 7.1.2.2 Storage of discovered data

Storing multiple copies of the same discovered data (i.e. redundant data) at an NF is an issue addressed by solutions #1, #2, #5, #8, #9 and #12. All solutions propose the same principle, i.e. store the discovered shared data only once and use the IDs as a pointer to the shared data.

Similar solution is also proposed in solution #4, by identify the discovered shared data using shared-data IDs together with the corresponding higher level NRF instance ID.

### 7.1.3 Discovery procedure

Solutions #1, #2, #4 and #5 propose optimizations, differing with regard to discovery response message length and number of message roundtrips.

Solution #1, alternative 2 of solution #2, solution #4 and solution #12 propose an additional roundtrip to retrieve shared data from the NRF if a shared data ID (or set profile ID) was received while the corresponding shared data (or set profile) has not yet been received. This additional message roundtrip occurs only when the shared data is not cached.

Solutions #5 and alternative 1 of solution #2 propose to always include (a single copy of) the shared data in the discovery response message, so to avoid an (potentially seldom occurring) additional roundtrip at the cost of (an always) increased message length.

With solution #3 the response message length is reduced at the cost of more roundtrips.

Solutions #8 and #9 propose optimizations for NF profile change subscription and notification that simplify the discovery of shared data.

Solution #11 does not imply any change on the discovery procedure by consumers of the NRF Discovery service.

### 7.1.4 Which data should be shareable?

Standardizing the exact list of data that may be shareable comes with the drawback that for any extension in future releases the decision (and specification) needs to be taken whether or not the extended data is worth being shareable.

Solution #1 currently has an editor's note indication that this aspect is ffs.

Solution #5 has the same issue as #1 (although not explicitly stated).

Solution #2 avoids the need to address the issue as the NF set profile would be of the same type as the NF profile.

Solution #7 listed the data which can be shared in the NF profile items.

Solution #11 is restricted to the Routing Indicator data set, as shareable data.

### 7.1.5 New NF services operations

It is beneficial to extend existing service operations rather than defining new service operations.

Solution #5 proposes that the NRF retrieves (and subscribes to) shared data from the NF during the NF registration procedure. The registering NF can be of any type. Consequently, any NF of any type must offer a new service allowing the NRF to retrieve and subscribe to shared data.

Solution #2 option A avoids new service operations by re-using and extending the Nnrf\_NFManagement service operations (NFRegister, NFUpdate, NFDeregister) allowing any NF to push (register) its set profile to the NRF, modify it at the NRF and delete it from the NRF. While it is not mentioned how the NRF can determine whether the NF Set profile is not contained due to this feature or some error cases, as no shared-data-ID is provided to the NRF in the NFRegister request.

Solution #3 avoids new service operations by re-using and extending the Nnrf\_NFDiscovery service operation (NFDiscover) allowing pagination of discovery responses.

Solution #11 does not require any modification on the interaction between the producer and the NRF.

### 7.1.6 Applicability Scope

Solutions #2, #12 and #5 restrict sharing of data to NF sets and does not address sharing of data that may be common to NFs (of same or different type) from different sets.

Solution #3 is restricted to NF instance level.

Solution #1 has no restriction, i.e. it is applicable on PLMN level.

Solution #4 can be applicable on PLMN level or level NRF instance level.

Solution #11 is restricted to the Routing Indicator data set, as shareable data, and therefore it is restricted to the producer NF types that make use of such data.

### 7.1.7 Backward compatibility

All solutions except Solution #11 require feature support indications to ensure backward compatibility with no-supporting legacy NFs.

Solution #11 does not require any modification on the interaction between the producer and the NRF, and it simply implies that the producer will no longer include the list of supported Routing Indicators in its NFProfile.

### 7.1.8 Comparison

Detailed comparison of the solutions can be listed as below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Solution #1 | Solution #2 | Solution #3 | Solution #4 | Solution #5 | Solution #8 | Solution #9 | Solution #12 |
| Synchronization of Shareable Data | Synchronized configuration in all NFs and all NRFs is required | OptionA: no synchronized configuration between NFs and NRFs required  Option B: synchronized configuration between NF sets and NRF(set) required | N/A | Synchronized configuration in all NFs and higher level NRF is required | no synchronized configuration between NFs and NRFs required | Avoid duplicate subscription and change notification via UDSF. | Avoid duplicate subscription and change notification via the NRF. | no synchronized configuration between NFs and NRFs required |
| Support for change of shareable data | FFS | FFS | FFS | FFS | FFS | FFS | FFS | FFS |
| Storage requirements | Shared data are stored/ configured in all/multiple NFs and NRFs | Option A: Shared set profile is stored/configured at the NF set only  Option B: Shared set profile is stored/configured at the NF set and at singleNRF | N/A | Shared data are stored/ configured in all/multiple NFs and in a higher level NRF | Shared data are stored/configured at the NF set only | Shared data are stored/configured at a single NF set | Shared data are stored/configured at a single NF set | Shared data are stored/configured at the NF set and the NRF(s) serving the NF Set |
| Discovery procedure | One additional message roundtrip only when shared data are not cached | Alternative 1: No additional Roundtrip at the cost of higher message size  Alternative 2: One additional message roundtrip only when shared data are not cached | Response message length is reduced at the cost of more message roundtrips | One additional message roundtrip only when shared data are not cached | No additional Roundtrip at the cost of higher message size | No additional Roundtrip at the cost of higher message size | No additional Roundtrip at the cost of higher message size | No additional Roundtrip at the cost of higher message size |
| Which data should be shareable? | ffs | No need to standardize | N/A | N/A | ffs | N/A | N/A | ffs |
| New NF service operations | Not required | Not required | Not required | Not required | All NFs must provide a shared data retrieval service operation to be consumed by the NRF | Not required | Not required | Not required |
| Applicability scope | PLMN level | NF Set level | NF Instance Level | NRF level or PLMN level | NF Set level | NF Set level | NF Set level | NF set level |
| Backward Compatibility | By including Shared Data support indication | By including Set profile Support Indication | By indicating support of paginated data retrieval feature | By indicating support for sharedData feature | By indicating support for sharedData feature | Not required | By indicating support of sharing NF profile change notification | By indicating support for sharedData feature |

## 7.2 Evaluation of Solutions for Key Issue#2

Solution for Key issue#2 aim to address the signalling overhead of NF Discovery Responses between PLMNs.

Solution #10 reduce the signalling overhead of NF discovery response by only containing value that match the NF discovery request parameters in the response. Solution #10 might increase the number of NF discovery message, however, the overall overhead of NF discovery procedure between PLMNs will be optimized.

Solution #6 would shorten NF discovery response messages by removing certain parameters from the message, and operators need to determine which parameters can be removed without affecting services.

Solution #10 reduce values of parameters of array or map type in NF discovery response, while solution #6 reduce unnecessary parameters. And they could be applicable to different scenarios.

Solution #10 could apply on most parameters of array or map type, while solution #6 might be applicable to a few scenarios and hard to define the rules for reducing which kind of parameters.

# 8 Conclusions

## 8.1 Conclusion of Solutions for Key Issue#1

The following principles are concluded for key issue #1:

1. For backwards compatibility, the selected solution shall make use of the supported feature negotiation mechanism.

2. For forward compatibility, the selected solution shall try to avoid the need to standardize, which data are shareable.

3. Whether new NF services are needed will be decided during normative work.

4. The selected solution shall allow to take advantage of the NF set concept, by defining NF Set profiles. NF set profiles may contain one or more shared data IDs pointing to PLMN-level shared data.

5. The selected solution shall allow to use shared-data concept across NFs not belonging to an NF-Set.

6. The selected solution should avoid the need for shared data configuration at multiple NFs within the network. The solutions of both OAM based and OAM-less solution will be further worked on during normative phase of work.

7. The NRF should optionally allow other NFs to register (and modify and delete) those NfSetProfiles/Shared Data at the NRF the other NFs may make use of. The details are ffs during normative work.

8. The NRF shall allow other NFs to retrieve (and subscribe to) NfSetProfiles/Shared Data

9. The selected solution shall ensure that a change of NfSetProfile/Shared Data is propagated between NF set and NRF set involving only one NF instance from the NF set and only one NRF instance from the NRF set.

10. The selected solution shall allow to avoid the need of including large lists of UE Routing Indicators in NF profiles registered at the NRF, e.g. by NRF configuration or by interaction with the UDR.

## 8.2 Conclusion of Solutions for Key Issue#2

Based on the evaluation of solutions for key issue#2, it is recommended to adapt solution #10 as basis for normative work. With solution#10, NF discovery response messages could be greatly simplified in most scenarios of inter PLMN by reducing values of parameters of array or map type in the messages, while these parameters cannot not be removed from the messages.

The new indication added to NF discovery response is optional, and NF consumer could either ignore it without any impact, or do extension processing with it. It is FFS what the impact of the new indication in NF discovery response on NF consumer, and whether it is necessary.

Annex A (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2022-08 | CT4#111e | C4-224138 |  |  |  | Skeleton | 0.1.0 |
| 2022-08 | CT4#111e | C4-224206 |  |  |  | Pseudo-CR on Scope | 0.1.0 |
| 2022-08 | CT4#111e | C4-224454 |  |  |  | Key Issue for Avoiding Duplicate Configuration and Transmission of Shareable Data | 0.1.0 |
| 2022-08 | CT4#111e | C4-224455 |  |  |  | Solution for Avoiding Duplicate Configuration and Transmission of Shareable Data | 0.1.0 |
| 2022-11 | CT4#113 | C4-225327 |  |  |  | New KI on Signaling overhead of NF discovery responses between different PLMNs | 0.2.0 |
| 2022-11 | CT4#113 | C4-225517 |  |  |  | Solution for Support of NF (Service) Set Profiles | 0.2.0 |
| 2022-11 | CT4#113 | C4-225520 |  |  |  | Updates to Solution #1 | 0.2.0 |
| 2022-11 | CT4#113 | C4-225521 |  |  |  | Solution for Downloading Large Volume Data within NF Profile | 0.2.0 |
| 2022-11 | CT4#113 | C4-225549 |  |  |  | Update of Key Issue #1 on Support of NF (Service) Set Profiles | 0.2.0 |
| 2022-11 | CT4#113 | C4-225557 |  |  |  | Updates on Key Issue #1 for FS\_NRFe | 0.2.0 |
| 2022-11 | CT4#113 | C4-225558 |  |  |  | New Solution for Key Issue #1 of FS\_NRFe with enhanced inter NRF interaction | 0.2.0 |
| 2022-11 | CT4#113 | C4-225560 |  |  |  | New Solution for Key Issue #1 of FS\_NRFe to support set level shared data | 0.2.0 |
| 2022-11 | CT4#113 | C4-225561 |  |  |  | Updates on Baseline | 0.2.0 |
| 2022-11 | CT4#113 | C4-225573 |  |  |  | Updates to Key Issue #1 | 0.2.0 |
| 2023-03 | CT4#114 | C4-230669 |  |  |  | NRF Policy Control on Information Exposure for different PLMNs | 0.3.0 |
| 2023-03 | CT4#114 | C4-230656 |  |  |  | Evaluation of Solutions | 0.3.0 |
| 2023-03 | CT4#114 | C4-230518 |  |  |  | List of Shareable Data Types | 0.3.0 |
| 2023-03 | CT4#114 | C4-230521 |  |  |  | Solution for NF Profile Change Notification to NFs in an NF Set | 0.3.0 |
| 2023-03 | CT4#114 | C4-230657 |  |  |  | New Solution on KI#2 of simplifying the NF service discovery response of inter-PLMN | 0.3.0 |
| 2023-04 | CT4#115e | C4-231098 |  |  |  | Update of Solution #6 | 0.4.0 |
| 2023-04 | CT4#115e | C4-231365 |  |  |  | Solution Evaluation for Key Issue #1 | 0.4.0 |
| 2023-05 | CT4#116 | C4-232264 |  |  |  | Solution Update for Solution #5 of FS\_NRFe | 0.5.0 |
| 2023-05 | CT4#116 | C4-232536 |  |  |  | Solution Update for Solution #10 of FS\_NRFe | 0.5.0 |
| 2023-05 | CT4#116 | C4-232634 |  |  |  | Evaluation for Key Issue #1 of FS\_NRFe | 0.5.0 |
| 2023-05 | CT4#116 | C4-232635 |  |  |  | Solution Evaluation for Key Issue #1 of FS\_NRFe | 0.5.0 |
| 2023-06 | CT#100 | CP-231023 |  |  |  | Presented for information | 1.0.0 |
| 2023-09 | CT4#117 | C4-233046 |  |  |  | Clean Up | 1.1.0 |
| 2023-09 | CT4#117 | C4-233122 |  |  |  | Completing the Comparison Table for Key Issue #1 | 1.1.0 |
| 2023-09 | CT4#117 | C4-233272 |  |  |  | Storage of RID ranges in UDR | 1.1.0 |
| 2023-09 | CT4#117 | C4-233374 |  |  |  | New Solution for Key Issue #1 of FS\_NRFe to support set level shared data | 1.1.0 |
| 2023-09 | CT4#117 | C4-233778 |  |  |  | Conclusion for Key Issue #2 of FS\_NRFe | 1.1.0 |
| 2023-09 | CT4#117 | C4-233847 |  |  |  | Solution Evaluation for Key Issue #1 of FS\_NRFe | 1.1.0 |
| 2023-09 | CT4#117 | C4-233878 |  |  |  | Solution Evaluation for Key Issue #2 of FS\_NRFe | 1.1.0 |
| 2023-09 | CT4#117 | C4-233886 |  |  |  | Conclusion for KI#1 | 1.1.0 |
| 2023-09 | CT#101 | CP-232023 |  |  |  | TR Presented for approval | 2.0.0 |
| 2023-09 | CT#101 |  |  |  |  | TR Approved | 18.0.0 |