

3GPP TR 23.845 V10.0.0 (2011-03)

Technical Report

3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Study on User Data Convergence (UDC) evolution (Release 10)



Keywords

GSM, UMTS, network, IP, Multimedia MAP, IN

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.

© 2011, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, 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 currently being 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, symbols and abbreviations	7
3.1 Definitions	7
3.2 Symbols.....	7
3.3 Abbreviations.....	7
4 UDC evolution topics.....	7
5 Multiple UDRs in a network	8
5.1 Description.....	8
5.1.1 Introduction.....	8
5.1.2 Multiple UDRs for very large networks.....	8
5.1.3 Multiple UDRs when no common data model.....	8
5.1.4 Multiple UDRs when many applications	9
5.2 Alternative solutions.....	9
5.2.1 Routing Solutions for very large networks	9
5.2.1.1 Introduction.....	9
5.2.1.2 Solution 1	9
5.2.1.3 Solution 2	10
5.2.1.4 Solution 3	10
5.3 Comparison of solutions and conclusions.....	10
5.3.1 Multiple UDRs for very large networks.....	10
5.3.2 Multiple UDRs when many applications	10
6 Bulk data operations.....	11
6.1 Description.....	11
6.2 Conclusion and recommendation.....	11
7 Handling of common data.....	11
7.1 Description and consequences of the common data concept.....	11
7.2 Alternative Solutions	12
7.2.1 Solutions for FEs that are not provisioning FEs	12
7.2.1.1 FEs are not aware of the common data concept.....	12
7.2.1.2 FEs are aware of the common data concept.....	13
7.2.2 Solutions for provisioning FEs	16
7.3 Solutions comparison.....	16
7.3.1 Comparison for non provisioning FEs aware of the common data concept	16
7.3.2 Comparison between non provisioned FEs aware or not of the common data concept.....	16
7.3.3 Comparison for provisioning FEs	17
7.4 Conclusions and recommendations	17
8 FE to FE communication	17
8.1 Description.....	17
8.1.1 Introduction.....	17
8.1.2 Request Relaying.....	17
8.1.3 Authentication Vector Request Forwarding	19
8.1.4 Replacing Indirect FE to FE Communication with direct FE to FE Communication	20
8.1.5 Summary.....	21
8.1.6 Standardisation considerations.....	21
8.1.7 Conclusion.....	22

9 Collision detection control enhancements22

10 Enhancements to subscription to notification.....22

11 Notifications and transactions22

11.1 Description..... 22

11.2 Alternative solutions..... 23

11.3 Solutions comparison..... 23

11.4 Conclusions and recommendations 23

Annex A (informative): Change history.....25

Foreword

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

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

Version x.y.z

where:

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

1 Scope

The present document is addressing a number of topics regarding User Data Convergence evolution. Some of these topics were identified in the Re19 work but have been delayed, others are new.

These topics are largely independent of each other and are studied separately in this document.

They will be normally addressed through the following steps:

- a description of the topic to be addressed with its interest, the associated requirements to cover or issues to be solved
- a description of the alternative solutions including their impact on 3GPP specifications
- a comparison of the solutions
- a conclusion with recommendations.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".
- [2] 3GPP TS 23.335: "User Data Convergence (UDC); Technical realization and information flows; Stage 2".
- [3] 3GPP TS 32.181: "User Data Convergence (UDC); Framework for Model Handling and Management".
- [4] IETF RFC 4511: "Lightweight Directory Access Protocol (LDAP): The Protocol".
- [5] 3GPP TS 29.335: "User Data Convergence (UDC); User Data Repository Access Protocol over the Ud interface; Stage 3".
- [6] 3GPP TR 29.935: "Study on UDC Data Model".
- [7] 3GPP TS 29.229: "Cx and Dx interfaces based on the Diameter protocol; Protocol details".

3 Definitions, symbols and abbreviations

3.1 Definitions

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

common data: sets of permanent data where the values are common to a large number of users.

3.2 Symbols

For the purposes of the present document, no symbols apply.

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].

ANDSF	Access Network Discovery and Selection Function
CDM	Consolidated Data Model
FE	Front-End
LDAP	Lightweight Directory Access Protocol
MMTEL	Multimedia Telephony
OSS	Operations Support System
RDM	Reference Data Model
SOAP	Simple Object Access Protocol
SLF	Server Locator Function
TAS	Telephony Application Server
UDC	User Data Convergence
UDR	User Data Repository

4 UDC evolution topics

The study on UDC evolution addresses the following topics:

- Multiple UDRs in a network
- Bulk data operations
- Handling of common data
- FE to FE communication
- Collision detection control enhancements
- Enhancements to subscription to notification
- Notifications and transactions

5 Multiple UDRs in a network

5.1 Description

5.1.1 Introduction

In 3GPP TS 23.335[2], it is written: "In the architecture, the User Data Repository (UDR) is a functional entity that acts as a single logical repository of user data and is unique from Application Front End's perspective". This statement may be interpreted as there is no more than one UDR in a given PLMN.

The present TR topic tries to address the assumption of multiple UDRs in a PLMN, to identify consequences and the possible impacts on existing UDC specifications.

From a practical point of view, even if the aim is to have one single logical repository, a certain number of considerations may drive to have more than one UDR in a PLMN.

It should be assessed if such considerations are of interest for operators and if so, what would be the possible impacts on standardization.

5.1.2 Multiple UDRs for very large networks

For very large networks with a very large amount of users, although an UDR may be implemented in a distributed architecture and multiple database servers with geographical distribution and geographical redundancy, an operator may consider to deploy several UDRs between which it will distribute the users.

Various reasons may drive an operator to deploy multiple UDRs, such as distribution of users in different administrative areas, progressive deployment of the UDC architecture starting with separate UDRs, UDRs from multiple vendors, scalability considerations, introduction of Ud reference points considered as a good way to go forward UDC without applying the complete UDC architecture. So it appears relevant to not only consider the target UDC architecture with one unique UDR, but to analyse how the UDC concept may also apply when multiple UDRs.

It is assumed that the user data of a given user is stored on only one UDR.

With regard to application FEs, we may distinguish two cases:

- 1) for clusters of HLR or HSS FEs (or application FEs behaving the same way) that are linked to only one UDR, with multiple UDRs, there would be several clusters of HLR or HSS FEs, one cluster being linked to only one UDR. It is to the interfaces between the FEs and the other core network entities to ensure the right routing of requests for a given user to the right FE cluster. Such routing is ensured by MAP or Diameter (e.g. Diameter proxies).
- 2) for some other application FEs, it may be somewhat different. In the example of the ANDSF covered by 3GPP TS 23.335 [2], a given ANDSF server may be contacted for any user of the network (to be checked), in that case according to the user, the ANDSF server should send a Ud request to the right UDR. Two sub-cases may be considered about the ANDSF-FEs:
 - the ANDSF supports several functional ANDSF-FEs each being connected by a Ud interface (e.g. a LDAP TCP-IP connection) to a given UDR. Then, the ANDSF has to find how to select the right Ud interface.
 - the FE concept is extended (extended FE) to support several Ud interfaces towards different UDRs. There is the same routing question on how to find the right UDR.

NOTE: In 3GPP TR 29.935 [6], it is studied to standardise the MMTEL data in the Reference Data Model to be used over Ud, meaning that a telephony application server (TAS) would also be considered as an example of AS that may support the Ud interface. It would be an additional example for the above case 2).

5.1.3 Multiple UDRs when no common data model

For full multivendor interoperability between FEs and UDR either a standardized Reference Data Model (common to all FEs and UDR) is required (see 3GPP TR 29.935 [6]), or the UDR needs to support multiple proprietary data models

(all data models that are used by the different FEs). As long as these options are not available, networks that deploy FEs from different vendors (using different vendor specific proprietary data models) may want to deploy multiple UDRs, one from each of the vendors. Vendor x FEs are connected to the vendor x UDR, vendor y FEs are connected to the vendor y UDR. The resulting architecture is the same as the one described in 5.1.2, but it is justified due to missing of a common data model rather than due to the very large amount of users.

5.1.4 Multiple UDRs when many applications

In this case, where there are many different applications each with their user data, the UDC logic would be to group all these user data into only one logical repository (UDR). An operator may want to avoid to group all these user data in a unique database, but nevertheless to use the UDC concept and to have one UDR grouping the user data of a set of applications and another one grouping user data of another set of applications etc.

Various reasons may drive an operator to deploy multiple UDRs, an UDR addressing a certain set of user data for a certain set of applications. Currently each application is storing its user data in a dedicated database (the "silo" view); the integration of all these user data into one UDR is a target and will probably be progressive. Intermediate steps may appear where there may still be separate logical repositories, each with its own Ud accesses, so appearing as multiple functional UDRs. Some pragmatism may be observed in the transition towards the UDC architecture. So it appears relevant to not only consider the target UDC architecture with one unique UDR storing all user data, but to analyse how the UDC architecture with Ud reference points may also apply when multiple UDRs, each storing the user data for a certain set of applications, are introduced.

In principle, for a given application FE, it would only see the UDR supporting its user data, so it complies to 3GPP TS 23.335[2] statement that "UDR is unique from Application Front End's perspective".

What can appear is that a given application has its own user data stored in a UDR and may need to access user data associated to another application (eg some HSS user data).

In this context, should such an application present two application FEs, one with a Ud interface to the first UDR, the other connected to the other UDR? The choice to use one of the functional FEs is based on the requested data, so it should not be an issue.

On the UDR side, the same user will have user data in one UDR attached to a first set of applications and other user data in other UDR(s) for other set(s) of applications. It clearly has an impact on the provisioning side as two or more UDRs may have to be provisioned for the same user.

The other point is about data that would be common to applications in the first UDR and applications in another UDR. This situation should be avoided, as it implies a synchronized management of this data.

5.2 Alternative solutions

5.2.1 Routing Solutions for very large networks

5.2.1.1 Introduction

It is here analysed the possible solutions to address the routing issues identified in sub-clause 5.1.2 for the case 2).

3 solutions are identified.

5.2.1.2 Solution 1

In this solution, a network element having to access multiple UDRs through the Ud interface have a local way to define to which UDR belongs a user (e.g. through ranges numbering).

This solution can apply if this network element supports several application FEs each connected to a UDR, or with an extended FE supporting several Ud interfaces.

This solution requires to configure all network elements in the network acting according to the case 2). It is a solution less powerful than the one classically achieved with MAP or Diameter routing.

5.2.1.3 Solution 2

A second solution is based on a central server (a bit similar to a SLF) that is able to return the UDR identity that is storing user data of a given user to the requesting network element. Then the requesting network element will use the relevant Ud interface towards the selected UDR.

The Ud interfaces towards the different UDRs may be already established and permanently maintained, so to optimise the performances.

This solution introduces a new functional entity in the UDC architecture.

A candidate interface to this central server could be a Ud interface.

5.2.1.4 Solution 3

This solution would be based on a proxy approach (somewhat like with a Diameter proxy). Here the network element through its Application FE always uses the same Ud interface towards an entity that will support a proxy function able to route the requests towards the right UDR.

This entity having this proxy function will appear as a UDR ("a functional entity that acts as a single logical repository" as defined in 3GPP TS 23.335 [2]) for the Application FE. Then 2 approaches are identified:

- the multiple UDRs of the 5.1.2 sub-clause are grouped into a "super UDR" that itself complies to the functional content of a UDR. And as UDC architecture does not address the internal functional structure of a UDR for which many possibilities may exist, this aggregation of UDRs into one UDR is out of the scope of the UDC architecture.
- UDC architecture evolves to consider such several UDRs driving to specify the proxy function and the interface between the proxy function and the various UDRs, keeping in mind that for the application FE, the Ud interface should keep its functional content. The introduction of a proxy may also alter the access performances to user data.

5.3 Comparison of solutions and conclusions

5.3.1 Multiple UDRs for very large networks

The hereafter analysis also applies to the case of multiple UDRs with no common data model.

In the subclause 5.1.2, it was indicated the option to extend the FE concept to support several Ud interfaces towards different UDRs. No strong arguments have been delivered for this new FE concept compared to several FEs, each handling a Ud interface towards a given UDR. The solution with several FEs is compliant with the existing definition of a FE and does not require additional standardization, whereas the concept of a FE accessing multiple UDRs requires additional standardization.

So it is currently recommended to build solutions with multiple UDRs for very large networks on the basis of the current FE definition in 3GPP TS 23.335 [2].

Regarding routing and the 3 solutions described in subclause 5.2.1, solution 1 presents a disadvantage as it requires to populate (and update over time) the routing information to find the right UDR according to the fetched user in each network element accessing multiple UDRs. So solution 2 or 3 are preferred. Further investigation on the comparison of solutions 2 and 3 should be made.

Both solution 2 and 3 may rely on the standardized Ud interface, so not strictly requiring additional standardization.

5.3.2 Multiple UDRs when many applications

Although it is not recommended to build solutions where user data of a single user are spread over multiple UDRs since this contravenes the general concept of user data convergence, intermediate steps may appear where there are separate logical repositories, each with its own Ud accesses and each storing the user data for a different set of applications, so appearing as multiple functional UDRs. For such cases, it is possible to build solutions with multiple UDRs, when many

applications, on the basis of the current FE definition in 3GPP TS 23.335 [2], so avoiding to introduce the concept of an extended FE connected to several UDRs.

6 Bulk data operations

6.1 Description

The main use case for bulk data operations is the provisioning from the OSS. 3GPP TS 32.181[3] states: "The Provisioning Gateway provides a single logical point for consistent provisioning of user data for all services in the UDR. The mechanism used by the Provisioning Gateway to access UDR data using the CDM is outside the scope of the present document". So Ud is not retained as an interface to handle provisioning from the OSS. 3GPP TS 32.61x series specifies bulk operations in OAM environment. So, it should be to SA5 to assess the use of bulk operations with OSS when related to UDR.

In 3GPP TS 23.335 [2], provisioning FEs are described in sub-clause 4.2.2 and address the following use cases:

- Provisioning from self care systems interfacing subscribers or users that should be allowed to initiate provisioning actions with a good response time.
- Provisioning via Applications servers that often offer user service configurations facilities (e.g. via Ut interface) and that will control the validity of user requests before storing the data in the UDC.

In these use cases, provisioning actions that provisioning FEs have to execute are not correlated and are triggered on a per user basis. They are not justifying to use bulk data operations.

6.2 Conclusion and recommendation

There is no justification to develop bulk operations over the Ud interface.

7 Handling of common data

7.1 Description and consequences of the common data concept

In a service profile, several sets of permanent data (in particular subscription data) may have the same values for a large or a very large number of users, so instead of repeating/ instantiating these sets of permanent data with the same values for each user, an important optimisation in the database resource and management is to generate a given set only once that is considered as common data, then a service profile of a given user only contains a reference or an identifier to this set of parameters. An equally important optimisation in the network resource usage is the retrieval of this common data once and stored locally by the relevant FEs such that subsequent subscription data retrievals only contain a reference to the common data.

Hereafter are presented some consequences of this common data concept in the UDC environment:

- This concept is first applied to the UDR that will store common data. It should be considered that given the value of this concept, it is largely applied in many existing implementations and cannot be ignored in a UDR implementation.

The common data concept impacts the information and data models listed in 3GPP TS 32.181 [3] (CBIM, SpIM, AIM, CDM). The analysis of this impact is for further study and should be studied with 3GPP WG SA5.

- Then, the concept of common data may or not apply to FEs and is analysed in sub-clause 7.2.
- Modification of Common Data by OSS may be regarded a bulk data modification since it impacts a large or very large number of users. Synchronizing the data with UDC-external entities (VLR/MME/SGSN/S-CSCF) should not result in signalling flood from FE to that entity. Instead existing reset mechanisms may be used.

7.2 Alternative Solutions

7.2.1 Solutions for FEs that are not provisioning FEs

7.2.1.1 FEs are not aware of the common data concept

In this solution, FEs are not aware of the common data concept and do not store such common data. It means that when a FE requests a user profile from the UDR, the UDR should map the necessary common data into attributes specific to this user according to the Application Data view and Application Data model mapping described in 3GPP TS 32.181 [3]. No common data references or identifiers are transmitted over Ud.

This solution assumes that such a FE has never to create, delete or modify common data.

This solution is depicted in the following figure:

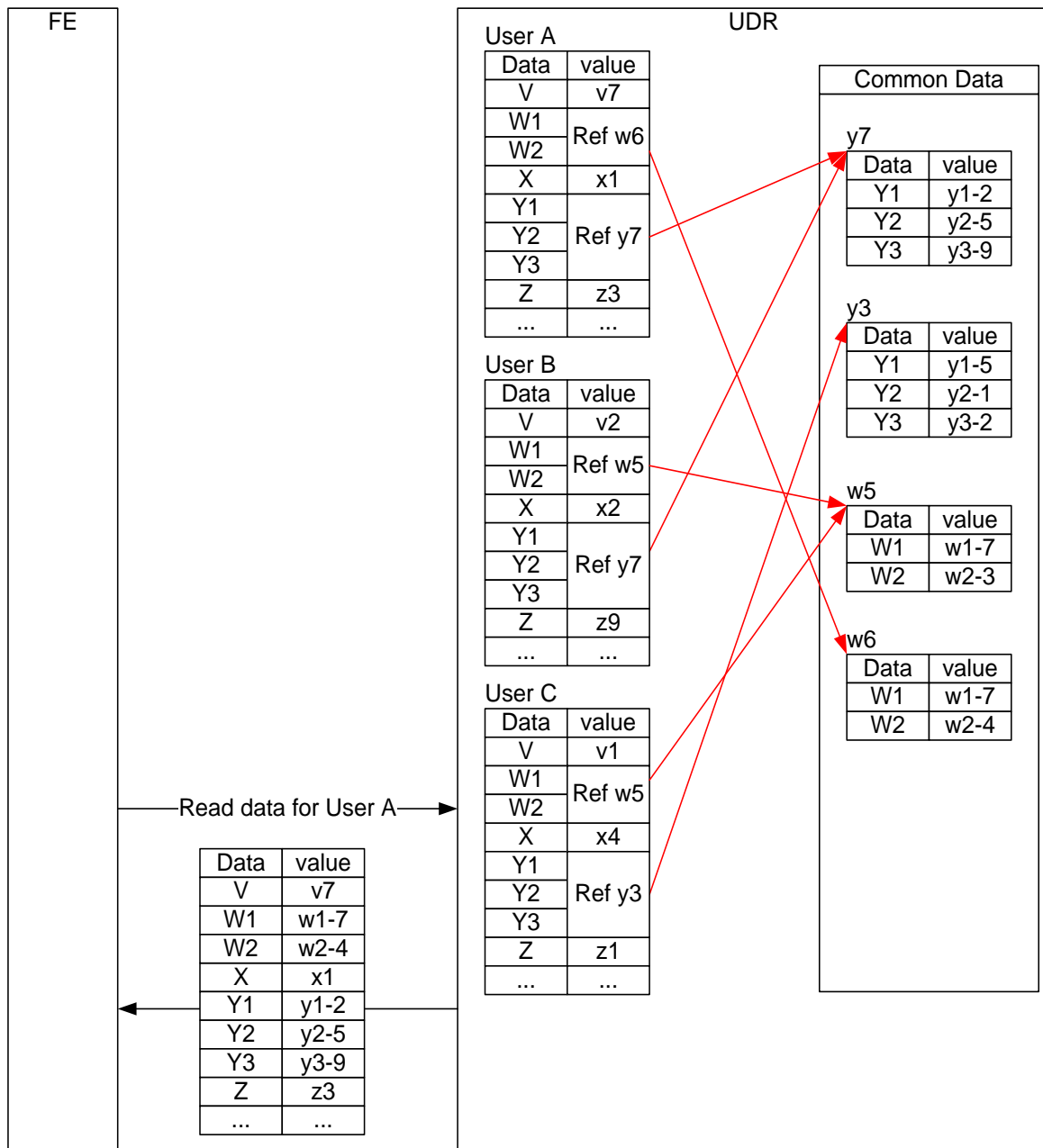


Figure 7.2.1.1-1: FE not aware of common data

Modification of common data by OSS has no impact on the Ud interface.

7.2.1.2 FEs are aware of the common data concept

In this solution, FEs are aware of the common data concept. It means that a FE can receive common data identifiers in user profile over Ud and then is able to retrieve the common data associated to this identifier, either locally if such common data are locally configured in the FE or have already been downloaded to the FE, or by doing a separate request to the UDR.

These FEs that are not provisioning FEs don't have to create, delete or modify common data.

The separated downloads of common data from the UDR to the FE can be done through the Ud interface or through another interface within the scope of UDC.

If it is through Ud, it will require an extension of the applicability of 3GPP TS 23.335[2] procedures to handle common data. The FE will retrieve common data through the Query Ud procedure, by supplying the identification of the common data and without any user identity. For common data that is downloaded and locally stored in a FE, the Ud subscription/notification procedures allow the synchronisation of the common data in the FE and in the UDR.

There is no identified standardisation issue to use the Ud interface for common data for FEs that are not provisioning FEs.

It has also impact on the Application Data Model and the Application Data View for which information common data and the related identifier syntaxes shall be defined. It drives to consider the standardisation of this common data and the related syntaxes of identifiers that, when used over the Ud interface, should be part of the Data Reference Model for HSS user data as studied in 3GPP TR 29.935 [6]; when common data are locally configured in the FE and not sent over the Ud interface, there is no need to standardize the RDM for common data.

These solutions are depicted in the following figures:

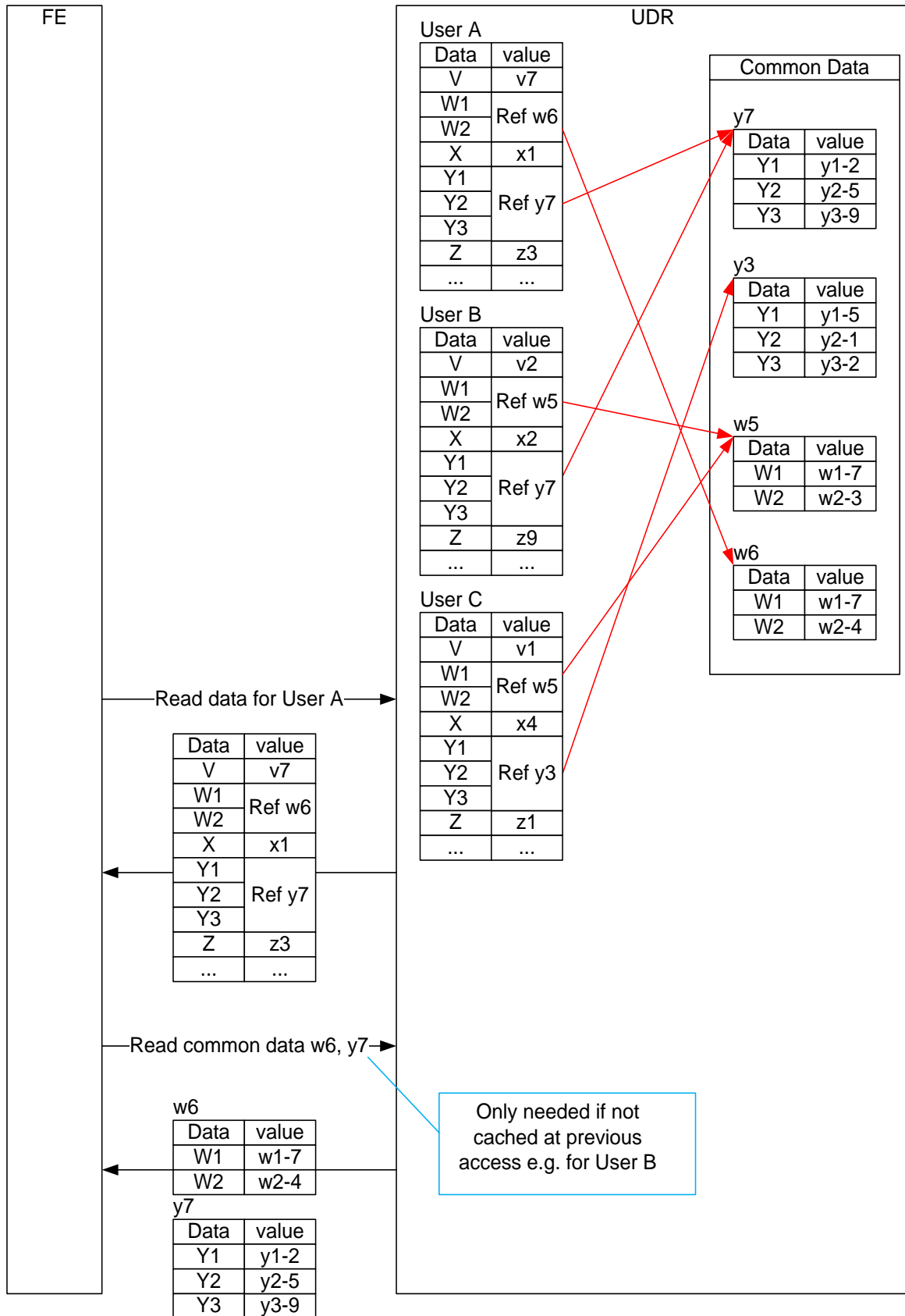


Figure 7.2.1.2-1: Common data retrieved via Ud

Modification of common data by OSS results in broadcasts of notifications to all FEs explicitly or implicitly subscribed to these data modifications.

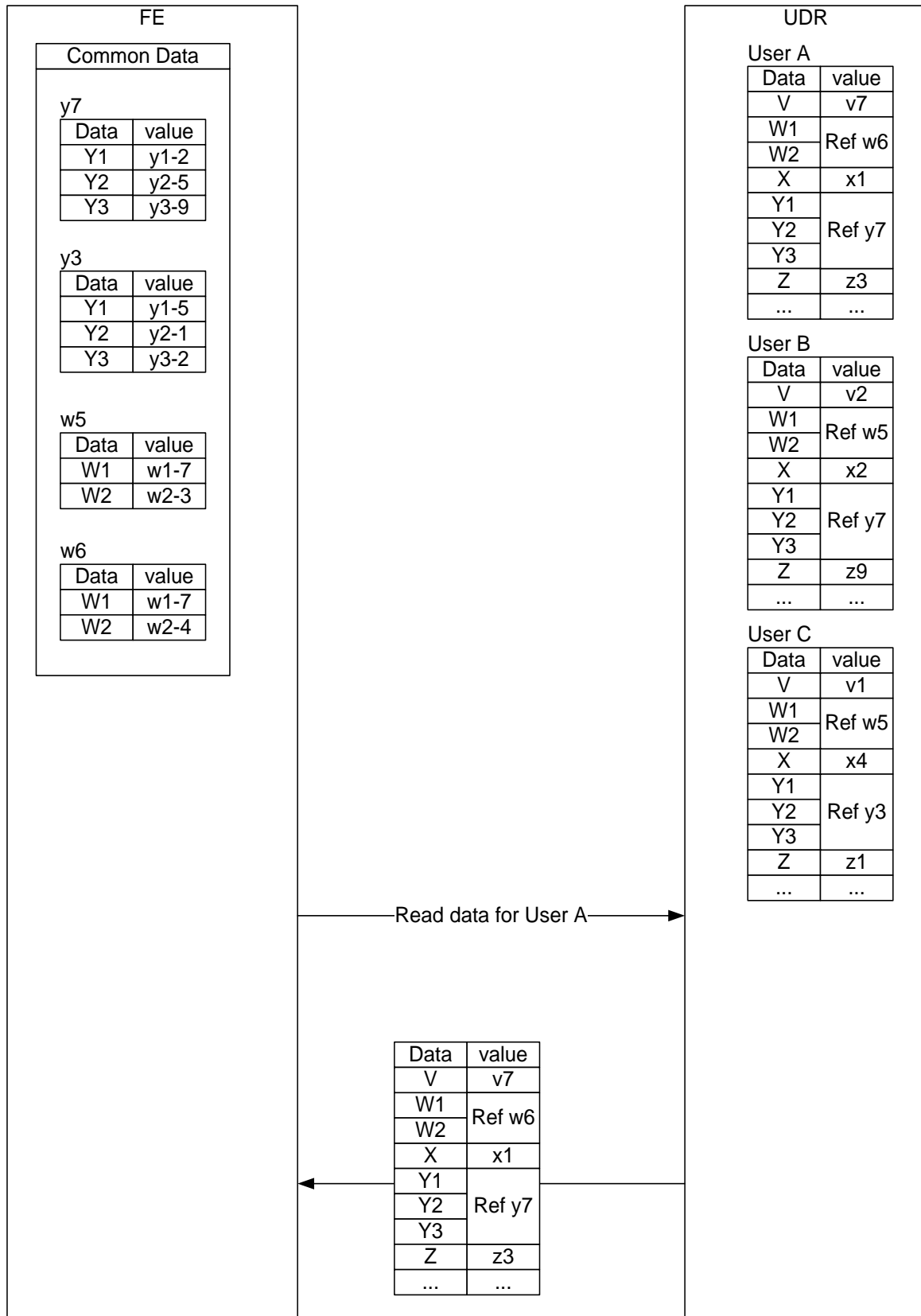


Figure 7.2.1.2-2: Common data locally configured in FE

Modification of common data by OSS is directly done in the FEs and has no impact on the Ud interface.

The determination on which user data are applicable for the common data concept with FE awareness is part of the study in 3GPP TR 29.935 [6].

7.2.2 Solutions for provisioning FEs

Provisioning FEs may have to be aware of the common data concept, as many subscription data associated to a user are in fact common data. A provisioning FE when populating a user profile shall provision the identifiers referring to common data. The common data needs also to be provisioned (before any user profile provisioning referring to this common data), this provisioning can be done through the Ud interface or through another mean out of the scope of UDC. If done via the Ud interface, the provisioning FE will be able to create, delete or modify common data. It may require an extension of the applicability of 3GPP TS 23.335[2] procedures to handle common data.

For the HSS application, the provisioning of common data is not handled by HSS front-ends but only by the OSS through a provisioning gateway as described in 3GPP TS 32.181 [3].

7.3 Solutions comparison

7.3.1 Comparison for non provisioning FEs aware of the common data concept

For non provisioning FEs that are aware of the common data concept, the comparison is between a solution using the Ud interface for the download of common data and a solution with a local configuration of common data in the FEs using another interface to configure these common data in the FEs.

- Solution using the Ud interface for the download of common data:

The analysis shows that the Ud interface does not present issues for such common data downloads; the Ud subscription / notification mechanism with implicit or with explicit subscription ensures data consistency of cached data in the FE with data in the UDR.

The use of Ud requires to describe the common data and their identifier syntaxes within the RDM to ensure interoperability.

- Solution with a local configuration of common data in the FEs:

Common data are not subject to frequent change. It is therefore beneficial (from a standardization point of view, i.e. less standardization effort) to configure common data locally in FEs (in FE specific format) rather than storing them in the UDR and transmitting them on Ud (in standardized RDM format). Local configuration of common data in FEs may be considered a first step; in later releases the retrieval via Ud (including subscription/notification) may be introduced.

Although this solution does not require a full standardization of common data, it requires to standardize the identifiers to be used over the Ud interface towards this common data and which data semantics they cover; it nevertheless avoids a standardisation of the formats of this common data.

Such a local configuration may raise a security breach if data are directly provisioned from OSS in the FE (and not in the UDR). There may be caching in the FE, but no data storing. FEs and UDR are not secured the same way. One of the main objectives in UDC is also to ensure independence between network and database problematics. So, even when common data is locally configured in a FEs, the master of the common data may have to remain in the UDR.

A drawback of another interface to configure the common data in the FEs is to add a new reference point and another protocol. If it is the same protocol as for Ud (LDAP), there is no indication on what it should be different from Ud. If this interface is not standardised (including its associated data model of the common data) it largely reduces interoperability between such a FE and the UDR.

7.3.2 Comparison between non provisioned FEs aware or not of the common data concept

- For FEs aware of the common data concept:

It is expected that signalling load on the Ud interface is significantly reduced when FEs are common data aware.

It requires to have a RDM that specifies identifiers to the common data and the common data themselves as indicated in sub-clause 7.3.1. It may add some more complexity in the structure RDM.

- For FEs not aware of the common data concept:

Regarding to the support of FEs not aware of the common data model, it will require mapping functions in the UDR to fit to the internal database of the UDR that, in practice, will be structured with common data.

7.3.3 Comparison for provisioning FEs

Currently no specificity regarding support or no support of common data by provisioning FEs is identified when compared to the case of non provisioning FEs.

7.4 Conclusions and recommendations

The solution where FEs are aware of common data and common data are transferred (read, subscribed, notified) on Ud (see figure 7.2.1.2-1) is recommended. Definition of common data and their identifier syntaxes shall be part of the used data model. A broadcast of Ud notifications from UDR to all FEs that cache common data is recommended and may be followed by reset procedures synchronizing the data in UDC-external entities.

8 FE to FE communication

8.1 Description

8.1.1 Introduction

This section identifies scenarios where one FE may need to communicate with another FE on a new UDC-specific interface. Existing non-UDC specific interfaces between FEs (e.g. Sh interface between HSS-FE and AS-FE) are not in the scope of this section.

According to current UDC principles it cannot be expected that one FE (of type A) is aware of the existence of another FE (of type B). As a consequence direct communication between these FEs is not foreseen. However, when two FEs (FE1 and FE2) are of the same type (e.g. FE1 is HLR-FE, FE2 is HLR-FE) or are partial implementations of the same type (e.g. FE1 is HSS-IMS-FE, FE2 is HLR-FE) it can be expected that the application logic in the FE1 is aware of the existence of FE2.

The following sections identify scenarios that need to be considered.

8.1.2 Request Relaying

This scenario addresses the case where one FE (FE1) receives a request that needs to be relayed to another FE (FE2) due to the fact that the other FE is currently serving the user (i.e. there is an ongoing FE-session in the other FE). An example scenario is shown in figure 8.1.2-1:

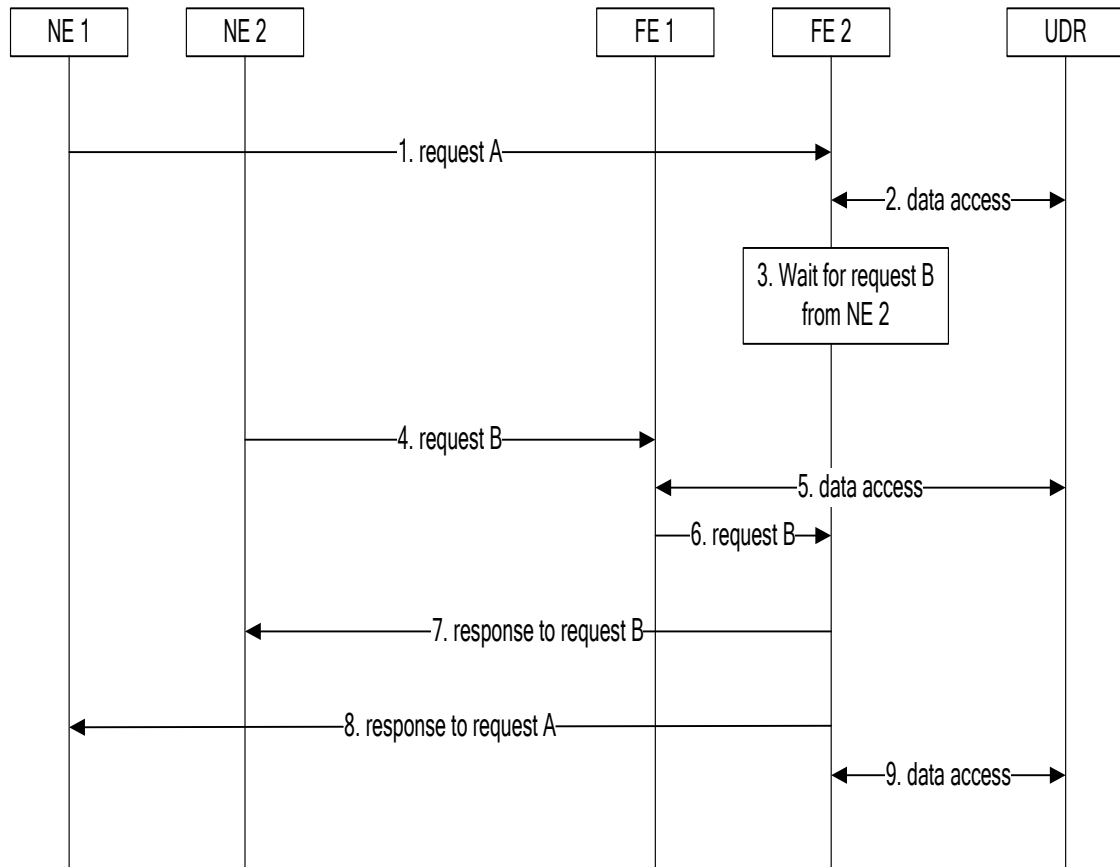


Figure 8.1.2-1

1. NE 1 sends request A. Routing is configured to route the request message to FE 1 or to FE 2. In this example FE 2 is chosen.
2. FE 2 accesses the user's data from the UDR and starts running its application logic.
3. FE 2 detects (as part of its application logic) that it needs to wait for a request B from NE 2. FE 2 starts a supervision timer and keeps the dialogue to NE 1 open.
4. NE 2 sends request B. Routing is configured to route the request message to FE 1 or to FE 2. In this example FE 1 is chosen.
5. FE 1 cannot handle the request and needs to relay it to FE 2. To this end it is assumed that FE 2 has stored in step 2 information in the UDR indicating that FE 2 is currently performing application logic related to request A. FE 1 reads this information (address of FE 2) from the UDR.
6. FE 1 forwards request B to FE 2.
7. and 8. FE 2 completes its application logic and sends responses to NE 1 and NE 2.
9. FE 2 removes from the UDR the information indicating that it is currently performing application logic related to request A.

The described solution is based on the following principles:

1. FE2 in step 2 writes information into the UDR indicating that FE2 is currently processing an FE session related to Request A. This information is read by FE1 in step 5 and deleted by FE2 in step 9. Consequently, the **Reference Data Model is impacted.**

- FE1 acts as a signalling relay function, i.e. it forwards request B to FE2, FE2 may not be aware that the request was forwarded by FE1; hence the response in step 7 is sent to NE2 rather than FE1. Consequently **protocol definition** for request B in step 6 (direct FE to FE communication) is **not impacted**.

An example where the given solution is applicable is in the area of CCBS (see 3GPP TS 23.093): The MAP message StatusReport (Event Report) sent from MSC to HLR-FE may be routed to any HLR-FE but must be relayed to the HLR-FE that has previously sent MAP SetReportingState (Start Reporting) i.e. to the HLR-FE that is currently running the CCBS application logic.

8.1.3 Authentication Vector Request Forwarding

This example shows a scenario where an HLR-AuC-FE and an IMS-HSS-FE are deployed.

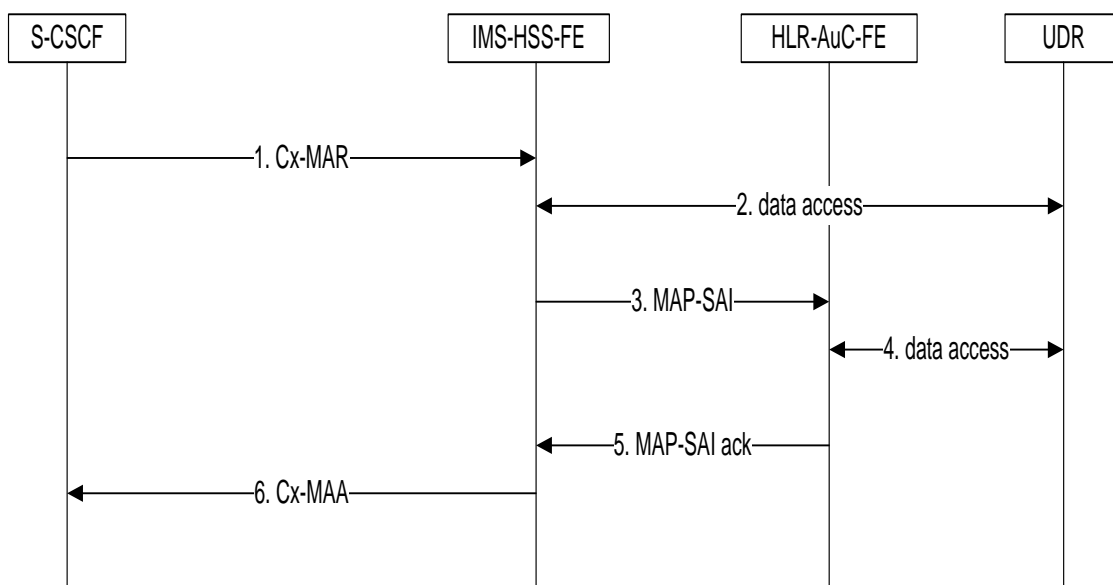


Figure 8.1.3-1

- The IMS-HSS-FE receives Cx-MAR from an S-CSCF.
- The IMS-HSS-FE accesses relevant user data (e.g. read IMSI) in the UDR.
- In this example the IMS-HSS-FE needs to make use of the AuC within the HLR-AuC-FE. It therefore constructs a MAP-SAI message and sends it to the HLR-AuC-FE. The MAP-SAI message contains a RequestingNodeType of "s-cscf".
- The HLR-AuC-FE accesses relevant user data from the UDR and calculates Authentication Vectors.
- HLR-AuC-FE returns the calculated vectors to the IMS-HSS-FE.
- IMS-HSS-FE sends the vectors to the S-CSCF.

The described solution is based on the following principles:

- The MAP-SAI message in step 3 contains a RequestingNodeType of "s-cscf", i.e. the MAP message is slightly enhanced for use on the FE-FE interface (it must be noted that this enhancement is already introduced to 3GPP TS 29.002). **Protocol definition is impacted**.
- There is no need to store additional data in the UDR. **Reference Data Model is not impacted**.
- Data access by the different FEs (step2 and step4) are on different data, i.e. IMS-HSS-FE and HLR-AuC-FE may be connected to different UDRs (see chapter 5), or IMS-HSS and/or HLR-AuC may be monolithic.

8.1.4 Replacing Indirect FE to FE Communication with direct FE to FE Communication

This example shows a scenario where an EPS-HSS-FE (supporting only S6a/S6d, but no Gr) communicates with an HLR-FE (supporting Gr) via the UDR.

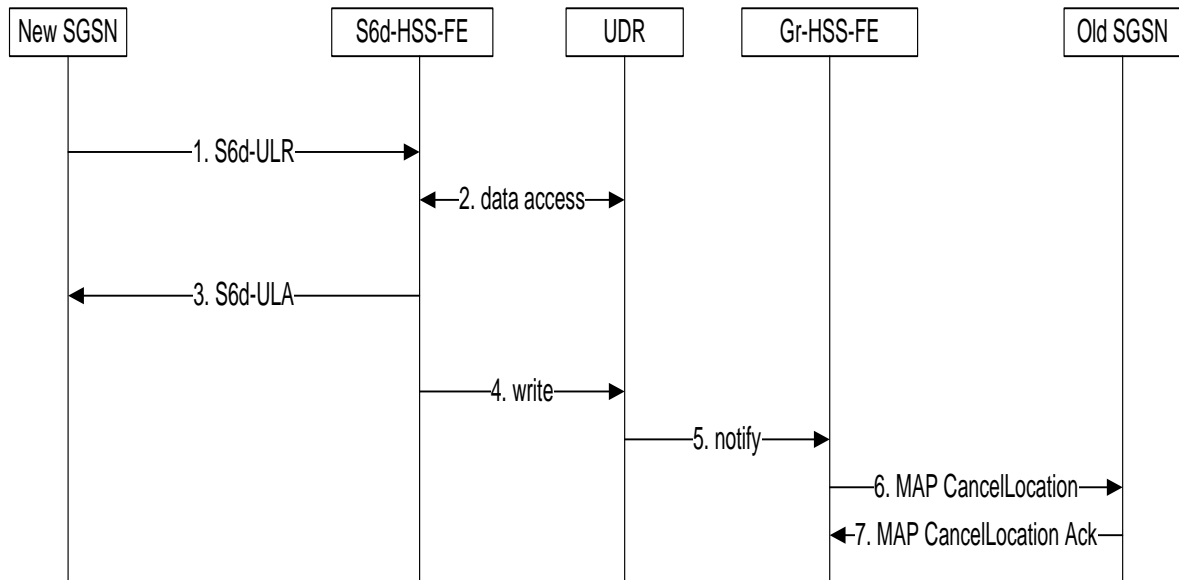


Figure 8.1.4-1

1. The S6d-HSS-FE receives ULR from a new SGSN.
- 2.- 3. The S6d-HSS-FE performs its application logic (read/write data in the UDR, send ULA,...) While doing so, it detects that the old SGSN cannot be contacted via S6d but only via Gr.
4. The S6d-HSS-FE writes some specific data "X" to the UDR. It is assumed that the Gr-HSS-FE has subscribed to the specific data "X"
5. The Gr-HSS-FE receives a notify message indicating that "X" has been modified.
- 6.- 7. Gr-HSS-FE performs the part of the application logic that could not be performed by the S6d-HSS-FE (i.e. sends MAP-CancelLocation and receives the acknowledgement).

The described solution is based on the following principles:

1. There is some data "X" defined in the Reference Data Model that is subscribed by the Gr-HSS-FE. Data "X" could be the SGSN-Number of the SGSN to which a MAP-CancelLocation message needs to be sent and could include also the content of the MAP-CancelLocation message. The S6d-HSS-FE (which cannot send MAP Gr messages) would write data "X" in the UDR (rather than sending MAP-CancelLocation), which results in a notify message to the Gr-HSS-FE. Consequently, the **Reference Data Model is impacted**.

As an alternative solution, the indirect FE-FE communication (step 4 and step 5) could be replaced by a direct communication, i.e. instead of write/notify via the UDR a new direct message between the two FEs could be used. Consequently the RDM would not be impacted but a new message needs to be defined.

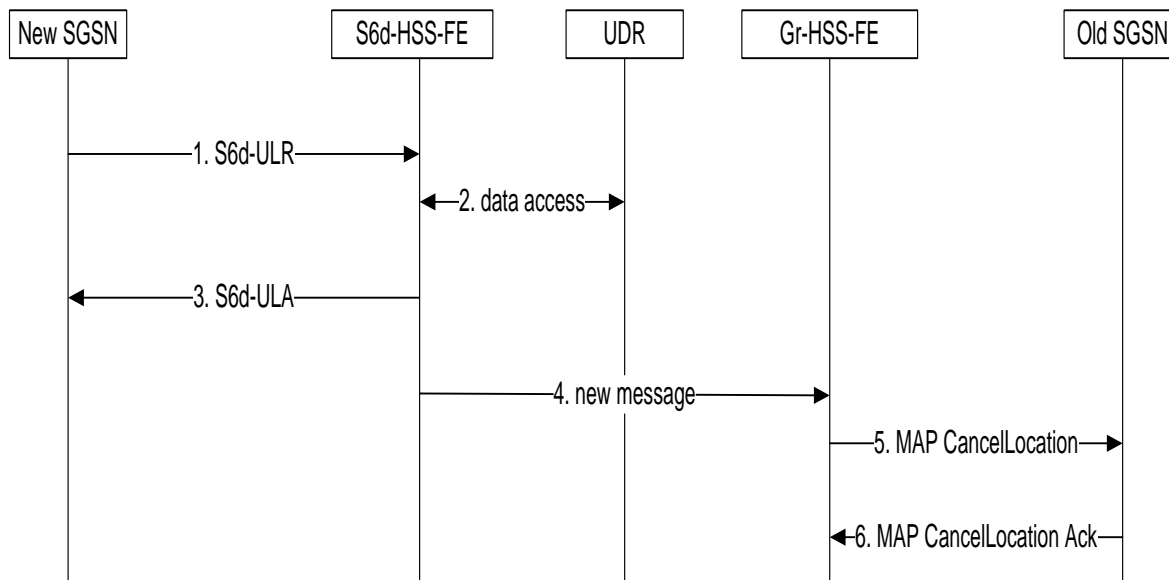


Figure 8.1.4-2

The new message in step 4 (in the general case) could be a re-used (unmodified) message from another interface, a re-used, slightly modified message from another interface (see e.g. section 8.1.3 step 3), or a newly defined message e.g. based on SOAP. For the specific case of figure 8.1.4.-2, only the last option seems to be available.

8.1.5 Summary

Two use cases have been identified for direct FE to FE communication:

1. Both FEs are of the same application type. FE to FE communication is needed because one FE receives a request for a user who is currently served by the other FE.
 Solution 1a: The request is relayed between FEs (see figure 8.1.2-1)
 Solution 1b: ffs
2. Both FEs are (different) partial implementations of an FE of the same type. FE to FE communication is needed because one part of the application logic needed to serve the initial request is performed by one FE and the other part by another FE.
 Solution 2a: No direct FE-FE communication, but indirect via UDR (see figure 8.1.4-1)
 Solution 2b: Direct FE-FE communication; re-use existing message (unmodified)
 Solution 2c: Direct FE-FE communication; re-use existing message (modified), see example in figure 8.1.3-1
 Solution 2d: Direct FE-FE communication; define new message e.g. based on SOAP
 Solution 2e: Avoid the need for FE-FE communication by mandating that an FE-session (application logic) is not distributed between FEs, e.g. with regard to figure 8.1.4-1 or 8.1.4-2, the FE receiving S6d-ULR from the new SGSN must be capable of sending MAP-CancelLocation to the old SGSN.

8.1.6 Standardisation considerations

Two use cases have been identified for direct FE to FE communication, one where the two FEs are of the same application type, the other where the two FEs are different and partial implementations of an application. For each of them, solutions have been identified.

A main remaining question is if these solutions enter the scope of standardisation.

- If a reference point is defined between such two FEs, there will not be a generic protocol. The relevant protocol will depend of the application type.
- In the case where a first FE is simply relaying a request towards another FE (as in the use case described in 8.1.2), this direct FE to FE communication will rely on a subset of the standardised protocol used by the first FE with the network and may be not far from a standardised case, especially if the standardisation allows the use of proxy or relay nodes with the used protocol. Whether or not, and if - to which FE - the message is relayed may

depend on temporary data stored (e.g. by the receiving FE) in the UDR as part of a standardized Reference Data Model.

- When achievable, a possible solution is to solve the communication issue between the two FEs through an indirect communication via the UDR (solution 2a in subclause 8.1.5). Such a solution will use the standardised Ud interface and will also use some data shared by the two FEs: e.g. a data update by one FE generating a notification to the other, so allowing an indirect communication between the two FEs. This data may be part of a standardised Reference Data Model over Ud.

8.1.7 Conclusion

It is recommended that solution 2e (see chapter 8.1.5) is applied, i.e. that FE-FE communication should be avoided.

Direct FE to FE communication should be regarded an exception case. It should be limited to the following situations:

- Where the two FEs are of the same application type in which case solution 1a (see chapter 8.1.5) is recommended. Standardization is currently out of scope of 3GPP.
- Where the two FEs are different partial implementations of the same application type, in which case solutions 2b and 2c (if possible, i.e. when existing mechanisms are available) are recommended. Standardization is currently out of the scope of 3GPP.

9 Collision detection control enhancements

Assertion controls (see IETF RFC 4511 [4]) may be used by the Application Front-Ends to ensure that the UDR shall not process any data modification request unless certain condition(s) are met. Since the FEs may not include this control (i.e. any FE with the right permissions shall modify the data with no further checks performed by the UDR), based on local configuration policy, the UDR may reject a modification request for certain data if the required assertion control is not included. If included, and the condition(s) indicated in the assertion control are met, the UDR shall process the required modification. This ensures that the FE modifying specific data is managing the latest version of that data (i.e. data value managed by the FE is the most recent)

10 Enhancements to subscription to notification

Currently, subscriptions to notifications over Ud do not allow to indicate any condition to require a notification. Hence, every time the data subscribed in the UDR changes, a notification is sent to the Front End, and the Front End possibly discards the notification if other conditions are not fulfilled.

It should be possible for an Application Front-End to request notifications about specific data changes only when certain conditions are met. The FE may indicate these conditions in the subscription request (e.g. request notifications when data A changes from value "x" to value "y" and data B contains value different from "z").

11 Notifications and transactions

11.1 Description

3GPP TS 23.335 [2] has defined notifications used when data changes occur and transactions that allow to perform several data updates operations in one unit of interaction. But the relationship between notifications and transaction has not yet been addressed.

The 3GPP TS 23.335 [2] information flows describing the operations for creating, deleting or updating data contains a step to perform notification procedure. This notification procedure may run before, after or in parallel of sending Create, Delete or Update data answer.

When the data operations belong to a transaction, it is only at the end of the transaction and if it is successful that notifications procedures can take place. Initiating a notification procedure triggered by an operation before the end of the transaction is not possible as a rollback of the transaction may occur restoring the data to its previous state.

11.2 Alternative solutions

Notifications generated by operations belonging to the same transaction may be notified to the relevant FEs according to the two following solutions:

- a first solution is that the notifications generated by the various operations remains independent and sent by independent notifications messages to the relevant FEs according to their subscription;
- the second solution is that the notifications addressed to a FE or a cluster of FEs and issued from the same transaction are grouped into only one notification built according to the associated subscription(s). The Notification protocol described in 3GPP TS 29.335 [5] allows the grouping of an unbounded number of independent objects with their changed attributes.

In this second solution, it is proposed that the grouping of notifications applies independently of the fact that the requested notified data changes have been defined by one or several Subscriptions to notifications.

In both solutions, are only notified the changes due to previous subscriptions to notifications procedure or due to local configuration policy in the UDR as specified in 3GPP TS 23.335 [2].

These 2 solutions may be exclusive meaning that only one of them is applied to a UDR and all its FEs, or they may coexist, meaning that for example, in the Subscription to Notifications, a parameter indicates if the notifications issued from the same transaction shall be grouped into one notification or kept separated. If the second solution is supported in a UDR, it is considered that this second solution should be then always used as there is no strong argument to keep the first one for some cases that have not been identified in this study. So, there is no need to introduce an additional parameter in the Subscription to notification request indicating if notifications associated to a transaction shall be grouped or not.

11.3 Solutions comparison

The main difference between the two solutions is that the second solution avoids inconsistent transitory situations for the notified FE and propagates the unit of interaction concept brought by the transaction into the notification process towards the FE.

In this example, Data A and Data B are modified through a transaction with an operation modifying Data A from value A1 to value A2 followed by another operation modifying Data B from value B1 to B2. In the first solution, the notified front end will receive a first notification where Data A has now value A2, but Data B that the FE may temporarily store will still be with value B1, which is not consistent until the FE receives the second notification with value B2 for Data B.

Another difference is that with the first solution, with a cluster of FE and according to 3GPP TS 23.335[2], the notifications may be delivered to different FEs of the cluster, although the objective could be that all the data changes made in the same transaction be notified to the same FE within a cluster.

The two solutions have different performances impacts: the first solution generates more notifications messages (requests and answers) than the second on the Ud interface. This performance impact may also be further observed on the interfaces between the Front-end and other network entities (over Map or Diameter interfaces), when each Ud notification generates a corresponding procedure over Map or Diameter.

Nevertheless the second solution requires the additional logic and processing to group the notifications into one.

11.4 Conclusions and recommendations

Due to the interest presented by the second solution as described in 11.2 and in 11.3, this solution grouping notifications associated to a transaction, this solution may be offered on the Ud interface as an optional feature of the UDR.

When this optional feature is supported, the notifications associated to a transaction are always grouped to the relevant FEs, taking into account the subscriptions to notifications about the data changes to be notified. The grouping of notifications applies independently of the fact that the requested notified data changes have been defined by one or several Subscriptions to notifications. It is not required to introduce an additional parameter in the Subscription to notification requests indicating if notifications associated to a transaction shall be grouped or not.

Annex A (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
2011-03	CT#51	CP-110096			V1.0.0 for information and approval in CT#51	1.0.0	10.0.0