ETSI TS 123 335 V12.0.0 (2014-10)

Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
User Data Convergence (UDC);
Technical realization and information flows;
Stage 2
(3GPP TS 23.335 version 12.0.0 Release 12)

Reference
RTS/TSGC-0423335vc00

Keywords
GSM,LTE,UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

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

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.
The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2014.
All rights reserved.

DECT™, **PLUGTESTS**™, **UMTS**™ and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ and **LTE**™ are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Intellectual Property Rights

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

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

Foreword

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

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

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

Modal verbs terminology

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

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

Contents

Foreword

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

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

Version x.y.z

where:

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

Introduction

The UDC concept (3GPP TS 22.101 [3]) supports a layered architecture, separating the data from the application logic in the 3GPP system, so that user data is stored in a logically unique repository allowing access from core and service layer entities, named Application Front Ends.

Network elements and functionalities should be designed to access user data remotely and without storing them permanently locally, i.e. the Front Ends shall work in a subscriber dateless configuration.

1 Scope

The present document describes the procedures and signalling flows associated to the technical realization of the 3GPP User Data Convergence (UDC). It furthermore indicates some requirements for the stage 3 specifications.

Special consideration is put in the following areas:

- reference architecture for the UDC concept

- general description of procedures for the user data manipulation (e.g. create, delete, update, etc.)

- identification of the requirements on the UDC for the applicability of the mechanisms described in this document.

User data convergence is an optional concept to ensure data consistency and simplify creation of new services by providing easy access to the user data, as well as to ensure the consistency of storage and data models and to have minimum impact on traffic mechanisms, reference points and protocols of network elements.

Standization of the Data Model for the Ud interface between Front-Ends and the UDR is out of the scope of 3GPP.

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 TR 22.985: "Service requirement for the User Data Convergence (UDC)".

[3] 3GPP TS 22.101: "Service aspects; Service principles".

[4] 3GPP TS 29.002: "Mobile Application Part (MAP) specification".

[5] 3GPP TS 23.002: "Network architecture".

[6] 3GPP TS 32.182: "User Data Convergence (UDC); Common Baseline Information Model (CBIM)".

[7] 3GPP TS 33.210: "3G Security; Network Domain Security; IP network layer security".

[8] 3GPP TS 32.181: "Telecommunication management; User Data Convergence (UDC); Framework for Model Handling and Management".

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].

Front End: a core network functional entity or service layer entity or provisioning entity that can access user data stored in a unique repository.

Front End Identifier: A name that uniquely identifies an FE within the set of all FEs accessing an UDR.

Front End Cluster: FEs handling the same application may be grouped in clusters to differentiate between them e.g. with regard to geographical location, feature support, vendor, or other characteristics. All FEs within a cluster are treated equally for required purposes (e.g. authorization, notifications, etc.).

Application type: The application handled by a FE (e.g. HLR) determines the application type of the FE. The application type is derived from the name indicated by a FE.

Front End Cluster Identifier: A name that identifies a cluster grouped with FEs supporting the same application.

User Data Repository: facility where user data can be accessed stored and managed in a common way.

Transaction: a transaction is a sequence of operations towards the User Data Repository (one or several), performed as a single logical unit of work. A logical unit of work must exhibit four properties, called the ACID (Atomicity, Consistency, Isolation, and Durability) properties, to qualify as a transaction.

A transaction usually consists of the following steps, start transaction, making updates, and end of transaction; the transaction can finish successfully, in that case it is said the updated data is committed, or unsuccessfully, in that case the transaction is cancelled and the updates made till that moment are rolled back so that the data remains as it were before the transaction.

Atomicity: a transaction must be an atomic unit of work; either all of its data modifications are performed or none of them is performed.

Consistency: when completed, a transaction must leave all data in a consistent state. In the User Data Repository, all rules must be applied to the transaction"s modifications to maintain all data integrity. All internal data structures must be correct at the end of the transaction.

Isolation: modifications made by concurrent transactions must be isolated from the modifications made by any other concurrent transactions. A transaction either sees data in the state it was in before another concurrent transaction modified it, or it sees the data after the second transaction has completed, but it does not see an intermediate state. This is referred to as serialization because it results in the ability to reload the starting data and replay a series of transactions to end up with the data in the same state it was in after the original transactions were performed.

Durability: after a transaction has completed, its effects are permanently in place in the User Data Repository. The modifications persist even in the event of a system failure.

3.2 Symbols

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

Ud reference point between a FE and the UDR

3.3 Abbreviations

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

FE Front End
UDC User Data Convergence
UDR User Data Repository

4 User Data Convergence architecture

4.1 UDC System architecture

Figure 4.1-1 presents the reference UDC architecture. UDC is the logical representation of the layered architecture that separates the user data from the application logic, so that user data is stored in a logically unique repository allowing access from entities handling an application logic, hereby named Application Front Ends.

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. Entities which do not store user data and that need to access user data stored in the UDR are collectively known as application front ends.

NOTE: Depending on the different network deployment, there may be more than one UDC in an operator"s network.

Application Front Ends connect to the UDR through the reference point named Ud to access user data.

Reference points towards network elements are marked in discontinuous lines in Figure 4.1-1, and are just shown for information purposes only. Details of the roles of these functional entities are described in sections 4.2.1, 4.2.2, 4.2.3 and 4.2. 4.

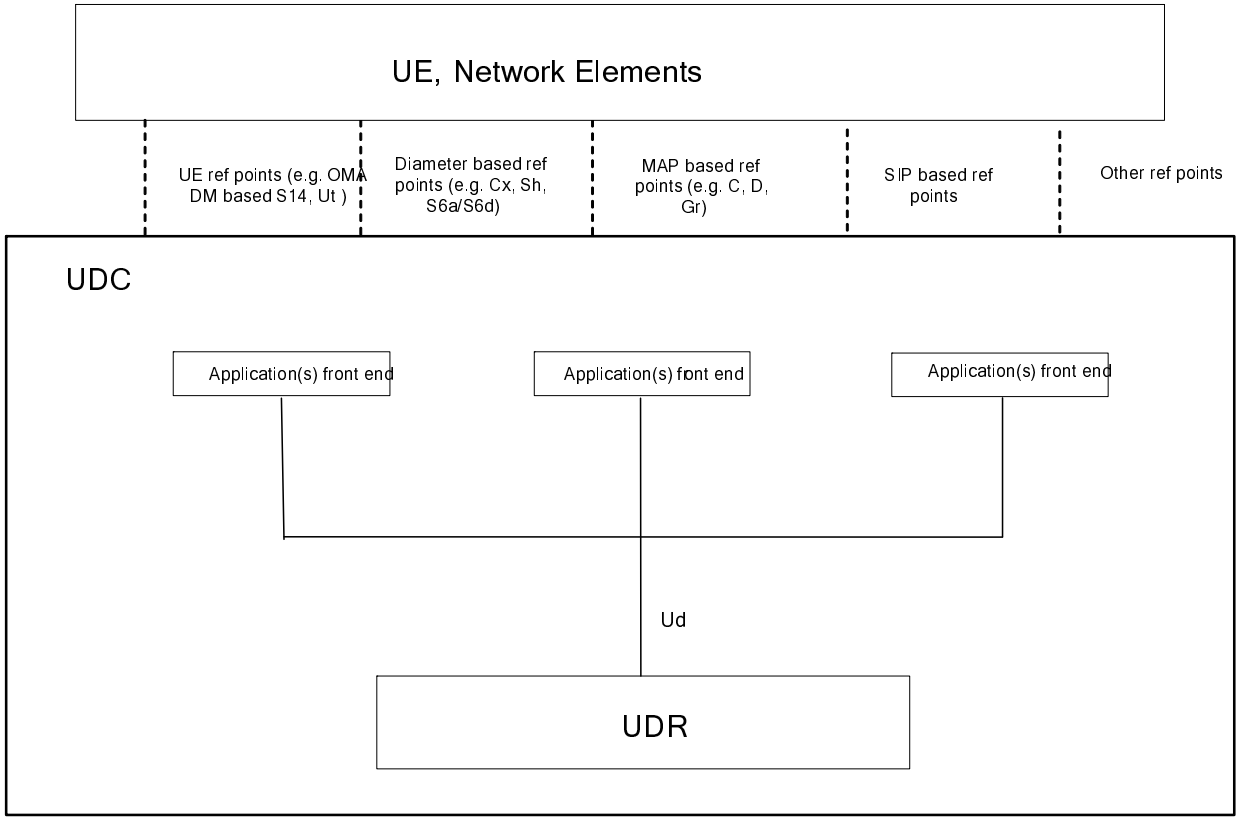

Figure 4.1-1: UDC reference architecture

Figure 4.1-2 shows how the UDC reference architecture is related to the overall network architecture by comparing a Non-UDC Network with an UDC Network. In the non-UDC Network, the figure shows NEs with their own database storing persistent user data and a NE accessing an external database; in both cases, when UDC architecture is applied, the persistent user data are moved to the UDR and concerned NEs are becoming Application-FEs (NE-FEs) according to the UDC architecture. This figure also shows that network interfaces between NEs are not impacted .A Network Element (NE), which in its original form represents application logic with persistent data storage, when the

UDC architecture is applied, may become a NE Front End, since the related persistent data storage is moved to the UDR.

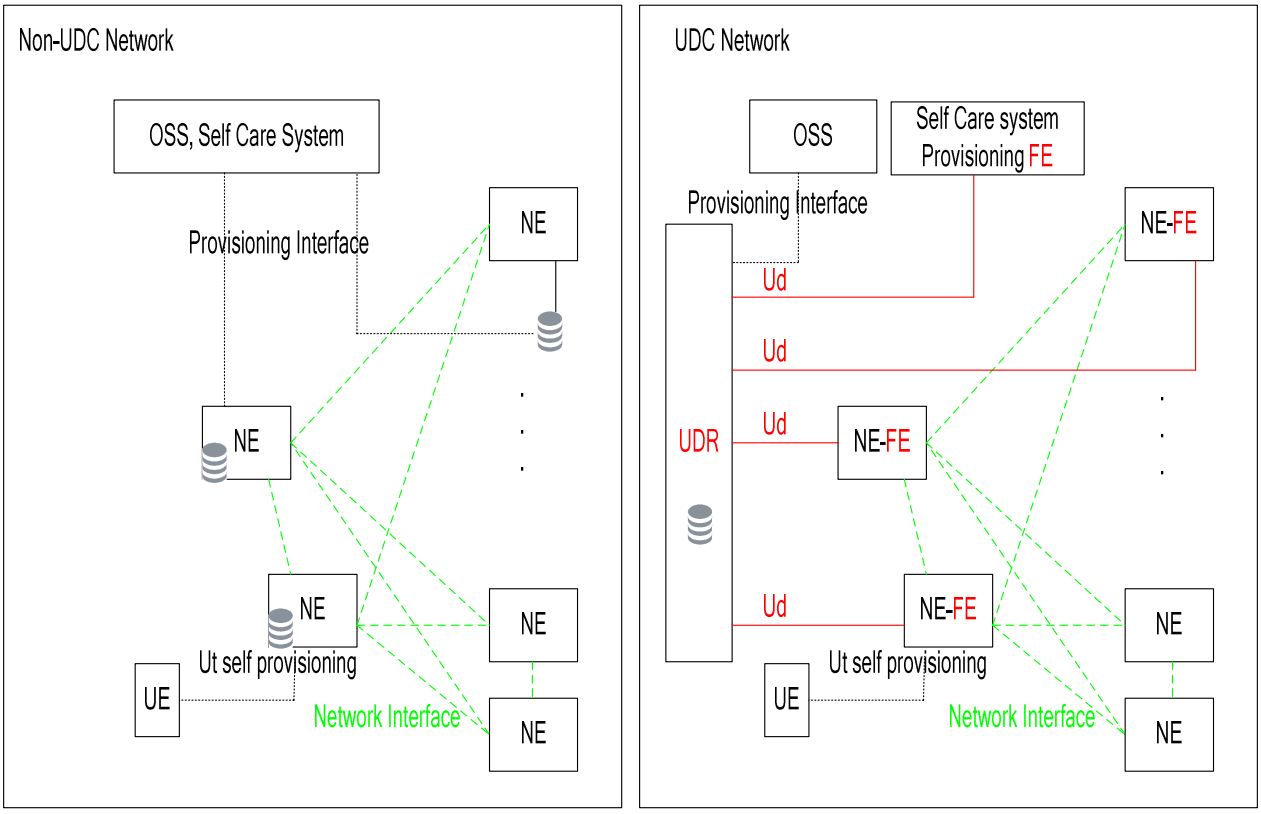

Figure 4.1-2: Comparison Non-UDC Network to UDC Network

4.2 Functional Entities

4.2.1 Application Front Ends

Functional entities, such as the HLR/HSS/AUC, Application Servers, Access Network Discovery and Selection Function in Home Network (H-ANDSF), any other Core Network nodes, Provisioning system, etc., when the UDC architecture is applied, keep the application logic, but they do not locally store user data permanently. These data-less functional entities are collectively known in the UDC architecture as Application Front Ends. The application that is handled by an FE determines the type of FE. A HSS Front End may implement a full or a part of the HSS functionalities as listed in 3GPP TS 23.002 [5], this choice being implementation dependant. The reference points between the different Front Ends and the core and service layers are not affected by the UDC architecture.

4.2.2 Provisioning Front Ends

A Provisioning Front End is an Application Front End for the purpose of provisioning the UDR. A Provisioning Front End provides means to create, delete, modify and retrieve user data. However, the provisioning should not be allowed to manipulate on the common baseline information model.

Provisioning may be associated to an application/implementation and may comprise semantic control specific to this application. It may correspond to different types of provisioning FEs corresponding to different applications logics.

The UDC may support the following provisioning possibilities of user data via Provisioning Front Ends:

- 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 the Ut interface) and that will control the validity of user requests before storing the data in the UDC.

The interoperation between UDC and the provisioning system is out of the scope of this specification.

4.2.3 User Data Repository

The User Data Repository (UDR) is a functional entity that acts as a single logical repository that stores converged user data. The user-related data that traditionally has been stored into the HSS /HLR/AuC, Application Servers, etc., is now stored in the UDR according to a UDC information model. UDR facilitates the share and the provisioning of user-related data throughout services of 3GPP system.

UDR provides a unique reference point to one or more Applications Front Ends, such as: one or more HSS/HLR/AuC-FEs, and one or more AS- FEs. The reference point is named Ud. UDR shall provide support for multiple applications simultaneously.

Application FEs should only be able to access the user data after proper authentication and authorization taking into account security and privacy requirements, i.e. it shall be possible to present different views on the data to the parties which require access, dependent on the authorization. The UDR shall take care of the authorization of the access to the user data. The authorization shall be based on the requestor information, the requested data, and the performed operation.

Ud reference point shall make use of Network Domain Security (3GPP TS 33.210 [7]) where applicable. For applications requiring sensitive data to be transferred over Ud (e.g. permanent authentication keys), encryption shall be required when storing these data in the UDR and transferring it over Ud.

The Application FEs managing these data shall support common algorithm(s) and key(s) for encryption/decryption.

NOTE: Given that different FEs (e.g. HLR/AuC) belonging to the same cluster can serve the same user, the encryption algorithm and key must be known by all FEs serving a given user. The key is not subscriber-specific.

The UDR functional entity may be distributed over different locations or be centralized; it may support replication mechanisms, back up functions and geographical redundancy to secure the storage of data. These functions are out of the scope of the present specification. They do not impact the functional content of the reference point Ud.

The UDR shall be able to store the following types of data:

- Permanent subscriber data: this is subscription data and relates to the necessary information the system ought to know to perform the service. User identities (e.g. MSISDN, IMSI, IMPU, IMPI), service data (e.g. service profile in IMS) and authentication data are examples of the subscription data. This kind of user data has a lifetime as long as the user is permitted to use the service and may be modified by administration means.

- Temporary subscriber data: this is data which may be changed as a result of normal operation of the system or traffic conditions (e.g. transparent data stored by Application Servers for service execution, SGSN number, user status, etc.).

There are certain types of data, as specified in 3GPP TR 22.985 [2] that in principle are not required to be stored in the UDR. However, it might be beneficial to converge these data to the UDR due to, e.g., sharing of the data within a cluster of Application Front Ends. It shall not be required that the UDR stores the following types of data:

- User-content data: content defined by the user and that may be quite large in size (e.g. Photos, videos, SMS, voice mail).

- User data that concerns event data records that can be generated on various events in the usage of services by a user and that can be used not only for charging or billing purposes but e.g. for user profiling regarding user behavior and habits, and that can be valuable for marketing purposes.

- User traffic data: this kind of user data contains call-related or session-related dynamic data (e.g. MSRN, P-TMSI), which are typically stored in VLR, SGSN or S-CSCF. These dynamic data are only used by their owner transitorily and proprietarily, and hardly shared by other services in the short term.

The UDR shall offer a Provisioning Interface to the OSS which serves as a single logical point for consistent provisioning of user data from the operator's OSS system. This interface is out of scope of this specification.

4.2.4 Other Network Elements

Other Network Elements, which in their original form represent pure application logic with no persistent data storage functionality but with user data access towards an external database, when the UDC architecture is applied, may be assumed to perform the functionality of an Application Front End, i.e. access the user data via the Ud interface from the UDR.

4.3 Reference point Ud

This reference point shall allow the different FEs to create, read, modify and delete user data stored in the UDR using the harmonized access interface.

This reference point shall support subscriptions/notifications functionality which allows a relevant FE to be notified about specific events which may occur on specific user data in the UDR. The events can be changes on existing user data, addition of user data, and so on.

Through the reference point, an Application Front End shall only interface with the UDR for the data relevant to its function, and not be impacted by other data that UDR stores for other applications.

The user data that an Application Front End accesses in the UDR through the reference point Ud shall comply with an agreed data structure between the Application Front End and the UDR. Such data structure shall comply with the Application Specific Data Model, specified in 3GPP TS 32.182 [6] and in 3GPP TS 32.181 [8].

Reference point Ud shall support transactions. Operations and transactions carried out over Ud shall support the ACID (Atomicity, Consistency, Isolation, and Durability) characteristics.

4.4 Front-End Session

The application logic in a Front End is performed during an FE-Session. An FE-Session starts either with the receipt of a request message on one of the supported interfaces from the UE, Core Network, service Layer or OSS, or with receipt of a notification message on the Ud interface from the UDR.

Before an FE-Session starts, the FE does not have any user data stored.

During an FE-Session the FE

- may (depending on the application logic) read user data from the UDR and store them temporarily locally;
- may (depending on the application logic) write user data to the UDR;
- may (depending on the application logic) communicate with entities of the Network or Service Layer or OSS on supported interfaces;
- shall delete all temporarily locally stored user data when the FE-Session is completed.

After completion of an FE-Session, the FE does not have any user data stored and the FE does not maintain any state information. A subsequent FE-Session for the same user triggered by a new request message on one of the supported interfaces from the UE, Core Network, service Layer or OSS, or a new notification message on the Ud interface from the UDR may be performed by a different FE.

It must be noted that the FE-Session concept may have the following side effects on the core network and service layer:

- While in networks that do not support UDC, means are needed to route messages destined for an HSS to the correct HSS (i.e. the HSS that serves the user in question), networks that support UDC may deploy several HSS-FEs all of which have access to the UDR and hence can serve any user. Instead of routing towards the correct HSS, routing towards one of the available HSS-FE is needed. HSS-FE selection may allow load sharing or failover; details of the selection algorithm are operator specific and out of scope of this specification.
- While the FE session duration may vary (depending on the application logic), means should be provided by the different FEs (e.g. storing temporary data in the UDR to be shared by other FEs) so that any request may be handled by any FE at any given time.

4.5 UDR Session

The UDR needs to process read accesses and write accesses received via the Ud reference point.

When receiving a read-request the UDR shall check whether the requesting FE is allowed to read the requested data. If it is not, the request shall be rejected, otherwise the requested data value shall be returned to the FE respecting the FEs data view.

When receiving a write-request the UDR shall check whether the requesting FE is allowed to write the requested data. If it is not, the request shall be rejected, otherwise the UDR shall

- check whether a notification message needs to be sent via the Ud reference point to a suitable and available FE. If so, the UDR shall send the notification message;
- perform the write-request and return a successful response to the FE.

5 User Data convergence information flows

5.1 General

This section documents the main procedures on the Ud reference point that are used by the different Front Ends. These procedures are described using text description as well as information flow diagrams. The procedures described in this document are meant to provide a high level description and are not intended to be exhaustive.

In the following sections, the multiple network elements are depicted as a single entity, since the procedures are common for all applications.

These procedures assume that the existing network elements accept a request message sent by any FE, e.g. a MSC/VLR accepts a MAP Cancel Location sent by any HLR-FE, an AS accepts Sh-Notif from any HSS-FE and an S-CSCF accepts Cx-Deregister from any HSS-FE. Figure 5.1-1 shows the general UDC information flow. See Annex A for specific examples.

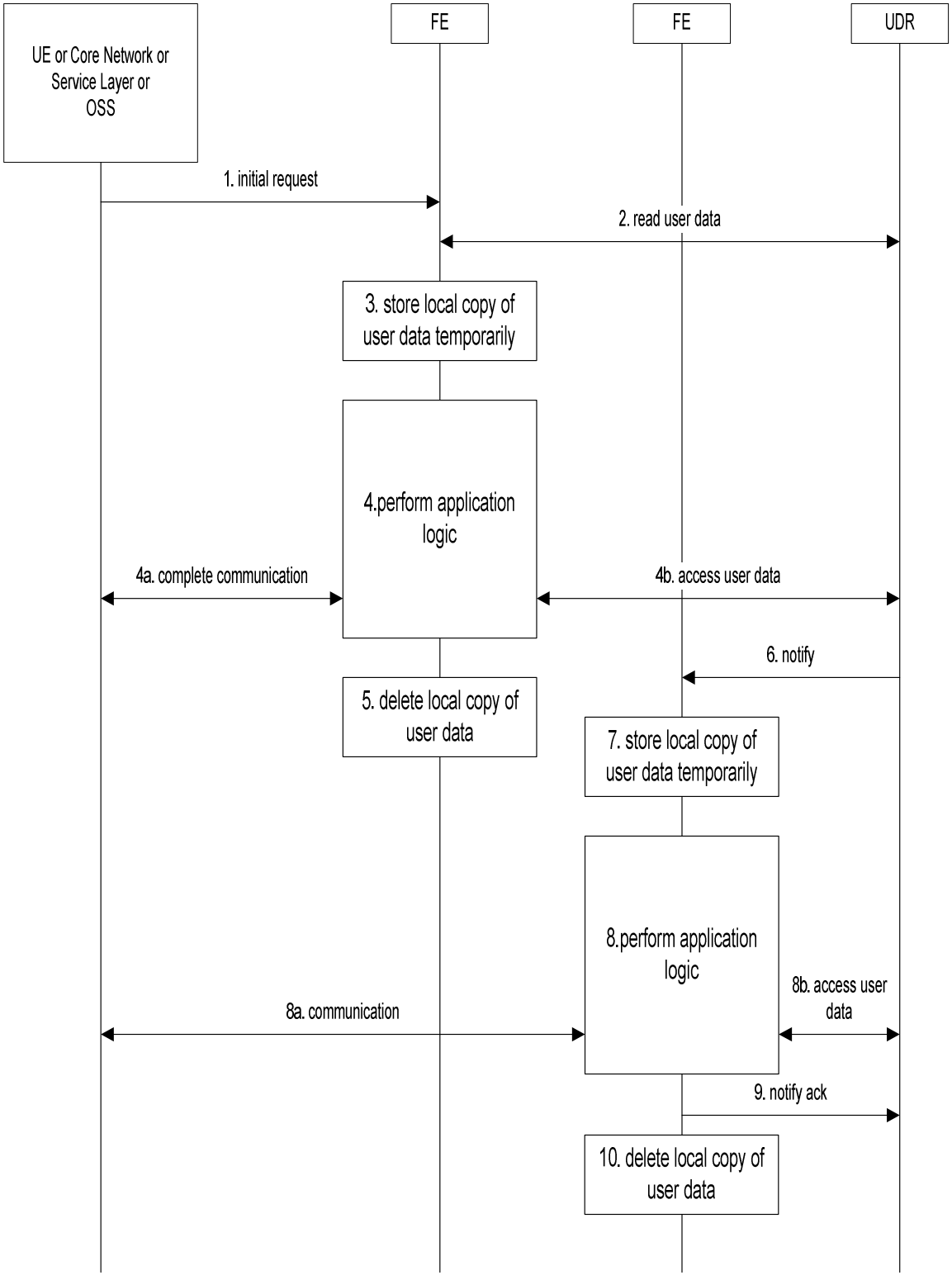

Figure 5.1-1: General UDC Information Flow

1. The FE receives an initial request on one of the supported interfaces from UE, Core Network, Service Layer or OSS.

2. When receiving an initial request message, the FE may read user data from the UDR.

3. The FE shall store the read user data (if any) as a temporary local copy and use it when performing its application logic. There may be applications that do not need to retrieve and store user data from the UDR in order to perform the application logic.

4. The FE performs its application logic.

4a. As part of performing the application logic, the FE shall continue and complete the communication with the UE, Core Network, Service Layer or OSS. This may include sending messages to and receiving messages from entities within the UE, Core Network, Service Layer or OSS other than the entity that sent the initial request.

4b. As part of performing the application logic, the FE shall access user data in the UDR if so required by the application. This may happen more often than once. Steps 4a and 4b may be performed in any order of sequence.

5. After application logic is completed, the FE shall delete its temporary local copy of user data.

6. The UDR shall send a notification message to an appropriate FE if the data modified in step 4b are subscribed data for notification. The notification message shall include user data. More than one notification messages are sent if the modified data are subscribed more often than once.

7. The FE shall store the received user data as a temporary local copy and use it when performing its application logic.

8. The FE performs its application logic.

8a. As part of performing the application logic, the FE shall communicate with the UE, Core Network, Service Layer or OSS if so required by the application.

8b. As part of performing the application logic, the FE shall access user data in the UDR if so required by the application. This may happen more often than once. Steps 8a and 8b may be performed in any order of sequence.

NOTE: In the UDR step 8b may result in another notification message being sent to an appropriate FE, which is not shown in the figure.

9. After application logic is completed, the FE shall send a notify acknowledgement to the UDR. Depending on the application the notify acknowledgement may be sent earlier, e.g. immediately after step 6.

10. After application logic is completed, the FE shall delete its temporary local copy of user data.

5.2 Requirements

The following points are considered as requirements for the purpose of these procedures.

1. It shall be possible for an authorized Front End to read relevant user data stored in the UDR.

2. It shall be possible for an authorized Front End to modify (i.e. create, update, and delete) relevant user data stored in the UDR.

3. The UDR shall support notifications to the related Front Ends about changes of user data which they have subscribed to. Specifically, the UDR shall allow applications to subscribe to specific events on specific data of specific users.

4. The UDR shall support controlled access. Accordingly, UDR shall authenticate and authorize Application Front Ends.

   The authentication shall be based on the identity provided by the FE when accessing the UDR.

   The authorization/access control shall be based on the following criteria:

   - the requested user"s network (e.g. PLMN identity)
   - identity provided by the FE
   - application type
   - the user data which are requested
   - the request type (e.g. query, modify)

NOTE: The UDR derives the application type from the name indicated by the FE. If the FE provided the Front End Identifier the UDR can also derive the Front End Cluster Identifier(s).

5. Access to the UDR shall be independent of the structure of the data models, i.e. the changes in the data models shall not affect these procedures.

6. It shall be possible to present different views on the user data to the different FEs which require access.

7. A group of Front Ends (or a single Front End) for a specific application type shall be distinguished by a unique Front End cluster identifier.

8. The UDR may store the current Front End identifier/Front End cluster identifier(s) when certain user data (e.g. user's location, Sh AS subscription data, service data settings, etc.) is changed as part of the user data and may use this information to determine which Front Ends should be used for notifications.

5.3 Querying data from the UDR

The Query data procedure is used by an FE to retrieve data from the UDR.

The information flow for the Query data procedure is shown in figure 5.3-1:

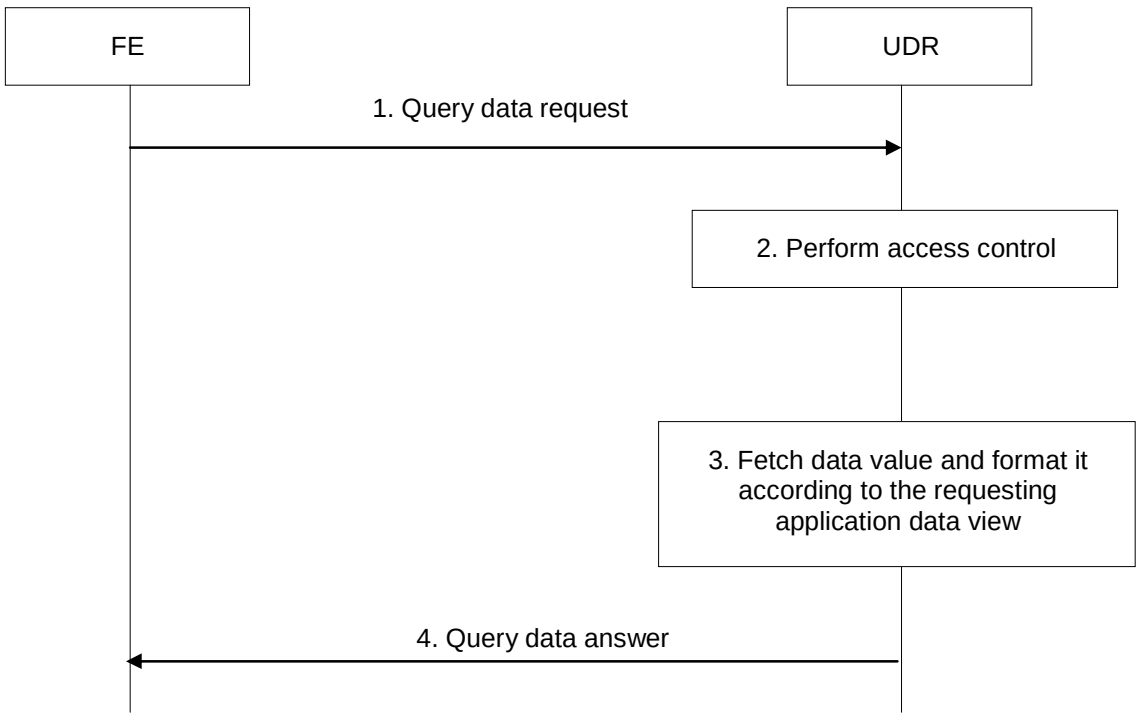

Figure 5.3-1: Query data procedure

1. When an application FE - during processing of its application logic - needs to retrieve user data from the UDR it shall issue a Query data request message and send it over the Ud reference point to the UDR. The message shall contain:

   - the FE Identifier or the FE Cluster Identifier

   - the user identity
     an identity of the user, e.g. IMSI, MSISDN, IMS public user identity, IMS private user identity

   - the requested data
     the identification of the data of which the value is requested

Data identification and structure shall comply with the requesting FE's data view.

2. The UDR shall perform access control to check whether the FE/application type is allowed to read the requested data. If it is not, an unsuccessful response shall be returned to the FE and steps 3 and 4 shall be skipped.

3. If the access control check is successful, the UDR shall fetch the value of the requested data and format it according to the requesting FE's data view.

4. The UDR shall return a Query data answer message to the FE. The FE then shall continue processing its application logic with the retrieved data.

5.4 Creating data within the UDR

Create data procedure is used by a FE to insert a new user data record into the UDR, e.g. when a provisioning FE creates an account for a new user, or creates a new service profile for an existing user.

The information flow for create data procedure is shown in figure 5.4-1.

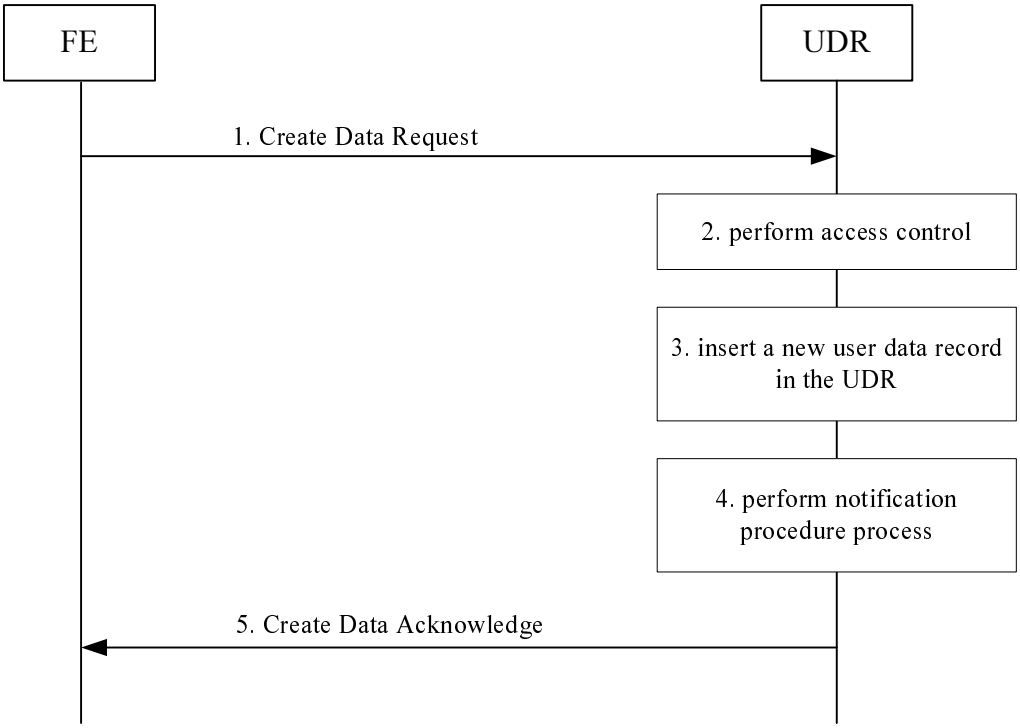

Figure 5.4-1 Create Data flow

1. When an application FE - during processing of its application logic - needs to insert new user data in the UDR, e.g. open a new user account in the database, it shall issue a Create Data Request message and send it over the Ud reference point to the UDR. The message shall contain:

   - the FE Identifier or the FE Cluster Identifier

   - the user identity
     an identity of the user, e.g. IMSI, MSISDN, IMS public user identity, IMS private user identity

   - the identification of data to be inserted.

   - the new data value.

2. The UDR shall perform access control to check whether the FE/application type is allowed to create the requested data. If it is not, an unsuccessful response shall be returned to the FE and steps 3 and 4 shall be skipped.

3. If the access control check is successful, the UDR shall insert the user data with the new data value.

4. If the notification triggering conditions are met, the UDR shall perform notification procedure. This procedure may run before, after or in parallel of sending Create data answer (see step 5). See section 5.8.

5. The UDR shall return a Create data answer to the FE. The FE then shall continue processing its application logic.

5.5 Deleting data from the UDR

Delete Data procedure is used by a FE to delete user data stored in the UDR, e.g. when a provisioning FE deletes a service profile for an existing user, or removes the account of a user.

The information flow for Delete Data procedure is shown in figure 5.5-1.

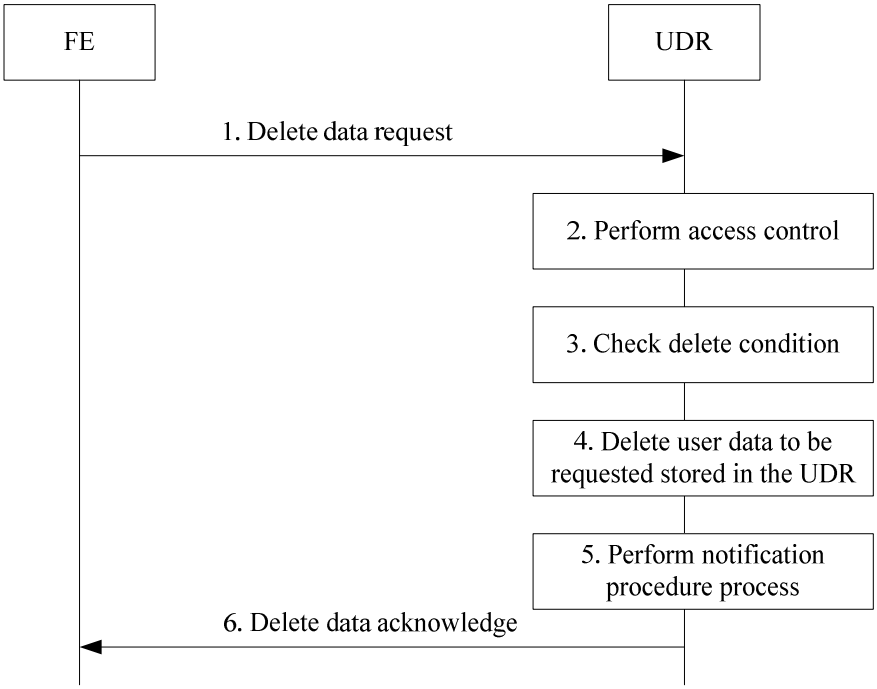

Figure 5.5-1 Delete Data flow

1. When an application FE - during processing of its application logic - needs to delete user data from the UDR, e.g. delete user service profile, it shall issue a Delete Data Request message and send it over the Ud reference point to the UDR. The message shall contain:

   - the FE Identifier or the FE Cluster Identifier
   - the user identity
     an identity of the user, e.g. IMSI, MSISDN, IMS public user identity, IMS private user identity
   - the identification of data to be deleted.

   The message may contain:

   - the delete condition, if supported by the UDR

2. The UDR shall perform access control to check whether the FE/application type is allowed to delete the requested data. If it is not, an unsuccessful response shall be returned to the FE and steps 3-5 shall be skipped.

3. If the message contains the delete condition, the UDR shall check if the delete condition is satisfied. If it is not, an unsuccessful response shall be returned to the FE and steps 4-6 shall be skipped

4. If the access control check is successful and the delete condition is satisfied, the UDR shall delete the requested user data.

5. If the notification triggering conditions are met, the UDR shall perform notification procedure. This procedure may run before, after or in parallel of sending Delete data answer (see step 6). See section 5.8.

6. The UDR shall return a Delete data answer to the FE. The FE then shall continue processing its application logic.

5.6 Updating data within the UDR

The Update data procedure is used by an application FE to modify user data in the UDR.

The information flow for the Update data procedure is shown in figure 5.6-1:

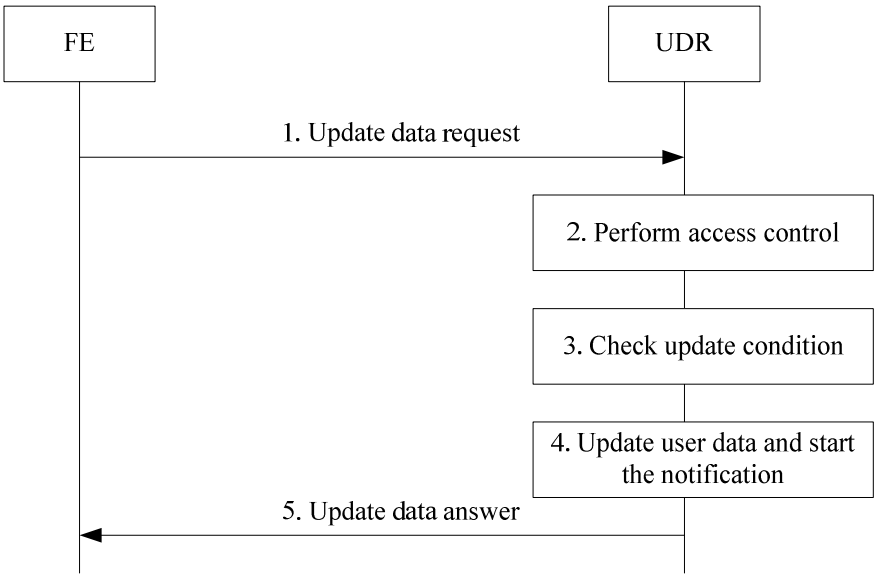

Figure 5.6-1: Update data procedure

1. When an application FE - during processing of its application logic - needs to update user data in the UDR it shall issue an Update data request message and send it over the Ud reference point to the UDR. The message shall contain:

   - the FE Identifier or the FE Cluster Identifier

   - the user identity
     an identity of the user, e.g. IMSI, MSISDN, IMS public user identity, IMS private user identity

   - the requested data
     the identification of the data of which the value is to be updated.

   - the new data value
     value of the data that is to be written in the UDR

   The message may contain:

   - the update condition, if supported by the UDR

   Data that is updated may have a complex structure with multiple attributes. So the data update achieved through the Update data procedure may comprise addition, modification or deletion of some attributes of this data.

   Data identification and structure shall comply with the Data view associated to the application FE.

2. The UDR shall perform access control to check whether the FE/application type is allowed to update the requested data. If it is not, an unsuccessful response shall be returned to the FE and steps 3-5 shall be skipped.

3. If the message contains the update condition, the UDR shall check if the condition is satisfied. If it is not, an unsuccessful response shall be returned to the FE and steps 4-5 shall be skipped.

4. If the access control check is successful and the update condition is satisfied, the UDR shall update the requested data with the new data value. If the notification triggering conditions are met, the UDR shall perform the Notification Procedure. This procedure may run before, after or in parallel of sending Update data answer (see step 5). See section 5.8.

5. The UDR shall return an Update data answer message to the FE. The FE then shall continue processing its application logic.

5.7 Subscription to Notifications

This procedure is used by FE to perform the subscription/un-subscription to notification to specific events which occurs on specific user data stored in the UDR. The events can be changes on existing user data, addition of user data, and so on. The information flow for the Subscription to Notifications procedure is shown in figure 5.7-1.

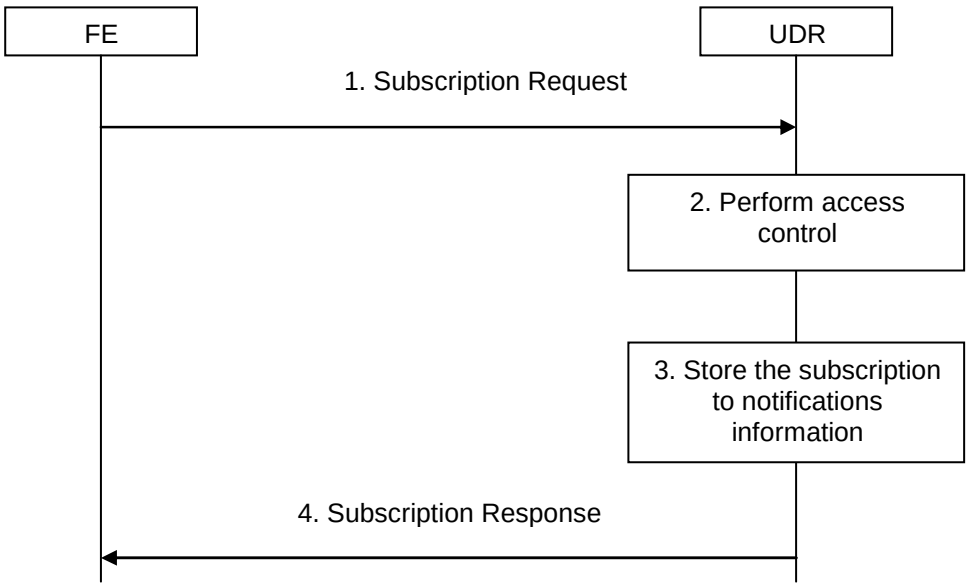

Figure 5.7-1: Subscription procedure

1. When an FE wants to receive notifications of specific events on specific user data stored in the UDR, it shall construct a Subscription Request message and send it over the Ud reference point to the UDR. The message may consist of the following information:

   - the FE Identifier or the FE Cluster Identifier

   - User identity, e.g. IMSI, MSISDN, IMS public user identity or IMS private user identity. The User Identity may not be present indicating that the subscription is applicable to all users in the UDR.

   - Subscription Type

     The Subscription Type indicates whether this request is to subscribe or to unsubscribe.

   - Notification Type. It indicates whether this subscription request should result in a notification to any FE of the application type or cluster identifier, or in a notification to the FE requesting the notification.

   - Subscription to Notifications information, which consists of:

     - Identification of the requested data

       The Identification of the requested data indicates the data to which notifications are subscribed.

     - The notification condition(s)

The notification condition indicates the specific events on which the notification shall be triggered. It shall consist of the addition of the requested data, the deletion of the requested data or the changes of the requested data.

- The expiry time

  If the subscription is not permanent, the expiry time indicates the point in time when the subscription to notifications expires.

- The original entity identity

  If the subscription request is related to a UDC external entity (e.g. AS) subscription request, this item indicates the original entity which sent the subscription request message to the FE in order to subscribe to the notification of the specific events of the requested data. The item shall contain the address or the name of the original entity.

2. When receiving a Subscription Request from a specific FE, the UDR shall perform access control to check whether the FE is allowed to perform the subscription to the UDR on the requested data. If not, an unsuccessful response shall be returned to the FE, and steps 3 and 4 are skipped.
3. The UDR shall store the subscription to notifications information.
4. The UDR shall return a successful response message to the FE.

NOTE 1: Since Subscription to Notifications information cannot be shared among different FEs (i.e. a FE cannot read from the UDR subscription information related to another FE), and for interoperability purposes, careful consideration is to be taken into account when using subscriptions over Ud. For example, an application FE could subscribe to notifications about certain data changes without being aware that another FE was already subscribed for the purpose of performing the same operations towards the external UDC entities.

NOTE 2: Applications that require subscribe/unsubscribe to notifications about many data changes on a per user basis and for common conditions (e.g. HLR/HSS) can make use of UDR pre-configured subscriptions in order to decrease the signalling over Ud. See information flows (annex A).

5.8 Notification of data modification

5.8.1 Description

The Notification procedure shall be used by the UDR to notify an FE about modification of data, when data in the UDR is added, modified or deleted, and an FE needs to be informed about this, due to a previous subscription to notifications procedure (as defined within section 5.7) or due to local configuration policy in the UDR. No notification should be done towards the FE at the origin of a data modification or to a FE belonging to the cluster of the FE at the origin of a data modification.

The information flow for the Notification procedure is shown in figure 5.8-1:

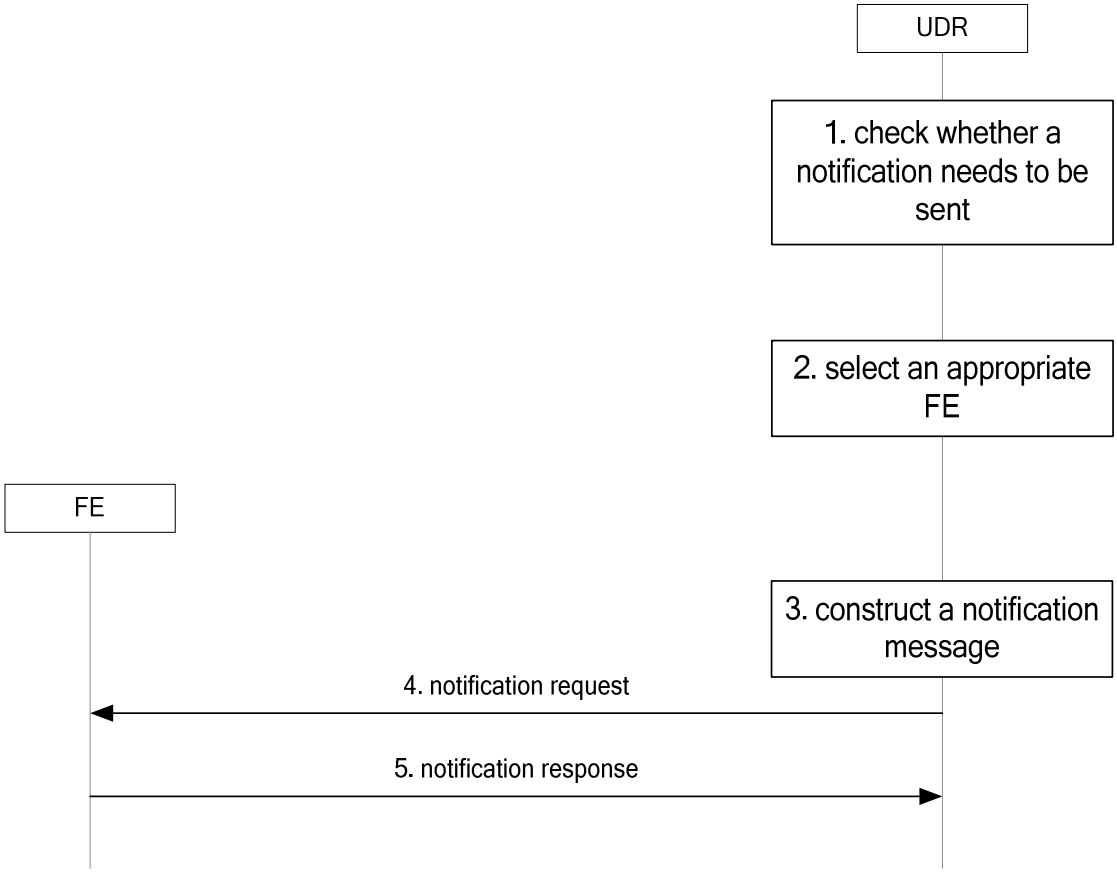

Figure 5.8-1: Notification procedure

1. The Notification procedure in the UDR is started by the Update data procedure, the Create procedure, or the Delete procedure; see section 5.6, 5.4 and 5.5 respectively.
   The UDR shall check whether the relevant notification condition(s) are met. If not met, the following steps shall be skipped and the procedure terminates.

NOTE: The conditions based on local configuration policy in the UDR (e.g. application type of the FE which performs the create, delete or update procedure, presence or absence of other user data) are operator specific and out of scope of this specification.

2. If the notification condition(s) are met the UDR shall select an available FE that supports the relevant application. If the notification is the result of a Subscription to Notification procedure, the FE selection shall take into account the value of the Notification Type information element:

   - If the Notification Type indicates that the notification is to be sent to the FE requesting the subscription, the UDR shall select the FE that requested the notification.

   - If the Notification Type indicates that the notification is to be sent to any FE of the application type or cluster identifier, the UDR shall select an appropriate FE of the application type or cluster identifier as applicable.

NOTE: Details of the FE selection algorithm can be operator specific and are out of scope of this specification.

3. The UDR shall fetch the data that are needed by the FE to perform the relevant application logic, such as the value of updated data, and may fetch other additional data based on local configuration policy in the UDR, such as the previous value of updated data, the original subscribing entity identity, etc. to construct a notification request message that includes data (in FE data view).

4. The UDR shall send the notification request message to the selected FE. The FE shall perform the relevant application logic.

5. The FE shall return a response message to the UDR to indicate success or failure. If no response is received or a failure is indicated, the UDR shall repeat the procedure starting with step 2 and selecting a different FE.

5.8.2 Notifications and transactions

An optional feature allows grouping of notifications associated to the data changes occurring within a transaction. When this optional feature is supported, the notifications of data changes associated to a transaction shall be grouped into one notification procedure to each relevant FE or cluster of FEs, taking into account the subscriptions to notifications or the local configuration policies that requested the notifications of the data changes. The grouping of notifications shall apply independently of those subscriptions to notifications or local configuration policies that requested the notifications of the data changes.

Annex A (informative): Information flows

A.0 Introduction

The following information flows examples show the possible use of Ud procedures by application front-ends such as HLR, HSS front-ends when executing some of their own procedures with the network

A.1 Information flows with Query data procedure over Ud

A.1.1 General

When user data (temporary or permanent) has to be fetched, the application front-end shall perform a query towards the UDR.

The following flows show an example of a terminating call in CS network and an example of a re-registration procedure in an IMS network. These scenarios do not address the mechanisms for access authorisation in the UDR. These scenarios only address the specific actions of traffic events that are currently in effect.

A.1.2 CS terminating call information flow example

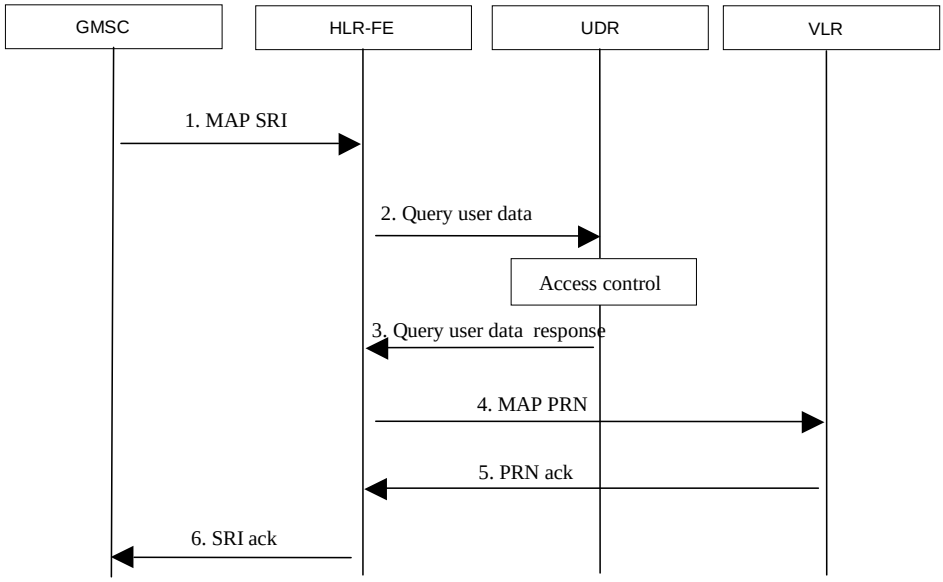

Figure A.1.2-1: CS terminating call example with Ud Query from HLR-FE

1. The GMSC initiates MAP Send Routing Info towards the HLR
2. Upon reception of SRI, the application specific front-end (HLR-FE) fetches all the user data it needs to perform its application logic (e.g. VLR number, barring indicators, call forwarding data) from the UDR through a Ud Query procedure.
3. After applying the proper access control (i.e. the front-end is allowed to read that type of data), the UDR responds with the requested user data.

4. The HLR-FE sends MAP Provide Roaming Number to get a MSRN from the VLR.

NOTE: For simplicity, this example does not show any other procedures that the HLR-FE can possibly do

5. If the user is reachable, the VLR provides a MSRN in the response. This MSRN is not stored in the UDR, since it is temporarily reserved and consumed in the VLR,

6. The HLR-FE responds to GMSC with the provided roaming number. No user data is kept in the front-end after the procedure is ended.

A.1.3 IMS re-registration information flow example

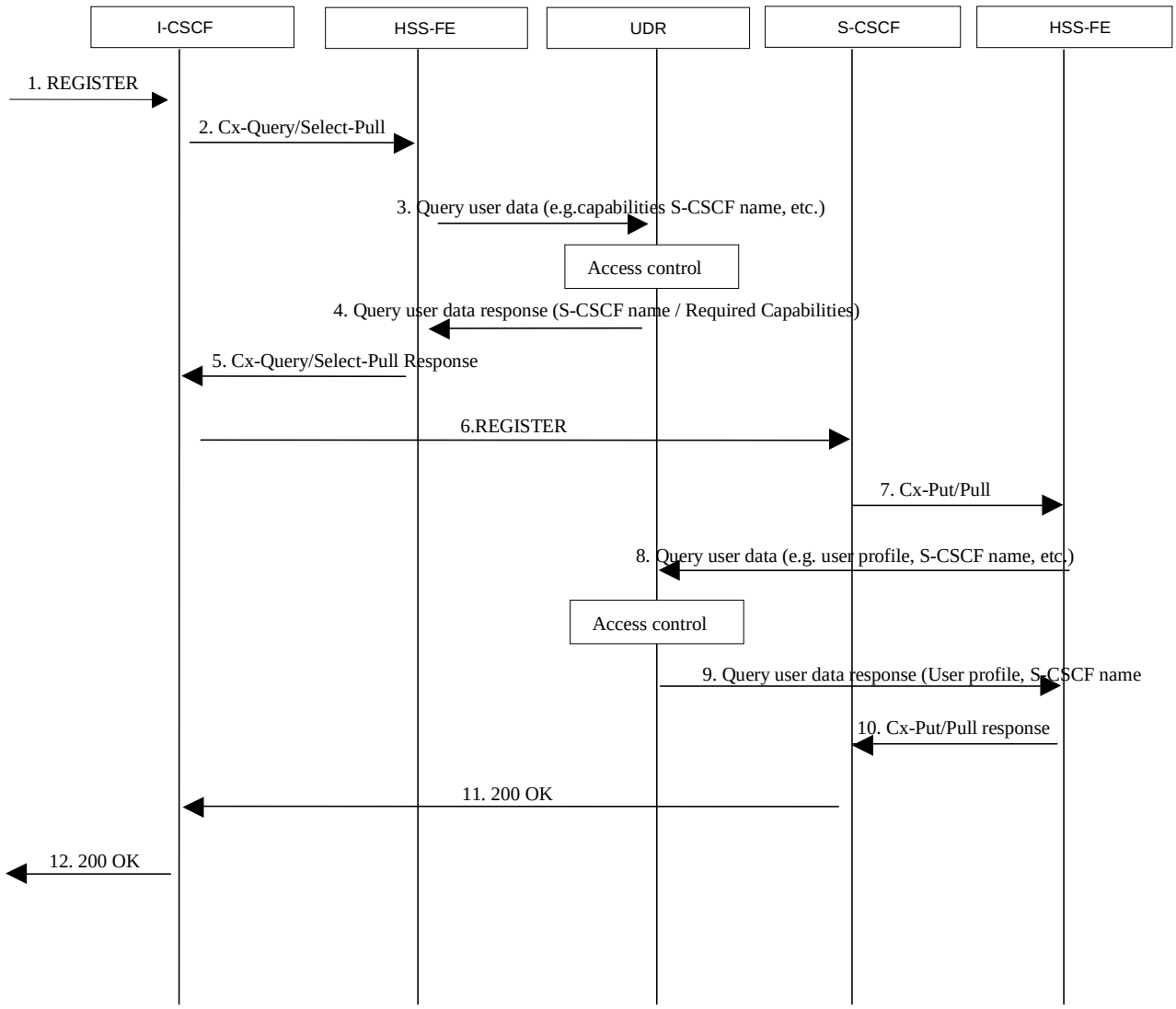

Figure A.1.3-1: IMS re-registration flow example with Ud Query from HSS-FE

1. I-CSCF receives an incoming REGISTER request.

2. The I-CSCF sends the Cx-Query/Cx-Select-Pull to the HSS with the Public User Identity.

3. Upon reception of Cx-Query, the application specific front-end (HSS-FE) fetches all the user data it needs to perform its application logic (e.g. capabilities associated to the user subscription, list of visited networks allowed, S-CSCF name, etc.) in the UDR through a Ud Query procedure.

4. After applying the proper access control (i.e. the front-end is allowed to read the type of data), the UDR responds with data requested (user location, required capabilities, etc.).
5. The HSS-FE includes the S-CSCF name in the response. No user data is kept in the HSS FE.
6. The I-CSCF sends the REGISTER request to the received S-CSCF.
7. The S-CSCF sends to HSS Cx-Put/Pull. This request, in this example, is received by a different HSS-FE.
8. Upon reception of Cx-Put/Pull, the HSS-FE fetches the user profile, S-CSCF name, etc. from the UDR through a Ud Query procedure .
9. After applying the proper access control, the UDR responds with data requested (e.g. user profile).
10. The HSS- FE detects that this is a re-registration, so it sends Cx-Put/Pull Resp to S-CSCF. No user data is kept in the front-end after the procedure is ended.
11. The S-CSCF returns 200 OK to I-CSCF
12. The I-CSCF forwards the response to P-CSCF.

A.2 Information flows with Updating data procedure over Ud

A.2.1 General

When user data (temporary or permanent) has to be modified, the application front-end shall perform an update procedure towards the UDR.

The following flows show an example of a location update in CS network and an example of a service data update in an IMS network. These scenarios do not address the mechanisms for access authorisation in the UDR. These scenarios only address the specific actions of traffic events that are currently in effect.

A.2.2 CS location update information flow example

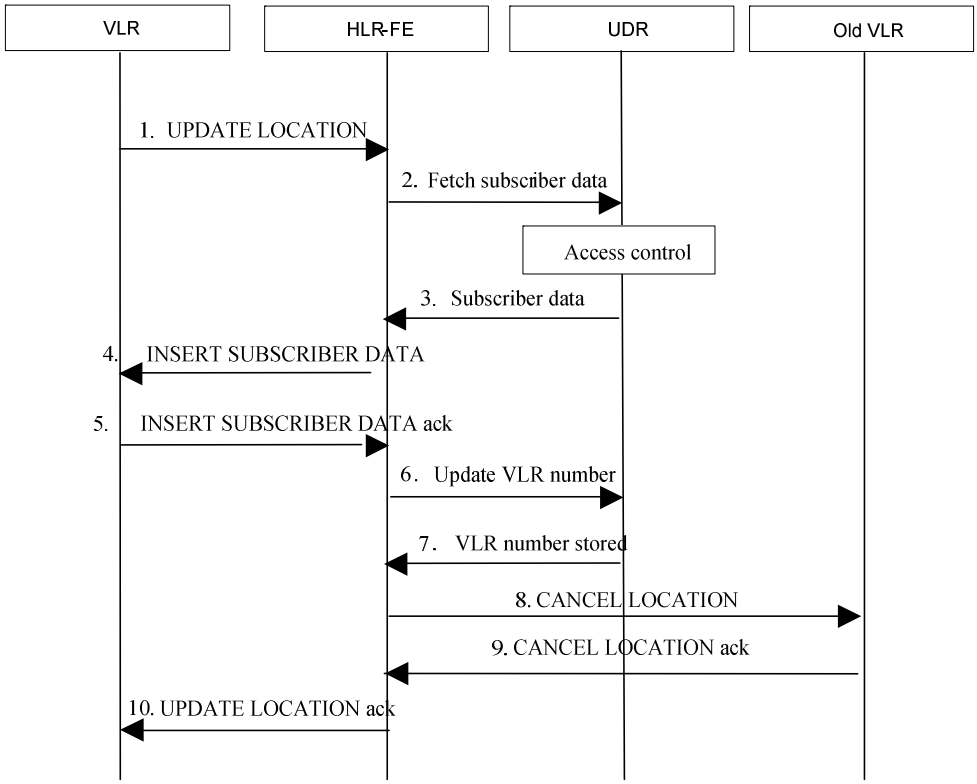

Figure A.2.2-1: CS location update with Ud Update from HLR-FE

1. The VLR initiates MAP Update Location message towards HLR.
2. Upon reception of UL, the application specific front-end (HLR-FE) fetches all the user data it needs to perform its application logic (e.g. barrings, VLR number, etc.) from the UDR through a Ud query procedure.
3. After applying the proper access control (i.e. the front-end is allowed to read that type of data), the UDR responds with the user data requested.
4. If the request is allowed (e.g. no barring is to be applied), the HLR-FE sends MAP Insert Subscriber Data to provide the user data to VLR.
5. The VLR acknowledges the request. HLR-FE receives the information about services supported by the new VLR. This can lead to certain additional modifications on user data.
6. If the previous and new VLR are not the same, the HLR-FE updates the new user data (e.g. VLR number) in the UDR through a Ud Update procedure.

NOTE: Step 6 can be done immediately after step 3

7. If the front-end is allowed to modify the type of data, the UDR updates the new VLR number.
8. The HLR-FE sends MAP CANCEL LOCATION to cancel the location in the old VLR.
9. The old VLR acknowledges the request and remove all data related.
10. The HLR-FE informs the new VLR about the result of the Update Location procedure. No user data is kept in the front-end after the procedure is ended.

NOTE: In this example, by updating the VLR-Number in the UDR after receiving MAP-Update-Location, the HLR-FE implicitly activates a pre-configured subscription to notification of relevant Subscriber Data changes. By deleting the VLR-Number in the UDR after receiving MAP-PurgeMS, a HLR-FE implicitly deactivates the pre-configured subscription to notification. The presence of a VLR-Number in the UDR is equivalent to a HLR-FE subscription over Ud for sending Notification Messages at modification of any relevant Subscriber Data.

A.2.3 IMS service data change information flow example

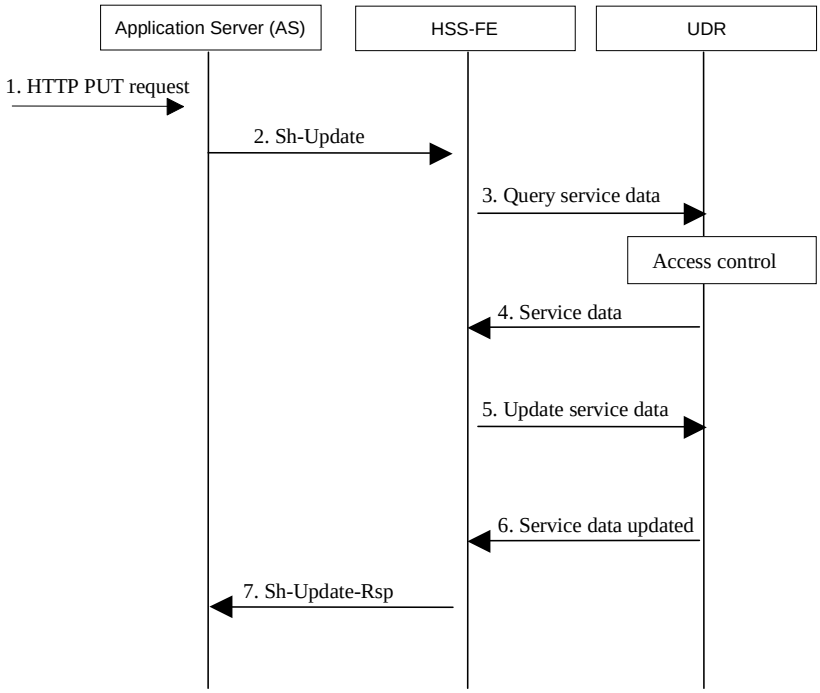

Figure A.2.3-1: IMS service data change information flow example with Ud Update from HSS-FE

1. UE initiates a service data configuration update (e.g. activates call waiting) via Ut interface towards the AS.
2. AS performs Sh-update to store the new service data (i.e. transparent data) in the HSS.
3. Upon reception of Sh-Update, the application specific front-end (HSS-FE) checks the AS permission list. If AS is allowed, the HSS-FE fetches the service data for the user from the UDR through a Ud Query procedure. These data include the sequence number to synchronize the writing of data among different ASs.
4. If the front-end is allowed to read the type of data, the UDR responds with service data.

NOTE 1: At this step the HSS-FE can also receive the subscription information stored for data requested (i.e. list of ASs subscribed to transparent data changes)

5. If sequence number is correct, the application specific front-end (HSS-FE) updates the service data for the user in the UDR through a Ud Update procedure. The update data request message contains an update condition to ensure that the data is to be updated if the sequence number was not updated by other FEs.
6. If the front-end is allowed to update the type of data and the update condition is satisfied, the UDR stores the new service data configuration and the new sequence number

NOTE 2: If the HSS-FE knows from step 4 that no other AS has subscribed to be notified when transparent data change, the HSS-FE does not send Sh-PNR to any other AS. If no other AS subscribed to transparent data changes and no other notifications are required (based on local configuration policy in the UDR), no Ud Notify is sent.

NOTE 3: It is assumed that the UDR has not allowed to write the transparent data and sequence number by a HSS-FE during steps 3-6.

7. The HSS-FE sends the response to Sh-Update to AS. No user data is kept in the front-end after the procedure is ended.

A.3 Example Information flows for subscriptions to notifications

A.3.1 General

When an UDC-external entity subscribes/un-subscribes to notifications of specific events which occurs on specific user data stored in the UDR, the application front-end should perform a subscription procedure towards the UDR.

The following flows show examples of an application server subscription to notification and un-subscription to notification in IMS network. These scenarios do not address the mechanisms for access authorisation in the UDR.

A.3.2 Application Server Subscription information flow example

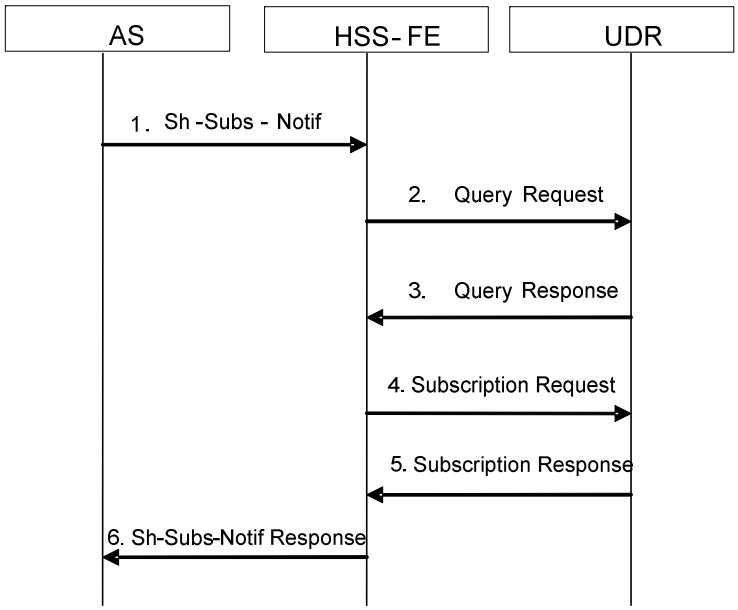

Figure A.3.2-1 Subscription over Sh with Ud subscription from HSS-FE

1. AS sends the Sh-Subs-Notif request to the HSS-FE to subscribe or un-subscribe to the notifications of changes on the Repository Data with the related Public User Identity, the Service Indication, the Application Server Identity, the Expiry Time (not for un-subscribing) and other information.

2. Upon reception of the Sh-Subs-Notif request, the HSS-FE checks the AS-permission list to see whether the AS is allowed to contact the HSS. If so, the HSS-FE sends the Query Request to the UDR to fetch the user data needed to process the Sh-Subs-Notify request from the UDR.

3. After applying the access control, the UDR responds to the HSS-FE with the requested data.

4. The FE sends a Subscription Request to the UDR.

The Subscription Request message includes:

- The Application type of the FE

- Application FE identity
- Public User Identity
- Subscription Type: indicates whether this request is to subscribe or to un-subscribe.
- Notification Type
- the following Subscription-to-Notifications information:
  - Identification of the requested data: Repository Data with Service Indication.
  - The notification condition: change on user data.
  - The original entity identity: Application Server Identity which issues the Sh-Subs-Notif request.
  - The expiry time: the Expiry Time in the Sh-Subs-Notif Request, if any (possibly modified by the HSS-FE); expiry time is not applicable for un-subscriptions.

5. The UDR checks whether the subscription/un-subscription is allowed. If it is permitted, the UDR shall store/delete the Subscription-to-Notification information. The UDR responds with the Success Indication.
6. Upon reception the Subscription Response, the HSS-FE sends the Sh-Subs-Notif Resp to the AS with the value of the requested data and with the result code set to DIAMETER_SUCCESS.

A.4 Information flows with notification procedure over Ud

A.4.1 General

When user data (temporary or permanent) has been modified, the application front-end may require a notification.

When an UDC-external entity modifies subscribable data in the UDR via an application FE, this application FE shall send notifications to other UDC-external entities on supported UDC-external interfaces as part of its application logic (if so required). Notifications to other UDC-external entities to which this FE is not connected, or to other AS which subscribed to notifications, require notification on the Ud-Interface.

The following flows show examples of IMS capability change for a user, an application server subscription to notification and un-subscription to notification in IMS network. These scenarios do not address the mechanisms for access authorisation in the UDR. These scenarios only address the specific actions of traffic events that are currently in effect.

A.4.2 IMS user capability change with notification information flow example

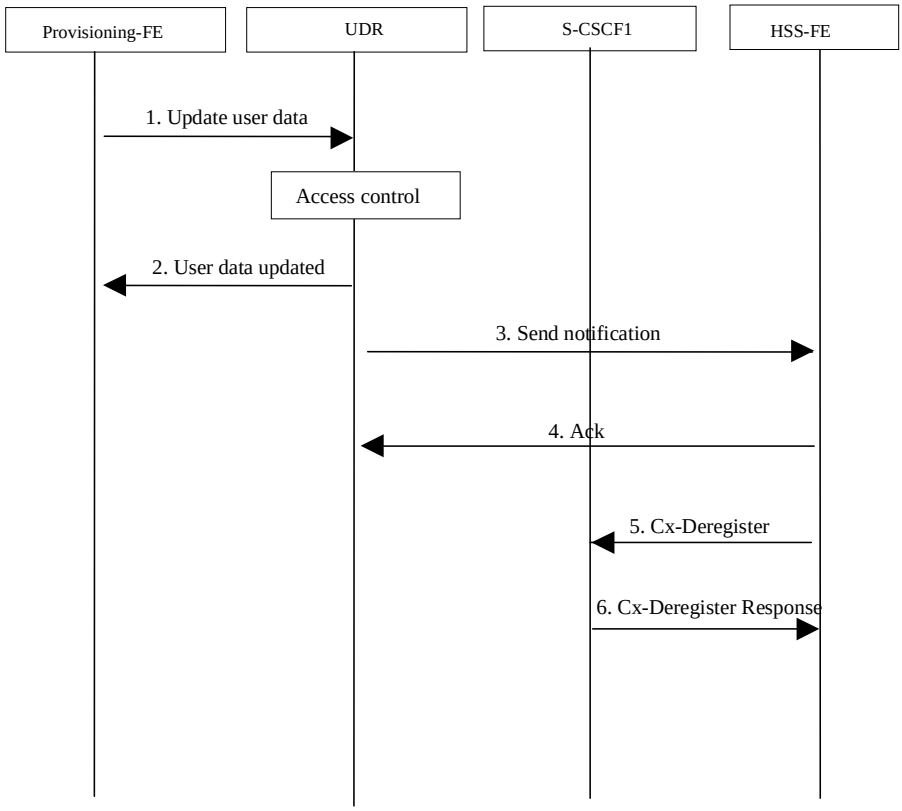

Figure A.4.2-1: IMS service data change information flow example with Ud notification towards HSS-FE

1. The operator changes the capabilities to move the user to a new S-CSCF, the user is administratively de-registered and S-CSCF is cleared (via Provisioning FE).

2. After applying the proper access control (i.e. the front-end is allowed to write that type of data), the UDR updates the requested data (e.g. registration status, user capabilities, etc.)

3. The UDR checks if a FE is required to be notified upon these modified data. If so, it selects a FE supporting the HSS application and sends a notification which contains the user data needed by the FE to perform its application logic (old/new user status, S-CSCF name, etc.)

NOTE 1: The conditions to trigger the notification can be common for all users and can be based on local configuration policy in the UDR, e.g. a Provisioning FE modifies user data that is relevant for another application type (HSS) and other related user data is present (S-CSCF name).

NOTE 2: The FE selection mechanism performed by the UDR (e.g. FE cluster identity if it was stored at user"s registration, application supported, load balancing, etc.) is out of scope of this specification.

4. HSS-FE acknowledges the notification.

5. The user is de-registered from S-CSCF1 by HSS-FE.

6. S-CSCF1 acknowledges the Cx command.

A.4.3 Application Server Notification information flow example without Ud-Notify

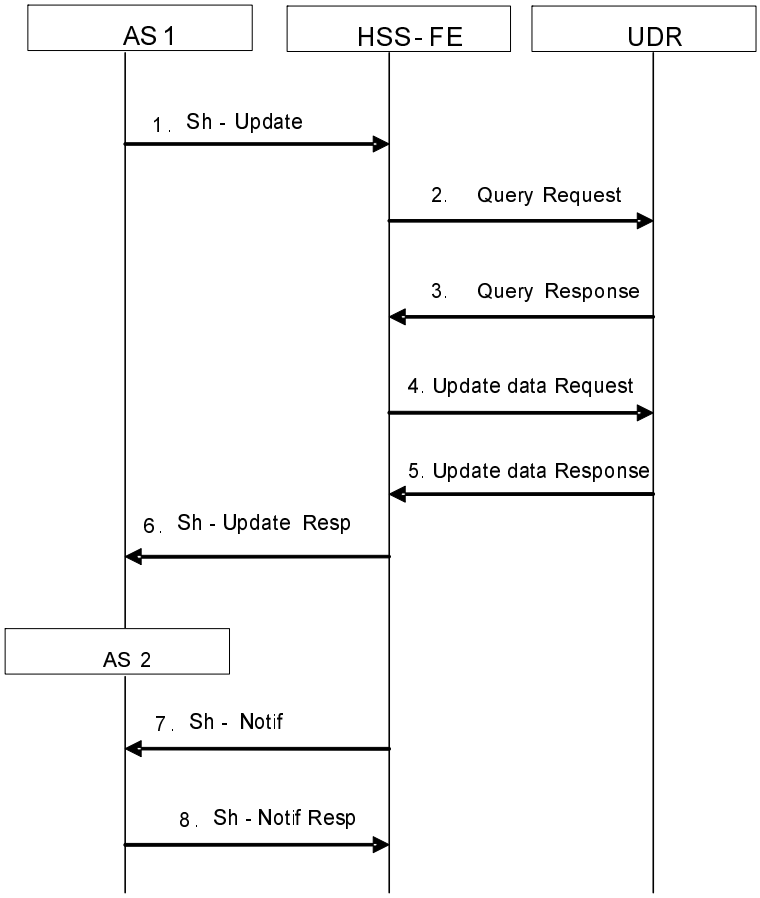

Figure A.4.3-1 Notification over Sh without Ud notification

1. AS1 sends the Sh-Update request to the HSS-FE to update the Repository Data with the related Public User Identity, the Service Indication, the Application Server Identity, and other information.

2. Upon reception of the Sh-Update request, the HSS-FE checks the AS-permission list to see whether the AS is allowed to contact the HSS. If so the HSS-FE sends the Query Request to the UDR to fetch the user data needed to process the Sh-Update request from the UDR.

3. After applying the access control, the UDR responds to the HSS-FE with the requested data.

4. The HSS-FE sends an Update Data Request to the UDR.

5. The UDR checks whether the update is allowed. If it is permitted, the UDR shall update the repository data. The UDR responds with the Success Indication. Since the update data request was sent by an FE which can contact AS2 and there is no other AS which subscribed to changes in repository data, the UDR does not send Ud-notify.

NOTE: The UDR knows that the FE can contact AS2 based on local configuration policy

6. Upon reception the Update Data Response, the HSS-FE sends the Sh-Update Resp to the AS with the result code set to DIAMETER_SUCCESS.

7. The HSS-FE checks data received in step 3 to see whether an AS has subscribed to notifications (i.e. whether a HSS-FE has previously added/stored an AS in the Application Server Identity List for the Repository Data of the user upon receiving a previous Sh-Subs-Notif). If so the HSS-FE checks the notification conditions. In this example the conditions are met (i.e. AS expiry time stored in the Application Server Identity list has not elapsed) and the HSS-FE sends Sh-Notif to AS2.

8. AS2 sends Sh-Notif Resp to the HSS FE.

A.4.4 Application Server Notification information flow example with Ud-Notify

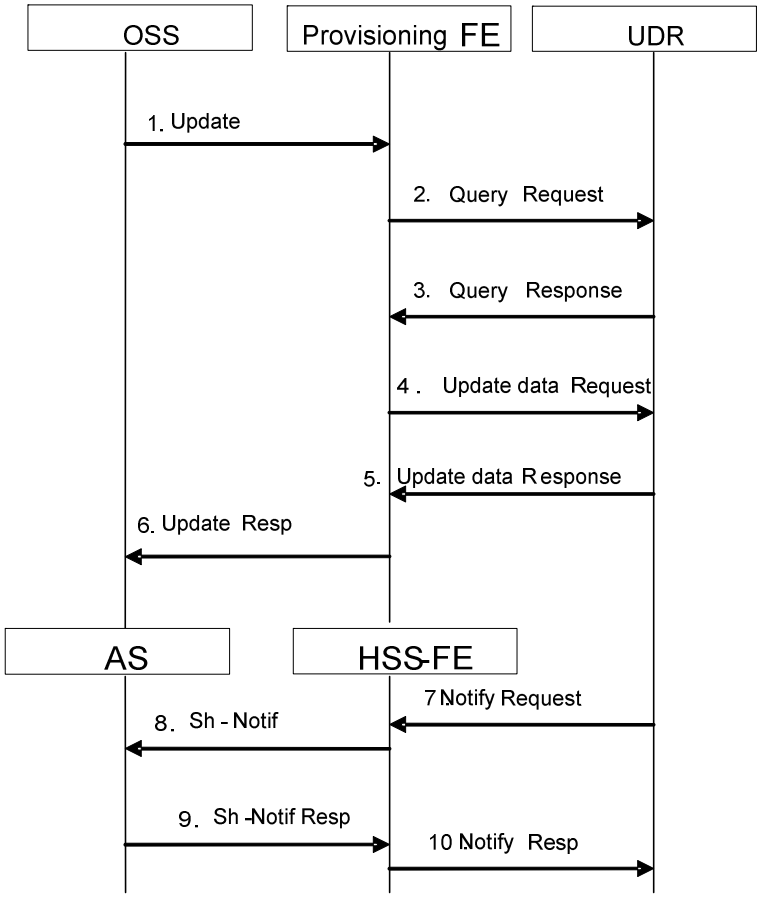

Figure A.4.4-1 Notification over Sh with Ud notification

1. OSS sends the Update request to the Provisioning-FE to update the Charging Information with the related Public User Identity.
2. Upon reception of the Update request, the Provisioning-FE sends the Query Request to the UDR to fetch user data from the UDR.
3. After applying the access control, the UDR responds to the Provisioning-FE with the requested data.
4. The Provisioning-FE sends Update Data Request to the UDR to update the Charging Information.
5. The UDR checks whether the update is allowed. If it is permitted, the UDR shall update the charging information. The UDR responds with the Success Indication.
6. Upon reception the Update Data Response, the Provisioning-FE sends the Update Resp to the OSS.
7. Since an AS has subscribed to notification of update of Charging Information (i.e. a HSS-FE has previously added/stored an AS in the Application Server Identity List for the Charging Information of the user upon receiving Sh-Subs-Notif) and the update data request was sent by an non HSS-FE (that is, a FE which cannot contact the AS), the UDR selects an HSS-FE and sends Notify Request to the selected FE.

   The Notify Request includes the following items:

   - Public User Identity.
   - Identification of the requested data: Charging Information.

- Original data: Original value of the Charging Information before update.
- Updated data: updated value of the Charging Information.
- Additional data: the additional information needed by the FE to perform the Notification procedure in addition to the requested data.
- The original entity identity: the identity of the Application Server which issues the Sh-Subs-Notif request.

NOTE: In this example, it is assumed that a pre-configured subscription exists in the UDR and that a modification to the Charging Information is done by a non HSS-FE, which results in a notification to an HSS-FE.

8. The HSS-FE checks data received in step 7 to see whether an AS has subscribed to notifications on modification on charging information (i.e. whether a HSS-FE has previously added/stored an AS in the Application Server Identity List for the Charging Information of the user upon receiving a previous Sh-Subs-Notif). If so the HSS-FE checks the detailed notification conditions. In this example the conditions are met (i.e. AS expiry time stored in the Application Server Identity list has not elapsed) and the HSS-FE sends Sh-Notif to the AS.

9. AS sends Sh-Notif Resp to the HSS FE.

10. HSS-FE sends Notify Resp to the UDR.

A.4.5 Application Server Notification information flow example without Ud-Notify – Subscription expired

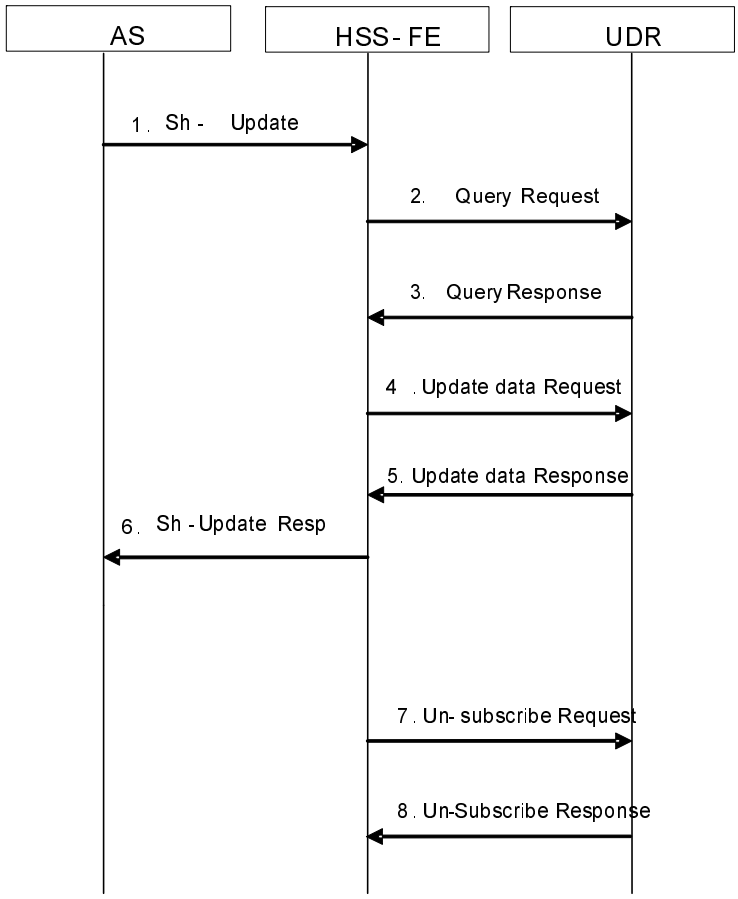

Figure A.4.5-1 Notification over Sh without Ud notification – Subscription expired

1-6. See A.4.3.

7. The HSS-FE checks data received in step 3 to see whether an AS has subscribed to notifications (i.e. whether a HSS-FE has previously added/stored an AS in the Application Server Identity List for the Repository Data of the user upon receiving a previous Sh-Subs-Notif). If so the HSS-FE checks the notification conditions. In this

example the conditions are not met (i.e. AS expiry time stored in the Application Server Identity list has elapsed). The HSS-FE un-subscribes the expired AS subscription in the UDR by removing the AS from Application Server Identity List for the Repository Data for the user.

8. UDR sends Un-Subscribe Resp to the HSS FE.

A.4.6 Application Server Notification information flow example with Ud-Notify – Subscription expired

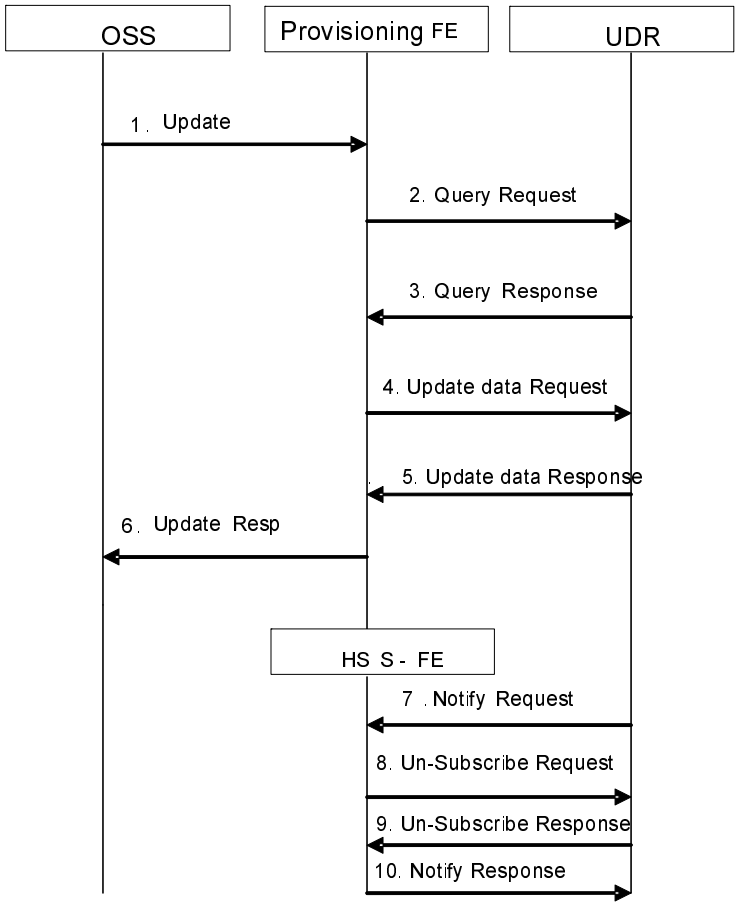

Figure A.4.6-1 Notification over Sh with Ud notification

1-7. See A.4.4

8. The HSS-FE checks data received in step 7 to see whether an AS has subscribed to notifications on modification on charging information (i.e. whether a HSS-FE has previously added/stored an AS in the Application Server Identity List for the Charging Information of the user upon receiving Sh-Subs-Notif). If so the HSS-FE checks the notification conditions. In this example the conditions are not met (i.e. AS expiry time stored in the Application Server Identity list has elapsed). The HSS-FE un-subscribes the expired AS subscription in the UDR by removing the AS from Application Server Identity List for the Charging Information for the user.

9. UDR sends Un-Subscribe Resp to the HSS FE.

10. HSS-FE sends Notify Resp to the UDR.

Annex B (informative): Applicability of the UDC concept to network nodes

B.1 Introduction

This informative annex outlines a step by step approach which aims to help clarifying whether a given network node is applicable to the UDC concept.

B.2 Basic Prerequisite

A fundamental concept of UDC is to separate a network node's user data from a network node's application logic.

It is understood that a network node's application logic is running in sessions, which last for a limited duration. During a session the application logic makes use of user data, i.e. user data cannot be separated from the application logic while a session is ongoing. Only when no session is ongoing user data can be separated from the application logic.

NOTE: The terms "session" and "application" in this context have special meaning which must not be mixed up with a user's communication session or service application.

Consequently, Network Nodes that (in non-UDC networks) do not store user data (while no session is ongoing), are not applicable to the UDC concept.

Furthermore, Network Nodes that (in non-UDC networks) do not perform application logic (i.e. pure data bases), are not applicable to the UDC concept.

B.3 Step 1 – Separating User Data from Application Logic

In principle, any network node (in non-UDC networks) that stores user data while no session is ongoing and that performs application logic is a candidate for separation of user data and application logic:

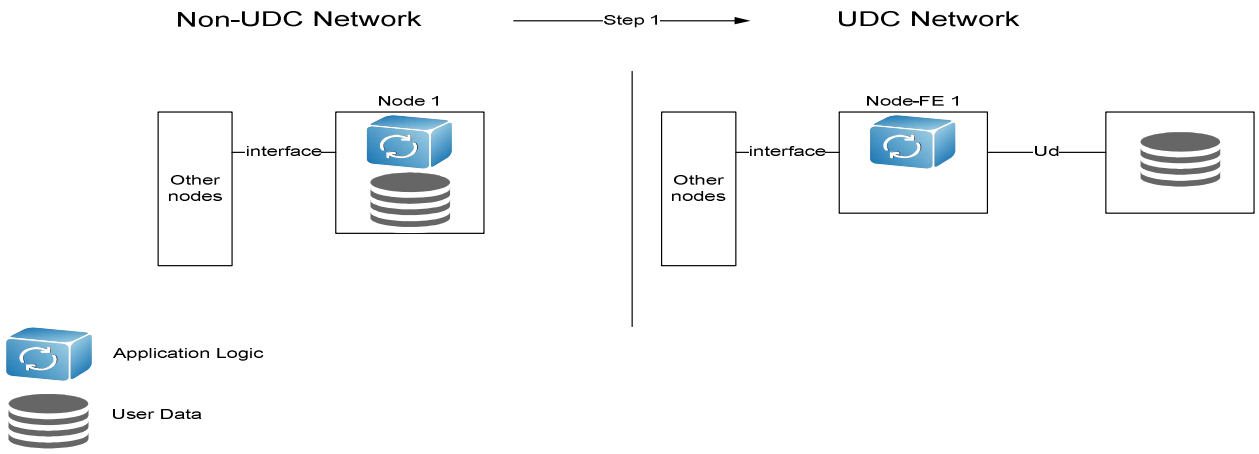

Figure B.3.1/1: Separating User Data from Application Logic

However, separating user data from application logic without any additional steps does not provide much benefit (if any).

B.4 Step 2 – Introducing Provisioning FEs

If part of the candidate node's user data is permanent data which is applicable to provisioning at that node, provisioning-application logic could also be separated from the node's application logic.

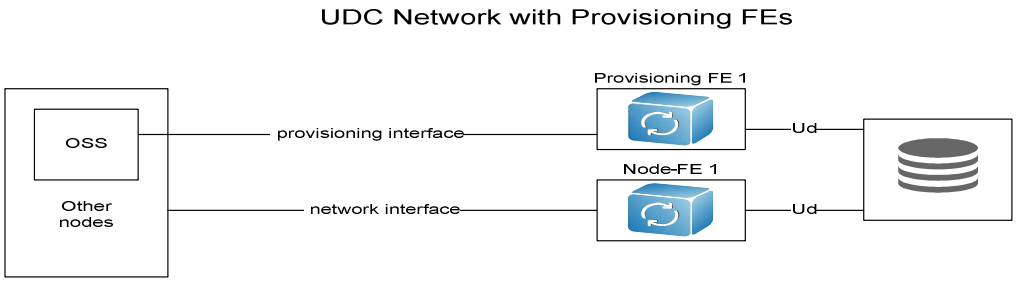

Figure B.4.1/1: Introducing Provisioning FEs

Again, this step alone does not provide much benefit.

B.5 Step 3 - Storing outsourced user data in a logically single UDR

This step is an essential part of the UDC concept. The outsourced user data are stored in a logically unique UDR:

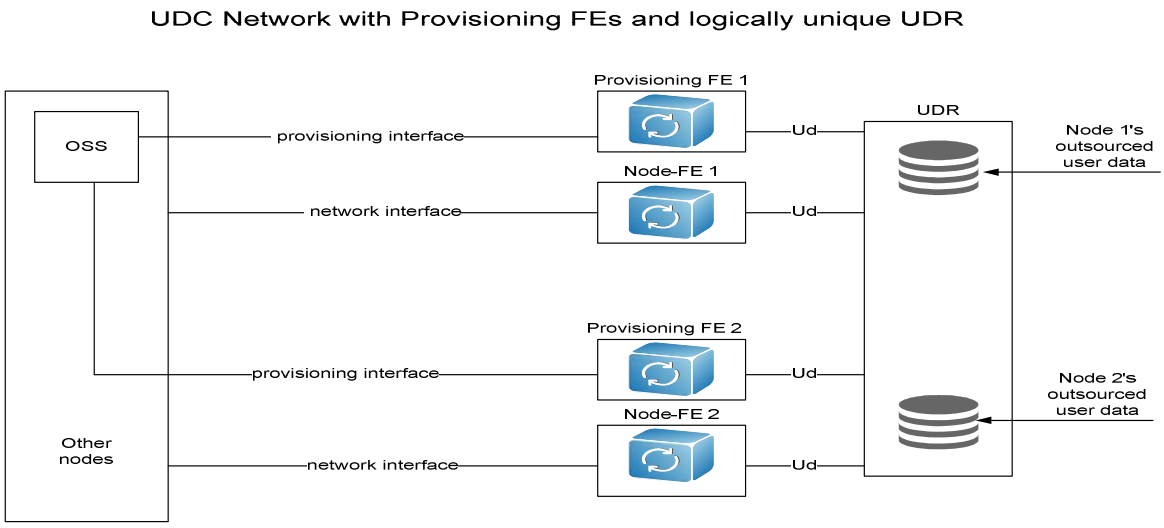

Figure B.5.1/1: Storing outsourced user data in a logically single UDR

One of the benefits of this step is that e.g. Provisioning FE 2 could not only access via Ud those user data which were separated from the application logic in Node-FE 2, but also those user data that were separated from the application logic in Node-FE 1.

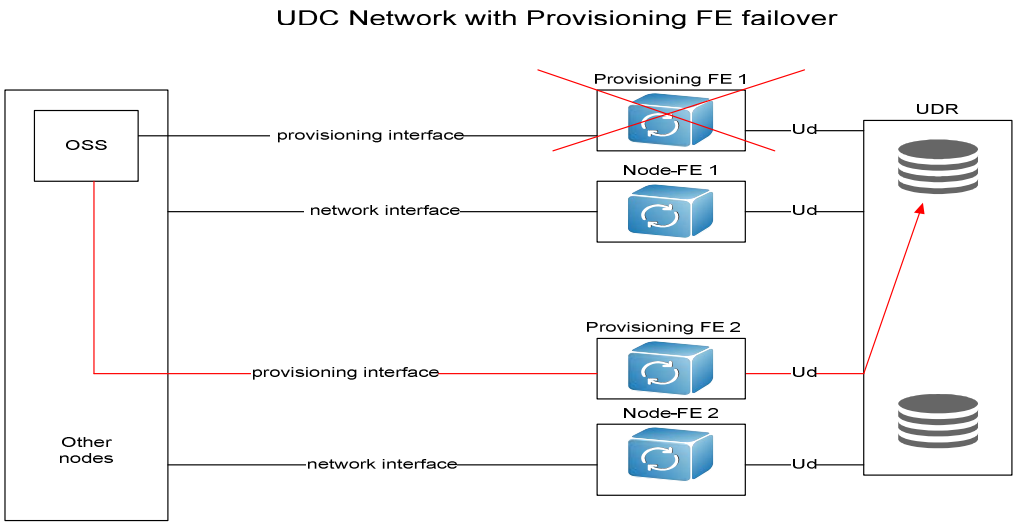

Figure B.5.1/2: Storing outsourced user data in a logically single UDR

B.6 Step 4 –Full Load Sharing and Failover functionality

What is said for the provisioning FEs in step 3, in theory also holds for other node-FEs: Node-FE 2 could access via Ud not only its own outsourced data, but also data sourced out by Node-FE 1. However, careful analysis is needed before deciding the cases in which "cross data accessibility" should be allowed. The main point to consider is whether or not parallel sessions in different FEs result in problems.

If routing mechanisms used on the network interfaces allow configuration of alternative routes to address e.g. load sharing, link loss or node outage, and parallel sessions in different FEs are not regarded a problem, FEs of the same application type could share load or could take over in cases of node-FE outage.

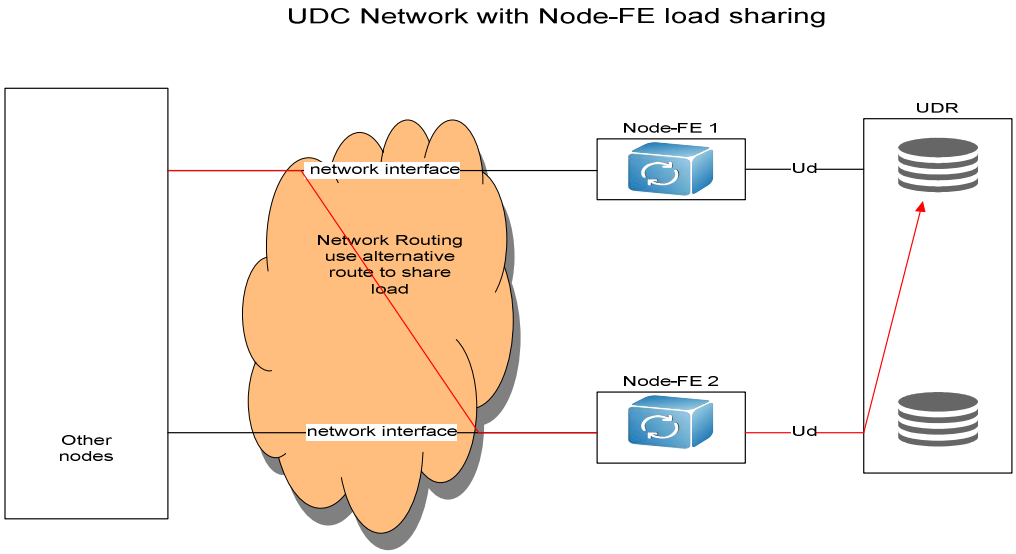

Figure B.6.1/1: Full Load Sharing and Failover functionality

This step does not impact FE or UDR implementation but is a routing configuration issue outside the UDC. Furthermore, this step – although it provides great advantages - is believed not to be essential to the UDC concept. Deployments may or may not take advantage of this step if applicable. Non-applicability of this step is not a knock-out criterion for the decision on applicability of the UDC concept for a given node.

B.7 Step 5 – Converging user data in the UDR

It is believed to be beneficial when the user data outsourced by several nodes are not stored separated within the UDR but are somehow converged:

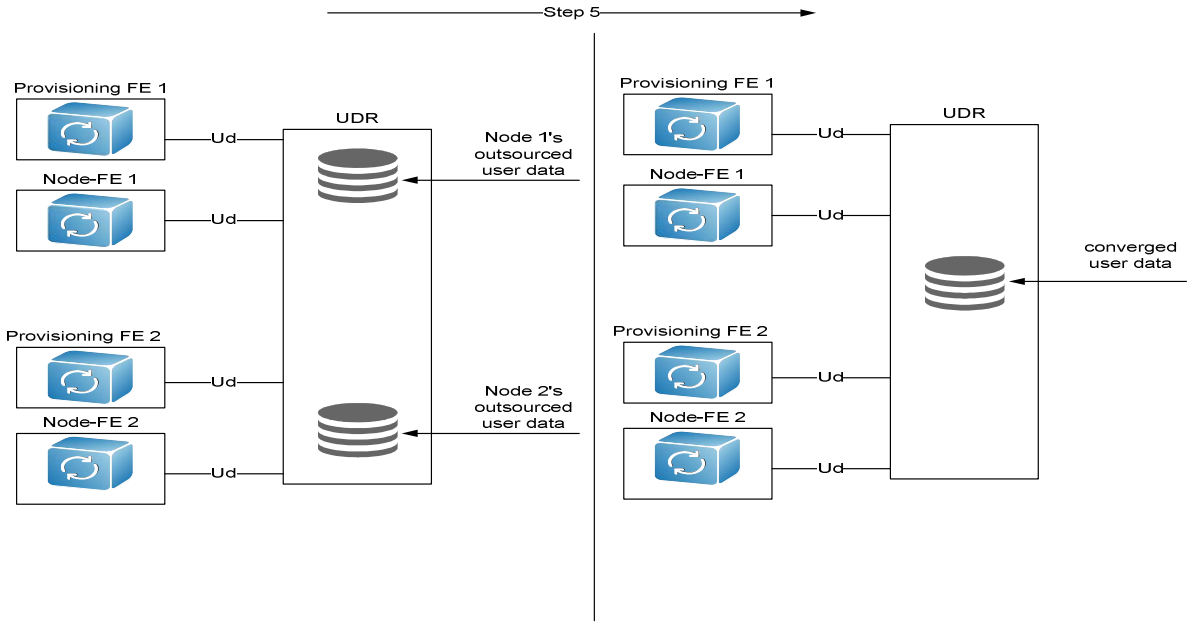

Figure B.7.1/1: Converging user data in the UDR

This step is purely UDR internal and therefore out of scope of CT4 standards, and not further looked at in this annex.

B.8 Example analysis for HLR

1. An HLR (in non-UDC Networks) stores user data (while no session is ongoing) and performs application logic, therefore step 1 is applicable.

2. An HLR (in non-UDC Networks) stores permanent user data provisioned at the HLR, therefore step 2 is applicable.

3. Outsourced HLR user data may be stored in a common UDR to allow access from any Provisioning-FE connected to the UDR. Step 3 is applicable.

4. Parallel sessions in two different HLR-FEs: It is believed that e.g. one HLR-FE could process a MAP-SendRoutingInfoForSM message while another HLR-FE processes a MAP-AnyTimeInterrogation message for the same user at the same time. This in general holds for all HLR-FE session initiating messages. Furthermore the duration of an HLR-FE session in general is in the range of seconds or minutes rather than hours or days. The transaction mechanism supported on the Ud interface is suitable to address data synchronization.

Step 4 is applicable.

In summary all steps are applicable and provide the full benefits of the UDC concept.

B.9 Example analysis for S-CSCF

1. An S-CSCF (in non-UDC Networks) stores (or may store) user data (while no session is ongoing i.e. no SCSCF-application logic is running) and performs application logic, therefore step 1 is applicable.

2. An S-CSCF (in non-UDC Networks) does not store permanent user data provisioned at the S-CSCF, therefore step 2 is not applicable.

3. Outsourced S-CSCF user data may be stored in a common UDR, but access from Provisioning FEs is not applicable. Step 3 is applicable, but does not provide any benefit.

4. Parallel sessions in two different S-CSCF-FEs may be regarded a challenge.

Step 4 is not applicable.

In summary the applicable steps (1 and 3) do not provide any benefit.

B.10 Example analysis for PCRF and SPR

1. A PCRF (in non-UDC Networks) does not store user data (while no session is ongoing); on the other hand an SPR does not perform application logic. It seems that the concept of separating application logic from user data (see step 1) is already in place.

2. Also step 2 (direct provisioning of user data in the SPR rather than via PCRF) seems to be already in place.

3. Also step 3 (combining several SPRs to a logically unique database) is applicable.

4. Parallel sessions in two different PCRFs needs further study.

Step 4 may not applicable.

In summary steps 1, 2, and 3 are applicable and provide some benefits. Applicability of step 4 may provide further benefits but it is for further study.

B.11 Summary

Any network entity that performs application logic (in sessions which are limited in duration) and that stores user data (during the time where no session is running) is in principle applicable to the UDC concept. However some nodes may not fully benefit from the UDC concept, especially if

- permanent user data stored at the node - in a non UDC network - is not provisioned at that node
- parallel sessions for the same user at the same time cannot run independently

Annex C (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2010-03	CP#47	CP-100051			Approved in CT#47	2.0.0	9.0.0
2010-06	CP#48	CP-100282	0001	1	Removal of normative behaviour of the HLR-FE during MAP Update Location flow	9.0.0	9.1.0
			0002	1	Removal of security editor"s note		
			0005		FE data view		
2010-09	CP#48	CP-100459	0004	3	UDC Reference Architecture	9.1.0	9.2.0
2010-12	CP#50	CP-100680	0006		Provisioning FE correction	9.2.0	9.3.0
			0007	1	Notifications triggered by a FE to its cluster		
			0008	1	Provisioning from self care systems		
2011-03	CP#51	CP-110074	0013	2	Notifications and transactions	9.3.0	10.0.0
2012-03	CP#57	CP-120469	0014	1	Adding note on Data Model for User Data Convergence architecture	10.0.0	11.0.0
2014-09	-	-	-	-	Update to Rel-12 version (MCC)	11.0.0	12.0.0

History

Document history		
V12.0.0	October 2014	Publication