Draft EG 201 076 V1.1.1 (1997-06)

ETSI Guide

Network Aspects (NA); Universal Personal Telecommunication (UPT); Phase 2: Architecture and functionalities for interworking

Reference
DEG/NA-064004 (9yo00icq.PDF)

Keywords
Architecture, IN, interworking, UPT

ETSI Secretariat

Postal address
F-06921 Sophia Antipolis Cedex - FRANCE

Office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - 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

X.400
c= fr; a=atlas; p=etsi; s=secretariat

Internet
secretariat@etsi.fr
http://www.etsi.fr

Contents

Intellectual Property Rights

ETSI has not been informed of the existence of any Intellectual Property Right (IPR) which could be, or could become essential to the present document. However, pursuant to the ETSI Interim IPR Policy, no investigation, including IPR searches, has been carried out. No guarantee can be given as to the existence of any IPRs which are, or may be, or may become, essential to the present document.

Foreword

This ETSI Guide (EG) has been produced by ETSI Technical Committee Network Aspects (NA).

1 Scope

The present document defines the functional architecture and network interconnections for Universal Personal Telecommunication (UPT) Phase 2, based on CS2. It gives a high-level description of network functions required to support UPT, categorizing them into several classes. It describes the architecture and interconnection scenarios and allocates network functions to the functional entities of the architecture. The requirements of UPT on the functional entities of the Intelligent Network (IN) model are assessed. The UPT architecture is then applied to several example cases of call handling and service management. Evolution from the Phase 1 architecture to the Phase 2 architecture is taken into account.

2 References

References may be made to:

a) specific versions of publications (identified by date of publication, edition number, version number, etc.), in which case, subsequent revisions to the referenced document do not apply; or

b) all versions up to and including the identified version (identified by "up to and including" before the version identity); or

c) all versions subsequent to and including the identified version (identified by "onwards" following the version identity); or

d) publications without mention of a specific version, in which case the latest version applies.

A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

2.1 Normative reference

[1] ETS 300 790 (1996): "Universal Personal Telecommunication (UPT); Specification of the security architecture for UPT Phase 2".

2.2 Informative references

[2] TC-TR 015 (1993): "Universal Personal Telecommunication (UPT); Phase 1: network functionalities for charging, billing and accounting".

[3] TR 121: "Universal Personal Telecommunication (UPT); Architecture and functionalities for interworking".

[4] DEG/NA-064003: "UPT Phase 2; Requirements on Information flows and Protocols".

[5] ETR 055-02: "Universal Personal Telecommunication (UPT); The service concept; Part 2: General service description".

[6] ETR 067: "Universal Personal Telecommunication (UPT); Network considerations and requirements on dialling, routeing and numbering".

3 Definition and abbreviations

3.1 Definition

For the purposes of the present document, the following definition applies:

CS2: Capability Set number 2

3.2 Abbreviations

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

AP	Access Point
API	Access Point Identity
CCAF	Call Control Agent Function
CCF	Call Control Function
CCS7	Call Control for Signalling system number 7
CL	Calling Line
CS	Capability Set
CUSF	Call Unrelated Service Function
DSS1	Digital Subscriber Signalling System number 1
DTMF	Dual Tone Multi Frequency
GSM	Global System for Mobile communications
IN	Intelligent Network
INAP	Intelligent Network Application Part
ISDN	Integrated Services Digital Network
O&M	Operations and Maintenance
PSTN	Public Switched Telephone Network
PUI	Personal User Identity
RA	Routing Address
SCEF	Service Creation Environment Function
SCF	Service Control Function
SCUAF	Service Control User Agent Function
SDF	Service Data Function
SI	Service Identity
SLP	Service Logic Program
SMAF	Service Management Agent Function
SMF	Service Management Function
SRF	Specialized Resource Function
SSF	Service Switching Function
SSP	Service Switching Point
UPT	Universal Personal Telecommunication
UPTN	UPT Number

4 UPT network functions

4.1 Introduction

This subclause attempts to identify UPT specific network functions, that is, network functions required to support UPT which are not provided by current networks. These functions have been grouped into a number of classes, based on the aspect of UPT they are related to:

- functions related to the Access Point;
- functions related to the Personal User Identity (PUI);
- functions related to service provider agreement;
- functions related to authentication;
- functions related to UPT call control;
- functions related to registration/deregistration;
- functions related to service profile management;

- functions related to services;
- functions related to charging;
- O&M functions;
- functions related to transfer of data;
- functions related to user interaction.

Each of these classes is described in a separate subclause. Within a given class the functions are individually described, and, for most of them, an example of a situation where it may be used is given, in order to illustrate the general description. It is to be noted that the level of detail may vary greatly from one description to the other: This fact simply reflects that some functions are easier to detail at this stage, while others will require further study before they can be more precisely described.

The definition of network functions required for UPT relies on some important facts about UPT, which are briefly summarized below:

- a unique UPT Number (UPTN) is assigned to every UPT user, who may render it public. This is the number dialled by other users when they wish to call the UPT user. It can also be used by network entities when they need to send a request to the UPT user's home Service Data Function (SDF);
- every UPT user is also identified by a unique Personal User Identity (PUI), for security and management purposes. This identity is not public. If no UPT device is used, this identity is known to the UPT user, who will have to supply it every time he accesses the UPT service. If a UPT device is used, it may store the PUI, in which case the UPT user does not have to memorize it;
- there is a one-to-one relationship between the UPTN and the PUI of a given user;
- a UPT user can register for specific services on specific Access Points. These Access Points are identified by Access Point Identities (APIs). The correspondence between Service Identity (SI) and API may vary according to the time of the day and other parameters;

  it should be noted that a Routing Address can be derived from an API;
- the characteristics of an Access Point have an impact on the services which may supplied on it. It might therefore be useful to store the characteristics of Access Points (i.e. their terminal/network capabilities);
- some services may have specific requirements on the capabilities of the Access Point on which they are supplied. It may therefore also be useful to store the relationship between services and the capabilities required to supply them.

4.2 Class "Access Point"

The functions listed in this subclause are related to the Access Points used for access to the UPT service. These functions allow the input/output of an API from/to the user; they allow the verification of an API input from the user and the conversion of an API to a Routing Address; they allow the storage of an association between an API and a service and the storage of an association between an API and a set of capabilities.

- **Request API** function to request and input an API from the UPT user. For example, this function can be used during registration procedures (InCall Registration, OutCall Registration or AllCall Registration) to request the identity of the Access Point on which the user wishes to register.
- **Determine API** function to determine the API of the Access Point currently used.
- **Provide API** function to provide an API to the user. This function may be used by interrogation procedures to output to the user the identity of the Access Point(s) he has associated to a given service.
- **Verify API** function to verify an API specified explicitly by the user. This verification mainly asserts that the format of the API given by the user is acceptable from the point of view of the network where the verification takes place. It is not in general a complete verification, as the API may belong to an other network, in which case sufficient information may not be available to perform a complete verification.

- **Determine Routing Address** function to derive a Routing Address from an API.
- **Store API/Routing Address** function to store an API/Routing Address in connection with a PUI and a specific service. For example, this function may be used at the time when a service is activated by the user, to store the identity of the Access Point on which the service is to be provided.
- **Retrieve API/Routing Address** function to retrieve a stored API/Routing Address. This function may be used by service logic when the service is actually in action, to retrieve the identity of the Access Point on which the user requested the service to be provided. It may also be used by interrogation procedures if the user wishes to be reminded of the API he has associated with a given service.
- **Store Access Point capabilities** function to store relevant capability information on an Access Point of the visited network. This function may be used in conjunction with **Store API/Routing Address** to store additional information about the access which is to be used for a given service. This is subject to the ability of the network to supply such information, and is for further study.
- **Retrieve Access Point capabilities** function to retrieve stored capability information for an Access Point. This function may be used by service logic to find out the capabilities of the Access Point on which the service is to be supplied, before attempting to match them against the capabilities required by the service.
- **Store network capabilities** function to store capability information for the Access Points of a given network.
- **Retrieve network capabilities** function to retrieve capability information for the Access Points of a network. This function is used when the capabilities of a particular Access Point shall be determined. This happens when a UPT user registers to an Access Point: The UPT service needs to know about the capabilities of the Access Point to adapt the services it supplies.
- **Store network capabilities** function to store capability information for the Access Points of a given network. This function deals with the general capability in the given network. The Access Points that appear as particular cases (compared to this general capability) are treated with the two previous functions.

4.3 Class "PUI"

The functions described in this subclause are related to the PUI of a UPT user. They allow the retrieval of a user's PUI (from his UPTN or by querying it) as well as its verification. These functions are mostly used when access to the UPT service is requested by the user.

- **Determine PUI** function to derive a PUI from a UPTN. This function may, for example, be used when an incoming UPT call is to be handled by the UPT logic. The calling party dials the UPTN, whereas the service logic needs the user's PUI. This function provides the required mapping between the two.
- **Request PUI** function to request and input a PUI from the UPT user or from his UPT device. For example, all requests for UPT service from an unregistered terminal will start with this function, as the user cannot be identified until his PUI is known.
- **Verify PUI** function to check if a PUI is still valid, if it is blacklisted, etc. This function can be used immediately after the **Request PUI** function, to check whether the UPT user is allowed access to the service or not.

4.4 Class "Agreements"

The functions listed in this subclause are related to agreements between service providers and network operators. They allow the determination of a given UPT user's home service provider, of the originating network of an incoming call and the interrogation of agreements passed between operators and service providers.

- **Determine service provider identity** function to derive the identity of a user's home service provider from his UPTN or his Personal Identifier. This identity is required when checking for an agreement between the visited network operator (if having carrier's capabilities and UPT functionalities) or service operator and the home service provider (see next function) (see note).
- **Check service provider agreement** function to verify if there is an agreement between the UPT user's currently visited network operator or service operator (where this function is used) and his home service provider. This

function may be used when a UPT user attempts to place an outgoing call, to assess whether he should be allowed to do so or not (note).

- **Determine originating network/service operator identity** function to derive a network operator identity or service operator identity from a calling party address. This identity is required when checking for an agreement between the operator of the network (if having carrier's capabilities and UPT functionalities) where the request for UPT service originates and the corresponding UPT service provider (note).

- **Check network/service operator agreement** function to verify if there is an agreement between the UPT user's home service provider (which uses this function) and the network/service operator from which a request for UPT service has been received (note).

- **Determine supporting network identity** function to derive a supporting network identity from routing information. This identity is required when checking for an agreement between a service operator and a supporting network operator. This supporting network is the originating network (and a transit network if necessary) not having the UPT functionalities (note).

- **Check supporting network agreement** function to verify if there is an agreement between the service operator (which uses this function) and a supporting network operator.

The two functions **Determine service provider identity** and **Check service provider agreement** work in tandem, as do the two functions **Determine originating network/service operator identity** and **Check network/service operator agreement** and the functions **Determine supporting network identity** and **Check supporting network agreement**. The three couples of functions are dual from each other, the first one operating from the point of view of the visited network operator or service operator, the second one operating from the point of view of the home service provider and the third one operating from the point of view of the service operator.

- **Select supporting network operator** When a call shall be established, an incoming call or an outgoing call, the UPT user may have the possibility to choose which network operator will be in charge of doing it. The choice is enabled by this function.

- **Select visited service operator** As for the previous function, the UPT user may have the choice between several service operators to deal with his procedures. This choice is realized through this function (note).

NOTE: A new entity appears in the previous functions: the service operator. Indeed, in case the originating network would not have the IN functionalities, it would ask for the help of an assisting network in order to offer the UPT service. But, for UPT services, there is no need that this assisting network should own a complete network infrastructure (only UPT functionalities are then required): we could then call it service operator to make the difference with a network operator such as a "carrier".

4.5 Class "Authentication"

The functions described in this subclause are related to the authentication of the UPT user, when he is requesting service. They allow storage, retrieval, input and validation of authentication information.

- **Store authentication information** function to store authentication information related to a UPT user.

- **Retrieve authentication information** function to retrieve authentication information related to a UPT user.

- **Modify authentication information** function to modify authentication information related to a UPT user.

- **Request authentication information** function to request and input authentication information from the UPT user and/or his UPT device. For example, all requests for UPT service from an unregistered terminal will use this function together with the **Request Personal Identifier** function, as UPT service will not be provided to a user until he is both identified and authenticated. The general principles and methods used for authentication in UPT Phase 2 are specified in ETS 300 790 [1].

- **Authenticate** function to validate the authentication information supplied by the UPT user. This information can be obtained by means of the previous function. This function may need algorithms to be performed.

4.6 Class "Call control"

The functions listed in this subclause are related to UPT call handling. They allow the recognition, processing and routing of UPT calls. They have been divided into two categories: the functions dealing with incoming UPT calls and the functions dealing with outgoing UPT calls.

4.6.1 UPT incoming calls

- **Recognize UPT incoming call** function to recognize that a call is directed to a UPT user. This function shall be provided by low-level network call handling mechanisms. Its role is to trigger the following high-level function:
  - **Process UPT incoming call** function to provide service logic for handling a UPT incoming call: locate the UPT user, retrieve service profile information, check services to be supplied against the capabilities available, supply available services, etc. This function uses the functions related to services described in subclause 2.9. When it is done, it triggers the following low-level function:
    - **Route UPT incoming call** function to route a call to a UPT user. This function uses the Routing Address supplied by the **Process UPT incoming call** function to route the call to the UPT user. Another UPT specific aspect of this routing function is that it may add an indication that the call to be routed is a UPT call in the signalling messages it sends.
    - **Indicate UPT incoming call** function to indicate to the called UPT user that an incoming UPT call is present.
    - **Negotiate UPT incoming call** function to negotiate with the UPT user his acceptance of an incoming UPT call, and to handle the response. If the corresponding supplementary services are activated, this function may involve the presentation of the identity of the calling party, the authentication of the calling party and/or of the called party, the indication of charging rates for the call and the negotiation off the supplementary services to be provided for the call, if restrictions are imposed by the capabilities of the Access Point.

4.6.2 UPT outgoing calls

- **Recognize UPT outgoing call** function to recognize a UPT outgoing call request. This function may have to be provided by low-level network call handling mechanisms, in the case where a specific dialling prefix or number is defined for UPT outgoing calls. Its role is to trigger the following high-level function:
  - **Process UPT outgoing call** function to provide service logic for handling a UPT outgoing call: retrieve service profile information, check services to be supplied against the capabilities available, supply actual services, etc. This function uses the functions related to services described in subclause 2.9.

Note that there is no need for a function to route a UPT outgoing call, as the normal network mechanism is used.

4.7 Class "Registration"

The functions listed in this subclause are related to UPT registration and deregistration. They allow the recognition and handling of registration and deregistration requests.

- **Recognize registration/deregistration** function to recognize a UPT registration or deregistration. This function may have to be provided by low-level network call handling mechanisms, in the case where a specific dialling prefix is defined for UPT registration or deregistration. Its role is to trigger the following high-level function:
  - **Process registration/deregistration** function to provide service logic for handling UPT registrations and deregistrations: retrieve service profile information, perform housekeeping (time of day dependant routing, etc.), activate default registration upon deregistration, etc. This function uses the functions related to services described in subclause 2.9.
- **Indicate registration** function to indicate that a user has registered on a terminal for specific types of calls. It warns a third party that his terminal is currently used by a UPT user. It can also be used for outgoing UPT calls when an outcall registration already took place.

4.8 Class "Service profile"

The functions listed in this subclause are related to service profile management by the user. They allow the recognition and handling of service profile management requests, including functions to retrieve, modify and store the service profile.

- **Recognize service profile management** function to recognize a service profile management request from the user. This function may have to be provided by low-level network call handling mechanisms, in the case where a specific dialling prefix or number is defined for UPT service profile management. Its role is to trigger the following high-level function:
  - **Process service profile management** function to provide service logic for handling service profile management requests from the user: check the user access to his service profile, it retrieves service profile information, modifies, updates and stores it back. These requests are handled by the means of the five following functions:
    - **Store service profile** function to store the whole service profile (or a part of it) of a given UPT user.
    - **Check service profile user access** function to check service profile user access against network capabilities and user's authorization.
    - **Retrieve service profile** function to retrieve the whole service profile (or a part of it) of a given UPT user, according to storing possibilities.
    - **Edit service profile** function to interact with the user, for editing his service profile.
    - **Modify service profile** function to modify the whole service profile (or a part of it) of a given UPT user.
    - **Present service profile** function to interact with the user, for presenting his service profile.

4.9 Class "Service"

The functions listed in this subclause are related to service profile management by the user. They allow the recognition and handling of service profile management requests, including functions to retrieve, modify and store the service profile.

- **Recognize UPT request** function to recognize a request for UPT service. This function shall be provided by low-level network call handling mechanisms. Its role is to trigger the recognition functions (outgoing call, registration, service management, etc.).
- **Recognize UPT service** function to interact with the UPT user, to identify the type of service requested (outgoing call, registration, service management, etc.) and to run its service logic.
- **Retrieve service information** function to retrieve information on a specific service for a given UPT user. This function provides from the service profile the data corresponding to this given service.
- **Check service against profile** function to check the service requested against the corresponding information in the UPT user's service profile: if the user has not subscribed to the service, it cannot be provided.
- **Check service against capabilities** function to check the capabilities needed by the requested service against the capabilities of the user's current Access Point(s).
- **Service assisting request** function that enables a controlling SCF (SCFo) to establish a link with a supporting SCF (SCFh) and request assistance from the latter.

4.10 Class "Charging"

The functions listed in this subclause are related to the charging process for the UPT service. They allow the collection of charging information, the transfer of this information between operators and service providers, and the provision of charging information to the UPT user.

- **Charge** This function covers the whole charging process, which includes the recording of charging information during the call, and the storage of this information after the end of the call. This complex function may need to be split into simpler functions (for example: **Charge determination**, **Charge generation**, **Charge registration**), but this has no impact on UPT architecture. More information can be found in TCR-TR/NA-72305 [2].

- **Check credit against limits** function to check if the credit used by the user exceeds the amount authorized by the service provider. If the authorized credit has been exceeded, access to the UPT service is forbidden until some administrative action takes place. This credit checking may take place off-line, at regular time intervals, or in real-time, after each UPT call.

- **Transfer charging information** function to transfer collected charging information between network operators and service providers.

- **Provide charging information to user** function to indicate to the UPT user the cost of a call, either in real-time (i.e. during the call), or off-line (i.e. after the call is completed).

- **Determine charging rate** function to give the charging rate that is applicable at the current Access Point of the user and given the charging option he has activated. This function can also retrieve other charging information attached to a user (e.g. temporary non-roamed location).

- **Update credit limit** function used by the networks to update the credit still authorized for a given user.

- **Store charging information** function to store collected charging information concerning a given UPT user.

4.11 Class "O&M"

Functions related to Operations and Maintenance (O&M) aspects of the UPT service:

- Creation and management of reports.

- Evaluation of traffic and quality of service figures.

- Subscriber and network configuration handling.

- Traffic management flow control.

4.12 Class "Transfer of data"

The functions listed in this subclause are related to UPT registration and deregistration. They allow the transfer from the SDFh to a SDFo of data that could be useful, according to access restrictions.

- **Copy data** function to copy data from the SDFh to the current SDFo.

- **Update data** function to update the data transferred from the SDFh.

- **Erase the copied data** function to delete the transferred data when it becomes useless, in a current or a former SDFo.

4.13 Class "User Interaction"

The function listed in this subclause are related to the interactions between the network and the UPT users, thanks to the Specialized Resource Function (SRF). They enable the collection of the answers given to the queries asked by the network.

- **Send message** function to send a message to a UPT user. It may be an asking or the echo of an answer given by this user (in order to make sure of the accuracy of this answer).

- **Collect information** function to collect the answer of a UPT user after the use of the previous function.

5 UPT architecture and interworking

5.1 General UPT functional architecture

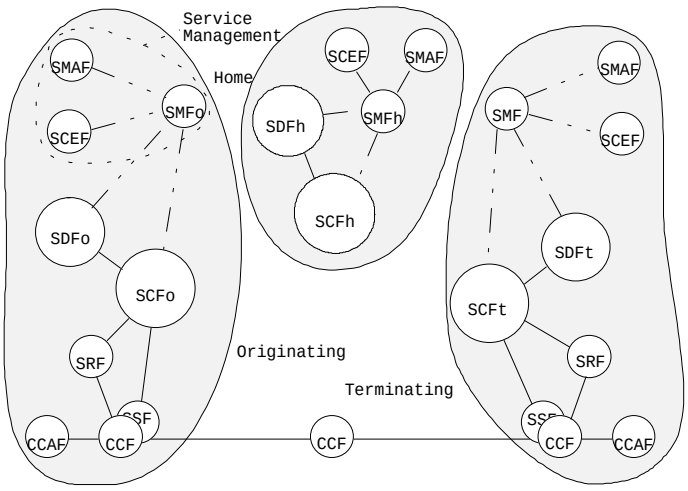


Figure 1: General UPT functional architecture

The general functional architecture for UPT, regardless of implementation phases, is described in TR 121 [3], where several scenarios are envisaged for interaction between UPT entities. Figure 1 is adapted from TR 121 [3] (interactions between UPT entities have been omitted for the sake of clarity) and gives an overview of the general UPT functional architecture. Apart from standard Intelligent Network (IN) terminology, the following notations are used in the figure above:

- **SCFh** Home SCF.
- **SDFh** Home SDF.
- **SMFh** Home SMF.
- **SCFo** Local ("visited") SCF, originating side.
- **SDFo** Local ("visited") SDF, originating side.
- **SMFo** Local ("visited") SMF, originating side.
- **SCFt** Local ("visited") SCF, terminating side.
- **SDFt** Local ("visited") SDF, terminating side.

The three interconnection scenarios described in TR 121 [3] are the following:

- Direct SDF - SDF dialogue.
- Direct access of SDFs from SCF.
- SCF - SCF dialogue.

All of them are considered in UPT Phase 2. The next subclause describes them in more detail.

5.2 Phase 2 UPT functional architectures

5.2.1 Interconnection functional architecture

Considering the interconnection requirements, CS2 enables three possibilities, described hereafter (see figure 2). Indeed, three functional interfaces exist in CS2 which are SCF-SDF, SCF-SCF and SDF-SDF. However, if the CS1 interface (SCF-SDF) was sufficient to offer a basic UPT service, for CS2 some enhancements are needed. For example, we may transfer data (service profiles) or request service logic assistance between networks.

Consequently, the general interconnection architecture is the addition of the three CS2 interfaces. Depending of the case, the UPT procedures will use one or several CS2 interfaces.

The management relationship may be used, for example, for sharing information off-line. The present document does not specify the management relationship.

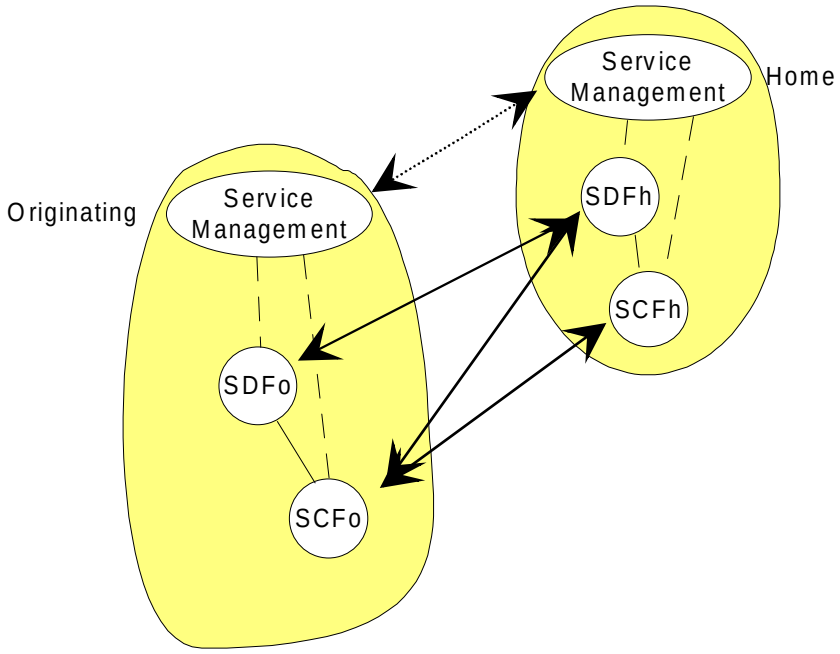


Figure 2: Interconnection functional architecture

NOTE: The use of a terminating network is not described hereafter. However, some procedures may involve such a network (e.g. secure answering).

5.2.1.1 Use of the SCF-SDF interface

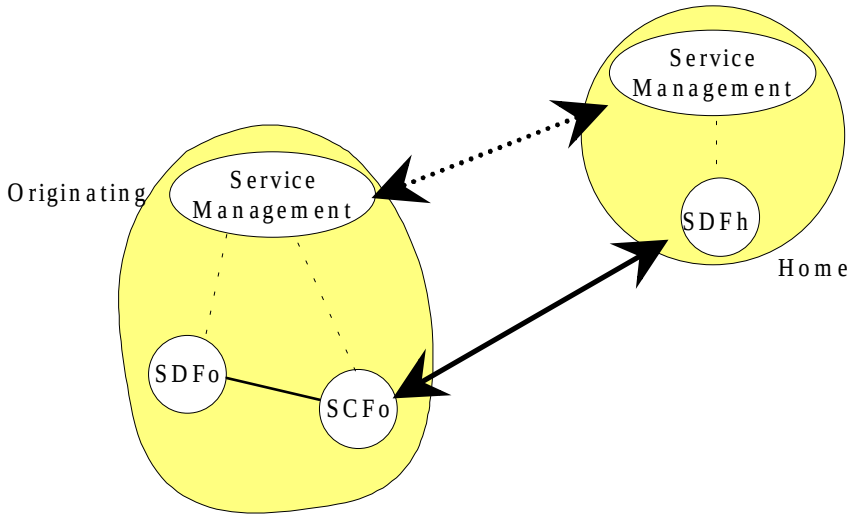
The functional architecture for UPT Phase 1 is described in figure 3. The differences with the general UPT functional architecture are:

- The interconnection of networks takes place between the SCFo and SDFh functional entities, as indicated by the arrow on the figure. The interface between SCF and SDF is specified in IN CS1.
- SDFh stores all data related to the UPT user (i.e. the database is centralized).
- SDFh shall as a consequence provide access control functions to check whether or not requests received from remote entities are authorized requests or not.
- SDFh performs the authentication of the UPT user.
- SDFo stores a list of agreements, which indicates the identity of all the service providers whose subscribers are allowed to access UPT service in SDFo's network.
- SDFo stores a list of service limitations resulting from agreements with service providers or network limitations.

- SDFo also stores information related to the management of the UPT service in its network, e.g. charging records which will be used later on for accounting.

- SDFo also stores service profile (or a part of it).

This architecture is derived from the standard architecture by assembling the architecture of the originating network with the data management part of the architecture of the home network (see figure 3). The terminating network is only mentioned for the sake of completeness on the figure, as no UPT-specific functions are attributed to it. The SCF-SDF interface is used to perform user authentication operations and to retrieve/update user profile information stored in the home network.



NOTE:
- The entity *Service Management* contains the entities dedicated to the management of services, that is to say Service Management Agent Function (SMAF), Service Management Function (SMF) and Service Creation Environment Function (SCEF), according to the IN architecture.
- Continuous links are service and call control links; dotted ones are management ones.

Figure 3: Use of the SCF-SDF interface

In this description, service control is always assumed by SCFo (i.e. by the originating network), while user profile management is always performed by SDFh (i.e. in the home network).

SCFo houses service logic for UPT, including call control logic, supplementary services logic, logic for interacting with the user through the SRF entity and logic for interacting with SDFh.

SDFo stores information on agreements with other operators, and call records created by SCFo. SDFo could also store some part of the service profile (with security restrictions) since there is the possibility to do some transfer of data. Data would be transferred from the SDFh to the SDFo thanks to the SCFo, after a SCFo request. Of course, it would be possible not to transfer the whole requested data: this would depend on agreements between operators, service providers and on the status of these data and would be decided by the SDFh.

SDFh stores subscriber and user profiles for all subscribers of the operator. It performs classic data storage and management functions, as well as access control over the SCF-SDF link (authentication of the originating network) and authentication of the UPT user, upon request from the originating network.

This description imposes a centralized approach to service control (SCFo is the only control point for a UPT call, excluding complex services which would require the co-operation of several SCF's) and to data management (SDFh holds the only copy of the UPT user's profile). While limiting the grade of service which can be offered, this architecture avoids service interaction problems which come up with decentralized architecture.

There is no transparency on the location of data: queries on a UPT user's service profile shall explicitly be addressed to the SDFh entity which stores it.

This architecture requires SDFh to be more than a simple data repository, since it shall be able to perform access control on the SCFo-SDFh link and to authenticate the UPT user.

The SCF-SDF relationship (which is defined in IN CS-1) is already used for UPT Phase 1.

5.2.1.2 Use of the SCF-SCF interface

This architecture is derived from the standard architecture by assembling the IN architecture for the originating, home and terminating networks (see figure 4). The SCF-SCF interface is used for relations between the originating and the terminating networks (e.g. call control and service logic related operations) during call related procedures.

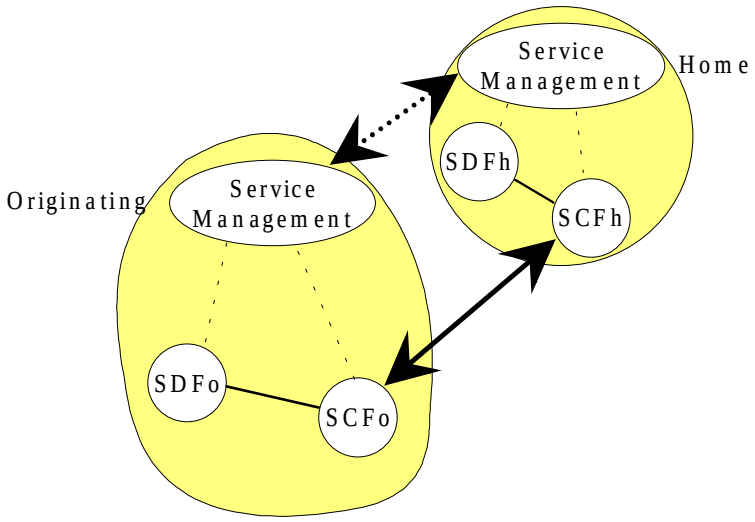


Figure 4: Use of the SCF-SCF interface

In this description, service control is assumed by SCFo, while data management is performed by SCFh. SDFh permanently stores subscriber and user profiles [*and SDFo or SDFt may store temporary copies under control of SCFh (which keeps all copies coherent, if required)*].

SCFo and SCFt house service logic for UPT, including call control logic, supplementary services logic, logic for interacting with the user through the SRF entities, logic for accessing UPT subscriber and user profiles (through interaction with SCFh) and logic for providing multi-point services (through interaction with each other). This description implies that SCFo has some knowledge of the user's home network (see the role of SCFo in clause 7).

SDFo stores information on agreements with other operators, and call records created by SCFo.

SCFh manages the data stored by SDFh and any copy of it transmitted to other networks, performs access control over the SCF-SCF link (authentication of the originating network), and may authenticate the UPT user, upon request from SCFo. Besides, there is the possibility for SCFh to play the part of an assisting SCF for SCFo in order to offer to the UPT user some part of the service that SCFo could not offer on its own.

SDFh only stores subscriber and user profiles for all subscribers of the operator.

This description allows distribution of service logic over several SCFs. During a call, SCFo is activated first, and may then invoke service logic contained in SCFh. While allowing a richer service logic, it opens up the possibility of complex service interaction scenarios between the service logic programs running on different nodes.

This relationship may also be used to extend the query language available in the SDF entities, by making an SCF entity act as a front-end to an SDF entity. The SCF may receive sophisticated and/or service-dependent requests and translate them into an equivalent sequence of elementary operations which are transparently sent to the SDF. This allows for open evolution of the data management facilities provided by the intelligent network.

It is of course possible to combine the different descriptions described above to obtain more elaborate ones.

NOTE: In fact, a terminating network may be involved in the SCF-SCF interface use, for example when secure answering for incoming calls is processed by this terminating network. This needs further study in order to describe the procedure.

5.2.1.3 Use of the SDF-SDF interface

This architecture is derived from the standard architecture by assembling the IN architecture for the originating and terminating networks with the data management part of the architecture of the home network (see figure 5). The SDF-

SDF interface is used for relations with the home network (user authentication operations and retrieval/updating of the user profile information stored in the home network).

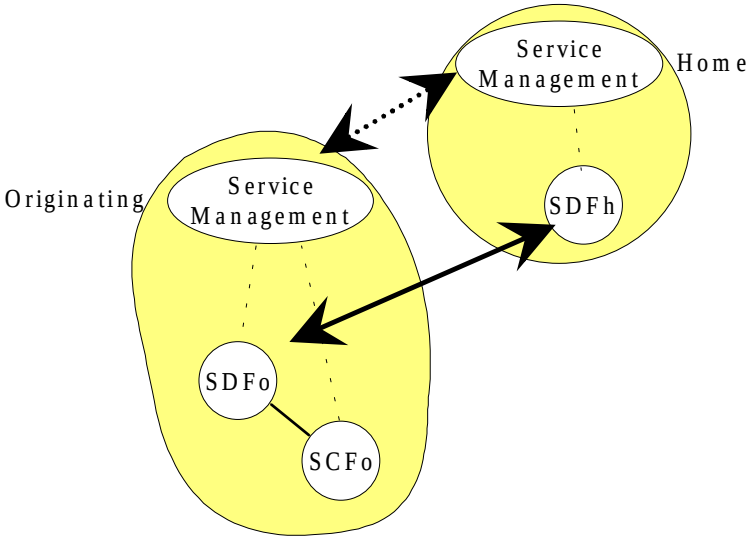


Figure 5: Use of the SDF-SDF interface

In this description, service control is assumed by SCFo, while data management is co-operatively performed by SDFh and SDFo. SDFh permanently stores subscriber and user profiles, but SDFo may store temporary copies under its control.

SCFo houses service logic for UPT, including call control logic, supplementary services logic, logic for interacting with the user through the SRF entities, logic for accessing UPT subscriber and user profiles (through interaction with their local SDF).

SDFo stores information on agreements with other operators, and call records created by SCFo and SCFt. They provide logic for transparent access to the user and subscriber profiles (distributed database with location transparency). This description implies that SDFo has some knowledge of the user's home network (see the role of SDFo in clause 5).

SDFh stores subscriber and user profiles for all subscribers of the operator, manages any shadow copy stored in SDFo or another SDF, and performs access control over the SDF-SDF link (authentication of the originating network).

This description allows distribution of data management functionalities and *location transparency* for data manipulation operations. No matter where the data is located, SCFo issues queries to SDFo. The SDFo entity is responsible for locating the data requested by the SCF entities, which may be in another network. Such a separation extends the ideas of the IN model to a distributed environment: the logic of the service is decoupled from the data for the service.

Furthermore, this approach allows UPT data to be duplicated on several nodes, if necessary (shadowing technique). The SDF entities co-operate to keep track of the data and maintain copies up to date, thus providing *replication transparency* to the service logic program run by the SCF entities.

SDF entities are required to be more than simple data repositories, since they shall be able to perform access control on the SDF-SDF link and to co-operate in order to provide location transparency and replication transparency to the SCF entities.

5.2.2 User access scenarios

Compared to CS1, a new entity, CUSF, exists in CS2 and can be used for call unrelated procedures. Thus, Dual Tone Multi Frequency (DTMF) and Digital Subscriber Signalling System number 1 (DSS1) can be used as follows.

5.2.2.1 DTMF user access

The following architecture is derived from the standard architecture by assembling the IN architecture for user and the network (see figure 6).

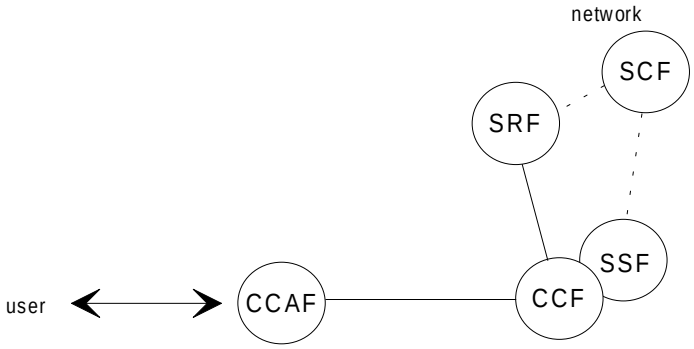


Figure 6: Functional interface of DTMF user access

In this architecture, the functional interface used for interaction between user and the network is CCAF-CCF/SSF.

CCAF establishes B-channel path between user and network.

CCF establishes B-channel path between CCAF and SRF.

SRF sends speech guidance to the user according to direct from SCF and receives information (authentication information, information to modify/retrieve own service profile etc.) using DTMF from user. SRF analyses this information and sends the results to SCF.

Some conclusions of such an architecture are:

- in this scenario, B-channel path shall be established for interaction between user and network;
- this scenario allows interaction for user and network using only DTMF through the B-channel path;
- in this scenario, information from user through the B-channel path is converted to signalling by SRF.

5.2.2.2 DSS1 user access

5.2.2.2.1 ISDN terminals which behave in functional mode

Functional mode is one-shot request type procedure.

Call related outchannel user-network interaction

The following architecture is derived from the standard architecture by assembling the IN architecture for user and the network (see figure 7).

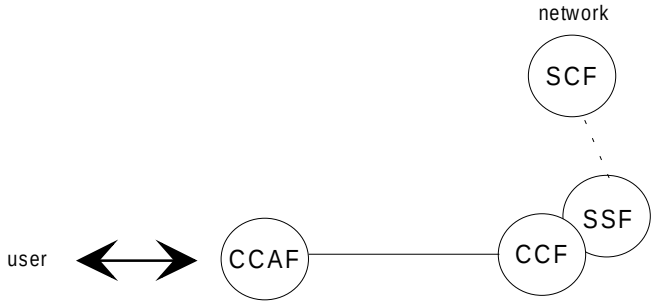


Figure 7: Functional interface of ISDN terminal (functional mode) - call related

In this architecture, the functional interface used for interaction between user and the network is CCAF-CCF/SSF. This scenario is valid to ISDN environments.

CCAF invokes functions to modify or retrieve own service profile not establishing B-channel path between network. And CCAF is activated by network function (e.g. challenge/response authentication).

CCF/SSF sends/receives information (authentication information, information to modify/retrieve user's service profile etc.) using D-channel messages to/from user/CCAF. CCF/SSF sends this information to SCF.

Some conclusions of such an architecture are:

- in this scenario, B-channel path shall be established for interaction between user and network;
- this scenario allows interaction for user and network using D-channel messages in case of call related procedures;
- in this scenario, information is transferred between SCF and user/CCAF by way of CCF/SSF.

Call unrelated user-network interaction

The following architecture is derived from the standard architecture by assembling the IN architecture for user and the network (see figure 8).

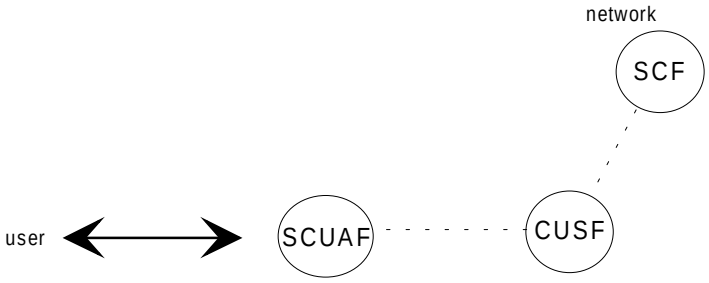


Figure 8: Functional interface of ISDN terminal (functional mode) - call unrelated

In this architecture, the functional interface used for interaction between user and the network is SCUAF-CUSF. This scenario is valid to ISDN environments.

SCUAF invokes functions to modify or retrieve own service profile not establishing B-channel path between network. And SCUAF is activated by network function (e.g. challenge/response authentication).

CUSF sends/receives information (authentication information, information to modify/retrieve user's service profile etc.) using D-channel messages to/from user/CCAF. CUSF sends this information to SCF.

Some conclusions of such an architecture are:

- in this scenario, B-channel path shall not be established for interaction between user and network;
- this scenario allows interaction between user and network in case of unrelated procedures;
- in this scenario, information is transferred between SCF and user/SCUAF by way of CUSF.

5.2.2.2.2 ISDN terminals which behave in stimulus mode

Stimulus mode is manual request/response type procedure.

Call related outchannel user-network interaction

The following architecture is derived from the standard architecture by assembling the IN architecture for user and the network (see figure 9).

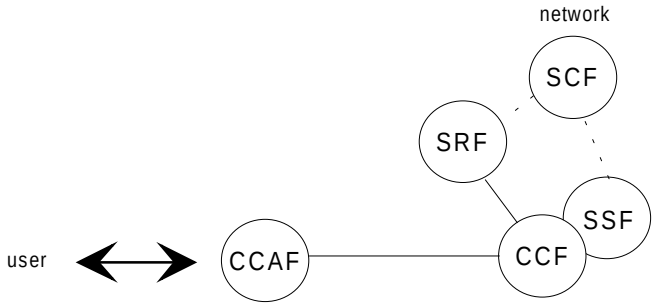


Figure 9: Functional interface of ISDN terminal (stimulus mode) - call related

In this architecture, the functional interface used for interaction between user and the network is CCAF-CCF/SSF. This scenario is valid to ISDN environments.

CCAF establishes B-channel path between user and network. Once the B-channel path is established, CCAF sends information (authentication information, information to modify/retrieve user's service profile, etc.) to CCF/SSF using D-channel messages. CCF/SSF sends this information to SCF.

Some conclusions of such an architecture are:

- this scenario allows interaction for user and network using D-channel messages in case of call related procedures;
- in this scenario, B-channel path shall be established but D-channel is used for interaction between user and network;
- in this scenario, information is transferred between SCF and user through CCF/SSF and SRF.

Call unrelated user-network interaction

The following architecture is derived from the standard architecture by assembling the IN architecture for user and the network (see figure 10).

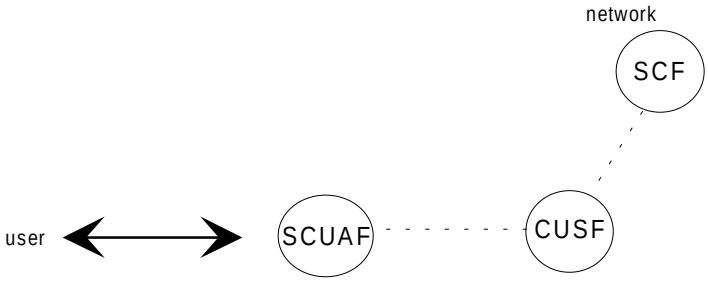


Figure 10: Functional interface of ISDN terminal (stimulus mode) - call unrelated

In this architecture, the functional interface used for interaction between user and the network is SCUAF-CUSF. This scenario is valid to ISDN environments.

SCUAF sends information (authentication information, etc.) to CUSF using D-channel messages.

CUSF receives information (authentication information, information to modify/retrieve user's service profile etc.) using D-channel messages from user/SCUAF and CUSF sends this information to SCF.

Some conclusions of such an architecture are:

- this scenario allows interaction for user and network using D-channel messages only in case of call unrelated procedures (registration, service profile management procedures etc.);
- in this scenario, D-channel path is used for interaction between user and network;
- in this scenario, information is transferred between SCF and user/SCUAF through CUSF;

- in CS2, the SRF is defined only in a call-related case. Consequently, this scenario can be considered only if the SRF is not a separate entity and if an equivalent is implemented in the terminal or the SCF.

5.3 Allocation of network functions to functional entities

Note that the class "O&M" is not covered.

5.3.1 Call related in-band user interaction

5.3.1.1 CCF

Table 1: CCF

Class	Function	Functional Entities		
		SCF-SDF scenario	**SCF-SCF scenario**	**SDF-SDF scenario**
Agreement	Select supporting network operator	CCF + SCFo + SDFh	CCF + SCFo + SCFh	CCF + SCFo + SDFo + SDFh
	Select visited service provider	CCF		
Call control	Route UPT incoming call	CCF		
	Indicate UPT incoming call	SCFo + SRF + CCF (note 3)		
	Negotiate UPT incoming call	SCFo + SRF + CCF + SSF + SDFh (note 4)	SCFo + SRF + CCF + SSF + SCFh + SDFh (note 4)	SCFo + SRF + CCF + SSF + SDFo + SDFh (note 4)
Registration	Indicate de/registration	SCFo + SRF + CCF (note 3)		
Charging	Charge	CCF + SDFo + SCFo (note 1)		
	Provide charging information to user	CCF + SRF / SCFo + CCF + SRF / SMFh (note 2)		
User interaction	Collect information	SRF + CCF + SSF + SCFo (note 6)		
	Send message	SCFo + SRF + CCF + SSF (note 5)		

NOTE 1: The charging information produced by the CCF entity will be transferred to SMFo and/or SMFh, which will lead to an advanced data transfer between SMFo and SMFh for the service management. On the other hand, if real-time credit limit checking is required, this information shall also be transferred from SCFo to SDFh. Note that this function may be split further to refine the description of the charging process. Several allocation scenarios are then possible, but they have no impact on the allocation of UPT functions.

NOTE 2: This function may be implemented by any three functional entities, depending on the type of information provided to the user. For UPT incoming calls, real-time charging information can be provided by the CCF (by emitting a pulse every time a charging unit is spent) or by the SCFo (by emitting voice messages to indicate specific charging conditions). Off-line charging information can be provided by the SMFh (by listing the costs incurred so far since the last bill, etc.) on request of the user.

NOTE 3: This function needs these three entities. Indeed, the message is decided and ordered by SCFo and sent by SRF, through CCF.

NOTE 4: These entities are necessary to negotiate and handle the UPT user's choice for incoming calls. This function is an advanced interactive function and could be split in several functions in coming versions of the present document.

NOTE 5: This function needs these entities. Indeed, the message is decided and ordered by SCFo and sent by SRF, through CCF/SSF.

NOTE 6: These entities are involved to collect the information given by the user.

5.3.1.2 SSF

Table 2: SSF

Class	Function	Functional Entities		
		SCF-SDF scenario	**SCF-SCF scenario**	**SDF-SDF scenario**
Access point	Request API	SSF + SCFo (note 1)		
	Determine API	SSF + SCFo (note 1)		
	Provide API	SSF + SCFo + SDFh (note 2)	SSF + SCFo + SDFo / SSF + SCFo + SCFh + SDFh (note 2 and 3)	SSF + SCFo + SDFo / SSF + SCFo + SDFo + SDFh (note 2 and 3)
Call control	Recognize UPT incoming call	SSF + SCFo (note 4)		
	Negotiate UPT incoming call	SCFo + SRF + CCF + SSF + SDFh (note 7)	SCFo + SRF + CCF + SSF + SCFh + SDFh (note 7)	SCFo + SRF + CCF + SSF + SDFo + SDFh (note 7)
	Recognize UPT outgoing call	SSF + SCFo (note 4)		
Registration	Recognize registration/deregistration	SSF + SCFo (note 4)		
Service profile	Recognize service profile management	SSF + SCFo (note 4)		
Service	Recognize UPT request	SSF (note 5)		
User interaction	Collect information	SRF + CCF + SSF + SCFo (note 6)		
	Send message	SCFo + SRF + CCF + SSF (note 8)		

NOTE 1: This function is realized by SCFo thanks to SSF. SSF is involved to obtain the Calling Line Identity at registration.
NOTE 2: The different functions help to provide the API from the SDFh to the user.
NOTE 3: The choice between the different possibilities depends whether there is a transfer of data before the processing of the function.
NOTE 4: These functions are implemented by the two functional entities, working in collaboration: the SSF triggers on a specific prefix or number and sends a Provide Instruction request to the SCFo, which then determines which service to activate (incoming call, outgoing call, etc.), depending on the prefix or number used.
NOTE 5: This function only recognizes the fact that a request for UPT service exists. The exact nature of the request is determined later by the SCFo, using the recognition functions, which interact with the user to determine which service is requested. Only the SSF is therefore involved.
NOTE 6: See note 6 of table 1 on the CCF.
NOTE 7: See note 4 of table 1 on the CCF.
NOTE 8: See note 5 of table 1 on the CCF.

5.3.1.3 SCFo

Table 3: SCF

Class	Function	Functional Entities		
		SCF-SDF scenario	**SCF-SCF scenario**	**SDF-SDF scenario**
Access point	Request API	SSF + SCFo (note 1)		
	Determine API	SSF + SCFo (note 1)		
	Provide API	SSF + SCFo + SDFh (note 2)	SSF + SCFo + SDFo / SSF + SCFo + SCFh + SDFh (note 2 and 3)	SSF + SCFo + SDFo / SSF + SCFo + SDFo + SDFh (note 2 and 3)
	Verify API	SCFo + SDFh (note 4)	SCFo + SDFo / SCFo + SCFh + SDFh (note 3 and 4)	SCFo + SDFo / SCFo + SDFo + SDFh (note 3 and 4)
	Determine Routing Address	SCFo + SDFh (note 4)	SCFo + SDFo / SCFo + SCFh + SDFh (note 3 and 4)	SCFo + SDFo / SCFo + SDFo + SDFh (note 3 and 4)
PUI	Request PUI	SCFo		
Agreement	Select supporting network	CCF + SCFo + SDFh	CCF + SCFo + SCFh	CCF + SCFo + SDFo + SDFh
Authentication	Request authentication information	SCFo		
	Authenticate	SCFo + SDFh (note 5)	SCFo + SCFh + SDFh (note 5)	SCFo + SDFo + SDFh (note 5)
Call control	Recognize UPT incoming call	SSF + SCFo (note 6)		
	Process UPT incoming call	SCFo + SDFh (note 7)	SCFo + SDFo / SCFo + SCFh + SDFh (note 3 and 7)	SCFo + SDFo / SCFo + SDFo + SDFh (note 3 and 7)
	Indicate UPT incoming call	SCFo + SRF + CCF (note 13)		
	Negotiate UPT incoming call	SCFo + SRF + CCF + SSF + SDFh (note 15)	SCFo + SRF + CCF + SSF + SCFh + SDFh (note 15)	SCFo + SRF + CCF + SSF + SDFh (note 15)
	Recognize UPT outgoing call	SSF + SCFo (note 6)		
	Process UPT outgoing call	SCFo + SDFh (note 7)	SCFo + SDFo / SCFo + SCFh + SDFh (note 3 and 7)	SCFo + SDFo / SCFo + SDFo + SDFh (note 3 and 7)
Registration	Indicate de/registration	SCFo + SRF + CCF (note 13)		
	Recognize registration/deregistration	SSF + SCFo (note 6)		
	Process registration/deregistration	SCFo + SDFh (note 7)	SCFo + SDFo / SCFo + SCFh + SDFh (note 3 and 7)	SCFo + SDFo / SCFo + SDFo + SDFh (note 3 and 7)
Service profile	Recognize service profile management	SSF + SCFo (note 6)		
	Process service profile management	SCFo + SDFh / SMFh + SDFh (note 3, 7 and 8)	SCFo + SDFo / SCFo + SCFh + SDFh / SMFh + SDFh (note 3, 7 and 8)	SCFo + SDFo / SCFo + SDFo + SDFh / SMFh + SDFh (note 3, 7 and 8)
	Present service profile	SRF + SCFo + SDFh / SMFh + SDFh (note 3, 7 and 8)	SRF + SCFo + SDFo / SRF + SCFo + SCFh + SDFh / SMFh + SDFh (note 3, 7 and 8)	SRF + SCFo + SDFo / SRF + SCFo + SDFo + SDFh / SMFh + SDFh (note 3, 7 and 8)
Service	Service assisting request	-	SCFo + SCFh (note 16)	-
Charging	Charge	CCF + SDFo + SCFo (note 9)		
	Determine charging rate	SCFo + SDFh (note 4)	SCFo SDFo / SCFo + SCFh + SDFh (note 3 and 4)	SCFo + SDFo / SCFo + SDFo + SDFh (note 3 and 4)
	Transfer charging information	-	SCFo + SCFh (note 18)	-

(continued)

Table 3 (concluded): SCF

Class	Function	Functional Entities		
		SCF-SDF scenario	**SCF-SCF scenario**	**SDF-SDF scenario**
	Provide charging information to user	CCF + SRF / SCFo + CCF + SRF / SMFh (note 10)		
Transfer of data	Copy data	SCFo + SDFo + SDFh (note 11)	SCFo + SDFo + SCFh + SDFh (note 11)	SDFo + SDFh (note 11)
	Update data	SCFo + SDFo + SDFh (note 11)	SCFo + SDFo + SCFh + SDFh (note 11)	SDFo + SDFh (note 11)
	Erase the copied data	SCFo + SDFo (note 12)		
User interaction	Send message	SCFo + SRF + CCF + SSF (note 17)		
	Collect information	SRF + CCF + SSF + SCFo (note 6)		

NOTE 1: See note 1 of table 2 on the SSF.
NOTE 2: See note 2 of table 2 on the SSF.
NOTE 3: See note 3 of table 2 on the SSF.
NOTE 4: The entities work in collaboration to realize the function.
NOTE 5: The authentication of the UPT user is performed by SDFh, upon request from SCFo.
NOTE 6: See note 6 of table 1 on the CCF.
NOTE 7: SCFo processes the function with the help of the data given by SDFh (through different entities depending the scenario).
NOTE 8: As discussed later in the present document (in the subclause 6.9 on UPT requirements on SMAF), the user has two methods to perform service profile management: either through call handling entities or through management entities. Both need to have the corresponding functions.
NOTE 9: See note 1 of table 1 on the CCF.
NOTE 10: See note 2 of table 1 on the CCF.
NOTE 11: Data are transferred from SDFh to SDFo (directly or through SCFo or SCFh and SCFo, depending the scenario).
NOTE 12: Data are erased from SDFo on request of SCFo.
NOTE 13: See note 3 of table 1 on the CCF.
NOTE 14: See note 6 of table 2 on the SSF.
NOTE 15: See note 4 of table 1 on the CCF.
NOTE 16: In that case the controlling SCF make a request for assisting to the supporting SCF (SCFh here). It covers both HIR and SCF Bind Request flows.
NOTE 17: See note 5 of table 1 on the CCF.
NOTE 18: In case of assisting between service logics, some charging information may be transferred between these two logics.

5.3.1.4 SDFo

Table 4: SDFo

Class	Function	Functional Entities		
		SCF-SDF scenario	**SCF-SCF scenario**	**SDF-SDF scenario**
Access point	Provide API	-	SSF + SCFo + SDFo / SSF + SCFo + SCFh + SDFh (note 1)	SSF + SCFo + SDFo / SSF + SCFo + SDFo + SDFh (note 1)
	Verify API	-	SCFo + SDFo / SCFo + SCFh + SDFh (note 2)	SCFo + SDFo / SCFo + SDFo + SDFh (note 2)
	Determine Routing Address	-	SCFo + SDFo / SCFo + SCFh + SDFh (note 2)	SCFo + SDFo / SCFo + SDFo + SDFh (note 2)
	Store network capabilities	SDFo		
	Retrieve network capabilities	SDFo		
	Retrieve API/Routing Address	SDFo / SDFh (note 3)		
	Retrieve Access Point capabilities	SDFo / SDFh (note 3)		
Agreements	Determine service provider identity	SDFo		
	Check service provider agreement	SDFo		
	Determine supporting network identity	SDFo		
	Check supporting network agreement	SDFo		
	Select supporting network operator	-	-	CCF + SCFo + SDFo + SDFh
Authenticate	Authenticate	-	-	SCFo + SDFo + SDFh (note 9)
Call control	Process UPT incoming call	-	SCFo + SDFo / SCFo + SCFh + SDFh (note 4)	SCFo + SDFo / SCFo + SDFo + SDFh (note 4)
	Process UPT outgoing call	-	SCFo + SDFo / SCFo + SCFh + SDFh (note 4)	SCFo + SDFo / SCFo + SDFo + SDFh (note 4)
	Negotiate UPT incoming call	-	-	SCFo + SRF + CCF + SSF + SDFo + SDFh (note 10)
Registration	Process registration/deregistration	-	SCFo + SDFo / SCFo + SCFh + SDFh (note 4)	SCFo + SDFo / SCFo + SDFo + SDFh (note 4)
Service profile	Process service profile management	-	SCFo + SDFo / SCFo + SCFh + SDFh / SMFh + SDFh (note 5)	SCFo + SDFo / SCFo + SDFo + SDFh / SMFh + SDFh (note 5)
	Retrieve service profile	SDFo / SDFh (note 3)		
	Present service profile	-	SRF + SCFo + SDFo / SRF + SCFo + SCFh + SDFh / SMFh + SDFh (note 5)	SRF + SCFo + SDFo / SRF + SCFo + SDFo + SDFh / SMFh + SDFh (note 5)
Service	Retrieve service information	SDFo / SDFh (note 3)		
Charging	Charge	CCF + SDFo + SCFo (note 6)		
	Transfer charging information	-	SCFo + SCFh (note 11)	-
	Determine charging rate	-	SCFo + SDFo / SCFo + SCFh + SDFh (note 2)	SCFo + SDFo / SCFo + SDFo + SDFh (note 2)

(continued)

Table 4 (concluded): SDFo

Class	Function	Functional Entities		
		SCF-SDF scenario	**SCF-SCF scenario**	**SDF-SDF scenario**
Transfer of data	Copy data	SCFo + SDFo + SDFh (note 7)	SCFo + SDFo + SCFh + SDFh (note 7)	SDFo + SDFh (note 7)
	Update data	SCFo + SDFo + SDFh (note 7)	SCFo + SDFo + SCFh + SDFh (note 7)	SDFo + SDFh (note 7)
	Erase the copied data	SCFo + SDFo (note 8)		

NOTE 1: See notes 2 and 3 of table 2 on the SSF.
NOTE 2: See note 3 of table 2 on the SSF and note 4 of table 3 on the SCFo.
NOTE 3: This function is realized by SDFo or SDFh: indeed, it depends on whether a copy of data was made from SDFh to SDFo (see the class "*Transfer of data").*
NOTE 4: See note 3 of table 2 on the SSF and note 7 of table 3 on the SCFo.
NOTE 5: See note 3 of table 2 on the SSF and notes 7 and 8 of table 3 on the SCFo.
NOTE 6: See note 1 of table 1 on the CCF.
NOTE 7: See note 11 of table 3 on the SCFo.
NOTE 8: See note 12 of table 3 on the SCFo.
NOTE 9: See note 5 of table 3 on the SCFo.
NOTE 10: See note 4 of table 1 on the CCF.
NOTE 11: See note 18 of table 1 on the SCFo.

5.3.1.5 SCFh

Table 5: SCFh

Class	Function	Functional Entities SCF-SCF scenario
Access point	Provide API	SSF + SCFo + SDFo / SSF + SCFo + SCFh + SDFh (note 1)
	Verify API	SCFo + SDFo / SCFo + SCFh + SDFh (note 2)
	Determine Routing Address	SCFo + SDFo / SCFo + SCFh + SDFh (note 2)
Agreement	Select supporting network operator	CCF + SCFo + SCFh
Authentication	Authenticate	SCFo + SCFh + SDFh (note 3)
Call control	Process UPT incoming call	SCFo + SDFo / SCFo + SCFh + SDFh (note 2)
	Negotiate UPT incoming call	SCFo + SRF + CCF + SSF + SCFh + SDFh (note 4)
	Process UPT outgoing call	SCFo + SDFo / SCFo + SCFh + SDFh (note 2)
Registration	Process registration/deregistration	SCFo + SDFo / SCFo + SCFh + SDFh (note 2)
Service profile	Process service profile management	SCFo + SDFo / SCFo + SCFh + SDFh / SMFh + SDFh (note 5)
	Present service profile	SRF + SCFo + SDFo / SRF + SCFo + SCFh + SDFh / SMFh + SDFh (note 5)
Service	Service assisting request	SCFo + SCFh (note 7)
Charging	Determine charging rate	SCFo + SDFo / SCFo + SCFh + SDFh (note 2)
	Transfer charging information	SCFo + SCFh (note 8)
Transfer of data	Copy data	SCFo + SDFo + SCFh + SDFh (note 6)
	Update data	SCFo + SDFo + SCFh + SDFh (note 6)

NOTE 1: See notes 2 and 3 of table 2 on the SSF.
NOTE 2: See note 3 of table 2 on the SSF and note 4 of table 3 on the SCFo.
NOTE 3: See note 5 of table 3 on the SCFo.
NOTE 4: See note 4 of table 1 on the CCF.
NOTE 5: See note 3 of table 2 on the SSF and notes 7 and 8 of table 3 on the SCFo.
NOTE 6: See note 11 of table 3 on the SCFo.
NOTE 7: See note 16 of table 3 on the SCFo.
NOTE 8: See note 11 of table 3 on the SCFo.

5.3.1.6 SDFh

Table 6: SDFh

Class	Function	Functional Entities		
		SCF-SDF scenario	**SCF-SCF scenario**	**SDF-SDF scenario**
Access point	Store API/Routing Address	SDFh		
	Retrieve API/Routing Address	SDFo / SDFh (note 1)		
	Store Access Point capabilities	SDFh		
	Retrieve Access Point capabilities	SDFo / SDFh (note 1)		
	Provide API	SSF + SCFo + SDFh (note 2)	SSF + SCFo + SDFo / SSF + SCFo + SCFh + SDFh (note 2)	SSF + SCFo + SDFo / SSF + SCFo + SDFo + SDFh (note 2)
	Verify API	SCFo + SDFh (note 10)	SCFo + SDFo / SCFo + SCFh + SDFh (note 3)	SCFo + SDFo / SCFo + SDFo + SDFh (note 3)
	Determine Routing Address	SCFo + SDFh (note 10)	SCFo + SDFo / SCFo + SCFh + SDFh (note 3)	SCFo + SDFo / SCFo + SDFo + SDFh (note 3)
PUI	Determine PUI	SDFh		
	Verify PUI	SDFh		
Agreements	Determine orig. network/service op. identity	SDFh		
	Check network/service operator agreement	SDFh		
	Select supporting network operator	CCF + SCFo + SDFh	-	CCF + SCFo + SDFo + SDFh
Authentication	Store authentication information	SDFh		
	Modify authentication information	SDFh		
	Retrieve authentication information	SDFh		
	Authenticate	SCFo + SDFh (note 4)	SCFo + SCFh + SDFh (note 4)	SCFo + SDFo + SDFh (note 4)
Call control	Process UPT incoming call	SCFo + SDFh (note 11)	SCFo + SDFo / SCFo + SCFh + SDFh (note 5)	SCFo + SDFo / SCFo + SDFo + SDFh (note 5)
	Negotiate UPT incoming call	SCFo + SRF + CCF + SSF + SDFh (note 9)	SCFo + SRF + CCF + SSF + SCFh + SDFh (note 9)	SCFo + SRF + CCF + SSF + SDFo + SDFh (note 9)
	Process UPT outgoing call	SCFo + SDFh (note 11)	SCFo + SDFo / SCFo + SCFh + SDFh (note 5)	SCFo + SDFo / SCFo + SDFo + SDFh (note 5)
Registration	Process registration/deregistration	SCFo + SDFh (note 11)	SCFo + SDFo / SCFo + SCFh + SDFh (note 5)	SCFo + SDFo / SCFo + SDFo + SDFh (note 5)
Service profile	Process service profile management	SCFo + SDFh / SMFh + SDFh (note 6 and 11)	SCFo + SDFo / SCFo + SCFh + SDFh / SMFh + SDFh (note 5 and 6)	SCFo + SDFo / SCFo + SDFo + SDFh / SMFh + SDFh (note 5 and 6)
	Store service profile	SDFh		
	Check service profile user access	SDFh		
	Modify service profile	SDFh		
	Retrieve service profile	SDFo / SDFh (note 1)		
	Present service profile	SRF + SCFo + SDFh / SMFh + SDFh (note 6 and 11)	SRF + SCFo + SDFo / SRF + SCFo + SCFh + SDFh / SMFh + SDFh (note 5 and 6)	SRF + SCFo + SDFo / SRF + SCFo + SDFo + SDFh / SMFh + SDFh (note 5 and 6)
Service	Retrieve service information	SDFo / SDFh (note 1)		
	Check service against profile	SDFh		
	Check service against capabilities	SDFh		

(continued)

Table 6 (concluded): SDFh

Class	Function	Functional Entities		
		SCF-SDF scenario	**SCF-SCF scenario**	**SDF-SDF scenario**
Charging	Determine charging rate	SCFo + SDFh (note 10)	SCFo + SDFo / SCFo + SCFh + SDFh (note 12)	SCFo + SDFo / SCFo + SDFo + SDFh (note 12)
	Check credit against limits	SDFh / SMFh (note 7)		
	Update credit limit	SDFh		
	Store charging information	SDFh		
Transfer of data	Copy data	SCFo + SDFo + SDFh (note 8)	SCFo + SDFo + SCFh + SDFh (note 8)	SDFo + SDFh (note 8)
	Update data	SCFo + SDFo + SDFh (note 8)	SCFo + SDFo + SCFh + SDFh (note 8)	SDFo + SDFh (note 8)

NOTE 1: See note 1 of table 4 on the SDFo.
NOTE 2: See notes 2 and 3 of table 2 on the SSF.
NOTE 3: See note 3 of table 2 on the SSF and note 4 of table 3 on the SCFo.
NOTE 4: See note 5 of table 3 on the SCFo.
NOTE 5: See note 3 of table 2 on the SSF and note 7 of table 3 on the SCFo.
NOTE 6: See note 8 of table 3 on the SCFo.
NOTE 7: This function may be implemented by two functional entities, depending on the type of credit limit checking required. If real-time checking is needed, SDFh does it. If off-line credit checking is needed, SMFh does it.
NOTE 8: See note 11 of table 3 on the SCFo.
NOTE 9: See note 4 of table 1 on the CCF.
NOTE 10: See note 4 of table 3 on the SCFo.
NOTE 11: See note 7 of table 3 on the SCFo.
NOTE 12: See note 3 of table 2 on the SSF and note 4 of table 3 on the SCFo.

5.3.1.7 SMFo & SMFh

Table 7: SMFo & SMFh

Class	Function	Functional Entities		
		SCF-SDF scenario	**SCF-SCF scenario**	**SDF-SDF scenario**
Service profile	Process service profile management	SCFo + SDFh / SMFh + SDFh (note 1)	SCFo + SDFo / SCFo + SCFh + SDFh / SMFh + SDFh (note 1)	SCFo + SDFo / SCFo + SDFo + SDFh / SMFh + SDFh (note 1)
	Present service profile	SRF + SCFo + SDFh / SMFh + SDFh (note 1)	SRF + SCFo + SDFo / SRF + SCFo + SCFh + SDFh / SMFh + SDFh (note 1)	SRF + SCFo + SDFo / SRF + SCFo + SDFo + SDFh / SMFh + SDFh (note 1)
Charging	Check credit against limits	SDFh / SMFh (note 2)		
	Transfer charging information	SCFo + SCFh (note 3)		
	Provide charging information to user	SRF + CCF / SRF + CCF + SCFo / SMFh (note 4)		

NOTE 1: See note 3 of table 2 on the SSF and notes 7 and 8 of table 3 on the SCFo.
NOTE 2: See note 7 of table 6 on the SDFh.
NOTE 3: See note 11 of table 3 on the SCFo.
NOTE 4: See note 2 of table 1 on the CCF.

5.3.1.8 SRF

Table 8: SRF

Class	Function	Functional Entities		
		SCF-SDF scenario	**SCF-SCF scenario**	**SDF-SDF scenario**
Call control	Indicate UPT incoming call	SCFo + SRF + CCF (note 1)		
	Negotiate UPT incoming call	SCFo + SRF + CCF + SSF + SDFh (note 2)	SCFo + SRF + CCF + SSF + SCFh + SDFh (note 2)	SCFo + SRF + CCF + SSF + SDFo + SDFh (note 2)
Registration	Indicate de/registration	SCFo + SRF + CCF (note 1)		
Service profile	Present service profile	SRF + SCFo + SDFh / SMFh + SDFh (note 3)	SRF + SCFo + SDFo / SRF + SCFo + SCFh + SDFh / SMFh + SDFh (note 3)	SRF + SCFo + SDFo / SRF + SCFo + SDFo + SDFh / SMFh + SDFh (note 3)
Charging	Provide charging information to user	SRF + CCF / SRF + CCF + SCFo / SMFh (note 4)		
User interaction	Send message	SCFo + SRF + CCF + SSF (note 6)		
	Collect information	SRF + CCF + SSF + SCFo (note 5)		

NOTE 1: See note 3 of table 1 on the CCF.
NOTE 2: See note 4 of table 1 on the CCF.
NOTE 3: See note 3 of table 2 on the SSF and notes 7 and 8 of table 3 on the SCFo.
NOTE 4: See note 2 of table 1 on the CCF.
NOTE 5: See note 6 of table 1 on the CCF.
NOTE 6: See note 5 of table 1 on the CCF.

5.3.1.9 Others

No UPT-specific functions have been identified for SCEF, CCAF and SMAF, so far.

5.3.2 Call related outchannel user interaction

As in subclause 5.3.1, but without the use of the SRF.

5.3.3 Call unrelated user interaction

This subclause describes the delta modifications compared to subclause 5.3.1. This is for outchannel and call unrelated interaction.

5.3.3.1 SCUAFo

Table 9: SCUAFo

Class	Function	Functional Entities		
		SCF-SDF scenario	**SCF-SCF scenario**	**SDF-SDF scenario**
Authentication	Request authentication information	SCUAFo + CUSFo + SCFo (note 2)		
Registration	Indicate de/registration	SCUAFo + CUSFo + SCFo (note 1)		
User interaction	Send message	SCUAFo + CUSFo + SCFo (note 1)		

NOTE 1: The functionality is requested/initiated by the terminal, relayed by the CUSF and processed by the SCFo. In case of "Send message", for service profile interrogation, Facility messages could be used to carry information.
NOTE 2: When call unrelated, the functionality is requested by the terminal, relayed by the CUSF and processed by the SCFo.

5.3.3.2 CUSFo

Table 10: CUSFo

Class	Function	Functional Entities		
		SCF-SDF scenario	SCF-SCF scenario	SDF-SDF scenario
Authentication	Request authentication information	SCUAFo + CUSFo + SCFo (note 2)		
Registration	Recognize registration/deregistration	CUSFo + SCFo (note 3)		
	Indicate de/registration	SCUAFo + CUSFo + SCFo (note 1)		
Service profile	Recognize service profile management	CUSFo + SCFo (note 3)		
User interaction	Send message	SCUAFo + CUSFo + SCFo (note 1)		
NOTE 1: See note 1 of table 9 on the SCUAFo. NOTE 2: See note 2 of table 9 on the SCUAFo. NOTE 3: The SCFo is triggered by the CUSFo.				

5.3.3.3 SCFo

Table 11: SCFo

Class	Function	Functional Entities		
		SCF-SDF scenario	SCF-SCF scenario	SDF-SDF scenario
Authentication	Request authentication information	SCUAFo + CUSFo + SCFo (note 1)		
Registration	Recognize registration/deregistration	CUSFo + SCFo (note 3)		
	Indicate de/registration	SCUAFo + CUSFo + SCFo (note 1)		
Service profile	Recognize service profile management	CUSFo + SCFo (note 3)		
User interaction	Send message	SCUAFo + CUSFo + SCFo (note 2)		
NOTE 1: See note 2 of table 9 on SCUAFo. NOTE 2: See note 1 of table 9 on the SCUAFo. NOTE 3: The SCFo is triggered by the CUSFo.				

6 UPT requirements on IN functional entities

The requirements or consequences related to the evolution from Phase 1 to Phase 2 are more precisely described in clause 10 and in DEG/NA-064003 [4].

6.1 CCAF

A major service requirement for UPT is that there be as few requirements as possible on the CCAF: existing terminals should be used as far as possible with minor or no modifications. In the PSTN, which is likely to be considered for the first implementation of UPT, the majority of terminals are DTMF or decadic-pulse telephone sets. The interaction between the UPT user and the network will therefore be based on decadic characters with, possibly, the addition of "*" and "#".

A distinction is made between the digits which are dialled in order to route the UPT call to an appropriate network point (i.e. an SSP) and the digits/characters dialled after those, which are used to exchange information between the UPT user and the UPT service. These digits/characters shall be passed by all the exchanges from the user up to the SSP (by transforming decadic pulses into DTMF tones if necessary) and shall be recognized by the SSP. The SRF is expected to play a significant role in the user-to-network interaction.

The UPT CCAF is a combination of the terminal, the subscription device and network functionalities.

6.2 SCUAF

As for CCAF, a major service requirement for UPT is that there be as few requirements as possible on the SCUAF: existing terminals should be used as far as possible with minor or no modifications.

The UPT SCUAF is a combination of the terminal, the subscription device and network functionalities.

6.3 CCF

Ideally, UPT should have no requirements on the CCF functional entity. However, it has already been discussed in the previous subclause that CCF's may be required to convert decadic pulses into DTMF tones in some cases. Also, UPT being a complex service, it may require that additional information be passed through the network by the signalling system. Some examples of possible requirements are: transmission of Calling Line identity to the SSP, transmission of UPT subscriber identity to the called terminal, etc.

Further study is required on this issue, which should be considered with a high priority for the following reasons:

- Some of the information required by UPT may not be available in traditional trunk signalling systems. This could imply the availability of Call Control for Signalling system number 7 (CCS7) throughout the network.
- Some of the information required by UPT may imply the modification of existing signalling protocols standards.

6.4 SSF

The capabilities currently under study for this functional entity may be sufficient for the provision of UPT. However, some UPT services (like the registration for outgoing calls) may require that the SSF be available in local exchanges or that additional functionalities be made available to local exchanges if the SSF is available in transit exchanges only.

6.5 CUSF

The capabilities currently under study for this functional entity may be sufficient for the provision of UPT. However, some UPT services (like the registration procedures) may require that the CUSF be available in local exchanges.

6.6 SCF

Some such requirements may be foreseen: UPT being an international service, the SCF should be able to query an SDF (SDFh) in another network (in case of a SCF-SDF scenario) or ask for the help of another SCF (SCFh; in case of a SCF-SCF scenario). The main requirements for the SCF Phase 2 are the following ones:

- the SCF will have the capabilities to address other SCFs in order to request assistance while it is keeping the control of the call. In this state, the service logic program (SLP) running in SCFo is halted until the proper response is got from SCFh or the co-operation is aborted;
- related to the former requirement, SCFh may request additional information to SCFo before providing the response invoked by the request of assistance. Consequently, SCFh may have an enhanced service logic.

6.7 SRF

In UPT Phase 1, the interaction between the user and the network will take place via in-band communication: DTMF tones will be used in one direction and voice prompts in the other direction. The SRF will be the interworking unit between the network and the SCFo: it will translate the dialling tones received from the user into messages to the SCFo and will transform SCFo requests into voice announcements sent to the user.

The definition of the appropriate announcements is left to the document on service aspects. However, some features already foreseen in that document (such as the availability of announcements in different languages) may have some impact on the complexity of the implementation of the SRF and therefore need to be studied from a network perspective as well.

In UPT Phase 2, the SRF may have to send also outchannel messages (not in-band), see subclause 5.2.2.

6.8 SDF

This functional entity will store all data related to the UPT service and to the UPT subscriber. *Urgent* study is required on this functional entity, for at least the following reasons:

- UPT will require a considerable amount of subscriber information to be stored in the network (location information, service profile, authentication information, etc.). It shall be ensured that the structure and contents of this information can be reflected in the SDF.
- The information about the UPT service as a whole will be distributed across several SDF entities. Co-operation between these SDFs and integrity of the data they collectively maintain shall be guaranteed.
- Furthermore the SDFs co-operating to maintain the UPT database will be spread across several networks and be under the supervision of different operators. Suitable interfaces and security mechanisms shall be devised to ensure this.
- In UPT, SDFh is not limited to data storage and management functions. It shall also be able to:
  - Perform the authentication of a UPT user upon request from an SCFo.
  - Perform credit limit checks after UPT calls (i.e. in real-time) if required.

UPT Phase 2 requirements will be the following ones:

- no additional requirement is proposed for SDFo, as long as SDFo remains a local SDF;
- in UPT Phase 2, SDFh performs the authentication of a UPT user, data control access, checks related to the service profile (credit limit, subscribed services, etc.) upon request from its SCFh or from the SCFo; indeed, data retrieval is made thanks to the SCF-SDF relationship (SCF-SCF is not intended for the access to the data). The SDFh will have at least the same capabilities as in Phase 1 (see clause 10). The SCFh may be involved in data transfer with SDFo, thanks to the SDF-SDF relationship.
- In UPT Phase 2, in the case of a SDF-SDF scenario, a SDF should be able to query a SDF (SDFh) in another network. Besides, because of this non-use of SCFh, each SDFh will have to check the access possibilities of SDFs and forbid this access if necessary.

6.9 SMAF

There are two ways in which UPT procedures other than call handling procedures can be performed in an IN-structured network:

- The user interacts with the same terminal he uses for making and receiving calls, and the exchange of information related to the UPT procedure flows through the CCAF, CCF, SSF, SCF and SDF entities, in this order. Due to limitations imposed by the terminal this method may not be used for the more complex management procedures.
- The user interacts with a terminal dedicated to service management (the SMAF). The exchange of information related to the UPT procedure then flows through the SMAF, SMF, SCF and SDF entities, in this order. For technological reasons, this method is in general more powerful, more flexible and more user-friendly than the previous one. It may, however, not be available to all users at all times, since it requires access to some specific equipment.

From a PSTN access, the first method will be used for those procedures which are simple in nature and need to be performed quite often and from different places (personal mobility procedures). The second method will be used for those procedures which are more complex but less frequent and can be performed from selected locations only (service management procedures).

From an ISDN access, a more user-friendly terminal may be used, allowing the use of more complex procedures than with a PSTN access. The use of an SMAF may, however, still be required for the more complex procedures. The objective is to allow, in future phases at least, all service management procedures to be used from an ISDN terminal.

Most terminals in UPT Phase 1 will use a PSTN access. UPT users will therefore have the need for an SMAF in order to use the service management procedures. Thus, the SMAF should be flexible enough to allow the use of these procedures, which may be quite complex in some cases (e.g. a registration on multiple terminals combined with the creation of a decision pattern for incoming calls).

6.10 SMF

Beyond the requirements on SMF implied by the subclause on the SMAF above, the following are identified:

- Requirements related to the edition of the user's service profile.
- Requirements related to the checking of user credit limits.

6.11 SCEF

No requirements on the SCEF are identified at the moment.

7 Applications of the functional architecture

This clause applies the functional architecture and the CS2 interconnection descriptions proposed in clause 5 to several aspects of the UPT service:

- Authentication.
- Registration for incoming calls.
- Locate UPT user.
- Called party specified secure answering of incoming call.

The proposed descriptions are indicative and the CS1 descriptions can be applied.

NOTE: It is still to be proven that all procedures can be used on the SCF-SCF interface; this is not obvious for the call unrelated procedures. See CS2 and the definition of SCF-SCF (only for call related procedures in CS2).

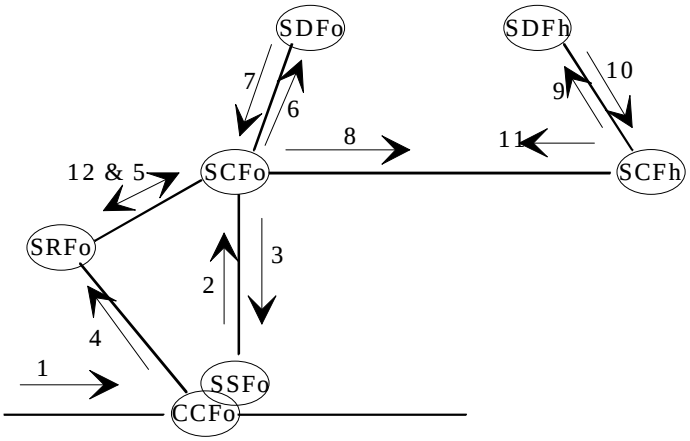
7.1 Authentication

7.1.1 Use of the SCF-SCF interface

If the SCF-SCF link is secure, the authentication of the UPT user can be performed in a different way than for Phase 1, by using this CS2 SCF-SCF interface described in subclause 5.2.1 (figure 4). The authentication procedure is described by assuming that the UPT user has dialled the UPT access code, that SSFo has handed over the control to SCFo and that SCFo has connected to the SRF in order to receive information from the user (this is called the initiation phase on the figure). The knowledge of the authentication algorithm is only in the home domain.

The authentication phase then begins: SCFo uses the user's PUI and authentication data to formulate an authentication request to SCFh. SCFh exchanges information with SDFh and checks the blocking status of the PUI, checks the validity of the authentication data, etc. Moreover, with this description, the algorithm used for authentication could be performed by the SCFh.

Next, SCFh informs SCFo about the outcome of the authentication procedure.



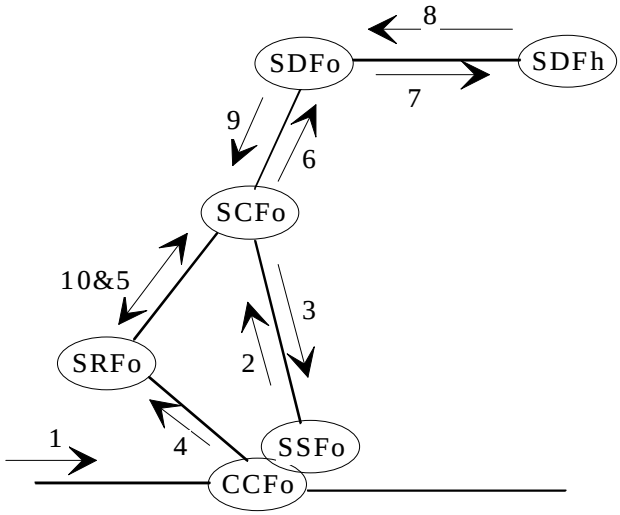
Initiation (same as Phase 1)
1. Dialled UPT access code.
2. Provide instructions.
3. Connect to resource.
4. Set up.
5. Collect user information.
6 & 7. Check of agreements between orginating and home networks.

Authentication
8. Request for authentication.
9. Check blocking status, check authentication data.
10. Response.
11. Authentication result.
12. Prompt 1 collect feature.

Figure 11: Authentication (use of the SCF-SCF interface)

7.1.2 Use of the SDF-SDF interface

The authentication with the SDF-SDF interface is described below:



Initiation (same as Phase 1)
1. Dialled UPT access code.
2. Provide instructions.
3. Connect to resource.
4. Set up.
5. Collect user information.
6. Check of agreements between orginating and home networks.

Authentication
6 & 7. Request for authentication d checking of blocking status, authentication data, through SDFo.
8 & 9. Response and authentication result.
10. Prompt 1 collect feature.

Figure 12: Authentication (use of the SDF-SDF interface)

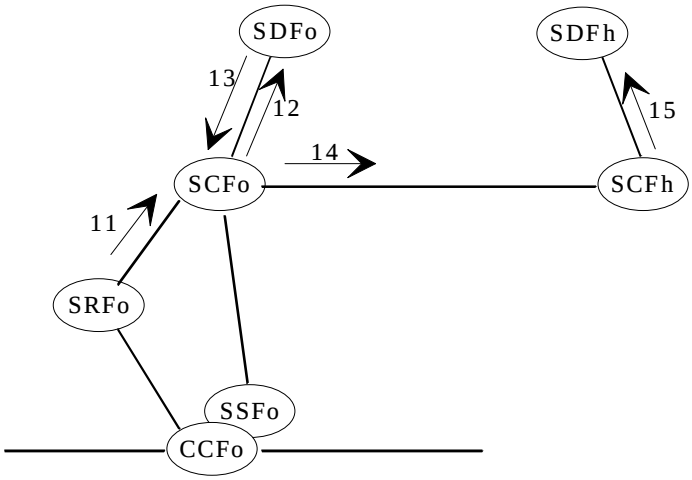
The difference with the use of the SCF-SCF interface is that SCFh does not have anything to do for authentication. This one is requested by SCFo (not in the home network). When no copy is made between SDFo and SDFh, the authentication is performed in the SDFh in the home network (see figure 12).

In the case when a copy of the service profile and authentication data is realized between SDFo and SDFh, then, the flows 7 and 8 above are used for this transfer. The authentication is then performed in the SDFo.

7.2 Registration for incoming calls

7.2.1 Use of the SCF-SCF interface

The process of registration for incoming calls is described by assuming that the initiation and authentication phases described in the previous subclause have been successfully performed.



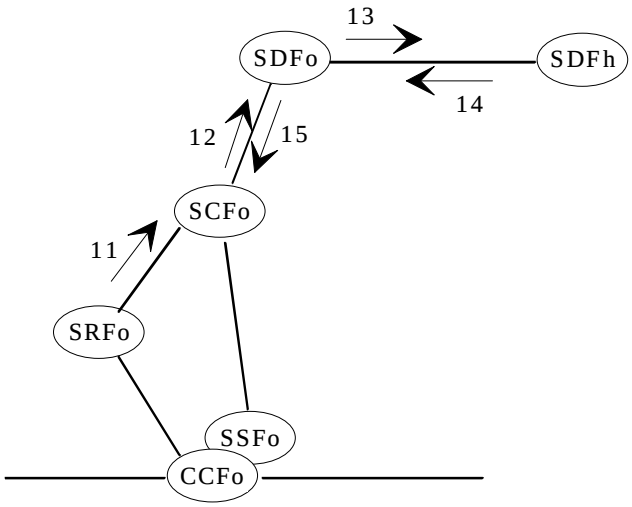
Registration
11. Collected user information.
12 & 13. Check of agreements between orginating and home networks.
14. Register.
15. Registration.

Figure 13: Registration for incoming calls (use of the SCF-SCF interface)

In this description, the access control on information can be done by the SDFh and the SCFh. It has an interest only if the registration (flow 15) shall be done in several passes. Otherwise, the current SCF-SDF interface appears more appropriate and needs less functionalities.

7.2.2 Use of the SDF-SDF interface

The process of registration for incoming calls is described by assuming that the initiation and authentication phases described in the previous subclause have been successfully performed.



Registration
11. Collected user information.
12. Check of agreements between orginating and home networks. Then, register.
13. Registration.
14 & 15. Acknowledge.

Figure 14: Registration for incoming calls (use of the SDF-SDF interface)

The access control on information is done only by the SDFh in this description.

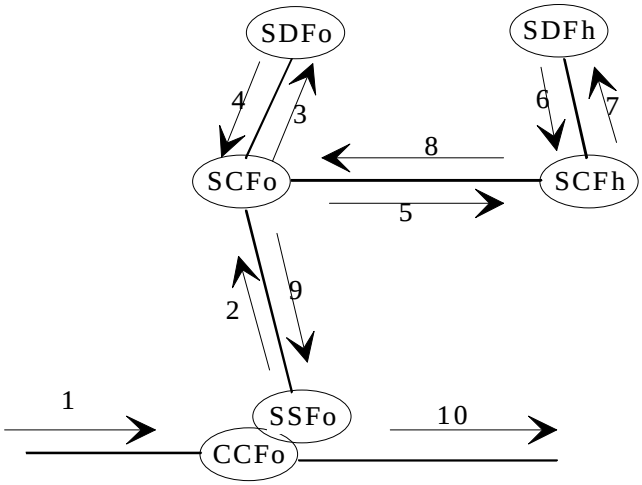
On the other hand, the transfer of information will be easier through databases (that is to say with the SDF-SDF interface).

The main interest of this description is the transparency from access: indeed, SCFo registers "in" SDFo but does not need to know exactly where data are stored: it could be in SDFh or in SDFo (when the profile has been transferred). SDFo is responsible for data storage and retrieval and is the only interlocutor to the SCFo.

7.3 Locate UPT user

7.3.1 Use of the SCF-SCF interface

The process is described in figure 15; it is assumed that a calling user has dialled the UPT number of a UPT user and that SSFo has handed over control to SCFo.



Incoming call
1. Dialled.
2. Provide instructions.
3 & 4. Check of agreements between originating and home networks.
5. Ask for query.
6. Query.
7. Response.
8. Response.
9. Connect.
10. IAM.

Figure 15: Locate UPT user (use of the SCF-SCF interface)

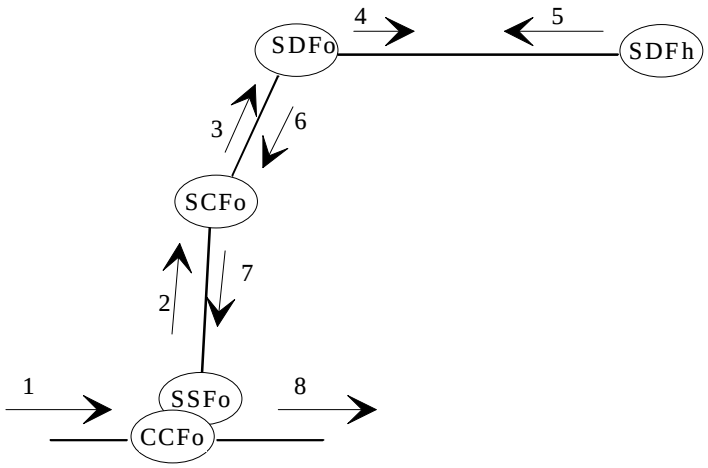
In order to locate the called UPT user, SCFo asks SCFh to query SDFh. The information found by SDFh is sent back to SCFh, then passed to SCFo. The search, made by SCFh, relieves SCFo to do this task.

This description has an interest only if the query to locate a UPT user shall be done in several passes. Else, the current SCF-SDF interface is more appropriate and needs less functionalities.

7.3.2 Use of the SDF-SDF interface

The process of locating a UPT user (for instance, during incoming call processing, to determine on which access(es) the call should be presented) may use the SDF-SDF interface. It is described (see figure 16) by assuming that a calling user has dialled the UPT number of a UPT user, and that SSFo has handed over control to SCFo.

In order to locate the called UPT user, SCFo queries SDFo. SDFo requests SDFh for information, receives the required information from SDFh and passes it to SCFo. This description relieves the SCF of the task to search for information itself, resulting in more efficient use of the SCF.



Incoming call
1. Dialled.
2. Provide instructions.
3. Check of agreements between originating and home networks. Then, query.
4. Search (get location).
5. Response.
6. Response.
7. Connect.
8. IAM.

Figure 16: Locate UPT user (use of the SDF-SDF interface)

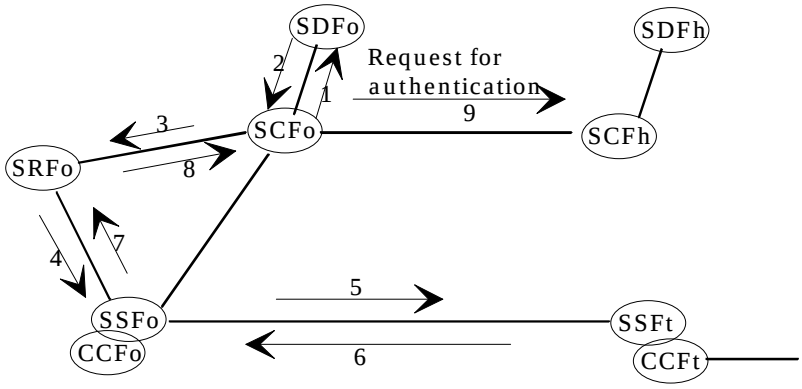
The main interest of this description is the transparency from access: indeed, SCFo registers "in" SDFo but does not need to know exactly where data are stored: it could be in SDFo or SDFh. SDFo is responsible for data storage and retrieval and is the only interlocutor to the SCFo.

7.4 Called party specified secure answering of incoming UPT calls

"*Called party specified secure answering of incoming UPT calls is a feature by which the called UPT user has specified that incoming UPT calls cannot be answered unless the answering party first successfully authenticates as the called user*." (ETR 055-02 [5]). This scenario is described by assuming that an incoming call has been initiated and that the database has already been interrogated to find the access(es) at which the UPT user is registered.

Authentication information are collected from the called user by the **originating** side (SCFo/SRFo).

7.4.1 Use of the SCF-SCF interface



1 & 2. Check agreements between originating and home networks (if not yet).
3, 4 & 5. Ask for Secure Answering Personal Identifier Number (SAPIN).
6, 7 & 8. Response.
9. Request.

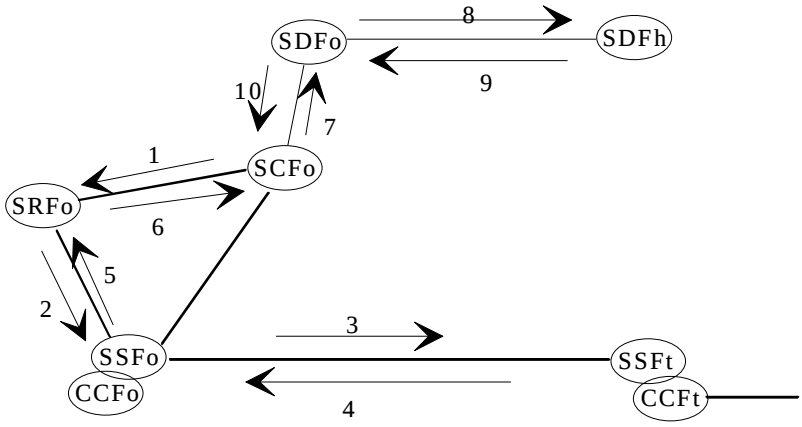
Figure 17: Secure answering of UPT calls

As in subclause 7.1.1, this scenario has an interest if the registration between SCFh and SDFh is done thanks to several passes. Besides, this scenario enables that the algorithm is performed by the SCFh.

The following comments apply to this procedure:

- the connection (which might be an expensive international link) between the originating and terminating sides is established from the beginning of the procedure;
- this connection can be charged only after a successful authentication.

7.4.2 Use of the SDF-SDF interface



1, 2 & 3. Ask for SAPIN.
4, 5 & 6. Response
7 & 8. Check of agreements (if not yet) and request for authentication.
9 & 10. Response.

Figure 18: Secure answering of UPT calls

This scenario has an interest only if the authentication of a UPT user (flow 7) shall be done several times in a short period of time. Then a data transfer could be done between SDFo and SDFh and the use of steps 8 and 9 avoided. Otherwise, the current SCF-SDF scenario is still appropriate.

The following comments apply to this procedure:

- the connection (which might be an expensive international link) between the originating and terminating sides is established from the beginning of the procedure;
- this connection can be charged only after a successful authentication.

7.5 Recall: Use of the CS1 interconnection scenario

7.5.1 Call handling

7.5.1.1 Outgoing UPT call originating in a UPT network

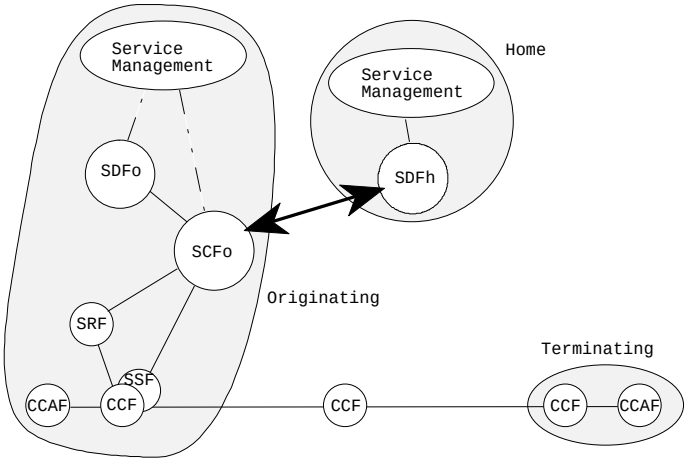


Figure 19: Outgoing UPT call originating in a UPT network

Since the originating network is a UPT network, the call can be handled locally (by SCFo) and optimal routing can be achieved.

7.5.1.2 Outgoing UPT call originating in a non-UPT network

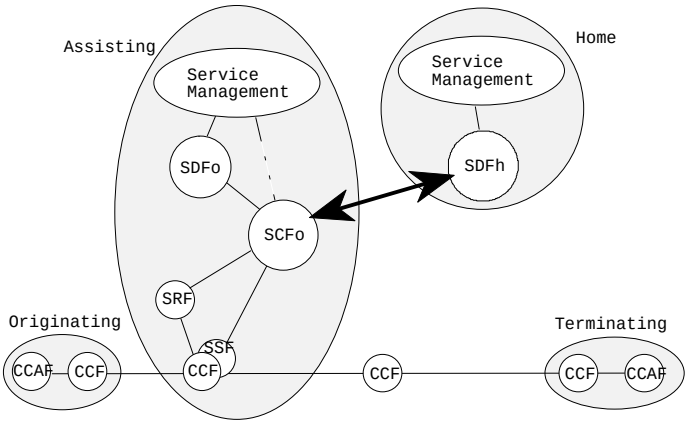


Figure 20: Outgoing UPT call originating in a non-UPT network

This scenario requires some co-operation between the originating (non-UPT) and the assisting (UPT) networks. The difference with the call from a UPT network is that routing may not be optimal and that tromboning can take place if the UPT user is calling someone who is also in the originating network.

7.5.1.3 Incoming UPT call originating in a UPT network

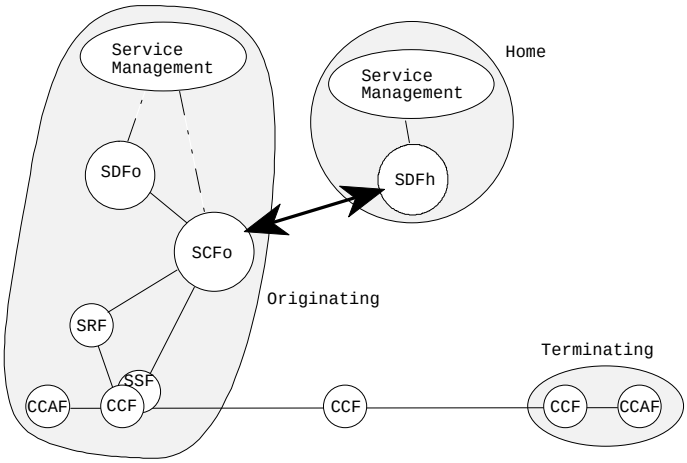


Figure 21: Incoming UPT call originating in a UPT network

Since the originating network is a UPT network, the call can be handled locally (by SCFo) and optimal routing can be achieved.

7.5.1.4 Incoming UPT call originating in a non-UPT network

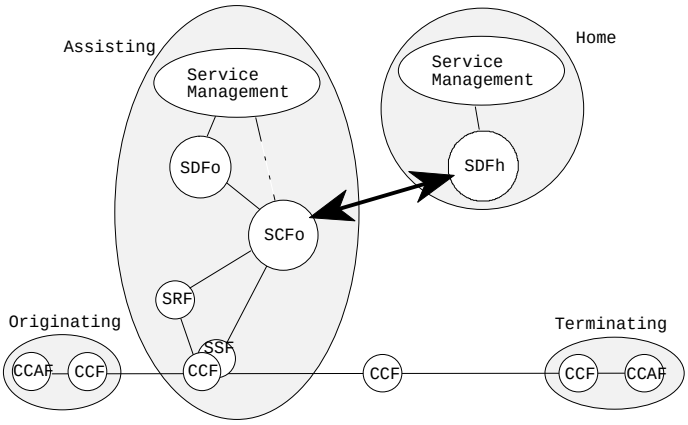


Figure 22: Incoming UPT call originating in a non-UPT network

7.5.1.5 UPT user to UPT user call

The SCFo will likely communicate with both the caller's SDFh and the called's SDFh.

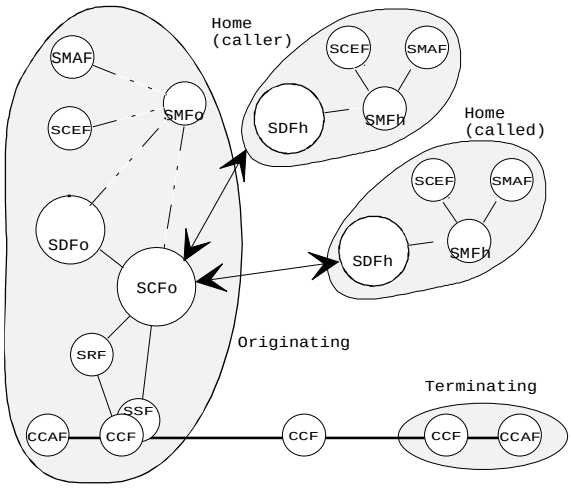


Figure 23: UPT user to UPT user call

7.5.2 Registration and service management

7.5.2.1 From a UPT network

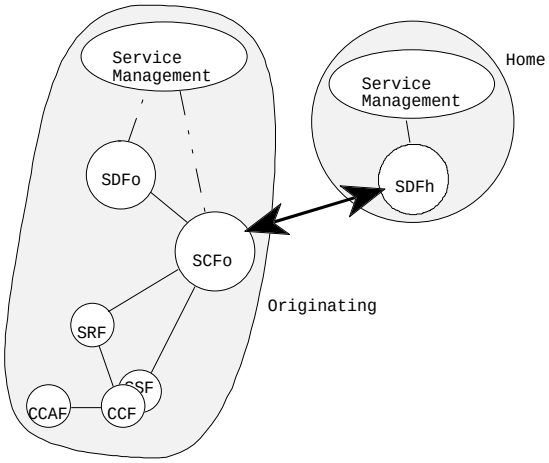


Figure 24: Registration and service management from a UPT network

Only two networks are involved in this scenario. Call handling is performed locally (by SCFo), but registrations and the service profile are modified in the home network (by SDFh).

7.5.2.2 From a non-UPT network

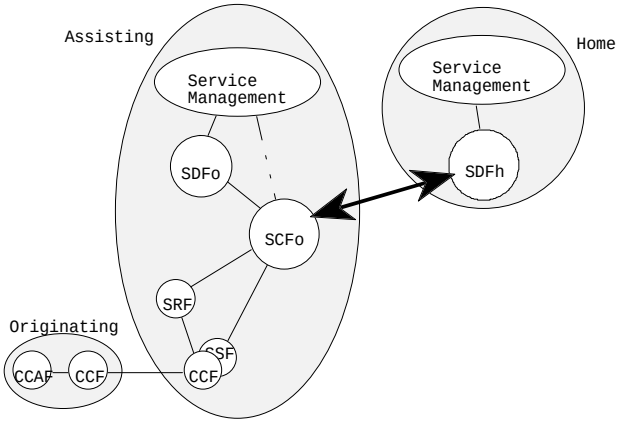


Figure 25: Registration and service management from a non-UPT network

Three networks are involved in this scenario: registration handling is not local, as it is performed by the assisting network (where the SCFo is located). A normal (non-UPT) call therefore needs to be set up from the UPT user's Access Point in the non-UPT network, to the assisting network. The branch of the call going from the originating to the assisting network is not seen as a UPT call and is therefore subject to different charging principles (it may be free).

7.5.2.3 Using IN Service Management procedures

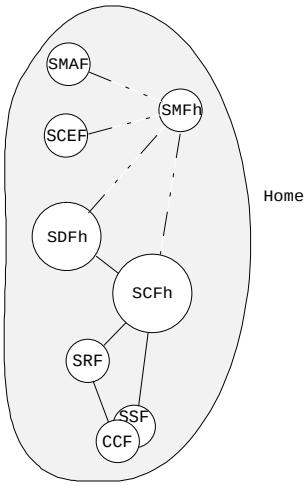


Figure 26: Registration and service management using IN Service Management procedures

Only one network is involved in this scenario. Moreover, not all entities in this network are used, as only SMAF, SMFh and SDFh are used to let the UPT user interact with the network in order to query and modify his service profile.

8 Interworking with non-UPT networks

Although the ultimate aim is to provide UPT functionalities in all networks, it shall be recognized that this will not be feasible in the near future (especially not for UPT Phase 1). UPT networks (that is, networks which provide UPT functionalities) and non-UPT networks will therefore coexist for some time. On the other hand, even for UPT Phase 1, user mobility should not be restricted to a small set of countries (for example, those which have an IN-structured network) or to certain types of terminals (for example, PSTN but not GSM).

Some networks will therefore not be able to offer full UPT support, but it may still be desirable that UPT users be able to use part of the functionalities of UPT when they visit these networks. This is possible, if the aid of another network with UPT capabilities, the assisting network, is sought. The corresponding architecture is shown on the figure below.

A UPT user visiting the non-UPT network could then use the UPT facilities of the assisting network to access the UPT service.

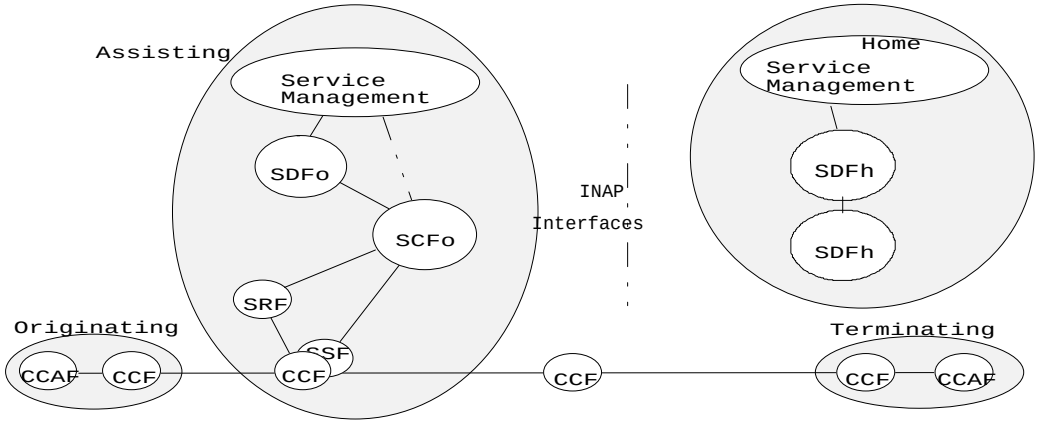


Figure 27: Interworking with non-UPT networks

In order to achieve this, agreements between the operator of the non-UPT network and the home/assisting network are needed, as we have to consider a non-standardized environment: the lack of standards should be compensated with bilateral agreements. As these considerations are out of the scope of the present document, a more detailed description is given below.

Also, numbering and routing aspects may have some implications on the way of interworking with non-UPT networks. See ETR 067 [6] for more information.

Methods for interworking with non-UPT networks

As stated above, an issue that needs consideration for the short term provision of the UPT service is the interworking with non-UPT networks, since the service will not be introduced all over the world at the same time. Therefore, at least for the initial phase, there is a need to consider both UPT areas (where users can subscribe to and access the UPT service), and non-UPT areas (where there are no specific network functionalities for the provision of the UPT service).

The simplest solution would be to rule that UPT users cannot access the service while in non-UPT areas. But this is not the only solution, if interworking with an assisting UPT network is possible. Such interworking will be based on bilateral agreements between the non-UPT network operator and the assisting UPT operator. Different levels of agreements between the operators are possible and they can impose some limitation to the UPT service provided or may require specific access procedures.

In the following, three methods are described, showing how different levels of agreement can influence the grade of service provided.

Method 1

The non-UPT network only accepts incoming calls to UPT users that have previously registered from another network, on an Access Point of the non-UPT network. No registration is possible from the non-UPT network. This grade of service is always available from any non-UPT network, since it does not require any extra effort from the network operators, either on the UPT or on the non-UPT side.

Nevertheless, a check on the number provided by the user should be performed by the originating network during a remote registration, in order to avoid that the number used for the registration be a special number (e.g. an emergency number). The originating network would then have to perform a check on a number of the non-UPT network, which would require a bilateral agreement between the two operators, as the format of these numbers differs from one network to an other.

Method 2

The non-UPT network accepts incoming calls to UPT users. In addition, a UPT user can perform a registration for incoming calls and other simple service profile management operations by dialling a special "UPT access" number in the assisting network. From the point of view of the originating, non-UPT network, this is an ordinary international call and

will be charged as such (the non-UPT network cannot avoid charging this call). The assisting network may charge the call as appropriate.

If this simple solution is used, it is not possible to let the UPT user place outgoing UPT calls using the facilities of the assisting network, as part of the call (the call from the non-UPT network to the assisting UPT network) would be charged to the owner of the access used, which is not UPT service any longer. While this deviation from the UPT definition may be acceptable in the case of simple registration and service profile management operations (because of the small cost involved and the benefit derived from it), it is not acceptable for outgoing UPT calls, however, and an agreement between the two networks is required, in order to refuse requests for outgoing UPT calls.

Method 3

Full UPT service is available to the UPT user, by either:

1) dialling a special number supplied by the non-UPT network;

2) dialling a freephone "UPT access" number in the non-UPT network, which is supplied by the assisting network.

The call is then redirected to the UPT facilities of the assisting network, without the need for the UPT user to be aware of it. Appropriate charging may be applied by both networks. An agreement is required between the two operators in the first scenario, while none is required in the second one (the assisting network supplies a true UPT service by offering freephone access to its network; the costs incurred may of course be charged back to the account of the UPT user).

Note that in the three methods, the fact of providing UPT service in a non-UPT network may result in inefficiencies and non-optimal use of network resources. This may be the case for the routing of UPT incoming calls: if the calling party is also in a non UPT network, tromboning can take place; the calling user will be charged according to the country code of the UPT number and the called UPT user will pay the rest of the call. The use of methods 2 or 3 may have other consequences:

- Since the UPT user sets up a call to an other network in order to access all or part of the UPT service, the procedures and the dialogues involved may have to be different from the normal procedures.

- The UPT user is not recognized as such by the non-UPT network. Conflicts with the features attached to the Access Point he uses in the non-UPT network may therefore occur (e.g. for restrictions, call forwarding, etc.).

NOTE: Registration for outgoing UPT calls explicitly requires UPT capabilities in the network where it takes place, since it puts some requirements on its local exchanges. The solution described here will therefore not give full access to UPT service in non-UPT networks beyond Phase 1, since registration for outgoing calls will then be an integral part of UPT. It is, however, an appropriate solution for Phase 1, since registration for outgoing calls has been excluded from it.

9 Handling of UPT data

1) Per enquiry, similar to the existing INAP CS1, no data are stored in the visited network. Location Update will need to store the Visited Network Identity upon Location Updating. The Routing Number strategy is therefore independent from this storage strategy. **This is highly inefficient as very many messages are sent between networks**.

2) Per call, temporary data are requested/retrieved and stored in the visited network throughout the call only; and is deleted at the end of the call. Location Update will need to store the Visited Network Identity upon Location Updating. The Routing Number strategy is therefore independent from this storage strategy. **This is highly efficient for rapid movement and a low calling rate but may give the visited network only limited control of the visitors access, like difficulty when resetting the network Access Point**.

3) Per location registration, the down loading of a portion of the basic service data and authentication parameters (an efficient subset, let to service providers choice) throughout the visit; the complementary data from the service profile being sent to the visited network for storage throughout the call or service request. This strategy will require data management function to track and handle the deletion of redundant data. **This may be difficult to standardize, who will agree which data are an efficient subset; this will also require both types of protocol, "in call" and "at registration" and so represent the worst of both worlds**.

4) Per location registration, the down loading of a basic service data or a part of the service profile (fixed "in advance") to the visited network for storage throughout the visit. This strategy is like GSM and will require data management function to track and handle the deletion of redundant data. **This is efficient, for a high calling likelihood and low rate movement; as large quantities of non-call related data are delivered and stored in a number of networks, irrespective of the likelihood of calls and service usage**.

The cases 2, 3 and 4 will use CS2 feature internetwork service profile transfer mechanism to distribute the data.

10 Evolution from Phase 1

From the architecture point of view, the main evolution from Phase 1 is the possibility to use the interconnection scenarios described in Capability Set 2.

The consequences of the SCF-SCF scenario and of the improvements of CS2 are the following ones:

- the new roles of the SDFs and SCFs: control of access, transfer of data (authentication, user profile etc.), depending on the scenario used;
- the introduction of new functions to deal with the above consequences;
- SCFh could be used as an assisting SCF to help SCFo to offer specific home services.

It can be noted that the SCF-SCF scenario is more relevant for service logic assisting.

The consequences of the SDF-SDF scenario and of the improvements of CS2 are the following ones:

- the possibility to realize transfers of data between the SDFs. Sharing data can be done according to several manners (see clause 7);
- the new roles of the SDFs and SCFs: control of access, transfer of data (authentication, user profile etc.), depending on the scenario used;
- the introduction of new functions to deal with the above consequences.

It can be noted that the SDF-SDF scenario will be more relevant within large networks (with distributed databases).

History

Document history				
V1.1.1	June 1997	Membership Approval Procedure	MV 9734:	1997-06-24 to 1997-08-22