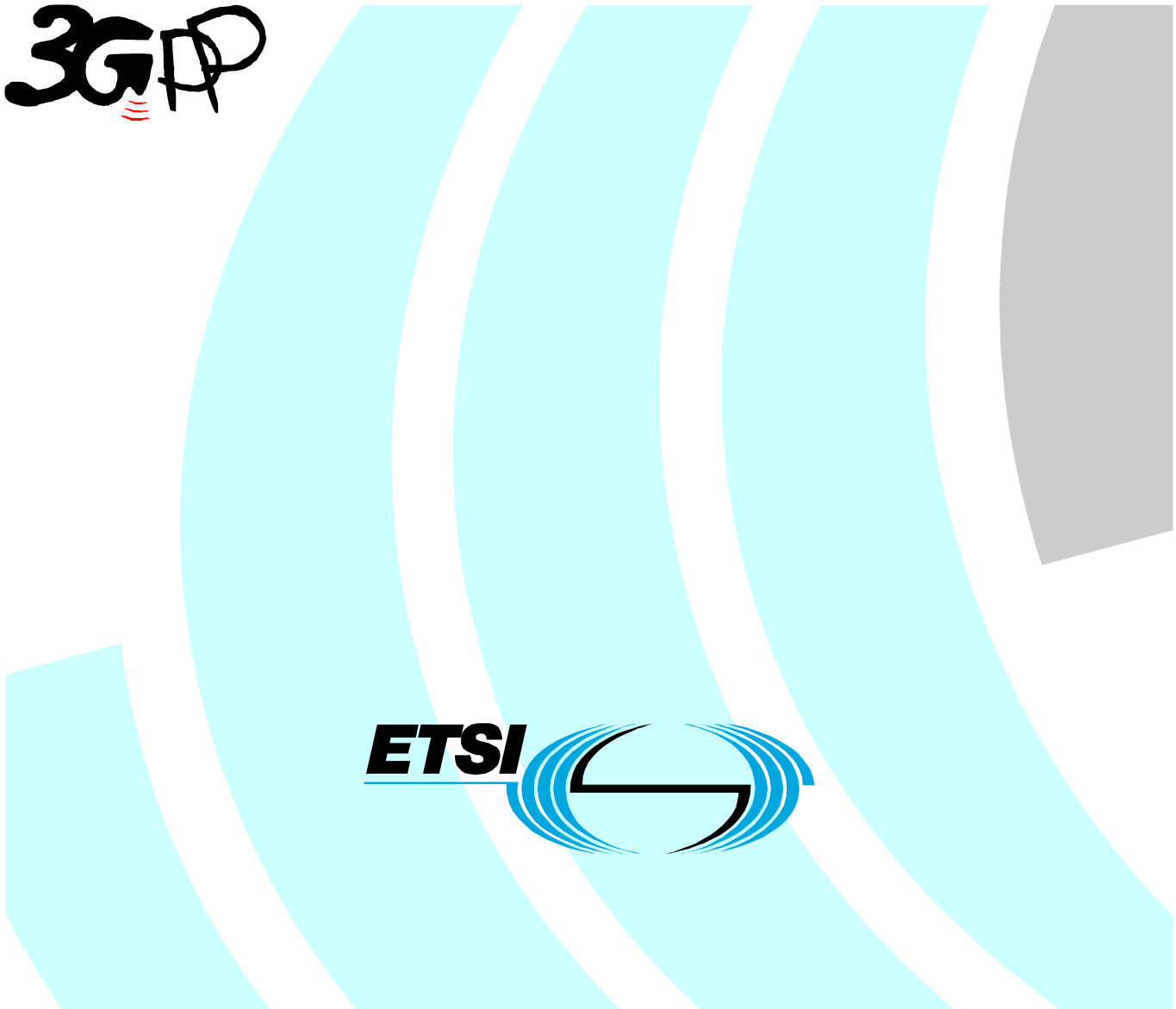
ETSI TS 123 127 V4.3.0 (2002-03)

Technical Specification

Universal Mobile Telecommunications System (UMTS);
Virtual Home Environment (VHE) / Open Service Access (OSA);
Stage 2
(3GPP TS 23.127 version 4.3.0 Release 4)



Reference
RTS/TSGS-0223127Uv4R2

Keywords
UMTS

ETSI

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

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

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

Intellectual Property Rights

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

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

Foreword

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

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

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under www.etsi.org/key .

Contents

Foreword

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

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

Version x.y.z

where:

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

1 Scope

The present document specifies the stage 2 of the Virtual Home Environment.

Virtual Home Environment (VHE) is defined as a concept for Personal Service Environment (PSE) portability across network boundaries and between terminals. The concept of VHE is such that users are consistently presented with the same personalised features, User Interface customisation and services in whatever network and whatever terminal (within the capabilities of the terminal and the network), wherever the user may be located.

For Release 4, e.g. CAMEL, MExE, OSA and USAT are considered the mechanisms supporting the VHE concept.

Stage 2 specifications for CAMEL, MExE and USAT are addressed in other TS documents. However, there is no separate stage 2 specification document for OSA. Therefore, the present document addresses stage 2 aspects for OSA.

2 References

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

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

[1] GSM 01.04: "Abbreviations and Acronyms".

[2] 3G TS 22.057: "Digital cellular telecommunication system (Phase 2+); Mobile Execution Environment (MExE); Service description".

[3] 3G TS 23.057: "Mobile Execution Environment (MExE); Functional description - Stage2".

[4] 3G TS 22.078: "Customised Applications for Mobile network Enhanced Logic (CAMEL); Service description - Stage 1".

[5] 3G TS 23.078: "Customised Applications for Mobile network Enhanced Logic (CAMEL) (Phase3); Stage 2".

[6] 3G TS 21.111: "USIM and IC card requirements".

[7] 3G TS 22.101: "Universal Mobile Telecommunications System (UMTS); Service aspects; Service principles".

[8] 3G TS 22.105: "Universal Mobile Telecommunications System (UMTS); Services and Service Capabilities".

[9] 3G TS 22.121: "Universal Mobile Telecommunications System (UMTS); Service aspects; The Virtual Home Environment; Stage 1".

[10] 3G TR 21.905: "Universal Mobile Telecommunications System (UMTS); Vocabulary for 3GPP Specifications".

[11] RFC1994 (1996): "PPP Challenge Handshake Authentication Protocol (CHAP)".

[12] World Wide Web Consortium (W3C) Composite Capability/Preference Profiles (CC/PP): "A user side framework for content negotiation".

[13] Wireless Application Protocol: "User Agent Profiling Specification". (http://www.wapforum.org/).

[14] Object Management Group: "The Complete formal/99-10-07 CORBA/IIOP 2.3.1 Specification" (http://cgi.omg.org/corba/corbaiiop.html).

[15] 3G TS 22.127: "Service Requirement for the Open Service Access (OSA); Stage 1".

[16] World Wide Web Consortium (W3C): "Simple Object Access Protocol (SOAP) 1.1" (http://www.w3.org/TR/2000/NOTE-SOAP-20000508/).

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3G TS 22.101 and 3G TR 22.905 and the following apply:

Applications: software components providing services to end-users by utilising service capability features

HE-VASP: See [9].

Home Environment: responsible for overall provision of services to users

Interface: listing and semantics of the methods and attributes provided by an object that belongs to a Service Capability Feature

Local Service: See[9].

OSA API: standardised API used by applications to access service capability features

OSA Internal API: standardised API between framework and service capability servers

Personal Service Environment: contains personalised information defining how subscribed services are provided and presented towards the user

NOTE: The Personal Service Environment is defined in terms of one or more User Profiles.

Service Capabilities: See [15].

Service Capability Feature: See [15].

Service Capability Server: Functional Entity providing OSA interfaces towards an application

Services: See [9].

User Interface Profile: See [9].

User Profile: See [9].

User Services Profile: See [9].

Value Added Service Provider: See [9].

Virtual Home Environment: See [9].

3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in GSM 01.04 and in 3G TR 21.905 and the 7following apply:

API	Application Programming Interface
CAMEL	Customised Application For Mobile Network Enhanced Logic
CSE	Camel Service Environment
HE	Home Environment
HE-VASP	Home Environment Value Added Service Provider
HLR	Home Location Register
IDL	Interface Description Language
MAP	Mobile Application Part
ME	Mobile Equipment
MExE	Mobile Execution Environment
MS	Mobile Station
MSC	Mobile Switching Centre
OSA	Open Service Access
PLMN	Public Land Mobile Network
PSE	Personal Service Environment
SCF	Service Capability Feature
SCS	Service Capability Server
SIM	Subscriber Identity Module
SOAP	Simple Object Access Protocol
USAT	Universal SIM Application Tool-Kit
USIM	Universal Subscriber Identity Module
VASP	Value Added Service Provider
VHE	Virtual Home Environment
WGW	WAP Gateway
WPP	WAP Push Proxy

4 Virtual Home Environment

The Virtual Home Environment (VHE) is an important portability concept of the 3G mobile systems. It enables end users to bring with them their personal service environment whilst roaming between networks, and also being independent of terminal used.

The Personal Service Environment (PSE) describes how the user wishes to manage and interact with her communication services. It is a combination of a list of subscribed to services, service preferences and terminal interface preferences. PSE also encompasses the user management of multiple subscriptions, e.g. business and private, multiple terminal types and location preferences. The PSE is defined in terms of one or more User Profiles.

Please see TS22.121 [9] for more details.

5 Open Service Access

In order to be able to implement future applications/end user services that are not yet known today, a highly flexible Framework for Services is required. The Open Service Access (OSA) enables applications implementing the services to make use of network functionality. Network functionality offered to applications is defined in terms of a set of Service Capability Features (SCFs). These SCFs provide functionality of network capabilities which is accessible to applications through the standardised OSA interface upon which service developers can rely when designing new services (or enhancements/variants of already existing ones).

The aim of OSA is to provide a standardised, extendible and scalable interface that allows for inclusion of new functionality in the network in future releases with a minimum impact on the applications using the OSA interface.

Network functionality offered to applications is defined as a set of Service Capability Features (SCFs) in the OSA API, which are supported by different Service Capability Servers (SCS). These SCFs provide access to the network capabilities on which the application developers can rely when designing new applications (or enhancements/variants of

already existing ones). The different features of the different SCSs can be combined as appropriate. The exact addressing (parameters, type and error values) of these features is described in stage 3 descriptions. These descriptions (defined using OMG Interface Description Language™) are open and accessible to application developers, who can design services in any programming language, while the underlying core network functions use their specific protocols.

The standardised OSA API shall be secure, it is independent of vendor specific solutions and independent of programming languages, operating systems etc used in the service capabilities. Furthermore, the OSA API is independent of the location within the home environment where service capabilities are implemented and independent of supported service capabilities in the network.

To make it possible for application developers to rapidly design new and innovative applications, an architecture with open interfaces is imperative. By using object-oriented techniques, for example CORBA, SOAP, etc., it is possible to use different operating systems and programming languages in application servers and service capability servers. The service capability servers serve as gateways between the network entities and the applications.

The OSA API is based on lower layers using main stream information technology and protocols. The middleware and protocols (for example CORBA/IIOP, SOAP/XML, other XML based protocols etc.) and lower layer protocols (for example TCP, IP, etc.) should provide security mechanisms to encrypt data (for example TLS, IP sec, etc.).

5.1 Overview of the Open Service Access

The Open Service Access consists of three parts:

- **Applications:** e.g. VPN, conferencing, location based applications. These applications are implemented in one or more Application Servers;

- **Framework:** providing applications with basic mechanisms that enable them to make use of the service capabilities in the network. Examples of framework functions are Authentication and Discovery. Before an application can use the network functionality made available through Service Capability Features, authentication between the application and framework is needed. After authentication, the discovery function enables the application to find out which network service capability features are provided by the Service Capability Servers. The network service capability features are accessed by the methods defined in the OSA interfaces;

- **Service Capability Servers:** providing the applications with service capability features, which are abstractions from underlying network functionality. Examples of service capability features offered by the Service Capability Servers are Call Control and User Location. Similar service capability features may possibly be provided by more than one Service Capability Server. For example, Call Control functionality might be provided by SCSs on top of CAMEL and MExE.

The OSA service capability features are specified in terms of a number of interfaces and their methods. The interfaces are divided into two groups:

- framework interfaces;

- network interfaces.

NOTE: The CAMEL Service Environment does not provide the service logic execution environment for applications using the OSA API, since these applications are executed in Application Servers.

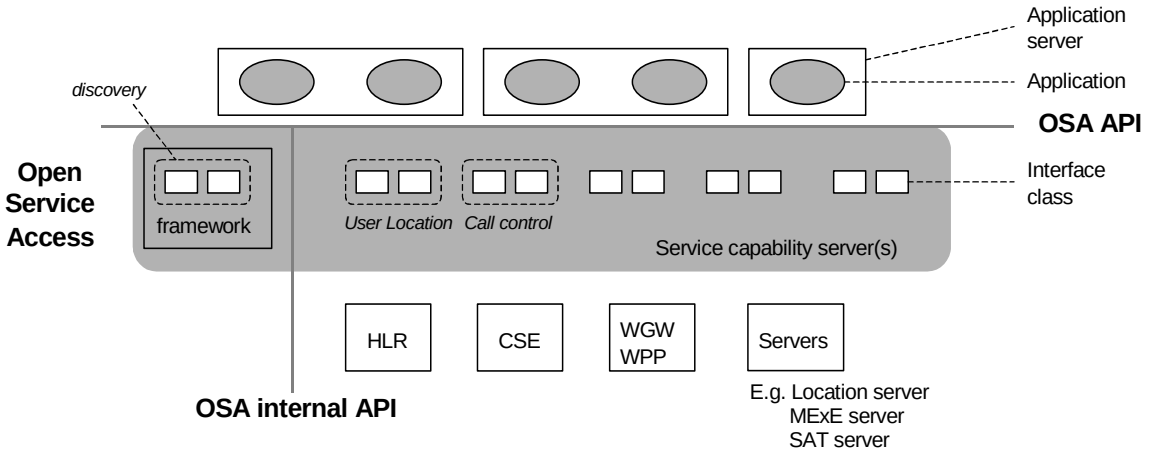


Figure 1: Overview of Open Service Access

The present document, together with the associated stage 3 specification, defines the OSA API and the OSA internal API between the framework and the service capability servers. OSA does not mandate any specific platform or programming language.

The Service Capability Servers that provide the OSA interfaces are functional entities that can be distributed across one or more physical nodes. For example, the User Location interfaces and Call Control interfaces might be implemented on a single physical entity or distributed across different physical entities. Furthermore, a service capability server can be implemented on the same physical node as a network functional entity or in a separate physical node. For example, Call Control interfaces might be implemented on the same physical entity as the CAMEL protocol stack (i.e. in the CSE) or on a different physical entity.

Several options exist:

Option 1

The OSA interfaces are implemented in one or more physical entity, but separate from the physical network entities. Figure 2 shows the case where the OSA interfaces are implemented in one physical entity, called "gateway" in the figure. Figure 3 shows the case where the SCSs are distributed across several "gateways".

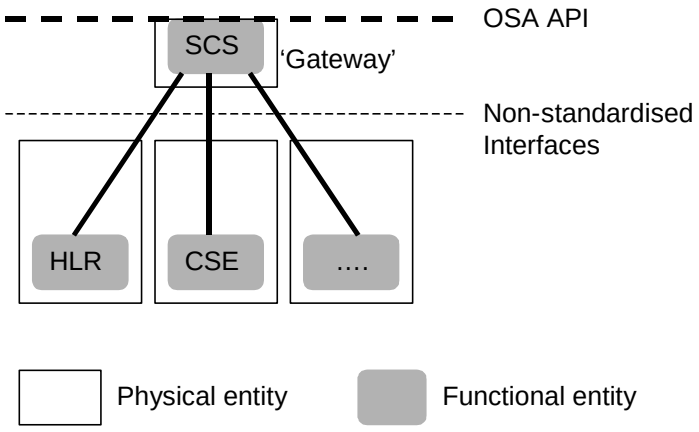


Figure 2: SCSs and network functional entities implemented in separate physical entities

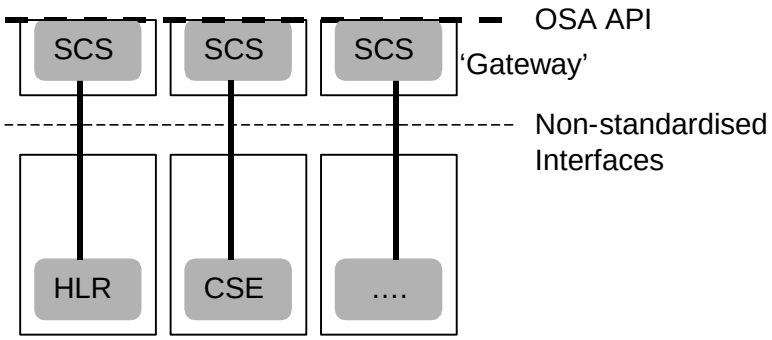


Figure 3: SCSs and network functional entities implemented in separate physical entities, SCSs distributed across several 'gateways'

Option 2

The OSA interfaces are implemented in the same physical entities as the traditional network entities (e.g. HLR, CSE), see figure 4.

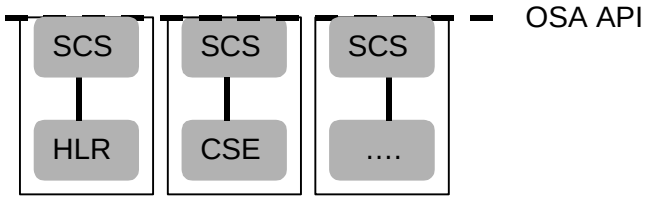


Figure 4: SCSs and network functional entities implemented in same physical entities

Option 3

Option 3 is the combination of option 1 and option 2, i.e. a hybrid solution.

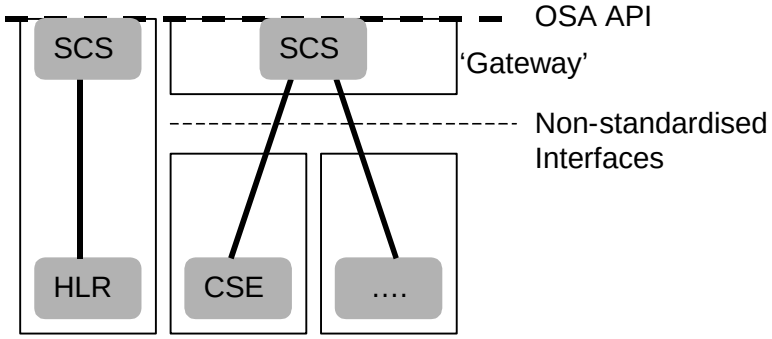


Figure 5: Hybrid implementation (combination of option 1 and 2)

It shall be noted that in all cases there is only one framework. This framework may reside within one of the physical entities containing an SCS or in a separate physical entity.

From the application point of view, it shall make no difference which implementation option is chosen, i.e. in all cases the same network functionality is perceived by the application. The applications shall always be provided with the same set of interfaces and a common access to framework and service capability feature interfaces. It is the framework that will provide the applications with an overview of available service capability features and how to make use of them.

5.2 Basic mechanisms in the Open Service Access

This subclause explains which basic mechanisms are executed in OSA prior to offering and activating applications.

Some of the mechanisms are applied only once (e.g. establishment of service agreement), others are applied each time a user subscription is made to an application (e.g. enabling the call attempt event for a new user).

Basic mechanisms between Application and Framework:

- **Authentication:** Once an off-line service agreement exists, the application can access the authentication function. The authentication model of OSA is a peer-to-peer model. The application must authenticate the framework and vice versa. The application must be authenticated before it is allowed to use any other OSA function.

- **Authorisation:** Authorisation is distinguished from authentication in that authorisation is the action of determining what a previously authenticated application is allowed to do. Authentication must precede authorisation. Once authenticated, an application is authorised to access certain service capability features.

- **Discovery of framework functions and network service capability features:** After successful authentication, applications can obtain available framework functions and use the discovery function to obtain information on authorised network service capability features. The Discovery function can be used at any time after successful authentication.

- **Establishment of service agreement:** Before any application can interact with a network service capability feature, a service agreement must be established. A service agreement may consist of an off-line (e.g. by physically exchanging documents) and an on-line part. The application has to sign the on-line part of the service agreement before it is allowed to access any network service capability feature.

- **Access to network service capability features:** The framework must provide access control functions to authorise the access to service capability features or service data for any API method from an application, with the specified security level, context, domain, etc.

Basic mechanism between Framework and Service Capability Server:

- **Registering of network service capability features**. SCFs offered by a Service Capability Server can be registered at the Framework. In this way the Framework can inform the Applications upon request about available service capability features (Discovery). For example, this mechanism is applied when installing or upgrading a Service Capability Server.

Basic mechanisms between Application Server and Service Capability Server:

- **Request of event notifications**. This mechanism is applied when a user has subscribed to an application and that application needs to be invoked upon receipt of events from the network related to the user. For example, when a user subscribes to an incoming call screening application, the application needs to be invoked when the user receives a call. It will therefore request to be notified when a call setup is performed, with the user number as Called Party Number.

5.3 Handling of end-user related security

Once OSA basic mechanisms have ensured that an application has been authenticated and authorised to use network service capability features, it is important to also handle end-user related security aspects. These aspects consist of the following.

- End-user authorisation to applications, limiting the access of end-users to the applications they are subscribed to.

- Application authorisation to end-users, limiting the usage by applications of network capabilities to authorised (i.e. subscribed) end-users.

- End-user's privacy, allowing the user to set privacy options.

These aspects are addressed in the following subclauses.

5.3.1 End-user authorisation to applications

An end-user is authorised to use an application only when he or she is subscribed to it.

In the case where the end-user has subscribed to the application before the application accesses the network SCFs, then the subscription is part of the Service Level Agreement signed between the HE and the HE-VASP.

After the application has been granted access to network SCFs, subscriptions are controlled by the Home Environment. Depending on the identity of an authenticated and authorised end-user, the Home Environment may use any relevant policy to define and possibly restrict the list of services to which a particular end-user can subscribe. At any time, the Home Environment may decide, unilaterally or after agreement with the HE-VASP, to cancel a particular subscription.

Service subscription and activation information need to be shared between the Home Environment and the HE-VASP, so that the HE-VASP knows which end-users are entitled to use its services. Appropriate online and/or offline synchronisation mechanisms (e.g. SLA re-negotiation) can be used between the HE and the HE-VASP, which are not specified in OSA release 4.

End-to-end interaction between a subscribed end-user and an application may require the usage of appropriate authentication and authorisation mechanisms between the two, which are independent from the OSA API, and therefore not in the scope of OSA standardisation.

5.3.2 Application authorisation to end-users

The Home Environment is entitled to provide service capabilities to an application with regard to a specific end-user if the following conditions are met:

1) the end-user is subscribed to the application;

2) the end-user has activated the application;

3) the usage of this network service capability does not violate the end-users privacy settings (see next subclause).

The service capability server ensures that the above conditions are met whenever an application attempts to use a service capability feature for a given end-user, and to respond to the application accordingly, possibly using relevant error parameters). The mechanism used by the SCS to ensure this is internal to the HE (e.g. access to user profile) and is not standardised in OSA release 4.

5.3.3 End-user's privacy

The Home Environment may permit an end-user to set privacy options. For instance, it may permit the end-user to decide whether his or her location may be provided to 3rd parties, or whether he or she accepts information to be pushed to his or her terminal. Such privacy settings may have an impact on the ability of the network to provide service capability features to applications (e.g. user location, user interaction). Thus, even if an application is authorised to use an SCF and the end-user is subscribed to this application and this application is activated, privacy settings may still prevent the HE from fulfilling an application request.

The service capability server ensures that a given application request does not violate an end-users privacy settings or that the application has relevant privileges to override them (e.g. for emergency reasons). The mechanism used by the SCS to ensure this is internal to the HE and is not standardised in OSA release 4.

6 Framework service capability features

6.1 Trust and Security Management Functions

The Trust and Security Management functions provide:

- the first point of contact for an application to access a Home Environment;
- the authentication methods for the application and Home Environment to perform an authentication protocol;
- the application with the ability to select a network service capability feature to make use of;
- the application with a portal to access other framework functions.

The process by which the application accesses the Home Environment has been separated into 3 stages, each supported by a different framework function:

1) Initial Contact with the framework;
2) Authentication to the framework;
3) Access to framework functions and network service capability features.

6.1.1 Initial Contact

The application gains a reference to the Initial Contact function for the Home Environment that they wish to access. This may be gained through a URL, a Naming or Trading Service or an equivalent service, a *stringified* object reference, etc. At this stage, the application has no guarantee that this is a reference to the Home Environment.

The application uses this reference to initiate the authentication process with the Home Environment.

Initial Contact supports a particular method to allow the authentication process to take place (using the Authentication SCF defined in subclause 6.1.2). This method must be the first invoked by the application. Invocations of other methods will fail until authentication has been successfully completed.

Once the application has authenticated with the provider, it can gain access to other framework functions and network service capability features. This is done by invoking a method, by which the application requests a certain type of access service capability feature. The OSA Access function is defined in subclause 6.1.3.

6.1.2 Authentication

Once the application has made initial contact with the Home Environment, authentication of the application and Home Environment may be required.

The API supports multiple authentication techniques. The procedure used to select an appropriate technique for a given situation is described below. The authentication mechanisms may be supported by cryptographic processes to provide confidentiality, and by digital signatures to ensure integrity. The inclusion of cryptographic processes and digital signatures in the authentication procedure depends on the type of authentication technique selected. In some cases strong authentication may need to be enforced by the Home Environment to prevent misuse of resources. In addition it may be necessary to define the minimum encryption key length that can be used to ensure a high degree of confidentiality.

The application must authenticate with the framework before it is able to use any of the other interfaces supported by the framework. Invocations on other interfaces will fail until authentication has been successfully completed.

6.1.3 OSA Access

This function supports stage 1 requirements related to authorization and service registration.

During an authenticated session accessing the Framework, the application will be able to select and access an instance of a framework function or network service capability feature.

In order to use network SCFs, the application must first be authorised to do so by establishing a service agreement with the Home Environment. The application uses the discovery SCF to retrieve the ID of the network SCF they wish to use.They may then check that they are authorised to use the SCF. The Home Environment is informed that the application wishes to use the SCF. Finally, a service agreement is signed digitally between the two parties.

Establishing a service agreement is a business level transaction, which requires the HE-VASP that owns the application to agree terms for the use of an SCF with the Home Environment. Service agreements can be reached using either off-line or on-line mechanisms. Off-line agreements will be reached outside of the scope of OSA interactions, and so are not described here. However, applications can make use of service agreements that are made off-line. Some Home Environments may only offer off-line mechanisms to reach service agreements.

After a service agreement has been established between the application and the Home Environment domains, the application will be able to make use of this agreement to access the SCF.

6.2 Discovery

Before a network SCF can be discovered, the application must know what "types" of SCFs are supported by the Framework and what "properties" are applicable to each SCF type. Once the HE-VASP finds out the desired set of SCFs supported by the network, it subscribes (a sub-set of) these SCFs using the Subscription framework function. The HE-VASP (or the applications in its domain) can find out the set of SCFs available to it (i.e., the SCFs that it can use).

6.3 Integrity Management functions

6.3.1 Load Manager

The Load Manager function permits to manage the load on both the application and network sides.

The framework API should allow the load to be distributed across multiple machines and across multiple component processes, according to a load balancing policy. The separation of the load balancing mechanism and load balancing policy ensures the flexibility of the load balancing functionality. The load balancing policy identifies what load balancing rules the framework should follow for the specific application. It might specify what action the framework should take as the congestion level changes. For example, some real-time critical applications will want to make sure continuous service is maintained, below a given congestion level, at all costs, whereas other applications will be satisfied with disconnecting and trying again later if the congestion level rises. Clearly, the load balancing policy is related to the QoS level to which the application is subscribed.

6.3.2 Fault Manager

The Fault Manager function is used by the application to inform the framework of events which affect the integrity of the framework and SCFs, and to request information about the integrity of the system.

6.3.3 Heartbeat Management

The Heartbeat Management function allows the initialisation of a heartbeat supervision of the client application. In case of SCF supervision, it is the framework's responsibility to check the health status of the respective SCF.

Since the OSA API is inherently synchronous, the heartbeats themselves are synchronous for efficiency reasons.

6.3.4 OAM

The OAM function is used to query the system date and time. The application and the framework can synchronise the date and time to a certain extent. Accurate time synchronisation is outside the scope of the OSA API.

7 Network service capability features

Network service capability features are provided to the applications by service capability servers to enable access to network resources.

7.1 Call Control

The Call Control SCF supports stage 1 requirements related to CS call control and call charging.

The Call control network service capability feature supports the following functionality:

1) management function for call related issues, e.g. enable or disable call-related event notifications.

2) call control, e.g. route, disconnect.

7.2 Data Session Control

The Data Session Control SCF supports stage 1 requirements related to PS call control.

The Data Session Control network service capability feature supports the following functionality:

1) management functions for data session related issues, e.g. enable or disable data session-related event notifications

2) session control, e.g. route, disconnect.

7.3 Mobility

The Mobility SCF addresses stage 1 requirements for user location, and user status based on network-related information.

The Mobility SCF provides terminal location information, and general terminal status monitoring. The following information is reported when requested provided that the network is able to support the corresponding capability:

- user whom the report concerns;
- VLR number;
- Cell Global Identification or Location Area Identification;
- location number (network specific, refer to ITU-T Q.763);
- geographical location (e.g. in terms of universal latitude and longitude co-ordinates);
- accuracy (value depending on local regulatory requirements and level of support in serving/home networks; note that the accuracy of the serving network might differ from that in the home environment);
- age of location information (last known date/time made available in GMT);
- status of the user's terminal.

An application uses this SCF to perform the following:

- user location requests;
- requests for starting (or stopping) the generation by the network of periodic user location reports;
- requests for starting (or stopping) the generation by the network of user location reports based on location changes;
- report of location information;

- notification of location update.

The application can also for each user start/stop receipt of notifications and modify the required accuracy by selecting another option from the network provided options.

7.4 Terminal Capabilities

It shall be possible for an application to request Terminal Capabilities as defined by MExE (MExE User Profile) [3]. The terminal capabilities are provided by a MExE compliant terminal to the MExE Service Environment either on request or by the terminal itself.

Terminal Capabilities are available only after a capability negotiation has previously taken place between the user´s MExE terminal and the MExE Service environment as specified in [3].

NOTE: For Release 4 only WAP MExE devices can supply terminal capabilities.

7.5 User Interaction

User Interaction SCFs support stage 1 requirements for information transfer.

There are two user interaction SCFs:

- Generic User Interaction: used by applications to interact with end users;
- Call User Interaction: used by applications to interact with end users participating to a call.

7.6 User Profile Management

User Profile information may be distributed between the Home Environment and the Home Environment Value-Added Services Providers. The HE-VASP may manage information specific to the services supported by their OSA applications. For this, they may use models and mechanisms, which are out of the scope of OSA release 4.

Home Environment User Profile information consists of various user interface and service related information. Of particular interest in the context of release 4 is the following information:

- list of services to which the end-user is subscribed;
- service status (active/inactive);
- privacy status with regards to network service capabilities (e.g. user location, user interaction);
- terminal capabilities.

Home Environment user profile information may be stored centrally, or the information may be distributed over relevant physical entities.

Terminal capabilities may be accessed by OSA applications through the network Terminal Capabilities SCF.

7.7 Charging

The Charging SCF addresses stage 1 requirements for charging related to service usage (and not call/session control).

This SCF permits an application to access subscriber accounts maintained by the network and charge subscribers for service usage.

Provided, that these functions are supported by the underlying network an application providing a service to the subscriber can use the Charging SCF to:

- Check, if – for the service to be provided by the application – the charge is covered by the subscribers account or credit limit.
- Reserve – for the service to be provided by the application – a charge in the subscribers account, that can be deduced from the account after service delivery.
- Deduct an amount from the subscriber's account.
- Release a reservation acquired earlier.
- Add non-monetary units to a subscriber's account.
- Deduct non-monetary units from a subscriber's account.

Reverse a completed charge transaction, e.g. after repudiation.

7.8 Account Management

The Account Management SCF addresses stage 1 requirements related to the features to monitor subscriber's account:

- retrieval of transaction history for a certain subscriber's account;
- query of the balance of the account of one or several subscriber's;
- request of notifications on certain criteria for one or several subscribers.

8 OSA Internal API

The OSA internal API between framework and service capability servers (SCSs) supports registering of network service capability features (SCFs), permits the framework to retrieve a network SCF manager interface when an application is granted access to a network SCF, and enables integrity management by means of load management, heartbeat management and fault management.

8.1 OSA Access and Discovery

To support registration, the OSA Access and Discovery interfaces shall be supported at the OSA internal API.

8.2 Registration of network service capability features at the framework

The Framework needs to know the Service Capability Features provided by the SCSs, in order to make them available to applications. For this purpose network service capability features have to be registered with the Framework, and they need to be registered in such a way that applications can discover them.

NOTE: Framework and Service Capability Servers are located within the same trusted domain. Therefore no authentication mechanisms are required between them.

8.2.1 Service Registration

The Service Registration interface provides the methods used for the registration of network SCFs at the framework.

8.2.2 Service Factory

The Service Factory interface allows the framework to get access to a manager interface of a network SCF. It is used, in order to return an SCF manager interface reference to the application. Each SCF has a manager interface that is the initial point of contact for the network SCF.

8.3 Integrity Management

Integrity Management interfaces allow the framework to perform load management, heartbeat management and fault management.

8.3.1 Load Management

Load management enables the framework to manage the load allowing it to be distributed across multiple SCSs by means of load balancing.

8.3.2 Heartbeat Management

Heartbeat management allows the initialisation of a heartbeat supervision of a network SCF.

8.3.3 Fault Management

Fault management allows to inform the framework of events which affect the integrity of the framework and network SCFs, and to request information about the integrity of the system.

Annex A (informative): Change History

Change history					
TSG SA #	**Version**	**CR**	**Tdoc SA**	**New Version**	**Subject/Comment**
SA_09	3.1.0	012	SP-000452	3.2.0	CR on Parlay-OSA alignment: basic service interface
SA_09	3.1.0	013	SP-000452	3.2.0	CR on Parlay-OSA alignment: initial contact interfaces
SA_09	3.1.0	014	SP-000452	3.2.0	CR on Parlay-OSA alignment : access SCF
SA_09	3.1.0	015	SP-000452	3.2.0	CR on Parlay-OSA alignment: load manager SCF
SA_09	3.1.0	016	SP-000452	3.2.0	CR on Parlay-OSA alignment: fault manager SCF
SA_09	3.1.0	017	SP-000452	3.2.0	CR on Parlay-OSA alignment: service factory SCF
SA_09	3.1.0	018	SP-000452	3.2.0	CR on Parlay-OSA alignment: authentication interface
SA_09	3.2.0	011r1	SP-000452	4.0.0	Change of TS 23.127 title for version 4.0.0 and up
SA_11	4.0.0	020	SP-010118	4.1.0	Alignment of 23.127 due to transfer of stage 3 content to 29.198
SA_11	4.0.0	021	SP-010118	4.1.0	Alignment due to additional Release 4 capabilities.
SA_12	4.1.0	026	SP-010332	4.2.0	CR on addition of transport examples in addition to CORBA
SA_15	4.2.0	029	SP-020133	4.3.0	OSA Mobility SCF
SA_15	4.2.0	030	SP-020133	4.3.0	OSA Charging and Account Management SCFs
SA_15	4.2.0	031r1	SP-020133	4.3.0	OSA Internal API, Integrity Management

History

Document history		
V4.1.0	April 2001	Publication
V4.2.0	June 2001	Publication
V4.3.0	March 2002	Publication