Technical Specification Group Services and System Aspects ***TSGS#12(01)0332***

Meeting #12, Stockholm, Sweden, 18-21 June 2001

Source: **TSG SA WG2**
Title: **CRs on 23.127**
Agenda Item: **7.2.3**

The following Change Requests (CRs) have been approved by TSG SA WG2 and are requested to be approved by TSG SA plenary #12.
Note: the source of all these CRs is now S2, even if the name of the originating company(ies) is still reflected on the cover page of all the attached CRs.

CR#	***re***	***Rel***	***title***	***cat***	***in***	***out***	***S2#***	***WI***
025		R99	Adding transport examples in addition to CORBA	F	4.1.0	4.2.0	S2-011480	OSA
026		R4	Adding transport examples in addition to CORBA	F	4.1.0	4.2.0	S2-011481	OSA1

Keywords

VHE, OSA

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Copyright Notification

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 and Open Service Architecture.

Virtual Home Environment (VHE) is defined as a concept for personal service environment (PSE) portability across network boundaries and between terminals. The concept of the 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 1999, e.g. CAMEL, MExE and SAT are considered the mechanisms supporting the VHE concept.

The Open Service Architecture (OSA) defines an architecture that enables operator and third party applications to make use of network functionality through an open standardised API (the OSA API). OSA provides the glue between applications and service capabilities provided by the network. In this way applications become independent from the underlying network technology. The applications constitute the top level of the Open Service Architecture (OSA). This level is connected to the Service Capability Servers (SCSs) via the OSA API. The SCSs map the OSA API onto the underlying telecom specific protocols (e.g. MAP, CAP etc.) and are therefore hiding the network complexity from the applications.

Applications can be network/server centric applications or terminal centric applications. Terminal centric applications reside in the Mobile Station (MS). Examples are MExE and SAT applications. Network/server centric applications are outside the core network and make use of service capability features offered through the OSA API. (Note that applications may belong to the network operator domain although running outside the core network. Outside the core network means that the applications are executed in Application Servers that are physically separated from the core network entities).

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.

2.1 Normative references

[1] GSM 01.04 (ETR 350): "Digital cellular telecommunication system (Phase 2+); Abbreviations and acronyms".

[2] GSM 02.57: "Digital cellular telecommunication system (Phase 2+); Mobile Station Application Execution Environment (MExE); Service description".

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

[4] 3G TS 22.078: "Customised Applications for Mobile network Enhanced Logic (CAMEL) (Phase3); Service description - Stage 1".

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

[6] GSM 11.14: "Digital cellular telecommunication system (Phase 2+); Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment; (SIM - ME) interface".

[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); Virtual Home Environment".

[10] 3GPP TR 22.905: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Vocabulary for 3GPP Specifications".

[11] IETF PPP Authentication Protocols - Challenge Handshake Authentication Protocol [RFC 1994, August1996].

[12] World Wide Web Consortium Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation (http://www.w3.org).

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

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

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

2.2 Informative references

[1] 3GPP TR 22.970: "Universal Mobile Telecommunications System (UMTS); Virtual Home Environment".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions 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. The Personal Service Environment is defined in terms of one or more User Profiles.

Service Capabilities: see [9].

Service Capability Feature: see [9].

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].

Further UMTS related definitions are given in 3G TS 22.101 and 3G TR 22.905.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations 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 Station (Application) Execution Environment
MS	Mobile Station
MSC	Mobile Switching Centre
OSA	Open Service Architecture
PLMN	Public Land Mobile Network
PSE	Personal Service Environment
SAT	SIM Application Tool-Kit
SCF	Service Capability Feature
SCS	Service Capability Server
SIM	Subscriber Identity Module USIM User Service Identity Module
SOAP	Simple Object Access Protocol
VASP	Value Added Service Provider
VHE	Virtual Home Environment
WGW	WAP Gateway
WPP	WAP Push Proxy

Further GSM related abbreviations are given in GSM 01.04. Further UMTS related abbreviations are given in 3G TR 22.905.

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.

The user profiles consist of two kinds of information:

- interface related information (User Interface Profile); and

- service related information (User Services profile).

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

5 Open Service Architecture

In order to implement not known end user services/applications today, a highly flexible Open Service Architecture (OSA) is required. The Open Service Architecture (OSA) is the architecture enabling applications to make use of network capabilities. The applications will access the network through the OSA API that is specified in this Technical Specification.

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 aim of OSA is to provide an extendible and scalable architecture that allows for inclusion of new service capability features and SCSs in future releases of UMTS with a minimum impact on the applications using the OSA API.

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 server 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, ~~like~~ 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 (~~e.g.~~for example CORBA/IIOP, SOAP/XML, other XML based protocols etc.) and lower layer protocols (~~e.g.~~for example TCP, IP, etc.) should provide security mechanisms to encrypt data (~~e.g.~~for example TLS, IP sec, etc.).

6.1.1 Initial Contact

The application gains a reference to the Initial Contact SCF 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 the initiateAuthentication 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 and network service capability features. This is done by invoking the requestAccess method, by which the application requests a certain type of access service capability feature. The OSA Access service capability feature is defined in subclause 6.1.3.

The Initial Contact framework SCF is defined by a unique interface, consisting of the following methods.

Method	**`initiateAuthentication()`** The application uses this method to initiate the authentication process.
Direction	Application to Framework
Parameters	This identifies the application domain to the framework, and provides a reference to the domain's authentication interface. appDomain The authInterface parameter is a reference to call the authentication interface of the client application. The type of this interface is defined by the authType parameter. If the interface reference is not of the correct type, the framework returns an error code (`P_INVALID_INTERFACE_TYPE`). authType This identifies the type of authentication mechanism requested by theapplication. It provides operators and HE-VASPss with the opportunity to use an alternative to the OSA Authentication interface, ~~e.g.~~for example CORBA/IIOP Security, SOAP/XML security etc.
Returns	fwDomain This provides the application domain with a framework identifier, and a reference to call the authentication interface of the framework.
Errors	

Method	**`requestAccess ()`** Once application and framework are authenticated, the former invokes the requestAccess method on the Initial Contact SCF. This allows the application to request the type of access it requires. If it requests OSA_ACCESS, then a reference to the OSA Access interface is returned. (Home Environments can define their own access interfaces to satisfy application requirements for different types of access.)
Direction	Application to network
Parameters	accessType This identifies the type of access SCF requested by the application. appAccessInterface This provides the reference for the framework to call the access interface of the application.
Returns	fwAccessInterface This provides the reference for the application to call the access SCF of the framework.
Errors	INVALID_AUTHENTICATION The application is not authenticated.

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.

1) The application calls initiateAuthentication on the Home Environment's framework Initial interface. This allows the application to specify the type of authentication process. This authentication process may be specific to the Home Environment, or the implementation technology used. The initiateAuthentication method can be used to specify the specific process, (~~e.g.~~for example CORBA/IIOP security, SOAP/XML security, etc.). OSA defines a generic authentication service capability feature (Authentication), which can be used to perform the authentication process. The initiateAuthentication method allows the application to pass a reference to its own authentication interface to the Framework, and receive a reference to the Authentication interface supported by the framework, in return.

2) The application invokes the selectAuthMethod on the framework's Authentication SCF. This includes the authentication capabilities of the application. The framework then chooses an authentication method based on the authentication capabilities of the application and the framework. If the application is capable of handling more than one authentication method, then the framework chooses one option, defined in the prescribedMethod parameter. In some instances, the authentication capability of the application may not fulfil the demands of the framework, in which case, the authentication will fail.

3) The application and framework interact to authenticate each other. Depending on the method prescribed, this procedure may consist of a number of messages e.g. a challenge/ response protocol. This authentication protocol is performed using the authenticate method on the Authentication interface. Depending on the authentication method selected, the protocol may require invocations on the Authentication SCF supported by the framework; or on the application counterpart; or on both.

The Authentication framework SCF is defined by a single interface, consisting of the following methods.

Method	**selectAuthMethod ()** The application uses this method to initiate the authentication process. The mechanism returned by the framework is the mechanism it prefers. This should be within capability of the application. If a mechanism that is acceptable to the framework within the capability of the application cannot be found, the framework returns an error code (INVALID_AUTH_CAPABILITY).
Direction	Application to network
Parameters	authCaps This is the means by which the authentication mechanisms supported by the application are conveyed to the framework.
Returns	prescribedMethod This is returned by the framework to indicate the mechanism it prefers for the authentication process. If the value of the prescribedMethod returned by the framework is not understood by the application, it is considered a fatal error and the application must abort.
Errors	INVALID_AUTH_CAPABILITY No acceptable authentication mechanism could be found by the framework.

Method	**`authenticate ()`** *`(application to network)`* This method is used by the application to authenticate the framework using the mechanism indicated in prescribed Method. The framework must respond with the correct responses to the challenges presented by the application. The clientAppID received in the `initiateAuthentication()` can be used by the framework to reference the correct public key for the application (the key management system is currently outside of the scope of the OSA specification). The number of interactions and the order of the interactions is dependent on the prescribedMethod.
Direction	Application to network
Parameters	prescribedMethod This parameter contains the method that the framework has specified as acceptable for authentication (see selectAuthMethod). challenge The challenge presented by the application to be responded to by the framework. The challenge mechanism used will be in accordance with the IETF *PPP Authentication Protocols - Challenge Handshake Authentication Protocol* [RFC 1994, August1996]. The challenge will be encrypted with the mechanism prescribed by `selectAuthMethod()`.
Returns	response This is the response of the framework to the challenge of the application in the current sequence. The response will be based on the challenge data, decrypted with the mechanism prescribed by `selectAuthMethod()`.
Errors	

Method	**`authenticate()`** *`(network to application)`* This method is used by the framework to authenticate the application using the mechanism indicated in prescibedMechanism. The application must respond with the correct responses to the challenges presented by the framework. The number of interactions and the order of the interactions is dependant on the prescribedMethod. (These may be interleaved with authenticate() calls by the application on the Authentication interface. This is defined by the prescribedMethod.)
Direction	Network to application
Parameters	prescribedMethod This parameter contains the agreed method for authentication (see selectAuthMethod on the Authentication interface.) challenge The challenge presented by the framework to be responded to by the application. The challenge mechanism used will be in accordance with the IETF *PPP Authentication Protocols - Challenge Handshake Authentication Protocol* [RFC 1994, August1996]. The challenge will be encrypted with the mechanism prescribed by `selectAuthMethod()`.
Returns	response This is the response of the application to the challenge of the framework in the current sequence. The response will be based on the challenge data, decrypted with the mechanism prescribed by `selectAuthMethod()`.
Errors	`INVALID_AUTHENTICATION` The application could not be authenticated.

Method	**`abortAuthentication()`***`(application to network)`* The application uses this method to abort the authentication process. This method is invoked if the application no longer wishes to continue the authentication process, (e.g. if the framework responds incorrectly to a challenge.) If this method has been invoked, calls to the requestAccess method on Initial Contact will return an error code (INVALID_AUTHENTICATION) until the application has been properly authenticated.
Direction	Application to network
Parameters	
Returns	
Errors	

Method	**`abortAuthentication()`***`(network to application)`* The framework uses this method to abort the authentication process. This method is invoked if the framework wishes to abort the authentication process, (e.g. if the application responds incorrectly to a challenge.) If this method has been invoked, calls to the requestAccess method on Initial will return an error code (INVALID_AUTHENTICATION), until the application has been properly authenticated.
Direction	Network to application
Parameters	
Returns	
Errors	

Annex A (informative): Change History

Change history					
TSG SA #	**Version**	**CR**	**Tdoc SA**	**New Version**	**Subject/Comment**
SA_07	2.0.0	-	-	3.0.0	Approved at SA#07 and placed under TSG SA Change Control
SA_08	3.0.0	001R1	SP-000286	3.1.0	OSA Internal API
SA_08	3.0.0	002R1	SP-000286	3.1.0	Editorial changes and improvements
SA_08	3.0.0	003R1	SP-000286	3.1.0	Alignment with stage 3 (TS 29.198)
SA_08	3.0.0	004	SP-000286	3.1.0	Removal of data-related parameters in call control SCF
SA_08	3.0.0	005	SP-000286	3.1.0	Replacement of "Camel" by "Network" in Network User
SA_08	3.0.0	006R1	SP-000286	3.1.0	Introduction of improved notification mechanism
SA_08	3.0.0	008R1	SP-000286	3.1.0	Modification of call control
SA_08	3.0.0	009	SP-000286	3.1.0	Data Session Control
SA_08	3.0.0	010	SP-000286	3.1.0	Modification of call control SCF
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_ 10	3.2.0	019	SP-000590	3.3.0	CR on Alignement with 29.198 in getTerminalCapabilities()

Rapporteur: Christophe Gourraud, Ericsson
Email:christophe.gourraud@lmc.ericsson.se Telephone: +1 514 345 7900 (#5795)

Keywords

VHE, OSA, MexE, CAMEL, USAT

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

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 specification 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.

2.1 Normative references

[1] 3G TR 21.004: “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) (Phase3); Service description - Stage 1".

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

[6] 3G TS 21.111: “USIM Application Toolkit (USAT)”

[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); Virtual Home Environment".

[10] 3G TR 21.905: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Vocabulary for 3GPP Specifications".

[11] IETF PPP Authentication Protocols - Challenge Handshake Authentication Protocol [RFC 1994, August1996].

[12] World Wide Web Consortium Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation (http://www.w3.org).

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

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

[15] 3G TS 22.127: “Stage 1 Service Requirement for the Open Service Access (OSA)”

[16] The 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 following terms and definitions 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. 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].

Further definitions are given in 3G TS 22.101 and 3G TR 22.905.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations 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

Further GSM related abbreviations are given in GSM 01.04. Further 3GPP related abbreviations are given in 3G TR 21.905.

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, ~~like~~ 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 (~~e.g.~~for example CORBA/IIOP, SOAP/XML, other XML based protocols etc.) and lower layer protocols (~~e.g.~~for example TCP, IP, etc.) should provide security mechanisms to encrypt data (~~e.g.~~for example TLS, IP sec, etc.).