

## Introduction

During the last VHE ad-hoc meeting a discussion about terminology with regards to VHE and OSA was held. The conclusions of this discussion was open. This contribution provides an attempt to clarify the situation and proposes changes to the 22.21 document.

## Background

### *Concepts/Definitions*

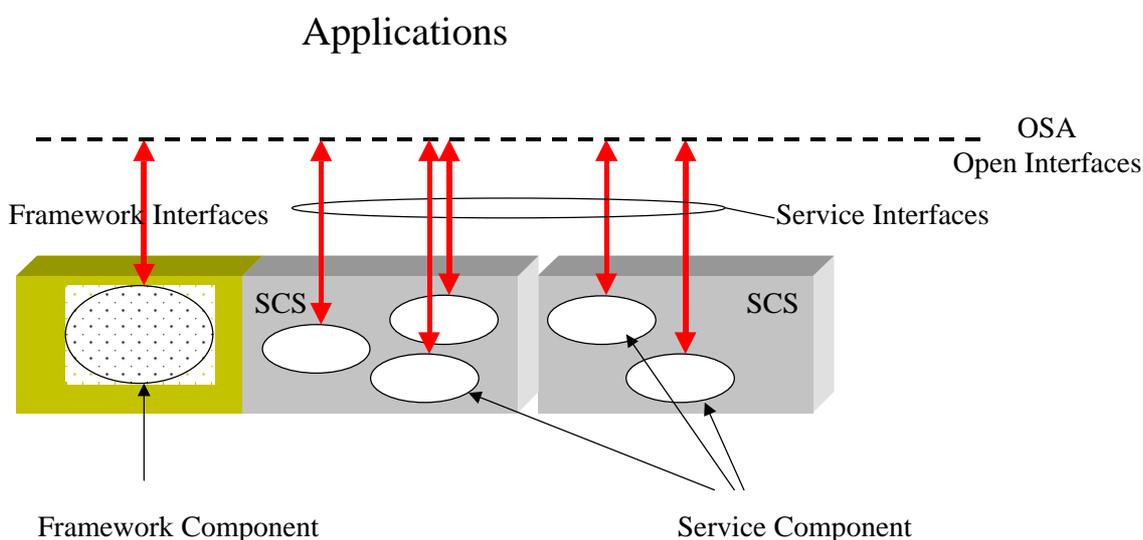


Fig 1 Service Capability overview

**Service Capabilities :** Bearers defined by QoS parameters and/or mechanisms needed to realise services. In UMTS phase 1 release-99, the Service Capabilities are access to bearers, CAMEL, MExE and SAT.

**Service Capability Feature:** Functionality offered by a service capability that is accessible via open standardised interfaces.

#### **Service Capability Service Component (Service Component).**

Service components are logical groupings (functionality sets) of Service Capability Features. Examples of service components are:  
Call Control, Conversational Multimedia Call Control, Location/Positioning, Charging, User Profile Manipulation, Message Transfer (WAP), SIM-Card Download, QoS Bearer Establishment, etc.  
Service Components publish at least one interface.

**Service Capability Server (SCS) :** Enables Applications to access and use functionality provided by Service Components over standardized interfaces (APIs). An SCS hosts one or several service components. Service Capability Servers connect to network functionality by using standardised UMTS and/or GSM protocols.

As defined above, the total functionality offered by a Service Component consists of Service Capability Features. Examples of service capability features of the Call Control service component are: Create call, Route Call, Release Leg, etc.

Clients (applications or other service components) access these service capability features via the open interfaces (APIs) that a service component publishes.

That is, a single service capability feature is accessible and visible to applications via the methods/operations in the interface<sup>1</sup>.

**Framework- and Service interfaces.**

Service interfaces:

- provided by the service components hosted by an SCS.

Framework interfaces:

- provided by a framework component and used to authenticate/authorize Applications, to broker applications to service components and SCSs, let applications discover new service components, etc.

An application should use **one** framework component even if several SCSs are hosting the service components.

**Generic Interfaces towards Service Components**

It should be possible to “re-use service components” and their interfaces (methods defined to access service capability features) when more than one Service Capability Server offer the same functionality.

By re-use is here meant that the service components appear to be identical from an application’s point of view, but implementations of the service components may differ.

This can be achieved by defining a generic interface class that is common over service capabilities and provided by service components<sup>2</sup> on different SCSs.

One example could be a Call Control service component provided either by a CAMEL SCS and/or a WTA SCS (see figure 2). Others could be SCSs for H.323 VoIP and ISDN/PSTN service control (not shown in figure 2).

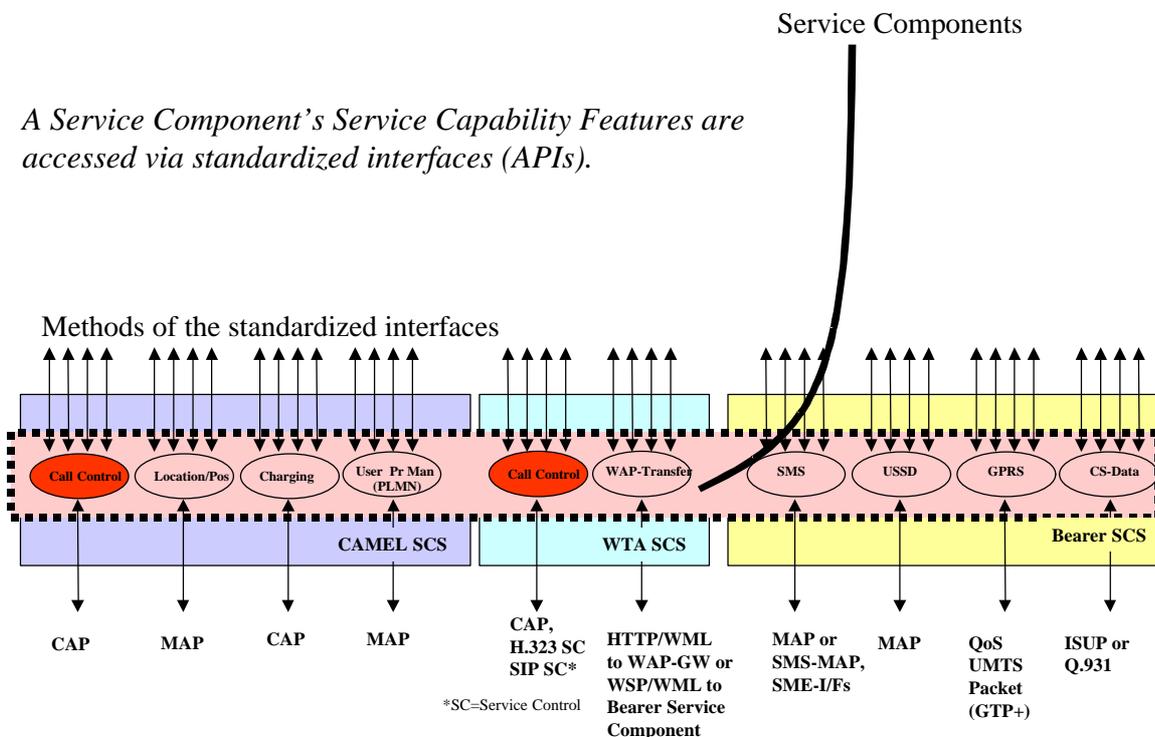


Fig 2 Re-use of generic service components

<sup>1</sup> Stage 2 and stage 3 descriptions will determine whether there shall be a direct correspondence between a Service Capability Feature and a method/operation of the interfaces.

<sup>2</sup> Implementation of the interfaces and service capability features.

Note : The framework component is not shown in Fig 2.

The call control service components of the CAMEL SCS and the WTA SCS in figure 2, appear to applications as “identical”. That is, the interface to them is the same, only the **address** to the call control service component differs. If an ORB (Object Request Broker, e.g. used in CORBA) is used, the ORB will “route” the method/operation to a call control service component based on the address given by the application (the address can e.g. be an object reference or a URL).

If one service capability offers more functionality, this generic interface class may be inherited and extended.

When applications use the generic interface class, they become independent of (portable over) underlying service capabilities. It should however be possible to use the extended interface for more functionality specific to a service capability, but with increased dependency of the used service capability.

In this case, the service components appears to applications as “almost identical”, depending on how much the generic interface class has been extended.

It is important that the generic interface part becomes as large as possible.

### Implementation

It should be possible to implement the OSA framework in a distributed (fig 3) or centralised fashion (fig 4). Distributed object-oriented techniques (such as CORBA, etc.) can be used to locate and access service components and their service capability features.

#### Distributed Implementation

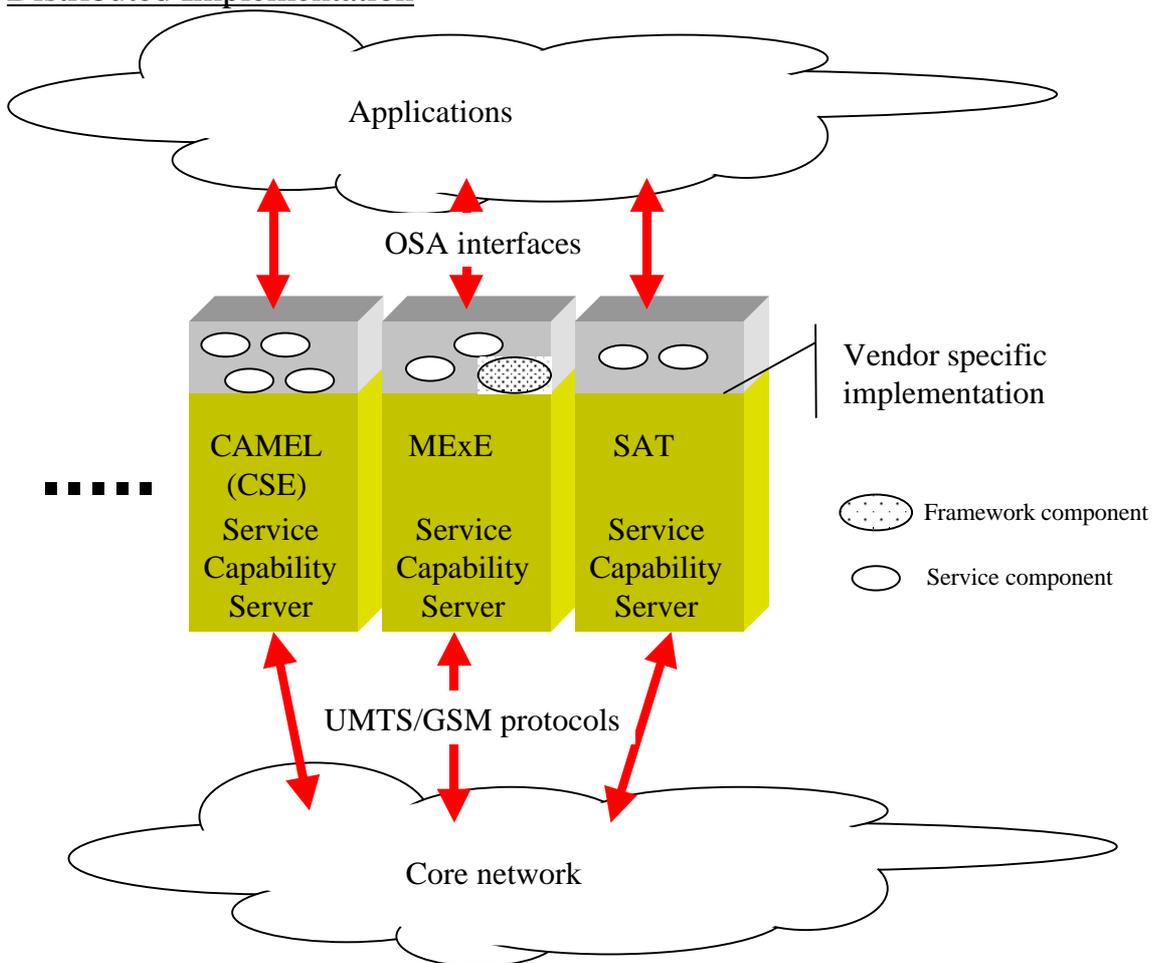


Fig 3. Distributed OSA implementation

The distributed implementation allows for a highly scalable, flexible solution. Applications can use the framework component to discover what service components that are provided.

Applications may also query the framework component which service capability servers that provide a particular service component e.g. call control.

Based on the result, the applications choose one of the service components and address this service component in the method/operation invocation. Applications may also be preconfigured to use a certain service components.

If an ORB is used, the ORB brokers the method/invoke to the requested service component based on the address (e.g. an object reference or a URL).

## Centralised Implementation

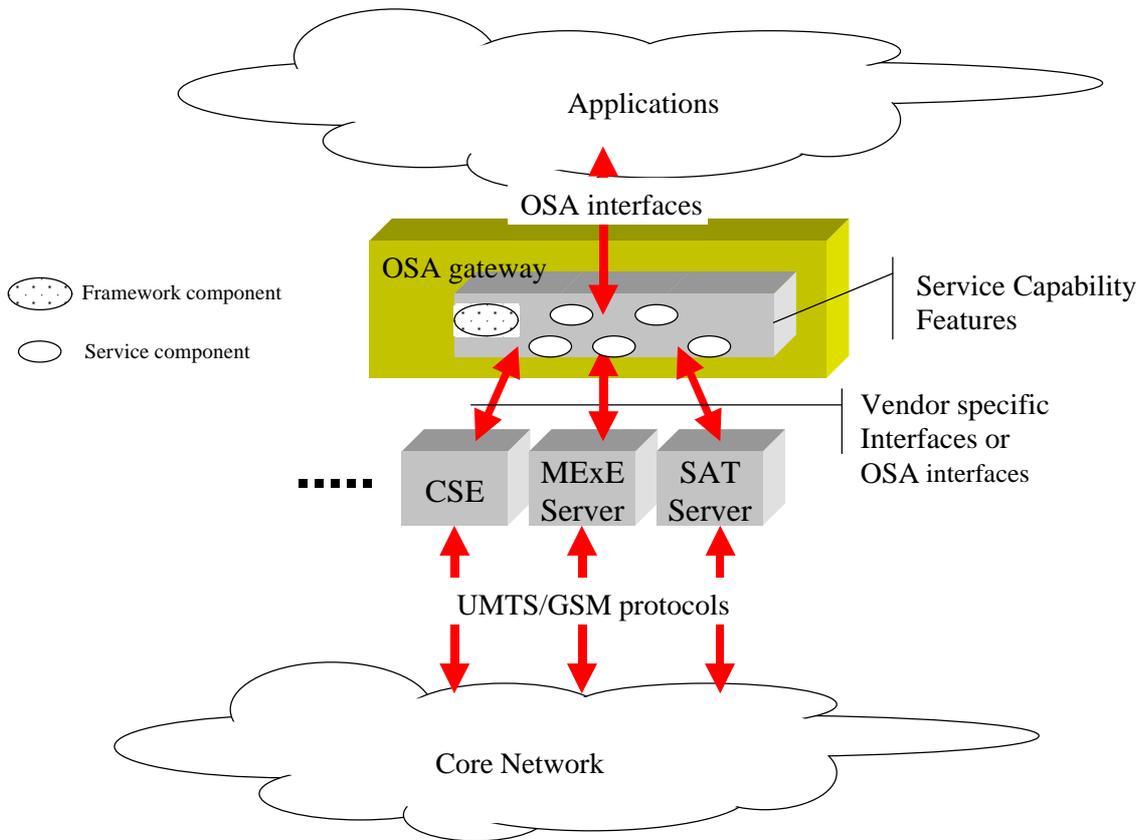


Fig 4. Centralised OSA implementation

In the centralised implementation shown in figure 4, the OSA gateway contains the service components visible to the applications (client). An ORB could still be used for communication between applications and the service components. The difference here is that the OSA gateway acts as a “service broker” towards the applications, i.e. chooses which service capability server that should be used (which SCS hosting a particular service component that should be used). In a multi-tiered Client/Server architecture, the OSA gateway would form a middle-tier between the applications and the Service Capability Servers.

The interfaces between the OSA Gateway and the CSE, MExE Server and SAT Server in figure 4 can in this case be either :

- Vendor specific (e.g. legacy equipment with other interfaces than the OSA interfaces). Here, the SCS functionality has been divided upon the OSA gateway and the CSE, MExE, etc servers.
- OSA interfaces if the CSE, MExE etc servers support OSA (i.e. become SCSs hosting service components). An ORB could be used also here to route the OSA gateway invoked methods/operations to the correct service component (on an SCS).
- A combination of the two.

This approach introduces an extra “node, the OSA gateway and could possibly imply that the OSA gateway becomes a critical resource e.g. from a capacity point of view.

## Conclusion

The VHE/OSA specifications should allow for both implementation alternatives, the distributed and centralised solutions.

## Changes to 22.21

\*\*\* changed section \*\*\*

# 5. Framework for Services

This framework for services will provide the scope for the users to personalise to some degree the way in which services operate.

VHE in release 99 shall support both GSM phase 2+ release 99 teleservices, bearer services and supplementary services as applied in TS 22.00 and new services built by service capabilities.

The goal of standardisation in UMTS with respect to services is to provide a framework within which services can be created based on standardised service capabilities. UMTS services will generally not rely on the traditional detailed service engineering (evident for supplementary services in second-generation systems), but instead provides services using generic toolkits.

Services are realised based on a number of service features or service capabilities features  $\{[2],[3],[4],[9], [10]\}$ . Service features in turn are realised based on service capabilities features, with standardised interfaces between them (see figure 2). Figure 2 does not impose specific implementation techniques of the interfaces shown.

Service Features are not required in release 99.

VHE enables the creation of services by providing access to service features and service capabilities features by means of standardised interfaces. Personalisation of services and user interface will be supported across network and terminal boundaries by providing the services to users with the same “look and feel” irrespective of the network type and within the limitations of the network and terminal.

In addition to services implemented on top of service capabilities (OSS), a generic interface towards the service capabilities shall be provided.

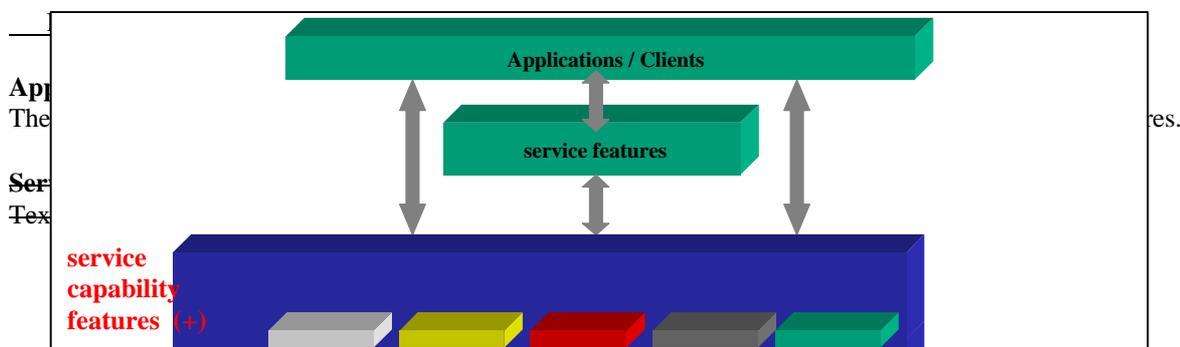
The generic interface shall be:

- Independent of vendor specific solutions,
- Independent of the location where service capabilities are implemented,
- independent of supported server capabilities in the network and,
- independent of programming languages of service capabilities.

In addition to that, ~~the same~~ the same kind of service capability features shall be made visible by a single, generic interface. The generic interfaces in the VHE concept is encapsulated and made visible through the Service Capability Features. The interface between Service Capability Features and the Service Capabilities could be implemented using a middleware layer. Further studies are requested in this area.

Access to these Service Features and Service Capability Features shall be realised using distributed object oriented access technologies.

The access to Service Features and Service Capability Features shall be independent of vendors technology used. It shall be secure, scaleable and extensible.



Bearer defined by QoS parameters and/or mechanisms needed to realise services.

In UMTS phase 1 release-99, the Service Capabilities are access to bearers, CAMEL, MExE and SAT.

**Service Capability Feature:**

Functionality offered by a service capability that is accessible via open standardised interfaces.

**Service Capability Service Component (Service Component).**

Service components are logical groupings (functionality sets) of Service Capability Features.

Examples of service components are:

Call Control, Conversational Multimedia Call Control, Location/Positioning, Charging, User Profile Manipulation, Message Transfer (WAP), SIM-Card Download, QoS Bearer Establishment, etc.

Service Components publish at least one interface.

Examples of service capability features of the Call Control service component are: Create call, Route Call, Release Leg, etc.

Clients (applications or other service components) access these service capability features via the open interfaces (APIs) that a service component publishes.

That is, a single service capability feature is accessible and visible to applications via the methods/operations in the interface .

**Service Capability Server (SCS) :**

Enables Applications to access and use functionality provided by Service Components over standardized interfaces (APIs). An SCS hosts one or several service components.

Service Capability Servers connect to network functionality by using standardised UMTS and/or GSM protocols.

Functionality provided by a Service Capability over standardised interfaces.

**Service Capability Server:**

Service Capability Servers (SCS) enable applications to access and use functionality provided by a Service Capability over standardised interfaces. An SCS hosts one or several service components. Service components are logical groupings (function sets) of Service Capability Features provided by a Service Capability Server. Examples of service components are:

Call Control, Conversational Multimedia Call Control, Location/Positioning, Charging, User Profile Manipulation, Message Transfer (WAP), SIM Card Download, QoS Bearer Establishment.

Service Capability Servers connect to network functionality by using standardised UMTS and/or GSM protocols.

**Service Features:**

Service Features provides sets of service capability features. Such sets—especially in the area of call handling—may serve as building blocks for more complex applications provided by third parties.

The above idea should be used to improve Service Features descriptions.

It is assumed to refer to service capabilities as CSE, HLR, MEXE, SAT server and network entities. The latter in the list needs more considerations for the forthcoming meetings.

The figure 2.0 above shows the different possibilities to implement services as existed in a GSM network and proposed for a UMTS network.

**Guidance on the implementation of services:**

Within UMTS there will be a number of ways to implement and offer applications:

**STANDARDISED SERVICES** (Supplementary Services, Tele-Services, etc.) are implemented on existing GSM/UMTS entities (e.g. HLR , MSC/VLR and terminal) on a vendor specific basis, using standardised interfaces (MAP, etc.) for service communication (e.g. downloading of service data). Availability and maintenance of these Services is also vendor dependent.

**OPERATOR SPECIFIC SERVICES (OSS)** are not standardised and could be implemented at the GSM/UMTS entities (e.g. HLR) on a vendor specific basis or using GSM ph 2+ mechanisms (CAMEL, SAT, MExE). These tool-kits use standardised interfaces to the underlying network (CAP, MAP) or use GSM Bearers to transport applications and data from the MExE/SAT server to the MS/SIM. The implementation of these operator specific services on the different platforms (CSE, MExE/SAT Server, MSs) is done in a completely vendor specific way and uses only proprietary interfaces.

Other **APPLICATIONS** are like OSS not standardised. These applications will be implemented using standardised interfaces to the Service Capabilities (Bearers, Mechanisms). The functionality offered by the different Service Capabilities will be defined by a set of so-called Service Capability Features. ~~This set~~ These Service Capability Features will be standardised and can be used by the application designers to build their applications.

Within the terminals Service Capabilities are accessible via the existing MExE and SAT APIs, i.e. there will be no service capability features and thus no service features within the terminal. The terminal can communicate, using GSM/UMTS bearers, with applications in the network via the service capability features defined for MExE- and SAT-servers.

The implementation of the Service Capability Features on the different Service Capabilities is still manufacturer specific, i.e. each manufacturer has to implement the functionalities of these standardised interfaces (Service Capability Features) on his platform.

Service Features offer high level functionality via standardised interfaces by combining individual Service Capability Features. Service Features are fully based upon Service Capability Features themselves.

This would leave it to the application designers to use either the Service Features to build their applications or base them directly on the Service Capability Features.

~~NOTE: Within the Open Service Architecture Work Item it is assumed to apply open standardised interfaces to GSM/Bearer, MExE and SAT Servers although these servers are not yet defined. These parts of the above figure are indicated as grey squares and arrows (Further clarifications requested)~~

---

## 6 — Open Service Architecture (OSA)

~~The objective to invent open service architecture is to allow secure access to core and advanced capabilities embedded in the UMTS network.~~

~~From a service designer point of view, the most important goal to be achieved is to provide a generic interface to the network as described in sub clause “chapter 5”~~

~~Access to these Features and Capability Features shall be realised using distributed object oriented access technologies.~~

~~The access to Features and Capability Features shall be independent of vendors technology used. It shall be secure, scaleable and extensible.~~

**Model for Implementation of Open Interfaces (APIs)**

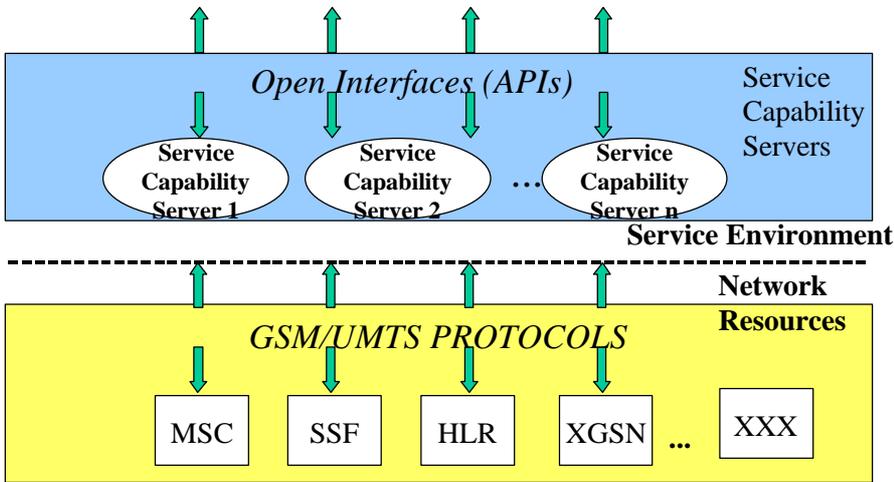


Fig 4.0 — Open Service Architecture principle (to be redrafted)

**6.1 — Service Capability Servers**

The Service Capability Servers reflect the service capabilities in UMTS phase1, i.e. access to bearers, call control, mobility management, tele services or supplementary services. CAMEL-, MExE- and SIM-Toolkit- capabilities may also be accessed. The functionality of these can further be subdivided into server components.

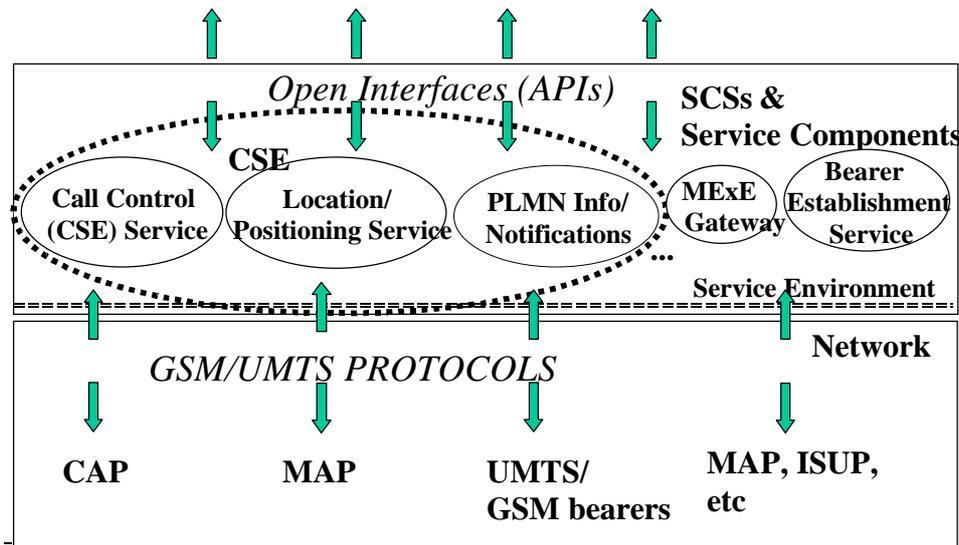


Fig. 5.0 — UMTS service capabilities (to be redrafted)

A service capability server consists of one or several service components. Taking CAMEL Services as an example, the service components could be Call and session Control, Location/Positioning, PLMN Information & Notifications. The same kind of capability of service components implemented in one or several service components is encapsulated and made visible using a service capability feature. Access to service capability features is implemented independent of technologies and defined by standardised interface.

~~Service capability Features will have access to service components over non-standardised, platform dependent procedures.  
The communication between service components and the core network is implemented by using GSM/UMTS protocols.~~

**\*\*\* changed section \*\*\***

## 11 Service Capability Features

Services Capability Features are open, technology- independent building blocks accessible via standardised and extensible interfaces. ~~This~~ These interfaces shall be applicable for a number of different business and applications domains (including beside the telecommunication network operators also service provider, third party service providers acting as HE-VASPs, etc.).

All of these businesses have different requirements, ranging from simple telephony and call routing, virtual private networks, fully interactive multimedia and using MS based applications.

~~This~~ These interfaces shall provide an secure and open access to service capabilities (e.g. CSE, MExE, etc) of the underlying UMTS network.

It is proposed that two categories of access should be defined:

- Service interfaces, which offers the applications the access to a range of network capabilities.
- Framework interface, which shall be commonly used providing "surround" capabilities necessary for the Service interfaces to be open, secure, resilient and manageable.

### 11.1 Framework interface

These Service capability Features offered by this interface will be used e.g. for authentication, registration, notification, etc..

Other commonly used service capability features ~~ies~~ haves to be defined (FFS).

#### 11.1.1 Authentication

This service capability feature shall provide the authentication of an application by the network, i.e. before any application can interact with the network a service agreement will have to be established or an existing agreement will need modification or indeed termination if it is superseded.

Once an application has been authorised to use one, more or all service components and their service capability features, no further authorisation is required as long as the "allowed" service capability features are used.

#### 11.1.2 Registration

This service capability feature deals with the registration of functionality that a service capabilitys provides. After the registration of functionality of a service capability this functionally could be used by authenticated applications.

#### 11.1.3 Notification

This service capability feature provides the functionality to the applications to enable, disable and receive notifications of service application related events that have occurred in the underlying UMTS network, e.g. indication that a new call is set-up or a message is received.

A Notification is defined as an event, which occurs in the UMTS network. It is monitored by the Notification Service Capability Feature and reported to the application / client.

NOTE: It should be further studied if the event notification should belong to the Framework component's Notification Service Capability Feature, or by the Service Components providing the other Service Capability Features related to the Event notification (example incoming call event notification handled by the Call Control Service Component).

#### 11.1.4 Discovery

This service capability feature allows application to identified available service components and service capability features .

#### 11.1.5 FFS

There might be other capability, e.g. for charging and billing of services, for logging or fault management. This is currently for further study.

### 11.2 Service interface

These service capability features provide the application the access to network capabilities. The set of interfaces and service capabilities features will be defined in such a generic way to hide the network specific or Service components specific implementation.

To provide such a generic interface it is necessary to identify the specific functionality offered by ~~the different~~ particular service component and to abstract and generalise functionality which is offered by more than one service component e.g. Call control, messaging services etc.

It is important that the generic interface part becomes as large as possible.

When applications use the generic interface class, they become independent of (portable over) underlying service capabilities. It should however be possible to use the extended interface for more functionality specific to a service capability, but with increased dependency of the used service capability.

In the following sections it is proposed to define such generic service ~~capabilities~~ components, e.g. for call control, messaging services, ~~etc.~~ and to show how this generic functionality could be mapped to the specific server components

#### 11.2.1 Call Control Service Component

This section details the requirements for the service capabilities ~~features for in~~ the Call Control service component, which will be used by the applications. The Call Control service ~~capabilities~~ components shall offer the functionality to establish, maintain, modify and release calls.

To define the necessary service capability ~~features~~ ies it is proposed to use a generic ~~a~~-call model (including ~~also~~ the call leg handling).

The following call control service capabilities were identified:

- Create call
- release call
- route call to destination
- establish new call leg
- release call leg
- ~~attached~~ call leg
- detach call leg
- request call status information
- define call duration
- supervise call

The mapping to service capabilities ~~server components~~ is for further study. (It shall be investigated to which extend the requirements above fit to CAMEL, MEXE and other service capabilities.)

The sections below are deleted due to the description shall follow the above one. It is not intended to delete the content, but to seek for a more generic description model.