Future Open Mobile Services

Do van Thanh
Wireless Communications, Telenor R&D
Snarøyveien 30 NO-1131 Fornebu, Norway; thanh-van.do@telenor.com

Abstract — **The major barriers for the success of mobile data services are the lack of comprehensible mobile service architectures, their confusing business models and the complexity combined with the inconsistency of the technology enablers. This paper attempts to present a more structured and comprehensive analysis of the current mobile service architectures and their technology enablers. The paper starts with a thorough study of the evolution of mobile services and their business models, and a collection of expectations of the different actors, including the end-user. Next, starting from the original mobile services architecture and environment, an attempt to place the different technology enablers in relation to each other and in relation to their position in the mobile system, will be carried out. Each technology enabler together with their contribution in the enhancement of mobile services are then summarised in a complete and comprehensive way. The paper concludes with a recapitulation of the achievement of the state-of-the-art technology enablers and an identification of future improvements.**

Index Terms— **mobile services, mobile applications, mobile technology enablers, mobile service architectures.**

I. INTRODUCTION

With the success of mobile telephony expressed by the increasing number of mobile subscriptions and number of mobile handsets and the advent of the Internet with its plurality of fancy, rich and useful applications, it is natural to expect a blooming of mobile data services. Unfortunately, this has not happened yet. There are, of course, emerging new data services like browsing, presence-based services, location-based service, gaming, data synchronization services, etc. but their popularity and usage are still quite limited. The dominant mobile services remain the communication ones. Voice communication or voice telephony is by far still the most used service. Next comes the Short Message Service (SMS) that allows people to communicate by exchanging short message of 128 characters. SMS is getting very popular, especially among youngsters. One of the explanations could be that these communication services address real needs of the users while the data services are superficial and "nice-to-have" one. More careful studies have revealed causes that are related to the architecture of mobile services, their business models and the complexity and incompleteness of the technology enablers. Indeed, from a closed architecture with rigid standard specifications, mobile service architecture should evolve to allow the development, deployment and operation of mobile services by third parties. New and more flexible business models should be used to allow the entrance of new players.

Many initiatives were started to propose and realise technology enablers for the development and deployment of advanced open mobile services. Unfortunately, these initiatives were quite often uncoordinated and sometimes they are even competing with each other. The consequence is that the technology enablers are often incomplete, inconsistent, overlapping or conflicting each other. The landscape of mobile services is rather complex and chaotic.

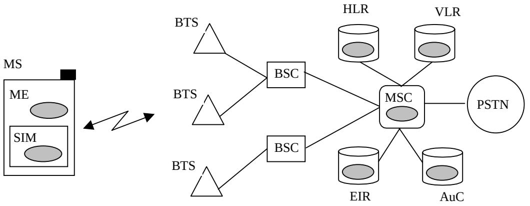
This paper attempts to present a more structured and comprehensive analysis of the current mobile service architectures and their technology enablers. The paper starts with a thorough study of the evolution of mobile services and their business models, and a collection of expectations of the different actors, including the end-user. From the original mobile services architecture and environment, the different technology enablers will be introduced. Their role, their relation with other technology enablers and their position in the mobile system, will be shown. Each technology enabler together with their contribution in the enhancement of mobile services are then summarised in a complete and comprehensive way. A recapitulation of the achievement of the state-of-the-art technology enablers and an identification of future improvements conclude the paper.

II. Overview of the Evolution of Mobile Services

A. *Evolving mobile services*

At the early stage of mobile communication, mobile services are centred on voice communication, i.e. telephony. They are available only in the telecommunication networks. The mobile services are implemented partly as logic installed in the network elements in the mobile telecommunication network and partly as logic in the mobile handset. They are totally managed and controlled by the mobile operator and are quite "closed" in terms of implementation, deployment and operation.

Figure 1 shows that the mobile telephony service is realised by components represented by grey ovals that are distributed both on the mobile phone, also called Mobile Station (MS), and on the mobile network. On the MS, there are components both on the Mobile Equipment (ME) and on the Subscriber Identity Module (SIM).



BTS: Base Station Transceiver – BSC: Base Station Controller – HLR: Home Location Register – MSC: Mobile service Switching Center - VLR: Visitor Location Register - EIR: Equipment Identity Register - AuC: Authentication Center – PSTN: Public Switched Telephony Network

Figure 1 Telephony service components in mobile communications system

However, it does not take long time before a variety of quite different services started to emerge. The typical mobile services are as follows:

- Real time communication (voice, multimedia)
- Browsing (WAP, HTTP, voice-enabled)
- Messaging (SMS, MMS, email, push-to-talk)
- Presence-based services
- Location-based services
- Data synchronization (calendars, contacts, files)
- Device management (remote configuration)
- Data services (file transfer, email download)
- Gaming (download, interactive)
- Streaming media (music, video, events)
- Peer-to-peer communication (local, remote)
- M-commerce (micro-payments, finance)

These mobile services require new and quite different technologies.

B. *Evolving business models*

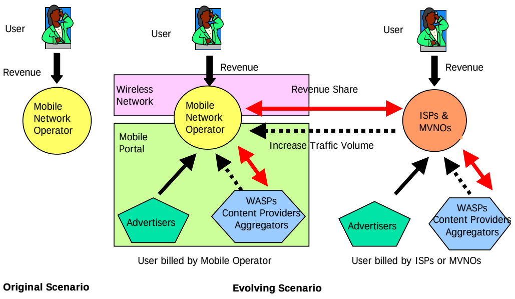


Figure 2 Business models in mobile communication

As shown in Figure 2, from a simple business model in which the user or subscriber pays the *Mobile Network Operator (MNO)* for the service usage, the mobile telecommunication business has evolved to a much more complex business model with higher number of actors involved. First, there was the arrival of *Wireless Application Service Providers (WASP), Content Providers (CP), Content Aggregators (CA)* that deliver their service or content using the MNO's wireless network. The MNO's revenues are increased by the increase in traffic volume and by the revenue sharing with WASPs, CPs and CAs. Another revenue source are from advertisements on the MNO's mobile portal. The business model is getting more complicated with the emergence of *Mobile Virtual Network Operators (MVNO)* or *Internet Service Providers (ISP)* that appear as a mobile network operator to their customers. They have their circle of WASPs, CPs, CAs, and advertisers. They are sharing their revenues with the MNOs and contribute to increase the traffic volume.

C. Evolving requirements and expectations

These different actors have different interests and put different requirements on mobile services and their technology enablers.

The end user's perspective: Mobile services should be highly available at hand and as cheap as possible. Access to services should be unhindered, which means they should be available from multiple devices, network domains (home and visited) using different access technologies. The service spectrum should be rich and varied, thus access to 3rd party services should be ensured. Also, services should be easy to use, have good performance and when relevant, be possible to personalise. Services should also be secure to use and protect the privacy of the users.

The mobile network operator's perspective: Mobile services should be easy to deploy, stable, efficient and require little administration and maintenance. Thus, a rich and varied service spectrum can be provided to the subscribers. Provision of 3rd party services should be possible to increase the service spectrum. It should be possible to employ several different business models for each type of service.

The wireless application service provider's perspective: Service providers should have access to a large population of mobile users. This means access to subscribers of different network operators and that services could be accessible on terminals from different vendors. Also, access to key network functionalities should be allowed via generic APIs. Provisioning of services should be independent of access network technologies. Different business models should be possible to employ.

The content provider's perspective: It should be possible to provide content using a standardised format. The content download procedure should be standardized, reliable and ensure billing opportunities. In addition, it should be possible to protect content using digital rights management (DRM) [1] solutions.

The equipment vendor's perspective: It should be possible to differentiate equipment using various preinstalled services and user opportunities, but the equipment should still be compatible with already existing services and equipment to ensure maximum adoption by users.

III. Analysis of Current Mobile Technology Enablers and Their Contributions

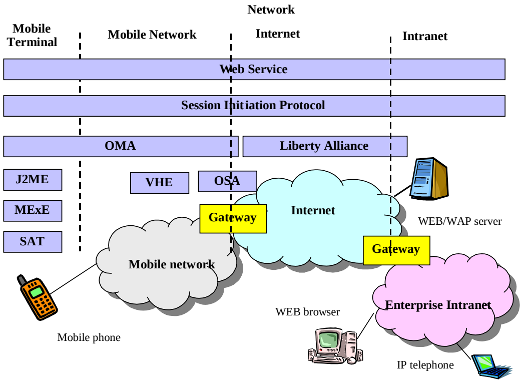


Figure 3 Classification of technology enablers into domains

To fulfil the requirements of the different actors mentioned in the previous section, many technology enablers have been introduced by different initiatives and organisations focusing one requirement subset. The result is a quite complex mobile landscape where the technology enablers are incomplete, inconsistent, competing and even conflicting with each other. As mentioned in earlier sections, mobile services are originally implemented as logic installed in the mobile handset and in the mobile network. To improve mobile services, technology enablers both in the mobile handset and the mobile network are required. In addition, to enable the introduction of fancy services, there should also be technology enablers that bridge the mobile network to the Internet and the Intranet. To make the technology enabler landscape easier to understand, an attempt to place the technology enablers in relation to each other and to the four domains: Mobile Terminal, Mobile Network, Internet and Intranet.

As shown in Figure 3, some enablers such as SAT, MExE, J2ME are confined to one domain, Mobile Terminal while others like Web services and SIP span over all the four domains.

It is worth emphasizing that a technology enabler can be a concept, a framework, a specification, an implementation or even the initiating organisation. For example, the

Mobile Execution Environment (MExE), earlier known as the Mobile Station Application Execution Environment [3], sounds like a runtime environment for executing services in, while it in fact is a framework describing some requirements that should be fulfilled by actual runtime environment implementations.

A. Technology enablers in the mobile terminal

A mobile terminal or more precisely a GSM terminal, actually consists of two devices: the ME (Mobile Equipment), which is the mobile apparatus itself, and the SIM (Subscriber Identity Module), which is a smart card containing the user subscription. A smart card is a tamper-resistant device that has its own processor and memory. The SIM contains the authentication mechanisms necessary to allow the network to authenticate the subscriber. While the ME is owned by the user or subscriber, the SIM is the mobile operator's propriety. The ME is the master that gives commands to its slave, the SIM.

To upgrade the mobile terminal from a telephony device to an advanced device offering multiple services, both the SIM and the ME can be enhanced to host advanced services. For the SIM, there is one technology enabler, SAT (SIM Application Toolkit). For the ME there are two, MExE and J2ME.

1) SIM Application Toolkit (SAT)

The SIM can be used to host advanced services but, unfortunately, although the SIM is a smart card having both processing and storage capabilities necessary for new services, it is useless due to the lack of interfaces with input unit (keypad, microphone) and output units (display, loudspeaker). The SIM is supposed to be the slave executing orders from its master, the ME. To remedy this, the SIM Application Toolkit (SAT) is introduced to allow applications/services residing on the SIM to control the input and output units. It is actually an API between the SIM card and the mobile equipment (see Figure x), enabling applications stored on the SIM card to control other facets of the terminal, i.e., the user interface (display and input capabilities like the keypad) and also communication capabilities.

The SAT API is made up of proactive commands, which can be issued by the application on the SIM, and event downloads, which are events sent by the mobile equipment to the SIM (and eventually received by the application that registered for it). There are four categories of proactive commands:

- *Application commands* are used to control the user interface, i.e. showing information in the display and getting user input.
- *Smart Card commands* are used in interaction with an optional secondary Smart Card reader connected to the terminal.
- *General communications* commands are general purpose interfaces (e.g. to open a channel, send and receive data etc.) to access all communication bearers supported by the terminal.
- *System commands* are used to coordinate and synchronise operation with the mobile equipment and network, e.g. used by the SIM to obtain more processing time from the mobile equipment.

A standardised API is not enough for enabling dynamic services on the SIM; an actual runtime environment for the services to operate in is also needed. There are generally two different types of runtime environments for the SIM, namely a SAT interpreter and a SAT virtual machine.

A SAT interpreter is very similar to a browser (e.g. a WAP browser), although it can be even more lightweight and primitive. It is used to access transient pages of information. As such, saves storage space on the resource constrained SIM.

A SAT virtual machine is a dynamic runtime environment, which allows procedural programs to be downloaded to and run on the SIM. Thus, services provided through a virtual machine can be more advanced than simple, static pages provided through an interpreter, but this solution requires more and permanent storage space on the SIM.

Limitations and Achievements

With SAT it is possible to develop applications on the SIM but there are many restrictions. First, SAT applications should be small in size and developers must have access to SIM application development environment, which is both difficult and costly. Second, the installation of applications on the SIM is controlled by operators who are reluctant to open the access due to security. The results are that SAT applications are usually operator-owned and are typically security related since the SIM is a tamper-resistant device.

Recently, the JAVA SIM cards start to emerge and it will be very interesting to have collaboration between SIM JAVA components and JAVA components on the Mobile Equipment enabled by J2ME.

2) Mobile Execution Environment (MExE)

The Mobile Execution Environment (MExE) is aiming at facilitating the development of advanced services on the Mobile Equipment (ME), i.e. the mobile phone itself and ensuring their portability across mobile terminals by classifying their capabilities. MExE is more of a framework describing the requirements of runtime environments to support services, than an environment itself. MExE defines several classmarks that shall be used to designate the capabilities of terminals and the functionality required by services. The classmarks currently defined are:

- Classmark 1: WAP
- Classmark 2: PersonalJava
- Classmark 3: Java 2 Micro Edition (J2ME)
- Classmark 4: CLI Compact Profile

A service of MExE classmark X can be run on a MExE device of classmark X. Vice versa, a MExE device of classmark Y can run services of classmark Y. Classmarks are not exclusive, so a device can be capable of running services of both classmark X and Y. Clearly, a number of combinations are possible.

The primary goal of the MExE initiative is to standardise the requirements to open service runtime environments in order to allow the same services to run on devices from different device manufacturers.

MExE also defines 4 service access models. These are:

1. Remote execution of services in the MExE Service Environment (MSE)
2. Download of service clients from the MSE
3. Download of standalone applications from MSE
4. Peer-to-peer services through MSE

Limitations and Achievements

MExE is only a framework for standardising runtime environments on handheld terminals. As such, to have any value it must be adopted by terminal manufacturers as well as supported by service developers. Interoperability testing is thus important. 3GPP has previously launched interoperability tests, but the results and participation are not known.

3) Java 2 Micro Edition (J2ME)

Historically, Java has been denoted as a programming language that allows a developer to write an application once and run it anywhere. This was initially the goal of the J2ME platform. To have a Java environment on mobile terminals will improve service portability and facilitate service deployment; the Java programmer community is quite large and still increasing.

The J2ME platform is standardised through the Java Community Process (JCP), which consists of participants from the industry, and specified in Java Specification Requests (JSR). J2ME platforms can be composed to fit the resources and capabilities of a particular type of target device. Acting as a foundation for such an environment is a configuration, which consists of Java core libraries and a Java virtual machine (JVM).

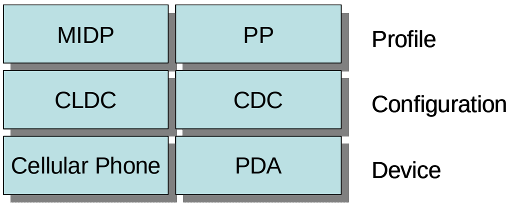


Figure 4 A J2ME platform consist of a device, a J2ME Configuration and a J2ME Profile

On top of a configuration a profile is added to provide further functionality to the user applications (see Figure 4). Several different configurations and profiles have been defined, but the pair suited for most of today's resource constrained mobile terminals is the Connected Limited Device Configuration (CLDC) [5] with the Mobile Information Device Profile (MIDP) [6] on top, which together results in minimal memory and processing requirements on the device and still provide a lot of functionality.

Another combination of configuration and profile is the Connected Device Configuration (CDC) [7] together with the Personal Profile (PP) [8]. This combination is resource demanding for current cellular phones, but within short time, cellular phones will be powerful enough to run this environment as well. Applications developed for the CLDC/MIDP environment should, in theory, run without modifications in a CDC/PP environment. Whereas MIDP has its own library for creating a graphical user interface (javax.microedition.lcdui), PP adds support for the standard J2SE Abstract Windowing Toolkit (AWT) and applet support.

State-of-the-art J2ME

The current specification of CLDC is version 1.1 and MIDP version 2.0. In MIDP 2.0, communication support has been extended from only HTTP to also supporting HTTPS, datagram, sockets, server sockets, and serial port communication.

Another major functional upgrade of the specification is the inclusion of a push model for MIDlet activation (MIDlets are the applications running on top of MIDP). This means that a MIDlet can remain in the background on the terminal until the runtime environment receives an event from the network, then activating it. This upgrade is very useful, and Instant Messaging is perhaps the first application to use it. MIDP 2.0 also adds end-to-end encryption of data through the use of Secure Sockets Layer (SSL) and Wireless Transaction Layer Security (WTLS).

Optional Packages for MIDP

Many valuable functions are left as optional packages, which are not included in the MIDP 2.0 profile. Some of these are briefly discussed next.

Wireless Messaging API – This is an API that allows access to native messaging functionality of a handset, e.g. SMS messaging capabilities in a GSM cellular phone. Such APIs can potentially expand the range of possible services based on short messages. Examples are intelligent automatic action based on incoming messages (e.g. filtering or statistical analysis etc.). However, the architecture of the J2ME Wireless Messaging solution does not allow direct access to the SMS inbox, and thus restricts this type of service development.

Web Services API – Web Services have a common goal with many other initiatives for mobile services (e.g., MExE and J2ME in general), namely platform independence. Remote, networked services should be accessible from any device regardless of what type of platform the service (server side) was developed and deployed on and what platform and implementation language was used for the client.

Security and Trust Service API – The final specification for this API was released 17th of June 2004, and the main contributions by the specification are APIs that provide security and trust services by using a Security Element (SE) on a terminal. An SE can typically be a Smart Card, and in the GSM context this usually means the Subscriber Identity Module (SIM). An SE provides secure storage of sensitive data, e.g., cryptographic data (like private keys and certificates) but also personal information and similar. Also, the SE can provide cryptographic operations which can be used together with the cryptographic data to support payment protocols, data integrity and data confidentiality. In addition, the SE can be used for custom security features used to realise other value-added services (e.g., authentication and identification).

MIDlet suites can be certified to belong to a particular protection domain, and access by MIDlets to restricted APIs on the handset can be differentiated among stakeholders in the mobile value chain based on the certified protection domain.

The defined protection domains are:

- Operator
- Manufacturer
- Trusted Third Party
- Untrusted

Limitations and Achievements

J2ME on handheld devices has until lately been best suited for standalone applications, due to the lack of advanced networking capabilities. With newer versions, distributed applications are easier to realise due to the existence of communication primitives like the socket abstraction.

Using XML Web Services can be an ideal approach for developing mobile services. If the Web Services API had been included in the MIDP 2.0 specification, the development of Web Services for mobile terminals would have been easier and the execution of them less resource demanding. Today, each client that needs the Web Services functionalities must either re-implement parsers and generators or integrate a 3rd party library (e.g. kSOAP[1]). This is a waste of resources on an already resource constrained device. On the other hand, some will argue that most handheld devices are still too constrained to consider the use of XML based protocols at all.

Another severe limitation is the lack of compatibility that removes the ultimate goal of Java. Several optional J2ME packages are specified. Furthermore, CDC implements a complete J2SE 1.3 API and has a complete Java Virtual Machine called the Compact Virtual Machine (CVM), whereas CLDC implements an almost complete J2SE API and utilises a Kilo Virtual Machine (KVM). Whereas MIDP has its own library for creating a graphical user interface (javax.microedition.lcdui), PP adds support for the standard J2SE Abstract Windowing Toolkit (AWT) and applet support.

B. Technology Enablers in the network

1) Open Mobile Alliance (OMA) Initiatives

OMA works with specifying market driven mobile service enablers with a primary goal to ensure service interoperability across devices, geographies, service providers and networks. OMA has taken over responsibility for the Wireless Application Protocol (WAP), and thus many of their standards are related to WAP, e.g., browsing, presentation of content, protection of content with digital rights management etc.

The following is a list of the specific areas OMA are working with:

- Browsing
- Transport and Presentation
- Messaging
- Push Services, Email, MMS
- Premium Content Consumption
- Digital Rights Management (DRM), Reliable Download
- Data Synchronization, Device Management
- Instant Messaging and Presence
- Location-based Services
- (M-commerce, mobile web services, mobile gaming etc.)

The results of OMA standardisation are called approved enabler releases. Some of the current approved enabler releases of particular interest for mobile services are:

OMA Data Synchronisation – These are universal specifications for data synchronisation. Data synchronisation must be used by services to allow access to the same information from many different locations and devices.

OMA Device Management – These are specifications that define how third parties can carry out procedures of configuring mobile devices on behalf of the end user. As devices become more complex, this is needed to ease the operation of devices for the customers.

OMA Digital Rights Management – These specifications enable controlled consumption of digital media objects. Management of content to ensure the rights of the content developers/providers is important in future mobile services, to ensure interest in providing such material.

OMA Mobile Web Services – Web Services is a middleware technology for creating and providing distributed services that decouple the service logic from specific platforms. It can prove particularly important for mobile services. XML Web Services will be treated in general in Section 3.2.5 of this paper.

Limitations and Achievements

OMA does not deliver platform implementations. Rather, the organisation provides specifications of service enabling functionalities, and it is thus left to others to implement the platforms and functionalities they describe. As such, it is not necessarily ensured that all implementations comply completely with the specifications and thereby allowing cross-platform service usage and service mobility.

On the other hand, interoperability testing is the only way to ensure that services can work across different platforms. The process that OMA follows, with designating enabler releases only after proper interoperability testing has been performed, is therefore an appropriate approach to work towards ubiquitously available services.

The common consumer is often not familiar with new features of technology; the device management initiative can ease the adoption of

[1] http://ksoap.enhydra.org/

new services by remotely applying the necessary configuration on an end-user device. Indeed, the many parameters needed for WAP configuration can be seen as one of the reasons for the slow adoption of this technology (although other reasons like small displays have also had an influence).

2) Parlay/Open Service Access (OSA) APIs

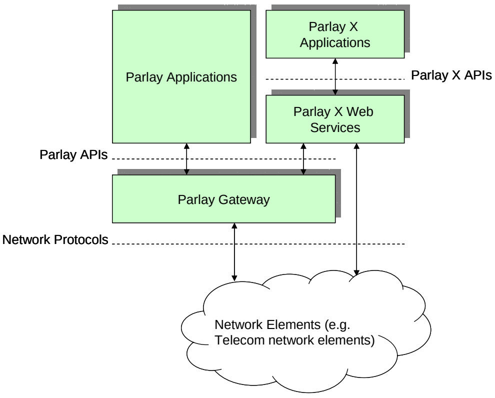


Figure 5 The Parlay APIs with several abstraction levels

The Parlay Group specifies open and technology independent APIs in cooperation with ETSI and 3GPP, referred to as the OSA/Parlay APIs [10]. These APIs allow a maximum number of market players to develop and offer advanced telecom services by making it possible for application servers not in the telecom network to interact with telecom network capabilities. It also allows development of applications that can across multiple networks.

Since the OSA/Parlay APIs should be technology independent, they are defined using UML [11] notation. However, the specifications include a normative appendix that describes a CORBA realisation using OMG Interface Definition Language (IDL) [12] and an informative appendix describing a SOAP realisation using the Web Services Description Language (WSDL) [13].

In addition to the original OSA/Parlay APIs there exists a Parlay X Web Services specification [14] which keeps a higher abstraction level (see Figure 5) and allows even easier access to the features provided by a telecom network operator.

Features, which are provided access to through the OSA/Parlay APIs, are called Service Capability Features (SCF) and they are provided by a Service Capability Server (SCS) in the telecom network. The OSA/Parlay specifications cover a lot of ground and are currently divided into 14 parts. The following is a list of functional groups of APIs which describes the access to SCFs:

- Policy management
- Presence and availability management
- Account management
- Call control
- Charging
- Connectivity manager
- Data session control
- Framework
- Generic messaging
- Mobility
- Terminal capabilities
- User interaction

In addition to these APIs, which are for specifically accessing features (service capability features) of a network, the Parlay APIs also include framework functionality. Between an application and the framework the basic mechanisms are authentication (optionally mutual) between applications and framework, authorisation of access to features by applications, discovery of framework and network service capability features, establishment of service agreements and access to network service capability features. The basic mechanism between the framework and a service capability server is the registering of network service capability features, whereas between the framework and an enterprise operator there is a service subscription function.

Limitations and Achievements

Parlay and OSA only provide specifications of interfaces, which leave implementers with the responsibility for utilising the specifications in a proper manner.

However, Parlay eases development of services in a number of ways. First, the APIs are open and platform independent. This means that services can be developed and deployed in a maximum number of heterogeneous environments and still interoperate with the required domain specific functionality (e.g., telecom functionality). Second, they define APIs with several abstraction levels. This eases the efforts needed by developers to start realising services; less domain-specific knowledge is needed and also less programming efforts are needed. Developers with CORBA knowledge can utilise this, but accessing functionality through XML Web Services eases development and debugging of services even more and improves the flexibility by utilising e.g., WSDL for publishing the APIs.

3) Virtual Home Environment (VHE)

VHE [15] is a concept for portability of a Personal Service Environment (PSE) across network boundaries and between terminals. It is defined as part of the UMTS standard through ETSI/3GPP processes. VHE relies on several other technologies for different platforms for realising its concepts. Of these, we only consider MExE (and in particular J2ME), OSA/Parlay and U/SAT in this paper. These have already been discussed. Figure 6 illustrates the relationships and reliance of VHE on other specifications and technologies.

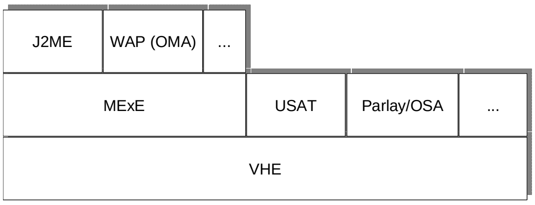


Figure 6 VHE is a concept which depends on several other specifications and technologies

Of greatest importance to a PSE are the user profiles. In these, all information that is needed to render personalised services is stored. Two aspects of profiles are crucial; the construction of the profiles is important to ensure that they can be dynamically changed and the distribution of them across the various domains is important to ensure their availability everywhere the PSE should be present.

Management of the profiles is considered an important part of the VHE, and so is the correct identification of a user's personalised data and service information.

In addition, the VHE is concerned with:

- Being able to provide and control services to the user in a consistent manner also if the user is roaming
- Provide the necessary means to create and maintain a set of user profiles
- Support the execution of services – through its service toolkits in the network, the USIM and in the ME
- Uniquely identify the user in the telecommunication networks supported by the home environment

Limitations and Achievements

VHE tries to tie together other initiatives that target different parts of the mobile services hosting domain. It is very useful to have such a high-level concept to guide the development of other standards. However, it can be very difficult to both define and realise a concept based on standards and technologies that do not consider this high-level goal themselves.

4) Session Initiation Protocol (SIP)

SIP (Session Initiation Protocol) is an important technology enabler for mobile services because it enables telephony over IP. Till now, with GSM, mobile telephony is always based on circuit-switched networks. With the popularity of IP, it is desirable to extend mobile telephony on IP-based networks. SIP is a signalling protocol for voice over IP (VoIP) and the most recent version (2.0) is specified in RFC3261 [16]. The protocol is text based, extremely flexible (thus also complex) and basically defines how to setup and tear down calls between two parties, or across a more complex architecture involving proxies, gateways to the Public Switched Telephone Network (PSTN) and other components. In addition to voice-over-IP applications, support for messaging and presence services has been added as part of the protocol.

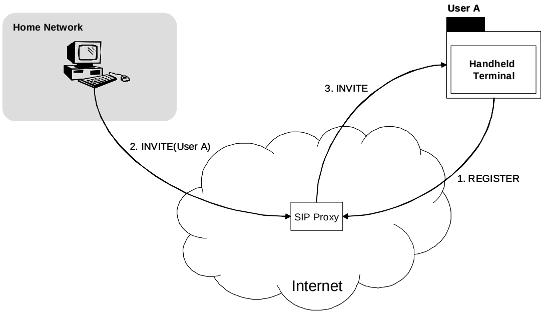


Figure 7 Simplified illustration of mobility handling in SIP

SIP depends on the Session Description Protocol (SDP) [17] for describing the specific capabilities of each end-point and the Real-time Transport Protocol (RTP) [18] is used for transport of voice data over User Datagram Protocol (UDP) [19]. The Real-time Transport Control Protocol (RTCP) [18] can in addition be used to dynamically change the behaviour of the RTP implementation, e.g. to increase/decrease the size of jitter buffers and similar.

SIP includes its own mobility enabling feature through the use of registrars and incoming proxies (see Figure 7). A SIP client (owned by User A) registers its current location (IP address) and identity (UserA) with a SIP Registrar/Proxy. When another SIP client (here illustrated as a computer in a home network) wants to contact UserA, it sends the request to the SIP Proxy and addresses the user with identifier UserA. The identifier will be the same even if User A moves to another access network or terminal, as long as it registers its current location with the SIP Proxy. This type of mobility is called personal mobility.
Today, mobile voice services are realized by circuit-switched technologies like GSM.

However, in the future, these services could in theory be provided using IP and SIP. One effect to this is to improve service continuity further. The same service could be accessible across different networks (GSM and Internet) and domains (Home and Enterprise). This does not mean that voice is enabled as a service all these places, voice services are readily available already at these locations already. What it means is that a user can be reached by the same unique identifier no matter if he is at work, home or travel, as earlier illustrated by the Device Unifying Service (DUS) [20].

Limitations and Achievements

It is very likely that SIP has a role in mobile services in the future. However, due to bandwidth requirements (64kbit/s for the G.711 codec a-law/u-law) as well as processing requirements on the terminal for encoding and decoding the RTP streams, current cellular phones are not yet suited for using SIP for voice-over-IP. However, for some Personal Digital Assistants (PDAs) with for example the XScale 400 MHz processor, this is possible today. The interoperability between SIP implementations from various vendors is a matter of concern. Since the protocol is flexible, extensions added by some vendors are not necessarily understood and handled properly by other implementations. The key to success here is to implement forgiving solutions that do not crash when unexpected SIP messages are received.

On the other hand, the SIP specifications include at lot of other usage scenarios than voice, e.g., for messaging and presence services. These do not impose the same requirements to bandwidth and processing power as voice-over-IP and are already possible to realise for restricted devices like cellular phones.

5) Web services

Today, a new computing model for building distributed software systems is emerging: Web services. They are a collection of standards for developing, deploying and providing flexible, platform independent services that are distributed through the Internet. A typical Web service consists of a Web service client and a Web Service server (which is commonly referred to as the Web service itself). The client can invoke operations on the server, which in turn returns the result. The minimum set of enabling technologies and standards of a Service-oriented architecture utilizing Web services are:

Web Services Description Language (WSDL) – These are XML documents describing the nature of a service, which includes the methods that can be invoked, and their parameters, as well as the return data type.

Simple Object Access Protocol (SOAP) – SOAP is the XML-based protocol used for communicating the service invocations between client and server. These invocations follow a request/response programming model, similar to HTTP; in fact, HTTP is the usual bearer for SOAP, although SOAP can use other transport like SMTP.

Universal Description, Discovery and Integration (UDDI) – This is a registry where services can be published. The specifications define how services are registered by service providers and how services can be discovered by service consumers (clients).

Limitations and Achievements

As SOAP travels across HTTP, it will pass through most firewalls that accept HTTP traffic. This is beneficial from a service availability view, but might pose a security risk also. However, anyone making an HTTP server accessible on the Internet should always take appropriate measures for keeping malicious code away from a production server. SOAP could therefore not be blamed directly for posing great risks.

As illustrated by earlier sections, Web Service technology has been adopted by several initiatives (e.g., J2ME, OMA and OSA/Parlay) as a model for service distribution and invocation. It has quickly become a recognised solution to distribution. Web service development is supported by most development platforms, also by open source platforms from Apache. This means that the technology is readily at hand.

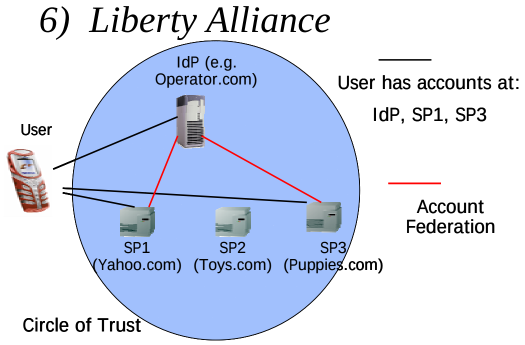


Figure 8 Circle of trust and the Account Federation

Today, a user's identity is fragmented across different service providers. A typical user has a lot of accounts for e.g. web-based services. Due to this situation, service creation and service usage can be cumbersome and time consuming. In addition, using mobile, handheld devices for this purpose decreases the usability significantly. The goal of Liberty Alliance is to enable seamless and simplified mobile web transactions across providers, without compromising security and privacy. The Liberty Alliance provides technology and certifications to include secure identification of users as a component of mobile and web based services. The standards developed by Liberty Alliance are open and platform-agnostic specifications to support a federated network identity. A federated network identity comprises the following key concepts (illustrated in Figure 8):

A *Circle of Trust* – This is an affiliation of identity providers (IdP) and service providers (SP) based on Liberty-enabled technology and on operational agreements.

An Account Federation – User accounts within the circle of trust are linked between service providers and identity providers.

The Actual Service Usage – Users can now perform web transactions within the circle of trust in a secure, simplified and seamless manner. The users authenticate with the identity provider and hop across linked service provider accounts.

Limitations and Achievements

To ease account management for users and provide the proposed benefits, the Liberty Alliance technology must become widespread among service providers. Also, a lot of enterprises that are trusted by consumers must take on the role as identity providers. For mobile services accessed through cellular phones and telecommunication networks, it would be natural for a telecom operator to take on this role. However, the identity provider role is not similarly obvious for other types of services provided through other networks.

The technology specified by Liberty Alliance eases the consumption of mobile services n general, and has additional value for services accessed through small and unmanageable devices. The technology also takes care of users' privacy and makes transactions secure. These are all key functions to ascertain a growth in both the availability and usage of mobile services in the future.

IV. Conclusion

In this paper, the major technology enablers for mobile services are presented in a comprehensive way and in relation with each other. Mobile services are subject to requirements not only from end-users and mobile network operators but also from wireless application service providers, the content providers and equipment manufacturers. Each technology enabler is usually aiming at only a subset of the requirements and can be incomplete, inconsistent and overlapping with other technology enablers.

According to the analysis of technology enablers presented in this paper, it is possible to conclude many technology enablers are aiming at promoting the diversity of mobile services by offering easy and flexible service development on different environments such as in the SIM card with SIM Application Toolkit), in the mobile terminal with J2ME, at third parties with OSA/Parlay and in the Internet with Web services. OMA proposes enablers for the construction of rich services such as instant messaging, multimedia messaging and synchronisation. The OMA intends to satisfy the requirement from the content provider with the Digital Right Management. SIP is aiming at enabling IP telephony on mobile networks and hence promoting convergence of the mobile networks and the Internet. Other technology enablers have the user on focus such as Liberty Alliance for the user's identity management and VHE for the user's personalisation of services.

Unfortunately, VHE is only a concept and quite a lot remains to be done to enable personalisation of services. The topics like user profile structure, distribution, operation, management and interfaces with the mobile services are especially relevant for further work. Another important issue is how to provide service continuity across different domains controlled by different players.

REFERENCES

[1] Renato Iannella, Digital Rights Management Architectures, D-Lib Magazine, Volume 7, Number 6, ISSN 1082-9873, June 2001

[2] Jørstad, I., van Do, T., Dustdar, S. (2004). Personalisation of Future Mobile Services. 9th International Conference on Intelligence in service delivery Networks, Bordeaux, France, 18-21 October 2004.

[3] ETSI, Digital cellular telecommunications system (Phase 2+) (GSM), Universal Mobile Telecommunications System (UMTS), Mobile Execution Environment (MExE), Functional description, 2002

[4] Jørstad, I., van Do, T., Dustdar, S. (2004). An analysis of service continuity in mobile services. 2nd International Workshop on Distributed and Mobile collaboration (DMC), WETICE conference, 14-16 June, Modena, Italy, 2004, IEEE Computer Society Press.

[5] Sun Microsystems Inc., JSR-139: Connected Limited Device Configuration 1.1, Final Release, March 2003, available online at: http://java.sun.com/products/cldc/index.jsp

[6] Sun Microsystems Inc., JSR-118: Mobile Information Device Profile 2.0, Final Release, November 2002, available online at: http://java.sun.com/products/midp/index.jsp

[7] Sun Microsystems Inc., JSR-218: Connected Device Configuration 1.1, Under Public Review, available online at: http://java.sun.com/products/cdc/index.jsp

[8] Sun Microsystems Inc., JSR-62: Personal Profile 1.0, Final Release, available online at: http://java.sun.com/products/personalprofile/index.jsp

[9] Sun Microsystems, Java Card Platform Version 2.2.1, Application Programming Interface, October 2003, available online at: http://java.sun.com/products/javacard/specs.html

[10] The Parlay Group, Parlay 4.1 Specification, available online at: http://www.parlay.org/specs/archive.asp

[11] Martin Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition, Addison-Wesley, ISBN 0-321-19368-7

[12] Object Management Group, Common Object Request Broker Architecture: Core Specification, Version 3.0.3, March 2004, available online at: http://www.omg.org/docs/formal/04-03-01.pdf

[13] World Wide Web Consortium, Web Services Description Language, Version 2.0, Part 1: Core Language, August 2004, available online at: http://www.omg.org/docs/formal/04-03-01.pdf

[14] The Parlay Group, Parlay 4.0: Parlay X Web Services Specification, Version 1.0.1, June 2004, available online at: http://www.parlay.org/specs/index.asp

[15] 3rd Generation Partnership Project, The Virtual Home Environment, Service Aspects (Release 5), 2002, available online at: http://www.3gpp.org/ftp/Specs/2004-09/Rel-5/22_series/

[16] Internet Engineering Task Force, RFC3261: Session Initiation Protocol, 2002, available online at: http://www.ietf.org/rfc/rfc3261.txt?number=3261

[17] Internet Engineering Task Force, RFC2327: Session Description Protocol, 1998, available online at: http://www.ietf.org/rfc/rfc2327.txt?number=2327

[18] Internet Engineering Task Force, RFC1889: A Transport Protocol for Real-Time Applications, 1996, available online at: http://www.ietf.org/rfc/rfc1889.txt?number=1889

[19] Internet Engineering Task Force, RFC0768: User Datagram Protocol, 1980,

available online at: http://www.ietf.org/rfc/rfc0768.txt?number=0768

[20] Erik Vanem, Dao Van Tran, Tore Jønvik, Pål Løkstad, Do Van Thanh: Managing heterogeneous services and devices with the Device Unifying Service implemented with Parlay APIs, Proceedings of the 8th IFIP/IEEE Symposium on Integrated Network Management, Colorado Springs, Colorado, USA, 24-28 March 2003

Prof. Dr. Do van Thanh obtained his MSc in Electronic and Computer Sciences from the Norwegian University of Science and Technology and his PhD in Informatics from the University of Oslo. In 1991 he joined Ericsson R&D Department in Oslo after 7 years of R&D at Norsk Data, a minicomputer manufacturer in Oslo. In 2000 he joined Telenor R&D and is now in charge of PANDA (Personal Area Network & Data Applications) research activities with a focus on SIP, XML and next generation mobile applications and terminals. He holds also a professorship at the Department of Telematics at the Norwegian University of Science and Technology in Trondheim. He is author of over 100 publications at international conferences and journals. He is inventor of 14 patents and a dozen of pending notes.