Towards a Uniform IMS Client on Heterogeneous Devices

Do van Thanh – Telenor & NTNU, Norway – thanh-van.do@telenor.com
Paal Engelstad – Telenor & Unik, Norway – paal.engelstad@telenor.com
Tore Jønvik – Oslo University College, Norway – tore.jonvik@iu.hio.no
Do van Thuan – Linus, Norway – t.do@linus.no
Ivar Jørstad – Ubisafe, Norway – ivar@ubisafe.no

Abstract – The paper presents current IMS client frameworks, and points out shortcomings of these for deployment in a mobile-fixed convergent multi-access IMS environment. Based on this analysis, a new IMS client framework is proposed. A prototype implementation is also presented.

Keywords: IMS client, IMS client framework, IMS terminals, SIP client

I. INTRODUCTION

In the traditional mobile telecommunication world, mobile phones were considered as “terminals”, i.e. devices terminating the mobile network systems. Quite often, they were viewed as peripheral devices outside the system that did not deserve much attention from the operators. This negligence has led to the absence of mobile phones at the launch of 3G networks in the early 2000s which lasted many months and caused a lot of losses to mobile operators. Unfortunately, this lesson seems not to be learned and the same situation is about to happen again with the launch of IMS (IP Multimedia Subsystem) [1][2]. Indeed, an acceptable IMS mobile phone is still missing. The major mobile terminal manufacturers has, of course come up with their IMS mobile phones but these phones are, unfortunately not fully compatible and interoperable with each other. Most importantly, they fail to support the concept of rich communication promised by IMS. To remedy the situation, there are currently several international activities at OMA (Open Mobile Alliance) [8], OMTP (Open Mobile Terminal Platform) [9], JCP (Java Community Process) [10], Eurescom, RCS (Rich Communication Suite) [11], etc. aiming at specifying a sound and realistic IMS client framework. This paper presents some of the IMS client works done in the EUREKA Mobicome project, which is aiming at providing unified subscription for a Fixed-Mobile Convergent IMS environment. Basically, the paper presents an innovative implementation of the IMS client, which provides a uniform user experience across a range of heterogeneous devices.

The paper presents related works in Section II and state-of-the-art of current IMS client frameworks in Section III. Section IV presents requirements to an enhanced IMS client framework that is target at deployment in convergent, multi-access environments, with heterogeneous devices. Based on these requirements, a novel IMS client framework is proposed in Section V, and a prototype implementation of this is presented in Section VI. Finally, conclusions follow in Section VII.

II. RELATED WORKS

Recognizing the urgent need for a standardized and well-defined IMS client, there are currently several international activities related to IMS clients. Some of the most significant works will now be briefly summarized.

A. The Java Community Process

Recently, the Java Community Process acknowledged the need to define client APIs for IMS networks and started work on JSR 281 [3]. The standardization work is lead by Ericsson and BenQ. The JSR 281 architecture shown in Figure 1 below identifies three layers within an IMS client:

- Application layer
- IMS Engine layer
- Protocol stacks layer including the network.

The Java API is the boundary between Java applications and the IMS Engine layer. The JSR specification contains the Core package and the Service package. The Core API is mandatory and includes IMS functionalities such as IMS session management and monitoring, QoS settings, Security and Access functions.

The Service API is optional, and covers the IMS services PoC (Push-to-Talk over Cellular) [4], IMS Presence [5] and XDM (XML Document Management) [6], and utilities to support those activities.

1 The EUREKA Mobicome project (Nov 2007-april 2010) has as partners: Telenor. Telefonica, HuaWei, HyC-Ericsson, Polytechnical University of Madrid, Oslo University College, Blekinge Institute of Technology, Linus, Ubisafe and WIP.
B. The Open Mobile Terminal Platform (OMTP)

OMTP (Open Mobile Terminal Platform) is an operator-sponsored forum founded in June 2004 that serves the needs of each and every link in the mobile phone value chain by gathering and driving mobile terminal requirements. OMTP released the first version of the IMS Client Framework requirements v.1.0 report in May 2007.

This recommendation defines requirements for:

- **IMS Terminals:** for example the terminal MUST allow simultaneous CS and PS communications
- **Protocols:** SIP, SDP, IPv4, IPSec(3GPP), DHCPv6, DNS, RTP, RTCP, RTSP, HTTP/XCAP, HTTPS, SigCOMP
- **IMS Core:** Session Management, QoS Management, Basic Messaging, Registration and Authentication.
- **Applications:** Functionality expected from IMS terminals regarding the integration of IMS applications within the terminal IMS framework (for example redirecting IMS messages to applications and application configuration)
- **OMA defined Service Enablers:** OMA PoC, OMA Presence, OMA XDM, IM SIMPLE (not released yet by OMA)
- **Application Programming Interface (API):** Core API and Service API (optional)
- **Service Enabler for Video Share (GSMA)**

C. The Rich Communication Suite initiative

The Rich Communication Suite initiative is a joint effort to facilitate the evolution of mobile communication towards rich communication from a group of key operators, infrastructure and device vendors comprised of Orange, Telecom Italia, Telefónica, TeliaSonera, Ericsson, Nokia Siemens Networks, Nokia, Sony Ericsson and Samsung founded in March 2008. RCS is aimed primarily at helping operators meet the consumer demand for richer communication. This initiative will make the total process - from implementation to customer care - easier and less expensive for operators. Further, it attempts to align consumer needs with device and network capability.

III. EXISTING IMS CLIENT FRAMEWORKS

There are a number of commercial IMS clients on the market provided by different companies, including Aricent, Conneon, Ecrío, Ericsson, Nokia, Pctel, NMS communications, Radvision, etc. In addition, there are some open source initiatives where IMS Communicator might be the most well-known. The majority of these IMS clients are using a layered IMS client framework as shown in Figure 2.
The signalling plane is typically divided into a number of sub-layers (Figure 3). In this layer, the Transport sublayer takes care of the transmission of actual IMS messages according to the set of protocols used. The Transaction sublayer takes care of a single IMS transaction, which can be a 2-way request/response or a 3-way request/response (e.g. INVITE), and must handle request/response matching and reliability (i.e. retransmissions and timeouts). The Dialogue sublayer binds together different transactions within the same session, which is identified by the ‘To’ and ‘From’ tags and the Call-ID. Finally, the Call sublayer deals with the entire IMS call, which might consist of a number of IMS transactions.

**Figure 3: The signalling plane is typically divided into different sublayers**

The signalling plane is typically divided into a number of sub-layers (Figure 3). In this layer, the Transport sublayer takes care of the transmission of actual IMS messages according to the set of protocols used. The Transaction sublayer takes care of a single IMS transaction, which can be a 2-way request/response or a 3-way request/response (e.g. INVITE), and must handle request/response matching and reliability (i.e. retransmissions and timeouts). The Dialogue sublayer binds together different transactions within the same session, which is identified by the ‘To’ and ‘From’ tags and the Call-ID. Finally, the Call sublayer deals with the entire IMS call, which might consist of a number of IMS transactions.

**IV. FUNCTIONAL REQUIREMENTS**

The Mobicome project aims at enhancing state-of-the-art of IMS client frameworks to a fully convergent, multi-access environment, based on shortcomings identified for existing client frameworks presented in the previous section.

The first observation is that today, there is often a tight binding and nearly one-to-one mapping between the IMS-based application and a specific underlying service enabler of the IMS client framework (Figure 1). Instead, we aim at letting various applications provide services composed of a number of underlying IMS service enablers at run-time. This means that the IMS client framework is capable of supporting rich communication. The Rich Communication Suite (RCS) is an effort using IMS to enhance mobile phone communication with interoperable, convergent and rich communication. It is also seen as an initiative to help IMS take off by accelerating IMS and SIP development.

The rich communication concept means that a number of standardized services shall be available to consumers. These services include presence & buddy lists, voice calls, instant messaging (IM), video share, image share, SMS and MMS. A generic API for rich communication is required between the applications and the enablers in Figure 2.

IMS was originally designed by the 3rd Generation Partnership Project (3GPP) for the 3G mobile networks [7]. This is probably the main reason that the IMS client frameworks seen today are mainly developed with mobile phone usage in mind. Currently an IMS client implemented in a mobile phone will not be able to operate in a wireless LAN environment. It is not only because of the differences between the 3GPP-SIP of IMS and plain IETF-SIP, but also because of the differences in terms of fundamental subscription differences between the mobile world and the fixed one. Indeed, in the mobile world, each user is associated to one subscription, while in the fixed network a whole household with many users can share one subscription. Furthermore, the authentication schemes used in the WLAN hotspots vary widely and they are different to the ones used in the mobile network. Consequently, due to the lack of compatibility, the IMS services will not work in the WLAN hotspots. As a result of these fundamental differences, there is no satisfactory mechanism defined for smooth handover between a wireless LAN environment and a 3G network.

The motivation of the Mobicome project, however, is to expand IMS to cover the fixed broadband networks as well, enabling it to be a common service delivery platform on top of fixed, mobile and IP-based networks. This calls for an IMS client framework that is developed for heterogeneous devices, and that can be used for both mobile phones, stationary PCs, set-top-boxes and so forth.

A related shortcoming is that existing client frameworks do not accommodate SIP (a.k.a. IETF-SIP) and IMS (a.k.a. 3GPP-SIP) within the same common framework. This means that today, a user will often need two different sets of client frameworks on his mobile device; one for SIP and one for IMS, even if the application might want to be able to use both, and might also want to switch between the two on-the-fly, e.g. during a handover between the mobile domain (e.g. using UMTS) and the fixed domain (e.g. using fixed broadband access in combination with WLAN). Instead of using two different frameworks, there should be a common client framework that accommodates the use of both IETF-SIP[14] and 3GPP-SIP[15].

Finally, yet another problem with most existing IMS client framework is that they are not open. Either the software is shipped with the embedded device (e.g. phone) or the software is proprietary. The users – and application developers – are dependent on the features of the client framework without having the opportunity to influence its functionality. Not only should the client framework be open (as it for the few open source IMS clients available), they should also be customizable in run-time, allowing full flexibility in controlling the functionality dependent on network access type, SIP-version in use, type of authentication, and so forth.

In summary, we argue that a future-proof IMS client framework shall
- be executable on heterogeneous devices, stationary or mobile,
- be agnostic of the network access, and support smooth handover between the mobile and the fixed broadband domains,
- support both 3GPP-SIP (IMS) and plain IETF-SIP,
- be customizable,
- support rich communication,
- provide a uniform user interface on multiple heterogeneous devices.
- have a loose coupling between the application layer and the service enabler layer.

With a client framework having these features, the complexity of dealing with both the IETF-SIP domain with fixed broadband access and the mobile IMS domain with UMTS access is handled by the client framework instead of by each separate application. The client framework also handles additional complexity that involves the two domains, such as handover between the two domains and authentication within the two domains.

From a middleware perspective, the advantages of this approach are evident: Implementing these capabilities within the IMS client framework means that we are taking the implementation burden away from the application developers. Moreover, user experience might be better, since many functions (e.g., such as handover and authentication) perform better when implemented further down the communication stack, simply because the lower layers of the stack are involved in the realization of the given function.

V. THE MOBICOME IMS CLIENT FRAMEWORK
A. Architecture
The new IMS client framework must implement both plain SIP and IMS signalling. It does so by supporting the IMS extensions of 3GPP-SIP in addition to the plain IETF-SIP signalling.

![Figure 4: Signalling Plane supporting both IETF-SIP and 3GPP-SIP IMS extensions](image)

In order to realize a new IMS Client Framework that satisfies the requirements described in the previous section, one has to deviate a little from the traditional layered model. The reason is that the new framework requires tight control with functionality at different layers. To deal with this challenge, a cross-layer control plane is introduced in addition to the Media plane and Signalling plane of the original client frameworks (Figure 5).

![Figure 5: The new IMS Client Framework introduces a Control Plane with modules for cross-layer control](image)

First, the control plane contains a module for internal routing, that makes sure that a request initiated by an application is directed to the right service enabler, and that an IMS/SIP response that comes back from the network for this request is directed back to the same service enabler and application.

A module for SIP/IMS selection is also required. For example, a generic request from an application or service enabler needs to be mapped into actual SIP/IMS signalling. The module investigates the networking conditions, and decides which kind of SIP shall be used, be it 3GPP-SIP or IETF-SIP.

Furthermore, session transfers and handovers, which might affect other modules in the control plane (such as SIP/IMS selection), are also handled by a separate module. Likewise with authentication control.

Moreover, as Figure 5 shows, we have also moved the media control module out of the Media Plane, and put it logically as a separate module in the Control Plane.

Finally, to enable the composition of applications offering rich communication, the Mobicome IMS Client Framework will include APIs representing all the service enablers like Presence, VoIP, PoC, etc. This API is shown in the upper part of Figure 5.

B. Thinner IMS client
Figure 6 shows the traditional way of implementing an IMS Client, namely deploying directly at the user’s client device.
To enable IMS on a range of heterogeneous devices with different capabilities, we propose instead to move much of the functionality of the IMS client to a network server. With this new proposal, much of the capabilities are implemented on and accessed from network servers (Figure 7). Indeed, the new proposed IMS client framework is network agnostic and allows for this kind of architecture.

The advantages of this approach is that simple mobile devices do not need the complete and complex IMS client framework, but only need a simple module to control the media delivery and to access the media client framework on the Mobicome proxy server.

The application has features for controlling media delivery, and a module that communicates and controls the functionality of the Mobicome proxy.

C. Web Service[16] API

As observed in Figure 7, the Mobicome proxy is implemented with an Application Server part that accesses the underlying client framework (with various service enablers etc.) through an API. (This API corresponds to the Rich Communication API shown in Figure 5.).

With a thin-client deployment, as shown in Figure 7, it makes sense to implement this API as an XML Web Service API which reduces the construction of applications to an orchestration using for example BPEL[17]. It is crucial that for each service enabler only a few simple Web Service methods are defined. This is to avoid repeating the weakness of Parlay, i.e. having a complete but rather complex set of APIs.

D. Using the browser as GUI

As a further development of the thin-client architecture, it is proposed to use the web-browser as the GUI when accessing a service. This kind of thin-client implementation is illustrated in Figure 8.

The client-side module can for example be implemented as a Browser plugin, so that it is easy to integrate call control with web-technology, allowing for easy implementation of click-to-call services and easy integration of IMS/SIP and Web 2.0 [18] technology.

The browser connects to the web server part of the Mobicome proxy, which accesses the Client framework through a Web Services API.

Implementing browser-based services has a number of advantages. First, it is easy to customize, and it is easy to add new functionality. Furthermore, it gives a uniform User Experience over heterogeneous devices. Indeed, IMS Clients with this kind of implementation can be deployed on any type of devices and still maintain a uniform user experience. For differentiation, service providers can deploy widgets instead of the simple web browser/plug-in scheme.

VI. IMPLEMENTATION

To verify the feasibility of the proposed IMS Client Framework concept, a prototype has been implemented as shown in Figure 9. The user can use three devices to browse and visit the IMS GUI Web server:

- **PC with Generic Bootstrapping Architecture (GBA) Smart dongle**: The login authentication is done using GBA[19]
PC with SIM Strong Authentication Smart dongle: The login authentication is done using the SIM strong authentication[20]

Cellular phone with HTML[21] browser: The login authentication is done using SIM Strong Authentication via SMS[22].

The prototype is working properly and has been demonstrated at the Mobile World Congress in Barcelona in February 2008[24].

VII. CONCLUSION

The paper presents current IMS client frameworks, and points out shortcomings of these for deployment in a mobile-fixed convergent multi-access environment. Based on this analysis, a new IMS client framework is proposed. The new framework is executable on heterogeneous devices, stationary or mobile, is agnostic of the network access, and supports smooth handover between the mobile and the fixed broadband domains, and between usage of both 3GPP-SIP (IMS) and plain IETF-SIP. It aims at being customizable, supporting rich communication and providing a uniform user interface on multiple heterogeneous devices. Furthermore, a thin client architecture is proposed, where a web-browser is used for GUI and where the client framework functionality is accessed through a web-services API on a server located in the network. A prototype implementation is also presented.

REFERENCES

[4] Open Mobile Alliance (OMA) PoC (Push to Talk Over Cellular) V1.01, online: http://www.openmobilealliance.org/release_program/poc-v1.0.1.html
[5] Open Mobile Alliance (OMA) Presence SIMPLE V1.01, online: http://www.openmobilealliance.org/release_program/Presence_simple_v1.0_1.html
[8] Open Mobile Alliance (OMA), online: http://www.openmobilealliance.org/
[15] ETSI/3GPP (2003), “Universal Mobile Telecommunication System (UMTS); Signalling Interworking between the 3GPP profile of the Session Initiation Protocol (SIP) and non-3GPP SIP usage (3GPP TR 29.962 version 6.1.0 Release 6)”, ETSI
[19] ETSI/3GPP (2008), “Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); SIM card based Generic Bootstrapping Architecture (GBA); Early implementation feature (3GPP TR 33.920 version 7.5.0 Release 7)