

Facilitating Mobile Service Provisioning in IP Multimedia Subsystem (IMS) Using Service Oriented Architecture

Igor Radovanović¹, Amit Ray², Johan Lukkien¹, and Michel Chaudron¹

¹ Technische Universiteit Eindhoven, Den Dolech 2,
5600MB Eindhoven, The Netherlands

² Atos Origin Technical Automation Nederland B.V., De Run 1121,
5503LB Veldhoven, The Netherlands

{i.radovanovic, j.j.lukkien, m.r.v.chaudron}@tue.nl
amit.ray@atosorigin.com

Abstract. This paper presents an extension of the IMS software architecture using a service orientation, which provides flexibility of mobile service provisioning. The suggested extension facilitates composition of new mobile services in run-time based on the existing services and enables the end users to become mobile service providers. Moreover, it enables addition of new mobile services with plug-n-play similar to an addition of an end user device to the network. With this extension, the core system architecture of IMS is not affected. The extension is realized by introducing Web services in combination with the SIP protocol. Using SIP will enable operators to remain in control of mobile service provisioning as they are the owners of the IMS networks. This property of the extension may also facilitate its acceptance by the operators. A proof-of-concept prototype is discussed to demonstrate feasibility of the proposed extension.

Keywords: Service Oriented Architecture, IP Multimedia Subsystem, SIP, Web services.

1 Introduction

The mobile systems beyond 3G in Europe are aiming at providing enhanced user experience using various mobile applications anytime, anyplace using any type of end user devices [1]. The main motivation is to meet user's wishes to be able to choose among variety of mobile services (applications) and to communicate in an intuitive way.

However, the new systems will also give a boost to operators as their networks will not be used as bit pipes only by the service providers providing the mobile applications. With the new system in place, they will get the possibility to provide mobile applications to the end users as well. These mobile applications comprise the information retrieval services (e.g. weather forecast, stock reports and car navigation), the multimedia services (e.g. video-on-demand) and the communication services (e.g. messaging, voice).

To enhance the user experience, operators intend to introduce and exploit the IP Multimedia Subsystem (IMS) SIP-based technology [2,3] that provides flexible IP media management and a session control platform, which they can lay on top of their current IP-based and (virtual) circuit-switched infrastructures. In Fig. 1, the basic elements of the IMS architecture are shown [4]. The SIP

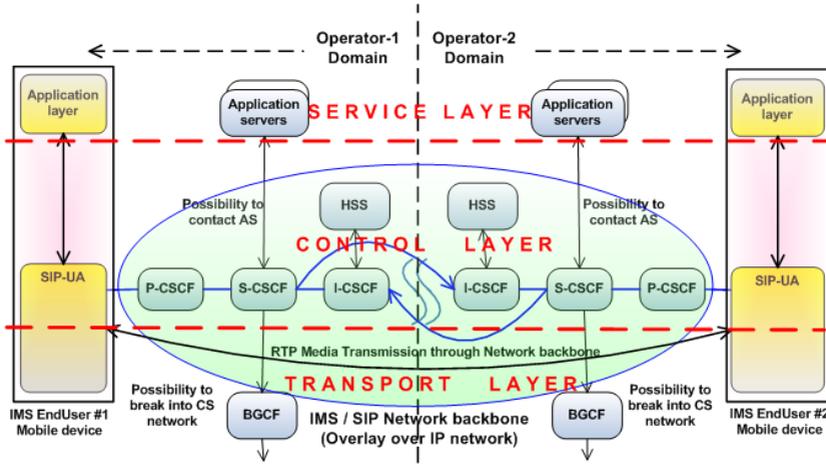


Fig. 1. The IMS system architecture

signalling is used during session establishment, maintenance and termination. The application servers (AS) provide mobile applications to the end users and the other AS's belonging to the same network administrative domain. The CSCF (Call Session Control Function) elements play a role during a user registration and a session establishment and form a SIP routing machinery in IMS. The HSS (Home Subscriber Server) is the main data storage for all subscriber- and service-related data of IMS in an operator's network. BGCF (Border Gateways) are used for connecting IMS to the circuit-switched telephone network. Other elements of the IMS infrastructure (like Resource List Server, Presence Server etc.) are not shown for the sake of clarity of representation.

2 The Effect of Limitations of the IMS Architecture

A lack of the proposed IMS architecture is that it does not provide remote access to mobile services provided by the third parties from a different administrative domain as the interfaces for using those mobile services are not standardized across different network administrative domains and there is no standard way of publishing these services. This is also important if the mobile services offered to the end users are to be built from other mobile services at run-time. For example, if we consider a car navigation service, it could be possible that an operator providing this service to the end users actually needs to compose it

from other services like a location service and a map service, possibly offered by the operators in different network domains.

One consequence of these drawbacks is the loss of revenue for the operators that lose opportunity to offer enabling services to other operators. Another one is a limited possibility to enhance user's experience as the system is unable to provide the tailor-made services to the end users that are particularly suitable to support their needs. Finally, a mobile service user might have a different experience when using mobile services provided in different network domains.

The current IMS architecture also fails to provide end users with a possibility to share their services over the network, just like they can do that over the Internet (e.g. using peer-to-peer technology [5]). The cellular terminals are mostly statically provisioned with a fixed number of services, and there is little possibility for an easy upgrade of services, which is traditionally done in the Internet. Another drawback of the current IMS architecture is that information conveyed in the SIP protocol does not inherently provide separation of mobile services and devices on which those are running.

To overcome these drawbacks, an architecture that (1) increases flexibility of mobile service provisioning, (2) allows end users to become service providers and (3) incrementally extends the IMS architecture, should be used. A solution in this paper is based on using a Service Oriented Architecture overlayed on the existing IMS infrastructure. By introducing web services on top of the SIP protocol, it is shown that new mobile services could be dynamically introduced from an end user device and that these services could easily become accessible outside the operators' domain. The operators still retain control of the system, owing to their ownership of the respective SIP based networks. This work is an extension of the work done by other authors ([6], [7]) who discussed combining SIP protocol (a part of IMS technology) with Web services, but not in the context of IMS.

3 Extended Software Architecture

Fig. 2 shows that a new Web service layer is placed between the top Application layer and the SIP connection layer. This results in a slightly modified software architecture in the end user devices. The Web service layer will represent the mobile services from a remote AS. In cases when the end user also acts as an AS, the Application layer will have its own web service part, which will expose its local services to the Web service layer. The Web service layer will in turn, expose the local services to the outside world by sending out the WSDL information through the NOTIFY (or PUBLISH) message of the SIP stack. For the SOAP request messages received from the Application layer, the Web service layer will forward these SOAP requests through the SIP stack. With this architecture, the services from a value added AS (VAS) would be available to users outside the operators domain. However, usage of the services of an existing AS would still be restricted, and might be used primarily for control of the network by the operator. The IMS core network infrastructure is not affected by this proposed

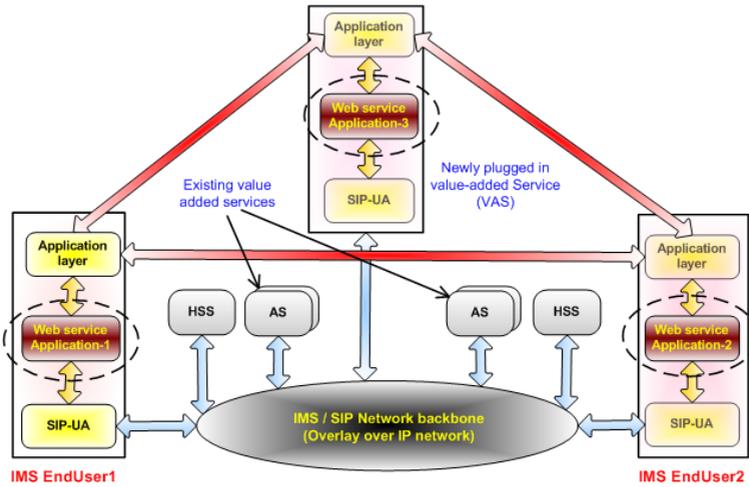


Fig. 2. A modified end-user software architecture in SOA-enabled IMS

architecture. However, if there are services provided by the existing ASs that are not used for network control purposes, these might better be moved over to the plug-n-play VASs under the operators control. This way the services would be available to a broader section of users.

4 Implementation

The interaction between the Application layer and the Web service layer will be in both directions. The Application layer asks for remote services from the Web service layer and the Web service layer asks for local services from the Application layer. In addition to the usual service and proxy components in the Application and the Web service layers, the Web service layer contains elements for (un-) wrapping the SOAP messages (from) to the SIP messages. For this, the design from [6] is extended to suit the proposed IMS architecture.

In the representation of the IMS end user in Fig. 3, the "Proxy for Remote services" in the Application layer is an object locally representing web services from the Web service layer. There is a corresponding object in the Web service layer representing the service(s) from the remote AS, and it acts like a call forwarder by forwarding all SOAP requests from the end user to the remote AS. The "SOAP message constructor" component in the Web service layer remembers the origin of the SOAP requests and the WSDL messages by maintaining a table of remote AS versus a SIP message ID. Additionally, this component has the responsibility to combine the received WSDL messages from various external sources and present a composite web service interface for the currently available remote services. There are tools available to build the proxy class automatically from the WSDL description of a web service. For example, in Microsoft's ASP.NET, a tool named WSDL.exe can build the proxy class. However, to realize

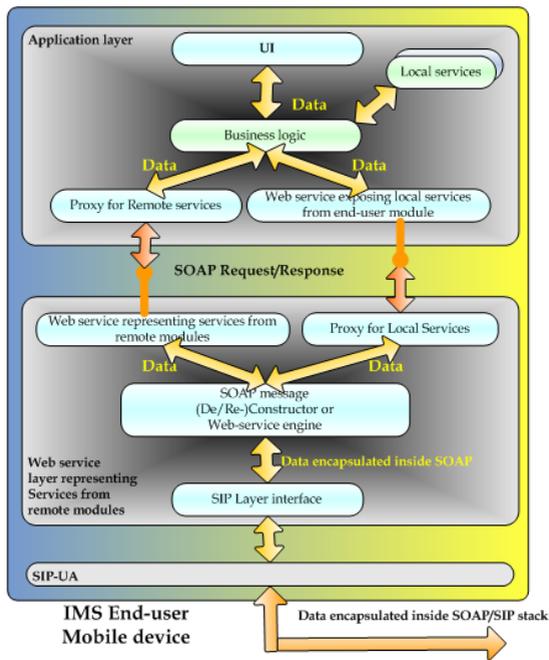


Fig. 3. The IMS End-user Module

the availability feature to its full potential, it is required that update of service states is done at run-time. In order to have this feature, the relevant components in the Application layer and the Web service layer should be dynamically updated based on the WSDL from the Web service layer. The dynamic creation and update of classes is possible with many of today's high level programming languages (.NET, Java) using a feature called "Reflection". A general guideline from [8] is adapted here for implementing this with Microsoft's .NET framework tool for ASP.NET development.

There are 3 implantation constraints at the terminal. The terminal should be equipped with all the features required to communicate as an IMS end user device (SIP-UA). Additionally, it must contain a running web server that can send and receive http messages and it must have the run-time environment for a high level language that supports reflection (e.g. Java runtime, .NET Framework).

5 Experiments

As a proof of concept, we built two software based IMS end user modules (UserA and UserB) with layered structures, similar to that shown in Fig. 3. Both modules offer a "Text" (i.e. messaging) service that can be turned on and off through a checkbox control in the UI. Although, this is a very simplistic interpretation of the "Text" service, this is done in order to simplify the implementation of the

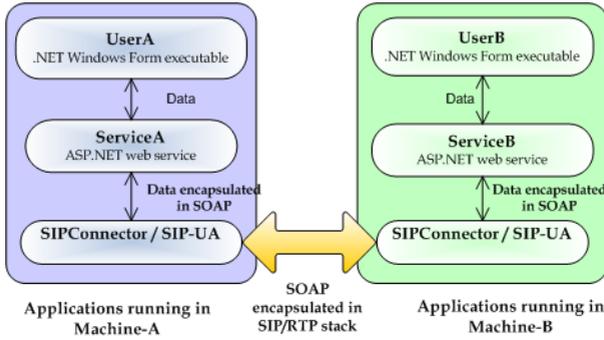


Fig. 4. Experimental setup

prototype without diluting the concept behind it. The experimental set-up is shown in Fig. 4. For the SIP protocol stack, the third party freeware named PJSIP is used, that provides the SIP protocol implementation in a C library. The two Microsoft .NET based user interface applications (UserA and UserB) are also created to enable activities like sending out a text message and displaying messages received from the remote user. The prototype User Interface (Fig. 5) has the text boxes to specify the address of the other end user it wants to connect to. A green color button indicates the availability of the “Text” service at the other end. If UserA starts to type in some characters in the box “Msg to User

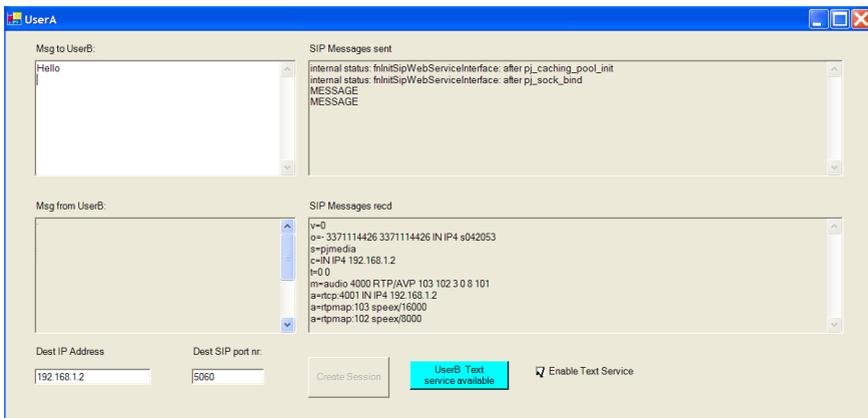


Fig. 5. An UserA console

X” (X being either A or B), the same characters start to appear in the box “Msg to User Y” (Y being either B or A) at the UI on the other machine. In the meanwhile, the boxes designated “SIP messages sent” and “SIP messages recd” display the SIP message exchanges between the two ends. The SIP message “MESSAGE” is used here to send text messages to the other user.

The "SIP messages recd" box shows the media type information received from the other user (User B) in SDP protocol format. The experiments are performed with the simple text here, but the same could be extended to any data, including video. When the check-state of the "Enable Text Service" checkbox of a user is changed, the other user's UI immediately reflects this changed service availability state. This illustrates the possibility of service *availability* as opposed to device availability only. Fig. 6 shows the scenario for the experimental set-up. Note that

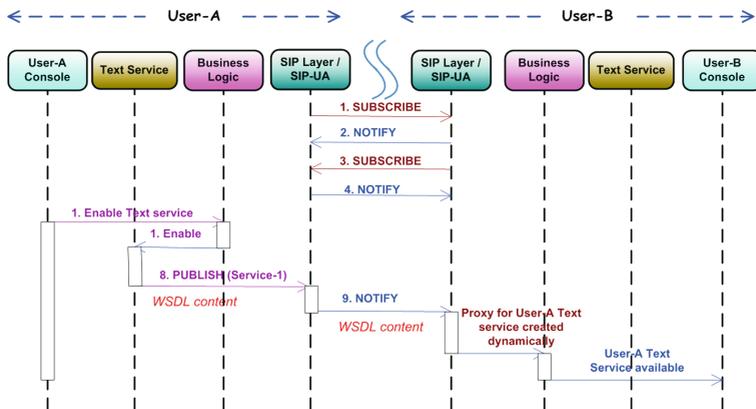


Fig. 6. A scenario for the experimental setup

there is no distinction made between an end user and an Application server. From the principle of web service applications, a web service from an end user can be reused by other web service applications. Moreover, a new end user (with a new web service) could be added to the service infrastructure as easily (flexibly) as plugging in a new IMS end user device to the network.

6 Conclusions

The extension of the IMS capabilities through a service orientation presented facilitates composition of new mobile services from the existing ones and introduction of new mobile and enabling services. It also enables the end users to become service providers. The new IMS software architecture does not require changes in the core IMS architecture, and provides a solution to the SIP message extension, including SIP version mismatch.

References

1. CONverged MESSaging Technology (2007), <https://www.comet-consortium.org>
2. SIP Architecture: (2007), http://www.protocols.com/pbook/sip_arch.htm
3. IETF working group on SIP/SIMPLE web page (2007), <http://www.ietf.org/html.charters/simple-charter.html>

4. Poikselkä, M., Mayer, G., Khartabil, H.: Niemi: The IMS – IP Multimedia Concepts and Services, 2nd edn. John Wiley & Sons, Ltd., West Sussex, England (2006)
5. Milojevic, D.S., et al.: Peer-to-Peer Computing, tech. report HPL-2002-57, Hewlett-Packard Laboratories, Palo Alto, Calif. (2002)
6. Liu, F., et al.: WSIP – Web Service SIP Endpoint for Converged Multimedia/Multimodal Communication over IP. In: ICWS'2004. Proceedings of IEEE International Conference on Web Services, pp. 690–697. IEEE Computer Society Press, Los Alamitos (2004)
7. Liscano, R., Dersingh, A.: Projecting Web Services using Presence Communication Protocols for Pervasive Computing. In: MCETECH2005 Web Services Workshop, Montreal, Canada (2005)
8. Google discussion group item on Microsoft dot net framework web service (2007), <http://groups.google.nl/group/microsoft.public.dotnet.framework.aspx.webservices/>