Mobile P2P Web Service Creation using SIP

Guido Gehlen 1), Fahad Aijaz 2), Yi Zhu 3), Bernhard Walke 4)

Abstract:
Telecommunication networks and the Internet are growing together. Peer-to-Peer (P2P) services which are originally offered by network providers, like telephony and messaging, are provided through VoIP and Instant Messaging (IM) by Internet service providers, too. The IP Multimedia Subsystem (IMS) is the answer of the telecommunication industry to this trend and aims at providing Internet P2P and multimedia services controlled by the network operators infrastructure. The IMS provides mobility and session management services as well as message routing, security, and billing.

This paper introduces a Mobile P2P Web Services framework which enables the creation of arbitrary P2P application on mobile devices within the IMS. The terminals are sharing and publishing a Web Service interface, which provides the capability of coupling arbitrary applications using Web Service technologies and the support of the IMS infrastructure. The concept, the necessary Web Service adaptations, and extension as well as the interworking of Web Services, IMS nodes, and protocols are described and proofed by an exemplary mobile chess game application. In addition, performance considerations are performed by means of measurements.

1 Introduction

The success of the Internet and Web Services along with the success of mobile networks in the last few years let forecast a high economical potential of the combination of both, called Mobile Web Services. From technological point of view this step is ongoing, but many challenges have to be coped to combine technologies from the Internet and Web domain with technologies from the telecommunication domain.

The IMS is from the network infrastructure point of view a promising system to enable

1)RWTH Aachen University, Faculty 6, Communication Networks, Kopernikusstr. 16, 52074 Aachen, guge@comnets.rwth-aachen.de
2)RWTH Aachen University, Faculty 6, Communication Networks, Kopernikusstr. 16, 52074 Aachen, fah@comnets.rwth-aachen.de
3)RWTH Aachen University, Faculty 6, Communication Networks, Kopernikusstr. 16, 52074 Aachen
4)RWTH Aachen University, Faculty 6, Communication Networks, Kopernikusstr. 16, 52074 Aachen
the interworking of the telecommunication and Information Technology (IT) domain. On
the network layer the convergence has already been realized by the introduction of Internet
Protocol (IP) service through the General Packet Radio Service (GPRS). This line is further
continued by introducing IPv6.

Telecommunication services are naturally P2P services, e.g. a telephony/video call session,
or the exchange of messages. But these are only a narrow subset of P2P services. In future,
also “sharing” applications, like sharing the personal photo album, a whiteboard, or a multi-
player games will be normal for users of mobile terminals. In general, arbitrary services could
be shared with the contacts of the buddy list. To enable such applications, the application
developers need strong support in order to cope with the mobility of the users, the changing
networks and network conditions, management of fluent application sessions, changing envi-
ronmental and user contexts, variety of device types and operating systems and variety of
network types and protocols.

The proposed Mobile Web Services framework is hiding application developers the complexity
of the overall system and providing services which transparently take the above listed chal-
lenges into account. The IMS is used for session/mobility management in order to enable a
seamless transparent sessions even if the Radio Access Network (RAN) or the currently used
terminal changes. For example, a user playing a chess game with a friend on a PC can switch
without restarting the game to his mobile terminal keeping the chess session running while
the user changes his location. In addition, new network types and conditions do not cause a
session termination (mobility support, roaming and message routing are applied). Furth-
more, the IMS enables the billing of arbitrary services, either online or offline, dependent on
the business model.

Application developers are supported by the concept of the Service Oriented Architecture
(SOA) and Web Services technologies to focus more on the actual application than on dealing
with device or communication specific features. Using these technologies, a distributed appli-
cation is as simple to handle as a local software component. The Web Services are provided
and published like conventional services, e.g. a laundromat, a post office, or a translator.
The application developer has the ability to search and investigate services of his interest by
accessing a registry like somebody is looking up the yellow pages.

The Mobile Web Service framework described in the following focuses on the contact points of
IMS and Web Service technologies, their interworking, and enhancements. These adaptations
enable a seamless integration of Mobile Web Service in the IMS (combination of SOAP and
SIP) and an enhanced Web Service communication over mobile networks using alternative
protocol bindings 2.4.

An important aspect of Mobile Web Services is their performance, since mobile networks are
offering frequently changing and in general bad Quality of Service (QoS). For Mobile Web Service applications the latency of Remote Procedure Call (RPC) calls is the most important QoS parameter which has to be reduced to a minimum in order to guarantee fluent interactive applications. A brief conclusions of Mobile Web Service performance based on measurements is given in section 2.4.

2 Mobile Web Services

The term “Mobile Web Services” is not clearly defined and is used with different meanings (in different contexts/domains). In this work the term “Mobile Web Services” is used in any case where mobile networks and devices are involved in Web Service interactions. In the following, the Mobile Web Service domain is further divided into 3 sub-domains.

A “Mobile Web Service” (Mob-WS) is according to the definition in [1] a self-contained software component identified by an Uniform Resource Identifier (URI), i.e. an Internet address or a SIP URI, which is deployed on a mobile device in a wireless/mobile network. In short, a Mobile Web Service is a Web Service which is deployed on a mobile and published within a wireless/mobile network.

Basically, Mobile Web Services are classified in this work into three classes, classified by means of the Web Service requestor (WS-R) and provider (WS-P) deployment (see figure 1). The deployment of the Web Service broker (WS-B) is irrelevant.

The Mobile Web Service classes are

1. Mobile Web Services Request (mobile WS-R/fixed WS-P)
2. Mobile Web Services Provisioning (mobile WS-R/fixed WS-P)
3. Mobile P2P Web Services (both peers are WS-R and WS-P)
Web Services access from mobile devices is working with no or with slightly changes of Web Service technologies, e.g. the improvement of Web Service communication over mobile networks using alternative protocol bindings. The provisioning of Web Services and Mobile P2P Web Services demand more efforts and changes on the default Web Service technologies, i.e. mobility support to enable Mobile Web Service mobility and session management by using Mobile Web Services in a P2P manner. There, the cooperation with the IMS is vital.

According to the Mobile Web Service definition and the classification of the last sections, a Mobile P2P-Web Service is containing of the following components form the software architecture point of view:

- SOAP node and SIP UA
- Mobile Web Service Proxies (SOAP client)
- Mobile Web Services (SOAP server)

These components are introduced and described in section 2.3, but before, the service creation mechanism 2.1 and the protocol point of view 2.2 are discussed.

### 2.1 Service Creation

Mobile P2P Web Services can be used in order to simplify the sharing of arbitrary application interfaces with other peers. These services have to be well defined, specified, and identified by an unique Mobile Web Service ID. If there are different Mobile Web Service interfaces with the same functionality a standardization have to take place or application developers have to implement for one application different interfaces in order to enable the interworking of nearly all kind of applications of this domain.

For example, somebody has designed a chess game Web Service interface, one can load the Web Service Description Language (WSDL) description and generate with common Web Service tools the needed server and client components in the programming language of the specific device. Using this middleware as a basis, the application developer can implement the Graphical User Interface (GUI) and logic which is appropriate to the device capabilities. Thus, the same Mobile Web Service interface definition of the exemplary chess game can be implemented for instance on a mobile game console with a 3D graphic environment in C++ on Symbian OS and for a simple mobile phone under Java2 Micro Edition (J2ME) and rudimentary graphical capabilities. The interworking of these application is always warranted in case the Mobile Web Service interface definitions has been used. The automated process of the interface integration in the programming environment guarantees the standard conform implementation.
2.2 Mob-P2P-WS Protocol Architecture

From the protocol point of view, one can arise two possible arrangements of combining the Web Services protocol Simple Object Access Protocol (SOAP) [2] with Session Initiation Protocol (SIP) [3]. SOAP can be used on top of SIP or in parallel on the same layer.

Using SOAP on top of SIP in the user (data) plane enables a transmission of SOAP messages inside of the body of a SIP message. This is for sure an appropriate alternative to the default case of transmitting SOAP messages inside of a Hypertext Transfer Protocol (HTTP) message, but from the protocol point of view it makes no difference. HTTP and SIP have a similar syntax and similar services.

In [3] SIP is declared only as an application layer signaling protocol. Thus, this work is focusing on the use of SIP on the control (signaling) plane in parallel of SOAP in the user plane, as depicted in figure 2. This strict separation of user and signaling plane has advantages in respect to protocol design, communication software design, and performance (network load of SIP infrastructure is reduced).

![Figure 2: Mobile Web Service User and Control Plane](image)

On the User (Data) Plane the application specific user data is transmitted over SOAP on top of various alternative underlying Internet protocols (using different SOAP bindings, see 2.4). On the Control (Signaling) Plane, SIP is used to transmit “application layer” signaling messages.

2.3 Mob-P2P-WS System/Software Architecture

In order to support the mentioned extensions and changes of Web Service technologies to Mobile Web Services the following architecture is proposed. The Mobile Web Service endpoint is a SIP URI which is resolved by the IMS infrastructure to the actual URI based on the terminal’s IP address. An unique Web Service ID is used for each Mobile Web Service in order to easily identify and register Mobile Web Services. This ID is used in the session establishment of a Mobile P2P Web Service session (inside the SDP session description) and by registering the Mobile Web Service to the network infrastructure (SIP/IMS).
The integration of Mobile Web Services into the IMS entails the design of interfaces between Web Service and IMS, SIP components. Figure 3 is showing the Mobile Web Service components of two user terminals, UT1 and UT2. UT1 is acting as a Web Service Proxy and terminal UT2 is providing the Mobile Web Services. Generally, each terminal is able to provide and use Mobile Web Services at the same time and within the same SIP session.

![Figure 3: Mobile Web Service and SIP UA integration](image)

The Mobile Web Service and proxy components are not only connected to SOAP nodes, but also to a SIP UA. Thus, the Mob-WS signalling is enabled as described in section 2.2 and 2.5. The Mobile Web Services and Proxies have to register to the SIP UA in order to be notified about IP address changes (mobility management) and session establishment/release events (session management).

### 2.4 Alternative SOAP Bindings

SOAP is a transport neutral mechanism for exchanging messages. This means that a SOAP message can be carried by any transport or application layer protocol as far as it conforms to the formal set of rules defined for carrying a SOAP message. The specification defining these rules for transmitting SOAP within or on top of the underlying protocol is called SOAP Binding.

By default SOAP is bound to HTTP. Focusing on mobile networks and short RPC message transmission, the default HTTP binding entails high latencies due to a bad performance of the Transmission Control Protocol (TCP) under these conditions caused by TCP’s connection establishment (3-way handshake) and the congestion control (slow-start phase), see [4].

Therefore, a binding of SOAP to the User Datagram Protocol (UDP) as an alternative to the default HTTP binding has been proposed, see [4], which enables a fast transmission of SOAP messages over UDP. Naturally, UDP enable only a connection-less and unreliable transmission, but a further improved binding introduces a Selective Repeat Automatic Repeat Request (ARQ) protocol on top of UDP which enables a fast and reliable transmission of short SOAP messages. Long-lived data transmissions, e.g. huge SOAP messages with attachments should use the default binding to HTTP and TCP in order to avoid a congestion of the network. In addition, the unreliable UDP binding can be used to advertise and discover Universal Plug...
and Play (UPnP) services.

The correlation of SOAP messages within a SOAP Message-Exchange Pattern (MEP) is realized by using the standardized Web Service Addressing [5] specification. All messages are identified and assigned by using an Universally Unique Identifier (UUID). The reliable UDP binding reuses these IDs in order to assign incoming datagrams to the corresponding SOAP message. The Selective-Repeat ARQ mechanism is working with a window size which is equal to the SOAP message size divided by the UDP payload size. For more details, please see [4].

![Figure 4: Comparative analysis of HTTP and Reliable/Unreliable UDP SOAP binding running on a phone emulator connected by ethernet. Depicted is the mean round-trip time (RTT) averaged over 30 executions of each service](image)

In figure 4 the latencies by means of round-trip times (RTT) of Mobile P2P Web Service calls between two mobile emulators are depicted. The emulator platforms are connected by an ethernet cable. Different “echo” methods are invoked 30-times per method. The averages of the RTTs is depicted and show that a short SOAP message exchange performs 4 to 6 times better using a UDP binding. If a mobile network, like UMTS or GPRS is used, the difference is even more obvious due to the higher transmission delays of the mobile networks.

### 2.5 SIP based Mobility- and Session Management

A distinguishing characteristic of Mobile Web Services is the mobility of the service, since the users with their terminals regular change their location and, thus, the RAN they are connected with (RAN handover). In addition, equal Mobile Web Services are may deployed on different terminals (only the interfaces are the same). This enables the user to start a Mob-WS session on one terminal and to continue later the session with another terminal (terminal handover).

The use of Mobile Web Services in a P2P manner is enabled by establishing a SIP session between the terminals (see figure 5). The Mobile Web Service endpoints are SIP URIs, the Web Services endpoints on both clients are URIs containing the current IP address.

During the SIP session establishment, the unique Mobile Web Service IDs are shared in terms of the OFFER/ANSWER model of the Session Description Protocol (SDP) [6]. The SDP offer
is carried inside of the \textit{SIP:INVITE} message. The SDP answer could already be transmitted with the provisional response.

The following example illustrates a SDP OFFER for a Mobile Web Service session. The unique Mob-WS ID
\url{http://www.comnets.rwth-aachen.de/Mob-WS/IDs/MChess4711}
labels the service and is used to register the service within the IMS resp. a SIP \textit{REGISTRAR} server. In addition, the Mob-WS interface description and format of the SOAP messages is defined in a WSDL document located at
\texttt{a=wsdl:\url{http://schemas.rwth-aachen.de/example.wsdl}}.

\begin{verbatim}
v=0 o=- 2890124 2890124 IN IP4 comnets.rwth-aachen.de s=- t=0 c=IN IP4 0.0.0.0 m=data 80 HTTP application/soap+xml a=contact:http://www.comnets...de/Mob-WS/MChesss1 a=direction:passive a=Mob-WS-ID: http://www.comnets...de/Mob-WS/IDs/MChess4711 a=wsdl: http://www.comnets...de/Mob-WS/schemas/MChess.wsdl
\end{verbatim}

It is also possible to associate another session, such as an audio session, while having a SOAP session. Thus, a game session could be coupled with an audio or video session in order to have a chat about the current game.

The corresponding SDP \textit{ANSWER} either confirms or rejects the session establishment in case the technical requirements of the invited peer are not fulfilled.
After the invited peer, UT2 in figure 5, accepts the Mob-WS session, the eventual response
\textit{SIP:200 OK} is sent back and a final \textit{SIP:ACK} message heralds the start of the Mob-WS
session.

During the session, both peers (terminals) are using the Web Service of their counterparts. After finishing the session, both peers are able to close the session by sending from the SIP UA a \textit{SIP:BYE} message. In order to enable a terminal handover or the freezing of a Mob-WS session, the application state has to be captured and submitted either to the terminal that continues the session or to the IMS resp. SIP infrastructure. A re-establishment of the session, identified by the SIP session ID (SIP dialog), is possible, if the SIP dialog can be located within the session history of the IMS, e.g. on the Home Location Register (HLR).

\subsection{2.6 Mobile P2P Chess game example}

In order to proof the concept, a chess game for J2ME has been extended with a Mobile Web Service interface in order to enable a game session between two users over the network. Doing this, Mob-WS methods are defined which move the pawn in a game. The Mob-WS framework is providing functionalities to easily expose these methods as a Web Service using either the default HTTP binding or a reliable or unreliable UDP binding.

The Mob-WS game session establishment is illustrated in figure 6.

![Figure 6: Mobile Chess session establishment sequence](image)

Using a conventional buddy list where all contacts of the users with their present states are listed. The contact list and the presence state management is defined in the IMS specification resp. the SIP presence event package [7].

In addition to the presence state, the intersection of own and the contact persons Mob-WS
are indicated by a symbol. In this example, the user’s terminal and the all of his contacts are providing the same chess game Mob-WS, which is indicated by a “knight” symbol.

In figure 6 user 1 on the left hand side invite user 2 to a chess game session. The SIP UA’s starting the SIP session establishment and exchanging within the OFFER/ANSWER procedure their interfaces related to that application. First, the SIP:INVITE request is answered by the provisional response SIP: 180 Ringing and after the user 2 has accepted the session, the response SIP:200 OK is received by the user’s 2 terminal. An SIP:ACK message ends the Mob-WS session establishment and the SOAP nodes are configured in order to exchange SOAP messages between the terminals. The SOAP messages are containing the user data (chess maneuvers). If the terminal or the access network changes during the session, SIP mobility management signalling takes place and the SOAP nodes will be reconfigured.

3 Conclusion

The paper has presented a novel P2P service creation framework based on Mobile Web Services and a SIP infrastructure, like the IMS. The service Creation is based on a Mobile Web Service middleware supporting HTTP and UDP client and server bindings which are coupled with SIP user agents. The framework has been proved by an exemplary mobile chess game for J2ME devices.

References