

Mobile Web Service Provisioning

Satish Narayana Srirama¹, Matthias Jarke^{1,2}, Wolfgang Prinz^{1,2}

¹*RWTH Aachen, Informatik V*

Ahornstr.55, 52056 Aachen, Germany

²*Fraunhofer FIT*

Schloss Birlinghoven, 53754 Sankt Augustin, Germany

{srirama, jarke}@i5.informatik.rwth-aachen.de

wolfgang.prinz@fit.fraunhofer.de

Abstract

This paper, discusses the confluence of two major recent trends in distributed information systems engineering: the evolution from static content via personalized adaptive information provisioning to Web Services, and the emergence of mobile terminals with sufficient speed to serve as parts of information systems. The combination of both trends yields the idea of mobile Web Services. While a few pioneering mobile Web Service client systems have appeared recently, this paper explores the logical next step. Whether it is feasible to use mobile terminals such as Smart Phones also as Web Service providers ("Mobile Hosts"). We first discuss the desirability of such Mobile Hosts, then discuss the challenges for design and implementation, and finally present a prototype implementation that has been developed and evaluated in cooperation with a major mobile phone vendor.

1. Introduction

It is well accepted by now that the Internet can be seen as a large-scale distributed information system with numerous information users and providers. From the view-point of information systems engineering, this has influenced three major trends:

- the evolution from static content to Web Services,
- the evolution from client-server systems to peer-to-peer and pervasive computing systems, and finally – in conjunction with novel developments in wireless communication technology –
- the trend from stationary to mobile distributed information management.

Web Services are software components that can be accessed over the Internet using established Web mechanisms and protocols such as SOAP and HTTP

[2]. Public interfaces of Web Services are described using Web Service Description Language (WSDL). Examples of Web Services range from simple requests, such as stock quotes or user authentication, to more complex tasks, such as comparing and purchasing items over the Internet.

With the introduction of Third and Interim Generation mobile communication technologies in the cellular domain like UMTS, GPRS/EDGE, the speed of wireless data transmission has increased significantly. Also processing power and device capabilities of mobile phones have increased drastically, thereby enabling better applications and usage of mobile devices in different application domains.

Combining these developments it is a logical next step to turn mobile devices into wireless Web Service requestors (clients). This enables communication via open XML Web Service interfaces and standardized protocols also on the radio link, where today still proprietary and application- and terminal-specific interfaces are required. Mobile Web Service clients lead to manifold opportunities to mobile operators, wireless equipment vendors, third-party application developers, and end users. It is easy to imagine that in the future mobile applications based on Web Service clients will generate a large percentage of all Web Service requests, and the first such solutions are currently appearing on the market. [5, 9]

However, in a sense, this role of mobile Web Service clients is still basic and the combination of cellular and Web Services domains would only be completed if it would become feasible to also offer standard Web Service providers on small mobile devices. In this paper, we explore this idea based on experiences from a cooperation project with a major mobile phone provider in which the prototype of such a

Mobile Host was developed and tested on a smart mobile phone. The paper is organized as follows:

Section 2, defines the concept of mobile Web Service provisioning, whereas section 3 describes uses and use cases for Mobile Hosts. Section 4 discusses the challenges faced in designing and implementing such a Mobile Host, and outlines a general solution architecture. Section 5 describes our prototype implementation and its evaluation results, and section 6 discusses conclusions and further research.

2. Basic architecture of mobile Web Service provider (“Mobile Host”)

The basic architecture for Web Services is built upon its three components: Service Requestor (Client), Service Provider and Service Registry. The service provider publishes its Web Services with the service registry. The service requestor searches (“Find”) the UDDI registry for the services, and the UDDI compatible service registry refers the respective WSDL. The service requestor accesses the described Web Service, using SOAP. [1]

Similar to this, the basic architecture of the mobile terminal as Web Service provider can be established as shown in *figure 1* with the Web Service provider (from now on: “Mobile Host”) being implemented on the Smart Phone.

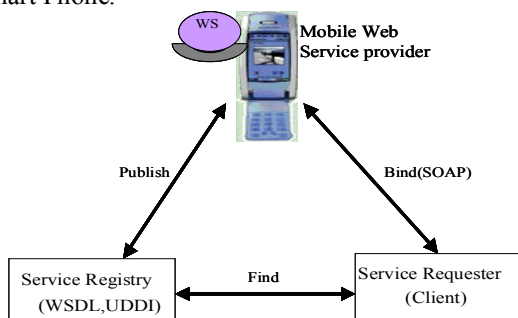


Figure 1. Basic architectural setup of Mobile Host

Even though the Web Service provider is implemented on the Smart Phone, the standard WSDL can be used to describe the services, and the standard UDDI registry can be used for publishing and unpublishing the services. This of course presents the challenge to design the mobile terminal with the same general architecture as on any standard desktop system, even under the low-resource considerations of the Smart Phone.

Alternate architectures for mobile Web Service provisioning are also possible with SOAP complaint proxy or gateway in between the Mobile Host and the Web Service requester. The communication between the client and the proxy is using SOAP and the

communication between the proxy and the Mobile Host would be using a protocol, efficient for the data transport across the mobile networks. Many such proprietary protocols and implementations have evolved like WSOAP, gSOAP, eSOAP, etc. The following paragraphs give a brief discussion of two of these protocols.

Wireless SOAP (WSOAP) [12] is basically a set of optimization techniques. The WSOAP aims to provide static encoding based on SOAP schema, leverages WSDL service description to create adaptive encoding for WS interfaces, limits computational cost, and concentrates on functional message equivalence rather than exactness. This protocol can be extremely useful between mobile devices and gateways where the resources are very limited as WSOAP can reduce SOAP message sizes by 3-12 times.

The gSOAP toolkit is a platform-independent development environment for C and C++ Web Services [13]. gSOAP provides transparent SOAP API through the use of compiler technology that hides irrelevant SOAP-specific details from the user. The compiler automatically maps native and user-defined C and C++ data types to semantically equivalent SOAP data types and vice-versa. As a result, full SOAP interoperability is achieved with a simple API.

We are considering this option of using SOAP proxy in our study, which handles the security and the scalability issues, there by providing better QoS for the Mobile Host. For scalability issues, compression technologies with Fast Web Services [14] are being considered. SOAP compression is highly efficient in the mobile Web Services domain, because of the poor connectivity and high communication costs of the mobile networks. [16]

But the processing of these proprietary protocols might require additional resources like processing power and time, delaying the response time of the Mobile Host. In addition having such a proxy or gateway might affect the interoperability of the Web Services. Moreover for the study addressed by this paper, we considered only the basic architecture, as we wanted to check the feasibility and performance analysis of having such a standard Mobile Host on the Smart Phone.

3. Applications of Mobile Host

Mobile Hosting of Web Services opens up a new set of applications that has not been explored very much up to now, since a realization seemed beyond the resource capabilities of present mobile terminals. In the subsections below, we therefore explore usability and usefulness of mobile Web Service hosting.

Generally speaking, the Mobile Host has the following major potential advantages over the currently discussed mobile Web Service client solutions:

- As a Mobile Host, the mobile terminal becomes a multi-user device where the owner/carrier of the device can work in parallel with users of the Web Service without explicit effort on his/her side.

- From a commercial viewpoint, there is a reversal of payment structures. While traditionally the information-providing Web Service client has to pay to upload his or her work results to a stationary server (where then other clients have to pay again to access the information), in the Mobile Host scheme responsibility for payment shifts to the actual clients -- the users of the information/services provided by the Mobile Host.

- Another commercial aspect is the possibility for small mobile operators to set up their own mobile Web Service business without resorting to stationary office structures, thus going one step further in the move from central to P2P architectures.

- From a management perspective, the Mobile Host supports more directly the concept of ad-hoc computer-supported cooperative work (CSCW), playing and learning in mobile settings, as studied in many current projects.

Of course, this additional flexibility generates a large number of interesting research questions which need further exploration.

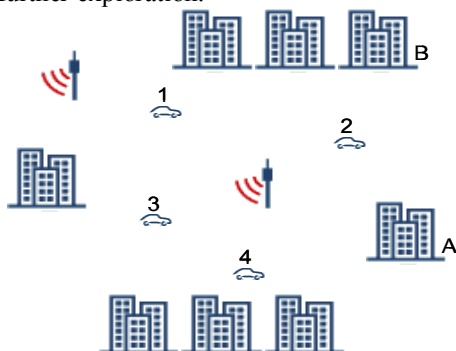


Figure 2. Guided parcel service scenario

As an example, the Mobile Host can be used for providing detailed location information of the mobile terminal, such as GPS [10] data and voluntary personal profile information describing the user of the device [15]. Such location based services can be used in applications like guided parcel service, where the client can exactly track his parcel throughout the delivery process. The cars in *figure 2*, used for the parcel delivery, are equipped with Mobile Hosts, which provide location details to the client. [6]

4. Mobile host design

The design of a Mobile Host faces some tricky challenges as very little research exists so far how to offer service provisioning on strictly resource-limited devices. After summarizing these challenges, we describe the solution concept developed for our Mobile Host system in general terms. The discussion has to get into some technical detail, but we believe that this kind of approach can also carry over to many other of the forthcoming Ambient Intelligence applications of mobile information systems and may therefore be worthwhile presenting here.

4.1. Challenges

When preparing this study, we were surprised to see that the state-of-the-art in mobile information provisioning (as opposed to mobile information requesting) seems to be still in its infancy. More recently Nokia has defined and is trying to address this issue with its Nokia Web Services Framework. [15]

Traditionally, mobile systems typically have been designed as client-server systems in which thin clients such as PDAs or data-capable cell-phones use wireless connections to gain access to resources (data and services) provided by central servers. Even though the emergence of wireless ad-hoc networks and powerful mobile devices makes it possible to design mobile systems using a peer-to-peer architecture, the concepts are still very fresh and require further research. The approach described below is intended as a contribution to address this need.

The key challenges addressed in our approach are threefold: to keep the Mobile Host fully compatible with the usual Web Service interfaces such that clients will not notice the difference; to design the Mobile host with a very small footprint that is available in the Smart Phone world (for example, the footprint of our fully functional prototype is only 130 KB); and to limit the performance overhead of the Web Service functionality such that neither the services themselves nor the normal functioning of the Smart Phone for the user is seriously impeded.

Considering these challenges in addition to the architectural and use case requirements identified in section 2 and section 3, the mobile Web Service provider for Smart Phones should provide the following features:

- A standard Web server handling HTTP requests,
- A basic Web Service provider handling requests from Web Service requestors using SOAP over HTTP,
- Capability to handle concurrent requests,
- Support for deployment of services at runtime,

- Support for the performance analysis.

Three additional challenges – the “grid-building” of multiple Mobile Hosts in a mobile P2P information system, the mobile management of Web Service evolution, and the security management – are subject of our current research but not yet within the scope of this paper.

4.2. Architecture

The Mobile Host has been developed as a Web Service handler built on top of a normal web server. The Web Service requests sent by HTTP tunneling are diverted and handled by the Web Service handler.

Figure 3 shows the core architecture of the Mobile Host. At the HTTP interface, the Mobile Host listens for incoming HTTP GET/POST requests on a sever socket. When the Mobile Host gets a request, the server socket accepts it, creates a socket for communication, and initiates a new thread of execution by creating an instance of the request handler. The request handler extracts the incoming message from the input stream of the socket, and checks for Web Service requests sent via HTTP tunneling. If it is normal HTTP request, the request handler processes the HTTP request just as the standard web server, and returns the response by writing to the output stream of the socket.

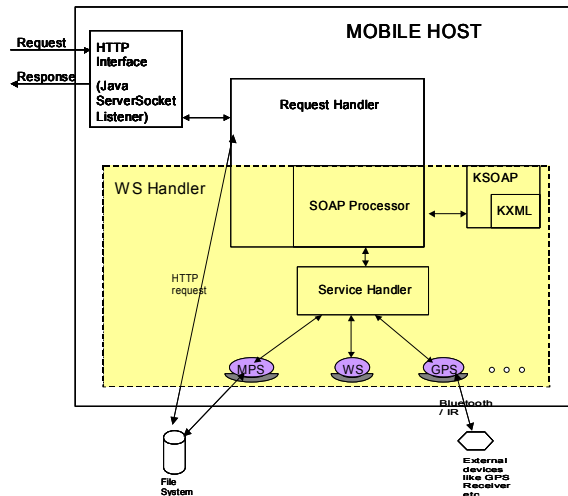


Figure 3. Core architecture of the Mobile Host

If the message comprises a Web Service request, the Web Service Handler component of the Mobile Host processes the message. The request handler reads the HTTP message body and de-serializes the SOAP request to Java objects, using the SOAP processor. The request handler passes these objects to the service handler, which extracts the service details and invokes the respective service. The business logic of the service

method is then executed and the service handler returns the response to the request handler.

The Web Services deployed on the Mobile Host can access the local file system, or any external devices like a GPS receiver, using Infrared, Bluetooth etc., and can implement business logic.

The request handler serializes the response and prepares the HTTP response message, which is returned to the client as a HTTP response by writing to the output stream of the socket.

Considering the low-resource constraints of Smart Phones, no deployment environment can be easily provided. Hence, all services have to be deployed at the installation of the Mobile Host. Alternatively, the Mobile Host was configured to look for services at other locations apart from the main JAR location, where the services could then be deployed at runtime.

4.3. Mobile terminal access

Once a Web Service is developed & deployed with the Web Service provider implemented on a mobile terminal (MT), the mobile terminal, that is registered and connected within the mobile operator network, requires some means of identification and addressing, that allows the Web Service to be accessible also from outside the mobile network operator’s network domain.

Generally, computers and devices in a TCP/IP network are identified using an IP address. The IP address, that is required for the data transfer to and from Smart Phones (as for any other IP communication client as Web servers, Intranet workstations, etc.), is assigned during the communication configuration phase. Typically, the IP address assigned to mobile devices using GPRS is only temporarily available, and is known only within the mobile operator’s network, which makes it difficult to use the IP address in the client applications.

The study has identified different means of resolving the IP address in HSCSD (High-Speed Circuit Switched Data) dial-up connection, GPRS environments and thereby making the data transmission with a mobile terminal, possible. A detailed discussion of these methods is beyond the scope of this paper. [6]

5. Implementation and evaluation of a Mobile Host prototype

5.1. Implementation details

To prove the feasibility of the Mobile Host in the scope for peer-to-peer, ad-hoc, distributed mobile information networks, a prototype mobile Web Service

provider was developed for the Java enabled Smart Phone, SonyEricsson P800.

The P800 Smart Phone supports two types of Java namely PersonalJava and J2ME CLDC/MIDP 1.0. After the analysis of both the development platforms, PersonalJava was selected for the implementation of the Mobile Host as, PersonalJava is faster in processing and has a richer application environment and can interact more extensively with the P800 phone software than J2ME. [4]

Using PersonalJava may have affected the portability of the Mobile Host on different Smart Phones, but this was not a big constraint, as the main interest was to prove the feasibility of mobile Web Service provider on some given Smart Phone.

Apart from this, the dedicated mobile Web Service provider required means for handling the SOAP requests from the WS clients. As the service provider was targeted at mobile terminals with limited resources, the parser should have a small memory footprint. With this intent, after an extensive search and analysis of different resources in the WWW, the kSOAP [3] toolkit from enhydra.org was chosen.

kSOAP is an open source API for SOAP parsing. It is based on kXML. kSOAP provides a SOAP parser with special type mapping and marshalling mechanisms. Both kSOAP and kXML are thin, easy to use, and well documented, and hence can be used for resource-constrained devices like mobile phones. Most recently kSOAP is redesigned to kSOAP 2 and the Mobile Host was upgraded to this new version, without much effort.

Alternative SOAP implementations like gSOAP, eSOAP were not considered for the implementation as they were developed in C++ and their compatibility with Symbian C++, supported by P800 could not be addressed easily.

The Mobile Host was developed as a combination of 5 core classes and 9 basic services, within a small memory foot print. It also has extensive support for performance analysis.

5.2. Evaluation

Once the Mobile Host was developed, it was extensively tested for performance issues like the memory load, server-processing load etc. The evaluation of the system was conducted using services like Mobile photo album service, which allows the Web Service client to browse the pictures taken by the mobile, the Location (GPS) data provisioning service and some more basic services like echo, 'ls' services and etc.

The test setup comprised a Mobile Host developed and deployed on the P800 Smart Phone and a standalone Apache Axis WS client. The client calls for different services (Within the context of this paper, it is assumed that the client knows the exact location (URI) of the service and the service description;) deployed on the Mobile Host and the performance of the Mobile Host was observed, by taking timestamps and memory foot prints, while the Mobile Host was processing the WS request. The tests were conducted both in HSCSD and GPRS environments.

As the test cases for the mobile photo album service, 15 different images were selected with memory sizes ranging from 3KB to 100KB, and the client can browse through these pictures. The location (GPS) data provisioning service uses an external GPS device for providing the GPS [10] data.

The evaluation showed that service delivery as well as service administration can be done with reasonable ergonomic quality by normal mobile phone users. As the most important result, it turns out that the total WS processing time at the Mobile Host is only a small fraction of the total request-response time (<10%) and rest all being transmission delay. *Figure 4* shows the time delays of different activities, for the Location data provisioning service. The results clearly showed that the processing capability of the Mobile Host can be increased by increasing transmission rates and the results are in harmony with the performance analysis of the mobile Web Service clients [11].

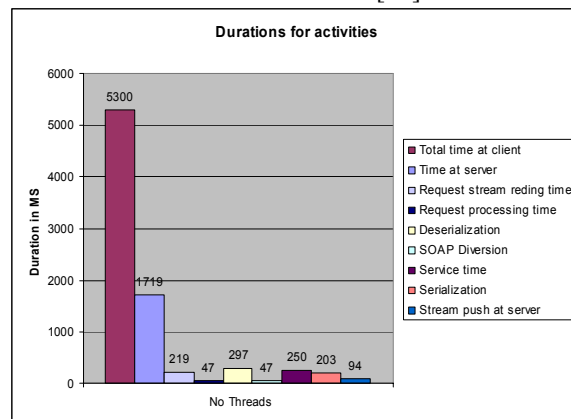


Figure 4. Time delays for the GPS provisioning service

In terms of performance, the key question was whether a reasonable number of clients could be supported by the Mobile Host with an overhead that would not prevent the main mobile user from using his or her Smart Phone in the normal fashion (either to supply the services or just for usual local phone functions). The results of this analysis are very encouraging and the Mobile Host was successful in handling 8 concurrent accesses for reasonable service

like location data provisioning service, with a response size of approximately 2KB.

The study of the memory footprints revealed that memory usage was not a problem with Mobile Host, as most of the time, the amount of free memory was at least 20% of the total memory allocated for the JVM (max value approximately 330 KB) and the “Out of Memory error” was never encountered during the execution of the tests. Approximately 200 data traces were observed as the experiments were repeated several times in order to have statistically valid results.

6. Conclusions and future work

This paper proposed to use modern low-resource mobile terminals such as Smart Phones not just as Web Service requestors, but also as Mobile Hosts that can themselves offer services in a true mobile peer-to-peer setting. A number of use cases, some of them experimented with in our prototype implementation, demonstrated the potential of this new approach. A prototype Web Service provider demonstrated the technical feasibility of our approach in terms of resource consumption, standard compliance, and performance. The approach truly paves scope for the P2P, ad-hoc and distributed mobile information networks.

Next steps on the technical side include a broader and more detailed, component-oriented performance analyses and means of improving the performance, a user-friendly approach to the handling of mobile Web Service evolution, and – perhaps most importantly – a detailed study of the security implications of this approach. Equally important is the study of specific application domains and usability analysis; our research mainly focuses on mobile community support and pervasive gaming.

Connected to this is the domain of mobile gaming which we are studying as part of the IPerG European Integrated Project. The mobile device turns from an interaction and information access device to a service providing device. I.e. elements of a game can be distributed over a network of Mobile Host users which makes them independent of a central game host, enabling ad hoc and on the fly reconfigurations of the game, based on the availability of players and mobile resources. While our research so far has focused on making such applications feasible from the viewpoint of mobile service provisioning, these applications will focus more on the interplay of multiple such servers and clients, complementing our earlier work on requirements analysis and simulation of such environments in the TROPOS project. [7, 8]

7. References

- [1] Web Services Activity, <http://www.w3.org/2002/ws/>, May 2004
- [2] HTTP, Hypertext Transfer Protocol version 1.1, IETF RFC 2616, <http://www.ietf.org/rfc/rfc2616.txt>
- [3] kSOAP, a open source SOAP implementation for kVM, <http://ksoap.enhydra.org/>, May 2004
- [4] “Java support in SonyEricsson mobile phones P800 and P802”, Jan. 2003 Developer guidelines from SonyEricsson Mobile CommunicationsAB, www.SonyEricssonMobile.com
- [5] “J2ME Web Services Specification”, JSR 172 from Java community process
- [6] S. Srirama, “Concept, implementation and performance analysis of mobile Web Service provisioning for Smart Phones”, MasterThesis, RWTH Aachen University, Jun. 2004
- [7] G. Gans, M. Jarke, G. Lakemeyer, D. Schmitz, “Deliberation in a modeling and simulation environment for inter-organizational networks”, In J. Eder / M. Missikoff (eds.): Advanced Information Systems Engineering. 15th Intl. Conf. CAiSE 2003, Velden/Austria 2003, Springer-Verlag, LNCS 2681, pp. 242-257
- [8] G. Gans, M. Jarke, S. Kethers, G. Lakemeyer, “Continuous requirements management for organisation networks: a (dis-)trust-based approach”, Requirements Engineering Journal, Vol. 8, No. 1, pp. 4-22, 2003
- [9] Mark Jones and Paul Krill, InfoWorld: JavaOne: JavaFirst brings Web Services to mobile devices, <http://www.javaworld.com/javaworld/jw-06-2003/jw-0612-idgns-mobile.html>
- [10] GPS, Global Positioning System, data format available at <http://www.navcen.uscg.gov/pubs/gps/sigspec/gpssps1.pdf>
- [11] M. Laukkanen, H. Helin, “Web Services in wireless networks: What happened to the performance”, in the proceedings of the 2003 Int. Conf. on Web Services – ICWS ’03, CSREA Press, pp. 278-284.
- [12] Apte, Deutsch, Jain, “Wireless SOAP: Optimizations for Mobile Wireless Web Services”, [www \(2005\), http://www.2005.org/cdrom/docs/p1178.pdf](http://www.2005.org/cdrom/docs/p1178.pdf)
- [13] R. van Engelen, K. Galliva, “The gSOAP toolkit for web services and peer-to-peer computing networks”, In 2nd IEEE International Symposium on Cluster Computing and the Grid, 2002.
- [14] P. Sandoz, S. Pericas-Geertsen, K. Kawaguchi, M. Hadley, and E. Pelegri-Llopert, “Fast Web Services”, Aug. 2003, <http://java.sun.com/developer/technicalArticles/WebServices/fastWS/>
- [15] Nokia Web Services Framework for Devices, http://www.nokia.com/NOKIA_COM_1/Operators/Technologies/Web_Services/web_services_a4_1603.pdf
- [16] Min Tian, Thiemo Voigt, Tomasz Naumowicz, Hartmut Ritter, and Jochen Schiller, “Performance Considerations for Mobile Web Services”. Elsevier Computer Communications Journal, Volume 27, Issue 11, 1 July 2004, Pages 1097-1105.