Mobile Enterprise – A Case Study of Enterprise Service Integration

Satish Narayana Srirama, Matthias Jarke
Information Systems and Databases Group
RWTH Aachen University
Ahornstr. 55, 52056 Aachen, Germany
{srirama, jarke}@dbis.rwth-aachen.de

Abstract

A Mobile Enterprise can be established in a cellular network by participating Mobile Hosts, which provide web services from smart phones, and their clients. Mobile Hosts enable seamless integration of user-specific services to the enterprise, by following the web service interfaces and standards, also on the radio link. Moreover, services provided by the Mobile Host can be integrated with larger enterprise services, bringing added value to the mobile services. However, establishing such a mobile enterprise poses several technical challenges, like the quality of service (QoS) and discovery aspects, for the network and as well as for mobile phone users. The paper summarizes the challenges and research in this domain, along with a case study where the results have been applied. The study and the results are of significant value in the enterprise service integration domain.

1. Introduction

Mobile data services in combination with profluent web services [7] are seemingly the path breaking domain in current information systems research. In mobile web services domain, the resource constrained smart phones are used as both web service clients and providers (Mobile Host). Mobile terminals accessing the web services cater for anytime and anywhere access to services. Some interesting mobile web service applications are the provisioning of services like e-mail, information search, language translation, company news etc. for employees who travel regularly. There are also many public web services accessible from smart phones like the weather forecast, stock quotes etc. Mobile web service clients are also significant in the geospatial and location based services [2]. While mobile web service clients are common, the research with providing web services from smart phones is still sparse. The scope of mobile web service provisioning was studied by two projects at RWTH Aachen University since 2003 [13, 6], where Mobile Hosts were developed, capable of providing basic web services from smart phones. Mobile web service clients and the Mobile Hosts in a cellular network, together form a Mobile Enterprise.

Mobile Hosts enable seamless integration of user-specific services to the enterprise, by following the standard web service interfaces and standards also on the radio link. Moreover services provided by the Mobile Host can be integrated with larger enterprise services bringing added value to these services. For example, services can be provided to the mobile user based on his up-to-date user context. The context details like the device capabilities, network capabilities, location details etc. can be obtained from the mobile at runtime and can be used in providing most relevant services like maps specific to devices and location information. Besides Mobile Hosts can collaborate among themselves in scenarios like Collaborative Journalism and Mobile Host Co-learn System and bring value to the enterprise. [12]

Once the Mobile Host was developed, an extensive performance analysis was conducted to prove its technical feasibility [13]. While service delivery and management from Mobile Host were thus shown technically feasible, the ability to provide proper quality of service (QoS), especially in terms of security and reasonable scalability, for the Mobile Host is observed to be very critical. Similarly, the huge number of web services possible, with each Mobile Host providing some services in the wireless network, makes the discovery of these services quite complex. Proper QoS and discovery mechanisms are required for successful adoption of mobile web services into commercial environments. Moreover, the QoS and discovery analysis of mobile web services has raised the necessity for intermediary nodes helping in the integration of Mobile Hosts with the enterprise. Based on these requirements a Mobile Web Services Mediation Framework (MWSMF) [15] is designed and established as an intermediary between the web service clients and the Mobile Hosts within the Mobile Enterprise, using the Enterprise Service Bus (ESB) technology. The paper summarizes the establishment of the Mobile Enterprise along with a scenario, where the concept was used. The sections are organized as follows:

Section 2 discusses the details and challenges associated with establishing a Mobile Enterprise. Section 3 discusses the Enterprise Service Integrations issues of the Mobile Host. Section 4 discusses the application scenario of the Mobile Enterprise where mobiles try to access LAS services and section 5 summarizes and concludes the paper.

2. Mobile Enterprise

A *Mobile Enterprise* can be established in a cellular network by participating Mobile Hosts and their clients, where the hosts provide user-specific services to the clients as per the WS* standards. However, such a Mobile Enterprise established poses many technical challenges, both to the service providers and the mobile operator. Some of the critical challenges and the associated research are addressed in this section. Figure 1 shows the Mobile Enterprise and hints the critical challenges posed to the mobile phone users and the operators.

2.1. Challenges for establishing Mobile Enterprise

As the Mobile Host provides services to the Internet, the devices should be safe from malicious attacks. For this, the Mobile Host has to provide only secure and reliable communication in the vulnerable and volatile mobile ad-hoc topologies. In terms of scalability, the Mobile Host has to process reasonable number of clients, over long durations, without failure and without seriously impeding normal functioning of the smart phone for the user.

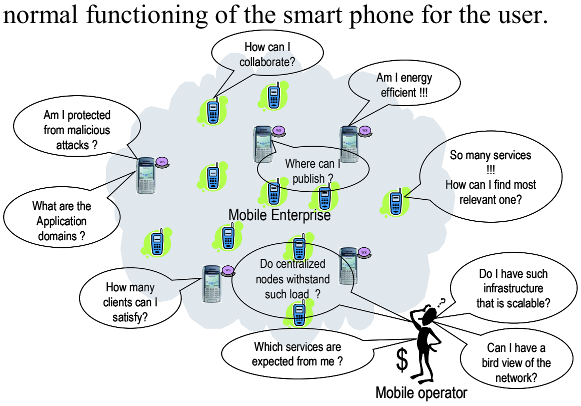

Figure 1. Mobile Enterprise and the critical challenges posed to the mobile phone users and the operator

Similarly, the huge number of web services possible, with each Mobile Host providing some services in the wireless network, makes the discovery of the most relevant services quite complex. Proper discovery mechanisms are required for successful adoption of Mobile Enterprise. The discovery, moreover, poses some critical questions like: where to publish the services provided by the Mobile Hosts? Should they be published with the centralized Universal Description, Discovery, and Integration (UDDI) registries available in the Internet or the operator is going to offer some help? This also raises questions like whether centralized nodes can withstand such high loads or some alternatives are to be looked at?

From the mobile operator's perspective the Mobile Enterprise poses questions like: what are the services expected by the mobile users from the operator? Can the operator monitor the communication and have a bird view of the complete network, so that business scenarios can be drawn out of it? Do the operators have such infrastructure that can scale to such requirements?

Our research in this domain focused at addressing most of these issues and the remaining parts of this paper summarizes the research and the results.

2.2. QoS aspects of the Mobile Host

Providing proper QoS, especially, appropriate security and reasonable scalability, for mobile web service provisioning domain was observed to be very critical. In terms of security, the Mobile Host has to provide secure and reliable communication in the vulnerable and volatile mobile ad-hoc topologies. In terms of scalability, the layered model of web service communication, introduces lot of message overhead to the exchanged verbose XML based SOAP messages. This consumes a lot of resources, since all this additional information is to be exchanged over the radio link. Thus for improving scalability the messages are to be compressed with out effecting the interoperability of the mobile web services. The message compression also improves the energy efficiency of the devices as there will be less data to transmit.

2.2.1. Security. The security analysis of the Mobile Host studied the adaptability of WS-Security specification to the mobile web service provisioning domain and concludes that not all of the specification can be applied to the Mobile Host, mainly because of resource limitations. The results of our analysis suggest that the mobile web service messages of reasonable size, approximately 2-5kb, can be secured with web service security standard specifications. The security delays caused are approximately 3-5 seconds. We could also conclude from the analysis that the best way of securing messages in a Mobile Enterprise is to use AES (Advanced Encryption Standard) symmetric encryption with 256 bit key, and to exchange the keys with RSA 1024 bit asymmetric key exchange mechanism and signing the messages with RSAwithSHA1. But there are still high performance penalties when the messages are both encrypted and signed. So we suggest encrypting only the parts of the message, which are critical in terms of security and signing the message. The signing on top of the encryption can completely be avoided in specific applications with lower security requirements. [14]

2.2.2. Transmission efficiency. In the scalability analysis of the Most Host, we have adapted BinXML [5] for compressing the mobile web service messages. BinXML is a light-weight XML compression mechanism and replaces each XML tag and attribute with a unique byte value and replaces each end tag with 0xFF. By using a state machine and 6 special byte values including 0xFF, any XML data with circa 245 tags can be represented in this format. The approach is specifically designed to target SOAP messages across radio links. So the mobile web service messages are exchanged in the BinXML format, and this has reduced the message of some of the services by 30%, drastically reducing the transmission delays of mobile web service invocation. The BinXML compression ratio is very significant where the SOAP message has repeated tags and deep structure. The binary encoding is also significant for the security analysis as there was a linear increase in the size of the message with the security incorporation. The variation in the WS-Security encrypted message size for a typical 5 Kb message is approximately 50%. [12]

2.3. Discovery aspects of the Mobile Enterprise

In a commercial mobile enterprise with Mobile Hosts, and with each Mobile Host providing some services for Internet, the number of services expected to be published could be quite high. Generally web services are published by advertising WSDL (Web Services Description Language) descriptions in a UDDI registry. But with huge number of services possible with Mobile Hosts, a centralized solution is not a best idea, as they can have bottlenecks and can make single points of failure. Besides, mobile networks are quite dynamic due to the node movement. Devices can join or leave network at any time and can switch from one operator to another operator. This makes the binding information in the WSDL documents, inappropriate. Hence the services are to be republished every time the Mobile Host changes the network.

Dynamic service discovery is one of the most extensively explored research topics in the recent times. Most of these service discovery protocols are based on the announce-listen model like in Jini. In this model periodic multicast mechanism is used for service announcement and discovery. But these mechanisms assume a service proxy object that acts as the registry and it is always available. For dynamic ad hoc networks, assuming the existence of devices that are stable and powerful enough to play the role of the central service registries is inappropriate. Hence services distributed in the ad-hoc networks must be discovered without a centralized registry and should be able to support spontaneous peer to peer (P2P) connectivity. [4] proposes a distributed peer to peer Web service registry solution based on lightweight Web service profiles. They have developed VISR (View based Integration of Web Service Registries) as a peer to peer architecture for distributed Web service registry. Similarly Konark service discovery protocol [9] was designed for discovery and delivery of device independent services in ad hoc networks.

Considering these developments and our need for distributed registry and dynamic discovery, we have studied alternative means of mobile web service discovery and realized a discovery mechanism in the P2P network. In this solution, the *virtual P2P network* also called the *mobile P2P network* is established in the mobile operator network with one of the nodes in operator proprietary network, acting as a JXTA super peer. JXTA (Juxtapose) is an open source P2P protocol specification. Once the virtual P2P network is established, the services deployed on Mobile Host in the JXME virtual P2P network are to be published as JXTA advertisements, so that they can be sensed as JXTA services among other peers. JXTA specifies '*Modules*' as a generic abstraction that allows peers to describe and instantiate any type of implementation of behavior representing any piece of "code" in the JXTA world. So the mobile web services are published as JXTA modules in the virtual P2P network. Once published to the mobile P2P network, the services can later be discovered by using the keyword based search provided by JXTA. The approach also considered categorizing the services and the advanced features like context aware service discovery. We address the discovery solution as *mobile P2P discovery* mechanism. The evaluation of the discovery approach suggested that the smart phones are successful in identifying the services in the P2P network, with reasonable performance penalties for the Mobile Host. [16]

3. Enterprise Service Integration

Mobile Hosts with proper QoS and discovery mechanisms, enable seamless integration of user-specific services to the Mobile Enterprise. Moreover services provided by the Mobile Host can be integrated with larger enterprise services bringing added value to these services. However, enterprise networks deploy disparate applications, platforms, and business processes that need to communicate or exchange data with each other or in this specific scenario addressed by the paper, with the Mobile Hosts. The applications, platforms and processes of enterprise networks generally have non-compatible data formats and non-compatible communications protocols. Besides, within the domain of our

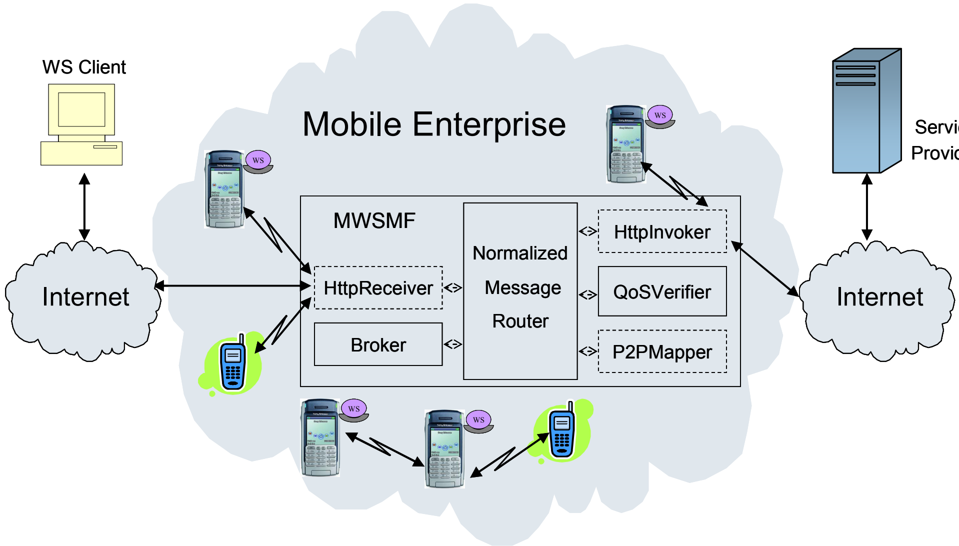

Figure 2. The Enterprise Service Integration scenario

research, the QoS and discovery study of the Mobile Host offered solutions in disparate technologies like JXTA. This leads to serious integration troubles within the networks. The integration problem extends further if two or more of such enterprise networks have to communicate among themselves. We generally address this research scope and domain, as the *Enterprise Service Integration*.

3.1. Integration options

As the integration troubles are well known and observed quite for sometime, the research in the domain has offered different solutions. [8]

Point-To-Point: One of the first methods used to integrate applications has been by using point-to-point integration solutions. Under this scheme a protocol or format transformer is built at one or either end between a pair of applications. One of the advantages of this scheme is that both applications have good knowledge about each other thus getting to a tight coupling. The principal disadvantage of this scheme is the difficulty to integrate a new application to the system due to the high number of protocols or format transformers which have to be implemented. This architecture is very popular solution for small scale integration problems.

Hub-And-Spoke: This architecture is also known as the message broker and it provides a centralized point where all applications connect to by using lightweight connectors. This centralized point is called hub or broker. The connectors act as adapters and translate data and messages between different applications to canonical formats. Two of the advantages of the hub-and-spoke architecture are that all the message transformation and routing is done by the hub and the number of connections for the integration is reduced with respect to the point-to-point architecture. Unfortunately like any centralized architecture, the centralized hub becomes a single point of failure.

Enterprise Message Bus: an enterprise message bus is an infrastructure of communication where the integration between applications can be done in a platform and language independent way. It is composed by a message router and publishing (subscription) channels where applications use request and response queues to interact with each other via a message. Consumers write request to the request queue and providers listen to the request queue waiting for request to their services. The result messages are then added to the response message queues.

Enterprise Service Bus: By using Enterprise Service Bus integration, applications do not communicate to each other directly, they communicate by using a SOA based middleware backbone. ESB basically consists of a set of service containers that are interconnected with a reliable messaging bus. The ESB supports multiple integration paradigms in order to fully support the variety of interaction patterns that are required in a comprehensive SOA between these service containers. So it has support for service-oriented architectures in which applications communicate through reusable services with well-defined and explicit interfaces, message-driven architectures in which applications send messages through the ESB to receiving applications and event-driven architectures in which applications generate and consume messages independently of one another.

3.2. Mobile Web Services Mediation Framework

The mobile web services mediation framework (MWSMF) is established as an intermediary between the web service clients and the Mobile Hosts in mobile enterprise. ESB is used as the background technology in realizing the mediation framework. Figure 2 shows the basic components of the mediation framework. For realizing the mediation framework we relied on ServiceMix, an open source implementation of ESB, based on the JBI specification. JBI architecture supports two types of components *Service Engines* and *Binding Components*. Service engines are components responsible for implementing business logic and they can be service providers/consumers. Service engine components support content-based routing, orchestration, rules, data transformations etc. Service engines communicate with the system by exchanging normalized messages across the *normalized message router* (NMR). The normalized messaging model is based on WSDL specification. The service engine components are shown as straight lined rectangles in the figure. Binding components are used to send and receive messages across specific protocols and transports. The binding components marshall and unmarshall messages to and from protocol-specific data formats to normalized messages. The binding components are shown as dashed rectangles in the figure.

The *HttpReceiver* component shown in figure receives the web service requests (SOAP over HTTP) over a specific port and forwards them to the *Broker* component via NMR. The main integration logic of the mediation framework is maintained at the Broker component. For example, in case of the scalability maintenance, the messages received by Broker are verified for mobile web service messages. If the messages are normal Http requests, they are handled by the *HttpInvoker* binding component. If they comprise mobile web service messages, the Broker component further ensures the QoS of the mobile web service messages and transforms them as and when necessary, using the *QoSVerifier* service engine component, and routes the messages, based on their content, to the respective Mobile Hosts. The framework also ensures that once the mobile P2P network is established, the web service clients can discover the services using mobile P2P discovery mechanism and can access the deployed services across MWSMF and JXTA network. [15]

Apart from security and improvements to scalability the QoS provisioning features of MWSMF also includes message persistence, guaranteed delivery, failure handling and transaction support. External web service clients, that does not participate in the mobile P2P network, can also directly access the services deployed on the Mobile Hosts via MWSMF, as long as the web services are published with any public UDDI registry or the registry deployed at the mediation framework and the Mobile Hosts are provided with public IPs. This approach evades the JXME network completely. Thus the mediation framework acts as an external gateway from Internet to the Mobile Hosts and mobile P2P network. The framework also provides a bird view of the mobile enterprise to the cellular operator, so that business scenarios can be drawn out of it.

4. Mobile access to LAS services – A use case of Mobile Enterprise

The main benefit with Mobile Enterprise is the achieved integration and interoperability for the mobile devices. It allows applications written in different languages and deployed on different platforms to communicate with Mobile Hosts over the cellular network. Moreover, the paradigm shift of smart phones from the role of service consumer to the service provider is a step towards practical realization of various computing paradigms such as pervasive computing, ubiquitous computing, ambient computing and context-aware computing. For example, the applications hosted on a mobile device provide information about the associated user (e.g. location, agenda) as well as the surrounding environment (e.g. signal strength, bandwidth). Mobile devices also support multiple integrated devices (e.g. camera) and auxiliary devices (e.g. Global Positioning Systems (GPS) receivers, printers). For the hosted services, they provide a gateway to make available their functionality to the outside world (e.g. providing paramedics assistance).

While the paper until here, have discussed the technical details involved with mobile web service provisioning, the following subsections try to explain, where the research can be applied, with one of the realized application scenario.

4.1. The Light Application Server (LAS)

LAS is a lightweight application server designed as a community middleware that is capable of managing users and multiple hierarchically structured communities along with their particular access rights as well as a set of services accessible to users. LAS mainly offers MPEG-7 multimedia services to the users. A community application can make use of the offered services by simply connecting to the server and then remotely invoking service methods. A very prominent feature of LAS is that the server functionality is easily extensible by implementing and plugging in new services and respective components. LAS also has security management support to guarantee access controls. [10, 11]

Even though, LAS is a reliable application server offering MPEG-7 multimedia services, it is not pure web service architecture; it was not designed under the SOA paradigm and important aspects like scalability and distributed services were not taken into account. QoS and performance problems have been observed recently by LAS users. For many years, LAS has been used on top of traditional networks infrastructures for providing the services required by social softwares such as Virtual Campfire [3]. Load balancing and cluster support were observed to be the immediate requirements for improving the performance of the LAS. Most recently the multimedia services are being offered to the mobile phone users and this requirement further enhances the problem.

4.2. Load balancing for LAS

The QoS of the LAS can be improved either by changing the architecture of the LAS to have the cluster support or to employ a middleware framework taking care of the load balance issues. We went with the second option and tried to adapt our knowledge from mobile web service provisioning domain to the LAS. Basically we developed the components that provide a web service interface to the LAS services. Later, we designed scenarios such that the requests are diverted to the less occupied server among a cluster of LASs.

The approach can be of three modes 1. Directly accessing the MPEG services through the Mobile Host. 2. Accessing the service through the mediation framework of the mobile enterprise 3. Indirect way of using the Mobile Host to connect to the mediation framework. All the three scenarios were tested for performance loads and can be perceived from figure 2.

4.2.1. Accessing LAS services through Mobile Host. Since LAS, as of now, supports the access of services only across the HTTP protocol, the invocation of these services from mobile phones becomes a little tricky. While studying mobile web service clients and providers, we have taken the design decision that the exposure and access of services from/to mobiles would be only through web services and WSDL. Mobile web services basically enable 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.

Based on our design decision, the Mobile Host provides a web service interface of the LAS services, for the external nodes. Under this architecture only the Mobile Host directly connects to the LAS and it is the only access point to the LAS services from mobiles. Theoretically, the Mobile Host can be replaced by a standalone node that provides web service interfaces for the LAS services. Since we are also interested in personalized services and user specific profiles, we were still proceeding with Mobile Host. Services provided by the Mobile Host are published using WSDL in order to enable the consumer to find those services and invoke them. We were also successful in publishing and discovering these services using the mobile web service discovery mechanism.

4.2.2. Mobile access through MWSMF. In this solution the services are directly accessed from the MWSMF. The middleware takes care of all the load balancing issues. The main advantage of the middleware solution is its transparency to users and programmers on the client side as well as on the server side. Inside LAS there are no necessary changes to do, in order to couple it with the MWSMF and users of LAS only need to connect to a single point, the MWSMF, in order to access any LAS server they are interested in. Imagine a set of LAS servers, which could be different instances of the same implementation or servers completely different from each other. Now, consider a group of Mobile Hosts or mobile web service clients trying to access many services from those LAS servers. The current way to achieve this goal would be configure each Mobile Host with the specific connection to the right LAS server based on the services offered by it.

But with the MWSMF solution, the middleware framework replaces all direct contact between Mobile Hosts and LAS servers, so that all communication takes place via the bus. The MWSMF receives all the requests from mobile clients and applies a Java based management process that selects the right LAS server to be invoked. Once the LAS server has been selected, the framework calls the appropriate service. As soon as the response from the LAS server is received by the MWSMF, it is forwarded to the mobile requester.

4.2.3. Accessing the middleware through the Mobile Host. In this solution the services are directly accessed from the Mobile Host. Here the Mobile Host contacts the middleware framework instead of contacting the LAS server directly. From the middleware the invocation scenario is as explained in section 4.2.2. The benefit with the scenario is that the mobile web service client can be unaware of the middleware framework. This is the most realistic scenario in terms of the applications, as it is the Mobile Host that is providing some personalized solution; The Mobile Host is integrating its services with some services from the LAS to polish the result, and the mobile requester is unaware of the complete backend process.

This is the most realistic scenario, but worst of the three solutions, in terms of the performance. One invocation cycle of this scenario has two invocation cycles across the radio link and one across the Internet. Thus the solution adds lot of network load.

Assuming that the next generation mobile communication technologies achieve higher data transmission rates and the operators provide flat rates this is the most feasible scenario for Mobile Enterprise.

5. Conclusions

The developments in the web services domain, the improved device capabilities of the smart phones and the improved transmission capabilities of the cellular networks have lead to the mobile web services domain. With this paper, we tried to summarize the challenges and research associated in this domain and especially with mobile web service provisioning and establishing the Mobile Enterprise. The QoS aspects of the developed Mobile Host, like providing proper security and scalability, are briefly addressed. However, the bulk of web services possible with Mobile Hosts, pose problems with the discovery of these services. The mobile P2P discovery mechanism proposes an alternative to this problem, with the help of P2P networks. Further, the QoS and discovery analyses of the Mobile Host have raised the necessity for a middleware framework and the features and realization details of the MWSMF are discussed.

The research has significant application scope, and the discussed Mobile access to the LAS services scenario, explains, where and how the study has been applied. The study in short concludes that the Mobile Enterprise can be established with reasonable performance latencies. Moreover, the services provided from the Mobile Hosts and the Mobile Enterprise offer significant help for integrating personalized mobile services to the regular enterprise and vice versa. Thus the study and the results are of significant value to the enterprise service integration domain.

6. Acknowledgement

This work is supported by the Ultra High-Speed Mobile Information and Communication (UMIC) research cluster (http://www.umic.rwth-aachen.de/) at RWTH Aachen University. The authors would also like to thank R. Levenshteyn and M. Gerdes of Ericsson Research for their help and support.

7. References

[1] Apache Software Foundation, "Apache ServiceMix", http://incubator.apache.org/servicemix/home.html [URL last visited on 30th April 2009].

[2] B. Benatallah and Z. Maamar, “Introduction to the special issue on m-services”, IEEE transactions on systems, man, and cybernetics - part a: systems and humans, 33(6):665–666, November 2003.

[3] Y. Cao, M. Spaniol, R. Klamma, D. Renzel, “Virtual Campfire - A Mobile Social Software for Cross-Media Communities”, K. Tochtermann, H. Maurer, F. Kappe, A. Scharl (Eds.): Proceedings of I-Media'07, International Conference on New Media Technology and Semantic Systems, Graz, Austria, September 5 - 7, 2007.

[4] S. Dustdar and M. Treiber, "Integration of transient Web services into a virtual peer to peer Web service registry", Distributed and Parallel Databases, 20: 91–115, 2006.

[5] M. Ericsson and R. Levenshteyn, "On optimization of XML-based messaging", In Second Nordic Conference on Web Services (NCWS 2003), pages 167–179, 2003.

[6] G. Gehlen, “Mobile Web Services - Concepts, Prototype, and Traffic Performance analysis”, PhD thesis, RWTH Aachen University, October 2007.

[7] K. Gottschalk, S. Graham, H. Kreger, and J. Snell, ”Introduction to web services architecture”, IBM Systems Journal: New Developments in Web Services and E-commerce, 41(2):178–198, 2002.

[8] M. Keen, S. Bishop, A. Hopkins, S. Milinski, C. Nott, R. Robinson, J. Adams, P. Verschueren, and A. Acharya, “Patterns: Implementing an SOA using an Enterprise Service Bus”, IBM RedBooks, July 2004.

[9] C. Lee, A. Helal, N. Desai, V. Verma and B. Arslan, "Konark: A System and Protocols for Device Independent, Peer-to-Peer Discovery and Delivery of Mobile Services", IEEE transactions on systems, man, and cybernetics - part a: systems and humans, 33(6):682–696, 2003.

[10] J. M. Martinez, C. Gonzalez, O. Fernandez, C. Garcia, and J. de Ramon, “Towards universal access to content using MPEG-7”, In Proceedings of the 10th ACM International Conference on Multimedia, ACM Press, 2002, pp. 199–202.

[11] M. Spaniol, R. Klamma, H. Janßen, D. Renzel, “LAS: A Lightweight Application Server for MPEG-7 Services in Community Engines”, K. Tochtermann, H. Maurer (Eds.): Proceedings of I-KNOW '06, 6th International Conference on Knowledge Management, Graz, Austria, September 6 - 8, 2006, J.UCS (Journal of Universal Computer Science) Proceedings, Springer, pp. 592-599.

[12] S. N. Srirama, “Mobile Hosts in enterprise service integration”, PhD thesis, RWTH Aachen University, October 2008.

[13] S. N. Srirama, M. Jarke, and W. Prinz, “Mobile web service provisioning”, In AICT-ICIW ’06: Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, page 120. IEEE Computer Society, 2006.

[14] S. N. Srirama, M. Jarke, and W. Prinz, "Security analysis of mobile web service provisioning", International Journal of Internet Technology and Secured Transactions (IJITST), 1(1/2):151–171, Inderscience publishers, 2007.

[15] S. N. Srirama, M. Jarke, and W. Prinz, "MWSMF: A Mediation Framework Realizing Scalable Mobile Web Service Provisioning", In International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications (Mobilware 2008), Innsbruck, Austria, ACM Int. Conf. Proceeding Series; Vol. 278, Article No. 43, 2008.

[16] S. N. Srirama, M. Jarke, and W. Prinz and H. Zhu, "Scalable Mobile Web Service Discovery in Peer to Peer Networks", In IEEE Third International Conference on Internet and Web Applications and Services (ICIW 2008), Pages 668-674, IEEE Computer Society, 2008.

[17] R. Ten-Hove and P. Walker, "Java TM Business Integration (JBI) 1.0 -JSR 208", Final Release, Technical report, Sun Microsystems Inc., 2005.