Mobile Web Services Mediation Framework

Satish Narayana Srirama [1],
Matthias Jarke [1,2]
[1] RWTH Aachen, Informatik V
Ahornstrasse 55,
52056 Aachen, Germany
{srirama, jarke}@cs.rwth-aachen.de

Wolfgang Prinz [1,2]
[2] Fraunhofer FIT
Schloss Birlinghoven
53754 Sankt Augustin, Germany
wolfgang.prinz@fit.fraunhofer.de

ABSTRACT

Mobile data services in combination with profluent Web services are seemingly the path breaking domain in current information systems research. In mobile Web services sphere, resource constrained mobile terminals are used as both Web services clients and providers. While service delivery and management from Mobile Host are technically feasible, the ability to provide proper quality of service (QoS) and discovery mechanisms for the huge number of services possible with Mobile Hosts is observed to be very critical. We have studied the security, scalability and discovery aspects of the mobile Web services and the analysis has identified the necessity of a mediation framework. This paper summarizes our QoS and discovery research and discusses the realization details and features of our enterprise service bus technology based integration framework for mobile Web service provisioning.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Architecture and Design, Distributed Systems

Keywords

Mobile Web services, QoS, Service discovery, Enterprise service bus

1. INTRODUCTION

Mobile data services in combination with profluent Web services [11] are seemingly the path breaking domain in current information systems research. Mobile Web services 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. To support the mobile Web services, there exist many organizations such as OMA [17], LA [26] on the specifications front; some practical data service applications such as OTA (over-the-air provisioning), application handover etc. on the commercial front; and SUN, IBM toolkits [24], [13] on the development front. Thus, though this is early stages, we can safely assume that mobile Web services are the road ahead.

Mobile Web services lead to manifold opportunities to mobile operators, wireless equipment vendors, third-party application developers, and end users. While mobile Web service clients, i.e. smart phones accessing Web services, are common these days [8, 3], we have studied the scope of *mobile Web service provisioning*, in one of our projects. In this project, we have developed a *Mobile Host*, capable of providing basic Web services from smart phones. Once the Mobile Host was developed, we have conducted extensive performance analysis to prove its technical feasibility. In the analysis, as the most important result, it turns out that the total Web service processing time at the Mobile Host is only a small fraction of the total request-response cycle time (<10%) and rest all being transmission delay. This makes the performance of the Mobile Host directly proportional to achievable higher data transmission rates [22].

While service delivery and management from Mobile Host are technically feasible, the ability to provide proper Quality of Service (QoS), especially in terms of security and scalability, from the Mobile Host is observed to be very critical. Moreover 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. Hence proper QoS and discovery mechanisms are required for successful adoption of mobile Web services into commercial environments. Based on our Mobile Host's application, QoS and discovery research, we have identified the need for a mediation framework or proxy for deploying Mobile Hosts in the cellular networks. The paper addresses our QoS and discovery research and discusses the features and realization details of the mobile Web services mediation framework. The rest of the paper is organized as follows.

Section 2 addresses the QoS realization issues of mobile Web services in terms of their security and scalability aspects. Section 3 summarizes the details of publishing and discovery of mobile Web services in P2P networks. Later section 4 discusses the need, features, realization and performance details of our mobile Web services mediation framework. Section 5 concludes the paper with future research directions.

2. QOS ASPECTS OF MOBILE WEB SERVICE PROVISIONING

Providing proper QoS, especially, appropriate security and

MW4SOC '07, November 26, 2007 Newport Beach, CA, USA

reasonable scalability, for mobile Web service provisioning 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.

2.1 Security analysis of Mobile Host

Once the Web services are deployed with the Mobile Host, the services are vulnerable to security breaches like denial-of-service attacks, man-in-the-middle attacks, intrusion and spoofing etc. For the traditional wired networks and Web services, a lot of standardized security specifications, protocols and implementations like WS-Security [2], SAML [5] etc. exist, but not much has been explored and standardized in wireless environments. In our security analysis of the Mobile Host [20], we have analyzed the adaptability of WS-Security to the mobile Web service provisioning domain. WS-Security provides ways to add security headers to SOAP envelopes, attach security tokens and credentials to a message, insert a timestamp, sign the messages, and encrypt the message. We mainly observed the additional latency caused to performance of the Mobile Host, with the introduction of security headers into the exchanged SOAP messages. The performance penalties of different encryption and signing algorithms were calculated, and the best possible scenario for securing mobile Web services communication was suggested.

From this analysis we could conclude that not all of the WS-Security specification can be applied to the Mobile Host, because of resource limitations. The results of our security 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 mobile Web service provisioning is to use AES 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 [18]. 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.

But a potential client of mobile Web services from the Internet can follow full WS-Security standard. This pushes the necessity for some mediation framework as the legitimate intermediary in the mobile Web service invocation cycle, transforming the messages to supported standard. For clarity, if we consider the encryption scenario, the messages sent by client can be encrypted with any other symmetric encryption algorithm like TRIPLEDES, AES-128, AES-192 [18] etc. Since the Mobile Host can not implement the complete WS-Security, the security of the message is to be verified at the intermediary and the message is to be encrypted again using AES-256 before sending the message across the radio link to the Mobile Host.

2.2 Scalability aspects of the Mobile Host

Web services communication is a layered communication and across different protocols. Considering *SOAP over HTTP* binding, at the lowest level is the transportation protocol, TCP. On top of TCP lies the HTTP communication. Then SOAP communication is over the HTTP protocol. The application communication and protocols for example WS-Security lie on top of SOAP. So any message exchanged over the Web service communication, consists some overhead across all the different layers. The size of the message

$$B_{msg} = B_{tp} + B_{mtp} + B_{soap} + B_{app}$$

Where B_{tp}, B_{mtp}, B_{soap}, B_{app} are the message overheads over transportation, message transportation, SOAP, application protocols respectively. Since we are considering message exchange over the radio link, B_{msg} has to be reduced to the minimum possible level [14]. For this the messages are to be compressed/encoded in the optimal way. The minimal encoding may not always be the best solution. First reason for this is that the encoding should be efficient, both in terms of message size reduced and extra performance penalties added to the devices. Secondly the encoding mechanism should not affect the interoperability. If an attempt is made to reduce the overload at B_{tp} or B_{mtp}, the interoperability of the Web services is seriously impeded. So the best position to target the encoding process is at the B_{soap} and upper levels. So the XML based SOAP messages are to be compressed.

The messages can be compressed with standard compression techniques like *Gzip* or XML-specific compression techniques like *XMill*. Recently there is an effort with the *Fast Web Services*, *Efficient XML*, *BinXML* [9] etc. to specify a binary format for XML data that is an efficient alternative to XML in resource constrained environments. Most recently there is also some effort with *BiM* (Binary Format for Metadata) [12] standard for the binary encoding of MPEG-7 Metadata. BiM is designed in a way that it allows fast parsing and filtering of the XML data at the binary level itself, without having to decompress again. [10] gives a comparison of different compression technologies for XML data and specifies the best scenario for the Web service message exchange across smart phones. The analysis suggests that BinXML is the best option (BiM was not considered in this analysis) to compress Web service messages considering compression ratio, processing time and resource usage.

Based on the analysis at [10] we have adapted BinXML for compressing the mobile Web service messages. BinXML replaces each 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% [20]. Similar to our security analysis, the scalability analysis also raised the necessity for an intermediary node, encoding/decoding the mobile Web service message to/from XML/BinXML formats in mobile operator proprietary networks.

3. MOBILE WEB SERVICES DISCOVERY

In a commercial wireless environment 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 descriptions in a UDDI registry [11]. 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 all these mechanisms assume a service proxy object that acts as the registry and 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 connectivity. [7] 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 [15] 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 JXTA/JXME network [23]. Project JXTA offers a language agnostic and platform neutral system for P2P computing. JXME is a J2ME based light version of JXTA for mobile devices [28]. By following the P2P architecture, mobile Web service discovery surely surpasses the problems of announce-listen mechanisms. The approach is conceptually similar to Konark but is truly scalable by following the open standards of Web service technologies and P2P protocols of JXTA. Our approach also significantly differs from context-aware VISR where all nodes are registries.

3.1 Publishing and discovery of mobile Web services in JXTA

A virtual P2P network is established in the mobile operator network, as shown in figure 1, with one of the node in the operator proprietary network, acting as a JXTA super peer. The super peer can exist at Base Transceiver Station (BTS) and can be connected to other base stations extending the JXTA network into the mobile operator network. Any Mobile Host or mobile Web service client in the wireless network can connect to the P2P network using the node at BTS as its rendezvous peer. Rendezvous peers cache and maintain an index of advertisements published by other peers. Within this mobile P2P network, the participating smart phones can be addressed with both peer ID and the mobile phone number, thus eliminating the need for a public IP. The necessity of public IP for each smart phone is observed to be major hindrance for the commercial success of Mobile Host, even though IPV6 promises such availability.

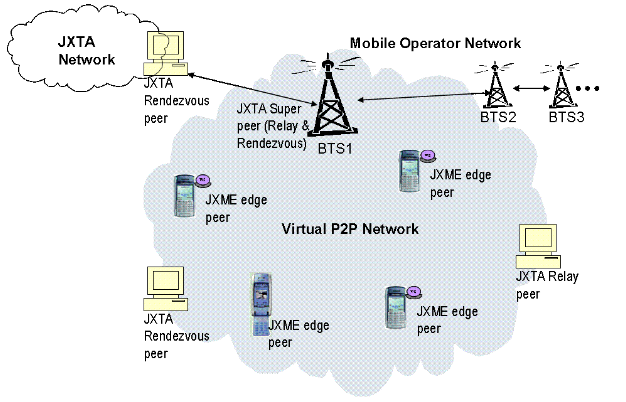

Figure 1: Virtual mobile P2P network with Mobile Hosts

The services deployed with Mobile Host in the mobile P2P network are published as JXTA *Modules*, so that they can be sensed as JXTA services among other peers. The module abstraction includes a *module class*, *module specification*, and *module implementation*. The WSDL descriptions of mobile Web services are incorporated into the module specification advertisements (MSA), and are published into the P2P network. The categorization of services is maintained with module class advertisements (MCA). We have maintained the categorization of mobile Web services according to the North American Industry Classification System's (NAICS) [27] service industries classification. The module advertisements are published into JXME network with an approximate life time that specifies the amount of time the Mobile Host wants to provide the service. The advertisements are cached at rendezvous peers or any other peers, with sufficient resource capabilities. Once the life time expires the module advertisements are automatically deleted from the P2P network, thus avoiding the stale advertisements. If the Mobile Host wants to extend the life time of the provided services, the advertisements can be republished.

Once published to the mobile P2P network, the services can later be discovered by using the keyword based search provided by JXTA. This basic search returns a large number of resulted services, returning every service that matches the keyword, even though the categorization of services already filters the list. Since the discovery client in mobile Web services scenario is a smart phone, the result set should be quite small so that the user can scroll through the list and can select the intended services. Hence the JXTA search resulted services are ordered according to their relevancy using *Apache Lucene* tool [6]. Lucene is an open source project that provides a Java based high-performance, full-featured text search engine library. Lucene allows adding indexing and searching capabilities to user applications. Lucene can index and make searchable any data that can be converted to a textual format. Using the tool and its index mechanism the search results were ordered/filtered and the advanced matched services were returned to the discovery client.

Similar to QoS analysis, the mobile Web service discovery also raised the necessity for intermediary nodes, acting as super peers and helping in the JXME publishing and discovery processes by hosting services like lucene based filtering mechanisms. Moreover mobile Web service clients generally prefer using services of the Mobile Host based on several context parameters such as location, time, device capabilities, profiles, and load on the Mobile Host etc. Most of these details can not be provided just based on keywords. Semantic matching of services gives the most appropriate and relevant results for mobile Web service discovery. Currently we are trying to describe the service context and device profiles using ontology-based mechanisms. For describing the semantics of services, we are trying to use Web Ontology Language for Services (OWL-S) [16]. OWL-S is an ongoing effort to enable automatic discovery, invocation, and composition of Web services. The semantic discovery process is heavy, in terms of resource consumption and need to be performed at a standalone intermediary or distributed middleware framework.

4. INTEGRATION FRAMEWORK FOR MOBILE WEB SERVICE PROVISIONING

The discussion of QoS and discovery analysis of mobile Web services, discussed in previous sections, has raised the necessity for intermediary nodes helping in the integration of mobile Web service provisioning deployment scenario. *Enterprise Service Bus* (ESB) technology is the latest development in SOA world. Gartner et al. defines enterprise service bus as a new architecture that exploits Web services, messaging middleware, intelligent routing, and transformation [19]. It basically consists of a set of service containers that are interconnected with a reliable messaging bus. Service containers adapt IT assets to a standard services model, based on XML message exchange using standardized message exchange patterns. The ESB provides services for transforming and routing messages, as well as the ability to centrally administer the distributed system. ESBs mainly help in achieving enterprise requirements of higher QoS.

Considering these developments, the mobile Web services mediation framework (MWSMF) is established as an ESB acting as intermediary between the Web service clients in the Internet and the Mobile Hosts in the virtual P2P network. The planned deployment scenario of the mediation framework is addressed in [21]. Web service clients can discover the services using P2P discovery mechanism and can access the deployed services across MWSMF and JXTA network. The mediation framework ensures the QoS of the mobile Web service messages and transforms them as and when necessary and routes the messages based on content. Apart from security and scalability, the QoS provisioning features of MWSMF also include message persistence, guaranteed delivery, failure handling and transaction support. External Web service clients that do not participate in JXTA can directly access the mobile Web services deployed on the Mobile Hosts via MWSMF, as long as the Web services are published with the UDDI registry at the mediation framework and the Mobile Hosts are provided with public IPs. The latter approach evades the JXME network completely. Thus the mediation framework acts as external gateway from Internet to the mobile P2P network.

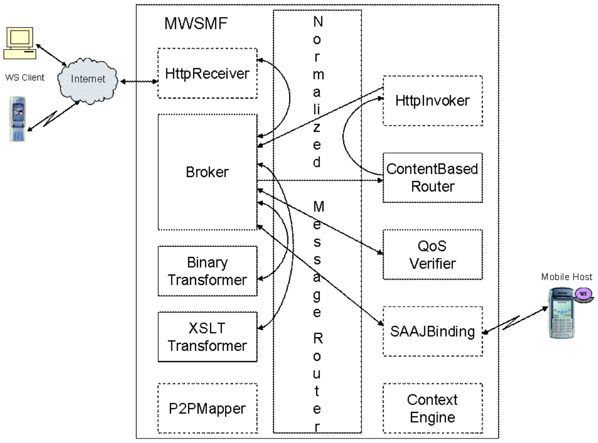

Figure 2: Basic components of the MWSMF

4.1 Realization of the MWSMF

For establishing the mediation framework many of the current ESB products were studied. Many products have already hit the market like *Sonic SOA Suite* and *FioranoESB Suite*, which are mainly based on proprietary message middleware. Then there are products like *Cape Clear* and *Cordys* that use truly open and distributed SOA. Alternatively there are products like *Iona Artix* which are extensions of the traditional EAI architectures [4]. SUN has defined JSR 208, *Java Business Integration specification* (JBI) [25], a Java-based standard addressing the EAI and B2B issues based on SOA paradigms and principles. MWSMF is realized based on open source *ServiceMix* ESB, which is based on JBI [1].

Figure 2 shows the components of the mediation framework. ServiceMix by following the 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. Binding components and service engines interact with the NMR via a delivery channel that provides a bidirectional delivery mechanism for message packets called *messages exchanges* (ME). JBI system supports four message exchange patterns *In-Only*, *Robust-In-Only*, *In-Out* and *In-Optional-Out*.

The *HttpReceiver* component shown in figure 2 receives the Web service requests (SOAP over HTTP) over a specific port and forwards them to the *Broker* component via

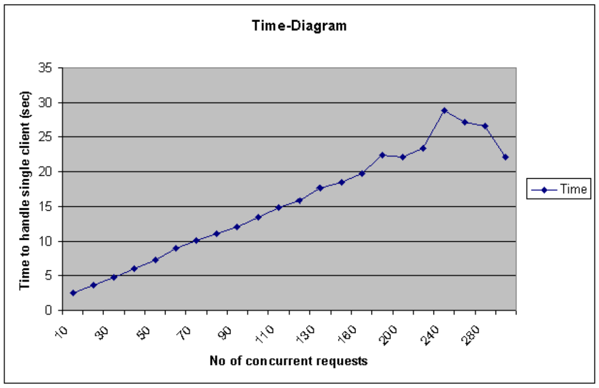

Figure 3: Times to handle a client at the MWSMF, under different concurrency levels

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 the messages comprise Web service messages, the messages are transformed to BinXML format at the *BinaryTransformer* component. The request messages are then sent to the Mobile Host via *SAAJBinding* component. SAAJBinding component invokes Web services according to SOAP with Attachments specification. The BinXML adapter at the Mobile Host first decodes the binary messages back to the XML format and then the Web service requests are processed. The response messages follow the same track along the NMR and are returned to the mobile Web service client via the HttpReceiver component. The scenario is evaluated for verifying the stability of the MWSMF, on a laptop. The laptop has an Intel Pentium M Processor 2.00GHz / 1GB RAM. Figure 3 shows the times taken for handling a client under multiple concurrent requests. Each concurrent client in turn generated 5 successive requests. The steady increase in average durations to handle a client is quite normal and the mean duration of handling a single request, across all concurrent successful requests, at the MWSMF is about 150 milliseconds. The mediation framework was successful in handling up to 110 concurrent requests with out any connection refusals. The sharp decline in times after 240 concurrent requests is because of a large number of failed requests at this high concurrency level. The analysis shows that the mediation framework itself is scalable and improves the scalability of the Mobile Host by BinXML encoding the messages.

The *QoSVerifier* and the *XSLTTransformer* components of the MWSMF come in handy in ensuring security of the mobile Web service messages. All the components are dynamically deployed into the mediation framework by following the Spring framework. Currently we are working with the *P2PMapper* binding component that transforms the JXTA messages to/from normalized messages and thus making the invocation of the mobile Web services feasible across the JXME network. Presently the mobile Web service invocations are possible only through the SAAJBinding component. We are also trying to realize a binding component for the context engine that helps in context aware service discovery of the mobile Web services. Figure 2 shows only those components that are most relevant for the discussion in this paper. Apart from these main components, the mediation framework also supports supplementary components that help in the successful deployment of Mobile Hosts in the operator proprietary mobile network. Some of these features are discussed in the following subsection.

4.2 Supplementary features of the MWSMF

Apart from acting as the integration framework for mobile Web service provisioning, the MWSMF also provides some critical services required in the QoS and discovery maintenance of mobile Web services. Some of the Web services deployed with the MWSMF include Identity related services that store and provide the asymmetric keys used in security analysis of the messages exchanged by participating Mobile Hosts and mobile Web service clients. The services also maintain and provide authentication and authorization details of the participants, thus help in realizing the end point security. Similarly context aware services provide the context information of Mobile Hosts and deployed services and profiles of the mobile Web service clients. The services thus help in realizing context aware service discovery.

As discussed earlier, the mediation framework also hosts a UDDI registry. The mobile Web services can therefore be published with the UDDI registry at the MWSMF. Any external client that does not participate in JXTA can search the registry and can access the Web services directly. Thus the mediation framework supports the access of mobile Web services both across P2P and standard Web service protocols. For the latter approach to be valid, the smart phones should still posses the public IP feature.

MWSMF also supports automatic startup of the Mobile Hosts. Generally hand-held devices have many resource limitations like low computational capacities, limited storage capacities, limited battery power etc. So to conserve these resources, the Mobile Host features of the smart phones are to be turned-on only when the provider is prepared to deliver and receives a request from the mobile Web service client. The MWSMF identifies the contact details of the phone, when the request is for particular Mobile Host, and sends a Short Message Service (SMS) to the device. A generic program is run on the smart phone that starts the Mobile Host automatically and activates its services and features, when the message is received. The SMS messages follow specific application protocol. Currently, our protocol has support for only the basic features of the Mobile Host like starting the server and authenticating the client. The person with the Mobile Host can also opt to turn down this request from client.

5. CONCLUSIONS AND FUTURE WORK

This paper first addressed QoS aspects of mobile Web services with respect to mobile Web service provisioning from smart phones. The main topics of interest in this discussion are the mobile Web service's security and scalability issues. The paper later addressed the discovery issues and proposed our mobile Web service publishing and discovery approach in P2P networks. From these observations the paper identifies the necessity of a proxy/middleware for successful deployment of mobile Web service provisioning from smart phones in cellular networks. Later the features, realization details and scalability analysis of such a mobile Web services mediation framework are discussed.

The MWSMF provides several directions for future research. Currently we are working with many auxiliary features for the mediation framework like the identity related services and context aware services already addressed in the paper. In terms of developing components for the framework, we are currently working with P2Pmapper and context engine components discussed in section 4.1. The components are crucial in achieving a context aware service discovery of mobile Web services across P2P networks. The security domain also can be extended with proper end-point security and access control mechanisms still missing for mobile Web services. We are also attentive of all the performance penalties added to the smart phone, with these additional features. Another detailed performance analysis of the Mobile Host is required, once the complete deployment scenario is ready for evaluation, checking that these added features won't restrict the main mobile user from using his smart phone in general fashion like making phone calls.

6. ACKNOWLEDGMENTS

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, 2007.

[2] B. Atkinson, G. Della-Libera, S. Hada, M. Hondo, P. Hallam-Baker, and etc. Web Services Security (WS-Security) - Version 1.0. Technical report, IBM Corporation, Microsoft Corporation, VeriSign, Inc., April 2002.

[3] 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.

[4] J. R. Borck. Enterprise service buses hit the road. *Infoworld*, pages 26–40, July 2005.

[5] S. Cantor, J. Kemp, R. Philpott, and E. Maler. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. Technical report, OASIS Standard, March 2005.

[6] D. Cutting. Apache Lucene, 2007.

[7] 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.

[8] J. Ellis and M. Young. J2ME Web Services 1.0 - Final Draft (JSR 172). Technical Report Revision 11, Sun Microsystems, Inc., October 2003.

[9] M. Ericsson. A study of compression of XML-based messaging. Technical report, Växjö University, 2003.

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

[11] 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.

[12] J. Heuer, C. Thienot, and M. Wollborn. *Introduction to MPEG-7: Multimedia Content Description Interface*, chapter Binary Format, pages 61–80. Jon Wiley and Son, 2002.

[13] IBM Corporation. IBM WebSphere Studio Device Developer. IBM developerWorks, 2007.

[14] M. Laukkanen and H. Helin. Web services in wireless networks: What happened to the performance. In *proceedings of the Int. Conf. on Web Services (ICWS '03)*, pages 278–284. CSREA Press., 2003.

[15] 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, November 2003.

[16] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, and etc. OWL-S: Semantic Markup for Web Services. Technical report, W3C Member Submission, November 2004.

[17] OMA. Mobile Web Services Requirements - Version 1.1. Technical report, Open Mobile Alliance Group, March 2006.

[18] RSA Labs. Techniques in cryptography, http://www.rsa.com/rsalabs/node.asp?id=2212, 2007.

[19] R. W. Schulte. The Enterprise Service Bus: Communication Backbone for SOA. Gartner Inc., 2007.

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

[21] S. N. Srirama, M. Jarke, and W. Prinz. A mediation framework for mobile web service provisioning. In *Middleware for Web Services (MWS 2006) Workshop @ 10th IEEE EDOC Conference*, page 14, 2006.

[22] 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.

[23] S. N. Srirama, M. Jarke, and W. Prinz. Mobile web service discovery in peer to peer networks. In *5th International Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS 2007), a CAiSE'07 workshop*, pages 531–542, 2007.

[24] Sun Microsystems. Sun Java Wireless Toolkit for CLDC. Sun Developer Network, 2007.

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

[26] J. Tourzan and Y. Koga. Liberty ID-WSF Web Services Framework Overview - Version: 2.0. Technical report, Liberty Alliance Project, 2006.

[27] U.S. Census Bureau. North American Industry Classification System (NAICS), 2007.

[28] B. J. Wilson. *JXTA*. New Riders Publishing, June 2002.