Semantic Authorization of Mobile Web Services

Anton Naumenko[1], Satish Srirama[2], Vagan Terziyan[1] and Matthias Jarke[2]

[1] University of Jyväskylä, Department of Mathematical Information Technology, annaumen@cc.jyu.fi, vagan@it.jyu.fi
[2] RWTH Aachen University, Information Systems, {srirama, jarke}@cs.rwth-aachen.de

Abstract

With the recent developments in the cellular world, the high-end mobile phones and PDAs are becoming pervasive and are being used in different application domains. Integration of the web services and cellular domains lead to the new application domain, mobile web services. Mobile web service provisioning offers many of its applications in domains like e-commerce, collaborative applications, social systems, mobile community support etc. This paper introduces the concept of mobile web services, inspects possible technical usage scenarios, and elaborates on commercial applications and usage scenarios based on previously conducted case studies. We have designed prototypes and conducted experiments towards quantitative feasibility study for this emerging research area of mobile web services, especially focusing on the secure communication and access control. In order to provide proper qualitative justification of security measures, we evaluated security threats in mobile environments, reviewed conventional security requirements for web services, analyzed security- sensitive characteristics of mobile web services, and finally defined critical success factors for controlling access to mobile web services. We proposed to utilize distributed architectures of semantics-based authorization mechanism as an adequate access control solution for mobile web service provisioning.

Key words: Access Control, Mobile Web Service, Ontology, Secure Communication

1 Introduction

Traditionally, the hand-held devices have many resource limitations like low computational capabilities, limited storage capacities, and small display screens with poor rendering quality. With the recent developments in the cellular world, the high-end mobile phones and PDAs are becoming pervasive and are being used in different application domains like location based services, mobile banking services, cooperative systems etc. The higher data transmission rates achieved with third and fourth generation mobile communication technologies also boosted this fast growth in the wireless market. The market capture of such smart phones is quite evident and in fact in 2003 itself 12.1 million PDA-sized devices were sold, including all PDA-phones and smart phones [19], [1]. The number of java enabled mobile phones sold, in the same time, has outnumbered the number of PCs sold [52]. The situation brings out a large scope and demand for software applications for such high-end mobile devices.

Paralelly the web services and their specifications are getting standardized and are quickly being adapted in different application domains. The main purpose of web services would be the integration of software applications across heterogeneous protocols to deliver sophisticated added-value services [60]. The biggest advantage of web services lies in their simplicity in expression, componentized development, communication and servicing. Integration of the web services and cellular domains lead to the mobile web services. While accessing the web services from mobile phones is common these days [30], [8], the scope of providing web services from smart phones was studied in the mobile web service provisioning project [57]. Mobile web service provisioning offers many of its applications in domains like e- commerce, collaborative learning, social systems, mobile community support etc. While the applications with mobile web services are quite welcoming, the ability to provide secure and reliable communication in the vulnerable and volatile mobile ad-hoc topologies is becoming obligatory. Mobile web services are easily readable across the network, as there might be many legitimate intermediaries in the web service communication. This easy readable nature of web services further enhances the complexity of security realization in wireless environments. Secure provisioning of mobile web services needs proper identification mechanism, access control, data integrity and confidentiality.

In our current research, we are trying to provide proper security for the realized mobile web service provider ("Mobile Host"). Different message-level and end-point security strategies are studied and analyzed to secure the mobile web service communication. The security analysis suggests that proper message-level security can be provided in mobile web services domain with reasonable performance penalties on the Mobile Host [58]. While the basic message-level security can be provided, the end-point security comprising proper identity and access control mechanisms, still poses a great challenge for the Mobile Host.

Amongst different areas of security, the access control directly impacts confidentiality, availability, and integrity of data and applications. The authorization faces the problem to describe (policy) rights and enforce (mechanism) a decision of access control. An access control policy, also referred as security policy, means laws, rules, principles, conditions, regulations and practices of managing, protecting, and sharing of computing and information assets. A policy can be application or platform specific or it can span boundaries of applications. An access control mechanism enforces an access control policy. It is desirable to use one common access control mechanism for a wide range of policies and to enforce one policy in wide variety of environments using native access control mechanisms. An access control model is a mathematically precise statement of a security policy that represents the state of a security system and transitions from one state to another state. An access control model mediates between the security policy and the access control mechanism. An access control to mobile web services is a challenging task due to the mobility of both clients and services; the demands of ubiquity and pervasiveness of all applications including security mechanisms; and the technical limitations of capabilities of communicational and computational components of mobile and wireless technologies. We propose to utilize distributed architectures of semantics-based authorization mechanism to ensure extensibility, flexibility, usability, applicability, etc of access control solutions.

In our research we have employed different research methods. Initially, we conducted case studies [63] and conceptual-analytical research [28] in order to collect technical and business usage scenarios, to evaluate security threats in mobile environments, to review conventional security requirements for web services, to analyze security-sensitive characteristics of mobile web services, and finally to define critical success factors for controlling access to mobile web services. These were used for proper qualitative (analytical) justification and evaluation of research ideas. Then we developed prototypes using system development [45] in order to make our research ideas tangible. Using the prototypes, we quantitatively justified and evaluated our initial ideas and proposals based on experiments.

The remainder of the paper is organized as follows. The next section provides overview of the mobile web service provisioning and of real-world usage scenarios. The section 3 presents analysis of security-relevant issues of mobile web service provisioning. The section 4 briefly introduces Semantics-Based Access Control (SBAC), proposes an access control model and policy language, and presents analysis of possible architectures for the policy enforcement function. The section 5 concludes this paper with summary and future research directions.

2 Mobile Web Services

Service-Oriented Architecture (SOA) describes a new component model which relates distributed components, called services, to each other by means of formally defined interfaces [17]. Thus SOA provides loose coupling of cleanly encapsulate services. Usually, SOA is implemented by means of web services which enable application-to-application communication over the Internet. Component-orientation is not new and a SOA can also be implemented with technologies like Common Object Request Broker Architecture (CORBA) and CORBA IDL (Interface Definition Language). But using web services for SOA provides certain advantages over other technologies. Web services are based on a set of still evolving, though well-defined W3C standards that allow much more than defining interfaces. Web services are self-contained, modular applications with their public interfaces defined and described using Web Services Description Language (WSDL). Web services provide access to software components through standard Web technologies and protocols like SOAP and HTTP. A service provider develops and deploys the service and publishes its description and access details (WSDL) with the UDDI registry. Any potential client, who queries the UDDI, gets the service description and accesses the service using SOAP [18]. The quest for enabling these open XML web service interfaces and standardized protocols also on the radio link, with the latest developments in cellular domain, lead to a new domain of applications mobile web services. The developments in cellular world are two folded; firstly there is a significant improvement in device capabilities and secondly with the latest developments in mobile communication technologies with 3G and 4G technologies higher data transmission rates in the order of few mbs were achieved [19], [1].

2.1 Mobile Web Service Provisioning (Mobile Host)

In the mobile web services domain, the resource constrained mobile devices are used as both web service clients and providers, still preserving the basic web services architecture in the wireless environments. While mobile web service clients are quite common, the research with providing web services from smart phones is still sparse. In mobile web service provisioning project one such Mobile Host was developed proving the feasibility of concept [57]. Mobile Host is a light weight web service provider built for resource constrained devices like cellular phones. It 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 component. The Mobile Host was developed in PersonalJava on a SonyEricsson P800 smart phone [27]. The footprint of the fully functional prototype is only 130 KB. Open source kSOAP2 [37] was used for creating and handling the SOAP messages. 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 un-publishing the services, as the basic web services architectures is still preserved in mobile web services [57]. The detailed evaluation of this Mobile Host clearly showed that service delivery as well as service administration can be done with reasonable ergonomic quality by normal mobile phone users [56]. 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. This makes the performance of the Mobile Host directly proportional to achievable higher data transmission rates.

2.2 Technical Usage Scenarios

Once a web service is developed and deployed with the Mobile Host, 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 Internet. 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 connection 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. Very few operators in the market today provide the facility that provides the smart phone with the public IP in GPRS network. The operational setup for accessing the mobile terminal in a GPRS network is shown in figure 1 with the interaction numbered 1. The mobile TCP/IP connection between the web service client and the Mobile Host is deployed on top of a GPRS link into the mobile operator network. From there the traffic is routed through the Internet to/from the web service client. The problem of addressing each mobile node with IP is not a big issue and it could be solved with Mobile IP version 6 (Mobile IPv6) [34].

The mobile web service provisioning project also has identified other means of addressing the Mobile Host in HSCSD (High-Speed Circuit Switched Data) [30] and P2P (Peer-to-Peer) environments [24]. In the HSCSD addressing scenario, a HSCSD connection is established between the smart phone and the prototyping network, which is connected to the Internet. HSCSD is an enhancement of CSD (Circuit Switched Data) data services of current GSM networks. HSCSD allows the access of non-voice services with a data rate about 3 times higher than that of CSD. Higher rates are achieved by using multiple channels for the data transmission. With this technology subscribers can send and receive data from their portable computers or mobile devices at a speed of up to 28.8 kbps. The HSCSD connection uses a Public Land Mobile Network (PLMN) and the Public Switch Telephone Network

(PSTN / ISDN) for making the data call to the server. The connection is setup by using PPP (Point-to-Point Protocol) over a circuit-switched data call to a modem that is connected to one of the servers in the network. On top of this PPP link a TCP/IP end-to-end connection between the mobile terminal and the dial-in server is established. Hence, as long as the data call persists, the mobile terminal can be addressed using the IP address assigned to it by the dial-in server. Thus the Web Service deployed on the mobile terminal can be accessed from any client within the network environment. [56] The interaction is shown in figure 1 with number 2.

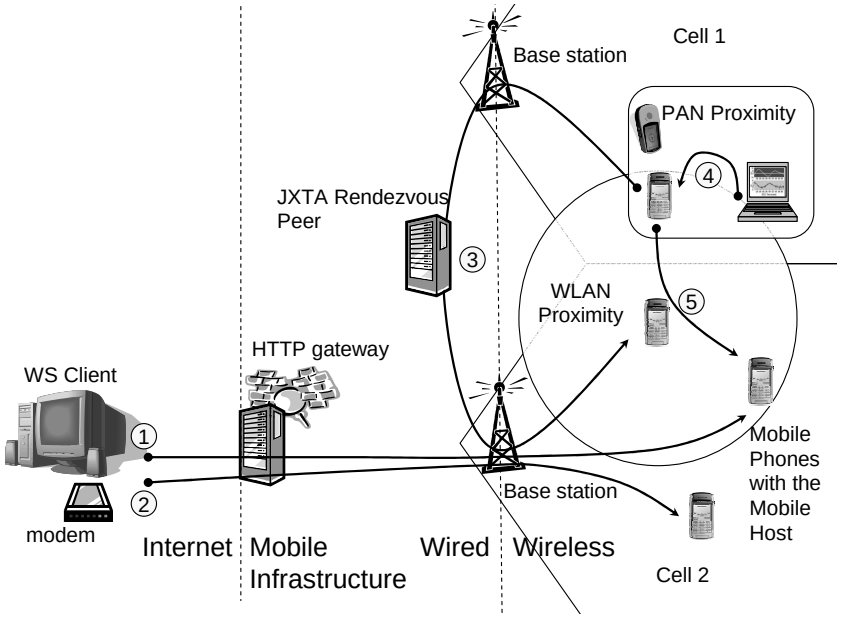

Figure 1: Mobile web service provisioning and interactions

The need for public IP for each of the participating Mobile Hosts was observed to be the major hindrance for commercial success of the Mobile Host. So, alternative architectures were studied for the addressing of mobile web services. In a JXTA network [35], each peer is uniquely identified by a static peer ID, which allows the peer to be addressed independent of its physical address like DHCP based IP address. This peer ID will stay forever with the device even though the device supports multiple network interfaces like Ethernet, WiFi or Bluetooth for connecting to the P2P network. Hence, the scope of the Mobile Host in the P2P networks was studied, so as to address the Mobile Host with peer ID. A virtual P2P network can be established by connecting the Mobile Hosts to JXTA superpeers deployed at the base stations. The base stations in turn connected to each other and thus extended the P2P network into the operator proprietary network. The mobile web service clients and the providers connect to the JXTA network, using the proxied JXME (JXTA for J2ME) version. Now by using the peerID, Mobile Host does not have to worry about changing IPs, operator networks, and is always visible to the web service client. Mapping the peer ID to the IP is taken care by the JXTA network, thus eliminating the need for public IP. The JXTA based P2P network also helps in better discovery of huge number of web services possible with the Mobile Hosts, by acting as a dynamic cache of advertisements of the mobile web services [55]. This Mobile P2P interaction in figure 1 is numbered 3.

Provisioning of mobile web services in totally decentralized manner is even more challenging. This kind of interaction between peers is also referred as pure P2P. Pure P2P is a setup like the classic Gnutella file sharing network [23]. The interactions 4 and 5 in figure 1 represent this pure P2P network idea. In our case of mobile web services, this means that discovery, invocation and integration of web services directly occur between mobile devices without any centralized entities like base stations. We have not studied how to provide mobile web services according to this kind of technical usage scenario, but it promises to have the best cost-effectiveness as long as interactions between clients and providers of mobile web services do not involve proprietary mobile networks. Bluetooth could be a possible technical solution for establishing such a pure P2P network. This kind of interactions tends to enable personal computing using various devices in Personal Area Network partially or fully based on mobile web services.

2.3 Sample Mobile Web Services: Building Bricks

Mobile photo album service: Today's high-end mobile terminals become more and more advanced, and are generally being equipped with an integrated digital camera. The photographs taken with these smart phones can later be uploaded or transferred to personal computers through cables or by using different wireless methods like Infrared or Bluetooth. Using currently available technologies, if a smart phone user wants to publish the photographs she had taken to the public or friends, she has to upload the pictures to a Web server. The user can also send the

images through MMS or some other means of messaging to the clients. With the Mobile Host deployed on the smart phone, interested people can access the Mobile Host using a standard web service client or a Web client, and can browse through the pictures they are interested in. The service is comparable to any other online image album service or blog service but implemented on the mobile terminal.

Location Data (GPS) Provisioning Service: This dedicated web service from the Mobile Host, provides the exact location information of the smart phone, as GPS (Global Positioning System) data [59]. The service uses a Socket GPS receiver for fetching the GPS co-ordinates. The external device is connected to the smart phone via Bluetooth. The GPS data can also be collected while taking the pictures and these two details can be mapped together, giving scope for many interesting scenarios like the traveller's diary and etc. The GPS co-ordinates can always be mapped to geo spatial maps.

2.4 Commercial Applications and Usage Scenarios

An interesting commercial usage scenario involves the coordination between journalists and their respective organizations. Journalists can be at different locations across the globe, covering different events like the sport events, conferences, etc. An editor can always keep track of the location of journalists and the content they have gathered. Standard client applications can be developed for the editor, which synchronize the information stored by editor and data at the Mobile Host. The key difference to the more traditional solutions where journalists upload their contents to a server held by the Editor is that parallel access to the Mobile Host by both the journalist and the editor is possible; even other journalists in the team can look at the mobile information thus better synchronizing their activities, e.g. in the coverage of some major distributed event. Thus, the journalists can concentrate more on their job of collecting, as they don't have to upload the data, every time they get something interesting [56]. Everyday life usage scenarios driven by regular people are even more interesting. These scenarios of use of mobile web services include citizen journalism, emerging monitoring under crisis management, traffic monitoring, and other. For example of citizen journalism, regular people may create, collect, consume, comment, edit and share news in different media. In average the quality of this information will obviously be lower than professionally prepared news. However, the network effects of participation and the power of "The Long Tail", proved by emerging second-generation of web-based services [47], make this kind of usage scenarios promising.

In addition, Mobile Host provides a large scope for many applications in the m-learning (mobile learning) domain. As the Mobile Host, the mobile terminal can provide access to information like pictures, audios, videos, tags, documents, location details, and other learning services [13]. Many m-learning application scenarios can be envisioned, like a mobile learning media sharing service and expertise finder service. In the mobile learning media sharing scenario, learners can share audio or video lecture recordings or go for the field study and take the pictures of the location. Peers can then browse through the pictures taken, add tags, and give their suggestions or comments. In an expertise finder learners can look for reliable access to learning resources, persons who share the same interests, and experts with the required know-how that can help achieving better results. In the e-learning aspect these experts can share the information among the other users. Examples of these use cases could be exchanging the mathematical formulas [9] and the experts validating them or even correcting them.

Regarding industrial applications, we have studied the business of remote maintenance services for machinery equipment in the pulp&paper industry [38], [51], [43]. The case company, Metso Paper Inc. specializes in pulp and paper industry processes, machinery, equipment, related know-how, and after sales services. There is a need for the cross-organizational secure provisioning of maintenance services by experts who are usually on the move, and who have only their handheld devices, or laptops in the best case [51]. The maintenance experts use mobile phones, PDAs, laptops to access traditional web services that provide functionality for the condition monitoring, billing, maintenance, faults analysis, repair management, and other activities. These traditional web services are provided by machinery control systems or their vendors through Internet. A case study of one such a service could be found in [33]. Mobile phones host applications which are clients for these services. On the other hand, there is a need to collect data from mobile phones. These data includes history of maintenance activities (logs), collected information on customers' sites of different media (text notes, photos, videos, audio recordings, etc), current locations of experts, their availability (load of experts with other tasks and/or maintenance requests), and other. Mobile web services may facilitate and generalize the design of solutions for accessing the data that reside on experts' mobile devices.

Another promising industrial commercial area is a business of decentralized network-centric management of power-networks which are owned by different businesses. We have also conducted a case study in the domain of distributed power network management [42] that we performed in collaboration with ABB company (Distribution Automation unit). ABB is a global vendor of hardware and software for power networks. The power networks themselves are owned, controlled and maintained by some local companies. The power networks have geographically distributed complex structures with different equipment. Different companies cooperate in order to manage power networks. Operators and experts remotely monitor the power networks and prescribe changes. Field maintenance crews collect information and implement prescribed changes on sites. The communication between different actors may be very important for fault localization, network reconfiguration, network restoration, and other. Moreover, rapid access to the onsite information can highly improve some of the activities and it is especially important for the decision making operators and experts. For example, onsite information about weather conditions,

ongoing forest or construction works, or natural phenomena can be used for evaluating existing threats, cause and effects for the power network. This information, which is usually collected by mobile devices, may greatly improve the quality of power-network management. Onsite information may be also used just to extend the operators' view of the power network. Thus, there is a need to integrate existing control systems, tools and application with the mobile devices that are able to provide contextual onsite information. Web services on the mobile devices provide the generic access solution to the information that is collected by the field crews.

3 Security Considerations for Mobile Web Services

Widely recognized generic security goals are confidentiality, availability, reliability, manageability, accountability, responsibility, integrity, non-repudiation, anonymity, and privacy. These generic security goals cannot effectively serve for evaluation of designed access control solutions. In order to provide proper justification of access control measures, we evaluate security threats in mobile environments, review conventional security requirements for web services, analyze security-sensitive characteristics of mobile web services, and finally define critical success factors for controlling access to mobile web services.

3.1 Security Threats in Mobile Environments

As web services use message-based technologies for complex transactions across multiple domains, traditional point-to-point security paradigms fall short. Potentially, a web service message traverses through several intermediaries before it reaches its final destination. Therefore, the need for sophisticated end-to-end message-level security becomes a high priority and is not addressed by existing security technologies. The adhoc nature of mobile environments further complexes the realization of this end-to-end security.

The mobile web service are prone to different types of security breaches like spoofing, unauthorized access, tampering, network eavesdropping, denial-of-service attacks, man-in-the-middle attacks, intrusion and etc [58], [38]. Spoofing is a means of accessing a system with false identity. With spoofing the original source of an attack can be hidden and access to a service can be gained as a legitimate user or host, thereby elevating sensitive privileges. Proper authentication and authorization principles are to be used to cover spoofing and unauthorized access. Tampering is an act of unauthorized modification of a web service message in the network. Potential targets of this attack are the messages with out proper encryption and signing. Network eavesdropping is to monitor traffic for sensitive data such as plaintext passwords by placing sniffers in the middle of the network. Proper encryption and digital signatures can help in avoiding tampering and network eavesdropping attacks.

Replaying a valid changed or unchanged message to a web service by impersonating the client is referred as replay attack. The unchanged message replay attack, also called basic replay attack, can be avoided by using nonce, a cryptographically unique value, with the web service message. In the changed replay attack also known as the Man in the middle attacks, the attacker captures the messages, changes the contents and replays them to the web service. Denial-of-service is a process of making a system, server or application unavailable. For each individual service, maintaining and understanding the collection of data can help in protecting it from denial-of-service attacks. But having such a scenario implemented on the resource constrained mobile phones could be impractical. Security policies and high-level access control should help to a certain extent in this regard.

3.2 Conventional Web Service Security Requirements

Almost all existing standards, related to security of web services, explicitly specify requirements for security measures. These requirements are derived from the generic security goals in response to possible security threats. For example, W3C had a Web Service Architecture working group [60]. This group produced several standards which are closely related to the focus of our research. The Web Services Architecture specification [11] defines standard reference architecture of web services with fundamental functional components and their relationships between each other. Amongst several supporting specifications, the Web Services Architecture Requirements specification [5] defines architecture goals and requirements including security related considerations. The top-level architecture goals involve a goal of providing "a secure environment for online processing". This top-level goal is broken down to requirements towards a web services security framework. Requirements for the web services security framework define that this framework must enable authentication of communicating parties and authorship of data, authorization, confidentiality, integrity, non-repudiation, auditing, and it must provide means for management of a web services' security policies. Additionally, this security framework should provide means for the availability of web services and means for security administration. Privacy requirements define that web services architecture must support P3P [14], i.e. expressing, delegating, propagating, accessing the privacy policy, and interaction of anonymous parties.

In response to the identified requirements in [5], the web service architecture specification [11] addresses issues of web services security. This includes definition of a policy model and discussions on security policies, message level security threats, security requirements, security consideration of the architecture, and privacy considerations. The policy model serves as a framework for policy-based security management. This framework is based on policy descriptions (machine processable), permissions (allows actions or states), obligations (prescribe actions or states), and policy guards (enforce policy) like permission guards (enforce permissions) and audit guards (enforce obligations). According to this specification [11], the fundamental concepts for security policies are resources to be protected, policy guards i.e. security mechanisms, and machine-processable policies. The message level security threats are subset of above discussed security threats in mobile environments (section 3.1). Discussion of security requirements, identified in [5], gives shallow insight into providing end-to-end security for web services. In conclusions the Web Services Architecture specification [11] acknowledges that the designed architecture effectively meet identified requirements except related to security and privacy.

3.3 Existing Standards to Secure Web Services

The web service communication is based on the SOAP protocol, which in turn exchanges information in XML format. The SOAP protocol from W3C does not directly provide means for secure communication. OASIS standards with the help of W3C XML standards like XML Signature and XML Encryption provide cryptographic protection, and thus help in securing a web service message. The WS-Security specification from OASIS is the core element in web service security realm [46]. It 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. However, this specification does not take into account authorization mechanisms.

Apart from WS-Security, web service security specifications also include WS-Policy [6] which defines the rules for service interaction, WS-Trust [2] which defines trust model for secure exchanges and WS-Privacy which states the maintenance of privacy of information. In combination WS-Policy, WS-Security, and WS-Trust, can state and indicate conformance of organizations to stated privacy policies. WS-Privacy specifies a model for how a privacy language may be embedded into WS-Policy descriptions and how WS-Security may be used to associate privacy claims with a message. WS-Privacy also describes how WS-Trust mechanisms can be used to evaluate these privacy claims for both user preferences and organizational practice claims. Built with these set of basic specifications are the specifications, WS-SecureConversation [3] that specifies how to establish and maintain secured session for exchanging data, WS-Federation [7] which defines rules of distributed identity and its maintenance, and WS-Authorization which processes the access rights and exchangeable information [25].

Security Assertion Markup Language (SAML) is an extension of WS-Security from OASIS [54]. SAML specifies the language to exchange identity, attribute and authorization information between parties involved in web service communication in an interoperable way. SOAP is used as the SAML request/response protocol transport mechanism. SAML requests and responses reside with in the SOAP body. SAML could help in achieving Single-sign-on (SSO). SSO is a mechanism where the authentication context of a consumer, can be maintained across multiple services.

The Extensible Access Control Markup Language (XACML) specification [39] provides the language to express access control policies. It additionally standardizes mechanisms, algorithms and architectural components for managing these access control policies. While XACML is the state-of-the-art approach in SOA, there are some limitations of applicability of this specification in emerging environments that are highly open, distributed and dynamic [62]. After analysis of security threats, conventional security requirements and provided solutions, we can conclude that the area of secure provisioning of (mobile) web services still has open research and development questions.

3.4 Ensuring Message-Level Security of Mobile Web Services

In order to ensure the message-level security of mobile web services, we tried to adapt the WS-Security specification in the mobile web service communication. The mobile web service messages were processed with different encryption algorithms, signer algorithms and authentication principles, and were exchanged according to the standard. The performance of the Mobile Host was observed during this analysis, for the caused extra delay and the variation in stability of the Mobile Host with the launched security overhead.

For this analysis, a SonyEricsson P910i smart phone was used. The device supports J2ME MIDP2.0 [31] with CLDC1.0 [29] configuration. For cryptographic algorithms and digital signers, we used java based light weight cryptographic API from Bouncy Castle crypto package [12]. kSOAP2 was modified and adapted by us according to WS-Security standard and utilized to create the request/response web service messages. The Mobile Host was redesigned with J2ME and the adapted kSOAP2. The Mobile Host with the J2ME does not have the extensive performance analysis and results as with PersonaJava based Mobile Host. Since the processing capabilities of both the JVMs are not significantly different we can safely assume that most of the observations from our previous analysis would still be valid for J2ME based Mobile Host [56]. To achieve confidentiality, the web service messages were ciphered with symmetric encryption algorithms and the generated symmetric keys were exchanged by means

of asymmetric encryption methods. The messages were tested against various symmetric encryption algorithms, IDEA 128, IDEA 256 and DES 64, along with the WS-Security mandatory algorithms, TRIPLEDES, AES-128, AES-192 and AES-256 [53]. The PKI algorithm used for key exchange was RSA-V1.5 with 1024 and 2048 bit keys. Upon successful deployment of confidentiality, we considered data integrity on top of confidentiality. The messages were digitally signed and were evaluated against two signature algorithms, DSAwithSHA1 (DSS) and RSAwithSHA1 [58].

The results of our study are welcoming and the mobile web service messages of reasonable size, approximately 2-5kb, can be secured with web service security standard specifications. With this study we are recommending that the best way of securing SOAP messages in mobile web service provisioning domain is to use AES-256 bit key for encrypting the message and RSAwithSHA1 to sign the message. The symmetric keys are to be exchanged using RSA 1024 bit asymmetric key exchange mechanism. The detailed performance analysis also suggested that not all of the WS-Security specification can be adapted to the mobile web service communication [58].

3.5 Specifics of Mobile Web Services Related to Access Control

In this section we summarize characteristics of mobile web service provisioning from the perspective of access control. We identify critical success factors for the access control solutions. They are derived taking into account specifics of mobile web service provisioning, technical and commercial usage scenarios, implemented sample mobile web services, security threats in mobile environments, conventional security requirements of web services, existing security solutions and standards, and our evaluation of message-level security.

For mobile web services, we thoroughly reuse all the standards that define how to describe (WSDL), to discover and to publish (UDDI), to interoperate (SOAP), etc with conventional web services. Consequently, access control measures must be based on and compatible with these widely adapted standards. In addition, access control solutions should rely on already existing standards to secure web services. In other words, despite of specifics of mobile web services, they should follow standardized web services architectures whenever possible and only sound arguments may justify incompatibilities.

It is critical for applicability of access control solutions to consider all possible technical use cases. Unauthorized access to mobile web services should be prevented regardless of type of technical use cases. Each of the above described (section 2.2) technical use cases of mobile web service provisioning has some specifics. These specific characteristics create complex real-world settings for the realization of access control. For example, a request from an Internet client to a mobile web service goes through several components along its route in the case of GRPS connection. There are HTTP gateway on the border between mobile operator's network and Internet, and another HTTP reverse proxy could be deployed in Internet for mapping a permanent DNS name of mobile web service to its dynamically allocated IP address [61]. Access control guards or policy enforcement guards, may be also added as standalone components and nodes along the route of request. These all should be aligned with the notion of intermediaries that are defined by SOAP.

Even for the two simple mobile web services (section 2.3), there is an obvious need to provide proper security measures in general and access control in particular. Mobile users demand to have expressive, flexible, pervasive and ease-to-use means for authorizing access to the sensitive personal information. For these sample mobile web services, users should be able to fully control the process of sharing image album data and location information. Commercial and business usage scenarios highlight an additional issue associated with mobile web services. Holders of mobile phones are not always their owners. In the cases of mobile journalism, power-network management and paper machinery maintenance, journalists, engineers and maintenance experts are not authorities that define access control policies for news, for on-site collected information, and even for their presence and location. Businesses and organizations that hire these people and commission their work should have full or partial authority over access control process depending on concrete needs. Usage scenarios (section 2.4) also reveal that mobile web services may provide access to the content with the great variety of media types. Therefore, access control solutions should support the major standardized formats of mobile content. It is desirable that access control solutions could be easily extensible for new and emerging formats. Heterogeneity in mobile web service provisioning has wider scope than just variety of content formats. There are different wireless and wired network protocols, client devices and devices of service providers with different sets of features and characteristics, different settings of mobile operators, convergence of mobile and WLAN networks, different available security measures, and other.

While industrial business applications of mobile web services mainly support business-to-business relationships in "closed" or "limited" environments, access control measures should be more generally oriented to "open" environments. This is needed to support customer-to-business and customer-to-customer interactions. Mobile environments are open in a sense that they create mobile social networking of mobile users. Who can both use the environment (consume mobile web services) and contribute to the environment (provide mobile web services). The Mobile Host is built on the top of open standards, software and technologies developed by open communities. Open environment of mobile web service provisioning introduces more challenging security problems due to a greater amount of risks and threats. Also, open environments require complex and special access control solutions to establish relationships from scratch with previously unknown clients and providers of mobile web services.

Another important requirement in the context of mobile infrastructures and wireless communication is that access control solutions should be tolerant to ad hoc nature of interactions and to dynamism of mobile users. This ad hoc nature, complexity and dynamics lead to unpredictable changes of environment's states. This complicates adapting traditional techniques to secure mobile web services. Mobility is one of the major factors behind dynamism and complexity. However, access control solutions should not limit mobility and spontaneous behavior of users.

Ubiquity and pervasiveness of mobile and wireless technologies have tightened the digital and physical worlds to the extent when security becomes the ultimate issue. The major implication of penetrating technologies on security is that the risks and negative consequences of security threats become higher than ever. Thus specific and conventional requirements to the security level of mobile web services are demanding. Additionally, the security infrastructure itself has to become pervasive enough to naturally fit current experience and expectations of mobile users. Therefore, access control has to reduce to the minimum attracting attention and gaining time of users.

Limitations of mobile devices comparing to the stationary computers are largely discussed in the literature and are known by users. Access control guards for mobile web services must be usable within limited capabilities of mobile phones. Access control solutions should particularly bear with limited connectivity and low transmission rates, battery power shortages, lower processing power and storage space, limited means for user input/output interactions like tiny screens, keyboards, etc. There are also other limitations which are not technical. Mobile users tend to simultaneously perform multiple tasks while using mobile phones, e.g. driving a car. Mobile users are also prone to errors. Due to technical limitations, access control solutions as well as other utility applications should introduce low overheads in terms of traffic, performance, storage, power consumption, etc. In other words, access control should be cost-effective not only in monetary fees but also in battery power consumption for example.

Context-awareness is the next important factor of mobile web services and access control. Location, time, user's activity, battery power level, history, etc are typical contextual data for mobile computing. Authorization of incoming requests may rely only on these contextual factors. For example, an access control decision may deny new requests because of overloading a mobile web service without evaluation of other information. In this example, an enforcement mechanism ensures availability of mobile web services, in contrast with the most common support of confidentiality. While we cannot envision all possible contextual factors that users want to take into account, it is hard to define the border of the access control system. Thus, the access control should be extensible enough to leave a space for user-driven personalization or intelligent self-configuration of access control policies and mechanisms. Context-awareness, extensibility, intelligence, and personalization demand a great expressiveness of underlying access control models and languages for policy specification.

Flexibility is vital for applicability and interoperability of access control solutions in the heterogeneous mobile environments. This flexibility of access control functionality should reflect extensibility and expressiveness of access control models and languages. The architecture of access control must follow component-oriented style of design. This is needed in order to compose and to reconfigure required access control functionality for specific usage scenarios using modular and flexible access control mechanisms and tools.

The last important issue to consider is security of access control solutions. Access control infrastructure should obviously be secure by design. For example, the guards are typically designed in such a manner that requests cannot bypass them. The availability of security-related components should be high. Accountability of access control mechanism must be ensured by proper logging for audit practices and technologies. Manageability is also an issue for the access control administration. These generic security goals could be further elaborated under a concept of self-management of access control solutions. Self-management for security is self-protection. It is a vision of pro-active context-aware autonomic security mechanisms [36], [20].

We qualitatively evaluated characteristics of mobile web service provisioning and identified a concise list of ten critical success factors of access control solutions for mobile web services provisioning: *compatibility, applicability, extensibility, openness, nomadic nature, pervasiveness, context-awareness, usability, flexibility, and self-security*.

4 Semantics-Based Access Control for Mobile Web Services

For the management of access control to mobile web services, we propose to use Semantics-Based Access Control (SBAC). SBAC [40] is the result of adoption of the Semantic Web vision and standards [10] to the access control research and development field. SBAC encompasses administration and enforcement of access control policies based on semantics of related concepts. This is the most suitable approach to handle *openness*, *dynamics*, *pervasiveness*, *heterogeneity*, and *distributed nature* of the mobile web service provisioning.

4.1 Access Control Model and Policy Language

The model-theoretic semantics of SBAC [40] is an extension of the direct model-theoretic semantics defined in the Web Ontology Language (OWL) standard [48] and Semantic Web Rule Language (SWRL) [21]. The SBAC model is

a result of introducing vocabularies and interpretations of specific security-related concepts inheriting all features of OWL and SWRL due to the *compatibility* with their direct model-theoretic semantics. The SBAC model has been expressed in the form of ontologies [41]. Thus, SBAC policies are OWL ontologies. The ontology engineering constitutes the traditional domain modeling. The SBAC ontologies consolidate and formally specify knowledge of the access control research and development domain. Similar to the traditional access control models, the SBAC ontologies aim to support the formal specification of policies with respect to standards, legal regulations, domain practices, agreements, approaches, traditions, etc. Ontologies also define languages. The SBAC ontologies define the SBAC policy language. Figure 2 shows the core part of the SBAC model [40].

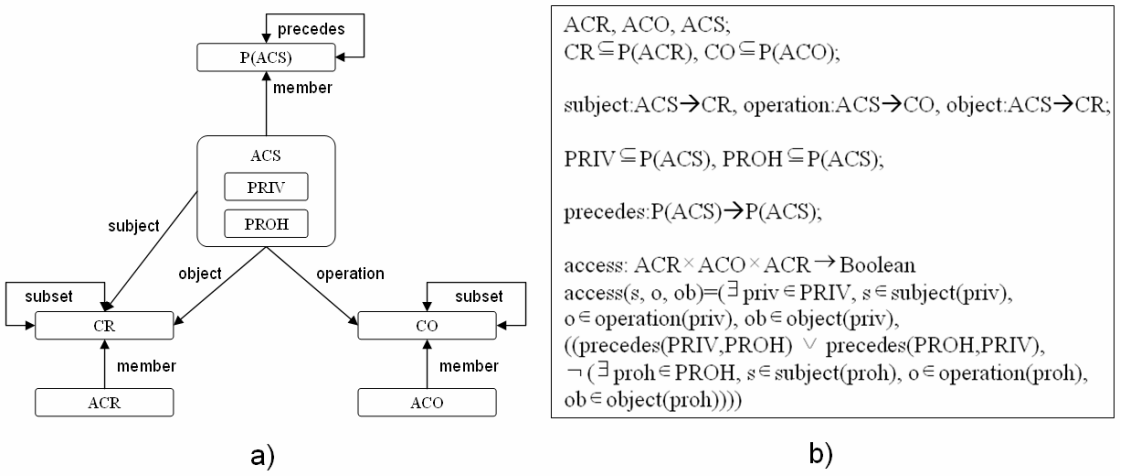

Figure 2: The SBAC model: graphical a) and textual b) representations

ACR is a set of individual resources. A resource is an entity of physical or digital world that is a subject or an object of access. Definition of the resource as a set for subjects and objects gives more flexibility in access control rights specification. Humans and intelligent applications are subjects of access to mobile web services. Domain and policy ontologies engineers annotate and classify resources that are possible subjects of access. Web services have two distinct types of results. A web service can return data or/and affect some real world objects. Generally, SBAC must protect objects of the both types. We have studied this *heterogeneity* of access objects in the context of semantic web services [43]. Our considerations and decisions for semantic web services remain the same for mobile web services. We decided to define access control statements based on inputs and outputs of mobile web services to represent protected objects. Inputs indirectly determine both information objects to be returned and real-world objects to be affected. Access control statements on outputs cannot always be verified before the invocation of services. On the other hand, access control decisions cannot be made after the actual invocation of services in cases when services have effects on real-world objects bundled with outputs. Thus, to support the maximum levels of *applicability*, *usability* and *flexibility*, there is a need to provide pre-authorization and post-authorization mechanisms. ACO is a set of individual operations that could be actions, transactions, access modes, etc. Each web service may have many operations. The WSDL operation is the lowest granularity level modeling concept that denotes operations used by subjects accessing protected objects in SOA. Thus, the WSDL operation is the most appropriate concept to be considered as an access control operation.

CR is a set of subsets of resources. CO is a set of subsets of operations. Resources and operations are classified and collected to named sets. CO and CR sets can be partially ordered by the transitive subset relation. This forms hierarchies of resources and operations. ACS is a set of access control statements that denotes a many-to-many abstract relation between subject, operation and object of access. We use three binary relations from access control statements to subject resources, operations, and object resources. The main feature of the access control statement semantics is that these statements are specified between classes instead of individuals. PRIV is a set of privilege statements. It is a subset of ACS. A privilege is an authorization of resources to access other resources using some operations. A decision of access granting or prohibiting depends on memberships of subject, operation and object elements in sets that are in definitions of privileges. The decision algorithm evaluates the containment relation (member) between individual elements and sets taking into account partial order of sets. PROH is a set of prohibition statements. It is a subset of ACS. Support of only positive authorizations in the form of privileges guaranties a conflicts free specification of access control policies. However, even in this case the model has an implicit prohibition that everything is prohibited unless it is privileged. For opposite example, block lists in mobile phones prohibit accepting calls from given phone numbers while there is a general implicit privilege to accept calls from everybody. Introducing means for the specification of prohibitions in the SBAC model enhances expressivity (*usability* and *applicability*) of the policy language. Policies with privileges and prohibitions are not free from conflicts in an arbitrary case. These policies require mechanisms to resolve conflicts and ambiguity. Following the fundamental principle of security, safer is better, prohibitions always precede privileges. Although in the most cases policies will follow the

fundamental principle, there is a need to explicitly specify the precedence between privileges and prohibitions. This precedence between sets of access control statements is modeled with a binary relation “precedes”.

Interpretation of ontologies is the key issue for evolution, consistency, reasoning and organizing features of SBAC, domain knowledge and concrete policies separately in different ontologies. This is needed for the *flexible* and *extensible* knowledge reuse with the high conceptual granularity. Annotation and ontology properties help to record a history of evolution of SBAC and domain ontologies, policies, trust agreements, etc. The definitions of when and how a collection of ontologies and axioms and facts is consistent and entails an ontology or axiom or fact provide background for reasoning and maintaining integrity of SBAC policies [48].

4.2 Policy Enforcement Function

The SBAC functionality is naturally separated to two parts: the run-time authorization function, also called enforcement, and the administrative function. The enforcement function controls run-time access of requestors to protected resources, according to ontology-based access control policies, attributes of subjects, objects and operations. The SBAC administration function defines mechanisms of manipulation with SBAC data e.g. semantic annotations of resources and operations, domain ontologies, ontology-based policies, configuration settings for the enforcement function, and other.

The enforcement function involves several architectural components and nodes. A subject of access is a web service client that invokes protected web services to access data or to affect real world. The client accesses the mobile web services on the Mobile Host from a mobile phone or regular computer through Internet and mobile networks. A guard mediates access to the protected mobile web service and enforces rules of corresponding access control policies. The guard must evaluate all requests, correctly evaluate semantic profiles and policies, be incorruptible and nonbypassable. A policy is an ontology. It has access control statements that define what users may access what data using what mobile web services. A subject and object descriptors are well known patterns that provide access to attributes of subject and objects of access. In SBAC, we specialize the descriptor pattern into semantic profiles for users, data, mobile web services, policies and context. Figure 3 illustrates a control flow for the enforcement function that is driven by the SBAC guard.

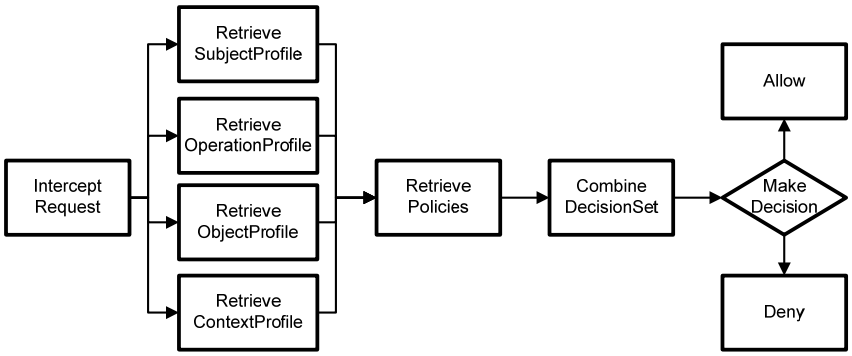

Figure 3: The SBAC enforcement function

Let us consider a concrete simple example in order to reveal roles of the above described components and roles of activities that comprise the algorithm of the enforcement function. In personal area networks (PAN) users can provide access to information about location of their mobile phones using a GPS device and mobile web service that was described in the section 2.3. A client sends a SOAP\XML request over Bluetooth to a mobile web service in order to get location data of a provider (figure 1, iteration 4). A guard intercepts this request. After that, the guard initiates the process of request evaluation. The guard extracts from request’s header a URI of web service’s operation and supplied credentials of the user. These are input parameters for the activities of retrieving or creating of semantic profiles for the user and operation. Our web service’s operation does not have any input parameters and always provides the current location only. Thus, there is no need to retrieve the semantic profile of object of access. However, for a more generic case, the guard can create or retrieve the semantic profile for the object based on the input SOAP\XML massage and its WSDL description. After collecting semantic profiles of the user and operation, the guard retrieves applicable access control statements. In our example, this retrieval uses information about memberships of the user and operation to classes, which were used for specification of privileges and prohibitions. The semantic profiles of the user and operation, retrieved policies, SBAC ontologies are loaded to a decision set for a further decision making. The provider grants privileges to all users that try to access location information using the mobile web service over Bluetooth protocol. There is also a prohibition for a set of users (blacklist) to access the service. This set is specified by enumeration of Bluetooth IDs for some of users. The provider specifies that prohibitions precede privileges. Assuming the user of our example is not in this blacklist, the decision making activity allows this access, otherwise the access is denied.

The architectural components of the SBAC enforcement function are deployed to the Mobile Host and middleware nodes depending on characteristics and requirements of usage scenarios. There are several reasonable options of deployment of the SBAC components for protected mobile web service provisioning with unique characteristics and implications on the level of security and quality of mobile web services. Figure 4 shows these deployment options.

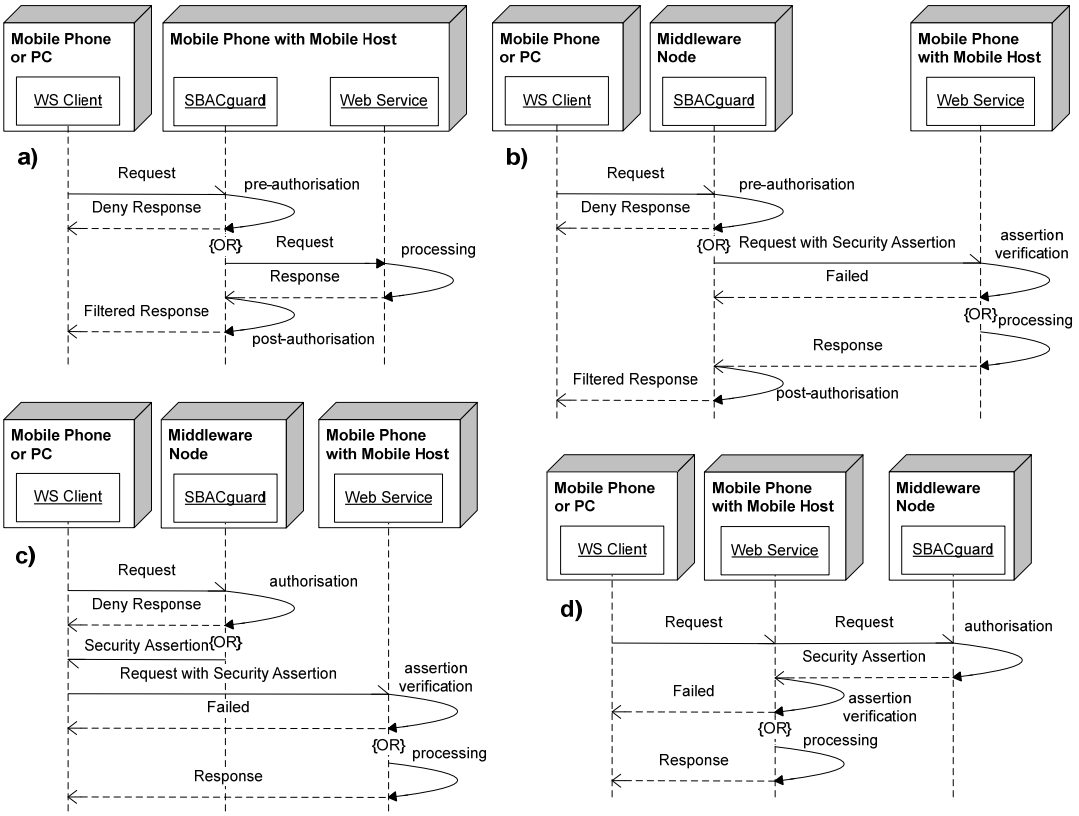

Figure 4: Deployment options

The embedded guard (option a) is the most *applicable* option for the *pervasive* mobile web service provisioning within the P2P usage scenarios between mobile clients and Mobile Hosts [55]. The clients directly access services. Interactions between the embedded guard and services can be done using procedure calls without delays of wireless or wired asynchronous communication. One crucial advantage of this option is the opportunity to perform post-authorizations i.e. procedures of access control that must be performed after service enactment e.g. filtering of the response. This option supports the principle of end-to-end security in contrast to other options. However computational limitations of mobile phones demand the *nomadic* functionality of the guard. This undermines the possibility to use complex semantics-based algorithms for the embedded decision making process.

The deployment option b) illustrates the middleware guard that is an intermediate web service proxy. This guard provides the same interface as the original mobile web service, decorates web service invocation with the SBAC policy enforcement mechanism, and delegates authorized requests to the mobile web service. The middleware guard is deployed in the Internet or in the proprietary mobile operator infrastructure. When the guard is in the Internet, clients are able to access it in the traditional way. Moreover Mobile Hosts receive less number of requests or, in other words, only authorized requests. This improves the scalability of the Mobile Host. The post-authorization is still possible. The middleware guard can represent several Mobile Hosts and web services. Mobile-to-mobile requests experience delays of wireless communication twice when the guard is not embedded but is a middleware component. An additional component of the Mobile Host has to validate security assertions (using SAML) of the guard. The validation of security assertions is necessary in order to check that a security assertion is consistent with a request.

The deployment option c), where the guard is a third-party authorization authority, creates additional inconveniences for clients. They have to get authorization assertions prior to access protected mobile web services. Then the component deployed on Mobile Hosts validates security assertions provided with requests like in the previous option. Although this case might look too complex, however this is probably the most *applicable* option for the industrial, commercial or professional use of mobile web services when clients can get security tokens with long period of validity on the basis of their memberships in or subscriptions to different organizations, social networks, commercial

services, etc. This option allows direct multiple requests to mobile web services using the same security token over time without overheads of the authorization decision making process for each request.

Delegation of authorization of option d) is the last option we considered. Mobile web services initially receive all requests directly from clients and then outsource the decision making procedure to the middleware guard. While such kind of deployment is possible, it has several significant shortcomings without clear advantages over the above described options. There are following needs: to embed the enforcement component for authorization messaging with all possible time overheads; to verify signatures of the guard; to process all requests from clients; and other. One advantage is that the performance demanding SBAC functionality is executed by the middleware guard.

4.3 Prototyping Decision Making Procedure and Testing Performance

We piloted the decision making procedure to test its performance (Make Decision activity in Figure 3). This prototype implementation is *applicable* for the middleware guard as it is, as discussed in section 4.2 with figure 4. The prototype has to be modified to fit limitations of java virtual machine of mobile phones, because we used Java 2 standard edition development kit version 1.5 [26] as a programming language. For the management of policies that are OWL ontologies, we used a semantic web framework for java Jena [32]. Jena was developed within the HP Labs Semantic Web Programme [22]. Jena has also a SPARQL processor ARQ [4]. SPARQL is a query language for Semantic Web data [50]. Protégé [49] was used for piloting SBAC ontologies in the RDF/XML exchange syntax of OWL. Protégé is an open source and free editor of Semantic Web data with the number of plugins for editing and visualizing OWL ontologies. For the programming and testing performance we used tools from Eclipse. Eclipse is an open source community [15] that produces extensible integrated development environment (IDE). The Eclipse Test & Performance Tools Platform (TPTP) project consists of four subproject one of which provides tools for tracing and profiling java applications for further analysis of their performance [16].

The internal structure of the decision maker has in-memory knowledge base (decision set) in the form of ontology model provided by the Jena framework and the query engine (query processor) provided by the ARQ processor of SPARQL queries. All ontologies were placed into the web server and were accessible via HTTP protocol. The fastest response time of the decision making process corresponded to the simplest policy and domain ontologies. The policy ontology consisted of one class for mobile web service clients with one user, one class for protected objects with one individual and one class for mobile web service operations with one operation. The policy had the only one privilege statement defined using the above described classes. All RDF statements of SBAC and policy ontologies were loaded into the decision set during the start-up process. The SPARQL query corresponded to the authorization rule for policies defined using only privilege statements.

The average cumulative CPU time of the decision making process was 0.827 seconds. The query execution over the decision set took the major fraction of this time. The hosting computer was IBM PC with the CPU AMD Athlon XP 3000+, 1 GB of RAM, and OS Microsoft Windows XP Professional version 2002 with Service Pack 2.

5 Conclusions and Future Research Directions

This paper contributes ideas to several areas that are related to mobile web services, secure mobile web service provisioning, and access control solutions based on Semantic Web standards. We have been providing ideas and developing prototypes for the emerging research area of mobile web services. This paper adds more technical and commercial usage scenarios of mobile web services in order to widen demands to applicability of enabling solutions. We recognized that security is an issue in the most usage scenarios. The thorough analysis of critical success factors for access control solutions and for security in general forms a solid background for future requirements engineering of concrete and practical access control measures. The concise list of critical success factors served for the qualitative evaluation of SBAC. The paper presents the SBAC model, architecture, deployment options, and piloted decision making function in the context of mobile web service provisioning. Regarding the prototype that is fully based on open source and freeware components, it shows applicability of our approach towards SBAC for middleware guards. Development of the embedded guard following SBAC would require a more robust solution.

Practical expectations for mobile web services are quite positive. Security measures are important components in order to ensure applicability of mobile web services in real-world settings. Practical implications of our research can be realized with industry-driven developments and deployments of mobile web services with adequate security measures. We are sure that mobile web services can change the design of potential applications for personal computing, and pervasive and distributed information systems. Access control solutions based on Semantic Web standards might help to realize some advanced mechanisms, e.g. access control based on learning of users' behavior, ratings-based fair provisioning of mobile web services, pro-active authorization based on monitoring of peers in user's proximity, and other.

Acknowledgement

The work has been supported by German Research Foundation (DFG) as part of the Graduate School "Software for Mobile Communication Systems" at RWTH Aachen University, and by the Rector of the University of Jyväskylä, Finland. We are also grateful for the funding from SmartResource, MODPA and MWSP research projects. The SmartResource (http://www.cs.jyu.fi/ai/OntoGroup/) is hosted at the Agora Center of the University of Jyväskylä and is financially supported by Tekes (National Technology Agency of Finland), and cooperating companies (ABB, Metso Automation, TeliaSonera, TietoEnator, and Jyväskylä Science Park). The Mobile Design Patterns and Architectures (MODPA, http://www.titu.jyu.fi/modpa) had been hosted at the Information Technology Research Institute (University of Jyvaskyla), funded by the National Technology Agency of Finland (Tekes) and industrial partners: Metso Paper, Nokia, SysOpen Digia, SESCA Technologies, Tieturi, and Trusteq. The MWSP (http://www-i5.informatik.rwth-aachen.de/lehrstuhl/staff/srirama/MWSP.html) is hosted at Information Systems Group (RWTH Aachen University) and funded by German Research Foundation (DFG). The authors would also like to thank Ericsson GMBH for its support in the MWSP project.

References

[1] 4G Press. (2005, November). World's First 2.5Gbps Packet Transmission in 4G Field Experiment. [Online]. Available: www.4g.co.uk/PR2006/2056.htm.
[2] Anderson, S., Bohren, J., Boubez, T., Chanliau, M., Della-Libera, G., Dixon, B. et al. Web Services Trust Language (WS-Trust). [Online]. Available: http://specs.xmlsoap.org/ws/2005/02/trust/WS-Trust.pdf.
[3] Anderson, S., Bohren, J., Boubez, T., Chanliau, M., Della-Libera, G., Dixon, B. et al. Web Services Secure Conversation Language (WS-SecureConversation). [Online]. Available: http://specs.xmlsoap.org/ws/2005/02/sc/WS-SecureConversation.pdf.
[4] ARQ, a SPARQL processor. [Online]. Available: http://jena.sourceforge.net/ARQ/.
[5] Austin, D., Barbir, A., Ferris, C., and Garg, S., (eds.)., Web Services Architecture Requirements, W3C Working Group Note, 2004, W3C. [Online]. Available: www.w3.org/TR/2004/NOTE-wsa-reqs-20040211/.
[6] Bajaj, S., Box, D., Chappell, D., Curbera, F., Daniels, G., Hallam-Baker, P. et al. Web Services Policy Framework (WS-Policy), Version 1.2, http://specs.xmlsoap.org/ws/2004/09/policy/ws-policy.pdf.
[7] Bajaj, S., Della-Libera, G., Dixon, B., Dusche, M., Hondo, M., Hur, M. et al. Web Services Federation Language (WS-Federation), version 1.0, http://specs.xmlsoap.org/ws/2003/07/secext/WS-Federation.pdf.
[8] Balani, N., Deliver Web Services to mobile apps, IBM developerWorks, 2003
[9] Belov, N., Braude, I., Krandick, W., and Shaffer, J. Wireless Internet Collaboration System on Smartphones, 3rd International Workshop on Ubiquitous Mobile Information and Collaboration Systems, UMICS 2005, a CAiSE'05 workshop.
[10] Berners-Lee T., Handler J. and Lassila, O. (2001), The Semantic Web, Scientific American.
[11] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., and Orchard, D., (eds.). Web Services Architecture, W3C Working Group Note, 2004, W3C; www.w3.org/TR/2004/NOTE-ws-arch-20040211/
[12] Bouncy Castle, Bouncy Castle lightweight cryptography API, http://www.bouncycastle.org/documentation.html.
[13] Chatti, M., Srirama, S., Kensche, D., and Cao, Y. Mobile Web Services for Collaborative Learning. in Proceedings of the 4th International Workshop on Wireless, Mobile and Ubiquitous Technologies in Education (WMUTE 2006), November 16-17, Athens, Greece, 2006.
[14] Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., Reagle, J. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C Recommendation, 2004, W3C; www.w3.org/TR/P3P/
[15] Eclipse, http://www.eclipse.org.
[16] Eclipse Test & Performance Tools Platform, http://www.eclipse.org/tptp/.
[17] T. Erl: Service-Oriented Architecture. Concepts, Technology, and Design, Prentice Hall, 2005.
[18] Gottschalk, K. and Graham, S., Introduction to Web Services Architecture, IBM Systems J. 41(2): 178-198, 2002.
[19] GSM World. [Online]. Available: www.gsmworld.com/technology/index.shtml.
[20] P. Horn, Autonomic computing: IBM's perspective on the state of information technology, IBM Corporation, Tech. Rep., 15 Oct. 2001. www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf
[21] Horrocks I., Patel-Schneider P., Boley H., Tabet S., Grosof B. and Dean M. SWRL: A Semantic Web Rule Language combining OWL and RuleML, W3C Member Submission, 2004, W3C, www.w3.org/Submission/SWRL/
[22] HP, Semantic Web Programme, http://www.hpl.hp.com/semweb/.
[23] Hughes, D., Coulson, G., Walkerdine, J. Free riding on Gnutella revisited: the bell tolls?, Distributed Systems Online, IEEE Volume 6, Issue 6, June 2005
[24] Hummel, J. and Lechner, U. Business models and system architectures of virtual communities. From a sociological phenomenon to peer-to-peer architectures, Int. Journal of Electronic Commerce, 2002 6(3), 41–53.
[25] IBM. Security in a Web Services world: A Proposed Architecture and Roadmap, IBM Developerworks, 2002.
[26] Java 2 standard edition development kit 1.5, http://java.sun.com/j2se/1.5.0/.
[27] Java support in SonyEricsson mobile phones P800 and P802, Jan. 2003 Developer guidelines from SonyEricsson Mobile CommunicationsAB, www.SonyEricssonMobile.com
[28] Järvinen, P. 2001. On research methods. Tampere: Opinpajan Kirja.
[29] JCP, Connected Limited Device Configuration Version 1.0, JSR 30, http://jcp.org/en/jsr/detail?id=30.
[30] JCP, J2ME Web Services Specification, JSR 172, http://jcp.org/en/jsr/detail?id=172.

[31] JCP, Mobile Information Device Profile 2.0, JSR 118, http://jcp.org/en/jsr/detail?id=118.
[32] Jena, a semantic web framework for java, http://jena.sourceforge.net/
[33] Johansson, N., and Mollstedt, U., Revisiting Amit and Zott's model of value creation sources: The SymBelt Customer Center case, Journal of Theoretical and Applied Electronic Commerce Research, ISSN 0718–1876 Electronic Version, 1(3), 2006, pp 16 – 27
[34] Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. IETF (2002)
[35] JXTA: The JXTA home page, www.jxta.org/
[36] O. Kephart and D. M. Chess, The vision of autonomic computing, Computer, vol. 36, no. 1, pp. 41--50, Jan. 2003. http://portal.acm.org/citation.cfm?id=642200.
[37] kSOAP2, A open source SOAP implementation for kVM, http://kobjects.org/.
[38] Meier, J., Mackman, A., Dunner, M., Vasireddy, S., Escamilla, R. and Murukan, A. Improving Web Application Security: Threats and Countermeasures, MSDN, Microsoft Corporation, June 2003.
[39] Moses, T., (ed.), (2005). eXtensible Access Control Markup Language (XACML) Version 2.0, OASIS Standard, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf
[40] Naumenko, A., Contextual rules-based access control model with trust, In Shoniregan C. A. and Logvynovskiy A. (Eds.), Proceedings of the International Conference for Internet Technology and Secured Transactions, ICITST 2006, 11-13 September, London, UK, e-Centre for Infonomics, pages 68-75.
[41] Naumenko A., Semantics-Based Access Control: Ontologies and Feasibility Study of Policy Enforcement Function, In: J. Filipe and J. Minguillon (Eds.), In Proc. of the 3rd Int. Conf. on Web Information Systems and Technologies (WEBIST-07), March 3-6, 2007, Barcelona, Spain, 6 pp. (In press).
[42] Naumenko A., Katasonov A., Terziyan V., A Security Framework for Smart Ubiquitous Industrial Resources, In: J.P. Müller and K. Mertins (Eds.), In: Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (IESA-07), March 28-30, 2007, Madeira Island, Portugal, 13 pp. (In press).
[43] Naumenko, A. and Luostarinen, K., 2006, Access Control Policies in (Semantic) Service-Oriented Architecture, In Schaffert S. and Sure Y. (Eds.), Semantic Systems From Visions to Applications, Proceedings of the SEMANTICS 2006, Austrian Computer Society, Vienna, Austria, pages 49-62.
[44] Naumenko A., Nikitin S., Terziyan V., Zharko A., Strategic Industrial Alliances in Paper Industry: XML- vs. Ontology-Based Integration Platforms, In: The Learning Organization, Special Issue on: Semantic and Social Aspects of Learning in Organizations, Emerald Publishers, Vol. 12, No. 5, 2005, pp. 492-514.
[45] Nunamaker, J.F., Chen, M. and Purdin, T.D.M. 1991. Systems development in Information Systems research. Journal of Management Information Systems 7(3), 89-106.
[46] OASIS, WS-Security version 1.0, www.oasis-open.org/specs/#wssv1.0.
[47] O'Reilly T., What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software. 2005, www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.
[48] Patel-Schneider P., Hayes P. and Horrocks I. (eds.). OWL Web Ontology Language semantics and abstract syntax, W3C Recommendation, 2004, W3C; www.w3.org/TR/owl-absyn/
[49] Protégé, http://www.protege.stanford.edu.
[50] Prud'hommeaux, E., and Seaborne, A. (eds.). SPARQL Query Language for RDF. W3C Candidate Recommendation, 2006, W3C, http://www.w3.org/TR/rdf-sparql-query/.
[51] Pulkkinen, M., Naumenko, A., and Luostarinen, K., Managing Information Security in a Business Network of Machinery Maintenance Services Business - Enterprise Architecture as a Coordination Tool, , In: Sangkyun Kim (Ed.), Special Issue on Methodology of Security Engineering for Industrial Security Management Systems, Journal of Systems and Software, ELSEVIER, (In press).
[52] Rollman, R. and Schneider, J. Mobile web services, XML 2004 Proceedings by SchemaSof., www.idealliance.org/proceedings/xml04/papers/73/MobileWebServices.pdf.
[53] Security algorithms, http://www.rsasecurity.com/.
[54] Security Assertion Markup Language (SAML) v2.0, OASIS Security Services TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security.
[55] Srirama, S. Publishing and Discovery of Mobile Web Services in Peer to Peer Networks, Proceedings of First International Workshop on Mobile Services and Personalized Environments (MSPE'06), Aachen, November 16-17, 2006, GI. pp. 99-112
[56] Srirama, S., Jarke, M. and Prinz, W. Mobile Host: A feasibility analysis of mobile Web Service provisioning, 4th Int. Workshop on Ubiquitous Mobile Information and Collaboration Systems, UMICS 2006, a CAiSE'06 workshop, June, pp. 942-953.
[57] Srirama, S., Jarke, M., and Prinz, W, Mobile Web Service Provisioning, Int. Conf. on Internet and Web Applications and Services, ICIW06, IEEE Computer Society, Feb 2006, pp. 120-125
[58] Srirama, S., Jarke, M., Prinz, W., and Pendyala, K., Security Aware Mobile Web Service Provisioning, In Shoniregan C. A. and Logvynovskiy A. (Eds.), Proceedings of the Int. Conference for Internet Technology and Secured Transactions, ICITST'06, Sep 2006, London, UK, e-Centre for Infonomics, pp. 48-56.
[59] USCGNC, 1995. Global Positioning System Standard Positioning Service signal Specification, 2nd Edition. United States Coast Guard Navigation Center. www.navcen.uscg.gov/pubs/gps/sigspec/gpssps1.pdf
[60] Web Services Activity, http://www.w3.org/2002/ws/.
[61] Wikman, J., and Racz, F. Mobile Personal Website, Mobisys2006 - The Fourth International Conference on Mobile Systems, Applications, and Services. Uppsala, Sweden, June 19-22, 2006.
[62] Yagüe M., Maña A. and López, J. A Metadata-based Access Control Model for Web Services, Internet Research, Emerald, 2005 25(1):99–116.
[63] Yin, R. K. (1994). Case Study Research - Design and Methods. 2nd ed.: Sage.