SOA Security - Secure Cross-Organizational Service Composition

Michael Menzel[1], Ivonne Thomas[1], Christian Wolter[2], Christoph Meinel[1]

[1]Hasso-Plattner-Institute,
{michael.menzel, ivonne.thomas, meinel}@hpi.uni-potsdam.de
[2]SAP Research, CEC Karlsruhe,
{christian.wolter}@sap.com

Abstract: Service-oriented Architectures (SOA) facilitate the interoperable and seamless interaction of services. The need to communicate with business partners demands a seamless integration of services across organizational boundaries. In fact, the integration and composition of services represent important aspects of a SOA to enable an increased responsiveness to changing business requirements. However, the interaction of independent organizations requires the establishment of trust across all involved business partners as a prerequisite to ensure secure interactions. In particular, the integration of scalable security solutions into SOA is highly demanded. This paper outlines approaches and open issues regarding secure service compositions and cross-organizational service invocation. Finally, new approaches are described to overcome current limitations regarding the dynamic composition of services based on semantic technologies, the specification and modeling of security requirements in business processes and the management of security policies based on trust levels.

1 Introduction

Service-oriented architectures are an abstract concept which exposes capabilities in distributed, domain-spanning environments as services[MLM$^+$06]. In general, SOA facilitates the interoperable and seamless interaction of service consumer and service provider to meet the consumer's needs by the service's capabilities. Several key aspects can be derived from this paradigm as described in [Erl05]: *Loose coupling* to reduce dependencies between services, *service contracts* to define interaction agreements, *autonomy* and *abstraction* to hide service logic, *reusability* and *composability* of services, *statelessness* to minimize the information specific to an activity, and *discoverability* to enable visibility of services.

The SOA paradigm provides a vast amount of flexibility in the way complex software systems are implemented. Especially in terms of an enterprise SOA, composability and reusability of services are the important concepts enabling the mapping of capabilities exposed as services to abstract activities in complex business processes that can be rearranged in an easy way at any time. Furthermore, the cooperation with business partners demands the utilization of capabilities across organizational boundaries. The involvement of independent trust domains constitutes the key aspect regarding security in service-oriented architectures. Collaborations requiring the integration of foreign services represent a considerable security threat.

The important question to address is: How can security be assured in such an unsteady environment while preserving scalability and flexibility? In traditional software systems, authentication and authorization are performed in a relatively fixed manner with a dedicated registration and authentication process which was chosen at the time of design. This is not the case in SOA anymore. The exchange of simple security credentials is insufficient when multiple trust domains are involved. Each domain may have a different understanding of security attributes (such as business roles), may support different security mechanisms and may require different information for access control. In addition, users may have multiple accounts registered with different service providers.

In this paper we provide a classification of security concepts to guarantee security goals, a description of standards implementing these concepts and, finally, introduce new approaches to overcome revealed limitations concerning the secure composition of services. Our solutions are based on modeling concepts, semantic technologies and trust levels to express, manage and negotiate security requirements in a technology-independent way.

The rest of this paper is organized as follows. In Section 2 we introduce basic concepts to guarantee the security goals confidentiality, integrity, authentication, and authorization. A classification of approaches to implement access control in SOA is presented as well. Section 3 presents various security mechanisms, which were developed or adapted to address these new security requirements and concepts. Afterwards, in section 4 we will discuss open issues regarding secure service compositions, which are in the focus of our research group, and propose solutions for selected problems. The last Section concludes this paper.

2 Classifying Security Solutions for SOA

The abstract concept of security can be defined precisely by specifying a set of security goals [PP02]. In this chapter we will present security concepts regarding the characteristics of SOA stated above. Due to space limitations, the concepts introduced in this section are related to the security goals *Authorization*, *Authentication*, *Integrity*, and *Confidentiality*. In general, we can distinguish the concepts related to confidentiality and integrity in SOA from those realizing authentication and authorization. Confidentiality and integrity provide protection of stored, processed, or transferred information in terms of properness and secrecy, while authentication and authorization are related to a digital identity regarding the establishment of trust and granting permissions to identities.

2.1 Protecting stored, processed, or transferred information

Traditional security solutions enabling a secure communication regarding confidentiality and integrity - such as SSL - provide transport security by creating a secure pipe between two hosts. Since security mechanisms are just related to the secure pipe, these solutions are not sufficient to secure information permanently. Since messages can be passed through several intermediaries in a SOA based on document exchange (i.e. Web Services using SOAP), mechanisms are required that are applied to the message itself to preserve this security information. This facilitates compliance and enables that some parts of a message, which are important to the involved intermediaries, can be kept visible. Further, different security mechanisms can be applied to different message types.

However, enhanced flexibility provided by message-based security comes also along with increased complexity, since different security mechanisms may be required by service consumer and service provider. Security requirements of services regarding confidentiality and integrity have to be described by security policies and negotiated with the service consumer.

2.2 Authenticating and authorizing a digital identity

A digital identity consists of several personal attributes with different privacy requirements that unambiguously represent a related subject. The process of authenticating the subject's identity information establishes a trust relationship between a subject and a party that relies on claims stated by the subject. Authorization concerns the determination of rights granted to the subject based on the quality of the trust relationship and attributes that are related to the subject's identity.

Security solutions that facilitate a trusted service invocation in SOA can be categorized in three groups based on the distribution of authentication and authorization information [MWM07]. For each category we present a short description along with some examples.

2.2.1 Service Managed Policies

Approaches based on *Service Managed Policies* enable the service to store and handle all information for access control. The identity of the service requester and its role are usually the most important aspects to grant access. Since all this information needs to be maintained for each user who is allowed to access the service, an initial registration of users is required to create a new account in a particular trust domain (cf. Figure 1). However, this approach requires the user to maintain different accounts and to reauthenticate when he tries to access a service in another domain. Moreover, the user has to adopt the authentication method specified by the service provider. The interaction between user and service provider will fail if different security infrastructures are used, probably supporting incompatible ways for authentication.

For example, security solutions in this category may implement identity-based access control based on a public key infrastructure (PKI). Infrastructure components are linked to keystores containing the certificates of either authorized users or the issuing certificate authority. Although a basic secure cross-domain invocation of Web Services is enabled by using a PKI, the general problem remains that such a trust domain cannot interact with a

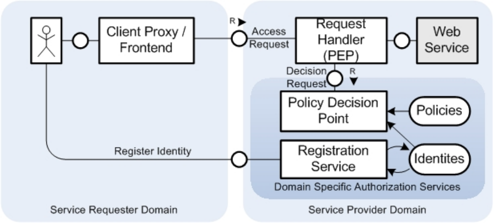

Figure 1: Service Managed Policies

domain that is based on another security solution, such as Kerberos.

2.2.2 Equal Sharing of Policy Information

Equal sharing means that the policy information is maintained by the client along with the service provider. This can be realized based on direct policy exchange, a central federation policy repository, or dedicated authentication/authorization services (cf. Figure 2). This approach simplifies administrative aspects and represents a common way to realize security solutions in collaborations. Moreover it constitutes the traditional way to implement single sign-on based on a central database.

Nevertheless, the establishment of collaboration to enable cross-domain service interaction is complicated due to the necessity to adopt the central security settings for each local infrastructure. Moreover, domain-specific individual security requirements are hard to be supported by this approach.

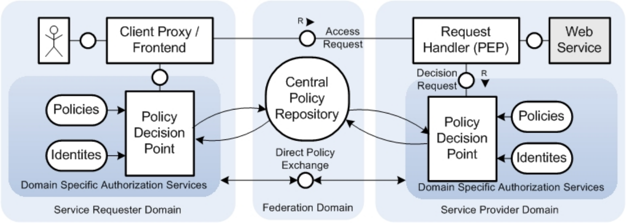

Figure 2: Equal Shared Policies

2.2.3 User Managed Policies

In the context of *User Managed Policies* the identity of the service users - and therefore the authentication policies - are solely managed and known in the user domain. The service provider may store some local policies necessary on the provider's side to define requirements for access control, but there are no cross-domain policies used - the policies of all trust domains are restricted to the respective security domain.

This approach is based on an identity federation. The key concept in an identity federation is the brokering of trust whereby all parties in a federation are willing to rely on assertions representing claims about a digital identity. For instance, these claims can represent authentication/authorization decisions to implement single sign-on, can state permissions such as 'the user is allowed to perform orders that are limited to 10.000 Euro' or additional information such as the authentication context.

Trust is usually stipulated by contracts specifying the business relationships and technically realized using security tokens that contain the assertions. Dedicated components (*Identity Providers*) in a federation are able to assert identity attributes that can be promoted to service providers acting as *Relying Parties (RP)*.

A service will grant access based on this asserted user information if the asserting authority in the user domain is trustworthy. Since a trustworthy communication is enabled although the user is unknown to the service, this approach provides scalability and flexibility that is needed in a SOA composing independent services. Furthermore, each domain is able to utilize an own security model independently from others.

Although, this approach decouples the security infrastructure used in the different trust domains, a common understanding of the exchanged attributes is still required. For example, the involved organisations may have a different understanding of roles and identities. This requires mapping mechanisms to translate these attributes.

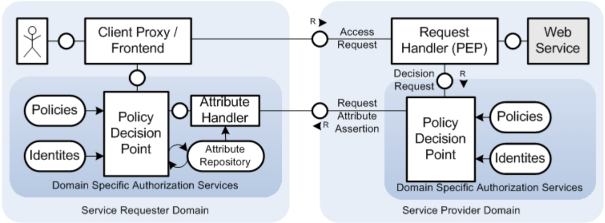

Figure 3: Client Managed Policies

3 Security Solutions in SOA

This chapter presents a selection of standards and mechanisms to secure a SOA based on the concepts which were introduced in the previous section.

3.1 WS-Security Standard

WS-Security has been proposed as a standard by Microsoft and IBM [IBM02] in 2002 and was established as an OASIS standard in 2004. This standard defines enhancements to SOAP-messaging in order to provide security to the messages transmitted between a consumer and a provider. For this purpose, additional information is included in the header of a SOAP message based on further specifications such as XMLEncryption for message confidentiality, XMLSignature for message integrity and many more. WS-Security is used to apply a wide range of different security technologies and models such as X.509 certificate and Kerberos.

3.2 Security Tokens

As described in the previous chapter, security tokens are an important concept to build trust on a technical layer by sending security credentials encapsulated in a special structure to the other party. WS-Security, as one way to define security tokens, supports for example the following types: an unsigned token (UsernameToken) to pass information like user name and password and a signed token (BinarySecurityToken) that has been endorsed by a third party, such as X.509 certificates or Kerberos. These security tokens can be used by a service to perform the authentication or authorization.

Apart from the WS-Security specification, the Security Assertion Markup Language is another standard – specified by OASIS – to describe security tokens [RHPM06]. With SAML, assertions about the authentication, authorization or attributes of a user can be stated and exchanged between service consumer and provider domain. In order to request security token and to exchange them between services, WS-Trust [NGG$^+$07] can be used.

3.3 Communicating policies

In a Web Service environment the standard way to expose service capabilities is the utilization of WSDL. However, the requirements of a service are described and communicated by security policies to enable a service consumer to determine, which security tokens he requires to access a service. In April 2006, the WS-Policy specification [DLGea05] has been submitted to the W3C as a proposed standard. This proposal describes an extensible and flexible grammar for expressing the general characteristics, capabilities and requirements of entities as policies in an XML Web Service system. By this specification, a base set of constructs is defined which can be used by other web wervice standards to describe a variety of service requirements. Another policy specification is XACML, which is already an OASIS standard. The latest Version, XACML 2.0, has been accepted as an OASIS standard in February 2005. In contrast to WS Policy, XACML is not a general-purpose policy language. It is focused on access control and authorization and specifies the architecture to enforce these policies as well. A policy in XACML is a set of rules containing a boolean expression that can be used to determine who is allowed to perform an action on a resource.

3.4 Solutions for Web Service Federation

As described in 2.2.3, a federation between Web Service consumer and Web Service provider is necessary to perform the authentication process on the user's side. Several implementations and standards for Web Service federation exist, but the two major approaches are WS-Federation and Liberty Alliance.

3.4.1 WS-Federation

The *Web Service Federation language (WS-Federation)* [NKea06] defines a framework to federate independent trust domains by leveraging WS-* Standards such as WS-Security [NKMHB06], WS-Trust [NGG+07] and WS-SecureConversation [GNea05]. This specification provides a model for security token exchange to enable the brokering of identities, the discovery and retrieval of attributes, and the provision of security claims in a Web Service based architecture. The token exchange is based on generic Secure Token Services using WS-Trust. A meta-data model to describe and establish a federation is introduced as well [GHN+ay]. Altogether, WS-Federation is designed to enable the use of identity attributes across trust domains to facilitate authorization decisions specified by WS-Policy.

3.4.2 Liberty Alliance

Liberty Alliance provides specifications for federated network identity management that is not just limited to Web Services. This project has been supported by a broad range of companies (Sun Microsystems, Novell, Intel, Oracle, ...) acting in different business areas.

The specification defines a basic framework for federation including protocols, bindings and profiles to enable account federation and cross-domain authentication based on SAML 1.0 (specified in Liberty Identity Federation Foundation (ID-FF)). In addition, bindings for Web Service Federation are defined (Liberty Identity Web Service Framework (ID-WSF)) and a set of standard services (Liberty Identity Service Interface Specifications (IS-SIS)).

In contrast to WS-Federation that can be used to exchange any type of security token, Liberty Alliance is totally based on SAML. However, this federation specification has been merged in SAML 2.0.

4 Challenges of Service Compositions in SOA

In the previous section, several standards to enable a secure federation of Web Services have been introduced. However, the application of these standards in terms of service compositions is still challenging regarding the generation, verification and negotiation of security policies. The application differs whether service compositions are deployed in cross-organisational scenarios or not.

4.1 Organizational Service Compositions

Service compositions in terms of business process modeling represents a cornerstone of process-aware information systems. Process modeling notations would provide a suitable abstract perspective to specific security goals on a more accessible level, but current notations do not support the specification of security goals at the business process level. Our research is focused on a model-driven approach addressing the difficulties to manage security mechanisms and their seamless integration into process-aware information systems by providing an abstract security goal specification, see Figure 4. This specification is translated to security policies that are deployed to provide security at the service level as well as at the business level. As a result, the security goal specification would be consistent with the affected business processes and the used security configuration as has been shown in [WS07].

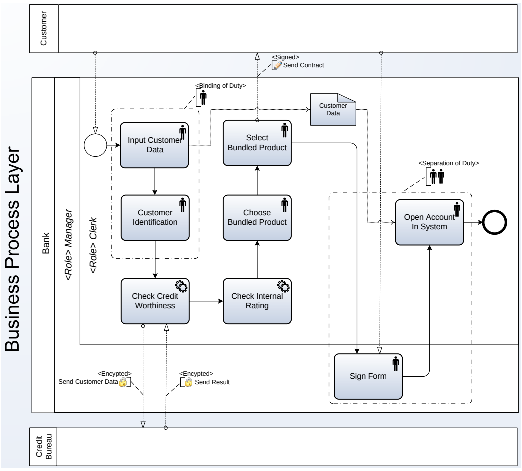

Figure 4: Modeling business goals in business processes

4.2 Cross-organizational Service Compositions

Although the generation and distribution of policy information is feasible within a single organisation, it will fail in a federation comprising multiple organisations due to the need to exchange policy information. Each organization in the federation may use its own security mechanisms or requires specific information for authorization. Therefore, services in a particular organization may have its own security requirements expressed as security policies in a specific language. These services may be mapped to an abstract activity in a service composition, as shown in Figure 5. Since service compositions are exposed as a service to the users, the security requirements of the composite service depends on the security policies of the basic services.

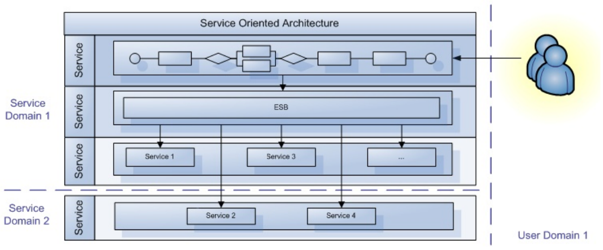

Figure 5: Layers in a Service Oriented Architecture

The federation frameworks introduced in the previous section support this scenario by allowing services to negotiate and resolve needed attributes at runtime. However, problems will arise if not all needed attributes can be resolved, e.g. due to privacy requirements. Another reason that causes a negotiation to fail is that the services located in one trust domain have no relationship to the client in another domain and no possibility to resolve attributes at all. Dynamic service compositions may be an additional reason that requires the determination of security preconditions in advance to enable a proper matchmaking.

Our research is focused on the prior calculation, verification and negotiation of the workflow's security requirements. A simulation environment should ensure in advance that a process can be executed successfully. This requires the determination of security preconditions defined by basic services. Security preconditions describe the security mechanisms that must be supported in order to invoke a service, the required security tokens and claims that must be provided comprising several attributes. Therefore, a security ontology is needed to describe security information and their relationships.

Using a formal workflow model based on petri nets - as described in research work about the calculation of preconditions in semantic workflows[Mey07] - the security requirements of the composite service can be determined.

4.3 Security Ontology

As aforementioned, our approaches concerning the modeling of security goals in organizational business processes and security verification of cross-organizational composite services require a security ontology to express the security preconditions of services and the relationship among these requirements. Several approaches have been described to define security in semantic web and Web Services, but this work is based on simple security annotations for services. We introduce a security model in this section that describes SOA-related security aspects including the relationship to policy definitions and security goals.

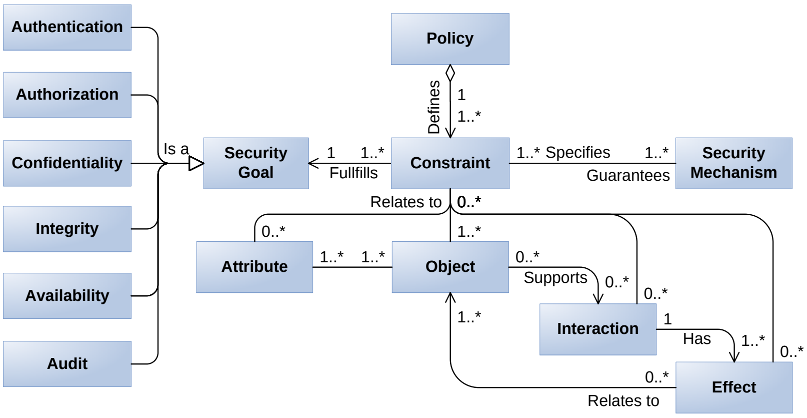

Figure 6: Security Policy Model

As shown in Figure 6, a security policy is composed of constraints that typically describe the relationship among security goals and affected entities. The basic entity in such a model is an *Object*. We define an object as an entity that is capable to participate in an *Interaction* with other objects. This interaction always leads to an *Effect*, which can comprise the provision of information or the change of state in a system. For example, one object could be a client application and another object could be a resource, such as a database. The process of accessing this database would be the interaction resulting in the effect that data in the database is changed or information is returned to the application.

Each object is related to a set of attributes describing its meta information. For instance, if the object represents a subject, attributes that constitute the digital identity will be related. Altogether, policy constraints always refer to a set of objects, a particular set of objects' attributes, and optionally a set of interactions and effects that are related to the objects. Based on these relations, specific constraints for particular security goals can be derived. These specific constraints define requirements for associations between the entities with regard to the particular security goals.

As shown in Figure 6, constraints specify security mechanisms that guarantee the defined

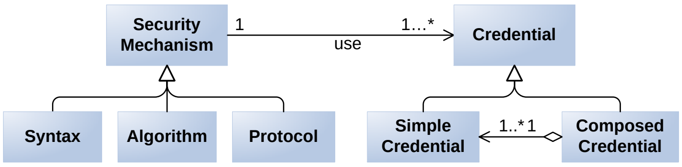

Figure 7: Security Mechanisms Model

constraint. For instance, a confidentiality policy usually specifies an algorithm (e.g. DES) that must be used to guarantee this requirement. In our model a *Security Mechanism* is designed to characterise techniques that are used to enforce security constraints, see Figure 7. In general, these mechanisms can be classified as algorithms (e.g. DES), protocols (e.g. WS-Security) or syntax (e.g. XML). Besides security mechanisms, a *Credential* represents another important entity in our model that subsumes evidences used by security mechanisms. A detailed classification of security credentials was presented by Denker et al. [DKF^{+}03]. In this work they introduced an ontology that divides credentials in simple credentials (e.g. key, login, certificate) and composed credentials (e.g. Smart Card, SAML, WS-Security Token) that contain a set of simple credentials.

The strength of our model is a general description of security goals, omitting technical details. Thus, the provided models can be mapped to an arbitrary application or technical specification. For example, we mapped our models to the technical specification of the WS-Security standard described in the previous section. Besides the potential mapping to a technical implementation, we addressed the issue of security goal specification. In particular the projection to general business process models allowing to directly specify security goals in the context of business processes. It has been revealed, that just two basic ontologies are needed to describe security preconditions in an SOA - *security mechanisms* and *security credentials*.

4.4 Managing and Negotiating Policies

As described in the previous section, several specifications such as WS-Policy or the XACML language specification can be used to express constraints and therefore security requirements. However, as stated earlier, service-oriented architectures can involve computers with all kinds of security infrastructures. Especially in a cross-organizational service composition services must be capable to rely on a large set of different security tokens and claims. This makes it very hard to list all the requirements for a service, since the access control decision can depend on many attributes of these architectures. If we consider for instance the authentication process, there are many attributes on which the access control decision can depend. In particular, it is not sufficient to consider the authentication method which was used to authenticate a user, but the whole number of attributes, which describe such an authentication as for example the length of a password, whether it was auto-generated or user-chosen, the encryption method used to transmit and store such a password, etc. Regarding all these attributes, a password-based authentication performed by one service can be much stronger or weaker than the one performed by another service i.e. due to different encryption methods used. This means the quality of the same aspect can differ tremendously between services. And this is not only true for the authentication aspect, but for all aspects, which characterize such a service and which are part of the access control decision. Consequently, it is necessary to state the attributes which characterize such differences in the security policies of the services. This however makes policy definition as well as the negotiation of such attributes very cumbersome. Furthermore, the policy structure reaches a complexity, which is very hard to handle by a human being and may therefore lead to policy inconsistencies and to the very end to the disruption of the related business process execution.

Therefore, mechanism are required to simplify the policy structure. One possible solution that we propose is to use a quantitative model, which expresses the expected security for an aspect and the achieved security for this aspect by a numerical value. For this purpose a numerical value is assigned to each aspect, as for example to each authentication mechanism which represents the confidence in this mechanism, which is determined by the service provider. This way, the attributes which characterize a certain security aspect are summed up in a numerical value indicating the trust, which has to be reached by a certain attribute combination. In order to calculate a trust value, classical probability theory is used.

5 Conclusion

This paper has pointed out which new security paradigms need to be applied in order to bring service-oriented architectures to their full potential. In an environment, in which services can be composed across organizational borders, security concepts are required which can deal with the unsteady and flexible nature of service-oriented architectures. Especially in a federated environment, that comprises multiple independent trust domains, the establishment of trust and the provision of identity information has been revealed as the key aspect to secure information, services, and interactions.

Several solutions for federated identity management have been introduced that are designed to be used with Web Services. However, it is challenging to apply these solutions in SOA since current approaches neither consider security aspects at the business process level nor enable a seamless composition of services that have different security requirements. We revealed that the design of service compositions under security constraints and the enabling of automatic service compositions require a generic security model.

A model has been introduced that specifies security goals, policies, and constraints based on a set of basic entities. The strength of our model is that these entities can be mapped to an arbitrary application domain and all layers in an SOA. This model constitutes the foundation to express security aspects at the business process level and provides an ontology to calculate the security preconditions of a workflow, which can be used for policy negotiation with clients from other trust domains. Finally, trust levels have been introduced to simplify the definition of policies that express service requirements.

References

[DKF$^+$03] Grit Denker, Lalana Kagal, Timothy W. Finin, Massimo Paolucci, and Katia P. Sycara. Security for DAML Web Services: Annotation and Matchmaking. In *International Semantic Web Conference*, pages 335–350, 2003.

[DLGea05] Giovanni Della-Libera, Martin Gudgin, and et all. Web Services Security Policy Language (WS-SecurityPolicy). Public Draft Specification, Juli 2005.

[Erl05] Thomas Erl. *Service-Oriented Architecture : Concepts, Technology, and Design*. Prentice Hall PTR, August 2005.

[GHN$^+$ay] Marc Goodner, Maryann Hondo, Anthony Nadalin, Michael McIntosh, and Don Schmidt. *Understanding WS-Federation*. Microsoft and IBM, 2007 May.

[GNea05] Martin Gudgin, Anthony Nadalin, and et al. Web Services Secure Conversation Language. public draft Specification, February 2005.

[Mey07] Harald Meyer. On the Semantics of Service Compositions. In *Proceedings of The First International Conference on Web Reasoning and Rule Systems (RR 2007)*, 2007.

[MLM$^+$06] Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter Brown, and Rebekah Metz. Reference Model for Service Oriented Architecture 1.0. OASIS Committee Specification, February 2006.

[MWM07] Michael Menzel, Christian Wolter, and Christoph Meinel. Access Control for Cross-Organisational Web Service Composition. In *Proceedings of the International Multiconference on Computer Science and Information Technology*, volume 2, pages 701–711, 2007.

[NGG$^+$07] Anthony Nadalin, Marc Goodner, Martin Gudgin, Abbie Barbir, and Hans Granqvist. WS-Trust 1.3. OASIS Standard, March 2007.

[NKea06] Anthony Nadalin, Chris Kaler, and et al. Web Services Federation Language (WS-Federation) V 1.1. Specification public draft, December 2006.

[NKMHB06] Anthony Nadalin, Chris Kaler, Ronald Monzillo, and Phillip Hallam-Baker. Web Services Security: SOAP Message Security 1.1. OASIS Standard Specification, February 2006.

[PP02] Charles P. Pfleeger and Shari Lawrence Pfleeger. *Security in Computing*. Prentice Hall Professional Technical Reference, 2002.

[RHPM06] Nick Ragouzis, John Hughes, Rob Philpott, and Eve Maler. Security Assertion Markup Language (SAML) V2.0 Technical Overview, 2006.

[WS07] Christian Wolter and Andreas Schaad. Modeling of Task-Based Authorization Constraints in BPMN. In *BPM*, pages 64–79, 2007.