A SECURITY ARCHITECTURE FOR INTER-ORGANIZATIONAL WORKFLOWS

Putting Security Standards for Web Services together

Michael Hafner, Ruth Breu, Michael Breu
Institute for Computer Science, University Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

Keywords: Web Services, Security, Model driven security architecture, eGovernment, XACML

Abstract: Modern eBusiness processes are spanning over a set of public authorities and private corporations. Those processes require high security principles, rooted on open standards. The SECTINO project follows the paradigm of model driven security architecture: High level business-oriented security requirements for inter-organizational workflows are translated into a configuration for a standards based target architecture. The target architecture encapsulates a set of core web services, links them via a workflow engine, and guards them by imposing specified security policies.

1 INTRODUCTION

The reliable implementation of security requirements is crucial in the development of trustworthy B2B, B2C or eGovernment-solutions. Security issues must be carefully planned and addressed during the development process, because they are not isolated aspects, but relevant in all phases of the development.

Modern B2B or eGovernment-applications use a web service centric architecture to communicate with each other. Web services standards, based on SOAP, WSDL, and UDDI, have facilitated the integration of B2B application upon platform independent components and tools. While SOAP (Mitra, 2003) provides the underlying mechanisms for standardized exchange of messages between senders, receivers and intermediaries, the security problems are not (yet) addressed. To this end OASIS developed an extension (Nadalin et al, 2004) of SOAP that defines mechanisms to add security information such as digital signatures and encryption to the standard.

However the handling of such secure SOAP Messages is quite complex, low-level and error prone. More abstract methods and tools are needed to disclose this complexity from the application developer.

In Breu et al. (2004a, 2004b, 2004c) we presented an approach that extends the concepts of model driven architecture (MDA) to a model driven security architecture. A set of models was proposed to model workflow and associated security aspects on an abstract level. These models can be classified by two orthogonal views: The *interface view* specifies the interfaces of the services an individual partner provides. The *workflow view* describes the orchestration of the rendered services. The workflow is described on the global level as a workflow of cooperating partners and on the local level to describes the behaviour of each partner's node. These models are specified using UML models (mainly by class and activity diagrams).

The UML models are extended by specific stereotypes to enable the specification of security requirements, such as confidentiality, integrity, or non repudiation.

The guiding idea of the SECTINO project is to link these models to the configuration of a predefined target architecture. The target architecture defines the structure, the components and the services, needed to build an implementation on. The functionality is then generated from the UML-based specification.

We will concentrate here on the security aspects. The workflow aspects will be defined in a separate document. For an overview see (Breu et al., 2004a).

The document is organized as follows: To make this paper self-contained, the next chapter gives an overview of the techniques and artefacts for modelling inter-organizational workflows and security requirements. We do this by presenting a running example taken from an ongoing case study taken from the field of eGovernment. Then we give an overview to security standards for web services

Hafner M., Breu R. and Breu M. (2005).
A SECURITY ARCHITECTURE FOR INTER- ORGANIZATIONAL WORKFLOWS - Putting Security Standards for Web Services together.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems - DISI*, pages 128-135

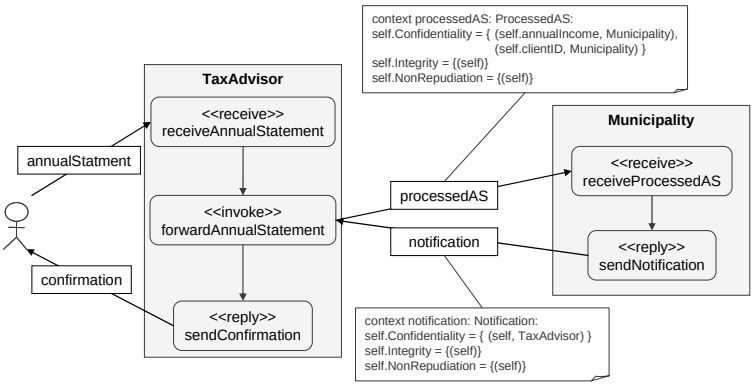

Figure 1: A Sample Document Flow with Security Requirements (Global Workflow Model)

and related technologies. In the following two sections we show how the target architecture, developed in the SECTINO project, is structured and the standards are employed. Finally we compare the chosen architecture with other approaches.

2 MODEL DRIVEN SECURITY FOR INTER-ORGANIZATIONAL WORKFLOWS

This section briefly sketches our approach for the systematic design and realization of security-critical inter-organizational workflows.

2.1 Case Study

Our research efforts are based on a real life use case that was elaborated within the project SECTINO, a joint research effort between the research group of Quality Engineering at the University of Innsbruck and the Austrian Research Center Seibersdorf, and is based on a case study involving a major Austrian municipality.

Our methodology for the systematic design and realization of security-critical inter-organizational workflows is illustrated by a portion of a workflow drawn from the use case "Processing of an Annual Statement" which describes the interaction between a business agent (the Tax Advisor) and a public service provider (the Municipality).

In Austria, all wages paid to employees of an enterprise are subject to the municipal tax. Corporations have to send the annual tax statement via their tax advisor to the municipality which is responsible for collecting the tax by the end of March of the following year. The municipality checks the declaration of the annual statement and calculates the tax duties. As a result, a notification with the amount of tax duties is sent to the tax advisor by mail.

One of the project goals is to analyze security issues that may stem from the migration of the workflow to an e-government based solution and create the necessary run-time artefacts for the target architecture through model transformation. Ultimately, the workflow should allow the declaration of the municipal tax via the internet.

2.2 Security Engineering

The development of a security-critical inter-organizational workflow starts with the analysis and the design of the workflow, followed by a risk and threats analysis, and the security requirements specification. Security requirements are then modelled in a platform-independent way at different levels of abstraction.

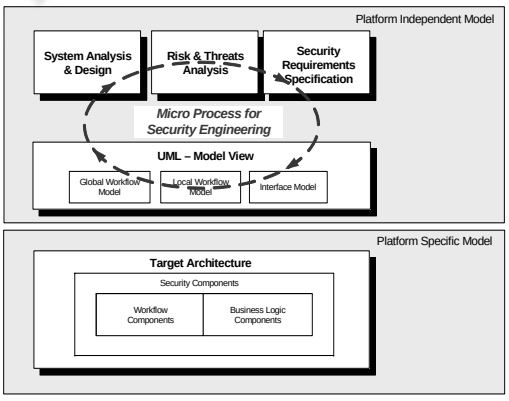

Figure 2: Model Driven Security for Inter-Organizational Workflows

These steps are executed iteratively, following a five step approach for security analysis – also called Micro Process for Security Engineering (Breu et al. 2004a). The artefacts produced in this process range from informal textual descriptions (e.g. covering legal requirements) until graphical and formal

descriptions that can be transformed into runtime artefacts for the target architecture.

For a detailed description of the approach for the management of security related aspects within the development process (figure 2) please refer to (Breu et al. 2004a, 2004b).

2.3 Model Views

In the context of this paper a *workflow* describes a network of partners cooperating in a controlled way by calling services and exchanging documents. Our method of designing security-critical inter-organizational workflows is based on two orthogonal views: the *interface view* and the *workflow view*. The workflow view is further divided into the *global workflow model* describing the message exchange between cooperating partners, and the *local workflow model* describing the behaviour of each partner.

In our approach workflows are specified by class and activity diagrams as defined in (Amsden, et al., 2004). We mainly use UML class diagrams for modelling the structure of exchanged documents (as XML messages) and UML activity diagrams to describe the flow of service calls. In this paper we only sketch the diagrams that are relevant to security modelling.

The global workflow describes the interaction of partners abstracting from internal processing steps and does not contain any connection to the business logic. Activities are qualified by BPEL stereotypes (Andrew et al., 2003). Security requirements in the global workflow model refer to the exchange of (XML) data among the partners involved. In the current version we consider the confidentiality and integrity of the messages (leading to the encryption and signing of the data exchanged) and non-repudiation of the message exchange. Both confidentiality and integrity may refer to the whole message or only to parts of it.

Figure 1 depicts as example security requirements a qualification of the document exchange. The security requirements *Integrity* and *Non-Repudiation* are assigned a set of document nodes in the form of an OCL navigation expression. The requirement of *Confidentiality* is assigned one or more pairs - consisting of document nodes and optionally actor roles. The latter implicitly carries information about permissions to view the information, as the security gateway signs the node with the corresponding public key of the referenced actor.

The local workflow models define the portion of the global workflow each partner is responsible for and therefore they are designed for each partner type. The local workflow corresponding to a swim lane in the global model is an executable process description that considers service calls from the outside, and contains internal actions as well as connections to the business logic. It is a direct input for a local workflow management system and is typically developed internally by partners.

The interface model describes a component offering a set of services with given properties and permissions. Usually, partners map the interfaces of the operations of their local business logic to web services operations in the interface model. Internal interfaces remain hidden to the other partners. Thus, the interface model of every partner's node describes the public part of the local application logic, which is accessible to the inter-organizational workflow and conforms to a uniform technical, syntactical and semantic specification the partners agreed upon - information typically published in WSDL files and technical Models (tModels) of UDDI registries.

Security requirements at this level of abstraction involve the support of a role model and the specification of access rights for particular web service operations. We describe access rights formally and platform-independently using an OCL dialect, a predicative sublanguage of UML (OMG, 2004). The predicative specification is then transformed into an XACML-policy file (see section 4.3) via automatic generation. A more detailed description of the interface model can be found in Breu et al. 2004a.

The application of these orthogonal perspectives allows us to combine the design of components offering services that may be called in different contexts by different partners with the design of workflows that focus on particular usage scenarios. Please refer to Breu et al. 2004b for a detailed description of the model dependencies.

Formal approaches based on Petri Nets (Van der Aalst, 2000) for the design of inter-organizational workflows guarantee local autonomy (at the partner level) without compromising the consistency of the global process. In our terms, this means that – through peer-to-peer interaction – the local workflows should exactly realize the behaviour as specified in the global workflow.

Security requirements specified in the global workflow model have to be mapped in a consistent way to security requirements of the local workflows of all cooperating partners, which reflect the business logic in their local environment.

All three model types together carry all the information that is needed by the Security Components in the Reference Architecture to implement the secure distributed workflow. The models are exported into XMI files and security relevant information is extracted and mapped into a table that directly configures the security components of the target architecture (see section 4.3).

3 RELATED SECURITY STANDARDS

The world of web service standards, recommendations and drafts is quite complex and has grown considerably in the last years. IBM and Microsoft have published a roadmap (2002) for the further evolution of the web service security standards, comprising mechanisms for trust, policy definition, authorization, etc. Although these specifications are still under development, we have designed a target architecture that adheres to these standards.

For a basic overview to security standards see e.g. (Gutiérez, 2004). We just introduce the most important ones, used in context of the target architecture.

The basic web services standards are SOAP (Mitra 2003) for basic message transfer and WSDL (Christensen et al., 2001) for the definition the abstract functionality of a web service.

OASIS has proposed an extension of SOAP to enable the addition of security features to Web Service Messaging (Nadalin et al., 2004): It allows the definition of (signed) security tokens in the header of SOAP messages and to digitally sign and encrypt the message content with these security tokens. The WS-Security standard in turn relies heavily on the underlying standards for signing and encryption of XML documents (Eastlake, 2002a, 2002b). There is a prototypic extension of Axis for web service security which we use in the reference implementation.

The XACML is another OASIS standard (Moses, 2004) that allows for the specification of access policies to (web) services. The standard defines a language for the formulation of policies and related queries. The XACML standard also identifies specific functionalities in the process of access control and defines an abstract data flow model between dedicated functional components. XACML is closely related to the Security Assertion Markup Language (SAML) (Mishra et al., 2004), which is an XML-based framework for exchanging security information.

We are using web services policies to achieve two different purposes. Web services endpoints have to negotiate the parameters and advertise the set of requirements potential service requesters have to comply with when consuming the service (like authorization, quality-of -service, privacy, etc.). Web services policy standards (Bajaj et al, 2004, Moses, 2004)) allow the expression of requirements for web services in their interaction with other services in a standardized way. On the other hand, these wide spread XML-based standards are well suited for a platform independent configuration of the security components. We are currently working on a declarative security model based on XACML and the WSPL-profile for the configuration of the security components of the reference architecture.

4 A SECURE TARGET ARCHITECTURE

The SECTINO target architecture is defined to provide a configurable framework to implement the requirements specified through the artefacts presented in section 2. Much care was taken to base the architecture on open web service and security standards.

In this section we first discuss the basic requirements and assumptions on which the architecture is built and then we present the components in more detail. In the following section we will show how security standards are used to interface these components.

4.1 General Requirements

We assume that the target architecture is built around a given core of atomic web services, that implement the internal steps of an overall business process. This is e.g. the view taken by Microsoft Biztalk. These core web services must apply the internal role model and are therefore guarded by internal access policies.

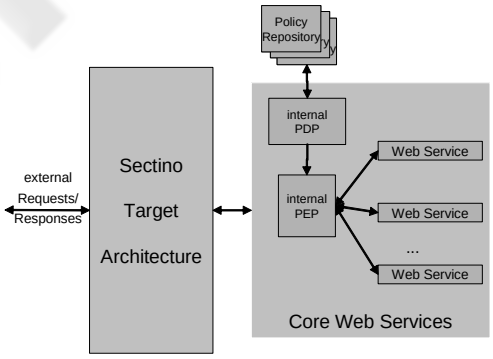

Figure 3: Access Control with XACML in the Application Core, Secured by the Target Architecture

As a backbone for our security architecture we use the workflow architecture introduced by the XACML standard. This workflow consists of a so-called Policy Enforcement Point (PEP) that acts as a gatekeeper for accessing the resources. The PEP queries a Policy Decision Point (PDP) to allow (or deny) access to web services. The PDP in turn can select the applicable policy from a Policy

Repository, decides the access query and returns either the result *permit*, *deny* or *not applicable*.

The target architecture encapsulates this internal core by a layer that realizes the interface to the external partners.

Hence the major requirements to the target architecture are

1. to provide functionality wrt. the policy enforcement for the encryption and signing of SOAP messages and interfaces for non-repudiation protocols (e.g. logging and timestamp facilities),
2. to map external roles and entities onto internal roles and entities,
3. to implement the internal workflow model on top of the atomic web services,
4. to map external web service interfaces to entry points of corresponding workflow processes that implement the interfaces on top of the atomic web services.

The first two requirements are implemented in a *Security Gateway*, the latter two requirements are implemented by a *Work Flow Engine*.

The Security Gateway has to take over two responsibilities: First it has to enforce access policies, i.e. who is allowed to access the called web service. Second it has to enforce that encryption and signing requirements are handled adequately by the partners, e.g. rejecting non-encrypted messages, signing outgoing messages or handling non-repudiation requirements.

In detail the Security Gateway has to check incoming secured requests for

- the correct signature of the message (or of parts of the message),
- the encryption of the message (or of parts of the message),
- generating return receipts if necessary to implement a non-repudiation requirement.

It thus acts in the same way as a *Policy Enforcement Point* (PEP) in the sense of the XACML standard, which passes the signature and encryption information to a *Policy Decision Point* to look up the applicable policy and to decide to grant the web service access.

Further the Security Gateway has to manipulate outgoing (yet unsecured) responses. According to the specification of the PEP Configuration the message (or parts of it) have to be signed and encrypted.

Signature and encryption used in a public environment usually need also a public key infrastructure to manage security certificates.

The Workflow Engine manages the mapping of the local workflow to the web services and the appropriate local roles. As workflow choreography language we choose BPEL (and as its implementation e.g. Biztalk, or the BPEL4WS Engine provided by IBM) . However the workflow engine will not be in the focus of this presentation.

We also assume that there are given services that provide

- access to a PKI infrastructure, in order to validate signatures, and to encrypt messages for respective recipients,
- logging functionality, i.e. a service that implements non-repudiation facilities.

We consider them as external services, because these services are typically implemented by some other service provider.

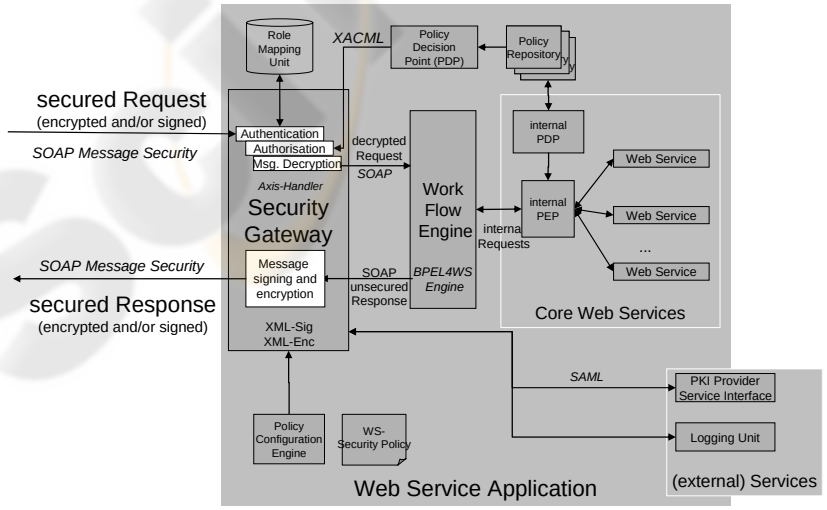

Figure 4: Reference Architecture

4.2 Components and Interfaces of the Target Architecture

Figure 4 gives an overview to the components and their interfaces. These components are explained in the following subsections.
The security gateway implements the functionality of policy enforcement, user authentication and security treatment of messages. It not only handles incoming requests and outgoing responses as illustrated in figure 4, but also outgoing requests and incoming responses initiated by the internal core web services (which was left out for simplicity)
For inbound messages handling the security gateway has to carry out the following tasks:

- Caller authentication by investigation of the provided signatures, or other credentials included in the inbound message.
- Caller Authorization: A role mapping unit maps the user internally to a set of roles. These roles define the permissions of the caller according to a given policy. To this end a PDP is queried for an appropriate policy that gives (or denies) access. Thus we follow a role based access control approach (Sandhu, et al., 1996) for permission control.
- Finally the gateway checks the compliance of the message with the security requirements as specified by the policy configuration engine. This comprises the check of required signatures of parts of the message, and the decryption.

For the handling of outbound messages the security gateway has to enforce the security requirements, as e.g. the signing and encryption of the complete outgoing message or parts of it.

4.3 Security Gateway Configuration

The global workflow model and the interface model define two types of security requirements.

Access restrictions, expressed in an OCL style, can be translated into XACML rules (Alam, et al. 2004), which can be stored in the policy repository.

Security requirements as e.g. mandatory signatures or encryption requirements have no direct correspondence in XACML. However XACML supports so-called obligations which are forwarded to the PEP as a part of the query result. We have chosen to model those security requirements as additional obligations for the security gateway.

Since XACML policies are quite flexible, but rather hard to read, we use here a more user friendly presentation of the obligations presented in table 1.

This table shows all security requirements for the security gateway at the domain boundaries of the Actor "Tax Advisor". As specified in the global workflow model (figure 1) the Tax Advisor sends a document called **`ProcessedAS`** to the Municipality qualified as partially encrypted (Node: **`processed-AS/annualIncome`**) and signed (Documentroot: **`processedAS`**) by calling the outbound operation **`sendProcessedAS`** of the interface **`AS_Service`**, which is implemented by the Municipality taking the role of **`Municipality_AS_Provider`**.

Both the permissions and the configuration file are stored in (separate) XACML policies as rules and obligations, respectively. We are currently evaluating whether the WS-PL profile for XACML (Moses, et al., 2003) is even better suited to store the requirements.

One major advantage of WS-PL is that it is planned to integrate it with WSDL-files that are distributed as the description of a web services port. Thus it would allow to share the access and security requirements between partners.

4.4 Reference Implementation

The target architecture is currently implemented by a conceptual prototype in order to study the details of the transformation process and configuration requirements. The intention was to base the architecture as near as possible on available or emerging non-proprietary security standards for web services. In figure 4 we also present the involved standards (noted in italics).

The reference implementation is based on Axis which is a basic middleware to implement web services. Axis implements an engine that provides mechanisms to define handlers for inbound and outbound messages. We use axis handlers to integrate security handling. The handlers employ XML-Sig and XML-Enc and are based on the WSS4J package that extends Axis to implement WS-Security. The involved handling of XACML requests is based on sun's XACML implementation. The workflow engine is based on a BPEL4WS Engine. Finally the PKI Provider Service Interface is based on SAML. However we need only a small subset for X.509 certificate lookup.

5 CONCLUSIONS

5.1 Results

This article outlined the SECTINO approach for model driven security and the underlying target

architecture of the SECTINO project. It is based on open standards such as WS security and XACML, which are currently starting to penetrate the market of application servers. The reference implementation shows the applicability of the approach and serves as an object to study the architecture in detail.

The guiding paradigm was to specify security requirements adequately all the way along the development process. Building on established (or at least most probably upcoming) standards is a major prerequisite when defining inter-organizational processes. However it turned out during the project that there is a universe of complimentary XML-based security standards. Also the standards themselves are quite hard to understand, and even harder to apply directly. This confirms our opinion that the model driven approach is the right way to go, in order to provide more understandable and thus more secure applications.

5.2 Other Approaches

Lodderstedt et al. 2002 originally inspired the paradigm of *model driven security architecture.* They proposed SecureUML as s special profile of UML to specify role based access control (Sandhu, et al., 1996) rules in UML with the help of OCL. They demonstrated their ideas in a J2EE prototype. Access constraints are translated into deployment descriptors and java assertion code of enterprise java beans (EJBs).

Besides access control rules our approach also integrates the specification of confidentiality, integrity and non-repudiation on UML level. Since we translate these requirements into XACML, these access and security constraints are exchangeable between partners working with different middleware architectures such as .NET or J2EE, but with the same open standard to defines security requirements.

The idea of a web service security architecture was also pursued by Vasiu and Donciulescu (2004). They propose a web service management architecture (WSMA) which covers among others the aspects of trust management in a web services landscape. The basic idea is that WSMA "creates a sort of 'bubble' around the XML Web Services managed." The requirements that are addressed, are (among others), the control of authorizations, maintenance of access rights, and the verification of client identity. The authors concentrate to demonstrate how to manage and combine hierarchical trust contexts. However the authors do not give details, how such an WSMA is structured, nor how it could be implemented. Indeed, the architecture we presented in section 4 could be considered as an implementation of a WSMA (or at least of a part of it).

The SECTINO architecture is comparable to the virtual mail office architecture, for centralized signing and encryption, proposed by the BSI (2003), since it also allows the central management of security policies. However the major focus of our architecture is to implement inter-organizational security policies.

5.3 Future Work

As presented in section 4.4 there exists up to now a first architectural prototype to study all the aspects of the proposed architecture. This prototype will be extended gradually to a fully functionally reference architecture.

In this presentation we have discussed exchanges of messages between systems as "Tax Advisor" and "Municipality", that sign and encrypt messages. However in many eGovernment processes real persons are signing (and potentially receiving encrypted) messages, as e.g. an employee of the tax advisor, or an officer in the municipality. To address this aspect the modelling has to require principles of delegation and authorization. The basic principles of our approach should remain stable, however this point needs major consideration.

A final point is the automatic transformation of security requirements in UML as they are presented in section 2.3 to a configuration as presented in section 4.3. For the time being this transformation is defined, however the ultimate goal is to provide tool support for an fully automatic generation of the

Table 1: Tax Advisor Security Gateway Configuration File

Interface	Operation	Parameters	Calling Role	SignedNodes	EncryptedNodes
TA_Service	**sendAnnualStatement**	in: AnnualStatement:**annualStatement** out: Confirmation:**confirmation**	**TA_Service_Requester**	annualStatement confirmation	annualStatement/annualIncome confirmation/taxDutiesAmount
AS_Callback	**sendNotification**	in:Notification:**notification**	**Process_AS_Provider**	notification	notification

Outbound

Interface	Called Operation	Parameters	Receiving Role	SignedNodes	EncryptedNodes
AS_Service	**sendProcessedAS**	ProcessedAS:**processedAS**	**Process_AS_Provider**	processedAS	processedAS/annualIncome

Internal WS Calls

Interface	Operations	Parameters
Internal_Proces	**checkMandate**	Result:**result**
Internal_Proces	**processASDocument**	ProcessedAS:**processedAS**
Internal_Proces	**processNotification**	Confirmation:**confirmation**

configuration artefacts for the target architecture.

REFERENCES

Alam, M., Breu, M., Breu, R. 2004. Model Driven Security for Web Services, INMIC 04, 8th International Multitopic Conference, Lahore, Pakistan

Amsden, J., Gardner, T., Griffin, C. 2004. Draft UML 1.4 profile for automated business processes with a mapping to BPEL 1.0. See http://www-128. ibm.com/developerworks/rational/library/4593.html

Andrews, T., et al. 2003. Specification: Business Process Execution Language for Web Services V. 1.1. See http://www-128.ibm.com/developerworks/library/ws-bpel/

Apache <Web Services/> Project, http://ws.apache.org/

Bajaj, S., et al. 2004. Web Services Policy Framework (WS-Policy) September 2004. See: ftp://www6.software.ibm.com/software/developer/library/ws-policy.pdf

Breu, R., Hafner, M., Weber, B., Alam, M. Breu, M. 2004a. Towards Model Driven Security of Inter-Organizational Workflows. In: Proceedings of the Workshops on Specification and Automated Processing of Security Requirements (SAPS2004), pp. 255-267.

Breu, R., Hafner, M., Weber, B., Novak, A. 2004b. Model Driven Security for Inter-Organizational Workflows in e-Government. TED Conference on e-Government, Bozen, 2005.

Breu, R., Hafner, M., Weber, B. 2004c. Modeling and Realizing Security-Critical Inter-Organizational Workflows. In: W. Dosch, N. Debnath (Eds.), Proceedings IASSE 2004, ISCA, ISBN 1-880843-52-X.

BSI, Bundesamt für Sicherheit in der Informationstechnik, Fachkonzept für die virtuelle Poststelle, 30. May 2003, See: http://www.bsi.de/fachthem/egov/download/6_VPS_FKP.pdf

Christensen, E., Curbera, F., Meredith, G., Weerawarana, S. 2001. Web Services Description Language (WSDL) 1.1. See: http://www.w3.org/TR/wsdl

Eastlake, D. (ed.), et al. 2002a. XML-Signature Syntax and Processing. W3C Recommendation 12 February 2002. See: http://www.w3.org/TR/xmldsig-core/

Eastlake, D. (ed.), et al. 2002b. XML Encryption Syntax and Processing. W3C Recommendation 10 December 2002. See: http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/

E-Government Gesetz der Bundesrepublik Österreich, See http://www.parlament.gv.at/portal/page? pageid=908, 145843&_dad=portal&_schema=PORTAL

Gutiérez, C., Fernández-Medina, E., Piattini, M. 2004. Web Service Security: *is the Problem solved?* In Proceedings of the 2nd International Workshop on Security In Information Systems, WOSIS 2004, In conjunction with ICEIS 2004, Porto, Portugal.

IBM and Microsoft. 2002. Security in a Web Services World: A Proposed Architecture and Roadmap. *A joint security whitepaper from IBM Corporation and Microsoft Corporation. April 7, 2002, Version 1.0.* See: http://www-106.ibm.com/developerworks/webservices/library/ws-secmap/

Lodderstedt, T., Basin, D., Doser, J. 2002. SecureUML: A UML-Based Modeling Language for Model-Driven Security, in LNCS 2460, Jezequel, J.M.; Hussman, H.; Cook, S. (eds.) Proceedings of the 5th International Conference on The Unified Modeling Language, pp. 426-441.

Microsoft, Biztalk, http://www.microsoft.com/biztalk/

Mishra, P. (ed.), et al. 2004. Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0 Committee Draft 02, 24 September 2. See: http://www.oasis-open.org/committees/download.php/9452/sstc-saml-conformance-2.0-cd-02.pdf

Mitra, N., 2003. SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation 24 June 2003. See http://www.w3.org/TR/2003/REC-soap12-part1-20030624

Moses, T. (ed.), et. al. 2003. XACML Profile for Web-Services. XACML TC Working draft, Version 04. September 29, 2003. See: http://www.oasis-open.org/committees/download.php/3661/draft-xacml-wspl-04-1.pdf

Moses, T. (ed.). 2004. eXtensible Access Control Markup Language (XACML) Version 2.0. Committee draft 02, 30 Sep 2004. See: http://docs.oasis-open.org/xacml/access_control-xacml-2_0-core-spec-cd-02.pdf

Nadalin, A., Kaler, C., Hallam-Baker, P, Monzillo, R., 2004. Web Services Security: SOAP Message Security 1.0 (WS Security 2004), OASIS Standard 200401, March 2004. See http://docs.oasis-open.org/wss/-2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

Apache WSS4J. See: http://ws.apache.org/ws-fx/wss4j/

OMG. 2004. UML 2.0 OCL Specification. See: http://www.omg.org/docs/ptc/03-10-14.pdf

Sandhu, E.S., Coyne, E.J., Feinstein, H.L., Youman, C.E. 1996. Role-based access control models, IEEE Computer, 29(2):38-47.

Van der Aalst, W.M.P. 2000. Loosely Coupled Inter-organizational Workflows: Modeling and Analyzing Workflows Crossing Organizational Boundaries. In: Information and Management 37 (2000) 2, pp. 67-75.

Vasiu, L., Donciulescu, C. 2004. A Requirement for a XML Web Services Security Architecture, in ICEIS 2004, Proceedings of the 6th International Conference on Enterprise Information Systems, Porto.