Security Requirements for a Semantic Service-oriented Architecture

Stefan Dürbeck, Rolf Schillinger, Jan Kolter
Department of Information Systems
University of Regensburg
D-93040 Regensburg, Germany
{stefan.duerbeck, rolf.schillinger, jan.kolter}@wiwi.uni-regensburg.de

Abstract

Service-oriented architectures (SOAs) are a commonly used paradigm for IT infrastructures in various fields. Due to their flexibility and the easy accessibility of their underlying web services, SOAs are the architecture of choice for more and more service providers. Semantic SOAs (SSOAs) are going one step further and are enhancing the common SOA with semantic components.

However, a major success criterion of any SOA is the existence of a reliant security infrastructure. Therefore, this paper identifies security requirements for an eGovernment SSOA focusing on communication security, trust, privacy and access control. Our work is based on the architecture designed within the scope of the European research project Access-eGov, which envisions the development of a SSOA-based eGovernment platform.

1 Introduction

Service-oriented architectures (SOAs) are increasingly gaining popularity in large-scale computer networks. More and more IT-providers wrap their applications into web services, thereby enabling them to be easily registered, discovered and integrated. Over the last few years this trend did not only occur in profit-oriented IT companies but also to a large extent in the field of eGovernment. Tight budgets and pressure to save costs are only two reasons why many public agencies offer their services electronically. On the other hand, citizens also benefit from this option, since they can now access a requested service at any time and from almost any place.

Besides the benefits a SOA offers, a major drawback is the lack of unique standardized vocabularies. As the source of service providers is very diverse, they tend to use different vocabularies when registering a web service. To overcome this issue, ideas from Tim Berners-Lee’s vision of the Semantic Web [3] can be employed. Enhanced with semantic components a SOA can perform the semantic matching of user queries and web services annotated with different ontologies. For such a Semantic SOA (SSOA) this involves the mapping of different ontologies.

Targeting eGovernment and the vision of SSOAs, the goal of the European research project Access-eGov\(^1\) is to build a trans-national eGovernment platform that facilitates user-friendly registration of eGovernment services for service providers (e.g. public authorities) and easy discovery of web services for citizens. In order to extend the already decentralized nature of SOAs, the Access-eGov platform is built on a flexible peer-to-peer network, which can easily be enhanced with new nodes in the network.

As Access-eGov is based on a SSOA, the registration of an Access-eGov service does not only involve the common quality of service data; Access-eGov goes one step further and attaches a semantic description to each service. This allows citizens to semantically discover a requested service. Another feature of Access-eGov is the semantic description of workflows, i.e. a complex sequence of basic web services that need to be executed by the citizen in a given order to fulfill a certain goal. A rich user interface, the Personal Assistant client will hereby support the user.

Since the processing of eGovernment services generally involves citizen data of sensitive and private nature, a solid security architecture is not only an integral part of Access-eGov, but a key requirement for any functional and usable middleware platform. The goal of this paper is to pinpoint the security requirements we imposed after a careful study of the underlying architecture. We identify common requirements for system security as well as those that are especially inherent to the environment of service-oriented architectures. They range from privacy and trust related issues to communication security and access control features which all need to be addressed when creating a secure service-brokering architecture for the electronic government. For these requirements we recommend a couple

of candidate technologies that need to be taken into consideration in order to meet the security needs best. Finally, we outline the physical arrangement of the different security features in the various system parts.

The remainder of this paper is organized as follows: Section 2 introduces the Access-eGov architecture including general requirements, the functional system view, the data view and the system architecture. In Section 3 we introduce the security requirements we derived from the designed Access-eGov architecture. After presenting related work in Section 4, Section 5 finally concludes the paper and gives an outlook on future work.

2 Semantic Service-oriented Architecture

The security requirements we impose in Section 3 are based on the SSOA built in the project Access-eGov (see Section 1). In order to provide the reader with a basic overview of the environment that the architecture will be operating in, we first describe the required qualities and the functional features of the designed Access-eGov architecture. This introduction will be followed by a brief insight into the system components and the technology used.

2.1 General Platform Requirements

During the early design phase in Access-eGov, several user requirements for the platform functionality were collected [5]. Since Access-eGov aims at becoming a useful tool in daily work for public agencies and citizens, its development is entirely driven by domain requirements. Based on the analysis of a series of interviews, we extracted those recommendations that were considered crucial to the design of the overall Access-eGov system architecture [6].

From these interviews, we identified the following categories of functional requirements that Access-eGov’s platform is supposed to meet:

- High Availability
  In contrast to the monolithic architecture developed within the research project Webocracy [8], in which a classic client/server technology was used as the underlying architectural paradigm, Access-eGov is required to be less failure-prone. The service-brokering middleware of this trans-national semantic e-Government platform requires a large-scale networking architecture that is available at all times. Since the platform will interconnect numerous e-Government service providers and thousands of citizens from several European countries, the architecture should be outage-resistant and scalable in order to handle thousands of service requests at a time. In addition, it should be ready to allocate responsibilities and delegate requests dynamically.

- De-centralized Management
  A de-centralized administration of all system components is logically resulting from the demand for high availability. Since various public service providers are located in different countries, the need for a scattered administration of the overall brokering platform is obvious. The platform has to guarantee stable and reliable operation and we therefore propose a de-centralized and meshed management of the overall Access-eGov system. This will empower service providers from different domains to manage and maintain the infrastructure independently from one another without the need for structured coordination beyond the negotiation of common interfaces. Also, Access-eGov is required to be integrated smoothly into the already existing database-landscape. Ideally, the existing domain knowledge storages are only loosely coupled with new components introduced by Access-eGov.

- Semantic Description of Web Services
  All services brokered and orchestrated via the Access-eGov platform will have to be semantically enriched. The semantic meta-information will enable the Personal Assistant clients as the citizen’s interface to easily look up service information in distributed directories registering public service information. For this purpose, the Personal Assistant will actively search in registries, where each semantically annotated e-Government service is listed. The Personal Assistant interface then looks for those service descriptions matching a personal life or business event and bundles them to high-quality e-Government services. In order to implement such a scenario, the platform must provide ways to annotate web services as well as a domain ontology that controls the vocabulary. The platform will also provide rich functionality to semantically annotate already existing services that it will broker over the network.

- Composition of Workflows
  Almost all public services of high-quality and use for the citizen will require more than one e-Government service. Therefore, the platform should be supported by an ontology that takes care of the correct composition and execution of workflows for each event. Following that information, the overall system can locate all relevant web services and combine them in a given order that makes sense to the citizen and to the public agencies.

- Mapping between Semantics
  More and more larger organisations are developing and already using their own taxonomies and controlled vo-
recognizing vocabularies to semantically enrich the services they offer. Service providers from different countries will most unlikely use the same terminology for the annotation of their services. In order to enable Access-eGov to broker service information in a large-scale environment with a multitude of different organisations all having their own semantics, the platform should be capable of semantically mapping descriptions from different domains. To overcome this semantic incompatibility, we shall use mediation that performs mapping operations between certain domain ontologies as well.

- Openness of Standards
The system will have to be able to generically support all legacy systems currently used in those public agencies that are taking part as user partner in the project. This challenge could be dealt best with by using wrapper engines on the premises and intelligent connection nodes in order to resolve heterogeneity on the various system levels. Therefore, XML-based interfaces should be offered, allowing the standardized exchange of processing data.

- Security
Since most of the public services (both traditional and electronic) require the exchange of citizen-related sensitive data, special care must be taken when designing a service-oriented architecture for electronic government purposes. Section 3 will deal in depth with all security related requirements that a system like Access-eGov will need to fulfil.

The requirements as outlined above demand for new ways in information brokering architectures. Therefore, already proven technologies have to be extended and brought together in order to build a performant and reliable middleware architecture that lives up to the demands of the e-Government environment in particular.

2.2 Functional System View

After having outlined the general requirements that a semantic service-oriented architecture shall satisfy, we focus on the functional system view. We hereby have to differentiate between the requirements of service-end-users (the information consumers) and those of service-providers that are using Access-eGov to have their public services brokered (the information providers).

2.2.1 Information Provider View

An information provider has to tackle three main tasks as described in Figure 1, namely registering new services, annotating services and building generic workflows out of already defined and registered services:

- Annotating a Service
  The public administration official chooses from a set of available ontologies. He uses the therein contained concepts and relationships to mark up the important aspects of the service or website he currently describes. This will include information about who is eligible to actually use that service, what form of identification is required up to the point of the privacy policies that the service itself can offer. Finally, the annotator has to add some general descriptions, including contact details for persons that are responsible for the service’s operations.

- Registering a Service
  After successful annotation of the service, all generated data is stored persistently; the service can now be retrieved from all participating system components connected to the network.

- Creating a Workflow
  A far more challenging task for the public administration official is to build generic workflows based on already annotated services or already existing real-world workflows. First, the annotator has to either find suitable services or already existing, suitable workflows. This is done through either searching or browsing the service and workflow repositories.

After having found suitable services or workflows, the annotator has to concatenate them in a meaningful combination that is imposed by internal procedural guidelines or legal conventions.

2.2.2 Information Consumer View

By nature, the information consumer (i.e. the citizen or a business user) has a significantly different view on a service-oriented platform as can be seen in Figure 2. A citizen has two major possibilities of interaction with the platform; he can specify his goal and can command the platform to execute the retrieved services.

- Specifying a Goal
  The citizen has to articulate wishes to the system and has to tell the actual task that he wants to accomplish. Therefore the Personal Assistant presents a list of life events or services for browsing and the possibility to search for life events. Once the citizen has chosen one of the offered life events, the Personal Assistant will translate the selection into a goal. It has to be noticed that the life event could either be connected to an atomic service, or to a predefined, generic workflow composed of several atomic services.
Figure 1. An Information Provider View on a SSOA

- Executing a Scenario
  When the user wants to achieve his goal, he lets the Personal Assistant start the execution of the retrieved service or workflow. The current progress of this run is always visible to the user through the Personal Assistant client.

- Executing a Traditional Offline Activity
  In many cases, the execution phase of such services or scenarios will also include activities that are only available offline. Access-eGov will, in this case, simply wait until notified of the completion of a specific offline activity.

2.3 Data View

Within a service-oriented platform like Access-eGov, "ontologies" are necessary to categorize real-world concepts (e.g. objects, procedures) and to express them in a semantic way that is defined and agreed upon by communities of users (e.g. collaborating organisations). In technical terms, an ontology constitutes an explicit formal specification of a shared conceptualization. This definition entails a number of essential components which are common for all sorts of ontologies. Ontologies define an agreed common terminology by providing concepts and declaring relationships between these concepts. In order to capture semantic properties of relations and concepts, the ontologies need to provide sets of logical expressions in a structured machine-readable language. We use three basic ontologies in several parts of the Access-eGov system. They are:

- Life Events Ontologies
  Life events denote specific situations (in the life of a citizen or a life cycle of an organization) that require a set of public services to be performed. It contains information about particular “Goals”, which can be grouped into several “Generic scenarios”. This ontology is capable of describing complex real-world scenarios like “marriage”.

- Service Profiles Ontologies
  A service profile specifies what the service logically provides from a user’s perspective and is used by the public administration to advertise services. This ontology will have to be powerful enough to describe all non-functional and functional properties that a public service can have. This ontology corresponds to the real-world descriptions of a service’s capability (e.g. a service resulting in issuing a passport).

- Access-eGov Domain Ontologies
  Domain ontologies are considered lower level ontologies within the overall system. They are used to describe all the relevant domain information related to a user’s scenarios. That means they will describe the physical functional and non-functional properties of a particular service, like required data types, and the like.

2.4 System Architecture

As already mentioned before, the architecture of our semantic service-oriented platform, as depicted in Figure 3, will be divided into three major component groups:

- Infrastructure
  The core part of the system is the so called Access-eGov Infrastructure, the service-oriented middleware which itself is subdivided into different component categories. All services that Access-eGov is supposed
Figure 2. An Information Consumer View on a SSOA

to broker will need to be shared and found by all participants of the network. Thus, public administrations have to annotate the services they want to share and have them published in publicly accessible repositories that are distributed across the network mesh. These repositories are supposed to persistently store goals (users’ aims), semantic service descriptions (the description of the actual service, for example its capabilities), scenarios (structured service concatenations) and ontologies. These distributed database federations are interconnected and have to be mutually updated on a regular basis in order to offer a system-wide view of all the services that the different organizations and public agencies offer via Access-eGov.

On top of these persistent data storage facilities, generic executable platform services will be required to maintain the middleware components and to manage the dataflows. These so called Core Components transparently enable the service users to execute operations on the basic data repositories. It is here, where the actual Semantic Web service ontology API, i.e. the object model for Semantic Web service ontologies, will need to be located. Another crucial module, the Connection Manager will need to be utilized as the central hub for communication within the whole system in order to make the infrastructure components available to other system components as well as to the outside world. The Security Components are an integral part of the whole architecture and will therefore also be incorporated into the main infrastructure. The actual requirements regarding this component will be found out in Section 3. The most important operation functionality of the whole system, the discovery and execution of web services, is carried out by the respective Discovery, Composition and Execution modules which are using a mediation module to map the different domain ontologies to comparable semantic descriptions.

- **Personal Assistant Client**
Access-eGov will also offer a generic user interface to the citizens, the Personal Assistant client. The client, and its counterpart the Personal Assistant platform component in the middleware, perform another very important aspect of the Access-eGov semantic service-brokering system: to guide the user through public service workflows and to help him in finding the services that satisfy his needs best. Apart from the platform’s own client interface, there will also need to be a fully-featured, generic XML-API available to other front-end implementations.

- **Administration and Management Tools**
During the first introduction phase, only a small amount of services will be available that have explicitly been designed for Access-eGov and are available in a form that the middleware platform is able to directly digest. All the other services, no matter if they are "traditional services” like signing a document in an administration office or "legacy services”, which could for example be a web service or a web page, need to be semantically annotated. Public agencies are supposed to annotate those services that they are willing to expose to the public. These kinds of service-related meta-data will be transferred to the data repositories via the "Access-eGov Annotation Service”. Public administration domain experts may use this generic annotation service component to create said semantic annotations for all kinds of services. Thus, it will be the most important administration tool for our semantic
The offered services are still under control of the public authorities offering them. It is at the authorities' discretion to become part of the Access-eGov single-sign-on domain or implement their own access control and user management. The actual service execution will be carried out on the premises of the participating public agencies or in their respective data centers. The services are only externally made available through Access-eGov and thus do not form an integral part of the overall service-brokering system. They are either electronically accessible (directly via web service interfaces or web forms) or represent "traditional" office services that may merely be described and registered to the middleware. Only executable services will dispose of an electronic XML-interface to the Access-eGov Infrastructure.

3 Security Requirements

After depicting the general system requirements and the components of Access-eGov, the following section will therewith derive several security requirements that such a system will have to satisfy. Already during the early design phase of Access-eGov, security-related requirements were collected from the user partners regarding robustness and security of the overall system [1]. Some of these recommendations already influenced the overall conception of the system components. During the first months of the project, more security requirements arose that are unique to the field of service-oriented architectures. This section refers to the ongoing research that has been invested and features an overview of the requirements as well as an outlook on how security recommendations are considered to be realized inside the several system components.

- Communication Security
  Regarding physical security on the network, user partners expect messages to be sent over secure channels and only to those citizens who are authorized to make use of a certain service. Thus, physical security features in Access-eGov are required to encompass encryption functionality and the ability to issue and digest digital signatures. Since the Access-eGov platform shall be operating as a service-oriented architecture network, XML-messages should be encrypted in accordance with international standards issued by W3C and OASIS. Access-eGov could therefore make use of a selected subset of WS-Security mechanisms on application level. The Web Services Security (WSS) specification from OASIS is a candidate technology defining how to apply XML signature and XML encryption concepts in SOAP messaging. This will satisfy three main security demands: Message Integrity, User Authentication and Confidentiality.

On the application level, the signatures and the integrity of the encrypted messages should also need to be controlled. Before forwarding a service request to the respective public service provider’s server, XML-firewalls should be set in place to check each single XML-encoded message for its encryption and...
signature part. This will also disburden the service provider’s application server and reduce network traffic in the back-office sector. In addition, traffic between participating parties in Access-eGov should be encrypted via SSL technology on network level.

- Trust
  A requirement that arises from the nature of a widespread network with a multitude of nodes is the need to mutually trust the communication partner when requesting a certain service or inquiring a certain database entry. The nature of a peer-to-peer system also mandates that new system nodes can be added without reconfiguration of the whole network thus also relying on a working concept of trust between nodes. This also applies to the communication channels between the various Personal Assistant clients and the platform itself. Mutual trust may be reached with the network components authenticating themselves when establishing communication relations to the user interface clients; different components of the system or security architecture, which may very well run on the same node, may also have the need to establish trust between them.

- Privacy
  In order to gain citizens’ trust in the architecture, the Access-eGov platform needs to provide reliant privacy mechanisms that protect user’s data both on client side and in the infrastructure of the service provider. Citizens must be able to tune individual privacy settings with their Personal Assistant. Possible candidates for a language to define privacy preferences are "A P3P Preference Exchange Language" (APPEL 1.0) [4], the "Enterprise Privacy Authorization Language" (EPAL 1.2) [2] and the "eXtensible Access Control Markup Language" (XACML 2.0) [7].
  In addition to default privacy settings, fine-tuning of privacy preferences should be possible on a per-process basis. Privacy preferences should always be taken into account when discovering and executing web services. Ideally, mechanisms on the service provider’s premises should be in place to enforce users’ privacy settings.

- Access Control
  Considering all requirements to the architecture, we believe that employing an attribute-based access control model is the best choice for the Access-eGov architecture. Following an access control infrastructure as defined in XACML [7], we consider Access-eGov to provide separated components for policy administration, policy information, policy decision and policy enforcement. Such a flexible and dynamic access control infrastructure suits well into a loosely coupled service-oriented architecture like the Access-eGov platform.
  Furthermore, we require each service provider to annotate a service with security attributes in addition to the functional semantic description. This way, Access-eGov can efficiently impose platform-wide access control mechanisms that can be executed automatically by the above mentioned XACML policy components. Upon receiving a request for a service, the distributed access control components take into consideration the service’s security level as required by the public administration and the validity of the credentials provided by the Personal Assistant client. Only then, a user’s Personal Assistant should be granted access to a service and forward the request messages to the public agency in question.

  The actual security mechanisms in Access-eGov are split between the different system components, notably between the Personal Assistant and the middleware platform. The Access-eGov Personal Assistant will handle authentication credentials provided by the user and forward them to the respective public service agency. After choosing a goal and before access is granted, the Personal Assistant lets the platform check the user credentials.

- Platform Security
  Most of the security mechanisms within Access-eGov will need to be implemented as part of the service-oriented architecture middleware, respectively inside its distributed components. The dedicated access control components will need to act as access gateways for the Personal Assistant clients and as guardians for granting service access. The platform will also host the security descriptions for all services. These databases shall be used as persistent repositories when annotating services with generic information about the required user credentials.

- Back-Office Security
  Legacy systems on the premises of the public agencies will most favor the installation of XML-firewalls and traffic-controlling proxy servers. This way, the current data centers will be enabled to securely connect to the semantic service-brokering system of Access-eGov.

- Personal Assistant Security
  The Personal Assistant will mainly be occupied by administering the personal security settings that each citizen can fine-tune. This will give the citizen full control over his personal information set (including credit card information, etc.). The main task of this component will be to ensure end-to-end communication en-
4 Related Work

One research direction regarding security aspects of Semantic Service Oriented Architectures (SSOAs) is to use the semantic information present in the SWSs’ descriptions for access control decisions [9] [11]. In [10] the concept of a semantic firewall is introduced, which acts as an access control component in SSOAs.

Many research projects are also starting to build networks based on SSOAs, like the contributors to the ESSI cluster 2. None of the 6 projects in this cluster even hints at planned security architectures in the projects’ summaries. The TAO project 3 as another example does not explicitly state security requirements as well. Especially in the field of eGovernment, many projects like e.g. OntoGov 4 do not have built-in security mechanisms on their research agenda.

On the other hand, many projects taking security into consideration, like Terregov 5, are lacking of the semantic perspective to service-oriented architectures. Treating a high level of security and semantic service-orientation as equal targets will therefore be a challenging task for our future work.

5 Conclusions

With the emergence of service-oriented architectures the need for suitable security infrastructures arises. Especially in large-scale environments like many eGovernment applications such a security infrastructure is a decisive success factor.

In this paper we introduced requirements for a security infrastructure in a semantic service-oriented architecture carried out in the European research project Access-eGov. After we described all architectural layers we introduced requirements of a corresponding security infrastructure, concentrating on communication security, trust, privacy and access control. We also laid out necessary security mechanisms that should be in place.

Our underlying semantic service-oriented architecture bears several opportunities for a complete security infrastructure. The existence of a user-friendly Personal Assistant enables the easy input and fine-tuning of individual privacy preferences. Furthermore, the existence of semantic information in the platform can be used for powerful and flexible access control mechanisms. Finally, the loosely coupled character of the architecture even facilitates the distributed storage of security components.

Future Work will involve the design and the development of the described security components in accordance with the Access-eGov infrastructure. This will involve the thorough testing of the infrastructure in pilot scenarios of three different countries.

Acknowledgment

We would like to thank our project partners for helpful comments and stimulating discussions. This work is done within the Access-eGov project, which is supported by the European Union under the IST Programme, contract No. FP6-2004-27020. The content of this publication is the sole responsibility of the authors, and in no way represents the view of the European Commission or its services.

References