Engineering Authorization Services for the Service Oriented Architecture

Sarath Indrakanti

Information and Networked Systems Security Research,
Department of Computing, Macquarie University,
Sydney, NSW, 2109, Australia
sindraka@ics.mq.edu.au

Abstract. The service-oriented architecture (SOA) can be used to build new solutions leveraging services, to cleave together existing applications or to cleave apart existing applications. The SOA provides many benefits such as cost saving to organizations by increasing the speed of implementation of any application(s) required and reducing the expenditure on integration technologies. However, security is one of the main roadblocks for enterprises to delay development and deployment of their services. Although there are standards for providing confidentiality, integrity and message authentication for services, there is not yet a standard specification for authorization services for the SOA. We address this important gap in the area of security for the SOA. In particular, we will propose an authorization policy language as well as an authorization framework for the SOA.

1 Background

The SOA is an architectural style and its aim is to achieve loose coupling among interacting distributed software systems. The SOA can be defined as a way of designing and implementing enterprise applications that deals with the intercommunication of loosely coupled, coarse grained, reusable artifacts (services). The SOA is made up of independent services interconnected via messaging. Platform independent service interfaces are defined to invoke these services. The SOA consists of service providers and service consumers. Service providers define what the service looks like and how to invoke it through an implementation independent service interface. Service consumers use this interface to invoke the service. The SOA also provides discovery mechanism to act as an intermediary. Service providers publish their service interface using the discovery mechanism for consumers to find and invoke the service.

There are three broad categories of service use. 1) To build new solutions with services. Services enable enterprises to have independent pieces within applications that can be developed and maintained independently and also to scale-out. 2) To cleave together existing applications. Services enable connectivity Business-to-Business (B2B) and Enterprise Application Integration (EAI). Services also let us formulate business processes to enable workflows across heterogeneous environments in turn letting us tap in to the IT infrastructure, and 3)

To cleave apart existing applications. Use cases of the SOA include Supply Chain Management, Customer Relationship Management (CRM), Enterprise Application Integration (EAI), Enterprise Resource Planning (ERP), and Portal Web Sites amongst others. The SOA enables cost saving by increasing the speed of implementation of any application(s) required and reducing the expenditure on integration technologies. The SOA provides enterprise agility, to quickly respond to rapidly changing market needs and business requirements by quickly building new applications and quickly updating old applications.

2 Securing the SOA

In general, security for the SOA is a broad and complex area covering a range of technologies. At present, there are several efforts underway that are striving to provide security services such as authentication between participating entities, confidentiality and integrity of communications. A variety of existing technologies can contribute to this area such as TLS/SSL and IPSec. There are also related security functionalities such as XML Signature and XML Encryption and their natural extensions to integrate these security features into technologies such as SOAP and WSDL.

The WS-Security specification describes enhancements to SOAP messaging to provide message integrity, confidentiality and authentication. There is also work on XKMS defining interfaces to key management and trust services based on SOAP and WSDL. However, while there is a large amount of work on general access control and more recently on distributed systems authorization, research in the area of authorization for Web services is still at an early stage. There is not yet a specification or a standard for Web services authorization. There are attempts by different research groups to define authorization frameworks [1] and authorization policy specification mechanisms [2] for Web services. Currently most Web service based applications, having gone through the authentication process, make authorization decisions using application specific access control functions that results in the practice of frequently re-inventing the wheel. This motivated us to have a closer look at authorization service requirements for the SOA. We will first address the area of design of an authorization policy language for the SOA. Then we will propose an authorization framework for the SOA. Finally, we will demonstrate these authorization services using a case study in the health care domain.

3 Authorization Policy Language for the SOA [3]

Languages have long been recognized in computing as ideal vehicles for dealing with expression and structuring of complex and dynamic relationships. Over the recent years, a language-based approach to specifying access control policies has (rightly) gained prominence, which is helpful for not only supporting a range of access control policies but also in separating out the policy representation from policy enforcement. One standard authorization policy language defined

for the SOA can replace dozens of application-specific languages. Administrators save time and money because they are not required to rewrite their policies in many different languages. Developers save time and money because they need not invent new policy languages and write code to support them. They can reuse existing code. If one policy language specification is standardized, good tools for writing and managing policies for that language will emerge. Policy languages in which one can specify policies using XML have an advantage over other languages such as Ponder, as Web services based applications using the language can leverage on the benefits of XML such as inter-operability over multiple platforms in a heterogeneous environment. A policy language based on XML technology with its own namespaces and schemas is necessary in a heterogeneous environment of Web services. Also if an XML based policy language is used, standard specifications such as XML Encryption and XML Signature can be leveraged to secure and sign those policies where required. Such a policy languages policies can be specified and referred to by any service based application whether it is running on a Java based platform or the .NET platform. When designing the authorization policy language, we took into consideration the support for a range of authorization policies such as the dynamic separation of duty policy, that are likely to be required for services deployed in a commercial environment.

In particular, we will discuss the following in the dissertation: (a) Survey of authorization policy languages for distributed systems, (b) Analysis of the authorization policy language used by .NET MyServices [4], (c) Extensions to the authorization policy language used by .NET MyServices, (d) Design of XML-based authorization policy language for the SOA, (e) Design of standard APIs to the policy language and the policy engine, and (f) Implementation of the policy language and the policy evaluation engine leveraging existing XACL policy language and engine.

4 Survey and Analysis of Authorization Frameworks[5]

We have carried out a comprehensive survey of existing authorization models both for traditional distributed systems as well as authorization models built for different layers of the SOA.

We will discuss the following in this section: (a) Survey and analysis of authorization frameworks built for stand-alone systems, (b) Survey and analysis of authorization frameworks built for distributed systems, and (c) Survey and analysis of authorization frameworks built for the SOA.

5 Authorization Framework Design Requirements [6]

We will then lay out the design requirements for authorization services built for different layers of the SOA. Broadly speaking, the SOA comprises of Web services and business workflows built using Web services. These workflows are called business processes. Figure 1 shows the layers comprising the SOA.

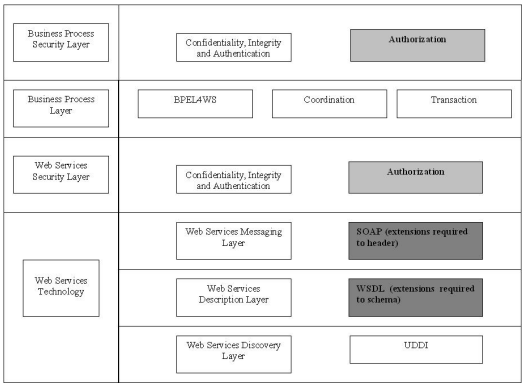

Fig. 1. Layers in the SOA

Authorization requirements differ for the Web services and business processes layers of the SOA. Authorization services for the Web services layer have special design requirements as Web services present a complex layered system. For instance, a service may be a front-end to an enterprise system and the enterprise system may access information stored in databases and files. Web services may be used by enterprises to expose the functionality of legacy applications to users in a heterogeneous environment. Or new business applications could be written to leverage benefits offered by Web services. This means an authorization architecture for Web services must support multiple models of access control. This enables legacy applications to use the access control models they have already been using as well as new Web services applications to use new models of access control or other well-known models of access control such as role-based access control (RBAC). An authorization architecture for the business process layer of the SOA must provide orchestration services to coordinate the authorization decisions from individual partners authorization policy evaluators. Each partner must be allowed to control its own authorization policies and also not require disclosing them to the entire workflow or to the workflow engine. Even in cases where the binding to actual end-points of partner services happens dynamically at runtime, the authorization architecture must be able to orchestrate the partners authorization policy evaluators and arrive at an authorization decision. Currently existing authorization frameworks are either designed for the Web services layer [1] or the business process layer [7] of the SOA. There is no unified model available that provides a comprehensive design of an authorization framework that provides authorization support to both Web services and business processes comprising the SOA.

6 Authorization Framework for the SOA

We will propose a unified authorization framework for the SOA. The framework will provide two separate architectures (indicated by the light grey colored boxes

in Figure 1) that extend the security layers of Web services (WSAA) and business processes (BPAA). The BPAA relies on the features provided by the WSAA. In particular, we will discuss the following:

6.1 Web Services Authorization Architecture (WSAA) [8]

(a) Design of the authorization administration and runtime APIs. The administration API is useful to group together a set of related Web services into collections to efficiently manage their authorization related information such as what privileges are required to be sent by a client before invoking a Web service. The runtime API is useful to invoke the necessary authorization components and receive an authorization decision, (b) Implementation of the authorization architecture APIs within the .NET framework, and (c) Benefits of the proposed architecture.

6.2 Business Processes Authorization Architecture (BPAA)

(a) Design of the authorization administration and runtime APIs. The administration API is useful to define a business process and mange its authorization related information. The runtime API is useful to invoke the necessary authorization components and receive an authorization decision, (b) Implementation of the authorization architecture APIs within the .NET framework, and (c) Benefits of the proposed architecture.

6.3 Extensions to Other Layers of the SOA

We will also describe our extensions to the Web services description and messaging layers to support the authorization frameworks for Web services and business processes (indicated by the dark grey colored boxes in Figure 1). Extensions to the description layer are required to make prospective clients aware of what authorization privileges are required to invoke a Web service or a business process. Extensions to the messaging layer are required to carry authorization related privileges, for instance, in the form of credentials.

7 Demonstration of Authorization Services in a Health Care Application

We will demonstrate the proposed authorization services the policy language and the authorization framework for the SOA using a case study in the health care domain. In particular, we will discuss the following:

(a) Demonstration of the proposed policy language and policy engine features, (b) Demonstration of the authorization services provided by WSAA and BPAA, and (c) Demonstration of the proposed extensions to the service description and messaging layers.

8 Conclusion

The service-oriented architecture (SOA) can be used to build new solutions leveraging services, to cleave together existing applications or to cleave apart existing applications. The SOA provides many benefits such as cost saving to organizations by increasing the speed of implementation of any application(s) required and reducing the expenditure on integration technologies. However, security is one of the main roadblocks for enterprises to delay development and deployment of their services. Although there are standards for providing confidentiality, integrity and message authentication for services, there is not yet a standard specification for authorization. We will address this gap in the area of security for the SOA. In the dissertation, we will first address the area of authorization policies and describe our proposal of an XML-based authorization policy language and its evaluation engine for the SOA. Then we will discuss the authorization framework requirements for the SOA. They differ for the Web services and business processes layers comprising the SOA. We highlighted the major differences in this paper. In particular, we will describe two separate authorization architectures designed using the requirements we laid out after carrying a comprehensive survey of authorization models built for stand-alone and distributed systems as well as for the SOA. Finally, we will demonstrate the authorization policy language and the authorization framework for the SOA using a case study in the health care domain.

References

1. R. Kraft. Designing a Distributed Access Control Processor for Network Services on the Web. In *ACM Workshop on XML Security*, Fairfax, VA, USA, 2002.
2. S. Godik and T. Moses. eXtensible Access Control Markup Language v1.1 (XACML), 07 August, 2003.
3. S. Indrakanti, V. Varadharajan, and M. Hitchens. Authorization Service for Web Services and its Application in a Healthcare Domain. International Journal for Web Services Research, Idea Group Publishing, vol. 2, issue 4, pages 94-119, 2005.
4. Microsoft Corporation. Microsoft .NET MyServices Specification, Microsoft Press, 2001.
5. S. Indrakanti, V. Varadharajan, and M. Hitchens. Analysis of Existing Authorization Models and Requirements for Design of Authorization Framework for the Service Oriented Architecture. In *The 2005 International Symposium of Web Services and Applications*, Las Vegas, USA, 2005.
6. S. Indrakanti, V. Varadharajan, and M. Hitchens. Principles for the Design of Authorization Framework for the Service Oriented Architecture. In *International Conference on Internet Technologies and Applications (ITA 05)*, Wrexham, North Wales, UK, 2005.
7. H. Koshutanski and F. Massacci. An Access Control System for Business Processes for Web Services. Informatica e Telecomunicazioni, University of Trento, Technical Report DIT-02-102, 2002.
8. S. Indrakanti and V. Varadharajan. An Authorization Architecture for Web Services In *19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, Storrs, Connecticut, USA, 2005.