Towards a Reference Architecture for Access Control in Distributed Web Applications

Worachet Uttha[1], Clara Bertolissi[1,2], and Silvio Ranise[2]

[1] LIF, CNRS UMR 7279 & AMU, Marseille, France
[2] FBK, Trento, Italy

Abstract. Web services are independently written and managed, each with its own access control policy, thus it is challenging to control the access to the information they own. A particularly difficult case occurs when a service invokes another service to satisfy an initial request. We call this "Transitive access problem". To tackle this issue, we propose to use XACML for defining Attribute based Access Control (ABAC) policies for web services. We focus on the authorisation issue of access control and solve the transitive access problem by integrating in the XACML architecture a module for supporting multiple attribute domains.

1 Introduction : Problem and Motivation

Nowadays organisations increasingly employ distributed systems in order to improve their service performance. Web services, which are a form of distributed system architecture, seem to become the preferred implementation technology for realising the integration and interaction between various systems in Internet and Intranet environments. They also offer many benefits over other types of distributed computing architectures, such as maximum service sharing, reuse and interoperability.

Web services, each managing their own security policies, must interoperate while maintaining secure access to their information. Therefore, in this context, access control becomes one of the challenging issues that must be well-defined in order to ensure a secure cooperation. A particularly difficult case involves a service that invokes another service to complete its computation and serve a request. We call it the "Transitive access problem". We assume here that there is a unique path from one service to another. Web service orchestration issues are beyond the scope of this work.

To explain it in a more concrete way, we give next an example about a medical scenario inspired from [1] that can be seen as an instance of the scenario in Figure 1. We suppose that Attribute Based Access Control (ABAC) is in place in the different services provided by the medical clinic. The ABAC model is well-adapted for open, dynamic and distributed scenarios, since users can identify themselves by using their attributes unlike Identity Based Access Control where the access is directly associated with a user's identifier[4].

In our example, we consider a clinic composed of four entities, each entity providing one or more web services protected by ABAC.

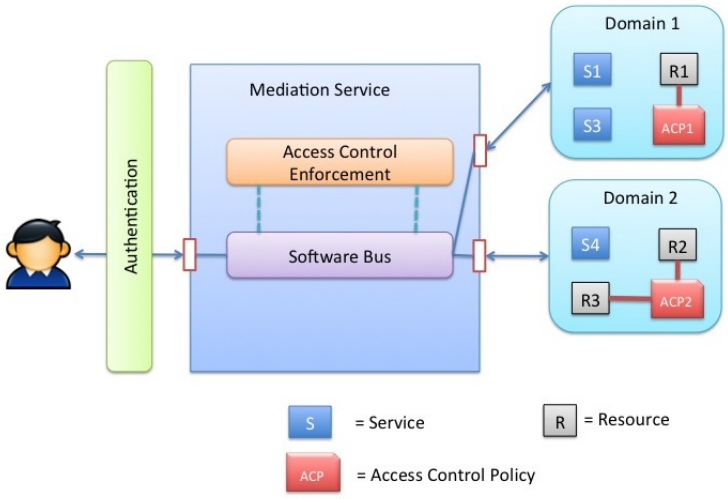

Fig. 1. Access control with a mediation service

- Clinical management: manages the scheduling of patients and captures the actions performed by doctors and nurses.
- Laboratory information system: tracks the tests to be performed and their results.
- Patient records: maintains historical data about patients health.
- Web portal: provides convenient web access to the previous three services. It does not store any confidential data locally. Instead, when the user requests a page, the portal makes service calls to the other services using the requesting user's attributes.

The transitive access problem occurs for instance when a user requests a page from the Web portal that calls the Clinical management service in order to retrieve details of tests that have been ordered by the doctor. In this case, the Clinical management service needs to invoke the Laboratory information system to satisfy this request. As each service is protected by its own access control policy, the requester may have the right to make a call to the Clinical management but this does not mean that he has the right to invoke the Laboratory information system to retrieve the test results. To solve this problem, we will work with the eXtensible Access Control Markup Language (XACML)[9]. XACML is an OASIS standard for authorisation decision making that includes a flexible attribute based authorisation model where access control decision can be made based on the attributes of the subject, the action and the target. We propose to add to the XACML architecture a component that can delegate attributes of one entity to someone else. This implies the definition of a mapping on a (sub)set of attributes that belong to different policy domains.

2 Aims and Objectives

Our main goal during this Ph.D. thesis is to provide an efficient and adaptive solution for access control in the context of Web Services. We can divide our main goal into four sub-goals :

- Specification. We aim at identifying and representing the different features of access control needed in a distributed and dynamic context such as web services. We want to highlight key challenging issues and outline possible ways to overcome them.
- Design. We want to extend capacities and abilities of existing standard tools in order to respond to the problematic issues identified in the first part.
- Implementation. We expect to have an implementation of our solutions able to provide both the desired functionality and the required security of the system.
- Validation. We will validate our approach through concrete case studies.

In particular, in the specification phase, we have focused on the transitive access problem for web services, since managing access in the case of transitive calls is a challenging issue and no satisfactory solution is available in the literature (see next Section for a discussion). Therefore, the main expected contribution of this part is providing an alternative specification of the access control model that could solve the transitive access problem. One of the important issues to address is the presence of multiple domains, since each service has its own access control policy based on attributes and users may not be recognised in every domain. We propose to delegate (a subset of) user's attributes from one domain to another. For that, we need to define a delegation graph, which is a Directed Acyclic Graph (DAG), that describes who (i.e. the delegator) has the right to delegate what (i.e. part of his attributes) to whom. This suppose to have previously reached an agreement between the different participants on a set of attributes that are allowed to be delegated. Concerning the design, we have chosen to improve the access control for Service Oriented Architecture (SOA) based on Web Services standard. For the implementation, we add a support to the XACML architecture in order to delegate requester's attributes from one service to another in the case of multiple domains. Finally we want to test our results on practical case studies, suitable to represent the transitive access problem.

3 Related work and expected contributions

The transitive access problem occurs frequently in big organisations which employ many services, each with its own access control policy. In [2], a solution for transitive access has been proposed but it addresses to the case of a single organisation: all services have the same access control model which is defined as Authorisation Based Access Control (ZBAC). In this context, there is no need to federate identities and find a global agreement on the meaning of attributes. [3] considers multiple policy domains and dynamic delegation of authority from

one user to another focusing on the use of credentials. However, it does not specifically consider the problem of access request evaluation and access decision making in the case of transitive calls. [5] addresses the problem of access control for web service composition. The access policies are specified in Pure-Past Linear Temporal Logic (PPLTL) that allows to exploit the history of service invocations to make access control decisions. However, we think that in practice the specification of policies in PPLTL is not very friendly from the point of view of a security designer. [6] and [7] also discuss access control in web service composition. Nevertheless, their approach is different from ours. They consider the issue of service unavailability along a pathway to a target service, and they solve it by invoking dynamically alternative services belonging to different domains.

As Single Sign On (SSO) [8] has provided a single authentication mechanism thus enhancing the interoperability of web services, in the same way we aim to reach a standard for the authorisation aspect of web service access control. We do not consider only the specification of security policies or the management of security in the system, but we aim at considering the whole security architecture of the system, keeping in mind the satisfaction of the security requirements while guaranteeing the desired functionality.

Summarising, we aim at producing a reference architecture and implementation which will provide a significant improvement towards the standardisation of access control in Services Oriented Architectures.

4 Work progress

Specification of the model We have decided to use access control policies specified following the ABAC model. Due to the presence of multiple domains, we have defined a delegation policy based on the mapping of a sub-set of attributes belonging to different policy domains. This is formalised by a delegation graph in the form of DAG which describes the way requesters' attributes (e.g. roles in the medical clinic example) are delegated in each service. The delegation graph is of crucial importance to determinate access request decisions in the case of transitive service invocation.

Architecture At architectural level, adding a support for the delegation of attributes in XACML requires some considerations before deciding where such a component could be incorporated into the XACML model (see Fig. 2). The major actors in the XACML model are: the Policy Administration Point (PAP) which manages access authorisation policies, the Policy Decision Point (PDP) which evaluates access requests before issuing access decisions, the Policy Enforcement Point (PEP) which is the endpoint for authorisation request and response, the Policy Information Point (PIP) acts as the source of various attribute values and the Context handler which converts a request from its native form into XACML format and an XACML response into its native representation. The delegation module could be called by the PEP. In this case each application will need to be modified in order to use a delegation module, since the PEP is an application

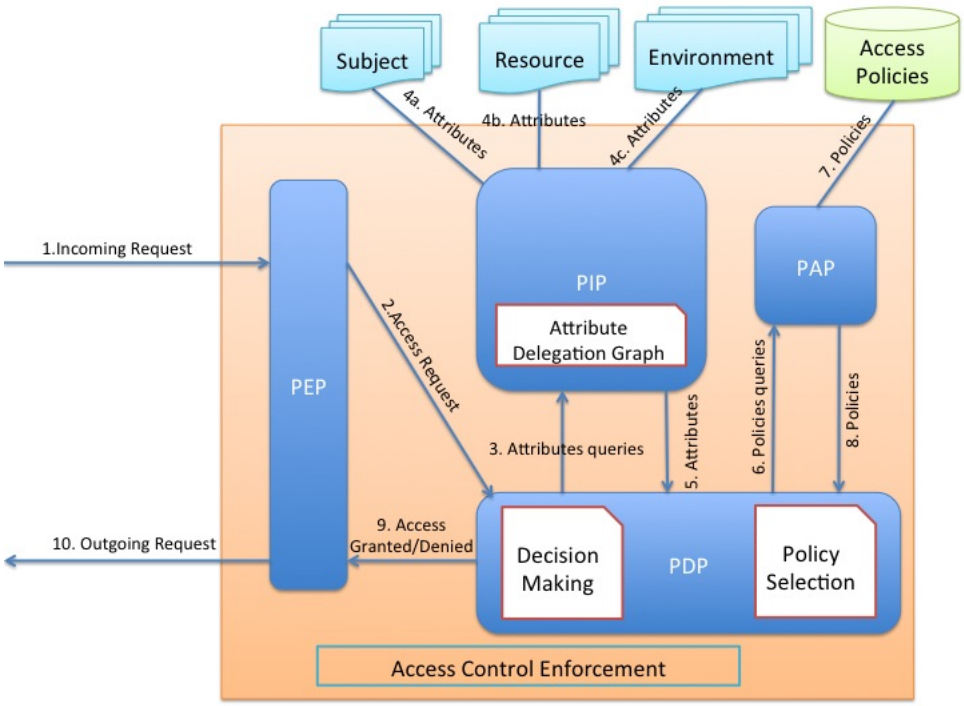

Fig. 2. XACML Extended : PIP with delegation

dependent component. It could be also called by the Context Handler. In this case, existing components, i.e. PEP, do not need to change. The only one that needs to change is the Context handler itself. Another solution is to integrate the delegation module to the PIP, in which case existing application do not need to be modified since the PIP is the closest component and has a direct link to subjects, resources and environment. The last one is our preferred approach since in our opinion it represent an easy and efficient way of incorporating the delegation module into the XACML architecture.

Our approach has been adapted from [3]. The authors in [3] propose to add to the XACML conceptual model a component for the delegation of authority dealing with the managing of attribute credentials issued by trusted authorities. We do not consider attribute trusting issues in our work, instead we introduce the delegation graph as a means to delegate attributes, and thus privileges, to other users across different domains.

Implementation We are currently implementing our extended XACML model. We use the WSO2 Identity Server, which has an XACML engine embedded and acts as PAP and PDP. We have chosen as case study the medical clinic and implemented all services as a web service standard (UDDI for services discovery,

WSDL for interface definitions and SAOP for invocations, all of which use XML as the communications format) based on SOA.

Future work Once we have a functional implementation prototype, we plan to apply our solution to concrete case studies such as the online services provided by the University of Trento for students (see the Smart Campus Project `http://www.smartcampuslab.it`).

5 Conclusion

We have modelled our problem by using ABAC policies and introduced an XAML module for the delegation of attributes. This allows us to solve the access control problem in case of transitive access requests. We are currently implementing our extension in the XACML standard and would like to apply our solution to concrete case studies. The main contribution of our research to the field of engineering secure systems is on the one hand, to ensure both the required security and the desired functionality of systems based on SOA, in particular in the case of transitive access requests; on the other hand, we provide the specification and implementation of policies and security requirements by adding a support to existing standard tools.

Acknowledgments This work was partially supported by the RESTATE Programme, co-funded by the European Union under the FP7 COFUND Marie Curie Action—Grant agreement no. 267224.

References

[1] Fischer, J.; Majumdar, R. "A Theory of Role Composition", IEEE International Conference on Web Services, ICWS '08, pages 320-328, 2008.
[2] Karp, A.H.; Jun Li, "Solving the Transitive Access Problem for the Services Oriented Architecture." ARES, page 46-53. IEEE Computer Society, (2010).
[3] D. W Chadwick; S. Otenko; T. A. Nguyen, "Adding Support to XACML for Dynamic Delegation of Authority in Multiple Domains", in Communications and Multimedia Security. LNCS 4237. Springer , pp. 67-86, 2006.
[4] A. Esfandi; M.. Sabbari. "Study of Access Control Issue in Web Services", International Journal of Computer Applications 49(1):11-16, July 2012.
[5] Srivatsa, M.; Iyengar, A., "An Access Control System for Web Service Compositions," IEEE International Conference on Web Services, ICWS'07, pp.1-8, 2007.
[6] Wei She; I-Ling Yen; Thuraisingham, B.; Bertino, E., "Security-Aware Service Composition with Fine-Grained Information Flow Control," Services Computing, IEEE Transactions on , vol.6, no.3, pp.330,343, July-Sept. 2013
[7] M. Mecella, M. Ouzzani, F. Paci, and E. Bertino. 2006. "Access control enforcement for conversation-based web services". In Proceedings of the 15th international conference on World Wide Web (WWW '06). ACM, New York, NY, USA, 257-266.
[8] Single Sign On (SSO), `http://www.opengroup.org/security/sso`
[9] "eXtensible Access Control Markup Language (XACML)" v3.0, 22 Jan. 2013, available from `https://www.oasis-open.org/committees/xacml`