Access Control System Using Web Services for XML Messaging Systems

Ali Kaplan, Ahmet E. Topcu, Marlon Pierce, and Geoffrey Fox
Community Grids Lab, Indiana University
501 North Morton Street, Bloomington, IN 47404-3730
{alikapla, atopcu, marpierc, gcf}@indiana.edu
Phone for Corresponding Author: 812-856-1212
FAX for Corresponding Author: 812-856-1537

Presenting Author: Ahmet E. Topcu

Abstract

We describe the design of an Access Control System using Web Services for information and content management. In this paper we described the solution for control mechanism of the information systems which have unique message id, requesting and giving permissions to different channels, and accessibility of the user to the specific channels by giving permissions to the user to use XML Messaging Systems using Web Services. This paper presents an overview of the research efforts undertaken by our group to build access control services around a Web Services model.

Keywords

Authorization, Web Services, XML Metadata

Introduction

Information and content management systems require access controls to information content. In publish/subscribe style systems, for example, there will be various restrictions on access to both topics and particular privileges associated with that topic, or channel. We are particularly interested in access control systems that are associated with XML metadata systems. XML metadata is important for future Web service applications [1]. Open Grid Service Architecture (OGSA) [2] and Semantic Web [3] both depend on metadata [4]. We have examined the low-level requirements for managing the XML nuggets of such systems, which include the following: composing tools for creating valid, correct XML metadata nuggets; an architecture and implementation for delivering metadata in the form of messages to listeners who has access to use specific message channel; metadata browsers that can sort and display the nuggets by category; and user role/access control system to define user levels and privileges. The implementation of these ideas is discussed in Ref. [5]. Specific applications of this system include a news group system and a book reference manager.

As a next phase of research, we want to a) decouple our prototype systems into Web Services with well-defined interface, b) formalize the various roles and privileges that exist in the system. Web services invoke remote methods by using (typically) XML-based protocol and interface definitions. The message protocol, usually SOAP [6], is bound to a lower level transport protocol such as HTTP. The method interface (expressed in WSDL [7]) describes agreed-up set of methods, parameters and return type for the services. We want to take advantage of these technologies in our systems. Particularly, we want to use them to build a reusable Access Control System that can be used within other systems besides our metadata system.

Access control systems are important to any system with multiple users, and we highlight here other work in this area. For example, UNIX [8] provides a very simple but powerful form of access control that partially inspires some of our system. Akenti [9] is an access control system for distributed resources based upon Public Key Infrastructure. An important recent development for Web Services is WS-Policy [10], which provides secure environment for Web Services. Also for Web Services, SAML [11] can be used to convey both authentication and access control assertions in SOAP headers.

Access Control System Requirement and Structure Data Models

XML metadata may be organized into various categories, or channels. In our newsgroup system, for example, we have many news topics. Each topic includes users with different types of roles, with multiple access privileges. Individuals may also possess different roles in different topics. The default "user" role has privileges to read and optionally write to one or more message channels. Users have additional options with regard to the choice of message delivery mechanism. That is, a user may request message notification by email, through a web interface, or both. Each channel users has been assigned to the role and group. Each role has several privileges (properties). For example, a user may additionally request that attachments to topic postings be sent to him through email. User can make requests to change these properties, which are granted or denied by channel administrators.

Other possible roles include "administrator", and "super-administrator". Message Channel Administrators have the authority to assign users to a specific message channel. A channel user may have administration privileges over more than one message channel, and a specific channel has one or more administrators. They also modify the access rights of a user, denying a user the privilege of writing to a particular channel, for example. Super administrators manage the entire messaging system. In addition to possessing administrator authorities for all channels, this role has the authority to create new messages channels and assign administrators to them.

The channel defines the main object for information systems which defines as unique URI. Each channel has assigned roles, and each role has rights to be confirmed by administrator in order to use the channels. These channels can be information messaging channel or file systems or any type of systems needed access control systems. The relation is shown in Figure 1: Each channel unique name, and defined roles. Role object have name and rights.

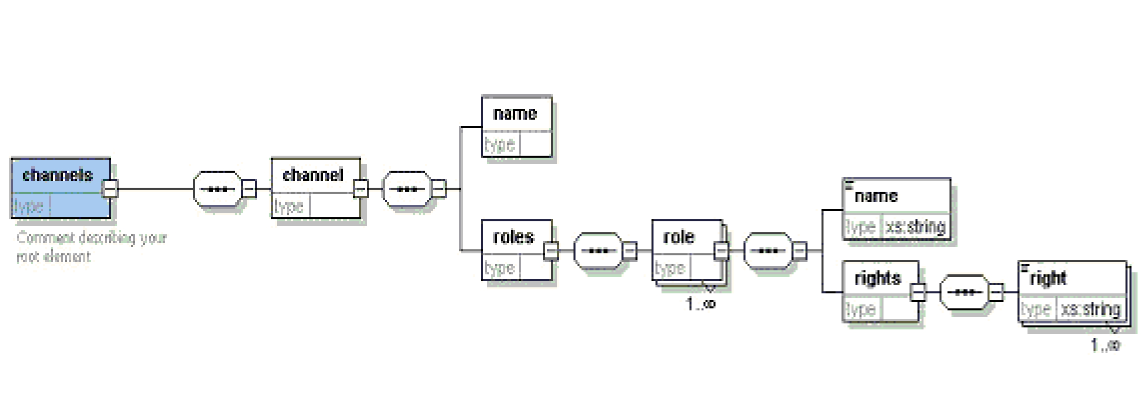

Figure 1 The data model for roles and privileges is expressed in XML.

The current system has an option for selection mechanism for channels. We define roles for message channel. Admin role, user role, and super administrator role are defined in the newsgroup system, for example. In the access control system, super administrators control the channel administrators, and channel administrators control the channel users.

Each channel users has been assigned to the role and group. Second group can modify, remove user rights from the message channel group by having these control structure.

The current system has an option for selection mechanism for channels. Each channel has its own administrator, who can grant or deny the requests permissions and services for a channel. This is shown in Figure 2. These are used by individuals to request changes to privileges. There are three sections. The User Request Pool contains requested objects that are initiated by different users of the channels. The Admin Pool captures the request objects and handles the results based on the administrator of the channel. The Result Pool sends users the results of their request. User might have different roles for each news channels. In the figure, for example, Administrator of Ch1 confirms the user request for both User U1 and User U2 for the channel Ch1. Then, both User U1 and User U2 may use the channel by having Role R1 in the channel system. However, for the channel Ch2, Administrator of Ch2 confirms only the User U1's request having a role R1. Administrator of Ch3 rejects User U3's request for channel Ch3. In summary, different administrators can handle user request objects, which have different roles and different channels.

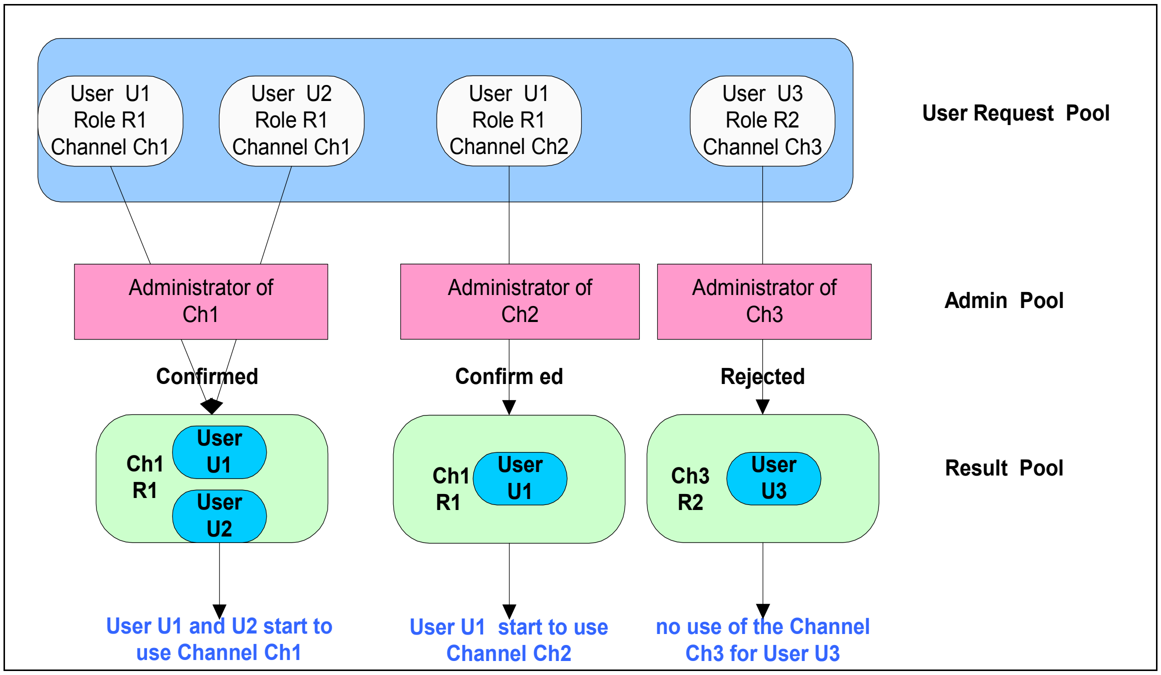

Figure 2 User request and confirmation model.

We assign all the channels a unique, hierarchical URI [12], which provides a method of inheriting permissions, similar to the UNIX file structure. Consider a message group for researchers in Signal Image Processing (SIP) This channel should have http://[(unique)destination]/Organization/Newsgroup/SIP. This system can be integrated to the XML Messaging System. If user want to create a new group derived from the SIP message channel, the new URI for the new channel is http://[(unique)destination]/Organization/ Newsgroup/SIP/SIPArchive. The administrator of the SIP channel will be automatically assigned to the SIPArchive channel. However, if the owner of channel doesn't want to have the upper level administrator to be assigned to the new channel, only the initiator of the channel becomes administrator of the new channel.

Web Service for Access Control

Web services have many advantages over others: protocol independent services, well-defined interfaces for distributed services, separation of interface from implementation (transparency). Due to transparent services capability of web service, the underlying data-storage implementation may be XML database, relational databases, or flat file system.

The system architecture is shown in Figure 3. The end user sends its request in the SOAP message format. The web server forwards incoming requests to the web service engine, which executes the proper web service in the web service pool. The invoked web service ingests the SOAP message, connects to database using JDBC connection, and performs the incoming service request. If there are responses for the incoming service request, it wraps it into a SOAP message and passes it to the Web service engine. The Web service engine delivers the output to requestor by passing it to web server.

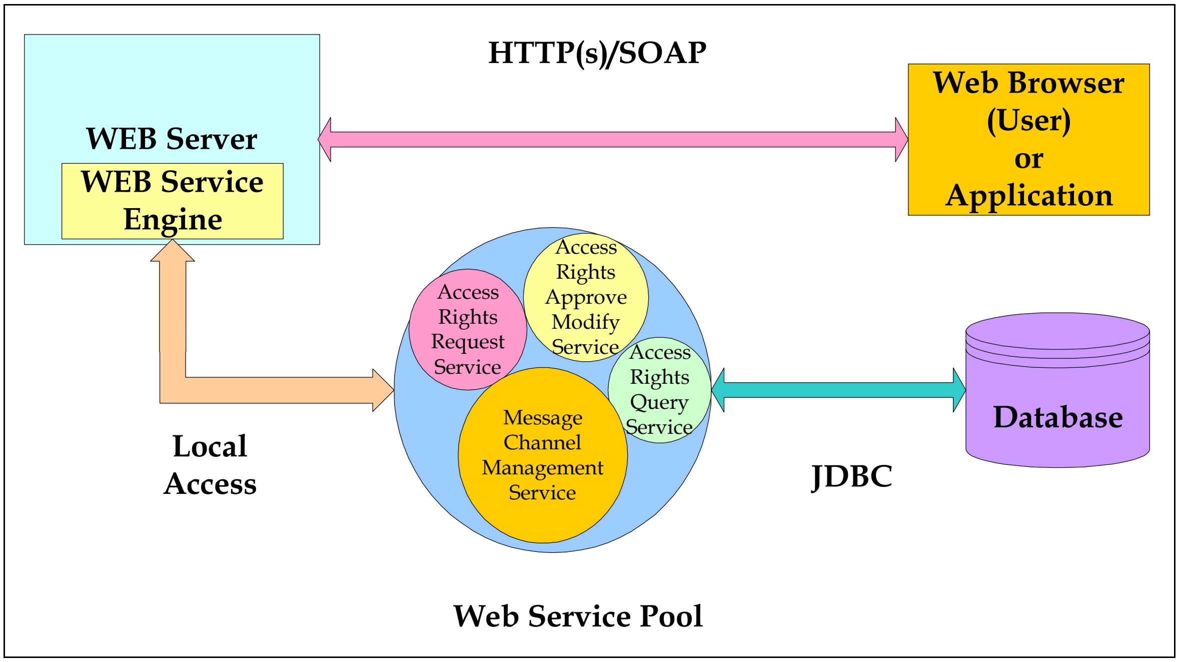
HTTP(s)/SOAP
WEB Server
WEB Service Engine
Web Browser (User) or Application
Access Rights Request Service
Access Rights Approve Modify Service
Access Rights Query Service
Message Channel Management Service
Database
Local Access
JDBC
Web Service Pool

Figure 3: Interaction of users with the Access Control service.

Let us now examine services needed for manipulating our data model.

Access Rights Request Service:

This service allows a user to make a request on a desired message channel to have proper access rights. For instance, a new user makes a request for “read” and “write” access rights on the “java beginners” message channel by calling this service transparently using her browser. When Access Rights Request Service receives this incoming request, it uses the JDBC to connect to database and enters user’s request.

Access Rights Approve, Modify Service:

Channel administrators or super administrators use this service. A channel administrator may approve, modify or reject any request. In addition, she may modify the current access rights of the subscribed users by using this web service. Super administrators assign users as a message channel administrators or remove such privileges by using this service.

Access Rights Query Service:

This service is designated for users who do not have the right to enter any information (access rights request, modification of access rights, etc.) into database but need to access rights information in the database. For example, a message-brokering publisher may need this service to verify the current message sender has write access on given message channel before publishing his/her message. The following are the current methods of this service: getReadableTopics, getWritableTopics, getAllUsers, getUsersHaveReadAccess, hasReadAccess, hasWriteAccess, getUsersHaveWriteAccess, getAllTopics, hasReadWriteAccess.

Message Channel Management Service:

Super administrators use this service. They can add new message channel into system, remove expired message channels from the system and manage their channel administrators by calling this service.

Summary and Future Work

We have presented a data model for access control definitions on an information channel. We implement services that manipulate instances of this data model using a Web services approach.

Access controls are part of larger security framework. Authorization must typically be coupled with two other security concepts: authentication and transport level security. In authentication part, the correctness of the user identity is verified. Data integrity and privacy are typically provided by transport level security mechanisms such as SSL. The access control system we have presented here depends on an external authentication method. Currently, we implement only HTTP-based authentication. Future versions may incorporate other authentication systems such as PKI, Kerberos or Shibboleth.

References

[1] M.Champion,C.Ferris,E.Newcomer,D.Orchard "*Web Services Architecture*", W3C Working Draft 14 November 2002.Avaliable from http://www.w3.org/TR/2002/WD-ws-arch-20021114/.

[2] I. Foster, C. Kesselman, J. M. Nick, S. Tuecke1 *"The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration".* Available from http://www.globus.org/research/papers/ogsa.pdf .

[3] R.V.Guha, P. Hayes "*LBase: Semantics for Languages of the Semantic Web W3C Note*" ,23 January 2003. Avaliable from http://www.w3.org/TR/2003/NOTE-lbase-20030123/.

[4] K. Selçuk Candan , H. Liu , R. Suvarna "*Resource description framework: metadata and its applications"* ACM SIGKDD Explorations Newsletter July 2001.

[5] G. Aydin, A.Kaplan, A.E. Topcu, B. Yildiz "An *XML Based System for Dynamic Message Content Creation, Delivery, and Control" IASTED International Conference on Information and Knowledge Sharing (IKS 2002)* ISBN: 0-88986-325-3. Available from http://grids.ucs.indiana.edu/ptliupages/publications/XMLMessaging2.pdf .

[6] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Frystyk Nielsen, S Thatte, D. Winer. "*Simple Object Access Protocol (SOAP) 1.1"*, W3C Note 08 May 2000. Available from http://www.w3.org/TR/SOAP.

[7] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana. *"Web Service Description Language (WSDL) 1.1"* W3C Note 15 March 2001. Available from http://www.w3c.org/TR/wsdl.

[8] F. Grampp and R. Morris, *"UNIX Operating System Security"*, BSTJ, Vol. 62, No. 8, 1984.

[9] S.S. Mudumbai, W. Johnston, M. R. Thompson, A. Essiari, G. Hoo, K. Jackson "*Akenti-A Distributed Access Control System*". Avaliable from http://www-itg.lbl.gov/Akenti/sc98/akenti.pdf.

[10] B. Atkinson, G.Della-Libera, S.Hada,M.Hondo, P. Hallam-Baker, C.Kaler, J.Klein, B.LaMacchia, P.Leach, J.Manferdelli, H. Maruyama *"Web Services Security (WS-Security) Version 1.0"* April 5, 2002. Available from: http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnglobspec/html/ws-security.asp.

[11] J. Hodges, C. Knouse, J. Moreh, R. Philpott *"Metadata for SAML 1.0 Web Browser Profiles"* Working Draft 01, 1 February 2003. Available from http://www.oasis-open.org/committees/security/docs/draft-sstc-saml-meta-data-01.pdf.

[12] T. Coates, D. Connolly, D. Dack *"URIs, URLs, and URNs: Clarifications and Recommendations 1.0"* W3C Note 21 September 2001 Avaliable from http://www.w3.org/TR/uri-clarification/.