Role-Based Access Control for e-Service Integration

Peter Lamb, Robert Power, Gavin Walker and Michael Compton
CSIRO ICT Centre
GPO Box 664
Canberra ACT 2601 Australia
+61 2 6216 7000
{Peter.Lamb,Robert.Power,Gavin.Walker,Michael.Compton}@csiro.au

ABSTRACT
We present a role-based access control (RBAC) mechanism for a Web Services based data integration system. The RBAC model is extended to allow for role hierarchies and permissions to be project-specific, and the authorization structure is designed to allow custodians contributing data to the integration system to control the authorization over their own data. To address the issue of allowing custodians without IT support to comply with privacy law or ethical standards, a simple access control language amenable to being edited in a GUI is used to express policies. This is translated into XACML for standards-based implementation.

Categories and Subject Descriptors
H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *data sharing, Web-based services*

General Terms
Design, Security

Keywords
Data integration, Access control.

1. INTRODUCTION
The e-Services Integration (e-SI) framework [1], which grew out of earlier work in the CSIRO ICT Centre in the Health Data Integration (HDI) project, is a powerful environment for the integration of Web Services, including Web Services representing databases and other data sources. However, the participation in such integration systems by the custodians of data and services is often only possible when the custodians retain control over the access to the data and services by the users of the integration system. The original e-SI architectural model makes all custodian services available to all integrator users.

This work adds the ability to define projects in e-SI; each user session associates the user of the integration system with a project in the authentication system, and with the user's roles within that project. Sets of permissions are associated with roles in a project, defining the access that users have in the project through their roles.

Access permissions can be defined over the schema elements, tables and columns in data services, and also over the rows of a table visible to a role. Contextual information for the particular user is used for row access control, answering a requirement in the e-Health Research Centre (e-HRC) HDI project for access controls like "view outcomes in *my* hospital" for a specialist (role) registered with a particular hospital.

2. RBAC FOR E-SERVICE INTEGRATION

2.1 Project-based roles
The components of the standard role-based access control model [2] are: *Users* in a many-to-many relationship with *Roles* and *Roles* in a many-to-many relationship with *Permissions*. *Roles* may also be related to other *Roles* to form a role hierarchy as a partial order, with superior roles inheriting the permissions of inferior roles.

Our access control model makes two extensions to the standard model. The first is that role sets may be partitioned by project – the same role name may exist in different projects and have different sets of users and permissions associated with it. It exists in a separate role hierarchy. The second is that there is an inheritance hierarchy over *Permissions* that allows the expression of some necessary relationships – for example that if permission is granted for read access to data in some part of the schema, then access to view that part of the schema must also be granted, since the integration system cannot formulate the queries to access the data without knowledge of the schema.

2.2 Access control architecture
The access control architecture embeds the eXtensible Access Control Language (XACML) [3] Policy Decision Point/Policy Enforcement Point (PDP/PEP) architecture in the e-SI architecture. The PDP (*Authorization*) takes an access request from a PEP (e.g. *Planner*) as a standard XACML *Request*, consisting of *Subject, Action* and *Resource*, and evaluates the request against the XACML policies in the PDP, and returns an XACML *Response* permitting or denying access. The PEP translates user access requests into XACML *Requests* and enforces the *Responses* by permitting or denying the user's actions. Custodians implement an *Authorization* service that resolves *Requests* from the integration *Authorization* service that refer to *Actions* on that custodian's data *Resource*. Integrator policies, enforced by the integration *Authorization* service calling itself with an appropriate *Request*, determine which custodian's *Authorization* service will be consulted to resolve a particular

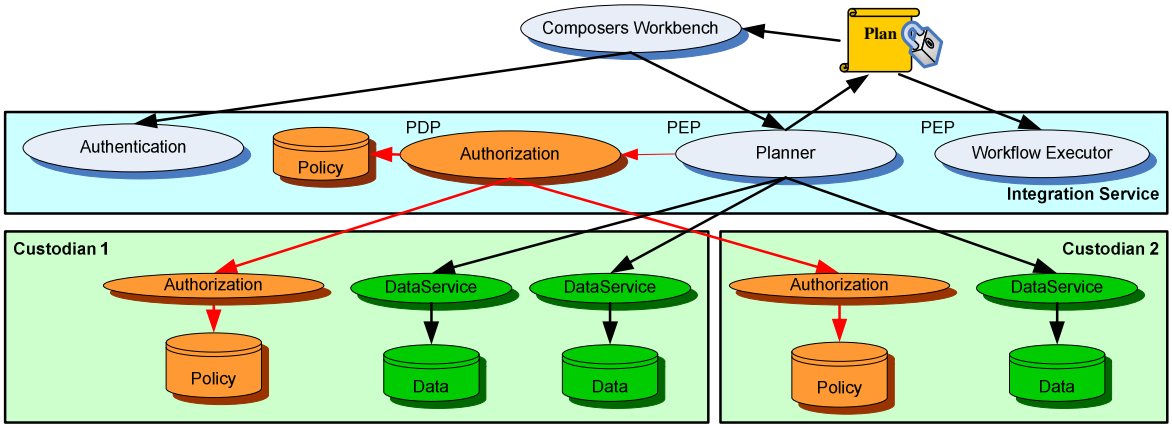

Figure 1 XACML architecture and e-SI query planning

client's access request. Since this requires a *Request* to be made regarding each custodian for every incoming client *Request*, the results of these internal requests are cached for efficiency.

Figure 1 shows the flow of data and control when a user presents a query to the integrator. The *Planner*, acting as the PEP, examines the access requests in the user query, and checks them through the PDP (*Authorization*), and the planning is completed only valid requests. The plan's access requests are noted in the plan and the plan is signed by the planner. Data access is traced through Web Service calls in the query. When the plan is run in the *WorkflowExecutor*, the executor can be assured that the plan is valid, that the plan has not been altered, and it rechecks the access permissions. This ensures that plans are properly re-checked if policy documents have changed since the plan was created, and prevents a plan created using a role with more permissions being misused by a role with fewer. The authorization checking in the *WorkflowExecutor* follows a similar pattern to the *Planner*.

Not shown in the diagram is the *Registry* service, which is responsible for fetching the *DataService* schemas, and passing on to the client only those parts of the schemas that the client is authorized for. Authorization to view the schema is a distinct action from authorization to read data described by the schema. We are also able to control access to database join operations.

Authorization to view the schema inherits from data access and join authorization, so that normally no separate authorizations need be given for schema access.

2.3 Users and Usability

The integration system is intended to support a wide range of data users and data custodians. Custodians, as well as users, may be individual researchers, and so the mechanisms to control access need to be usable by individuals who need to comply with privacy law or ethical standards, and who do not have high levels of knowledge of access control methods.

For this reason, policies are defined in a simple XML-based language that reflects the XACML 2.0 RBAC profile [3] and the access control needs for data integration, and which is translated into XACML. The language is one that is more readily understood and more amenable to editing in a simple GUI than XACML. We have implemented a simple editor which presents the schema elements to the user, allows the role hierarchy to be edited, and shows both the inherited and resultant permissions for the schema elements. The intention is to show the results of access control choices directly to the custodian. The custodian, whose main interest is in knowing what access is being granted, needs no knowledge of XACML and how RBAC is expressed in it, neither is knowledge of our simplified access control language required.

3. CONCLUSION

This work will help extend the applicability of the integration services provided by the e-SI framework, and it is being developed with a view to its use in both the e-SI system and in the e-HRC Health Data Integration System.

4. ACKNOWLEDGMENTS

This work was carried out in collaboration with the e-Health Research Centre, Australia, and funded through the CSIRO Preventative Health Flagship Program.

5. REFERENCES

[1] Ackland, R., Taylor, K., Lefort, L., Cameron, M. and Rahman, J., Semantic Service Integration for Water Resource Management, *The Semantic Web – ISWC 2005: 4th International Semantic Web Conference,* November 6-10, 2005. Springer LNCS Volume 3729 / 2005. 816 - 828.

[2] Sandhu, R. S., Coyne, E. J., Feinstein, H. L. and Youman, C. E. Role-Based Access Control Models. *IEEE Computer*, 29, 2 (Feb 1996), 38-4.

[3] OASIS Standard. *eXtensible Access Control Markup Language (XACML) version 1.0*, http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf, February 2003.

[4] OASIS Standard. *Core and hierarchical role based access control (RBAC) profile of XACML v2.0*, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf, February 2005.

[5] S. Rizvi, A. Mendelzon, S. Sudarshan and P. Roy. Extending Query Rewriting techniques for Fine-Grained Access Control, *2006 ACM SIGMOD International Conference on Management of Data* (*SIGMOD 2004)*, June 2004.