SAML Enhancement of ebXML Registry/Repository:

prepared by
Ed Buchinski, Aziz Abouelfoutouh, Richard Lessard

A Proposition

Great strides have been made by the IT community in developing standards and products to safeguard the authenticity and security of electronic information. For example, the Security Assertion Markup Language (SAML) has been defined to enable systems to exchange identity and privilege information defined by the XML Access Control Markup Language (XACML). The time is right to look at SAML as the means of sharing access management information that is managed by a run time registry service. *This paper discusses how the XACML features of an ebXML registry/repository can use SAML to deliver access control information that is trusted to represent the common understanding of business activities, privileges and resources that have been defined and securely managed by a community of interest (COI).* The salient and unique features of an ebXML registry/repository represent the only mechanisms, that are available today, for ensuring that the access control information is trustworthy and will be uniformly applied by the members of a given COI.

Registry – a Trusted Information Management Tool

In a paper presented at XML Europe 2004, Farrukh Najmi described how an ebXML registry/repository (reg/rep) could provide run time support for content management applications. He described how a reg/rep could provide the metadata about any resource, including information resources, and enable access to the resources managed by content management systems.[1]

This contribution to the ebXML registry/repository Technical Committee proposes that the registry/repository service be extended to support the SAML specification in order to facilitate access control to resources available to a community of interest. The proposition is that the COI, its services, resources and member privileges should be defined in a uniform and trusted manner and that all this information needs to be supported by a secure, registry/repository service that is SAML enabled.

As noted by Najmi, a registry functions as a catalog (i.e. manages metadata that model objects) and “indexes” instances of those objects stored in a repository (either external or integrated with the registry). Thus a registry could include metadata that identifies organizations, individuals, business entities, etc. located in one or more repositories that may be specialized to hold data about particular kinds of resources. Furthermore the registry capability to define “associations” between registry contents allow user, with XACML-controlled administrative privileges, to assign access privileges to resources identified by the registry. This enables entries for individuals to be linked to

[1] http://idealliance.org/papers/dx_xmle04/papers/04-02-02/04-02-02.html

organizations via explicit roles and thereby to specify the business activities that individuals can perform on behalf of the organization. Having the means to define these associations (i.e. privileges defined as roles) and to assign them to specific resources is critical to the implementation of a privilege management infrastructure – a trusted and COI-shared access control capability that has proven difficult to achieve using traditional directory protocols, schemas and technologies.

Role-Based Access Control (RBAC)

As defined by the XACML specification, RBAC is specified as a policy statement. This statement is made up of three elements:

- A business process activity
- A role
- A resource

The policy statement states that a business activity is associated with a specific role and that role may perform a given activity on a particular resource (i.e. business entity such as a purchase order or invoice).   RBAC is realized by delegating the role to a particular organization or individual. Based on this delegation a XACML engine is able to enforce the privileges that the organization or individual is granted to execute a given activity and to access a specific resource.

Figure 1 illustrates the four architecture nodes that make up a XACML implementation. These nodes can be deployed in various configurations:

- PEP – policy enforcement point
- PDP – policy decision point
- PIP – policy information point
- PAP – policy administration point

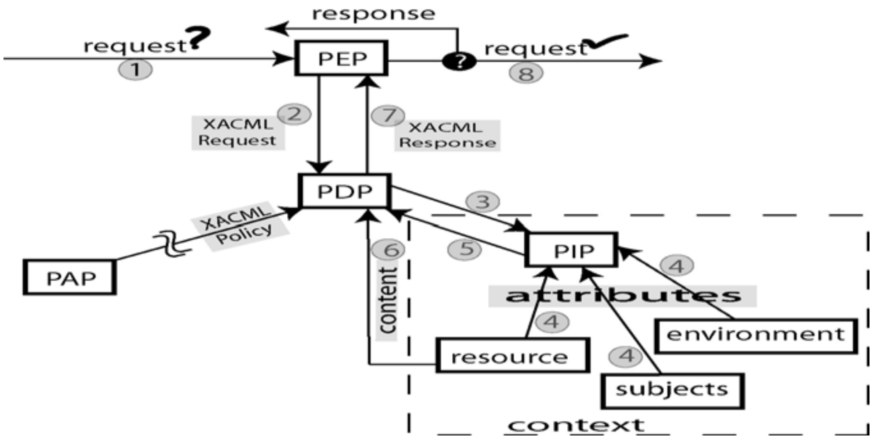

Figure 1 - How XACML Works

Standardized Electronic Business Specifications and RBAC

To ensure that RBAC policy statements are applied correctly within a COI, that community must ensure that business process activities, roles and resources are defined in a uniform manner using common modeling methodologies. The requirements are well document by the international initiative that was undertaken by UN/CEFACT and OASIS in 1999 to standardize the definition of electronic business and to promote its use in international trade. This led to the development of a business process specification schema (actually two separate but related specifications available from each of the founding bodies) for defining business process, roles and business entities (i.e. resources) in accordance with the United Nations Modeling Methodology – a specialization of rational unified process and the Unified Modeling Language. Other modeling methodologies could also be supported in defining and promoting model-driven business specifications.

This initiative also advocated that the electronic business specifications and related artifacts be stored and managed by an ebXML registry/repository.

Enabling Role Based Access Control (RBAC) for a COI

Figure 2 illustrates the kinds of relationships that could be defined and managed in RBAC-based business collaborations. Its development was inspired by a multi-organizational COI that used the ebXML Business Process Specifications Schema and the UMM worksheets to model business processes and concepts (entities) that were pertinent to the community members.

The following notation describes the relations between objects X and Y. There is 1 X for every Y and many Ys for every X →. There can be many Xs for every Y and there can be many Ys for every X. ↔. Note: The elements (attributes) that have been added to the entities (objects) have been added for purposes of clarity of definition only, to indicate the significant primary keys. These are not meant to portray a comprehensive normalized key structure for the entity.

Service Output Types - This is a concept that is used to describe service type patterns and business entity types. It is envisaged to be the primary classification method for business processes and business information entities. One of the Service Output Types is Enforcement. There is a one to one correspondence between service types and business processes.

General Service Process - The processes that make up the pattern for the Service Output Type. Notify Client of Inspection is a process outlined in the Enforcement Service pattern.

***Specialized Service Process* -** The business process pattern as implemented by a specific program or organization. This definition should be described at the same level as the Service pattern. The business process will also be enhanced based on any specialized processing as required by the specific organizations.

Process Activities - A decomposition of the Specialized Service Process to the level of granularity required to apply specific RBAC polic(y)ies. These will include;
ebXML Registry activities (e.g. store new CPP or store new activity description (Initiate notification of Inspection) OR run time business activities (e.g. Notify John Doe of inspection on March 8, 2004 by Officer Jones.

Resource (Data Object) - This includes all data objects stored or referenced by the Registry/Repository.

MOU - Memorandum of Understanding. The agreement amongst the COI organizations regarding the privacy and security properties (context) assigned to the data objects[2]. This is interpreted by specific legislation in the form of statutes. Note: legislation is only one context type that could be assigned. Others include jurisdiction, industry, process and role. These contexts could be defined and managed as classifications schemes in an ebXML registry and used "associates" metadata instances with individual nodes in each classification.

Statutes - Specific Legislation (Acts, Regulations or business policy statements)

Role Types - This entity contains the general information that applies to all roles of this type. When a new role is defined, it will be attached to a role type and inherit all the attributes of the role type.

Roles - This is a specialization of the Role Type for a business process activity and would be "associated " with specific programs, organizations and individuals. The role type attributes will be extended to describe this specialization.

Role Assignment - This entity links a specific subject (User, client, officer, registry administrator, etc.) to a specific role. This effectively provides the subject all the rights and privileges assigned to the role.

Subject - A specific individual in a specific organization.

Policy - The policy provides the rights, privileges, conditions and obligations assigned to a role to perform a specific process activity. (e.g. all drug enforcement officers may perform the activity of << Initiate notification of an enforcement Inspection >> under the condition << a particular jurisdiction >>, and are obliged to << check the registry for prior convictions, a pre-condition identifying a business process activity that would have to be executed first >>.

[2] An MOU is the traditional approach to enabling collaboration. This could be formalized in an Collaborative Protocol Agreement (CPA) or through use of the evolving ISO 15944 –4 specification for Resource, Event, Agent which adds formalisms for modeling agreements, commitments, etc.

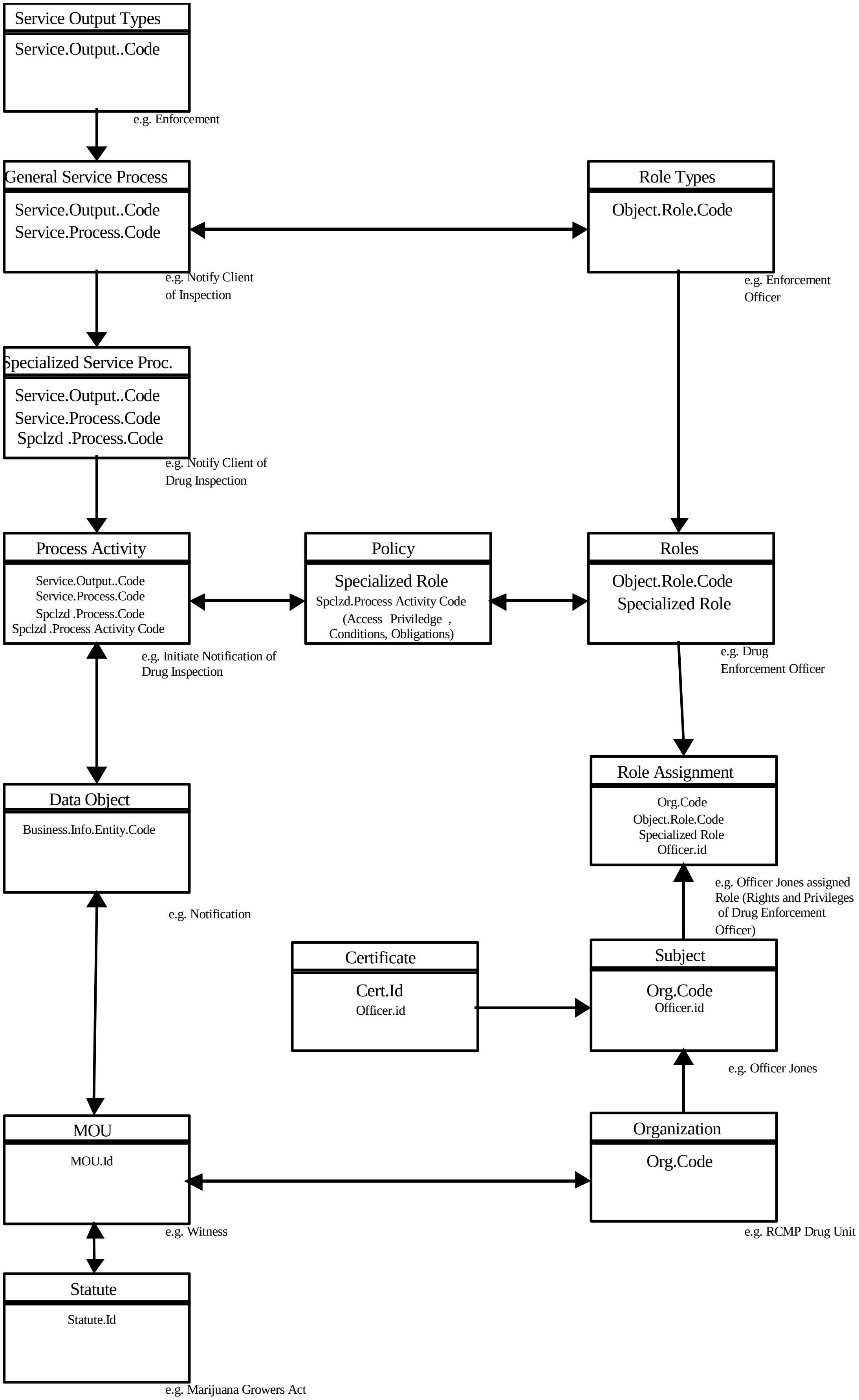

Figure 2 – Sample Objects Supporting RBAC Specification

SAML Assertions

An assertion is a package of information that supplies one or more statements made by a SAML Authority. Three different kinds of assertion statements can be created by a SAML Authority in accordance with the SAML assertions and protocols specification[3]:

1. ***Authentication***: the specified subject was authenticated by a particular means at a particular time.
2. ***Attribute***: The specified subject is associated with the supplied attributes (e.g. Ed is a employee of Treasury Board Secretariat)
3. ***Authorization Decision***: A request to allow the specified subject to access the specified resource has been granted or denied (e.g. Richard Lessard can modify the GoC registry entries for PWGSC)

This SAML specification defines the syntax and semantics for XML-encoded assertions about authentication, attributes and authorization, and for the protocols that convey this information.

The following figure illustrates the relationship between SAML and XACML within a security infrastructure.

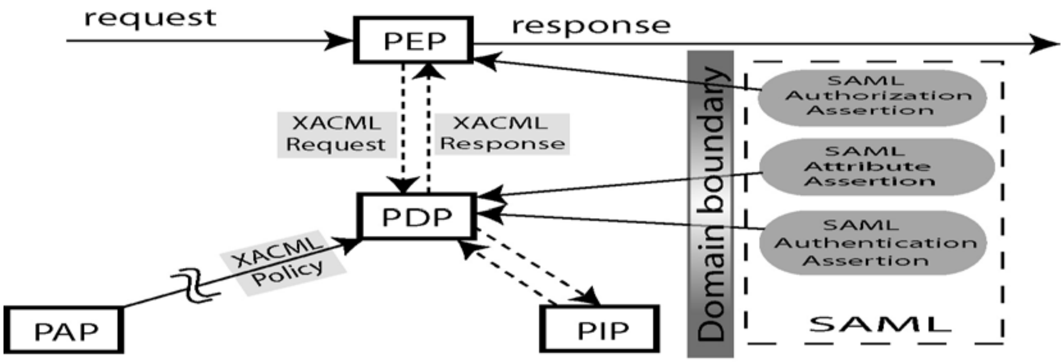

Figure 3 – SAML /XACML integration within a security infrastructure

SAML Profiles[4]

Currently, the core SAML specification is oriented to providing single sign-on (authentication) services and does not address the need for authorization service support.

[3] OASIS. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML). v2.0. (Last-call working draft). July 13, 2004.
[4] OASIS. Profiles for OASIS Security Assertion Mark-up Language (SAML). v2.0. (Last-call working draft. (July 13, 2004.

While there is much convergence on SAML for single sign-on, use of SAML for federated policy management is not yet mainstream thinking. To set the stage for federated authorization services it would be prudent to use SAML, in conjunction the ebXML registry, to achieve federated management for single sign-on services.

The profiles specify[5] the use of SAML assertions and request-response messages in communications protocols and frameworks, as well as attribute syntax for use in attribute statements:

1. One type of SAML profile outlines a set of rules describing how to embed SAML assertions into and extract them from a framework or protocol. Such a profile describes how SAML assertions are embedded in or combined with other objects (for example, files of various types, or protocol data units of communications protocols) by an originating party, communicated from the originating party to a receiving party, and subsequently processed at the destination. For example, a SOAP profile of SAML describes how SAML assertions can be added to SOAP messages, how SOAP headers are affected by SAML assertions, and how SAML-related error states should be reflected in SOAP messages.

2. Another type of SAML profile defines a set of constraints on the use of a general SAML protocol or assertion capability for a particular environment or context of use. Profiles of this nature may constrain optionality, require the use of specific SAML functionality (e.g. attributes, conditions, bindings), and in other respects define the processing rules to be followed by profile users.

Registry Support for RBAC

Using SAML, one could communicate the “privileges or rights” of users (individuals, organizations, automated agents, etc.) to access registered resources. The following figure illustrates the interactions that could be supported by an SAML enabled registry service. It assumes that an identity provider would use a registry to maintain a persistent identifier for a “subject” and the subject’s access rights and privileges. This information would be presented to a service provider in response to a confirmation of the “subjects” rights to exercise a given “role” in accessing a specific resource type.

[5] Ibid. p. 6

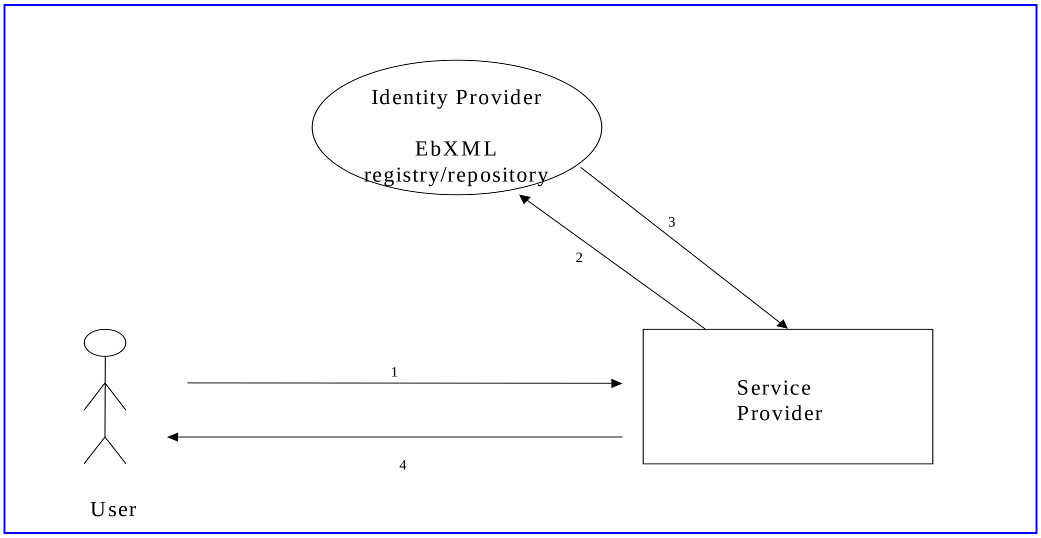

Figure 4 – User Access authorization via Identity Provider Service

In this scenario a "subject" requests access to a resource provided by a service provider (arrow 1). The service provider seeks verification that subject has "authorization" to access service types and associated resources (arrow 2) by submitting a SAML request to the Identity Provider. A SAML response (arrow 3) from the Identity Provider confirms the subject's identity, privileges and the identity of the certification authority that provided the requested assurance information. In this instance, the ebXML registry is providing a XACML policy statement to a XACML engine that operates as a "policy decision point" and supplies the information that is embedded in the SAML response. Arrow 4 indicates that the service provider authorizes the subject to access the requested resource and in so doing is acting as a XACML "policy enforcement point". However, the service provider could also seek confirmation of the subject's privileges from another identity provided, if additional or a higher level of trust was required concerning the subject's credentials.

Proposed Extension ebXML Registry Services

Based on figure 1, it is readily apparent that that a registry could assume the role of a policy information point (PIP) since it could easily serve as a trusted source for the kind of information that is needed to specify access control policies in accordance with the XACML standard. To realize this type of functionality OASIS would need to develop an ebXML profile of SAML. In addition, the registry would need to be enhanced to receive SAML requests and to generate SAML responses. This would enable the registry/repository to provide a number of services including:

1. Verify that a subject (user) was part of a particular audience (community of interest) using the SAML Authentication Request protocol.

2. Specify which identity providers a subject is using with the Web Browser SSO profile.
3. Allow SAML single sign-on service to be used to access the ebXML registry/repository
4. Implement the XACML Attribute Profile and provide “mapping” services that convert SAML attributes to XACML attributes that can be used as input to XACML authorizations decisions[6].

Issues for Discussion

This proposal was prepared without any consideration of the security constraints and vulnerabilities that might be created by placing access control policy information in an ebXML registry/repository. Apart from the security considerations there may be concerns about the registry/repository becoming a performance bottleneck if it is enhanced to work as a PIP. It is assumed that issues such as these would be addressed by the OASIS Registry/Repository Technical Committee when considering the proposal.

Acknowledgement

The authors gratefully acknowledge the comments on the initial draft provided by Dan Demos, Kevin O’Shea and Andrew Robinson. Appreciation is also extended to Ed Simon supplying Figures 1 and 3 and to Dan Demos for the source material for Figure 2. Further consultation was not possible due to the very short time that was available for preparing the submission.

[6] Ibid. p. 47