ACCESS MANAGEMENT IN FEDERATED DIGITAL LIBRARIES

Kailash Bhoopalam, Old Dominion University, Norfolk, Virginia, USA

Kurt Maly, Old Dominion University, Norfolk, Virginia, USA

Mohammed Zubair, Old Dominion University, Norfolk, Virginia, USA

Ravi Mukkamala, Old Dominion University, Norfolk, Virginia, USA

ABSTRACT
With the growth in digital libraries and standardization of protocols for metadata sharing, it is becoming feasible to build federated discovery services which aggregate metadata from different digital libraries (data providers) and provide a unified search interface to users. One of the obstacles that keep data providers, especially the commercial ones, from joining the federation is the lack of an infrastructure to support their business model. Commercial publishers are interested in sharing their metadata with a service provider if they can provide controlled access (who can see what) according to their business model. Driven by the need of commercial organizations (both for-profit and non-profit) to have controlled access to their digital objects and, furthermore, to have that control be inherited by aggregators that federate several contributing organizations, we have implemented a demonstration system based on Shibboleth, OAI-PMH, and XACML. Our system enables contributors to manage controlled access to their information to privileged users using a role-based policy editor. It enables aggregators to reduce their maintenance overhead of dynamic user communities by delegating authentication and credential management responsibility to authorization authorities of the users. Finally, it enables the end users to access information irrespective of their location. This is achieved by providing consistent quality of service irrespective of user location.

KEYWORDS
Access control, digital libraries, remote authentication, Shibboleth, XACML.
1. INTRODUCTION

For the last few years, we have been noticing an exponential growth of online resources, especially due to the widespread use of Internet and increased availability of advanced tools for generating contents. However, a variety of obstacles (e.g., lack of metadata) hamper discovery of such materials and hinder their widespread use. Digital libraries (DLs) address these problems by providing an infrastructure for publishing and managing content so it is discovered easily and effectively. Digital libraries provide cost-effective access to a wealth of information for remote and geographically dispersed users. Digital libraries manage information that can be broadly classified into “full text” and “metadata.” Full Text is the complete information of an item in its original published form and structure. Meta-data is an abstraction of information in the full text such as author’s name(s), publication date, subject, and abstract. In addition, to managing such information, digital libraries provide value-added services such as reference linking, similar subject search, similar authors search, and annotation service.

Non-uniformity among the technologies used by various digital libraries has been one of the biggest obstacles for information dissemination among user communities. The Open Archives Initiative (OAI) has been extremely successful in addressing this interoperability problem. Many digital libraries have decided to become OAI-compliant and expose their metadata to various services including federation and reference linking services. In fact, several services are now available that harvest data from data providers and make the aggregated metadata available to other users. For example, Arc, DP9, Archon, and Kepler (http://dlib.cs.odu.edu) are aggregation services developed by our research group at Old Dominion University.

For the aggregator services to be effective, it is necessary that they can harvest metadata from a large set of OAI-compliant data providers including commercial publishers. One of the obstacles that keep data providers, especially the commercial ones, from joining the OAI framework is the lack of an infrastructure to support their business model. Commercial publishers are interested in sharing their metadata with a service provider if they can control who can see what according to their business model. For example, consider a commercial publisher, American Physical Society (APS), which has contractual agreements with different institutions for their users to have access to the APS metadata at varied levels of granularity depending on the attributes of the users. For instance, a contractual agreement could be such that faculty of institute A has access to all the metadata, and its students have limited metadata access. However, faculty at Institute B has access to metadata and full-text. So APS will be prepared to share its resources only if federation members that harvest (and reharvest) its metadata/data can enforce this policy. The design and implementation of an access control management system for such a federation is challenging, especially when the federation is aggregating metadata from several data providers with different access policies. Building such an information federation involves several tasks including:

1. Designing a policy specification schema and editor to manage contributor’s (one who contributes content to a federation) policy. The editor should hide the complexities involved in the specification mechanism and provide the non-technical (content) managers of the contributors with domain dependent access policy views of contractual agreements with various subscribers.
2. **Developing a policy enforcement framework at the federation resource.** The framework should be based on standard tools, specifications and practices and be applicable to different content domains.

3. **Developing a distributed management system to handle user authentication and role assignment.** The management system leverages user’s home organization authentication system to authenticate its users and then sends information about them to the resource site, which can then make an informed authorization decision. This avoids the problem of resource site managing the dynamically changing user community at different home organizations.

In this paper, we summarize our experience with building a demonstration system and framework that allows for specification of access rules by high-level content managers of the contributors and enforcement of these access rules at an aggregator that federates several collections. The paper is organized as follows. Section 2 provides background of the tools, standards and processes we have used. Section 3 describes our approach. Section 4 describes the architecture of the demonstration system and section 5 shows how the various components of the system are integrated. Finally, we conclude with a summary and future work in section 6.

# 2. BACKGROUND

**Digital Libraries:** Archon [Hesham A. et al. 2003] is an Open Archives Initiative [Carl L. and Herbert Van de Sompel (eds.) 2001, Kurt M. et al. 2001, Simeon W. 2001] compliant federated digital library with an emphasis on physics for the National Science Digital Library. The Open Archives Initiative (OAI) is an international effort focused on furthering the interoperability of DLs through the use of metadata harvesting. The major contribution of the OAI-PMH [Simeon W. 2001] is that it defines a common format for metadata exchange that is independent of the underlying database. We use Archon to provide a technological demonstration of secure federated digital libraries to support a sustainable business model described in the introduction.

**Remote Authentication:** Shibboleth [Scott C. and Marlena E. 2002], Liberty [Thomas W. (ed.) 2003] and the advanced Web Services [Siddharth B. et al. 2003] framework provide remote authentication. When users are affiliated to certain organizations or communities, service providers use remote authentication to distribute the management of user identities so that users are authenticated by authorities that respond soonest to the changes in the membership of the community. This releases the service provider from the overhead of having to maintain user identities current. We use the Shibboleth framework to enable the aggregator (Archon) to delegate user authentication to the user’s home organization. Our system requires only remote authentication and none of the other features provided by Liberty or the Advanced Web Services framework.

**Access Specification:** XACML [Simon G. and Tim M. 2002] is an evolving standard being developed by OASIS and it defines a schema and namespace for the expression of access policies in XML. The access control model used in our system is based on the Role-Based Access Control (RBAC) [David F. and Richard K. 1992] model where access is based on the user’s role (or credential) in their home organization. RBAC simplifies security administration by using roles instead of identities. We chose a generic access specification technology that
came bundled with a policy engine, to simplify the process of access evaluation. Standards like XACML, Object Digital Rights Language [Renato I. (ed.) 2002] and eXtensible Rights Management Language [Xin W. et al. 2002] provide a vocabulary to specify access control rules and also a policy engine that evaluates requests based on the encoded rules. Such policy engines free developers from having to build custom systems.

3. APPROACH

Our approach for building a secure access management system for a federated digital library (e.g., Archon) is based on Shibboleth framework and XACML access specification standard. A federated digital library such as Archon contains holdings from different contributors and supports higher-level services such as discovery, cross referencing, and annotation. Additionally, because of its varied contributors, Archon requires differential granularity of access both in terms of objects and services. This requires a flexible access control management system. Shibboleth framework enables a resource provider (e.g., Archon) to delegate the administration of user identities and attributes to the search user’s origin (or home) site and, hence, freeing the federated digital library from having to maintain user identities. This framework allows users to access information when they are connected to the Internet and use a browser that is Secure Socket Layer (SSL) [John V., Matt M. and Pravir C. 2002] compliant. It also makes the system more responsive to changes in the user community as the origin organization is the first responder to such changes. We have developed a system in which we integrate Shibboleth at the home organization with the Unix password based authentication mechanism. The resource provider (Archon) relies on the authenticated information and uses it to evaluate information access.

To enable Archon to provide differential granularity, we use XACML and its policy engine to express and evaluate policies under which a contributor releases its content to the federation. XACML is a very syntax sensitive specification language and is strictly validated against its schema definitions. Managing XACML in its native format would require knowledge of XACML schema. For content managers, who are typically not technical-savvy, we have developed a high-level policy editor to specify access policies using a “click-and-play” editor.

In summary, the proposed system integrates (a) OAI-PMH based Archon (a DL that federates physics collections) that is part of NSF’s National Science Digital Library (NSDL) project with (b) Shibboleth framework developed by Internet2/Mace that provides for the secure remote authentication and transport of authenticated information and (c) XACML, an emerging XML based standard proposed by OASIS to specify access-control rules.

4. ARCHITECTURE

The proposed system is implemented using the architecture shown in Figure 1. The stakeholders in the system are the search users, the contributors (e.g., APS and CERN), the subscribers (e.g., ODU, CMU, and TWRC) and the aggregators (e.g., Archon).
Users: Users are individuals seeking information online from a digital library and are generally required to provide their identity or their rank (role or affiliation) in their home organization to access privileged or value added information from the digital library.

Contributor: A contributor is the source of digital information in the federation. A contributor could be an individual author, an online preprint archive, or a commercial publishing organization.

The contributors and the users are the stakeholders who are at the two ends of the digital library spectrum. The intermediary stakeholders provide the following value in a digital library.

Figure 1. Architecture Diagram

Subscribing Institution (and Shibboleth Origin): A subscribing institution is an organization that enters into contractual agreements with contributors to enable its members to access information published by the contributors. A subscribing institution provides end-users (i.e., its members) who may individually not have the capability (monetarily or technologically) to access privileged information published by the contributor. To provide secure access to information, the subscribing institution authenticates its members (i.e., users) when they seek to access digital library information. After authentication, it transmits the user’s authenticated information and the user’s attributes to the aggregator so that the aggregator can make an access-control decision about what information it can provide to the user. To support this functionality of secure transport of authentication information, the aggregator uses a Shibboleth Origin.

Aggregator (and Shibboleth Target and Federated DL): The aggregator assists the contributor in supporting their subscription models and in making their information available.
to a wider target audience. By doing so, it increases the sustainability of the contributors. The aggregator exposes its content and services under the policies specified by different contributors. For authenticating users and transmitting their attributes, it uses Shibboleth framework. To manage contributor’s policies specified in XACML, it uses a high-level policy editor and a policy enforcement engine.

The following are the major software components of our proposed system.

**Registration Server:** This is a graphical user interface that allows contributors to register their intent to provide information to the aggregator. Registration requires the contributor to list its subscribing institutions, the metadata it provides, and the services the aggregator should provide.

**Policy Editor:** This is a graphical user interface that allows contributors to specify fine-grained role-based access policies for each of its subscribing institutions. The policy editor provides a domain dependent managerial “click-and-play” abstraction of the details of the access policies, these abstractions are then translated by the system into XACML. These access policies represent the contractual agreements between the contributor and the subscribing institutions.

**Federated DL Store and Harvester:** After the contributor registers at the aggregator, the harvester gathers digital information at regular intervals from the contributors using OAI-PMH and stores the information in the federated DL store. The DL store optionally normalizes the information gathered to provide enhanced search services. The aggregator also contains the policy decision point and the policy enforcement point that enables it to provide secure access.

**Policy Decision Point (PDP) and Policy Enforcement Point (PEP):** The PDP [Simon G. and Tim M. (eds.) (2002)] is a software package that evaluates access requests against an XACML policy. The PEP [Simon G. and Tim M. (eds.) (2002)] converts domain dependent requests to XACML-compliant requests and submits them to the PDP for evaluation. The PDP receives access requests from the PEP and also responds to the PEP with its access decisions.

Figure 1 shows two distinct interaction sequences. In the numeric sequence, the user seeks to access privileged information that is protected by Shibboleth and the access control mechanism. Upon such a request, the Shibboleth “Target” at the aggregator and the Shibboleth “Origin” at the users home organization (also a subscribing institution) collaborate to provide the aggregator with the required attributes of the user, after which, the aggregator performs an access evaluation using the PDP and PEP and subsequently serves or denies the user’s request. In the alphabetic sequence, the contributor registers with the aggregator and encodes the access policies using the editor. After the policies are encoded, they are submitted to the policy engine at the PDP where they are evaluated against user’s requests made in the numeric sequence. These interaction sequences are the enabling mechanism towards the development of a secure digital library framework.

### 5. IMPLEMENTATION

As described in the architecture section and in Figure 1, the two interaction sequences towards enabling a secure digital library are: (1) User’s access to privileged information, and (2) Contributor’s registration and access control specification. Enabling (1) requires the
integration of the Shibboleth Target with the aggregator and the Shibboleth Origin with the subscribing institutions and also the integration of the XACML compliant PDP and the PEP with the aggregator. Sections 5.1 and 5.2 describe how this was implemented. For (1) we follow the specification of Shibboleth framework [Scott C. and Marlena E. 2002]. Enabling (2) requires a contributor registration system and a policy editor that can be used to specify and manage access policies by non-technical content managers. Section 5.3 describes the design of the policy editor and how it achieves the goals of customizability and ease of use.

5.1 Integrating Shibboleth with Archon

The Shibboleth v1.1 that we employed, in our implementation consists of the origin written in java with the target and the “Where Are You From” (WAYF) module written in the C language. The user’s attributes are exchanged between the origin and the target using Security Assertion Markup Language (SAML) [Scott C. et. al. (eds.) 2005] and Simple Object Access protocol (SOAP) [Martin G. et. al. (eds.) 2003]. Prior to the exchange of attributes, the origin and the target authenticate each other by exchanging and verifying each other’s certificates. This section elaborates how the origin and the target have been setup in the prototype and how cryptographic certificates are used to enable trust between them.

Integrating Shibboleth at the users Home Organization: The Shibboleth origin uses servlets to accept attribute requests and to post responses. It is installed in an off-the-shelf IBM-compatible PC with Redhat Linux 8.0, apache web-server v 1.3.27, and the tomcat servlet container v1.4.26. Openssl and mod_ssl were used for secure communication. We chose these release versions because they were the latest that were supported by the shibboleth v1.1 package at the time of release and additionally because documentation and support were available in good measure for these versions. Using Openssl tools, we created certificates for the origin machine so that it can authenticate to the user and the Shibboleth target. The certificates were created using the java keytool and signed using tools provided in Openssl. We explain the mechanism of establishing trust and certificate creation in greater detail later in this section.

The apache web server in the origin is configured to authenticate the user by presenting the user with a username and password challenge. If the user is successfully authenticated, the origin and the target authenticate each other and the origin sends an opaque handle---where the identity of the user is hidden---for the user to the target. The origin later receives the user’s handle along with a request from the Shibboleth target for the additional attributes about the user. In the prototype, the user accounts are grouped into one of three groups--- student, staff and faculty. The groups are represented as Unix based Group Ids. We have developed a plugin for Shibboleth that converts groupids of authenticated users into the corresponding namesake roles (i.e., groups staff, student, faculty). These roles are embedded as SAML assertions by Shibboleth and sent to the target in a SOAP envelope.

Integrating Shibboleth at target (Archon): The Shibboleth target package is similar to the Shibboleth origin package (a C based web enabled application and is installed in an off-the shelf IBM compatible PC with Redhat Linux 8, apache 1.3.27, openssl and mod_ssl). Though a servlet container is not required for Shibboleth, it is required for Archon, a servlet based digital library aggregator.
When a user accesses Archon using a browser, he is assigned the role of a guest and is eligible to the privileges that each contributor specifies for a guest. The privileges specify what the user can view and what services he can use. If the user is from an organization, which has a contractual agreement with any of the contributors (and hence, with the target, Archon) of Archon, the user may “login” to Archon in order to use the privileged features provided by Archon to members of these organization. The “login” into Archon is implemented as a user access to a web page that is protected by Shibboleth. This web page is specified as being protected in the apache configuration file.

When the user chooses to login, he is redirected to the WAYF. The location of the WAYF is specified in the Shibboleth target configuration file Shibboleth.ini. The WAYF is provided with the target package of Shibboleth but may be installed in any computer. We have installed the WAYF in the same machine as the origin (there is no apparent reason for choosing the origin over the target or a separate machine). The WAYF is configured with a list of origin sites. When the user chooses his home organization, he is redirected to the home organization, which presents the user with the challenge as explained earlier. During the Shibboleth handshake, the origin and the target authenticate each other and the target gets possession of the user’s credentials (role of the user in his home organization.) In this entire process, the user is first redirected from the target URL to the WAYF, then to the origin site and then back to the target URL to where the user chose to login. The POST based web redirection ultimately results in the target having the users credentials as HTTP request attributes. The policy enforcer explained later in 5.2 uses these user credentials to provide contributor specified access privileges to the user.

Enabling Cryptographic Key-Based Trust between the user’s Home Organization and target (Archon): The home organization and the Shibboleth target at the aggregator authenticate each other prior to the request and provision of user attributes. The target authenticates the home organization to ensure that it receives user attributes from a recognized and valid attribute authority of the requesting user. The home organization authenticates the target to ensure that it reveals the user information only to organizations with which it has contractual agreements. The development of trust between organizations is a non-technical issue that is specified in legal contractual documents. The contents of the document contain agreed upon privacy and attribute usage processes and practices of participants in the contract.

Once the legalities involved in establishing trust is complete, participants in the contract share and verify cryptographic identity certificates to manage trust. To support verification of certificates, the target and origin have to possess certificates signed by the same certificate authority (CA) or signed by other certificates in the chain leading to a common CA. Shibboleth allows for the specification of such a certificate chain. In our implementation, we created a self-signed certificate to create a CA, and signed the origin and target certificates with this CA. The origin and target store each other’s certificates and the chain leading to a common CA in a certificate file. This certificate file is available at the origin and the target. When the target and the origin present their certificates to each other, they are verified to check whether they are valid (not modified, not expired and not revoked).

After mutual authentication is complete, the attribute authority at the origin first presents an opaque handle of the user to the target, subsequently when the target requests attributes related to the opaque handle, the attribute authority at the origin presents the signed user attributes to the target. The entire protocol involving user authentication and subsequent provision of user attributes occurs within a few seconds. The clock skew between the origin and the target machines is set low to prevent replay attacks.
5.2 Policy Enforcement Point–Policy Decision Point Interaction

Archon provides users with metadata of technical articles like author information, description of articles, date of publication, subject etc, and services like search on similar subjects, equation extraction, annotation, etc. Figure 2 illustrates how information and services are categorized in Archon. Digital libraries may also provide access to the full text of articles, which may be available as a separate category of information or through a service. In our test bed, we have not implemented a protection environment for full text, however, it can be implemented with no change to the access control framework.

Archon extends the benefits it provides to its contributors by providing suitable access control mechanisms to control access to portions of the metadata or to certain services for articles it hosts. Figure 3 illustrates how the components of the protection system at Archon, namely the Policy Enforcement Point (PEP) and the Policy Decision Point (PDP) interact to provide secure access to metadata and services. The rest of the section elaborates the pre-requisites for access control enforcement and how access control is enforced during user request.
Prior to exposing the resources for potential users, it is necessary for the PEP and the PDP to be aware of what resources (metadata and services) are being subject to access control, against whom the resources are protected and how the resource are protected. The resources that Archon needs to protect for various contributors are specified as configuration files that are prepared by the registration system when each contributor registers with Archon. The registration system at Archon provides a master set of available services and metadata among which contributors can choose to make their content available. Each contributor’s choice is stored in an XML configuration file as a list of chosen metadata and services that need to be protected and the list of user roles that the contributor recognizes. These configuration files specify information about what needs to be protected for each contributor and the user types or roles that are of interest to this contributor. These configuration files are used by the policy editor to provide appropriate views with the just the relevant resources and roles to content administrators. The content administrator of each contributor uses the policy editor to articulate the access control policies for roles of each of its subscribers on the contributor’s resources. The features of the policy editor are described in section 5.3, for this section it suffices to realize that the policies created using the editor are stored in the XACML format and are used by the PDP to evaluate access requests. The configuration files and the XACML policies provide the PEP and PDP with sufficient information to enforce access control on resources it hosts.

When the user’s attributes are available at the target, the PEP fetches the user’s attributes, and for each metadata and service it embeds the user attributes in an XACML compliant request and submits it to the policy engine. The PDP evaluates the requests and for each request it responds with a decision encoded within the response. Hence if there are ‘X’ metadata fields and ‘Y’ services offered by Archon for the registered contributor, the PEP generates X + Y request to the PDP and stores a similar number of responses. If the home organization of the user is a subscriber to N contributors at the aggregator, the total number of requests is N * (X + Y).

Figures 3 and 4 list technical items (articles or journals) that are retrieved by Archon from the contributor archives based on a search criterion submitted by a guest user and a privileged user respectively. By clicking on an item in the list, the user is shown additional information (metadata and services) about the item in a separate pop-up window. The information shown to the guest user (popup of Figure 4) is very little compared to the information available to a faculty of a hypothetical institution test1 (popup of Figure 5.) The information that each type of user is allowed to view is specified as access-rules using the policy editor explained in 5.2 and enforced by the PEP-PDP interaction explained earlier in this section.
5.3 Policy Editor

The policy editor provides an interface for non-technical content managers to specify access policies for the digital libraries services and metadata based on the Role-Based Access Control (RBAC). We have chosen XACML for access specification as it provides a flexible and scalable framework for the specification of access rules and the evaluation of access decisions. XACML is a syntax-sensitive language that is strictly validated against its schema definitions. Hence, it is unreasonable to expect a non-technical content administrator to manage XACML in its native format. The policy editor we have developed empowers contributors’ content administrators with a simple and yet powerful tool to manage access. This section describes how the policy editor assists the content administrator to specify consistent rules and how it reduces errors in policy specification by providing customizable views of the policy and the stakeholders. It also describes how the rules are translated and organized as XACML policies.

Customizable Views: Different subscribing institutions may choose to expose different roles based on the services they require from a contributor. The contributor specifies these roles for each of its subscribing institutions when it registers with the aggregator. The policy editor we have developed uses contributor’s registration information to generate views to the content administrator that are relevant to each subscriber. In the absence of such views, administrators would have to rely on other external mechanisms to obtain relevant roles or perform non-intuitive role mapping to a restricted set of predefined roles. This feature reduces errors in specifying access policies and also reduces the time required to manage access rules.

Consistent Access Rules: The matrix shown in Figure 6 enforces consistent specifications of rules. Using the matrix view, it is impossible to specify inconsistent access rules where in one rule a certain resource is made available to a role and in another rule it is denied access. By using the matrix view to prevent inconsistencies, we have overcome one of the limitations of XACML.
Structuring Access policies: Figure 6 is a screen shot of the policy editor that provides views of the access policies for a guest user who is not affiliated to any subscribing institution and access rules for roles of a hypothetical institution (test1.edu.) The columns namely, “faculty,” “staff,” “student” and “guest” allow the administrator to specify access rules to the metadata listed in the first column (e.g., archive, identifier, and creator) for members of the roles listed in the first row.

When the manager clicks on a box next to a metadata field, the interpretation is that this field is to be accessible to a member of the respective role. The guest, though not affiliated to any subscriber is replicated in the subscriber specific views for reference. Using the editor, the contributor specifies one XACML policy for each of its subscribing institutions and one for the guest. Each access policy is a collection of rules, where each rule represents the access allowed for a role to a set of resources or services. Hence, if N is the number of roles and M is the number of subscribing institutions, the number of XACML access rules created by each contributor is of the order O(N*M). The manager only interacts with the interface while the system generates the XACML rules.

6. RELATED WORK

There are few other projects that have implemented distributed authentication and authorization and controlled access by multiple authorities. Each of these systems has a unique approach that concentrate on either authentication or access control. We have brought together similar technologies and have used them along with XACML to support a business model that allows for the sustainability of digital libraries.
The Community Authorization Service (CAS) [Laura P. et. al. 2002] reduces the administrative overhead by separating resource administration from the administration of user’s identity and organization-role administration. In CAS resource administrators provide bulk rights to a particular community, and a user accessing a resource is provided a capability certificate based on the user’s identity or the user’s group identity. The resource manager enforces the privileges encoded in the capability certificate. CAS uses proxy certificates and a proprietary policy statement format. We have used the XACML standard as an alternative and more viable solution. Additionally, our system can respond to changes in authentication and authorization more quickly as they are handled by the immediate stakeholders and not delegated to a third party.

Akenti [Mary R.T. et. al. 2003] addresses issues raised when multiple authorities control access to resources. Akenti provides a policy language, as well as infrastructure components to enforce, flexible access control policies. Akenti uses proprietary XML schemas to construct policies for resource hierarchies and requests. The Akenti team is working on XACML based policy specification. The results of the Akenti team would be an excellent complement to the access specification of digital resources in our future work.

7. CONCLUSION

Driven by the need of commercial (for-profit and non-profit) organizations to have controlled access to their digital objects, and, furthermore, to have that control be inherited by aggregators that federate several contributing organizations, we have implemented a demonstration system based on Shibboleth, OAI-PMH, and XACML. We have created a secure version of Archon that federates collections from CERN, Los Alamos National Laboratories, and APS that allows users from different organizations access with different granularity level. Performance of the secure versus the non-secure implementation is comparable. For guests that see only minimal metadata and no services, there is practically no degradation in performance. Login takes about 10 seconds and is the most time consuming aspect of the system as it involves a number of interactions between the home and the target organization. Display of the search results takes slightly longer (on the order of 10%) as the enforcement engine has to check the rules before it displays (or not) metadata and services. The changing of existing and creation of new rules is almost instantaneous as they are done at the guarded resource directly. As any search uses stored configuration files that are modified by such editor changes, users will see the impact of the changed rules immediately. What we have not yet determined is either the acceptability of the editing interface to managers or the willingness of subscribing institutions to install a Shibboleth server at their home organizations to interact with the target Shibboleth server. We plan on a field test with APS and selected organizations subscribing to the digital library of APS. Only such a field test will show if this model’s advantages outweigh these costs of installation and maintenance.
REFERENCES