An Assessment of the VOMS and GridShib VO Management Systems

David Aikema, Cameron Kiddle and Rob Simmonds
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
{aikema,kiddlec,simmonds}@cpsc.ucalgary.ca

February 12, 2007

1 Introduction

In the past, high performance computing consortia in Canada have received separate grants for resources. Furthermore, a consortium's resources have been primarily intended for users local to member institutions of the consortium. Recently, the seven high performance computing consortia in Canada worked together on a successful single CFI (Canadian Foundation for Innovation) grant proposal, called the National Platform Fund. While users will still be expected to try to meet computing needs from their local consortium first, there will be a greater emphasis on the national sharing of resources. With a much larger user base nationally, mechanisms for authenticating users and authorizing access to resources need to be explored.

Currently the users registering with a consortium may receive an account with the same login name and password on all of the consortium's resources to simplify administration. The user may not make use of all of the resources, but the user can easily remember their login name and password and get access to any of the consortium's resources that they need. While creating accounts for all users on all resources at the consortium level has been successful, it is not really practical to do this at a larger scale. Requiring users to register with each consortium separately is also not practical. That approach would require each consortium to separately verify the identity of each user. It may also require users to remember multiple usernames and passwords. A visiting researcher may spend a significant amount of time just getting accounts on the resources that their research group has access to.

X.509 certificates [22] and the Grid Security Infrastructure (GSI) of the Globus Toolkit [7] can be used to aid in authentication and authorization. A user's certificate, signed by a certificate authority such as Grid Canada [13], gives the user a global identity. Sites that trust the certificate authority can validate the authenticity of a certificate and map a user to a local account if authorized. Users need not worry about remembering login names and also have single sign-on capability where a password (for the certificate) only needs to be entered once. While single sign-on can also be accomplished with SSH keys, the SSH key approach still requires users to remember login names, requires the user to copy SSH keys to all of the sites and can pose a greater security risk.

Additional mechanisms are still needed to aid in deciding who is authorized to use resources and to deal with aspects of user account management such as creation and deletion of accounts. Solutions have and are being developed to provide more efficient handling of accounts. These solutions are built around the concept of a *virtual organization (VO)*. Many definitions of a virtual organization exist, but for our purposes we will consider a virtual organization to

be a group of individuals that are working together on a common project or that are entitled to use a particular class of resources (e.g., users entitled to use government-funded HPC resources). A virtual organization does not need to provide any resources of its own, but instead negotiates with resource providers to gain access to resources. Grid computing projects such as Enabling Grids for E-Science (EGEE) [4] and the Open Science Grid (OSG) [31] make use of virtual organizations. Any user that wants to make use of either the EGEE or OSG resources must first belong to a virtual organization. There are virtual organizations covering many disciplines and projects [32, 5]. A user is authorized to use a subset or all of the resources that the VO is authorized to use.

This document proceeds by discussing management of VOs and the goals that we want to be achieved by VO management tools in Section 2. A survey of various VO management tools is then given in Section 3 followed by an account of our experiences with these tools in Section 4. This account includes an assessment of these tools based upon their availability, ease of installation and ability to achieve the goals discussed in Section 2. The document finishes in Section 5 with a conclusion summarizing our findings .

2 VO Management

Management of a virtual organization involves controlling who the members are and specifying their roles and privileges. VO management tools and services aid administrators of VOs in achieving these tasks. They also aid resource administrators in managing and mapping accounts for VOs that are authorized to use the resources.

Virtual organizations can be created and managed in a variety of ways. A virtual organization could consist of researchers from around the world that are collaborating on a specific project such as the various virtual organizations affiliated with the LHC Computing Grid project [25]. Users of a particular application, such as GROMACS [26] or Gaussian [9], could be grouped together as a virtual organization with authorization granted on resources where the application is supported and which have the necessary licenses. Another possibility would be to group members of a consortium together as a virtual organization. Other consortia could grant access to members of a consortium's VO as they deem fit.

An important aspect to consider is trust. Resource providers must not only trust the certificate authorities that issue user certificates, but they must also trust the virtual organizations. As the virtual organization decides who its members are, the resource providers must trust that the virtual organization confirms the identity of its members and their right to be a member of the virtual organization. They must also trust that the VO will terminate the membership of any users who no longer have the right to be part of the virtual organization, or are not abiding by rules and regulations set forth.

In the remainder of this section we discuss goals that we want to be achieved by VO management tools and services. The goals are discussed in terms of three different functionalities: adding users, assigning roles and removing users.

2.1 Adding Users

Consider a collaborative project involving research groups from multiple institutions across Canada. Project members are authorized to use various clusters from several high performance computing consortia as well as resources owned and managed by the individual research groups.

Currently when a new member joins one of the research groups participating in the collaborative project, acquiring access to all of the available resources is often a tedious process. The new member must first determine all of the resources that they are eligible to use, and then must contact each of the resource providers separately to apply for accounts. This requires the involvement of administrators at each site. Each site must also verify the identity of the user which would likely involve contacting the corresponding research group leader. The time to acquire all accounts could be considerable and requires involvement of many people. If the new member is just visiting or participating in the project for a short time period, such as the summer, then a large portion of the available time could be spent just in acquiring accounts.

With the virtual organization approach, the new member would only need to apply to become a member of the virtual organization. Once a member of the virtual organization they would automatically have access to all of the resources that the virtual organization is authorized to use, subject to having an appropriate role. This would significantly reduce the amount of administrative overhead and likely decrease the time required to get access to resources. This

also simplifies the process of providing VO members access to new resources in the future. The resource provider of the new resource would only need to authorize access to the VO and all its members would automatically have access without everyone having to apply for new accounts.

VO management tools should facilitate the addition of new members and allow for automatic mapping of members to accounts on the authorized resources. The addition and automated mapping should be done as described in a contract established between the VO and resource providers, without additional administrative involvement. In other words, the VO and the resource provider should only have to setup an initial agreement governing use and how users should be mapped to accounts. After this point, a new member of a VO should automatically be mapped to an account the first time they use a resource authorized for use by the VO. This should not require any further negotiation or account setup with the resource administrator.

2.2 Assigning Roles

Within a particular virtual organization users may take on one or more roles. These roles could be used to distinguish subprojects a user may be working on. They could also be used to indicate different privileges of the user. Based on the role, the user may be mapped to a different user account.

An example where a role could be used to restrict access to resources is as follows. All Canadian researchers in a particular virtual organization may be granted access to all of the resources in Canada that the virtual organization is authorized to use, whereas international researchers in the virtual organization might be granted access to a more limited set of Canadian resources.

An example where a user may need to be mapped to a different user account based on role is as follows. A user may have a role for running production experiments and also a role for testing. These roles may map to different user accounts for accounting purposes. The production account may have a higher priority and be allowed a larger resource allocation. The test account may not have special priority and receive a default resource allocation. Note that it could still be possible to accomplish the same thing by mapping the user to the same account but to a different accounting group.

VO management tools should allow virtual organizations to assign roles to users and should allow users to specify the role they are acting in. They should also enable resource providers to differentiate between user roles.

2.3 Removing Users

Turnover of members in a VO can be high due to students or visiting researchers that only participate in the project for a short duration. It is important that members no longer get access to resources after they have left or if they are not abiding by the rules of the VO. Keeping track of which users are still entitled to access a resource can be difficult for individual resource administrators, particularly if the users are from a different institution. With VOs, only the VO, which is likely more familiar with its members, needs to keep track of the status of its members. Once a member no longer has the correct status, the VO can remove the member. This will automatically prevent the member from gaining access to resources authorized for use by the VO.

In the case of malicious users, the certificate authority also has the ability to revoke the user by adding the user's certificate to its certificate revocation list. Note that this will prevent a user from accessing any resources using this certificate and not just the resources authorized for use by the VO.

VO management tools should have mechanisms to remove a member such that they no longer have access to resources authorized for use by the VO.

3 Survey of VO Management Tools

The de facto standard middleware in the grid computing realm is the Globus Toolkit. By default it utilizes a flat text file called a gridmap file to map users. When a user attempts to authenticate to a system using this authorization method, the system validates the user's identity by checking whether or not the certificate provided to the system has been signed by a recognized and trusted CA. With a stamp of approval from the CA, the system knows that the user is who the distinguished name (DN) of the certificate declares them to be. The system goes on to compare the certificate's

DN with a list of DN-to-account mappings in a flat text file to associate the user with an account provided that such a mapping exists. This method of mapping users works on a small scale reasonably well. However, as the quantity of resources available and/or the number of users increases, or user mappings become more dynamic, this method becomes more difficult to manage.

Beginning in version 3.2 of the Globus Toolkit, some extensions were added to allow 3rd party code to provide authorization services. Such code now exists for both the pre-Web service [11] and Web service [12] versions of Globus. These extensions allow new authorization infrastructure to be plugged into an installation of the Globus toolkit merely by editing a few configuration files to point to the new module. There is no need to recompile any of the toolkit.

A number of different projects have attempted to create tools utilizing the virtual organization concept which augment the Globus Toolkit's authorization mechanisms using the means described above. Some of these tools are presently being utilized, while others are still in development. A set of tools centred around an application called VOMS offers one approach to security and has seen production usage. The GridShib approach attempts to extend Shibboleth to the grid environment. Shibboleth is a framework originally intended for browser-based authentication and authorization for Web resources. There are a number of other related projects which will be discussed briefly, but the frameworks based around VOMS and GridShib have been the primary focus of investigation and will thus be the focal points of this paper.

3.1 VOMS Approach

One approach to explicitly incorporating the idea of virtual organizations into the authorization infrastructure combines the efforts of several projects. The list of software tools utilized in this approach includes VOMS [1], GUMS [20], and PRIMA [36]. For the sake of simplicity and brevity, future references to VOMS as a framework should be taken to include these other components as well.

VOMS, short for the Virtual Organization Membership Service, provides a database in which the membership lists of virtual organizations may be maintained. It also allows administrators to define subgroups and roles held by various members within each virtual organization. If they have the VOMS client software installed, users who are members of the virtual organization myVO may execute the following command when initializing a proxy:

```
voms-proxy-init -voms myVO
```

This command allows them to obtain an X.509 proxy certificate which is backwards compatible with unmodified Globus installations and has their VO-membership attributes for the specified VO appended to their certificate. When contacting a resource which is running PRIMA, the resource will then pass this information on to the GUMS server, allowing an authorization decision to be based on VO-membership information.

GUMS [20] was developed at Brookhaven National Laboratory (BNL), and it attempts to provide centralized management of user authorization at resource providers. Each of these resource providers may provide many resources with differing access control policies. The means through which it accomplishes this goal is by giving a trusted server control over all user authorization decisions. This service has been in production usage at BNL since 2004.

A configuration file must exist on this central server, containing the account mapping information for all services which rely upon it for an authorization decision. The configuration file does not require each client machine to have its own separate configuration. Rather, client machines are mapped to configuration information either by examining their hostname or the DN of their certificate, and the usage of wildcards in these mappings is permitted. By using different certificates, a single host can also be associated with multiple security configurations for each of its different services.

There are a number of ways in which GUMS can map certificates of users to accounts. The system supports not only basic one-to-one mappings, but also group accounts and account pools. One-to-one is where a user is mapped to a single account, and the account is also only mapped to by the one user.

Group accounts allow multiple users to access a single account, but there is no corresponding mapping back from an account to an individual user. This may limit auditing as multiple users may be utilizing the same account at a given time. If sufficient auditing information is kept, however, a user might be traced back to their original login using process numbers. If using a group account, all involved in the physics experiment BaBar could be assigned the same account, BaBarAccount.

Account pools retain some of the advantages of grouping by virtual organization while at the same time providing a unique inverse mapping of account to user. On a system utilizing account pools, administrators set aside a range of unallocated accounts. When a user connects to the system for the first time, if authorization rules map them to the account pool, then one of these unallocated accounts will be set aside and a permanent mapping created tying this to the certificate. When this same user comes along again later they will receive the same account as they were allocated the first time. If the same hypothetical organization defined in the previous paragraph used account pools instead of a group account, system administrators at a resource site might set aside a range of accounts — BaBarAccount01 - BaBarAccount99. The first time that the hypothetical user Bob logged into this resource, he would be allocated the first unallocated account remaining - e.g., BaBarAccount43. Each time in the future that Bob sent a job request to this same resource provider, he would receive the same account - BaBarAccount43. The reverse mapping also exists, unlike the group account case. Any activity by the user account BaBarAccount43 is traceable back to Bob.

GUMS supports two different authorization mechanisms - gridmap files and PRIMA - the latter of which provides more flexibility. Partly as a way of providing backwards compatibility, GUMS is able to generate gridmap files based upon information retrieved from VOMS. When it does not have the PRIMA module installed, a client machine can install a cron script which contacts the GUMS server to download gridmap files generated according to the GUMS configuration. Thus, even using the gridmap file approach, some use may be made of VOMS. However, using the gridmap file mechanism requires that all potential users have accounts preallocated. Users also can't take advantage of roles when the resource utilizes gridmap files.

PRIMA [36] is the plug-in authorization module utilized by GUMS to perform on-demand authorization for machines running the Globus Toolkit. The plug-in was originally developed as part of another authorization framework [28]. This other framework does not appear in widespread use today, and so GUMS seems to be the primary user of the PRIMA authorization module today.

3.1.1 Overview of an authorization

Figure 1 outlines the interactions occurring between the user, the virtual organization and the resource provider when using the PRIMA plug-in module. The user first contacts the VOMS server operated by the VO to initialize a proxy which contains a signed assertion testifying to membership in the virtual organization. This proxy is then passed to the resource when the user attempts to access it. The PRIMA module at the resource contacts the GUMS server for the site to get a mapping. The GUMS server utilizes a locally cached VO-membership list retrieved from the VOMS server to verify membership. Finally a username is returned to the resource, and the user gains access. If using gridmap files generated by a cron job, when a user attempts to access a resource the resource looks no further than its own gridmap file. This gridmap file is updated at a regular interval, at which point the interaction with the GUMS server and VOMS takes place.

3.1.2 Addressing goals

There are two ways in which a user can be added to the virtual organization in VOMS. In the first case the administrator manually adds the user's certificate DN to the VO-membership database using voms-admin. Alternately, the user can request to be added to the VO by using the Web based voms-admin interface. After verifying their email address, this request is then passed on to the administrator for further verification. A decision can then be made whether or not to approve their application. A member can then attach a membership assertion to their X.509 certificate using

```
voms-proxy-init
```

and then authenticate to resources as a member of the virtual organization.

When initializing a proxy with a VOMS assertion attached, the user can specify that a particular role that the VO grants them permission to assume be appended to their assertion. In the example above, a user could be permitted to assume production and test roles. GUMS, when it performs the mapping can check the different roles and map the user differently according to the role which they suggested. Different roles function only when using the PRIMA plug-in authentication module, and do not work if authentication is done using gridmap files generated by the GUMS server as the result of a cron job.

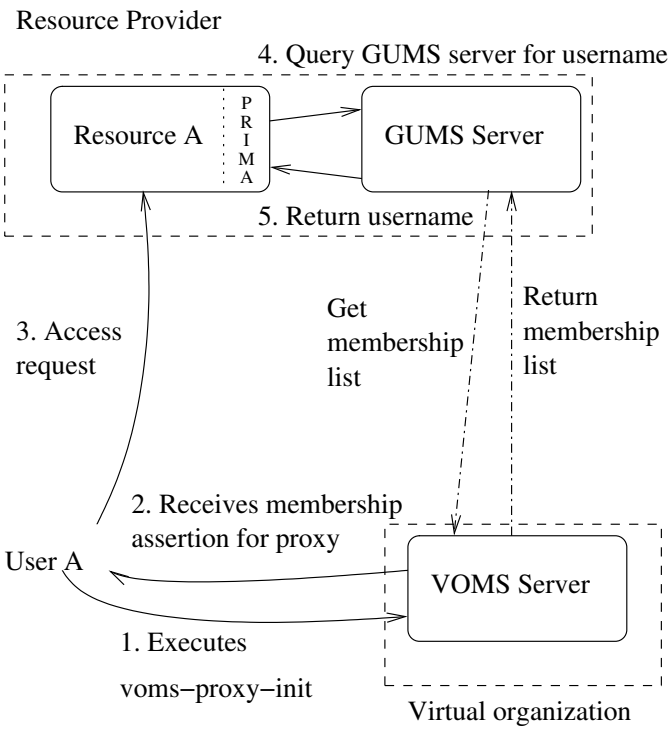

Figure 1: Setup for VOMS.

Access by a user may be revoked, and this revocation takes effect when the GUMS server's cache is updated. Users removed from the organization will thus no longer be able to launch jobs shortly thereafter. Where authentication is against a gridmap file generated by a cron job the user will be denied access following the next execution of the cron job. In the remaining cases when the VO membership cache at the GUMS server is updated, access will be revoked. The VOMS approach allows for hacked user accounts or users in violation of usage policy to be tracked down. VOMS retains log files which detail the mapping process and allow the perpetrator to be tracked down and/or the compromised account to be suspended.

3.1.3 High Availability

The Large Hadron Collider at CERN [25] is a large-scale Physics project which is scheduled to begin generating data at a rate of 15 petabytes per year once the facility goes into operation. This data will be accessed by a large community of scientists scattered around the world. Given the scale of the project, a high level of availability is required for the VOMS server used. They are using the Linux-HA technology [27] to provide redundancy [52]. Linux-HA allows a hot-spare machine on the same network to assume the failed node's IP address and take over its operations in case the original node fails. However, Linux-HA leaves the system vulnerable to network failure, as both the original system and the hot-spare reside on the same network. If there are problems routing packets to and from this network, even if the machine running VOMS is operational it will be unable to communicate with its clients.

3.2 GridShib Approach

Another VO-management framework that has been experimented with is GridShib [16], an extension to Shibboleth [41]. Shibboleth was developed to provide browser-based authentication to Web resources such as article databases, and it possesses a relatively large installation base.

When a user attempts to access a resource which is protected by Shibboleth they are redirected to a WAYF if they have not already authenticated. This WAYF, short for "where are you from", is a Web portal which allows the user to select the organization they wish to be associated with. Following selection, the user is redirected to the identity provider (IdP) which they selected at the WAYF. An IdP is a trusted service that maintains attributes associated with users and represents an institution or virtual organization. After being redirected by the WAYF to the IdP the user must authenticate by some means. Shibboleth allows any authentication method to be used to validate the user's identity.

After authentication at the IdP, the user is assigned an identifier that is generally intended to be anonymous. Shibboleth was developed to protect user privacy, and thus resource sites are provided with the minimal amount of information necessary. After the IdP has assigned the user an anonymous identifier, the user then communicates this identifier to the resource. The resource can then query an attribute authority at the IdP site with this identifier to retrieve those attributes to which it has been granted access permission. Based on the values of these attributes, the resource can then determine whether or not to grant access to the user.

GridShib [16] is a project which attempts to take advantage of the investment which many institutions have put into the Shibboleth infrastructure by extending its applicability to the grid world. It utilizes certificate registries at the identity providers in which users must register their certificates in order to be mapped appropriately when they provide the certificate to a resource to request access.

In the current implementation of GridShib resources must be configured with metadata and a particular identity provider to authenticate against. When a user attempts to authenticate, their certificate DN is provided to the IdP's certificate registry in an attempt to map it to a user account. Once this mapping has been completed, the resource can then query for user attributes and make an authorization decision based on the information retrieved.

This section describes how authorization using GridShib works in practice. It also addresses how this system meets the goals of virtual organization management, and the practical issue of availability that production systems must address. Finally, it concludes with a brief discussion of myVocs [30], a project which further augments the virtual organization management capabilities which GridShib provides.

3.2.1 Overview of an authorization

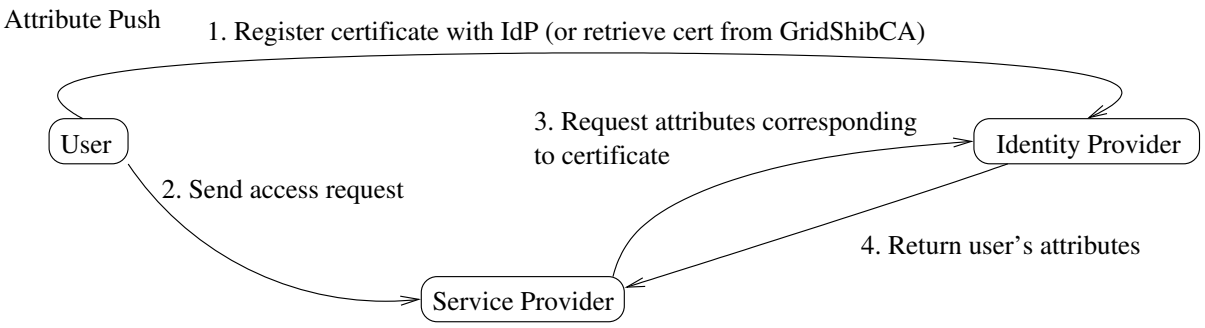

Figure 2: Setup for GridShib.

Figure 2 outlines the authorization process of GridShib. A user registers a certificate with a certificate registry. Then, when the user sends a access request to a service provider, each service the provider offers is configured with a particular identity provider. In regular Web based Shibboleth authentications a WAYF would allow the user to select a particular identity provider, but GridShib's work to embed assertions in proxies to accomplish this same goal is not finished. Thus a particular identity provider must be specified in the resource's security configuration. The service provider requests the attributes corresponding to the certificate from this identity provider and makes a decision whether or not to allow the user's access request based on the values of these attributes.

Unfortunately there are some limitations to user mappings in GridShib, preventing the use of such things as account pools at the present time. These limitations are something that future work promises to eliminate. The additional limitation of requiring a particular identity provider to be selected also promises to go away. There are plans [15] to

develop tools to embed an identity provider assertion in X.509 certificates, but as of yet the actual implementation of these tools remains to be completed. Some initial work on embedding assertions was completed, but that work was deprecated and awaits replacement [24].

3.2.2 Addressing goals

When a virtual organization is setup using a regular Shibboleth IdP, a new user is added to whatever user database the VO has chosen to employ. At present, a resource must be configured with the identity of a particular IdP to use. In the future once the tools to deal with SAML assertions attached to user certificates have been fully implemented, the user may then attach an assertion to their X.509 certificate. This assertion specifies the virtual organization's IdP as the location to authenticate against. myVocs, which will be outlined later, may allow users to be added more easily.

GridShib does not map users to a specific account in the same way that GUMS does and this makes selecting roles difficult. Rather a list of accounts to which the user has access privileges is calculated. The service provider will select one of these accounts, unless the user specifies otherwise. By adding a request for a specific username to a request, the user can be mapped to the specified account. This can be used in a similar way to VOMS, but suffers from the problem that the account names will frequently be system specific, and it may not be easy to determine which account should be utilized for which purpose.

Shibboleth and Gridshib deal also with the issue of revoking user access to resources. Log files which detail the mapping process are produced by Shibboleth. These logs allow the perpetrator of any inappropriate activity to be tracked down and/or the compromised account suspended. Users removed from the organization will no longer be able to launch jobs shortly thereafter. Where time-limited attributes are pushed to the resource site or remain cached from a previous authorization attempt, the user will be denied access when the pushed or cached attributes' validity expires. In the remaining cases the virtual organization will be contacted as part of the authorization attempt, and will deny the user whose account has been revoked access virtually instantaneously.

3.2.3 High Availability

Georgetown University has a project called HAShib [21] which attempts provide high-availability by sharing states between nodes running Shibboleth IdP software.

Another way to improve availability is to eliminate the need for the service provider to query an identity provider for user attributes. While GridShib currently pulls attributes from the Identity Provider, the GridShib team is also working towards a push model of attributes, where attributes (signed by the identity provider) could be attached to a proxy certificate [54].

3.2.4 myVocs

One thing to consider when dealing with virtual organizations is what entity controls a user's attributes. To simplify management of users, it is desirable that the organization's administrators exercise control over those user attributes related to their project. The organizers can then make changes as need requires without waiting for any of the system administrators to make changes. At the same time the users do not need to be registered at all sites individually. Each site maintains full control of all attributes which it defines, and the VO retains full control over its membership and their attributes.

There are two alternatives to make this a reality. The first option involves setting up the virtual organization similar to a regular organization. A user database is setup and all accounts and associated users' attributes must be created by the VO administrator. Just as in any other Identity Provider's certificate registry an X.509 certificate can be associated with the user. When attempting to use a resource the user specifies this virtual organization's Identity Provider as the authorization target. Authorization then proceeds as normal. An alternative to this way of doing things is offered by the University of Alabama's myVocs project [30]. myVocs functions conceptually like an Identity Provider proxy. It maintains a list of membership criteria, but may pull some attributes from downstream identity providers to which users authenticating against the myVocs Identity Provider will be redirected for authentication. The myVocs IdP is only able to get hold of those attributes which the attribute release policy of the home institution allows. Users must trust the myVocs IdP, and can verify its signature from the federation metadata.

Thus, the myVocs approach attempts to retain the advantage of full control by administrators over membership and attributes but at the same time lowering the administrative cost. myVocs seems promising, but only recently has a means of deploying the software become available [29], so testing of this remains to be done.

3.3 Other authorization frameworks

VOMS and GridShib are just two of the authorization frameworks that have been developed. In this section a few of the other players in the VO management space are briefly described.

3.3.1 PRIMA

Although PRIMA [28] produced an authorization module that sees use as part of the VOMS approach to virtual organization management, the PRIMA project consists of more than just this component. Just as when utilized by GUMS, PRIMA allows the services to run without requiring changes to their code to enforce VO restrictions. The way in which PRIMA alone differs from PRIMA combined with GUMS and VOMS is that it does away with the central VO server and instead uses attributes provided directly by attribute authorities. It is intended to deal with ad-hoc and short term group management, avoiding the extra infrastructure required by VOMS.

3.3.2 CAS

Similar to VOMS, CAS [34] is an approach to VO management centered around a server which delegates rights to users. It differs though in that it requires modification to application code to enforce security restrictions. This is accomplished using a CAS API to query attributes attached to user certificates. CAS is distributed as part of the Globus Toolkit [10].

3.3.3 Akenti

Akenti [47, 46], like CAS, requires modifications to application code in order to enforce restrictions upon those utilizing VO credentials. It allows more finely-grained authorization decisions through the use of an XML policy language, producing role-based access control.

3.3.4 PERMIS

PERMIS [2, 35] provides a form of role-based access control similar to Akenti. It uses a policy allocator to create attribute certificates which are then stored in an LDAP server to be allocated to users when requested.

3.3.5 Grid Grouper

The cancer Biomedical Informatics Grid (caBIG) developed Grid Grouper [14] as part of the GAARDS framework [8]. This tool extends an Internet2 project called Grouper [18] which attempts to create a common API for group management, allowing both composite and subgroups to be created.

3.3.6 UK National Grid Service projects

Multiple projects exist to investigate other ways to incorporate Shibboleth authentication with the UK National Grid Service [48]. SHEBANGS [39, 38] (Shibboleth Enabled Bridge to Access the National Grid Service) and ShibGrid [43] are two examples of such projects.

4 Experiences with VO Management Tools

VOMS and GridShib were the two frameworks which our testing focused upon, and they are described in that order. The testbeds established and the installation process for each VO framework are described. Overviews of the tests conducted are given, as well as an overall assessment of each framework.

4.1 VOMS

4.1.1 Testbed information

VOMS was the first VO management framework examined by us in any significant detail. It was tested using a setup consisting of three virtual machines running on an x86-based system. These virtual machines were setup using Xen [55], a system which allows multiple machine instances to execute concurrently on a single physical machine. Xen allows each virtual machine to run a different operating system, but in this case all machines were setup with Fedora Core 4 [6] and each was allocated 512 megabytes of RAM. One of these machines was primarily a VOMS server, and another served as a GUMS server.

4.1.2 Compilation / Installation

VOMS was initially created at INFN in Italy, although effort was required to find the current CVS repository as the information posted was outdated. The latest version of the source is now available through CVS at CERN [51].

Getting VOMS to work required that several other packages be installed, and there were some difficulties getting these working. GUMS, the component allowing centralized authorization at resource sites, has detailed documentation on its Web site [20]. However, the information as to where the source for this application could be downloaded was outdated. The source code is now available through Subversion [19] from an alternate maintainer, John Hover [23].

Attempts to build PRIMA, the plug-in authentication module for Globus, from source were ultimately a failure. Although there is some source code available through the PRIMA website [36], the build process for this is dependent upon having a particular computing environment, details of which were unavailable. Although building from source is desirable for applications placing an important role in security, it was suggested by the developers to utilize the build contained in the Virtual Data Toolkit (VDT) [50]. Further inquiry resulted in the discovery that the VDT project was not performing their own builds, but rather having the application built for them by the PRIMA group. Having only a single site at which builds are conducted in addition to a nontrivial build process is a concern.

The last of the components in the VOMS framework is PRIMA-WS. PRIMA handles authentication for the pre-Web services version of Globus, and PRIMA-WS extends this authentication to the web-services version of Globus. PRIMA-WS is available from John Hover at BNL [23].

As there were difficulties building from source, VOMS was tested using the binaries provided by the Virtual Data Toolkit (VDT). VDT, developed for the Open Science Grid [31], attempts to facilitate easy, binary installations of grid software packages utilizing Pacman [33].

Once a test machine had successfully been installed and configured with VDT, extra machines could be setup relatively easily. However, getting to the point where the initial installation was functional was not a straight-forward process. The software expected that not only certain packages be installed but that these be specific versions. This caused some problems initially, as error messages were lost in log files, and occasionally errors went undetected for some time. Thus, strange problems were sometimes tracked down to issues that were relatively simple to fix. For example, Tomcat was at first left unconfigured, and this was due to a seemingly unrelated missing package. Better visibility of error messages as well as more extensive error checking would be good to see in VDT.

Weak security is another concern about VDT. At the present time, securing an instance of some VDT-supplied applications is problematic and dependent upon denying non-administrative users any access to the host computer. Whereas the applications themselves may be locked down reasonably when installed on their own, in the current release VDT stores sensitive information in a MySQL database using only host-based security. While this makes script-writing easier, it definitely poses security concerns. Progress is being made to install services in a more secure fashion, but presently this concern remains.

Availability of VDT only in binary form is also somewhat problematic. For security-critical services, a category into which VO-management tools fit, having direct access to the source code to perform security audits or bug fixes is important. Additionally, as was discussed earlier in this section, there are some difficulties limiting the extent to which the component services of VDT can be built separately. VDT only supports a limited number of platforms. Even though VDT targets the scientific community, and Scientific Linux version 4 dates from April 2005 [37], Scientific Linux 4 was not supported even by the VDT 1.5.2 release in December 2006. Users are still restricted to Linux platforms [49].

VDT has made some significant improvements over the course of the past year, and for many projects may work well. The VDT support team has proved attentive to support requests. Some concerns remain, but VDT has provided a platform whereupon the VOMS framework can be utilized.

4.1.3 Tests conducted

Tests were conducted to confirm the applicability of the VOMS framework to each of the goals outlined in Section 2. As using GUMS-generated gridmap files is functionally no different than standard gridmap files, testing was primarily performed using the PRIMA module. Tests pertaining to the addition of users were conducted both by having administrators manually add users as well as by having users apply using the Web interface provided. This required that users configure their Web browser to work with their certificates. Both means of user addition proved effective. Additional testing was performed to address working with multiple roles. By default VOMS does not append anything beyond a default role to a proxy. If a particular role is desired, it must be requested when initializing the proxy. Basic testing was also done to confirm that malicious users could be tracked down, and account revocation tests were successful in denying the former user access once revocation had taken place.

Some additional testing was done to see how well VOMS and MyProxy interoperated, but this work did not meet with success. Instructions in an OSG Wiki [53] were tested, but segmentation faults were encountered as a result.

4.1.4 Assessment

In terms of availability, source code for the tools was difficult to find due to outdated Web sites. While installation of VOMS and GUMS from source was successful, difficulties were encountered in installing PRIMA from source due to various dependencies. Installation of the binaries for all tools was successful using VDT, although not straightforward. Also, VDT has limited platform support and only provides binaries which is a security concern.

On a positive note, the VOMS approach provides much functionality in terms of the goals outlined in Section 2. Mechanisms for easily adding/removing users and for mapping users to accounts in various ways exist. There is also good support for managing roles within a VO. Furthermore, there is support for both pre-Web service and Web service components of the Globus Toolkit.

VOMS is primarily being used by large Physics projects that mainly use x86-based Linux systems and as such limited platform support has not been an issue. With improved availability and platform support, VOMS and its supporting tools would be a viable approach for VO management in a more general context.

4.2 GridShib

4.2.1 Testbed information

Similar to the VOMS testbed, testing of GridShib and Shibboleth was conducted using virtual machines run on an x86-based system using Xen [55]. The testbed for GridShib consisted of a total of four virtual machines, each running Fedora Core 4. One of these machines was set aside with TestShib [45] on it to serve as a reference platform. Another of these machines hosted the WAYF for the federation and could also have been used to distribute cryptographically-signed federation metadata updates. The remaining two machines were setup as institutions offering a full set of services as well as acting as identity providers. These last two machines required the installation of the latest version of Globus. Some issues that cropped up when version 4.0.1 of the Globus Toolkit was installed seemed to disappear following an upgrade to Globus 4.0.3.

4.2.2 Compilation / Installation

The Shibboleth project is intended to serve as an extension of an existing site authorization system. Gaining access to the source for this project was easy compared to obtaining that for VOMS, GUMS, or PRIMA. This source can be found on the Shibboleth Web site [41].

With VOMS, given a certificate, that certificate just needed to be added to the VOMS database. Prior to being able to use Shibboleth to any great extent, a prototypical user database with user attributes needed to be created. Shibboleth Identity Providers are able to deal with user attributes stored in a variety of manners. Built in are means of handling

static attributes, as well as attributes stored in SQL databases or available through LDAP. The testbed that was setup using an OpenLDAP server as an attribute store. The majority of attributes created were taken from the eduPerson schema [3], a schema in common use amongst Shibboleth's predominantly academic installation base.

Getting Shibboleth up and running required that a metadata file be setup describing the entities in our testbed federation. The Shibboleth Wiki [42] has quite a bit of information available, but the documentation emphasized using a preexisting federation where possible rather than creating a new one. As such, additional documentation describing the metadata files which contain information about the Identity Providers and Service Providers is would be beneficial. Working files for a testbed federation were created, based largely upon examples contained in the Identity Provider installation package.

As Shibboleth originated as a tool for securing access to Web based databases using a browser, some experimentation took place in this realm prior to moving to testing the grid extensions. Some usage of the Web interface to Shibboleth may also be required when using the grid extensions, as will be described later. In the Web realm a component called the WAYF (where are you from) is regularly utilized to select an Identity Provider to authenticate against. This WAYF may allow the user to select any identity provider from any federation that the resource trusts. There is a WAYF that operates using standard Shibboleth metadata [40], but due to difficulties building this project a second WAYF developed by SWITCH, the Swiss academic computing organization, was utilized instead [44]. The developers of the Shibboleth WAYF appear to be making progress, so it is expected that in the future this service will be usable.

GridShib is presently under development at the University of Chicago, and has source code available through its Web site [16]. As of yet there are some features described that have not yet been implemented. Thus, current experimentation with GridShib does not take full advantage of all the functionality that it will eventually provide. As such this must be taken as only a preliminary evaluation. Note also that GridShib authorization also presently works only in the Web services realm. Pre-Web service Globus is not supported by GridShib.

The GridShib installation involves two components as well as some changes to the federation metadata. A component must be added to the Shibboleth Identity Provider to extend it with a certificate registry. A plug-in for the Globus Toolkit must also be installed. These components work together in order to create an environment in which one can authenticate against a Shibboleth-protected resource using an X.509 certificate. Changes to the federation metadata are required to declare the usage of X.509 certificates as a acceptable authentication mechanism for resource sites.

One reason for which the Web based interface to Shibboleth remains important is that users must first login to the Identity Provider in order to either register their existing certificate or to create a new short-term, anonymous certificate using a software package GridShibCA [17]. Thus far the former approach of registering certificates has proved successful, but problems which have yet to be resolved have been experienced when attempting to utilize GridShibCA. To register a certificate users must go to a Shibboleth-protected access point, e.g. https://myidp.com/shibboleth-idp/Certificate-Registry, and then paste in the contents of their certificate file (usually contained in ~/.globus/usercert.pem).

Some of the features that are yet to be implemented in the GridShib project involve tools to handle assertions appended to X.509 certificates. These tools [15] would allow a greater degree of automation during authentication. Presently, when testing with the Globus Toolkit, the security configuration files for the protected resource must specify a single Identity Provider to which all authentication requests will be directed. Rather than forcing users to have accounts at multiple identity providers in order to be able to access all resources to which they are entitled, it is desirable to allow the user to register at a single Identity Provider and then take advantage of the trust fabric of the grid federation.

Use of SAML assertions offers the potential for users to be authenticated against a particular Identity Provider that they specify. The resource could then check its federation metadata files to verify that a trust relationship exists with that Identity Provider and then authenticate against it. This will remain inoperable until the handling of SAML assertions is complete.

4.2.3 Tests conducted

As a set of relatively minor yet compounded problems delayed the establishment of a working GridShib platform, initial testing was conducted using Shibboleth the traditional way. Shibboleth was established to perform authentication for electronic databases through Web browsers, and this was the original testing arena for this platform. The setup of a WAYF allowed users to select what virtual organization they were a member of, but GridShib is somewhat lacking in

this area at the moment. GridShib is under active development, and when these tests were conducted the only way positive results were achieved was by specifying a particular identity provider in a resource's security configuration files. Thus the identity provider was fixed for a particular resource configuration instead of being variable and selectable by the user.

The goals previously described were considered in relation to how they worked in GridShib. Shibboleth and GridShib build on top of existing site user databases, so users could be added in the site's normal fashion. Selection of roles in GridShib is done by adding an attribute to an access request. There a user can request a username, but this requires advance knowledge of the user namespace of each site. Similar to VOMS, GridShib's log files appear extensive, seemingly providing sufficient detail to determine who a malicious user might be.

An attempt was also made to get GridShib and MyProxy functioning together. One way in which this can take place is through the use of GridShib CA [17] which can use a MyProxy-based certificate authority backend. Thus far success has eluded this endeavour, so some uncertainty remains as to how this software functions.

4.2.4 Assessment

Usage of Shibboleth is quite widespread in the academic realm for securing access to Web databases. However, GridShib is still under development, and so this must be taken as only a preliminary evaluation.

In comparison to the tools incorporated in the VOMS approach, those utilized in this approach were much easier to obtain. Documentation was also more thorough and few difficulties were encountered in installation.

Since pre-Web service tools are presently in widespread use, the fact that GridShib only supports Web services is a problem. However, if this problem was fixed, and the work to handle SAML assertions for proxy certificates was completed, GridShib would be a viable option. Support for the account mapping capabilities that GUMS provides as part of the VOMS approach is also needed.

5 Conclusion

High performance computing consortia in Canada are moving towards a more cooperative environment with resources being shared nationally. VO Management services have the potential to significantly aid in authorization and account management in such an environment. They allow users to become members of virtual organizations which would automatically grant them access to resources the virtual organizations are authorized to use, subject to them holding an appropriate role. This would greatly reduce the administrative overhead involved in acquiring accounts.

Two particular VO management approaches were examined in this paper. One approach involves the use of VOMS, GUMS and PRIMA. The other approach makes use of Shibboleth and a grid extension to it called GridShib. The approaches were assessed on their availability, ease of installation and functionality in terms of adding users, assigning roles and removing users.

The VOMS approach has greater functionality but has limited platform support. The GridShib approach is limited in functionality but is standards-based. Unlike for the VOMS approach, the source code is readily available and easy to compile. Improving platform support for VOMS would help make it a more viable approach in a general context. However, as the GridShib approach is standards-based, it appears to be the more promising solution once it is able to provide the same level of functionality as VOMS. Adoption of either set of tools currently requires significant effort and development in a multi-platform computing environment. The benefits of such effort could be substantial and would improve the ability to share resources.

References

[1] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Frohner, K. Lorentey, and F. Spataro. From gridmap-file to VOMS: Managing authorization in a grid environment. *Future Generation Computer Systems*, 21(4):549–558, 2005.

[2] David W. Chadwick and Alexander Otenko. The PERMIS X.509 role based privilege management infrastructure. *Future Generation Computer Systems*, 19(23):277–289, 2003.

[3] eduPerson schema. http://www.nmi-edit.org/eduPerson/internet2-mace-dir-eduperson-200604.html. Accessed December 12, 2006.

[4] Enabling Grids for E-Science (EGEE). http://www.eu-egee.org/. Accessed January 9, 2007.

[5] EGEE VOs. http://cic.gridops.org/index.php?section=home&page=volist. Accessed February 9, 2007.

[6] Fedora project, sponsored by Red Hat.
http://fedora.redhat.com/. Accessed January 2, 2007.

[7] Ian Foster. Globus Toolkit version 4: Software for service oriented systems. In *Proceedings of the IFIP International Conference on Network and Parallel Computing*, pages 2–13, 2005.

[8] GAARDS. http://www.cagrid.org/mwiki/index.php?title=GAARDS:Main. Accessed February 7, 2007.

[9] The official gaussian 03 web site. http://www.gaussian.com/. Accessed January 9, 2007.

[10] GT 4.0: Security: Community authorization service. http://www.globus.org/toolkit/docs/4.0/security/cas/. Accessed January 8, 2007.

[11] Globus authorization callouts (pre-web services). http://www-unix.globus.org/security/callouts/. Accessed December 12, 2006.

[12] GT4 WS AA admin guide. http://www.globus.org/toolkit/docs/4.0/security/authzframe/admin-index.html. Accessed December 12, 2006.

[13] Grid Canada. http://www.gridcanada.ca/. Accessed February 2, 2007.

[14] Grid grouper. http://www.cagrid.org/mwiki/index.php?title=GridGrouper:Main. Accessed February 7, 2007.

[15] GridShib: SAML assertion tools. https://spaces.internet2.edu/display/GS/ SAMLAssertionTools. Accessed January 12, 2006.

[16] GridShib: A policy controlled attribute framework. http://gridshib.globus.org/. Accessed December 12, 2006.

[17] GridShib certificate authority. https://spaces.internet2.edu/display/GS/ GridShibCertificateAuthority. Accessed January 12, 2007.

[18] Grouper. http://middleware.internet2.edu/dir/groups/grouper/. Accessed February 7, 2007.

[19] GUMS source code using subversion. https://svn.usatlas.bnl.gov/svn/griddev/gums-multiproject/. Accessed December 12, 2006.

[20] Grid user management system. http://grid.racf.bnl.gov/GUMS/. Accessed December 12, 2006.

[21] HA Shib. https://www.middleware.georgetown.edu/confluence/display/MW/hashib. Accessed January 2, 2007.

[22] R. Housley, W. Ford, W. Polk, and D. Solo. Internet X.509 public key infrastructure certificate and CRL profile. RFC 2459, The Internet Society, 1999.

[23] John Hover. http://www.usatlas.bnl.gov/twiki/bin/view/Main/JohnHover. Accessed December 12, 2006.

[24] IdP discovery based on embedded SAML authentication assertion. http://dev.globus.org/wiki/IdP_discovery_based_on_embedded_SAML _authentication_assertion. Accessed December 19, 2006.

[25] LHC computing grid.
http://lcg.web.cern.ch/LCG/. Accessed January 3, 2007.

[26] Erik Lindahl, Berk Hess, and David van der Spoel. GROMACS 3.0: a package for molecular simulation and trajectory analysis. *Journal of Molecular Modeling*, 7(8):306–317, 2001.

[27] High Availability Linux. http://www.linux-ha.org/. Accessed December 19, 2006.

[28] M. Lorch, D.B. Adams, D. Kafura, M.S.R. Koneni, A. Rathi, and S. Shaw. The PRIMA system for privilege management, authorization and enforcement in grid environments. In *Proceedings of the Fourth International Workshop on Grid Computing (GRID'03)*, pages 109–116. IEEE, 2003.

[29] myVocs box. http://myvocs-box.myvocs.org/. Accessed December 19, 2006.

[30] myVocs. http://www.myvocs.org/. Accessed December 12, 2006.

[31] Open Science Grid. http://www.opensciencegrid.org/. Accessed December 12, 2006.

[32] OSG VOs. http://www.opensciencegrid.org/?pid=1000137. Accessed February 9, 2007.

[33] Pacman. http://physics.bu.edu/pacman/. Accessed November 15, 2006.

[34] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. A community authorization service for group collaboration. In *Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks*, pages 50–59, 2002.

[35] What is PERMIS? http://sec.cs.kent.ac.uk/permis/documents/concept.shtml. Accessed December 12, 2006.

[36] PRIMA "PRIvilege Management and Authorization". http://computing.fnal.gov/docs/products/voprivilege/prima/prima.html. Accessed December 12, 2006.

[37] Scientific Linux distribution list. https://www.scientificlinux.org/distributions/. Accessed December 12, 2006.

[38] Shebangs presentation at OGF19. http://www.ogf.org/OGF19/materials/575/federatedIdentity.ppt. Accessed February 7, 2007.

[39] Shebangs. http://www.mc.manchester.ac.uk/research/shebangs. Accessed February 7, 2007.

[40] Shibboleth WAYF. https://spaces.internet2.edu/display/SHIB/DiscoveryService. Accessed January 12, 2007.

[41] Shibboleth. http://shibboleth.internet2.edu/. Accessed December 12, 2006.

[42] Shibboleth wiki. https://spaces.internet2.edu/display/ SHIB/WebHome. Accessed January 12, 2007.

[43] Shibgrid. http://www.oerc.ox.ac.uk/activities/projects/index.xml?ID=ShibGrid. Accessed February 7, 2007.

[44] SWITCH WAYF. http://www.switch.ch/aai/wayf/. Accessed December 12, 2006.

[45] Testshib.org. http://www.testshib.org. Accessed December 19, 2006.

[46] M. Thompson, A. Essiari, K. Keahey, V. Welch, S. Lang, and B. Liu. Fine-Grained Authorization for Job and Resource Management Using Akenti and the Globus Toolkit. *ArXiv Computer Science e-prints*, June 2003.

[47] Mary Thompson, William Johnston, Srilekha Mudumbai, Gary Hoo, Keith Jackson, and Abdelilah Essiari. Certificate based access control for widely distributed resources. In *Proceedings of the 8th USENIX Security Symposium*, pages 215–228, 1999.

[48] Uk national grid service. http://www.grid-support.ac.uk/. Accessed February 7, 2007.

[49] VDT 1.5.2 supported platforms list. http://vdt.cs.wisc.edu/releases/1.5.2/contents.html. Accessed December 12, 2006.

[50] Virtual Data Toolkit. http://vdt.cs.wisc.edu/. Accessed January 11, 2007.

[51] EGEE JRA1 middleware CVS repository. http://jra1mw.cvs.cern.ch:8180/cgi-bin/jra1mw.cgi/. Accessed December 12, 2006.

[52] High availability implementation for VOMS. https://twiki.cern.ch/twiki/bin/view/LCG/VomsWlcgHa. Accessed December 19, 2006.

[53] VOMS and MyProxy integration testing. http://osg.ivdgl.org/twiki/bin/view/Integration/VOMSandMYPROXY. Accessed December 19, 2006.

[54] Von Welch. GridShib project update. http://events.internet2.edu/2006/fall-mm/sessionDetails.cfm?session=2979&event=258. Accessed December 19, 2006.

[55] Xen. http://www.xensource.com/xen/. Accessed January 2, 2007.