

**Cross-Domain Authorization for Federated Virtual Organizations Using  
the myVocs Collaboration Environment**

Journal:	<i>Concurrency and Computation: Practice and Experience</i>
Manuscript ID:	CPE-07-0128.R1
Editor Selection:	Prof. Geoffrey C. Fox
Wiley - Manuscript type:	Research Article
Date Submitted by the Author:	13-Mar-2008
Complete List of Authors:	Gemmill, Jill; Clemson University, Cyberinfrastructure Technology Integration; Clemson University, School of Computing Robinson, John-Paul; University of Alabama at Birmingham, Information Technology Scavo, Tom; National Center for Supercomputing Applications Bangalore, Purushotham; Univerity of Alabama at Birmingham, Computer and Information Sciences
Keywords:	Shibboleth, Federated Identity, SAML



## Cross-Domain Authorization for Federated Virtual Organizations Using the myVocs Collaboration Environment

Jill Gemmill<sup>1,2</sup>, John-Paul Robinson<sup>3</sup>, Tom Scavo<sup>5</sup>, Purushotham Bangalore<sup>4</sup>

<sup>1</sup>Clemson University Cyberinfrastructure Technology Integration; <sup>2</sup>Clemson University School of Computing; <sup>3</sup>University of Alabama at Birmingham, Information Technology; <sup>4</sup>University of Alabama at Birmingham, Department of Computer and Information Sciences; <sup>5</sup>National Center for Supercomputing Applications

Correspondence to: Jill Gemmill, Clemson University  
340 Computer Court  
Anderson, South Carolina  
864-656-3343

Email: [gemmill@clemson.edu](mailto:gemmill@clemson.edu)

## ABSTRACT

This paper describes our experiences building and working with the reference implementation of myVocs (my Virtual Organization Collaboration System). myVocs provides a flexible environment for exploring new approaches to security, application development, and access control built from Internet services without a central identity repository. The myVocs framework enables VO self-management across unrelated security domains for multiple, unrelated VOs by leveraging the emerging distributed identity management infrastructure. myVocs provides an accessible, secure collaborative environment using standards for federated identity management and open source software developed through the National Science Foundation Middleware Initiative. The Shibboleth software, an early implementation of the OASIS SAML standard for browser single sign-on, provides the middleware needed to assert identity and attributes across domains so that access control decisions can be determined at each resource based on local policy. The eduPerson object class for LDAP provides standardized naming, format and semantics for a global identifier. We have found that a Shibboleth deployment supporting virtual organizations requires the addition of a new VO service component allowing VOs to manage their own membership and control access to their distributed resources. The myVocs system can be integrated with Grid authentication and authorization using GridShib.

**Keywords:** SAML; Shibboleth; authentication; authorization; single sign-on; federated identity

## INTRODUCTION

Collaborations today are formed dynamically, crossing institutional and administrative boundaries. These collaborations are called virtual organizations (VOs) [1] where the ability of VO members to work collectively requires sharing data and resources that are distributed across administratively independent domains. There are many obstacles preventing transparent, shared access to the VO's collective resources, especially when sensitive information is involved. Collaborations facing this challenge today include: multi-institutional research programs; scientific communities such as those represented by IEEE or Internet2 working groups and the TeraGrid [2]; and consortia such as regional healthcare networks. A participant in any of these collaborations should be able to securely access data and resources anywhere, in any order, without constantly authenticating using different credentials at each location. Providing self-management so that VOs can easily create and manage their own memberships and roles and also manage access control for their own resources is an additional challenge, especially when those shared resources are hosted at multiple enterprises.

By way of an example scenario, scientists from three medical research institutions are aggregating their clinical research data on heart disease so that a fourth collaborator can mine the entire data collection. Each clinical site is managed by an IT department that is responsible for securing this data and logging all access to it. The four scientists, their lab staff and graduate students need to collaboratively analyze the data, review findings from the data mining, collectively document steps taken and decisions made, and jointly write papers describing their methods and findings. These functional requirements dictate that all data sets and documents, although stored in four locations, must be readable by every collaborator and writeable by the data owners only. In addition, some private but shared file store for common documents that can be edited by all VO members must be available.

A typical solution used today to provide a set of applications across a common security context involves some type of single domain identity repository and portal where users are required to register an identity. Familiar examples include collaboration platforms such as Yahoo or Google Groups [3,4], JSR-168 [5] portals such as Pluto [6] or Jetspeed [7], and Microsoft Active Directory [8,9]. All three examples provide ease of access to shared, secured resources, but with major limitations. The Yahoo/Google Groups solution requires that all user identities, group memberships, services and data reside under a single management domain (Yahoo or Google). The benefits achieved include self-management and autonomy; the disadvantages are that the domain dictates which services are available, each participant must have an identity (login/password) in that domain, and all shared data must reside at that domain. Available portal solutions are similarly limited by their requirement for a single identity repository and the complexity of applications integration. Adding applications or other resources to the portal requires permission from the portal administrator along with some non-trivial technical work. Active Directory (AD) provides easy group creation and management. Although it is possible to configure some types of trust relationship between AD domains it is difficult to limit the scope of that trust to a single resource and certain services will not work if the foreign directory schema does not contain the necessary object classes. Each of the solutions mentioned provides a centralized approach to managing distributed users and resources, but in many situations what is needed is a decentralized approach. The limitations of centralized solutions led us to examine emerging middleware solutions for federated identity.

This paper is organized as follows: the design goals for myVocs are described and our evaluation of candidate middleware solutions is provided. Identity management, authentication

and authorization in the Grid and federated approaches are compared as background justifying the rationale for components selected to construct myVocs. Integration of middleware components in myVocs so as to provide a system environment supporting distributed collaboration is described. Use of myVocs by a hypothetical collaboration is considered and real-world use of myVocs to date is summarized. The conclusion identifies limitations, remaining challenges and our planned next steps.

### **myVocs DESIGN GOALS**

By selecting some of the software components developed through the National Science Foundation Middleware Initiative (NMI) [10], we leverage distributed identity management services to provide a consistent security context for VOs across systems and applications. With the addition of a new service that associates enterprise asserted identity with VO-managed attributes, it is possible to build a distributed collaboration environment. Our solution combines identity federation with self-managed VO attributes and is named *myVocs* for “my Virtual Organization Collaboration System.” Our work is the first known attempt to address VO membership functionality in a federated environment without mandating any particular authentication mechanism. While the Virtual Organization Membership Service (VOMS) [11,12] also allows VOs to manage their own memberships, it is tied very strongly to X.509 public key authentication and attribute certificates.

The design of myVocs stemmed from three fundamental design goals that we believe are necessary for accessible, secure collaborative environments in higher education. These goals are: (a) to leverage the identity management systems provided by increasing numbers of universities; (b) to consider the VO to be authoritative for its own creation and membership assignments, thus maximizing VO autonomy, and (c) to make it possible for a service provider in any domain to easily authorize access based on VO membership. Due to middleware activities at Internet2 and elsewhere, many universities in the United States, UK, Australia and Europe are building identity management infrastructure using an Internet2-developed LDAP object class for US higher education called eduPerson [13] and are also deploying cross-domain single sign-on (SSO) services using Shibboleth [14,15]. The eduPerson standardized attribute vocabulary and Shibboleth-based SSO infrastructure provide a reliable and growing foundation for the first goal. Leveraging this infrastructure leaves the burdens of identity verification, identifier assignment, secure authentication and password management to individual organizations while allowing VO members to use their familiar campus authentication service to identify themselves to resources located anywhere on the Internet. Compared to email-based user account creation, as provided by Yahoo/Google and others, enterprise managed identity typically involves processes for stronger identity verification, such as examination of government issued documents.

The second goal of VO self-management is crucial for autonomy and scalability. Our model is capable of simultaneously supporting thousands of small collaborative activities as well as TeraGrid-scale operations. The current trend in science is towards multidisciplinary teams while the Internet allows team members to live almost anywhere in the world. These collaborations and their organizations are self-organizing and dynamic, and myVocs facilitates autonomous relationships by making it easy to provision a VO-centric security context.

The third goal addresses the common problem of how to securely share data, documents, or computational resources across domains and between applications. We wanted each VO to be able to easily assemble a customized, web-based collaboration environment that meets their unique requirements. We believed that the approach envisioned would become reality if

collaborative applications would readily interface with the emerging middleware infrastructure [16,17].

## EVALUATING CANDIDATE MIDDLEWARE

Available middleware components were evaluated with these goals in mind. We are aware of three approaches to distributed access control with working open source software implementations: Globus grid software [18,19], Privilege and Role Management Infrastructure Standards validation (PERMIS) [20,21], and Shibboleth.

### Grid Identity, Authentication and Authorization

The Grid Security Infrastructure (GSI) [22] is built on Public Key Infrastructure (PKI), an architecture based on X.509 public key certificates that is an International Telecommunications Union (ITU) [23] and Internet Engineering Task Force (IETF) [24] standard. PKI binds an identity to one or more public/private key pairs that are signed by a trusted third party called a Certificate Authority (CA). These keys are used to establish trust and to secure communications between PKI participants. The X.509 public keys are presented to clients when identifying a resource, and to resources when identifying the requester. Thus public keys and their exchange might be considered as a “language” used by grids to establish identity.

PKI scalability was intended to be achieved via a hierarchical arrangement of CAs. Each CA in the hierarchy is signed with the private key of the authority one level above, thus establishing a chain of trust that can always be traced back to a root CA. PKI was originally designed to provide identity only; the X.509 standard does not specify standard certificate contents, other than requiring that the certificate’s Distinguished Name (DN) be globally unique. (Standardized industry practice, as exemplified by the IETF PKIX Working Group, has relaxed this requirement to a Distinguished Name (DN) that is unique within the domain of the issuing CA). It was hoped that by adhering to this simple concept that a global PKI could be achieved.

GSI is built on X.509 identity certificates with identity management left as an exercise for the deployer. In practice, certificates used in grids today have been issued using inconsistent and even unknown identity management practices, requiring case-by-case trust decisions. Some grids operate as ad-hoc identity federations by installing a common set of relevant CA root bundles at each resource, providing cross-domain access but with limited scalability. Bridged Certificate Authorities [25,26] improves scalability by construction of certification paths between CAs (through the bridge)..

Practical management difficulties with PKI include providing reliable private key storage for shared systems, mobile systems, and single user/multiple device scenarios. An important limitation of PKI is that only identity is provided, while access control decisions often require additional information about that identity such as group memberships or roles. A final consideration is that the grid software’s client-server architecture requires a higher than average level of end-user computing expertise. A friendlier user interface and more transparent key management are addressed by web-enabled grid portal middleware such as OGCE [27] and GridSphere [28]. These solutions are more user-friendly and can also be designed to provide group and/or role information about registered users. However, portals typically require registration of user identities in a single domain. A considerable security drawback is that a user must authenticate at the portal in order to obtain their grid credential, and the portal must have access to the user’s password in order to provide this service.

GSI authorizes access to grid resources based on a mapping of the digital certificate's DN to some local system account name. Augmenting this identity-based approach to authorization, VOMS and PERMIS rely on X.509 attribute certificates [29] for authorization. PERMIS introduced the addition of a Privilege Management Infrastructure (PMI) to a PKI, using the X.509 data structure to store attributes and roles. PERMIS requires that users have an authenticated LDAP Distinguished Name (DN) but does not address authentication. X.509 Attribute Certificates have not been adopted beyond the PERMIS infrastructure with the exception of the Virtual Organization Membership Service (VOMS). VOMS is the most successful attribute-based grid authorization model in use today, used in Europe (EGEE), the U.S. (OSG) and elsewhere.

A VOMS server issues a signed X.509 attribute certificate that a client typically binds to an X.509 proxy certificate. Attribute certificates are X.509 certificates with no public key but with standard attribute fields that declare group memberships and/or roles associated with the certificate's DN. Although X.509 attribute certificates are standardized, this approach to authorization has gained little traction outside of the Open Grid Forum [30]. Since VOMS attributes are encoded in standard X.509 attribute certificates, VOMS has found wide applicability in X.509-based grid environments. Moreover, the VOMS attribute profile is very well suited for expressing groups and roles. However, a major limitation of VOMS is that while a user may request his or her own attributes, it is not possible for another entity to obtain attributes on the user's behalf (unless of course that entity impersonates the user in some way). Thus VOMS is not suitable in portal environments, for instance, where the portal requests and aggregates attributes on behalf of the user. Moreover, the federating technology with the greatest presence on today's campuses is Shibboleth, an open source implementation of the SAML Browser Profiles [31,32]. Since Shibboleth is based on SAML, we conclude that VO middleware based on SAML (myVocs) is more desirable than middleware based on X.509 ACs (VOMS).

VOMS and PERMIS have each recently added a SAML interface. These features were not available while the work described in this paper was being done, but adoption of a SAML interface indicates a growing consensus among the community of distributed system developers of the utility of Shibboleth for grids, an approach pioneered by myVocs.

### **Shibboleth Federated Identity and Authorization**

Federated identity is predicated on a trust relationship between independent security domains. The Shibboleth Project is an Internet2 initiative to develop an open, standards-based solution for the exchange of information among institutions participating in a Federation. Shibboleth security is built on the Security Assertion Markup Language (SAML) specification, a standard developed by the Organization for the Advancement of Structured Information Standards (OASIS) [33]. SAML is an XML [34] standard for exchange of identity and attributes between identity providers and service providers. A SAML assertion contains a set of statements that can be securely exchanged between federated domains. Thus SAML assertions and their exchange might be considered as a "language" used by Service Providers to establish identity and other user attributes from a set of distributed Identity Providers. Like GSI, Shibboleth was designed to solve the problem of authorizing entities identified in one domain to use resources owned and managed by an unrelated domain. Unlike the typical grid use case, Shibboleth and SAML were designed for the case where the client is a standard web browser.

The OASIS Security Services Technical Committee (SSTC) was chartered in 2001 to "define an XML framework for exchanging authentication and authorization information" [35]

and produced the SAML V1.0 specification as an OASIS Standard in November 2002. Meanwhile, the Liberty Alliance [36] proposed an extension to the SAML standard. Like Shibboleth, the Liberty Identity Federation Framework (ID-FF) [37] describes a standardized, cross-domain, web-based single sign-on infrastructure where each participating domain is trusted to accurately document the processes used to identify a user, the type of authentication system used, and any policies associated with the authentication credentials. Other members of the federation may examine these policies to determine whether to trust these credentials.

While Liberty was developing ID-FF, the OASIS SSTC began work on SAML V1.1, a minor upgrade to the SAML Standard that was ratified in November of 2003 and is widely implemented and deployed today. During that same month, Liberty contributed ID-FF to OASIS, thereby sowing the seeds for SAML V2.0, a major revision of the SAML Standard that was announced by the SSTC in March 2005 [38]. Implementations and deployments of SAML V2.0 are increasingly becoming available.

### **Advantages of a Federated Approach**

SAML provides significant advantages over the PKI approach to distributed identity. First, federation allows each participating domain to deploy its authentication technology of choice. For example, one domain can use Kerberos while another uses simple username/password. In each case the Shibboleth software generates a standardized SAML authentication assertion to be transmitted to a trusted partner. A second advantage is that private key management is greatly simplified since certificates need to be issued only to servers and not to individuals, reducing the use of digital certificates by at least an order of magnitude as compared to traditional PKI. Finally, federation eliminates the need for a single root authority trusted by all. The Federation makes all participants' credentials easily available but does not vouch for any of these credentials; each issuing party vouches for their own credentials and each relying party determines its own roots of trust. Both Identity Providers (IdPs) and Service Providers (SPs) publish information about themselves in special XML files called metadata files. The metadata are aggregated and digitally signed by a trusted third party (such as the InCommon Federation [39]) that acts as a federation provider. The metadata includes the entity's public key, and therefore serves as a key distribution service. Shibboleth deployments are configured to periodically fetch fresh metadata from the federation provider. The provider's digital signature is verified with each refresh.

Shibboleth is a secure attribute transporting mechanism that is triggered by a user's attempt to use a web browser to access some protected resource. The security domain that creates and manages the user's attributes can be independent of the security domain in which the resource resides. User attributes including identity, group memberships, and roles supplied by the identity provider are transported by Shibboleth in a "just in time" manner to remote service providers for use in site-managed access control. Agreements regarding which attributes will be released and under what circumstances are negotiated in advance as part of the federation process. Shibboleth securely transmits those attributes across security domains. A Cross-domain transport mechanism with standardized attribute profiles is a key advantage of the Shibboleth model that is missing from most other solutions. Given Shibboleth's agnostic view of authentication methods, its roots in the broader OASIS standardization efforts and its focus on secure attribute transport, we chose Shibboleth as the vehicle for our VO self-management service. The InQueue Federation [40], a Shibboleth deployer's sandbox environment, served as our federated trust fabric. At present, InQueue has been replaced by the InCommon Federation production service and the TestShib [41] sandbox.

## SHIBBOLETH ARCHITECTURE AND MESSAGE FLOW

The Shibboleth architecture consists of three components: an Identity Provider (IdP), a Service Provider (SP), and an optional “Where Are You From?” (WAYF) service. During the development of myVocs we worked with Shibboleth versions 1.2 and 1.3 that implement the SAML V1.1 browser profiles. Shibboleth adds two distinguishing features to the SAML profiles: (a) protection of privacy by the use of an anonymous “handle” in place of a name-revealing identifier, and (b) an SP-initiated message flow. The reason for the emphasis on privacy is that higher education is required by federal law to preserve student privacy [42]. Access to many resources, such as licensed on-line journal subscriptions, is available to any member of the licensed community without requiring more specific identification, and so Shibboleth was designed to provide anonymity when accessing a resource.

A Shibboleth Identity Provider exposes an enterprise identity management system by leveraging a local authentication service. The identity data store can be an LDAP directory server or some other data store, such as a database. The authentication system can be any system chosen by the enterprise. Shibboleth components at the Identity Provider include a Shibboleth SSO Service and a Shibboleth Attribute Authority (AA). The SSO Service is configured to seamlessly interact with the local authentication service and to produce authentication assertions bound to HTTP [43] responses according to the SAML V1.1 Browser/POST or Browser/Artifact profiles. The Attribute Authority processes incoming SOAP [44] requests for attributes and issues attribute assertions according to the enterprise Attribute Release Policy (ARP).

A Shibboleth Service Provider (SP) protects web-based content and resources. Access control in Shibboleth is currently implemented using the `mod_shib` web server module, available for both Apache and IIS. Shibboleth components at the Service Provider include an Assertion Consumer Service and an Attribute Requester, components that work together to protect web content according to the access control rules expressed in `.htaccess` and/or web server configuration files. Both Identity Providers and Service Providers publish information about themselves in special XML files called metadata files that are digitally signed to precisely identify the provider and web-based services offered by that provider. Public keys are distributed in metadata files which simplifies the negotiation of trust. Because the signed metadata files may be self assertions, trust in them is achieved by out of band means.

The Identity Provider selection mechanism is implemented by Internet2 as a Federation “Where Are You From?” (WAYF) service and results from Shibboleth’s unique SP-initiated profile. The WAYF user interface is typically deployed as a pull-down selector listing all identity providers in the federation, which allows the user to select their preferred Identity Provider. To illustrate, an anonymous user (or principal) arrives via a browser at a Service Provider and attempts to access a protected resource. Since the principal is anonymous the Service Provider needs to send the user back to her Identity Provider to fetch the required SAML assertion. If the federation has multiple participating IdPs, as is typical, it is not possible to know in advance which Identity Provider is associated with this anonymous user. Therefore, some type of IdP selection mechanism is required.

The message flow among these Shibboleth components for the browser/POST profile is detailed in Figure 1; this Shibboleth message flow is leveraged by myVocs. Figure 1 illustrates how an anonymous user’s attempt to access a Shibboleth protected web resource with a browser at a Service Provider (SP) initiates a sequence of messages that eventually provide that SP with an assertion that the user has authenticated at the Identity Provider. Additional assertions

regarding the user's attributes may also be received. These assertions can be trusted due to out-of-band agreements made among federating parties and are used by the Service Provider in managing access to its own resources using a local access control policy. Solid arrows in Figure 1 indicate front-channel communications with the user agent (browser); dotted arrows indicate optional back-channel communication between Shibboleth components. Note that the Identity Provider and Service Provider can reside in unrelated security domains, and that Shibboleth does not specify any particular authentication method; the authentication step is external to Shibboleth. Finally, because resources are housed on web servers and web browsers are used to access these resources, Shibboleth leverages HTTP to direct the browser from one Shibboleth component to another. See the Shibboleth Technical Overview [45] for further detail regarding this and other message flows.

### **myVocs AGGREGATES ATTRIBUTES**

The architecture for myVocs developed from requirements to leverage enterprise identity, provide VO self-management, and provide a framework supporting single sign-on for numerous applications. How to integrate enterprise attributes with VO attributes became a central design issue. An obvious solution considered was to add attributes to the data store already being managed by each participant's Identity Provider. Aggregating all attributes associated with a particular identity at the IdP might appear to be a simple solution but in practice would involve obtaining permission to insert VO membership information into the attribute repositories of numerous enterprises. We believed that VOs would quickly find themselves mired in the policies and priorities of hundreds of unrelated domains, which is clearly not a path to autonomy. Instead, we built myVocs as a VO membership management service and associated VO attributes created by this service with enterprise asserted identity.

Our solution provides a consistent and trustworthy security context across distributed, independently administered domains allowing a user to directly and transparently access a protected resource. Shibboleth provided the necessary transport mechanism but with the unreasonable expectation that each participating enterprise directory would store all attributes associated with each identity managed at that enterprise. The problem in that assumption is that in collaborations that are distributed across enterprise boundaries, the authoritative source for VO membership may be from an unrelated security domain. As a result, the entity that owns and can assign the attribute is unlikely to have write permission into some other enterprise's attribute store.. As a VO attribute management tool, myVocs addresses this issue by facilitating the aggregation of attributes.

### **myVocs COMPONENTS**

The myVocs functional components, illustrated in the central portion of Figure 2 inside the square surrounded by double lines, include: (1) a Shibboleth Service Provider labeled "myVocs Service Provider"; (2) a Shibboleth Identity Provider labeled "VO Identity Provider"; (3) a VO membership management system labeled "Membership Manager (Sympa)"; and (4) a federated identifier and VO attribute data store labeled "Federated Identifier and Attribute Database". Note that the myVocs security domain and its access control policies are unrelated to the security domain of the Identity Provider and also unrelated to the domain of the Service Provider. The Shibboleth Identity Provider labeled "Enterprise X Identity Provider" represents just one of the many federated IdPs. Likewise, a Shibboleth Service Provider, labeled "VO Service Provider" protects some resource on one of many distributed web servers. Comparing

Figure 2 with Figure 1, observe that from the perspective of a federation Identity Provider, myVocs appears to function as a Shibboleth Service Provider and from the perspective of a VO's Service Provider myVocs appears to function as an Identity Provider. The message flows in Figure 2 will be described in a later section.

### **The Mailing List Manager as a VO Self-management Tool**

We consider self-managed VO membership preferable to any process requiring manual intervention by an external administrator. The mailing list management (MLM) application is a familiar, frequently used collaborative tool and in practice often serves as the authoritative source for VO name and membership information. Creating a new mailing list is equivalent to creating a new VO while adding and removing mailing list subscribers is equivalent to adding and removing VO members. MLM's already provide a variety of well-known methods for adding or approving list managers and subscribers. In addition, MLM's also provide per-list role information such as owner, moderator, subscriber, and editor; these designations can be considered rudimentary VO roles as well. Therefore, using an MLM application as the myVocs membership management component was a natural design choice. Of the open source mailing list packages available, the Sympa [46] MLM was selected as the best choice. Sympa provided a well-developed web interface that could be leveraged for use with Shibboleth. Sympa was also the most advanced with respect to middleware integration, supporting SQL and LDAP data stores and also authentication external to the MLM application. Finally, the Sympa developers were extremely interested in extending their Shibboleth integration and in collaborating with our project.

To understand the use of the MLM as a membership management tool for VOs, consider myVocs as a service provided to all identities managed by all enterprises participating in a federation. In order to benefit from the myVocs service each Identity Provider in the federation has agreed to release a persistent identifier, usually of the form user@domain, to myVocs. Our prototype leverages the eduPersonPrincipalName (ePPN) attribute for this purpose; however, any persistent global identifier could be used. How would participants in the usage scenario mentioned in the introduction begin using myVocs? The current version of myVocs requires a one-time registration. Let's assume Dr. Coeur has already registered at myVocs. Dr. Coeur visits myVocs with her browser and creates a new VO (mailing list) named 'HeartMine'. As owner of the HeartMine VO, Dr. Coeur can subscribe herself to the HeartMine list ("join the VO") and can use any of the well-known MLM subscription methods to decide whether she wants to invite additional members to join by email invitation, by open subscription with her approval, or by pre-population. These are standard MLM functions provided by Sympa.

Each VO member is notified of their HeartMine membership by the method Dr. Coeur has selected for notification, and they are asked to visit the myVocs URL to complete the one-time registration or verify an existing registration. Dr. Valentine has never used myVocs before, so when he clicks on the URL provided by Dr. Coeur he sees and clicks on a "Federated Login" button which takes him to the federation WAYF and then on to the Shibboleth single-sign-on service at his home institution. As Dr. Valentine's browser is redirected back to myVocs, the released ePPN is examined but myVocs does not recognize his identifier and therefore, Dr. Valentine is guided through the myVocs registration process. During registration Dr. Valentine is asked to specify a preferred email address. After the email address provided by Dr. Valentine is validated by the usual message-response mechanism, the ePPN and preferred email address are stored in the myVocs Federated Identifier and Attribute Database (illustrated in Figure 2 as

dotted line A). Now Dr. Valentine has either confirmed his invitation or subscribed himself to HeartMine, and is considered to have joined the HeartMine VO.

By integrating Sympa with Shibboleth, authentication is performed by each VO member's institutional Identity Provider and not by myVocs. What myVocs does is trust the federation to authenticate the session owner and provide a persistent identifier. In our prototype implementation, registered myVocs users must have an Identity Provider participating in the federation. Anyone who can authenticate successfully at a participating Identity Provider is authorized to create and manage VOs, and anyone who can be authenticated by a participating Identity Provider can join and participate in myVocs. But, what if some of Dr. Coeur's collaborators do not have the required Identity Provider? As a bootstrap mechanism, our project provides a service called OpenIdP.org [47] a Shibboleth-enabled Identity Provider that is a member of the InQueue Federation. Anyone with a working email address may enroll at OpenIdP.org.

Up to this point, our description of myVocs illustrates a use case for Figure 1: the Service Provider is myVocs, which protects the Shibboleth-enabled MLM resource. MLM users identify themselves using distributed authentication and the ePPN attribute released by each IdP is used as a global identifier. The MLM application stores the released ePPN along with the member's preferred email address.

Unlike most other open source MLMs, Sympa is built on a relational database backend. This was advantageous because the corresponding VOs, VO memberships, and VO roles were easily accessible via relational queries. Relational databases are a natively supported data store in Shibboleth so it was straightforward to make attributes managed by Sympa available for transport by Shibboleth. Tables 1–3 illustrate the myVocs backend database. The ePPN serves as the primary key for identifying myVocs participants. Table 1 shows the mapping of the user's preferred email address to ePPN. In the future we intend to introduce a separate myVocs primary key to use as an anchor for managing any number of external identifiers.

Table 2 shows the HeartMine VO membership table, which currently has two members. While myVocs currently utilizes ePPN as the primary key for identifying registered users, Sympa was written to use a unique email address as primary key and several internal functions are dependent on that design choice. This legacy design made it desirable to include both myVocs and Sympa primary keys in these tables. For example, Sympa has the ability to map multiple email addresses to the single address used as primary key so that people can use any email address they like on a per-list basis while being able to manage all their subscriptions and subscription preferences through a personalized web interface. Making use of this Sympa feature, each VO member can specify which email address or addresses should be used for HeartMine communications. The myVocs prototype also utilizes roles that are native to the Sympa MLM, such as ListOwner, ListModerator, ListEditor, and ListMember. (Table 3 is described further ahead in the section MyVOCS AND GRIDS).

### **myVocs as a Proxy Identity Provider and Attribute Aggregator**

A Shibboleth Identity Provider is responsible for authenticating its user community, but Shibboleth is agnostic regarding how that authentication actually occurs. Shibboleth's role is to assert that a valid authentication occurred and to release attributes about that authenticated user according to its release policy. In its role as the VO Identity Provider, myVocs relies on enterprises for authentication rather than providing its own authentication service. Since the exact authentication method used is external to Shibboleth, the myVocs IdP operates like any other IdP at the technical level; use of external authentication is a policy choice. What is special

about the myVocs VO IdP is that it is treated as a resource and is protected by the myVocs Service Provider. As a result, requests arriving at the myVocs VO IdP are redirected to federation's Identity Providers (Figure 2, Steps 3-4). Upon return from that IdP to myVocs, the principal's identifier is available (Figure 2, Step 9) and can be released to the VO Service Provider along with VO memberships associated with that identifier (Figure 2, Step 11).

myVocs supports at least two trust models. (1) myVocs can consume the attributes asserted by the user's home IdP, cache them, and re-assert these attributes, along with VO-managed ones, as appropriate and needed, or (2) myVocs can capture the signed attribute assertion issued by the user's home IdP, issue attributes of its own, and subsequently transmit both sets of attributes to the SP. In the first case, the SP fully trusts myVocs to re-assert the attributes it obtains from the user's home IdP. In the second case, it is up to the SP to consume and accept attributes from the user's home IdP. The choice of trust model is a deployment decision.

The message flow among the various myVocs components is illustrated in Figure 2 and described in full, below. Previous sections in this paper have described the myVocs one-time registration process; dashed arrow A represents initial entry of the global identifier. Arrow B indicates use of the Membership Management component (Sympa) to update VO and VO membership information.

1. A browser client requests a VO web resource protected by the VO Service Provider. If a security context for this principal already exists at this service provider, skip to step 14.
2. The client is redirected to the myVocs VO Identity Provider component (that is protected by the myVocs Service Provider component).
3. The client makes a Shibboleth Authentication Request to the VO Identity Provider. This request is similar to Step 1, above, but here the protected resource is the VO IdP itself. If a security context for this principal already exists, skip to step 10.
4. The client is redirected to the appropriate Identity Provider (for simplicity, ignoring a possible WAYF interaction).
5. The client makes a Shibboleth Authentication Request to the principal's preferred Identity Provider. If a security context for this principal does not exist at the Identity Provider, authentication occurs (details omitted) to establish the principal's identity.
6. The Identity Provider returns an authentication response to the client. The response may contain user attributes.
7. The client submits the authentication response to the Service Provider at myVocs, where it is validated. If necessary, the Service Provider queries the Identity Provider for attributes (7a).
8. The Service Provider updates its security context for this principal and redirects the client to the VO Identity Provider.

9. The client makes a Shibboleth Authentication Request to the VO Identity Provider at myVocs (the same request as at step 3 but this time a security context exists).
10. The myVocs IdP filters the attributes from the request header, stores the value of the identity attribute to the VO database (C), and returns an authentication response to the client. The response may contain any attributes available in the Federated Identifier and Attribute Database. Which attributes to release is a policy decision that can be implemented at any chosen level of granularity by configuration of the VO Identity Provider (D).
11. The client submits the authentication response to the VO Service Provider where the assertion is validated. If necessary, the VO Service Provider queries the VO Identity Provider for attributes (11a).
12. The VO SP updates its security context for this principal and redirects the client to the VO resource.
13. The client requests the VO resource (the same request issued at step 1 but this time the security context exists).
14. The resource filters the attributes from the request header, makes an access control decision, and returns the resource to the client.

The myVocs system treats the redirect for authentication and successful return as the out-of-band Shibboleth authentication step (Figure 2, Steps 4-9). In this manner, myVocs functions as an IdP proxy for a set of federated Shibboleth IdPs. As a proxy IdP, myVocs is able to aggregate a set of attributes useful for VO membership based access control. The aggregation can be self-managed and is thus highly scalable. Using myVocs, multiple unrelated security domains can provide a federated set of resources to each VO. In this manner, myVocs functions as a bridge (or proxy) between a federation of Shibboleth Identity Providers and a federation of Shibboleth Service Providers forming a federated set of distributed applications. Figure 3 illustrates this composite view of myVocs. The upper half of the figure shows the federated Identity Providers; the lower half of the figure shows one VO's provisioned applications and resources. The myVocs system federates the user's identifier, as provided by their IdP, with VO memberships and roles. MyVocs interacts with the federated IdP's in the role of Shibboleth Service Provider, and interacts with the federated applications in the role of Shibboleth Identity Provider.

### **myVocs Trust Management Framework Compared to Portals**

Portals are a popular method for providing a single web gateway to a set of users to log in and access documents, resources, and services. In addition to customized content presentation, portals provide an Application Programming Interface (API) that is intended to simplify integration of applications with a common runtime environment. The portal container provides a consistent environment across the set of integrated applications, termed portlets, with access to a user's identity and profile information and standard storage of persistent settings such as group memberships. A user's portal environment is activated when the user visits the portal and logs in. As mentioned previously, most portals are designed in a manner similar to the web applications described above: the user population is expected to reside in a single domain and

access control rules are local to the application (although in the portal case it is actually a suite of applications).

myVocs is a generalization of the authors' earlier work in enterprise single sign on for Grids [48]. It can be visualized as providing a consistent set of user, group, and role attributes to the myVocs SPs just as a portlet container provides a consistent user context to all portlets. myVocs distributes attribute context using Shibboleth rather than the shared runtime environment common to portals. In doing so, it gains the advantage of having applications distributed across multiple administrative domains while still receiving a trusted set of identity attributes. By leveraging federated identity infrastructure, myVocs can receive identities from external identity providers, aggregate them with locally defined VO attributes, and distribute the combined set to participating applications. A portal application, for example, could be one of the applications that is provisioned by myVocs as explained in the next section.

### **Provisioning Collaboration Resources**

Auto-provisioning of applications, accounts, roles, and access control makes myVocs a highly scalable service that can be managed autonomously. The myVocs demonstration environment includes a VO Management component that automatically creates a set of collaborative applications for each new VO. Creating a new VO automatically instantiates a wiki, a shared file space, and a content management system (CMS) in addition to the new mailing list. The applications chosen for the myVocs prototype were PHPwiki [49], Drupal [50], and WEBinstaFM Manager [51]. These particular applications were selected because they are web-based and provide functionality arguably essential for collaboration: joint content development, information and blog sharing and file sharing. Our goal was to demonstrate proof of concept: that a set of useful collaborative tools can be integrated with Shibboleth to produce a seamless working environment that identifies users and performs VO-based access control across unrelated security domains. In principal, any tool can be substituted or added to the ones we've selected, and a growing number of applications have already been Shibboleth enabled [52].

Essential modifications to each application were to modify the native account management, authentication, and authorization (AAA) systems and to integrate them with the security model provided by myVocs. Each of the modified applications, while different in collaborative functionality, shared similarity in their approach to AAA. Each application maintains an internal data store of identifiers, passwords, and roles. Typically, a default administrator role and password are assigned first. That administrator is then allowed to create users and roles and to assign access rules. In order to automatically provision applications, myVocs assigns the VO creator/owner to this administrator role. As VO members are added and removed, myVocs adds and removes these users to each application's data store, using the ePPN as the local account identifier. In the prototype myVocs, the native listserv roles (owner, moderator, editor, and member) were mapped onto each application's native set of roles. For example, a ListOwner has read and write permissions in the VO's instance of WebInstaFM, while VO members have read-only permission. If a VO member's role changes, myVocs updates each application's account store accordingly. VO members are unaware of this background activity. Our modifications to the selected applications make it possible to automatically provision accounts and application-specific roles, allowing the modified applications to simply respond to the attributes presented by the myVocs IdP. The application's internal authentication system is disabled and instead the Shibboleth authentication assertion is used to identify the session owner

myVocs also automatically provisions access control for the collaborative applications. This is currently accomplished by creating and inserting `mod_shib` access control statements instructing the web server to authorize access and account creation in each application based on membership in a specific VO. This approach requires that, in addition to the user's identifier, the VO membership and VO role attributes are also available to `mod_shib`. The identifier, VO membership and VO role are available in the myVocs Federated Identifier and Attribute Database and therefore are available to be transported by Shibboleth, but only if Service Providers hosting these applications are configured to use myVocs as the Identity Provider. Attempts to access any Shibboleth protected application at the VO Service Provider are redirected to the myVocs VO Identity Provider (Figure 2, steps 1-3). Upon return to the VO Service Provider (Figure 2, step 11), identifier, VO membership and VO role attributes are available to be used in the Service Provider's access control decision. The next section will describe how this takes place without any authentication service in myVocs or its security domain.

### **myVocs SCALABILITY**

Two important features of myVocs are (a) a single instance of myVocs can be used to manage many virtual organizations and (b) a single VO Service Provider is capable of serving multiple VOs. One possible myVocs deployment is its use as a federation-wide VO management service. In this scenario, Service Providers have flexibility in selecting the number of services and VOs to support at any one location. A single web server can, for example, provide unrelated applications to multiple VOs, or can provide multiple instances of a single application, each configured for use by a different VO. In the current myVocs prototype, all VO applications are created on the myVocs server; this was done for ease of auto-configuration, but is not an architectural requirement. The Shibboleth access control mechanism would work identically for any Service Provider in the Federation desiring to manage access based on VO membership attributes. The SP must be configured to use myVocs as the Identity Provider, and access to the application must be configured for restricted use by the VO. That configuration can be automated by insertion of additional `mod_shib` configuration parameters or by creation of a simple `.htaccess` file. A VO can assemble a highly variable set of services that each use myVocs and the underlying Shibboleth mechanisms for single sign-on and access control based on VO membership, but that are hosted in multiple locations and security domains. Automated application provisioning by myVocs in unrelated security domains is more difficult, but should be possible using a mechanism such as the Grid security infrastructure.

### **myVocs AND GRIDS**

While the work described in this paper focuses on middleware for web applications, certainly non-web applications can also benefit from federated identity and VO managed attributes. The GridShib project [53] uses Shibboleth to transport attributes from identity providers to resources in a grid infrastructure. The GridShib Certificate Authority (GridShib CA) [54] is a Shibboleth-enabled web application that allows an individual to obtain a short term X.509 credential by authenticating at their home Identity Provider. Upon successful authentication, the GridShib CA signs a newly minted short-term end-entity certificate and stores it in a well-known location on the user's system. Two implementations of the GridShib CA are provided, backed by either OpenSSL [55] or the MyProxy [56] online CA. The GridShib

project also provides tools that convert Shibboleth SAML assertions into a format useful for grid access control.

Like myVocs, the GridShib project hoped to benefit from distributed identity management, but the GridShib developers discovered that the attributes of greatest interest for grids—VO membership attributes—were not present in university directories. Once they learned about myVocs, a myVocs-GridShib integration project began [57,58]. As a result, the myVocs system was extended to allow association of the user's grid-issued Distinguished Name (DN) with their federated identifier. Two integration scenarios have been explored. In one approach, the GridShib CA is protected by a VO SP that belongs to the Shibboleth federation (like any other VO resource) and requires some type of database connector to the myVocs Federated Identifier and Attribute Database. Alternatively, the GridShib CA can be an integrated as a myVocs service. Either way, the software allows a user to acquire a short-lived Grid Credential through their web browser. After successfully authenticating at their home Identity Provider via Shibboleth protection of the CA, a credential suitable for use with Grid middleware such as Globus Toolkit 4.0 is installed. What myVocs provides is federation of the certificate's DN, their Grid identifier, with their ePPN and VO attributes. To support GridShib, the myVocs identity table was extended to add the GridShib CA issued DN, as shown in Table 3. When a VO member requests a Grid resource, the Grid software presents the user's Grid credential to the Globus software on the remote system; at this point in time the user is identified to the remote resource by their grid-issued DN only. The GridShib for Globus Toolkit component, configured to use myVocs as its Identity Provider queries myVocs requesting attributes associated with that DN. MyVocs locates the federated identifier associated with the DN and then determines the VO memberships and roles associated with that identifier. Those attributes are released to the GridShib for GT software, which then authorizes access based on VO membership and local policy, and logs the job appropriately. The advantages of myVocs integration with GridShib include web single sign on for web and grid services, and the use of short-lived end-entity certificates that simplifies certificate management.

## USING myVocs

The HeartMine collaborators are likely to think of myVocs as a membership management tool only, because myVocs is otherwise transparent. Dr. Coeur begins her workday by analyzing her previous day's experiments, blogging her thoughts into the HeartMine Content Management System. She can go directly to the HeartMine site without logging in through a portal. Her first attempt to access the blog will cause her familiar Clemson login screen to appear, followed by her blog. She knows that only the other members of HeartMine can read this page. Dr. Coeur remembers that the HeartMine collaboration has an annual report deadline within the week, so she again uses a saved URL to check the HeartMine WebInstaFM file share for the current, collaboratively edited version of the document. The HeartMine file share is hosted in a security domain that is unrelated to the domain hosting her Blog, but Dr. Coeur's security context is managed via Shibboleth so she does not need to authenticate again; Shibboleth is working behind the scenes to provide the correct security context at each site, in each domain. Dr. Coeur is not aware that her browser has followed a path of redirects, nor was she aware that her identifier and membership attributes were being made available (although this process can be exposed to protect privacy). Dr. Coeur's view of the HeartMine collaboration is summarized in Figure 4A. From the user's perspective, the myVocs Collaboration Environment provides an environment that makes her identity, VO memberships and roles

consistently and transparently available across a set of unrelated applications. If the HeartMine collaboration security environment were managed by a portal, Dr. Coeur would always have to first visit the portal home page portal, log in, and then access her applications. Because Shibboleth is being used to create a consistent, distributed security environment she can access applications directly and the login screen will appear (only if fresh credentials are required) without her having to navigate to it.

When Dr. Valentine begins his work day, he wants to check the HeartMine calendar for upcoming events and deadlines. Dr. Valentine prefers to use the myVocs search interface, shown in Figure 4E. This interface is not a portal; it is an application that converts short text strings like “cms heartmine” into the correct URL. As Dr. Valentine attempts to access the calendar, the web server’s `mod_shib` module intercepts the access attempt and redirects his browser to the myVocs VO IdP. Upon return, the browser carries a Shibboleth payload that is received and interpreted by `mod_shib`; this payload contains Dr. Valentine’s identifier and a list of his VO memberships and roles, as illustrated in Figure 4B. The web server module authorizes the principal’s access to the calendar page content, and makes the identifier and VO attributes available to the file share application as CGI environment variables, illustrated in Figure 4C. The application can provision an account for Dr. Valentine, if he didn’t already have one, because if he weren’t authorized to have an account he wouldn’t have access to the application.

Dr. Hart is the data mining expert member of the HeartMine collaboration. She will be using Grid data storage and compute clusters today as illustrated in Figure 4D. Dr. Hart begins her job submission and is prompted to authenticate at her home web single sign on service; in the background, a short-term public/private key pair has been created on her desktop and signed by the GridShib CA. As job submission begins her digital certificate is presented to the remote Grid software. The Globus GridShib plug-in extracts Dr. Hart’s DN and queries the myVocs VO IdP for additional attributes, receiving a response payload identical to the one shown in Figure 4B. The job begins its run and the use log shows it as Dr. Hart’s job; this is important because she wants to be contacted by the Help Desk if there are any problems. The accounting file logs the job as HeartMine use, so that the run time can be charged to the HeartMine Community Grid Allocation.

### **myVocs Use Today**

The myVocs prototype is available for download as a stand-alone distribution [59], and we have identified some user communities interested in using myVocs as their collaboration environment. We are also collaborating in planning a TeraGrid testbed that will demonstrate and evaluate attribute-based access control for Grids [57]; myVocs is one of the approaches that has been chosen for deployment in this TestBed [60]. VOMS and PERMIS have each recently added a SAML AA interface. These features were not available while the work described in this paper was being done, but adoption of this interface indicates a growing consensus among the community of distributed authentication/authorization system developers in the utility of Shibboleth for grids, an approach pioneered in myVocs.

The UABgrid project [61,62] was designed to use the UAB campus authentication service and campus identifiers for web-based access to grid resources. That architecture is now being re-engineering for integration with myVocs.

## LIMITATIONS, REMAINING CHALLENGES AND FUTURE WORK

The main limitation of myVocs is that the prototype software implementation caches the content of attribute assertions rather than the entire signed assertion. This implementation was a choice made for ease of software development, and as a result myVocs trusts the campus to correctly identify the principal and the VO Service Provider trusts that myVocs is correctly federating the asserted identifier with VO managed attributes. In the current implementation, the SP assumes and trusts that myVocs is the source of all user attributes. However, by adding the ability to cache the entire digitally signed attribute assertion the SP would be able to directly verify the signature and then determine whether to trust the assertion contents. The resulting set of attributes, available to a SP at the time the user requests access to the resource, may be aggregated without impersonating the attribute owner. Other trust models are possible, and these should be explored.

Other concepts being developed are an enhanced IdP discovery mechanism that would employ the user's login history to suggest the appropriate Identity Provider; and a myVocs identity aggregation function that would assign each user a myVocs persistent identifier so that users could aggregate their multiple persistent identifiers and to whom they are released Shibboleth-enabled multimedia tools, especially video and voice over IP standards-based products, would provide valuable additions to the collaboration toolkit available to VOs. Previous work in this area discovered two hard problems: (a) the H.323 protocol had a flawed security design and (b) the SIP protocol had only MD5 hash security and TLS defined [63]. While some work on asserted identity has been done for SIP [64,65], there has been less progress for the H.323 protocol. However, the GridShib CA-myVocs solution that essentially converts an act of authentication into a short-term digital certificate could be used in conjunction with H.323 Annex E, a digital signature security profile. Federation peering remains an important issue to resolve, especially in support of international collaborations.

## Acknowledgments

Work described in this paper was supported by NSF ANI-0330543 "NMI Enabled Open Source Collaboration Tools for Virtual Organizations" (Gemmil (PI), Robinson, *et al.*), with partial support from N01-LM-3-3513 "Advanced Network Infrastructure for Health & Disaster Management" (Orthner (PI), Terndrup, Grimes, Gemmill, *et al.*). The GridShib work is funded by the NSF National Middleware Initiative (NMI awards 0438424 and 0438385). Opinions and recommendations in this paper are those of the authors and do not necessarily reflect the views of NSF. Special thanks to collaborator Von Welch (NCSA) for his work on myVocs-GridShib integration, and to excellent critique from our anonymous reviewers. Collaborators in this project include Jason L.W. Lynn (UAB); Shibboleth team members Bob Morgan, Ken Klingenstein, Scott Cantor and Nate Klingenstein; Sympa developers Serge Aumont and Olivier Salaun (Comité Réseau des Universités); members of the Internet2 MACE-MLIST Working Group, and Charlie Catlett (TeraGrid).

## REFERENCES

1. Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of Supercomputer Applications* 2001; 15(3).
2. Reed DA. Grids, the TeraGrid and beyond. *Computer* 2003; 36(1): 62-68.

3. Yahoo Groups. <http://groups.yahoo.com/> [26 August 2005].
4. Google Groups. <http://groups.google.com/> [26 April 2007].
5. JSR 168: Portlet Specification. <http://www.jcp.org/en/jsr/detail?id=168> [10 April 2006].
6. Pluto 1.1 Reference Implementation of the Java Portlet Specification. <http://portals.apache.org/pluto/> [26 February 2006].
7. Jetspeed. <http://portals.apache.org/jetspeed-1/> [26 February 2008].
8. Microsoft Windows 2000 Server Active Directory. <http://www.microsoft.com/technet/prodtechnol/windows2000serv/technologies/activedirectory/default.mspix> [26 February 2008].
9. White Paper: Technical Overview of Windows Server 2003 Active Directory. <http://support.microsoft.com/kb/818626/en-us> [26 February 2008].
10. National Science Foundation Middleware Initiative (NMI). <http://www.nsf-middleware.org/> [14 April 2005].
11. Alfieri R, Cecchini R, Ciaschini V, Dell'Agnello L, Frohner A, Gianoli A, Lorentey K, Spataro F, Fernandez Rivera F, Bubak M, Gomez Tato A, Doalio R. VOMS, an Authorization System for Virtual Organizations. *Lecture notes in computer science 2003*; 2970(Grid computing (Santiago de Compostela, 13-14 February 2003, revised papers)): 33-40.
12. Alfieri R, Cecchini, Ciaschini, Dell'Agnello L, Frohner A, Lorentey K, Spataro F. From gridmapfile to VOMS: managing authorization in a grid environment. *Future Generation Comp.Syst.* 2005; 21(4): 549-558.
13. eduPerson Object Class. <http://www.educause.edu/eduperson/> [4 March 2006].
14. Morgan RLB, Cantor S, Carmody S, Hoehn W, Klingenstein K. Federated Security the Shibboleth Way. *EDUCAUSE Quarterly* 2004; 27(4): 12-17.
15. Shibboleth Web Site. <http://shibboleth.internet2.edu/> [12 December 2005].
16. Gemmill J. 2006 *Trust-Relationship Management Framework for Federated Virtual Organizations*. Ph.D. Dissertation.
17. Gemmill, Jill, Robinson, John-Paul, and Shealy, David. 2003 *NMI Enabled Open Source Collaboration Tools for Virtual Organizations*. Grant Proposal, <http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0330543> [12-23-2005].
18. Foster I, Kesselman C. Globus: A Metacomputing Infrastructure Toolkit. *International Journal of Supercomputer Applications* 1997; 11(2): 115-128.
19. Foster I. Globus Toolkit Version 4: Software for Service-Oriented Systems. Springer Verlag (LNCS):2-13.
20. Chadwick DW, Otenko A. The PERMIS X.509 role based privilege management infrastructure. 2002; Symposium on Access Control Models and Technologies 135-140.
21. Chadwick D, Otenko A, Ball E. Role-based access control with X.509 attribute certificates. *Internet Computing, IEEE* 2003; 7(2): 62-69.
22. Foster I, Kesselman C, Tsudik G, Tuecke S. A security architecture for computational grids. 1998; Conference on Computer and Communications Security 83-92.
23. International Telecommunication Union (ITU). Recommendation C.509 The Directory: Public-Key and Attribute Certificate Certificate Frameworks. 2000.
24. RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. <http://www.ietf.org/rfc/rfc3280.txt> [13 October 2006].
25. Jokl J, Basney J, Humphrey M. Experiences using Bridge CAs for Grids.7-8-2004.;

26. Alterman P, Weiser R, Gettes M, Stilson K, Blanchard D, Fisher J, Brentrup R, Norman E. The EDUCAUSE - NIH PKI Interoperability Pilot Project. *Proceedings of the 1st Annual PKI Workshop 2002*; 177-191.
27. Open Grid Computing Environments Collaboratory (OGCE). <http://www.collab-ogce.org/nmi/index.jsp> [16 February 2006].
28. Novotny J, Russell M, Wehrens O. GridSphere: a portal framework for building collaboratoin. *Concurrency and Computation-Practice & Experience 2004*; 16(5): 505-513.
29. RFC 3281: An Internet Attribute Certificate Profile for Authorization. <http://www.faqs.org/rfcs/rfc3281.html> [13 October 2006].
30. Open Grid Forum Web Site. <http://www.ogf.org/> [16 October 2006].
31. Security Assertion Markup Language (SAML) v1.1 [OASIS 200308]. <http://www.oasis-open.org/specs/index.php#samlv1.1> [13 October 2006].
32. Profiles for the OASIS Security Assertion Markup Language (SAML) V 2.0. <http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf> [26 February 2008].
33. Organization for the Advancement of Structured Information Standards (OASIS). <http://www.oasis-open.org/home/index.php> [14 April 2005].
34. Extensible Markup Language (XML). <http://www.w3.org/XML/> [14 April 2005].
35. Security Services Technical Committee. [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=security](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security) [31 October 2006].
36. Liberty Alliance Project. <http://www.projectliberty.org/> [14 April 2005].
37. Liberty ID-FF Architecture Overview. <http://www.projectliberty.org/specs/draft-liberty-idff-arch-overview-1.2-errata-v1.0.pdf> [14 April 2005].
38. Security Assertion Markup Language (SAML) version 2.0. <http://www.oasis-open.org/specs/index.php#samlv2.0> [14 April 2005].
39. InCommon Federation Web Site. <http://www.incommonfederation.org/> [16 October 2006].
40. Inqueue Federation. <http://inqueue.internet2.edu/> [16 October 2006].
41. TestShib. <http://testshib.org/index.html> [7 November 2006].
42. Family Educational Rights and Privacy Act (FERPA). <http://www.ed.gov/policy/gen/guid/fpco/ferpa/index.html> [23 August 2005].
43. RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1. <http://www.w3.org/Protocols/>
44. Simple Object Access Protocol (SOAP) 1.2. <http://www.w3.org/TR/soap/>
45. Shibboleth Technical Overview, Working Draft 02. <http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf> [13 October 2006].
46. Sympa Mailing List Manager. <http://www.sympa.org/> [19 February 2006].
47. OpenIdP.org. <http://www.openidp.org/> [28 February 2006].
48. Robinson J-P, Gemmill J, Joshi P, Bangalore P, Chen Y, Peechakara S, Zhou S, Achutharao P. Web-Enabled Grid Authentication in a Non-Kerberos Environment.
49. phpwiki. <http://sourceforge.net/projects/phpwiki/> [20 February 2006].
50. drupal. <http://drupal.org/> [20 February 2006].
51. WebInsta FM Manager. <http://www.webinsta.com/fm.php> [20 February 2006].
52. Shibboleth Enabled Applications and Services. <https://wiki.internet2.edu/confluence/display/seas/Home> [26 February 2008].
53. GridShib: Grid-Shibboleth Integration. [http://www.globusworld.org/2005Slides/Session%201b\(1\).pdf](http://www.globusworld.org/2005Slides/Session%201b(1).pdf)
54. GridShib CA. <http://gridshib.globus.org/docs/gridshib-ca-0.5.0/> [25 February 2008].

55. OpenSSL. <http://www.openssl.org/> [25 February 2008].
56. Basney J, Humphrey M, Welch V. The MyProxy Online Credential Repository. *Software: Practice and Experience* 2005; 35(9): 801-816.
57. GridShib-myVocs Integration: Federated Identity and Attribute-Based Access Control for Grids. <http://grid.ncsa.uiuc.edu/presentations/i2mm-myvocs-gridshib-april06.ppt> [9 March 2008].
58. myVocs-GridShib Wiki. <https://spaces.internet2.edu/display/GS/MyVocs> [26 February 2008].
59. myVocs Box. <http://myvocs-box.myvocs.org/> [26 February 2008].
60. Welch V, Foster I, Scavo T, Siebenlist F, Catlett C, Gemmill J, Skow D. Scaling TeraGrid Access: A Testbed for Identity Management and Attribute-based Authorization.
61. UABGrid: A Campus-Wide Distributed Computational Infrastructure. <http://staff.psc.edu/lfm/PSC/Grid/PGS-RG/GridsOnCampus/Gemmill.pdf>
62. UABgrid web site. <http://uabgrid.uab.edu/> [29 November 2005].
63. Gemmill J, Srinivasan A, Lynn JLW, Johnson TM, Verharen E, Tulu B, Abhichandani T. Middleware for Scalable Real-time Multimedia Communications Cyberinfrastructure. *Journal of Internet Technology* 2004; 5(4): 405-420.
64. Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP) [draft-ietf-sip-identity-01]. <http://tools.ietf.org/html/draft-peterson-sip-identity-01> [13 October 2006].
65. SIP SAML Profile and Binding draft-ietf-sip-saml-00.txt. <http://www.ietf.org/internet-drafts/draft-ietf-sip-saml-00.txt> [19 October 2006].

**TABLES**

<b>Identifier (ePPN)</b>	<b>Preferred email</b>
coeur@clemson.edu	coeur@dept.clemson.edu
xyz1234@myu.edu	valentine@myu.edu

**Table 1 myVocs Registered Members**

<b>Identifier (ePPN)</b>	<b>Preferred email</b>	<b>List Address(es)</b>	<b>MemberRole</b>
coeur@clemson.edu	coeur@dept.uab.edu	coeur@clemson.edu	ListOwner, ListMember
xyz1234@myu.edu	valentine@myu.edu	valentine@myu.edu, dr.v@gmail.com	ListMember

**Table 2. myVocs HeartMine VO membership Records**

<b>Identifier (ePPN)</b>	<b>Preferred email</b>	<b>DN</b>
coeur@clemson.edu	coeur@dept.clemson.edu	CN=couer@clemson.edu, O=Shibboleth User, DC=computer, DC=ncsa, DC=uiuc, DC=edu
xyz1234@myu.edu	valentine@myu.edu	CN=xyz1234@myu.edu, O=Shibboleth User, DC=computer, DC=ncsa, DC=uiuc, DC=edu

**Table 3. myVocs Registered Members**

TOP

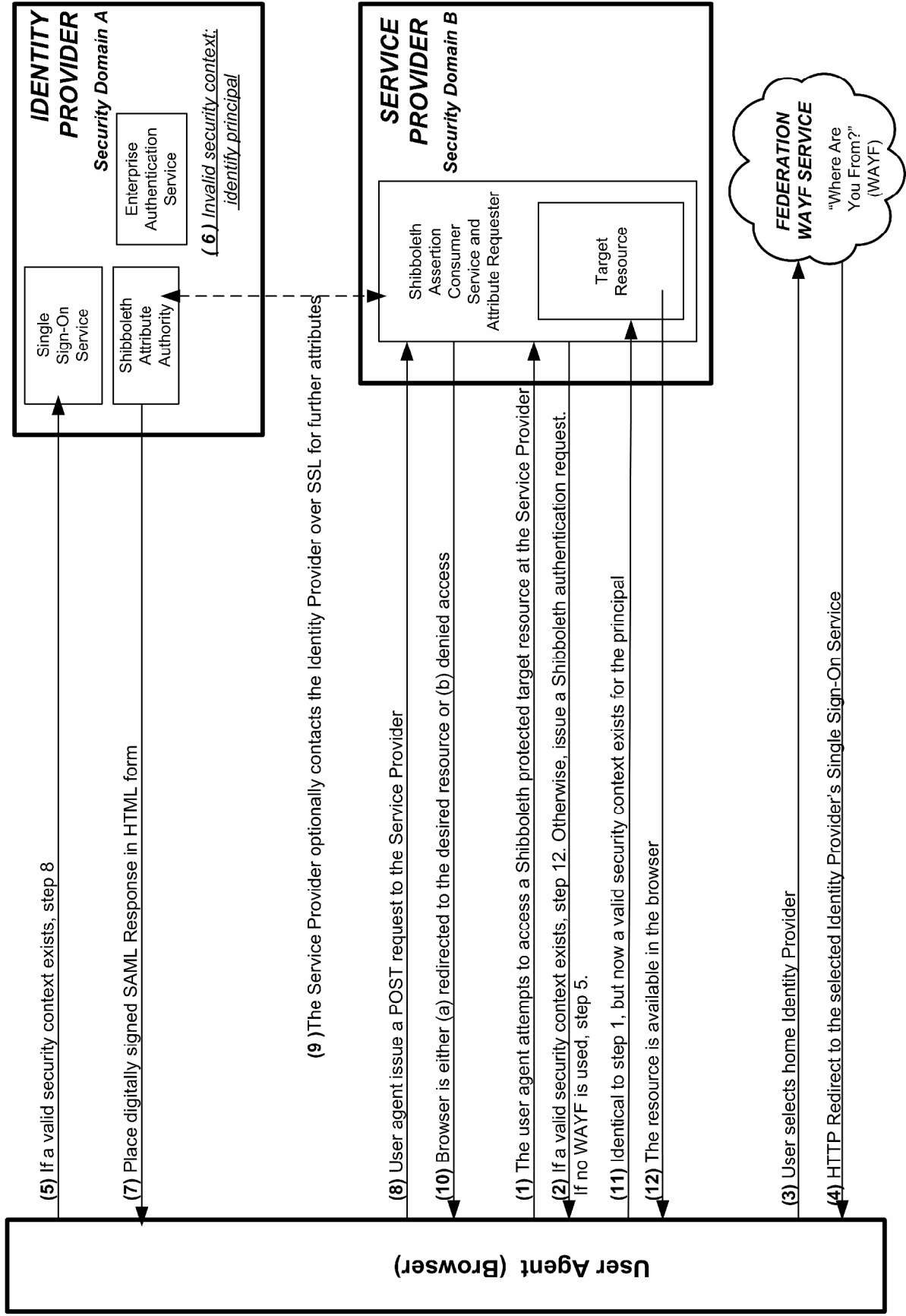
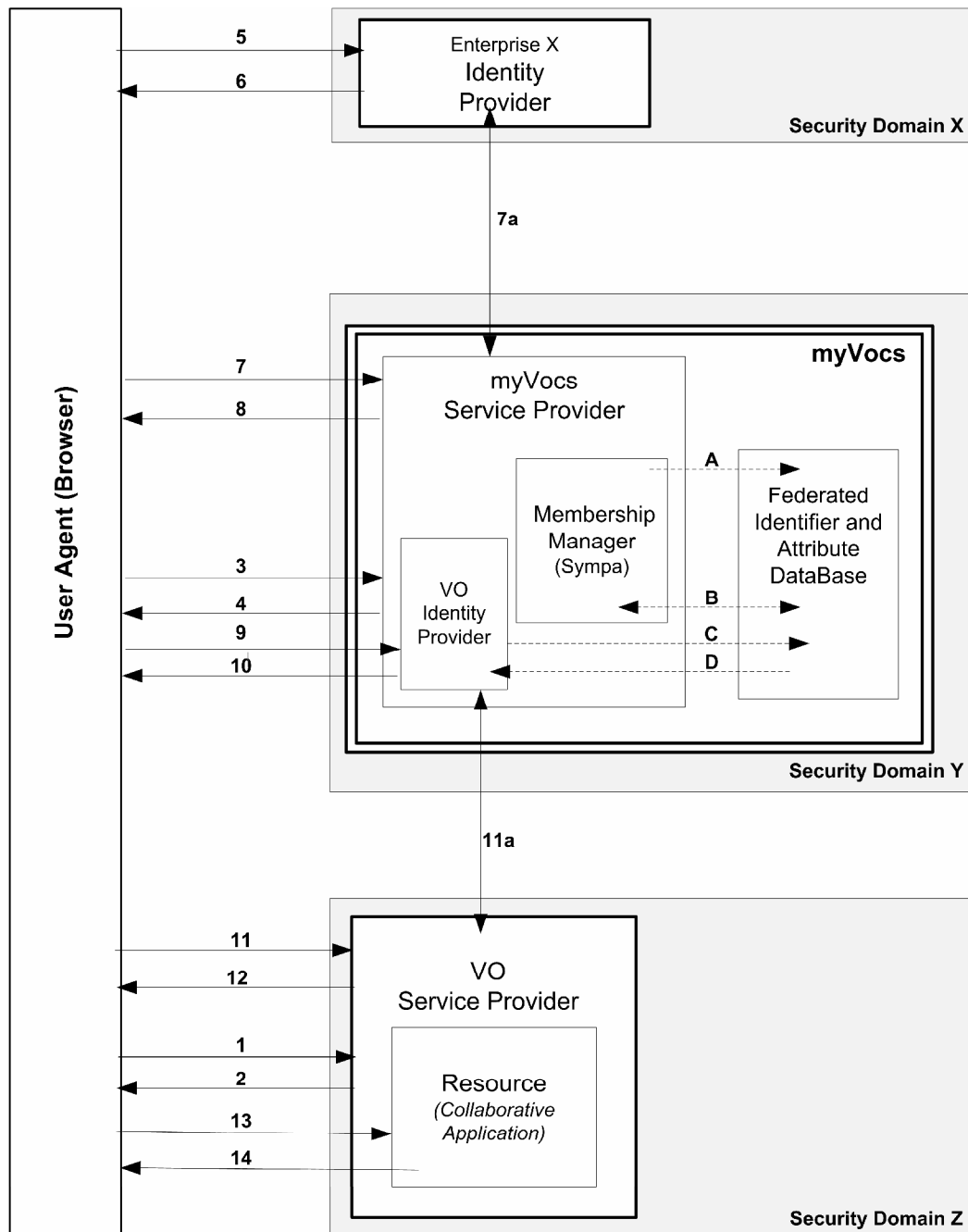
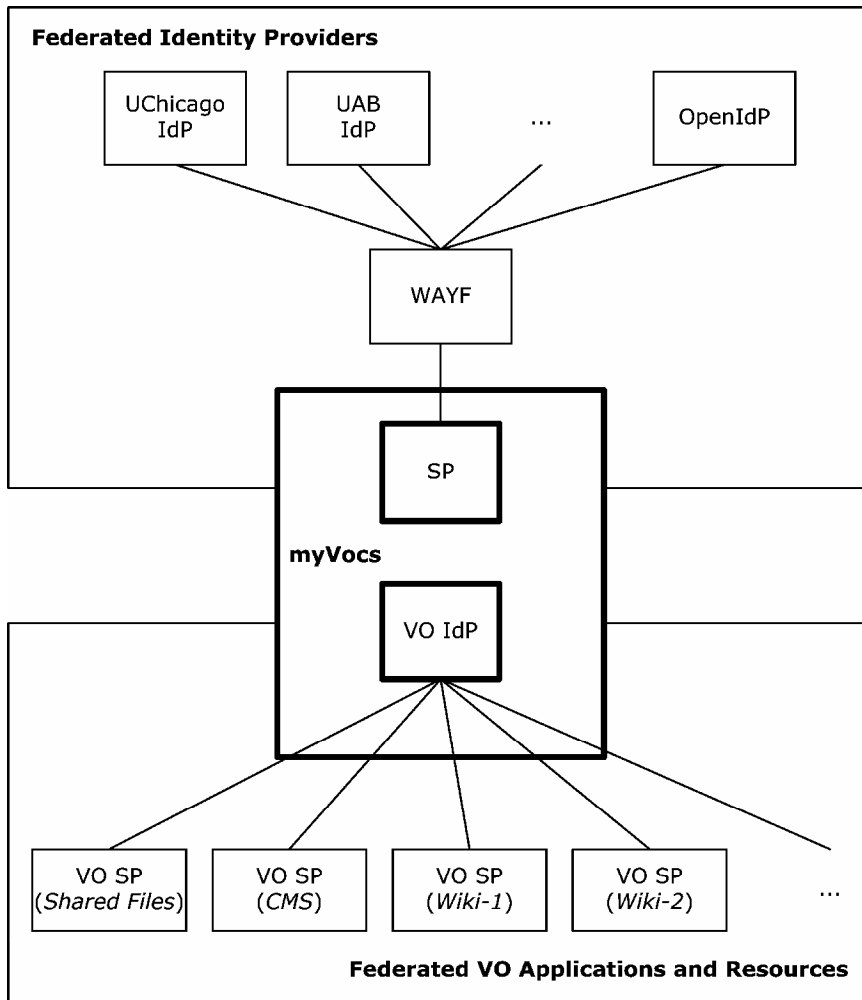


Figure 1. Shibboleth Message Flow



**Figure 2. myVocs Architecture Components and Message Flow.** Message flow is detailed in section “myVocs as a Proxy Identity Provider and Attribute Aggregator”.



**Figure 3. myVocs represented as a bridge between a federation of Identity Providers and a set of federated VO Service Providers.**

