IVOM: Interoperability and Integration of VO Management Technologies in D-Grid

Work Package 1: Evaluation of international Shibboleth-based VO Management Projects

Report v1.1, 7 June 2007

Peter Gietz, DAASI International GmbH, Tübingen
Christian Grimm, RRZN und L3S, Leibniz Universität Hannover
Ralf Gröper, RRZN und L3S, Leibniz Universität Hannover
Martin Haase, DAASI International GmbH, Tübingen
Siegfried Makedanz, AWI, Bremerhaven
Hans Pfeiffenberger, AWI, Bremerhaven
Michael Schiffers, Ludwig Maximilian University Munich

1 Introduction

The initial work package in the IVOM project [IVOM06] comprises the evaluation of selected VO management systems to understand their suitability for efficiently supporting Shibboleth as an authorization infrastructure in Grids. As Shibboleth lacks an appropriate concept of representing Virtual Organizations (VO), several international projects work on solutions for this key Grid requirement.

It is our objective to critically review the deliverables of these projects for suitability in IVOM and the D-Grid community projects. During the project kick-off several candidate projects were identified: gLite Shibboleth integration, GridShib, MAMS, myVocs, PERMIS, VOMS and VOMRS. This report contains the reviews and a comparison of these respective products or projects.

2 SAML and Shibboleth

2.1 Short Description

The *Security Assertion Markup Language* (SAML) [SAML] is an XML-based standard issued by OASIS [OASIS], the *Organization for the Advancement of Structured Information Standards*. It serves as a framework for exchanging authentication, entitlement and attributes information.

Based on the foundation of SAML, Shibboleth [Shib] is a software package for building federated *Authorization and Authentication Infrastructures* (AAI). It supports features such as *Single Sign on* (SSO) across organizational boundaries. The three major components of Shibboleth are:

- The Identity Provider (IdP) serves as a Policy Information Point (PIP). Each user has a home organization providing its own IdP. This IdP is responsible for managing the user's metadata and making it available to SPs belonging to the same federation.
- Shibboleth Service Providers (SPs) are arbitrary web-based resources that are protected by Shibboleth mechanisms. This means that a user trying to access such a resource is being forwarded to his home organization's IdP for authentication instead of being authenticated by the resource itself.
- The Where Are You From (WAYF) Service allows users to select their IdP, as it is impossible for an SP to determine the user's IdP automatically. Each federation has exactly one WAYF (or none if the federation has only one IdP).

This federated approach makes Shibboleth an ideal enhancement for Grid environments as VOs tend to be dynamically formed out of organizationally and geographically distributed users. By leveraging an existing (not necessarily Grid-specific) Shibboleth infrastructure, or implementing a new one specifically for Grid use, the creation of new VOs is simplified as user-specific data, such as name, address or affiliation, do not need to be re-acquired. Furthermore, IdPs can manage arbitrary attribute/value pairs. Thus IdPs can also manage VO-affiliation and roles within a VO and make this information available to Grid resources, which, in that case, act as Shibboleth SPs.

Despite the advantages of using Shibboleth in Grids there are some problems that need to be addressed:

- If an existing non Grid-aware federation is used it is not feasible to manage Grid-specific attributes like VO-membership by these IdPs. There is thus the need for another Attribute Authority (AA) specifically for Grid attributes. This can either be an additional Shibboleth IdP or another service like VOMS. This scenario is not yet covered by standard software but several international projects like myVocs and the efforts made by SWITCH described later in this document deal with it.
- When SPs pull attributes from the IdP, the IdP discovery problem occurs. This means that an SP might not know which IdP is responsible for the user whose request it received. This problem can either be solved by embedding information identifying the user's IdP or by embedding all necessary attributes into the request, i.e. by pushing all available attributes.

- When pushing attributes to the SP, e.g. by embedding it to the user's proxy certificate in a Grid context, a problem similar to the IdP discovery problem occurs: The SP discovery problem. This means that it might not be known a priori which attributes will be needed by the SPs involved for authorization of the current request. It is thus not known which of the available attributes are to be pushed to the SP. This problem can also be solved in two ways: First it is possible to embed all available attributes into the job. This solution is not favorable as it unnecessarily increases the number of attributes that need to be embedded and it also raises confidentiality concerns as not all attributes associated with a user might be suitable for public exposure. The second solution is to agree upon a set of attributes within a VO that are sufficient for making authorization decisions on the Grid resources and pushing only those to SPs.

2.2 Evaluation

Shibboleth is the state of the art software for building federated AA-Infrastructures. Today, many international AAI projects like MAMS [MAMS], the German DFN-AAI [DFNAAI], the Swiss SWITCH AAI [SWaai] and the Finnish Haka federation [Haka] use Shibboleth. As compatibility to national and international AAIs is a key aim of IVOM, the use of at least some Shibboleth concepts and/or components for building a D-Grid AAI is virtually inevitable.

3 Shibboleth and VO Management

Shibboleth's federation model is two-tiered. The core concept of Shibboleth-based authorization is to have a single source of authority per user, which is the Identity Provider, based on the identity management at the user's home institution. A Service Provider may only request a given user's attributes from a single Identity Provider, the one it gets to know through the user's WAYF selection.

With the wide acceptance of Shibboleth it was adapted in authentication and authorization realms outside the space of inter-institutional sharing of web resources. One such field was Grid AAI. The representation of virtual organizations in Shibboleth is a major issue for the integration of Shibboleth and Grid middleware. The „Grid problem" – as the specific problem that underlies the Grid concept – has been identified as flexible, secure, coordinated resource sharing and problem solving among dynamic collections of individuals, institutions, referred to as virtual organizations [FKT01].

A virtual organization is a source of authority of its own. Users have specific roles in it and it confers specific rights to users. So, to make a well-informed access control decision based on all available attributes of a user, a SP would have to request assertions from the home institution and the VO. Due to Shibboleth's architecture this is not feasible. Besides, it would also raise the IdP discovery problem [ShibDS].

Therefore the additional source of authority had to fit into the given model. Based on previous work by Von Welch [Wel05], the MAMS project and the myVocs project four options were identified to achieve this:

- VO management at the home institutions, based on participant's agreement on attributes, VO-specific information is located at the member's home institution. It is a moot point if institutions would accept modifications in their identity management systems. However, the major problem is trust (see chapter Trust Issues below): the home IdP is generally not the authoritative source for this information.
- The VO operates its own IdP, which means extra work to run separate Identity Management (IdM) systems and services. This approach would undermine the advantages of the Shibboleth concept of identity federation.
- Decentralized VO management: VO attributes are centrally managed by the VO and stored distributed in the institutional IdM systems. The Internet2 tools Grouper [Grouper] and Signet [Signet] may in the future be the appropriate provisioning tools. This approach would need a new set of trust relations and associated policies, e.g. on attribute or namespace usage. A proof of concept is an open issue.

- IdP Proxy: VO management hooks into the communication flow between IdP and SP by acting as a SP when facing the IdP and acting as an IdP when facing the SP. Thereby it gathers the user's home attributes, adds the VO-specific attributes and presents the resulting conglomerate assertion to the SP. This is the solution chosen by the developers of myVocs and IAMSuite, the MAMS VO system.

As already pointed out, the management of virtual organizations was not on the agenda of the Shibboleth architects. This might change in the future as a discussion on a Shibboleth delegation profile has begun and the roadmap for Shibboleth v2 and beyond mentions applicable features. These extension requests are not yet assigned to a specific future version. If realized, however, they would free VO management in Shibboleth from being "the man in the middle" in the authorization process [Shib2] through the ability:

- to link accounts between multiple Identity Providers and
- to use linked accounts to gather attributes from multiple Identity Providers.

3.1 Multiple VO Memberships

Users are often members of multiple VOs. It is a standard requirement in Grid middleware that a user may use the rights from those VOs simultaneously [DGV, MVO05, YBC+07]. In Shibboleth terms this requirement would translate into the combined release of a user's attributes from all his VOs.

In production grids it is known that a user in most cases does not know which VO membership or entitlement is required to access a specific resource. Operating experience in higher education computer centers has shown that the majority of users does not care what attributes or entitlements they have, they just want to get their work done without running into access control barriers. We expect the typical user's habit to be the complete release of all available attributes as long as there are no perceptible consequences.

The IdP Proxy approach allows for the complete release of a user's attributes from all his VOs, if all VOs are managed in one place. However, a massive problem occurs, if a user's VO memberships are managed at different VO management systems. The user may only select one IdP proxy and would only get a assertion containing the attributes from one VO management system. We refer to this as the „IdP Proxy problem". Today this problem can only be avoided by using one VO management system in a community or a federation.

3.2 Trust Issues

The TrustCoM project [TCM] recently remarked that VO management as developed for academic Grids, "*has previously only addressed membership issues and has previously ignored Trust, Security and Contract management issues*" [Trust05]. The introduction of Shibboleth and its federation concepts into Grids will at least address the trust issue.

Trust in Shibboleth is based on the respective federation policy. In addition special arrangements may be made between IdPs and SPs. In practice, trust is utilized when an IdP releases user attributes at the request of a known SP and signs the assertion to confirm the reliability of the information contained. In other words: Information is requested, released and consumed in a bilateral, trustful communication process.

VO management in Shibboleth as described above adds a third role to the process. The available systems intermediate the user's IdP to a SP. This can be done in different ways:

1. The home IdP's original assertion is included in the VO-generated assertion.
2. The VO extracts the attributes and includes them in its assertion.

The first solution has the advantage that the assertion the user's home IdP created is passed on unchanged. Still, to the SP the VO is the issuer of the conglomerate assertion and would be the first to contact in case of a problem. We see this as the best practice currently feasible. In general, the communities who will adopt this approach should be aware of the implicit trust issues.

The second solution leads to a core problem in distributed systems: Who do you trust to say what about what or whom [Mor06]. Here, the VO acts as if it were the authoritative source for these

attributes. We consider this to be a bad practice and to be potentially dangerous for each VO provider as well as for the trust fabric of a federation in its entirety.

4 gLite and Shibboleth: Work done by SWITCH

4.1 Short Description

The Swiss Research Network SWITCH has done work [FTLW07] in the area of providing interoperability between gLite VOMS and Shibboleth as part of the EGEE-II project[1].

The aim of this project is to integrate two sources of attributes: First, a Shibboleth IdP being part of a nationwide AA-Infrastructure provides general information about users, e.g. name, home organization and telephone number. Second, Grid-specific attributes are provided by a gLite VOMS. In order to make attributes of both AAs available to Grid resources for authorization decisions several possible implementations were evaluated by SWITCH:

- Embed Shibboleth attributes in attribute certificates (AC): This approach implies the need for changes in the IdP-Software to issue VOMS-compatible ACs in addition to SAML assertions.
- Embed Shibboleth attributes in the user's X.509 certificate: This approach does not allow exposing only a minimally required subset of all available attributes but always exposes all of them. Furthermore the whole certificate would have to be revoked if only one attribute changes.
- Grid resources request user attributes from the IdP: This approach implies the need for adding SAML processing functionality to all Grid resources.
- Store Shibboleth attributes in VOMS: This approach uses the newly implemented functionality in VOMS to store arbitrary attribute-value-pairs. Shibboleth attributes can either be pulled from the IdP by the VOMS or they can be pushed to the VOMS by some means. The pull model is not feasible as it implies changes to the IdP software to support delegated access. For the push model means have to be found to actually execute the push process.

The SWITCH project chose to store the Shibboleth attributes in VOMS using the push model. In order to allow for the pushing of Shibboleth attributes to VOMS they implemented the *VOMS Attributes from Shibboleth* (VASH) service. This service allows Grid users to control and initiate the transfer of Shibboleth attributes to a VOMS in form of a web-based application. One dedicated VASH service is needed for every desired combination of a Shibboleth federation and a VO managed by a VOMS.

In this implementation, Shibboleth attributes are transferred from the IdP to the VOMS once per semester. The users themselves initiate the transfer by accessing the VASH web front end and they have full control over which attributes are transferred from the IdP to the VOMS. After a lifetime of six months the attributes expire in the VOMS and the user receives a notification per e-mail to refresh his attributes.

This limitation has been caused by the current Shibboleth 1.3 version not offering a sufficient framework to implement more frequent automatic attribute updates. Building upon SAML 2.0 and the yet unreleased Shibboleth 2.0 it is planned by SWITCH to overcome this limitation and introduce attribute refresh times in the order of days or less. In that case users have to consent to the release of their attributes only once. Subsequent attribute updates will not need further user interaction.

Additionally, the VASH service is already capable of working together with SLCs and can thus be used in conjunction with a GridShib CA.

4.2 Evaluation

The authors of this document have not tested the VASH service itself, as the software is not publicly available. The approach taken by this project is viable and meets the requirements the

[1] http://egee-technical.web.cern.ch/egee-technical/

SWITCH project was initiated upon, i.e. usage in academic environments where Shibboleth attributes tend to change only at the beginning of a new semester. It has yet to be evaluated if its concepts or the VASH software itself can be integrated in the middleware-spanning VO-management concept to be developed by IVOM.

The main point of critique is the fact that Shibboleth attributes are replicated in the VOMS in time intervals in the order of months, defaulting to half a year. If Shibboleth attributes of a user change in that time, this change is not propagated to the VOMS. It is thus possible that authorization decisions on Grid resources are made based upon outdated attributes. The project does address this problem by assuming that the attributes managed by Shibboleth are not subject to frequent change. This is generally true for attributes like the user's name, but authorization decisions will more likely rely on attributes as affiliation to a university, institute or project. If that affiliation ends, a user might not be eligible to use Grid resources. Using the VASH approach this information might not be available to Grid resources for several months.

As SWITCH plans to overcome this limitation using Shibboleth 2.0 after its release, the VASH service needs to be re-evaluated at that time.

5 Globus Toolkit and Shibboleth: GridShib

5.1 Short Description

GridShib [GridShib] is being developed as part of the Globus Project [Globus]. It is a collection of software aimed at allowing grid resources to make authorization decisions based on attributes managed by Shibboleth federations, i.e. by Shibboleth IdPs. Furthermore it includes functionalities to enable users to access grid resources without the need for long-lived certificates issued by a certificate authority (CA). The components and their dependencies can be seen in Figure 1.

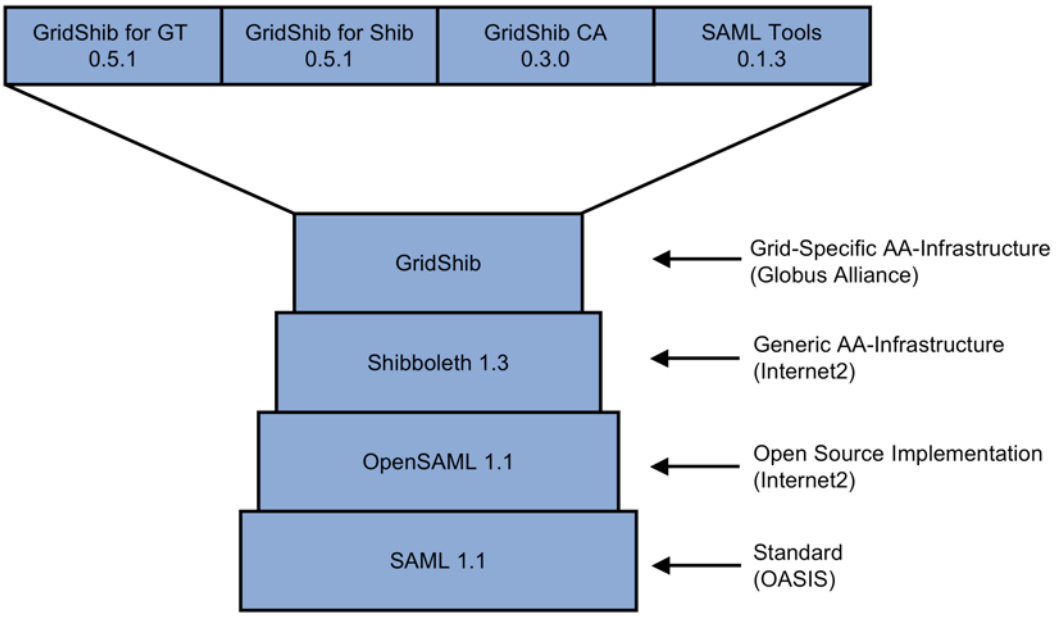


Figure 1: Components of the GridShib architecture with current version numbers

The four main components of GridShib are:

- GridShib for Globus: This component includes a *Policy Decision Point* (PDP) for web services in Globus such as WS-GRAM and RFT. This PDP makes authorization decisions based on Shibboleth attributes. At the current state this functionality is limited to "Yes-or-No" decisions. It is e.g. not possible to choose a specific batch queue based on Shibboleth attributes. The PDP does also contain an interface to query these attributes directly from an IdP. This implies the aforementioned IdP discovery problem. Evaluation of pushed attributes is planned for the near future but not yet available. Furthermore it is not yet

possible to make authorization decisions solely based upon Shibboleth attributes: As there is no concept in the Globus Toolkit similar to gLite's pool-accounts there is still need for a one-to-one mapping of Grid identities to local accounts. The grid-mapfile must thus still be present and contain the DNs of all authorized users.

- GridShib for Shibboleth: This component has to be installed together with the IdPs if attribute pull on the Grid resources is used. As the SAML assertion identifying the user is not present on the Grid resources (only the user's DN from his credentials is available), it is not possible for these Grid resources to query the IdP directly. GridShib for Shibboleth fills this gap and serves as the glue between the Grid services and the Shibboleth IdP software.

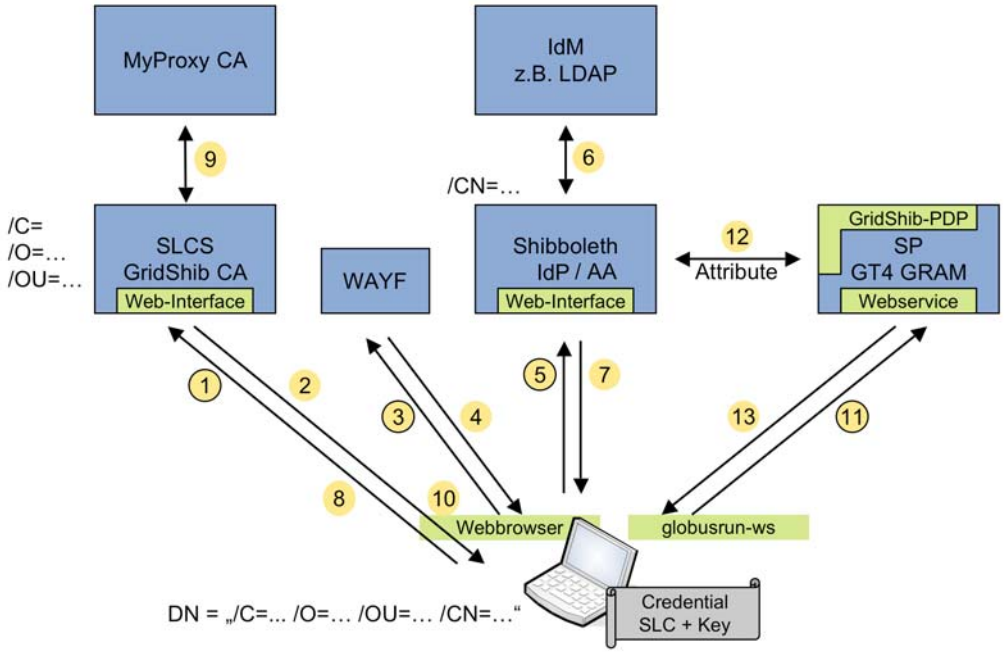


Figure 2: GridShib workflow using SLCS [GridShib]

- GridShib CA: This component is a Shibboleth Service Provider (SP) used to issue a *short lived certificate* (SLC), which the user uses instead of proxy certificates derived from a long-lived user certificate. A service issuing SLCs is called a *short-lived certificate service* (SLCS). The user needs to authenticate to the SLCS, e.g. by username/password or a user certificate imported into the web browser. If the SLCS is realized in form of a Shibboleth SP, authentication is handled by Shibboleth mechanisms described in this document.
- GridShib SAML Tools: These tools can be used to request SAML assertions from a SAML Attribute Authority (such as an IdP) and optionally bind them to a X.509 proxy certificate. Using these tools it will be possible to push attributes within such a proxy certificate to the Grid resources. This solves the IdP discovery problem and eliminates the need to install the GridShib for Shibboleth software on the IdPs.

In Figure 2 the complete process of acquiring a credential and submitting a Globus job using attribute pull is illustrated. The steps with a black circle require user interaction whereas the ones without are executed automatically. The steps are in detail:

1. The user accesses the web front end of the GridShib CA in order to request a SLC.
2. The SLC forwards the user to the federation's WAYF.
3. The user selects his home organization's IdP from a list of all available IdPs of the federation.
4. The WAYF forwards the user to the chosen IdP.

5. The user enters his username/password combination or authenticates himself by other means.
6. The IdP compares the user's credential with its *Identity Management* (IdM) system, e.g. an LDAP server. The user's attributes are also retrieved from this database.
7. The IdP issues a cookie to the user's web browser and forwards the user back to the SP he originally tried to access.
8. By accessing the cookie issued by the IdP and information embedded into the URL the IdP forwarded the client to, the Shibboleth SP protecting the GridShib CA can evaluate the user's authentication information. On the user's computer a Java WebStart application is launched which creates a key pair and sends a certificate request to the GridShib CA.
9. If the GridShib CA uses MyProxy as backend for signing certificate requests, it forwards the user's request to the MyProxy service. Alternatively the GridShib CA can use the OpenSSL libraries to sign certificate requests.
10. The user receives a SLC that, together with its associated private key, can now be used like a normal proxy credential known from the standard *Grid Security Infrastructure* (GSI) used by both gLite and the Globus Toolkit. The Java WebStart application stores the SLC and the private key in /tmp/x509_u<UID> as grid-proxy-init does with proxy credentials.
11. The user issues a Grid job using globusrun-ws just like he would do with normal proxy credentials.
12. Optional: If attribute pull is used and the SP knows which IdP to ask it is possible to pull further attributes from the user's IdP, e.g. VO membership information.
13. The result of the job submission is returned to the user.

GridShib introduces several features to Grid computing, depending on which components are used. (e.g., the Shibboleth PDP for Globus can be used independently from the GridShib CA and vice versa).

5.2 Evaluation

Using Shibboleth attributes for making authorization decisions on Grid resources will only be useful if complete user coverage in the grid-mapfile can be avoided. At the moment this is not possible because the mapping of the user to a local account is done by a static one-to-one mapping using the user's DN. Other possible advantages, such as enqueueing "premium users" to express queues, are at this point not available. The Globus Toolkit developers are in the process of implementing pool accounts similar to those used in gLite, but despite the current lack of software support many resource owners dislike the idea of pool accounts because of accounting and tracking issues caused by the lack of separation of users.

The advantage of the GridShib CA is that users do not need long running user certificates issued by a CA any more. If certificate handling and key hygiene on the user's end is considered a critical issue, the use of a GridShib CA is a viable solution. As the user still needs to authenticate to his home organization's IdP by some means it is still required that the user has some sort of credential, currently this means a username/password combination. Another open issue is the creation of the DN of the created SLC: First of all the creation needs to be consistent, i.e. a user must always receive SLCs with the same DN. As the DN is composed of one or more entries, such as Country (C), *Organization* (O), *Organizational Unit* (OU) and *Common Name* (CN), it has to be agreed upon a scheme how to create these entries. The CN will most probably be derived from an attribute issued by the user's IdP, e.g. the `eduPersonPrincipalName` attribute taken from the eduPerson [EduPerson] scheme, which is a common standard for Shibboleth attributes. At the moment, the GridShib CA statically generates the O and OU entries. It would be better to handle them in a more flexible way, e.g. by using an `eduPersonScopedAffiliation` attribute or the user's home organization identified by his IdP for the OU.

Although GridShib is mainly focused on the Globus Toolkit, it can easily be integrated with gLite's VOMS, as it is possible to embed a VOMS attribute certificate into a proxy certificate derived from an SLC issued by a GridShib CA by using the existing *voms-proxy-init* command, i.e. no new

software needs to be developed or adapted to allow this. If the Grid resources know the GridShib CA certificate as a trusted CA, all SLCs issued by it will be accepted and the existing DN- or attribute-based authorization process will be used. A possible integration with UNICORE will be evaluated by IVOM.

From an IVOM point of view, GridShib can be used on the one hand to leverage AA-Infrastructures currently being implemented by DFN. Additionally DFN is currently implementing a GridShib CA which is planned to be accredited by the EUGridPMA as soon as possible. As DFN is already active as a Grid CA and is evaluating authoritative policies [EGP] for SLCSs currently being worked out by the American TAGPMA [Gen05], it is advisable to use this infrastructure in a D-Grid AA-Infrastructure.

The GridShib for Globus package is needed if Shibboleth attributes, either pushed or pulled, are to be used for authorization decisions on Grid resources. When, in the future, more fine grained authorization decisions will be possible by evaluating attributes other than the user's DN and the need for fully-fledged grid-mapfiles will cease, a PDP evaluating these attributes is needed. In case of Shibboleth attributes this PDP will be GridShib for Globus.

GridShib for Shibboleth is from a current point of view not as valuable: The GridShib for Shibboleth package is needed only when Grid resources pull attributes directly from an IdP. As there is no easy solution for the IdP discovery problem, in the future attribute push will be the method of choice, thus rendering GridShib for Shibboleth superfluous. Furthermore, as IdPs in generic AAIs, like the DFN-AAI, will be operated by Grid-independent organizations it might not always be possible to add this functionality to the IdPs.

6 PERMIS

PERMIS [PERMIS] is a policy based authorization system (PMI - Privilege Management Infrastructure) being developed at the University of Kent (UK), which uses credentials such as X.509 attribute certificates stored in an LDAP directory to hold roles/attributes. Given a username, a resource and an action, it says whether the user is granted or denied access based on the policy for the resource. A core component of PERMIS is to provide a Policy Decision Point (PDP) functionality, another core component is the credential validation service (CVS) which is similar to a Policy Information Point (PIP).

Although PERMIS can be used in a variety of modes, the focus in this IVOM evaluation lies on its use together with Globus Toolkit and Shibboleth / GridShib. Thus the text first describes authorization in Globus Toolkit and in GridShib, where authentication and authorization is based on the Internet2-software Shibboleth, which is based on the OASIS standard SAML. Where appropriate the text refers to parallel sections within this document.

6.1 Background: Authorization in Globus Toolkit

In this section we will introduce the GridShib project from an *authorization* point of view, preparing for an understanding of the PERMIS and GridShibPERMIS projects. Before we describe the authorization part of GridShib, the reader is referred to the sections describing authorization in the Globus Toolkit itself (following hereafter) and in Shibboleth (SAML and Shibboleth).

6.1.1 Authorization in Globus Toolkit

Authorization is the process of deciding whether an entity, e.g. a user, is entitled to perform a certain action on a certain target. In the Globus Toolkit framework, in its newest version 4, a powerful and flexible authorization framework has been established [FA05, SW06]. Within this framework, an incoming authorization request will be processed by a chain of modules called *interceptors* (see also Figure 3). There are two basic types of interceptors:

- A Policy Information Point (PIP) queries suitable sources of information for attributes related to the initial request. Attributes can be a user's affiliation, some entitlement statement or just information about the user's request under consideration. The source can be external information, such as a directory, or information the client supplies with its request, such as SAML attribute assertions or X.509 Attribute Certificates. The attribute information will be

extracted from the medium that stored or transported it and normalized into a technology-neutral format ready for further processing. There can be multiple PIPs each adding information to the attribute store relating to the request.

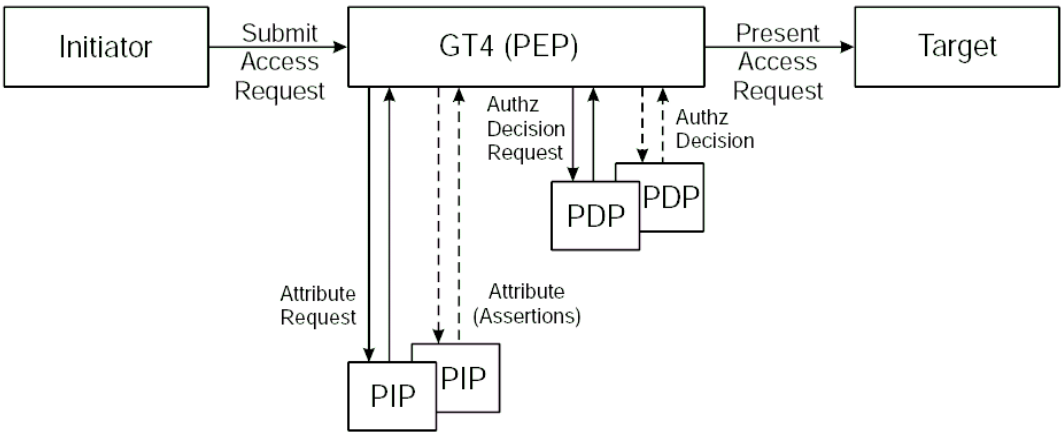


Figure 3: GT4 Authorization Architecture [CNO06]

- A Policy Decision Point (PDP) uses the supplied attributes to make a *Deny/Permit* decision for the request. There can be multiple PDPs each processing its set of attributes and each returning its decision. As of GT version 4.0.x, the answers of all PDPs are chained by the authorization engine using *AND* logic (*deny overrides)*, which means that if a `Deny` decision is returned by any one of the PDPs in the chain, the request is denied.

- The Policy Enforcement Point (PEP) is actually not an interceptor but can be seen as the GT Engine itself. It is responsible for enforcing the final decision of the authorization engine. Only if that returned *Permit*, the requested action will be performed on the requested target and the result returned to the requester.

6.1.2 Authorization with GridShib

The four components of GridShib (GridShib for Globus, for Shibboleth, CA, and SAML tools, cf. chapter Globus Toolkit and Shibboleth: GridShib) can be used mainly independently of each other. For example, the CA will be very useful for authentication of new grid users not possessing grid certificates. A complete scenario of the CA usage can be found in chapter Globus Toolkit and Shibboleth: GridShib.

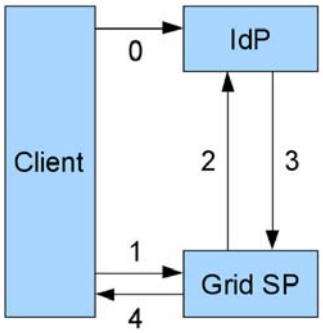


Figure 4: Classic GridShib pull profile

Here we will be more concerned with the combination of the GridShib for Shibboleth and GridShib for GT plugins from an authorization point of view. It is assumed that the user is in possession of a personal X509 certificate. Then GridShib leverages the existing *authentication* mechanisms in GT and provides attribute-based *authorization* by using the Shibboleth mechanism. A possible

workflow (Classic GridShib profile, see [Scav06c]) using attribute pull can be found in Figure 4.

0. Beforehand, a mapping from RFC2253 DNs to local principal names must have been established. This is done once at the IdP, either via one or more name-mapping-files (similar to GT's grid-map-file) or via a relational Database supporting the JDBC interface. For the latter, GridShib supplies a Certificate Registry servlet where users can register and unregister their X.509 certificates which eventually get stored in the database.

1. The Grid Client requests a service at the Grid SP. At the same time, it will present an X.509 (proxy) certificate and provide a pointer to its preferred IdP (IdP discovery problem). The latter is currently hard-wired in the Grid SP.

2. The Grid SP authenticates the Client and extracts the DN from the proxy certificate. The Grid SP queries the Attribute Authority (AA) at the IdP using the DN as a SAML name identifier.

3. At the Attribute Authority of the IdP, the requester will be authenticated by mapping the DN to a local principal name. Then the AA returns an attribute assertion to the Grid SP.

4. Finally, the Grid SP parses the assertion and performs the requested service, returning a response to the Grid Client.

Complementing the above workflow, there are several possibilities to integrate the GridShib Certificate Authority (back-ended by Myproxy) which will issue short-lived certificates (SLCs) for new Grid users into the authorization chain (cf. [Scav06c]):

- Myproxy-first with Attribute Pull: here the Online CA back-ended by MyProxy will insert a SAML authN assertion into a short-lived, reusable end-entity certificate, before the workflow follows the setup in Figure 4.

- IdP-first Attribute Pull: Here the client will first authenticate at their home IdP which will issue a SAML authN assertion. MyProxy then consumes the authN assertion from the IdP, inserts a SLC and produces another SAML AuthN assertion.

- IdP-first Attribute Push: Here MyProxy consumes both authN and authZ assertions issued by the IdP and produces both assertions including the SLC. In contrast to Attribute Pull, the Grid SP will not ask for attributes at the IdP but expects them in the SAML assertion it receives.

- Browser profiles, such as IdP-first Attribute Pull (where the normal Shibboleth profile is used), but extendend: the SP protects a web version of MyProxy, and control then proceeds to the Grid SP which pulls attributes from the IdP and returns the result via the Shibboleth SP

To summarize, GridShib can be used in various ways to account for several possibilities of authorization in the Grid context using Shibboleth. However, there are still some problems. One is that GridShib is not enough: usually, federation IdPs do not store grid-related attributes needed for authorization. IdP proxies such as myVocs can fill this gap. Another example is attribute aggregation, which is possible in the new GT AuthZ framework by parallelizing several PIPs, but is presently not implemented in GridShib.

As outlined in [CNO06], the GridShib PDP has a number of shortcomings which PERMIS promises to remedy. The GridShib PDP makes authorization decisions by comparing the received attributes to an access control list (ACL) that contains allowed attributes. This leads to users being granted access if they have any single attribute in the ACL. Combinations of attributes (e.g. member of University X and project Y) or dynamically changing conditions (e.g. time of day, amount of consumed resources) can not be accounted for. Likewise, parameters of the user's request are not recognized, such as the operation, the requested target or the job priority. Furthermore, GridShib is not able to verify whether the IdP issued the correct set of attributes that it was trusted to issue, i.e. there is no possibility to write something like a policy stating who is entitled to issue which attributes to whom.

6.2 PERMIS

6.2.1 Overview

PERMIS [PERMIS] (PrivilEge and Role Management Infrastructure Standards Validation) is a solution for policy-based *authorization*; it thus only complements existing *authentication* solutions. The following overview is in large parts quoting [PERMIS] without indicating this separately.

PERMIS is an infrastructure that provides the necessary facilities for users to manage privileges and authorization policies and for applications to make authorization decisions.

- The Attribute Certificate Manager is provided for **privilege management** and the Bulk Loader for managers to allocate privilege to users. The generated privilege information is stored in X.509 Attribute Certificate format. PERMIS also provides the Delegation Issuing Service, which allows users to delegate (a subset of) their privileges to other users in their domain, according to the site's delegation policy.
- The Policy Editor is provided for **policy management** to allow administrators to construct authorization policies for their applications and delegation policies for their Delegation Issuing Service. The policies are created in XML format, and may then be optionally protected by encapsulating them in an X.509 policy attribute certificate, digitally signed by the administrator.
- For **authorization decision making,** PERMIS provides a modular policy decision point (PDP) and a credential validation service (CVS).
    1. The **credential validation service** is used to validate whether the allocation of privileges is valid or not. (The need for this is due to the fact that privileges may be managed in a distributed manner, thus potentially anybody can allocate any privileges to anyone else, but only some of these allocations will be recognised by the PERMIS CVS as being valid). The CVS is a core component that will be integrated with applications, and it returns the set of valid attributes for a user, ready for the PDP to make an authorization decision.
    2. The **policy decision point** renders an authorization decision for a user's access request, normally in the form of *grant* or *deny*. The PDP is a core component that will be integrated with applications, and it is responsible for making the authorization decisions when applications need to verify whether a requested operation is authorized or not. The application is responsible for enforcing the decisions returned from the PDP.

PERMIS is based on the following concepts and technologies:

1. **Role Based Access Control (RBAC).** RBAC allows grouping all users into roles (or attributes), each role/attribute is associated with a collection of privileges. A user's membership of a role will allow the user to exercise the privileges associated with the role. Roles in PERMIS can be organized into hierarchies with superior roles inheriting the privileges of subordinate ones. Below we outline which parts of the RBAC standard [RBAC04] are supported by PERMIS.
2. **Policy based Management**. Authorization criteria are specified as a collection of rules, and these rules are stored as a policy. The policy is then used by the PERMIS PDP when it renders authorization decisions and by the PERMIS CVS when it returns the valid sets of user attributes. In this way, PERMIS is not hard coded with the authorization rules. Administrators can change the policy for an application, which in turn will change PERMIS's authorization decision results. Changing policies will not require any change of the applications' implementation or any recompiling of the application's code.

Other related technologies are:

3. **XACML.** (e**X**tensible **A**ccess **C**ontrol **M**arkup **L**anguage) - this XML schema (an OASIS standard) allows for representation and processing of authorization policies. The XACML interface has been introduced recently into the ensemble of supported technologies in PERMIS. This way the PERMIS CVS can interact either with the PERMIS PDP or another XACML-compliant PDP, e.g. SUNs XACML PDP.
4. **LDAP.** LDAP is used by PERMIS as a network accessible repository for storing policies and credentials. LDAP support is optional in modular PERMIS, as the system can use

other repositories such as local file stores. A new WebDAV repository is currently being tested by the University of Kent that will store X.509 certificates (PKCs, and ACs) in Apache web servers, accessible via the HTTP protocol. PERMIS also allows programmers to extend the capabilities of PERMIS to access other repositories such as databases, web pages, etc.

5. **X.509 Attribute Certificates**. X.509 Attribute Certificates were compulsory in early versions of PERMIS, to provide trust and tamper-proof resistance to policies and credentials. In the current version of PERMIS, X.509 attribute certificates are no longer compulsory, as other formats are supported, including plain XML policies and SAML attribute assertions. PERMIS also allows programmers to extend the capabilities of PERMIS to access other formats for credentials and policies.

The underlying architecture of PERMIS is a distributed architecture. Normally the following principals/entities will be involved:

1. **System Administrators (**called **Sources of Authority** in PERMIS). System administrators are principals responsible for composing the rules for the decision making and credential validation services. These rules are kept as policies. The rules for decision making specify the association of privileges to roles/attributes, saying what privileges have been assigned to every role/attribute in the system. The rules for credential validation specify the way that PERMIS recognizes valid Attribute Administrators and valid credentials that they have issued, as credentials may be issued by many parties that are not trusted by the system administrators in the current domain.
2. **Attribute Administrators (or Attribute Authorities)**. Attribute administrators (AAs) issue attributes to users. These attributes are normally used to associate users with roles. Thus with the issued attributes, PERMIS can know what roles a user has been assigned to. Attributes will be managed in the form of credentials. These are the user-role assignment rules of RBAC.
3. **Users**. Users are the principals that perform operations on the protected resources. Users can be human beings or applications.
4. **Applications**. Applications are the programs that do useful things for users and provide users with interfaces to access protected resources. Applications will need to intercept the users' request to access protected resources, and solicit authorization decisions from PERMIS. The application will then need to enforce the authorization decisions returned by PERMIS. This enforcement is normally to reject the user's request to access the resource if the authorization decision is "denied" and to allow access to the resource if the decision is "granted".
5. **Resources**. Resources are valuable computer based resources that need to be protected from being wasted, damaged or used improperly by users.

To summarize, system administrators will write authorization policies, specifying what roles have which privileges, and what kind of credentials will be recognized by PERMIS. The authorization policy will be used by PERMIS for all reasoning regarding authorization. Attribute administrators will issue credentials to users containing attributes, telling what roles the users have. When a user requests access to a protected resource, the user's credentials will be analyzed by PERMIS, and only those attributes that can be validated by the credential validation rules in the policy will be recognized as valid by PERMIS. Then PERMIS will use the association of attributes and privileges as specified in the policy to render an authorization decision for the user's request.

There is a project within the PERMIS framework called **SimplePERMIS**. SimplePERMIS represents the core of the PERMIS decision engine. It provides the core access control service (i.e. access control decision-making). It works in push mode only and the policy is stored in a plain XML file. It does not support X.509 attribute certificates, or LDAP directories. It can be considered as a lightweight PERMIS decision engine with the decoupling of credential validation and policy protection implementations.

6.2.2 Evaluation

A major advantage of the PERMIS tools is its *distributed privilege management*, i.e. the possibility to distribute the responsibilities across many domains. On the service provider side, this concerns the target SOA (Source of Authority) which creates the target access policy (TAP) and the subject

SOA which is responsible for the role assignment policy (RAP). The identity providers in turn each are responsible for allocation of attributes/roles to their users.

For a functioning PERMIS system, the following ingredients are needed:

- An LDAP server for storing the TAP and RAP policy (alternatively the policy can be stored in the file system)
- The policy formulated in XML, possibly created using the Policy Editor
- A certificate in .PKCS#12 format (that contains user and CA certificates as well as the users private key) to sign the policy
- A SOA (along with its DN) which is entitled to create the policy
- An LDAP directory to store user names and roles
- The above data are needed to configure the PERMIS engine. Additionally for each decision, the (Simple-)PERMIS engine needs the following arguments:
    - a user's DN,
    - the DN or URL of the target
    - and an action to be executed on the target.

In its first versions, PERMIS only implemented hierarchical RBAC (RBAC1) allowing inheritance of attributes. Permissions are not tied to users. However, the RBAC standard [RBAC04] allows a more fine-grained model including separation of duties (RBAC2) and other features. Moving towards the standard, RBAC2 recently has been implemented with PERMIS, details of which can be found in [CXO+07].

The most critical question for IVOM with respect to PERMIS is the question of licensing. Most PERMIS tools (except SimplePermis and SAWS, the Secure Audit Log Web Service) use the Stiftung SIC IAIK JCE library for signing X.509 attribute certificates. This library is licensed for educational and research use and evaluation only. Production and commercial use of the software is not covered by these terms (see `http://jcewww.iaik.at/sales/licences/`). There have been efforts to replace the IAIK library with Bouncy Castle as part of integrating PERMIS with OMII-UK. As of May 2007, development of this replacement has been completed and it will have to be re-evaluated at a later stage once all tests have been finished. . Additionally, the PERMIS binary downloads contain a similar license which requires to *„use the software solely for the purposes of academic research and teaching*“. However, most of the PERMIS source code has been released to the public recently and is now available under the name of OpenPERMIS with a BSD-like license. While this has eased the licensing problems, it has at the same time complicated the installation of PERMIS as the downloads solely contain the bare source code – without required libraries, without *ant* build files, without installation documentation etc.; the issue with the IAIK JCE library still remains as well.

6.3 GridShibPERMIS

6.3.1 Overview

In what follows we introduce the integration of GridShib with PERMIS. GridShibPERMIS was a third-year student research project carried out by Andrey Novikov and supervised by David Chadwick (the inventor of PERMIS) and Alexander Otenko, funded by UK JISC. Basically, the Globus Toolkit authorization chain was extended by a custom PDP (the GridShibPERMIS Context Handler) which interfaces to the PERMIS engine. Figure 5 can be consulted to gain an overview of the GridShibPermis architecture, which we describe in turn, cf. [CNO06].

Like in the classic GridShib profile, the user first authenticates using her long- or short-lived credential. Authentication could be realized by the GridShib SAML authentication PIP or some other means, but this will not be covered here. The according DN of the authenticated user is then extracted by the GridShib SAML Attribute PIP. The PIP uses the DN to request attributes at the

Attribute Authority of the Shibboleth Identity Provider, which will leverage the GridShib IdP name-mapping plug-in to resolve the DN into a local principal name. The Attribute Authority will then query the IdP's directory, e.g. an LDAP server, for this principal's attributes. In the case of PERMIS, this must include any information about this user's roles. As roles can be stored as attributes, this is compatible with directory services such as LDAP. The role information will then be returned in a SAML attribute assertion as in the standard GridShib case to the GridShib PIP.

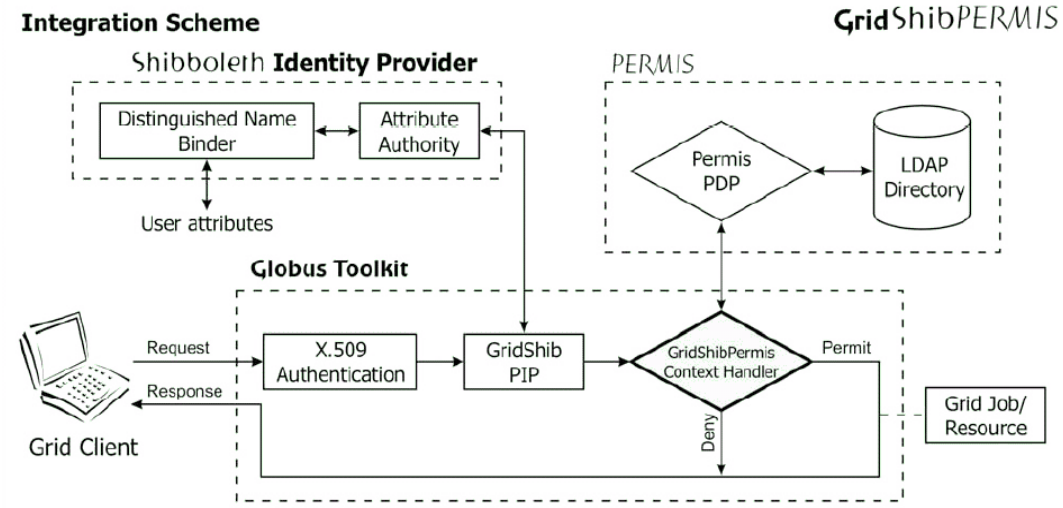


Figure 5: GridShibPERMIS integration scheme [CNO06]

The GridShib PIP will receive the SAML assertions. It parses them and converts the included attribute information into a plain Java object, which will be used further on. Next this information will be fed into the GridShibPermis Context Handler, which is the heart of the GridShibPermis project. It functions as a PDP in the Globus Toolkit authorization framework, extracting the attributes returned by the GridShib PIP. They will be converted into the internal Java format recognized by the PERMIS API. According to the PERMIS policy used, Permis' Credential Validation Service (CVS) will check whether the IdP is trusted and entitled to issue the attributes it has returned. In order to do this, either an LDAP directory or a local file will be queried for the policy to be applied. Next the PERMIS PDP proper is called and handed over the valid attributes together with the user's requested action and target. It is here where an access control decision is made according to the PERMIS policy in effect. The decision is returned to the GridShibPermis Context Handler which in turn hands it over to the Globus Toolkit PEP which is responsible for enforcing the decision.

An attempt is made to support the distributed management of attributes by allowing *scoped* domains as known from Shibboleth. When the PERMIS CVS is passed scoped attributes, the scope domains take the place of the Source of Authority (Subject SOA, i.e. the signer of the AC). The PERMIS role allocation policy specifies the scope domains as SOAs in place of attribute certificate issuers. If an origin site does not issue scoped attributes, the name of the origin site will be inserted as scope domain for all attributes, but only if its name is configured into the GridShib PIP's SAML metadata. This solution is very constraining as it allows only for a single origin site.

6.3.2 Evaluation

An advantage of GridShibPERMIS over plain PERMIS is that the licensing problems with the IAIK JCE library are less of an issue. This is because GridShib does not use X.509 attribute certificates but encodes its attributes in SAML. Consequently, the library that would handle ACs is not needed. However, this library is also used to sign the policies and store them in ACs. This means that GridShibPERMIS would have to work with unsigned policies. As described in [Nov06], omission of the IAIK library also resolves a compatibility problem the IAIK library had with the Claymore SSL Toolkit used by Globus and GridShib.

There are a number of issues in which GridShibPERMIS does not differ from plain GridShib. First, we have the IdP discovery problem. This is currently solved, as in GridShib, by hard-coding the preferred IdP into the GridShib PIP. Once the Globus team solves this it will be solved for

GridShibPERMIS as well. Second, the Shibboleth IdP has to maintain information about user's roles in its attributes which has to be in synchronization with the attributes/roles expected by the SP/PDP. Third, according to [CNO06], there is no possibility yet to use pseudonymous access in GridShibPERMIS as both the user's DN and attributes will be revealed to the grid application. Fourth, according to [Nov06], there is no possibility to merge attributes from multiple AAs and use those for the PERMIS decisions. However again, this is a GridShib issue as PERMIS itself can merge automatically from multiple IdPs by users' LDAP DNs.

By the time of writing this report (March 2007), it turned out that the current release of GridShib (0.5.1) had changed several names of Java classes compared to GridShib 0.4.0 which the original GridShibPERMIS project was based on. This meant that an error would be generated whenever these methods were called, resulting in failure of the whole GridShibPERMIS system. We thank the whole PERMIS team and especially David Chadwick and George Inman for tracking and fixing this issue. This part of the present report was written in close contact with them.

Another question in the IVOM context would be whether there is, besides Globus' GridShib, also an integration of PERMIS with gLite or UNICORE available. As for gLite, it might be included in a currently new project to integrate PERMIS and VOMS with GT4, OMII-UK and GT2. It will be using VOMS as an attribute authority and PERMIS as a decision engine. (See [VPMan]). UNICORE integration is not planned.

PERMIS also has the SAAM module which allows Shibboleth-enabled authorization for Apache-controlled websites. The DyVOSE project [DyVOSE] developed an extension to PERMIS for dynamic delegation of authority and pushing out ACs for use in VOs. An approach similar to PERMIS but for Grid portals is currently being developed as part of the OMII-SP project [OMII-SP].

7 MAMS' VO-related Work

7.1 MAMS Project Overview

MAMS was initiated in 2003 as a three-year project to integrate „multiple solutions to managing authentication, authorization and identities, together with common services for digital rights, search services and metadata management", locally within organizations and inter-institutional. The project's objective was to provide „an essential middleware" component to increase the efficiency and effectiveness of Australia's higher education research infrastructure" [MAMS].

To achieve this, MAMS aims to produce integrated solutions to identity, authentication, and authorization management, together with common services for digital rights, search services and metadata management, for users in mutual trusted organizations to share protected resources and services.

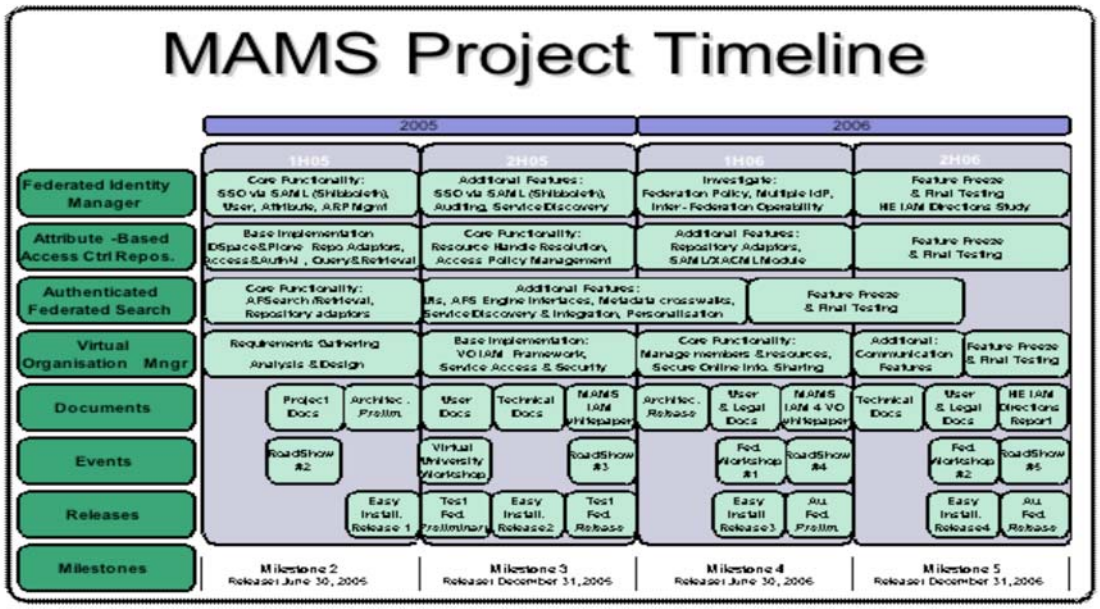


Figure 6: MAMS project timeline [VuDa05]

MAMS was a middleware project that developed a technical infrastructure for institutions to share content and resources in a trusted environment. It had no specific objectives in the field of grid computing (see Figure 6). Though, during the course of the project, work and concepts included grid aspects. MAMS conducted research and development work in the following fields:

- Federated identity and access management
- Federated search
- Customization of applications for use with Shibboleth (GridSphere, DSpace, Fedora etc.)
- Open Access
- Access Control with XACML

MAMS decided early to rely on Shibboleth [Shib] as authentication and authorization infrastructure. Therefore the project initiated the Australian AAI federation and provided the required resources. It contributed ShARPE, the Shibboleth Attribute Release Policy Editor [ShARPE], and Autograph, which enables personal control on Shibboleth attribute release [Autograph], to the Shibboleth community. These components will be included in future Shibboleth releases.

7.2 MAMS' definition of VO

MAMS has a broad understanding of the term virtual organization (VO). It follows the VO definition of T. Dimitrakos, D. Golby and P. Kearney [DGK04]:

> *„ A Virtual Organisation is understood as a temporary or permanent coalition of geographically dispersed individuals, groups, organisational units or entire organisations that pool resources, capabilities and information to achieve common objectives. Virtual Organisations can provide services and thus participate as a single entity in the formation of further Virtual Organisations.“*

In general, scientific projects, work groups and other inter-institutional collaborations are seen as virtual organizations. MAMS´ goal is to provide a appropriate VO infrastructure to allow for:

- Collaboration between project members,
- Collaboration with externals,
- Dissemination of research results,

- Authentication and authorization based on Shibboleth.

The VO is seen at the core of a general collaboration platform for eScience, which should provide for the same tools and technologies real organizations use: calendars, forums, wikis, grids, repositories, to name a few.

Product	Full Name	Type	Release	Availability	Documentation
ShibGS	Shibbolized GridSphere	GridSphere Plugin	Beta	Download at MAMS website	Short installation guide
IAMSuite	Identity Access and Management Suite	General collaboration toolkit, based on a VO management component	Prototype	Online Test	Architecture, Functionality

Table 1: State of VO-related work at MAMS (as of February 2007)

7.3 ShibGS: Shibboleth-enabled GridSphere

To develop a VO management system was a major work item in MAMS from start on. As there was no Shibboleth-based system available they started to build the required features into GridSphere [GS]. GridSphere was selected for being Open Source Software with access to the source code, for its JSR168 compliance, its integrated user management and its existing range of portlets.

MAMS released this product as ShibGS [ShibGS], a GridSphere plugin that enabled Shibboleth-based authentication. ShibGS is not a VO management system. It is included in this study as it is the only openly available product reviewed here.

7.4 IAMSuite

In February 2006, the MAMS team started to develop an ambitious toolkit for the Australian eReseach middleware infrastructure: The Identity and Access Management (IAM) Suite [IAMSuite]. On the base of secure portal-based VO infrastructure it shall integrate the Shibboleth and PKI AAI frameworks to support access to common Internet services, such as portals, repositories, Wikis etc., as well as access to Grid services, including Grid storage facilities (e. g. Storage Resource Broker) and high performance computing. Additionally, the IAM Suite serves as a general IT infrastructure toolkit for the management of projects, groups and workspaces, providing for easy set-up and access to collaboration tools like a CMS, calendar, Wiki, forum, mailing lists. This enhances research effectiveness for projects, especially those funded for short durations, by saving time to get the project IT infrastructure going. The conceptual model underlying the IAMSuite system is called the Trust Virtual Organization (TVO).

With the IAM Suite MAMS proposes to organize existing services in three layers (see Figure 7):

Layer 1: The **Federation Services** layer contains standard services for the Australian federation, including a WAYF service. Additionally, a Shibboleth-protected MyProxy server shall provide for conversion of a user's SAML assertion into a short-lived proxy certificate, giving the user access to grid facilities based on Globus Toolkit [GTK]. MAMS also developed a federation gateway called VO-WAYF to support cross-federation authentication and authorization (not shown in Figure 7: IAM Suite architecture).

Layer 2: Institutional Identity and Service Providers constitute the **Institutions** layer. These are the federation members.

Layer 3: **The eResearch Project or Virtual Organization** layer contains the general IT infrastructure components required by an eResearch project. At the core is the IAMSuite. It is the main access point for project work:

- On a user's request being the switchboard to collect a SAML assertion from the institutional IdP, adding an own VO-specific SAML assertion and presenting it to a SP in the VO domain.
- It is the Identity Provider for all the SPs in the VO domain.
- It will provide a user with a proxy certificate from the Federation MyProxy server to access Grid/HPC services.
- It contains a Group Manager to allow for VO administration.

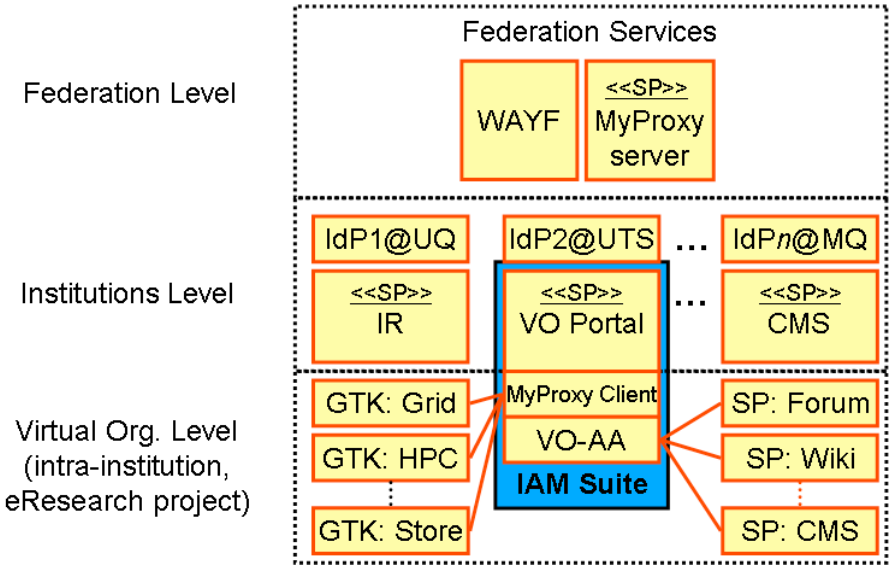


Figure 7: IAM Suite architecture

The IAM Suite shall enable eResearch projects to get an enriched SSO environment right out of the box. To allow for adaption to specific needs it will have a modular architecture [Vul06].

7.4.1 IAMSuite VO: The Trust Virtual Organization Model

The IAMSuite VO is a VO management system based on the Trust Virtual Organization (TVO) conceptual model [IAMSuite]. The IAMSuite system provides a work „environment for geographically dispersed individuals, groups and organizational units to construct and maintain their temporary or permanent trust relationships and share disperse protected resources and services with SSO to achieve common goals“ [IAMSuite]. IAMSuite main concepts are:

- Work space for VO members
- Consistent sharing space for a collection of distributed resources and services
- Trust bridge between IdPs and SPs across federations

The IAMSuite system incorporates both Shibboleth communications entities; it is a service provider as well as a identity provider. The SP part comprises the GUI for VO administration and user self-service. It also collects the SAML assertion from the user's home IdP during the user's login.

The outline of a typical IAMSuite workflow is shown in Figure 8 [VBD05]:

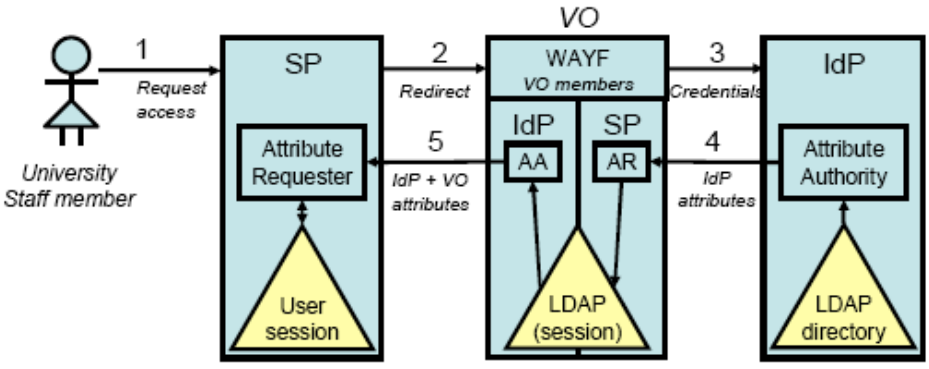


Figure 8: MAMS VO authentication and authorization model [VBD05]

1. The user tries to access a Grid service provider (SP).
2. As the SP doesn't know the user, she is redirected to the VO's Where Are You From

(WAYF) service, which consists of a list of VO member institutions. Additionally, the WAYF creates a cookie and stores the desired SP's address.

3. She selects her institution's Identity Provider, is redirected to the IdP, and provides the necessary login credentials. Typically, this could be a username and password, but could also be a PKI certificate provided by the institution's helpdesk and signed with the institution's key.
4. The IdP validates the login credential, and the Attribute Authority uses the user's Attribute Release Policy to determine which attributes should be sent to the VO, which is done accordingly using the SAML artifact method or the SAML post method. Additionally, the SAMLAuthenticationMethod variable is set to Software PKI (in case of PKI login) or Basic (in case of password login).
5. The VO verifies the received SAML assertion (do we trust this institute, is the signature valid) and stores the received attributes in its own directory (this could be session based, or over a longer period according to the policies of the VO). Typically, those attributes should contain the personal attributes of the user, like full name, email address. Now, additional VO-specific attributes can be added to the authorization flow.
6. The user is redirected again to the actual Service Provider (the SP's address was stored by the WAYF in step 2) she wants to visit, accompanied by another SAML assertion generated by the VO's IdP.

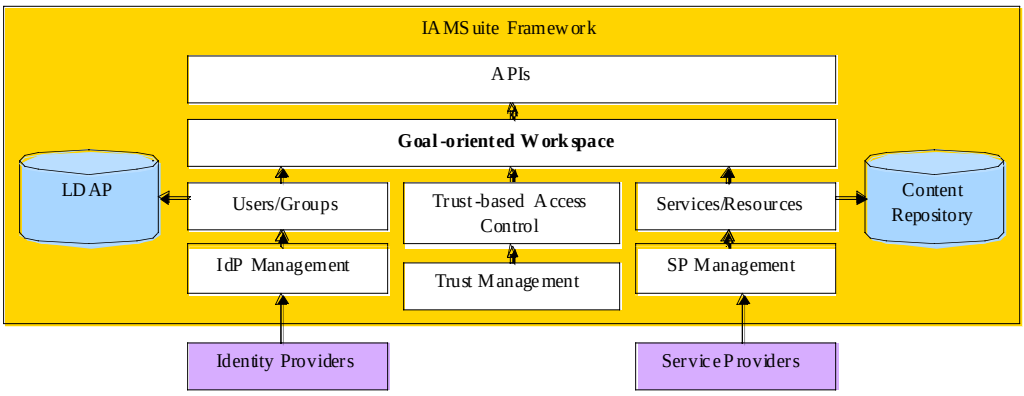


Figure 9: IAMSuite implementation of the TVO model [IAMSuite]

The IAMSuite is built on software from the Open Source projects Shibboleth (IdP and SP), MyProxy, OpenLDAP and Fedora, the OAI-compliant repository system. Figure 9 shows an architecture overview. The user and group management facilitates the life cycle (creation, modification, and deletion) of users or groups. It is based on the MAMS OpenIdP, which is a Shibboleth IdP connected to a LDAP server as user database. The Content Repository is based on the Shibboleth SP and Fedora to protect resources and services. Virtual Rooms serves as shared workspaces, in which users may collaboratively create, modify, and utilize protected resources and services.

MAMS plans to provide an API to support developers in constructing IAMSuite-enabled applications in the fields of collaborative learning and e-Science or eResearch, respectively.

7.5 A Review of MAMS' VO-related Work

A review of MAMS' VO-related work can only in the case of ShibGS be based on practical product experience. The other products are not available yet. A prototype of IAMSuite with restricted access is available on-line. Therefore we review the latter based on the prototype and available documents.

7.5.1 ShibGS

The integration of Shibboleth authentication and authorization with the widely used Grid portal

software GridSphere was an intermediate step for MAMS. It facilitates the use of a JSR168 compliant, rich Open Source portal software in a identity federation context. MAMS realized the integration by extracting common user attributes from the user's SAML assertion and storing them in GridSphere's user database.

ShibGS is not a true plugin as it needs some modifications in GridSphere itself to work properly. This is due to the intervention into GridSphere's user management. When a user logs in to ShibGS, he will be authenticated at his home IdP, then attributes, like the user id, surname, givename and mail address, will be requested from the user's home IdP and stored in the portal's user database. At each subsequent login, the user will be authenticated at his home IdP and, if needed, his attributes will be refreshed in GridSphere's user database.

A similar approach [WAJKS06] was chosen by the DyVOSE (Dynamic Virtual Organization in e-Science Education) project [DyVOSE], led by the National e-Science Centre, United Kingdom.

ShibGS is supported by MAMS as a part of IAMSuite. Still, we see further need for Shibboleth-based authentication and authorization being integrated in GridSphere and supported by the GridSphere development team.

7.5.2 IAMSuite

It is MAMS' objective to build an integrated middleware component to support out-of-the-box proliferation of eResearch infrastructure. IAMSuite is the VO management component for this planned solution. For the time being, the direction MAMS has taken with IAMSuite is the only feasible, as is reflected by UAB's myVocs [myVocs], which in general uses the same technique to realize VO management in Shibboleth.

IAMSuite works as an IdP proxy between SPs outside IAMSuite and the federation IdPs. When a user logs in to IAMSuite, he will first be authenticated at his home IdP. During the authorization process the VO management extracts the user attributes from the home IdP's assertion and stores it in a LDAP server as long as the session is active. When a user tries to access a SP outside IAMSuite during session lifetime, the user attributes will be released together with the VO attributes. The attributes are included in a single assertion, signed by the IAMSuite. This may lead to severe trust issues as we pointed out above (see Trust Issues above). MAMS is considering solving this problem in a future release. This trust problem does not affect the SPs that are closely integrated into the portal-based IAMSuite. Access to these services and resources is through the login to IAMSuite.

IAMSuite – and myVocs – address the IdP Proxy problem (see Multiple VO Memberships above) by implementing the IdP proxy as a bridge between a federation of SPs – in myVocs they are called VO SPs – and the federation IdPs. This integration of SPs with the IdP proxy provides for the management of VOs at a single place.

The IdP Proxy approach with its aggregation of SPs closely tied to the VO management is well suited for community Grids, where a proven trust fabric between providers and users is to be expected. However, with the further expansion of the Grid there is the need for further development to allow for the support of large-scale international projects with tens of thousands of members.

With IAMSuite, the MAMS project has integrated the VO management into the GridSphere portal software. This moves the VO management close to the services and resources that may be included in the portal as portlets or otherwise. As access to these services and resources is not protected by separate Shibboleth SPs, there has to be a trustful relationship between those and the IAMSuite portal.

7.5.3 Conclusion

At the current stage of development, MAMS' IAMSuite can not be recommended for use in D-Grid. As of early 2007 it is available only as online demo. Also, trust issues regarding the combined release of IdP and VO attributes are not addressed in the current version of IAMSuite.

The further development should be followed closely as the integration of VO management, repository systems, like Fedora or Storage Resource Broker, and Grid middleware is an interesting approach in the Australian e-Research context.

In must be emphasized that we continue to recommend other MAMS products, namely ShARPE, Autograph and ShibGS. Shibboleth-based authentication and authorization should be closely integrated into GridSphere and supported by the GridSphere development team.

8 VOMS and VOMRS

8.1 Short Description

gLite's *Virtual Organization Membership Service* (VOMS) [VOMS] and the *Virtual Organization Membershib Registration Service* (VOMRS) [VOMRS] developed by the VOX Project at Fermilab are both systems for managing members of VOs. The VOMRS user interface can be seen in Figure 10: VOMRS screenshot. The two systems have though a different focus and can be deployed individually or together as complementing systems.

The features are in detail:

Features VOMS and VOMRS have:

- A database backend for storing the users and their attributes
- Storing of VO-membership attributes as well as Group, Role and Capability attributes used by gLite's LCAS mechanism for authorization decisions.
- A web front-end for both users to register themselves and for VO-Administrators to manage the VO-members.

VOMS-only features:

- Web service based API to access the VO membership data
- Issuing of attribute certificates for inclusion in proxy certificates

VOMRS-only features:

- A, compared to VOMS, more streamlined workflow for registering new users and administration of their attributes. This lessens the burden on VO administrators by, e.g., requiring new users to validate their e-mail addresses before their membership application is presented for approval to the responsible VO administrator.
- Compulsory acceptance of an *Acceptable Usage Policy* (AUP).
- Storage of additional user information such as telephone number and additional arbitrary attribute-value pairs
- Forwarding of entered data to a complementing VOMS

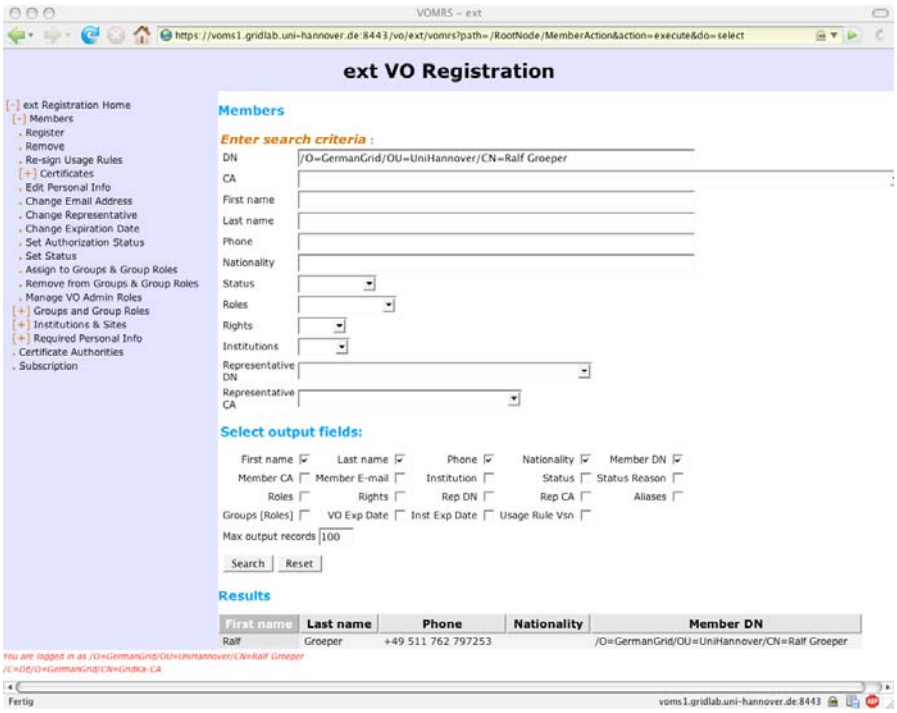


Figure 10: VOMRS screenshot

8.2 Evaluation

A standalone VOMRS without VOMS is useful if no attribute certificates need to be included in proxy certificates, e.g. in a Globus Toolkit-only environment. In this case, in order to use the stored data e.g. for creating grid-mapfiles, the information needs to be extracted by directly accessing the database as there is no API such as the one VOMS offers.

A standalone VOMS does not offer the streamlined process of user registration and can thus impose a major burden on VO administrators. Furthermore it is not possible to store additional metadata about VO members, such as a telephone number. If such features are not needed but attribute certificates based on the VO membership data have to be issued, VOMS can be used standalone.

As VOMRS is developed from the beginning with VOMS compatibility in mind, a combination of both services offers the combined features of both systems. It has to be considered that VOMRS must be used as the leading system as data migration only works in the direction from VOMRS to VOMS, not the other way round. If thus data is entered or altered directly in the VOMS, the two databases will become inconsistent.

With regard to IVOM, it is advised to use both systems in conjunction: gLite requires a VOMS to issue attribute certificates and the streamlined registration process of VOMRS will ease the burden on VO administrators of large VOs. Furthermore current developments need to be assessed as current versions of VOMS are extended to be able to store arbitrary attribute-value pairs and a more streamlined registration process is planned for future releases. If future VOMS releases incorporate these features the advantages of adding VOMRS to VOMS might become less relevant.

9 Virtual Organization Collaboration System (myVocs)

9.1 A Short Description of myVocs

9.1.1 myVocs' Objectives

myVocs' design goal was to "extend the access to emerging Internet collaboration tools and build a system environment that respects VO defined roles and attributes while preserving valuable institutional identity assertions" [myVocs].

myVocs thus manages *attributes*. It actually is a SAML-based Identity Provider proxy serving as a bridge between a federation of Shibboleth Identity Providers (IdP) and a federation of Shibboleth Service Providers (SP) (see Figure 11) for overcoming the somewhat unrealistic expectation that home organizations maintain their VO list of users. myVocs presents itself as a Shibboleth SP so that other services can rely on it to ensure that the user has been authenticated. The myVocs servers assert the attributes that the SPs in the VO need to base their authorization decisions upon.

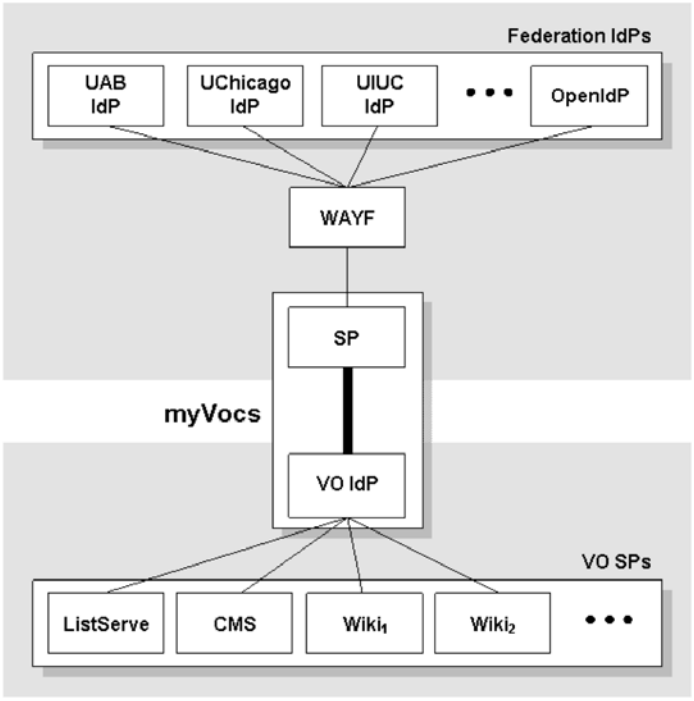


Figure 11: myVocs as bridge [Scav06]

myVocs allows several SPs (called VO SPs) to be aggregated into Virtual Organizations (VOs). myVocs considers VOs as people (more precisely: collections of attributes of people represented by lists), and the aggregated SPs as federated sets of distributed applications, the resources, accessible by this list of people. It is an important feature of myVocs that a single VO SP may serve multiple VOs and, hence, supporting overlapping VOs [Robi07].

Like the IdPs, the VO SPs may reside in arbitrary administrative domains. Using off-the-shelf, open source software components (such as Shibboleth, MySQL, and Sympa), myVocs provides the "glue" that authorizes access to a VO SP based on the membership in some specific VO. The resources are protected by VO SPs which are mutually trusted by a VO IdP. Figure 12 illustrates the orchestration of the various myVocs components during the process of authentication and authorization.

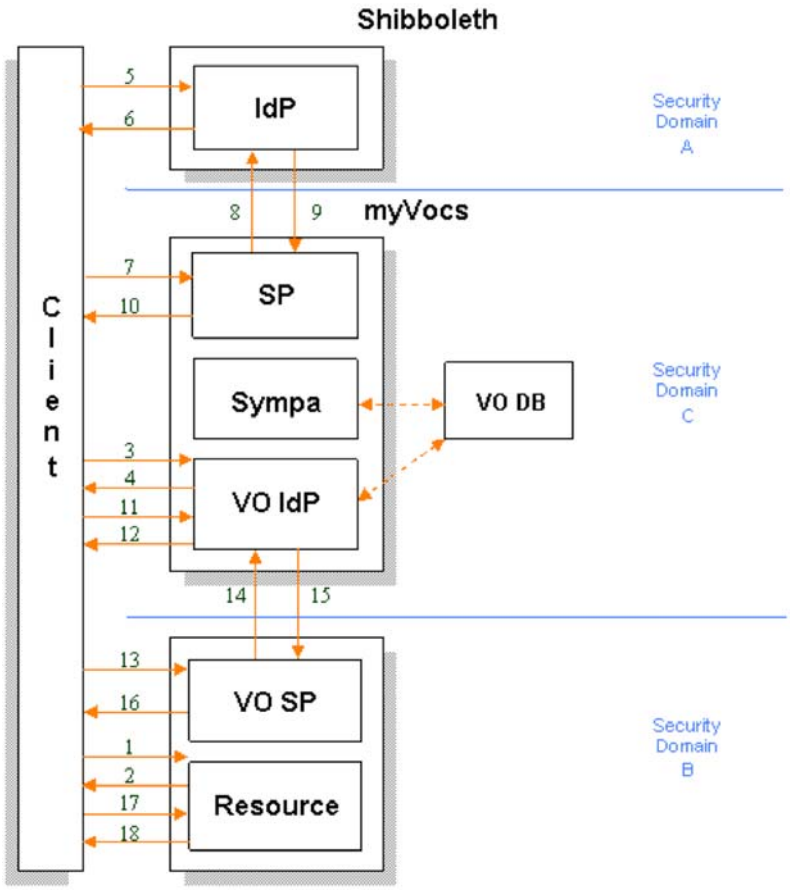


Figure 12: Typical myVocs workflow [Scav06]

The outline of a typical myVocs workflow looks as follows:

1. A browser client requests a VO web resource protected by a VO SP (e.g. a wiki or a listserv). If a security context for the principal (i.e. `eduPersonPrincipal`) already exists at the VO SP, skip to step 18.
2. The client is redirected to the VO IdP (which is protected by a federation SP).
3. The client makes a Shibboleth AuthnRequest (new in Shibboleth 1.3) to the VO IdP. If a security context for the principal already exists at the VO IdP, skip to step 12.
4. The client is redirected to the federation IdP (ignoring a possible interaction with the federation WAYF).
5. The client makes a second Shibboleth AuthnRequest to the SSO service at the federation IdP. If a security context for the principal does not exist at the federation IdP, the IdP identifies the principal (details omitted).
6. The IdP updates security context for this principal, issues an authentication assertion, and returns an authentication response to the client.
7. The client submits the authentication response to the assertion consumer service at the federation SP. The assertion consumer service validates the authentication assertion in the response and passes control to the attribute requester.
8. The attribute requester queries the attribute authority at the federation IdP.
9. The attribute authority returns an attribute response to the attribute requester.
10. The federation SP updates its security context for this principal and redirects the client to the VO IdP.
11. The client makes a Shibboleth AuthnRequest to the VO IdP, the same AuthnRequest made at step 3.
12. The VO IdP filters the attributes from the header of the request (by virtue of the attribute exchange in steps 8 and 9), persists these attributes to the VO database (if necessary), and returns an authentication response to the client.

13. The client submits the authentication response to the assertion consumer service at the VO SP. The assertion consumer service validates the authentication assertion in the response and passes control to the attribute requester.
14. The attribute requester queries the attribute authority at the VO IdP.
15. The attribute authority returns an attribute response to the attribute requester. Both federation attributes (persisted at step 12) and VO attributes are included in the response.
16. The VO SP updates its security context for this principal and redirects the client to the VO resource.
17. The client requests the VO resource, the same request issued at step 1.
18. The resource filters the attributes from the header of the request (by virtue of the attribute exchange in steps 14 and 15), makes an access control decision, and returns the resource to the client.

9.1.2 The myVocs box

The goal for *myVocs box* [myVocs] was to provide a framework for building federated system environments. In the case of distributed environments one of the strongest methods of binding systems together is by having a consistent definition of identity across the system. Essentially, the identity boundary defines the system boundary. Identities are the attributes that define users: username, email, and group memberships. The goal of the myVocs box is thus to support the definition and distribution of these identities that can then be available to all applications in the system environment, i.e. shared across Web-based and Grid-based applications.

myVocs box essentially includes a VO attribute store (implemented as a relational data base) that holds user groups and roles for the federated system environment. This attribute store is driven/controlled by the creation of mailing lists in the Sympa mailing list manager[2]. Although any attribute management system could be used, Sympa was selected by the developers simply as an easy tool to use that also provides a useful mailing list for VOs. The attributes can be distributed to applications via Shibboleth or GridShib for Web-based or Grid-based applications respectively.

myVocs box is packaged as a virtual machine. The current version is 0.1 and it is thus very preliminary. It has been released on December 3rd 2006. It may be downloaded as tar-ball from http://myvocs-box.myvocs.org/downloads/myvocs-box-v0.1.tar.gz . The technologies in this release include:

- A complete Shibboleth 1.3 identity system (IdP and SP)
- Simple collaboration group setup and management via Sympa
- Flexible resource integration powered by YubNub[3]
- Dynamically allocated Drupal[4], PHPwiki[5], and WEBInsta FM[6] collaboration tools
- Globus integration powered by Gridshib CA and GridShib for Shibboleth
- A short circuit identity provider for stand-alone operation.

9.2 Installation of myVocs box in a Nutshell

myVocs box comes as a ready to run virtual machine. If a VMware environment has already been established, myVocs box should be placed in the same directory as the other VMs and should be started from there. If such an environment is not yet available, the VMPlayer tool has to be installed first (available from http://www.vmware.com/products/player/). Using the VMPlayer myVocs box can be executed on the VMware provided NAT network. After having started it, the web browser needs to point to the box. This is accomplished by associating the host name *myvocs-box* with the running box. To do this the IP address, the myVocs box is running on (printed on the VM console after the machine started up), needs to be inserted into /etc/hosts. After that the browser needs to point at http://myVocs-box.

For testing myVocs box, the browser needs to trust myVocs box. This is achieved by clicking on the *Trust Me* link above the command prompt. Further usage and installation hints are given when using the *Tools* link. For “playing” with the tool: the root password is “root”, users can be created

2 http://www.sympa.org
3 http://yubnub.org/
4 http://drupal.org/
5 http://phpwiki.sourceforge.net/
6 http://www.webinsta.com/fm.php

using the standard Linux commands after having “ssh”-ed to the box.

We have setup myVocs box on an Intel Pentium 4 CPU, 3.00 GHz machine with 1 GB memory under Scientific Linux with kernel version 2.4.21-40.EL using Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-18 mod_jk/1.2.5 mod_fastcgi/2.4.2 mod_ssl/2.8.22 OpenSSL/0.9.7e.

9.3 Using myVocs Box

Based on its background as collaboration system myVocs looks at VOs as collections of attributes associated to members of mailing lists and at VO resources as Web applications. Consequently myVocs provides commands for creating, maintaining, monitoring, and managing such lists (i.e. VOs).

The commands which control myvocs-box are Web “commands” issued to the “command line” in Figure 13. These commands are conceptually similar to traditional unix commands but are simply expanded to HTTP GET or POST commands.

Figure 13: myVocs entry screen

As an example, *newvo* expands to the URL:

http://myvocs-box/sympa/create_list_request?listname=%25shttp://myvocs-box/sympa/create_list_request?listname=%25shttp://myvocs-box/sympa/create_list_request?listname=%s, where %s is a placeholder for the argument passed to the newvo command. This interface to myVocs is implemented using YubNub.

9.3.1 List of Commands

As mentioned above, myVocs offers both a conceptual command line interface and a Web-based navigation. The commands are:

- about: Some info about myVocs box
- addnewuser: Add a new user to the designated VO
- admin: Access the VO overview and member administration page
- archive: Access the discussion archives of a project
- book: Create a new book page in the project CMS
- bookmark: Post a bookmark to del.icio.us
- browse: Browse the existing virtual organizations
- cms: Open the Content Management System (CMS) for the specified VO
- create: Create new command
- del: View the latest del.icio.us entries for the given tag
- files: Browse the files available to the members of the VO
- g: google
- ge: Voted most popular commands
- gim: Search google images

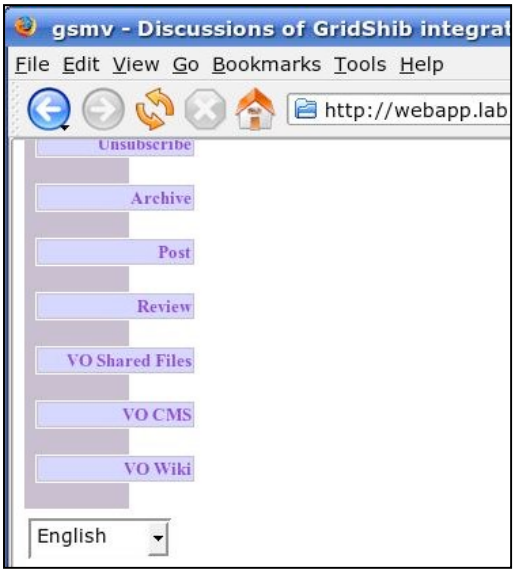
- gridlogin: Initialize grid credential for Globus using the myVocs-box instance of the GridShib CA
- join: join a VO of the given name
- ls: List all commands
- most_used_commands: The most used commands
- newblog: Bring up compose window for a new blog entry for the specified VO
- newvo: Create a new virtual organization.
- post: Send a message to the members of the VO
- project: Find out about the myVocs-box project
- tec: Search Technocrati
- viewfile: view a file in the file manager for a VO
- wiki: Open the wiki for the specified VO
- wp: Search the Wikipedia encyclopedia with the specified keyword

Due to myVocs' understanding of VOs, creating a VO is simply creating a mailing list, assigning a subject to it and categorizing it. Members join the VO by subscribing to the list. Note that this understanding of VOs implies a separation of VO membership and VO ownership: the owner of a VO does not necessarily need to be a member of the list; she explicitly has to subscribe to the list.

9.3.2 Web-Based Interface

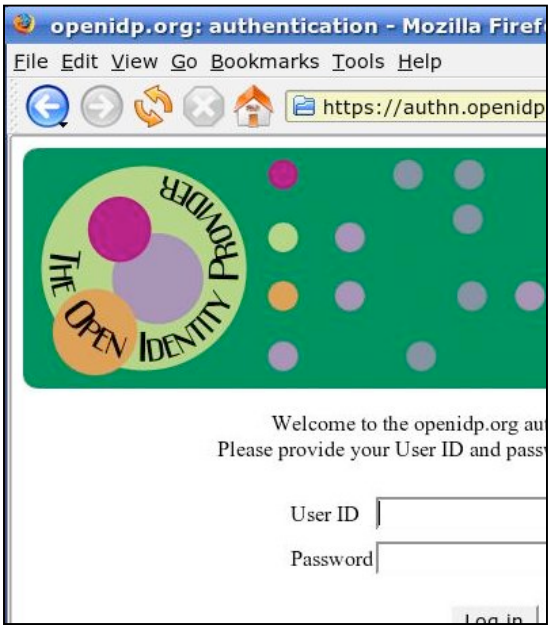
When using myVocs box through the Web interface the user will be navigating through several simple pages. Here are some examples:

9.3.2.1 Selecting a VO resource



9.3.2.2 Selecting an Identity Provider

9.3.2.3 Validating Identity



9.4 Who Is Using or Considering myVocs?

Currently the system is being used by the foundation of campus grid computing environment at the University of Alabama at Birmingham, USA. The Internet2 group is exploring VO management architectures to support user community and is considering myVocs for this. The Safari project of UK Data Archive[7] is exploring myVocs for access authentication to resources. The collaboration with GridShib is ongoing. The Teragrid project is considering myVocs for user registration[8].

9.5 A Review of myVocs

Focusing on attribute management within collaboration environments, myVocs allows for VOs based on Shibboleth identities. Users register via Shibboleth and can be added to myVocs-

[7] http://safari.data-archive.ac.uk/

[8] http://www-fp.mcs.anl.gov/tgmeeting/AAA-Agenda.htm and https://spaces.internet2.edu/display/GS/TeraGrid

maintained groups (in the form of mailing lists). myVocs acts as a Shibboleth proxy with a dual IdP and SP role to add group information to a user's normal Shibboleth information. myVocs thus follows a similar approach as the Australian MAMS (see chapter MAMS' VO-related Work).

As per today (i.e. as per this version of myVocs) there are some issues with myVocs:

- It is not clear which attributes need to be captured and persisted at step 12 in Figure 12. myVocs requires the federation IdP to release the attribute "`eduPersonPrincipalName`", a globally unique identifier for the principal. This global identifier is permanently bound to a local identifier in the VO database. It is this binding that permits myVocs to determine the VO attributes associated with the user. The local identifier is determined as a result of a one-time registration step. At the time of registration, the user's global identifier is bound to a local identifier in the VO database. As this is a kind of static approach, a more flexible registration process is required longterm.

- In the architecture diagram (Figure 11), the myVocs SP relies on an ordinary WAYF for IdP discovery. In order to gain more flexibility (and some knowledge of a user's history), myVocs proposes an enhanced IdP discovery process for myVocs based on the SAML 2.0 IdP Discovery Profile, which allows SPs to more easily discover the user's preferred IdP (see Figure 14).

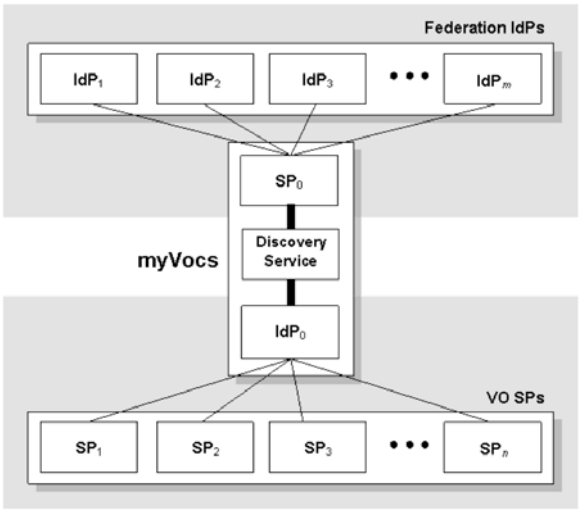


Figure 14: Enhanced IdP discovery [Scav06a]

The goal of this implementation is to display a simplified "confirmation page" to the end user. Instead of a complete list of federation IdPs, the user is presented with a short list of recently visited IdPs, in reverse chronological order. For the majority of users, however, this list will be of length one.

- Assuming most users have an account at at most one IdP, an obvious simplification is to automatically redirect to the only IdP on the list (without confirmation). The issue arises, however, if the user wishes to use a different IdP at a later time.

- Currently, myVocs requires a user to be pre-registered. An unregistered user will not be able to use the myVocs system. Consequently, the request at step 11 of the myVocs protocol flow (see Figure 12) will fail.

- myVocs' understanding of a VO is simply a collection of attributes of persons represented by lists. Consequently, VO membership is list membership. List membership is defined by respective `eduPersonPrincipalName` (ePPN) and `Mail` attributes. myVocs expects the provisioning of ePPN by the federation IdPs. ePPN is globally unique and needs to be mapped onto a local identifier in the VO data base which is being installed during the initial registration. From myVocs' perspective is this VO description sufficient. Whether

or not this will also be sufficient for D-Grid will be the topic of subsequent IVOM work packages.

- As both the ePPN and the Mail attributes indicate the VO membership these are passed along in an assertion to an SP. As per today, other attributes, while stored during registration, are not passed along. Simply passing these additional attributes along to the SPs comes down to extending the SQL query and ARP that is used by the Shibboleth infrastructure within myVocs. Providing administrator or end-user control interfaces for those attributes, however, is not trivial. A tool like Sharpe from MAMS could help to solve this issue.

- myVocs doesn't currently support sub-VOs directly. As mentioned before, the VO concept as defined in myVocs considers groups with a static set of sub-groups defined as "owner", "editor/admin", and "member/subscriber" that currently act as authorization sub-groups (aka roles). Because these groups/roles are directly inherited from the Sympa subsystem they can communicate using the email addresses "voname-owner@domain" and "voname-editor@domain". The only difference from these addresses and the "voname@domain" address is that they are simple email expansions and don't have the bounce/subscribe/archive options of the parent "voname@domain" list. To overcome the VO/sub-VO configuration issue either the Sympa umbrella lists or the Grouper effort my help, but both haven't been tested for supporting VO/sub-VO configurations in myVocs.

9.6 myVocs in Context

9.6.1 myVocs and Globus

myVocs box includes both the GridShib CA and the GridShib for Shibboleth components from the GridShib project [GeRo06]. The GridShib CA will issue short term certificates to users that can be used to access Globus resources. The GridShib for Shibboleth interface makes it possible to feed collaboration group membership information to Globus resources located in any domain by using GridShib through myVocs. To achieve this, identity federation would actually require a translation of the user's affiliation name to the name by which they are known in the virtual organization, which would, in turn, be translated into a X.509 DN [WeSi06].

myVocs handles the first translation through the use of its internal databases when a user registers. The second translation, from myVocs into the Grid domain, is processed by a certificate registry service, which is a Gridshib service on the myVocs Shibboleth server. It allows an X.509 user to assert an X.509 certification through the standard Shibboleth authentication process. The Shibboleth server then binds those two names for the purpose of future identification. Since the Shibboleth server can recognize that a particular X.509 identity is bound to a specific local identity, it can feed back the appropriate attributes [WeSi06].

In myVocs, a Virtual Organization (VO) manages any number of Web resources. With GridShib installed, the same VO may include any number of Grid resources protected by Grid SPs (see Figure 15)

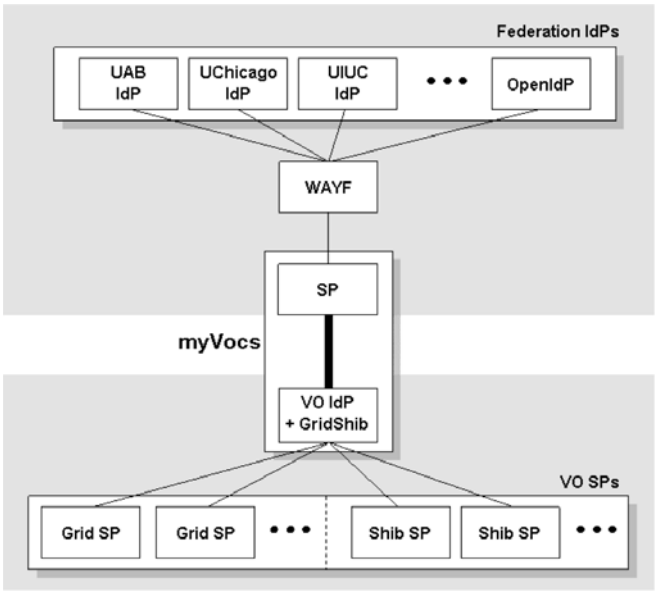


Figure 15: myVocs and Gridshib [Scav06b]

9.6.2 myVocs and VOMS/VOMRS

The Virtual Organization Membership Service (VOMS) stores attributes and X.509 DNs and issues short term certificates based on this information. The Virtual Organization Membership Registration Service (VOMRS), on the other hand, provides the workflows necessary for registering users, organizing them by institution, assigning groups, granting roles, and coordinating approval of these by institutional, VO and/or resource representatives. The resulting attributes are stored in a VOMRS database, which is used to fill the VO-specific VOMS databases. All attributes are separately queryable.

Like VOMS, myVocs places the control of VO attributes closer to the VO resources that require them. Rather than depending on federation IdPs, myVocs allows VOs to manage their own attributes. Thus, VO resources can leverage standard federation attributes (such as `eduPersonScopedAffiliation` and `eduPersonPrincipalName`) as well as any VO attributes that are maintained locally. Integrating VOMS and myVocs requires managing VOMS roles as myVocs attributes.

There are two opportunities of integrating such VO-based attributes with Shibboleth IdP concepts in Grids [Sill06]: one way is to keep all external Grid pieces the same as in the present VO-based workflows and introduce extra assertions via SAML (or, in principle, XACML) to augment the PDP decision. Mapping to accounts may then optionally be done as before, or by GridShib. The other way it to replace the external X.509 dependancy by one of many Shibboleth-based IdP replacement schemes like GridShib + myVocs and obtain the VO attributes from VOMRS by a Web Services call or by importing the VOMRS attribute database directly (as depicted in Figure 16).

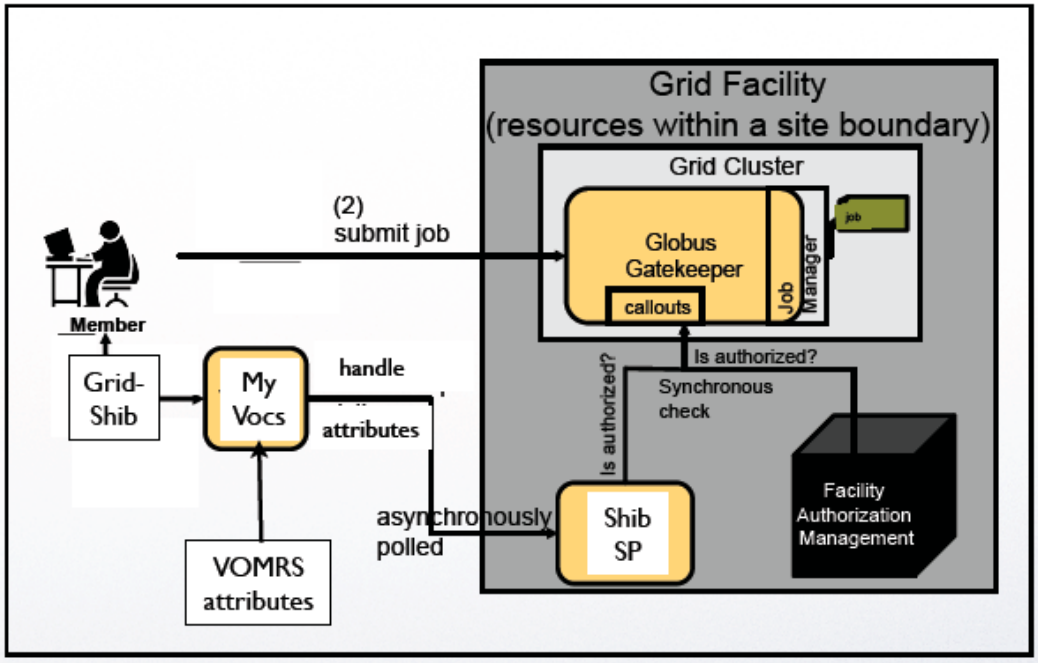


Figure 16: myVocs and VOMRS integration [Sill06]

9.6.3 myVocs and MAMS

myVocs and MAMS are pursuing the same goals. However, in order to overcome the difficulties of supporting attributes beyond e.g. eduPerson's `eduPersonAffiliation` and `eduPersonScopedAffiliation` or `eduPersonPrincipalName`, they use different approaches: a VO oriented approach and a user oriented approach. myVocs is – just like VOMS – a representative of the first category, whereas MAMS belongs to the latter class.

As a VO oriented tool, myVocs places the management of VO attributes close to the VO resources that require these attributes. Instead of depending on federated IdPs with a standardized central set of attributes, myVocs allows VOs to manage their own attributes (as depicted in Figure 11 and Figure 12). Thus, VO resources can be protected not only by standard federation attributes (such as the eduPerson ones) but also by any other VO-specific attribute defined and maintained locally. By contrast, in MAMS the user decides herself on the attributes (by using Autograph.).

Eventually, MAMS and myVocs will converge, as was presented by Neil Witheridge, Project Manager MAMS (http://cd-docdb.fnal.gov/0019/001976/001/Fermilab20061208_NW_MAMS.ppt).

10 Comparison

The integration of Shibboleth with Grid middleware and the VO management concept is an ongoing process. Even the most advanced product in this field, GridShib, is available only as a beta version. myVocs and IAMSuite were the first products that started the integration of VO management in Shibboleth. myVocs is currently available as a 0.1 release and IAMSuite as a online demo. These products are in an early stage of development. We do not expect mature public releases of these products before 2008. That said, we have to make clear that the evaluated products already contain sufficient functionality to start working with them.

Two products developed by the Internet2 community were previously seen as candidate VO management systems: Grouper [Grouper] and Signet [Signet]. So far they lack a Shibboleth Attribute Authority component and it is not clear if development of these systems will proceed into that direction.

The Grid-relatedness of the products evaluated in this work package differs to a considerable degree. IAMSuite and myVocs are true Shibboleth-enabled VO management systems. GridShib is a modular product to integrate Shibboleth with Globus Toolkit Grid middleware. PERMIS is a

RBAC authorization infrastructure that can interface to Shibboleth and GT. VOMS and VOMRS are classical certificate-based VO management systems, developed for the Grid.

As the evaluated products address diverse problem domains they cannot be compared with each other. Therefore we define the following categories:

- Shibboleth/Grid integration: GridShib (GT) and Interoperability Shibboleth-gLite in EGEE II.
- Shibboleth-based VO management: myVocs and IAMSuite.
- Certificate-based VO management: VOMS and VOMRS.
- Policy-based authorization framework: PERMIS.

In the Shibboleth/Grid integration category GridShib stands out as the first and most advanced project. GridShib has proven its ability to execute by continuously improving the product and extending the scope. The Interoperability Shibboleth-gLite project started later and is in a rather early phase.

In the area of Shibboleth-based VO management myVocs is available as an early release version with basic functionality. GridShib and myVocs cooperate and as a result myVocs box was recently released as an integrated package containing GridShib for Shibboleth, GridShib CA and myVocs itself. IAMSuite is available only as an online prototype. Both systems, as well as the Shibboleth-gLite integration implemented by SWITCH, extract attributes from the IdP assertion and assert them as their own. We believe this to be not a good practice as we pointed out in chapter 3.2. Due to the current architectural constraints of Shibboleth they function as IdP Proxies (see chapter 3). This may possibly change in the future, when Shibboleth gets enhanced to support multiple attribute authorities.

Certificate-based VO management systems are available for some time and have achieved a considerable level of maturity. Support for VOMS attribute certificates, originally developed within the gLite context, is currently implemented for the GT. It is possible to combine the GridShib PDP and the VOMS PDP on GT4 resources[9], thus allowing for the use of attributes taken from both, a SAML assertion and an attribute certificate, for authorization decisions. The main advantage of VOMS-based VO management in D-Grid is the currently available or planned support of VOMS attribute certificates in all three middlewares: gLite has complete and stable support for attribute certificates, a PDP for the Globus Toolkit is available as part of GridShib for GT 0.5.1 and support for UNICORE is under way in an IVOM work package.

As for policy-based authorization frameworks, PERMIS is the only candidate in this report. There have been several projects integrating PERMIS with e.g. Shibboleth, GT3 and GT4, GridShib, Apache Web Server, etc. However, gLite and UNICORE are by now not among the supported technologies.

In most cases the VO management systems are developed for a specific Grid middleware. Consequently, the use of a specific middleware determines the choice of the Shibboleth/Grid integration solution and subsequent tools like the VO management system. Therefore the listed products form distinctive ecosystems grouped around the respective Grid middleware:

- Globus Toolkit 4 ecosystem: GridShib, myVocs, IAMSuite, VOMRS, VOMS (porting in work), PERMIS.
- gLite ecosystem: Shibboleth/gLite integration in EGEE II, VOMS, VOMRS.
- UNICORE ecosystem: Shibboleth and VOMS integration is an ongoing part of IVOM.

Due to GridShib and myVocs the Globus Toolkit ecosystem is currently the most advanced in the field of Shibboleth and Grid integration. Work on the Shibboleth/gLite integration was recently started by SWITCH.

[9] http://gridshib.globus.org/docs/gridshib-gt-0.5.1/admin-index.html#VOMS

Features	IAMSuite	myVocs	VOMS	VOMRS
A. Profile				
1. Primary Grid ecosystem	Globus Toolkit	Globus Toolkit GridShib	gLite Globus Toolkit[10]	gLite Globus Toolkit
2. AAI base	Shibboleth	Shibboleth	X.509 PKI	X.509 PKI
3. Release state (April 2007)	Web prototype	Beta	Stable	Stable
4. Software base	GridSphere	Sympa	VOMS	VOMRS
5. Maintainer	MAMS	UAB	INFN[11]	USCMS/Fermilab
B. Interoperability with Grid Middleware				
1. Compatibility with GT 2	-	-	-	n/a
2. Compatibility with GT 4	Planned	Integration with GridShib	(X)[10]	n/a
3. Compatibility with gLite	-	-	X	n/a
4. Compatibility with Unicore	-	-	X (IVOM)	n/a
5. Compatibility with GridShib	-	X	X	n/a
C. Scalability				
1. Maximum number of VOs	unlimited	unlimited	1 DB per VO, # of DBS limited by system resources	1 DB per VO, # of DBS limited by system resources
2. Maximum number of users	unlimited	unlimited	No specific limit	No specific limit
D. VO Management				
1. Multiple memberships (user can be member of more than one VO)	X	X	X	X
2. Different rights/roles per VO user (static)	X	X	X	X
3. Different rights/roles per VO user (flexible)	X	(X)[12]	X	X
E. VO Administration				
1. Easy VO setup	X	X	X	X
2. VO admin can invite/enlist users	X	X	X	X
3. VO admin can de-list users	X	X	X	X
F. Interoperability with Short Lived Credential Services				
1. Supports own SLCS (one SLCS per VO server)	X	X[13]	-	-
2. Supports central SLCS (e.g. EUGridPMA accredited DFN-SLCS)	Planned	X[13]	X	X
G. Handling of IdP Assertions				
1. Attributes imported from IdP assertion to identify user	eduPersonTargetedID, mail	eduPersonPrincipalName, mail	n/a	n/a
2. Additional attributes imported from IdP assertion	eduPersonEntitlement, uid, cn, givenName, sn, o	all attributes released to myVocs by an IdP	n/a	n/a
3. Embedding of original IdP assertion in VO assertion	-	-	n/a	n/a
H1. Issuing of VO Attributes: SAML Assertions				
1. Issuing of VO assertions	X	X	n/a	n/a
2. Attributes used to represent VO membership	eduPersonEntitlement, employeeType	ePPN, mail from eduPerson; a custom "group" attribute in the format *role@vo*	n/a	n/a
3. Additional attributes included in VO assertion	eduPersonScopedAffiliation, mail, givenName, sn, o	-	n/a	n/a
H2. Issuing of VO Attributes: Attribute Certificates				
1. Support of Attribute Certificates	n/a	n/a	X	n/a

[10] VOMS-PDP for GT4 is available as „technical preview" and will be part of GT4.2
[11] https://twiki.cnaf.infn.it/cgi-bin/twiki/view/VOMS/WebHome
[12] Not via a *UI*.
[13] SLCSs are independent of the core of myVocs.

Features	IAMSuite	myVocs	VOMS	VOMRS
2. Representation of VO membership	n/a	n/a	FQAN (Fully Qualified Attribute Name: VO, Group, Role, Capability)	n/a
3. Additional attributes included	n/a	n/a	currently VOMS 1.7: arbitrary Attribute-Value Pairs	n/a
I. Comments				
			While VOMS does not rely on Shibboleth techniques, it is possible to combine it with Shibboleth-enabled environments for VO-Management, such as GridShib or VASH	VOMRS can be deployed together with VOMS. Then see VOMS for capabilities for points G and H.

Table 2: Comparison of VO management systems

X implies that the feature is supported. A dash means "not supported". "n/a" means "not applicable".

In Table 2 the Shibboleth- and PKI-based VO management systems are compared against a set of features, which were identified in the evaluation process.

11 Conclusion

It is our objective in this work package to lay the ground work for the process of selecting prospective Grid and Shibboleth integration technologies and VO management products in D-Grid. The decision on the choice of technologies and products will be based on the final set of requirements to be determined in IVOM work package 2.

A considerable set of products is emerging in the field of integration of X.509-based Grid environments with Shibboleth/SAML. We have evaluated a selection of these technologies as well as Shibboleth-based and PKI-based VO management systems to assess their suitability as integration and management tools in Grids. The projects under evaluation were the gLite-Shibboleth integration, GridShib, IAMSuite, myVocs, PERMIS, VOMS and VOMRS.

GridShib had a head start in the field of Grid and Shibboleth integration and maintains a lead over the peer projects. It currently offers the broadest set of solutions and is the best starting point for Grid and Shibboleth integration, given it takes place in the Globus ecosystem.

While myVocs is restricted regarding both the attribute handling and the user/admin support, it is however flexible enough to pave the way for a VO management in Grids utilizing Shibboleth-based federations of IdPs and Grid Service Providers. Bridging collections of IdPs and SPs is a requirement when transparently managing VOs in non-trivial configurations. myVocs supports this objective. Combined with functionalities from other projects myVocs would be a first-choice candidate to further explore in IVOM. IAMSuite, developed by MAMS, is not yet available as a software product and can therefore not be recommended here.

VOMS is a mature and stable VO-Management system developed as part of the gLite middleware. It is used in production environments, especially in the HEP communities, for several years and such is the de-facto standard in PKI-based VO management. Furthermore it is being actively enhanced with new features such as support for arbitrary attribute-value-pairs, which is an essential feature for flexible VO management. The importance of VOMS is also reflected by the ongoing integration of attribute certificates in additional Grid middlewares such as the Globus Toolkit 4. Due to its support in different Grid middlewares and its maturity it is advised to consider VOMS in work package 3 and assess its suitability for IVOM based on the results of work package 2. It has to be considered that VOMS itself does not offer the integration of Shibboleth-based campus attributes, which is an essential goal of IVOM. Means would have to be found to combine VOMS with Shibboleth, e.g. by using GridShib or an approach similar to the VASH service by SWITCH.

VOMRS offers only a subset of the features of VOMS, but implements them in a more streamlined way, thereby lessening the burden imposed on VO administrators. However, VOMRS can be used as a front-end of a VOMS-server, offering the complete functionality of VOMS and the

streamlined workflows of VOMRS. As the VOMS developers currently plan to overhaul the VOMS web interface, VOMRS might not be necessary any more when this VOMS version will be released.

PERMIS is a system for policy-based authorization, which has already a longer history, however, support for grid infrastructure and GridShib has been introduced rather recently. It is in active development. The system provides all components needed for establishing and maintaining an authorization infrastructure to be used in, but not limited to, grid environments.

In IVOM work package 3 we will consider all products evaluated for their suitability in regard to the community requirements, which will be the deliverable of work package 2. Based on our review we think that the following products are the best-of-breed approaches for VO management currently available:

- VOMS and VOMRS offer support for long-time Grid users with an established PKI infrastructure. If additional Shibboleth-based campus attributes are needed for authorization, means have to be found to make these attributes available to Grid resources, e.g. by using GridShib.
- GridShib used in co-operation with myVocs or VOMS offers support for Grids utilizing PKI-based authentication and Shibboleth-based authorization in the Globus Toolkit ecosystem. This approach especially supports the leveraging of the campus attributes managed by the user's home IdPs. gLite users can utilize their VO-attributes immediately if VOMS is used. Though, gLite users will have to wait for the deliverables of the gLite/Shibboleth Integration project to use their campus attributes or VO-attributes managed by myVocs.
- GridShib and myVocs offer support for Grids utilizing Shibboleth for both, authentication and authorization, in the Globus Toolkit ecosystem. The primary use cases are Portal-based Grid access and SLC-based Grid access. gLite users will have to wait for the deliverables of the gLite/Shibboleth Integration project at SWITCH.

This evaluation is a snapshot of products in a highly dynamic research environment. Readers should be aware that most of these products are in an ongoing development process.

12 Acknowledgements

We would like to thank the PERMIS developer David Chadwick, University of Kent (UK); James Dalziel, the MAMS chief investigator, Macquarie MELCOE (Australia); the PERMIS developer George Inman, University of Kent (UK); Alan Lin, developer of ShibGS and IAMSuite, Macquarie MELCOE (Australia); John-Paul Robinson from the University of Alabama at Birmingham (USA), the developer of myVocs box; Tom Scavo from the University of Illinois (USA), one of the GridShib developers; Richard Sinnott, technical director at NeSC Glasgow; and Erik Vullings, the former MAMS program manager, Macquarie MELCOE (Australia); for their valuable comments and their responsiveness.

The IVOM work is funded by the BMBF, the Federal Ministry of Education and Research (PT-IN grant FKZ 01AK810B [LRZ], 01AK810C [AWI], 01AK810D [RRZN], 01AK810E [DAASI]).

13 Contacts

This report is a team work product. The introduction and the closing chapters were written collectively by all authors. The work on the other chapters was split among the project members, the leading authors for these parts are given here as contact persons:

Chapter	Chapter Title	Contact
2	SAML and Shibboleth	Christian Grimm, Ralf Gröper

3	Shibboleth and VO Management	Siegfried Makedanz, Hans Pfeiffenberger
4	gLite and Shibboleth: Work done by SWITCH	Christian Grimm, Ralf Gröper
5	Globus Toolkit and Shibboleth: GridShib	Christian Grimm, Ralf Gröper
6	PERMIS	Peter Gietz, Martin Haase
7	MAMS' VO-related Work	Siegfried Makedanz, Hans Pfeiffenberger
8	VOMS and VOMRS	Christian Grimm, Ralf Gröper
9	Virtual Organization Collaboration System (myVocs)	Michael Schiffers

14 List of Abbreviations

AA	Attribute Authority
AAI	Authentication and Authorization Infrastructure
AC	Attribute Certificate
ACL	Access Control List
API	Application Programming Interface
AuthN	Authentication
AuthZ	Authorization
CA	Certificate Authority
CMS	Content Management System
CN	Common Name
DFN	Das Deutsche Forschungsnetz
DN	Distinguished Name
DyVOSE	Dynamic Virtual Organization in e-Science Education
ePPN	eduPersonPrincipalName
GT	Globus Toolkit
GUI	Graphical User Interface
HPC	High Performance Computing
IAM	Identity and Access Management
IdM	Identity Management
IdP	Identity Provider
IT	Information Technology
IVOM	Interoperabilität und Integration der VO-Management Technologien im D-Grid
JSR	Java Specification Request
LDAP	Lightweight Directory Access Protocol
MAMS	Meta Access Management System
O	Organization
OASIS	Organization for the Advancement of Structured Information Standards
OU	Organizational Unit
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
PKI	Public Key Infrastructure
RAP	Role Assignment Policy (PERMIS)
RBAC	Role Based Access Control
RFT	Reliable File Transfer
SAML	Secure Assertion Markup Language
ShARPE	Shibboleth Attribute Release Policy Editor
ShibGS	Shibbolized GridSphere
SLC	Short-Lived Credential

SLCS	Short-Lived Credential Service
SOA	Source of Authority (PERMIS)
SP	Service Provider
SSO	Single Sign On
TAGPMA	The Americas Grid Policy Management Authority
TAP	Target Access Policy (PERMIS)
TVO	Trust Virtual Organization
UAB	University of Alabama at Birmingham
UNICORE	Uniform Interface to Computing Resources
Vocs	Virtual Organization Collaboration System
VO	Virtual Organization
VOMRS	Virtual Organization Membership Registration Service
VOMS	Virtual Organization Membership Service
WAYF	Where Are You From
WS-GRAM	Web Service Grid Resource Allocation Manager
XACML	Extensible Access Control Markup Language

15 References

[Autograph] Autograph Website, http://federation.org.au/Autograph, last visited 14 Feb 2007.

[CNO06] Chadwick, D.W., Novikov, A., Otenko, A. "GridShib and PERMIS Integration." Campus-Wide Information Systems. Vol. 23, No. 4, 2006. pp297-308, ISSN 1065-0741. (Cf. also http://www.terena.org/events/tnc2006/core/getfile.php?file_id=753 .)

[CXO+07] David W Chadwick, Wensheng Xu, Sassa Otenko, Romain Laborde and Bassem Nasser. "Multi-Session Separation of Duties (MSoD) for RBAC". To appear in: First International Workshop on Security Technologies for Next Generation Collaborative Business Applications (SECOBAP'07), April 16-20, 2007, Istanbul, Turkey.

[DFNAAI] DFN-AAI - Authentifizierungs- und Autorisierungs-Infrastruktur im DFN, Website http://www.dfn.de/content/dienstleistungen/dfnaai/, last visited 02 Apr 2007

[DGK04] Theo Dimitrakos, David Golby and Paul Kearney: Towards a Trust and Contract Management Framework for dynamic Virtual Organisations. Proc. eChallenges 2004, Vienna 2004. http://epubs.cclrc.ac.uk/bitstream/701/E2004-305-TRUSTCOM-OVERVIEW-FINAL%5B1%5D.pdf, last visited 24 Feb 2007.

[DGV] VOMS Page at DataGrid Website, http://edg-wp2.web.cern.ch/edg-wp2/security/voms/voms.html, last visited 26 Feb 2007.

[DyVOSE] DyVOSE Website, http://labserv.nesc.gla.ac.uk/projects/dyvose, last visited 06 Jun 2007.

[EduPerson] eduPerson Object Class, Website http://www.educause.edu/eduperson/, last visited 02 Apr 2007

[EGP] EUGridPMA Guidelines and Authentication Profiles, Website http://eugridpma.org/guidelines/, last visited 02 Apr 2007

[FA05] Authorization processing for Globus Toolkit Java Web services, Tim Freeman and Rachana Ananathakrishnan, 2005-10-25, http://www-128.ibm.com/developerworks/grid/library/gr-gt4auth/, last visited March 2007

[FKT01] Ian Foster, Carl Kesselmann and Steven Tuecke: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 15(3), 2001. http://www.globus.org/alliance/publications/papers/anatomy.pdf, last visited 24 Feb 2007.

[FTLW07] P. Flury, V. Tschopp, T. Lenggenhager, C. Witzig: *Shibboleth Interoperability with Attribute Retrieval through VOMS*, https://edms.cern.ch/cedar/plsql/doc.info?document_id=807849. Version 0.95, January 2007, last visited 02 Apr 2007

[Gen05] Ed. Tony J. Genovese: Profile for Short Lived Credential Services X.509 Public Key Certification Authorities with secured infrastructure, http://www.tagpma.org/files/IGTF-AP-SLCS-20051115-1-1.pdf, last visited 02 Apr 2007

[GeRo06] Jill Gemmill and John-Paul Robinson: *myVocs and GridShib: Integrated VO Management*, Presentation at the Spring 2006 Internet2 Member Meeting in Arlington, VA (USA), April 2006, http://grid.ncsa.uiuc.edu/presentations/i2mm-myvocs-gridshib-april06.ppt.

[Globus] The Globus Allicance, Website http://www.globus.org/, last visited 02 Apr 2007

[GridShib] GridShib: A Policy Controlled Attribute Framework, Website http://gridshib.globus.org/, last visited 02 Apr 2007

[Grouper] Grouper Website, http://grouper.internet2.edu, last visited 24 Feb 2007.

[GS] GridSphere Website, http://www.gridsphere.org, last visited 25 Feb 2007.

[GTK] Globus Toolkit Website, http://www.globus.org/toolkit/, last visited 26 Feb 2007.

[Haka] Haka federation – CSC, Website http://www.csc.fi/english/institutions/haka, last visited 02 Apr 2007

[IAM] IAM Suite Website, http://www.federation.org.au/twiki/bin/view/VO/WebHome, last visited 24 Feb 2007.

[IAMSuite] IAMSuite Online Prototype: https://group-2.mams.org.au/gridsphere, last visited 06 Mar 2007.

[IVOM06] Projektantrag *Interoperabilität und Integration der VO-Management Technologien im D-Grid*, Version 1.3 vom 27. Juni 2006.

[MAMS] MAMS Project Overview Website, http://www.melcoe.mq.edu.au/projects/MAMS/, last visited 10 Feb 2007

[Mor06] RL ‚Bob' Morgan: Posting about „Multiple Scopes" in Shibboleth. Shibboleth Users Mailing List, 8 Sep 2006. https://mail.internet2.edu/wws/arc/shibboleth-users/2006-09/msg00064.html, last visited 24 Feb 2007.

[MVO05] EGEE Service Request: Multiple VOs affiliation, submitted by Johan Montagnat, 21 Feb 2005. https://savannah.cern.ch/support/index.php?func=detailitem&item_id=100558, last visited 26 Feb 2007.

[myVocs] myVocs Website, http://myvocs.org/, last visited 24 Feb 2007.

[Nov06] Andrey Novikov, GridShib and PERMIS Integration. CO620 Research Project Report 2005/06, University of Kent. http://www.cs.kent.ac.uk/pubs/ug/2006/co620-projects/gridshib/report.pdf, last visited March 2007

[OASIS] Organization for the Advancement of Structured Information Standards (OASIS), Website: http://www.oasis-open.org, last visited 24 Apr 2007

[OMII-SP] Open Middleware Infrastructure Institute UK – Security Portlets, website, http://labserv.nesc.gla.ac.uk/projects/omii-sp, last visited 06 Jun 2007

[PERMIS] PERMIS websites, http://sec.cs.kent.ac.uk/permis/index.shtml and http://www.openpermis.org/, last visited March 2007

[RBAC04] ANSI/INCITS 359-2004 (cf. http://csrc.nist.gov/rbac/, last visited March 2007)

[Robi07] John-Paul Robinson: Private Communication

[SAML] OASIS Security Services (SAML) TC, http://www.oasis-open.org/committees/security/http://www.oasis-open.org/committees/security/

[Scav06] Tom Scavo: *Introduction to myVocs*, https://spaces.internet2.edu/display/GS/MyVocs, last visited 3rd April 2007

[Scav06a] Tom Scavo: *myVocs IdP Discovery*, https://spaces.internet2.edu/display/GS/MyVocsIdPDiscovery, last visited 3rd April 2007

[Scav06b] Tom Scavo: *myVocs-GridShib Integration*, https://spaces.internet2.edu/display/GS/MyVocsGridShibIntegration, last visited 3rd April 2007

[Scav06c] GridShib - A Technical Overview, Tom Scavo, NCSA, April 2006, http://grid.ncsa.uiuc.edu/presentations/gridshib-tech-overview-apr06.ppt, last visited 3rd April 2007

[ShARPE] ShARPE Website, http://federation.org.au/ShARPE, last visited 14 Feb 2007.

[Shib] Shibboleth Website: http://shibboleth.internet2.edu/, last visited 24 Feb 2007.

[Shib2] Shibboleth 2 Roadmap, https://spaces.internet2.edu/display/SHIB/ShibTwoRoadmap, last visited 21 Feb 2007.

[ShibDS] Website Shibboleth Discovery Service, https://spaces.internet2.edu/display/SHIB/DiscoveryService, last visited 20 Feb 2007.

[ShibGS] ShibGS Website, http://mams.melcoe.mq.edu.au/wiki/display/MAMS/How+to+install+and+test+ShibGS+(Shibbolized+GridSphere)+project, last visited 30 Jan 2007

[Signet] Signet Website, http://signet.internet2.edu, last visited 24 Feb 2007.

[Sill06] Alan Sill: *VOMRS-Shibboleth Integration*, Presentation at the GGF18 Meeting, Washington D.C. (USA), September 2006, http://grid.ncsa.uiuc.edu/events/ggf18-shib-bof/VOMRS-Shibboleth%20Integration.pdf

[SW06] The Globus Toolkit Authorization Framework. Frank Siebenlist, Von Welch, Information leaflet for GlobusWorld2006, http://www.globus.org/alliance/events/gw06/gt-authz-gw06-v3.pdf, last visited March 2007.

[SWaai] SWITCHaai, Website http://www.switch.ch/aai/, last visited 02 Apr 2007.

[TCM] TrustCoM Website, http://www.eu-trustcom.com, last visited 24 Feb 2007.

[Trust05] Lutz Schubert et al.: TrustCoM Reference Architecture v1.0. Report, 2005. http://www.eu-trustcom.com/DownDocumentation.php?tipo=docu&id=208, last visited 24 Feb 2007.

[VOMRS] VOM Registration Service, Website: http://computing.fnal.gov/docs/products/vomrs/vomrs1_3/wwhelp/wwhimpl/js/html/wwhelp.htm, last visited 02 Apr 2007

[VOMS] VOMS, Website http://vdt.cs.wisc.edu/components/voms.html, last visited 02 Feb 2007

[VPMan] Integrating VOMS and PERMIS for Superior Secure Grid Management (VPMan), Website: http://sec.cs.kent.ac.uk/vpman, last visited 06 June 2007

[VBD05] Erik Vullings, Markus Buchhorn and James Dalziel: Secure Federated Access to GRID applications using SAML/XACML. APAC 2005, Gold Coast 2005. https://mams.melcoe.mq.edu.au/zope/mams/kb/all/20050630%20-%20Secure%20Federated%20Access%20to%20Grid%20Applications%20using%20SAML_XACML%20-%20Vullings-Buchhorn-Dalziel.pdf/download, last visited 24 Feb 2007.

[VuDa05] Erik Vullings and James Dalziel: Meta Access Management System – A Summary for DEST SII Proposals. June 2005. http://www.dest.gov.au/sectors/research_sector/policies_issues_reviews/key_issues/australian_research_information_infrastructure_committee/documents/mams_overview_rtf.htm, last visited 19 Feb 2007

[Vul06] Erik Vullings, Australian eResearch Infrastructure Proposal v2, June 2006, not published.

[WAJKS06] J. Watt, O. Ajayi, J. Jiang, J. Koetsier and R.O. Sinnott: A Shibboleth-Protected Privilege Management Infrastructure for e-Science Education. Glasgow 2006. http://labserv.nesc.gla.ac.uk/projects/ESP-GRID/Docs/papers/CLAG06-final.pdf, last visited 24 Feb 2007.

[Wel05] Von Welch: GridShib: Grid-Shibboleth Integration (Identity Federatin and Grids). UK eScience Security Workshop, Presentation, April 2005. http://grid.ncsa.uiuc.edu/GridShib/presentations/GridShib-uk-april05.ppt, last visited 14 Feb 2007.

[WeSi06] Von Welch and Frank Siebenlist: GridShib – An Interview with the Globus Consortium Journal (http://www.globusconsortium.org/journal/20060905/gridShib.html)

[YBC+07] D.R. Yocum, E. Berman, P. Canal, K. Chadwick, T. Hesselroth, G. Garzoglio, T. Levshina, V. Sergeev, I. Sfiligoi, N. Sharma, and S. Timm: FermiGrid. Submitted to Teragrid07, http://cd-docdb.fnal.gov/0020/002016/001/teragrid07-08-formated.doc, last visited 26 Feb 2007.