

# Dynamic Privilege Management Infrastructures Utilising Secure Attribute Exchange

J. Watt, R.O. Sinnott, A.J. Stell

National e-Science Centre, University of Glasgow, UK

## Abstract

Technologies which implement dynamic privilege management infrastructures will be crucial to the secure sharing of resources on the Grid, especially as the number of resources and participating sites increases. The DyVOSE project has successfully deployed Grid services secured with the PERMIS authorisation software implementing a static Privilege Management Infrastructure (PMI) model. The second stage of this project focuses on the extension of the current PERMIS infrastructure to include dynamic delegation of authority and cross-certification of institutional security policies. This paper describes the existing static PMI that has been used within the Grid Computing module as part of the advanced MSc at Glasgow University. We also outline an e-Science education use case that will be used to highlight how dynamic PMIs can be established using an extended version of PERMIS and utilising the Internet2 Shibboleth software to transfer user attributes and authentication tokens across institutional boundaries. This work addresses one of the key challenges in the Grid, supporting the dynamic establishment of secure Virtual Organisations (VOs).

## 1. Introduction

Security on the Grid is an area where much research still needs to be undertaken, standards developed and best practices documented. Security tends to be presented under the banner of AAA: Authentication; Authorisation and Accounting. Audit is sometimes added as a fourth A, closely related to Accounting and doesn't concern us here. Authentication (the establishment of identity) on the Grid is currently considered to be best achieved through Public Key Infrastructures (PKI), where users are identified by X509 Certificates which bind their unique username (DN) to a public key. These certificates are issued by a central Certificate Authority (CA) which is the root of trust for the whole PKI.

The establishment of identity is only part of the story, as Authentication provides no information on the access rights or privileges of that user on a particular resource. This problem is one of Authorisation, and is the focus of several parallel projects using a number of distinct technologies (Akonti [1], CAS [2] etc.). When resources need to be shared across institutional boundaries, the gatekeeper/Access control list approach of GSI becomes impossible to administer without surrendering some (if not all) of the local security policy. Further, this approach is far too coarse-grained to be taken up by certain security focused communities. The DyVOSE [3] project is

looking at using PERMIS [4] to make authorisation decisions based on a user's attributes, in particular, their role and function within an organisation.

The DyVOSE project is a JISC-funded 2 – year project involving the National e-Science Centre at the Universities of Glasgow and Edinburgh, and the University of Kent. The project aims to test technologies which support dynamic delegation of authority which is required to create scalable, secure virtual organisations (VOs) in an e-Science education context. The static model of privilege management has already been implemented, and it is the goal of the second phase of the project to implement a dynamic design, where multiple site policies are integrated into a single VO.

## 2. Technologies

### 2.1 PERMIS

The Privilege and Role Management Infrastructure Standards Validation (PERMIS) software is an authorization infrastructure that can realise Role Based Access Control (RBAC) [5] authorisation infrastructures. It is implemented as a Java-based API (or as SAML [6] request/response messages) which makes decisions on whether or not access to a particular resource is valid. PERMIS uses XML policies [7] (normally developed using bespoke editing tools) which define the rules specifying the access rights and associated actions that can

be invoked on resources within the VO. These policies include the definitions of roles (and their hierarchy), Sources of Authority (SOA) which are trusted to assign these roles, and the resource targets and actions which are governed by this policy. Roles are issued to users in the form of X509 Attribute Certificates (ACs) stored in a dedicated LDAP repository. The application gateway comprises an Application dependent Enforcement Function (AEF) and an Application independent Decision Function (ADF). A decision request currently contains: the user name, the target name, and the requested action. Optional parameters such as the time of day may be included. The decision is made based on the VO policy and the roles of the user retrieved from the AC at decision time. The PERMIS policy comprises a Role Allocation Policy (RAP) and a Target Access Policy (TAP). ACs are checked against the RAP by the AEF and valid attributes are passed to the ADF which returns a granted or denied response according to the enforced policy.

Static PMIs are formed by an SOA which issues RAPs and TAPs to its subordinate AA which allows it to assign roles to its users. Roles issued by other SOAs have no meaning in VOs formed with static delegation (unless hard wired into the local infrastructure, but this is not a scalable solution). Since we wish to facilitate multi-site collaboration without surrendering local security policy, a system for recognising the roles at collaborating institutions must be in place. Dynamic Delegation of Authority allows an AA controlled by an external SOA to be delegated the ability to assign roles meaningful to the home SOA. This is achieved through the issue of a RAP and a TAP to the remote SOA specifying the roles it may delegate to its users and their privileges. This way, a remote user can hold a role based in the home institution that will allow access to the home institution's target resources. This scenario is illustrated in Figure 1. A scenario where the remote roles are unknown to the home SOA requires complex dynamic delegation and role mapping and will be the focus of the final phase of the DyVOSE project.

## 2.2 Shibboleth

Shibboleth [8] is a system designed to assert attributes between a user's home organisation and the organisation hosting resources for the primary purpose of authorisation. In the simple use case, when a user requests access to a remote target resource, the user's home organisation can send specific

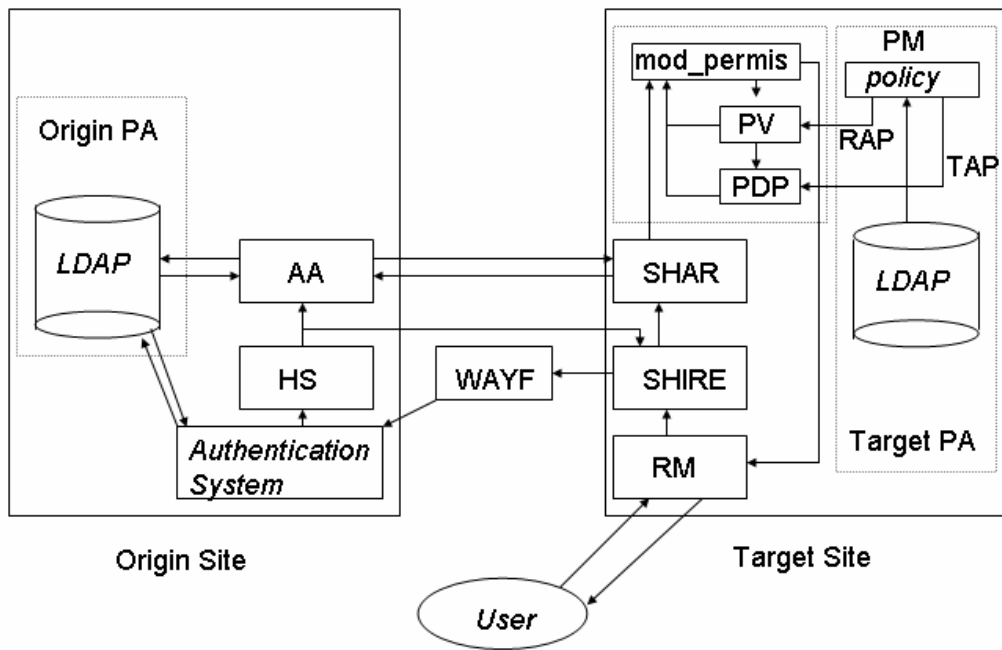
information about that user to the remote resource in a trusted attribute exchange. The presented attributes may then be used by the target resource to make decisions as to whether or not to grant the user access to that resource. User privacy is protected as the user may specify the specific attributes he wishes to release to the target resource, but additionally, attribute-based authorisation need not present any identifying information about the user, only information pertaining to their privileges on the remote system.

The components of Shibboleth required vary according to the role of the site. A complete Shibboleth system consists of an origin side (the user's home institution) and a target side (the resource they are attempting to access). The origin side comprises the following four components: the Handle Service (HS), the Attribute Authority (AA), the directory service (usually LDAP) and a local authentication system. The target side requires the Shibboleth Indexical Reference Establisher (SHIRE), the Shibboleth Attribute Requester (SHAR) and the resource manager (RM).

When a user attempts to access a Shibboleth protected target resource, the RM uses the SHIRE to redirect the user to a WAYF (Where Are You From) service which asks the user to select the institution (usually their home institution) that will authenticate them from a list of trusted organisations. Once authenticated, the HS at the origin site generates a temporary reference to the user by issuing a handle token (a SAML authentication assertion containing a user identifier). This handle is registered with the origin AA so it can match this with the user's attributes from their home AA. The user is then sent back to the target resource with their handle, which the SHAR uses to determine which AA can supply attributes about this user, and requests them. The origin AA then consults the Attribute Release Policies (ARPs) for the directory service entry corresponding to the supplied handle, queries the service for the user's attributes, and forwards to the target SHAR all the attributes it is entitled to know about the user according to the site ARP. The SHAR can perform some basic analysis using an Attribute Acceptance Policy, but passes the user's attributes back to the RM for it to make the final decision on target access.

## 2.3 SAAM Module

Integration of Shibboleth and PERMIS is currently implemented by the PERMIS Shibboleth Apache Authorisation Module (SAAM) [9], which allows PERMIS to enforce



**Figure 1:** Diagram showing the interoperation of the various subsystems within the Shibboleth-PERMISS SAAM integration. PERMISS SAAM components are shown within dotted lines.

access control on websites which use Shibboleth (or Apache) to provide user authentication. By careful tuning of the Apache configuration file, the SAAM can be loaded before Shibboleth thereby bypassing the Apache authorisation phase and making the authorisation decisions using PERMISS without interfering with the rest of the Shibboleth operation.

The PERMISS SAAM (very similarly to PERMISS) consists of three components. The Privilege Allocator (PA) subsystem, distributed to Origin sites, and is responsible for issuing attribute certificates and assigning privileges to users. The Policy Management (PM) subsystem contains tools which define the local RBAC policy, and stores the signed policy in an LDAP directory. Finally the PV/PDP (Privilege Verification/Policy Decision Point) has the job of validating the incoming Attribute Certificates and making the final access decision. Note the PV/PDP subsystem fulfils the same function as the AEF/ADF system described in Section 2.1.

The PV/PDP subsystem contains the important Shibboleth-Apache interface, `mod_permis`, which collects the information for the PV/PDP to make an access decision. Since a definitive access decision result is obtained from PERMISS SAAM when it is loaded, the Shibboleth authorisation is not invoked.

The integration of Shibboleth and PERMISS SAAM is detailed in Figure 1.

### 3. Static Privilege Management

Static delegation of authority implies that a central manager must be contacted to register managers who are authorised to assign roles. Once created these policies are stored and used to check and ensure that legal requests are permitted and illegal requests are denied.

The first phase of the DyVOSE project implemented a static PMI. This was focused upon supporting the Grid Computing module within the advanced MSc course taught at Glasgow University [10]. Specifically, the students were asked to write a GT3.3 Grid service which searched and sorted a large text file (complete works of Shakespeare) by submitting a set of Java universe jobs to the local NeSC Condor pool. Using the PERMISS Authorisation software, the students added a basic PMI to their Grid service, defining the roles of Lecturer, StudentTeam1 and StudentTeam2. The Grid service utilising the generic Grid-authorisation interface defined in GGF SAML AuthZ api was deployed so that only users holding the role were able to sort the text file, whereas the search functionality was made publicly accessible. This static PMI was successfully deployed and used to enforce

appropriate policy decisions by students taking the Grid Computing course.

Whilst supporting fine grained security models, static PMIs do not address a key aspect and perceived benefit of Grid technology – establishing VOs dynamically.

#### 4. Dynamic Privilege Management

Dynamically establishing VOs where multiple remote, fine-grained policies co-exist and need to be integrated and harmonised to meet the regulations of a particular VO poses numerous challenges. To scale such a system up requires that dynamic delegation of privilege is supported. Thus as the complexity and number of security policies increases, the ability of a given SOA to delegate responsibility to others is necessary.

Perhaps the biggest challenge in moving static PMI based approaches and technologies to support dynamic PMI infrastructures is a semantic one. Remote policies defining rules and regulations in terms of roles, targets and actions on those remote resources requires tool support that can facilitate the discovery, association, merging and promotion or suppression of policies denoting user privileges between sites. To support this, the DyVOSE project is looking towards the architecture in figure 2 as the basis for the investigation.

The Glasgow Privilege Management Infrastructure (PMI) will consist of an SOA which has the properties that it can assign ANY role to ANYONE in the world. The SOA also holds the PERMIS policy for that VO, which contains all the valid roles, rules and actions that are allowed on the target resource that PERMIS is protecting. The Glasgow SOA delegates via a RAP the ability to assign the role of Lecturer or Student to Glasgow users to Attribute Authority 1 (AA1). AA1 then can delegate a subset of its privileges to AA2, namely the ability to designate the role of Lecturer or Student to Glasgow Department of Computing Science users. This ability to delegate means that the SOA need not change its policy as new authorities which are delegated role assigning powers are created. A similar infrastructure will be created at Edinburgh [11], where the roles of Trainer and Trainee will have similar meaning to the Glasgow roles. Note that this role equivalence statement will become crucial when complex recognition of authority is implemented within the PERMIS software, as this will support the ability to delegate new roles that the delegating SOA is unaware of, but as the new role will contain a subset of the

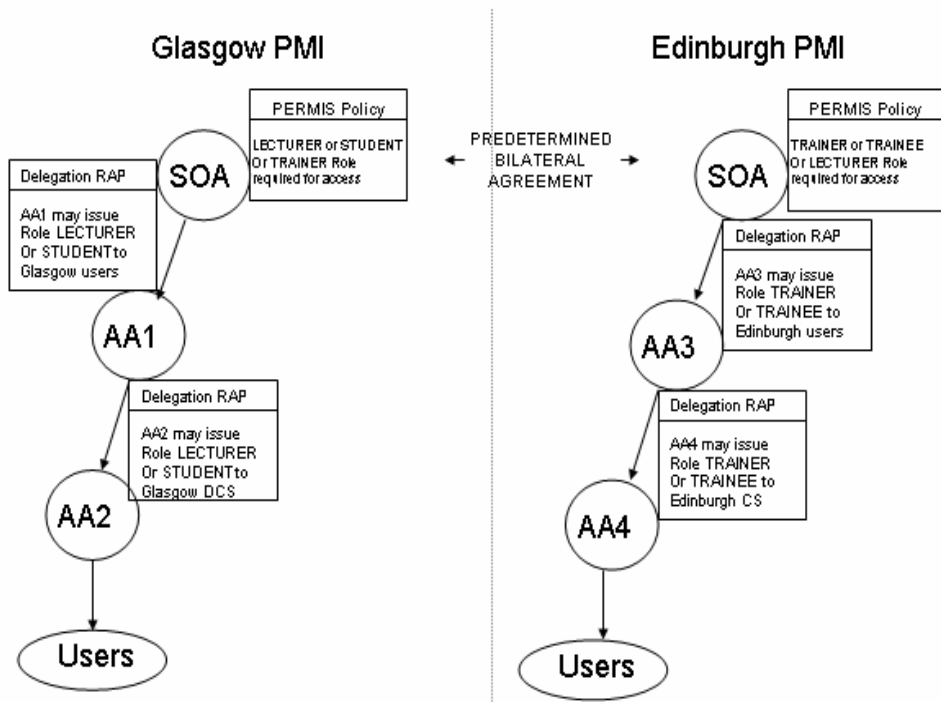
privileges of the old roles, the security of the original PERMIS policy remains uncompromised.

In static delegation, the roles at the remote institution would need to be hand written into the policy at the home institution, i.e. Glasgow would have to have the roles of trainer and trainee defined in the policy held by the SOA. Dynamic delegation factors away the role assigning powers to subordinate authorities, which may delegate the ability to assign Glasgow roles to Edinburgh Attribute Authorities, and vice versa. In this example, the Glasgow “Student” role may be assigned to Edinburgh Computing Science users, so they may access the Glasgow resource without the Glasgow SOA knowing about any Edinburgh roles. This trust relationship is agreed beforehand, where it is implicit that the role of Student at Glasgow and Trainee at Edinburgh are equivalent. Complex delegation allows new intermediate roles with less privilege than their superior role to be defined and assigned to remote attribute authorities. This will be investigated in the final phase of the DyVOSE project.

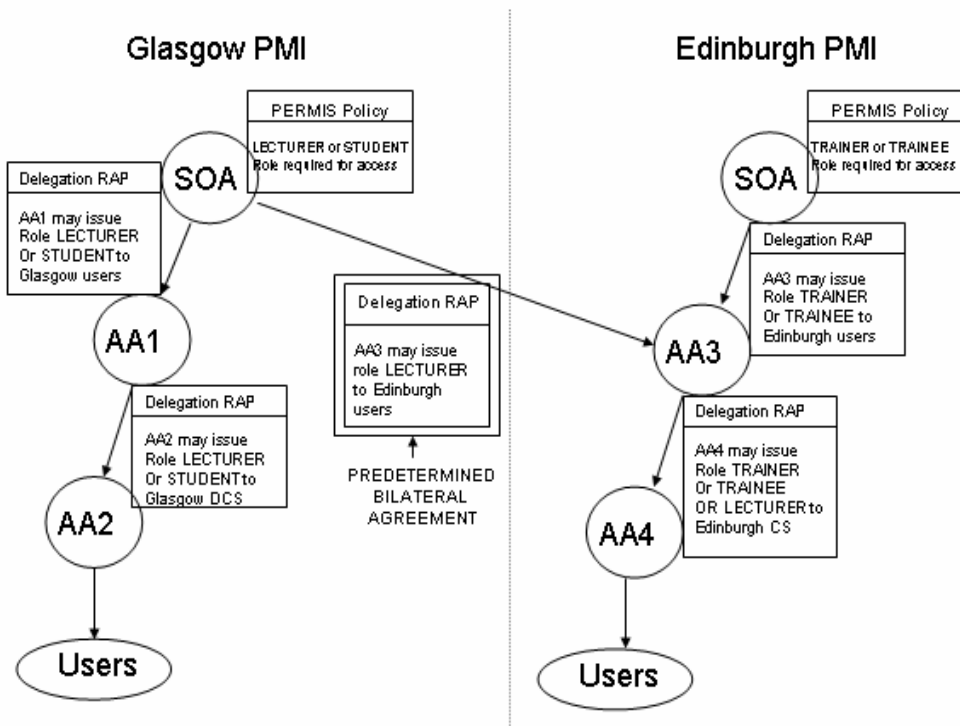
In this architecture, extensions to the PERMIS infrastructure will support the tool-supported identification and mapping of roles, actions and targets between the resources offered at Glasgow University within the Grid Computing module, and resources at Edinburgh used by the EGEE project trainers and trainees.

This infrastructure will use Shibboleth to transport authentication information to target resources from origin sites, for subsequent RBAC processing using PERMIS. The use of the PERMIS SAAM module will be explored for this purpose.

The testing of the dynamic PMI is being tackled in two stages. The first stage is the establishment of secure attribute assertion exchange using the Shibboleth software. This is being investigated on a local Glasgow testbed, where policies are stored in several LDAP servers, and Shibboleth targets and origins established. At present we are focused on static PMI based approaches to populating these policy servers, however exploration of bulk loader [12] approaches from PERMIS will also be investigated. The second stage will involve deploying this infrastructure (including the refinement of the policies to be targeted towards explicit roles, resources etc) at Edinburgh. In both of these stages, we will utilise an appropriate Grid service (to be determined based upon the student assignment). This will likely make use of local Condor pools, the



**STATIC DELEGATION OF AUTHORITY**



**SIMPLE DYNAMIC DELEGATION OF AUTHORITY**

**Figure 2:** The top figure shows delegation in a static PMI. All roles (including external ones) must be present in the site policy. In dynamic delegation, the policy only contains local roles, and the ability to assign local roles is delegated to an external AA. This method of delegation surrenders less local security.

Scotgrid [13] infrastructure and the National Grid Service [14]. A key part of this work will be in establishing how the PERMIS cross-certification tool will allow Edinburgh roles and resources to be seamlessly linked with Glasgow roles and resources to dynamically establish a VO.

All of this infrastructure will be in place for the Advanced MSc at both institutions, with the Glasgow students starting their course in January 2006. User manuals and best practise documents are pending for the static PMI work, with corresponding documents for the dynamic PMI following after successful integration.

## 5. Future Work

Integration of the authorisation infrastructure described above with the Globus Toolkit will be facilitated by the release of the SAML 2.0 based Shibboleth 1.3. This new version will be the final major release aligned with SAML 1.1, will include an RPM for Shibboleth and its dependencies, and will allow the use of non-URL addresses as targets (i.e. Grid URIs).

The GGF have also recently initiated standards work looking at issues fundamental to the dynamic establishment of VOs: the definition of standard attributes for OGSA Authorisation. This work is exploring the attributes and delivery mechanisms through which distributed policy agreements can be defined, agreed and enforced [OGSAAAuth]. Thus in the way in which eduPerson attributes underpin the Shibboleth technologies as the standard attributes for exchange, the wider challenges of attributes pertinent to Grid based VOs is being explored. As such the work is very much complimentary to the work defined here. The PERMIS project is continuing its releases supporting dynamic delegation, with the complete implementation expected in the first quarter of 2006. This implementation will support dynamic delegation of authority, cross-certification, support for SAML 2.0 and will provide a separation of duties implementation (which stops users with conflicting roles exploiting them).

The NMI GridShib [15] project is tackling the problem of using Grid Services as Shibboleth targets. It aims to provide an implementation whereby a Grid Service can authenticate a user using the GSI, following which, the address of the Shibboleth attribute service is determined and used to obtain selected user attributes from the Shibboleth service that the Grid Service is allowed to see. The Grid Service can then use these attributes to make authorisation decisions.

The findings of this project may provide useful input in how to complete the PERMIS-Shibboleth/Globus integration.

It is hoped the software will be available in time for the DyVOSE project to test these new approaches, which may form the basis of a new project further investigating advanced authorisation infrastructures.

## 6. Conclusions

Shibboleth is fast becoming the de facto method of securely exchanging attributes between enterprises. However the authorisation features it provides, in general, are too coarse-grained to be of use in building dynamic virtual organisations. Using the GGF SAML Authz callout, we have described the implementation of a VO that uses Shibboleth to exchange user attributes across two institutions, but calls the PERMIS authorisation software to make complex authorisation decisions based on separately defined user policies at each institution using the PERMIS SAAM Apache module. Building on the first half of the DyVOSE project, this infrastructure will be deployed in a University environment by students with little or no prior experience. Dynamic delegation allows each site to maintain complete control over its security policy, but delegates the authority to issue meaningful roles to members in other institutions without having to rewrite the local policy. This will be an important feature in the construction of VOs for Grid applications.

The experiences and findings of the real-life scenario detailed above is being fed into the GGF authorisation activities, and will provide input into the best practises of performing complex authorisation that will be a stepping stone to application within a Grid context.

## 7. References

- [1] W.Johnston, S.Mudumbai, M.Thompson, "Authorisation and Attribute Certificates for Widely Distributed Access Control", IEEE 7<sup>th</sup> International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Stanford, CA, June, 1998, p340-345 (<http://www-itg.lbl.gov/security/Akenti/>)
- [2] L.Pearlman et al., "A Community Authorisation Service for Group Collaboration" in Proceedings of the IEEE 3<sup>rd</sup> International Workshop on Policies for Distributed Systems and Networks. 2002.

- [3] Dynamic Virtual Organisations in e-Science Education project (DyVOSE)  
[www.nesc.ac.uk/hub/projects/dyvose](http://www.nesc.ac.uk/hub/projects/dyvose)
- [4] Privilege and Role Management Infrastructure Standards Validation project  
[www.permis.org](http://www.permis.org)
- [5] D. W. Chadwick, A. Otenko, E. Ball. "Role-based access control with X.509 attribute certificates", *IEEE Internet Computing*, March-April 2003, pp. 62-69
- [6] OASIS. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1, Sep 2003 <http://www.oasis-open.org/committees/security/>
- [7] D.W.Chadwick, A. Otenko. "RBAC Policies in XML for X.509 Based Privilege Management" in Security in the Information Society: Visions and Perspectives: *IFIP TC11 17th Int. Conf. On Information Security (SEC2002)*, May 7-9, 2002, Cairo, Egypt. Ed. by M. A. Ghonaimy, M. T. El-Hadidi, H.K.Aslan, Kluwer Academic Publishers, pp 39-53
- [8] Shibboleth, <http://shibboleth.internet2.edu>
- [9] W. Xu, D. Chadwick, A. Otenko, "Development of a Flexible PERMIS Authorisation Module for Shibboleth and Apache Server", accepted for 2<sup>nd</sup> European PKI Workshop, University of Kent, July 2005
- [10] R.O. Sinnott, A.J. Stell, J. Watt, "Experiences in Teaching Grid Computing to Advanced Level Students", Proceedings of the 5<sup>th</sup> IEEE International Symposium on Cluster Computing and the Grid, Cardiff, 19<sup>th</sup>-12<sup>th</sup> May 2005
- [11] R.Mann, e-Science Institute, Personal communications
- [12] D. Chadwick, E. Ball, P. Langley, "PERMIS Directory Bulk Loader User Manual",  
<http://sec.isi.salford.ac.uk/permis/private/dbl/DBLUserGuideV1.1.doc>
- [13] ScotGrid, <http://www.scotgrid.ac.uk>
- [14] National Grid Service  
<http://www.ngs.ac.uk/>
- [15] V.Welch, T.Barton, K.Keahey, F.Siebenlist, "Attributes, Anonymity and Access: Shibboleth and Globus Integration to Facilitate Grid Collaboration", submitted to 4<sup>th</sup> Annual PKI R&D Workshop (<http://grid.ncsa.uiuc.edu/GridShib/>)