Experiences of Applying Advanced Grid Authorisation Infrastructures

R.O. Sinnott[1], A.J. Stell[1], D.W. Chadwick[2], O.Otenko[2]
[1]National e-Science Centre, University of Glasgow
{ros@dcs.gla.ac.uk, ajstell@dcs.gla.ac.uk}
[2]IS Security Research Centre, University of Salford
{D.W.Chadwick@salford.ac.uk, o.otenko@salford.ac.uk}

Abstract

The widespread acceptance and uptake of Grid technology can only be achieved if it can be ensured that the security mechanisms needed to support Grid based collaborations are at least as strong as local security mechanisms. The predominant way in which security is currently addressed in the Grid community is through Public Key Infrastructures (PKI) to support *authentication*. Whilst PKIs address user identity issues, authentication does not provide fine grained control over what users are allowed to do on remote resources (*authorisation*). The Grid community have put forward numerous software proposals for authorisation infrastructures such as AKENTI [1], CAS [2], CARDEA [3], GSI [4], PERMIS [5,6,7] and VOMS [8,9]. It is clear that for the foreseeable future a collection of solutions will be the norm. To address this, the Global Grid Forum (GGF) have proposed a generic SAML based authorisation API which in principle should allow for fine grained control for *authorised* access to any Grid service. Experiences in applying and stress testing this API from a variety of different application domains are essential to give insight into the practical aspects of large scale usage of authorisation infrastructures. This paper presents experiences from the DTI funded BRIDGES project [10] and the JISC funded DyVOSE project [11] in using this API with Globus version 3.3 [12] and the PERMIS authorisation infrastructure.

1. Introduction

Today, collections of distributed individuals and institutions in science and industry are increasingly forming virtual organisations (VOs) to pool resources such as data sets, data archives, CPUs, or specialised equipment from astronomical radio-telescopes through to medical imaging scanners. Grid technology presents itself as one of the main ways in which such VOs can be established. With the open and collaborative nature of the Grid, ensuring that local security constraints are met and not weakened by Grid security solutions is paramount. PKIs represent the most common way in which security is addressed. Through PKIs, it is possible to validate the identity of a given user requesting access to a given resource. For example, with the Globus toolkit [12] solution, *gatekeepers* are used to ensure that signed requests are valid, i.e. from known collaborators. When this is so, i.e. the Distinguished Name (DN) of the requestor is in a locally stored and managed *gridmap* file, then the user is typically given access to the locally set up account as defined in the *gridmap* file.

There are several key limitations with this approach with regard to security however. Most importantly, the level of granularity of security is limited. There is no mention of what the user is allowed to do once they have gained access to the resource. Another issue with this approach is that it works on the assumption that user certificates are provided by an acknowledged certificate authority (CA). In the UK, a centrally managed CA at Rutherford Appleton Laboratories exists which (necessarily!) has strict procedures for how certificates are allocated. Users are expected to “prove” who they are in order to get a certificate, e.g. through presenting their passports to a trusted individual. This is a human intensive activity and one which is likely to have scalability issues once it is rolled out

to the wider community, e.g. to industry and larger groups such as students taking Grid/e-Science courses. Having users personally take care of their private keys is another limitation of this approach.

In short, current experiences with PKIs [13, 14] as the mechanism for ensuring security on the Grid have not been too successful [15, 16]. Authorisation infrastructures offer extended and finer grained security control when accessing and using Grid resources. Numerous technological solutions have been put forward providing various levels of authorisation capabilities e.g. AKENTI [1], CAS [2], CARDEA [3], GSI [4], PERMIS [5,6,7] and VOMS [8,9]. Examples of how these compare to one another is described in [17, 18, 19]. It is too early to say if large scale use of attribute certificates (ACs) for user authorisation, based on infrastructures such as PERMIS, will be successful or not. However, few other alternatives currently exist, so practical experience is required. In order for large scale use to be facilitated, dynamic (rather than static) delegation of authority is required. In the current PERMIS infrastructure, static delegation of authority means that a central authority has to be contacted, and register local managers in its policy, before managers are entitled to assign privileges to subordinates. With dynamic delegation of authority, local managers do not need to be registered, but are given the privilege to delegate when they are first given privileges to use the system. Managers can then allocate privileges to staff and students as required, without having to contact the central authority first to get permission. Through this, a federated and scalable model of security authorisation can be realised. In developing this federated privilege management infrastructure (PMI) model, key challenges have to be overcome which are common to most, if not all, uses of Grid technology – the dynamic establishment of Virtual Organisations (VO). VOs allow shared use of computational and data resources by collaborating institutions. Establishing a VO will require that efficient access control mechanisms to the shared resources by known individuals are in place. However, currently in the Grid community access control is usually done by comparing the authenticated name of an entity to a name in an Access Control List. This approach lacks scalability and manageability as discussed in [15]. Dynamic delegation of privileges offers a more realistic approach that could shape future Grid security, especially when it is rolled-out to the masses, e.g. Grid students, industry.

2. Authorisation Background

Authentication should be augmented with authorisation capabilities, which can be considered as what Grid users are allowed to do on a given Grid end-system. Thus “what users are allowed to do” can be interpreted as the privileges that the users have been allocated on those end-systems. The X.509 standard [20] has standardised the certificates of a privilege management infrastructure (PMI). A PMI can be considered as being related to authorisation in much the same way as a PKI is related to authentication. Consequently, there are many similar concepts in PKIs and PMIs. An outline of these concepts and their relationship are discussed in detail in [6].

A key concept from PMI are attribute certificates (ACs) which, in much the same manner as public key certificates in PKI, maintain a strong binding between a user’s name and one or more privilege attributes. The entity that digitally signs a public key certificate is called a Certification Authority (CA) whilst the entity that signs an AC is called an Attribute Authority (AA). The root of trust of a PKI is sometimes called the root CA – which in terms of the UK e-Science community is given by the Grid Support centre at RAL [21]. The root of trust of the PMI is called the Source of Authority (SOA). CAs may have subordinate CAs whom they trust and to which they delegate the powers of authentication and certification. Similarly, SOAs may delegate their power of authorisation to subordinate AAs. If a user needs to have their signing key revoked, a CA will issue a certificate revocation list. Similarly, if a user needs to have authorisation permissions revoked, an AA will issue an attribute certificate revocation list (ACRL). Typically, a given users’ access rights are held as access control lists (ACLs) within each target resource. In an X.509 PMI, the access rights are held within the privilege attributes of ACs that are issued to users. A given privilege attribute within an AC will describe one or more of the user’s access rights. A target resource will then read a user’s AC to see if they are allowed to perform the action being requested.

The X.812 | ISO 10181-3 Access Control Framework standard [22] defines a generic framework to support authorisation as depicted in Figure 1.

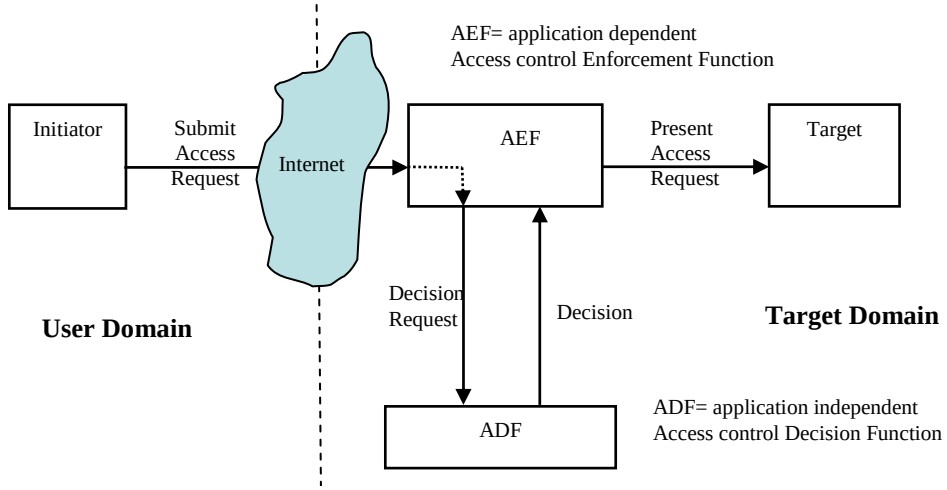

Figure 1: Overview of X.812 Access Control Function

With this model an *initiator* is attempting to access a *target* in a remote domain. Two key components are put forward in [22] to support authorised access to the target: an Access control Enforcement Function (also known as a Policy Enforcement Point (PEP)) and an Access control Decision Function (also known as a Policy Decision Point (PDP)). The PEP ensures that all requests to access the target are authorised through checking with the PDP. The PDP's authorisation decision policy is often represented through collections of rules (policies), e.g. stored in a Lightweight Directory Access Protocol (LDAP) server.

The different authorisation infrastructures associated with Grid technology have put forward their own mechanisms for realising PEPs and PDPs. Recently however, the GGF has put forward a generic API – the SAML AuthZ API - which in principle provides a generic PEP that can be associated with an arbitrary authorisation infrastructure [23]. The Grid specification is an enhanced profile of the OASIS Security Assertion Markup Language v1.1 [24]

2.1 GGF SAML AuthZ API

The OASIS SAML specification defines a number of elements for making assertions and queries regarding authentication, authorization decisions and attributes The OASIS SAML AuthZ specification defines a message exchange between a policy enforcement point (PEP) and a policy decision point (PDP) consisting of an *AuthorizationDecisionQuery* flowing from the PEP to the PDP, with an assertion returned containing some number of *AuthorizationDecisionStatements*

The *AuthorizationDecisionQuery* itself consists of

- *A Subject* element containing a *NameIdentifier* specifying the initiator identity
- A *Resource* element specifying the resource to which the request to be authorized is being made.
- One or more *Action* elements specifying the actions being requested on the resources

The GGF SAML profile specifies a *SimpleAuthorizationDecisionStatement* (essentially a granted/denied Boolean) and an *ExtendedAuthorizationDecisionQuery* that allows the PEP to specify whether the simple or full authorization decision is to be returned. In addition the GGF query supports both the pull and push modes of operation for the PDP to obtain attribute certificates, and has added a *SubjectAttributeReferenceAdvice* element to allow the PEP to inform the PDP where it may obtain the subject's attribute certificates from. The interactions supported by this API are depicted in Figure 2.

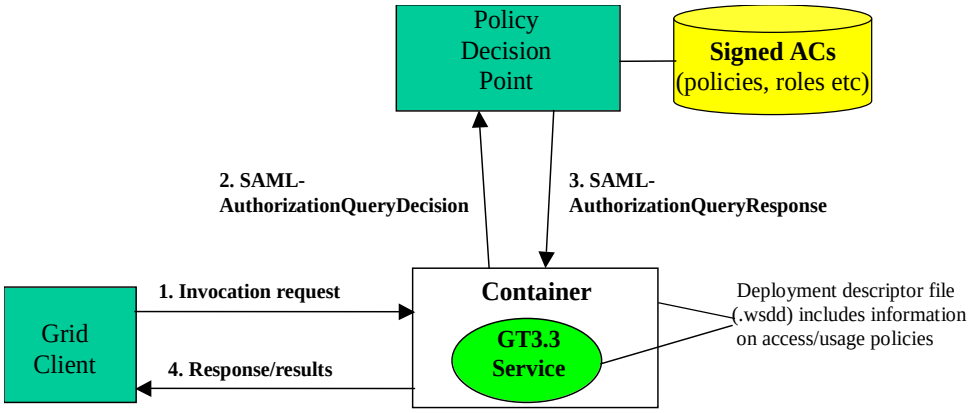

Figure 2: GGF SAML AuthZ API

Through this SAML AuthZ API, a generic PEP can be achieved which can be associated with arbitrary (GT3.3) Grid services. Thus rather than developers having to explicitly engineer a PEP on a per application basis, the information contained within the deployment descriptor file (.wsdd) when the service is deployed within the container, is used. Authorisation checks on users attempting to invoke "***methods***" associated with this service are then made using the information in the .wsdd file and the contents of the LDAP repository (PDP) together with the DN of the user themselves. Note that this "method" authorisation basis extends current security mechanisms such as GSI which work on a per service/container basis. This generic solution can be applied to numerous infrastructures used to realise PDPs such as PERMIS.

2.2 PERMIS Background

The Privilege and Role Management Infrastructure Standards Validation (PERMIS) project [7] was an EC project that built an authorisation infrastructure to realise a scalable X.509 AC based PMI. Through PERMIS, an alternative and more scalable approach to centrally allocated X.509 public key certificates can be achieved through the issuance of locally allocated X.509 ACs.

The PERMIS software realises a Role Based Access Control (RBAC) authorisation infrastructure. It offers a standards-based Java API that allows developers of resource gateways (gatekeepers) to enquire if a particular access to a resource should be allowed. The PERMIS RBAC system uses XML based policies defining rules, specifying which access control decisions are to be made for given VO resources. These rules include:

- definitions of subjects that can be assigned roles
- definitions of SOAs – local managers trusted to assign roles to subjects
- definitions of roles and their hierarchical relationships
- definitions of what roles can be assigned to which subjects by which SOAs
- definitions of target resources, and the actions that can be applied to them
- definitions of which roles are allowed to perform which actions on which targets
- the conditions under which access can be granted to roles.

Roles are assigned to subjects by issuing them with X.509 Attribute Certificate(s). A graphical tool called the Privilege Allocator (PA) and a simpler version termed the Attribute Certificate Manager (ACM) have been developed to support this process. Once roles are assigned, and policies are developed, they are digitally signed by a manager and stored in one or more LDAP repositories.

The process to set up and use PERMIS can be split into two parts: *Administration* and *Use*. To set up and administer PERMIS requires the use of a LDAP server to store the attribute certificates and reference the SOA root certificate. A local certificate authority (CA) is also required to be set up using OpenSSL – this designates the SOA and all user certificates created from this CA must have a

Distinguished Name that matches the structure of the LDAP server. The DN of the user certificate is what is used to identify the client making the call on the grid service. Establishing local CAs matching the structures of the LDAP repository is not without issues which need to be resolved, e.g. in ensuring that locally generated certificates are recognised (trusted) by other remote CAs since there is no root of trust.

From the user's perspective, once the administrator has set up the infrastructure, the PERMIS service is relatively easy to use. Unique identifiers are placed as parameters into the user's grid service deployment descriptor (.wsdd file). These are the Object Identification (OID) number of the policy in the repository, the URI of the LDAP server where the policies are held and the SOA associated with the policy being implemented. Once these parameters are input and the service is deployed, the user creates a proxy certificate with the user certificate created by the local CA to perform strong authentication. The client is run and the authorisation process allows or disallows the intended action.

3. Experiences of Authorisation

The GGF SAML AuthZ API offers, in principle, a generic way in which authorisation can be made. It is clear that direct experiences in applying/stress testing this mechanism are needed from a variety of different application domains. This has been undertaken within the BRIDGES project where the emphasis on security has been on life science data security, and the DyVOSE project where focus has been on education case studies looking at method level security.

3.1 Bridges Background

The Biomedical Research Informatics Delivered by Grid Enabled Services (BRIDGES) project [10] has been funded by the UK Department of Trade and Industry to directly address the needs of the Cardiovascular Functional Genomics (CFG) [25]. The CFG project is investigating the causes of hypertension and involves five UK and one Dutch site through pursuing a strategy combining studies on rodent models of disease with studies of patients and population DNA collections. Currently many of the activities that the CFG scientists undertake in performing their research are done in a time consuming and largely non-automated manner often requiring navigation to many different data resources, web sites and following multiple links to potentially relevant information. In their pursuit of novel genes and understanding their associated function, the scientists often require access to large scale compute facilities to analyse their data sets, e.g. in performing large scale sequence comparisons or cross-correlations between large biological data sources.

The BRIDGES project is investigating the application of the Globus toolkit [12] to support HPC bioinformatics BLAST services using large HPC facilities; and the Open Grid Services Architecture – Data Access and Integration (OGSA-DAI) [26] and IBM's Information Integrator product [27] to deal with federation of distributed biomedical data. A key requirement of the scientist and hence focus of the BRIDGES work is security. Broadly speaking, the CFG scientific data can be classified dependent upon its security characteristics into three groups: public data (with no/minimal security, e.g. publicly curated genomic databases); shared data (belonging to the CFG scientists/consortia, e.g. shared research data sets); private data (belonging to given CFG sites and unavailable to anyone else, e.g. personal medical records).

3.1.1 Bridges Security Considerations

Figure 3 provides an overview of the system used to explore the SAML AuthZ interface in Bridges. The GT3-PERMIS extensions realising the GGF SAML AuthZ profile allows for authorisation at portal access and subsequent Grid service invocations to be supported. The portal is personalised to CFG scientists based on the policies that have been defined for them, i.e. their role, targets etc. These policies are accessed when users log-in. Thus scientists are restricted to seeing and using services that are appropriate based on their roles.

A typical scenario that the infrastructure supports is:

- The user requests access to the CFG portal;
- The access request results in a SAML query being raised to ensure that this user is authorised to access the portal (by ensuring an appropriate policy is available in the secure LDAP repository);
- If successful (the user is authorised), the portal is configured/personalised to display the services that are associated with that user;
- At this point, the user can invoke various services (they are entitled to use) – one of these is a syntenic relation visualisation service (SyntenyVista).
- Upon launching SyntenyVista (using WebStart technologies) the users can use data available in the repository (which itself provides an OGSA-DAI front end and exploits IBM Information Integator to integrate and where possible federate various remote public data resources);
- The user may then visually explore genomic data sets and potentially export these onto the high throughput computing resources ScotGrid for sequence similarity checking (BLAST) against other query sequences.

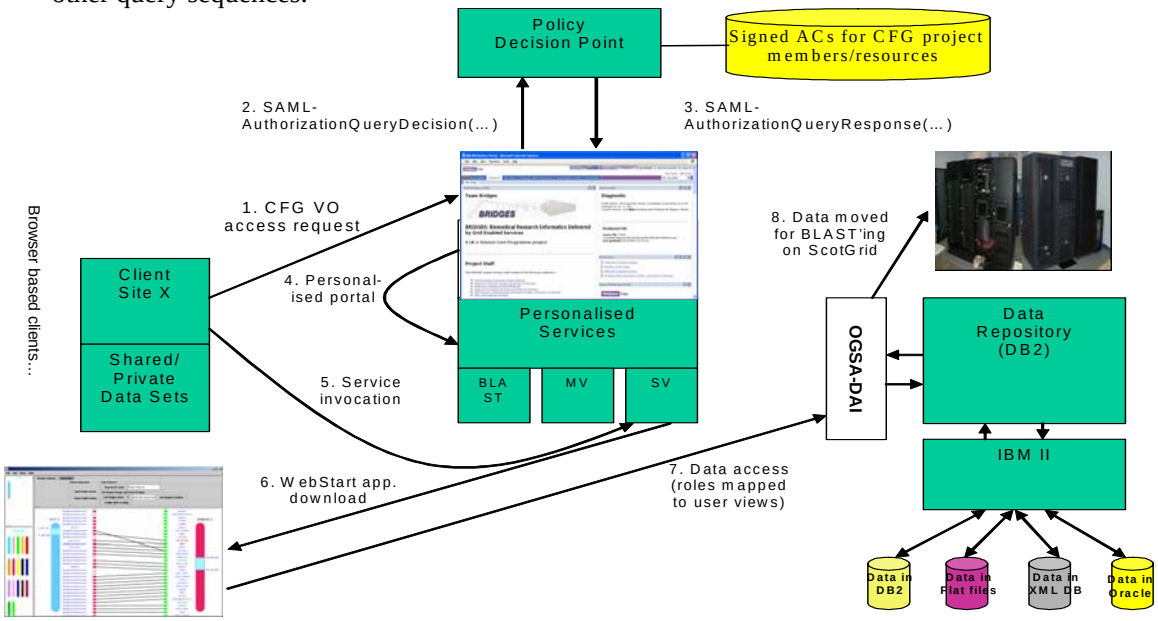

Figure 3: System Design and Usage Scenario

In the current implementation the usage of SyntenyVista offers direct visualisation of data sets available via the repository (from ensembl [28]). It is planned however that the user is restricted to seeing and visualising the data sets that they are entitled to see based upon their role within the CFG virtual organisation (VO), this applies also to the usage/invocation of GT3 based Blast services, i.e. that they will be restricted to those users ***and*** those data sets that meet appropriate security restrictions. For this purpose, the PERMIS Policy Editor tool (shown in Figure 4) has been used to develop appropriate policies based upon the specific roles in the projects and the capabilities to be associated with those roles.

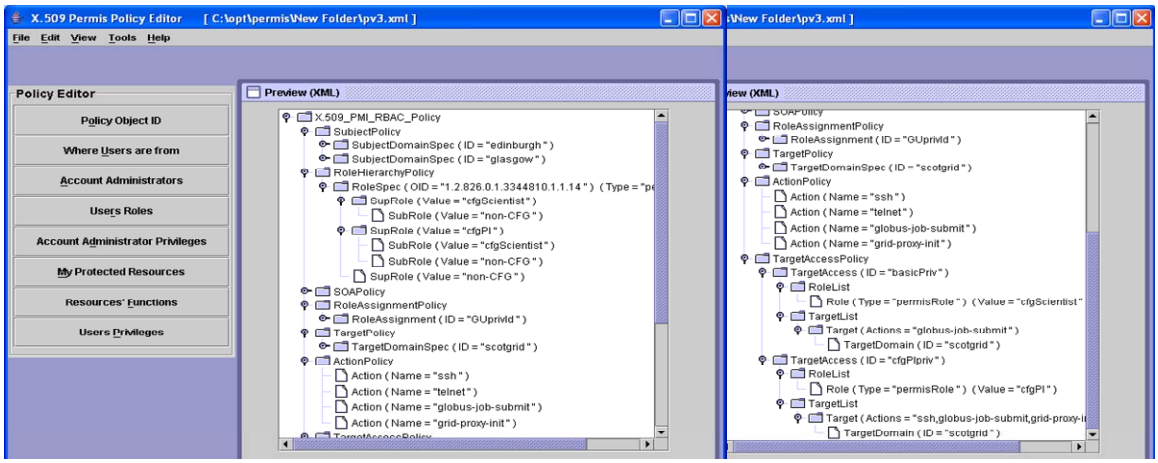

Figure 4: BRIDGES/CFG Policies Developed with the PERMIS Policy Editor

3.1.2 Bridges Security Experiences

The emphasis on security in BRIDGES is upon data security. Work has investigated how best to map advanced Grid authorisation infrastructures such as PERMIS/SAML AuthZ with best practice in the database management systems (DBMS) world. DBMS have extensive experience in addressing security aspects, e.g. with how to ensure users access data that they are entitled to. The relation between how much authorisation should be done through Grid software and how much should be left to the DBMS is not always clear in the Grid community. Explorations of BRIDGES in this area are that the PERMIS (Grid) roles within the CFG VO (as extracted from the AC repository) are mapped against specifically established user views of data sets available via the DB2 data repository. However one issue that has been encountered with the SAML AuthZ profile is the lack of granularity in how users might invoke actions. For example, different actions may or may not be allowed depending upon the data that they wish to access and potentially change. The SAML AuthZ profile does not currently allow actions to be distinguished based upon the parameters that might be associated with them. As a result, the GT3 based BLAST service cannot be restricted to BLAST those data sets that are appropriate to the invoker. Instead, the SAML AuthZ specification supports either a SecureGrid BLAST service or a non-secure BLAST service. Thus when the portal is personalised per user/role, it is not possible to distinguish the usage of individual operations, e.g. to allow arbitrary invocations of actions where the data sets themselves might change.

Further, the identification of explicit targets and actions applicable to the data in the DB itself is not easily reconciled. A naïve approach would be for example to explicitly have read/write actions on contents of the database itself, e.g. read/write access to individual tables. The difficulty in this situation is that the DB is perpetually being modified (extended) as new data sets are added and changed. As a result, new policies would have to be defined with each DB change which impacts directly upon the scalability of the approach. In addition, attaching policies to individual data elements would face immediate scalability problems.

To address this issue, the project is investigating how the schemas defining the secure data structures themselves might be extended in a more scalable way to include security attribute information. Thus policies can be formulated to query data sets that do/do not have appropriate security attributes depending upon the policy in place. Through this mechanism, a generic approach to secure authorised access to DB contents can be achieved.

3.2 DyVOSE Background

The Dynamic Virtual Organisations in e-Science Education (DyVOSE) project [11] began in May 2004 and involves the Universities of Glasgow, Salford and in the second phase of the project, the University of Edinburgh. It was funded through the JISC Core Middleware programme.

One of the initial goals of DyVOSE is to explore scalability issues in the usage of advanced authorisation infrastructures such as PERMIS. To this extent, the PERMIS technology is being applied in the advanced MSc Grid Computing module at the University of Glasgow. It is worth noting that the first lecture had over 50 students.

3.2.1 DyVOSE Security Considerations

Within the DyVOSE project the PERMIS tools such as the Policy Editor and Privilege Allocator have been used to create policies to authorise what the students are allowed to do as part of their programming assignment. To explore the authorisation infrastructure, the students have been asked to develop a GT3.3 service (*searchSortGridService*) which wraps a Condor based application (this service offers two methods to search (*searchMethod*) and sort (*sortMethod*) a large (5MB) text file[1]). The students themselves have been split into groups with the authorisation policy to ensure that method *sortMethod* can ***only*** be invoked by members of your student group and the lecturing staff, and that method *searchMethod* can be invoked by everyone.

Initially the students were asked to develop this policy themselves through the PERMIS Policy Editor. The usability of these tools is a key part in development of authorisation infrastructures. The output of the Policy Editor is an XML-based policy which identifies specific roles (*studentteam1, studentteam2* and *lecturer*), specific targets (*searchSortGridService*) and specific actions on that target (*searchMethod* and *sortMethod*). This XML policy is then input to the Privilege Allocator tool which is used to denote specific users associated with that given rule (i.e. the students themselves); to digitally sign the policy and store it in the LDAP server.

3.2.2 DyVOSE Security Experiences

All of the students were able to successfully create the policy defined above using the PERMIS Policy Editor with minimal help from staff. It should be noted that the students were informed of various background information that they would need to create the policy including the Policy Domain to use ("O=University of Glasgow, C=GB"), the Source of Authority to use ("CN=Administrator, O=University of Glasgow, C=GB") and the Policy Object Identifier (1.0.0.1 for student group 1 and 1.0.0.2 for student group 2).

The students were requested to critically evaluate the PERMIS tools for this purpose, with these results being sent back to the PERMIS team for HCI improvements and minor bug fixes, e.g. problems in cross platform (Unix/Windows) versions of the tool and functionality in the tool that has not yet been implemented (although the buttons/pull down menus exist).

The student policies themselves have been signed and stored as ACs within the LDAP server. At the time of writing the students are completing their assignment which is using these authorisation policies. The working solution demonstrating that these policies and the SAML AuthZ API are working has been produced however.

Establishing a working solution was not without issue however. For example, one overhead is in environment settings that must be configured before the PERMIS-GT3.3 solution can be used. The CLASSPATH environment variable, for instance, is sensitive to change: it must include most of the JAR files in the Globus installation library, but must not include certain specific ones if an Ant build script is to be used to run the service client. Once these environment settings are identified, however, these can be incorporated into a script, which then only needs to be run once.

---

[1] The complete works of Shakespeare.

4. Conclusions and Future Plans

It is clear that detailed explorations are needed to assess the suitability of next generation Grid middleware. The work undertaken within the DyVOSE project has shown that the GGF SAML AuthZ API does provide a generic and useful mechanism through which fine grained authorisation can be achieved using GT3.3 and the PERMIS infrastructure. The BRIDGES project has shown the current limitations of this API which are being addressed by the GGF security authorisation working group through support for parameters in actions.

Continued feedback on the PERMIS tools is an equally important activity. Students' experiences within the DyVOSE project are providing the PERMIS team with detailed feedback on the usability of these tools. These stem from needed functionality through to improvements to the HCI aspects of these tools.

The work in exploring the SAML AuthZ API has also identified issues with the Globus toolkit which have been fed back to the Globus team. Specifically, within the GT3.3 release, certain Globus source code was required to be commented out before PERMIS could run with version it. Delays were also incurred due to the GT3.3 version compatible with PERMIS only being accessible via the CVS repository as opposed to the web site link. It is worth noting that it has been stated by the Globus team [30] that this SAML AuthZ API will be supported in future versions of the Globus software.

This work is addressing scalability issues of security infrastructures. A local central CA has issues with the overall manageability of PKIs, and does not address authorisation issues. A more realistic model would be to have local CA infrastructure to issue certificates, e.g. to students as part of their matriculation. Within DyVOSE and BRIDGES a local certificate authority was established using OpenSSL [29]. Whilst relatively straightforward to achieve, there are issues in recognition of these certificates by other CAs within PKIs, such as the UK e-Science CA. Since no root of trust exists between these CAs, solutions might be based upon some form of bridging solutions [31]. However, given the limitations of PKIs a better solution would be to support dynamic establishment and recognition of trust to support authorisation. The second phase of the DyVOSE project will, through extensions to the PERMIS technologies, investigate how dynamic delegation of trust can be achieved. In this situation, collections of distributed policies issued by various remote SOAs will be dynamically recognised (locally) and used as the basis for establishing the rules through which the dynamic VOs will be managed and enforced. This will benefit from the Shibboleth suite of protocols [33] for transport of policy information.

The explorations being undertaken in the BRIDGES and DyVOSE projects are providing valuable insight into the scalability and suitability of advanced authorisation infrastructures to establish VOs. These experiences are feeding in to numerous other areas. These include applications of Grid technology to establish VOs within the clinical science domain as part of the VOTES project [32], and as input to the UK e-Science Grid Engineering Task Force – specifically the action line associated with authentication, authorisation and accounting. Experiences in the application of PERMIS infrastructure have also been presented to the UK e-Science Security Task Force as part of an on-going activity in establishing best practice and usage of Grid security software.

5. References

[1] Johnston, W., Mudumbai, S., Thompson, M. Authorization and Attribute Certificates for Widely Distributed Access Control, IEEE 7th Int. 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 3rd International Workshop on Policies for Distributed Systems and Networks. 2002.
[3] Lepro, R., Cardea: Dynamic Access Control in Distributed Systems, NASA Technical Report NAS-03-020, November 2003
[4] Globus Grid Security Infrastructure (GSI), http://www-unix.globus.org/toolkit/docs/3.2/gsi/index.html

[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] D.W.Chadwick, A. Otenko, The PERMIS X.509 Role Based Privilege Management Infrastructure, Future Generation Computer Systems, 936 (2002) 1–13, December 2002. Elsevier Science BV.
[7] Privilege and Role Management Infrastructure Standards Validation project www.permis.org
[8] VOMS Architecture, European Datagrid Authorization Working group, 5 September 2002.
[9] Steven Newhouse, Virtual Organisation Management, The London E-Science centre, http://www.lesc.ic.ac.uk/projects/oscar-g.html
[10] BioMedical Research Informatics Delivered by Grid Enabled Services project (BRIDGES), www.nesc.ac.uk/hub/projects/bridges
[11] Dynamic Virtual Organisations in e-Science Education project (DyVOSE), www.nesc.ac.uk/hub/projects/dyvose
[12] Globus, http://www.globus.org
[13] R. Housley, T. Polk, Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructures, Wiley Computer Publishing, 2001.
[14] ITU-T Recommendation X.509 (2001) | ISO/IEC 9594-8: 2001, Information technology – Open Systems Interconnection – Public-Key and Attribute Certificate Frameworks.
[15] JISC Authentication, Authorisation and Accounting (AAA) Programme Technologies for Information Environment Security (TIES), http://www.edina.ac.uk/projects/ties/ties_23-9.pdf.
[16] Whitten, A., and Tygar, J. D. Why Johnny can't encrypt: a *u*sability evaluation of PGP 5.0. Paper presented at the 9th USENIX security symposium, Washington, 1999.
[17] D. Chadwick, O. Otenko, A Comparison of the Akenti and PERMIS Authorization Infrastructures, in Ensuring Security in IT Infrastructures, Proceedings of ITI First International Conference on Information and Communications Technology (ICICT 2003) Cairo University, Ed. Mahmoud T El-Hadidi, p5-26, 2003
[18] Conceptual AuthZ Framework and Classification (DOC) https://forge.gridforum.org/docman2/ViewCategory.php?group_id=55&category_id=458
[19] A.J. Stell, Grid Security: An Evaluation of Authorisation Infrastructures for Grid Computing, MSc Dissertation, University of Glasgow, 2004.
[20] ITU-T Rec. X.509 (2000) | ISO/IEC 9594-8. The Directory: Authentication Framework.
[21] UK e-Science Certification Authority, www.grid-support.ac.uk
[22] ITU-T Rec X.812 (1995) | ISO/IEC 10181-3:1996, Security Frameworks for open systems: Access control framework
[23] V. Welch, F. Siebenlist, D. Chadwick, S. Meder, L. Pearlman, Use of SAML for OGSA Authorization, June 2004, https://forge.gridforum.org/projects/ogsa-authz
[24] OASIS. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1,. 2 September 2003, http://www.oasis-open.org/committees/security/
[25] Cardiovascular Functional Genomics project, http://www.brc.dcs.gla.ac.uk/projects/cfg/
[26] Open Grid Service Architecture – Data Access and Integration project (OGSA-DAI), www.ogsadai.org.uk
[27] IBM Information Integrator, www.ibm.com
[28] EMBL-EBI European Bioinformatics Institute, http://www.ebi.ac.uk/ensembl/
[29] OpenSSL to create certificates, http://www.flatmtn.com/computer/Linux-SSLCertificates.html
[30] Von Welch/Jennifer Schopf personal communications.
[31] J. Jokl, J. Basney and M. Humphrey, Experiences using Bridge CAs for Grids, Proceedings of UK Workshop on Grid Security Practice - Oxford, July 2004
[32] Virtual Organisations for Trials and Epidemiological Studies project (VOTES), www.nesc.ac.uk/hub/projects/votes
[33] Shibboleth, http://shibboleth.internet2.edu/