



Grid Security: Practices, Middleware and Outlook

Professor Richard Sinnott
National e-Science Centre
University of Glasgow
e-Science Hub
Kelvin Building
Glasgow
G12 8QQ
United Kingdom

Tel: 0141 330 8606
Fax: 0141 330 8625
Email: ros@dcs.gla.ac.uk

Document Control

Details

Document Title	Grid Security: Practices, Middleware and Outlook
Funded By	Joint Information Systems Committee (JISC)
Circulation	JISC Committee for Scientific Research (JCSR)
Author	Richard Sinnott
Version	1.1
Date	2 nd December 2005

Authorised By	Joe Hutcheon (JISC), Tony Hey (JCSR)
---------------	--------------------------------------

Document History

Version	Editor	Date	Description
1.0	Richard Sinnott	21 November 2005	Version 1
1.1	Richard Sinnott	2 December 2005	Version 1.1

This work has been funded by the Joint Infrastructure Systems Committee (JISC) October 2005.

Contents

1	Introduction.....	7
2	Security Practices Today.....	9
2.1	Public Key Infrastructures (PKI)	9
2.1.1	Problems with PKIs	11
3	Security Practices Tomorrow.....	13
3.1	Risk Assessment	13
3.2	Exploitation of Server Certificates and Core Services	15
3.3	Advanced Authorisation Infrastructures	17
3.3.1	GGF SAML AuthZ API.....	18
3.3.2	Privilege and Role Management Infrastructure Standards Validation (PERMIS)	19
3.3.3	Globus Security Infrastructure (GSI).....	20
3.3.4	Community Authorisation Service (CAS).....	22
3.3.5	Virtual Organization Membership Service (VOMS).....	22
3.3.6	Process Based Access Control (PBAC).....	22
3.4	The Shibboleth Dimension on Grid Security	23
3.4.1	Introduction to Shibboleth	23
3.4.2	Impact of Shibboleth in the Context of the Grid.....	24
3.4.2.1	Trusting IdPs for Authentication	25
3.4.2.2	Shibboleth Attributes for Grids.....	26
3.4.2.3	Shibboleth and Grid Implementations	28
3.5	Grid Security and Fabric Management	31
3.6	Web Service Security Standards.....	33
3.6.1	WS-Security	34
3.6.2	WS-Policy	35
3.6.3	WS-Trust.....	36
3.6.4	WS-Privacy	36
3.6.5	WS-SecureConversation	36
3.6.6	WS-Federation	37
3.6.7	WS-Authorization	37
3.6.8	Security Assertion Markup Language (SAML).....	38
3.6.9	Liberty Alliance	39
3.6.10	Extensible Access Control Markup Language (XACML).....	40
4	Conclusions and Recommendations	40
5	References.....	43

Table of Figures

Fig 1.	Scenario for Obtaining a Grid Certificate.....	10
Fig 2.	Usage of Server Certificates for Job Submission onto the Grid.....	16
Fig 3.	X.812 Access Control Framework.....	18
Fig 4.	Global Grid Forum SAML AuthZ API.....	18
Fig 5.	GridShib Integration of Grid and Shibboleth.....	29
Fig 6.	Web Service Security Standards.....	34

Executive Summary

Security underpins Grids and e-Research. Without a robust, reliable and simple Grid security infrastructure combined with commonly accepted security practices, large portions of the research community and wider industry will not engage. 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), and as such this model of security does represent a weakness and potential threat to existing security infrastructures. Understanding these issues and trying to balance them with good practice needs to be recorded and documented for the wider community.

Authorisation infrastructures offer solutions to this security limitation, however there are many authorisation infrastructures available today but little consensus on which is most suited for which purpose. We provide an overview of some of the most prominent of these infrastructures in this report, and our experiences in applying them at the National e-Science Centre (NeSC) at the University of Glasgow.

Underlying a large part of the problem in developing secure Grid infrastructures is lack of standards, or rather lack of consensus and implementations of standards. With the move of the Grid community towards web service based solutions and service oriented architectures, a multitude of security standards have been proposed and various roadmaps for delivery and implementation given. We outline the reality of web service security standards and their implementations today based on currently agreed roadmaps. Past experience has shown however that such roadmaps provide only a general guideline as opposed to concrete plans for development, release and implementation.

In addition to this, the UK academic community is currently in the process of deploying the Internet2 developed Shibboleth technologies to support local (existing) methods of authentication for remote login to resources. Through the Shibboleth model, sites are expected to trust local security infrastructures for example in establishing the identity of users (authentication) and their associated privileges (authorisation). We discuss the likely impact this model will have in the context of the Grid, and outline potential solutions for harmonising the Shibboleth and Grid worlds.

Whatever solutions are put forward for Grid security, they have to be tailored to the end user research community needs. It can be stated that the existing Grid community have until now largely been IT-centric. Solutions for the less IT-focused and more research oriented community are required. Why should a biologist require training on Grid infrastructures or security middleware when all they require is access to a BLAST service across HPC resources such as the National Grid Service (NGS)? We outline solutions at Glasgow that have been put forward which meet these requirements.

In summary, this document serves several purposes:

- to give a snapshot of current practices in Grid security and the potential issues that arise from them in terms of acceptance by the wider e-Research and infrastructure (system) support community;
- to give an overview and assessment of the various Grid security infrastructures and middleware solutions that are available to the Grid community today;

- to give an outline of developments related to the Internet2 Shibboleth technologies and the impact they will have on the way Grid based research is currently undertaken across the UK academic arena;
- provide an overview, assessment and categorisation of web service standards and initiatives in the Grid security area and their potential impact;
- to provide a basic set of recommendations that may be followed to ensure that Grid facilities are made as secure as possible together with recommendations on what should be done in case of security breaches.

Throughout this document we draw on examples highlighting real experiences taken from a selection of security focused e-Science projects at the NeSC at the University of Glasgow. These are representative examples and do not attempt to address all issues associated with Grid Security.

It should be noted that the opinions expressed in this report are my own. It is also worth noting that the NeSC at Glasgow has an application-oriented focus and as such is not expressly a security middleware developer. Most of the Grid research and work at NeSC Glasgow is targeted at security, especially in the e-Health domain.

1 Introduction

Fundamentally Grids are about sharing resources. With this in mind, it is essential that security is ensured, both of the underlying systems, and of the Grid infrastructures and applications running on top of them. This is especially the case as the Grid community moves from the academic, research-oriented background to more commercial arenas, and especially when one moves towards more security focused domains such as finance and e-Health. It is the case in computer security that the weakest link rule applies; this fact is magnified by Grid infrastructures due to their openness. Highly secure multi-million pound compute facilities can be compromised by inadequately secured remote laptops. Rigorous security procedures at one site can be made redundant through inadequate procedures at another collaborating site.

This problem is due in part to the lack of granularity in how security is currently considered. Grid security is still primarily based around Public Key Infrastructures (PKIs) which support validation of the identity of a given user requesting access to a given resource – so called *authentication*. There are several key limitations with authentication based approaches to security. 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. For example, users can in principle run arbitrary applications, starting a variety of local processes. In reality, a set of existing applications and infrastructure are often pre-deployed across the Grid nodes, hence the issue and risks of uploading executables is diminished. However, given the fact that common compilers for C++ etc are commonly available on these resources, the possibility to compile arbitrary code and run executables spawning arbitrary processes exists. There is typically no security middleware enforcement on what processes can be started, by whom and in what context, other than the local enforcement given by the privilege associated with the local account. As the Grid community moves towards more security focused domains such as e-Health, such a model will never be supported. Thus it is unlikely ever to be the case that the UK National Health Service will allow access to one of their servers behind their firewalls to run arbitrary code. We provide an overview of PKIs in chapter 2 and the more general processes that are necessary for a PKI to function. The issues are problems with PKIs that deter large sets of the research community from engaging with Grid based e-Research are also outlined.

In chapter 3 we focus on what a future Grid security framework should include and associated practices that may/should be followed. The terms *practices* is important here since security is a large area and provides numerous challenges especially with the open nature of Grids. Security is absolutely **not** just a technology issue. To paraphrase Bruce Schneier: “...if you think that technology can solve your security problems then you don't know enough about the technology, and worse you don't know what your problems are...”¹.

Risk analysis is one way in which potential security threats can be quantitatively and qualitatively assessed along with the likelihood of their occurrence and plans to limit their potential damage or minimise their chance of occurrence. We provide an outline of a risk assessment exercise that was undertaken at the NeSC in section 3.1. With the open and collaborative nature of Grids, such risk assessments exercises, should in principle, be performed by all collaborators. There are no mandatory risk

¹ Bruce Schneier, Secrets and Lies in a Digital Networked World.

assessment procedures and practices that have been put forward across UK e-Science right now. Rather, such activities and recommendations have largely been in what can best be described as informal or ad-hoc. That said, we do note that incident response processes have been set up at the Grid Operations Support Centre (www.grid-support.ac.uk), but these are primarily for post-security breaches.

A fundamental property of any future Grid security infrastructure is that it has to be simple – at least from the end user researcher perspective. We provide an overview of solutions from projects at NeSC in section 3.2 where we have engineered solutions which minimise the learning curve associated with Grid technology and undertaking Grid based research.

The security framework for the future Grid security infrastructure of tomorrow needs to meet numerous requirements. It needs to take on board advances in security solutions such as advanced authorisation infrastructures which allow defining and enforcing “what” end users are allowed to do or not do on resources thereby providing finer grained models of security. We provide a snapshot of some of the most prominent security infrastructures in section 3.3.

This Grid security framework needs to provide solutions that are harmonised with other security mechanisms being deployed by the wider academic community, e.g. the Internet2 Shibboleth software which is currently being rolled out across UK academia. Understanding and adapting to the paradigm shift associated with Shibboleth-based security needs to be defined in the context of the Grid. We outline some of the issues and proposed solutions to incorporating Shibboleth and Grid technologies in section 3.4.

It is also still the case that the coupling of Grid security technology and the more general issue of managing the security of the underlying fabric has not been resolved. It is still the case for example that a single site can jeopardise collaborating sites if it does not take all appropriate measures to ensure its own security. Ensuring all Grid nodes have the necessary OS and/or Grid middleware patches and the most up to date antivirus software protection is something that up until now has been left to the individual sites across a *virtual organisation* (VO). This is something that cannot be left to chance however. It needs to be managed at the VO level. Grid based systems dealing with medical records or other highly sensitive data sets demand that resources are protected as far as possible. It is the case that the middleware solutions up to now have been primarily targeted at isolated aspects of Grid security, e.g. advanced authorisation, or given e-Research projects have focused on specific aspects of security and tools of relevance to their needs. We explore the issues and outline potential solutions to addressing this integrated Grid security in section 3.5.

With the move to service oriented architectures and web services within the Grid community, much of the implementation of the future Grid security infrastructure will be driven by web service security standards. In section 3.5 we look at existing web service security standards and outline their content, status (including whether they have been implemented yet or are still being discussed) and the likely impact they will have.

Finally in chapter 4 we provide various recommendations on Grid security that the JISC, JCSR and wider Grid community should consider and potentially adopt.

2 Security Practices Today

Most Grid solutions today are based upon X.509 certificates to support public key infrastructures. We provide a brief overview of these now. More information on PKIs is available through the JISC funded TIES project [TIES]. The JISC funded ESP-Grid project [ESP] is also exploring the PKI technology and the extent to which it meets the needs of the Grid community.

2.1 Public Key Infrastructures (PKI)

Cryptography is one of the main tools available to support secure infrastructures. Using cryptographic technology, confidentiality can be established by encrypting and decrypting messages and their contents. Encryption and decryption are done using keys. When these keys are the same, this is called symmetric-key cryptography.

Public-key cryptography uses different keys: private and public keys. Messages encrypted with a public key can only be read by an individual who possesses the private key. Any user can direct a message to a known destination, knowing that it can't be read by anyone else, simply by encrypting it using the public key of that destination. The owner of the private key can encrypt messages with that key, and the receiver of the message can be sure that it was sent by the owner of the private key. Both public key agreement and public key transport need to know who the remote public key belongs to, i.e. who has associated private key. The public key certificate is the mechanism used for connecting the public key to the user with the corresponding private key. Public key certificates include a Distinguished Name (DN) which can be used for identifying a given user.

A PKI is responsible for deciding policy, managing, and enforcing certificate validity checks. The central component of a PKI is a Certificate Authority (CA). A CA is a root of trust which holders of public and private keys agree upon. CAs have numerous responsibilities including issuing of certificates, often requiring delegation to a local Registration Authority (RA) used to prove the identity of users requesting certificates. CAs are also required amongst other things to revoke older or compromised certificates through issuing Certificate Revocation Lists (CRL). A CA must have well documented processes and practices which must be followed to ensure identity management.

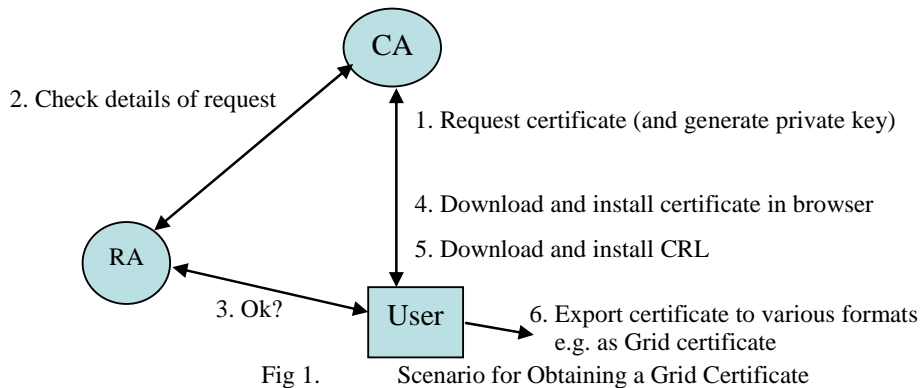
Various PKI architectures are possible and the selection of which depends upon numerous factors. Whether numerous CAs are to be trusted? How important to be able to add new CAs? What kind of trust relationships exist between CAs?

The simplest PKI involves a single CA which is trusted by all users. With this model, users only accept certificates and certificate revocation lists issued by this CA. This model makes certificate path analysis easy since there is a single step from a certificate to the CA who issued it. One danger of this PKI infrastructure is that the CA is single point of failure. Thus if it is compromised, then potentially all certificates that have been issued are compromised, requiring all users to be contacted and certificates revoked. The ramifications of such a compromise would be catastrophic with potentially all resources that had been accessed using certificates issued by this CA having to be completely reinstalled (in case backdoor software solutions had been installed). Perhaps more of an issue would be the level of trust and how Grids using PKIs were perceived by the wider community.

Other more complex PKI architectures also exist. For example, users may keep a list of trusted CAs. However, issues such as how to tell trustworthy one from

untrustworthy one arise? Hierarchical PKIs where there are chains of trust between the CA, sub-ordinate CAs and users may also exist. This model allows limiting the damage caused by a compromised subordinate CAs. Thus if a subordinate CA is compromised then only the certificates issued by them (or their subordinate CAs) need to be revoked. Other more complex architectures exist again, such as meshes of PKIs where trust relationships (webs of trust) are established on a peer-peer basis. This model often requires bridging solutions [PH,JBH] between CAs and results in certificate paths that are harder to establish – potentially containing loops.

The PKI architecture chosen for UK e-Science is based on a statically defined centralised CA with direct single hierarchy to users. The typical scenario for getting a certificate is depicted in Fig 1.



Researchers wishing to gain access to Grid resources such as the NGS (www.ngs.ac.uk) in the first instance have to acquire a UK e-Science X.509 certificate issued by the centralised Certification Authority (CA) at Rutherford Appleton Laboratory (RAL) (www.grid-support.ac.uk/ca). They will thus apply for a certificate via the Grid Support web site (www.grid-support.ac.uk). The CA will then contact their local Registration Authority (RA) who will in turn contact the user and request some form of photographic identification (such as a passport photo or university card). Once the identity of the user has been ratified, the RA contacts the CA who subsequently informs the user (via email) that their certificate is available for download. The user downloads the certificate and associated certificate revocation lists into their browser. Once in their browser they are required to export it to forms appropriate to the Grid middleware.

We note also that certificates can also be acquired for **both** users and servers/machines. Server certificates combined with core Grid services can allow for simpler security solutions to exist for the wider research community. We explore one implementation of such a simplified model of security in section 3.2.

The main benefit and reason for the widespread acceptance of PKIs within the Grid community is their support for single-sign on. Thus since all Grid sites in the UK trust the central CA at RAL, a user in possession of an X.509 certificate issued by RAL can send jobs to all sites, or rather to all sites where a user has requested and been granted access to those sites. Typically with Globus based solutions *gatekeepers* are used to ensure that signed Grid requests are valid, i.e. from known collaborators. When this is so, i.e. the DN of the requestor is in a locally stored and managed *grid-mapfile*, then the user is typically given access to the locally set up account as defined in the *grid-mapfile*.

2.1.1 Problems with PKIs

As stated, researchers wishing to gain access to Grid resources such as the NGS in the first instance have to acquire a UK e-Science X.509 certificate issued by the centralised CA at RAL. This process itself is off-putting for many of the wider less-IT focused research community since it required them to convert the certificate to appropriate formats understandable by Grid (Globus) middleware, e.g. through running commands such as:

```
$> openssl pkcs12 -in cert.p12 -clcerts -nokeys -out usercert.pem
```

Such requirements are likely to dissuade less IT-savvy researchers from engaging – especially as openSSL is not commonly available on platforms such as Windows. We note that the Certification Authority now suggests for researchers with Windows based PCs that they can use a Windows openSSL based solution (<http://www.openssl.org/related/binaries.html>) but this in turn requires them to install and configure additional software etc. In some circumstances this is not possible, for example if they do not have sufficient privileges on their PC (root access etc) – a not uncommon practice in certain departments and faculties at Glasgow University for example. In this case the researchers will instead have to refer to a local system administrator to help with the installation and configuration.

Assuming researchers have managed to obtain a certificate which they have converted into the appropriate format, they are then expected to remember strong 16-character passwords for their private keys with the recommendation to use upper and lower case alphanumeric characters. The temptation to write down such passwords is apparent and an immediate and obvious potential security weakness. Alternative, judgement based personalised authentication schemes drawing upon user knowledge bases are one way in which such issues can be resolved but as yet still largely a research area [LRA,KR].

This process as a whole does not lend itself to the wider research community which the e-Science and Grid community needs to reach out to and engage with. It is a well known adage that the customer is always right. Usability and addressing researcher requirements is crucial to the uptake and success of Grid technology. End user scientists require software which simplifies their daily research and not make this more complex. Given the fact that the initial user experience of the Grid currently begins with application for UK e-Science certificates, this needs to be made as simple as possible, or potentially removed completely. Scenarios where local IT staff can apply for batches of certificates for local users and preinstall and configure their environments for immediate usage on the Grid are one possibility. As part of the Grid Computing module taught at Glasgow University, local certificates were issued to students and their environments configured accordingly. Alternatively, solutions which do not require any user certificates represent another possibility. We explore different models and systems implemented at Glasgow in sections 3.2 and 3.4.

There are other issues with PKIs and Grid certificates as currently applied in the UK community. Thus for example *grid-mapfiles* are currently manually updated and managed based upon individual user requests. Solutions such as VOMS (discussed in section 3.3.5) offer capabilities for dynamically updating *grid-mapfiles* across multiple Grid resources. The dynamicity of this manual approach is also not conducive to the Grid-idea for establishing new short term VOs. Instead users have to statically have their DNs registered at collaborating sites which have previously made available/allocated local accounts.

Another issue with this approach is the human intensive nature of authentication utilising a centralised CA. Once the Grid scales to many hundreds of thousand or millions of users (there are currently over 3 million Athens accounts across UK academia from over 2,000 organisations to put this into context there are approximately 1800 UK e-Science certificates that have been issued right now) this centralised model of certification is likely to have scalability issues when the Grid is rolled out to the wider community, e.g. to industry and larger groups such as students taking Grid/e-Science courses.

The fundamental issue with PKIs however, is trust. Sites trust their users, CAs and other sites. If the trust between any of these is broken, then the impact can be severe, especially since users are currently free to compile and run arbitrary code. With the now global PKI and associated recognition of international CAs through efforts such as the International Global Trust Federation (www.gridpma.com), this basic trust model is naïve. Practices and solutions which help make Grid infrastructures safer are thus required.

3 Security Practices Tomorrow

In this chapter we outline various mechanisms that can be employed to make Grids more secure.

3.1 Risk Assessment

With the open nature of the Grid, there is always the potential knock-on effect when a site is compromised to collaborators (and collaborators of collaborators) as well as the risk to immediate projects that a given site might be involved in. Risk analysis can be used to better understand, protect and prepare sites for potential security breaches.

A risk analysis will normally involve several stages:

- identify all information and resources that needs to be protected;
- identify all sources of risk;
- determine the probability of occurrence of each risk item on each protected item;
- quantitatively and qualitatively assess the likely impact on the sites' business of the occurrence of each risk item on each protected item;
- identify actions that can mitigate the effects of each risk item;
- quantify the cost of implementing mitigating actions.

Once all of these stages have been documented, informed decisions about which mitigating actions to implement for each protected item can be made. A risk assessment at the NeSC was undertaken in 2003. There were many recommendations related to physical security and general working practices to prevent potential theft of equipment for example which have subsequently been implemented. The key risk it was deemed was from the wider *internet and Grid* communities. Both internet and Grid were considered since the two cannot be treated separately. Highly secure Grid middleware solutions can easily be made redundant from poorly configured firewalls, web services or general practices. The key conclusions to this risk assessment were:

1. It is absolutely paramount that Grid resources and private keys should not be compromised. Compromise of private keys can realistically only occur if the encrypted private key files are copied and passwords are stolen.
2. Password cracking software should be used to test the strength of existing passwords, e.g. Grid user credential passwords or passwords used for ssh (if this has not been disabled). Users should be advised immediately to change their passwords if necessary.
3. Backup copies of "strong" passwords should be stored ideally in a fireproof safe, not in the same physical location as the encrypted private keys. Passwords for user keys should NOT be written down and absolutely NEVER written down next to/near to the machines where the user keys are kept.
4. User keys should NOT be kept on laptops which are used elsewhere, i.e. outside of NeSC. If this is needed, e.g. to demonstrate software at a conference, then a copy of the keys should be kept separately from the laptop, e.g. on an external data storage device.
5. All machines of NeSC should be regularly updated with the newest virus detection programs and latest patches for software fixes. This should ideally be done on an automated basis, e.g. when users log in. NeSC employees should also ensure that

- they run firewall software on their laptops when used outside of the university.
6. NeSC should follow guidelines put forward by groups such as the UK e-Science Security Task Force (STF) on how to set up firewalls to run Grid software. Restricting the services that are available on the servers is also essential (e.g. disable Telnet, Rlogin, FTP etc.) and only activate those ports deemed absolutely essential for Grid. This included:
 - Restricting the ports that are available, e.g. only those needed for Grid software/services. Specifically this requires [SB,BOS]: Gatekeeper 2119/tcp; MDS Grid Resource Information Service (GRIS) 2135/tcp/2135udp; MDS Grid Information Index Service (GIIS) (site-selected); MDS Grid Information Index Service (GIIS) 2135/tcp/2135/udp; GridFTP 2811/tcp(control); GSI-Enabled SSH 22/tcp; MyProxy 7512/tcp. Based on the available port ranges and to avoid conflict with official port designations, it is suggested in [RAH] that values for the port range be selected from 65000-65256.
 - Firewalls should allow for packet filtering based upon whatever criterion is deemed appropriate. This will be at a minimum filtering based upon source and destination IP addresses, but also preferably in conjunction with some form of content filtering.
 - By default Globus Toolkit operations do not restrict the use of ports for communication to a specific range. With the exception of the Globus Toolkit gatekeeper and gridftp, all other communications utilise a dynamic selection of ports. In operation the ports used by the Globus Toolkit can be restricted through the GLOBUS_TCP_PORT_RANGE(min,max) environment variable which will result in listeners being created with ports in that specified range.
 7. Networks should be configured to minimise the problems that might occur if a given machine is compromised, e.g. avoid having Grid machines and other machines on the same direct network. Rather sub-networks should be set up with internal firewalls to both protect Grid related machines from other non-Grid machines and vice versa. This is a typical “Norman Castle” approach to security [MS].
 8. NeSC should ensure that it has taken sufficient measures to avoid possible threats from physical attacks/theft. The Kelvin Building where NeSC at Glasgow is based has CCTV and janitors. The primary compute clusters at Glasgow are in windowless rooms inside the building. This room has card swipe access/key pad protection on the door.

There were numerous other conclusions to this risk assessment which we do not describe here (since they are specific to NeSC) and not likely to be widely applicable.

The primary purpose of outlining in great detail the security risk assessment that was undertaken at NeSC, was that ideally all sites would conduct similar analyses. Needless to say, this assessment has since been superseded by evolutions in the Grid middleware. Thus for example the above analysis was undertaken when the UK Level2 Grid was being operated (which was built using Globus toolkit version 2 (GT2) [GT2], hence the GRIS, GIIS, MDS references). It is currently less clear what security policies should be followed given the move to the service oriented architecture solutions represented by later releases of the Globus toolkit (GT3+, GT4+) [GT4]; the Enabling Grids for E-Science (EGEE) [EGEE] gLite middleware [gLite] and the Open Middleware Infrastructure Initiative (OMII) [OMII] Grid

middleware solutions, amongst several others such as CROWN [CROWN], Condor [CONDOR], Unicore [UNICORE]. Instead, the focus has *to a certain extent* moved to draw on web service standards which we explore in section 3.6.

Irrespective of the technological solutions however, risk analysis is an important activity which should be undertaken by all sites to better understand their own internal security practices. This is especially the case for affiliate and full partner nodes of the NGS for example.

A body needs to be established that can check and validate that sites meet all appropriate security requirements. This body needs to be defined but might well include members of the STF, GOSC or some cross section of these.

One of the major issues with PKIs as implemented right now and the strong need for risk analyses is that researchers are in principle allowed to run arbitrary code. It is a fact however that in many cases, numerous researchers require access to the same kind of services and do not need to “tinker” with codes on Grid resources. In this situation, alternative and simpler security models can be supported.

3.2 Exploitation of Server Certificates and Core Services

The BRIDGES project (Biomedical Research Informatics Delivered by Grid Enabled Services (www.nesc.ac.uk/hub/projects/bridges) is a core project of the UK’s e-Science Programme aimed at developing Grid-enabled bioinformatics tools to support biomedical research. Its primary source of use cases is the Cardiovascular Functional Genomics Project (CFG) (www.brc.dcs.gla.ac.uk/projects/cfg), a large collaborative study into the genetics of hypertension (high blood pressure).

BRIDGES aims to aid and accelerate such research by applying Grid-based technology. This includes data integration tools but also Grid support for compute intensive bioinformatics applications such as BLAST. Solutions have been developed which provide simplified access to and usage of range of large scale compute resources including *all* nodes of the NGS, ScotGrid (www.scotgrid.ac.uk), other HPC clusters at Glasgow University and a collection of Condor pools.

One of the project requirements was that user authentication should not cause any additional learning or usability overheads for the users. Biology end users range widely in computer literacy and therefore systems providing a single mechanism for users of all abilities should aim at the lowest level of literacy. It was therefore decided to remove digital certificates from the end user environment altogether and replace them with simple username and password authentication at a central project web portal (see Fig. 2). Authentication at Grid sites such as the NGS is instead being carried out by means of a host proxy generated from the Grid server’s host credentials. The host’s identity is then mapped locally to a project account in the local *grid-mapfile* on the remote Grid nodes. Thus, all jobs run under the project’s identity on the NGS resources, and the logging and monitoring of user activity has to be moved up one level into the domain of the BRIDGES support staff.

We note that whilst we have removed the need for UK e-Science X.509 certificates from the biological end users, we have not omitted security. Rather, we have defined and enforced a much finer grained security model. For example, once a user has logged in to the portal, they have access to the complete set of tools available on the project portal. The finer grain control of what back end resources associated with a tool are accessible for a given user is implemented through the Grid authorisation software PERMIS (described in section 3.3). The identity of the user submitting the job can be extracted from the portal context, and is passed on with the job request. The Grid server sends a lookup request to a dedicated PERMIS

authorisation server maintained by the project team, where secure attribute certificates are used to store information about the roles a user has.

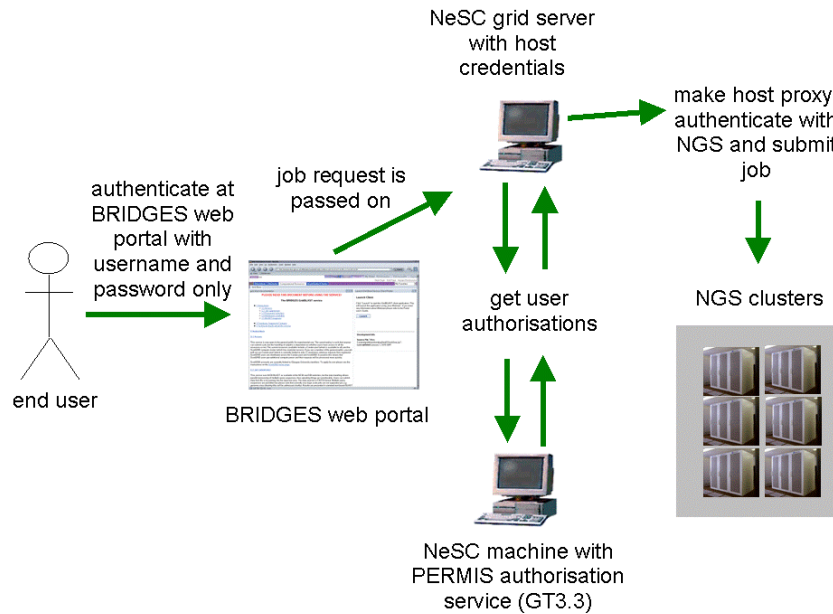


Fig 2. Usage of Server Certificates for Job Submission onto the Grid

Currently we support three security policies (which are enforced):

- If they are unknown users the job will only be submitted to the local Condor pool (we allow anyone access to the portal, however we restrict what they are allowed to do once there).
- If we recognise the users but they do not have a local ScotGrid account the job will be submitted to the Condor pool and NGS (we currently use all of the NGS nodes and are helping to define the generic datasets and services for the wider life science community on the NGS).
- If we recognise the users and they have an account on ScotGrid then the job will be submitted potentially to the Condor pool, the NGS and to ScotGrid (based on job numbers).

The selection of where to submit jobs is based on availability of resources (which is established dynamically).

This model of security through portals and server certificates is one way that increased security can be achieved. It does come with certain constraints however on the Grid application developers. We are required for example to keep a track of the users that are submitting jobs (logging of all activity through the portal is recorded and kept). The dangers that might otherwise arise with usage of server certificates for job submission by anonymous end users (from the point of view of the Grid resources the jobs are submitted to), are minimal however. Users that have successfully authenticated themselves at the portal via a username and password are given access to a fixed set of portlets such as the Grid BLAST service. Should a security breach occur and another masquerading user has managed to authenticate at the portal interface, the worst that can occur is that they will be allowed to run many BLAST jobs for example.

This solution is unlikely to be suitable for many Grid researchers who need to compile and tinker with their codes on the Grid resources. However there are *many* other researchers (not explicitly Grid-researchers) that require simple, secure access to large scale Grid infrastructures to run known services. Given the number of UK e-Science certificates that have been issued (approx. 1800), it is clear that simpler services tailored to the scientific community with minimal/no Grid learning or overheads are needed to engage with the much larger research communities. BLAST is one example of such a service. There are likely to be many other such solutions both within the life science as well as other research communities.

3.3 Advanced Authorisation Infrastructures

Authorisation is closely linked to authentication. Once a user has had their identity validated at a remote resource, it is essential that users actions are restricted based on who they are, what they are trying to do, and in what context etc. There are various methods of enforcing this restriction, the simplest method being the use of an Access Control List (ACL), which lists what users have access to a privilege. Essentially, a user presents their credentials at the gatekeeper to a resource, which consults a list of users. This basic authorisation structure extends the concept of authentication and no more. If the user cannot authenticate to the satisfaction of the gatekeeper then the resource request will be denied. A problem that arises when trying to apply this method to a dynamic Grid environment is that only one list exists, where there could be many privileges that require different ACLs. For example, a user might need access to a given resource for different purposes within a given VO. Having a single list with a predefined set of accounts and user DNs does not support this multi-role approach. This is a solution that would not scale well in a large VO. A more sophisticated method of applying authorisation controls is through use of Role-Based Access Control (RBAC) mechanisms, which allow Privilege Management Infrastructures (PMI).

The relationship between a PMI and authorisation is similar to the relationship between a PKI and authentication. Consequently, there are many similar concepts in the two types of infrastructure. Central to a PMI is the idea of the attribute certificate (AC), which maintains a binding between the user and their privilege attributes. It is similar in notion to the public key certificate in a PKI. The entity that signs a public key certificate is a CA; the entity that signs attribute certificates is called an Attribute Authority (AA). The root of trust of a PKI is often called the root CA, which can delegate this trust to a subordinate CA; the root of trust of a PMI is called the Source of Authority (SOA). The SOA may have subordinate authorities to which it can delegate powers of authorisation. Certificate Revocation Lists (CRLs), which show a list of certificates that should no longer be accepted as valid, exist in a PKI; Attribute Certificate Revocation Lists (ACRLs) exist in a PMI.

The critical idea in a PMI is that the access rights of a user are not held in an ACL but in the privilege attributes of the ACs that are issued to the users. This is the central idea behind RBAC – the privilege attribute will describe one or more of the user's rights and the target resource will then read a user's AC to see if they are allowed to perform the action being requested. This de-couples the user's privileges from their local identity and allows a more dynamic and flexible approach to access control.

The X.812 | ISO 10181-3 Access Control Framework standard [X812] defines a generic framework to support this type of authorisation, depicted in fig 3.

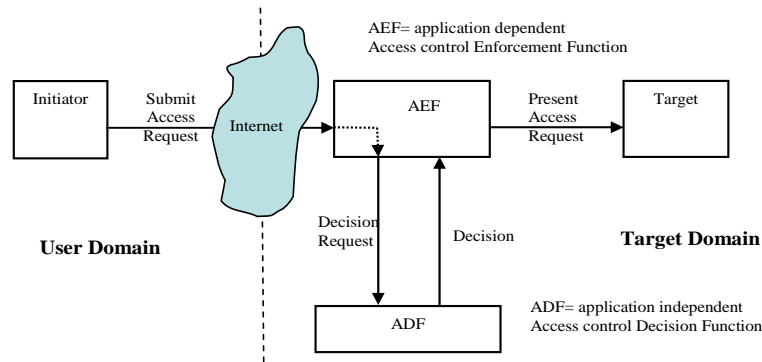


Fig 3. X.812 Access Control Framework

In this model, the initiator attempts to access a target in a remote domain. Two key components support authorised access to the target: a Policy Enforcement Point (PEP), described in the figure as the Access control Enforcement Point (AEF), and a Policy Decision Point (PDP), described as the Access control Decision Function (ADF). The PEP ensures that all requests to access the target are run through the PDP and the PDP casts the authorisation decision on the request based on a collection of rules (policies). To make this structure scalable and easily applicable within a Grid environment, a generic API to model the PEP has been proposed and created by the Authorisation Working Group of the Global Grid Forum (GGF) (www.ggf.org).

3.3.1 GGF SAML AuthZ API

The GGF have put forward an API that provides a generic PEP, which can be associated with an arbitrary authorisation infrastructure. The specification for Grid technologies is an enhanced profile of the OASIS [OASIS] Security Assertion Markup Language (SAML) v1.1 [SAML1-1].

The OASIS SAML AuthZ specification defines a message exchange between a PEP and PDP consisting of an *AuthorizationDecisionQuery* (which contains a *subject*, a *resource* and an *action*) going from PEP to PDP, and an assertion returned containing a number of *AuthorizationDecisionStatements*.

The GGF SAML AuthZ specification [WSCMP] defines a *SimpleAuthorizationDecisionStatement* (a boolean stating “granted/denied”) and an *ExtendedAuthorisationDecisionQuery* that allows the PEP to specify whether the simple or full authorisation decision is to be returned. Figure 4 shows the interactions supported by this API.

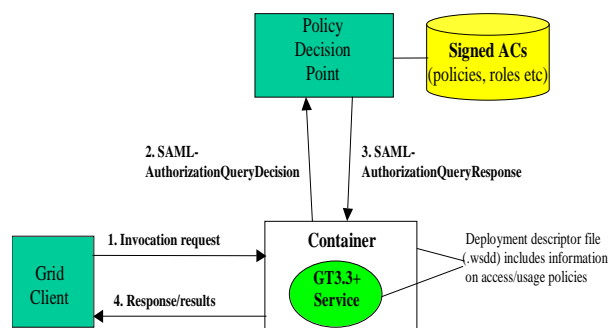


Fig 4. Global Grid Forum SAML AuthZ API

Through this SAML AuthZ API, a generic PEP can be achieved which can be associated with arbitrary 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. Releases of the Globus software since GT3.3 have supported this API.

At the time of writing and to the best of knowledge of the author, only two authorization infrastructures support this PDP: the Privilege and Role Management Infrastructure Standards Validation (PERMIS) initiative at the University of Kent lead by Prof. David Chadwick [CHAD] and the WSRF .net implementation at the University of Virginia lead by Dr Marty Humphrey (www.ws-rf.net).

We note that one issue that has been encountered with the SAML AuthZ profile in projects at NeSC in Glasgow is the lack of granularity in how users might invoke actions [SC]. 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, a query service cannot easily (at least in a manner that easily scales) be restricted to query those data sets in a given set of federated databases that are appropriate to the invoker. Instead, the SAML AuthZ specification supports either a secure query service or a non-secure query service. The GGF AuthZ working group is now working on a new version of this API (to support parameters). In addition, a recently funded JISC project will implement this API in PERMIS [AuthZ2].

3.3.2 Privilege and Role Management Infrastructure Standards Validation (PERMIS)

The Privilege and Role Management Infrastructure Standards Validation (PERMIS) project (www.openpermis.org) [COB,CO] 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 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; SOAs, e.g. local managers trusted to assign roles to subjects; roles and their hierarchical relationships; what roles can be assigned to which subjects by which SOAs; target resources, and the actions that can be applied to them; which roles are allowed to perform which actions on which targets, and 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) has been developed to support this process. Once roles are assigned, and policies 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

(www.openldap.org) to store the attribute certificates and reference the SOA root certificate. A local CA is required to be set up using OpenSSL [OpenSSL] – this designates the SOA and all user certificates created from this CA must have a DN 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.

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.

The PERMIS infrastructure offers very fine grained authorisation capabilities both in terms of policy expression and enforcement. The policy editing tools allow for easy development of the XML based policies. These tools have been developed with HCI considerations included, although we note that the advanced MSc students at the University of Glasgow raised issues with the tools, e.g. the XML that is generated is inconsistent with the tool user interface. For example, the XML has attributes for “subject domain”, whilst the tool has buttons for “where are users from”.

With support for the GGF SAML AuthZ api, PERMIS should in principle allow easy linkage between Grid services and authorisation infrastructures. It is still non-trivial linking an authorisation infrastructure and Globus based Grid service however. The NeSC at Glasgow have put users guides on how to set up PERMIS, Globus Grid services and link these together (see www.nesc.ac.uk/hub/projects/etf).

The PERMIS team have also included capabilities to link Shibboleth to the PERMIS authorisation infrastructure. The Shibboleth Apache Authorisation Module [XCO] allows for authorisation decisions on access to and usage of Apache based services to be made via PERMIS. Once again however, usage of this middleware is still a non-trivial activity and requires detailed configuration of the underlying software infrastructure.

Despite these difficulties, PERMIS is by far the most advanced authorisation infrastructure with software that meets the needs of the wider Grid and Shibboleth communities. Further detailed explorations of PERMIS and Shibboleth will be undertaken in a variety of other projects at NeSC Glasgow including the clinical trials domain (www.nesc.ac.uk/hub/projects/votes), wider e-Health domains linking genetics and healthcare (www.nesc.ac.uk/hub/projects/ghi) and the life sciences (microarray expression) domain (www.nesc.ac.uk/hub/projects/gemeps). As described, previously we have already shown how PERMIS allows for fine grained security models for access to and usage of the NGS (and numerous other resources) within the BRIDGES project. We have also implemented fine grained security solutions within BRIDGES utilising PERMIS where different genomic databases can be accessed depending upon the user role in the VO.

In short, PERMIS and the associated tool sets do allow for fine grained security to defined, enforced and seamlessly linked to Grid services.

3.3.3 Globus Security Infrastructure (GSI)

GSI [GSI] is an example of the classic Access Control List (ACL) used to enforce authorisation and provides a relatively coarse-grained approach to implementing

security. A list is compiled that maps each user's local account name to the DN that appears on their user certificates. When a user makes a method call on a service, this list is consulted and access is granted or denied depending on whether they appear on the list with the correct credentials. Rather than distinguishing between methods this restriction applies to that user for all secured services across the container.

To run the Globus container requires an administrative user (usually 'globus') to set up the container. Each user that wishes to run secure services within this container must have a user certificate located in their home directory. The machine upon which the container is running must also have a host certificate installed by 'root'. Once the container is running, any user should be able to run an unsecured service, with or without a certificate. However, using GSI, a measure of security can be introduced on the service that allows only those with the necessary credentials to run it, typically through a proxy certificate generated from their user certificate.

To use GSI, Grid clients must normally be in possession of a Grid (X.509) certificate which is used to encrypt the communication between client and Grid service. The Grid service is then able to check the identity of the user invoking the service against the local ACL (*grid-mapfile*) that an authorised client is invoking the service.

The latest release of the Globus toolkit [GT4] supports GSI-based authentication and authorization. This includes:

- WS Authentication with support for both message level and transport level security. Message level security is achieved through an implementation of the WS-Security standard that supports message protection at the Simple Object Access Protocol (SOAP) message level. Transport level security is achieved through use of X.509 certificates to establish Transport Layer Security (TLS) connections.
- WS Authorization through an authorisation framework (based upon the SAML AuthZ api defined in section 3.3.1) and use of the Community Authorization Service (CAS). We describe the CAS service in more detail in section 3.3.3.
- Credential Management through MyProxy (a credential storage and management system) and SimpleCA (which as its name implies provides a simple CA).

The MyProxy solution [MyProxy] in particular should be mentioned since this is gaining widespread acceptance as the way in which credentials should be managed within a Grid environment. Instead of users managing their own private keys and credentials, they can delegate them to a MyProxy repository. Through username and password access to MyProxy repositories, short lived proxy certificates can be created. MyProxy also allows for the creation of PKI credentials since later releases now include a CA.

MyProxy solutions are now being used in combination with portals for example, where users accessing a portal through a username and password will automatically have short lived proxy certificates created which can subsequently be used for Grid based job submission. This capability exists for example on the NGS (portal.ngs.ac.uk).

Of all of the authorisation infrastructures, GSI is arguably the most straightforward to establish and use. Unsurprising since GSI has been developed as an integral part of the Globus development. That said, the ACL based approach offered by *grid-mapfiles* is a limited form of authorisation however.

3.3.4 Community Authorisation Service (CAS)

CAS [CAS,CAS2] implements RBAC using an authorisation server. The central idea behind CAS is that while resource providers can specify a coarse-grained policy, the fine-grained policy decisions can be delegated to the administrator of the community that is served by CAS. Resource providers grant privileges to the community and establish a trust relationship with the representative of that community. That representative then uses CAS to manage the distribution of privileges within the community.

When a user wants to access resources served by CAS, the user issues a request to the CAS server (using their own X509 certificate). If the CAS server decides that the user associated with this certificate has sufficient privileges, then it will issue a proxy credential with an embedded policy giving the user the right to perform the requested actions (assuming that the user has sufficient privilege). The user then uses these CAS credentials to access the resource. The local resource then applies its own local policy to determine the amount of access granted. Currently the only resource that can be accessed through CAS credentials is gridFTP.

It is non-trivial to set up and use CAS (see www.nesc.ac.uk/hub/projects/etf). The centralised model of an authorisation server is also likely to have scalability issues when dynamic VOs are to be established or very large VOs. Given the fact that CAS can only be used right now for gridFTP, it is not immediately clear what the benefits of using this middleware are.

3.3.5 Virtual Organization Membership Service (VOMS)

VOMS [VOMS] is a system for managing authorisation data within VOs. VMS has been developed as part of the European DataGrid project (edg-wp2.web.cern.ch/edg-wp2) VOMS provides a database of user roles and capabilities and a set of tools for accessing and manipulating the database and using the database contents to generate Grid credentials for users when needed.

The VOMS database contains authorisation data that defines specific capabilities and general roles for specific users. A suite of administrative tools allow administrators to assign roles to users and manipulate capability information. A command-line tool allows users to generate a local proxy credential based on the contents of the VOMS database. This credential includes the basic authentication information that standard Grid proxy credentials contain, as well as role and capability information from the VOMS server.

One of the benefits of VOMS is that Grid applications can use the credential without using the VOMS data. Alternatively, VOMS-aware applications can use the VOMS data to make both authentication and authorisation decisions regarding user requests.

3.3.6 Process Based Access Control (PBAC)

PBAC [PBAC] is the authorisation system for the OMII system. With PBAC, the invocation of operations on a web service can be restricted depending upon the context. To support this, a repository of named authorisations exists. Each authorisation consists of a triple including the user, the operation and the conversation (which is an identifier for the context in which the operation is executed).

Through PBAC, history and context of authorisation can be supported. One of the restrictions with other authorisations solutions such as PERMIS that currently exist are that history unaware (or process more generally) security policy rules are defined

and enforced. The evaluation of such security policy rules either allow or deny access depending upon the rule (which typically does not change when an authorisation decision is enforced). As a concrete example, a doctor might have the privilege to issue prescriptions and PERMIS authorisation infrastructure will authorise these requests. However, if a doctor has issued a large number of prescriptions previously, then this information is ignored. PBAC solutions overcome such restrictions.

If an operation occurs that changes the state of the context, the authorisation database is updated accordingly.

PBAC itself utilise the WS-Security standards [WS-S] which are explored in more detail in section 3.6.

3.4 The Shibboleth Dimension on Grid Security

3.4.1 Introduction to Shibboleth

The UK academic community is currently in the process of deploying Shibboleth technologies (<http://shibboleth.internet2.edu/>) to support local (existing) methods of authentication for remote login to resources. Through this model, sites are expected to trust remote security infrastructures for example in establishing the identity of users (authentication) and their associated privileges (authorisation). To support this, the Shibboleth architecture [ShibA] and associated protocols [ShibP] identify several key components that should be supported including federations, Identity Providers, Service Providers² and optionally Where Are You From (WAYF) services. Through these components, end users will have single usernames and passwords from their home institutions (which they are more familiar with than PKIs!) which will provide for seamless access to a range of resources at collaborating institutions and service providers. Local security policies at service provider sites can then be used to restrict (authorise) what resources authenticated users are allowed access to.

The term “federation” has emerged in recent years to describe groups of organisations which agree to adopt common policies and technical standards to provide a common infrastructure for managing access to resources and services in a uniform way. Examples of Shibboleth-based federations are InCommon (<http://www.incommonfederation.org>), the federation formed by the Internet2 community in the United States, InQueue (<http://inqueue.internet2.edu/>) for sites wishing to test and explore the Shibboleth federated trust model, the SWITCHaai federation of the higher education system in Switzerland (<http://www.switch.ch/aai/>), the HAKA federation developed by the Finnish universities and polytechnics (<http://www.csc.fi/suomi/funet/middleware/english/>) with more in the pipeline such as the Meta Access Management System (MAMS) in Australia (<https://mams.melcoe.mq.edu.au/zope/mams/kb/shibboleth/>).

It was announced at the JISC Core Middleware Programme meeting on 15th November 2005 that the UK production federation will be based upon the SDSS federation based at the University of Edinburgh and managed by EDINA (www.sdss.ac.uk).

To understand the impact of Shibboleth technologies on Grid security it is first necessary to have an appreciation of the interactions that typically arise with Shibboleth. When a user attempts to access a Shibboleth protected service or Service Provider (SP) more generally, they are typically redirected to a WAYF server that asks the user to pick their home Identity Provider (IdP) from a list of known and

² In earlier versions of the Shibboleth documentation the service provider was referred to as the target.

trusted sites. The service provider site already has a pre-established trust relationship with each home site, and trusts the home site to authenticate its users properly.

After the user has picked their home site, their browser is redirected to their site's authentication server, e.g. an LDAP repository, and the user is invited to log in. After successful authentication, the home site redirects the user back to the SP and the message carries a digitally signed SAML authentication assertion message from the home site, asserting that the user has been successfully authenticated (or not!) by a particular means. The actual authentication mechanism used is specific to the IdP.

If the digital signature on the SAML authentication assertion is verified and the user has successfully authenticated themselves at their home site, then the SP has a trusted message providing it with a temporary pseudonym for the user (the handle), the location of the attribute authority at the IdP site and the service provider URL that the user was previously trying to access. The resource site then returns the handle to the IdP's attribute authority in a SAML attribute query message and is returned a signed SAML attribute assertion message. The Shibboleth trust model is that the target site trusts the IdP to manage each user's attributes correctly, in whatever way it wishes. So the returned SAML attribute assertion message, digitally signed by the origin, provides proof to the target that the authenticated user does have these attributes.

We note that later versions of the Shibboleth specification have introduced a performance improvement over the earlier versions, by allowing the initial digitally signed SAML message to contain the user's attributes as well as the authentication assertion. Thus the two stages of authentication and attribute retrieval can be combined.

We note that the connection from the IdP to the service provider can also be optionally protected by SSL in Shibboleth. Here SSL is used to provide confidentiality of the connection rather than message origin authentication. In many cases a confidential SSL connection between the IdP and SP will not be required, since the handle can be opaque/obscure enough to stop an intruder from finding anything out about the user, whilst the SAML signature makes the message exchange authentic. However the message exchange should be protected by SSL if confidentiality/privacy of the returned attributes is required. The attributes in this assertion may then be used to authorise the user to access particular areas of the resource site, without the service provider ever being told the user's identity. Shibboleth has two mechanisms to ensure user privacy. Firstly it allows a different pseudonym for the user's identity (the handle) to be returned each time, and secondly it requires that the attribute authorities provide some form of control over the release of user attributes to resource sites, which they term an attribute release policy. Both users and administrators should have some say over the contents of their attribute release policies.

3.4.2 Impact of Shibboleth in the Context of the Grid

Shibboleth offers numerous possibilities and potential advantages in the context of the Grid. Single sign-on via authentication at a home site and subsequent acceptance and recognition of the authentication and associated attributes released to remote sites is the most obvious advantage. Thus users need not remember X.509 certificate passwords but require only their own institutional usernames and passwords. Institutions can establish their own trust federations and agree and define their own policies on attribute release, and importantly SPs can decide upon what attributes and attribute values are needed for authorisation decisions.

The uptake and adoption of Shibboleth technologies within a Grid context is not without potential concerns however. Sites need to be sure that collaborating sites have adopted appropriate security policies for authentication. Strength of user passwords and unified account management is needed. We outline issues in deploying such an integrated account management system at the University of Glasgow in section 3.4.2.1.

Shibboleth is by its very nature much more static than the true vision of the Grid, where VOs can be dynamically established linking disparate computational and data resources at run time. Instead Shibboleth requires agreed sets of attributes that have been negotiated between sites. We explore proposed attributes in section 3.4.2.2 that may well map on to Grid. We also explore results from the JISC funded DyVOSE project where dynamic creation and recognition of attribute certificates is supported.

In section 3.4.3.3 we also outline proposals from the UK and US communities looking at integration of the Grid and Shibboleth technologies, and assess how they may (or may not) simplify Grid security. We also outline implementations at NeSC Glasgow have already shown how Shibboleth based access to Grid resources has been supported.

3.4.2.1 Trusting IdPs for Authentication

Ensuring that an institution in a Shibboleth federation can guarantee the authenticity of a user when accessing a remote resource is crucial to the overall principles upon which Shibboleth and Shibboleth federations are based. In short, institutions in a federation should trust one another. It is the case however, that users at larger institutions may well have numerous usernames and associated passwords that are used to access a variety of services. This is the case at the University of Glasgow for example! A unified institutional user account management system based which handles authentication and attributes is key.

Directory technology offers one solution to this. The directory is the part of any service which retains the authentication data, most commonly a username and a password. Until now this information has primarily been closely linked to specific operating systems or infrastructures. This has resulted in a myriad of solutions holding a variety of authentication information across the university. For example, within Windows NT this is the domain database; within UNIX it is commonly the NIS database, within Netware it is the eDirectory etc. As a result, it has until now been the case that members of large institutions and universities are in possession of multiple accounts for many systems that are needed to access the many services that are available. For example, to access their desktop at Glasgow University, users will typically need an NDS account for workstation management, an NT domain account for Exchange access and an MIS NIS account for SAMBA-based access to their filestore. In addition, many users may potentially get usernames and passwords for dial in services, for vpns, for departmental databases etc. One of the consequences of this is that the evolution of services can become tied to the platform which hosts the user identities, rather than the best platform for the job. In most cases these accounts are not necessarily the same - indeed in lots of cases they are very different, and often based on a combination of central and local accounts. Thus users are expected to keep multiple accounts and multiple passwords. Under these circumstances users tend to either leave the password at the value it was when they received it; change it to the same value as their other passwords; they have to remember multiple passwords, or they end up with passwords they can't change because changing it in one place means changing it everywhere. With multiple accounts, across multiple systems with

potentially multiple different administrators coordinating changes is almost impossible even within a single institution. Addressing such issues is crucial for the wide scale successful deployment and take-up of Shibboleth.

The above problems are not isolated. Until recently no mechanisms existed to keep the various user accounts synchronised across all of the systems used. This arrangement meant there was a high number of redundant accounts, which has meant that it was very difficult to ensure all access and privileges were removed in a timely fashion. In some circumstances users could retain rights to data and services long after they should. This was possible since different representations for the same users could in principle lead to situations where one account could be disabled, but users could retain access to services and data via a second account. A key challenge is therefore to address the whole user base since there may be no definitive source for authentication data, but rather a collection of sources.

To overcome these issues the University of Glasgow is moving to a system that offers a more consistent representation of staff and students across multiple systems that will allow: timely creation/modification and deletion of accounts; an audit trail against central records; a single authority for services covering the whole university; password synchronisation; and the implementation of a rigorous password policy.

To support this, the university is planning:

- a one to one representation between each user and their corresponding entry in the Human Resource/Registry database – the definitive sources for data;
- an agreed standard for unique identifiers for each user account;
- an agreed password policy;
- an agreed definition of department/faculty codes where user accounts should reside.

This system is currently being rolled out by Computer Services across the university. With this system, sites collaborating with Glasgow University can be assured that when Glasgow authenticates and releases attributes for a particular individual, then they are actual current members of the university, and not authenticated on some older and overlooked username and password. To make Shibboleth a success, all sites should ideally follow similar practices. Time will tell if this is the case.

3.4.2.2 Shibboleth Attributes for Grids

As well as authentication information, SPs are likely to need further information in order to allow (authorise) access to specific services. In the context of the Grid, membership of the University of Glasgow will not normally be sufficient information for a decision on access to a specific Grid service hosted and managed by a given VO. The eduPerson efforts [eduPerson] have identified a core set of attributes that may be of use within an academic environment. The JISC Blueprint for a Production Federation [RM] has also explored some potential attributes of relevance to the UK academic community.

A small core set of attributes is recommended for IdPs to support that SPs can subsequently use for authorisation decisions. It is essential that interoperability exists between attribute authorities issuing attribute assertions, policy writers defining access policies, and access decision functions that make decisions based on the initiator's attributes and sites target and resource policy. The overlap between Grid technologies (requiring in the first instance attributes for identification) and Shibboleth technologies is required.

The *eduPerson* attributes that have been recognised as providing the necessary core functionality for IdPs and SPs in the UK academic community include:

- *eduPersonScopedAffiliation*: which indicates the user's relationship (e.g., staff, student, etc.) with the institution.
- *eduPersonTargetedID*: is needed when an SP is presented with an anonymous assertion only, as provided by *eduPersonScopedAffiliation*. In this situation it cannot for example provide usage monitoring across sessions. The *eduPersonTargetedID* attribute provides a persistent user pseudonym.
- *eduPersonPrincipalName*: is used where a persistent user identifier, consistent across different services, is needed.
- *eduPersonEntitlement*: enables an institution to assert that a user satisfies an additional set of specific conditions that apply for access to a particular resource. A user may possess different values of the *eduPersonEntitlement* attribute relevant to different resources.

Each of these attributes can be used to provide the necessary information to SPs to make authorisation decisions. These attributes are versatile and likely to be sufficient for the great majority of applications.

Given the fact that Grids form VOs which themselves will have finer grained structuring, it seems sensible that the *eduPersonEntitlement* attribute can be used for this purpose. The *eduPersonEntitlement* attribute can utilise structured XML data representative of large scale Grid infrastructure users and IdPs. This might include the VO they are involved in, the roles that they might have in that VO etc.

It is important to note that these attributes are statically defined and agreed upon between the institutions prior to formulation of VOs or requests to access Grid resources, i.e. they are based upon statically defined PMIs. The JISC DyVOSE project has developed solutions which allow for the dynamic creation and acceptance of attributes. This is more aligned with the dynamic creation of VOs across Grid infrastructures where dynamic delegation of privilege is supported. As the complexity and number of security policies increases, the ability of a given SOA to delegate responsibility to others is necessary. Through extensions to the PERMIS software (www.permis.org), DyVOSE now supports dynamic delegation of authority whereby Grid sites can allow an attribute authority controlled by an external SOA to be delegated the ability to assign roles meaningful to a home SOA. Through this, a remote Grid user can hold a role based in the home institution that will allow access to the service provider Grid resources.

Perhaps the biggest challenge in moving from static PMI based approaches as exist with current Shibboleth solutions to supporting 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.

In static delegation, the roles at the remote institution would need to be hand written into the policy at the home institution. Dynamic delegation factors away the role assigning powers to subordinate authorities, which may delegate the ability to assign local roles to remote attribute authorities, and vice versa. Thus a 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 say 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 Delegation Issuing Service to support such dynamic creation and recognition of attribute certificates has been implemented and available for use (www.openpermis.org).

One of the key issues that have still to be resolved with attributes for the Grid community is related to the attribute release policy. At present an SP will request the attributes associated with the potentially opaque identifier (handle) that is returned from an IdP. If a user from the University of Glasgow is involved in numerous Grid projects and VOs however, and all of this information on what VOs this person is involved in, and what their role is in that VO etc are encoded in the core set of attributes, then it is difficult to restrict the information being released. Thus the *eduPersonEntitlement* attribute might encode much of the information on VO membership and roles etc. If an SP requests the attributes for a given user, and receives this *eduPersonEntitlement* attribute then they will receive more information than they might actually need to make an authorisation decision, e.g. if this SP was just one of the many VOs that the user was involved in, then this SP would know more about all VOs the user was involved in. Of course these attributes will be encoded, however, the SP will be able to decode the attributes due to the trust relationships and certificates previously put in place.

It is of course possible to have a richer array of attributes other than the core set of *eduPerson* attributes identified previously, but for interoperability and simplicity, having a core set is beneficial. Given that the focus of much of the Grid community as being represented by the NGS does not focus upon privacy or confidentiality, such issues are not immediately important. Once more security focused groups are involved however, attribute release policies will become more important and only those attributes absolutely needed, will be released.

Another potential solution to this situation is to have a proliferation of IdPs. Thus each individual Grid project might have their own IdP and be associated with different WAYF servers. This would allow for those sets of attributes to be released deemed necessary for particular SPs, however the more IdPs that exist requires more trust relationships to be put into place, thereby weakening the overall security.

Having multiple WAYF services and IdPs and SPs being involved in more than one trust federation also brings with it potential difficulties. Do we trust all federations equally? Do some treat authentication and identity management more stringently? If there are differences between the assurance levels, then multiple memberships will be problematic.

3.4.2.3 Shibboleth and Grid Implementations

There is much effort to reconcile the Shibboleth and Grid worlds. The GridShib project [GridShib] and the two recently funded JISC projects: ShibGrid and SHEBANGS are exploring use of Shibboleth and Grid. The GridShib project is focusing upon identity federation between the Grid and Shibboleth communities. In real terms the GridShib project is looking towards Grid (GSI) based authentication followed by Shibboleth based retrieval of attributes for making authorisation decisions. It is important to note that the GridShib project does not directly address Shibboleth single sign on to Grid infrastructures.

The basic scenarios through which Grid and Shibboleth technologies are being integrated in GridShib are outlined in Fig 5.

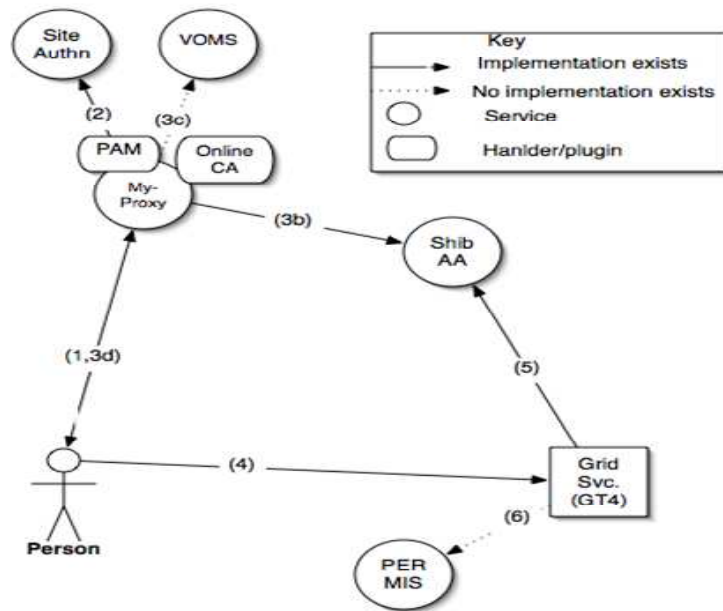


Fig 5. GridShib Integration of Grid and Shibboleth

Here the user contacts MyProxy and provides authentication information. MyProxy then verifies the authentication information using the site authentication system. MyProxy creates an X509 certificate for the user. This has the following sub-steps: a SAML Subject element is created and embedded in the certificate, providing details of how the user was authenticated and providing a subject which can be used to query a SAML Attribute Authority; a DN is created for the user, either algorithmically, or through querying a local Shibboleth AA for an attribute defined for this purpose; MyProxy can be configured to contact a VOMS server and inserts VOMS-generated attribute certificates into the returned certificate. The certificate is signed and returned to the user.

When the user makes a request of a Grid Service, authenticating with X509 credentials, the Grid service contacts the Shibboleth attribute authority using the Subject element from user certificate, and retrieves the user attributes. The Grid service then presents these attributes to the PDP (in this case PERMIS) which makes the authorisation decision.

The ShibGrid and SHEBANGS projects are both looking at supporting scenarios where Shibboleth is used for single sign-on and access to the NGS, both with MyProxy at the core. In addition to these projects, the DyVOSE and ESP-Grid projects at NeSC Glasgow have already put together working implementations demonstrating how Shibboleth can be used for access to and usage of Grid resources. We outline these briefly here.

3.4.2.3.1 Shibboleth Access to Grid Resources in DyVOSE

The DyVOSE project was funded as one of the JISC Core Middleware projects focusing on advanced security infrastructures in the education domain. The basic model being explored in DyVOSE of sites having their own security authorisation policies and associated attributes is very much consistent with Shibboleth where there will likely be several authorities that assert attributes for users. Various domains will then write their own authorization policies based on such attributes.

In teaching the Grid Computing module as part of the advanced MSc in Computing Science at the University of Glasgow a thorough exploration was made of the PERMIS authorisation software for forming static PMIs in a Grid context. In detail, students were initially expected to develop their own security policies for a basic GT3.3 based Grid service which was subsequently used in their main programming assignment.

This assignment required that the students were requested to create a policy for a GT3.3 service (*searchSortGridService*) which wrapped a Condor based Java application (this service offered two methods to search (*searchMethod*) and sort (*sortMethod*) a large (5MB) text file (the complete works of Shakespeare). The students themselves were split into groups (*studentteam1*, *studentteam2*) with the authorisation policy to ensure that method *sortMethod* could only be invoked by members of their student group and the lecturing staff, whilst method *searchMethod* could be invoked by everyone. This set-up was used to illustrate the use of RBAC, where users are allocated privileges based on what role they have been assigned rather than their local user credentials. The students were also requested to secure their service using Globus GSI and also with PERMIS. Performance aspects and benchmarks for the speed of the different systems were recorded by the students and are documented in [SSW].

The basic Shibboleth scenario currently supported in DyVOSE demonstrates how the Grid based search and sort service can be securely accessed and used via Shibboleth technologies. Specifically it supports scenarios demonstrating how the attributes related to users being members of *studentteam1* (or *studentteam2*) are returned from the IdP at NeSC Glasgow and used to restrict access to the service itself (which has been deployed as a portlet in a GridSphere web portal). In supporting this scenario we have utilised the PERMIS Shibboleth Apache Authorisation Module (SAAM) module [XCO] which allows use of the PERMIS infrastructure to make authorisation decisions, as opposed to the existing Apache authorisation module (*mod_auth_ldap*). Currently the IdP returns two attributes: the role that the student has (*studentteam1*) and the DN. These attributes are then used and linked through the SAAM module to make authorisation decisions. This system was successfully demonstrated at the JISC Core Middleware workshop in November 2005 (http://www.jisc.ac.uk/index.cfm?name=middleware_cmpm2). In this scenario we utilised server certificates to overcome the issues in creation of proxy certificates and for submission of jobs via Grid services to the Condor pool at NeSC. Thus client side certificates are not required. We are also working towards solutions utilising MyProxy – as are the SHEBANGS and ShibGrid projects.

Some of the challenges we faced in developing this solution were the Apache focus of Shibboleth and linking these solutions to GridSphere (which uses the tomcat container) and which housed our Grid services (portlets). It is also the case that having a Shibboleth target behind a portal introduces its own challenges (since the web pages are normally dynamically created). Thus it is not easy to directly have a portlet as a Shibboleth target. Instead, we focused on solutions where the portal itself was the Shibboleth target. In doing this we use remote authentication (at the NeSC Glasgow IdP) to gain access to the portal and retrieval of attributes needed for invoking the services.

Dynamically establishing arbitrary VOs where no prior agreements on security attributes have been arranged is a research challenge and is being explored within a PhD studentship associated with the NeSC MRC funded VOTES project (www.nesc.ac.uk/hub/projects/votes).

We are looking towards Shibboleth based single sign on to the services available within the BRIDGES project also, which will demonstrate how we are able to use Shibboleth for job submission to the NGS (amongst numerous other compute resources).

3.5 Grid Security and Fabric Management

Whilst considerable progress has been made in developing advanced security infrastructures that are well integrated into Grid middleware [PERMIS,GSI,CAS] the wider issue of general infrastructure and Grid security have not been adequately considered by the Grid or security communities. Integrated security frameworks comprising sets of management tools for ensuring the security of Grid infrastructures and the fabrics upon which they exist are not currently available.

To understand the need to consider the combined consideration of Grid security and the wider impact of fabric security, consider the following (realistic!) scenarios:

- A laptop used by a Physicist in CERN has become infected with a virus. Through exploiting the Grid infrastructure, the virus quickly spreads to all connected nodes.
- A PC connected to the Grid has been hacked through a non-blocked port. The hacker uses this PC to get the X.509 proxy certificate passwords of Grid clients. These passwords are used to access other Grid nodes and to install backdoor hacker's software for future unauthorised access.
- A Grid based application demonstration requires that certain ports are opened on the firewalls across collaborating sites. Several weeks of negotiation are necessary with all local system administrators to achieve this.
- Before authorising access to patient records as part of a clinical trial, an NHS review board requires that a security validation process is made of the target Grid infrastructure. The data is not released since all Grid sites are unable to ensure that all necessary security measures have been taken across all collaborating sites.

All of these scenarios represent current problems and potential dangers with existing Grid technologies and their underlying computational infrastructures. The fact that Grid infrastructures have not been more seriously compromised thus far, has been more due to lack of awareness of the non-Grid community, as opposed to techniques and tools to prevent such things occurring. With the intended "ramping up" of Grid technology to industry and academia, this anonymity no longer offers sufficient protection. The recent compromise of the TeraGrid infrastructure is testament to this [TG]. The consequences of a full blown compromise of large scale Grid systems would be disastrous. In the worse case, the machines themselves would have to be completely rebuilt, including the Grid middleware, to ensure that no backdoors had been added for future hacker's access to the machines. Perhaps more damaging is the perceived lack of security of the Grid middleware and the expertise of the sites themselves. Trust underpins Grids and e-Research – trust of both the Grid software, the wider software infrastructure upon which Grid middleware depends, and of course people. Blind trust of any of these is naïve and will prohibit the uptake of Grid-based e-Research and lead to future compromises.

What is required is a security management framework that will provide an infrastructure for measuring, quantifying and improving the security of the middleware and end system software across entire Grid fabrics. Through this, the level of trust across a given virtual organisation (VO) can be improved, and the need

for blind trust diminished. This will go some way to overcome the possible perception from the non-Grid community that Grids are something inherently dangerous and to be avoided - a perception that will not change until tools are available which allow for the unified treatment of Grid security and security of the underlying fabric. In short, VO members and partners need as far as possible to ensure that all sites take all appropriate security measures seriously as deemed necessary for the VO. The existing model of trusting VO sites to take all appropriate security measures is something that will in certain instances simply not be tenable. "Trust but verify" was the maxim of one well known US president. Tool support to automatically manage, configure and verify the security of VO nodes to minimise risks is therefore essential.

The Grid and the wider computing science communities have developed technologies that allow for the management and configuration of distributed collections of heterogeneous resources. Solutions such as CFEngine (<http://www.cfengine.org>), SmartFrog (<http://www.smartfrog.org>) and the ongoing work on OGSConfig (<http://groups.inf.ed.ac.uk/ogsconfig/>) are typical examples of distributed fabric configuration tools. Until now these technologies have not focused upon the issue of ensuring Grid security across a given VO - not least due to the perceived lack of security in previous Grid solutions identified earlier. Rather, these technologies have focused upon the more general issues of configuration of computational resources. With enhanced Grid security mechanisms however, exploitation and integration of these tools with Grid middleware will help improve Grid security and the security of the fabrics themselves.

There are numerous challenges in achieving an integrated framework encompassing Grid and fabric security management. These include:

- identification of compute resources that are potential risks to other nodes of a given VO;
- measuring the "security level" of these nodes;
- acquiring the privilege level required to install patches on end systems, or if this cannot be done then quarantining the system until it is patched;
- the difficulties of OS dependencies that make patching and anti-virus upgrades non-trivial; reacting in a timely manner to virus threats;
- the fluidity of Grid technologies and standards; the identification of the misuse of VO nodes, e.g. in launching distributed denial of service attacks, as well as the non-technical issues in education to ensure that appropriate security guidelines and recommendations are followed.

This latter point should be emphasised since as identified previously, computer security is not solely a technical issue.

Ideally this framework will provide a model and associated reference implementation of a security management infrastructure offering, in the long term, numerous capabilities for establishing and managing the security of VOs. The framework should provide core functionality that allows for the setting and enforcement of security policies and the measuring and subsequent management of the security of a VO, including the quarantining (and reinstatement) of infected or risky machines.

The development of such a framework should allow for several exemplar services as proof of concept including:

- dynamic deployment of anti-virus software;
- automatic security patching of OSs and Grid middleware;
- dynamic configuration and management of firewalls.

Additional services may subsequently be added to this framework, such as management of intrusion detection systems and secure auditing etc. Thus for example, before a VO is established it might well be require to run intrusion detection software to ensure that sites and site logs have not been tampered with. Such capabilities will push advanced authorisation infrastructures to the limit, since this is completely outwith the normal usage of authorisation infrastructure. To support such highly intrusive activities will require fine grained policy specification and enforcement activities, and capabilities not currently present in authorisation infrastructures. For example, the definition and enforcement of policy obligations and policy overrides (<https://forge.gridforum.org/projects/ogsa-authz>). An example of the former includes ensuring that local system administrators are (are obliged to be!) informed prior to a VO middleware patch. Examples of the latter include scenarios where a VO site policy might state that no reconfiguration or changes to the software are needed until a given production run is complete, but a credible security threat has been found and a patch is urgently needed. In this case the local system administrator will always have the possibility to override any policies (since Grids must be autonomous), but it might also be the case that a remote administrator could have privileges assigned (is trusted enough) that such changes can be made.

The development and realisation of such a framework represents arguably the most challenging security infrastructure to support complete Grid security, encompassing both OS level, Grid middleware level and application level security. One of the challenges in achieving this is the multitude of Grid solutions available today and the complexities in their associated software stacks. Thus for example, changes or patches to elements of the gLite software stack (or most other Grid stacks for that matter) will undoubtedly break some aspects of the overall functionality (hence the restrictions on specific OS versions (Scientific Linux 3) currently imposed by the EGEE community). It is a fact however, that other domains such as healthcare will at some stage require that the software has been validated across all VO nodes. This is a normal occurrence for example when clinical trials are to be conducted.

Despite these challenges, it is the case that fine grained Grid security is achievable now; toolsets for fabric management are available now; test suites for Grid infrastructures have been explored within the UK e-Science community; the dynamic installation of software is an artefact of Grid middleware. Combining these solutions is urgently needed, since without this work, the Grid will always be at best perceived as something inherently non-secure, and at worst a direct threat to academic, commercial, financial, and governmental systems.

3.6 Web Service Security Standards

The development of robust Grid security infrastructures is very much dependent upon agreements on technologies and practices. Standardisation plays an extremely important role in this regard. With the move of the Grid community towards web services and service-oriented architectures, web service security standards and their associated implementations are crucial. Unfortunately it is the case that a multitude of specifications and proposals for web service standards have been promised and put forward, or merely promised. There are often cases of web service standards covering similar topics resulting in multiple competing specifications such as WS-Notifications [WS-N] and WS-Eventing [WS-E]; WS-ReliableMessaging [WS-RM] and WS-Reliability [WS-R]; WS-Orchestration [WS-O], WS-Co-ordination [WS-Co] and WS-Choreography [WS-Ch], along with the many varieties of workflow or business process languages that have been put forward to name but a few examples of the

issues in the proliferation of web service standards. It is also the case that at the time of writing, many web services standards are only in working draft or draft status, often with no associated implementations or acknowledged conformance or interoperability definitions. Claiming conformance or compliance to a particular web service standard is thus often not possible (or meaningful!).

It is also apparent that although many standards use the common prefix “WS-”, this does not mean that there is an agreed WS-Architecture. This stems from a variety of reasons: vendor and commercial issues; political aspects and also the different bodies involved. For example the Internet Engineering Task Force (IETF) (www.ietf.org); the World Wide Web Consortium (W3C) (www.w3.org); the Organization for the Advancement of Structured Information Standards (OASIS) (www.oasis-open.org); and the Web Services Interoperability Organization (WS-I) (www.ws-i.org) are some of the most prominent bodies. The consequence of this profusion of standards and standards making bodies, and the lack of consensus on the core web service architecture, impacts directly upon development of Grid standards, architectures and associated implementations and middleware.

With this complexity in mind, several key standards have nevertheless been identified for web service security. Figure 6³ provides a snapshot of the current status of some of the web service security standards as of October 2005.

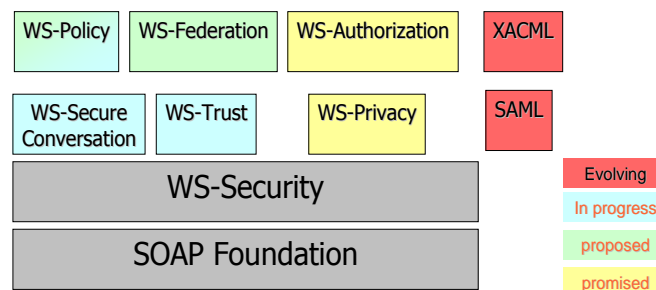


Fig 6. Web Service Security Standards

We provide a brief overview of these security standards. All of these standards build upon the basic SOAP foundations which include XML Signature [XMLSig] and Encryption [XMLEncrypt] for ensuring the security of messages. The XML Signature specification defines a methodology for cryptographically signing XML. The signatures are defined using a *<Signature>* element and accompanying sub-elements as part of a security header. The signature itself is computed based on the SOAP message content and a security token.

3.6.1 WS-Security

WS-Security describes enhancements to SOAP messaging to provide security enhancements for message integrity and message confidentiality. WS-Security also defines a general purpose mechanism for how to attach and include security tokens within SOAP messages including binary encoded security tokens such as X.509

³ This figure is taken from a set of slides presented by Dr Von Welch at the JISC security meeting in Edinburgh, October 2005, which itself was based upon “Security in a Web Services World: A Proposed Architecture and Roadmap, A Joint White Paper from IBM Corporation and Microsoft Corporation April 7, 2002, Version 1.0”

certificates. These mechanisms can be used independently or in combination to accommodate a wide variety of security models and encryption technologies.

Message integrity is provided by leveraging XML Signature in conjunction with security tokens to ensure that messages are transmitted and received without modifications. The integrity mechanisms are designed to support multiple signatures, potentially by multiple actors, and to be extensible to support additional signature formats. The signatures may themselves reference security tokens.

Message confidentiality is provided by leveraging XML Encryption in conjunction with security tokens to keep portions of SOAP messages confidential. Any portions of SOAP messages, including headers, body blocks, and substructures, may be encrypted. It should be noted that the encryption mechanisms of XML Encryption are designed to support additional encryption technologies, processes, and operations by multiple actors. The encryption itself can be realized using either symmetric keys shared by the sender and the receiver of the message or a key carried in the message in an encrypted form.

WS-Security defines a framework for securing SOAP messages, with the specifics being defined in profiles determined by the nature of the security token used to carry identity information. There are for example different profiles of WS-Security for various different security token formats such as X.509 certificates and Kerberos tickets. There is also a SAML token profile of WS-Security that specifies how SAML assertions can be used to provide message security. Additionally, SAML itself points to WS-Security as an approved mechanism for securing SOAP messages carrying SAML protocol messages and assertions.

WS-Security has now been fully implemented by several web service providers and the Grid middleware community. For example, the OMII server and client software stacks provide an implementation of WS-Security based upon Axis and WSS4J [WSS4J].

3.6.2 WS-Policy

WS-Policy [WS-Policy] describes how senders and receivers can specify their security requirements and capabilities. WS-Policy has been designed to be extensible and does not place limits on the types of requirements and capabilities that may be described. However, the specification does identify several basic attributes including privacy attributes, encoding formats, security token requirements, and supported algorithms. WS-Policy thus provides a flexible and extensible grammar for expressing the capabilities, requirements, and general characteristics of web service-based systems. WS-Policy also defines a framework and a model for the expression of these properties as policies. Policy expressions can include both simple declarative assertions as well as more sophisticated assertions. A policy itself can be regarded as a collection of one or more policy assertions. These assertions might include for example the authentication scheme, transport protocol selection, privacy policy, or quality of service characteristics. WS-Policy provides a single policy grammar to allow for such kinds of assertions to be reasoned about in a consistent manner.

It should be noted that WS-Policy stops short of explicitly specifying how policies are discovered or attached to a web service. It is envisaged that subsequent specifications will provide profiles on WS-Policy usage within given web services technologies and domains. For example, specifications for WS-PolicyAttachments, WS-PolicyAssertions, WS-SecureConversation have been put forward already as have various domain-specific assertions such as WS-SecurityPolicy and WS-ReliableMessagingPolicy. (See [WS-Policy] for further information).

3.6.3 WS-Trust

The goal of WS-Trust [WS-Trust] is to enable applications to construct *trusted* SOAP message exchanges. WS-Trust uses the basic mechanisms for secure messaging from WS-Security and defines additional primitives and extensions for security token exchange to enable the issuance and dissemination of credentials within and between different trust domains. Thus for example, to secure a communication between two parties, the two parties must exchange security credentials (either directly or indirectly). However, each party needs to determine if they can *trust* the asserted credentials of the other party. To support such situations, WS-Trust has defined extensions to WS-Security that provide methods for issuing, renewing, and validating security tokens; and ways to establish, assess the presence of, and broker trust relationships. Through these extensions, applications can engage in secure communication designed to work with the general web services framework including WSDL service descriptions, UDDI and SOAP messages.

The latest version of the WS-Trust language specification was released in February 2005.

3.6.4 WS-Privacy

The WS-Privacy specification was outlined in a joint white paper from IBM and Microsoft [WSW]. Here it was presented how the WS-Privacy specification could address how privacy practices could be stated and subsequently implemented and enforced by web services. By using a combination of WS-Policy, WS-Security and WS-Trust, organizations should be able to state and indicate conformance to stated privacy policies. The specification would describe a model for how a privacy language could be embedded into WS-Policy descriptions and how WS-Security may be used to associate privacy claims with a message. In addition, the WS-Privacy specification would describe how WS-Trust mechanisms could be used to evaluate these privacy claims for both user preferences and organizational practice claims.

At the time of writing, the WS-Privacy specification and associated implementation(s) have not materialised, nor is it clear when they will appear.

3.6.5 WS-SecureConversation

The Web Services Secure Conversation Language (WS-SecureConversation) [WS-SC] allows clients and web services to establish a token-based, secure conversation for the duration of a given session. The secure conversation itself is based on security tokens that are procured from a service token provider. Once obtained and a secure channel established, the client and service exchange a lightweight, signed security context token, which optimizes message delivery time compared with using regular security tokens. The security context token enables the same signing and encryption features as other security tokens such as X509 security tokens.

WS-SecureConversation itself is built on top of the WS-Security and WS-Policy models to provide secure communication between services. WS-Security focuses on the message authentication model but not a security context, and thus is subject several forms of security attacks. WS-SecureConversation defines mechanisms for establishing and sharing security contexts, and deriving keys from security contexts, to enable a secure conversation.

It should be noted that WS-SecureConversation by itself does not provide a complete security solution rather WS-SecureConversation is a building block that is used in conjunction with other web service and application-specific protocols such as

WS-Security to accommodate a wide variety of security models and technologies. It should also be noted that WS-SecureConversation is designed to operate at the SOAP message layer so that the messages may traverse a variety of transports and intermediaries. This does not preclude its use within other messaging frameworks. In order to further increase the security of the systems, transport level security may be used in conjunction with both WS-Security and WS-SecureConversation across selected links.

Several implementations of WS-SecureConversation are now available for example within Microsoft Web Service Enhancements for the .NET platform [WSE].

3.6.6 WS-Federation

The Web Service Federation Language (WS-Federation) [WS-Fed] defines how to construct federated trust scenarios using the WS-Security, WS-Policy, WS-Trust, and WS-SecureConversation specifications. For example, WS-Federation describes how to federate between Kerberos and PKI infrastructures. The WS-Federation specification defines the model and framework for federation between security domains. Subsequent documents define profiles which detail different ways that the WS-Federation language can be applied.

WS-Federation supports specification of a trust policy to identify and constrain the type of trust that is being brokered. Through this different security realms are able to federate by supporting the brokerage of trust of identities, attributes, and authentication information between participating web services.

The last version of the WS-Federation specification was released in July 2003. Since then various implementations of WS-Federation have been put forward. For example, Microsoft, IBM, RSA Security Inc. and various other vendors have implemented this specification and demonstrated interoperability between their implementations through passing a particular identity between six exemplar portals at a workshop organised in May 2004 [WS-FW].

3.6.7 WS-Authorization

A standard for authorization does not exist for web services. In the Microsoft/IBM roadmap for web services security white paper [WSW], an outline for WS-Authorization was loosely described. This document outlined how the WS-Authorization specification would “describe how access policies for a web service are specified and managed. In particular it will describe how claims may be specified within security tokens and how these claims will be interpreted at the endpoint. This specification will be designed to be flexible and extensible with respect to both authorization format and authorization language. This enables the widest range of scenarios and ensures the long-term viability of the security framework”.

However, the WS-Authorization specification has not (yet?) been published. Since this roadmap document was published, developments within the Grid community regarding authorisation and how such infrastructures can be seamlessly linked to Grid services have matured however (as described in section 3.3.1). As such, from a Grid community perspective, the question may well be asked, what would a WS-Authorization specification offer that can not yet be supported by Grid based solutions and existing authorisation infrastructures?

3.6.8 Security Assertion Markup Language (SAML)

The OASIS SAML specification [SAML1-1] is an XML-based framework for communicating user authentication, entitlement, and attribute information. SAML allows making assertions regarding the identity, attributes, and entitlements of a subject to other entities. SAML has been designed to be a flexible and extensible protocol which can be customised by other standards. For example, the Liberty Alliance, the Internet2 Shibboleth project, and the OASIS Web Services Security committee have all adopted SAML for various purposes.

SAMLv1.0 became an OASIS standard in November 2002. SAMLv1.1 followed in September 2003 and has seen significant success, gaining acceptance across a wide range of domains and is supported by numerous security technology providers.

SAML is defined in terms of assertions, protocols, bindings, and profiles. An assertion is a package of information that supplies one or more statements made by a SAML authority. SAML defines three different kinds of assertion statement that can be created by a SAML authority:

- **Authentication:** which indicates that the specified subject was authenticated by an identity provider through some means at some given time.
- **Attribute:** The specified subject is associated with the supplied attributes.
- **Authorization Decision:** A request to allow the specified subject to access the specified resource has been granted or denied.

The outer structure of an assertion is generic, providing information that is common to all of the statements within it. Within an assertion, a series of inner elements describe the authentication, attribute, authorization decision, or user-defined statements containing the specifics.

SAML defines a number of request/response protocols that allow service providers to request various things. For example, to request one or more assertions from given SAML authorities, or to request that an identity provider authenticate a principal and return the corresponding assertion.

Mappings from SAML request-response message exchanges into standard messaging or communication protocols are called SAML protocol bindings. A SAML SOAP Binding has been defined which outlines how SAML protocol messages can be communicated within SOAP messages.

A profile of SAML typically defines constraints and/or extensions in support of the usage of SAML for a particular application. For instance, the Web Browser Single Sign On [WebSSO] profile specifies how SAML authentication assertions are communicated between an identity provider and service provider to enable single sign-on for a browser user. This profile details how to use the SAML Authentication Request/Response protocol in conjunction with different combinations of the HTTP Redirect, HTTP POST, HTTP Artefact, and SOAP bindings.

Other SAML profiles also exist such as attribute profiles which provide specific rules for interpretation of attributes in SAML attribute assertions. For example the X.500/LDAP profile, describing how to carry X.500/LDAP attributes within SAML attribute assertions.

SAMLv2.0 unifies the building blocks of federated identity in SAMLv1.1 with input from the Internet2 Shibboleth initiative and the Liberty Alliance's Identity Federation Framework [LA-IFF]. As such, SAMLv2.0 is a significant step towards convergence for federated identity standards.

SAMLv2.0 includes numerous additional features from v1.1 including support for:

- opaque pseudo-random identifiers (pseudonyms) which can be used between providers to represent principals.
- identifier management allowing providers to establish and subsequently manage the pseudonym(s) for the principals for whom they are operating.
- metadata defining how to express configuration and trust related data to make deployment of SAML systems easier.
- attribute statements, name identifiers, or entire assertions may be encrypted in SAMLv2.0. This feature ensures that end-to-end confidentiality of these elements may be supported as needed.
- attribute profiles which simplify the configuration and deployment of systems that exchange attribute data. These include basic attribute profiles for string based attribute names and XML schema primitive type attribute value definitions; X.500/LDAP attribute profiles; and XACML attribute profiles.
- SAMLv2.0 supports situations where authenticated users can be automatically logged out of all service providers in the session at the request of the identity provider.
- SAMLv2.0 includes mechanisms that allow providers to communicate privacy policy and settings. For instance, SAML makes it possible to obtain and express a principal's consent to some operation being performed.
- In scenarios with more than one identity provider, service providers need a means to discover which identity provider(s) a principal uses. The identity provider discovery profile relies on a cookie written in a common domain between identity and service providers.

The SAMLv2.0 specification was release at the end of September 2005.

3.6.9 Liberty Alliance

The Liberty Alliance [LibAll] is an industry consortium defining standards for federated identity – including enabling simplified sign-on through federated network identification, as well as supporting and promoting permission-based attribute sharing to enable a user's choice and control over the use and disclosure of their personal identification information.

The Liberty Alliance Identity Federation Framework (ID-FF) [LA-IFF] is based on SAML. Recognising the value of a single standard for federated single sign on, the Liberty Alliance submitted their Identity Federation Framework to the OASIS Security Services Technical Committee as input to SAMLv2.0. It intends to use the new version of SAML in concert with its own technical and business guidelines for identity federation going forward.

Liberty's Identity Web Services Framework (ID-WSF) [LA-WSF] provides a platform for communicating identity information among web services and continues to be developed within the Liberty Alliance. The latest version of Liberty ID-WSF now uses SAMLv2.0 assertions as the security token format for communicating authentication and authorization information amongst web service actors.

SAML assertions can be used within SOAP messages in order to convey security and identity information between actors in web service interactions. The SAML Token Profile produced by OASIS specifies how SAML assertions should be used for this purpose with the WS-Security framework. The Liberty ID-WSF builds on these specifications to use SAML assertions for enabling secure and privacy-respecting access to web services.

3.6.10 Extensible Access Control Markup Language (XACML)

XACML [XACML] is an OASIS [OASIS] standard that describes both a policy language and an access control decision request/response language (both written in XML). XACML version 2.0 is the latest release and was published in February 2005. The policy language associated with XACML is used to describe general access control requirements, and has standard extension points for defining new functions, data types, combining logic, etc. The request/response language allows formation of queries to ask whether or not a given action should be allowed, and interpret the result. The response always includes one of four values: Permit, Deny, Indeterminate (an error occurred or some required value was missing, so a decision cannot be made) or Not Applicable (the request can't be answered by this service).

The typical setup is that someone wants to take some action on a resource. They will make a request to a PEP protecting a resource. The PEP will form a request based on the requester's attributes, the resource in question, the action, and other information pertaining to the request. The PEP will then send this request to a PDP, which will look at the request and some policy that applies to the request, and come up with an answer about whether access should be granted. That answer is returned to the PEP, which can then allow or deny access to the requester. In addition to providing request/response and policy languages, XACML also supports finding policies that apply to a given request and subsequent evaluation of requests against that policy. XACML also allows for generic, distributed policies to be supported. Thus a policy can be written which refers to other policies kept in various remote locations. Hence rather than having to manage a single monolithic policy, different people or groups can manage sub-pieces of policies as appropriate, and XACML supports combination of the results from these different policies into one decision.

XACML comes with a core base language which can be extended. The standard language supports a wide variety of data types, functions, and rules about combining the results of different policies. In addition to this, standards groups are working on extensions and profiles that will hook XACML into other standards like SAML and LDAP, which will increase the number of ways that XACML can be used.

XACML 2.0 and all the associated profiles were approved as OASIS Standards in February 2005.

4 Conclusions and Recommendations

In this report we have explored the current limitations of authentication only based solutions with PKIs as the basis for Grid security. Lack of granularity of authorisation will dissuade large groups of researchers from engaging. Perhaps more of an issue is the initial steps through which researchers are asked to proceed before they are able to do anything "on the Grid". X.509 certificates and the process of acquiring one and converting it to appropriate formats is a hurdle that a large swathe of the non-Grid research community will not overcome. It needs to be made simpler and ideally removed completely. Having local system administrators requesting batches of certificates which can be used at the local campus Grid level and setting up the Grid environments for researchers is one possibility. Alternative models might be based upon local institutions setting up their own CAs to issue certificates which, when signed by the local RA (using their UK e-Science certificate) are accepted by the wider Grid community. Through such approaches, the issues of obtaining certificates, converting them into appropriate Grid formats and setting up the environment for end

users to undertake their Grid-based research can be removed completely from the researchers and moved to the supporting e-Science centres for example.

The BRIDGES project and its use of server certificates and a fixed core set of services is one way in which the user can be shielded from security aspects. Users are comfortable with logging in with usernames and passwords to services/portals, hence such a solution should be explored more widely.

There are numerous authorisation frameworks available today and we have tried to give an overview of their functionality and suitability of some of the most prominent of these. Of those we have listed here, the PERMIS middleware is arguably the most mature solution with tools available for security policy specification and enforcement; for linkage to Grid services in a generic manner; and for linkage to Shibboleth. All of these tools and capabilities have been explored and proven to function at NeSC Glasgow. It is still the case that wider uptake and application across a range of different scenarios is needed before they solutions can be hardened into real products however. For example, considerable effort is still required for deployment and configuration of PERMIS and its interworking with Globus and Shibboleth solutions. This will no doubt resolve in time. For example the SAML AuthZ API (which should be a direct SAML call out) has not been backwardly compatible between earlier versions of Globus (GT3) and current versions of Globus (GT4) with PERMIS. Rigorous testing and further feedback from applications exploring these and other APIs are necessary.

Grids will always be seen as a threat if they ignore the issue of fabric management. A unified treatment and associated framework for analysing the security of Grid applications, Grid middleware and the underlying OS is needed. If VOs are to be truly secure, then blindly trusting partners to take necessary steps is naïve. We have outlined what sort of steps might be taken through the risk analysis that was undertaken at NeSC. Of course, such a risk analysis needs to be repeated, perhaps annually. In addition to the framework and hope that sites might undertake a risk analysis and put in place security policies and practices, it might well be beneficial to have a body authorised with ensuring UK wide Grid security. This body might well include members of the STF and GOSC (or other bodies). In security, the weakest link is always the one that counts. As such, all sites need to be educated in Grid security and the wider issue of fabric security.

Shibboleth represents a clear opportunity to overcome the current issues with PKI based security. Trust federations at an institutional level where users can authenticate at their home site and have appropriate attributes released to service providers (which will use them to make authorisation decisions) changes the dynamic of security. There has always been a large degree of trust in the Grid community: trust of users, trust of sites, trust of CAs etc. Hence Shibboleth does not add a new trust requirement especially. Instead trust is moved to IdPs (and ensuring that they have appropriately strong authentication and authorisation schemes) and WAYFs (which ensure that the “correct” IdPs are identified and matched with SPs).

Understanding what attributes are needed in the Grid community is essential. Many solutions may only require that the DN is passed over for example, so that accounting and logging of the resource usage for that individual can be achieved, i.e. not further attributes are needed to make an authorisation decision. Other more prescriptive VOs may require more information such as VO membership, role of the user etc. Mapping such attributes into a form that Shibboleth can make use of, e.g. *eduPersonEntitlement* attribute is needed. Once such scenarios can be supported,

more understanding of the attribute release policies and how they might be implemented can be achieved.

It is clear that the web service standards community are producing numerous specifications which in principle could help simplify Grid security. Single sign on solutions to services at numerous sites via SAMLv2.0, and complementary efforts within the Liberty Alliance consortia offer potential solutions of direct relevance to the Grid community in its move towards web based solutions and service oriented architectures.

5 References

- [PKI] R. Housley, T. Polk, *Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructures*, Wiley Computer Publishing, 2001.
- [X509] ITU-T Recommendation X.509 (2001) | ISO/IEC 9594-8: 2001, Information technology – Open Systems Interconnection – Public-Key and Attribute Certificate Frameworks.
- [TIES] JISC Authentication, Authorisation and Accounting (AAA) Programme Technologies for Information Environment Security (TIES), http://www.edina.ac.uk/projects/ties/ties_23-9.pdf
- [ESP] ESP-Grid project, e-science.ox.ac.uk/oesc/projects/index.xml.ID=body.1_div.1
- [PH] W. T. Polk and N. E. Hastings, *Bridge Certification Authorities: Connecting B2B Public Key Infrastructures*, <http://csrc.nist.gov/pki/documents/B2B-article.doc>
- [JBH] J. Jokl, J. Basney and M. Humphrey, Experiences using Bridge CAs for Grids, Proceedings of UK Workshop on Grid Security Practice - Oxford, July 2004
- [LRA] J. Liddell, K. V. Renaud, and A. De Angeli, *Authenticating users using a combination of sound and images*. HCI 2003, Bath, UK, September 2003.
- [KR] K. Renaud, *Quantifying the quality of web authentication mechanisms: a usability perspective*. Journal of Web Engineering, 3(2):95–123, 2004.
- [SB] S. Booth, *Grid Firewall Recommendations*, <http://www.grid-support.ac.uk/etf/firewalls/Firewalls.html>
- [RAH] A. Richards, R. Allan, D. Hanlon, *Globus Toolkit Firewall Port Selection*, <http://www.grid-support.ac.uk/etf/firewalls/FirewallPortSelection.pdf>
- [BOS] M. Baker, H. Ong, G. Smith, *A Report on Experiences Operating the Globus Toolkit through a Firewall*, <http://esc.dl.ac.uk/Papers/firewalls/globus-firewall-experiences.pdf>
- [MS] M. Surridge, *Rough Guide to Grid Security*, http://www.nesc.ac.uk/technical_papers/RoughGuidetoGridSecurityV1_1a.pdf
- [X812] ITU-T Rec X.812 (1995) | ISO/IEC 10181-3:1996, Security Frameworks for open systems: Access control framework.
- [WSCMP] 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>
- [OASIS] Organization for the Advancement of Structured Information Standards (OASIS), <http://www.oasis-open.org>
- [XACML] eXtensible Access Control Markup Language TC v2.0 (XACML), <http://www.oasis-open.org/specs/index.php#xacmlv2.0>
- [WS-S] Web Services Security (WS-Security), version 1.0 5th April 2002, www-106.ibm.com/developerworks/webservices/library/ws-secure
- [WS-E] Web Services Eventing (WS-Eventing), www-128.ibm.com/developerworks/webservices/library/specification/ws-eventing
- [WS-N] Web Service Notifications (WS-Notifications), <http://www-128.ibm.com/developerworks/library/specification/ws-notification/>
- [WS-RM] Web Services Reliable Messaging (WS-ReliableMessaging), <http://www-128.ibm.com/developerworks/library/specification/ws-rm/>
- [WS-R] Web Services Reliability (WS-Reliability), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrc
- [WS-C] Web Services Co-ordination (WS-Co-ordination), <http://www-128.ibm.com/developerworks/library/ws-coor/>

- [WS-Ch] Web Services Choreography (WS-Choreography), <http://www.w3.org/TR/ws-chor-model/>
- [WS-O] Web Services Orchestration (WS-Orchestration), <http://www-128.ibm.com/developerworks/webservices/library/ws-bpelcol2/>
- [WSS4J] Apache WSS4J, <http://www.ws.apache.org/axis>.
- [WS-Policy] Web Services Policy Framework, September 2004, <http://www-128.ibm.com/developerworks/library/specification/ws-polfram/>
- [WS-Trust] Web Services Trust Language, February 2005, <http://www-128.ibm.com/developerworks/library/specification/ws-trust/>
- [WS-Fed] Web Service Federation Language (WS-Federation), <http://www-128.ibm.com/developerworks/webservices/library/ws-fed/>
- [WS-FW] WS-Federation Passive Requester Profile Interoperability Workshop, <http://msdn.microsoft.com/webservices/community/workshops/wsfedprmar2004.aspx>
- [WS-SC] Web Services Secure Conversation Language, <http://www-128.ibm.com/developerworks/library/specification/ws-secon/>
- [WSE] Microsoft Web Service Enhancements (WSE), <http://msdn.microsoft.com/webservices/webservices/building/wse/>
- [WSW] *Security in a Web Services World: A Proposed Architecture and Roadmap*, A Joint White Paper from IBM Corporation and Microsoft Corporation, April 7, 2002, Version 1.0.
- [XMLSig] IETF/W3C XML DSIG Working Group, <http://www.w3.org/Signature/>
- [XMLEnc] W3C XML Encryption Syntax and Processing, W3C Recommendation, December 2002 <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>
- [SAML1-1] OASIS, Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1, 2 September 2003, <http://www.oasis-open.org/committees/security/>
- [SAML2] Security Assertion Markup Language (SAML) version 2.0, March 2005, <http://www.oasis-open.org/specs/index.php#samlv2.0>
- [LibAll] Liberty Alliance, www.projectliberty.org
- [LA-IFF] Liberty Alliance Identity Federation Framework, <https://www.projectliberty.org/resources/specifications.php>
- [LA-WSF] Liberty Alliance Identity Web Service Framework version 1.1., <https://www.projectliberty.org/resources/specifications.php#box2a>
- [COB] 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.
- [CO] 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.
- [OpenSSL] OpenSSL to create certificates, <http://www.flatmtn.com/computer/Linux-SSLCertificates.html>
- [ShibA] Shibboleth Architecture Technical Overview, <http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf>
- [ShibP] Shibboleth Architecture Protocols and Profiles, <http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-latest.pdf>
- [GT2] Globus toolkit version 2, <http://www.globus.org/toolkit/downloads/2.4.3/>
- [GT4] Globus toolkit version 4, <http://www.globus.org/toolkit/downloads/4.0.1/>
- [EGEE] Enabling Grids for E-science (EGEE) project, public.eu-egee.org
- [gLite] gLite software, glite.web.cern.ch/glite
- [OMII] Open Middleware Infrastructure Institute (OMII), www.omii.ac.uk

- [CROWN] China Research and Development environment over Wide Area Network (CROWN), www.crown.org.cn
- [Condor] Condor software, www.cs.wisc.edu/condor
- [Unicore] UNICORE Forum, www.unicore.org
- [RM] A. Robiette, T. Morrow, *Blueprint for a JISC Production Federation*, JISC Development Group, Version 1.1: issued 27 May 2005, http://www.jisc.ac.uk/index.cfm?name=middleware_documents
- [GridShib] GridShib project, <http://grid.ncsa.uiuc.edu/GridShib/>
- [SSCO] R.O. Sinnott, A.J. Stell, D.W. Chadwick, O.Otenko, Experiences of Applying Advanced Grid Authorisation Infrastructures, Proceedings of European Grid Conference (EGC), pages 265-275, Vol. editors: P.M.A. Sloot, et al June 2005, Amsterdam, Holland.
- [SSW] R.O. Sinnott, A.J. Stell, J. Watt, Comparison of Advanced Authorisation Infrastructures for Grid Computing, Proceedings of International Conference on High Performance Computing Systems and Applications, May 2005, Guelph, Canada.
- [TG] TeraGrid attack, <http://www.washingtonpost.com/ac2/wp-dyn/A8995-2004Apr13>
- [SC] R.O. Sinnott, D.W. Chadwick, *Experiences of Using the GGF SAML AuthZ Interface*, Proceedings of UK e-Science All Hands Meeting, September 2004, Nottingham, England.
- [CHAD] D.W Chadwick, *An Authorisation Interface for the Grid*, Proceedings of UK e-Science All Hands Meeting, September 2003, Nottingham, England.
- [MyProxy] MyProxy Credential Management Service, myproxy.ncsa.uiuc.edu
- [XCO] W. Xu, D. Chadwick, A. Otenko, "Development of a Flexible PERMIS Authorisation Module for Shibboleth and Apache Server", 2nd European PKI Workshop, University of Kent, July 2005.
- [eduPerson] eduPerson Specification, <http://www.educause.edu/eduperson/>
- [AuthZ2] Prof David Chadwick, JISC proposal, Authorisation Interface V2 for the Grid, June 2005 – accepted for funding.
- [CAS] Community Authorisation Server – <http://www.lesc.ic.ac.uk/projects/cas.html>
- [CAS2] 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.
- [GSI] Globus Grid Security Infrastructure (GSI), <http://www.globus.org/toolkit/docs/4.0/security>
- [VOMS] R. Alfieri, et al, *Managing Dynamic User Communities in a Grid of Autonomous Resources*, CHEP 2003, La Jolla, San Diego, March, 2003;
- [STELL] A.J. Stell, *Grid Security: An Evaluation of Authorisation Infrastructures for Grid Computing*, MSc Dissertation, University of Glasgow, 2004.