ShibGrid: Shibboleth Access for the UK National Grid Service

David Spence, Neil Geddes, Jens Jensen, Andrew Richards and Matthew Viljoen
CCLRC Rutherford Appleton Laboratory
D.R.Spence@rl.ac.uk, J.Jensen@rl.ac.uk, A.J.Richards@rl.ac.uk, M.J.Viljoen@rl.ac.uk

Andrew Martin, Matthew Dovey, Mark Norman, Kang Tang, Anne Trefethen and David Wallom
University of Oxford
Andrew.Martin@comlab.ox.ac.uk, Mark.Norman@computing-services.oxford.ac.uk,
Kang.Tang@oerc.ox.ac.uk, David.Wallom@oerc.ox.ac.uk

Rob Allan and David Meredith
CCLRC Daresbury Laboratory
D.J.Meredith@dl.ac.uk

Abstract

This paper presents work undertaken to integrate the future UK national Shibboleth infrastructure with the UK's National Grid Service (NGS). Our work, ShibGrid*, provides both transparent authentication for portal based Grid access and a credential transformation service for users of other Grid access methods. The ShibGrid support for portal-based transparent Grid authentication is provided as a set of standards-based drop-in modules which can be used with any project portal as well as the NGS project in which they are initially deployed. The ShibGrid architecture requires no changes to the UK national Shibboleth authentication infrastructure or the NGS security infrastructure and provides access for users both with and without UK e-Science certificates.*

In addition to presenting both the architecture of ShibGrid and its implementation, we additionally place the ShibGrid project within the context of other efforts to integrate Shibboleth with Grids.

1 Introduction

Since 1996 the Athens Access Management System [16] has provided a highly-successful national Single Sign-On (SSO) service to UK academic users for web-based services. Building on this, the Joint Information System Committee (JISC) has recently settled on Shibboleth [18] as the next generation authentication method for accessing web-based resources [12]. Following on from this decision JISC has been actively involved in funding gateways between Shibboleth and other authorisation services, including Athens. Within this there has been a drive for Grid resources, especially the National Grid Service (NGS) (which is partly funded by JISC), to also be able to inter-operate with Shibboleth. ShibGrid, and its 'sister' project SHEBANGS [19], are consequently investigating possible methods for Shibboleth and Grid integration.

In this paper we describe the motivation (Section 1), architecture (Section 3), security considerations (Section 4) and implementation (Section 5) of the ShibGrid system for providing a gateway for users with Shibboleth credentials to access Grid resources. The ShibGrid project is also placed in the context of other similar SSO integration projects in Section 2.

1.1 Use Cases

ShibGrid supports two classes of users; users with an X.509 certificate [8] from a traditional Certificate Authority (CA) who wish to use the Shibboleth framework to protect their proxy certificates [21] without the need to repeatedly enter pass-phrases; and users who do not possess any pre-existing Grid credentials (or choose not to use any) and who wish to access Grid resources.

In both cases ShibGrid supports access to Grid resources through portals (with Shibboleth also automating authentication to the portal) and through more traditional command-line tools.

ShibGrid also presents a solution to one of the main issues of SSO systems; how to dependably link users' identity in two different security domains (Shibboleth/SAML and Grid Security Infrastructure/X.509) where there is no algo-

rithmic mapping or database linking them.

The Shibboleth framework is an implementation of the browser profiles from the OASIS SAML v1.1 specification [15], which provides a SSO service and (possibly pseudonymous) attribute exchange from the user's home site to the site he is accessing. Therefore Shibboleth separates user authentication, which is performed by the user's home site, and authorisation, which is performed by the site to be accessed, based on attributes that have been passed to it. An instance of the Shibboleth framework is called a *Federation*, within which all sites are mutually-trusting.

1.2 Key Requirements

The ShibGrid project seeks to provide a service to users; facilitates access to Grid services provided by the NGS[1] and the UK e-Science Certificate Authority (CA)[2]; and will use the Shibboleth federation provided by JISC and the UK higher-education institutes. These bodies all put significant requirements upon the ShibGrid project. The key requirements are listed here.

- **User Requirements:**
  - Where possible certificate-less access should be used; many users do not want ever to handle certificates.
  - Where users do have certificates these should be used as transparently as possible.
  - Access to the Grid should be simple and intuitive; users are not interested in middleware or authentication infrastructures.
  - Users still want to use their application-specific portals; any portal changes should be trivial to apply regardless of the portal.
- **NGS/CA:**
  - Authentication to the NGS must still be accomplished through the use of X.509 certificates and proxies. This authentication infrastructure is known as the Grid Security Infrastructure (GSI) [27].
  - The private keys corresponding to UK e-Science certificates cannot be stored in any way which means that others can obtain them (even administrators).
- **JISC/Higher-education institutes**
  - JISC is encouraging UK universities and other research institutions to take part in a national academic Shibboleth infrastructure. ShibGrid must fit into this infrastructure, therefore ShibGrid should not require changes to Identity Providers (IdPs) or the Shibboleth protocol.

[1] *http://www.ngs.ac.uk*
[2] *https://ca.grid-support.ac.uk*

2 Related Projects

The first project to make a significant connection between Grid software and Shibboleth was *GridShib*, which uses Shibboleth to provide attribute-based authorisation in the context of the Globus toolkit [26]. The use cases of this work envisage users, identified using X.509 identity certificates, offering a brokering service to enable those identities to form the basis of queries against attribute authorities—run either by the user's home campus, a particular Grid project, or a combination of the two. GridShib suggests both a *pull mode* (in which the use of Shibboleth could be entirely hidden from the user) and a *push mode* (in which the user first contacts a Shibboleth service to obtain attributes, before contacting the grid service), but only the pull mode is currently supported.

More recent work [3] incorporates MyProxy [4] using means similar to our own. Another extension incorporates *PERMIS* [9], allowing a richer, role-based access regime to be implemented.

Our 'sister' project is *SHEBANGS* [19]. This project has similar goals to our own—access to the UK National Grid Service via its portal—using similar components. That project has adopted a proxied 'push' model, wherein users first contact a credential translation service, which, after Shibboleth-based authentication and attribute retrieval, generates a credential stored in a MyProxy server. Access to that credential is then achieved by the user logging on to the portal with details returned to them from the credential translation service. Our projects are evolving together, and will share and compare experiences as the projects mature.

Shibboleth also features in the architecture of the Australian *Meta Access Management Systems* (MAMS) project [23]. Here, Shibboleth is used both to authenticate the user, via their institution's IdP, and then to give access to multiple repositories through a single federated interface. A range of credential transformation and attribute management tools help to enhance usability and interoperability.

A number of other projects are also working at the conjunction of Grid and Shibboleth, including the Swiss national infrastructure *SWITCHaai*, Oxford University's *ESP-GRID* and the *DyVOSE* project at the UK's National e-Science Centre. These and others are summarised in an informational document of the Global Grid Forum [25].

Other authentication frameworks have also been integrated with the Grid Security Infrastructure. MyProxy has been used as the basis for many of these projects and many efforts are now part of the main MyProxy distribution. Authentication methods integrated with MyProxy include (ticket-based) Kerberos authentication via the use of SASL (see [4, 11]), Pubcookie [14] and Pluggable Authentication Modules (PAM). PAM support provides access to all the local password-based authentication PAM modules on the server running MyProxy and can be used to support (password-based) Kerberos, LDAP and One Time Password (OTP) [17] authentication. In addition work on Kerberos CAs for accessing websites [13] has been applied to generate short-term X.509 certificates for use with Grids. In the main part these solutions only answer the issues of site SSO not inter-site SSO.

The GAMA project [5] provides another method by which the details of the Grid Security Infrastructure can be hidden from the user. In this case initial authentication happens out-of-band, users request an account which creates a hidden Grid credential for the user to use with a portal. The advantage of a Shibboleth based infrastructure is primarily that the users do not need to remember anymore passwords to access the Grid, but also that they do not need to explicitly request credential conversion. In addition the initial authorisation of users is performed by the user's home institution and not the project/Grid.

3 Architecture

The architecture of the ShibGrid project is based on the generic SSO model presented in previous work [11]. This system extends the SSO concept to the national scale using Shibboleth as a national SSO infrastructure.

Figure 1 shows how the ShibGrid architecture would be used with a portal. The steps in this figure are described below. Steps 1–6 represent the standard scheme for Shibboleth authentication. This cannot be changed because the system will eventually be a part of a national Shibboleth infrastructure, a requirement from Section 1.2.

Steps 7–9 describe a procedure very similar to how Grid portals would normally access a standard MyProxy server. This similarity helps fulfil another requirement from Section 1.2, to allow users to still use their current portals. The ShibGrid login module should be trivial to use in place of the normal MyProxy login module in use by pre-existing Grid portals.

1. The user requests access to the portal, the Service Provider (SP) in Shibboleth terminology, through Shibboleth login. The user's browser is redirected to the Where Are You From? (WAYF) service.

2. The user chooses their home institution from the list of the institutions in the federation, as returned by the WAYF service.

3. The user's browser is re-directed to the authentication service (SSO service in Shibboleth terminology) of their home institution's IdP.

4. The user is authenticated by their home institution's IdP's authentication service, through the site's authentication infrastructure (such as Kerberos, WebAuth, One Time Password, etc.,).

5. The IdP redirects the user's browser back to the portal. A signed SAML authentication assertion is passed in this redirect, containing a unique pseudonymous Id or "handle" for the user and demonstrating that the user has been authenticated as a member of that institution.

6. The portal calls out to the IdP's Attribute Authority (AA) for attributes about the user, using the handle. This interaction takes place using a mutually authenticated HTTPS connection so normally the attribute assertion returned by the AA is not signed. In our case we require that it is signed as we wish to pass it on. This can be specified using a standard Shibboleth option. In addition, (6a) the attributes can be used by the portal to make an access authorisation decision and/or to gather user information.

7. The portal attempts to obtain a credential from the ShibGrid MyProxy server. The standard Java Globus CoG libraries [22] can be used, with the username derived from the attributes and the entire signed attribute assertions used as the password. The MyProxy server then validates the attribute assertion by verifying that it was generated for the server that is downloading the credentials and that it matches the supplied username. If the portal is authorised and the user authenticated and authorised, then the MyProxy server returns either a proxy of the user's real certificate (if the user has already uploaded a proxy to the MyProxy server; see Section 3.1) or a low-assurance certificate[3] which is automatically generated by MyProxy's built-in CA. Where a certificate is generated, the Distinguished Name (DN) [8] used is based on the attribute information in the attribute assertion (e.g. their name and organisation).

8. The proxy certificate or low-assurance certificate is returned. Optionally, other attributes can be added to this

[3] Low-assurance certificate: a certificate signed by a online CA that generates certificate based on authentication by an electronic method, not by presenting photographic id to a human representative of the CA. The maximum lifetime of these certificates is one million seconds as opposed to one year for traditional certificates.

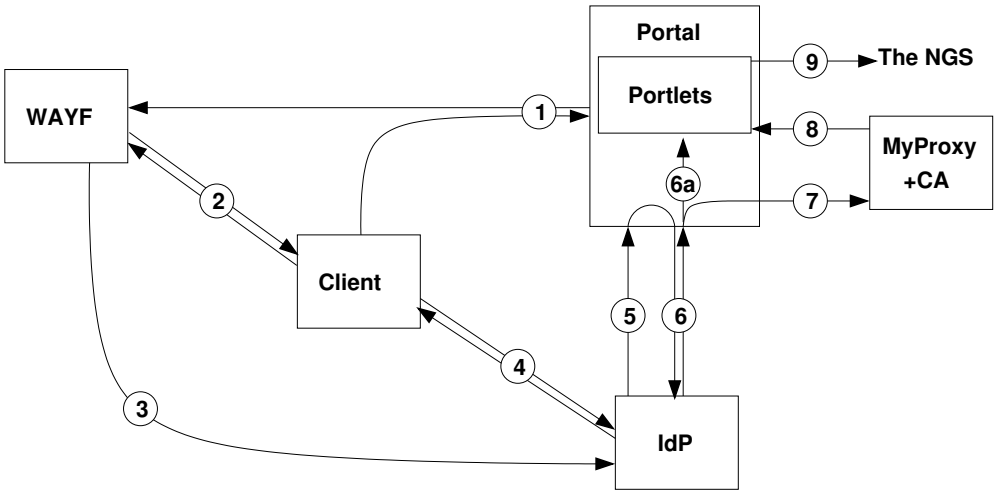

Figure 1. Using ShibGrid with a portal.

credential (through the use of certificate extensions). For example Virtual Organisation (VO) attributes retrieved from a VOMS [2] server could be written into the credential. These extensions can be used by gatekeepers to restrict access to resources.

9. The user can use the portal to access the NGS or other Grids with the returned credential. Alternatively, another website could pass the credential to the user to allow him to use traditional Grid access methods.

3.1 Uploading Credentials to the ShibGrid MyProxy Server

At an early stage of the project it was decided that if users have their own certificates and wish to use them, then there should be a robust way of linking their two identities (Shibboleth identity and X.509 certificate DN) together. Many systems using MyProxy servers allow users to upload proxies to the MyProxy server with no authentication (standard MyProxy has no support for authentication in addition to certificate-based authentication on upload) under any username, even when the download authorisation is using Kerberos or another link into their site authentication system. This requires users to upload their credentials under the correct username. This would not work in a Shibboleth environment as users will probably not know the username that ShibGrid uses for them to access the ShibGrid MyProxy as it must be unique across many organisations and so will not be their site user id (perhaps not even related to it). Given this consideration and the possibility of users maliciously uploading credentials to other people's accounts, we decided that the ShibGrid MyProxy server should also require Shibboleth authorisation to upload certificates.

Upload authorisation is complicated by the fact that Shibboleth credentials are obtained at websites, while certificate private keys should always remain on the user's machine. Figure 2 shows the architecture we have developed; as before, the initial steps are standard for Shibboleth.

1. User accesses the ShibGrid MyProxy upload page (SP) and the user's browser is redirected to the WAYF. This request contains the user's certificate DN.

2. The user chooses their home institution from those returned by the WAYF service.

3. The user's browser is re-directed to the correct IdP.

4. The user is authenticated by the authentication service of the IdP.

5. The IdP redirects the user's browser back to the ShibGrid MyProxy upload page. The signed SAML authentication assertions are passed in this redirect.

6. The ShibGrid MyProxy upload page calls out to the IdP's Attribute Authority for attributes about the user.

7. The username is extracted from the relevant attribute of the attribute assertion. The attribute assertion is placed within a signed delegation, with the user's DN specified as the delegate.

8. Using the ShibGrid MyProxy upload applet the user unlocks their certificate and uploads a proxy certificate to the ShibGrid MyProxy server using the username and password returned in Step 7. It is envisioned that users will be able to read their certificate either from their browser, or from two .pem files or from one

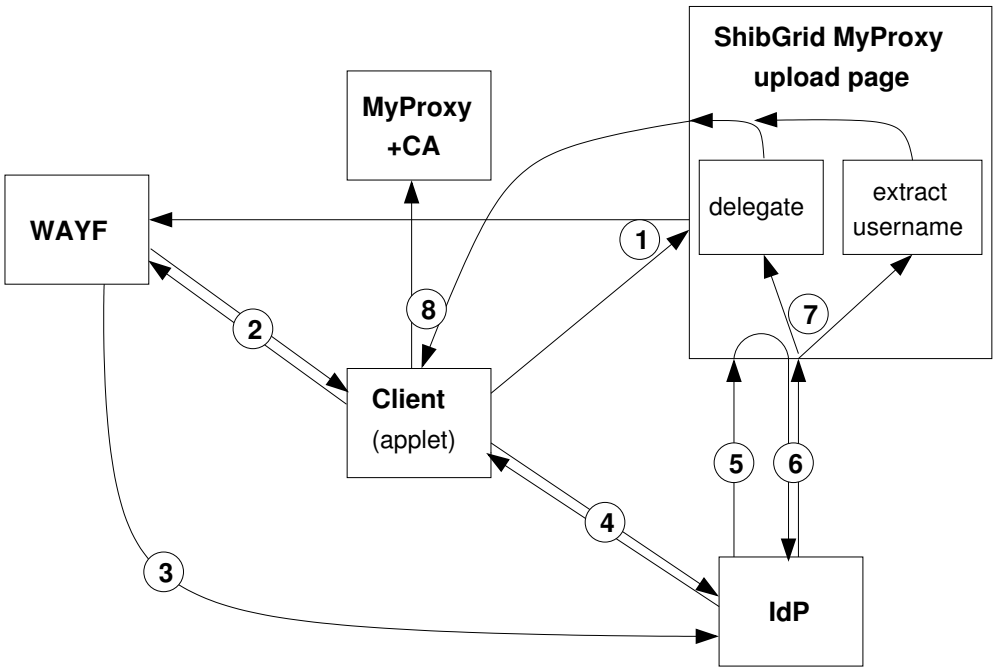

Figure 2. Uploading a proxy certificate to the MyProxy server in ShibGrid.

PKCS12 file. This tool also allows users to destroy their proxy on the ShibGrid MyProxy server.

4 Security Consideration

When the portal accesses the ShibGrid MyProxy server there are four operations which must take place before the MyProxy server will release a proxy or low-assurance certificate: user authentication, IdP authentication, portal authentication and portal authorisation. We do not assume that the portal is trusted by the MyProxy server as we foresee that the portal part of this work will be deployed into external project portals, not just portals that we manage.

User authentication is primarily performed by the IdPs and, within the Shibboleth federation, the portal and MyProxy server trusts the IdPs to authenticate users and make valid attribute assertions about them. In addition the MyProxy needs proof that the IdP has actually authenticated the user and made the attribute assertions that have been passed to the MyProxy server. This authentication (of the IdP) is primarily achieved through the requirement of signed attribute assertion.

The standard MyProxy server ensures authentication of the portal by requiring the use of GSI-authenticated connections in all communications. It provides authorisation of the portal through the use of server-scope and proxy-scope white-list schemes for proxy and low-assurance certificate download authorisation.

The ShibGrid MyProxy server also requires one more form of authorisation for portals: that the portal is mentioned as an audience for the attribute assertion. This authorisation reduces the exposure to badly configured download authorisation white-lists and the possibility of rogue portals stealing attribute assertions. This extra authorisation also has the explicit requirement that only servers in the Shibboleth federation can download proxies or low-assurance certificates.

A similar set of conditions are applied to the uploading of a proxy. In this case it is the user who contacts the ShibGrid MyProxy server, although a web server still needs to be present to act as the SP in the Shibboleth protocol. For the user to be authorised to use the resultant attribute assertion the web server then delegates its authorisation (to use the attribute assertion) to the user, by specifying the user's certificate's DN as the delegate.

The operations which therefore have to take place before proxy upload is allowed are: direct user authentication to the MyProxy server using the GSI-authenticated connection against their real certificate's DN; authorisation of the user in the MyProxy server's upload white-list; authentication of the user's Shibboleth identity with the IdP; and the authentication of the IdP (though the signed assertion) and the web server obtaining the Shibboleth attribute assertion on behalf of the user (through the signed delegation).

With this authentication/authorisation framework we seek to guarantee that a portal can obtain a proxy/low-

assurance certificate for a user if that user has been authenticated via Shibboleth to the portal within the last hour, as this is the lifetime of a Shibboleth attribute assertion. Note this is an improvement over the solution chosen by most portals in which users send their MyProxy pass-phrase to the portal and thus effectively give it access to their proxy on the MyProxy server until they next change the pass-phrase they use to protect proxies on the MyProxy server (which could be an extended period of time). In ShibGrid, or with normal portals, we can never mitigate against a portal using a user's certificate maliciously once the user has been authenticated.

For secrecy, all communication within the ShibGrid framework, especially those transmitting credentials (Shibboleth and X.509), takes place over HTTPS unless specified otherwise in the Shibboleth protocol.

5 Implementation Details

While the ShibGrid project aimed to reuse existing components, each of the components used required work to fully integrate into ShibGrid as they were designed for Shibboleth or the Grid Security Infrastructure, but not both. In this section we describe how the various components were changed to give one integrated infrastructure and the components that had to be developed from scratch.

5.1 ShibGrid MyProxy Server

The vast majority of work involved changes to the MyProxy server code. The security checks on Shibboleth attribute assertions from Section 4 are all implemented in the MyProxy server. The method used by ShibGrid for Shibboleth authorisation to MyProxy, sending an encrypted authentication token as the password and verifying this in the password checking code, was inspired by the MyProxy support for the Pubcookie [14] authentication framework. In addition to the checks from Section 4, there are thorough checks on the structure and validity of attribute assertions and delegations.

Where the portal's identity must be checked against an attribute assertion's audience, the MyProxy server searches through the federation's metadata for an SP entry with a matching entityID. The expected DN of the portal is extracted from the key information stored for that entity and this is compared against the DN of the certificate used to either authenticate the GSI connection to MyProxy or sign the attribute assertion delegation.

In all cases the username must match the attribute in the attribute assertion which is defined to map to the username (by default *urn:mace:dir:attribute-def:eduPersonPrincipalName*). If no low-assurance certificate is being generated this is the only attribute required in the attribute assertion.

If a low-assurance certificate needs to be generated then a DN pattern must be specified in the configuration of the MyProxy server and the attributes used to make up the DN must be present in the attribute assertion. A number of checks are made to ensure the correctness of the resulting DN. The default DN scheme for ShibGrid is:

/C=UK/O=eScienceMyProxy
/OU=*urn:mace:dir:attribute-def:o*
/UID=*urn:mace:dir:attribute-def:uid*
/CN=*urn:mace:dir:attribute-def:givenName*
urn:mace:dir:attribute-def:sn

5.2 Portals

The initial implementation of a ShibGrid-enabled portal is based on the NGS portal. There are two relevant modules: the initial login module which is based on the Java Authentication and Authorisation Service (JAAS) standard [20] and the proxy renewal module which is based upon the JSR 168 portlet specification [1]. As these two modules are standards-based it is hoped that they should be able to be deployed in other portals and portal frameworks with little or no effort.

Another local Shibboleth integration project, the Shibboleth-aware Portals and Information Environments (SPIE) project[4], has developed a Shibboleth JAAS Module [10] which can be used to authenticate users to a (non-Grid) portal via Shibboleth. This is used to provide the Shibboleth protocol support and passes to our modules the signed attribute assertion and the individual attributes. The standard Java Globus CoG libraries are then used to retrieve the user's Grid credential. The username is picked up from the attributes and the entire signed attribute assertion is used as the password.

5.3 ShibGrid Tools

To provide users with the ability to upload real certificate proxies to the ShibGrid MyProxy server and to allow users to download proxies/low-assurance certificates to their machines for non-portal Grid access, we have also developed special upload and download tools. In both cases a Java servlet and a Java applet work in tandem to provide the required functionality because obtaining Shibboleth credentials requires a presence on a web server; and access to the user's certificate (for upload) or proxy storage location on a local disk (for download) requires code running on the user's machine.

The upload tool works largely as described as in Section 3.1, except that initially the user logs on (over HTTPS)

[4] *http://spie.oucs.ox.ac.uk/*

to the upload tool website and at this time Shibboleth authentication takes place and the applet is downloaded (with the user's username as one of its arguments). Later when the user has chosen and unlocked their certificate, so their DN is known, the applet calls back to the servlet with this DN (over HTTPS) and receives the delegated attribute assertion. As both connections are to the same servlet and over HTTPS, there is no need for the user to be re-authenticated as Java picks up the relevant cookie and the *shibd daemon* running on the web-server remembers the attribute assertion. Using the returned delegated attribute assertion the applet uploads a proxy of the user's certificate to the MyProxy server via a GSI-authenticated (secure) connection.

The upload tools also allow proxies to be destroyed. In this case the GSI-authenticated connection provides sufficient authorisation to allow the proxy to be destroyed (after checking that the DN of the proxy and the DN of the certificate used to authenticate the connection match). In this case the tool does not call back to the servlet for a delegated attribute assertion. This is why the username is passed when the applet is downloaded.

The download tool works on a similar model. The user must login using Shibboleth to access the page containing the download applet. The applet has one option: the length of the downloaded proxy and this value is passed back to the servlet in a call-back, in the same style as for the upload tool. The servlet then calls the MyProxy server (in the same way as described for the portal) to obtain a proxy or low-assurance certificate for the user. This is returned to the applet which saves it on the user's machine. Once again, it is important that all these interactions take place over HTTPS to protect the certificates from being snooped.

6 Future Work

Within the ShibGrid project, our aim is to deploy a production environment for Grid authentication via Shibboleth. Therefore, much of the future work is concerned with the evaluation, testing and hardening of our prototype followed by deployment and on-going support of the final infrastructure.

Other further work will include support for new Shibboleth/SAML developments; for instance, versions 2.0 and 2.1 of Shibboleth which are currently being developed. In addition, innovations on the horizon such as SAML/Shibboleth Delegation [6, 24], the SAML Protocol Extension for Third-Party Requests [7] and Shibboleth support for non-browser profiles may be able to be employed to replace/supplement parts of the ShibGrid framework with standards-based protocols.

7 Conclusion

This paper has presented our work undertaken to integrate the future UK national Shibboleth infrastructure with the Grid Security Infrastructure, for access to the NGS. The ShibGrid architecture that has been presented provides users access to Grid resources via their own project portal, the NGS portal or standard command-line tools. ShibGrid only requires users to provide authentication via Shibboleth and allows users to easily use their real UK e-Science certificates if they have them. ShibGrid also does not require any changes to the standard Shibboleth infrastructure or the NGS security infrastructure. Therefore, ShibGrid fulfils all the requirements from Section 1.2.

Acknowledgements

We would like to thank JISC for funding the ShibGrid project.

References

[1] A. Abdelnur and S. Hepper. JSR-000168 portlet specification. Java Specification Request (JSR) 000168, October 2003.

[2] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, Á. Frohner, A. Gianoli, K. Lrentey, and F. Spataro. VOMS, an authorization system for virtual organizations. *Lecture notes in computer science*, (2970):33–40, 2004. Grid Computing, First European Across Grids Conference, Santiago de Compostela, Spain, February 13-14, 2003.

[3] T. Barton, J. Basney, T. Freeman, T. Scavo, F. Siebenlist, V. Welch, R. Ananthakrishnan, B. Baker, and K. Keahey. Identity federation and attribute-based authorization through the Globus toolkit, Shibboleth, GridShib, and MyProxy. In *Proceedings of the 5th Annual PKI R&D Workshop*, October 2005. (To appear April 2006).

[4] J. Basney, M. Humphrey, and V. Welch. The MyProxy online credential repository. *Software- Practice & Experience*, 35(9):801–816, 2005.

[5] K. Bhatia, A. Lin, B. Link, K. Mueller, and S. Chandra. Geon/telescience security infrastructure. Technical Report SDSC TR-2004-5, San Diego Supercomputing Center, 2004.

[6] S. Cantor. SAML 2.0 Single Sign-On with constrained delegation. Internet 2 document: draft-cantor-saml-sso-delegation-01, October 2005. Available at *http://shibboleth.internet2.edu/docs/draft-cantor-saml-sso-delegation-01.pdf*.

[7] S. Cantor. SAML protocol extension for third-party requests. OASIS Committee Draft sstc-saml-protocol-ext-thirdparty-cd-01. Available at *http://www.oasis-open.org/specs/index.php#samlv1.1*, March 2006.

[8] CCITT recommendation X.509: The directory - authentication framework. CCITT Blue Book, volume VIII, pages 48–81, 1988.

[9] D. Chadwick, W. Novikoc, and A. Otenko. GridShib and PERMIS integration. In *TERENA Networking Conference 2006*, May 2006.

[10] C. Fernau and F. Pinto. Shibboleth JAAS module architecture. Available at *http://spie.oucs.ox.ac.uk/Wiki.jsp?page=JAASmoduleArch*, February 2006.

[11] J. Jensen, D. Spence, and M. Viljoen. Grid single sign-on in CCLRC. In *to appear in the proceedings of the 2006 UK e-Science All Hands Meeting*, September 2006.

[12] JISC. Shibboleth: Connecting people and resources; briefing paper version 2. Available as *http://www.jisc.ac.uk/index.cfm?name=pub_shibboleth*, February 2006.

[13] O. Kornievskaia, P. Honeyman, B. Doster, and K. Coffman. Kerberized credential translation: A solution to web access control. In *Proceedings of the 10th USENIX Security Symposium, Washington, D.C., USA*, pages 235–249, August 2001.

[14] J. Martin, J. Basney, and M. Humphrey. Extending existing campus trust relationships to the Grid through the integration of Pubcookie and MyProxy. In *2005 International Conference on Computational Science (ICCS 2005), Emory University, Atlanta, GA, USA*, May 2005.

[15] OASIS Security Services Technical Committee. Security assertion markup language (SAMl) v1.1. OASIS Standard 200308. Available at *http://www.oasis-open.org/specs/index.php#samlv1.1*, August 2003.

[16] D. Orrell. Athens next generation core architecture. Eduserv Athens Whitepaper. Available at *http://www.athensams.net/local_auth/shibboleth/ng_core_architecture.pdf.*, October 2004.

[17] R. Petkus. One-time-password integration at BNL. Available as *http://hepix.caspur.it/spring2006/TALKS/4apr.petkus.otpbnl.pdf*, April 2006. (Talk given at HEPiX conference).

[18] T. Scavo and S. Cantor. Shibboleth architecture technical overview. Internet 2 document: draft-mace-shibboleth-tech-overview-02, June 2005. Available at *http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf*.

[19] SHEBANGS (Shibboleth Enabled Bridge to Access the National Grid Service). Details at *http://www.sve.man.ac.uk/Research/AtoZ/SHEBANGS*, June 2006.

[20] Sun Microsystems. Java authentication and authorization service (JAAS). See *http://java.sun.com/products/jaas/*.

[21] S. Tuecke, D. Engert, I. Foster, M. Thompson, L. Pearlman, and C. Kesselman. Internet X.509 public key infrastructure proxy certificate profile. Request for Comments (RFC) 3820, June 2004.

[22] G. von Laszewski, J. Gawor, P. Lane, N. Rehn, M. Russell, and K. Jackson. Features of the Java commodity Grid kit. *Concurrency and Computation: Practice and Experience*, 14:1045–1055, 2002.

[23] E. Vullings and J. Dalziel. Meta access management system. Technical report, Macquarie University, June 2005.

[24] J. Wang, D. Del Vecchio, and M. Humphrey. Extending the security assertion markup language to support delegation for web services and Grid services. In *Proceedings of the 2005 IEEE International Conference on Web Services (ICWS 2005), Orlando, Florida, USA*, July 2005.

[25] V. Welch. Report for the GGF 16 BoF for Grid developers and deployers leveraging Shibboleth. GWD-informational, Global Grid Forum, March 2006.

[26] V. Welch, T. Barton, K. Keahey, and F. Siebenlist. Attributes, anonymity, and access: Shibboleth and Globus integration to facilitate Grid collaboration. In *Proceedings of the 4th Annual PKI R&D Workshop*, April 2005.

[27] V. Welch, F. Siebenlist, I. T. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. Security for Grid services. In *12th International Symposium on High-Performance Distributed Computing (HPDC-12), Seattle, Washington, USA*, pages 48–57, 2003.