The GLASS Project: Supporting Secure Shibboleth-based Single Sign-On to Campus Resources

J. Watt, R.O. Sinnott, J. Jiang

National e-Science Centre, University of Glasgow
j.watt@nesc.gla.ac.uk

Abstract

Higher and Further education institutions in the UK are in the process of migrating their IT infrastructures to exploit Shibboleth technologies for federated access management. Ease of use and secure access are paramount to the successful uptake of these technologies, both from the end user and system administrator perspective. The JISC-funded GLASS project is a one-year project investigating the use of Shibboleth to support single sign-on to a variety of campus resources at the University of Glasgow including browser-based email access; the Moodle online virtual learning environment; the WebSURF online student records facility, and a network filestore browser. This paper describes the implementation issues and experiences gained in rolling out the Shibboleth technologies to support federated access management .

1. Introduction

The academic community in the UK is in the first stages of deploying a nationwide federated access management infrastructure to support local authentication mechanisms. Part of the motivation for this is driven by the desire to simplify collaboration between institutions, and also to make the end user experience easier for non-technical end users (both staff and students). In general, if two institutions wish to collaborate, any computing resources needed at either location will be managed by their own systems administrators. Typically, this results in students or staff being allocated remote accounts on all non-local resources. As the number of collaborating institutions increases, so does the number of separate accounts the user will hold. The result of this is a clutch of different usernames and passwords which inevitably will be misplaced or confused. Further, if a local member of an institution has their privileges or contract revoked, there is no immediate way for any collaborating remote institutions to know if this user is still valid.

The Shibboleth [1] federated access management system aims to rectify this issue by providing an infrastructure where the job of authenticating and providing information about a user is the sole responsibility of the user's home institution, and this single authentication step will be recognised across all resources within the Shibboleth federation. Individual resources will still be able to exercise full control over access to their systems, normally by the presentation of specific roles by Shibboleth, but the step of confirming a user's identity will have already been established by the home institution. In addition, Shibboleth supports the notion of single sign-on, whereby users will only have to enter their home username and password once each session to access and use multiple different resources in the federation.

The Joint Information Systems Committee (JISC) is steering the academic community in the United Kingdom towards federated access management over the next few years. The MATU (Middleware Assisted Take-Up) [2] service will be coordinating the changeover from the prevalent Athens-style authentication [3] to Shibboleth based on the SDSS Federation [4] (of which Glasgow is a member – currently operating two Identity Providers and four Service Providers).

The one-year JISC-funded GLASS (GLASgow early adoption of Shibboleth) [5] project aims to investigate how the centralised directory structure which is being rolled out at the University of Glasgow for unified account management can be utilised in a Shibboleth environment. The university has adopted the Novell NSure account management system [6], which provides the infrastructure necessary to support easy creation/deletion of accounts, unique IDs, a secure audit trail and the propagation of this information to any related user databases in real time. The project has investigated how Shibboleth can utilise the NSure directory information to allow one-stop authentication from the primary staff/student information directory, and also how any access control attributes required may be integrated into this system.

It should be noted that none of the applications within this paper rely on any non-identity based user

attributes (such as role or an eduPerson [7] type attribute) for access control, so no changes to the NSure directory or schema are required. However, this kind of functionality is required to enable fine-grained access control to protected resources. We will detail how we have tackled this problem through the use of multiple attribute authorities.

The University has defined 4 initial key services which would benefit from single sign-on:

- 'Moodle' online virtual learning environment
- 'WebSURF' online personal record update facility
- 'WebMail' browser based student email
- A Network filestore browser

The implementation for the final item, the network filestore, is incomplete at the time of writing, so this paper will focus on the issues encountered in Shibboleth-enabling the Moodle and WebSURF services which already exist, and also an open-source version of Novell Webmail called Hula [8]

2. Shibboleth

Shibboleth is an Internet2 Middleware initiative to establish a federated authentication mechanism based on SAML [9]. The SAML profile calls for the creation of several trusted entities who are responsible for exchanging information about users, and these entities communicated through encrypted channels, thus preserving information privacy. An Identity Provider (IdP), sometimes referred to as an 'Origin' is a body which releases digital credentials about its users. These credentials are typically released as multi-valued attributes, containing any information about the user which could be used to make an access control decision. The attributes may consist of direct, 'real-world' attributes such as Surname, Postal Address, Telephone number, or more abstract credentials like role, privilege or licences. There is no generic authentication method required for use with Shibboleth rather it is intended to support your local authentication system. Hence, systems like CAS [10] and Pubcookie [11] are popular, whereas in our infrastructure, we make use of the mod_auth_ldap [12] plug-in for Apache [13] which allows the login to use an LDAP directory. The IdP is responsible for defining the subset of user attributes which will be released to the federation of trusted sites (called an Attribute Release Policy or ARP), but has no policy on what should be done with them. A Service Provider (SP) is therefore defined, who acts as the resource or service which is to be Shibboleth protected. The Service Provider receives attributes from the IdP according to its own Attribute Acceptance Policy (AAP) and exposes them internally to applications running on the SP as HTTP headers. Any application may use the information encoded in these headers to make its access control decisions. A further (optional) entity is defined, called a Where Are You From (WAYF) service, which is provided by the UK SDSS federation. A WAYF normally takes the form of an intermediate web page listing all of the subscribed institutions which the user selects their home institution from.

A normal invocation of a Shibboleth-protected resource proceeds as follows. The user types the URL of the desired resource into their browser, at this point the SP will detect that the user is requesting a Shibboleth-protected resource and will forward the user to the WAYF service. The user selects their home institution from the drop-down list, and then the user is forwarded to their IdP for authentication utilising their local username/password.

Our implementation of Shibboleth uses the mod_auth_ldap Apache module to communicate with the central campus LDAP server for authentication and attribute retrieval. This module by default does the query and reception in non-encrypted cleartext which exposes the link between the Identity Provider and the campus LDAP as exploitable by packet sniffing software. There are two solutions being considered, one involves setting up a secure SSL connection between these two nodes which has proven to be problematic due to the lack of a clear defined standard for SSL to LDAP, but also NSure has stricter requirements on how external connections may and may not be made to its directories. The second solution involves migration of the Identity Provider to sit behind the same network switch as the campus LDAP. Then the fact that the IdP-LDAP connection is unencrypted is no longer an issue as this part of the network would be invisible to the outside world. Obviously the first option offers greater flexibility so efforts are ongoing in this area to supply a location-independent Identity Provider.

3. Moodle

Moodle is an open-source course management system which allows educators and teachers to create online virtual learning environments [14]. Moodle now has over 100,000 registered users in over 150 countries and is being adopted by educational institutions worldwide. As is the case in Glasgow, most institutions end up deploying multiple Moodles on a departmental basis, each controlling the content of their individual sites. In the current regime, a student who is logged into Moodle in one department will have to log in again if they change to a different Moodle site, the irony being that the separate Moodles will probably be extracting their authentication details from the very

same central directory. Moodle also transmits and receives its login information in cleartext, which carries heavy security implications if the resource is being accessed from outside the campus.

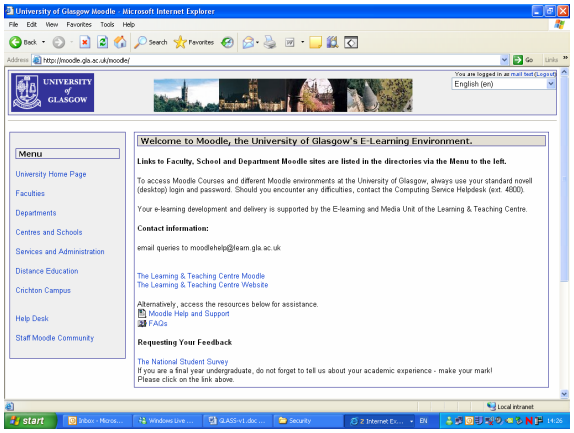

Figure 1: The University of Glasgow central campus Moodle

Moodle ships with login modules supporting various types of authentication mechanism, and a Shibboleth module is supplied with the standard build. The module is configured within the Moodle system admin page, where a configuration page allows various fields that Moodle can use (Surname, UID, email address etc.) to be mapped to HTTP headers that Shibboleth uses to expose its attributes. In Glasgow, the central Moodle uses five standard LDAP attributes to create a user session: uid (or cn); givenName; first name; surname and email address. The Moodle session username is created from the uid attribute as this is unique across campus. The other information can be used by Moodle in its accounting/audit feature.

The standard Shibboleth configuration contains a global Apache directive which states:

allowOverride None

This means that Apache will not consider parsing .htaccess files if the Shibboleth login fails. This is desirable for normal operation, but Moodle uses a .htaccess file to provide user attributes if a Shibboleth session already exists. Hence this macro must be set to 'all' to allow single sign-on to be achieved. Since this change need only be applied to the Moodle directory, this directive does not negatively impact upon any other Shibboleth protected services on that resource.

There is a further complication in that if the user had logged into Shibboleth from a different page than Moodle, the first time they visit Moodle they are pointed to a page which allows them to choose between the normal generic login and a Shibboleth login. This problem is alleviated if the Moodle front page is set to a "Shibboleth only" authentication, but this may not be desirable in the production environment. It should be noted that after the initial login to the Moodle, the Shibboleth single sign-on functions correctly. We are investigating ways in which an initial login to Moodle can detect whether or not a Shibboleth session is active and create the appropriate user session.

4. Email

The NSure account system offers an email service (NetMail) [15] viewable via a web browser interface. The NetMail service is deployed in its own web container, which would make Shibboleth enabling this application almost impossible without specific code changes from Novell. However, an open-source UNIX version of the NetMail service called Hula is available which aims to provide a version of NetMail configurable for non-Windows platforms, and more importantly can be run inside a separate web server application like Apache. Hula is called from the mod_python Apache plug-in module, and provides the same functionality as the proprietary NSure system.
There was no way to tie this work into the live WebMail system at present, so this effort remains documented to speed up any future implementation.

5. WebSURF

WebSURF [16] is a browser-based application which allows students and staff to view and update central student records at any time and from any internet-enabled location. Students can use WebSURF to register for courses, check and maintain personal information and view their examination results. University staff can also use the system to view student records or update faculty information. The WebSURF software was written by the Management Information Services group at the University of Glasgow.

In the current live implementation, WebSURF is a J2EE application which runs in a JBoss container utilising JAAS for login security. For students, an LDAP login module is used for communication with the central campus LDAP server. For staff, a remote EJB bean queries various staff authentication mechanisms which require pre-registration.

For interfacing with Shibboleth, the generic JAAS module in JBoss was replaced with the SPIE JAAS module from the University of Oxford [17]. The SPIE JAAS Shibboleth module allows JBoss to pass security context information received from Shibboleth to the WebSURF application, or indeed any application running in JBoss. SPIE JAAS allows Shibboleth attributes to be accessed through standard JSP Servlet API calls. As such, it is quite

easy to integrate with other existing applications with little changes to code. Shibboleth authenticates the user (if not already done so) and receives their attributes, passing this information to the SPIE JAAS Login module through the mod_jk2 connector under the 'Shib-Attributes' HTTP header. The SPIE Login module is responsible for extracting the user ID (in our case, the 'uid' attribute) from the header, and passes the other extracted attributes to the SPIE JAAS Module. The SPIE JAAS Module creates a random key for the passed attribute set and passes the key back to the SPIE Login module for it to use as a temporary password for the session. Once formed, the password and username extracted from the attribute assertion are presented to the original application login page as if this information was input manually. The SPIE JAAS module can also perform rudimentary access control based on any attributes that have been defined to be used as a user role, although this is not strictly necessary for WebSURF. How the SPIE JAAS functionality has been integrated within the framework of our WebSURF system is shown in Figure 1.

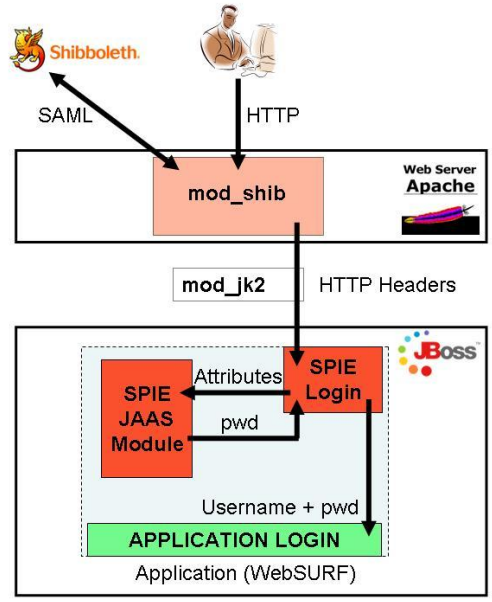

Figure 2: The SPIE JAAS Login module operating in a Shibboleth-protected JBoss application server. The SAML exchange with the Shibboleth federation will only be done if a session doesn't already exist.

A test application was deployed to mimic the WebSURF deployment environment as closely as possible. Using information retrieved by Shibboleth from the central campus LDAP, our application was able to utilise the unique user ID attribute to form an application session within JBoss. The application was able to use already established security contexts from Shibboleth, thereby fulfilling the single sign-on requirement. Since WebSURF takes its login information from the generic JAAS module, the SPIE JAAS module will provide exactly the same information so no modification of the production WebSURF code will be necessary.

However, the databases and resources utilised by the production version of WebSURF are too sensitive to allow deployment of the tool as it stands, and the tool is not designed for any more than one single deployment at the University of Glasgow. We negotiated a partial rewrite of the WebSURF functionality to demonstrate a test deployment which looks and operates like a normal WebSURF session, retaining an identical login implementation, but without compromising highly sensitive information sources.

6. Storing Attributes

By default, Shibboleth is configured with a single LDAP connector which it uses to retrieve user attributes. Since it is unlikely that a single attribute authority will be supplying all information about a user within an institution, there is provision in the Shibboleth configuration for the use of multiple data connectors so attributes from many sources may be located. A slightly different motivation for this may be found in the prevalent use of existing authentication mechanisms across campus – in our example, the Novell NSure system. With these established solutions comes a certain reluctance on the institution's part to allow a new piece of software to start demanding that particular attributes or schemas may be appended or at worst, replaced. For example, in our current use case, the central campus LDAP directory was queried by the mod_auth_ldap Apache plug in to confirm the identity asserted by the username/password combination given by the user when prompted. However, since the central LDAP has not adopted the eduPerson schema, there would exist no placeholder for the user attributes within that directory (as our portal pages require the eduPersonEntitlement attribute to provide user roles in the organisation).

However, using the principal identifier used to parse the central LDAP, we can search other LDAPs (AAs) for attributes pertaining to that user's rights as long as the user entries contain this one piece of linking information (uid) to associate the user entries across multiple LDAPs, possibly with distinct Distinguished Names (DNs). The University of Glasgow has, as part of its unified account mechanism, guaranteed that the 'uid' attribute is unique for all users across campus, therefore this attribute may be confidently used to link user entries in separate LDAP servers to uniquely identify users. This provision allows departments within Universities to set up their own Attribute Authorities (LDAP servers for example) to issue roles to their employees, adopting completely separate LDAP

namespaces if they so wish, but providing the 'uid' attribute is the same as the central campus LDAP, this AA may be parsed by Shibboleth for user attributes.

There are several precautionary measures that need to be taken however when configuring the Shibboleth JNDI connectors (which make the connection to LDAP). By default, a connector will return an error if a user is not found in a directory. While this is a desirable feature for robustness, it has the effect that a user would need an entry in every LDAP directory that the IdP searches. An error in one directory causes a global error that returns no attributes, even if a valid set is located in another LDAP. The solution to this is to use a configuration parameter in the JNDI connector definition of

propagateErrors="false" noResultIsError="false"

This has the effect of ignoring errors occurring from a user's lack of entries in certain directories.

As an illustration (Figure 3), consider the situation where a clinical medicine researcher may wish to collaborate with a department hosting geographical databases in order to link patient illness to census data, for example. He already holds an attribute in his home department (*Clin_projectY*) but requires roles to access the geography resources. In this case, the researcher would request a particular role (*Geo_project1*) from the geographical department, who would create a new user within their own LDAP representing the researcher, remembering to issue the user the same unique 'uid' as the central LDAP, and allocate the appropriate role there. This allows the collaborating department to fully control the attributes pertaining to access of *its own* services, yet if this user is removed from the University, it will not be critical for the department to remove this user's attribute as the user will fail the very first authentication step and the user's attributes would never get requested.

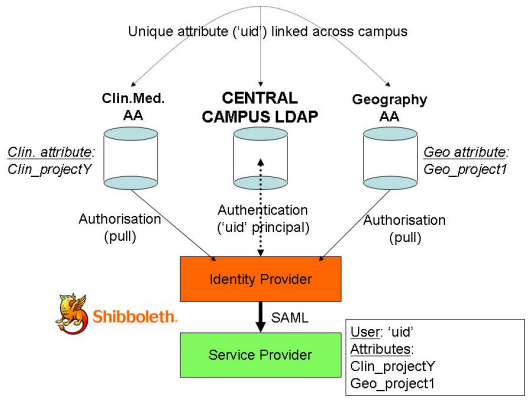

Figure 3: Distributed attributes from multiple campus Attribute Authorities, linked by a common institutionally unique 'uid'.

In contrast to the above 'distributed attributes' model, we have investigated in previous projects the idea of Dynamic Delegation. Put simply, this allows departments to collaborate by issuing roles and the right to issue roles to its fellow collaborators directly, rather than retaining them locally. Using the previous clinical/geographic example, the researcher would request an attribute to allow them to access the geographical database, and the geography department would issue the role directly to the researcher's LDAP server in clinical medicine. Therefore roles for many departments would be stored in a single placeholder representing the user's home department. See Figure 4 for a schematic of this interaction.

Since this model involves surrendering some control over your own attributes (as they are being stored externally) there must be a secure infrastructure to allow these roles to propagate in a controlled manner. The PERMIS Delegation Issuing Service (DIS) [18] is a web service which allows such a model to be implemented, but the attributes are stored in LDAP in the form of X.509 Attribute Certificates, which are digitally signed offering another layer of security. ACs are issued according to a local policy, so initially a collaborating department would know nothing of an external department's attribute set. The external department therefore issues a Role Allocation Policy to the home department, allowing the external department's role to be issued in the user's home LDAP while obeying any restrictions that the external department wishes to enforce.

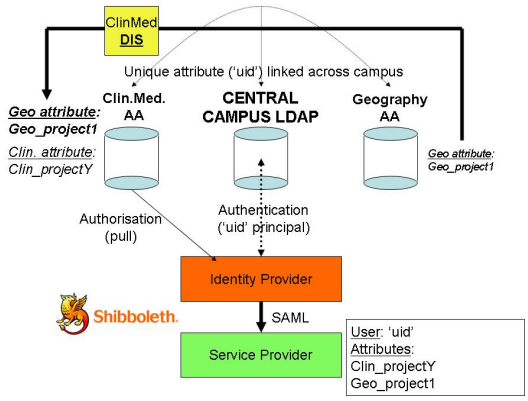

Figure 4: Use of the Delegation Issuing Service to allocate external department's roles to the user's home department. This allows a single Attribute Authority to contain all the information regarding a user's privileges across campus.

The DyVOSE (Dynamic Virtual Organisations in e-Science Education) [19] project investigated a use case involving PERMIS-protected access to databases located in two different locations, enforced by two separately authored policies. The two sites agreed to exchange Role Allocation Policies which allowed the remote institution to allocate roles required for access to its services to a user based at their home institution. The Attribute Certificates themselves are issued by the DIS Web Service on behalf of privileged users (systems administrators or directors). This means that these users do not require a signing key pair to issue ACs to their subordinates, the advantage gained being that subsequent revocation of any user's key pair at any point in a signing chain will not invalidate any legally issued end-user certificates that this revoked user had handed out. It should be noted that all certificate usage and interaction is done in the background, and no certificates need to be handled by the user. Also, because the roles are stored in the form of signed digital certificates, it means that the certificate needs to be decoded by openssl, for instance, in order to extract an attribute string. This model is appropriate for fine-grained access control, but is unlikely to be adopted by an institutional-level security system. It would be expected that individual departments/projects would make the decision to adopt this technology.

7. Scoping Attributes

One of the immediate issues with operating Shibboleth in a large scale federation such as the UK Access Management Federation [20] is that of implied trust. When a new Shibboleth Service Provider is implemented, the default behaviour is to accept the authentication assertions issued by any Identity Provider in the federation. Typically, coarse grained access control can be enforced using the attributes passed by Shibboleth, however there could arise a situation where two Service Providers require the same value of a generic attribute (such as eduPersonEntitlement) purely coincidentally. For example, Site A requires an eduPersonEntitlement attribute of 'full_access' to gain access to their biological data, whereas Site B requires the same attribute to access its particle physics data. Even if these sites are unaware of each other's existence, if a user authenticates at Site A and then finds the URL of the Site B service, their authentication assertion will be trusted anyway, but their authorisation token will be correct to access a system they are not intended to access, purely because the same attribute value has been presented.

A solution to this is to 'scope' the user attributes provided by Shibboleth. Using this concept, an Identity Provider may be configured to only release a certain attribute to a specific Service Provider (or set of Service Providers). In a similar way, Service Providers may be configured to only accept attributes from a particular Identity Provider, so the situation described above is avoided as Site A will not be able to assert Site B's eduPersonEntitlement attribute. Related to location scope is attribute value scope. Using this allows not only the location of an attribute to be filtered, but also its value. This is useful as a user may have many attributes relating to several projects, some of which may not be desirable to release to just anyone. Using value scope, a Service Provider can only accept the subset of attributes that it is interested in and reject the rest. This rejection happens within the Shibboleth Apache module so the sensitive attributes can never get through to the service itself.

8. Conclusions

We have demonstrated that it is possible to integrate Shibboleth with a central institution authentication infrastructure without impacting on their operations. We successfully integrated Moodle, a bespoke student WebSURF application, and our own Grid resources with the University's unified account management system. We have adopted an infrastructure which allows the campus LDAP to be the authoritative source of user authentication information, yet through the use of multiple attribute authorities, we have shown it would be possible for individual departments or groups within the University to issue attributes and roles to their users for access to their own systems or for access to resources provided by collaborators. This work is being used in several other projects at the National e-Science Centre at the University of Glasgow including the EPSRC pilot project "Meeting the Design Challenges of nanoCMOS Electronics"

project [21]. This and other projects are providing real life use cases testing this security infrastructure, and we expect will demonstrate scalability not just at institutional level but across national level Shibboleth-based access management federations.

9. Acknowledgements

Many thanks to Dr. Christian Fernau of the University of Oxford for his help in configuring their SPIE JAAS module, and also to Lukas Haemmerle of SWITCH, Switzerland for his help with the Moodle Shibboleth authentication module. Thanks to Robert Stewart of the Management Information Service at the University of Glasgow for information and executables for the WebSURF application, and to Dave Anderson of IT Services for his help with accessing the central campus LDAP server at the University of Glasgow. Thanks are also due to Scott Cantor, Nate Klingenstein and Chad Le Joie for their help through the Shibboleth support mailing list. GLASS is funded from a grant from JISC whom we also acknowledge here.

10. References

[1] Internet2 Shibboleth Technology, http://shibboleth.internet2.edu
[2] eduServe Middleware Assisted Take-Up Service (MATU) http://www.matu.ac.uk
[3] eduServe Athens for Education, http://www.athens.ac.uk/
[4] SDSS Federation http://www.sdss.ac.uk
[5] JISC Funded Glasgow University Early Adoption of Shibboleth (GLASS) Project http://www.nesc.ac.uk/hub/projects/glass
[6] NSure, Novell Identity Manager http://www.novell.com/products/identitymanager/
[7] eduPerson Specification http://www.educause.edu/eduperson
[8] The Hula Project, Novell, http://www.hula-project.org
[9] Security Assertion Markup Language (SAML) v2, March 2005 http://www.oasis-open.org/specs/index.php
[10] Community Authorisation Server http://www.lesc.ic.ac.uk/projects/cas.html
[11] Pubcookie: Open-source software for intra-institutional web authentication, http://www.pubcookie.org/
[12] mod_authz_ldap: X509 Certificates and LDAP http://authzldap.othello.ch/
[13] Apache Web Server, http://www.apache.org/
[14] Moodle – A Free Open Source Course Management System for Online Learning http://moodle.org
[15] Novell NetMail http://www.novell.com/products/netmail
[16] WebSURF Student Records Update Service, University of Glasgow (Robert Stewart, Management Information Services), http://www.websurf.gla.ac.uk
[17] SPIE JAAS Module Architecture, SPIE Project, University of Oxford, http://spie.oucs.ox.ac.uk/Wiki.jsp?page=JAASModuleArch
[18] 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.
[19] J.Watt, J.Koetsier, A.Stell, R.O.Sinnott, "DyVOSE Project: Experiences in Applying Privilege Management Infrastructures" in UK e-Science All Hands Meeting Proceedings, Nottingham, September 2006
[20] The UK Access Management Federation for Education and Research, http://www.ukfederation.org.uk
[21] "Meeting the Design Challenges of nanoCMOS Electronics" EPSRC Pilot Project http://www.nanocmos.ac.uk

Author/s:
Watt, J; Sinnott, R; Jiang, J

Title:
The GLASS Project: Supporting Secure Shibboleth-based Single Sign-On to Campus Resources

Date:
2007

Citation:
Watt, J., Sinnott, R. & Jiang, J. (2007). The GLASS Project: Supporting Secure Shibboleth-based Single Sign-On to Campus Resources. UK e-Science All Hands Meeting, 2007. Proceedings, UK e-Science All Hands Meeting.

Publication Status:
Published