Secure Federated Authentication and Authorisation to GRID Portal Applications using SAML and XACML

Erik Vullings and James Dalziel

MELCOE, Macquarie University, Sydney, NSW 2109, Australia
E-mail: {erik.vullings, james}@melcoe.mq.edu.au, Web: federation.org.au

Markus Buchhorn

Australian National University and APAC and GrangeNet, Canberra, Australia
E-mail: Markus.Buchhorn@anu.edu.au

Internationally, the need for federated Identity & Access Management continues to grow, as it allows users to get Single Sign-On access to external resources (a.k.a. Service Providers) using their home account and some attributes that are being released securely by their home organization (a.k.a. Identity Providers). In other words, it solves the problem of service providers needing to create and maintain accounts for external users who they may not know. Current implementations seem to either rely on SAML, the Security Assertion Markup Language, or PKI, where the latter is mainly popular for Grid services. However, there are some trends towards convergence, for example, the recent release of the Globus toolkit is SAML and XACML aware, and GridShib is another project that uses PKI for authentication and SAML for passing attributes for authorisation. Still, these projects do not use the full potential of SAML and XACML, so this paper focuses on a scalable approach using distributed attribute authorities to access Grid services.

Keywords – Federated Identity & Access Management, Virtual Organizations, Shibboleth, SAML, XACML, eResearch Toolkits, Virtual Research Environment (VRE), Grid

ACM Classification: H.3.5. (Online Information Services)

INTRODUCTION

In many cases, advanced research can only be done using very expensive equipment, like cyclotrons, Scanning Electron Microscopes (SEM), High Performance Computing (HPC) centres, or virtual observatories. Because of their nature, these resources need to be optimally used on the one hand and carefully protected against misuse on the other hand. To satisfy the former criterion, the resources are often GRID enabled, meaning that they theoretically can be accessed by many computers on the internet. The latter criterion is fulfilled by protecting access using PKI (Public Key Encryption) user certificates, so only a limited set of trusted users can actually use the resource. The GLOBUS toolkit (Foster, 2005) is an example of a software product that meets both of these requirements and is implemented world-wide.

The main problem with this approach is the inflexibility of the PKI infrastructure that needs to be in place for this to work. PKI, although already an old concept, has never really taken off beyond server-side protection due to a number of inherent limitations (Gutmann, 2001):

Manuscript received: 13 December 2006
Communicating Editor: Professor Bill Appelbe

- **Key lookup:** When I sign-up a user to access my GRID application, how do I get his true PKI certificate? Currently, they are mostly mailed around.
- **Enrolment:** How does a user get a key in the first place? Currently, it's difficult to obtain a certificate, it's difficult to install it, and it costs time and money to obtain and maintain it. As Australia does not have a single national Certificate Authority (CA) (yet), it is especially difficult for Australian researchers to obtain an appropriate certificate.
- **Validity checking:** Is the certificate that is presented to a user still valid? Currently, checking of the validity is often not done, as it is very time-consuming, and the information is often outdated (you might have a new valid certificate that hasn't been added, or you might have a ex-employee whose certificate hasn't been revoked).
- **User identification:** I have this certificate from John Doe in the US; but is he really the John Doe that can access my application? Although a PKI certificate contains a Distinguished Name or DN, the way this DN is constructed varies from place to place, and is often also influenced by national laws.
- **Quality Control:** Has the current certificate and the CA who issued it been properly verified, and does my application actually verify their quality? Currently, many applications ignore part of the certificate, and the fact that many of them aren't properly formatted does not help either.

It is important to note the distinction between authentication and authorisation. Authentication addresses the question of "Is the user who he claims to be?" which is typically a static attribute. Authorisation addresses the question of "What is the user allowed to do?" which can vary over time. PKI provides credentials that can help to solve the first question, assuming that the initial enrolment was properly performed, and assuming that the signing CA is trusted. If the user has a certificate, and (if set) knows the password that protects the certificate, you can be confident that the user is who the certificate claims to be. Authorisation, which is commonly based on a set of user attributes, however, is not usually solved by PKI: naturally, a PKI certificate can contain a set of user attributes that are used by the application for determining the user's authorisation. But these attributes have to remain static and might not have been properly validated during the enrolment, which generates a number of problems. For example, if the user changes his/her role, some attributes could typically change as well, thereby invalidating the certificate, and hence someone needs to revoke it. Another problem is that not every application needs the same set of attributes, so either you need to have multiple certificates for each service and remember which certificate to use where, or you need to create a super certificate, which means that many services will receive much more information about you than they need (and this may compromise your privacy). As a result, PKI is not a scalable solution for passing changeable attributes for authorisation, and you would rather have a delegated or distributed model: one service (IdP) for authenticating the user; another service, the Attribute Authority (AA), will provide the authenticated user's accompanying attributes like name, email, or role, and manage which attributes need to be released to a certain SP. The SP, based on the received attributes and optionally some other locally-managed attributes, decides what a user is authorised to do.

This paper will therefore discuss the problem of *how to seamlessly and securely access distributed GRID applications*, looking at both authentication and authorisation issues. Currently, this presents problems for GRID researchers as well as system administrators of GRID applications. Additionally, as the number of users increases, a system administrator would prefer to no longer control access directly via Access Control Lists (ACL), i.e. based on a user's identity, but based on role-describing attributes, such as their role within a project or within the organisation (Role-Based Access Control or RBAC), or even more generically, any type of attribute (Attribute-Based Access

Control). Our proposed solution uses the Security Assertion Markup Language (SAML) for distributed authentication, and the eXtended Access Control Markup Language (XACML) for distributed authorisation. These technologies are discussed below.

EXISTING SOLUTIONS AND THEIR LIMITATIONS

To access GRID applications, a number of solutions are already implemented, or are currently being developed, and we will briefly discuss them in this section.

GLOBUS and PERMIS

The Globus Alliance is a community of organizations and individuals developing fundamental technologies behind the "Grid", which lets people share computing power, databases, instruments, and other on-line tools securely across corporate, institutional, and geographic boundaries without sacrificing local autonomy. The GLOBUS toolkit (GT) is an implementation of open source Grid software, which consists of a number of tools to make deploying a developing Grid services easier (Chadwick and Welch, 2004). PERMIS (PrivilEge and Role Management Infrastructure Standards validation) is a project, whose fundamental objective is to set-up and to demonstrate a distributed "infrastructure" able to solve both authentication and the authorisation issues. GT's Version 3.3 supports SAML combined with PERMIS for authorisation, and the latest release, GTv4, also supports XACML. Basically, operation with PERMIS is as follows (PERMIS), see Figure 1: User requests are passed, together with their X.509 PKI identity certificate via a SAML request to an IdP. The IdP verifies the user's credentials, and generates a SAML response. The SAML response is passed to the AA and retrieves the user's attributes, which are returned in another SAML response. Finally, this last SAML response is passed to the Authorisation Service of the SP, and the final SAML response is generated, which permits or denies the user access to a target resource.

GridShib

(GridShib) is a new 2-year project that started in November 2004. It addresses the same problem, but it focuses on using (Shibboleth) as the Attribute Authority. Their approach is slightly different to the proposed above. GridShib assumes the users have a valid PKI certificate, containing at least the

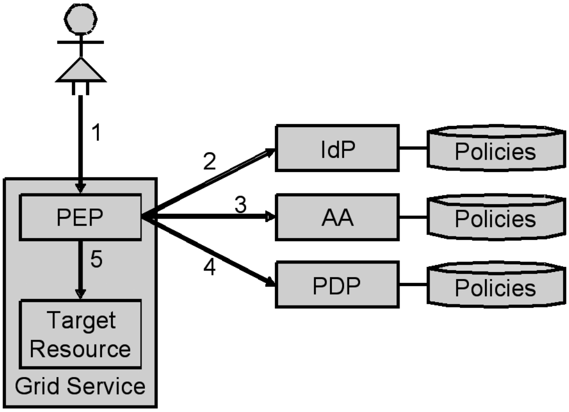


Figure 1: GLOBUS & PERMIS: 1) User request + credentials are offered to the Policy Enforcement Point (PEP); 2) SAML request to verify credentials; 3) Credentials are used to obtain the user's attributes; 4) Request + credentials + attributes are used to make a decision (Policy Decision Point); 5) Access resource.

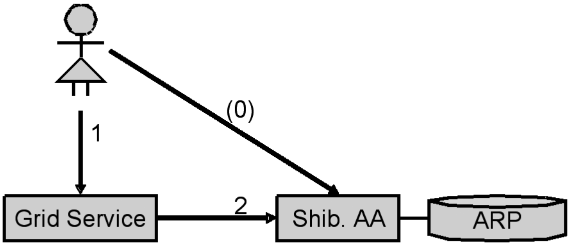


Figure 2: GridShib uses Shibboleth to obtain user attributes. 0) User modifies their Attribute Release Policy (ARP) to determine what attributes will be released to the Grid Service; 1) User presents her X.509 certificate to the service, 2) the attributes are pulled from the Shibboleth Attribute Authority using the user's certificate ID, and the Grid Service does internal attribute-based authorisation.

name of your institution (the Identity Provider or IdP). The PKI certificate is offered to the application, verified, and the IdP information is extracted. Next, the application contacts the IdP, supplies the certificate's ID, and receives in return a SAML assertion containing the user's attributes.

A major criticism against this approach is that the user still needs to obtain a valid certificate from a CA, and in most cases, the initial verification of a CA is not very good. Or, as Dr Ken Klingenstein, project director of the Shibboleth project, said: "If you buy a certificate for $20, you buy $20 of trust." The second problem is that the Grid service needs to create a reciprocal trust relationship with each Attribute Authority, which could be very time-consuming. Thirdly, how will this approach solve the 'multiple attribute authority' problem? For example, the Monash researcher who can only get access to a certain SP because he is from Monash *and* has a research grant from DEST. Or the USYD professor who is also an IEEE fellow, and wants to merge the privileges of both roles when accessing the IEEE repository. Clearly, the PKI certificate can be offered to a different AA with which you need to have a trust relationship, in which case the user would need to set up several ARPs – however, a link must be made between identity and attributes. Finally, there is no means for mapping of user attribute names and values, e.g. if the Grid Service is Chinese, and the user's IdP is Australian, how do we translate the required attributes? This is particularly a problem since the user cannot edit his attributes, as this would break the trust in the attributes.

FEDERATED ACCESS TO GRID APPLICATIONS WITH SAML AND XACML

This section will discuss the use of SAML to enable federated identity management and cross-institutional single sign-on, and XACML to enable federated access management.

SAML

SAML v2.0 is supported by OASIS and the Liberty Alliance (Liberty), a consortium of over 130 corporations. Many of Liberty's members, such as IBM, Novell, Sun, RSA, Verisign, Ping Identity, Oracle, etc. have implemented SAML in commercial products, but in Higher Education (HE), the open source implementation called Shibboleth® developed by the Internet2/MACE group in the USA, is the most popular.

As described in (The CoverPages), "SAML provides a standard way to represent authentication, attribute, and authorisation decision information in XML, and a series of web services-based request/response protocols for exchanging these statements. SAML v2.0 provides support for full federation and mapping of identifiers, session management, greater interoperability for attribute exchange and other features".

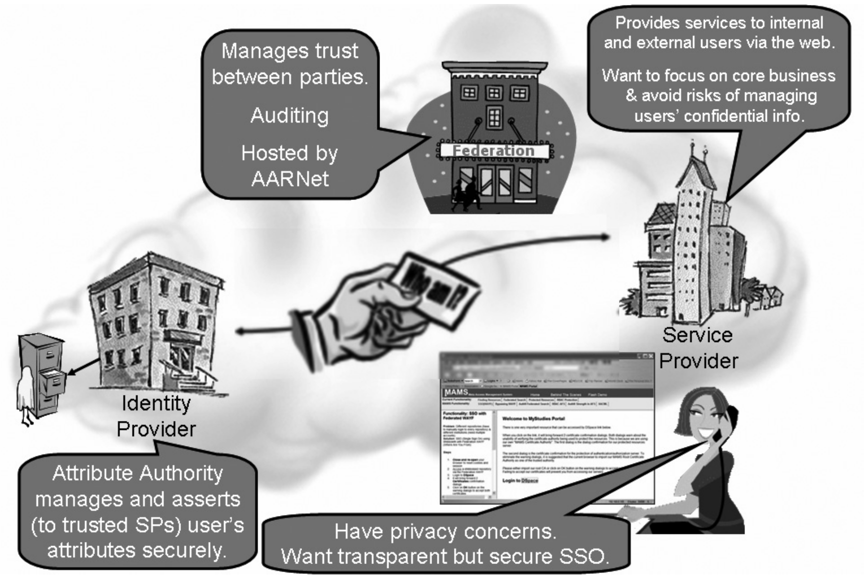

Manages trust between parties.
Auditing
Hosted by AARNet
Federation
Provides services to internal and external users via the web.
Want to focus on core business & avoid risks of managing users' confidential info.
Who am I?
Identity Provider
Service Provider
MAMS
Attribute Authority manages and asserts (to trusted SPs) user's attributes securely.
Have privacy concerns. Want transparent but secure SSO.

Figure 3: The Federation (top) establishes trust between IdP (left) and SP (right), such that a user (bottom) can use her IdP account to have SSO access to a service at the SP.

The Shibboleth architecture is depicted in Figure 3, and is already implemented in Australia (go to www.federation.org.au: as of November 2006, nine of the 39 universities have joined as an IdP, and eight others will join in the near future). The main assumption underlying Shibboleth is that service providers (SP) and identity providers (IdP) trust each other within a federation. This means that all IdP know all trusted SP in a federation and vice versa, and this trust is managed using PKI certificates.

Assuming the federation's trust infrastructure has been properly setup, a typical scenario of a user accessing a GRID application (referred to as SP) through a browser/portal is as follows below:

1. Using a browser, the user attempts to access a service provider (SP) in the federation. As the SP does not know the user, she is redirected (using a HTTP302 redirect message) to the Federation.
2. The Federation asks the user where she is from, and she selects her preferred IdP (typically her home institution) from the list.
3. She is then redirected to her IdP, which asks her to authenticate: often, a local username/password combination is sufficient, which will be verified against the institution's directory. In case stronger authentication is required, the user could also be requested to login using a PKI certificate. Note that in this case, the PKI certificate only needs to be trusted locally within the institution. Based on the target SP, which is conveyed to the IdP as part of the redirection, the IdP (after checking whether it can trust this SP) generates a SAML handle (an opaque identifier associated with her identity) for her and redirects her back to the SP with this handle.
4. The SP extracts the handle and uses it to query the IdP about the user's attributes.
5. The IdP sends some of the user's attributes (like role, email, affiliation) back to the SP (this is represented by the business card that the user presents to the SP), according to an attribute release policy (ARP), in a signed SAML assertion statement. As part of this assertion, it will

mention that the SAMLAuthenticationMethod = "Basic" or "Software PKI" was used to login. Note that the ARP is controllable by the user and IdP sysadmin (we have developed two tools to manage ARPs, ShARPE and Autograph, which can be found at www.federation.org.au)

6. Based on the attributes and the fact that a Software PKI was used to login, the SP gives certain access rights to the user and commences an authenticated session.

Note that, in order to maintain transport security, all traffic uses SSL/TLS encryption (HTTPS protocol). For more details see Vullings and Dalziel (2005).

XACML

In the earliest forms, authorisation was done using Access Control Lists (ACL), which clearly specified who was allowed to do what. Since it is difficult to maintain these lists in a quickly changing world (people get hired/fired regularly), they introduced Role-Based Access Control (RBAC). No longer did a person's identity determine what he can do, but the role(s) the user has. So there was a migration from access control based on a *name* attribute, to a *role* attribute. In a more general sense, you would like to use *any attribute* for authorisation, so more recently, the focus has been on Attribute-Based Access Control (ABAC), with its supporting language, XACML.

According to OASIS (XACML), "XACML enables the use of arbitrary attributes in policies, role-based access control, security labels, time/date-based policies, indexable policies, 'deny' policies, and dynamic policies – all without requiring changes to the applications that use XACML." Although XACML allows fine-grained access control, it does not provide all that Digital Rights Management (DRM) does, i.e. after you have accessed an XACML-protected resource, the resource may be no longer protected, and hence you could distribute it.

In order to get an overview of XACML, consider Figure 4, where Joe requests to edit the policy plan. XACML polices are based on a <u>subject</u>, performing an <u>action</u> on a <u>resource</u> (see step 1 above), under certain <u>circumstances</u> (like time of day). For example, you might have a policy stating that "<u>staff members</u> (subject) can <u>download</u> (action) any <u>thesis</u> (resource), if that staff member is a <u>member of the federation</u>, and if the resource has been <u>submitted more than two years ago</u> (two conditions).

In XACML, these authorisation requests are formalized, but everything still evolves around the triad *subject* performs *action* on *resource*, which can be *permitted*, *denied* or show you an *obligation*. This is called a *rule*, whereas the first part, the triad of *subject*, *action* and *resource*, is

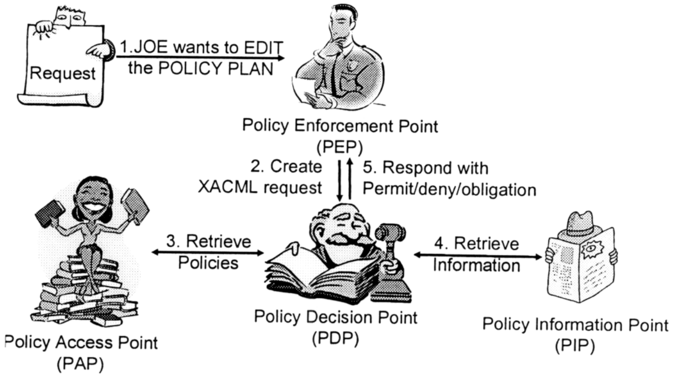

Request
1.JOE wants to EDIT the POLICY PLAN
Policy Enforcement Point (PEP)
2. Create XACML request
5. Respond with Permit/deny/obligation
3. Retrieve Policies
4. Retrieve Information
Policy Access Point (PAP)
Policy Decision Point (PDP)
Policy Information Point (PIP)

Figure 4: XACML in action: a user's request is translated to XACML: the decision is outsourced to the PDP which uses existing policies and context information to make its decision, which is then enforced by the PEP.

called the *target*, since the *target* will determine if the *rule* is applicable in the current situation. In most cases, you will create many rules, such as:

Students – Read –Thesis → Permit
Staff – Edit – Lecture notes → Permit

XACML allows you to go even one step further, and when a *target* is applicable, you can include additional *conditions* that need to be fulfilled as well, e.g. in the latter case, let's assume that a staff can only edit lecture notes of the courses he is teaching:

Staff – Edit – Lecture notes → Permit
iff Subject.CourseID matches Resource.CourseID

You can even include external information, like date and time, e.g.

Anyone – Read – Thesis → Permit
iff CurrentDate > Resource.PublicationDate + 2 years

Continuing from here you can see that we will end up with many rules, which can be grouped together in a *policy set*. Within a *policy set¸* some of the rules may be colliding with other rules, e.g. if the repository wants to be secure, it may decide to deny every action by default:

Anyone – AnyAction – AnyResource → Denied

Therefore, when rules collide, you need to determine which one takes precedence, and XACML employs *rule combining algorithms* (RCA) to do that, e.g. *permit overrides*.

Finally, a repository may wish to employ multiple policy sets, e.g. one for each collection of resources, or one for each user role like staff and student. In that case, the XACML engine processing the incoming request needs to determine which policy set applies, and, like the rules, it uses the policy target to find them. And again, similar to conflicting rules, you can now have conflicting policy sets, which are resolved using *policy combining algorithms* (PCA). For an overview, see Figure 5.

A typical use case would be accessing a large dataset:

1. The researcher would use Single Sign-On (SSO) via SAML to access the repository containing the dataset (see above).

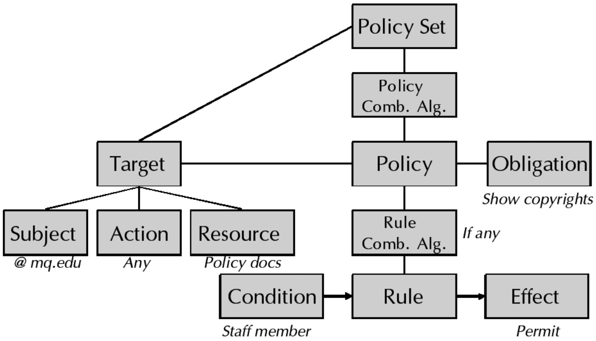


Figure 5: XACML policies are identified using targets, consisting of a SUBJECT doing an ACTION on a RESOURCE. Each policy contains rules and a way to combine them.

2. The repository containing the dataset would display its abstract page and each possible further action (e.g. view sample records, rank, download, comment, or edit) would be evaluated using XACML as explained in
3. Figure **4**: based on existing policies, the actual dataset's attributes and the researcher's SAML attributes, these actions would only be displayed if the researcher is permitted to execute them.
4. The researcher selects to download a set of records from the dataset.

Current work is underway to develop an XACML module for institutional repositories within the Research Activityflow and Middleware Priorities (RAMP) project, which can be found at www.ramp.org.au.

PROPOSED SOLUTION: AUTHENTICATING THE USER WITH SAML

Based on the description of SAML and XACML, we propose the following model, which is very similar to myVocs (Robinson, 2005), for federated Identity & Access Management to Grid services, especially where the user's attributes are not stored in one AA, see also Klingenstein (2007), but in several, and we need to consolidate information from multiple AA to access an application. We start with authenticating the user before we discuss the authorisation issues.

Figure 6 depicts a typical example of a user accessing a Grid portlet service using a browser (only for the first trial – later accesses will be faster, as all available data is already present at the Virtual Organisation or VO):

1. The user tries to access a Grid service provider (SP).
2. As the SP doesn't know the user, she is redirected to the VO's Where Are You From (WAYF) service, which consists of a list of VO member institutions. Additionally, the WAYF creates a cookie and stores the desired SP's address.
3. She selects her institution's Identity Provider, is redirected to the IdP, and provides the necessary login credentials. Typically, this could be a username and password, but could also be a PKI certificate provided by the institution's helpdesk and signed with the institution's key (so the user's certificate has no validity outside the institution, and there is no need to manage it – if the user leaves the institution, her account is blocked and the certificate is no longer useable).
4. The IdP validates the login credential, and the Attribute Authority uses the user's Attribute Release Policy to determine which attributes should be sent to the VO, which is done using the SAML artefact method or the SAML post method. Additionally, the *SAMLAuthenticationMethod* variable is set to Software PKI (in case of PKI login) or Basic (in case of password login).

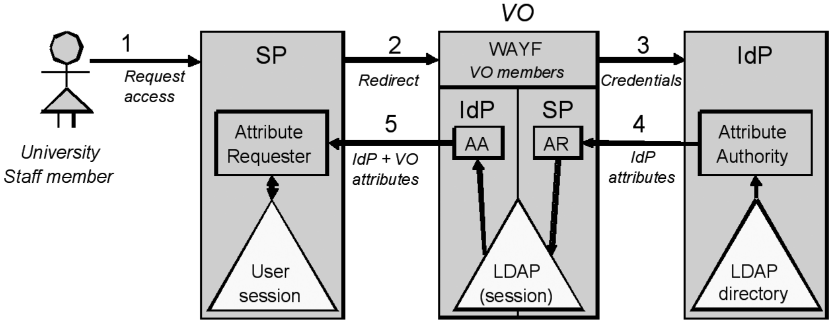


Figure 6: Authenticating users of the Virtual Organization (VO) wishing to access a Grid service provider, which is similar to the myVocs model.

5. The VO verifies the received SAML assertion (do we trust this institute, is the signature valid) and stores the received attributes in its own directory (this could be session based, or over longer period according to the policies of the VO). Typically, those attributes should contain the personal (but general) attributes of the user, like full name, email address, etc. Now, based on its own knowledge of the user, additional attributes can be added, e.g. that the user is a sys admin, has completed training course X, is authorised to access equipment Y, and has credit Z.
6. The user is redirected again to the actual Service Provider (the SP's address was stored by the WAYF in step 2) she wants to visit, accompanied by another SAML assertion generated by the VO's IdP. The SP subsequently 'knows' the user, i.e. the user is authenticated.

eResearch Toolkits: The IAM Suite

Our current implementation of this model is called the Identity and Access Management (IAM) Suite, which is a type of Virtual Research Environment (VRE). A VRE has been defined by JISC in the UK as follows: "A VRE will provide an integrated and interoperating set of networked tools, systems and resources to facilitate and enhance the practices of individual researchers, research teams and communities distributed across the UK and abroad. A VRE will be able to address the needs of hybrid teams and communities as well as those of individual researchers, and will provide them with 'different time – different place' access to experts, knowledge, collaboration tools and computational resources from a personalised access point. Lastly, a VRE will be able to provide a mechanism for the creation of a flexible layered architecture of distributed and interoperable resources and tools, in order to meet different requirements" (JISC).

The basic concept behind our IAM suite, which is an implementation of a VRE on top of the JSR168 compliant GridSphere portal (www.gridsphere.org), has already been depicted in Figure 6: You login to the VRE using Shibboleth SSO via your IdP. Within the VRE, you use Shibboleth again to access internal VRE services: the VRE's internal IdP is now used as an attribute authority (VO AA), combining personal user attributes (from the IdP) with VRE-specific attributes (e.g. group memberships, entitlements, etc.).

In addition to being able to act as an IdP (VO AA), for access to grid services a long-lived PKI certificate or a short-lived proxy certificate is often needed. Proxy certificates can be obtained using

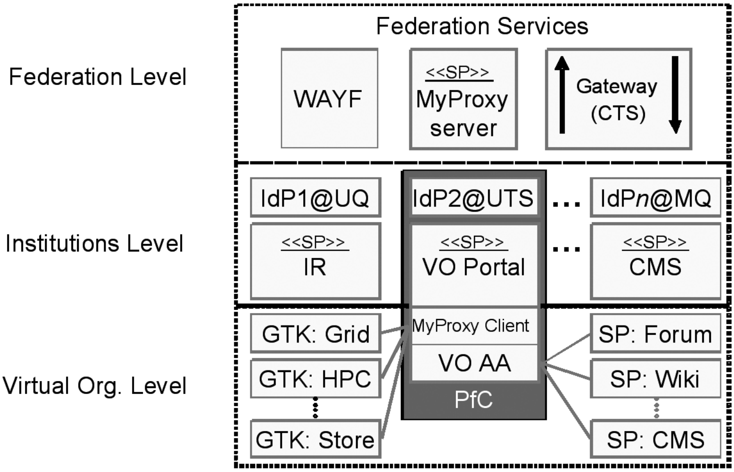


Figure 7: The IAM suite (VRE) extends the Federation with a Virtual Organization level.

MyProxy, an open source product developed by Basney (2005), which allows users to store their long-lived PKI in the MyProxy server: grid services would use the MyProxy client to login to the server using login name, password and, optionally, pass phrase, to retrieve a proxy certificate for the session. By Shibbolizing the MyProxy server, which sits at the top in the Federation Level, we effectively have created an Open Certificate Authority, which can convert a SAML assertion to a proxy certificate for use within the VRE.

SAML for Desktop Applications

Our second example shows the same setup, but this time for a desktop application (see Figure 8). The user first logs in to the application, after which it provides her credentials to the IdP: either by the application forwarding her (locally-issued) PKI certificate or other credentials that might be accepted by the IdP according to the Web Services- Security specification. Alternatively, for interoperability with the Shibboleth model, the desktop application could also include a simple HTTP-class, which would perform SSO via HTTPS to the IdP. As part of the initial request, the application will also request a SAML assertion for the VO (step 2), containing user attributes that are required for the next phase. Second, the application will do the same for the Virtual Organization (VO), offering the first SAML assertion (using the SAML profile in WS-Security) as proof of its identity, and receive a second SAML assertion containing additional VO-related attributes (step 3-4). Finally, both SAML assertions containing all necessary information are offered to the Grid service, which permits or denies the action (step 5). Alternatively, the VO could store the IdP issued attributes in its own directory, and create a new assertion for the Grid service, containing only its required attributes and signed by the VO, so the Grid service will not receive too much information about the user. And finally, the VO could provide the application with a token, which the SP can use to access the VO directory.

Cross-Federation Identity Management

The final example shows two versions of dealing with multiple attribute authorities by chaining several of them (see also Klingenstein, 2007), for alternative approaches. In Figure 9a, we present the simple case of aggregating attributes from two sources within the same federation (so IdPs and SPs trust each other directly).

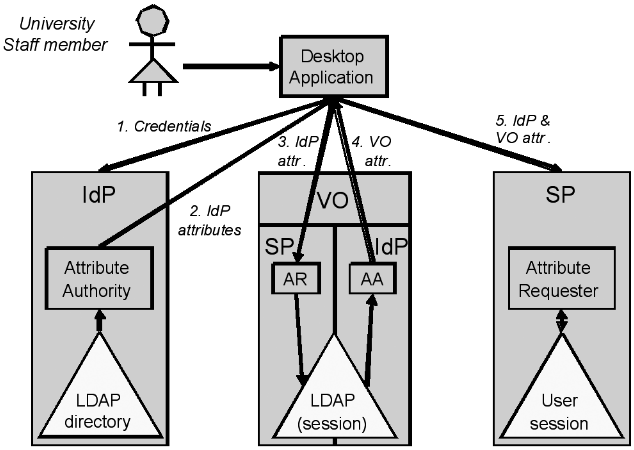


Figure 8: A user accessing a Grid service via a desktop application using SAML over SOAP to pass SAML attribute assertions.

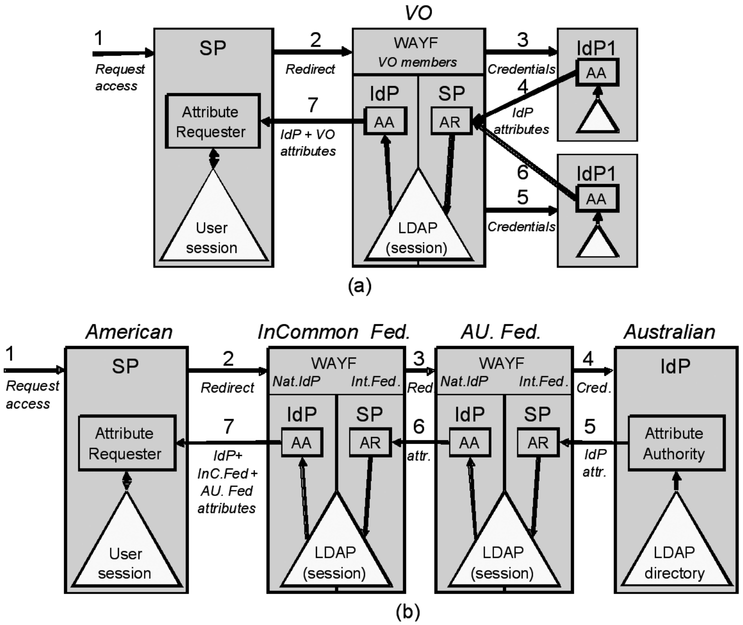


Figure 9: a) Chaining multiple attribute authorities; b) Cross-federation chaining via a Federation gateway

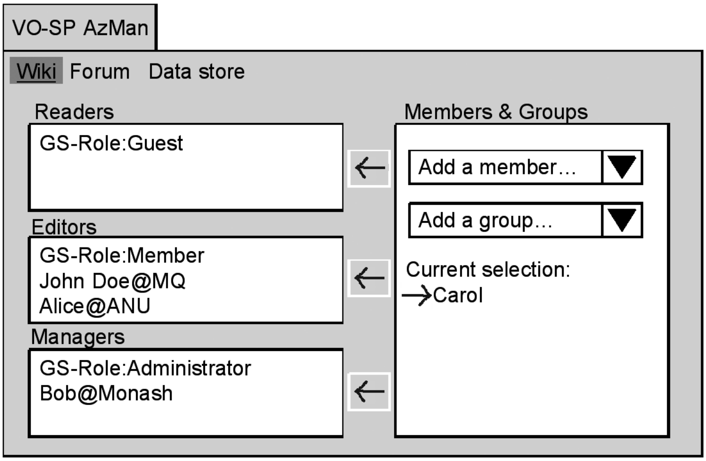
Figure 9b shows the slightly more complicated case when the SP and IdP are not part of the same federation, so we need to cross a federation boundary. Here, we have taken the example of the American Service Provider which wants to offer access to a staff member of an Australian IdP. Based on an assumed trust relationship between the American InCommon Federation and the Australian Shibboleth Trust Federation, attributes are passed along and translated, i.e. attribute names and values are mapped as they move towards the American SP. On a subsequent visit to another SP that belongs to the same federation, the IdP does not need to get involved as the federation will ‘know’ the user.

PROPOSED SOLUTION: AUTHORIZING THE USER

Assuming we ‘know’ the user, and her attributes are available to the Service Provider, we can now discuss how to authorize the user. In many cases, RBAC will be sufficient to control who can do what. For example, assume that the VO has an internal Wiki, and its members need to receive proper authorisation when accessing the Wiki. In that case, after logging in to the VO, the VO AA sends a SAML assertion on behalf of the user to the Wiki. Since the VO also contains a group and authorisation manager (a mock-up is shown in Figure 10), useful attributes for authorisation can be sent across to the Wiki, such that the VRE allows central authorisation management, e.g. Guest users within the VO can read information in the Wiki, John and Alice and all VRE members can edit the Wiki, and Bob is also an administrator.

More generally, in order to add an application to this framework, it first needs to allow external users to authenticate via Shibboleth, and additionally define some attribute values (e.g. a group name such as “WikiEditor”, of which John is a member) for RBAC.

In more complicated scenarios where the previous RBAC solution does not suffice, we should use XACML in either of the following three ways:



Readers
GS-Role:Guest

Editors
GS-Role:Member
John Doe@MQ
Alice@ANU

Managers
GS-Role:Administrator
Bob@Monash

Current selection:
→Carol

Figure 10: Role-Based Access Control to VO services

1. **Outsourcing model:** The user contacts the Policy Enforcement Point (PEP) with a certain request, and the decision to permit or deny the request is outsourced to the Policy Decision Point (PDP)
2. **Provisioning model:** The user contacts the PDP with a certain request and, if the user is permitted to do it, updates all services accordingly. For example, if the user wants to run a certain experiment, she contacts the PDP at the SP 'Scheduler' and requests to run the experiment. As a consequence, the SP 'Source' (e.g. a SEM analysing a material sample), the SP 'Sink' (e.g. a computer cluster that analyses the scans), and the SP 'Bandwidth (e.g. a high-speed optical fibre connection between SEM and the computer cluster) are all provisioned with the policies to allow the user to run the experiment at a certain time (see Figure 11).
3. **Token model:** The user contacts the PDP with a certain request and, if the user is permitted to do it, receives one or more tokens (keys) to access all services. Using the same example as above, the tokens would give access to all services necessary to perform the experiment.

CONCLUSIONS

The proposed solution offers a more flexible method for authentication and authorisation to Grid applications in a distributed environment compared to existing approaches, and without necessarily

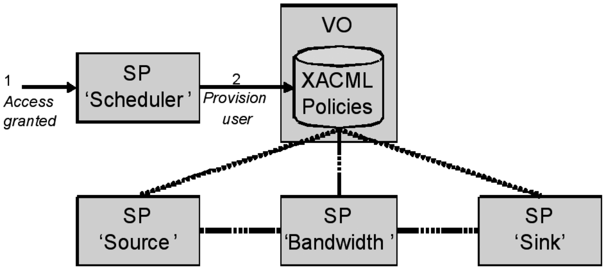


Figure 11: XACML in action: the authenticated user accesses the SP Scheduler, which updates the XACML policies so she can execute her experiment. Access to the other SPs will also be via the VO.

relying on (difficult to manage) user PKI certificates. Even if strong PKI authentication is necessary, this is dealt with by generating local PKI certificates by the institution (although it could be argued that PKI's stronger authentication actually improves security since many Grid users are not aware that they have a PKI certificate, and/or they do not protect it with a pass phrase. At least with Shibboleth, if the SSO environment allows access to HR details, people are probably more careful with their login credentials). Additionally, building upon the Shibboleth implementation, it can allow users to remain anonymous where appropriate, as the attributes do not necessarily have to contain any personal information. For example, it might be sufficient to tell the Virtual Organization that you are a member of 'Sandstone' university, after which 'Sandstone' will assert that the user is a postgraduate researcher at Sandstone's ICS department. This could be sufficient to allow you access to a Grid service at Gumtree. Also, our distributed approach does not rely on many end-to-end trust relationships, but builds upon trust between and inside federations. And finally, these existing trust relationships solve the issue of mapping attribute names and values for each VO separately, for example from an Australian IdP to an American SP.

ACKNOWLEDGEMENTS

We thank the Australian Department of Education, Science and Training for providing funding to this project under the program Backing Australia's Ability.

REFERENCES

BASNEY, J. (2005): MyProxy Protocol, *Global Grid Forum Experimental Document GFD-E.54*, http://grid.ncsa.uiuc.edu/myproxy

CHADWICK, D., OTENKO, S. and WELCH, V. (2004): Using SAML to link the GLOBUS toolkit to the PERMIS authorisation infrastructure, *Conf.Comm.&Multimedia Security (CMS)*

COVER, R. (2004): The CoverPages, http://xml.coverpages.org/ni2004-07-15-a.html

FOSTER, I. (2005): A Globus Toolkit Primer, http://www.globus.org/

GRIDSHIB PROJECT (in preparation): http://grid.ncsa.uiuc.edu/GridShib/

GUTMANN, P. (2001): Everything you never wanted to know about PKI but have been forced to find out, Tutorial, http://www.cs.auckland.ac.nz/~pgut001/

OASIS (2005): eXtensible Access Control Markup Language (XACML) TC, "XACML V2.0", http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

OASIS (in preparation): Security Services (SAML) TC, "SAML V2.0", http://tinyurl.com/a3h5l

KLINGENSTEIN, N. (2007): Attribute aggregation and federated identity, *Subm. SAINT 2007*

LIBERTY ALLIANCE PROJECT (in preparation): Digital Identity Defined, http://www.projectliberty.org/

PERMIS project (in preparation): http://www.permis.org/

ROBINSON, J-P. (2005): MyVOCS: My Virtual Organization Collaboration System. *Internet2 Member Meeting,* http://myvocs.org

SHIBBOLETH PROJECT (in preparation): Internet2 Middleware, http://shibboleth.internet2.edu/

VULLINGS, E. and DALZIEL, J. (2005): Searching and retrieving protected resources using SAML-XACML in a research-based federation, *IPSI Transactions on Internet Research*, 1(2), http://www.internetjournals.net/

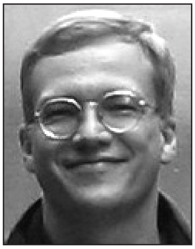
BIOGRAPHICAL NOTES

Erik Vullings

Erik Vullings was Program Manager of the Meta Access Management System (MAMS) project at Macquarie University from May 2004 until January 2007 (when he moved back to the Netherlands). In this role he led the development and implementation of a national testbed for federated access and identity management based on Shibboleth. He received his MSc and PhD from Delft University of Technology in the Netherlands. In January 1999, he joined Philips R&D as a systems engineer. In October 2003, he became the program manager of a 16M European Union funded FP6 project in assembly equipment.

James Dalziel

James Dalziel is Professor of Learning Technology and Director of the Macquarie E-Learning Centre Of Excellence (MELCOE) at Macquarie University in Sydney, Australia. James leads a number of projects including: LAMS (Learning Activity Management System), including roles as a Director of the LAMS Foundation and LAMS International Pty Ltd; MAMS (Meta Access Management System), a national identity and access infrastructure project for the Australian higher education sector; RAMP (Research Activityflow and Middleware Priorities), a project investigating open standards authorisation and e-Research workflows, and ASK-OSS (the Australian Service for Knowledge of Open Source Software), a national advisory service on open source issues for the Australia higher education and research sector. Prior to his current roles, James helped lead the COLIS (Collaborative Online Learning and Information Services) project, was a Director of WebMCQ Pty Ltd, an e-learning and assessment company, and was a Lecturer in Psychology at the University of Sydney.



Markus Buchhorn

Dr Markus Buchhorn is an ex-astronomer who crossed to the dark-side of IT many years ago. He has been focussed on how ICT infrastructure provides support for user activities, especially in research and education. His main interests include networking, collaboration technologies, data management and authentication/authorisation frameworks. His jobs have included roles in GrangeNet, APAC, APSR, APAN and Internet2, as well as his enduring jobs at the ANU, and engagement at various levels with the NCRIS processes. He is currently the Director for ICT Environments at the ANU.