Searching and Retrieving Protected Resources using SAML-XACML in a Research-Based Federation

Vullings, E. and Dalziel, J.

Abstract—*Extending IAM (Identity and Access Management) beyond an institution's boundaries to enable SSO (single sign-on) within a federation of trusting institutions currently receives much attention [1]. Within this area, we address the following two problems: how to effectively find protected content and protected metadata in numerous repositories across the federation and how to protect your own published material from unauthorized access.*

Existing solutions for finding resources like federated search assume that all metadata is freely available, but they do not give users an indication of the accessibility of the found results.

The solution we propose extends the SAML-based (Security Assertion Markup Language) federations as currently implemented by the Liberty Alliance and Shibboleth with XACML (eXtended Access Control Markup Language): SAML allows for asserting information about a user, which can be used to determine access privileges. XACML allows for defining rules that enable fine-grained access control to resources. Although this is more limited than digital rights management (DRM), it is sufficient for our user group in Higher Education.

Index Terms—*federated Identity & Access Mgmt, fine-grained access control, SAML, XACML*

1. Introduction

THE internet makes finding publicly available research material very easy, as can be concluded from the popularity of search engines like Google. However, if the content, or its metadata, is not public, you often have to login to a repository, and do a dedicated search, after which you move on to the next repository. Clearly, this is not very efficient, and it requires many accounts you need to manage. On the other hand, if you publish some information yourself, you would often like to limit the people who can access it.

Manuscript received 31 March 2005. This work was supported by the Australian Department of Education Science and Training (DEST) in its program 'Backing Australia's Ability', which financed the Meta Access Management System project for three years with AUS$4.2 million till the end of 2006.

Dr. Erik Vullings is the Program Manager of the MAMS project at Macquarie University's E-Learning Centre of Excellence (MELCOE), Sydney, Australia (erik.vullings@melcoe.mq.edu.au) .

Prof. James Dalziel is Director of MELCOE and Chief Investigator of the MAMS project (james@melcoe.mq.edu.au). .

Existing solutions for finding resources, like federated search, are limited in that they assume that all metadata is openly accessible. However, for repositories containing, say, sacred artifacts, this might not be the case. For example, a person might need to be a member of an indigenous community, male and initiated, in order to see the sacred artifact. Otherwise, you do not even know about its existence. And even if the metadata is available, it does not contain any information about who can actually access the resource, so you have to actually try to retrieve it in order to find out. And finally, the federated search system's functionality is limited by the fact that search results are based on the trust between the repository and the federated search system, and not on trust in the actual user that is using the system. Continuing our previous example, if the federated search system could inform the repository with sacred artifacts that the current search is by a member of the relevant indigenous community, male and initiated, the search would extend to more resources and potentially be more accurate.

This paper will look at ways to reduce the above-mentioned problems in the current approaches to finding and accessing resources. The layout is as follows. First, we briefly introduce the two core open standards that underlie our work, i.e. SAML and XACML. SAML is illustrated using one of the existing SAML implementations, Shibboleth, which is used to build a federation of higher education identity providers and service providers. XACML is not yet widely implemented, but the basic mechanism is easy to explain.

Next, based on our problem statement, we will describe the key requirements and usage scenarios in Section 3 that we have to address, which will lead to the proposed solution in Section 4, ending with our conclusions.

2. Federated Identity & Access Management

Extending SSO beyond your institution to a federation of trusting institutions in higher education (HE) is a key requirement as determined by the Meta Access Management System (MAMS) project [7]. Currently, there are

two standards that address this issue: SAML v2.0, supported by OASIS [2] and WS-Federation (Web Services). SAML is supported by the Liberty Alliance [5], a consortium of over 130 corporations, and Shibboleth, an open source implementation developed for HE by the Internet2/MACE group in the USA. WS-Federation is currently only supported by Microsoft and IBM, and without existing implementations, and therefore less relevant to our immediate needs, so we will continue our discussion using the SAML approach.

2.1 SAML and Shibboleth

As described in The CoverPages [3], "SAML provides a standard way to represent authentication, attribute, and authorization 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".

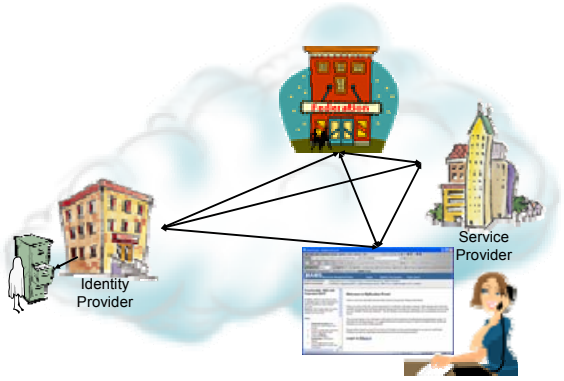

Figure 1. The Shibboleth implementation uses the HTTPS protocol for communication between browser, IdP, SP and Federation.

A high-level Shibboleth implementation is depicted in Figure 1. The main assumption underlying Shibboleth is that service providers (SP) and identity providers (IdP) trust each other within a federation. A typical use case of a user accessing a federated service goes as follows:

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 login if she hasn't already done so. 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, after extracting the handle, 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, according to an attribute release policy (ARP), in a signed SAML assertion statement. Note that the ARP is controllable by the user and IdP sysadmin.
6. Based on the attributes, 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).

The main components have the following responsibilities:

Identity Providers (IdP):

- Allow SSO, within the institution and federation.
- Maintain user attributes while protecting privacy.
- Know the SPs in the federation, so they only send user attributes to trusted SPs; and can format these attributes in a way the SP expects.
- Allow system administrators and individual users to control the attribute release.

Service Providers (SP):

- Control access to service (who can access what) based on the attributes received from an IdP, i.e. they implement attribute-based access control.
- Know the IdP in the federation, so they only accept user assertions from trusted IdP.

Federation:

- Manages the trusted IdP and SP in the federation, and informs member institutions about it.
- Offers additional services, like a Where Are You From service (WAYF), which allows users to select their IdP.

Finally, the user's **browser:**

- Has to enable Cookies for maintaining security contexts with IdP and SP, and enable JavaScript.
- Has to properly deal with HTTP302 redirect messages

2.2 The IdP as Attribute Authority

Essential for our later discussions is the part of Shibboleth's IdP called the Attribute Authority (AA). When a SP requests the attributes in step 5, it is the AA that actually has to perform several actions:

- Determine which attributes the SP needs, which is part of its default Attribute Release Policy (ARP) as configured by the IdP (e.g. 'Email address').

- Check if the user has overruled the ARP (do not send email address due to privacy concerns).
- Retrieve the user's attributes from the directory.
- Format them if necessary (e.g. if the directory contains an "E-mail" attribute, and the SP expects an "email" attribute.
- Put the attributes in a SAML assertion, sign it, and send it across.

Additionally, as already implemented by some Liberty Alliance members, you may need an interaction between IdP and SP to change the ARP 'on-the-fly'. For example, assume that a user visits a publisher, where he receives the "bronze-level" service based on his institution's contract with the publisher. However, the publisher offers the user the "gold-level" service in exchange for his full name and email address for marketing purposes. If the user agrees, he is redirected to his IdP, which asks for a confirmation before actually changing the ARP and releasing the full name and email address to the SP. Another example is a user who wants to download a certain protected document. Although the repository normally only needs to know his affiliation (but not necessarily the user's identity), in order to retrieve this document it also needs the user's full name (or enrolled courses, or any other user attribute). After the user's confirmation, it will receive this information only once, so on the next visit, the original ARP will be used again.

Finally, the ARP should also allow for privacy of information, and the user should be able to change his ARP, potentially hidden from the sys admin. For example, a user who often needs to order items needs to fill out his contact and credit card details for each site manually. Instead, the SP can offer a function to collect this information automatically from the IdP, where the user needs to confirm at the IdP to release this information. Optionally, the IdP could allow the user to choose a different 'persona', i.e. if he makes a personal purchase; take information from his personal description (John Doe, home address and MyPersonal credit card). For business purchases, he can use his business description (John Doe, company address and MyCompany credit card). However, in both of these cases, all information should only be accessible by the user, and not by someone else.

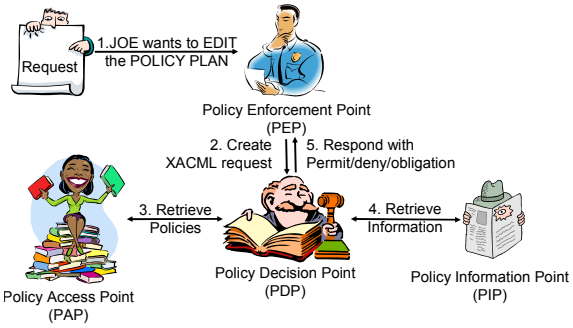

Figure 2. XACML in action: a user's request is translated to XACML, a decision is based using policies and context information, and enforced.

2.3 XACML

According to OASIS [4], XACML was chartered "to define a core schema and corresponding namespace for the expression of authorization policies in XML against objects that are themselves identified in XML. […] 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 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 2. A typical use case would be a researcher viewing a paper in a repository:

1. The researcher would use SSO via SAML to access the repository (see Section A).
2. The researcher would search for some keywords, discover the paper and request it.
3. The repository would display the paper's abstract page and each possible further action (e.g. view full paper, rank, download, comment, or edit) would be evaluated using XACML as explained in Figure 2: based on existing policies, the actual resource and the researcher's SAML attributes, these actions would only be displayed if the researcher is permitted to execute them.The researcher selects to view the full paper. The access control is expressed in XML policy documents, written in the XACML namespace. Policies are selected based on generic Target descriptions (see), and each policy consists of multiple specific rules and a way to combine the outcome of different rules (e.g. if any rule permits the action, it's permitted). Additionally, it allows specifying an obligation that must be fulfilled before you can perform the action. Note that access should be considered in its wider sense, as it is not limited to opening documents, but could also mean editing, ranking, or any other type of action.

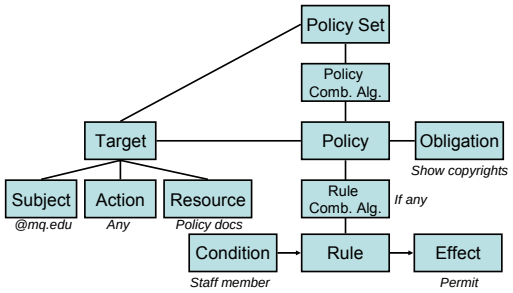

Figure 3. XACML policies are identified using targets, consisting of a SUBJECT doing an ACTION on a RESOURCE. Each policy contains rules and a way to combine them.

3. KEY REQUIREMENTS

Key requirements for federated identity and access management in the Australian Higher Education sector were elicited using two techniques: (1) voting on 14 requirements during several workshops with 148 people, where each had 4 votes, and (2) ranking 45 requirements by 33 people from the 148. The detailed outcomes are published in [7] but the top five is as follows (the #votes it received is added to the end):

(1) SSO with authentication strength (129)
(2) Repository Access & DRM (100)
(3) Federations, federated search & policies (76)
(4) Virtual organizations (59)
(5) Attribute management (57)

SSO can be addressed using SAML as described previously, with the ability to use many authentication methods (like passwords, Public Key Infrastructure (PKI), one-time Short Message Service (SMS) passwords, etc), leading to different authentication strengths. For an overview of the full range of DRM requirements, see [8]. Here, we will describe some key system requirements related to the scope for the MAMS project for repository access (2 – from list above) and federated search (3) from the perspective of the four types of stakeholders: authors; users of content; repositories (e.g. libraries); and publishers. Additionally, we will briefly discuss attribute management (5): who can change or view attributes in the IdP's directory, and what attributes will be sent to each SP.

From an **author's** perspective, a system could:

R1. Disseminate a work to a large audience
R2. Restrict access to a work
R3. Receive feedback from peers
R4. Rank a work based on input from peers
R5. Track the use of a work
R6. Recognize the author as the creator
R7. Enable proper attribution when citing a work
R8. Protect the integrity of a work
R9. Specify whether derivative works can be created
R10. Restrict the way a work can be used and redistributed (non-)commercially

For **researchers**, a system could:

R11. Find relevant content
R12. Provide access to the content
R13. Present the terms of the license
R14. Download content in batches (see 4E).
R15. Provide attribution requirements
R16. Allow anonymous access

For **digital repositories (e.g. libraries)**:

R17. Define the supported access privileges
R18. Provide authors with a method of defining access privileges and terms of use for the content they deposit
R19. Expose the terms of use and access privileges information to repository users, and provide them with a method of accepting or rejecting usage terms
R20. Protect content from unauthorized use
R21. Expose access privileges metadata for harvesting
R22. Define access privileges for the use of metadata (catalog records)

For **publishers**, a system could:

R23. Protect content from unauthorized access
R24. Uniquely identify each instantiation of digital content for licensing and tracking purposes
R25. Define usage rights, e.g. view, print
R26. Identify content users, either to confirm that they have a license or to assign them a license for use

4. PROPOSED SOLUTIONS

In Figure 4, we have drawn the context diagram of the system that should realize the required behavior as described in Section 3. Here, the components that are part of the blue square are under our control, whereas the other components are not or only limited. As a consequence, we are not able to guarantee the realization of all requirements. However, this paper realizes the following requirements: R2, R5, R11, R12, R14, R16, R17, R18, and R26. In two other projects within the MAMS project, we are improving the repository product by using XACML for access control and we are adding functionality and usability to the attribute authority within the IdP, so we can realize more requirements.

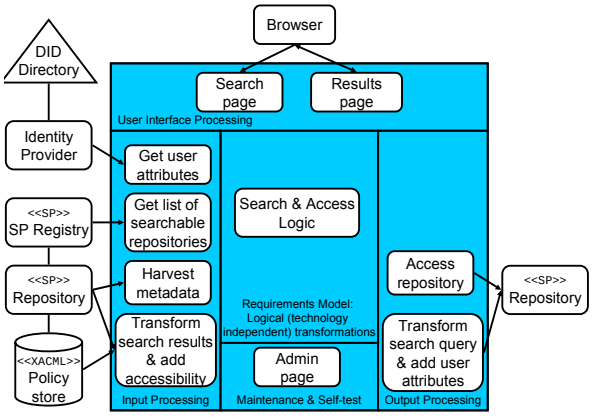

Figure 4. The context diagram shows the dependencies on external systems (DID = Digital Identity).

Our proposed solution to realize these requirements is depicted in **Figure 5**, where the (AFS) authenticated federated search engine extends the capability of the normal (anonymous) federated search engine by receiving attributes of the user that is accessing it. Based on this simple architecture, several scenarios can be implemented, which are discussed in the following subsections.

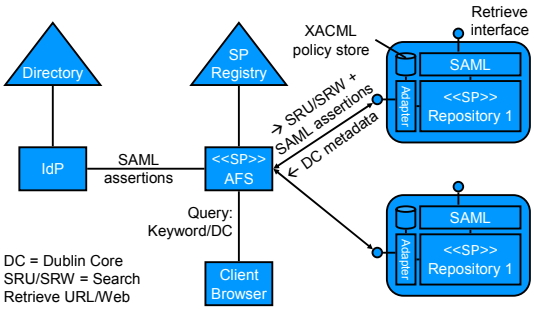

Figure 5. Simplified architecture for our authenticated federated search (AFS) system.

4.1 AFS as yet another SP

In the simplest case, the AFS retrieves the repositories it should search from a registry, either maintained by itself, the federation, or the user's home institution (in the latter case, the AFS could also search and retrieve from repositories with which the home institution might have arrangements, but which are not part of the federation). When the user visits the AFS using SAML/Shibboleth SSO, the AFS will receive the superset of all attributes that each single repository needs. So if repository SP1 needs your affiliation, and SP2 your email address, the AFS will receive both from the user's IdP. Now when it performs the query across the SPs, it will send the SRU/SRW search [9] query together with the SAML assertion containing the user's affiliation to SP1 and the email address to SP2. At the repository site, an adapter (see **Figure 5**) transforms the query to the repository's query language, and determines the user's accessibility to the returned results (metadata and content) using an XACML policy store: these policies are either used directly for authorization by the repository itself (preferred), or the policy store mimics the repository's internal authorization behavior. The search results are in this case returned in Dublin Core [10] metadata format, together with an indication of the user's ability to access the content. If the associated content is accessible, and the user decides to access it, the normal Shibboleth behavior is invoked, i.e. the user will be redirected to his IdP with a request to access the found result in repository X, which will redirect him to repository X with a SAML handle giving him access.

The main limitations of this model are the following:

- The AFS receives much information about the user, because it receives many attributes intended for different SPs, as well as a user's interest (query). This may be a privacy issue.
- The AFS will bypass the IdP's ARP, and schema transformation rules, hence the IdP and all SPs need to use the same schema i.e. attribute names
- The AFS will bypass the user's ARP and provide attributes required by each SP, e.g. if the user does not want SP2 to get his email address, but SP3 may, the AFS does not know this and will send it to both.
- All SPs need to trust the ASF's assertions.

4.2 AFS receives encrypted SAML assertions

A slightly more complicated model that deals with all of the above-mentioned problems is as follows: when the user visits the AFS, the IdP does not send all the attributes that the different repositories need in one SAML assertion to the AFS anymore. Instead of being bypassed, it directly sends signed (using its private key) and encrypted (using the SP's public key) SAML assertion packets, one packet for each SP. So the AFS gets one packet, which is based upon the ARP, for each SP, and it will forward them to the SP when it performs the query. Note that, as these packages are encrypted with the SP's public key, the AFS will not be able to read them. When performing a query, the encrypted and signed SAML assertion is received by the adapter (see **Figure 5**), it decrypts it using its private key and verifies the signature, and treated as discussed previously.

4.3 AFS extending the IdP

An alternative approach is to combine the AFS with the IdP, such that its assertions will receive the same level of trust as was given to the IdP (basically, the assertions would be indistinguishable). Additionally, it would have the advantage that the AFS could respect the user's ARP completely, and that it could be managed by the IdP themselves, i.e. the GUI could be made to fit with the institution's layout, their presentation preferences (do we search using Dublin Core or MARC) can be respected, and you could even query service providers residing outside the federation, as long as they trust you. The main downside to this approach is that now each IdP needs to manage and operate their own AFS, instead of only one site. From a security perspective, this should have no impact, as all trust is based on the IdP's assertion.

4.4 Harvesting and access privileges metadata

Our solution uses a distributed parallel search of repositories to find results as opposed to using harvested metadata. Although performing a distributed search can be more accurate and allows an indication of accessibility, searching local harvested metadata might be faster. If one wishes to combine the advantages of both, there are two approaches: first, you could return the search results using harvested metadata, performing an actual search in parallel, and

update the accessibility (and potentially also the search results) while the user is inspecting the results. This could be achieved using an XMLHttpRequest [9] object, which checks in the background if the updated accessibility information is available.

A second approach is to enrich the harvested metadata with access privileges metadata. This access privileges metadata could be a copy of the applicable XACML policies, but this may introduce other problems ("Why can Sharon from Sandstone University access this document while John from Gum-tree University cannot?"). A better alternative may be to agree on a common but simple semi-hierarchical set of access policies across a federation, for example you could have six classes:

A. Public Domain resources
B. Federation resources:
   (1) All members; (2) Undergraduate students, (3) Postgraduate students, and (4) Staff
C. MyUniversity resources:
   (1) All members; (2) Undergraduate students, (3) Postgraduate students, and (4) Staff
D. MyUniversity+Faculty resources:
   (1) All members; (2) Undergraduate students, (3) Postgraduate students, and (4) Staff
E. MyUniversity+Faculty+Course resources:
   (1) All members; (2) Undergraduate students, (3) Postgraduate students, and (4) Staff
F. Special class using XACML rules & policies

For example, if an author only wants federation staff and post-grads to access (or edit, rank, etc.) his work, he would select "Access protection class: B3" (or "Edit protection class: B3" etc.). Note that C1/2, D1/2 or E1/2 will never be sufficient. If a teacher wants to make content available to his students, he would choose "Access protection class: E2 (Course Physics101)". That is, all enrolled students would be able to access it. Each of these classes could be applied to all content (metadata, resources, and software services) and all actions (access, edit, rank, comment, etc.) although a smart GUI could hide these advanced options if they are not available to a user.

For all other cases, class F can be used, as it allows a user to specify specific rules, similar to a modern-day email program (like MS Outlook) specifies the rules. For example, it could only allow access to a list of specific persons (i.e. an Access Control List). Or the document is only available to members of a certain Virtual Organization (VO). Class F, as it is very specific, will probably not allow you to determine your access rights automatically, unless you are willing to make all necessary attributes part of your ARP. So if your VO uses repository X for storing all documents, which are only available to members of the VO, you should make your VO attribute part of the default ARP, so you can always see and access these documents without any additional SP-IdP redirection to retrieve these attributes.

4.5 Batch downloading

When performing a search across many repositories, the metadata is often not sufficient to actually determine the value of a document and you still need to access it. As described in Subsection 44.1, you can access each document separately, which is quite cumbersome. Instead, you would prefer to have a service, which is part of the IdP, which can do this for you automatically (R14). So you would store potentially interesting search results in your 'shopping list', sent this list to your IdP, which would then retrieve these resources on your behalf, informing you when ready. Using the approach presented in 44.3, where the AFS is already part of the IdP, would be an easer way to implement this functionality.

5. CONCLUSION

'Backing Australia's Ability' is our funding program, which funds many projects sharing the goal of increasing Australia's research effectiveness. Our proposed solutions to the AFS as described in Section 4 achieve this by:

- Making search of protected content more efficient by informing users beforehand which resources are available to them, so they do not try to access something and receive an "Access denied" message.
- By filtering inaccessible results, users can review a shorter list, so it will reduce the time they need to review information.
- Downloading resources in batches, or even by an agent running in the background which retrieves information for you, would save much time as well (imagine accessing 50 documents manually by going to the repositories one-by-one, versus downloading one big file containing all the information).
- If researchers have a more efficient and secure way to share content, they will be more willing to open their work to peers, and material that might have remained hidden on someone's PC otherwise now becomes available.

A minor disadvantage to the proposed AFS solution is that adding SAML assertions to search queries will require extra processing time on the AFS side as well as the repository for each query, although caching the SAML assertions by AFS and repository after the first access should reduce this processing time considerably, as these assertions do not need to be verified again. Additionally, the repository not only needs to find matching results, but also needs to add accessibility conditions. This could only be partly hidden from the user by using an approach as described in Section 44.3.

In all, we feel that the SAML-XACML model as described in this paper will increase research effectiveness considerably, and it only gives a glimpse of the functionality that we might expect in the future.

ACKNOWLEDGEMENT

The authors wish to acknowledge the contribution of Bruc Liong, Chi Nguyen, Dr.Yoichi Takayama, Scott Wilson and Neil Witheridge.

REFERENCES

[1] DigitalIDWorld, Identity is Center, 15 Identity Related White Papers, http://tinyurl.com/dshdh
[2] OASIS Security Services (SAML) TC, "SAML V2.0", http://tinyurl.com/a3h5l
[3] Cover, R, The CoverPages, http://xml.coverpages.org/ni2004-07-15-a.html
[4] OASIS eXtensible Access Control Markup Language (XACML) TC, "XACML V2.0", http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
[5] Liberty Alliance Project, Digital Identity Defined, http://www.projectliberty.org/
[6] Shibboleth Project - Internet2 Middleware, http://shibboleth.internet2.edu/
[7] Meta Access Management System project, http://mams.melcoe.mq.edu.au/zope/mams
[8] Collier, G., Robson, R., "Digital Rights Management for Research and Education", http://tinyurl.com/8dnnl
[9] Search/Retrieve Web Service, Z39.50 International, The Next Generation, http://www.loc.gov/z3950/agency/zing/srw/
[10] Dublin Core Metadata Initiative, http://dublincore.org/
[11] McLellan, D, "Very Dynamic Web Interfaces", February 2005, http://tinyurl.com/6ytrs
[12] Dalziel, J, Vullings, E, "MAMS and Middleware: The Easily Solved AuthN, AuthZ, Identitty, SSO, Federation, Trust, Security, Digital Rights and AAP Cluster of Problems," Educause 2005, Auckland, New Zealand, http://tinyurl.com/83smr

Erik Vullings (MSc 94 - PhD 99) is the Program Manager of the Meta Access Management System, a 3-year AUS$4.2 million research project which is developing demonstrators of federated Identity & Access Management solutions to increase the research effectiveness of Australia's Higher Education sector.

James Dalziel (BA Hons 94 – PhD 2000) is Professor of Learning Technology and Director of the Macquarie University E-learning Centre of Excellence (MELCOE). He is Chief Investigator on the MAMS project, and has been involved in a range of other e-learning and IT infrastructure projects such as COLIS and LAMS.