



Discovery and Access Work Package 1.

**Improve repository deposit rates, sharing and re-use by
allowing end-user control over who can access what**

Final Report

Version 0.1, MARCH 2007

Lead Investigator: Andrew Treloar
Development by: Sio Fai Keong (Sio.Keong@its.monash.edu.au)
Prepared by: Andrew Treloar and Sio Fai Keong

Executive Summary

This document describes the work completed in order to fulfil the requirement of the DART working package DA1.

The objective of the final report is to

- document the aims and objectives of the DA1 work package in terms of its contribution to the DART project,
- provide a detailed description of the work undertaken over the course of the project in order to achieve those aims,
- provide an archival record of any software, configuration instructions, hardware platforms that may have been built

Table of Contents

1	Introduction	4
2	Project Milestones	5
2.1	System Testing.....	5
2.2	Users' Access Control Requirements.....	5
2.3	Implementing Access Control Requirements in XACML.....	5
3	Project Outcomes	6
3.1	Overview.....	6
3.2	Architecture Issues.....	6
3.2.1	Collaborations	8
4	Archival Storage of Project Deliverables.....	9
5	Recommendations	10
6	Terms of Reference.....	11
6.1	Glossary	11
7	Report Signoff.....	12

1 Introduction

The requirements originally listed in the bid document were that DART would need to provide depositors with control over access to their contributions. This might be by a range of attributes, including user, role, time or location. The requirements for this work package were to improve repository deposit rates, sharing and reuse by allowing end-user control over who could access what.

The original activity statement was as follows:

- Enable controlled access to distributed archives through the RDF-data store – to resources and data stored both within SRB, DSpace and Fedora repositories
- Identify generic range of access control requirements
- Develop XACML statements that encode these
- Implement XACML statements in ARROW software

However, it became quickly clear that it would not be possible to achieve these requirements in this way.

2 Project Milestones

The original project plan had three milestones:

- System Testing including LDAP and XACML integration
- Identify users' access control requirements
- Implement access control requirements in XACML

2.1 System Testing

This phase identified early on that the implementation of access control in the Fedora repository software at the time had a number of deficiencies. These, and the solution developed are discussed in the next section.

2.2 Users' Access Control Requirements

The plan was for these requirements to come from the engagement of the information management professionals detailed in work package CR4. In practice, the engagement of these staff was focussed on a range of other aspects of the researchers' practice (see the CR4 work package for details) and they didn't get around to eliciting precise access control requirements.

In addition, it became clear from the CR4 work and from anecdotal evidence that in the collaboration space, the access control requirements were both fairly simple and dependent on infrastructure that wasn't yet in place. The simplicity was based on a stated requirement to restrict access to members of the research team. The required infrastructure for anything more nuanced is only now being put into place as the Access Australia Federation (AAF).

2.3 Implementing Access Control Requirements in XACML

Because of the issues identified above, this part of the work package was only completed to 'proof-of-concept' stage. In addition, the tools available at the time for constructing XACML policies were extremely primitive, requiring the use of an XML editor. It was not felt that this was something that end-users would want to interact with.

3 Project Outcomes

3.1 Overview

After meeting with some research teams and collaborating with ARROW project team, several key features that were missing from the FEDORA security module available at the time were identified. Specifically, in order to meet the needs of the ARROW and DART projects, the Fedora repository needed to be able to make authorization decisions based on particular fields in MARCXML (ARROW) and other metadata (DART).

After studying the FEDORA architecture, it was concluded that a software patch needed to be developed to improve the authentication in Fedora repository, to fulfil those requirements. In particular, based on feedback from the Fedora development team, it was confirmed that the version of Fedora we were using (2.1 and 2.1.1) could only make authorization decisions based on a pre-defined and limited set of attributes defined in FOXML. So the proposal that the repository might make authorization decisions based on particular attributes in MARCXML was not possible without further work.

By enhancing the features inside the Fedora XACML module, it made control access to resources and data stored within Fedora repositories more flexible, rather than using the insufficient default attribute values provide by Fedora itself.

After this was done, the gathered access control requirements are encoded in XACML format and made available to the FEDORA system.

3.2 Architecture Issues

XACML

XACML is an OASIS standard that describes both a policy language and an access control decision request/response language. The policy language is used to describe general access control requirements, and has standard extension points for defining new functions, data types, combining logic, etc. The request/response language lets you form a query to ask whether or not a given action should be allowed, and interpret the result. The response always includes an answer about whether the request should be allowed using one of four values: Permit, Deny, Indeterminate (an error occurred or some required value was missing, so a decision cannot be made) or Not Applicable (the request can't be answered by this service).

In brief there are two major modules inside XACML architecture: Policy Enforcement Point (PEP) and Policy Decision Point (PDP).

Fedora

Versions of Fedora prior to 2.1 did not provide for the possibility of authorizes incoming requests against XACML policies enforce engine. Fedora 2.1.X introduced a completely new authentication system using XACML engine developed by Sun Microsystems, which is extendable and configurable. However, due to the limitation of technology, the authorization module inside Fedora only implements XACML 1.1 Standard.

Authorization process within Fedora

As described in the below figure, there are few major actors within Fedora Authorization process:

- **Application** can be a GUI front end like Fez, Elated or VITAL making a request for a resource inside Fedora system.
- **Resource** can be the whole FEDORA object or a particular datastream
- **Policies** are in XACML format and define some authorization rules. Fedora pre-defines some system wide policies.
- **Attributes** are used by PDP to make the authorization decision, however FEDORA will only retrieve limited attributes defined in FOXML (FEDORA XML) only.

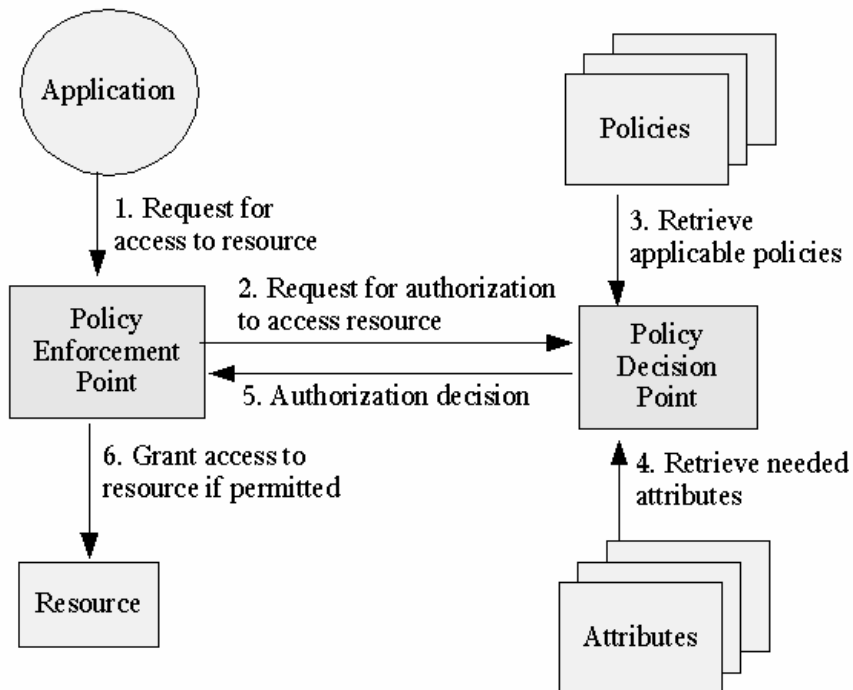


Figure 1, Authorization inside Fedora

3.2.1 Collaborations

In 2007, A DEST funded project “Research Activityflow and Middleware Priorities Project” (RAMP) successfully published the first authentication plug-in for the Fedora. This new security architecture is now being actively investigated by the Fedora core development team as a candidate to replace their current system. The ARROW project has decided to adopt this solution for the VITAL software. Further details at <http://drama.ramp.org.au/>

4 Archival Storage of Project Deliverables

As a result of this working package, a new fedora security extension has been implemented and encoding XACML statements has been completed. A brief demonstration of the work was presented to ARROW team members and DA1's Chief Investigator

Please download the software patch and follow the installation instruction at <http://www.dart.edu.au/>

As described before Fedora only retrieves limited attributes for making the authorization decision. By using the developed software patch, it will allow Fedora to accept attribute value from MARCXML datastream to make authorization decision.

Comment [AET1]: This will need to be updated to the final location of this software patch (preferably within the DA1 area)

5 Recommendations

After conducting the work described in this report, the limitation of the existing programming inside the Fedora 2.1.1 has been discovered, therefore a required software patch has been developed to satisfy the needs. This working package has shown how easily the FEDORA's XACML module can be extended to accept more information for making authorization decisions. However, since FEDORA only introduced this latest technology not long ago, key feature such as support for hierarchical authorization cannot be successful implemented.

Other future work that could be done in this area includes:

- A Policy editor
- Support for authorizing hierarchical related resource

6 Terms of Reference

6.1 Glossary

Acronym	Definition
XACML	XML Access Control Markup Language
FEDORA	Flexible Extensible Digital Object and Repository Architecture
RAMP	Research Activityflow and Middleware Priorities Project
MARCXML	MARC21XML schema

7 Report Signoff

It is agreed between

[Lead Investigator:](#)

and

[Chief Investigator: Andrew Treloar](#)

and

[DART Project Director](#)

That the **Final Report Document** for the [DART DA1 Improve repository deposits through end-user control](#) gives a full account of the work undertaken for the DART Project.

[Sio Fai Keong](#) [03 9902 0585](#)

- has been read and reviewed by all parties,
- shows that the [work package DA1](#) has been completed satisfactorily,
- clearly outlines the [functionality that was delivered](#).

Dated this [30th](#) day of [May](#) 2007

Signed by [Andrew Treloar](#) for
and on behalf of the Chief
Investigator

Signed for and on behalf of DART by
the Project Director [Andrew Treloar](#)