FeSL_Requirements

Required Features for first release

Timeline:

- Initial Partial Release by OpenRepositories '09 (May 1821)
- Final release by June 30, 2009

Overview

Create an alternate AuthN/AuthZ implementation for Fedora that can be bundled with Fedora and included in the installer. It will lend itself to integration with any Fedora client application.

FESL will run alongside existing Fedora code but will assume that the standard Fedora XACML component is turned off. In this context FESL will override the existing XACML implementation.

FESL will use the Muradora code as starting point and will be written with Jaas.

Vocabulary (Policy Templates)

Provide pre-vetted set of policy templates for:

- ACTIONS: Read, Create, Edit, Delete, and "Change Permissions"
  - One vocabulary covers all equivalent methods in SOAP and REST APIs (ie. policies decide at a higher level who can edit a datastream, rather than saying who can call the modifyDatastream SOAP method).
- TARGETS: Collections, Objects, and Datastreams
- SUBJECTS: User & Group
  - Assign permissions by User or by Group, regardless of where user attributes are coming from (ie. LDAP, Shibboleth, OpenId, CAS, etc.).

A general design principle of the FESL approach is that an object ideally belongs to one collection for authorization purposes, providing a simpler approach to policy interpretation. However, sample policy templates will be provided which show more complex examples with multiple parents for one object. FESL will look at an approach that allows an object to be assigned to a policy object in the policy repository using a special authorization predicate.

Authentication (AuthN)

- Provide documentation on how a GUI can allow a user to authenticate and pass the credentials to the Fedora server.
- Support LDAP, AD and Tomcat-Users by refactoring the existing servlet filters to make them more user friendly.
- Implement authentication in a modular way (ie. via Jaas) so that participating organizations can write their own adapters (ie. Drupal, OpenID, etc.).

Policy Manager / Authorization (AuthZ)

- Enforce policies at Datastream, Object and Collection level. (Rely on either RELS-EXT or Fedora's bundled RIsearch for evaluating collection memberships.) This is already supported in the Muradora AuthZ work by supporting the precedence rule, where the policy at at the lower level takes precedence over that at higher levels.
- The Muradora preference is to store the Policy in a separate Policy store (XML database) for efficiency and simpler implementation. The issue for existing Fedora installations which store policies in collection and object datastreams will be addressed in the initial FSL release with migration documents.
- Support for the new REST API.

General

- Keep the implementation stable & current
- Bundle solution with Fedora and include it in the installer
- Audit the Implementation for potential security flaws
- Support community innovation & allow people to completely replace the whole thing if they wish
- Ensure that there are points that allow for future development

Desirable Features (not required for first release)

- Support Shibboleth
- Support OpenID & OpenAuth
- Support Single Sign-on (SSO) - must be pluggable/overridable
- Allow for Custom AuthN
- FESL will implement an approach which will store policies in Fedora as Policy objects, which can then be subscribed to by the appropriate objects.
- Simple, intuitive, well documented vocabulary for controlling Read, Create, Edit, Delete, and "Change Permissions" for Collections, Objects, and Datastreams
- User interface & REST API for editing policies on Collections, Objects, and Datastreams
  - Allow repository managers to find out what policies apply to a given Object, Datastream, or Collection

Work Packages

In order to satisfy the Requirements for an initial release, the following work must be done.

- Add collections to PDP vocab
- Provide pre-vetted and fully documented set of policy templates (using templates provided by Muradora and UPEI)
- Provide and fully document API for Policy Manager
- Configure Muradora PDP to be aware of POLICY datastreams (*This will not be done as part stage 1)*
- Plug & play install experience (+Fedora Commons, MediaShelf)
- Add coverage for REST API
- Comply with Fedora 3.x
- Bundle solution with Fedora and include it in the installer (+Fedora Commons)
- Test & Refine Install Experience (+MediaShelf and UPEI)