# Open Repositories 2013 Session Proposal

# FRACAS: Fedora Role-based Access Control and Security

An implementation of role-based access control in the Fedora Enhanced Security Layer (FeSL)

Presenters: Gregory Jansen, UNC Chapel Hill Libraries

 Jill Sexton, UNC Chapel Hill Libraries

## The Problem of Success

As repositories mature into large collections with diverse content, workflow and user needs multiply. There comes a point at which the repository manager can no longer be involved in all operations. As large organizations take on digital stewardship, they need to create appropriate access controls to safeguard their collections. This is especially true as archives professionals migrate their practices into the digital realm and start managing digital collections. The successful repository will not remain a single bucket of materials managed by one or two people, but will provide a robust infrastructure to support the varied workflows of individuals in all areas of the curation lifecycle. Our presentation will explain how role-based access controls help create spans of control that allow collection managers, repository administrators and developers do their work.

### Fit and Function

We argue that you can only open up whole collections for greater sharing and reuse when you can also handle the exceptional cases and restrict certain items and actions. There are always items that are more restricted than the rest of the collection. If you cannot restrict access to these data, then you cannot share the rest of it. Therefore the security layer must be equal to the sophistication of the repository structure and its policies, otherwise opportunities for reuse are held hostage by the most restrictive objects. Fedora Enhanced Security (FeSL) is a sophisticated security framework that mirrors the Fedora API. Any action you can take in Fedora, you can also restrict with FeSL policies. You can extend FeSL to meet even more exacting requirements, making it the ideal instrument for protecting repository data and services. FeSL lets developers leverage the full power of the Fedora API to create services, without compromising the security of sensitive objects.

At the same time as overall security must be sophisticated, access control settings must be simple or they are incorrectly and rarely used. FeSL policies are too sophisticated for end users to author themselves. A large number of FeSL policies makes it hard for every day users, or even administrators, to generalize about the actions permitted them in the repository. However, a limited number of well-documented user roles are easy enough for end users and even repository visitors to understand. These roles can be assigned throughout the repository to different groups of people. Since the assigned user roles are tied to centrally managed FeSL policies, their enforcement can still be as sophisticated as necessary. The specific actions permitted to each role can be fine tuned over time, as new services and data are added. Roles help divide the work of controlling access into the frequently performed curation task of role assignment and the infrequent administrator task of policy maintenance.

As we take in new collections, we take on new audiences with their needs and preferences. Often these groups are best served by discipline or collection-specific access, through scientific data portals, finding aids, institutional repositories and digital exhibits. Visitors recognize these portals, which cater to their specific needs. Construction of appropriate access sites is routine in growing repositories. The question becomes how can we make the build-out and maintenance of portals as efficient as possible. We make the case that it is most efficient to leverage the ongoing work of the entire Fedora community. Securing access at the API level means you are free to select software stacks based on the Fedora API. You are also free to reveal that API to your community and let them create the portal. There are many existing front-end technologies for Fedora. With FRACAS and other FeSL implementations you are free to leverage these, while consistently enforcing permissions at the API level. FRACAS/FeSL makes your repository into a remix platform supporting rapid development of access sites and portals.

## Technical Overview

FRACAS is an implementation of the Fedora Enhanced Security Layer (FeSL) that is designed to meet the security needs of the Carolina Digital Repository. It secures the Fedora API based on the roles of the authenticated user with respect to the object being accessed or managed. Roles are assigned to groups throughout the repository tree and role assignments may be inherited to cover either a whole collection or a single object. The actions permitted to each role are specified in a set of repository-wide XACML policies. We will cover the architecture of FRACAS and the reference implementation of FRACAS in the Carolina Digital Repository (CDR). We will explain the straightforward layout of the access controls and how they neatly separate policy language from various points of enforcement within the repository tree. This mirrors a major design feature of XACML itself, which is to separate the policy enforcement points (PEP) from the policy definition point (PDP).

## Feature Set

### Use

* Hierarchical access controls make it possible to provide access to more objects
* Leverage LDAP, etc.. to grant roles to broad categories of people
* Makes it easy for curators to manage access controls
* Keeps policies centrally for administrators to manage
* Create private folders for content deposits and ongoing deposit streams
* Create review and publication work-flows
* Stops most uses of the all powerful Fedora administrator account and the risks that entails

### Re-use

* Enables remix projects based on repository content and Fedora APIs
* Stops use of proxies, enabling client access to Fedora data streams instead
* Allows third-parties to leverage select parts of the Fedora API on select parts of the repository
* Allows trusted third-parties to forward end-user credentials

### Reproduce

* Facilitates researchers finding objects and requesting access to them (list permission)
* Organize and protect research data in a custom hierarchy or content model
* Use APIs to create data-driven visualizations and demonstrate findings

## Session Audience

This presentation can fill a general session and we think role-based access control is an approach that can be applied to repositories beyond Fedora. While we want to spend significant time on the general benefits of this type of security, we are also open to presenting in the Fedora track. The ideal venue may depend upon the other submissions received this year, but the topic does seem like a good fit with the overall conference theme. Please direct suggestions and feedback to Gregory Jansen, greg\_jansen@unc.edu.