Hydra: A Technical and Community Framework
For Customized, Reusable, Repository Solutions

PAPER PROPOSAL for Open Repositories 2010

While repositories provide obvious benefits in hosting and managing content, it is equally clear that there is no "one size fits all" solution to the range of digital asset management needs at a typical institution, much less across institutions. A system that supports the submission, approval and dissemination of electronic theses and dissertations, for example, has demonstrably different requirements than a digitization workflow solution, an e-science data repository, or media preservation and access system.

There is a clear need in the repository community to readily develop and deploy content-, domain-, and institution-specific solutions that integrate the flexibility and richness of customized applications and workflows with the underlying power of repositories for content management, access and preservation.

Hydra is a multi-institutional, multi-functional, multi-purpose framework that addresses this need on twin fronts. As a *technical framework*, it provides a toolkit of reusable components that can be combined and configured in different arrays to meet a diversity of content management needs. As a *community framework*, Hydra provides like-minded institutions with the mechanism to combine their individual development efforts, resources and priorities into a collective solution with breadth and depth that exceeds the capacity of any single institution to create, maintain or enhance on its own.

Hydra's ultimate objective is to effectively intertwine its technical and community threads of development, producing a community-sourced, sustainable application framework that provides rich and robust repository-powered solutions as an integrated part of an overall digital content management architecture. Such solutions can meet the distinct needs of digital library, institutional repository, discipline repository, research, preservation and publishing workflows.

Now in its second year, Hydra's overall architecture has been established; its initial components have been developed; a half dozen applications using the framework are in operation across three institutions; and the collaborative has grown beyond its first set of founding partners to include a dozen institutions.

This paper will give the details of Hydra, including its underlying philosophy, its technical components and architecture, the functions it supports, and the current state of its community.

Hydra Philosophy

Underpinning Hydra are two fundamental assumptions: 1) no single application can meet the full range of digital asset management needs, and 2) no single institution or provider can resource the development or maintenance of a full set of solutions for the same needs. As implied by its very name, Hydra takes a "one body, many heads" approach to both needs. From a functional perspective, one body, many heads means that Hydra is designed to support tailored applications and workflows for different content types, contexts and user interactions (e.g., an ETD application, digitization workflow application, etc.), by building them from

- a common repository infrastructure,
- flexible, atomic data models, and
- modular services and configurable components

From a participants' perspective, many heads, one body also means…

- an open architecture built on a common core, with many contributors,
- collaborative ,working solutions that can be adapted and modified to suit local needs,
- a community of developers and adopters through which additional solutions and components will be shared, and the
- ability to integrate with institution-specific infrastructure and systems.

Altogether, this leads to rich applications, customized workflows, made up of modular components, and producing reusable objects.

Hydra Technical Framework: Primary Components and Functions

The primary components of the Hydra technical framework are:

- *Fedora*, providing a robust, durable repository layer for persisting and managing digital objects. Fedora's disseminator features allow us to place an abstraction layer between it and our Hydra heads, shielding an institution's applications from any future changes to the repository structure.
- *ActiveFedora*, providing a Ruby gem for creating and managing objects in Fedora, (developed by MediaShelf, LLC)
- *Solr* indexes, providing fast access to information about the institution's resources. Solr can be used as a *lingua franca*: content from any source that can generate a Solr index (perhaps an OPAC, or repository metadata records with different schema) can potentially be brought into a Hydra discovery environment.
- *Blacklight* plugin, a Ruby on Rails library that provides faceted searching, browsing and tailored views on objects
- *Hydra* plugin, a Ruby on Rails library that works with ActiveFedora to provide create, update and delete actions against objects in the repository
- A suite of web-based services, supporting granular actions against content to support their management, access and preservation (e.g., checksumming,

indexing, transform MARC to MODS, djatoka-based JPEG2000 image streaming)
- *Hydrangea*, a web application that bundles all the Ruby on Rails components and hooks to web services into a single package, with a library of screen widgets and user interactions to support various content management actions (e.g., upload file, edit metadata, change permissions)

Taken altogether, these technical components support the following five primitive functions:

Deposit – uploading simple or multi-part objects, singly or in bulk
Manage – editing and updating an object's content, metadata and permissions
Search – full-text and fielded search supporting user discovery as well as administration
Browse – sequential viewing of objects by collection, attribute or ad hoc filtering
Deliver – viewing, downloading and otherwise disseminating objects through Hydra applications, web services and third party applications

Finally, these components rely on several background services:
- *authorization*, provided by FESL (Fedora Enhanced Security Layer – a new Fedora framework service part funded by the Hydra partners and others in the community)
- *authentication*, provided by local institutional systems
- *workflow*, which can either be provided as a bundled part of the Hydra framework, or provided by a local institutional systems

Live Applications
The total combination of these components, functions and background services is a comprehensive set of "lego bricks" that can be combined and recombined to rapidly develop and deploy Hydra heads, tailored to different content types and workflows. As of February, 2010, less than a year after development first started, Hydra applications have been deployed at three separate institutions to support five use cases:
- an electronic thesis & dissertation solution
- an institutional repository front end access system
- a digital libray (combined repository and OPAC) discovery application
- digitized image and manuscript delivery system
- a digital archives management system

Over the next year, this suite of solutions will extend to include
- an application for describing, managing and presenting born digital special collections
- an end-to-end digitization workflow application
- an open access institutional repository application

- a scientific data curation system
- a personal repository application

Hydra Community Framework & Participants

Hydra has from its inception been designed to provide a generalizable, portable framework that would meet the needs not only of the three original institutions, but also those of a wider community. Originating as a multi-institutional project spanning three universities (Hull, Stanford and Virginia), and with support from Fedora Commons, in early 2010 Hydra began to expand to include like-minded institutions with similar needs, technical infrastructures and complementary systems.

Consider the use of content models as an example. The Hydra team has spent considerable effort designing a common approach to leverage Fedora content models. After a number of false starts that attempted to define a uniform, standard data model, Hydra has settled on an approach which fits the 'high reuse' philosophy. The project does not offer a single comprehensive content model for each category of object that a repository might store; rather it offers a content model for core metadata which can (and arguably should) form part of almost any object's structure and then supplements this with one or more further content models which provide for the object's particular content and/or local institutional variations in structure. Thus the overall content model is actually an aggregation of reusable components.

A number of institutions worldwide have already seen that there would be positive benefits in adopting this approach of reusable components and contributing to some of the Hydra developments taking place; in particular:

- shared content models,
- shared datastream structures, and
- shared code.

In fact, much of the approach and some of the components that Hydra is developing are relevant and useful in non-Hydra environments.

Summary

This paper will provide an overview of Hydra's philosophy, architecture, and components, as well as demonstrations of various Hydra installations. The paper will also provide a progress report on Hydra development to date and its overall roadmap, as well as provide observations on the successes and challenges of community-based development of shared repository solutions.