Next-Generation Technical Services (NGTS)
POT 1 LT 1C

Proposed Model for Systemwide Digital Asset Management System (DAMS) with Discovery and Display

Final Report

February 8, 2013

POT 1 LT 1C Members
Todd Grappone, UCLA (Co-chair)
Declan Fleming, UC San Diego (Co-chair)
Erik Hetzner, California Digital Library
Sue Perry, UC Santa Cruz
Brian Tingle, California Digital Library

Table of Contents

Executive Summary

The Power of Three 1 (POT 1) Lightning Team 1C (LT 1C) was presented with the following charge: Develop a model for a systemwide Digital Asset Management System with a discovery and delivery interface. LT1C consisted of Todd Grappone (UCLA), Declan Fleming (UCSD), Erik Hetzner (CDL), Brian Tingle (CDL) and Susan Perry (UCSC).

To meet the charge, the Lightning Team met in Oakland in September 2012 to develop a technical model, discuss requirements and lay out a framework for developing recommendations. Our initial meeting identified some high level goals both technical and philosophical:

1. The DAMS application should be a modular web application, built using principles of service-oriented architecture (SOA) and Representational State Transfer (REST).
2. The model should include best of breed components with open source tendencies that have broad adoption and community support.

Based on the second goal, the team decided to broaden the discussion. A blog was developed to facilitate community engagement and we held a "Birds of a Feather" discussion at the Digital Library Federation 2012 Fall Forum.

After our initial discussions, the team identified four technologies based on the criteria we outlined and conducted a thorough evaluation of each. The four technologies identified were Project Hydra, Islandora, Nuxeo and Alfresco. All of these products met our initial criteria and are in use at one or more University of California campuses. Pilot installations of each system were deployed and evaluated based on requirements developed by POT 1 LT 1A and members of LT 1C.

The LT recommends a progressive model for a system wide DAMS. We would like to address the immediate needs of UC Libraries without a DAMS while striving for a 10 campus future. In order to address short-term needs, the LT recommends adopting a DAMS with vendor support, Nuxeo. While we believe that achieving and ultimate goal of a full 10 campus DAMS would be achieved by adopting a

Fedora based system, acute needs of system libraries need to be addressed immediately. A technical model was developed and refined to reflect our recommendations (see Fig. 1).

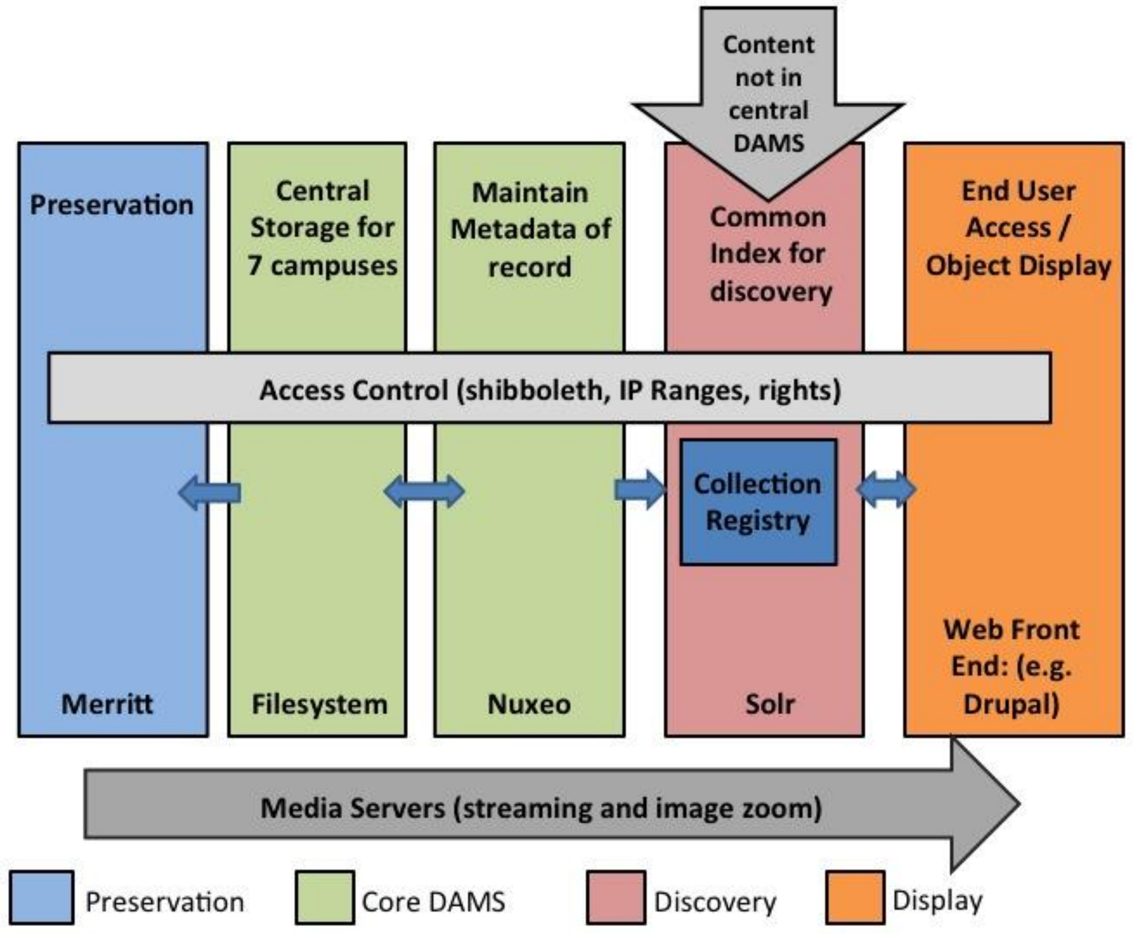
Content not in central DAMS
Preservation
Central Storage for 7 campuses
Maintain Metadata of record
Common Index for discovery
End User Access / Object Display
Access Control (shibboleth, IP Ranges, rights)
Collection Registry
Merritt
Filesystem
Nuxeo
Solr
Web Front End: (e.g. Drupal)
Media Servers (streaming and image zoom)
Preservation
Core DAMS
Discovery
Display

Figure 1: Technical Model

At the conclusion of writing this report, the LT was asked to develop a preliminary project plan and estimated budget for developing the DAMS. We worked with Joan Starr (CDL) and Lisa Spagnolo (UCD) on these estimations. Please see Appendix F for those details.

Report Background

> *1.1.10. Develop potential models for systemwide DAMS with Discovery and Display.*
>
> POT1 Lightning Team 1C Charge [PDF]

On September 14th 2012, the POT 1 Lightning Team 1C met face to face in Oakland to discuss and evaluate models for a systemwide DAMS with a discovery and display interface that would meet the needs outlined in the POT 1 Lightning Team 1A [PDF] and Lightning Team 3A [PDF] reports and sketched out a model in representing the consensus of our thinking.

We identified the following critical areas for the architecture to cover:

- Metadata management
- Index (search crawl, project registry)
- Media server
- Authentication
- Storage/persistence layer
- Import and export
- Discovery and display
- Workflow

After the initial meeting of LT 1C , a blog was developed to share our thinking with the community. We also initiated a series of meetings with solution providers and vended solution consultants to discuss a core group of technologies and their match to the technical architecture and requirements documentation. LT 1C met with representatives from Project Hydra and Islandora at the Digital Library Federation 2012 Fall Forum and had conference calls with representatives from Nuxeo and Alfresco (notes are available regarding these meetings) as well as held a discussion with Patrick McGrath, Chris Hoffman, and Richard Millet from UC Berkeley about their use of Alfresco (Research Hub) and Nuxeo (CollectionSpace).

The primary outcome of the LT 1C September 2012 meeting was agreement on the following technical model:

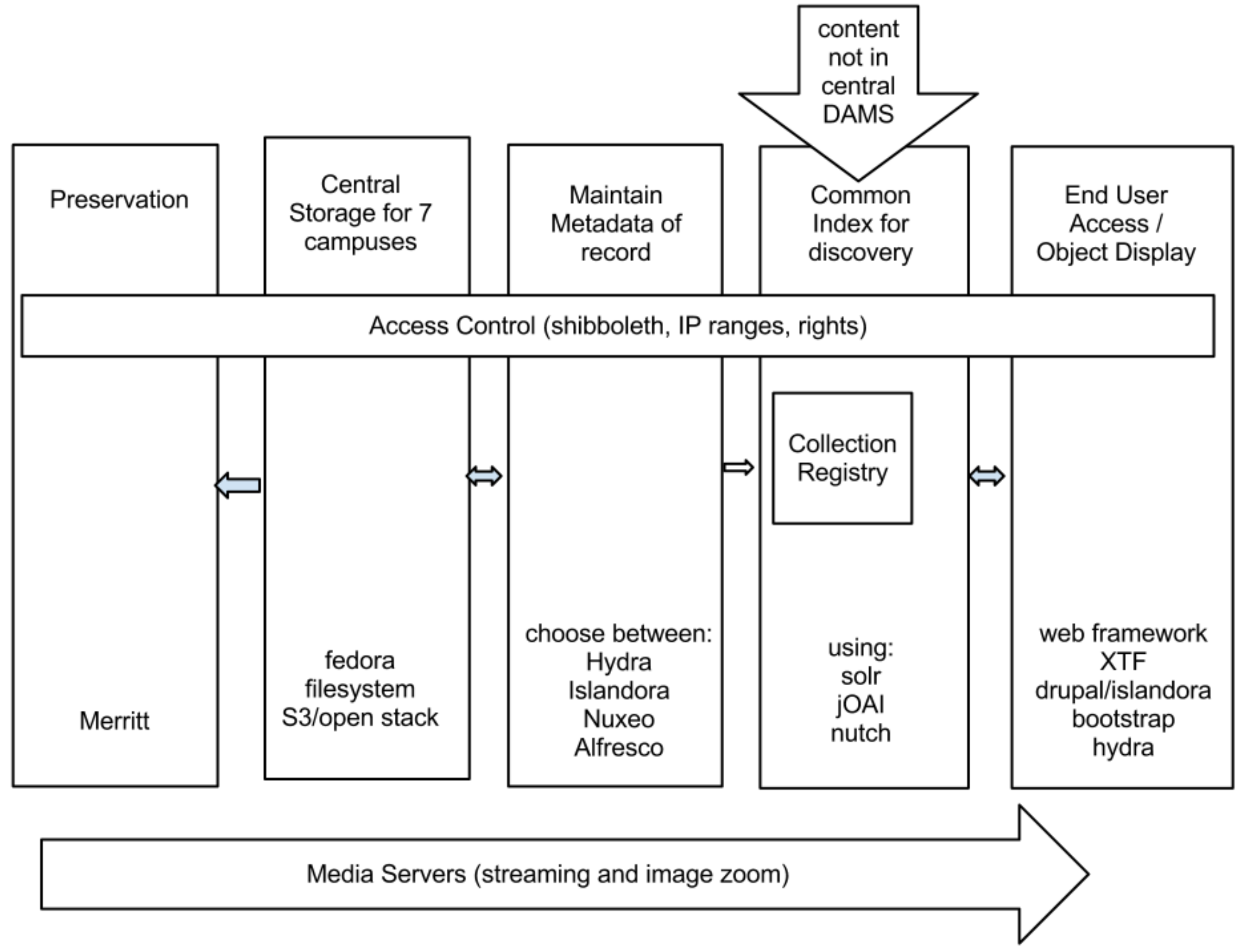
content not in central DAMS
Preservation
Central Storage for 7 campuses
Maintain Metadata of record
Common Index for discovery
End User Access / Object Display
Access Control (shibboleth, IP ranges, rights)
Collection Registry
Merritt
fedora
filesystem
S3/open stack
choose between:
Hydra
Islandora
Nuxeo
Alfresco
using:
solr
jOAI
nutch
web framework
XTF
drupal/islandora
bootstrap
hydra
Media Servers (streaming and image zoom)

Figure 1: Technical Model

1.1.11. Determine if there is an existing product that meets the requirements.

1.1.12. Decide on acquire vs. build; submit to NGTSMT / SOPAG / CoUL for sign-off

As a result of the meeting and modeling we had a good deal of clarity on our technology choices. For example, using Merritt for digital asset preservation and Solr for the common index that includes materials from both the systemwide and local DAMS are clear choices.

What was less clear in our September meeting was what technology should be recommended for managing content during the digitization and description process, such as for the 300 extant collections with identified digital asset management needs (20 of which are in urgent need of some solution).

The DAMS solution must simplify for staff members at campus libraries the activity of managing and processing digital assets under library control. The entire workflow--from digitization of files through the act of developing and maintaining metadata and publication in appropriate access and preservation systems--needs to be supported in the DAMS.

Some options were eliminated quickly during the discussions, e.g., taking a system from a campus and scaling it up for all the rest of the system was estimated to be too ambitious. CONTENTdm was discussed, but the consensus seemed to be this was a short-term option at best, as it does not meet many of the key POT 1 LT 1A and LT 1C requirements:

- Modular solution
- Best of breed component w/ open source tendencies
- Broad adoption w/ community support

The group identified four possible solutions (1.1.11 existing products) selected from two general approaches (1.1.12 build or acquire). We have determined that no DAMS solution exists today that meets all of the requirements expressed by LT 1A. To that end we will need to balance some product development with any selected solution. For example, neither Alfresco nor Nuxeo have discovery and display interfaces. They both are excellent managers of content files and some metadata types, however they would each require the development of a public interface, probably from scratch or integration with an existing CMS such as Drupal. To that end, Hydra and Islandora are types of solutions, though different from a pure DAMS. Islandora can be a monolithic content and display system, assuming you use the content types with existing solution packs - namely images and PDFs. Hydra is a framework based on Ruby on Rails that is highly "opinionated" meaning that there are a lot of assumptions about how to create interfaces because of its Rails and Solr/Blacklight underpinnings. It does not come "out of the box" ready to support content, but there are good examples for how to get started - namely Penn State's ScholarSphere. Additionally, both Alfresco and Nuxeo claim open source, but upgrade paths and development communities require a subscription and both are tied to a single vendor that currently controls development.

Articulation of Assumptions

Central storage for seven campuses:
The proposed model includes a DAMS that will be maintained centrally and be available to campuses without a local solution. Central storage of files in this model is based on a standard file storage system with Merritt as a preservation back end service.

Maintain metadata of record and support processing workflows:
Two open source library community projects (Project Hydra and Islandora) and two open source enterprise content management systems (Nuxeo and Alfresco) were evaluated to provide the metadata management system.

Common index for discovery:
In the proposed model, information from the centralized DAMS together with information from library resources held outside of the DAMS will be combined into a centrally maintained Solr index. (POT 1 LT 3C has conducted an evaluation of this approach, which is discussed in their final report.) A collection registry would be a database that would be maintained by campus libraries and contain metadata information used to OAI harvest and web crawl content into the central index. This would be content not in the systemwide DAMS.

End user access systems:
In the proposed model the central index exposes an API, either a read only Solr handler and/or a custom API. Campuses would be able to use this index with the API from local sites, and a public portal to the UC Libraries Digital Collection could be created in any language or framework that has a Solr library.

Both Hydra and Islandora are understood to the team to be roughly consistent with the model in as much as they use Solr indexing and have industry standard web frameworks on top of Solr.

Access control:
An access control layer that can use Shibboleth/InCommon and IP address ranges is required across all the layers of the proposed model.

Media servers:
Media servers including for audio, video, and zooming images that can be used by any content deposited into the system are part of the model.

Decision Criteria

The members of the lightning team identified the following principles during our September meeting in Oakland:

- Modular solution
- Best of breed component with open source tendencies
- Broad adoption with community support

Rather than examine every item in the functional requirements list developed by LT 1A, the team identified a group of key differentiating features and analyzed how difficult it would be to implement the required functionality in four identified technology options: Project Hydra, Islandora, Alfresco, and Nuxeo. The key differentiating features include:

- Security access control
- Support for complex objects (arbitrary hierarchy of mixed type)
- Audio/video support
- Workflow support (suppress in-process collections, restrict master/print quality files)
- Metadata editing (including support for custom fields and batch editing)

The following are items from the POT 1 LT 1A critical requirements list that should provide further opportunity for differentiating features and customization level-of-effort:

- Ability to upload preservation/master copies of content files (e.g., TIFFs for image-based objects) and have derivative/service copies automatically generated when possible (e.g., JPEG thumbnails) instead of submitting different copies.
- Metadata embedded in content files to be extracted on import into the system, for ease of working with and adding in the resulting record.

- Ability to batch edit metadata records associated with content files (i.e., search and replace on particular data elements; replace data across some subset of objects in a given metadata element).
- Integrated use of authoritative vocabulary terms from LC, Getty, VRA, TGN, locally created, etc.
- Ability to specify different levels of access restrictions in the metadata, for example: 1) reading room/department only, 2) library only, 3) specific UC campus only, 4) all UC campus users, 5) specific authorized users only, and 6) public access.
- Ability to limit what is shown to end users at different access levels (e.g. only the metadata record available to the public but a given digital object and the metadata record are should be available to campus users).
- A set of APIs to interact with the repositories where the objects are being managed.

Results and Recommendation

LT 1C conducted thorough evaluations of the four identified technologies:

1. Project Hydra
2. Islandora
3. Alfresco
4. Nuxeo

Recommendation: After discussing the evaluations the LT 1C recommends that Nuxeo be implemented as the first step towards a systemwide digital asset management system strategy.

While we had many factors to weigh in making a recommendation, the key factor is time. For the UC Libraries Digital Collection to be able to sustain momentum, it is assumed that systems to support next generation collection development must be deployed quickly. The 20 collections (estimated at 5.65 TB) that have been identified as high priority for a DAMS need to be available online -- in a system that integrates these 20 collections with related materials in local systems and Calisphere -- within 18 months, or there is a risk of the project stalling.

In order to get the at-risk assets in a supported DAMS and under digital library control in a timely fashion, LT1C recommends the use of an off-the-shelf digital asset module of Nuxeo, with limited customization. An estimated $12k/year should be budgeted for Nuxeo Connect subscription, which includes access to Nuxeo Studio. This provides an on-line configuration wizard for developing content and metadata models. CDL should allocate 3 TB of storage at the inception of the systemwide DAMS project.

The LT 1.C recommendation is for “acquire” rather than build, however the recommended product is not a turnkey solution. Work on developing workflows, metadata models and a web front end will need to be completed for a systemwide DAMS to be implemented. We added members to the LT to match requirements developed by LT 1.A, results of the LT 3.B report and informed by the collections outlined in LT 3.A with the intent of developing a project plan, time, effort and cost estimations for implementation of Nuxeo as an initial systemwide DAMS with Discovery and Display. The group consisted of the LT 1.C group with the addition of Patrick McGrath, Patrick Schmitz and Richard Millet from UC Berkeley, Joan Starr from CDL and Lisa Spagnolo from UC Davis. The team from Berkley have worked extensively with Nuxeo and offered insight into the tasks associated with the Nuxeo while Joan and Lisa developed Appendix F, a rough outline of a project plan and budget.

This short-term tactic of adopting a non-library-industry standard technology should be combined with a long-term strategy of participating in the library-specific Project Hydra and Islandora communities utilizing the Fedora (or Fedora Futures) repository framework. Several campuses and institutional partners are exploring or investing in these systems, and this work deserves our close attention and support. Nuxeo should be implemented with an eye toward an exit strategy that does not lock us in to their product for the long-term. While Nuxeo is an open source product, long-term alignment with library developed and governed communities would presumably align closer with systemwide goals.

It is recommended that the short-term (18 months to 3 years) DAMS with discovery and display implementation should be hosted and managed at the California Digital Library, in close consultation with collection owners. Long-term participation in a Fedora based system should involve co-development with library-based programs, as LT 1C believes this to be the best approach for a systemwide solution, due to the international and intra-UC digital library efforts involving Fedora. Assuming our recommendation regarding long-term participation in Fedora is accepted, we also recommend the planning and development of a business model for Fedora based repository support. We note that the Fedora Futures initiative launched at the Coalition for Networked Information 2012 Fall Forum will lead Fedora to be the type of non-preservation repository that the UC Libraries should ultimately adopt to complement the Merritt preservation repository currently running at CDL.

Supporting Documentation

Blog: https://blogs.library.ucla.edu/systemwidedams/

Appendix A: Vendor Comparison

	Requirements	Alfresco			Nuxeo			Hydra			Islandora		
		Yes	No	Partial	Yes	No	Partial	Yes	No	Partial	Yes	No	Partial
1	Security Access Control	X			X					X	X		
2	Complex Objects Support (Arbitrary Div Hiearchy of Mixed Type)			X	X			X			X		
3	Audio/Video Support	X			X					X	X		
4	Workflow support (suppress in process collections, restrict/print quality files)	X			X					X	X		
5	Metadata editing (including support for custom fields and batch editing)			X	X					X	X		
6	Modular Solution	X			X			X			X		
7	Best of breed component w/open source tendencies	X					X	X			X		
8	Look for broad adoption w/community support	X					X	X					X
9	Ability to upload preservation/master copies of content files (e.g. tiff's for image-based objects) and have derivative/service copies automatically generated when possible (e.g. jpeg thumbnails) instead of submitting different copies	X			X					X	X		
10	Metadata embedded in content files to be extracted on import into the system, for ease of working with and adding in the resulting record.	X			X					X	X		
11	Ability to batch edit metadata records associated with content files (i.e. search and replace on particular data elements; replace data across some subset of objects in a given metadata element)		X		X					X	X		
12	Integrated use of authoritative vocabulary terms from LC, Getty, VRA, TGN, locally created, etc.	X			X					X	X		
13	Ability to specify different levels of access restrictions in the metadata, for example: 1) reading room/department only, 2) library only, 3) specific UC campus only, 4) all UC campus users, 5) specific authorized users only, and 6) public access.	X			X					X	X		
14	Ability to limit what is shown to end users at different access levels (e.g. only the metadata record available to the public but a given digital object and the metadata record are should be available to campus users).	X			X					X	X		
15	A set of API's to interact with the repositories where the objects are being managed	X			X			X			X		
16	Search and Discovery		X			X		X			X		

Appendix B: Alfresco Evaluation

Alfresco is an enterprise content management system, with an open source community edition as well as a supported edition available for a fee. Cloud-based as well as on-premise versions are also available for a fee. It has an active library development community in Europe but library involvement is more limited in the United States. UC Berkeley's Research Hub project uses Alfresco and are launching a production service to push content into Merritt for archiving soon. Most library sites have created a public front-end using Drupal and Solr. Research Hub uses the out-of-the box interface for contributors to manage their digital objects.

High-Level Requirements:

- **Security access control**

Available

- **Support for complex objects (arbitrary div hierarchy of mixed type)**

This is limited at this time, though this is on their development roadmap. ResearchHub support staff see this as a limiting issue.

- **Audio/Video support**

Available

There are no transformations or metadata extractions out-of-the-box.

- **Workflow support (suppress in process collections, restrict master/print quality files)**

Available. Each object or collection can have a variety of simple or complex workflows attached to it.

- **Metadata editing (including support for custom fields and batch editing)**

Custom fields are available. Batch editing is not available out of the box. Batch loading of data (for example, from a spreadsheet) is only available to system administrators. This has been issue for the Research Hub staff.

Additional Requirements:

- **Ability to upload preservation/master copies of content files and have derivative/service copies automatically generated when possible instead of submitting different copies**

Not available out of the box, but one of their partners, Technology Services Group, has created the Open Migrate Toolset which provides this function.

- **Metadata embedded in content files to be extracted on import into the system, for ease of working with and adding in the resulting record.**

Available but limited. Each file gets a content type, size, creator assigned to it. Not on the same level as JHOVE though.

- **Ability to batch edit metadata records associated with content files (i.e. search and replace on particular data elements; replace data across some subset of objects in a given metadata element)**

Not available out of the box, but could be accomplished with the web scripting function. It is unclear whether users can run these web scripts or if system administrators are required to run them.

- **Integrated use of authoritative vocabulary terms from LC, Getty, VRA, TGN, locally created, etc.**

Available

- **Ability to specify different levels of access restrictions in the metadata, for example: 1) reading room/department only, 2) library only, 3) specific UC campus only, 4) all UC campus users, 5) specific authorized users only, and 6) public access.**

Available

- **Ability to limit what is shown to end users at different access levels (e.g. only the metadata record available to the public but a given digital object and the metadata record are should be available to campus users)**

Unclear whether this is available out of the box or not.

- **A set of APIs to interact with the repositories where the objects are being managed**

Available. These API's are supported and fully documented:

- RESTful API
- Alfresco Repository FreeMarker Template API
- Alfresco Repository JavaScript API
- Alfresco Surf Platform API

Appendix C: Hydra Evaluation

Draft Hydra Evaluation for LT 1C
Overall evaluation

Hydra is a platform for the development of repository software. It consists of a number of Ruby libraries for building Ruby on Rails applications to provide repository solutions. These libraries integrate Fedora (for the repository backend) with Blacklight (which provides discovery and display). Hydra's platform requires the user to develop the workflows and metadata-editing capabilities that are required by repository solutions.

Hydra seems to have an active community and is a well-organized project. The platform is tested and mature. The project's lines of communication are open and transparent. It is across-institutional development effort. As an open source model, it is based more upon the community developed product than the model of a product (which is at least partially open source) developed by a single vendor. (Think apache httpd v. mysql). Hydra seems a better organization than the alternatives that we have looked at to get involved with.

Comparing Hydra with a system like Alfresco or Nuxeo is difficult. Hydra is a platform upon which a DAMS could be built, with Fedora as the backend and a Ruby on Rails application as a front-end. But the user interface that we need (with rich metadata editing, authority control, automatic generation of derivative copies, management of object relationships, etc.) is not part of Hydra. All of this would require additional development.

The discovery layer, Blacklight, does provide a reasonably complete user-interface which could, with some development effort, satisfy all the discovery and display requirements identified by POT 1.

Differentiating requirements

Ability to upload preservation/master copies of content files and have derivative/service copies automatically generated when possible instead of submitting different copies.

No, not without additional development.

Metadata embedded in content files to be extracted on import into the system, for ease of working with and adding in the resulting record.

No, not without additional development.

Ability to batch edit metadata records associated with content files (i.e. search and replace on particular data elements; replace data across some subset of objects in a given metadata element)

No, not without additional development.

Integrated use of authoritative vocabulary terms from LC, Getty, VRA, TGN, locally created, etc.

No, not without additional development.

Ability to specify different levels of access restrictions in the metadata, for example: 1) reading room/department only, 2) library only, 3) specific UC campus only, 4) all UC campus users, 5) specific authorized users only, and 6) public access.

Provides user/group based access-control; presumably this could be used to support place-based access control.

Ability to limit what is shown to end users at different access levels (e.g. only the metadata record available to the public but a given digital object and the metadata record are should be available to campus users)

Differentiates between “discover (see the metadata but not download)” and “read (see the metadata and download)”. Could be extended to support more fine-grained intra-object access control.

A set of APIs to interact with the repositories where the objects are being managed.

Yes.

Appendix D: Islandora Comparison

- **Security access control**

Security in Islandora is the result of combining Drupal's Access Control infrastructure (Drupal Roles and Permissions) with Fedora's security framework. Fedora's framework offers a great deal of flexibility and customization.

In a Drupal site, you can allow (or prevent) people from doing things like creating accounts, or viewing your site by navigating to administer > user management > user settings. Drupal also gives you the ability to divide your site users into different groups, by creating "Roles" for users. A "Role" defines who your user is, and what they should be able to access, update, delete, or create in a Drupal site.

When you are using Islandora, Fedora's entire suite of security features are available to you. Fedora security starts with your repository setup, but can be refined further using object-specific XACML policies (written in eXtensible Access Control Markup Language).

- **Complex objects support (arbitrary div hierarchy of mixed type)**

The Book Solution pack in Islandora may be considered complex object support. The Book Solution pack creates a Book Collection object consisting of a MODS metadata record. After the Book Collection object is created, users can upload a zipped directory of uncompressed tiffs. These tiffs become Page objects that are members of the Book Collection Object. Page objects undergo an Optical Character Recognition (OCR) process, making their contents full-text searchable. You can use the Book module to create any paged content consisting of tiff images of pages.

We did not see the mixed type complex object support. Perhaps we can modify the existing module or create the new module to support for mixed type.

- **Audio/Video support**

Islandora has an Audio Solution Pack and Video Solution Pack.

The Audio Solution Pack takes a WAV file, and ingests it into the Fedora repository. On ingest, the WAV file is converted to an MP3 using LAME. The MP3 can be downloaded by users, and when viewing an item that has utilized the Audio Solution Pack, the audio file can be played in JWPlayer.

The Video Solution Pack takes a video file, and ingests it into your Fedora repository. On ingest, the video file is converted to an MKV (archival format) using FFMPEG. Two derivative formats are also created: MP4 (using FFMPEG) and OGG (using FFMPEG2Theora). JWPlayer is used to display one of the derivative formats (depending on the browser being used).

- **Workflow support (suppress in process collections, restrict master/print quality files)**

Islandora uses Drupal Permissions and Roles (Users are assigned roles and can have multiple roles), with the user possessing all the permissions that their various roles do. If you allow a role to **add fedora datastreams,** users with that role will be able to add a datastream to an object in your repository.

- **Metadata editing (including support for custom fields and batch editing)**

Islandora utilizes Fedora's ability to represent ddescriptive metadata in XML format via one or more **Datastreams** in an object. Fedora is written in such a way that any object may have multiple metadata Datastreams, which can store metadata following any schema, such as MODS, Dublin Core, or QDC.

Fedora requires that any object created in the system contain a default Dublin Core Stream. By extension, any object that is created in Islandora (and therefore in a Fedora repository) will have a default Dublin Core Datastream. However, Islandora's Solution Packs presume that users will often want to store an additional metadata stream outside of the default Dublin Core stream, in order to create richer descriptive metadata, and also to adhere to standards for metadata description of particular types of data and collections. For example, the MODS form that comes by default with any Solution Pack is designed to suit the most common cases for that solution pack.

Custom metadata editing forms can be built by using XML Forms Modules in Islandora.

XML Forms is a collection of Drupal modules that allow for the manipulation of XML documents though Drupal forms. The Islandora Form Builder (XML_Forms modules) makes it possible for users to create, copy, and edit ingest forms, and to affiliate them with Content Models in the repository.

- **Modular solution**

Islandora's front end is based on Drupal, which has many modules BUT Islandora does not use Drupal nodes, so most Drupal modules will not work without a lot of programming. Back end is Fedora and treated like a module. Solution Packs offered as add-ons for different content types.

- **Best of breed component w/ open source tendencies**

Islandora is open source and leverages other open source products.

- **Look for broad adoption w/ community support**

Gaining popularity across many libraries as a single source solution. Supported by community; centered around the company Discovery Garden.

- **Ability to upload preservation/master copies of content files and have derivative/service copies automatically generated when possible instead of submitting different copies**

Islandora has image, pdf and book Solution Packs, which can ingest JPEG, JPG, GIF, and PNG, uncompressed TIFF, PDFs into your Fedora repository, and uses ImageMagick to create thumbnail, medium, and large sized versions of the image.

The Audio Solution Pack takes a .wav file, and ingests it into your Fedora repository. On ingest, the .wav file is converted to an .mp3 using LAME.

The video solution pack takes a video file, and ingests it into your Fedora repository. On ingest, the video file is converted to an MKV (archival format) using FFMPEG. Two derivative formats are also created: MP4 (using FFMPEG) and OGG (using FFMPEG2Theora).

- **Metadata embedded in content files to be extracted on import into the system, for ease of working with and adding in the resulting record.**

Some Islandora Solution Packs utilize the EXIFtool to extract technical metadata from objects and store it in a separate datastream. EXIFtool is probably the most comprehensive, open source metadata extraction tool, which is a perl utility that can extract technical metadata from images, audio, video and more.

- **Ability to batch edit metadata records associated with content files (i.e. search and replace on particular data elements; replace data across some subset of objects in a given metadata element)**

We did not see batch editing support but we are able to edit metadata using the forms modules. Perhaps we can modify the existing module to do batch editing support.

- **Integrated use of authoritative vocabulary terms from LC, Getty, VRA, TGN, locally created, etc.**

Being worked on as of August:

https://groups.google.com/forum/?fromgroups=#!topic/islandora/PsNZ6H4VDl0

- **Ability to specify different levels of access restrictions in the metadata, for example: 1) reading room/department only, 2) library only, 3) specific UC campus only, 4) all UC campus users, 5) specific authorized users only, and 6) public access**.

Security in Islandora is the result of combining Drupal's Access Control infrastructure (Drupal Roles and Permissions) with Fedora's security framework. Fedora's framework offers a great deal of flexibility and customization.

- **Ability to limit what is shown to end users at different access levels (e.g. only the metadata record available to the public but a given digital object and the metadata record are should be available to campus users).**

Islandora Using Drupal Permissions and Roles combining with Fedora Security:

--Object-specific XACML policies
--Collection-specific XACML policies
--Global XACML Policies

The documentation states that youcan customize security policies that will restrict access to the items in your collection (overriding Fedora's default behavior).

- **A set of APIs to interact with the repositories where the objects are being managed**

The repository is Fedora, which has a rich set of documented APIs. Islandora is a front end to many of these APIs.

Islandora allows you to create, view, update, and delete (purge) content in your repository. By enabling the Collection Manager modules, you can:

--Add an Item to the Digital Collection
--Edit an Object’s Metadata
--Purge an Object
--Replace the Datastream of an Object
--Create a New Islandora Collection
--Batch Ingest Files
--Harvest Metadata Records and Advanced Collection Management:
--create new child collections
--Manage Collection Policies
--Change Content Models
--Purge All Objects From a Collection

Other criteria

- **Search and Discovery**

Islandora uses Solr and the Solr module in combination with GSearch to provide search functions to users on your site. Islandora uses Solr to make objects in your Islandora installation discoverable. The Solr search module uses an XSLT in Gsearch to index the **FOXML** documents in your repository, and allows you to configure search fields for searching and faceting. Whenever you add a new object in Fedora, the Solr module updates your index, and makes those results available to your users.

Appendix E: Nuxeo Evaluation

- **Security access control**

Nuxeo has a package for shibboleth authentication. The security model is so fine grained that it is possible to set security policies at the level of individual metadata fields. see section 11.6

- **Support for complex objects (arbitrary div hierarchy of mixed type)**

Object model supports arbitrary hierarchy of mixed types. see see also

- **Audio/Video support**

DAMS module had built in transcoding and key frame extraction support.

- **Workflow support (suppress in process collections, restrict master/print quality files)**

As of 5.x built in rules engine

- **Metadata editing (including support for custom fields and batch editing)**

DAMS module out of the box supports batch editing

- **Modular solution**

modular architecture (OSGi) see architecture document

- **Best of breed component w/ open source tendencies**

product is open source; leverages other open source products

- **Look for broad adoption w/ community support**

not widely used in the library community. There seems to be a robust user community. http://answers.nuxeo.com/

- **Ability to upload preservation/master copies of content files and have derivative/service copies automatically generated when possible instead of submitting different copies**

It is possible to configure workflows like this (verify, but I think w/o programming just by using studio?).

- **Metadata embedded in content files to be extracted on import into the system, for ease of working with and adding in the resulting record.**

Yes, it has a built in extractor that can be swapped out. Technical details can be found here.

- **Ability to batch edit metadata records associated with content files (i.e. search and replace on particular data elements; replace data across some subset of objects in a given metadata element)**

The DAMS module appears to be able to do this from its web interface

- **Integrated use of authoritative vocabulary terms from LC, Getty, VRA, TGN, locally created, etc.**

Has a vocabulary management feature that can import complex vocabularies.

- Ability to specify different levels of access restrictions in the metadata, for example: 1) reading room/department only, 2) library only, 3) specific UC campus only, 4) all UC campus users, 5) specific authorized users only, and 6) public access.

The security model can represent these types of access.

- **Ability to limit what is shown to end users at different access levels (e.g. only the metadata record available to the public but a given digital object and the metadata record are should be available to campus users)**

Security model can be extended into search, supports

- **A set of APIs to interact with the repositories where the objects are being managed**
- **Supports CMIS protocol.**

Appendix F: High Level Proposal for DAMS Project

Assumptions

- Requirements for End User Access and Display as well as DAMs are substantially covered by the Final Report of LT1A. (http://bit.ly/14ChKD1)
- The selection of content included in the Final Report of LT3A will be considered as the initial scope of the project. (http://bit.ly/YB13oe)
- Requirements for the crawler and harvesting components are substantially covered by the Final Report of LT3C. (http://bit.ly/XX4lkA)
- Based on the recommendations of the LT1C report, CDL will host the DAMS and build the End User Access and Display layer.
- Upon approval, a campus implementation group will be formed with representation from all key stakeholder groups.

Example Timeline

Summary: Assuming a May 2013 start date, initial system functions are available to collection managers in 3 to 6 months. Additional components follow approximately every 3 to 6 months later, with completion of full functionality to public approximately 24 months from start date. Training and documentation will be ongoing to support continuous rollout approach.

Date	Functionality	System Component	Audience	Audience can now…
May 2013	Metadata extraction negotiation	Nuxeo DAMS	Project team	Begin object model development.
Oct 2013	Object models for simple cases	Nuxeo DAMS	Collection managers	Log in to the Collection Registry dashboard and see harvested content.
	Collection Registry dashboard	Collection Registry		
	Collection harvester launched	Collection Registry		
Jan 2014	Object models for complex cases	Nuxeo DAMS	Collection managers	Participate in the development of detailed requirements for the user interface.
	User assessment for the User Interface	Public Interface	Campus implementation group	
July/Aug 2014	Ability to add new content, check metadata and edit	Nuxeo DAMS	Collection managers	Log into the DAMS and see existing content and new content.
	User Interface Design	Public Interface	Campus implementation group	
	Media Server installed—streaming media supported	Nuxeo DAMS	Campus technologists	Create custom user interfaces (UIs) if desired.
	Search API available	Collection Registry		
Jan 2015	Existing “target” collections loaded	Nuxeo DAMS	Collection managers	Assure the quality of their collections.
	Integration with Merritt.	Preservation		Understand that collections are now preserved.
	Index improvement	Collection Registry		Enjoy continuously better search results.
May 2015	User Interface available	Public Interface	Public	View the new UC Digital Library Collection

Human Resources & Roles

Role	Proposed Resource	Percent availability	Cost basis	Total Cost*
Project/Product Manager	CDL Staff	50%	$89,600/yr	$89,600
Sr. Programmer, Tech Lead	CDL Staff	100%	$110,500/yr	$221,000
Jr. Programmer	CDL Staff	100%	$82,600/yr	$165,200
UX Designer, Access	CDL Staff	50% for 1 year	$89,600/yr	$44,800.
Metadata design analyst	CDL Staff/contractor/campus	100 %	$86,800/yr	$173,600.
ETL (extract, transform, load) consultant	Contractor/campus	50% for 1 year	$105,000/yr	$52,500
DAMS Implementation consultant	Contractor/campus	2 months	$120,000/yr	$20,000
DAMS trainer	Contractor/campus	25% for 18 months	$89,600/yr	$33,600
Technical writer	Contractor/campus	20%	$86,800/yr	$17,360
Collection entry	Campus	Varies	$86,800/yr	Varies
Quality assurance	Campus	Varies	$86,800/yr	Varies

*All totals are for 2 years unless otherwise stated.

Note: CDL has a one-time allocation of $125,000 from the budget augmentation for 2012-2013 earmarked for this project which could be used for expenses such as contractors or other short term startup costs.

Software acquisition and support

Package	Purchase Y/N	If purchase, price	Purchase timeframe*
Nuxeo (DAMS)	Y	$12,000 (annual)	FY 2012-13
Wowza (Media server)	Y	$1145 (annual)	FY 2013-14
Merritt storage	Y	See below	FY 2014-15
Amazon Web Services	Y	$600 (annual)	FY 2013-14

*Timeframes based on Example Schedule

Hardware acquisition

Requirement	Location	Estimated price	Purchase timeframe*
Network Attached Storage	Kaiser Data Center	$2106 (annual)	FY 2013-14

*Timeframes based on Example Schedule

Reference information

Merritt Storage Costs: UC-only		
Quantity	**Annual**	**One-Time**
Up to 1 TB	$ 390.00	$ 2,900.00
Up to 2 TB	$ 780.00	$ 5,800.00
Up to 5 TB	$ 1,950.00	$ 14,500.00
Up to 10 TB	$ 3,900.00	$ 29,000.00
Up to 20 TB	$ 7,800.00	$ 58,000.00
Up to 50 TB	$ 19,500.00	$ 145,000.00

Initial Collections for DAMS Ingest*		
Collection	**Format**	**Size**
UC Davis		
1. Halberstadt Collection	images	TBD
2. 1937 Yolo County Aerial Photographs	images	48 GB
UC Davis Total (approx.)		> **50 GB**
UC Irvine		
3. Edward Cochems photographs	images	15 GB
4. Hugh McMillan photographs	images	21 GB
5. W. Gearhardt photographs	images	52 GB
6. Early campus photograph albums	images	456 MB
7. Eugene Loring films	video	16 GB
8. Richard Rorty born digital files	born dig	740 MB
UC Irvine Total (approx.)		**106 GB**
UC Merced		
9. WWII Japanese American Assembly Center	text	2 GB
10. Mugbook, Merced County	images	57 GB
11. McLean collection	images	58 GB
12. Angels Camp Museum	images	1 GB
UC Merced Total (approx.)		**118 GB**

Initial Collections for DAMS Ingest*		
Collection	**Format**	**Size**
UC Riverside		
13. Oral History Interviews	video	350 GB
14. Sabino Osuna papers	images	8 GB
15. Highlander student newspaper	text	3 GB
16. George Fujimoto diaries	text	12 GB
17. Tuskegee Airmen Archive	images	TBD
18. University Archives photographs	images	116 GB
UC Riverside Total (approx.)		**> 489 GB**
UC Santa Barbara		
19. EDVR Images	images	3.9 TB
20. Air Photos	images	1 TB
UC Santa Barbara Total (approx.)		**4.9 TB**
20 Initial Collections Total (approx.)		**>5.65 TB**

*Source of expected initial collection size:

Collection registry: http://dscl-dev.cdlib.org/collection_registry/