UCLDC Model and Major Components

Skip to topic:

- Overall Technical Model
- Digital Asset Management System (DAMS)
- Central Index / API
- Collection Registry
- Public Interface

Overall Technical Model

The technical model for the UC Libraries Digital Collection (UCLDC) was developed by POT1 Lightning Team 1C. This team articulated three major principles driving the design of the system and the technologies selected:

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

The lightning team report provides extensive documentation on the process of developing the model, as well as a comparison of other systems considered.

The diagram at right was used by the lightning team to visualize the technical model, and it continues to provide an excellent overall picture of how everything is going to work together in the UCLDC. Below, we explore the four major components of the product we are working on over this two-year implementation phase, highlighting parts of the model so it is clear where they fit in.

Digital Asset Management System (DAMS)
A digital asset management system (DAMS) is software for storing, describing, and managing digital content such as images, audio, and video files. The UCLDC DAMS will enable campus libraries to add metadata to digital assets, publish them in the access interface, designate them for preservation, and perform many other stewardship functions. The DAMS will offer the major benefit of allowing for “collections-in-process,” meaning librarians will be able to work on assets and metadata as they have the time, in a closed virtual environment, until they are ready to open the content for public viewing.

The DAMS we are configuring and deploying is a commercially supported, open-source product called Nuxeo.

The UCLDC DAMS will also have a mechanism for selecting digital assets for preservation and pushing them into Merritt for long-term preservation.

**Central Index / API**

The central index (Solr) is where all of the digital assets and metadata—those hosted in the central DAMS and those harvested from other platforms—will come together. The item-level index will allow for queries and faceting on predetermined fields, and therefore make it possible for all UCLDC collections to be searched and browsed in a single public interface.

The central index will also have an API, which will make it possible for campus libraries and potentially other partners to develop their own interfaces to the UCLDC and portions therein.

**Collection Registry**
The Collection Registry is an interface for campus library staff to manage and apply standardized information about repositories and digital collections. Collections may be defined in a few different ways. They may reside in the shared DAMS that is part of the UCLDC or they may be harvested from external sources (such as a campus OAI feed, the Internet Archive, or OAC/Calisphere) and brought into the UCLDC's central index.

The collection registry is an essential part of the UCLDC system with a few different use cases. Specifically, it will:

- Drive some essential display requirements for the UCLDC public interface. For example, public interface end-users will be able to facet search results by campus.
- Enable campus libraries to denote and track the status of digital collections, for example indicating the location of collections ready for harvest.
- Provide a platform for collaborative digital collection development and/or digitization planning

**Public Interface**

A public interface will be built on top of the system which will enable end-users to seamlessly search and browse across the whole UC Libraries Digital Collection—including content both hosted in the DAMS and harvested from other platforms. The public interface will make use of the Solr API described above. The public interface is intentionally “de-coupled” from the repository and other back-end systems in this model. This provides more flexibility, making it easier to change and enhance the public interface as the collection grows and user expectations evolve.