

Designing a Migration Path - Institutional Profiles

- [Front-end Application Analysis](#)
 - [Islandora](#)
 - [Samvera](#)
 - [Custom](#)
- [Data Model Analysis](#)

Front-end Application Analysis

We profiled eight institutions using different front-end applications (Islandora, Samvera, custom) supporting different use cases (institutional repository, special collections, research data, etc.), of different institutional and repository size. These profiles were used to establish commonalities and differences between Fedora 3.x repositories in order to assess the impact to migrating to the latest version of Fedora. The results are summarized below, broken down by front-end application category. The eight institutions are:

- Florida State University
- National Library of Medicine
- University of Wisconsin-Madison
- UNC Chapel Hill
- Michigan State University
- Stanford University
- Williams College
- Amherst College

Islandora

Islandora is a Drupal-based framework that interacts with Fedora 3.x largely through the REST-API. Islandora takes a modular approach, with a single core application that can be customized by enabling and configuring Drupal modules. This approach makes migrations easier because the majority of the community is using a common core application with similar data models. Islandora uses a [Solution Pack](#) framework; each Solution Pack handles a particular type of content and describes specific data models which are shared across the community. The profiled institutions use a common and relatively small set of file formats, most of which are already supported in Islandora 8. Migrations will also be made easier by using the [Drupal Migrate](#) ecosystem, which allows administrators to migrate from an Islandora 7 application to an Islandora 8 application based on Fedora 5.x. Such a migration is necessary because Islandora 7 only supports Fedora 3.x, while Islandora 8 only supports Fedora 5.x and higher.

Islandora 8 continues to follow the same model of providing a core application with configurable modules, which mitigates the need for adopters to rebuild all of their Islandora interfaces and workflow tools - many of these will be part of the standard Islandora 8 application stack. However, some institutions have customized their interfaces, and these customizations will need to be rearchitected in Islandora 8, which may be challenging for institutions with limited resources. Islandora 7 has also been around for many years and it has accumulated many modules and features which have not yet been built into Islandora 8. This will present a barrier to anyone using Islandora 7 features that don't exist in Islandora 8 yet, but fortunately as community members build these features they will be available for everyone to use.

Each of the profiled institutions with Islandora repositories have made some front-end customizations to their repositories, and these customizations are largely unavailable to the public and may or may not be documented. This presents a challenge for migrations since these local customizations are unlikely to be developed as part of the core Islandora 8 offering, thus requiring some resource investment on the part of the affected institutions in terms of updating the code to work with Islandora 8.

Samvera

The Samvera (formerly Hydra) community takes a different approach compared to Islandora, with an ecosystem of many different applications and tools used by different institutions. There is no single Samvera application stack, but most applications and tools are written in the Ruby programming language and Rails web framework. The most commonly used front-end application (Hyrax) has already been made compatible with Fedora 4.x and 5.x; in fact, the Samvera community was an early adopter of Fedora 4.x, and a previous version of Hyrax (Sufia) was the first application based on Fedora 4.x to be used in production. This has allowed some institutions to migrate to the new platform, though the diversity of community deployments and prevalence of local customizations makes migrations more challenging.

Stanford is an example of a founding Samvera community member that uses a diversity of applications and tools in a complex repository ecosystem. Many of these tools are customized to work within the local environment in particular ways, which makes a repository migration a daunting prospect. The level of customization and interdependency means that Stanford would likely need to undertake a migration of their application framework in-house using local resources; no one else in the community has a similar enough system to work together on a migration.

Custom

Custom Fedora 3.x implementations are the most difficult to profile because each one is different. The institutions we profiled tended to support similar services (ingest, search, administration, dissemination, etc.) but they provide these services in different ways using different applications and programming languages. In some cases these frameworks are available on GitHub, but most often they are managed in-house. Even when the source code is publicly available, it is often not well-documented and not designed for use outside the home institution.

Custom frameworks present a great challenge for migrations because they necessarily require local effort that can't be supported by others in the community. One of the profiled institutions has nearly completed a migration to the latest version of Fedora but it took a lot of effort over a period of several years. Local resources are therefore a potential barrier to migration in the case of repositories with custom front-ends, but as the data model analysis section below describes, shared community tools may be able to support the migration of the data independent of the front-end applications.

Data Model Analysis

Each of the eight profiled institutions provided representative sample data which was used to compare and contrast the data models used by each institution. The models themselves can be found in the Data Models section; what follows is a summary of the findings.

There are many commonalities between the types of data managed by each institution. For example, all eight institutions manage image, document (PDF), video, and paged content (books, serials) items. These objects tend to be modeled similarly across institutions, with the highest degree of consistency between the Islandora repositories. However, there is also some variance as we see with the University of Madison-Wisconsin; each object is contained by a parent, metadata-only object, which adds an extra layer of hierarchy that does not exist in the other profiled repositories. Digging into the details further reveals a number of differences between institutional data models.

Each Fedora 3.x object has a number of datastreams, each of which has an identifier. These IDs are configurable, and we see a high degree of variance between the datastream IDs used at each institution. In some cases datastream IDs are employed inconsistently between data models within the same institution. This presents a possible barrier to migration because shared tools will need to support custom datastream IDs. That being said, it would be possible to simply migrate each datastream as a binary file in Fedora 4.x or higher, thereby maintaining the basic structure of each object.

We also see differences in the focus of each repository. For example, UNC Chapel Hill and Stanford each make heavy use of PREMIS preservation metadata, while the other institutions do not include PREMIS metadata in their data models. Similarly, the National Library of Medicine and the University of Madison-Wisconsin maintain METS structural metadata, while the others do not. These differences are less important from a simple binary migration scenario as described in the previous paragraph, but they could present challenges in cases where, for example, PREMIS metadata is transformed from XML to RDF during a migration.

Data transformations are challenging because it is not always clear how the data should be transformed. Many of the profiled institutions use MODS descriptive metadata in Fedora 3.x, and it would be fairly simple to migrate these datastreams and store them as XML binary files in Fedora 4.x and higher. This is likely to be a sufficient migration for some institutions; however, others will want to take the opportunity to transform XML datastreams into RDF properties so they can take advantage of Fedora's LDP-based API. But the transformation from MODS XML to MODS RDF is not a simple one - a Samvera working group only recently published [recommendations](#) on how to handle the mappings, and this was a multi-year, multi-group effort. Even in this case, not all institutions will agree on the mappings, so transformation tools need to be robust and configurable.