



**Fedora**<sup>TM</sup>

# Migrating from Fedora 3 to Fedora 4

Andrew Woods, DuraSpace



# Learning Outcomes

- Understand the main differences between Fedora 3 and Fedora 4
- Understand the considerations necessary for migrating from Fedora 3 to Fedora 4
- Explore new possibilities for enhancing data in Fedora 4

# Differences between Fedora 3 and Fedora 4

# XML Objects vs. Resources

- Fedora 3
  - FOXML objects
  - Inline XML and XML datastreams
- Fedora 4
  - Web resources (Objects & datastreams)
  - RDF properties and XML datastreams

# Flat vs. Hierarchy

- Fedora 3
  - Objects and datastreams at the top level
  - No inherent tree structure
- Fedora 4
  - Objects and datastreams in a hierarchy
  - All resources descend from a root node

# File System

- Fedora 3
  - Objects directory and Datastreams directory
  - Both objects and datastreams are in a PairTree
- Fedora 4
  - Objects directory and Datastreams directory
  - Datastreams in a PairTree
  - Objects in a database (e.g. LevelDB)

# PID vs. Path

- Fedora 3
  - Objects have Persistent Identifiers (PIDs)
  - An object's PID can never be altered
- Fedora 4
  - Objects have an internal ModeShape UUID
  - Objects have a repository path
    - This can be user-defined or generated via a PID-minter

# Migration Considerations



# Ingest or Projection?

- Fedora 4 supports projection over content in an external system.
- Projecting over Fedora 3 REST-API is possible.

# Security

- What kind of security does your Fedora 3 repository use?
- Many Fedora 3 repositories use XACML security.
  - Fedora 3 XACML policies would need to be transformed for Fedora 4 (which supports XACML 2.0).

# Versions

- Does your Fedora 3 repository use versioning?
- Fedora 3 versions must be iterated through to create new versions in Fedora 4.
- How should version dates be handled? Will you use the system modified date, or a special date property?

# Content Models

- How are content models used in your Fedora 3 repository?
- Do they have any logic built into them, or is that handled at a higher application level (e. g. Islandora, Hydra)?

# Disseminators

- Does your Fedora 3 repository make use of disseminators?
- What are they used for? XSL transforms? Derivatives?
- How can we support the existing disseminator use cases in Fedora 4 without re-creating disseminators?

# Other Considerations

- Have we left anything out?
- What else needs to be considered when planning a migration?

# Enhancements

# Taking Advantage of Properties

- Lightweight and granular compared to XML.
- Inline XML is no longer supported.
- Converting Inline XML and/or XML Datastreams (e.g. RELS-EXT, RELS-INT) to RDF properties.



# New Query Possibilities

- New possibilities for complex queries that extend beyond the limits of the repository.
  - Linked data relationships can be exposed via a standardized REST-API
  - Web applications can take advantage of these standardized representations.
  - Data can be shared and manipulated in new and interesting ways.

# Enhancing Your Metadata

- XML metadata datastreams are still supported, but there are new opportunities to explore!
- XML metadata can be converted into RDF metadata using an RDF-based schema.
- RDF metadata is easier to query and share.
- Take advantage of linked data by pointing to authority URIs.