Fedora™

Introduction and Feature Tour

David Wilcox, DuraSpace

DURASPACE™

Learning Outcomes

- Understand the purpose of a Fedora repository
- Learn what Fedora can do for you
- Understand the key capabilities of the software

Introduction to Fedora 4

What is a Fedora Repository?

- Secure software that stores, preserves, and provides access to digital materials
- Supports complex semantic relationships between objects both within and outside the repository
- Supports millions of objects, both large and small
- Capable of interoperating with other applications and services

Exposing and Connecting Content

- Flexible, extensible object modeling
- Atomic objects with semantic connections using standard ontologies
- RDF-based metadata using Linked Data
- RESTful API with native RDF response format

Core Components

Fixity

- Over time, digital objects can become corrupt
- Fixity checks help preserve digital objects by verifying their integrity
- On ingest, Fedora can verify a user-provided checksum against the calculated value
- A checksum can be recalculated and compared at any time via a REST-API request

Backup and Restore

- A full backup can be performed at any time
- A full restore from a repository backup can be performed at any time

Export and Import

- A specific Fedora object, its children objects, and associated Datastreams can be exported
- Exported objects are serialized in a standard JCR/XML format
- An exported object or hierarchy of objects can be imported at any time

Versioning

- Versions can be created across the entire repository or on particular API calls.
- A previous version can be restored via the REST-API.

Data Modeling

Resources

- Both objects and datastreams are represented as resources.
- Object resources can have both Objects and Datastreams as children.
- The tree structure allows for inheritance of things like security policies.

Properties

- Resources have a number of properties, which are expressed as RDF triples.
  - Name-value pairs; translated to RDF on REST-API responses
- Properties can be RDF literals or URIs
- Any number of RDF namespaces can be defined and used.

Content Models

- Content can be modeled using Compact Node Definitions (CNDs).
- Mixins can be used to define any number of properties.
- An object can inherit properties from any number of mixins

Linked Data

- Fedora 4.0 is compliant with the LDP 1.0 spec.
- Metadata can be represented as RDF triples that point to objects outside the repository.
- Many possibilities for exposing, importing, sharing resources with other web applications.

External Components

Indexing

- Index repository content for external applications with the JMS Message Consumer.
- The Consumer relays repository updates to one or more external applications.
- Repository content needs to be assigned the rdf:type property "indexible".

Triplestore and Search

- An external triplestore can be used to index the RDF triples of content managed by Fedora.
- Any triplestore that supports SPARQL-update can be used; Fuseki and Sesame have been tested.
- An external search application can also be configured
    - Solr and Elastic Search have been tested

Authorization

- The authorization framework provides a plug-in point within the repository that calls out to an optional authorization enforcement module.
- Currently, two authorization implementations exist: Role-based and XACML

Role-based Authorization

- Role-based authorization compares the user's role(s) with an Access Control List (ACL) defined on a Fedora resource.
- ACLs can be inherited; if a given resource does not have an associated ACL, Fedora will examine parent resources until it finds one.

XACML Authorization

- A default policy must be defined for the repository, and each resource can override the default with another policy.
- A XACML policy referenced by a resource will also apply to all the resource's children, unless they define their own XACML policies that override the parent policy.

Performance

Transactions

- Multiple actions can be bundled together into a single repository event (transaction).
- Transactions offer performance benefits by cutting down on the number of times data is written to the repository filesystem (which tends to be the slowest action).

Clustering

- Two or more Fedora instances can be configured to work together in a cluster.
- Fedora 4 currently supports clustering for high-availability use cases.
- A load balancer can be setup in front of two or more Fedora instances to evenly distribute read requests across each instance.

Further Reading

- Fedora 4 Wiki
  - https://wiki.duraspace.org/display/FF/Fedora+Four+Prospectus
- Fedora 4.0 Features
  - https://wiki.duraspace.org/display/FF/Fedora+4.0+Feature+Set