DESY Computing Seminar on
Data Management in Climate Research

Ing. José A. Mejía Villar M.Sc.
jmejia@awi.de

Computing Center of the Alfred Wegener Institute for Polar and Marine Research

Hamburg, 31. January 2011

Contents

1. Fedora Commons

1.1. What is Fedora Commons?

1.2. Key Features

1.3. Digital Object Model

1.4. Content Model Architecture

1.5. Web Service Interfaces

1.6. Framework Services

1.1 What is Fedora Commons?

- Fedora stands for *Flexible Extensible Digital Object Repository*.
- Fedora is a general-purpose, open-source digital object repository system.
- Java based conceptual framework using a set of abstractions about digital information to provide the basis for software systems that can manage digital information.
- The Fedora software distributed by Duraspace (http://www.duraspace.org) is available from http://fedora-commons.org under the terms of the Apache License, version 2.0.

1.2 Key Features [1/2]

- Store all types of content and its metadata
- Scale to millions of objects
- Access to data via Web APIs (REST/SOAP)
- Provides RDF based Resource Index search
- Rebuilder Utility (for disaster recovery and data migration)
- The entire repository can be rebuilt from the digital object and content files.

1.2 Key Features [2/2]

- Content Model Architecture (define "types" of objects by their content)
- Many storage options (database and file systems)
- JMS messaging provider (your apps can "listen" to repository events)
- OAI-PMH Provider Service

1.3 Digital Object Model

- All content in Fedora is managed as data objects
- Data objects are made up of datastreams that store the content or metadata about it.
- Each datastream can be managed directly by the repository or left in an external, web-accessible location to be delivered through the repository as needed.
- A data object can consist of any number of data and metadata components, combining managed and external datastreams in any desired pattern.

1.3 Digital Object Model: FOXML

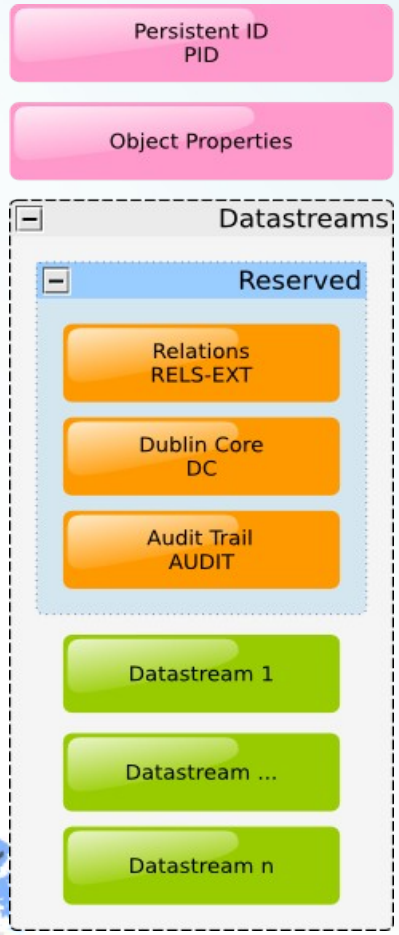
FOXML (Fedora Object XML) is a simple XML format that directly expresses the Fedora Digital Object Model.

FOXML 1.1 XSD Schema on: http://fedora-commons.org/definitions/1/0/foxml1-1.xsd

```
<digitalObject PID="uniqueID">

  <!-- there are a set of core object properties -->
  <objectProperties>
    <property/>
    <property/>
    ...
  </objectProperties>

  <!-- there can be zero or more datastreams -->
  <datastream>
    <datastreamVersion/>
    <datastreamVersion/>
    ...
  </datastream>

</digitalObject>
```

1.3 Digital Object Model: Datastreams

Fedora reserves three datastreams for its use, namely “DC” (Dublin Core), “AUDIT”, and RELS-EXT.



Basic Datastream Properties

- Datastream Identifier
- State: Active, Inactive, or Deleted
- Created Date
- Modified Date
- Versionable: true/false
- Label
- MIME Type
- Format identifier (optional)
- Alternate Identifiers (Handlers or DOI)
- Checksum
- Bytestream Content
- Control Group
  - Internal XML Content
  - Managed Content
  - Externally Referenced Content
  - Redirect Referenced Content

1.4 Content Model Architecture

- The Content Model Architecture (CMA) describes an integrated structure for persisting and delivering the essential characteristics of digital objects in Fedora.
  - Structural, behavioral, and semantic information.
  - Description of the permitted, excluded, and required relationships to other digital objects or identifiable entities.

- The content model is expressed in a modeling language.

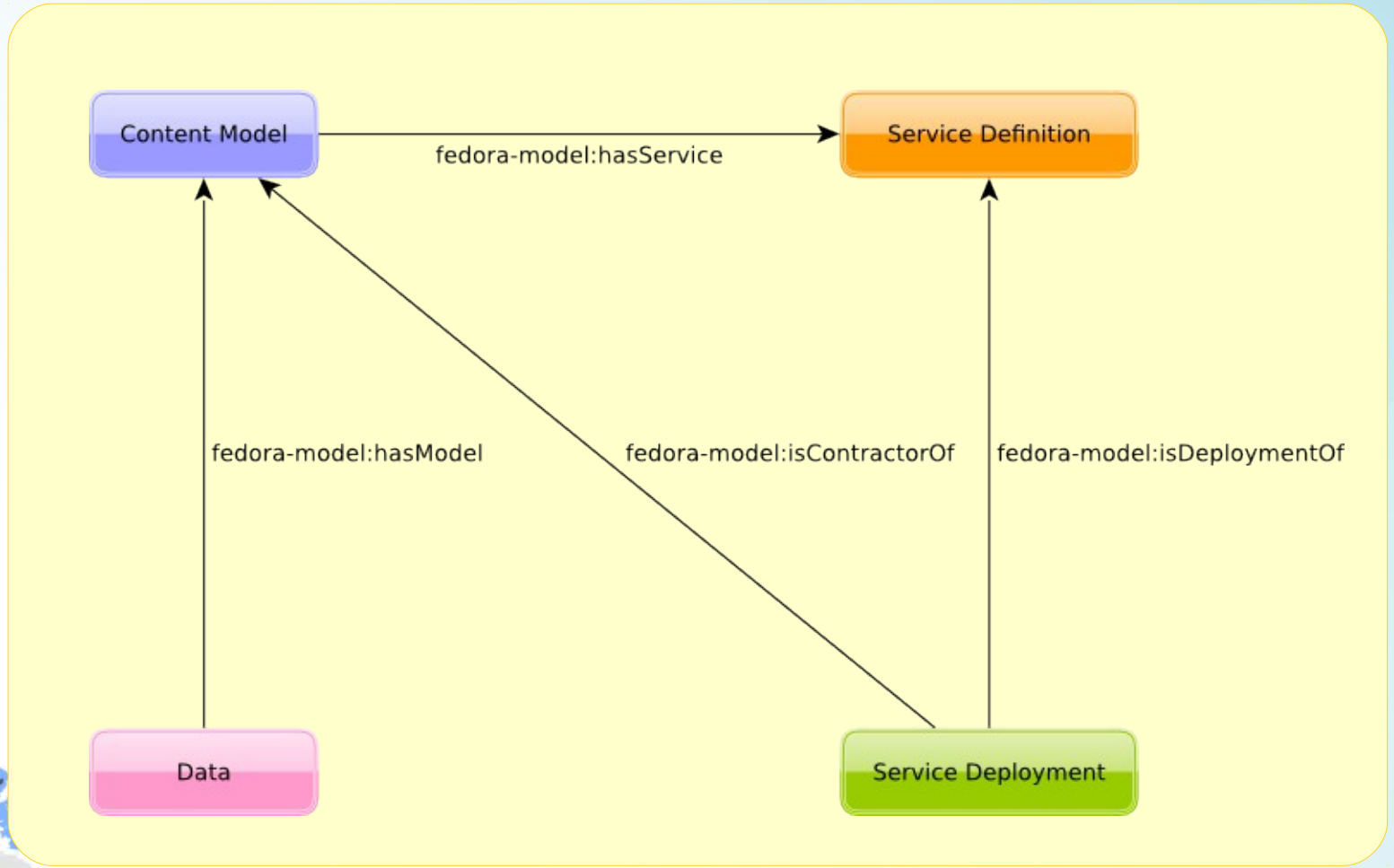
1.4 CMA: Object Types

Fundamental Fedora Object Types

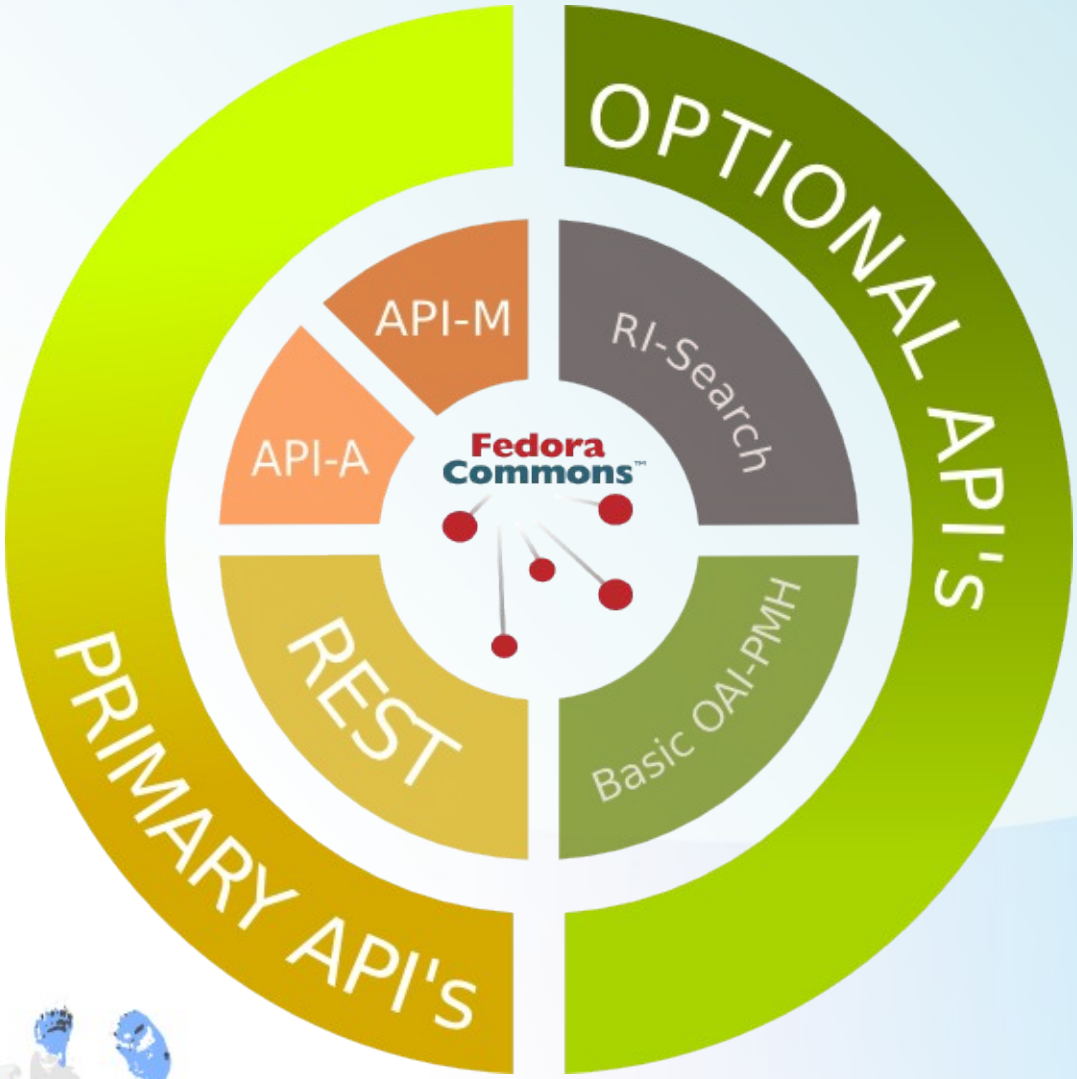
Object Type	Code	Description
Data	Data	A container for content
Service Definition	SDef	A container for the service definitions
Service Deployment	SDep	A container for service deployment bindings
Content Model	CModel	A container for content models

1.4 CMA: Object Types

Fundamental CMA Relationships



1.5 Web Service Interface



- Primary API's

  Allow the creation, reading, modification, and deletion of Fedora digital objects.

- Optional API's
  - Basic OAI-PMH
  - RI-Search

1.5 Web Service Interface: API-A

Fedora Access service methods

- Repository Access
  - DescribeRepository
- Object Access
  - findObjects
  - resumeFindObjects
  - getObjectHistory
  - getObjectProfile
- Datastream Access
  - getDatastreamDissemination
  - listDatastreams
- Dissemination Access
  - getDissemination
  - listMethods

1.5 Web Service Interface: API-M

Fedora Management service methods

- Datastream Management
  - addDatastream
  - compareDatastreamChecksum
  - getDatastream
  - getDatastreamHistory
  - getDatastreams
  - modifyDatastreamByReference
  - modifyDatastreamByValue
  - setDatastreamState
  - setDatastreamVersionable
  - purgeDatastream
- Relationship Management
  - addRelationship
  - getRelationships
  - purgeRelationship

- Object Management
  - modifyObject
  - purgeObject
  - export
  - getNextPID
  - getObjectXML
  - ingest
  - validate

1.5 Web Service Interface: RISearch [1/2]

- The Resource Index Search Service (RISearch) is a web service that exposes the contents of a repository's Resource Index guide for outside use.

RISearch Service Functionality

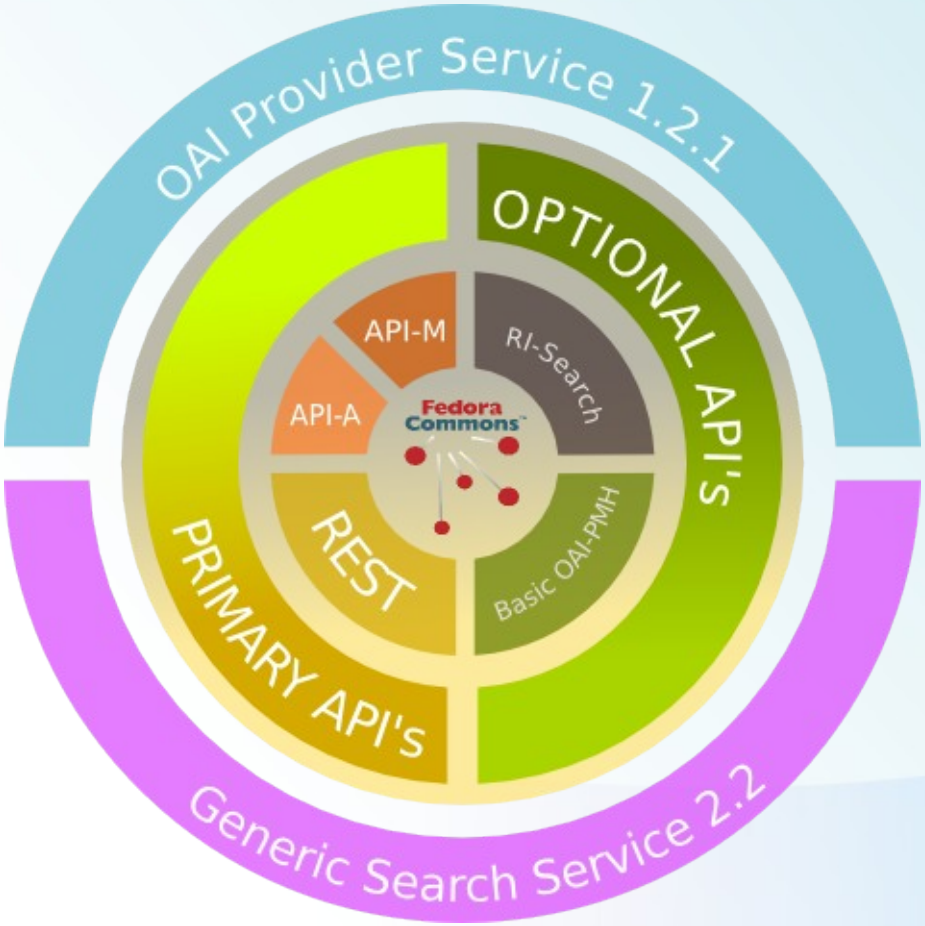
	Find Tuples	Find Triple
Query Language	SPARQL, iTQL	SPO
Response Type	CSV, Simple, Sparql, TSV, count	N-Triples, Notation 3 RDF/XML, Turtle, count

1.5 Web Service Interface: RISearch [2/2]

- Example iTQL Query:

```
select $object $label $description $owner $date $type from <#ri>
where $object <fedora-model:label> $label
and $object <fedora-model:ownerId> $owner
and $object <dc:date> $date
and $object <dc:description> $description
and $object <dc:type> $type
and $object <fedora-rels-ext:isMemberOf> <info:fedora/demo:1>
order by $label asc limit 15 offset 12
```

1.6 Framework Services



- Generic Search Service
- OAI Provider Service

2. Federico

2.1. What is Federico?

2.2. System Requirements

2.3. Use Cases

2.4. Content Model

2.5. Architecture

2.1 What is Federico?

- Fedora-Enabled Repository with Cocoon
- AJAX-based frontend for a C3Grid local repository of metadata
- Transparent Integration of Fedora with the Framework Services GSearch and OAI Provider
- Developed in the scope of the work package #3, Long-term Preservation of Digital Archives of Wissgrid, sponsored by the German Federal Ministry of Education and Research

2.2 System Requirements [1/2]

Hardware

- PC with a 1 gigahertz (GHz) processor or faster and network card
- 2 GB RAM
- 800 MB free disk space for the installation

Software

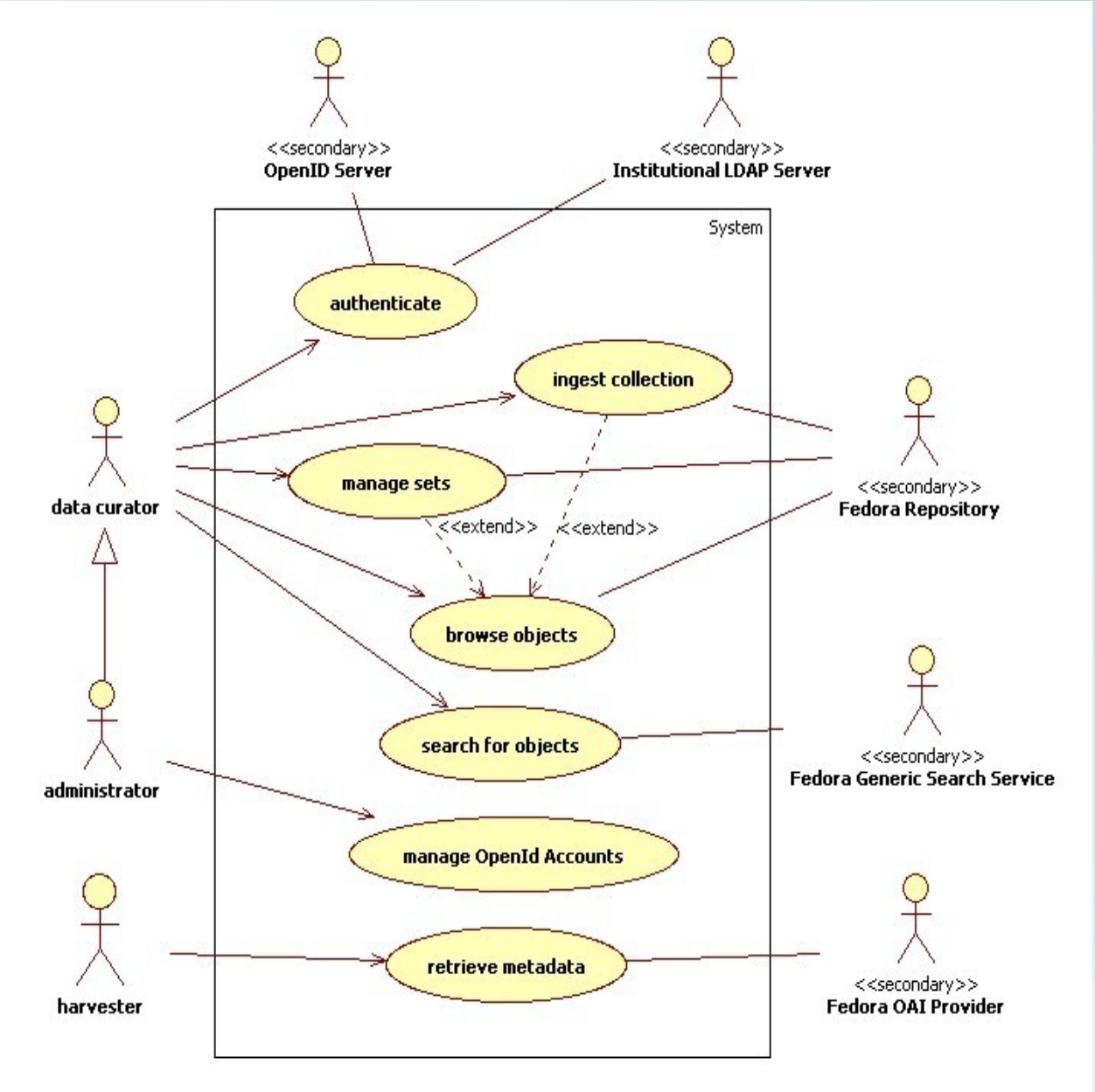
- Linux Distribution with X Window System
- Java JDK 1.6
- 3 MySQL Databases for Fedora Commons, Fedora OAI Provider, and openID accounts

2.2 System Requirements [2/2]

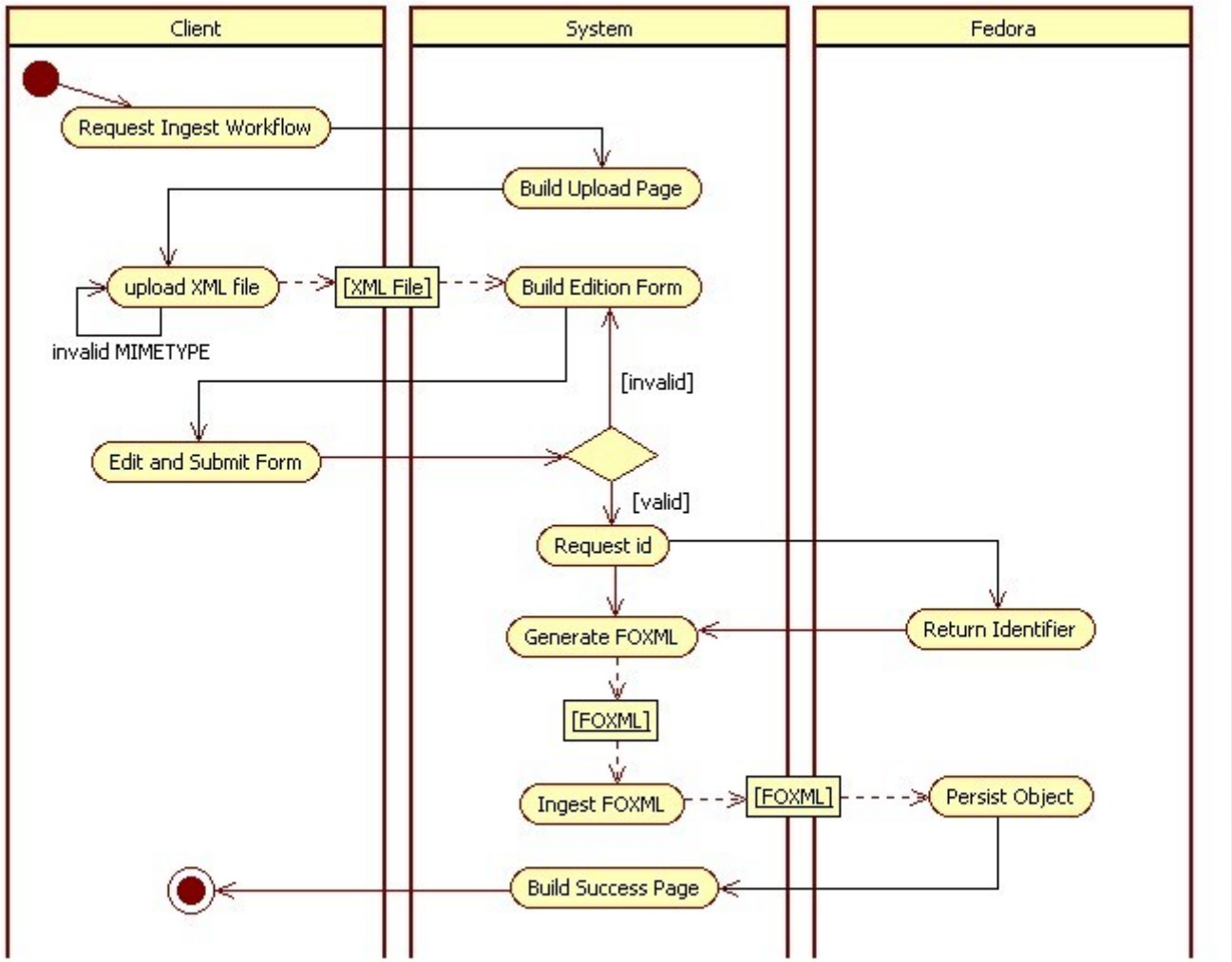
User

- PC with graphical interface and network card
- Keyboard and mouse
- Browser (preferably Mozilla Firefox) with Javascript enabled

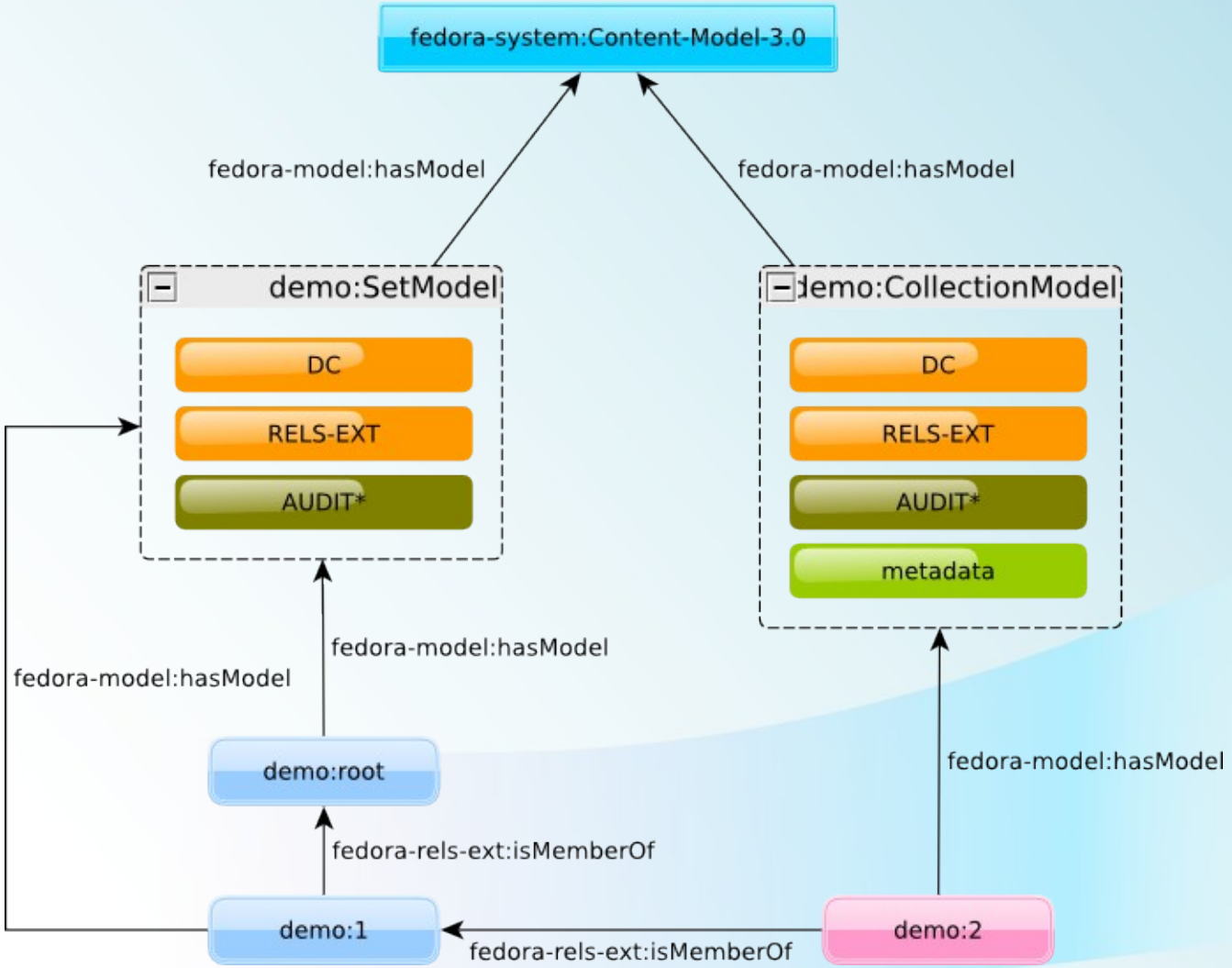
2.3 Federico Use Case



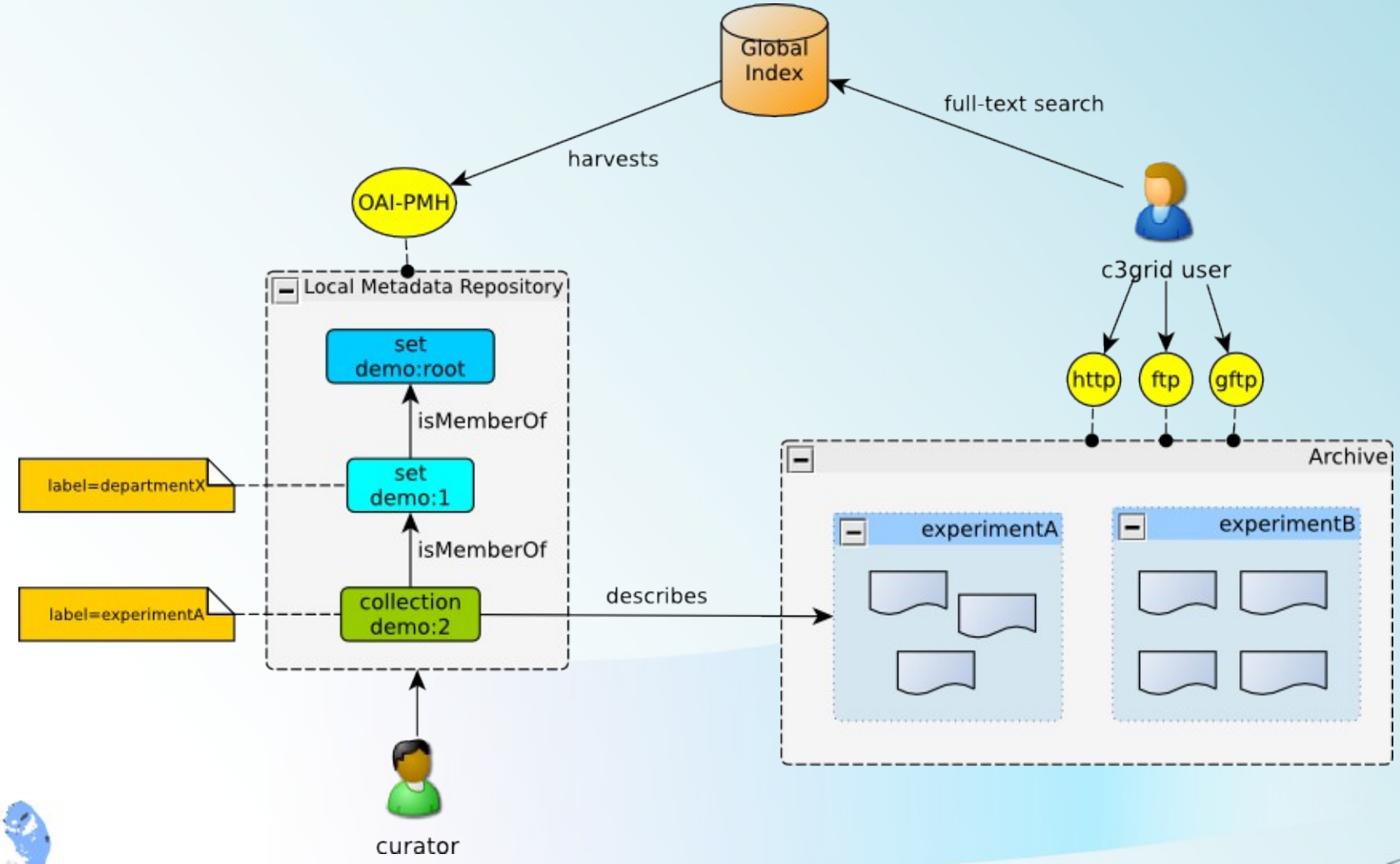
2.3 Activity Diagram – Ingest Collection



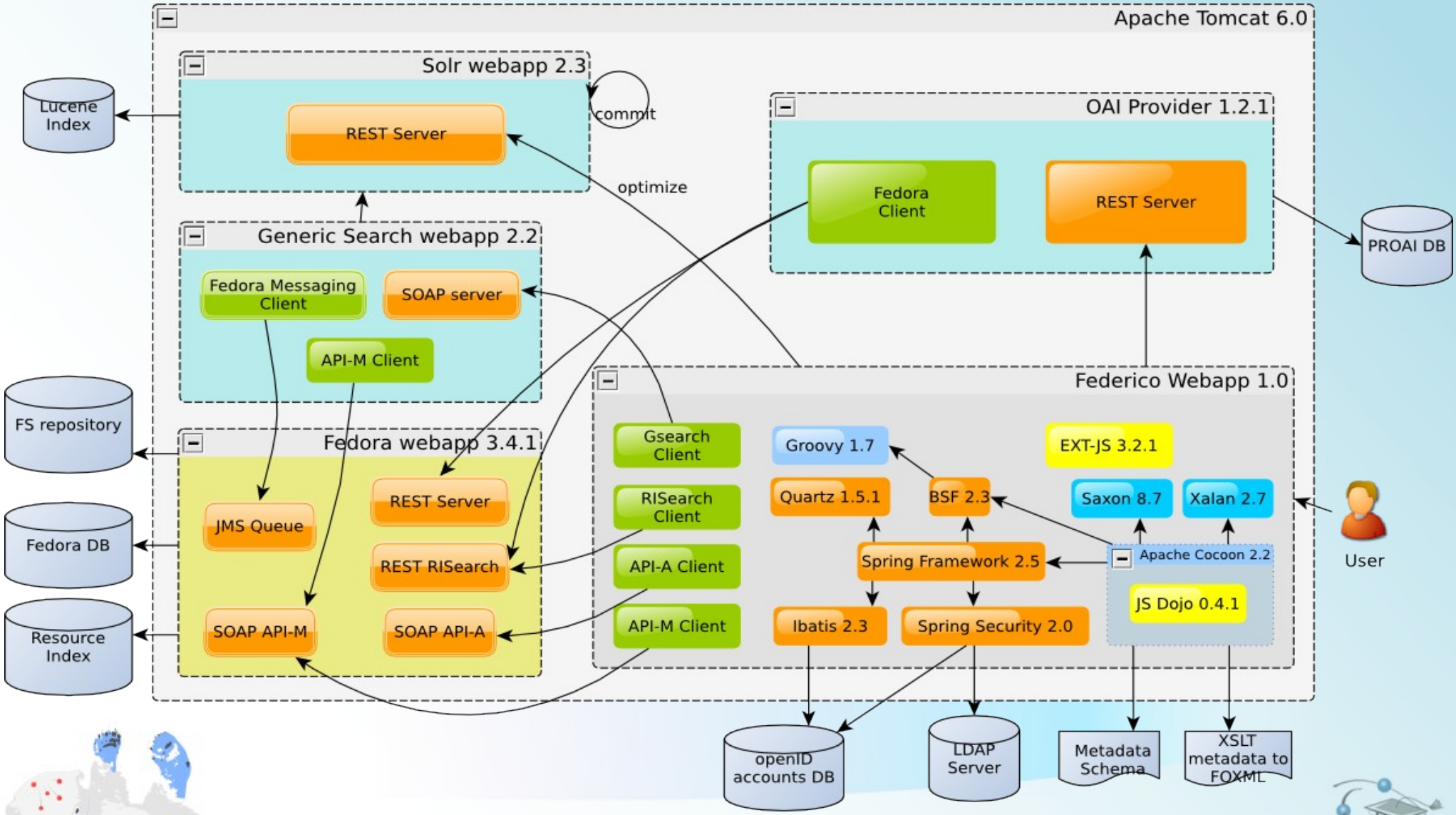
2.4 Content Model [1/2]



2.4 Content Model [2/2]



2.5 Architecture



3. Federico's Live Demo

3.1. User Interface

3.2. Authentication

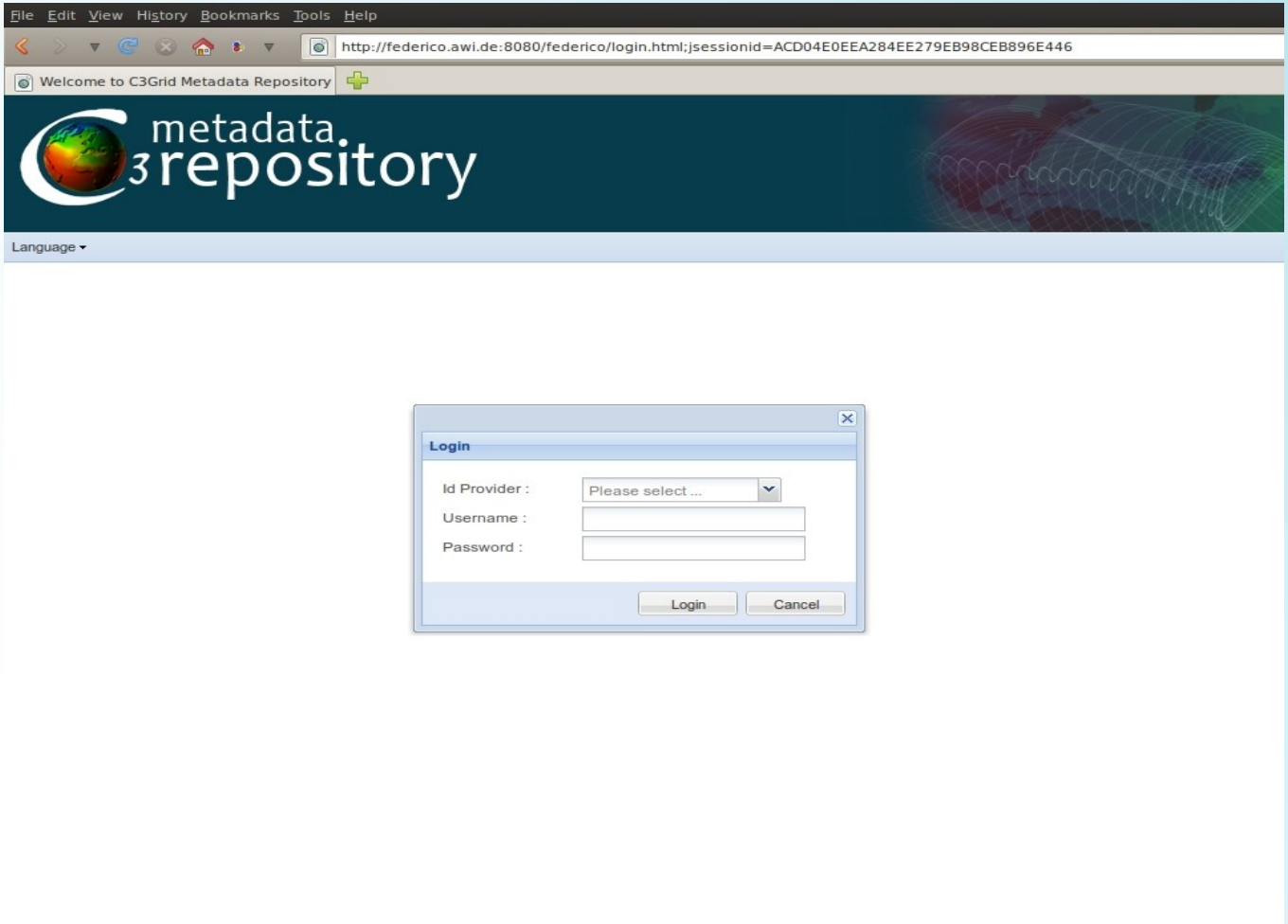
3.3. Ingest Collections

3.4. Full-text Search

3.5. OAI-PMH

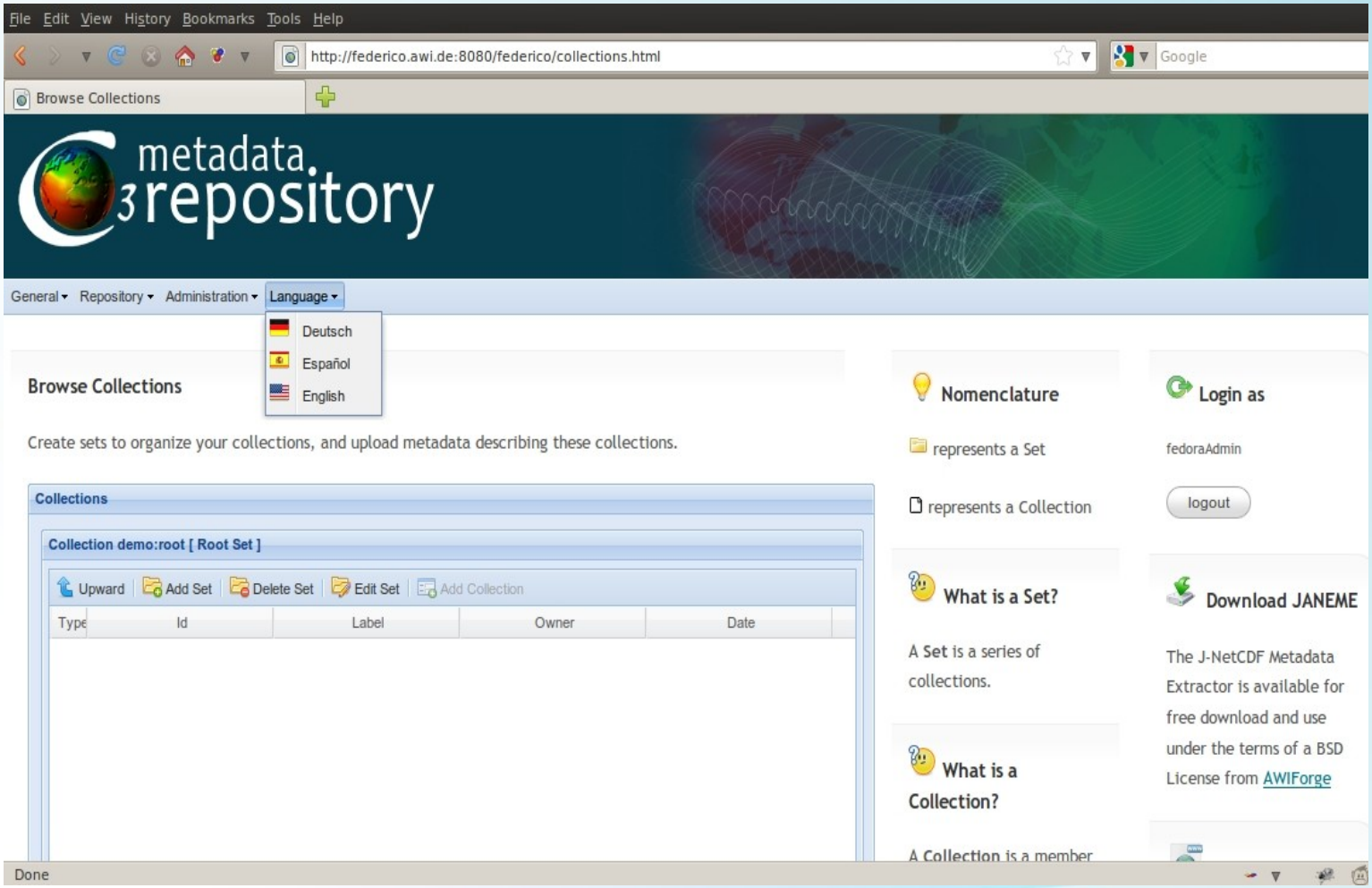
3. Federico's Live Demo: Screenshots

Login Form



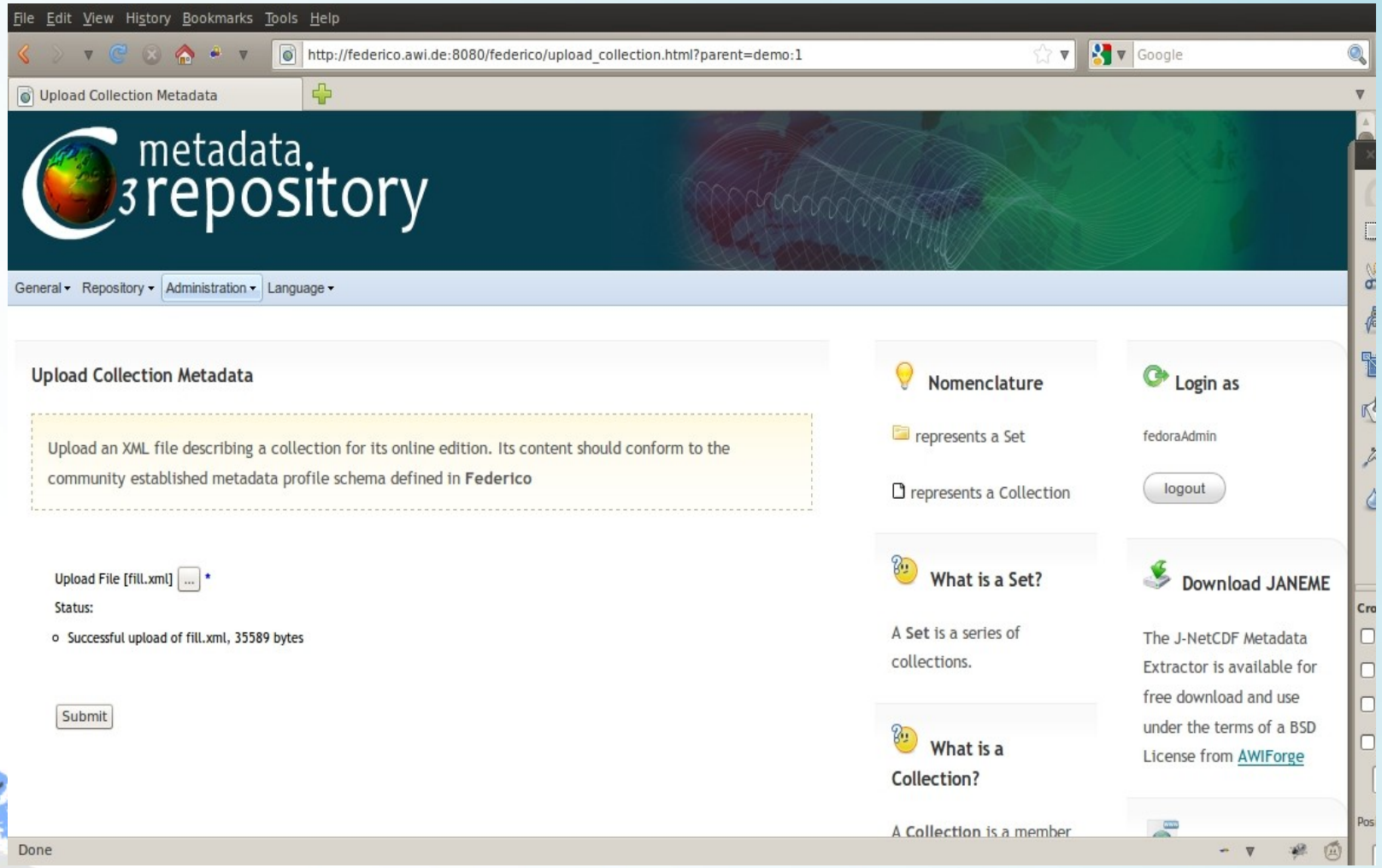
3. Federico's Live Demo: Screenshots

Browse Collections Form



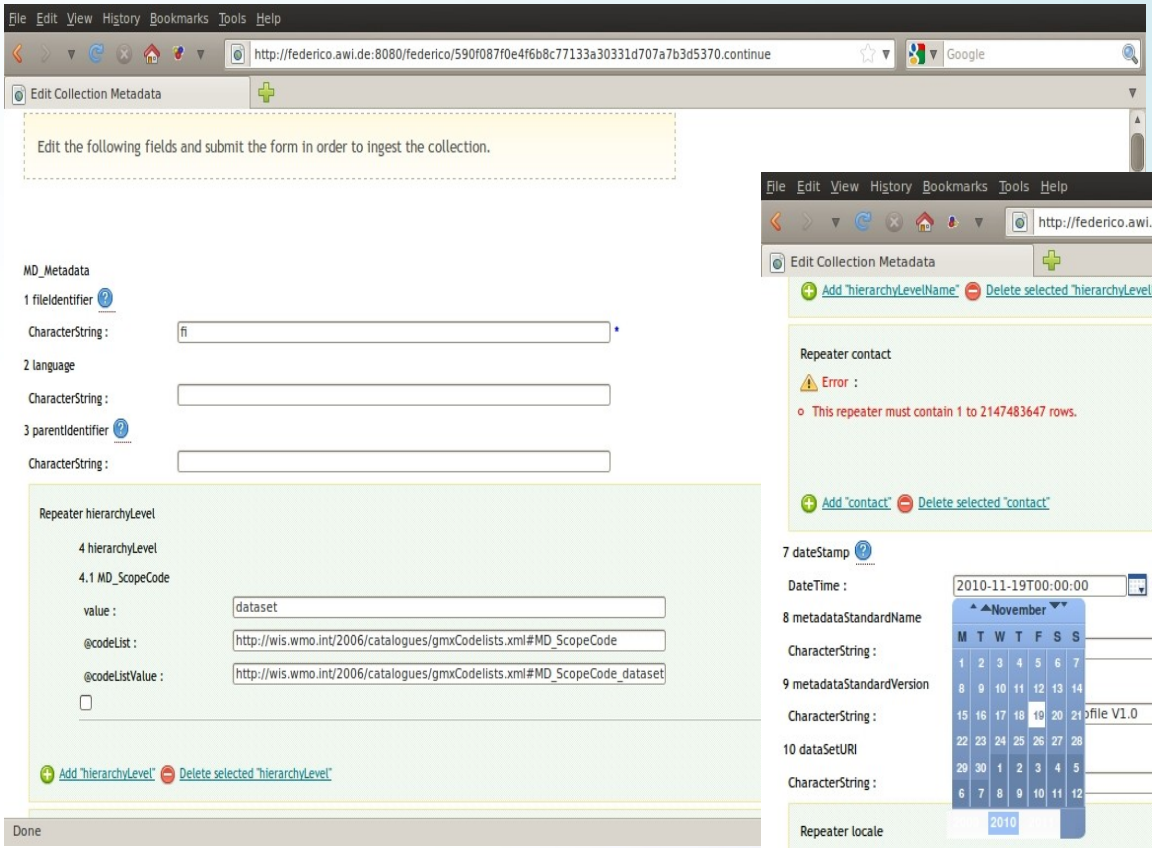
3. Federico's Live Demo: Screenshots

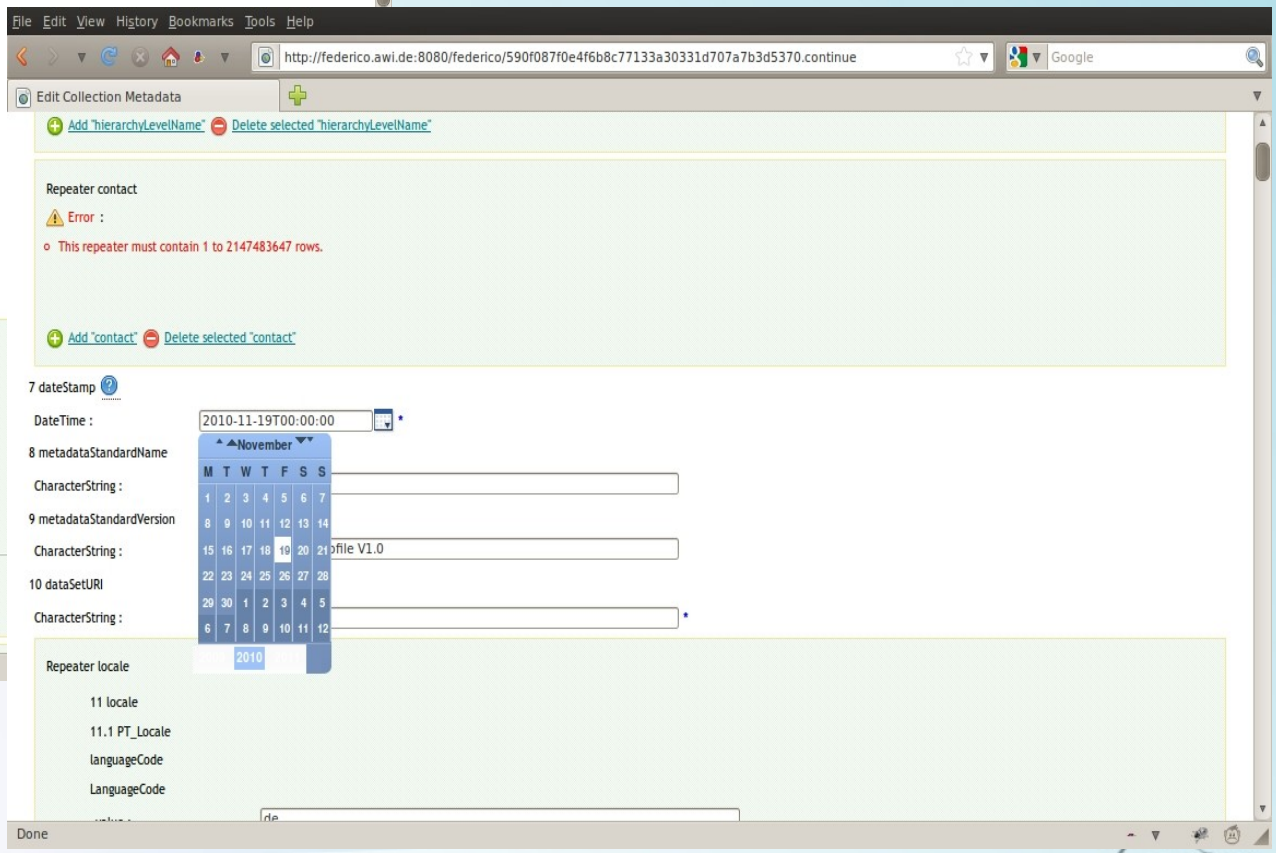
Metadata Upload



3. Federico's Live Demo: Screenshots

Metadata Edition





Summary

- Fedora as repository for digital information in research environment
  - Well defined API's
  - Content Model Architecture for the definition of “types” of objects
  - Harvesting through OAI-PMH
- Knowledge of XML is crucial
- Difficult UI implementation