

# IU DLP Infrastructure Update

Ryan Scherle  
Muzaffer Ozakca



INDIANA UNIVERSITY

# Outline

- What is the infrastructure project?
- Fedora
- Progress
  - Content models
  - Ingest tool
  - Delivery system
  - Policies
- Current status

**What is the infrastructure project?**

# IUDL infrastructure project

- 2-year project funded by UITS to re-engineer digital library infrastructure around Fedora
- Builds on experience with Fedora in context of EVIA Digital Archive (ethnomusicology video)
- 2 full-time staff, plus part-time from many others
- Dozens of legacy collections with roughly 100,000 digital objects
- New collections: some content-focused, some research-focused

# Digital objects

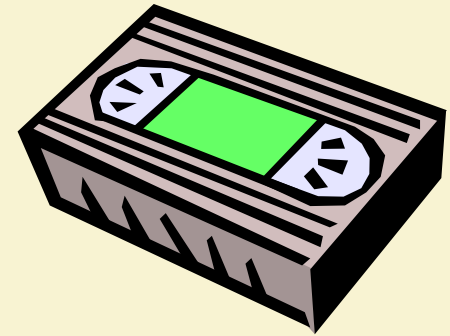
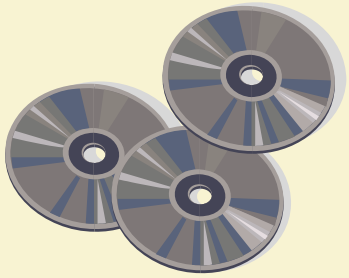
- Digital object  $\approx$  cataloged item
- Digital objects have many parts
  - Metadata
    - Descriptive, administrative, structural, preservation, ...
  - Preservation/archival files (several)
  - Delivery files (several)
  - Persistent identifier
- How do we keep them connected and organized?
  - Past: Good practice in file naming, directory organization, project documentation -not scalable!
  - Future: Digital object repository

Why do we need a repository?

The DLP Collections

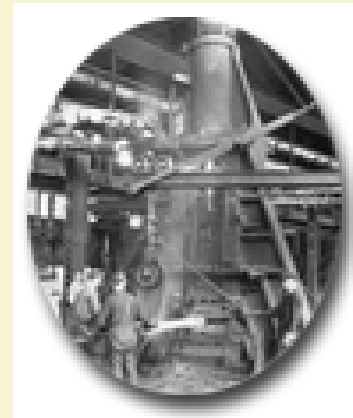
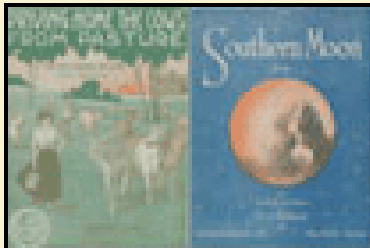
# Why do we need a repository?

- Centralize access and preservation functions for IU's digital collections
- Reduce DLP staff time and attention needed to create and maintain collections
- Enable librarians, curators, archivists to digitize new collections
- Stabilize costs to add objects to digital collections
- Enable coordination with other services (Sakai, OneSearch, etc.)
- Enable digital preservation



# Diversity

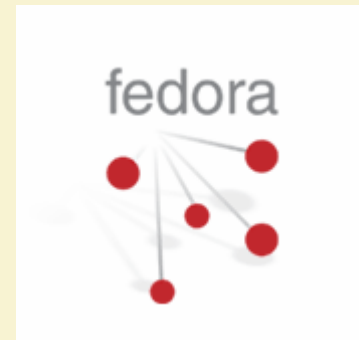
- Multiple media types
- Multiple brands
- Multiple tools



Fedora

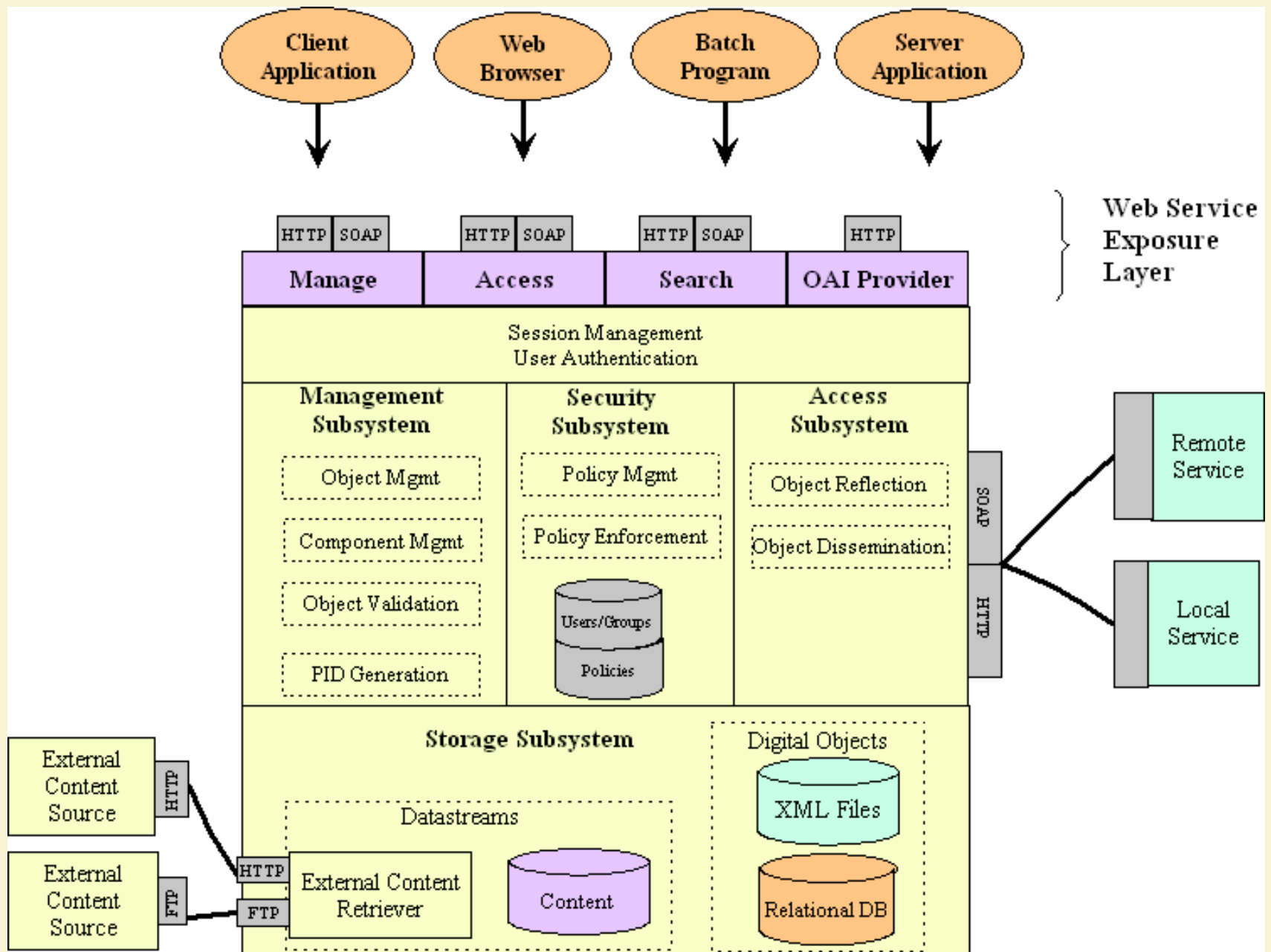
# Fedora

- FEDORA
  - Flexible
  - Extensible
  - Digital
  - Object and
  - Repository
  - Architecture



# What does Fedora do?

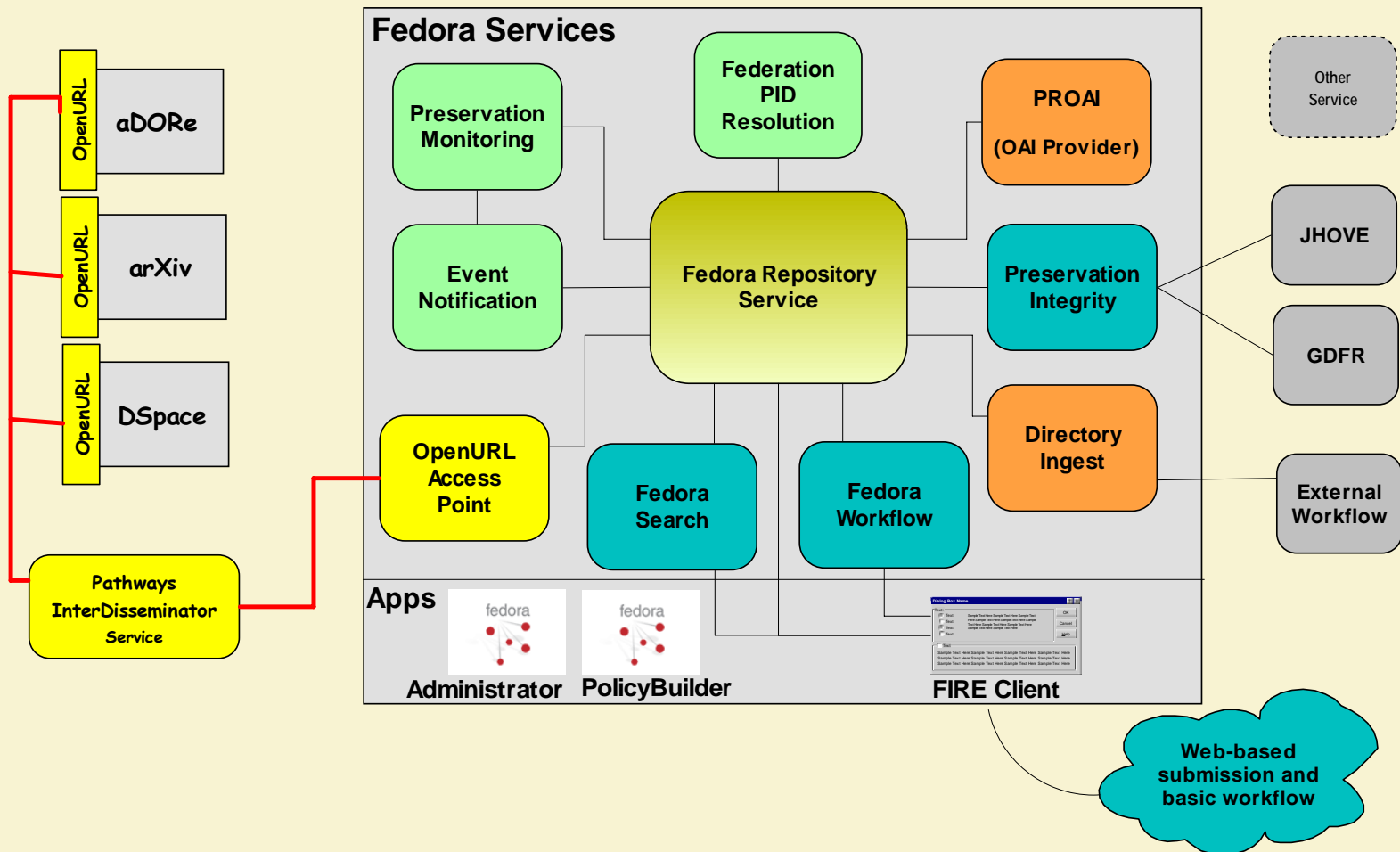
- Provides database features for digital objects
- Manages files or references to files that make up digital objects
- Manages associations between objects and interfaces
- Invokes behaviors of objects



# Critical Fedora features

- Core repository functions are separated from utilities that act on the repository
- Datastreams may be stored locally or distributed across the web
- Local data is stored in a straightforward manner
- Disseminators provide “just-in-time” transformations
- Growing user community

# Fedora Service Framework

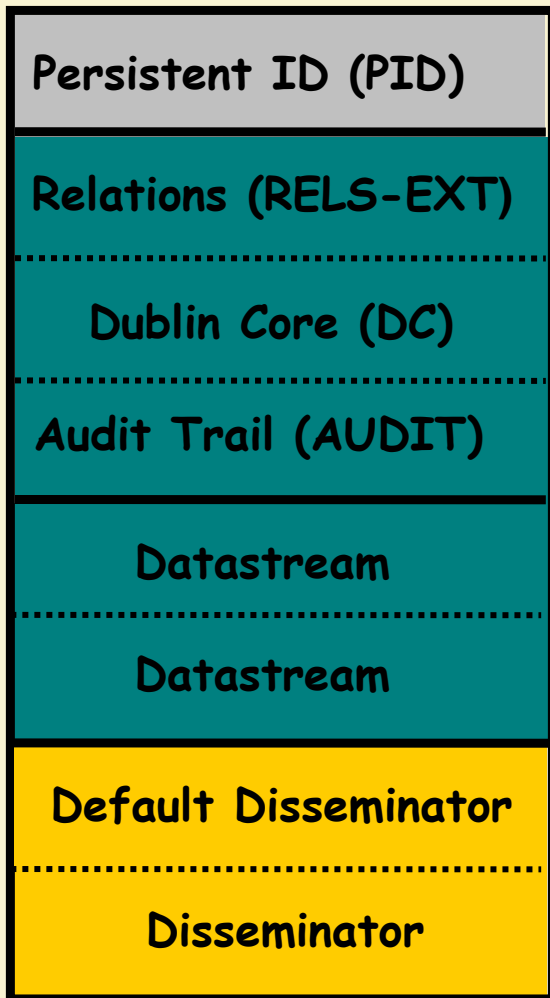


# Flexibility comes with a price

- Using Fedora takes significant work (right now)
  - Cataloging/ingest tools
  - Advanced searching/browsing
  - End-user user interface
  - Preservation services
- Fedora is not a complete system, it's just plumbing (right now)

# Content Models

# Fedora Object Model



} Digital object identifier

} Reserved Datastreams  
*Key object metadata*

} Datastreams  
*Aggregate content or metadata items*

} Disseminators  
*Pointers to service definitions to  
provide service-mediated views*

# Content models

- A content model describes the internal structure of a class of Fedora objects
  - Number & type of datastreams
  - Number & type of disseminators
- Benefits of a content model
  - A method to describe the structure of similar Fedora objects
  - Facilitate the creation of “batches” of objects
  - Standardize handling of Fedora objects by tools outside the repository

# Content model goals

- Maintain consistency with other Fedora users
- Standardize disseminators across objects, shifting the implementation to suit the needs of the collection
  - Makes it easier to build collection-independent applications on top of Fedora
  - It's possible to change implementations behind the scenes
- Maintain functionality of existing collections



# Standard disseminators

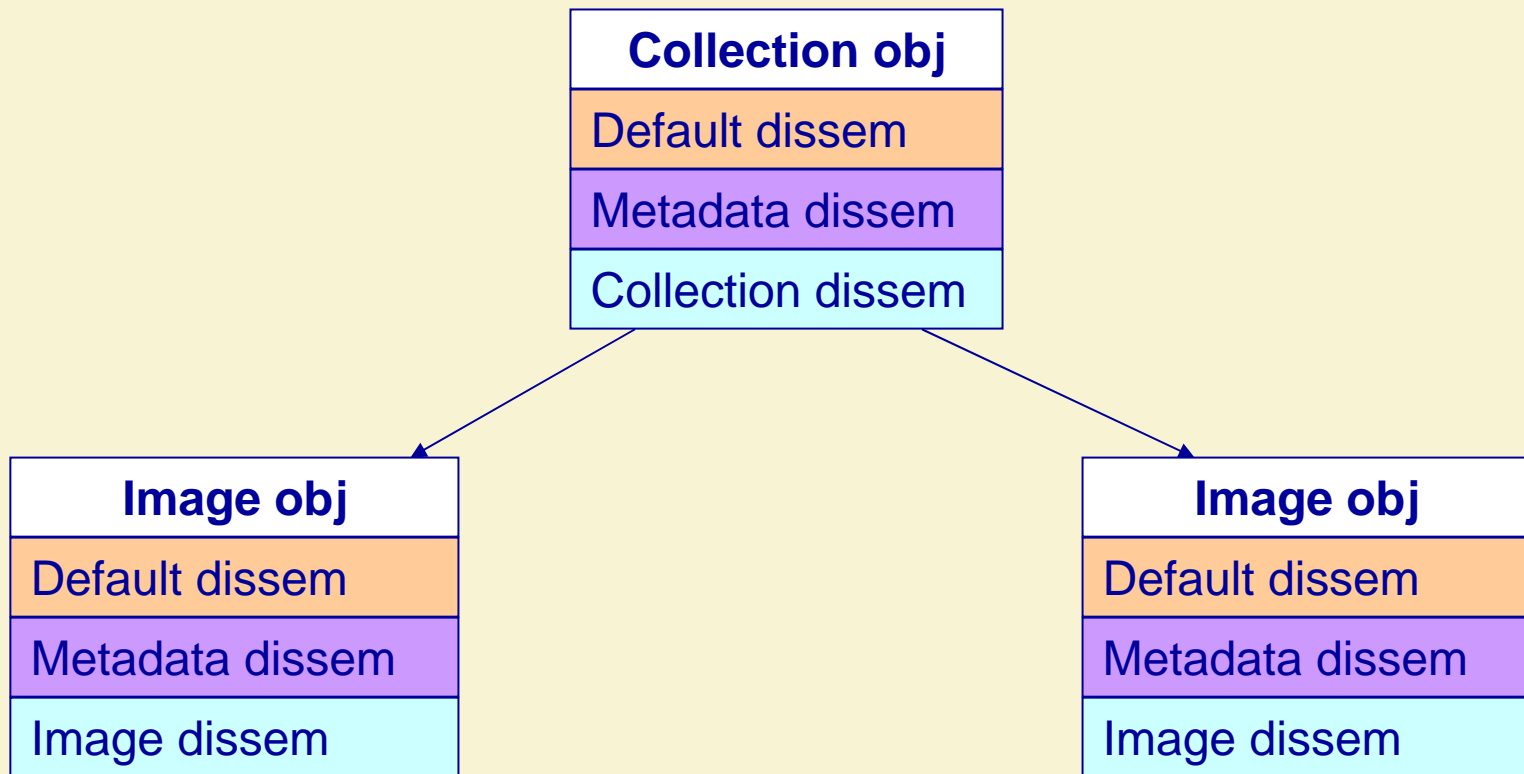
- All objects can implement the default disseminator for cross-collection functionality
- Most objects implement the metadata disseminator
- Most objects implement type-specific disseminators

<b>Default dissem</b>
getLabel
getDefaultContent
getPreview
getFullView

<b>Metadata dissem</b>
getDC
getMetadata(type)

# Content model for simple images

- Each image is a single Fedora object
- Images are available in a variety of sizes
- Each image belongs to a collection, which performs presentation



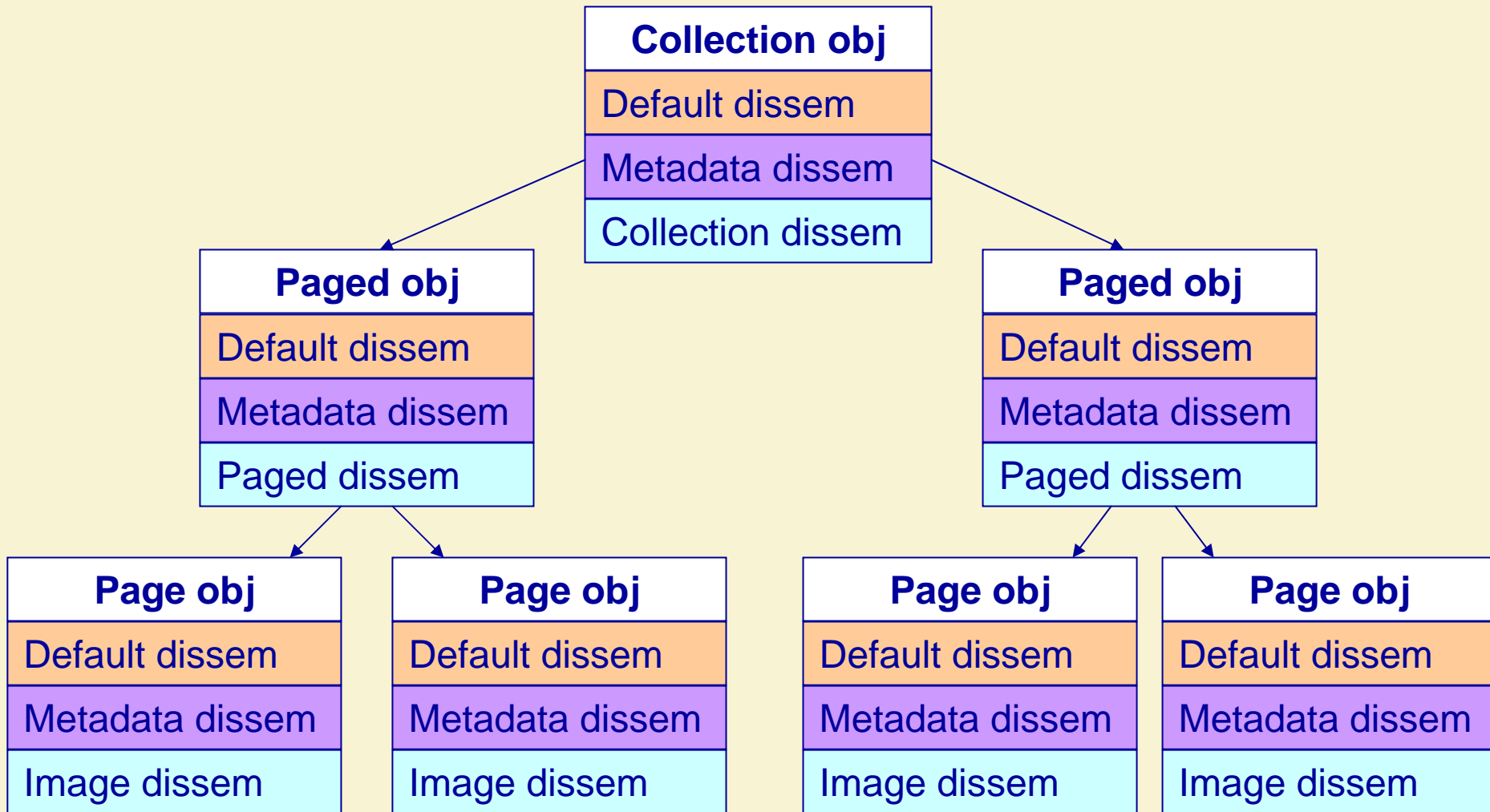
# But what about the metadata?

- Different content types have different types of metadata
  - MARC for general library holdings
  - MODS for collections we catalog
  - TEI for textual collections
  - EAD for archival collections
  - Combinations: Some items need METS for structure, TEI for text, MODS for description, etc.
- METS provides a standard way of dealing with all of these types of data

# Image Demo

- [Sam Park](#)
- [Hohenberger collection](#)

# Paged document content model



# Paged document demo

- Image
- Letter
- Collection
- Page turner

# Object-level disseminators

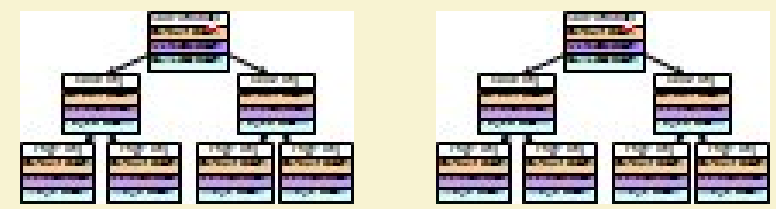
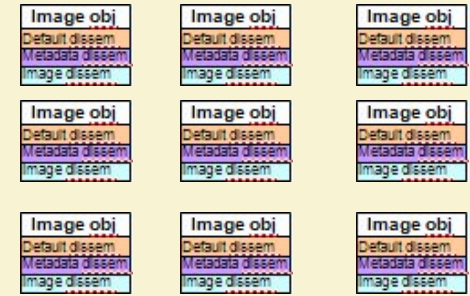
- **Image**
  - `getThumbnail`
  - `getScreenSize`
  - `getLarge`
  - `getMaster`
- **Video**
  - `getSmilFile`
  - `playSmilFile`
  - `getStructMap`
  - `getActionObject`
  - `getObjectID`
- **PagedImage**
  - `getNumChildren`
  - `getChildren`
- **PagedText**
  - `getSummary`
  - `getChunkList`
  - `getChunk(label)`
  - `getRawText`
  - `getFriendlyText`
  - `getTextPage(num)`
- **Printable**
  - `getPrintableVersion`

# Collection-level disseminators

- **Collection**
  - `getSize`
  - `listMembers(start,max)`
- **CollectionRender**
  - `renderItemPreview(pid)`
  - `renderItemFullView(pid)`
- **CollectionPagedImage**
  - `viewPageTurner(pid, pagenum)`
- **CollectionPagedText**
  - `viewText(pid, pagenum, style)`
  - `viewChunk(pid, label, style)`
  - `viewPage(pid, num, style)`

# Ingesting data

# The goal



# Required features

- Ingest common content types:
  - Images
  - Paged documents
  - Textual documents
- Allow for easy creation of new content types
- Must support several workflows
  - Metadata or media may be primary
  - Most objects include derived media
  - Systematic changes to metadata may be desired
  - May need to connect with external tools for metadata generation, validation, etc.
  - A workflow engine may sit on top of the ingest system

# Fedora admin client

- Comes with Fedora
- Geared towards admins rather than end users
- No systematic way of entering data or attaching files
- Very flexible
- The only way to create disseminators
- Tedious

Properties

Datastreams

Disseminators

SCREEN

THUMBNAIL

RELS-EXT

LARGE

DC

PURL\_REDIRECT

METADATA

New...

State **Active**

Control Group Managed Content

Created 2007-02-13T09:53:35.000Z

Label Palace : the joy spot of town - copy - 01 - page - 01

MIME Type image/jpeg

Format URI

Alternate IDs

Fedora URL <http://bl-ldlp-mz.ads.iu.edu:8080/fedora/get/iudl:25222/SCREEN>



View

Import...

Export...

Purge...



User: admin

terms Search

View Image Details:

Parent Collections: [Hohenberger Photographs](#)

<b>Title</b>	Alice in Wonderland float (take 2)
<b>Creator(s)</b>	
<b>Research Fields, Courses and Disciplines</b>	
<b>Keyword(s)</b>	Floats (Parades) Carnival
<b>Description</b>	
<b>Publisher</b>	
<b>Contributor</b>	Indiana University. Digital Library Program Lilly Library (Indiana University, Bloomington)
<b>Date</b>	Friday, December , 2006
<b>Type</b>	
<b>Format</b>	
<b>Source</b>	
<b>Language</b>	
<b>Relation</b>	
<b>Coverage</b>	
<b>Rights</b>	
<b>Additional Notes</b>	

### Datastream Preview



Attached Files		
Name	Description	MIMETYPE
<a href="#">Hoh027-000-0035.jpg</a>	Hoh027-000-0035.jpg	image/jpeg
<a href="#">LL-SLO-004696.jpg</a>	LL-SLO-004696.jpg	image/jpeg

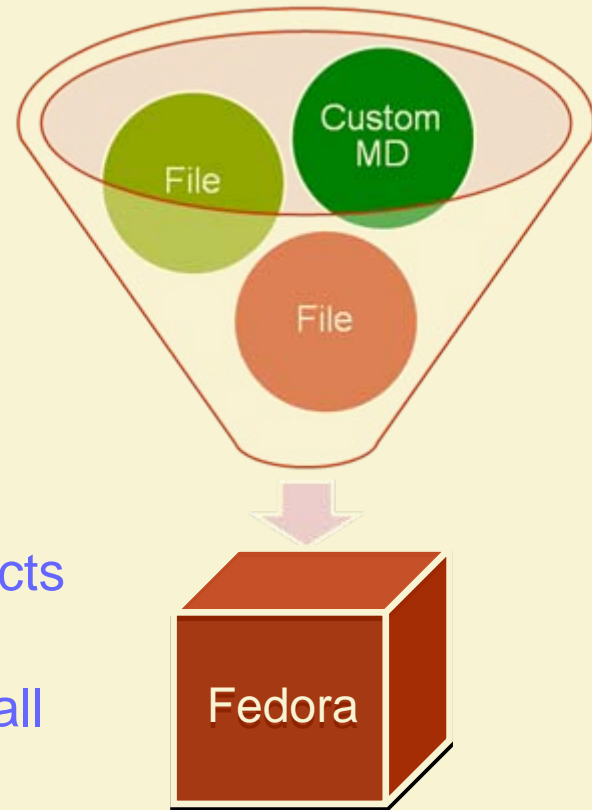
Related Links	
Link	Description

# Fez

- End-to-End GUI system
- Highly customizable content models, workflow, security
- Customizable role and group based access control
- Growing community
- Originally developed as an Institutional Repository
- Many preset content models
- Can create “extension” metadata based on an XSD
- External MySQL database for workflow/vocabulary data
- GPL

# Fez

- Single object ingest
  - Through Web UI
  - ImageMagick/JHOVE integration
- Bulk ingest:
  - Upload files to a directory
  - Also can import existing Fedora objects in bulks
  - Templates for metadata common to all objects, manual updates for the rest
  - Batches possible, but only one file per object
- No disseminators
- Custom metadata can be stored as a simple XML file
- Objects must use “compound” content model



## Collection

### [MUZO'S COLLECTION](#)

[View](#) collection information.

Displaying items 1- 15 out of 15

[Previous](#) | [Next](#)

Preview	Title	Description	Creation Date
	<a href="#">Hoagy Carmichael Collection</a>		Wed, April 5, 2006
	<a href="#">U.S. Steel Gary Works Photograph Collection, 1906-1971</a>		Mon, October 17, 2005
	<a href="#">IN Harmony: Sheet Music from Indiana</a>		Wed, April 5, 2006
	<a href="#">Dido Image Bank</a>		Mon, October 17, 2005
	<a href="#">Slocum Puzzles Collection</a>		Thu, June 1, 2006
	<a href="#">Item Four</a>		Sun, April 4, 2004
	<a href="#">Document folder</a>	Documents are stored in this folder	Thu, December 28, 2006
			

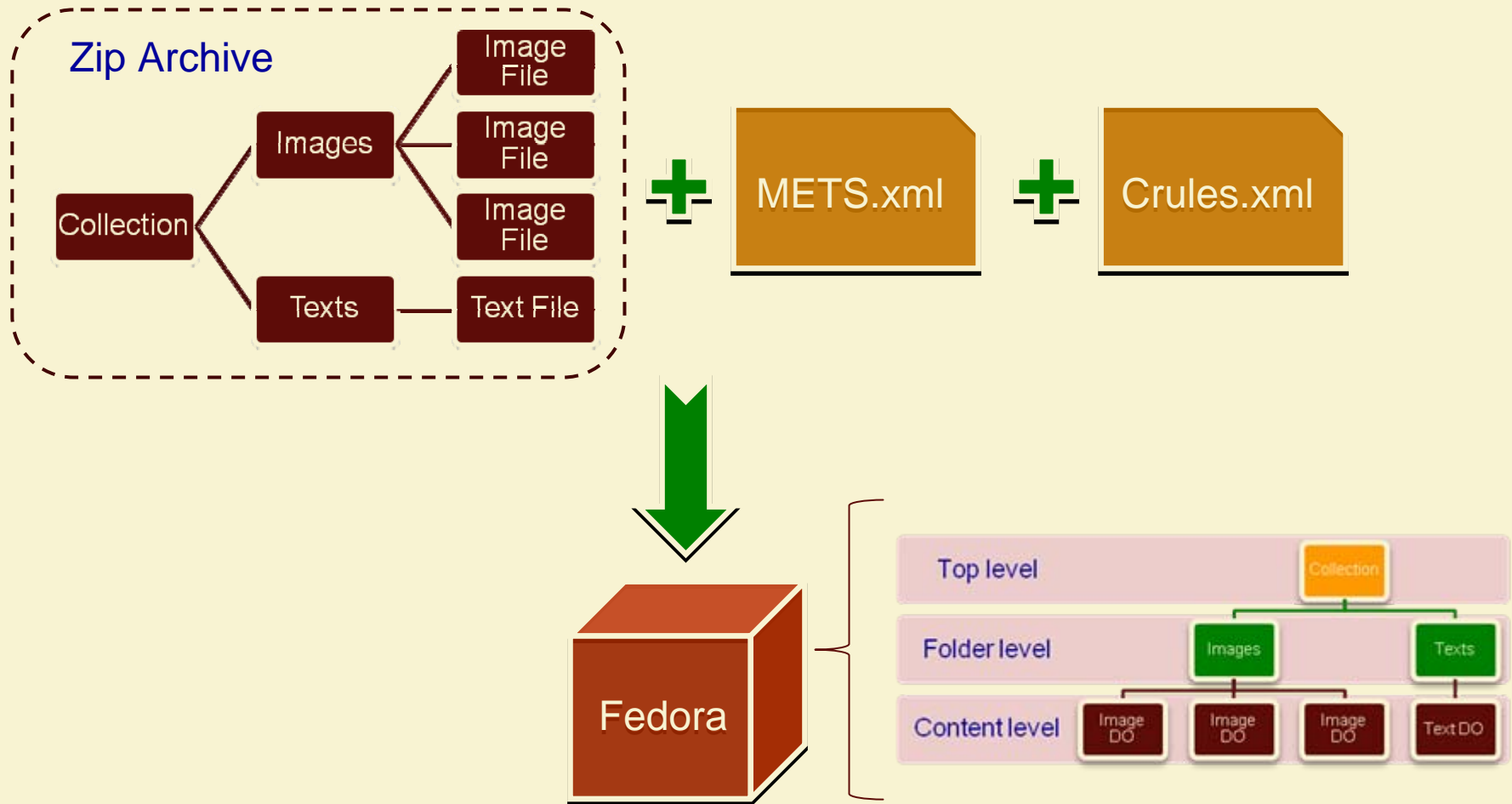
# Elated

- End to end complete system for digital collections
- Emphasis on being simple to install and use
- Simple customizable metadata and a simple workflow supported
- GPL

# DirIngest

- Ingests objects from a structured ZIP file
- Highly flexible
- User must create METS structure by hand
- Doesn't handle disseminators
- Can create some RELS-EXT data, but not fully flexible
- Cannot modify existing objects/collections
- Easy to use OhioLink Bulk Ingest































# Dirlingest



# Batch modify

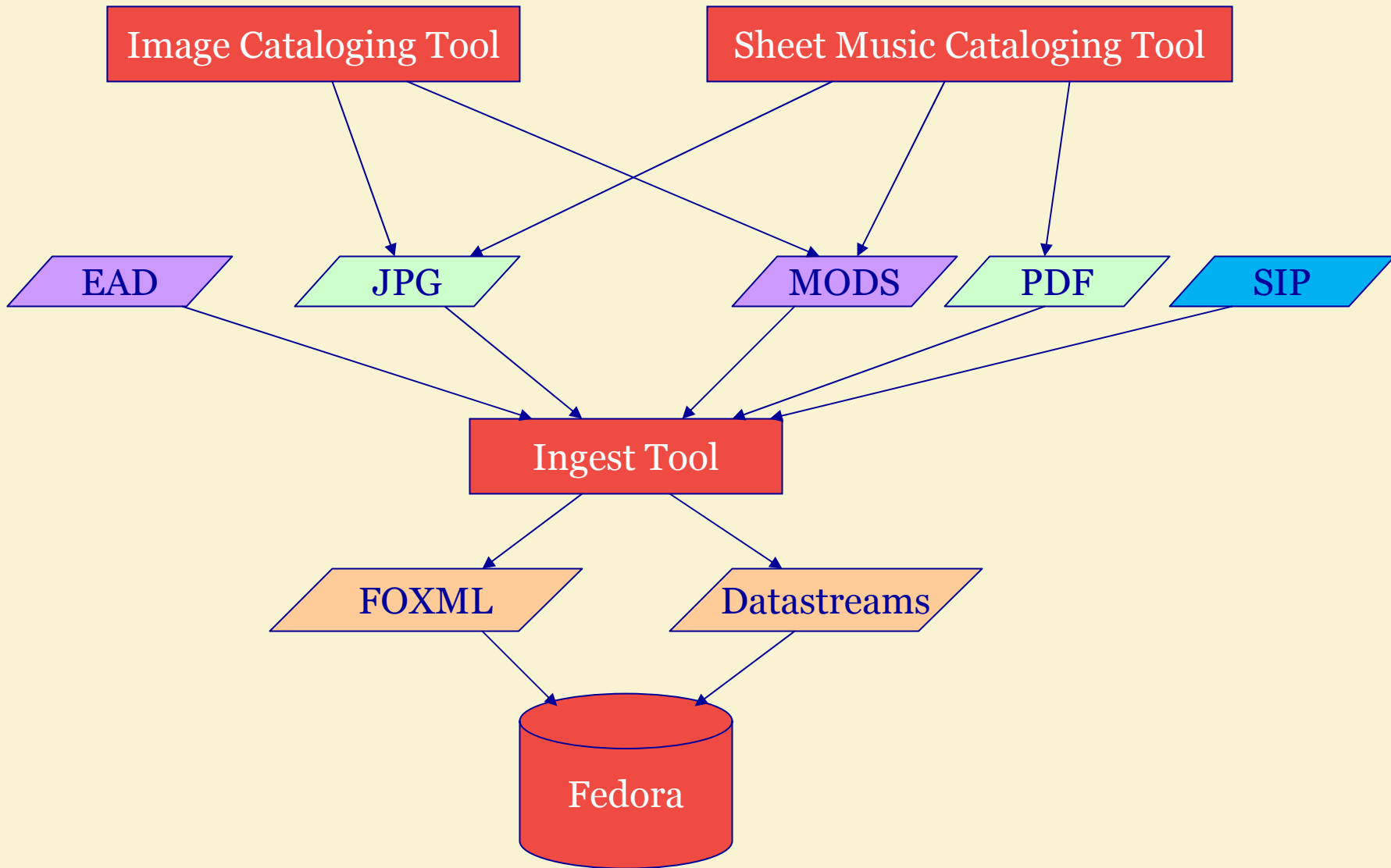
- A method of controlling API-M with simple XML statements
- Can create “empty” objects and change them in systematic ways.
- Requires manual (or programmatic) creation of the modify scripts
- Can be used in conjunction with other tools...

# Summary

	Fez	Elated	Valet	Dir Ingest	Batch Modify	Admin Client
Ease of install						
Native CM						
Custom CM						
Workflow Neutrality						
Batch ingest						

# Indiana Ingest Tool

- A structured interface between a workflow management or repository management GUI and the Fedora repository
- Focused on simple input formats for maximum flexibility
- Keeps the tools independent of the repository architecture
- Builds the FOXML, rather than requiring a full structure to be pre-built
- Binds disseminators
- Creates RELS-EXT relationships
- Can create and/or alter items in a collection
- Auto-generates technical metadata with JHOVE or XSLT.



# Performing an ingest

- Place source metadata in an accessible location (filesystem, website)
- Place media files (both master and derivative) in an accessible location
- Define the "collection configuration"
- Run the ingest process
- Receive report

# Sample collection config file

```
<cc:collectionName>Hoagy Carmichael Correspondence</cc:collectionName>
<cc:contentModel>paged</cc:contentModel>
<cc:collectionID>hoagy</cc:collectionID>
<cc:collectionPID>iudl:6</cc:collectionPID>
```

Collection defn

```
<cc:existingItem>
  <cc:federalItemExists action="alter"/>
</cc:existingItem>
```

What to do  
If item exists

```
<cc:masterContent type="image" subtype="tif">
  <cc:sourceLocation="local fs">{path to master images}</cc:source>
  <cc:extension>.tif</cc:extension>
</cc:masterContent>
<cc:derivedContent derivativeType="images">
  <cc:sourceLocation="local fs">{path to derivative images here}</cc:source>
  <cc:extension item="thumb">-thumb.jpg</cc:extension>
  <cc:extension item="screen">-screen.jpg</cc:extension>
  <cc:extension item="large">-full.jpg</cc:extension>
</cc:derivedContent>
```

File defn

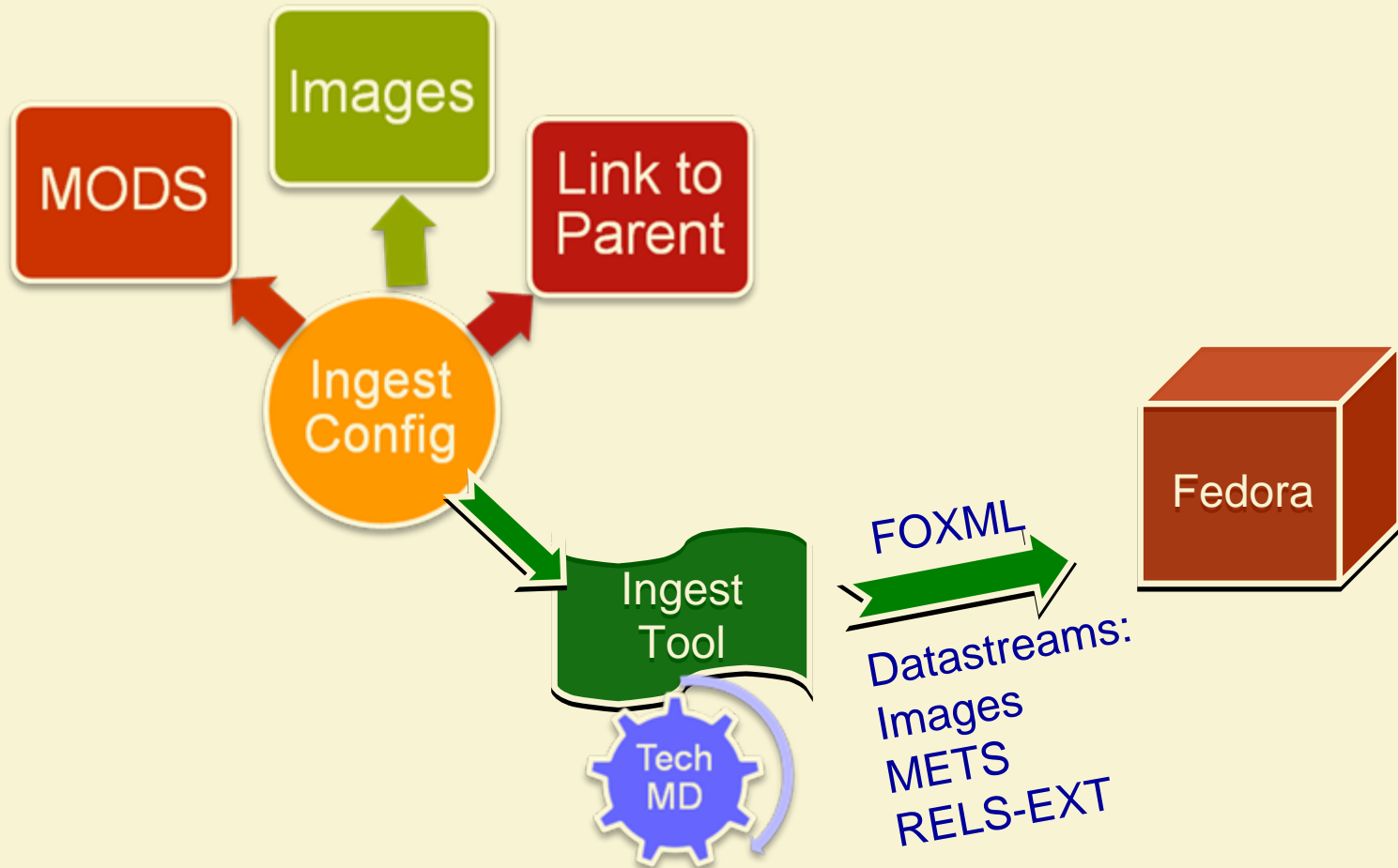
```
<cc:descriptiveMetadata>
  <cc:metadataItem type="ead" authoritative="true" level="collection">
    <cc:sourceLocation="local fs">{path to ead}</cc:source>
  </cc:metadataItem>
```

Desc. metadata

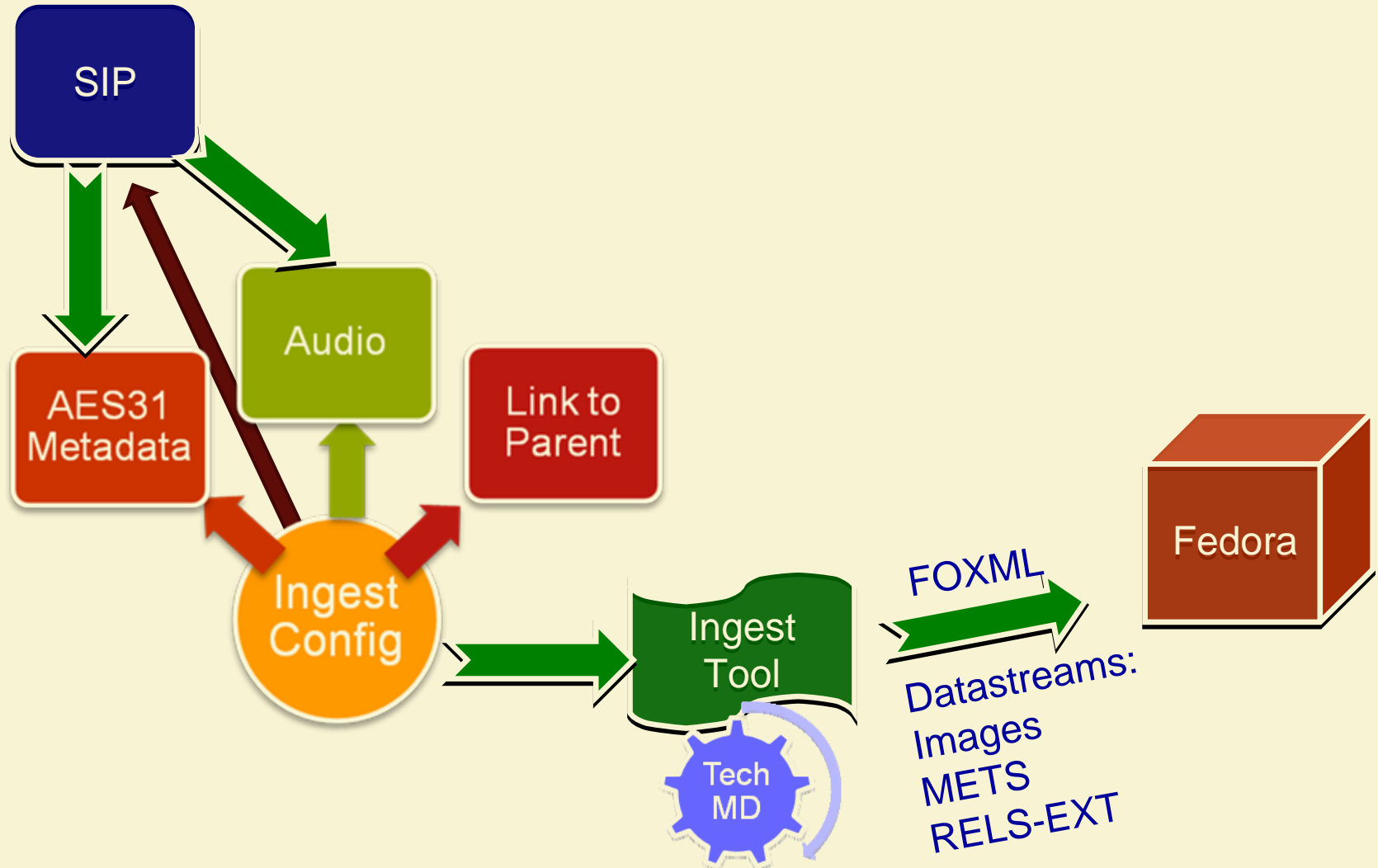
```
...
<cc:technicalMetadata>
  <cc:metadataItem type="mix" authoritative="true" level="masterContent">
    </cc:metadataItem>
  ...
```

Tech. metadata




































# Example – sheet music



# Example – preservation package



# Summary

	Fez	Elated	Valet	Dir Ingest	Batch Modify	Admin Client	IU Tool
Ease of install							
Native CM							
Custom CM							
Workflow Neutrality							
Batch ingest							

# Search and delivery

# Search system

- Uses Fedora Generic Search to extract objects from Fedora and index them
- The DLP SRU server is based on an implementation by OCLC
- Any SRU client can retrieve data from this server, but it is typically used by our tools

# The Jerry Slocum Mechanical Puzzle Collection

<http://www.dlib.indiana.edu/collections/slocum/>

# METS Navigator

- METS Navigator is a METS-based system for displaying and navigating multi-image digital objects.
- It was built to be extendible and configurable.
- Web pages with navigational structure are built from metadata in the repository.

# Using METS Navigator with Fedora

- METS document must meet minimal format requirements
  - Logical and physical structMap
  - Files marked with USE and GROUPLD attributes
  - Files are URLs that point to Fedora
- METS Navigator may be called from a disseminator, but it is better if called separately.

# Cross-repository functionality

[Aquifer Asset Actions Demo](#)

# Policies and documentation

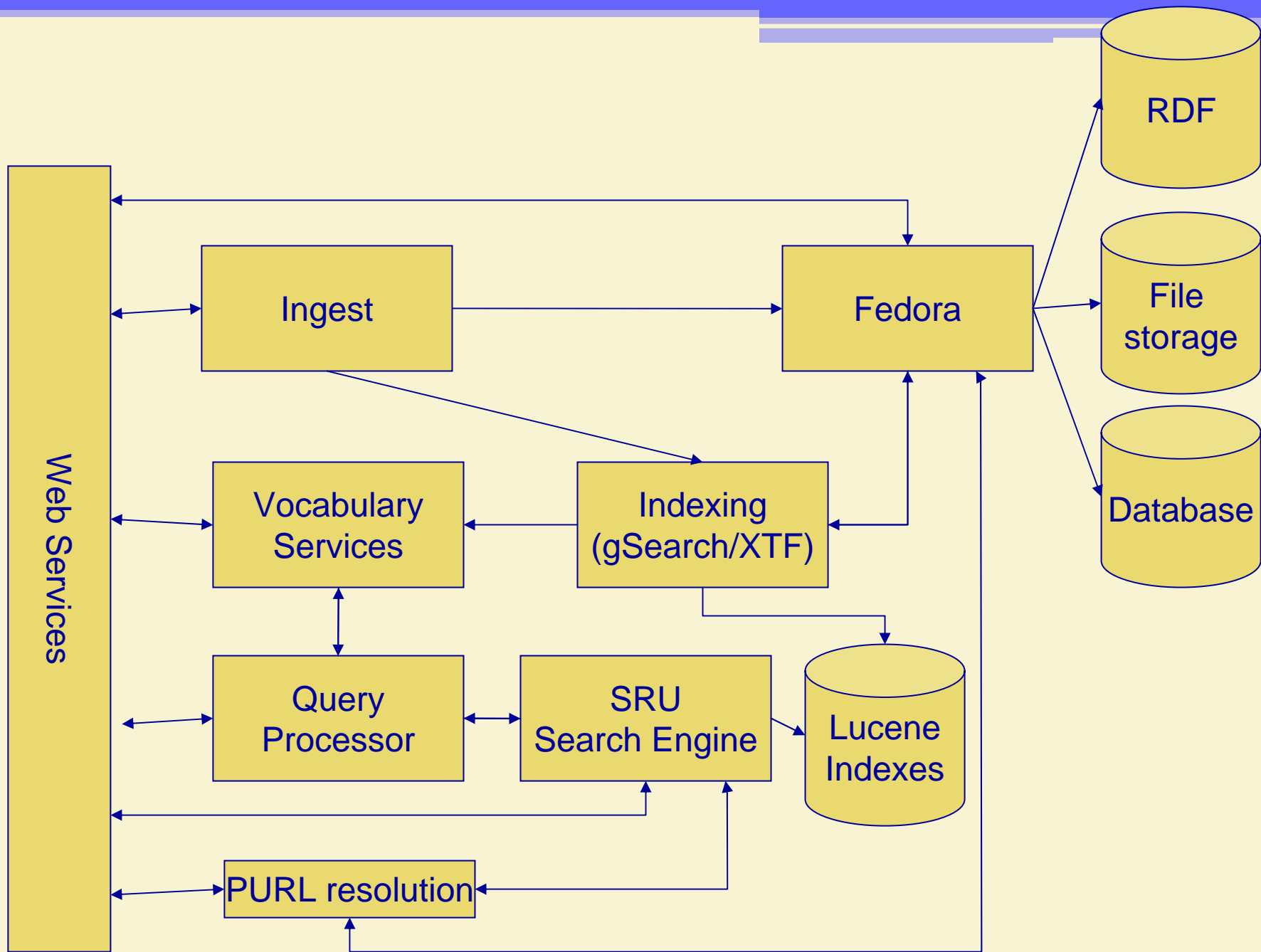
# Policies

- File naming
- Identifiers
- New objects checklist
- New collections checklist
- Preservation policies
- Turning policies into validation

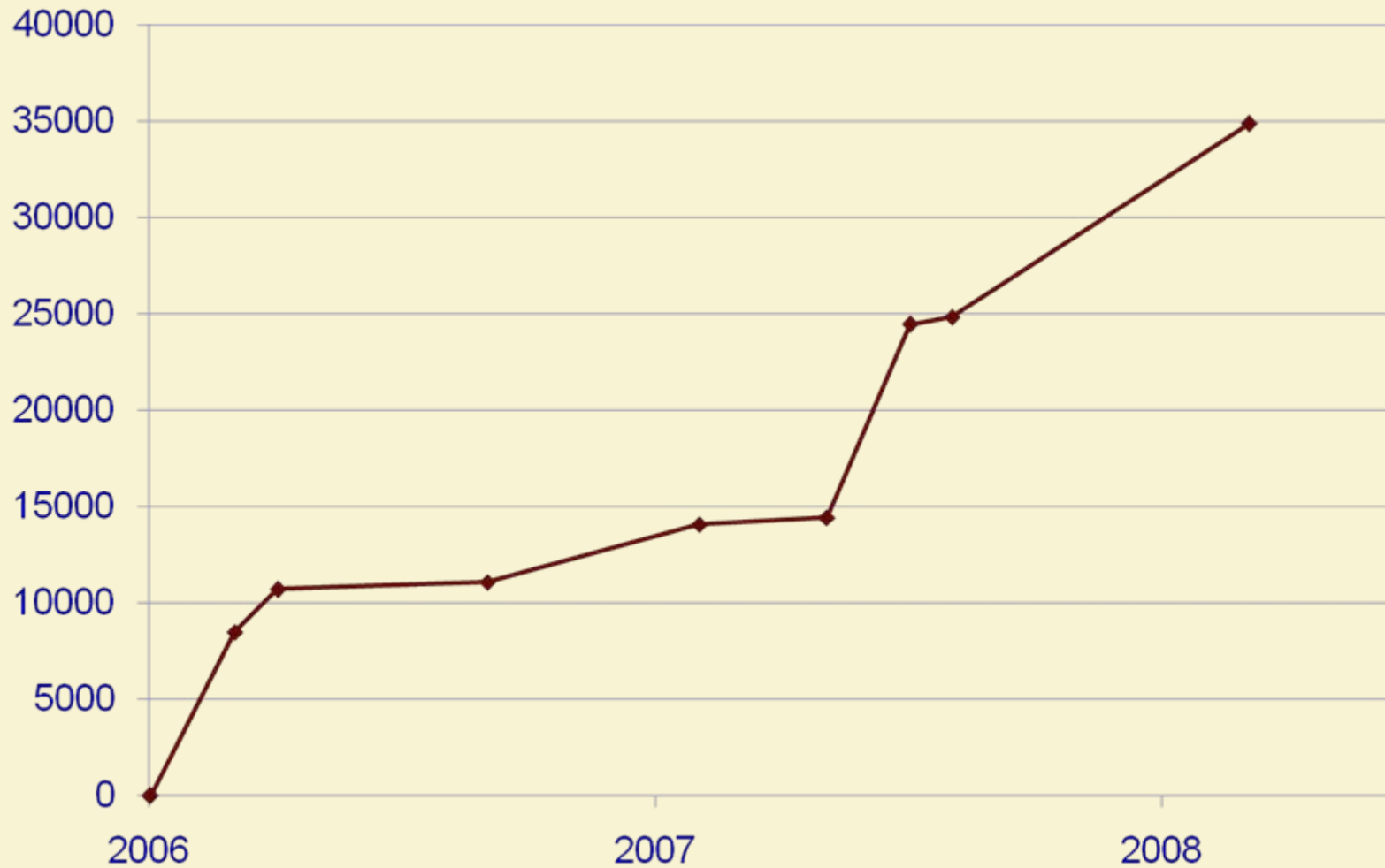
Where are we?

# Progress so far...

- Repository architecture
- Content models
  - Simple image, paged, video, multi-copy, audio
  - Content model standardization
- Basic tools
- Policy development
- Collections
  - Slocum Puzzles
  - Hohenberger
  - U.S. Steel
  - Hoagy Carmichael
  - New Harmony Correspondence



# Objects in repository



# Work in progress

- IN Harmony
  - Ingest
  - Interface development
- Sound Directions
  - Ingesting exchange packages
- Search enhancements
  - Fulltext search (XTF)
  - Faceted search
- Ingest enhancements
  - Validation (Xsubmit, content models)
  - Configurability
- Photo cataloging tool

# Work to be done

- Continue ingesting image-based collections
- Ingest text collections
- Better MDSS integration
- Develop processes for audio/video collections
- Enhance search system
- Release tools back to the community
- End-user submission system
- Preservation integrity system

# Thank You!

- Infrastructure project wiki:
  - <http://wiki.dlib.indiana.edu/confluence/display/INF>
- Contact info:
  - Ryan Scherle [rscherle@indiana.edu](mailto:rscherle@indiana.edu)
  - Muzaffer Ozakca [mozakca@indiana.edu](mailto:mozakca@indiana.edu)