Searching Digital Content via SRU

Ryan Scherle

Randall Floyd

October 25, 2006

Overview

- What is SRU, and why do we need it?
- SRU in practice
- SRU in the DLP
- Search and browse interfaces to SRU
- Going forward…

What is SRU?

- Search/Retrieve via URL
- The game of the name:
  - ZNG, ZiNG
  - SRW/U
  - SRU
  - SRU POST
  - SRU over SOAP
  - CQL

Why standardize search?

- Makes locating content easier for end users
- Increases visibility of existing digital content
- Eases implementation of new systems, features
- Search engines that follow standards can be combined in many ways

What about….

- Z39.50:
  - Complicated (eighteen native and extended services vs. one)
  - Requires connection-based sessions
  - Binary encoding transmitted directly over TCP/IP
- OAI-PMH:
  - Harvesting vs. searching
  - Focused on metadata only, not text
  - Can be used in conjunction with SRU
- Google, OpenSearch:
  - Very simple query syntax

Benefits of SRU

- Web services simplify communication
- Builds on the experience of the Z39.50 community, but sheds Z39.50's complexity
- Allows for any type of underlying search engine:
  - Simple grep applied against a text file
  - Direct access to a relational database
  - An index created from search engine software, such as Swish-e or Lucene

Benefits of SRU

- Web services simplify communication
- Builds on the experience of the Z39.50 community, but sheds Z39.50's complexity
- Allows for any type of underlying search engine:
  - Simple grep applied against a text file
  - Direct access to a relational database
  - An index created from search engine software, such as Swish-e or Lucene

Benefits of SRU

- Many institutions are implementing SRU:
  - Library of Congress
  - OCLC
  - Index Data
  - DSpace
  - Fedora
- There are SRU tools in many programming languages:
  - Perl
  - Python
  - C
  - Java
  - Ruby

Benefits of SRU

- Many institutions are implementing SRU:
  - Library of Congress
  - OCLC
  - Index Data
  - DSpace
  - Fedora
- There are SRU tools in many programming languages:
  - Perl
  - Python
  - C
  - Java
  - Ruby

Operations

- Explain – learn about the contents of the search system, available result formats
- Scan – learn about the usage of terms within the search system
- SearchRetrieve – retrieve documents that match a given query

Explain

- Describes the SRU Server
- Explain Response structure: Zeerex
  - Z39.50 explain, explained and re-engineered in XML
  - Basic administrative information
  - Fields to search
  - Result types
- Sample Explain record

Scan

- Allows listing of terms and their frequencies in the index
- Can be useful for browsing, zero results
- Could display a controlled vocabulary for a given collection
- Sometimes difficult to implement, but not required
- Sample Scan result

SearchRetrieve

- Submit a query, receive a response
- Any XML return type is allowed
- Result sets last for “a while”
- The simplest SRU client

SRU request parameters

- query
- startRecord
- maximumRecords
- recordSchema

SRU is simple, yet powerful

- Supports simple keyword queries, but allows much more complex queries to be specified.
- Complex queries may be formed by trained users or by task-specific interfaces.
- Default metadata type for returned records, but many types may be available.

Contextual Query Language

CQL has two goals:

- Be as simple as possible
  "Just do the right thing"
- Allow powerful searching

Some examples of
"CQL 1.2" adapted
from Ray Denenberg…

- cat

- cat

(That’s it. The whole query.)

Simple CQL Queries

- cat (simplest)
- cat and dog (simple boolean)
- title = cat (index)

Simple CQL Queries

- cat (simplest)
- cat and dog (simple boolean)
- title = cat (index)
- **dc.title = cat** (index qualified)

Qualified indexes

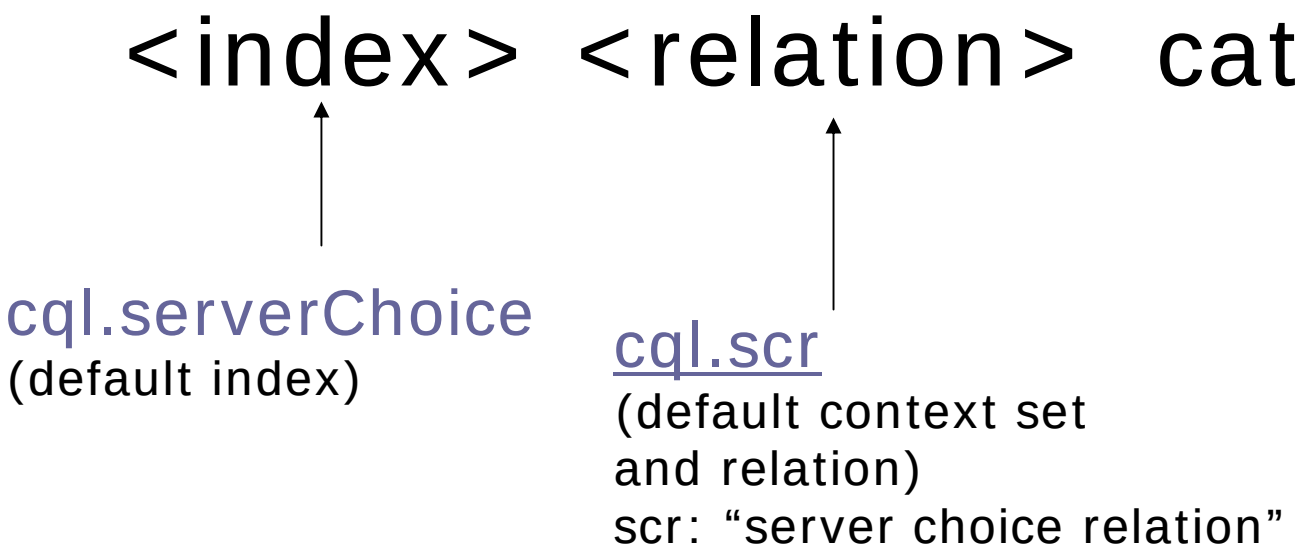
- title = cat
- dc.title = cat
- dc.title="Focus on grammar" AND bib.titleSub="basic level"

Relations

<index> *<**relation**>* *<search term>*

Indexes and relations come in pairs:

- cat
- title = cat


<index> <relation> cat
cql.serverChoice
(default index)
cql.scr
(default context set
and relation)
scr: “server choice relation”

About the = relation

= is the same as cql.scr

title = “the complete dinosaur”

means: find these three words, according to the server’s preferred equality metric (not necessarily as a phrase)

Other text Relations

- title = "the complete dinosaur"
- title all "complete dinosaur”
- title any "dinosaur bird reptile"
- title adj "the complete dinosaur"

All

title all "complete dinosaur“

matches
“the **complete** and unabridged **dinosaur**“

does not match
“the unabridged **dinosaur**“

All

title all "old man sea"

is the same as

title="old" and title="man" and title="sea"

Any

title any “dinosaur bird reptile”

does match

“the complete **dinosaur**" and

“the unabridged **dinosaur**"

Any

title any "old man sea"

is the same as

title="old" or title="man" or title="sea"

Adj

title adj "the complete dinosaur"

matches

"the complete dinosaur" and

“Follow the complete dinosaur home"

More relations

- **<** less
- **>** greater
- **<=** less or equal
- **>=** greater or equal
- **<>** not equal

Relation modifiers

- title =**/stem** "these completed dinosaurs“

  matches

- *The Complete Dinosaur*

Booleans

- **and**
- **or**
- **not**
- **prox**

  dc.title="cat" prox/distance=2 dc.title="hat"

Context sets

- CQL
- DC
- MARC
- Music
- Bib

  bib.namePersonal=/bib.role=author
  /bib.roleAuthority=marcrelator "George Orwell"

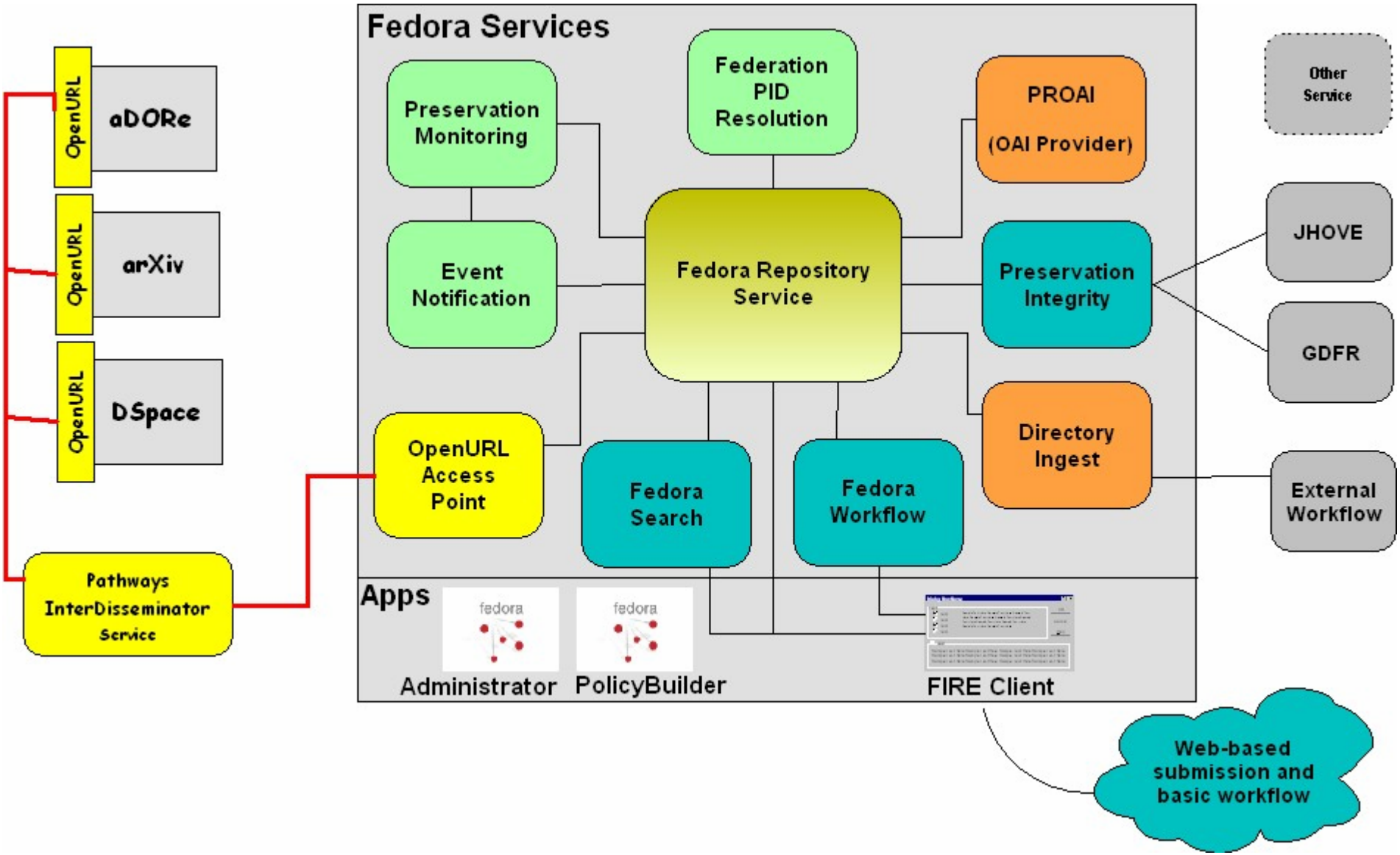
SRU around the world

- OAIster
- Library of Congress (dc, mods, marcxml)
- Rutgers Law Library
- UIUC OAI Registry
- The European Library and the Koninklijke Bibliotheek
- OCLC
  - Research publications
  - SRW registry

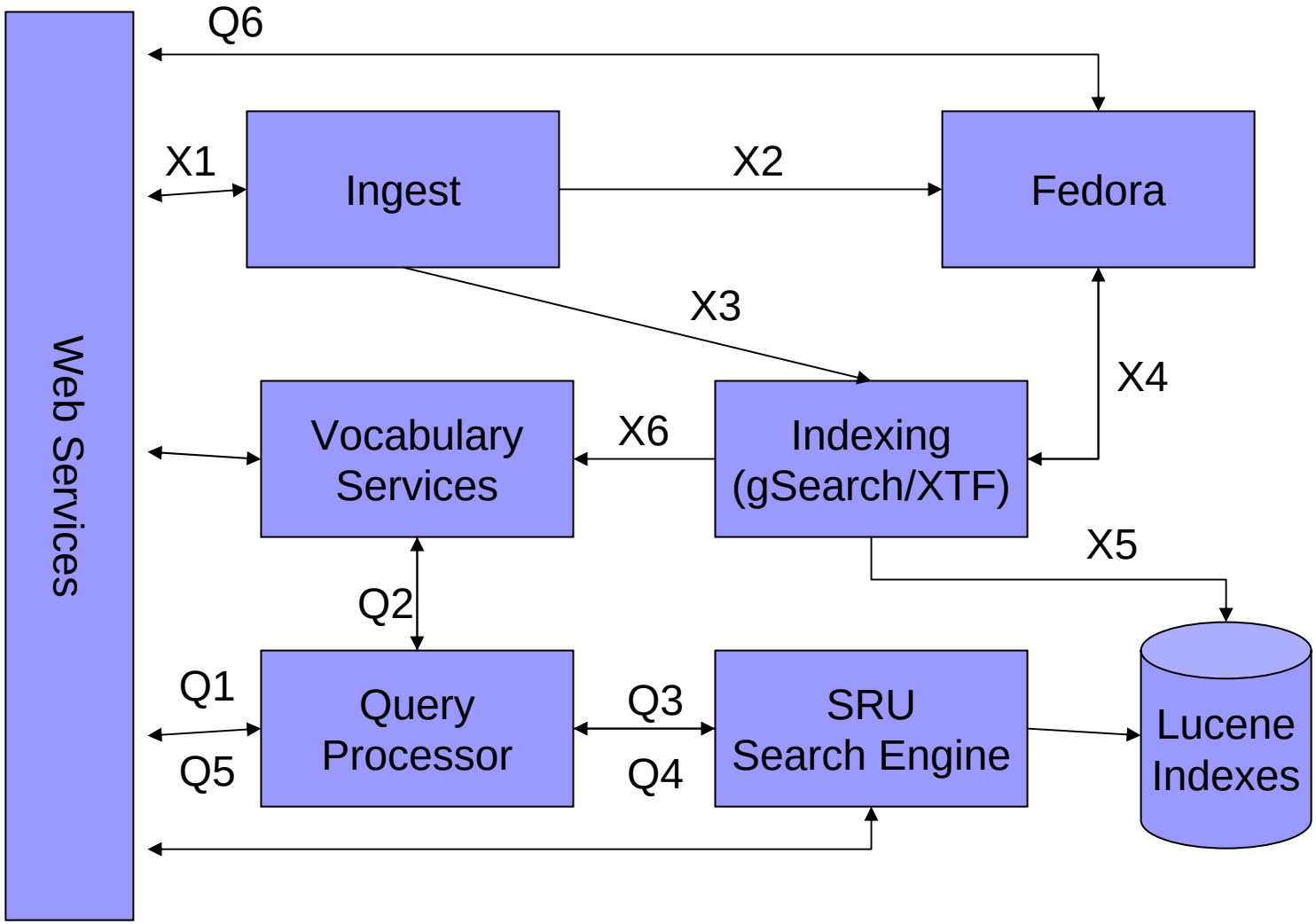
SRU in the DLP

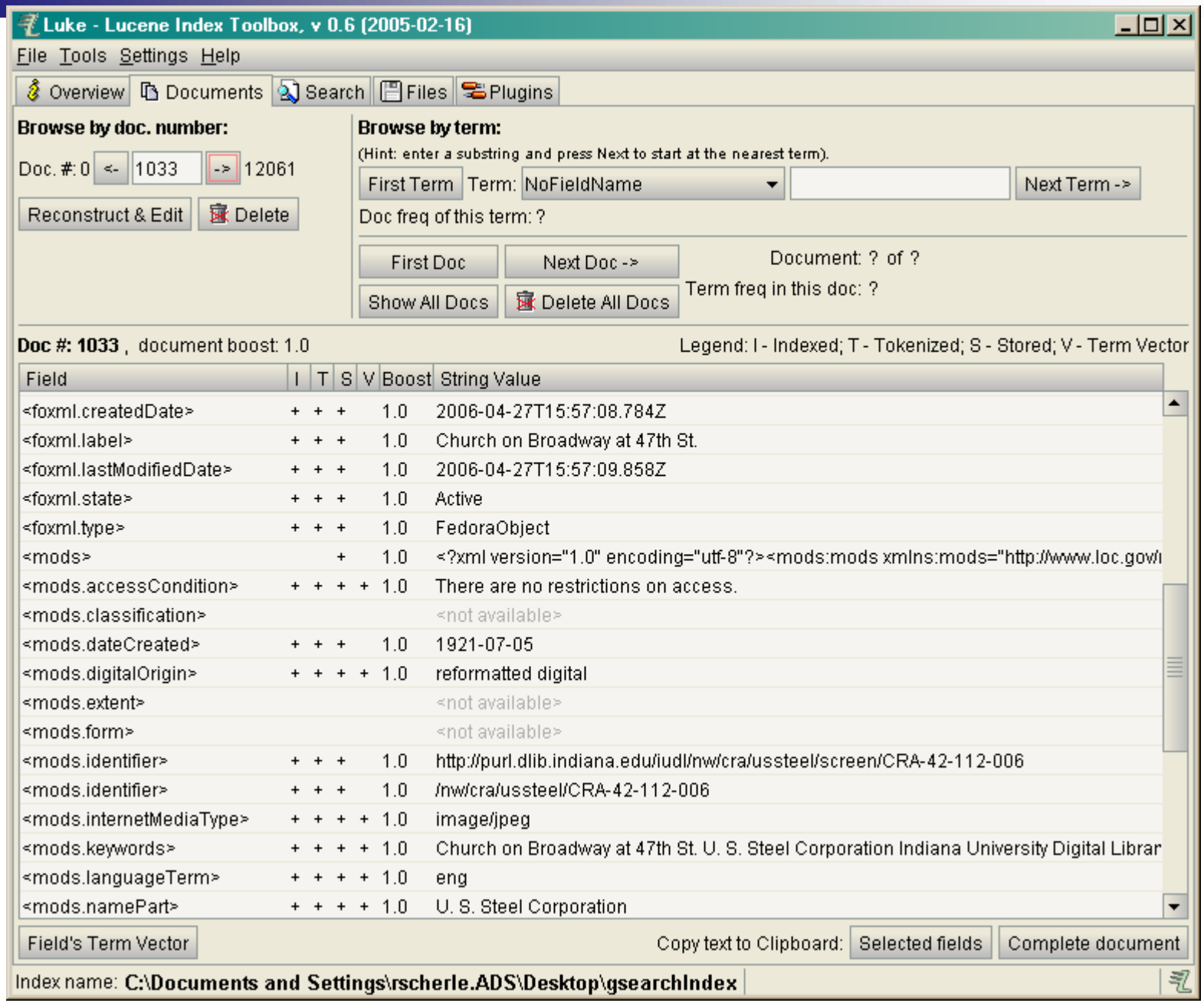
- New infrastructure based on fedora

- Indexing for SRU search
- Interaction between the search system and the index
- The other side of the world: SRU clients


Fedora Services
Preservation Monitoring
Federation PID Resolution
PROAI (OAI Provider)
Other Service
Event Notification
Fedora Repository Service
Preservation Integrity
JHOVE
GDFR
OpenURL Access Point
Fedora Search
Fedora Workflow
Directory Ingest
External Workflow
Apps
Administrator
PolicyBuilder
FIRE Client
OpenURL
aDORe
OpenURL
arXiv
OpenURL
DSpace
Pathways InterDisseminator Service
Web-based submission and basic workflow

Search system architecture




Luke - Lucene Index Toolbox, v 0.6 (2005-02-16)
File Tools Settings Help
Overview Documents Search Files Plugins
Browse by doc. number:
Doc. #: 0 <- 1033 -> 12061
Reconstruct & Edit Delete
Browse by term:
(Hint: enter a substring and press Next to start at the nearest term).
First Term Term: NoFieldName Next Term ->
Doc freq of this term: ?
First Doc Next Doc -> Document: ? of ?
Show All Docs Delete All Docs Term freq in this doc: ?
Doc #: 1033 , document boost: 1.0
Legend: I - Indexed; T - Tokenized; S - Stored; V - Term Vector
Field I T S V Boost String Value
<foxml.createdDate> + + + 1.0 2006-04-27T15:57:08.784Z
<foxml.label> + + + 1.0 Church on Broadway at 47th St.
<foxml.lastModifiedDate> + + + 1.0 2006-04-27T15:57:09.858Z
<foxml.state> + + + 1.0 Active
<foxml.type> + + + 1.0 FedoraObject
<mods> + 1.0 <?xml version="1.0" encoding="utf-8"?><mods:mods xmlns:mods="http://www.loc.gov/
<mods.accessCondition> + + + + 1.0 There are no restrictions on access.
<mods.classification> <not available>
<mods.dateCreated> + + + 1.0 1921-07-05
<mods.digitalOrigin> + + + + 1.0 reformatted digital
<mods.extent> <not available>
<mods.form> <not available>
<mods.identifier> + + + 1.0 http://purl.dlib.indiana.edu/iudl/nw/cra/ussteel/screen/CRA-42-112-006
<mods.identifier> + + + 1.0 /nw/cra/ussteel/CRA-42-112-006
<mods.internetMediaType> + + + + 1.0 image/jpeg
<mods.keywords> + + + + 1.0 Church on Broadway at 47th St. U. S. Steel Corporation Indiana University Digital Librar
<mods.languageTerm> + + + + 1.0 eng
<mods.namePart> + + + + 1.0 U. S. Steel Corporation
Field's Term Vector
Copy text to Clipboard: Selected fields Complete document
Index name: C:\Documents and Settings\rscherle.ADS\Desktop\gsearchIndex

The DLP SRU server

The DLP SRU server is based on an implementation by OCLC:

- Modular, can accommodate many databases of differing types
- Supports both REST (SRU) and SOAP (SRW) interactions
- Extended to support arbitrary Lucene indexes

Configuration:

- cql.serverChoice = mods.keywords
- dc.title = dc.title
- dc.creator = mods.namePart
- dc.subject = mods.subject
- bib.classification = mods.classification
- iudl.collection = collectionID

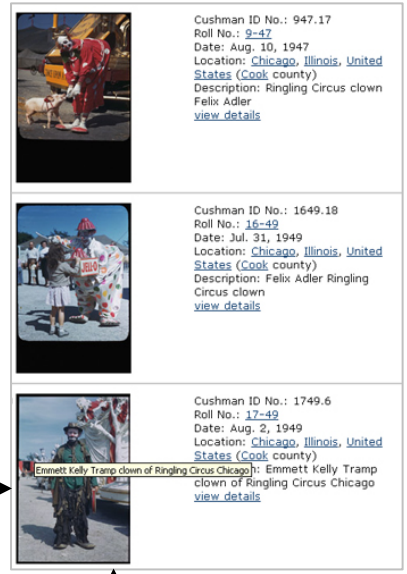
SRU in the DLP

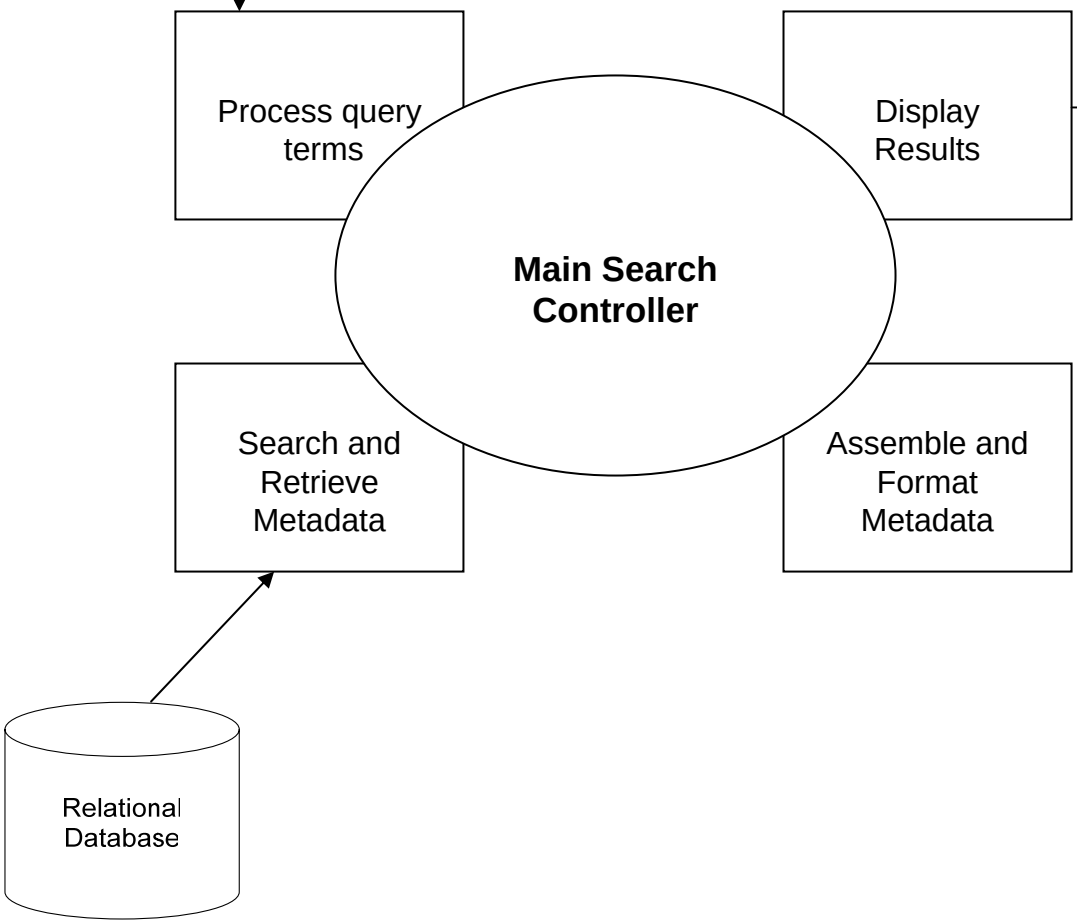
- Implementing SRU clients in search and browse interfaces

Search and Browse Interfaces

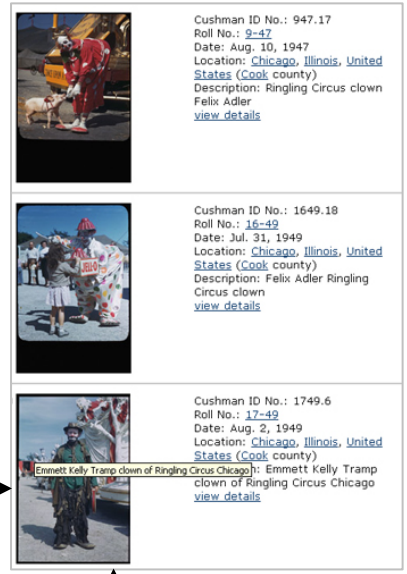
- Repository gives us a standardized way to store objects and metadata
- Still need a way to deliver collections online with minimal development
  - Past collection interfaces have been developed in a “one-off” manner

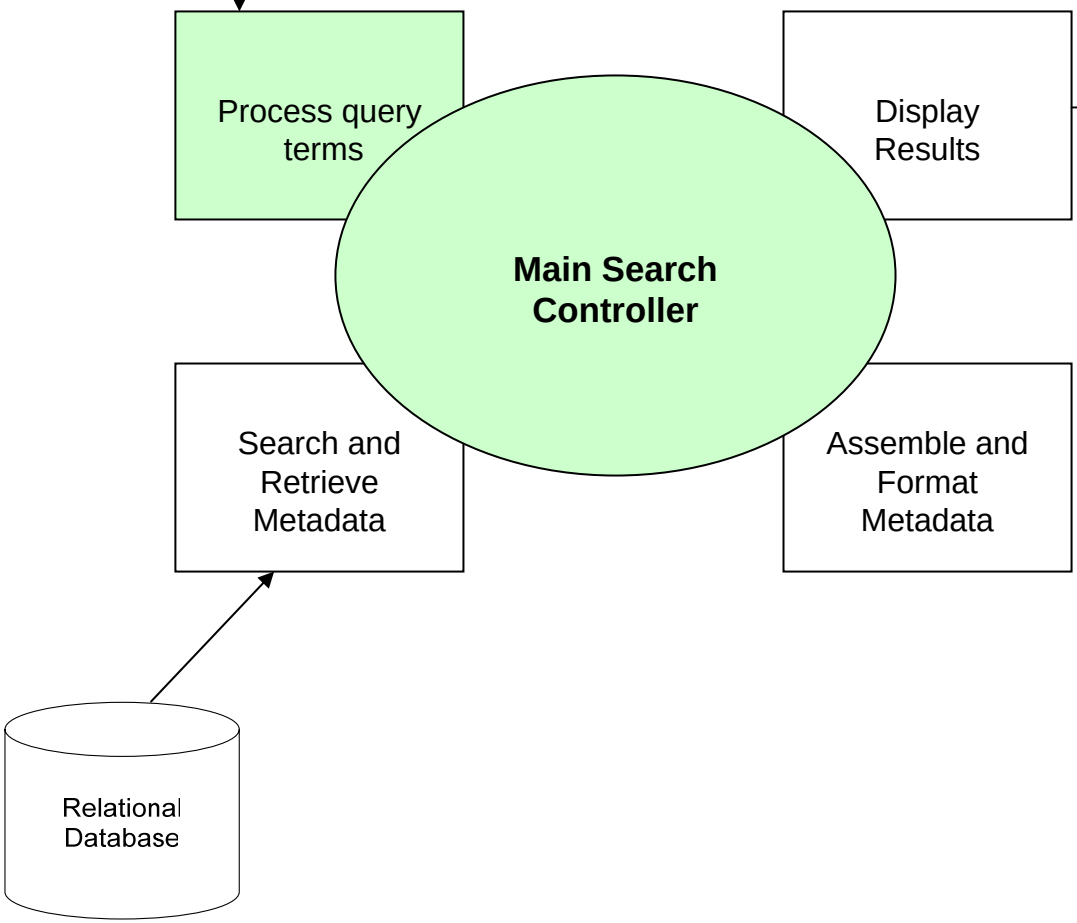

home ■ overview ■ browse ■ search ■ highlights ■ project info ■ site guide
Keyword Search
home >> keyword search |
Advanced Search
Search for: clown
search
clear


Cushman ID No.: 947.17
Roll No.: 9-47
Date: Aug. 10, 1947
Location: Chicago, Illinois, United States (Cook county)
Description: Ringling Circus clown Felix Adler
view details
Cushman ID No.: 1649.18
Roll No.: 16-49
Date: Jul. 31, 1949
Location: Chicago, Illinois, United States (Cook county)
Description: Felix Adler Ringling Circus clown
view details
Cushman ID No.: 1749.6
Roll No.: 17-49
Date: Aug. 2, 1949
Location: Chicago, Illinois, United States (Cook county)
Emmett Kelly Tramp clown of Ringling Circus Chicago
view details

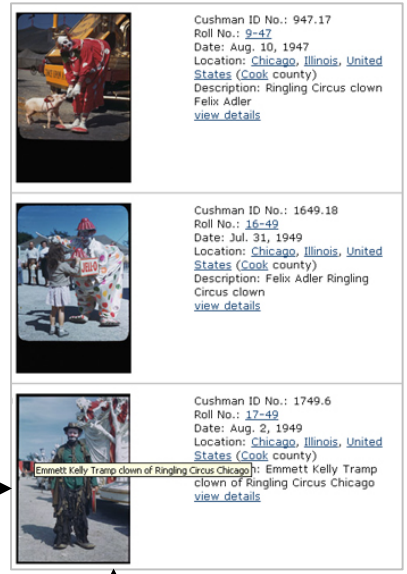

Process query terms
Display Results
Main Search Controller
Search and Retrieve Metadata
Assemble and Format Metadata
Relational Database
Image files on server

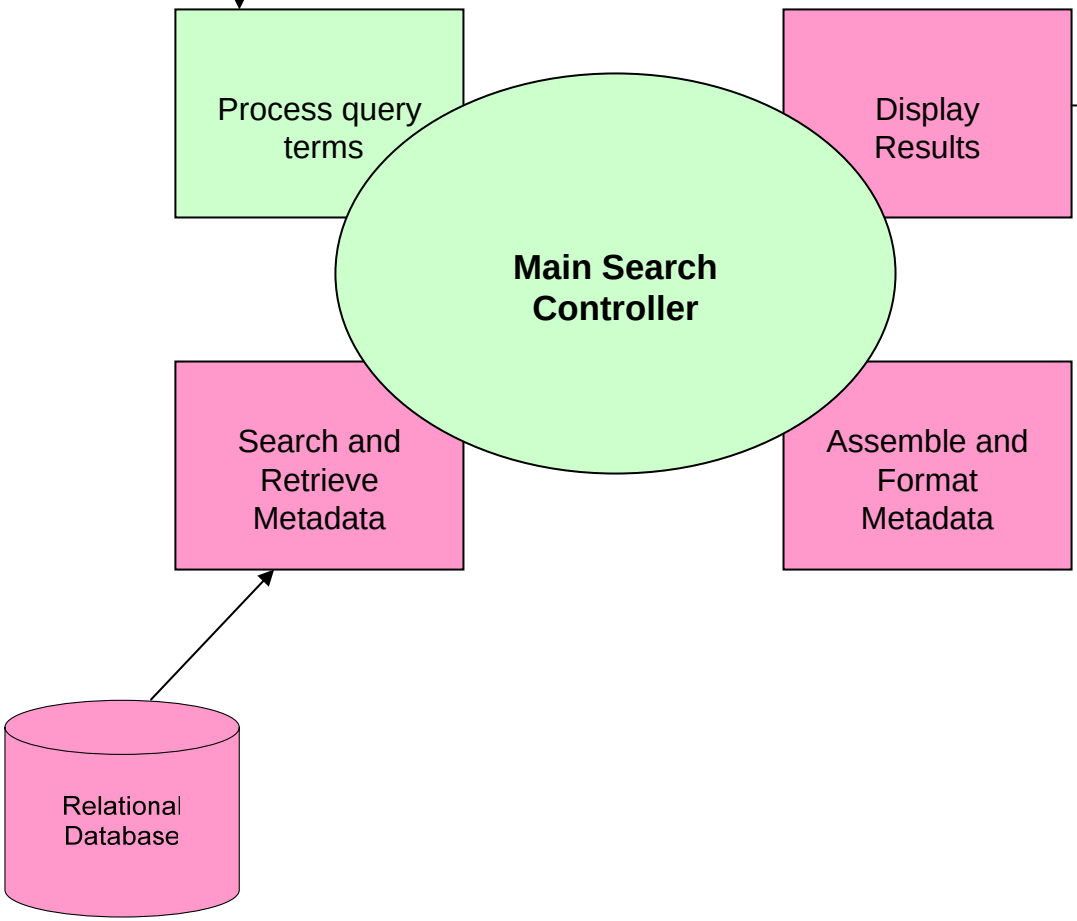

home ■ overview ■ browse ■ search ■ highlights ■ project info ■ site guide
Keyword Search
home >> keyword search |
Advanced Search
Search for: clown
search
clear

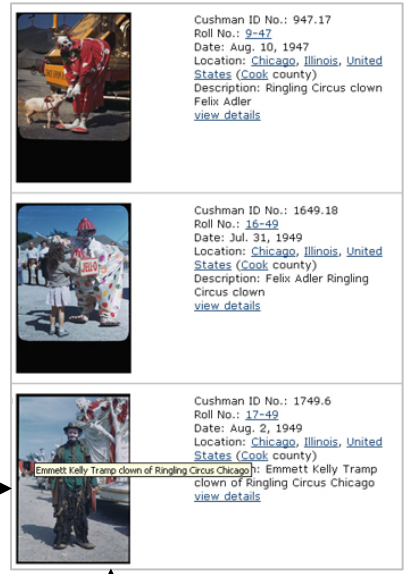

Cushman ID No.: 947.17
Roll No.: 9-47
Date: Aug. 10, 1947
Location: Chicago, Illinois, United States (Cook county)
Description: Ringling Circus clown Felix Adler
view details
Cushman ID No.: 1649.18
Roll No.: 16-49
Date: Jul. 31, 1949
Location: Chicago, Illinois, United States (Cook county)
Description: Felix Adler Ringling Circus clown
view details
Cushman ID No.: 1749.6
Roll No.: 17-49
Date: Aug. 2, 1949
Location: Chicago, Illinois, United States (Cook county)
Emmett Kelly Tramp clown of Ringling Circus Chicago
Description: Emmett Kelly Tramp clown of Ringling Circus Chicago
view details

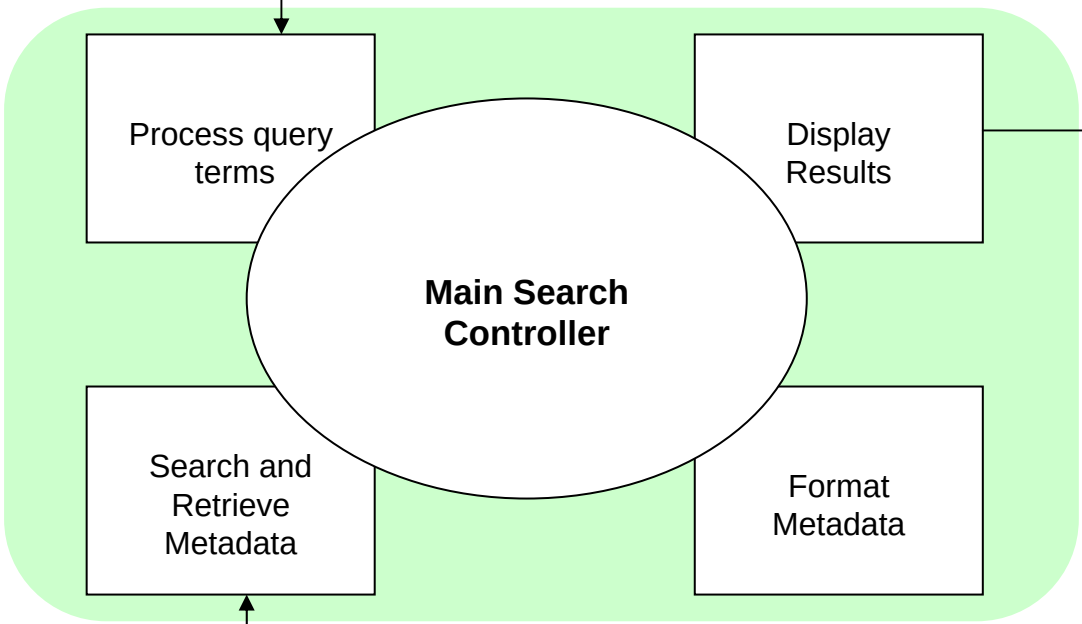

Process query terms
Display Results
Main Search Controller
Search and Retrieve Metadata
Assemble and Format Metadata
Relational Database
Image files on server


home ■ overview ■ browse ■ search ■ highlights ■ project info ■ site guide
Keyword Search
home >> keyword search |
Advanced Search
Search for: clown
search
clear

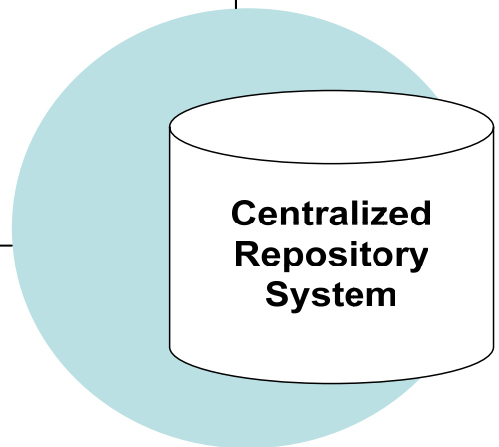

Cushman ID No.: 947.17
Roll No.: 9-47
Date: Aug. 10, 1947
Location: Chicago, Illinois, United States (Cook county)
Description: Ringling Circus clown Felix Adler
view details
Cushman ID No.: 1649.18
Roll No.: 16-49
Date: Jul. 31, 1949
Location: Chicago, Illinois, United States (Cook county)
Description: Felix Adler Ringling Circus clown
view details
Cushman ID No.: 1749.6
Roll No.: 17-49
Date: Aug. 2, 1949
Location: Chicago, Illinois, United States (Cook county)
Emmett Kelly Tramp clown of Ringling Circus Chicago
view details


Process query terms
Main Search Controller
Display Results
Search and Retrieve Metadata
Assemble and Format Metadata
Relational Database
Image files on server

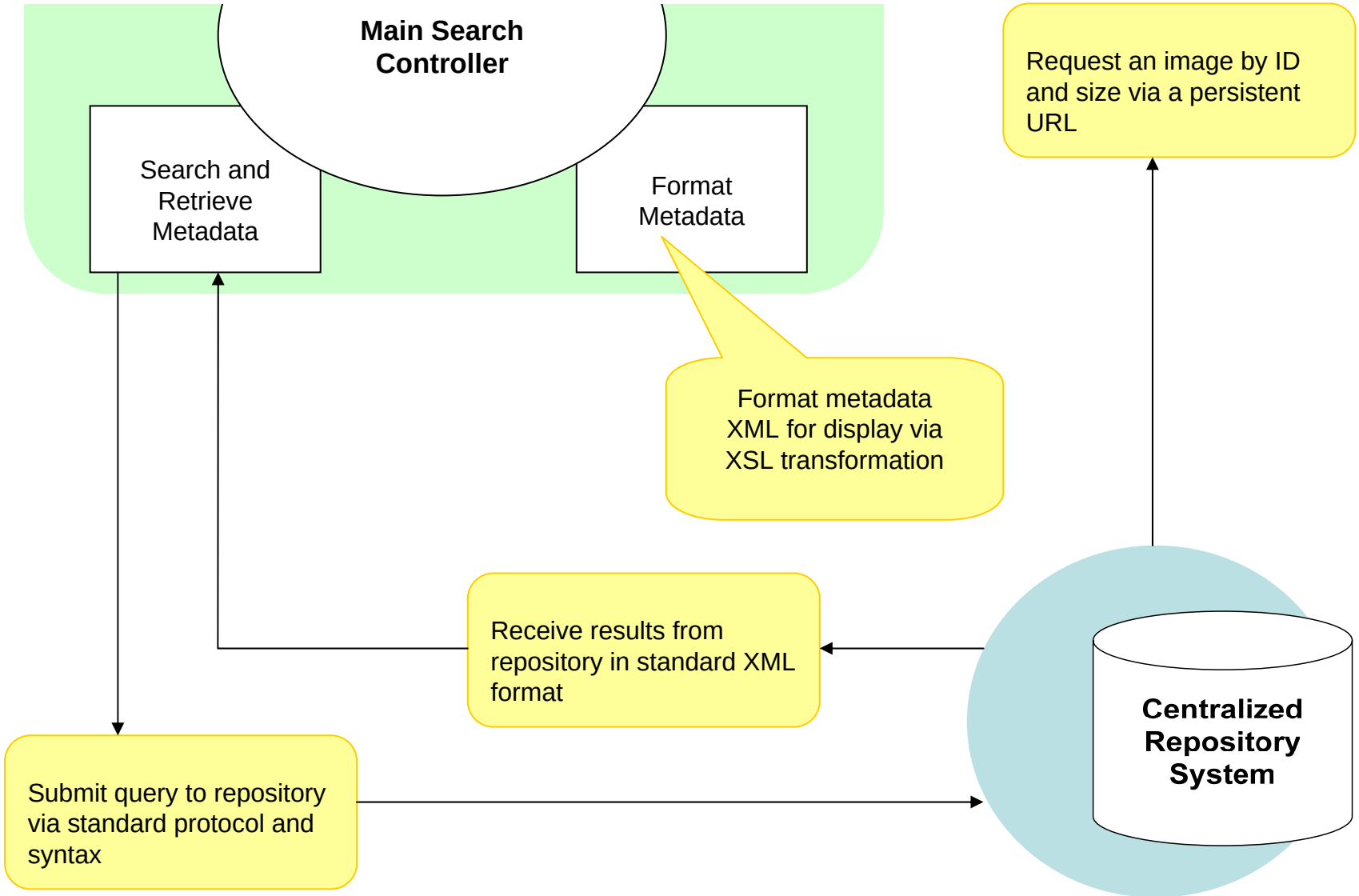


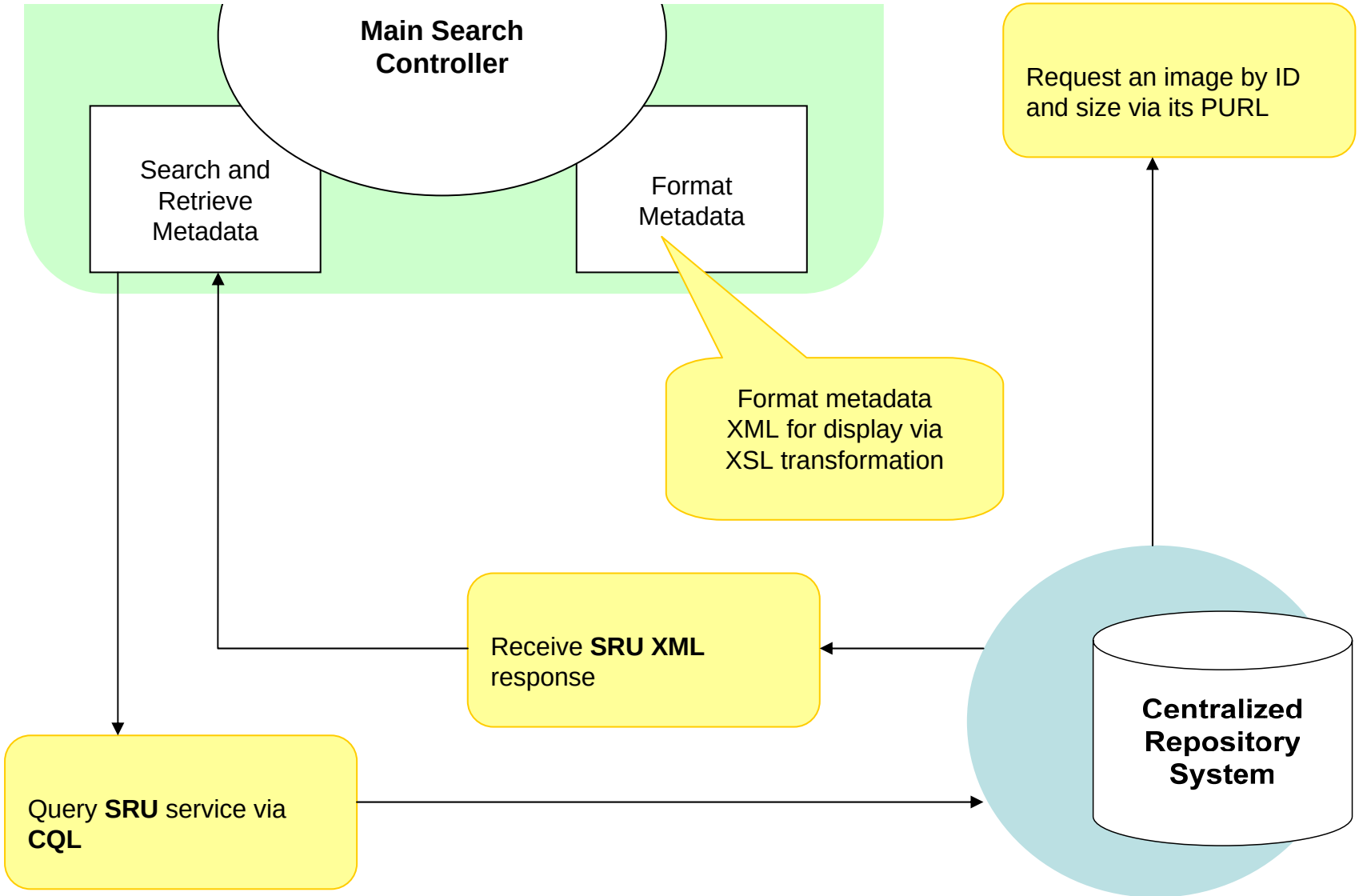


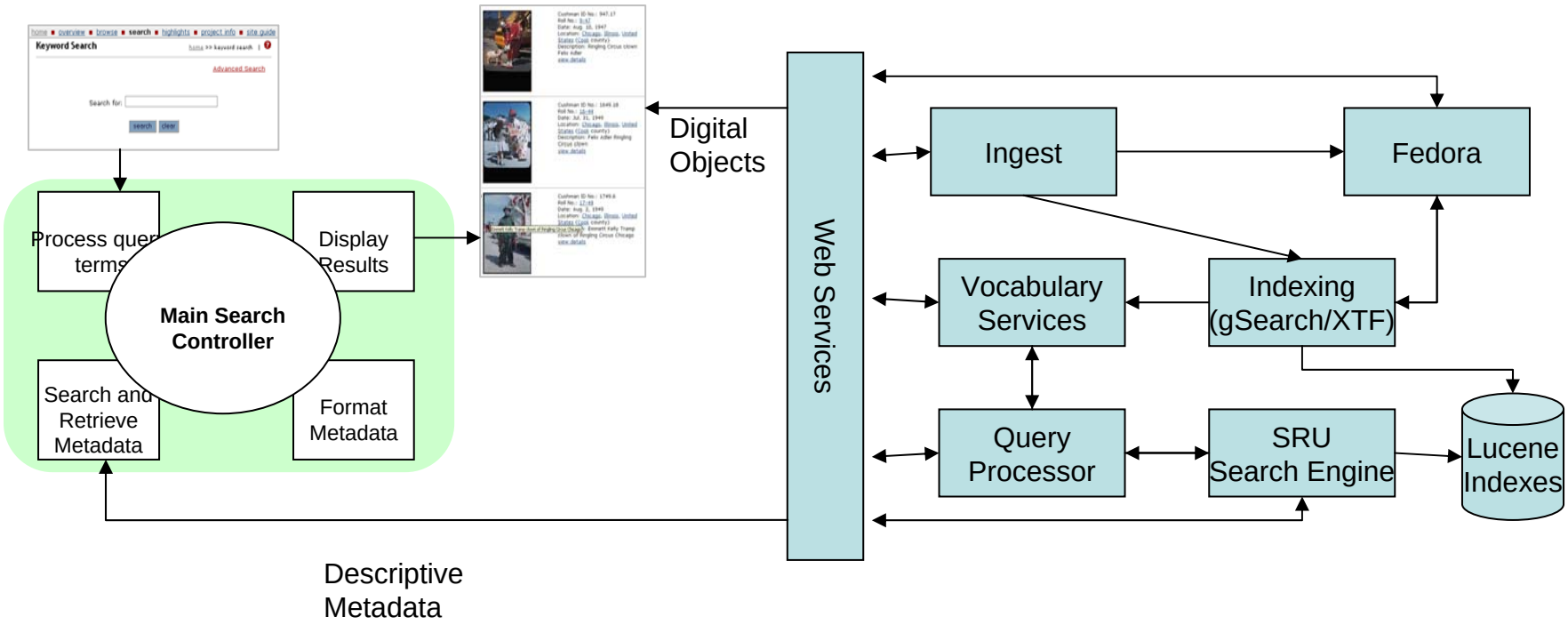
Digital Objects

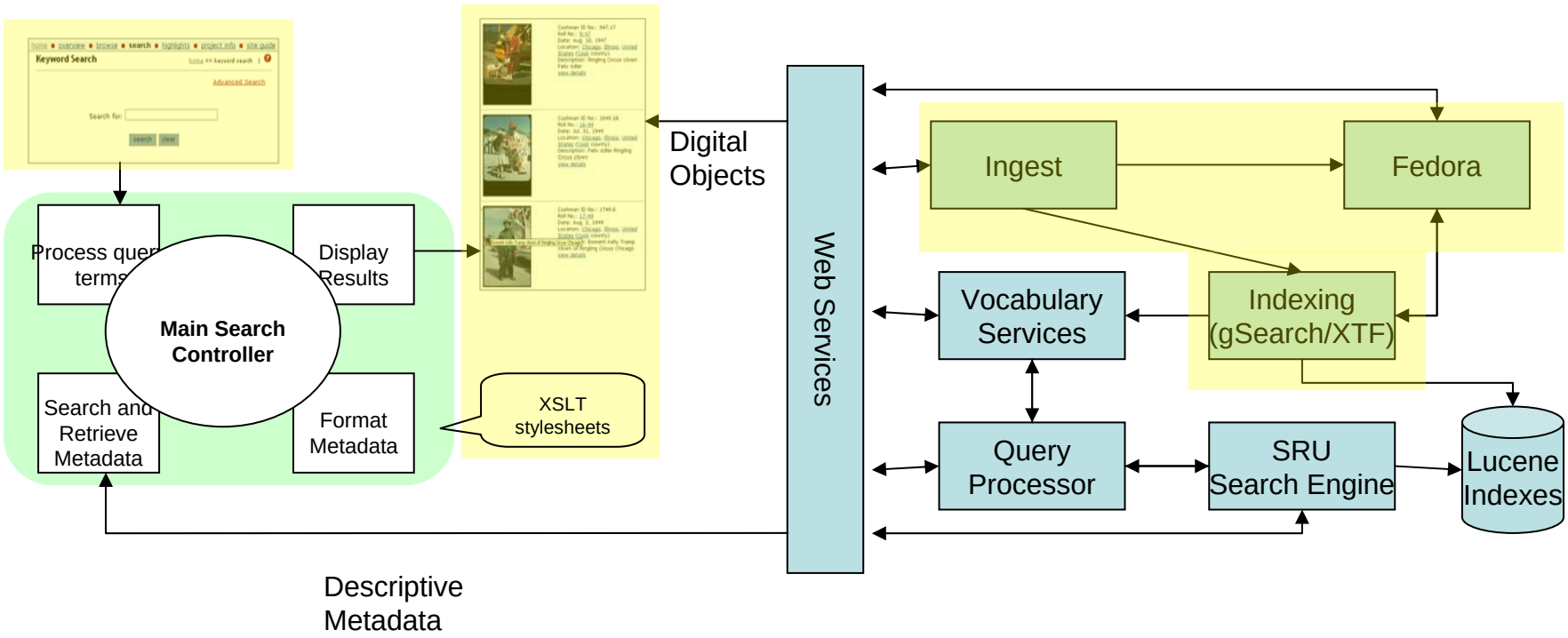


Descriptive Metadata


Main Search Controller
Search and Retrieve Metadata
Format Metadata
Request an image by ID and size via a persistent URL
Format metadata XML for display via XSL transformation
Receive results from repository in standard XML format
Centralized Repository System
Submit query to repository via standard protocol and syntax

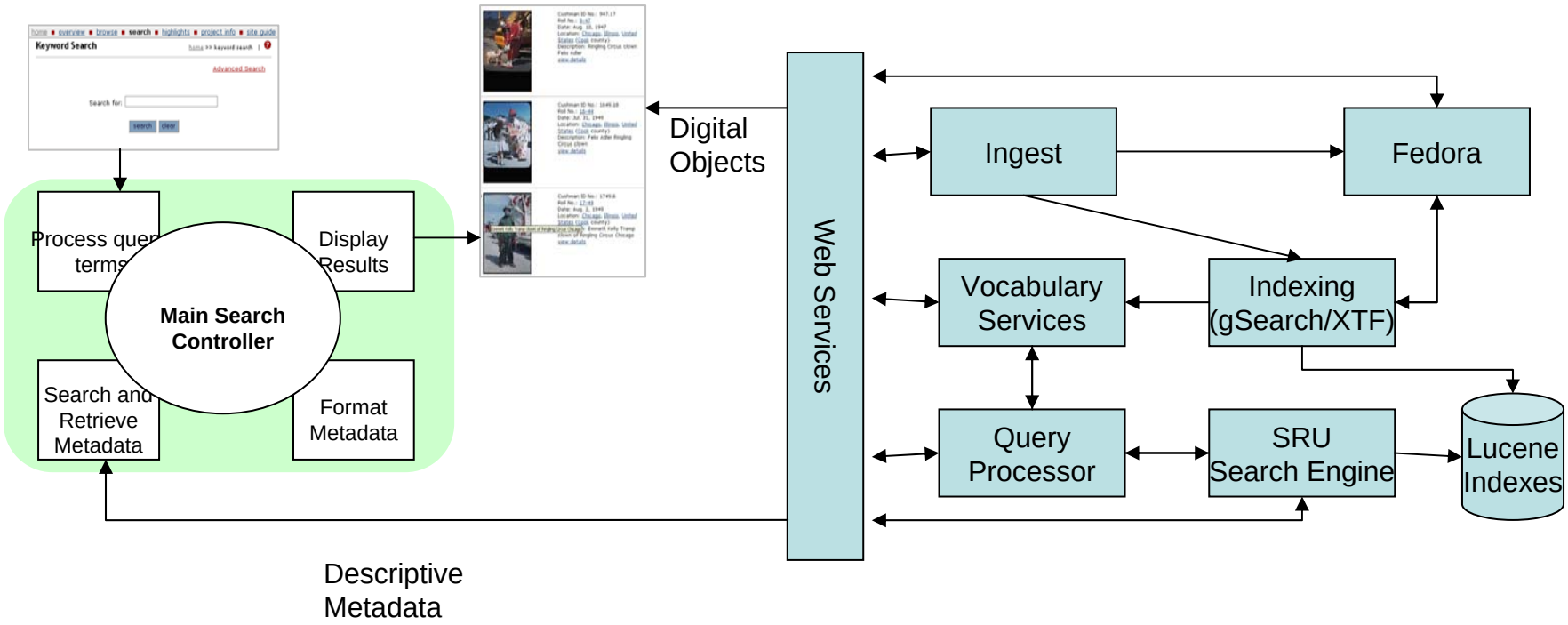

Main Search Controller
Search and Retrieve Metadata
Format Metadata
Request an image by ID and size via its PURL
Format metadata XML for display via XSL transformation
Receive SRU XML response
Centralized Repository System
Query SRU service via CQL


Keyword Search
Advanced Search
Search for:
search
clear
Process query terms
Main Search Controller
Display Results
Search and Retrieve Metadata
Format Metadata
Digital Objects
Web Services
Ingest
Fedora
Vocabulary Services
Indexing (gSearch/XTF)
Query Processor
SRU Search Engine
Lucene Indexes
Descriptive Metadata


Keyword Search
Advanced Search
Search for:
search
clear
Process query terms
Main Search Controller
Display Results
Search and Retrieve Metadata
Format Metadata
XSLT stylesheets
Digital Objects
Web Services
Ingest
Fedora
Vocabulary Services
Indexing (gSearch/XTF)
Query Processor
SRU Search Engine
Lucene Indexes
Descriptive Metadata

Sounds good in theory…

- Will it work in practice - will it keep us from having to build collection-specific search applications?
- As it turns out, we didn't have to wait long for an opportunity to find out.
- As usual, it started something like:
  - "By the way, we promised the Lilly Library that we would develop an online demo of the Slocum Puzzle collection to coincide with the grand opening of the Slocum Room"


Keyword Search
Advanced Search
Search for:
search
clear
Process query terms
Main Search Controller
Display Results
Search and Retrieve Metadata
Format Metadata
Digital Objects
Web Services
Ingest
Fedora
Vocabulary Services
Indexing (gSearch/XTF)
Query Processor
SRU Search Engine
Lucene Indexes
Descriptive Metadata

Delivering the Slocum Puzzle Collection

1. **Prepare the metadata**
   - Map Jerry Slocum's original Filemaker metadata fields to MODS
   - Export and convert the Filemaker data to MODS records

2. **Load the collection into Fedora**
   - Acquire preexisting digital images of puzzles
   - Load the MODS metadata and images into Fedora via the ingest system
   - Index the data using the gSearch system

Delivering the Slocum Puzzle Collection

3. **Develop a prototype for the collection interface**
   - Developed by Michelle Dalmau
   - Search application uses generic templates to render pages
   - Header, navigation, footers, etc. are separate pieces included as pages are rendered
     - Actual interface pages are reusable but still allows strong collection branding

4. **Develop XSLT stylesheets**
   - to transform the SRU XML response into HTML for the result and detail pages
   - Based on the prototype interface

The Jerry Slocum Mechanical Puzzle Collection

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

Order of events

1. Process search terms
2. Convert into CQL query
3. Invoke the SRU server with URL containing CQL query
4. Receive SRU response containing MODS records
5. Apply XSLT stylesheet to transform MODS records to formatted HTML
6. Assemble results page

Insert banner

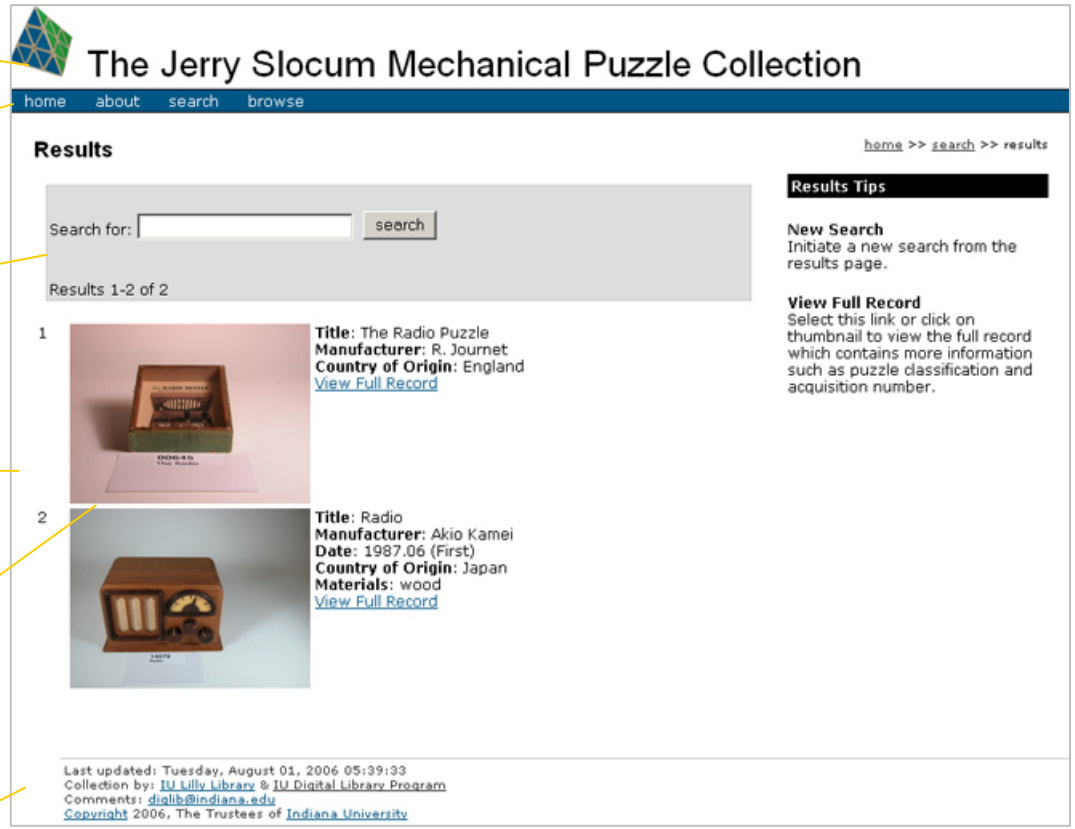
Insert navigation

Insert search controls

Insert formatted HTML from XSL transformation

Retrieve inline images via PURLs

Insert footer



Application log entries from search

```
Input terms: japan and wood collection:lilly/slocum

CQL query will be: cql.serverChoice="japan" AND
cql.serverChoice="wood" AND
iudl.collection="lilly/slocum"

About to contact SRU server with URL:

http://fedora.dlib.indiana.edu:8080/SRW/search/GSearch?query=cql.serverChoice%3D%22japan%22+AND+cql.serverChoice%3D%22wood%22+AND+iudl.collection%3D%22lilly%2Fslocum%22&startRecord=1

Our result total is: 45
```

Paging

- Pass startRecord parameter so that you start at any point you want
- Application handles calculation, then adds startRecord to SRU parameters
- Based on a page size of 20:
  - For page 2, set startRecord=21
  - For page 3, set startRecord=41

Going Forward…

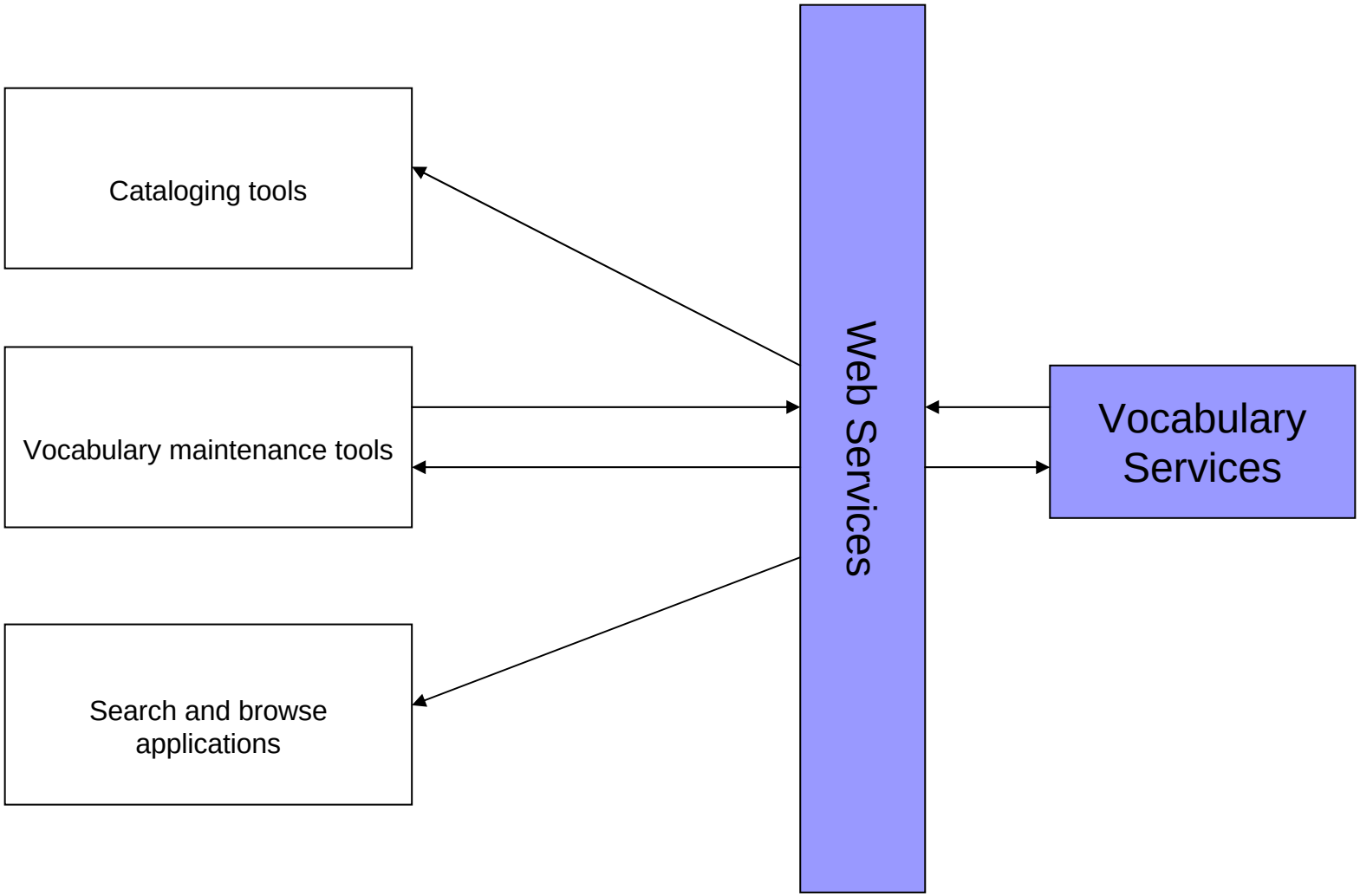
- Reusable application for *text* collections using SRU
- The eXtensible Text Framework (XTF)
  - Pioneered for use within the DLP by David Jiao and Tamara Lopez
  - Architecture that supports searching across collections of heterogeneous textual data, and the presentation of results and documents in a highly configurable manner.
  - Used to deliver the *Minutes of the Board of Trustees of IU* and *The Chymistry of Isaac Newton* collections.

Going Forward…

- DLP federated image search system
  - Need for a single interface to search all of DLP image content
  - The reusable search application easily adapted; it inherently works that way
  - Need to get all collection metadata into the repository and indexed
  - Need to create XSLT stylesheets to handle collection-specific display characteristics

Going Forward…

- Vocabulary and thesaurus services
  - Current search application does not include features to enhance search and browse via thesaurus relationships
    - A cornerstone feature of the *Charles W. Cushman Photograph* collection
  - Need a service to perform search term expansions
    - Lookup and include preferred term in search
    - Lookup and include narrower terms in search
  - Source of terms should be the same thesauri used for descriptive metadata


Cataloging tools
Vocabulary maintenance tools
Search and browse applications
Web Services
Vocabulary Services

Going Forward…

- Anticipated design of vocabulary and thesaurus service will be:
  - Vocabularies modeled in a relational database with close Z39.19 compliance
  - Resulting database indexed by Lucene for close integration to Fedora and SRU
  - Possibly even use SRU to query terms for expansion, using Zthes
    - An emerging thesaurus context set for CQL, with companion XML schema for vocabularies and relationships

Going Forward…

- Adapt existing collection interfaces to use new search system based on SRU
  - Each collection will present new challenges that will require new features throughout the system
- Work with OCLC to incorporate our changes to their SRU server
  - Biggest change was to allow SRU to use Lucene indexes versus their database-specific implementation

Going Forward…

- Integration with other search services
  - IUCAT, OneSearch @IU, WorldCat, Google
  - How do we allow external services to search and/or index us via our SRU server?
  - Conversely, how do we accept search input from external systems and translate to CQL/SRU queries?
    - For example, allowing Google search appliance at IU to interface with our search system -- “Gocqle”?

Comments or Questions?

Additional links

- The SRU standard
- DLP searching system documentation
- DLP SRU server documentation
- OCLC's SRU implementation