Architecture for Authority Lookup

Supports Samvera (aka Hydra) community linked data support for authority lookups with linked data.

NOTE: The Hydra Community recently rebranded as Samvera due to trademark issues with the name Hydra.

Technical Outputs

- Hyrax ld4l-qa-demo (code on github)
- Mockups for Lookup with Context
- Jena/Fuseki Caching Server
- Production authority lookup service at Cornell
    - connection validation
    - accuracy tests
    - monitoring
    - historical data
- Public access to authority lookup service code
- Authority configurations & documentation
    - AgroVoc
    - dbPedia
    - GeoNames
    - Getty (aat, tgn, ulan)
    - Library of Congress (LoC - authority names, subject headings, genres/form terms)
    - Medical Subject Headings (MeSH)
    - NAL Agricultural Thesaurus (NALT)
    - OCLC Fast
- Extension of Samvera Questioning Authority gem
    - min_context branch adding additional context to results

Presentations

- Linking the Data: Building Effective Authority and Identity Lookup

  presentation - SWIB (Dec 2017) (video)

- Efficient and Reliable Access to Authorities with Contextual Lookup

  presentation - DLF (Oct 2017)
- Linked Data in Hyrax

  lightning talk - LDCX (Apr 2017)
- Using Questioning Authority Gem to access Linked Data for Hydra and Beyond

  session - LDCX (Apr 2017)
- Questioning Authority Gem and Linked Data

  bi-annual meeting LD4L Labs (Nov 2016)
- Caching Refresh for Active Triples Linked Data Fragments

  bi-annual meeting LD4L Labs (Nov 2016)

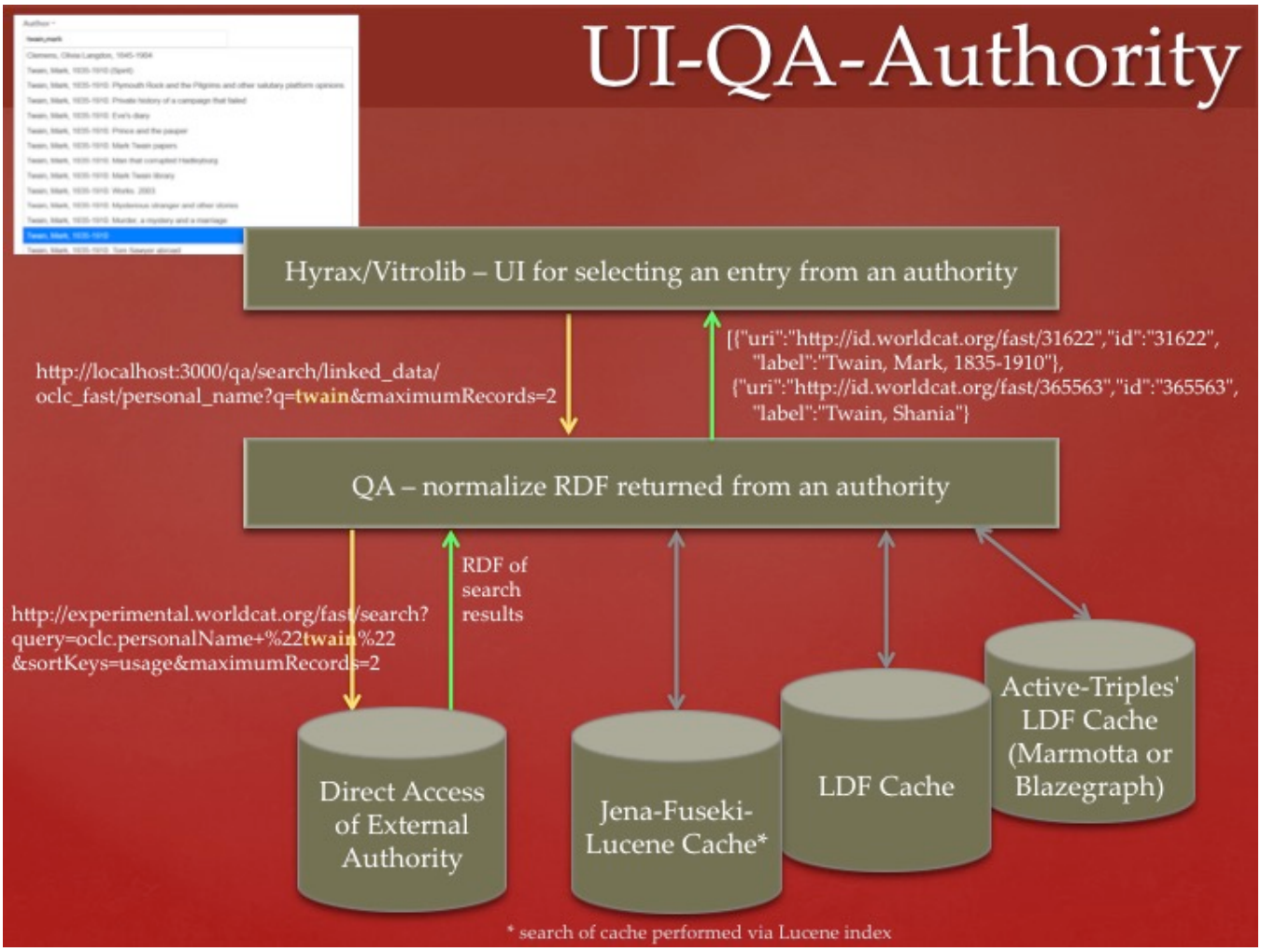
Table of Contents

- Overview
- Authorities
- Questioning Authority (QA) normalization layer
- Application UI
  - Getting Linked Data
  - Leverage Linked Data

Overview

Efforts in this area are targeting the Samvera development community to increase the ease of adoption of linked data. The tools being created by this work integrate with the Samvera stack (ruby based app). Several tools are standalone with an access API that allows them to be used in other development stacks which has been demonstrated with VitroLib (java based app) and the Hip Hop Collection visualizations (javascript based app).

The following diagram shows the architectural overview of the relationship between the Authorities (e.g. Library of Congress, OCLC Fast, dbPedia, etc.), the Questioning Authority (QA) normalization layer, and the display of content for selection in an application's UI.



Notes:

- The UI layer shows using either Vitrolib or Hyrax. Other applications could also be integrated with this stack.
- In this example, Hyrax/Vitrolib UI makes a query request through QA which is configured to make a query request against OCLC FAST directly. OCLC FAST returns RDF search results which QA normalizes to JSON with URI, ID, and Label included for each search result. Hyrax or Vitrolib uses the normalized JSON to back an autocomplete field.
- The example shows Hyrax/Vitrolib making a query request and receiving multiple results. QA also supports fetching additional information for a single term.
- QA could have been configured to go against a number of external authorities directly or against a server cache. Configurations exist for OCLC FAST, Agrovoc, and Geonames for query search and term fetch. LoC is configured for term fetch only.
- Three potential server cache technologies are included in the diagram. As a proof of concept, a test app was configured to access the Jena-Fuseki-Lucene cache server. The others have not been tested.
- Future work in QA will normalize RDF in an expanded JSON format that will include additional context values that can be used by the UI layer for lookup with context.

- All cache technologies can be supported without changes to QA if they meet these assumptions:
    1. Server (e.g. external authority or cache server) provides an API supporting two types of calls…
        a. QUERY: q=any full or partial string value
        b. FETCH: id or URI=id_or_uri_to_a_single_term
    2. Server returns RDF in just about any RDF serialization
- Supporting multiple requests to an LDF server requires modifications to QA. (Potential Future Work)

Authorities

Direct access to authorities is subject to outages and service delays that are outside the control of the library. To address stability of authorities, we are exploring several backend caching schemes. These include work at University of Iowa into caching authority data in a Jena-Fuseki Triplestore with a Lucene search layer for querying. Harvard has explored a http://linkeddatafragments.org linked data fragments server implementation. Cornell investigated the Samvera community's Active-Triples linked data fragments implementation which uses Marmotta or Blazegraph as the backing triplestore.

We are moving forward with extending the Jena-Fuseki-Lucene approach for caching external authority data. This approach will serve as the back-end for VitroLib installations for cataloging experimentation and Hyrax repository applications. The QA and UI components being developed will also be able to work with the other back-end approaches, but we will not be exploring them further at this time and they may require additional work to be fully functioning.

Questioning Authority (QA) normalization layer

The original Questioning Authority (QA) code used non-linked data APIs to access authorities. This required coding of a one-off process for each authority to access the API and interpret the proprietary results format. We extended the QA code to include a single process for accessing linked data based APIs and interpreting the linked data results. There are still differences in the access APIs and results can be returned in a number of ontologies. These differences are encoded in a configuration for each authority which includes the access URLs and identification of the predicates serving particular roles in the UI. The results are normalized into a JSON structure allowing front-end UIs to process all data the same.

Application UI

The UI experimentation looks at how to get linked data into an application and how to leverage linked data once it is in.

Getting Linked Data

The primary challenge that we are addressing for getting linked data into an application is through a selection process from an authority. The methods of selection we have explored are:

- **Autocomplete** - The user begins typing in a form field. As they type, a list drops below the box showing potential matches. The user can select from the list. The list is driven by data accessed from an authority and normalized by QA for use in an autocomplete list. These work has been successfully demonstrated in a Hyrax application and in VitroLib.
- **Lookup with Context** - The user chooses to do a lookup for a form field. The user types a search query that accesses an authority through a normalization process in QA. QA returns matching terms and values for predetermined context fields for the matching terms. The user is then able to select one or more terms using the additional context to aid in their decision making process. The additional context is expected to enable users to make more accurate selections thus improving overall data quality. This work is in the early design stages. See mockups for current design.

In both cases, the URI for the selected term is stored by the application. In some cases, the label of the selected term is also stored. For Hyrax applications, the URI is stored with the repository object and the label is stored in the solr index to allow for search results matching in the application.

Leverage Linked Data

Linked data can be leveraged in several ways.

- Fetch current label for a URI from the authority or a cache of the authority. (Demonstrated in Hyrax and VitroLib)
- Include additional information for a term from the same authority. (Demonstrated in Hyrax)
- Include additional information for a term from additional authorities. (Demonstrated in Hyrax and Hip Hop Collections visualizations)
- Allow users to follow linked data to explore the deeper graph. (Potential Future Work)

See Linked Data in Hyrax LDCX presentation for visual representations of the Hyrax app implementation leveraging data. The longer Using Questioning Authority Gem to access Linked Data for Hydra and Beyond presentation also includes screen shots from VitroLib and Hip Hop Collections.