Applied Linked Data Call 2015-04-29

Attendees:
- Steven Anderson (Boston Public Library)
- Trey Terrel (Oregon State University)
- Corey Harper (New York University)
- Andy Weidner (University of Houston)
- Justin Coyne (Data Curation Experts)

New Meeting Time:
- 9am Pacific, 12 Eastern -- Thursdays (Every other -- Still Off-Weeks of Hydra Metadata WG)

Caching Discussion
- Sidecar. Does LDF apply to this.
  - Oregon Digital uses MongoDB.
  - Justin uses Marmotta
- How to Cache?
  - Marmotta Option: Built-in Caching Logic
  - LDF Server as Vocab Repo. Processes Triple Pattern Frag
  - Question of how to do Cache Invalidation. Current approach just refreshes after 30 days.
  - Linked Data Fragments option would still have to require Marmotta or MongoDB or some other caching mechanism behind it.
    - Does allow a place to put configuration for the caching though.
    - Does make it easier to swap out the caching implementation.
    - Question on if we need to implement all of a Linked Data Fragments interface. We may only care about it being given a subject rather than supporting resolution of all parts of the triple.
    - Oregon Digital also needs geo-lookup (return Lat/Long) beyond just labels.
  - Mention of Stanbol but unsure exactly how it works. Previously sent link on details: https://stanbol.apache.org/docs/trunk/customvocabulary.html (Amherst has implemented it)

Timelines for a Linked Data Fragments Sprint
- June 8th - June 19th (conflicts with Open Repositories though)
- June 15th - 26th (conflicts with one of the members being on vacation for the 2nd week).
- Main advantages of this work for our applications: easier configuration of caching invalidation rules and switching out the caching backend.

Indexing Problem
- Local Solr Reflections of what’s in your cache
- Current: On save, retrieves asset from cache & save to solr
- Option 1: If you find a linked data element has changed, find the different objects with that reference & reindex
  - Done via Resque/Redis background reindex jobs
  - Slow!
- Option 2: intermediary Solr. Layer in between main solr and application that handles just the linked data (ie. resolves labels and the sticks them into the main solr response).
  - Down side, now you have another proxy thing.
  - Change still requires reindex, but maybe w/ atomic updates
- Requires stored fields: https://wiki.apache.org/solr/Atomic_Updates#Stored_Values
- Option 3: Sidecar Indexer
  - Application logic for reindexing happens outside of SOLR / Hydra.
  - Occasionally polls Solr for out of date objects and updates their reference (unlike option 1 that schedules jobs when an out of date reference is found).
  - Much easier to develop a sharable separate application that that others could then use than trying to make Option 1 reusable.
  - Does mean one has yet one more application running in addition to your Hydra Head / Linked Data Fragments Server / Caching mechanism...
  - Would require stored fields and atomic updates.
  - Leaning towards doing option 3 in the future. Would like feedback on other thoughts for handling this!

Alt Labels Searching and broader / narrower SKOS concepts
- For all labels, could just pull it with the normal label into a single multi-valued solr field. Then could return results from "Boston" based on a search of the alternate label “Beantown”.
- For broader / narrower, would be cool to be able to get those on a search. Not sure on the implementation and pushed off to the next call.

Stored Field default in Hydra
• Open an issue in Active-Fedora or Hydra-Jetty (likely Active Fedora).

• Keywords that don't use storable in Hydra indexing: Facetable and Searchable. Would want to remove these from the Solr config rather than change them.

• There is a stored searchable one that will do it as a stored field. Would need to check to ensure there is a stored facetable option (may need to have it added). Seems like it may be equivalent to :symbol.

• Issue created at: https://github.com/projecthydra/active_fedora/issues/778