Agenda:

1. Check in on LDCX action items ([https://docs.google.com/document/d/1yr7dlIKzWmaVLf-2easbaQvYSUF8f0_ylr8RlkWprY/edit](https://docs.google.com/document/d/1yr7dlIKzWmaVLf-2easbaQvYSUF8f0_ylr8RlkWprY/edit)):
   a. Get DGE/Tom’s LDF branch of QA merged
      i. The google doc probably misrecorded this as a QA branch; actually a branch on LDF.
   b. For LDF gem, Tom would advocate possibly axing rails engine. Wants buy-in for rack instead of rails. First step buy-in for this is Tom to create a PR on LDF gem.
   c. Next step for qa-ldef gem, lib/qa/ldef/authority.rb, Tom extracting some of separate concerns of a QA authority deals with - search, mapping between search responses - implement dependency injection to handle those.
      i. Anna putting this off for now; will check in again next time.
   d. QA roadmap: Looking at mapping of search responses, open issues that get to a QA either 1.1 or 2.0
      i. I think this is a restatement of the item above.
   e. Extending LDF to accept queries:
      i. Steven did push up the branch on which he started this work: "The branch is located at: [https://github.com/ActiveTriples/linked-data-fragments/tree/feature/ldf_query_experimental](https://github.com/ActiveTriples/linked-data-fragments/tree/feature/ldf_query_experimental) and has an updated README. I tested it again locally so let me know if there are problems if you try it. Note that it gave me an error in Ruby 2.4 for some reason so I switched back to my older Ruby 2.2.x version."
      ii. Anyone who is interested in this functionality is welcome to take a look and pick up this issue!
      iii. The LDCX action item is now complete (was simply to get this branch pushed, documented).

2. Summary of Action Items from Linked Data for Libraries bi-annual planning meeting (Lynette Rayle)
   a. Backend
      i. Jena-Fuseki as a cache with Lucene/Solr index for searching and potential mirroring
      ii. Explore ability to submit multiple LDF queries (°s° p° q°) to search and retrieve predicate values
      iii. Integration with ActiveTriples LDF implementation
      iv. Maintaining QA as usable in Hydra stack but not requiring Hydra stack
   b. Authorities
      i. Latest release of QA includes authority access requesting linked data normalized by QA
      ii. Current QA supports search query and term fetch
      iii. Add contextual lookup support in QA linked data configuration
      iv. QA Current set of linked data authorities: OCLC Fast, LoC (term fetch only), Agrovoc. Additionally Geonames is ready in a PR.
      v. Exploring expanding to: VIAF, Orcid, ISNI, dbPedia, LoC (adding search via caching server)
      vi. Combining of search results from multiple authorities
   c. Frontend
      i. UI for Hyrax that provides contextual lookup
      ii. UI for VitroLib (a cataloging tool) that provides contextual lookup
      iii. UI considerations when combining multiple authorities in a single result set
   d. Outstanding Questions
      i. How much of the lookup context will happen on the cache server side and how much in QA?
      ii. Which has the intelligence for searching the cache, the server or QA?
      iii. What search algorithm to use? Some options: purely lexical on primary label (e.g. skos:prefLabel) or on configured text fields (e.g. skos:prefLabel, skos:altLabel, etc.) or support contextual search (providing additional info like birth/death dates in addition to q=person's name)
      iv. Cache server returning a predicate with a search ranking as part of the results to allow maintenance of the search order when put into a graph
      v. Do we want to expand cache server search to include resume tokens to allow for larger result sets?

3. Do we have membership to sustain this group? If no, what are the next steps?
   a. There's value for us in continuing to touch base.
   b. We'll put out a call for agenda for the next call. If we don't get agenda items we'll do this again next time.

4. Add your project updates / planning topics here!

Next Meeting: June 19 at 12pst/3est