2016-06-23 Fedora API Extensions Meeting

Date: Thursday, June 23, 1pm EDT (-4 UTC)

- Dial-in Number: (712) 775-7035
  - Participant Code: 479307#
  - International numbers: Conference Call Information
  - You may also call in using the VoIP dialer from a web browser, or Android/iOS apps
- IRC:
  - Join the #fcrepo chat room via Freenode Web IRC (enter a unique nick)
  - Or point your IRC client to #fcrepo on irc.freenode.net

Attendees

- Nick Ruest
- Jared Whiklo
- Daniel Davis
- Elliot Metsger
- Unknown User (acoburn)
- Ruth Duerr
- Joshua Westgard
- Bethany Seeger
- Katherine Lynch
- Andrew Woods
- Stefano Cossu
- Diego Pino Navarro
- Hanh Vu

Agenda

1. OR '16 Updates
   a. Informal API-X + CLAW meeting, see notes on github issue #18
2. Review of design doc comments - ratification date: July 7th
   a. What is the target audience? developers? users? fedora-community?
   b. Terminology - are "Exposing" and intercepting the right term to use for extension modalities?
   c. Ontologies in the repository - necessary? As binaries? objects?
   d. Reasoning, are we OK with reasoning in general? Limit the scope to object + extension definition + any owl:imports?
3. Priorities for next two weeks

Minutes

Update from informal discussions from OR

- Discussion on the similarities and differences between CLAW and APIX:
  - Nick Ruest will serve as a Stakeholder in API-X since there are a few shared goals and technical points between the two projects …
  - Service binding aspect can be addressed via API-X: If we could have enough of a design and specification in API-X that there could be multiple implementations, CLAW could be an implementation of API-X in PHP (while we're currently working on the Java implementation of API-X.) See GitHub issue #18 for more details.
  - URL rewriting came up in discussion: APIX has a different approach than CLAW. Once implementations are available we'll be able to see how they work with services in place.
- Andrew Woods introduced Aaron to Justin Simpson (Artefactual). Aaron had a conversation with Justin on API-X potential to providing base level of plumbing that is desired by Archivematica.
- Presentation on API-X at OR went well/well received.
  - Andrew: presentation and design documents highlight that we need to have clear messages for different types of people. Many people at OR were blown away by the level of details and the moving parts in the presentation. We also need to craft a message at a higher level that focus on the business needs rather than the technical aspects.
  - Ruth: follow up on Andrew's comment. There's a combination of audiences. Design docs are meant to be communication mechanism between communities. Looking to the next agenda item, we need some design materials at a higher level for the stakeholders to take to their communities.

Design doc review comments - highlights and discussions

What is the target audience of the design docs?

- The original doc by Aaron were aimed toward developers and are very technical, as we have been trying to achieve a common understanding amongst ourselves about what API-X is and does. These documents aren't intended help the outsiders in figuring out what API-X is and where it address their need.
- TODO: Ruth will write up elevator pitch version of API-X description and overview. This will be a separate document from the existing design doc but will be linked to from the design doc. Draft document is available - currently in comment mode...
- TODO: Aaron will create GitHub issue for the elevator pitch document and assign it to Ruth

- Agreement on the existing Overview doc to serve as a way to navigate denser more detailed design docs and elevator pitch is best written by Stakeholders.
- Andrew address that we link to and reused material from the API-X proposal documents.

Terminologies:

- “Pipeline” used in the design doc is a confusing term.
- Aaron proposed to use “exposing” and “intercepting” instead.
- “Filtering” and “Matching” was suggested but was deemed to have different meanings in different context.
- Consensus around using “intercepting” as it more accurately describes the action performed by services

Ontologies in repository

- OWL is used to reasoned on objects to determine if they are of a certain class that can be acted upon by certain services. -> does it make sense for ontologies to be in the repository as resources, exposed to services so that the matching of objects and services can be performed.? This will guarantee that references to and from the ontologies are resolvable when needed.
- Discussion on storing ontologies as binaries or as RDF objects in the repository. There are no best practice. After a lengthy and broad discussion, a hybrid approach seemed acceptable:
    - Stored as binaries address the long term preservation concern, one which has plagued other communities, rendering services/tools unusable when namespaces, references become obsolete.
    - Stored as RDF triples in the repository help provide additional functionalities around the triples themselves.
- Conversation turned its focus on whether the guaranteeing of OWL imports is within API-X’s scope of responsibilities; does/would/should API-X be concerned about resolvavility of ontologies, addressing Ruth's observation that ontology resolution can be problematic.
    - Example: PURLs are now essentially deprecated/frozen by OCLC, shutdown of google code caused turmoil when ontologies hosted there had to move
    - While it seems that the responsibilities for guaranteeing OWL import resolution may lie with the services, there is concern over the practical/performance implication in that approach: without access to these ontologies directly, API-X would have to go through all of the registered services to determine which services apply.
        - Can a middle of the way approach work: having ontologies stored (in whichever way) in the repo in cache time?
    - Questions/options to evaluate:
        - Is API-X responsible for making sure that all OWL ontologies related links are resolvable by providing specific functionalities?
        - Is API-X just responsible for establishing best practices?
        - Should we just put the stake on the ground now about API-X making this resolvable guarantee and work out how to do that later, either via hosting the ontologies within the repository or else where?
- TODO: create a GitHub issue capturing the discussion around this topic. Continue this discussion offline.

Priorities for next two weeks

- Continue to review and comment on the design docs
- Design will be finalized at the next call on Feb 7th.

Next meeting July 7, 1:00 pm EDT (UTC -4)