2016-09-06 Meeting notes - Andrew Woods

Date

06 Sep 2016

Attendees

- William G. Cowan
- Randall Floyd
- Heidi Dowding
- Jon Dunn
- Andrew Myers
- Karen Cariani
- Michael Muraszko

Documentation Sent

PHYDO Data Model

Asynchronous Storage Proxy API

PHYDO asynchronous storage interactions

Open Repositories 2016 Demo

Features Roadmap August-December 2016

Questions

Add questions for Andrew here!

Discussion items

Time	Item	Who	Notes

Meeting Notes

Functionality of asynchronous storage as being a layer above Fedora or a native Fedora feature - wants to be involved with making sure that the work we're doing conforms or complies with the work being done with APIX

What is current status of APIX?:

Doesnt' have all the details, but still quite active, meet every week or two, at a good point for ensuring alignment for these description docs and a level deeper with the technical specifications of the APIX expectations (currently those documents are in a strong draft format)

Has been some prototyping (Hopkins, Amherst, AIChicago) -- have pulled in different code that implements APIX patterns

Have been touch points along the way with asynchronous storage -- there is agreement and interest, full support there

Randall: Nianli's work in Camel, hasn't gotten the impression that there's a piece of code to test this yet ; there's no gap here yet; development work would be to make one of those services the responder to APIX call

Trying to figure out how to make it work first, then we'll try and figure out how to make it work on the APIX front end

Andrew: would it be useful to start pushing the work HD2 is doing to push this into the APIX fold?

Randall: Would defer that to Nianli, not sure how familiar she is yet with the prototype examples, etc

Andrew: whenever the time is right, would be interested in tightening that relationship

Will: Interested in making that relationship happen sometime this fall, but not sure exactly when. Will be looking at how we can take our functioning Camel prototype and make that into a service that works within APIX. Haven't put that on a timeline.

Jon: Would encourage us to start talking about this sooner rather than later in terms of APIX

Will: what we're putting in place is our first draft, we don't need all the pieces to be working exactly right but we need to understand what all the working pieces are so when it comes time to start working with APIX we can know where those interactions will work best. We might decide that it's broke down into the three pieces we've decided, or it might look different.

Andrew: We can talk about details in a subsequent call maybe. But interested if Nianli or we have a code to be built, ready to be deployed, etc; would facilitate his working with the APIX folks to turn this into an APIX project

Randall: in the meantime we've made this service-oriented thing work, would just be a matter of implementation. We've also worked on the client part and the interaction part and that's what mostly was shown at OR. Important to recognize that for for this to be in the Hydra community there's a part of this that lives closer to the Fedora component, how that might be not camel based (possibly Sinatra based)

Andrew: Going forward, what communities could take advantage of the work we're doing

Randall: Pulled out 2016 OR demo link:

- Client library for the HD rails app that does the connection between the two of these; how the client-side acts on those responses; happening from a fairly abstracted library; views and presenters knows how to interact with the client

In APIX these calls would like something in the frontend of APIX that knew how to respond to the storage

Andrew: What's going on in Fedora and how this interacts with the HD2 project --

Randall: Worried that there might be projects taht could invalidate some of our work; Hybox results - webapp+cloud as a way to store binaries (coop the datastream way to store things and use S3 as the backend) -- what does the request look like for that? Wondering if that's a competing strategy to what we've done

Andrew: No real projects doing what we're doing

The S3 work is right now a very niave approach, using S3 as native storage; if there's a delay in responding to storage, the user experiences that delay

Additional phases right now, suspects as the S3 work matures they will be either looking at what's going on with HD2 or may possibly just put in caching so the user only experiences delay the first time

Wouldn't see this being in competition, hopefully it'll be a collaborative approach as they get further in their work

Jon: to clarify, S3 doesn't have the same latency issues as tape, but does have more latency; Glacier has classes of service, but S3 doesn't

Randall: Glacier as a use case more than S3

Andrew: Glacier hasn't been addressed in that project

Randall: Yeah, hopefully that will be a connection; it had always been a thought thta we'd need to understand the basic services; instead of that you just hand it over to the next port of call; this could provide the user interactions for asynchronous instead of it being completely blocking

Andrew: If this starts to converge -- Duraspace people are working with Hybox on that stuff, would be very easy to bring them into this loop

We're still definitely reasonable in this approach, according to Andrew

Andrew: A few recommendations around Fedora:

- 4.6.0 was just released, that will be the last that uses modeshape under the covers -- had a release that changed its underlying storage, but 4.6.0 still uses the older release; has an impact on installations, will have to do a migration

  - Starting on what is now not a release, be keeping in mind that you'll have to be thinking about the next release -- two months it will be released

- Mysql or postgres is encouraged to be used for new release, no more default database?

- To mitigate the possibility of running into issues of corruption, there are steps that can be taken

  - specific Javaops to pass into JVM environment he can recommend

    - Drew: what are red flags to look for with corruption? Andrew: has seen with the application runs out of memory, another reason to get into the 4.7 release (gets rid of infinispan, which causes problems)

  - There's work going on around client-side tooling for dumping repository as RDF; also dumping out the binaries

RDF

head requests -- people like that pattern and we don't expect it to change

with the authorization at the top level, what degree WebAC -- how are we doing authorization?

Randall: right now it's purely fedora code? use of URL redirects; still is a function of the hydra application, but need to figure out how something that doesnt fit that scenario can be authorized

...

Andrew: Would be good news if we could verify what hydra is doing is enforceable at the fedora level

When authorization time comes (November?) -- would be really good to prove that this has been actually accomplished; if it hasn't, making it happen

Jon: The content modeling as well

Andrew: no concerns, but doesn't fully understand the diagram; assumes that Julie H has her hand in this work; knows conversations with PCDM and PCDM 2.0

describes/describedBy

Drew: if we start putting more things on the files using the predicates, going to require a lot of code change through the hydra stack

- Technical metadata is kind of assumed; only one resource described, etc etc

Working through that discussion with Esme now

Other notes