Samvera Tech Call 2017-11-15

!!! Please note that Call-In Info has changed as of October 12th, 2017 !!!

Call-In Info:

Zoom:
https://psu.zoom.us/j/613720745 (Requires Zoom)

Telephone:
Meeting ID: 613 720 745
+1 646 876 9923 (US Toll)
+1 669 900 6833 (US Toll)
+1 408 638 0968 (US Toll)
International numbers available: https://psu.zoom.us/zoomconference?m=UZ_PRwQ56TNX1pDiisdDlnAu8XPVqtiX3

H.323:
Meeting ID: 613 720 745
162.255.37.11 (US West)
162.255.36.11 (US East)
221.122.88.195 (China)
115.114.131.7 (India)
213.19.144.110 (EMEA)
202.177.207.158 (Australia)
209.9.211.110 (Hong Kong)
64.211.144.160 (Brazil)
69.174.57.160 (Canada)

SIP:
613720745@zoomcrc.com

Time: 9:00am PDT / Noon EDT

Moderator: Carolyn Cole

Notetaker: Jennifer Lindner

Attendees:
- LaRita Robinson (Notre Dame)
- Jennifer Lindner (DCE)
- David Tolmie (University of Washington)
- Jim Coble (Duke)
- Justin Coyne (Stanford)
- Esmé Cowles (Princeton)
- Adam Wead (Penn State)
- Thomas Scherz (Cincinnati)
- Collin Brittle (Emory)
- Chris Colvard (Indiana University)
- Carolyn Cole (Penn State)
- Brendan Quinn (Northwestern University)
- Carrick Rogers (Northwestern University)

Agenda

1. Roll call by timezone per following order - ensure notetaker is present (moderator)
   a. folks outside North and South America
   b. Eastern timezone
   c. Central timezone
   d. Mountain timezone
   e. Pacific timezone
   f. folks who were missed or who dialed in during roll call
   g. Welcome all newcomers!
2. Agenda (moderator)
Call for new agenda items (moderator)


Notetaker and moderator for next time

Notes

A. Call for new agenda items (moderator)

-- No new items.


B. Data Mapper working group closed up a few weeks ago, with two issues we want to discuss today:

A PR to add a new back end for Redis, for back ends that can't be done synchronously, in this case cloud search doesn't commit right away, so he can cache into Redis. --- Trey thinks the implementation is fine, but question is whether we want to have a new adapter, knowing the maintenance burden of them. This is just a metadata adapter, not a storage adapter; is there a high demand for it? -- a lot of institutions would use it at some point, because it's significantly cheaper when using AWS -- cloud searching is much cheaper with it. I don't understand how it's supposed to be used because it doesn't fulfill a lot of queries -- nvm that's a null adapter I was looking at. It works the same way as an in-memory adapter, there's a version two for more complex or compound queries. So, is the theory that you'd use Redis rather than look in cloud search? Because of hours of look up time required in cloud search, you look in Redis cache first and then cloud -- also Redis first it will be accurate while cloud search will update later. There's a follow-on PR.

Moderator: is the question really is this PR technically sound, or is it whether to add new adapters? A: First question was do enough people need this to take on maintenance -- Justin has questions about its soundness.

-- There's another path, we have the plugin option, the plugin working group did a lot of work on this topic, this sounds like a good candidate for a plugin to me. Michael Klein has another PR which sounds like a cloud search adapter, -- ... plugins of all eventual-consistant stores? but maybe we should let it sit outside the call for a bit - agree these might be topics for further discussion and reflection outside, and maybe come up with a process for determining which plugins are going to be part of Valkyrie's code base; general consensus is maybe this should be a gem, which is fine.

C. Versioning discussion -- we have two PRs to support versioning inside Valkyrie, the linked one (https://github.com/samvera-labs/valkyrie/pull/312) is the sort-of preferred one. It's best we can do right now, not exactly a highly preferred option. With some things that don't support Versioning like disk we need to tree-ify ids or we say don't support version. We implemented a version method that either returns a list of ids or false -- difficulty is we are treating each version like main file so you can ask each version for more versions and if you do that in Fedora, it blows up.

What creates a new version rather than uploading a new file? -- We are always creating a new version.

How are two versions linked? The resource is the same - two files attached to same resource, resource assumes all files are versions. So how do you distinguish actual unique files? Don't know how to deal with this right now, unless we properly model versions and store it somewhere -- let's sort this out offline, off the call.

API versioning is different from other kinds like file and timestamps -- can we treat timestamp as identifier? yes ..

The second decision the Data Mapper working group made was to look at any issues that came up and discuss after the call, so anyone who wants to stick around to talk about them, let's do that.