2015-06-08 - Open Repositories Committers Meeting

What/When

Open Repositories Fedora committers/tech-folk meeting:

- https://www.conftool.com/or2015/index.php?page=browseSessions&form_session=17

Monday, 08/Jun/2015:

- 9:00am - 12:30pm

Remote access

- https://plus.google.com/hangouts/_/event/c1glu6soq43r1rr6ou17qtobug8

Those Who Expect to Attend

- Nick Ruest
- Unknown User (acoburn)
- Jared Whiklo
- Michael Durbin
- Andrew Woods
- Fedo Raadmin
- Eric James
- Osman Din
- Elliot Metsger
- Stefano Cossu
- Aaron Birkland
- Unknown User (westgardja)
- Harry Sidhunata
- Katherine Lynch
- Sarah Potvin
- Maureen Walsh
- Mark A. Matienzo
- Chad Mills
- Chris Colvard
- Jeff Leedy
- <add your name here>

Please also sign up for this meeting and any other OR2015 Monday workshops and tutorials that you plan to attend via the signup form linked on the Workshops and Tutorials page of the OR2015 website. Please make sure that you are already registered for OR2015 before signing up for workshops.

Agenda topics

1. Introductions (all)
    a. What is your current Fedora status?
    b. If you are not on F4, what are your migration/installation plans?
        i. What are your barriers?
    c. Vote for ModeShape issue: https://issues.jboss.org/browse/MODE-2109
2. Local implementations and/or implementation ideas, issues? priorities? (open forum)
    a. Data Conservancy: potential for community interest for an "API extension architecture" (work-in-progress draft)?
    b. Non-RDF source descriptions triples
        i. Eliminate `fcr:metadata`
    c. Fcrepo source code (OSGi support, Java8, RDF libraries, dependency proliferation)
    d. Web ACL / Storage Policy
    e. VIVO/Fedora integration
    f. ...
3. Long-term project strategy (Andrew)
    a. Core features
    b. Relevant API standards
        i. LDP, WebACL: How much (and what kind of) representation to W3C WGs do we need?
4. Developer community engagement
    a. Hackfests?
5. <add topic here>

Minutes (please edit if I missed anything)

****2a****

- For the Data Conservancy Fedora 4 ticks alot of the boxes, but missing some things.
- Domain specific apis is one such thing
- Is there a way to update Fedora to support these domain specific APIs.
- Put forward this use-case. Like to get a sense about supporting this effort and if you like it, perhaps some donated wiki space for use cases and discussion. Get support from the community.
- Dissemination architecture, kind of analogus to what dissemenators where in Fc3 but for Fc4. A lot more usable and share-able.
- The general problem is the Data Conservancy (or it's clients??) wanted to extend Fedora's API, associated with resources.
  - For example, Post a tar-gz to a resource and have it attached,
- To address these needs at a layer between Fedora and the outside world for addressing those concerns. Redirect traffic to this tool.
- Want to redistribute code to share these resources, and a "CMA" to reason about the options available on an resource.
- The Data Conservancy is already committed to be doing this anyways, want to see if this would fulfill the missing disseminators from Fc3 in Fc4.
- Stefano Cossu is also moving forward with their own implementation of a similar use case. Resolve whether resources from Sparql or Solr index, are properly referenced/controlled in the access control layer.
- Another possible use case could be having a content model being applied at the ingest process. If you have derivatives already created, upload a multi-form data and have it place content in appropriate containers based on content model.
- Part of the use case document touches on parts of the CMA, interested in whether defining content-models in Fedora 4 (these intersect).
- As far as ingesting content and having different actions occur based on a content model, is exactly what might belong in this layer.
- These actions exists outside the core and acts on resources internally. It would be advantageous to be able to distribute the capability by jumping the OSGi band layer and use something like Apache Karaf. One can imagine a Karaf features XML file which defines the necessary jars to implement these (actions). Within scope would be deciding what the convention to package up these resources in a usable package.
- Question: Is this something that will involve changes to core Fc4 code or is more akin to the PCDM community effort?
  - This is meant to be a layer between Fc4 and the outside world, perhaps in an entirely different JVM container.
  - Conceptually it would be a layer distinct from Fc4. This layer would be for anything "out of scope" for Fc4.
  - It may operate as a service proxy. The change is to have a service model with the object to share these special actions.
  - Andrew Woods: This would involve minimal changes to core code.
- Question: Would this work with Fc4 access controls right now?
  - You can currently do Fc4 access control right now or shuffle that need up a layer and handle it before Fc4. Principles of the original request would (or would not) flow through to Fedora. The service layer might change the request.
  - If you wanted to know a priori if when you invoked a service would you be able to actually complete the request. This service layer could need to know how Fc4 applies these authentication policies, perhaps the WebACL discussion would have some impact.
  - At this stage of producing a sketch of this proposal it would be part of a future discussion.
- Question: Would this be a place to implement WebACL to authenticate across repositories?
  - Yes, it is operating as a reverse proxy with a camel route which... if you can envision it, we can glue it in.
  - Hoping to generate more use cases for this, that could definitely be one.
- Check out the initial use cases on the proposal document. These are data conservancy's use cases, if there are people in the community that are interested in this we would want to flesh out these use cases.
- Question: Could another use case be a IIIF image server?
  - Yes and Johns Hopkins would be very interested in this.
- The idea would be to embrace OSGi, and things like Apache Karaf. The OAI-PMH provider specifically, then theoretically without having to worry about this layer. Deployment would be provided by OSGi framework.
- OSGi'"fying" Fedora core is a significant investment of time.
- This OSGi'ing could be external to Fedora, but there is a benefit to Fedora.
- OSGi is a standard for including additional information in Jar files such that packages could be hot-deployed. A means of packaging up. Smart class loaders in Java, allows this hot-deployment of services and technology. A technical framework.
This proposal would require Camel and the web apps in one framework and Fedora in another. To make a change, you would need to restart the container environment to make changes.
Karaf has a dynamic configuration framework, if I want to change the configuration file, the second I change the file it will auto-redeploy the appropriate routes. OSGi defines this lifecycle and the Karaf manages the lifecycle.
This has no actual impact on Fedora core, other than the idea of OSGi layers. To understand what is in scope of Fedora and what should go external.
- Question: What support for OSGi in service providers, is there such support?
  - Unsure of the level of support.
- Do we see a reason not to support this effort? No objections.
- What would be the most useful or helpful ways to rallying support as the data conservancy will be implementing this framework anyways. Get the document out to the community and schedule some calls to get some use cases. A wiki page to coalesce the ideas and discussion.
- Need to work out a timeline. First step is to put the document out to the community in general, and see what the uptake on it it. If people are interested in the layer, then figure out how.
- Very important feature to Art Institute of Chicago that they are implementing their own.
- Question: What about Hydra or a middle layer?
  - That would need to be sketched out, defining who is interacting with whom and where certain functionality lives. Perhaps there are services written in Java that does something and maybe Hydra uses that. There are many options that would need to be sorted out.
- The Data Conservancy are interested in Blacklight and maintaining an index of these resources which could be maintained by this layer.
- Data validation could be dealt with in Hydra or at this layer.

****2b****

- If you have a binary resource in Fc4, there is an associated resource that describes the binary datastream. Currently it is the binary URL + /fcr:metadata. We are moving towards a rule within the repository and you get the RDF that describes it, that the subjects in that RDF is the resource and not resource + /fcr:metadata. So this breaks the convention as some are fcr:metadata and some are the actual resource.
- Would it be possible to get rid of fcr:metadata and use content-negotiation. If you ask for an RDF format you get the RDF description and if you ask for JPG you get the binary back??
- An attractive end-goal is keeping fcr:metadata where everything you get back is all source or fcr:metadata.
- If you have an arbitrary RDF datastream and want to add it to the repository. You need to define a different content-type (other than application/rdf+xml).
- How do you define what resource describes "me"? It could be simple, but if not it could be very weird (Unknown User (escowles@ucsd.edu)).
- What would the binary be in an LDP sense? An LDP-NR and LDP-RS? Unless we default to either all binary or all RDF.

- With Marmotta, how does it deal with binary resources and their descriptions? The interaction pattern is that everything is a container. We are harvesting metadata container from providers, an LDP-RS gets created at the URI. Marmotta tries to guess the type and then appends the extension to the datastream. When you get the content with the suffix and description without.
- Interim step of single subject'izing. What is appropriate subject?
- the non-RDF resource, as this makes more sense.
- except administrative metadata, which is in fcr:metadata. That might be useful. But makes it a first-class resource itself.
- Do we have the last modified date separate from the binary? We don't think so.
- Is the last modified date of the fcr:metadata relevant? People might be storing information in the fcr:metadata and so having last modified date to determine changes to this might be in use, if not a mistaken implementation.
- Is fcr:metadata just another metadata container that can be attached or linked to the binary object.
- If fcr:metadata did not exist and I had a binary stream. How would I describe that binary. Opaque RDF would be how to do that.
- Could the solution be to have multiple properties, some reference the metadata around the binary and some reference the content of the binary.
- Use two different predicates, in this case.

- Look at how Lilo (sp?)
- Discussion around how fcr:metadata works right now.
- Cleaner if we had single subject and the binary being the subject, there are some use cases where metadata changes would be nice. But it would be cleaner if the rule was that when you request RDF you get only RDF around that subject.
- There are some issues, but are those issues significant that we don't do that. Is any reason to support the current framework?
- The strong argument for keeping fcr:metadata are based on a strong realamce on fcr:metadata and using it incorrectly. Perhaps we should have some community support around doing this correctly. To be LDP compliant we need to remove any LDP-RS associated with the LDP-NR.
- Perhaps restricting, what you can do on fcr:metadata and provide better error messages.
- Would the description of fcr:metadata includes a this describes binary resource.
- If you are using a representation of an object. If at the same URI we are presenting both a description and a representation of an resource, it could be messy.

****2c****

- OSGi, this is not some much making Fedora core fully runnable in OSGi, but getting us on a path to it. Currently we publish all Fc4 artifacts as OSGi bundles. The problem is it defines what packages it exports and imports.
- The problem is these Fc4 hacks in fedora-impl, there are some internal classes in the modeshape layer that are being hacked in to making changes. So Fedora is importing and exporting the same packages. It works in terms of war deployment.
- But removing this should be a priority, and it requires some understanding of how to do it.
- There might be a way to approach Modeshape to make some of these changes, also try to use some of the public methods to do the same things we are doing.
- Other than package invasions, is standard practice with OSGi that has internal classes that are not exported. The Maven bundle plugin with take any class that has impl in the package and excludes it, making it non-exported. Perhaps rename some of these packages to get around this. kernel -> kernel-api and kernel-impl -> kernel. Same thing with the other impl packages. Do the modeshape hacks we have "need" to be exported? As far as we know they do, but someone with more experience might be able to do that.
- That is for building them as proper OSGi bundles, the next is being able to hot-deploy extension modules. We would have to think that through.
- What would be the benefit of hot-deployment? If you have a Fc4 instance and there is a new module, and you want that to run inside Tomcat then you could be required to re-compile the entire application and restart the container. Webapp-plus with different configurations as an example of how we are doing this now. OSGi is just allowing you to deploy vanilla fedora and then add/remove any feature module without having to re-compile or cycle the container.
- Seems like a good way to facilitate bug reporting, to be able to test use-cases with a vanilla configuration. Imagine different profiles that could be deployed.
- What is the effort OSGi?
  - packaging and deploying with Modeshape could be complicated. Unknown User (acoburn) has made attempts but has been unable to resolve it so far. Better understanding the invasion of modeshape namespace in Fedora 4 namespace.
- RDF libraries - in the fc4 core we are using 2 different libraries (Jena and Sesame) in fcrepo-camel using Clarrisa. RDF is central to the idea of Fedora, but we are not doing a lot of complex RDF operations.
- There is a project called commons-rdf (Apache), very early. It defines an interface (Java 8) using Java 8 streaming api. Jena and Sesame are moving towards that. We should look at using this in fc4 core instead of specific Jena and or Sesame classes. There are implementations from Jena, Sesame and Marmotta. (A. Soroka: There is definitely *not* an impl from Jena, yet, although the Jena project is thinking hard about how to do one. Nor, to my knowledge, has Sesame produced one. See here for current known implementations.)
- Consolidate our RDF libraries for good reasons. We should keep an eye on this project as it could help us. We could get into the discussion, not much room for influence. More that these RDF libraries need to implement the new commons-rdf api. Will this commons-rdf API will meet all our needs? Not sure currently.
- Skipping dependency chain in Java 8.

****3a****

- What are the core services that Fedora should offer?
  - CRUD operations
  - versioning
  - authorization?
  - transactions?
  - fixity?
- What are existing standards that define services that we provide and which can be handed off to others. ie. LDP for CRUD, authorization - WebACL, versioning - Memento.
  - A. Soroka: Some standards may be usable for part of the service: e.g. PROV-O or PREMIS may be a suitable ontology for a fixity service even if they don't explain interaction.
- Where there is not a standard, is there an oppourtunity to influence or start a standard.
- Features like transaction and versioning are attractive of Fedora4. Are those features core to Fedora4 or come from Modeshape, etc?
- We want clear agreement and responsibility of the core features. Standards based implementation would allow easier to move the data into and out of Fedora.
- Avoids any sort of lock in.

- All for adopting standards, depending on those standards. Say we have serialization, whom else is using this? If we go towards standards, we should look at who is using these standards.
- Good to make some movement in this direction.

****4****

- Developer community engagement? Why don't people jump in? What can get more people in.
    - Web development (PHP, Ruby)
    - Best practices, IDEs, mocking up internal interfaces. Lots of different ways of doing this.
    - Just saying what contribution you are open to accepting?
- What are good channels? Nothing you are doing that is wrong? Give feedback?