2016-06-13 - Open Repositories Tech Meeting

What/When

Open Repositories Fedora committers/tech-folk meeting:

- https://www.conftool.com/or2016/index.php?page=browseSessions&form_session=87

Monday, 13/Jun/2016:

- 9:00am - 12:30pm

Remote access

- https://plus.google.com/events/c1ol74vk74ru7edlomnqdb405eo

Those Who Expect to Attend

- Andrew Woods
- Fedo Raadmin
- Andy Wagner
- A. Soroka
- Esmé Cowles
- Daniel Davis
- ½ of Benjamin Armintor
- Aaron Birkland
- Nick Ruest
- Diego Pino Navarro
- Fulgencio Sanmartín
- Clifford Frey
- Jeff Leedy
- Unknown User (matthias.razum)
- Raman Ganguly
- Peter Sefton
- Jon Gibson
- Michael Durbin
- Lutz Biedinger
- Richard Jones

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? What are your barriers?
2. Clarify distinction between NonRdfSource and its Description as one repository resource or two LDP resources
    a. Example, what are the implications of event messaging from actions on either "resource"?
3. Fedora API Specification
    a. Seeking initial agreement, followed by alignment of implementation to specification
4. ModeShape5, backend databases, and migrations
5. Atomicity in the Java kernel API
    a. Which methods on services and resource types are atomic and / or synchronous? This is going to matter to reimplementations that reuse the Java kernel API.)
6. Local implementations and/or implementation ideas, issues? priorities? (open forum)
    a. ...
7. <add topic here>

Minutes

API Specification

- Acceptance process. Should we finish CRUD, as others depend on it?
    - Not ready for acceptance process yet. Get CRUD and versioning to shape where acceptance can begin

Versioning

- When looking at a version that is a version of an original, is there a link to the original version?   Yes
- Most important thing is that a client understands if it a versioned resource or nor
- In the current implementation, if you look at a versioned resources, all links are to the snapshot version

  - Can be crawled by an LDP client that has no awareness of versioning mechanism
  - Links to resources in the version snapshot tree will be this way
  - Links to resources **outside** the versioned snapshot tree are to non-snapshot URI
    - So a client that is LDP-only client that is unaware of versioning semantics can escape the snapshot tree and not be aware of it
- The most consistent approach would be to link to canonical (current version URIs) always, have memento-aware client look for versions.
  - Could implement this using JCR in place currently
  - Representation of versioned resources and snapshots would be different
- Esme: I thought there was some way to set versioning policy
  - Mike: That was thrown out a year ago
- 7.2.2 is the default behavior in Fedora right now
- 7.3.3 would be the least challenging to implement right now
- Andrew: How important are versioning to people right now
  - Diego: We need them right now, everything is incremental
  - Sometimes versions can be in application logic, leave it as a separate concern from the repository. May be dubious value to rely on repository versioning
    - Different applications are going to have different notions of versioning. Some use cases may consider different versions to be different resources entirely.
    - Definitely some strong use cases for versioning as it is currently implemented now (and proposed in the spec),
    - "I need the ontology (resource in the repository) that matches its state at X time in history"
- If one deletes the original version of a versioned resource, what happens to the version history?
  - Currently, If you version binary directly and delete it, you'll delete all versions. If you if you have a binary in a container and version the container, deleting the binary will not delete the link to the binary for versions of the binary that are linked by versions of the container (somebody please fact check and/or state this)

Batch Atomic

- Basic idea is to start a transaction, perform requests, COMMIT or rollback. There had been some talk about timeouts
  - Use case for "timeout all transactions" e.g. scheduling a reboot
    - One idea: Just say "Implementation may choose to cancel transactions for any reason" in spec, don't specify how to do it
  - Use case - ability checkpointing/continuations. Complex process in atomic batch. Would like to define incremental points that can be rolled back and continued.
    - Continuations/checkpointing is probably a concern not for the core
    - Dan: At a general level, maybe batch atomic should not be a core concern of the repository at all. Definitely have use case for "everything, or nothing",
      - Adam: Probably should be in core, because broader Fedora community has asked for it and paid for it

Authorization

- Assumption is that requests are pre-authenticated, Fedora is enforcing authorization
  - Resource points to policy "I'm protected by this". Policy points to resource or class of resources that are protected by it.
- Mike Lynch couldn't figure out how to pass user attributes to Fedora in order to enact policy
  - user principal, on-behalf-of header should be in request, it's configurable (but needs to be defined at build time). The task is for something to populate these headers.
- In an unreleased version, users can be strings or URIs. It will be in 4.6.

Fixity Checking

- On ingest, or on-demand via REST request. Can store fixity events in repository.
- How does it extend to other checksum algorithms?
  - SHA-1 supported because we get it "for free"
  - How do we specify how to specify a checksum?
    - Ingest- a header. On-demand, in POST
- For huge files (video files) there is no spec for finer-grained checksums
- In the EU, checksums and signatures are indicated in an XML format, checked by external tools. Not sure if it's XML signature spec per se, but it's expressed in XML. The tool is called Checklex: https://checklex.publications.europa.eu/faces/AboutCheckLex.xhtml;jsessionid=1757FE945F9B55308A09C3E05723171D?lang=en (specs available under request)
  - For example, this signature: http://eur-lex.europa.eu/legal-content/EN/TXT/SIG/?uri=OJ:L:2016:154:FULL&lang=EN for this PDF: http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2016:154:FULL&from=EN
  - External tool can schedule re-checking of resourcces
  - When changed, a new signature file needs to be generated.
- Will/should fixity generate events picked up by audit?
- Use case: Fixity against entire collections (trees of resources)
- Is there a method of fixity checking for external sources. Is there a standard, e.g. like s3 buckets, swift?
  - Not aware of a standard, but maybe s3, swift indicate a defacto standard?
  - Can we say in the spec that we expect an external application to include a certain header?
  - Are external resources in the spec at all?
    - There is a note, this will make it into the spec at some point
    - Storing fixity of such resources seems to be reasonable for Fedora to do
      - Some desire just to defer to external system
      - .. but repository can't/shouldn't necessarily trust these external sources.
    - Maybe it's possible to have user specify checksum when creating external resource, or have repository pull in the external source's notion of fixity.
  - In a clustered scenario, do we consider fixity to be a consensus?
    - Self-healing needs to be hidden, or storage needs to be transparent
    - There used to be code related to this, there was a pattern

Projection/Federation

- Core spec "this is what it means to do Fedora". We got federation "for free" from Modeshape, but it it's an implementation detail
    - Recognized that projection is not a core capability, but will be extracted from core code base, made into an optional feature that depends on the modeshape impl of Fedora.
    - It has been an intriguing, but mysterious feature. Not clear to users if they can/should use it
    - Mike: It was too good to be true, but never really worked. For example, you could not create links from federated resources; federation breaks if two files share the same checksum
        - Will document the issues, since people seem interested

Performance

- People have been meeting monthly, have generated test profiles, have some reports.
- Have run tests against modeshape5. Some results have been better, but no conclusions yet
- Get a good understanding of performance characteristics, document it
    - What settings can improve performance on certain hardware
- With modeshape4, leveldb performed better than databases, but didn't scale as well
    - Princeton tests: leveldb started fast, collapsed. Postgres started slow, stayed performant
- With modeshape5, one datapoint suggests Postgres performs as well as leveldb.
- Andrew: Note, we've noticed leveldb can become corrupt, so we'd like move to a DB, recommend against leveldb.
    - Removing leveldb default configuration
    - One click still has leveldb, but may move to some embedded db like Derby
- Was there a comparison to Fedora 3?
    - There was, not any more. It seemed that fedora 3 was more performant on reads, fedora4 on writes.