Decisions and Open Questions

Decisions

Functional Decisions

1. Drop the Single Subject Restriction and allow out of domain subjects
2. Only enforce referential integrity on server generated triples
   a. No automatic enforcement or cleanup of user supplied RDF
3. Add Archival Group interaction model
4. Permissions are checked on staging content but not on commit.
   a. Implications
      i. User can commit anything that they were permitted to write when they staged it.
      ii. Thus it is possible that a userA would start a transaction at 10:00, update some resources at 10:01, subsequently find they no longer have permission to write at 10:03 because of a restrictive ACL change by userB at 10:02. userA's attempt to commit the transaction at 10:04 will succeed.
   b. This behavior is consistent with Fedora 4.7.x and 5.
5. Search service should be synchronous
6. For a request made within a transaction, access control decisions will be made against the state of WebACs as they exists within that transaction.
7. OCFL versions will map directly to Fedora versions
8. past versions will be deleted (at some point) when the OCFL client supports deleting versions.
9. Unversioned changes will be stored in the Mutable HEAD.
10. Transactional state will be stored in a scratch space and will not be rebuildable.
11. We need to store a digest for binaries regardless of whether the user supplies one.
12. Converter framework: should it stay or should it go? We are moving towards phasing it out as much as it is practically possible
13. We are evaluateing a "Minimal" Fedora 3 migration, where all datastreams from a fcrepo3 object are placed into a single OCFL object with no modifications
14. Autoversioning -
    a. Configurable repository wide? Yes we will support setting a repository wide default.
    b. We will support a per object versioning policy object, similar to fedora 3 (REST API#modifyDatastream (versionable param)) if there is community desire for this feature.
15. Minting versions within a transaction:
    a. Is there a use case for minting multiple versions within a transaction? No known use-case at this time.
    b. We will disallow minting versions within a transaction for starters. We will revisit if there is community interest in supporting this feature.
    c. Transaction side-car spec will indicate that some operations may not be allowed within a transaction.
16. Tombstones handling:
    a. What happens when you delete an AG? A new OCFL version is created that contains only the tombstone
    b. We will continue to handle tombstones from the fedora client perspective in the style of Fedora 5: When GETting the child of deleted parent, the response will be a 410 with a reference to the parent's tombstone.
    c. We may consider exposing the timemap of the deleted resource to administrators in order to allow them to restore a previous version.
17. Internal versus External Identifiers
    a. Fedora will allow for configuration operations to support both mapping between internal/external identifiers, as well as allowing some identifiers to be transported without modification
    b. See the details here: https://docs.google.com/document/d/1LVPeGwfhnqttcRs-KV_ZQSzRrtBEOiTHd1ZhJ5MtWxc/edit#
18. Server Managed Triples
    a. Fedora6 will relax server managed triples rules, such that SMTs present in the user submitted RDF will be ignored and dropped before persistence
    b. A subset of the SMTs will be stored to sidecar files in the OCFL persistence.
    c. See the details of the proposal here: https://docs.google.com/document/d/1LVPeGwfhnqttcRs-KV_ZQSzRrtBEOiTHd1ZhJ5MtWxc/edit#

Nearly Decided

1. Drop support for import of historic mementos in OCFL via the API
   a. Allow or disallow in other backends?
2. 1:1 mapping between Fedora info URIs and LDP paths
   a. This means fedora 3 objects migrated to single OCFL objects would inherently have LDP path http://example.org/fcrepo/rest/${pid}
3. Containment and membership relations will be interpreted from repository structure and indexes, not persisted to disk.
   a. See Containment/Membership triples management
4. Automatically generated checksums
   a. OCFL will generate a digest automatically for storage reasons (not necessarily always the same algorithm), should this be surfaced in Fedora?
      i. We probably want to do this but we should check in with Aaron Birkland first to verify that this is what we want.
      ii. Should the digests included in existing OCFL objects be surfaced in fedora?

Design Decisions

1. Refactor FedoraResource and its sub-classes to serve primarily as data encapsulation objects.
   a. Refactor modification operations into separate service classes.

Open Questions

1. What representation should be used for resources on disk?
   a. See examples towards the bottom of this document https://docs.google.com/document/d/1LVPeGwfhnqttcRs-KV_ZQSzRrtBEOiTHd1ZhJ5MtWxc/edit#
2. Canonicalization of RDF, checksumming metadata, and the possibility of byte-for-byte I/O of metata resources.
   a. Is this of use: http://json-ld.github.io/normalization/spec/index.html
3. Should Fedora support non-container RDF sources?
   a. https://www.w3.org/TR/ldp/#dfn-linked-data-platform-rdf-source
4. Are Pair-trees as a resource still a thing in Fedora 6?
   a. Must keep in mind use case of users migrating from fedora 4 where pair trees were enabled to fedora 6
   b. Options
      i. pair tree maintained (where each directory becomes a LDP Container).
      ii. pair tree paths collapsed down
      iii. pair tree behavior same as it is now in fedora 4/5
5. Should the filename for the triples file containing the user provided RDF of an AG or atomistic container be the last segment of the fedora id, the full fedora id, or a constant name?
   a. Given fedora id info:fedora/my/ag from URL http://localhost:8080/fcrepo/rest/my/ag
   b. Last segment: <ocfl_storage_root>/my_ag/versions/v1/content/ag.nt
      i. pros: Short filename. Filename is relative to the filesystem path, which is consistent with other objects.
      ii. cons: When going from ocfl/fedora id, requires some knowledge of how to calculate the filename.
   c. Full id: <ocfl_storage_root>/my_ag/versions/v1/content/my_ag.nt
      i. pros: filename would match the fedora id, and the directory name of the AG.
      ii. cons: could produce very long filenames, which could cause more inventory bloat.
   d. Constant name: <ocfl_storage_root>/my_ag/versions/v1/content/self.nt
      i. pros: properties for the object itself are always in the same path
      ii. cons: the name of the file is not descriptive and may not be readily understandable by users viewing the OCFL object directly.
   e. In all cases, the fedora id should be contained in the json file for the AG/container. Just have to know how to locate the json file.
6. Do we want|need to continue to maintain pure blank nodes and skolemized blank nodes? This is currently a configuration option. For context here is some information on blank node skolemization.
7. Rebuild from OCFL:
   a. Do we want to support lazy loading of indices?
      i. on demand?
      ii. on file system change?
   b. Do we want Fedora to populate the indices
      i. on demand?
      ii. on startup?
      iii. on startup if the indices are empty?
   c. Do we want an external tool for populating the indices?