Technical Working Group

Technical Working Group

- Ben Armintor - Columbia University
- Chris Beer - Stanford University
- Esme Cowles - University of California, San Diego
- Dan Davis - Smithsonian Institute
- Declan Fleming - University of California San Diego
- Neil Jefferies - Oxford University
- Adam Soroka - University of Virginia
- Andrew Woods - DuraSpace - Technical Team Lead
- Zhiwu Xie - Virginia Tech

The working group's charter.

Initial Objective

Given the areas of assessment enumerated below, the Technical Working Group has decided to prioritize and select the top four areas for initial review. The plans for the each of these four areas and their assessment outcomes can be found:

- F4 Assessment - Pre-Production

Areas of assessment

1. REST API
   - Are immediate updates required?
   - We should version the API independently
     a. This offers multiple backend implementations/optimizations
     b. A. Soroka: I think this requires a stronger definition of the API than currently exists in the form of user documentation. I suggest defining the API as ontology extensions to LDP.
     c. Clarifying and publicizing (formally and informally) the relationship between the Fedora API and LDP.
2. Performance
   a. Read
   b. Writes
      i. Many small files
      ii. Large files
      iii. High throughput
   c. Scalable serialization to disk
      - Need to measure scale of load that async serialization can meet
      - Need to clarify async approaches: messaging and sequencers
   d. Replication of objects to another repository instance
   e. Full re-indexing
   f. Full integrity checks
3. Multi-node / Clustered configurations / ~~Federation~~ Capable
   a. High availability
   b. Bulk ingest
   c. High read loads
   - Note: generally need to define what clustering provides (DWD - I suggest that a cluster acts like a single installation in which system state is closely shared among the members. Clusters usually imply a common implementation)
   - Federation - (nodes have a common definition for identifiers, interfaces, formats, protocol, business semantics, and policies that permit them to interoperate but otherwise act like independent installations that do not closely share system state. Federation does not need to be a common implementation but implies common governance)
4. ModeShape
   a. Assess persistence approach (i.e. bit-level object and datastream persistence)
      i. Some backup/restore details: Backup and Restore
5. Evolution-capability - The system permits graceful (incremental) changes without having to perform replacement of large parts of the system in one step
   a. The software permits the graceful replacement of old technology with new technology
   b. The software permits the integration of new technology gracefully
   c. New content formats can be added easily, and the system permits gracefully delivering new representations for existing content
   d. New capabilities can be added or old ones replaced gracefully
   e. Underlying hardware and software infrastructures can be replaced gracefully, and the system can use advances in technology or special characteristics of its technical infrastructure without changing the core Fedora software
   f. How does the content move forward in time?
   g. How do the interface contracts move forward in time?
   h. How does the implementation move forward in time?
6. Ability to use in various integration patterns
   a. Inbound and outbound transformation
      i. Permits ingested information to be transformed so it matches the supported ingest contracts, and the same in reverse for delivery
      ii. Also used internally to support interoperation with back-end integrations particularly storage (for example S3, DuraCloud)

        iii. Overlaps the Content Enrichment pattern for feature extraction, for example loading Search and Discovery indices
    b. Content Enrichment pattern for ingest at least
        i. For example, extra meta-information can be added to newly ingested content
        ii. Another example is extraction of meta-information from inside the ingested content
        iii. A third example, is connecting content or meta-information to other related items
    c. Internal and external event driven (notification) patterns (especially external notification that an asynchronous operation is complete)
        i. Internal event driven operation is likely to be well set up
        ii. A classic external case is a front-end system needs to know when all internal operations or delegated operations are finished so the front-end system can behave is a post ingest fashion. For example, update its indexes, removed staged content, possibly remove original content. The alternative is a polling approach (both could be used).
    d. Idempotent receiver pattern - Identical ingests could be received but it should be possible to ignore duplicates
    e. Message Bridge pattern - Permits inbound messages (all RESTful HTTP API calls are messages) to signal back-end integrations, possibly outside the repository to perform functions

7. Storage Options
    a. Tiered-storage
        i. Support having all or part of the content low performance storage including copies in near-offline storage
        ii. Support having all or part of the content on offline storage (like tape - where items are not available until after staging)
        iii. Support having meta information stored in offline or near offline
    b. Support storage other than file systems and using that storage's special features
        i. Bytestream-based object stores like S3, DuraCloud or Isilon
        ii. Streaming stores for low latency, low dropout functions such as audio and video delivery
        iii. Tape
    c. Support having specialized indices particularly for locating copies, metadata or discovery data, also removal of latency
        i. Direct queries to appropriate an appropriate index
        ii. Marshall results from multiple indices

8. Preservation-worthiness
    a. These comments are based on the assumption that the only form we currently know how to preserve is a serialized form, also some features overlap, If this is not true propose an alternative
    b. Permit copies to be made, maintained and validated at one or more geographically remote locations
    c. All archivally significant data is, at some point, stored in a serialized form
        i. A. Soroka: What is "archivally significant data"?
    d. No notification that results in the destruction of the original source materials is issue until all steps of the preservation policy are executed and verified (A. Soroka: This is as much to say that Fedora's performance will always be terrible.)
        i. e.g. content progresses from a (possibly) non-serialized form, to a serialized from and n copies are made, followed by a check of essential characteristics
        ii. There is some definition of the essential characteristics of the representations that can be delivered for the unit of preservation
        iii. There is some definition of the unit of preservation
    e. Bitstream level fixity of "preserved" representations can be verified
    f. Fixity of meta information can be verified
    g. Some approach to authenticity is selected and used including at least lifecycle records (one kind of audit record)
    h. Records of system operations including configuration changes are kept (a second kind of audit record)
        i. A. Soroka: This is not feasible in the current implementation and making it feasible would require bringing configuration into the repository, a massively-non-trivial task.
    i. Repeated from Evolution above since the subjects overlap: How does the content move forward in time?
    j. How do the interface contracts move forward in time?
    k. How does the implementation move forward in time?

9. Support for graphs of related stuff (carefully avoiding saying what kind of stuff yet)
    a. Linked data
    b. Semantic databases
    c. Specific representations
    d. Named graphs

Meetings

- 2014-08-13 - Fedora Technical WG Meeting
- 2014-08-20 - Fedora Technical WG Meeting
- 2014-08-27 - Fedora Technical WG Meeting
- 2014-09-03 - Fedora Technical WG Meeting
- 2014-09-10 - Fedora Technical WG Meeting
- 2014-09-17 - Fedora Technical WG Meeting
- 2014-09-24 - Fedora Technical WG Meeting
- 2014-10-08 - Fedora Technical WG Meeting
- 2014-10-15 - Fedora Technical WG Meeting
- 2014-10-29 - Fedora Technical WG Meeting