# 2015-02-20 - Audit Service Planning Meeting

### Time/Place

- Time: 3:00pm Atlantic Standard Time US (UTC-4)
- Call-in: DuraSpace conference line
  - · 1-209-647-1600, 117433#

### Attendees

- David Wilcox
- Andrew Woods
- Nick Ruest
- Mark Jordan
- John Doyle
- Doron Shalvi Susan Lafferty
- Unknown User (escowles@ucsd.edu)
- Peter Eichman
- Matt Critchlow
- Dr. Arif Shaon
- Charles Schoppet
- Joshua Westgard
- Yinlin Chen

## Agenda

- 1. Introduction and topic summary a. Proposed Plan
  - i. Establish common understanding of function of Audit Service
  - ii. Brainstorm use-cases
  - iii. Compile initial requirements
  - iv. Summarize and Post use-cases and requirements
    - · Iterative meetings, as required
    - Set deadline for feedback
  - v. Create strawman design
    - Set deadline for feedback
  - vi. Confirm commitments
    - developer and stakeholders (verification)
  - vii. Sprint (Mar 23, Apr 13)
- 2. Use case discussion
  - a. Audit service should automatically record who updated which resource when and with which action.
  - b. Audit service should be able to include/import events that were performed external to the repository.
  - c. Audit service should be able to purge events.
  - d. Audit service should be RDF-based, and use PATCH semantics for updates.
  - e. PROV-O ontology may be better suited than PREMIS.
  - f. Audit service would ideally support map-reduce-style analytics.
  - g. Evidence of fixity checking on a "routine basis", and with logs "stored separately or protected separately from the AIPs themselves" should be available.

  - h. Fedora 4 REST API should support dissemination of event/audit information.
- 3. Workplan and timelines
- 4. Testing and validation
- Questions

### Minutes

#### What is an audit service?

- Fedora 3 audit log
  - ° Recording events that affect resources within the repository
    - Events may occur within or without the repository
      - Not everyone agrees that external events should be included
  - No particular structure or semantics
- · Fedora 4 audit service
  - Should at least have the minimal features provided by Fedora 3
  - Information should be centrally accessible
  - ° Information should be captured in RDF and should be query-able using SPARQL
    - Should be a REST-API endpoint
    - Need to collect a list of common queries

- ° Supplementary information can be added to enrich event information
- Ontology to represent event types
- Purpose
  - Problem-solving: find out when something went wrong and how to fix it
  - ° Demonstrates to external entities that you are taking care of their assets
  - Meeting ISO/TRAC specifications
  - ° Selecting repository content for archiving
  - ARL stats
- Internal vs. external events
  - $^{\circ}~$  Is the scope of this audit service the repository or the resource?
  - This needs to be discussed further

#### **Use Cases**

- 1. Audit service should automatically record who updated which resource when and with which action.
- 2. Audit service should be able to include/import events that were performed external to the repository.
- a. Migrate audit logs from F3 to F4 for example
- 3. Audit service should be able to purge events.
  - a. This could be problematic
  - b. Maybe just retaining the most recent version of a checksum for example
  - c. Perhaps certain events can be hidden from queries?
- 4. Audit service should be RDF-based, and use PATCH semantics for updates.
- 5. PROV-O ontology may be better suited than PREMIS.
  - a. Need to do a comparative analysis
- 6. Audit service would ideally support map-reduce-style analytics.
- 7. Evidence of fixity checking on a "routine basis", and with logs "stored separately or protected separately from the AIPs themselves" should be available.
- 8. Fedora 4 REST API should support dissemination of event/audit information.

#### Next Steps

- 1. Refine use cases
- 2. Compile a set of requirements
- 3. Get commitments for developers and stakeholders
  - a. Development
    - i. Mohamed Mohideen Abdul Rasheed
    - ii. Unknown User (escowles@ucsd.edu)
    - iii. Need at least one more developer
    - b. Testing/validation
      - i. Matt Critchlow
      - ii. Nick Ruest
      - iii. Joshua Westgard
      - iv. Mark Jordan
- 4. Will schedule another call after making some progress on use cases and requirements