2015-03-19 - Audit Service Planning Meeting

Time/Place

- Time: 3:00pm Eastern Standard Time US (UTC-5)
- Call-in: DuraSpace conference line
    - 1-209-647-1600, 117433#

Attendees

- David Wilcox ⭐
- Andrew Woods
- Nick Ruest
- John Doyle
- Doron Shalvi
- Unknown User (escowles@ucsd.edu)
- ~~Matt Critchlow~~
- ~~Charles Schoppet~~
- Unknown User (westgardja)
- ~~Brian Caruso~~ (will not make it, will check minutes)
- ~~Bria Parker~~
- Eric James
- ~~Michael Friscia~~
- A. Soroka
- Dr. Arif Shaon
- ~~Stefano Cossu~~ (Out of the country – I will not make it, but will check out the minutes)
- ~~Aaron Coburn~~
- Mark Jordan

Agenda

1. Review proposed requirements
    - a. "Audit service MUST store logs separately or protected separately from the repository resources themselves": is this a valid requirement, or an implementation concern?
    - b. Is it possible to fulfill these requirements via integration patterns and practices (without new code)?
2. Review proposed queries
3. Discuss implementation options
4. Outstanding questions?

Minutes

Review proposed requirements

- Write/Import
    - *Audit service MUST automatically record who updated which resource when and with which action*
        1. How do we determine the "who" in this scenario?
            - a. e.g. Hydra/Islandora use a Fedora admin account rather than a particular user account
            - b. Fedora has a mechanism for passing additional user principals into a request, so these could be used in "on behalf of" entries
    - *Audit service MUST be able to include/import events that were performed external to the repository.*
        1. e.g. Fedora 3 audit logs, external services, past events, etc.
        2. Import format would be RDF
            - a. Outputs of events (e.g. FITS XML) would be stored separately and referenced via URI.
            - b. Need a minimum set of elements that need to be present on imported events
    - *Audit service MUST be able to purge events.*
        1. If the use case is to limit the results of a query, this could be accomplished with a filter
        2. Audit service should keep a log of deleted resources, which is a separate issue
        3. For the purpose of a trustworthy repository, users should not be able to alter or delete audit history
    - *Audit service MUST store logs separately or protected separately from the repository resources themselves*
        1. This is an issue for TRAC compliance
        2. Could be configurable - needs to allow administrators to store audit history in a separate location
    - *Audit service MUST import events with RDF triples drawn from the specified ontologies*
    - *Audit service MUST ensure that all events minimally include the following information*
        1. *Event Agent*
        2. *Event Date/Time*
        3. *Event Activity*
        4. *Event Entity*
- Read/Export
    - *Audit service MUST be RDF-based, and use PATCH semantics for updates*

1. We should not use PATCH to modify events - we should use POST to add new events
2. This requirement will be removed

- *Audit service MUST provide evidence of fixity checking on a "routine basis"*
    1. Audit service should support fixity checking events, but not everyone will use fixity checks.
- *Audit service MUST support dissemination of event/audit information*
    1. Need to be able to query the service
- *Audit service MUST be able to export full logs in RDF format*
- *Audit service MUST service queries that vary by:*
    1. *Single or all resources*
    2. *Date range*
    3. *Event type*
    4. *Agent*
- *Audit service MUST provide a single search endpoint for all repository resource-related events*
- *Audit service MUST provide a SPARQL-Query search endpoint*
    1. This could be accomplished with an external triplestore

- Requirements define what the audit service should do
- Implementation is separate - the service may not be a core function of Fedora but a sidecar service that meets the requirements
- Need to move Fedora toward standards, limit custom code

Actions

- Everyone should review the current set of requirements
    - If you have a requirement that is not listed, you should not expect it to be supported in the implementation