2015-10-28 - Asynchronous Storage Meeting

Dial In Details

Date: Wednesday, October 28 2015 at 3PM EDT (-4 UTC)

Dial-in Number: (712) 775-7035
- Participant Code: 479307#
- Web Access: https://www.freeconferencecallhd.com/wp-content/themes/responsive/flashphone/flash-phone.php

Meeting Goals

- Participant's roles and commitments
- Development process
- Define scope
- Review of collected use cases and requirements
- Define timeline and milestones (what can be accomplished and when)

Attendees

- Randall Floyd
- William G. Cowan
- Andrew Woods
- Bethany Seeger
- Brad Spry
- John Rees
- Aaron Birkland
- Yinlin Chen
- Don Brower
- Nick Ruest

Agenda

1. Introduction and review of meeting goals
2. Roles and commitments
    a. Identify Stakeholders, Designers, Developers
3. Development processes
    a. Sprint cycles and duration
    b. Communication modes
    c. Process for consensus on use cases, scope, and requirements
4. High-level design and implementation discussion

    a. Review of existing design discussions
    b. Likely use of API Extension Architecture
    c. Mediated asynchronous services vs. pluggable storage
5. Review existing use cases
    a. Are they adequately described?
    b. Do we have all the use cases we need or want? Are there more?
6. Discussion of scope
7. Timelines and milestones

Related Resources

- Design - Asynchronous and Pluggable Storage
- Meeting notes: 2015-08-17 - Indiana - Amherst F4 Storage
- Design - API Extension Architecture

Meeting Notes

1. Round table of intros
2. Roles and commitments
    a. IU (Randall and William) committed to lead and provide significant developer time
    b. UNC-Charlotte (Brad) stakeholder, QA over AWS testing
    c. Duraspace (Andrew) wrangler/encourager, feedback offeror
    d. NLM (Rees) stakeholder/lurker, contribute case studies
    e. Notre dame (Don) stakeholder, contribute use cases, perhaps some developer time
    f. Va Tech (Yinlin) stakeholder, cold storage use case, AWS testing
    g. Nick R. (role with Ontario cloud service) –stakeholder, tester, perhaps some development
    h. Amrherst (Bethany/Aaron) stakeholder, Aaron has a Glacier cold storage interest, not sure about commitment level

General agreement that we'd all like to better understand ties with the API-X effort/group. We should try to cross-pollinate, communicate, understand/identify synergies where they exist, but at a distance, too.

3. Process
   a. Assume work will progress under an Agile framework, with development sprints
   b. still need significant time to gain a shared understanding before scheduling any work
   c. expect user stories/context and development work will progress asynchronously

4. High-level design
   a. Existing use cases/design discussions -- are old but still hold relevance and resonate for many at a high level
   b. API extension architecture – relationship with API-X/F4 extension architecture could be strengthened, but agree it is a suitable framework to start but will likely evolve as requirements emerge
      i. Woods emphasizes F4's slim code core philosophy
      ii. Why asynch storage not core? Aaron B. some interactions with specific storage options may need more direct integration with F4 core, but other pieces of a plugable API may not; requirement details should start to reveal these divergences
   c. Mediated services vs. plugable storage – related to API-X. Randall making a distinction between the heavy lifting needed by any storage solution vs. F4 REST services/messaging interactions

5. Use cases
   a. Aaron–API-X experience demonstrated that use cases generally need some common structure and better task/outcome definitions to properly generate evaluation criteria
   b. Action items:
      i. Randall will look at API-X templates as inspiration to create some for this group
      ii. Attendees will look at current use cases with a fresh eye to ensure their needs are being met; Woods encourages use of the wiki's 'like' feature for lurkers or the uncommitted so at least your voice is recorded–it matters

Sidebar: Randy asks if perhaps Brad Spry has already done some of this plumbing with S3?

Brad Spry: Yes, in F3 for datastreamStore, by mounting S3 to Linux filesystem using YAS3FS.   datastreamStore's I/O characteristics are asynch friendly. I /O transactions/characteristics are too fast for F3 objectStore and resourceIndex.   Performance gains could be had if Fedora's read/write block sizes could be fine tuned, for example from 4k to 128k for datastreamStore. Matter–issue is how to fine tune Fedora in this regard.

More Info: UNC Charlotte's Islandora Deployment and System Schematic

7. Timelines and milestones

- Expectation is no development sprints until 2016
- Also have potential dependency on API-X development schedule

Meeting schedule: every two weeks, same day/time, to keep the ball rolling

Complete action items from #5 by next meeting.