Hyrax Working Group Breakout and Valkyrie Discussion

Agenda:

- Introduction and context (steve van tuyl Tom Johnson 10 minutes)
  - We’d like to come out of Connect with an emerging plan for Valkyrie/Hyrax
  - There is a bunch of work and thinking that has been done thus far, but we need to ensure that we’re taking the right approach
  - https://wiki.duraspace.org/display/samvera/Valkyrie+Discussion+2018-10-04
- What has been done? (30 minutes)
  - Valkyrie (Trey Pendragon)
    - Storage differences btwn valkyrie’s fedora 4 adapter vs. hyrax’s use of fedora:
      - valkyrie does has members differently than hyrax.
      - valkyrie doesn’t use indirect containers; it puts the triples straight on the graph
      - object > binary > metadata about binary (hyrax) vs object > metadata profile > binary (valkyrie); it’s really hard to flip this around in valkyrie
    - One idea would be to have read-only adapters meant for migration purposes. this would be a much easier way forward here.
    - do institutions want to stay on fedora and use a hyrax-y fedora adapter going forward
    - Tom: a couple possible approaches to moving folks over
      - Ship with an immediate / abrupt code cut-over and expect everyone to update their applications
        - use a read-only fedora-hyrax adapter; ship it and expect everyone to upgrade right away; whenever something gets read out of fedora in the old model, replace it with a version running through the new metadata adapter, this could be a data migration over time, with an immediate code migration
        - existing writes through active fedora are immediately breaking (lead to data loss)
      - Allow the fedora-hyrax adapter to be read/write, leave active fedora in place; allow adapters to decide when they want to make the code cut-over. Allow code migration to be step-wise
    - Why can’t I just power valkyrie up with active fedora? V has features not available in active fedora like arbitrarily deep nested objects and ordered properties
    - Let’s not call it valkyrie-hyrax!! because it is really more like a valkyrie-AF or valkyrie-pcdm
  - Valkyrie on Hyrax v.1 (Justin Coyne)
    - A 4-week effort which got valkyrie in by find/replacing all uses of active fedora. This code is out of date since collections work was merged
    - One difficult piece was versioning; only fedora has a "version this" button. They implemented in hyrax by saving the new resource directly, maintaining pointers, etc.
- Where do we go? (15 minutes)
  - CHECK IN; should we:
    - Continue discussion;
      - a wide expectation that Valkyrie is coming to Hyrax may exist in the community
        - is there an easy way of learning? poll/survey
          - give respondents options weighted across a variety of real-world paths/scenarios (around time/money/quality, code vs. data migration, etc.)
        - some members will not migrate (due to their resources) until there is something like Valkyrie
    - Break-out technical subgroups?
      - Whiteboarding
      - Hacking
  - Report back (last 20 minutes)
    - What were the concrete outputs from today?
    - What are the next steps?
    - What, if any, additional get togethers should there be at Connect?
    - What commitments do we need to accomplish this?
      - What should our ask be for the rest of the conference?

Q: It sounds like people may be comfortable moving away from PCDM is that true?
  - Princeton comfortable following it in local modeling; could easily serialize this way if we shared our data.
  - Application behavior is the question much more than the storage layer – pcdm a concern of an application (e.g. hyrax itself could guarantee /validate the pcdm-ness of data)

Q: Does anyone have any other use case for a read/write valkyrie-pcdm adapter other than the gradual migration case?

People use active fedora directly in their applications to support a variety of features which will be work to migrate.

Migration options:

- lazy migration one object at a time as objects are accessed
  - con - concern that the migration can languish and some objects might never be migrated
- batch lazy migration of 1000 objects on a nightly basis
  - pro - that we can track the progress on the migration
  - full migration
    - con - concern about time to migrate

**Development Approaches:**

**Low Bar**
- create a read-only activefedora-adapter
  - read using active fedora gem
  - use fedora-adapter or any other adapter to write out
  - con - this is abandoning current active fedora model which may or may not be tenable for the community

**Medium Bar**
- side-by-side approach with both current activefedora gem supported at the same time that parallel code is written supporting valkyrie read-write.
  - controllers and models will be created that use the read-only activefedora-adapter.
  - continue to support activefedora gem code
  - if required, extend activefedora-adaptor to be read-write
  - once valkyrie approach is complete, remove code using the activefedora gem approach. Complete is defined as...
    - EITHER read-write activefedora-adapter approach is completely functional
    - OR if it is decided that read-write is not required, once read-only activefedora-adapter AND parallel code for using other write adapters is complete
  - can use feature-flipper approach
    - pro - migration is not a line in the sand and development can respond to deprecation notices over time
    - Trey - use generators to generate one or the other; change is at controller level
    - con - can make hyrax even more complex as both approaches are supported at the same time
    - con - greatly increases the number of tests to run since you have to test both approaches

**High Bar**
- create a read-write activefedora-adapter
  - pro - makes data migration optional
  - pro - allows apps to use an alternate adapter
  - con - hard to create the write part of the activefedora-adaptor

**Challenges for All Bars**
- The various bars address data migration, but all bar options require code migration for any customizations by apps.

**SURVEY VOLUNTEERS:** The survey should include questions proposed by Tom Johnson and Steve van Tuyl. Let's document the survey process, and make it available on the samvera wiki, so we don't keep doing them ad-hoc.

Andrew Myers
Phil Suda
David McCallum
Brian McBride