March 2017 Developer Congress Meeting Agenda

Logistics

- **Date/time:** March 30-31, 2017 - 9am to 5pm
- **Location:** Bechtel Conference Center at Encina Hall (616 Serra Street), Stanford University - Main Hall
- **Facilitator:** Carrick Rogers
- **Registration form:** https://hdc-march-2017.eventbrite.com
- **Lodging and Transportation:** https://library.stanford.edu/projects/ldcx/2017-conference/location-and-travel

Note: please consider attending and separately registering for LDCX on the preceding Monday through Wednesday.

Goals

- Focused face to face time for Hydra developer community
- Community code exchange
- Move community development goals forward

Dev Congress Rough Schedule

Thursday AM

8:30 am to 9 am: Coffee and introductions at the Betchel Center

9 am-10:30 am: Start of Conference

**Fast Introductions**

**One Minute Curated Session Pitches**

- Adding File Upload to Valkyrie, Trey Pendragon
- Putting Active Encode into Hydra Derivatives, Chris Colvard
- Composable Jobs, Adam Wead
- Organization of Documentation Working Group, specifically: Documenting Gated Discovery in Hyrax, Documenting overloading of Roles Term, An drew Myers
- Making it easier to add attributes to your models (can we collapse Lynette's Tutorial, can we make a generator, etc?), Facilitator Needed
- Creating a ContentDM to Hyrax/Hyku Migration Gem, Braden
- Implement Display Sets Lynette Rayle

**Plenary Topics** (10 minutes max per topic, if further discussion is needed a group can be spun off)

- Any opposition on the promotion of Question Authority to Project Hydra from Labs
- Any reservations to the merging of the Linked Data branch merging into Question Authority
- Merging the Actors refactoring PR
- Any other long standing PRs we need to consider during this conference
- Do we want a standardized rubocop profile
- If we had commitment from our administrators what would a group that works just on hydra looks like (core group/sprints suggestion)
  - We need Higher visibility of demand for cycles and the supply of cycles, which a project hydra roadmap from management and development would give us
  - We need Clearly defined and advertised current work and next work
  - We would use working groups to clearly define and refine the requirements of a new feature
  - We would use sprints to implement and do work
  - We prefer 1-2 week sprints over longer partial time commitments for accomplishing work
  - We define the following sprint types with the deliverables they would provide
    - Documentation: Documentation and/or Test [code examples in the document should map to tests]
    - Release Sprints around release candidates: Quality Assurance Scripts, Release & migration documentation
    - Maintenance/ Bug fixing: Report on closed bug
    - Feature: Feature Code, Testing, and documentation
    - Refactoring: Report
  - A sprint team should include
    - Product Owner
    - Senior Developer
    - Developers (could include inexperienced developers at in-person sprints for onboarding)
    - Documenter
    - Quality assurance
  - Plugin working group report and discussion/improvement of the white paper
    - Here are the guidelines: https://github.com/projecthydra-labs/hydra_plugins_wg
    - Here's the WG charter info: Samvera Plugins Working Group
    - Here's some work we've done for "pluginifying" GeoConcers. It's the "plugin" branch of the (now old) GeoConcerns repo: https://github.com/projecthydra-labs/geo_concerns/commits/plugin
  - Report on Display Sets Working Group and feedback – Requirements - Display Sets
  - Any spots to call out for refactoring
    - Reform
• External URIs in Fedora - Handle asynchronous request using action cable
  • Need push notifications
  • Refactor our approach to Can Can abilities
  • Different Auth Library instead of cancan
• Can we say we are on rails 4?
  • Hyrax 2 Rails 5
• Additional discussion items or topics
  • Better on ramp for new developers - website
  • Redoing the website

10:30 AM to 11:40 AM: Group Work

11:40 am to Noon: Groups report on morning work

Thursday PM

1 pm to 5 pm: Continuation of Group Topics

5:15 PM: Dinner and Drinks At Lathrop Courtyard

Potential Topics

• Moving from Sufia and CurationConcerns to Hyrax
• Can we stop using the “include MyControllerBehavior” pattern in controllers/models? Justin Coyne
• Assess current ability to search and deposit works in Hyrax via built-in APIs
• Actionable work from FileSets WG, Plugins WG, or Display Sets WG
  • Pluginizing GeoConcerns and Google Analytics (current efforts by the working group)
  • WGBH's on going plugin development for PREMIS modeling and external (non FCR) storage of files
  • Pinging Stefano Cossu, Andrew Myers, Lynette Rayle
• Extract features into Hyrax (from Hyku or your/another app)
• ArcLight/indexing archival description
• How do we produce stable releases of Sufia and or Hyrax
  • More broadly, how do we sustainably develop software as a community given the relative lack of investment in shared development and shared quality assurance?
  • Please help build out this document listing the workflows available in Sufia and Hyrax: https://docs.google.com/document/d/1WUyYdbU4rCs62HHRmhe2_SXvHSbv1IJ2M6Un32yck3s4/edit?usp=sharing This can help us better distribute quality assurance testing at time of release, by being clear about what we're testing and how to test it.
• Compare the Notre Dame (Bento) approach to asynchronous storage with the IU approach to find best patterns and practices. ( Jeremy Friesen, Randall Floyd, Andrew Myers)

Development Ideas

• A sentence or more about your development idea, and how long you think it will take to implement with a team of ~5 developers. (Your name)
  Take a look here for more ideas: https://github.com/projecthydra/hydra/wiki/Hackfest-ideas and February Hydra Developer Congress

1. Implement a new ordering approach for PCDM members (Esmé Cowles). One of the pain points of implementing PCDM is poor performance in Fedora, especially around ordering large numbers of members. Hopefully taking input from discussions at LDCX, implement an alternate to the current linked-list approach.

2. Hydra Plugins stuff (Andrew Myers)
   a. Guidelines - this is mainly documentation. For each guideline, we'd like to provide a rationale as to why we think the guideline is needed. Also seeking feedback on structure and readability of the document, as it's supposed to serve as a reference for plugin developers. It would be great if we could get enough feedback and +1 to release v1.0 of the guidelines, if we haven't yet already.
   b. Pluginizing GeoConcerns - we've made some headway here already. If there's more to do at the end of March, we can look at it. If we're already done, then it can serve as a good case study.
   c. Pluginizing Google Analytics - less progress on this so far, so there may still be some work to do by end of March.
   d. Preservation Event Plugin - this is what we've been working on at WGBH and IU. Much of the functionality is there, but have yet to go back and ensure it follows the guidelines.
   e. Refactor Hyrax for better integration points - One of the things we hope to come out of our pluginizing, is identifying some interfaces in the stack that could be refactored to become better integration points for plugins. For instance...
     i. Derivative generation - when pluginizing GeoConcerns, there is a lot of logic around customizing how derivatives are created. We highlighted some interfaces within the stack that could provide better integration points.

3. Asynchronous storage (Randall Floyd, Daniel Pierce, Nianli Ma, and/or Andrew Myers)
   a. This is an effort by WGBH and IU to use Fedora's APIx framework to create a well-defined interface for communicating with tiered storage systems.
   b. Randall Floyd, Daniel Pierce, and Nianli Ma probably have a better idea of how the dev work re: APIx can be broken down into doable chunks.
   c. Hyrax (and lower in the stack) will also need to be refactored in places to provide UI for asynchronous interactions. Of course this depends on the asynchronous storage proxy api (see above) being sufficiently stable.

4. User notifications via ActionCable (instead of long polling) Justin Coyne
5. Rubocop rules - Make a hydra-wide Rubocop ruleset and utility code to install and keep it up to date.  
   Tom Johnson
   a. We have had problems with failing builds when Rubocop is updated, having a basic cop config shared across the project would prevent these kinds of breakdowns at the cost of some maintenance overhead.
6. Implementation of Display Sets - I am hoping that the working group will have requirements and mockups complete in time for LDCX such that implementation could begin as part of the developers congress.  
   Lynette Rayle
7. Look at Valkyrie (https://github.com/tpendragon/Valkyrie), explore building a buffer layer between our frontend engine (Hyrax) and data persistence.  
   Trey Pendragon
8. Try using Reform instead of our own form objects.  
   Trey Pendragon - do valkyrie instead at this congress
9. Push model validations into our forms.  
   Trey Pendragon - do valkyrie instead at this congress
10. Identify places to refactor that a group could come back to.  
    Trey Pendragon
11. Document best practices around actors, provide an easy-to-find place to inject your own actor.  
    Trey Pendragon
12. Organize and charter a group to tackle refactoring and other feature-less sprints on core gems  
    Trey Pendragon
13. Document (maybe a tutorial) how gated discovery works in Hyrax, think on how to simplify.  
    Trey Pendragon
14. Architecture solution for jobs in Hydra/CC/Sufia to allow "composable" jobs. Currently, jobs are defined in the gem and aren't extendable or mixable in any way so you have to override the entire class to make small changes. Default jobs could be in the gem, but the consuming application could define its own by reusing the existing defaults and extending/mixing where needed.  
    Adam Wead
15. Write some more shared specs for composable objects in Hyrax (See https://github.com/projecthydra-labs/hyrax/blob/master/lib/hyrax/specs/shared_specs/derivative_service.rb)  
    Trey Pendragon
16. Start mainstreaming our nascent technical onboarding documentation and addressing unresolved issues. (Suggested by Michael J. Giarlo; maybe Esmé Cowles is game?)
17. Make it easier to add new properties to your model.  
    Trey Pendragon
18. Dropbox has deprecated its v1 API. BrowseEverything will need to support it, but there are problems https://github.com/projecthydra/browse-everything/issues/159  
    Adam Wead

Development Ideas, Prioritized (for later use, pls do not edit)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Well Defined</th>
<th>Achievable by Friday</th>
<th>Feature/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>