Sufia/CurationConcerns Consolidation Plan

- Introduction
- Timing
- FAQ

Introduction

The Samvera Architecture Working Group produced two white papers in advance of Hydra Connect 2016, one of which was "Should Sufia and CurationConcerns Merge?" At Hydra Connect 2016, there was a well attended session to discuss these white papers. The community discussion of the Sufia/CC-oriented white paper at Hydra Connect was rich and spirited, and largely validated the costs and benefits highlighted in the white paper. Users of Sufia and CurationConcerns want more features in their applications, many of which should live in core gems rather than in institution-specific applications. They also want more control over what features they get “for free,” and how those features are added or removed, enabled or disabled. While a good number of the features that Sufia provides are slated for extraction into CurationConcerns, some are prime candidates for extraction into new, optional plugins. The advantage of a plugin architecture is that those who want the functionality can opt in by including it in their applications and those who don’t can avoid the cost of maintaining code they don’t need.

There was significant discussion of the mechanism by which Sufia and CurationConcerns might be merged, during which we concluded that “merger” is a misnomer for what is being proposed. We do not propose merging CurationConcerns into Sufia, and we do not propose merging Sufia into CurationConcerns; rather, we propose creating a new gem with a new name that:

- Contains all of Sufia and all of CurationConcerns, side by side, at first;
- Becomes a single codebase gradually, containing the features that Sufia users and CurationConcerns users want to see; and
- Integrates cleanly with more specialized features that are either extracted into plugins, or placed behind toggles that make it simple and straightforward to disable them

While this idea resonated with the audience and seemed to be “the sweet spot” for moving beyond our current bifurcated state, several people voiced concerns about the timing of such work and about the pains of keeping Hydra applications up-to-date, especially given the churn we’ve experienced over the past year-and-a-half which is just now starting to settle. We aim to do this work in a way that:

- Does not force any data migration from the latest releases that are available at the time the new gem is made available;
- Focuses on consolidating features already implemented in Sufia and CurationConcerns into a single codebase, while avoiding unnecessary refactoring unrelated to the consolidation; and
- Involves developers with Sufia and CurationConcerns applications already in production who can test the upgrade path.

Ultimately, the consolidation of Sufia and CurationConcerns into a single gem is intended to increase the Hydra community’s capacity to share features and collaborate on source code more efficiently. As we do this, we ought to consider new techniques for providing more stable, better managed gems, such as more consistent API documentation; a formal release schedule, which provides more foreknowledge of how and when our gems will be changing; more thorough integration testing before major releases; and increased discipline with regard to how we version our releases.

Timing

After Hydra Connect 2016, the Hydra-in-a-Box team discussed the Sufia/CC consolidation with the Hydra community, and what the timing for this work means for the Hydra-in-a-Box project’s objectives. The Hydra-in-a-Box team allocated resources to the consolidation in December 2016 for a few reasons:

1. It leads to a simplified Hydra codebase sooner, which will make it easier for developers contributed to community sprints in the early months of 2017 and beyond. The Hydra community wants community sprints to be more productive and less disorienting and this gets us on that path sooner.
2. Allocating resources to the consolidation slows feature work – which has been happening at a rapid pace – and allows time for current work in Hydra-in-a-Box, Sufia, and CurationConcerns to be better tested and QAed, and also allows time for further design work to be vetted and finalized.
3. Starting the consolidation earlier creates more of a buffer between when the consolidation work happens and when a hosted Hydra-in-a-Box pilot service is planning to launch. The team wants to avoid a situation where we are still in the throes of the consolidation when we need to be available to the pilot program.

This work is being undertaken without substantially disrupting current work in Sufia or CurationConcerns. This will not affect the master branches or existing feature and release branches in either Sufia or CurationConcerns; this work is happening in a new repository, in parallel with existing Sufia- and CC-based work.

FAQ
What will the new gem be called?

Hyrax!

A number of folks in the Hydra community have wanted to call a gem 'hyrax' for a long time (partly because it starts with 'hy'), and the gem name was available. A repository has already been created for it in the projecthydra-labs organization and we've taken the liberty of reserving the gem name.

What does this mean for my current Sufia or CurationConcerns application? Will I need to migrate my data?

That depends on what you're upgrading from. The work to consolidate Sufia and CurationConcerns pulls in all the latest code from each gem. This includes code from CurationCurations that changed the directionality of membership relationships and this will require applications based on CC 0.x or 1.x to run a data migration to reverse the direction of membership relationships. These changes have been released as CC 2.0.0, and are already included in Hyrax.

Thus, if you are upgrading from a CC 2.x-based application, a data migration will not be required. If you're upgrading from a CC 0.x- or 1.x-based application or a Sufia 7.x-based application, you will need to run a migration to change the directionality of membership relationships within your repository to avoid the gnarly Fedora performance issue. Though no Hydra partners have yet committed to this work, we anticipate early adopters of Hyrax who are migrating from CC or Sufia to help define and smooth over the upgrade path with documentation and/or tooling that the rest of the community can use for their upgrades.

I'm ready to migrate my application from Sufia or CurationConcerns to Hyrax. What version of Hyrax should I target?

You should use the most recent 1.x version that is available. We will be creating a 1.x-stable branch to cut 1.x releases from precisely so that folks coming from Sufia or CC could have a stable target for their migration. This branch will also contain merges of the latest work from Sufia and CurationConcerns, so that if you come over from Sufia 7.3.x or CC 2.0.x, you have access to all the same features and configuration. (The latest Sufia and CC work is also merged into Hyrax's master branch to ensure that Hyrax releases greater than 1.x also have the same code.)

Using a dedicated branch for migrations makes a clean separation between Hyrax as a migration target from Sufia and CC, and as a location for continuous advancements – for, e.g., the Hydra-in-a-Box project.

I'm developing, or planning to start developing, an application based on Sufia or CurationConcerns... should I wait for Hyrax?

There are Sufia and CurationConcerns users who will reasonably plan to upgrade from the latest Sufia or CC version to Hyrax, and the time-honored practice has been for early adopters to help smooth over the upgrade path. So in short, you do not need to wait for Hyrax 1.0 to start doing your development; you will not be left behind if you adopt Sufia 7.x or CC in the meantime, and there are many other community members who will be on the same path as you.

When will Sufia and CurationConcerns be consolidated into a single codebase? And what will happen to Sufia and CurationConcerns afterward?

The consolidation work started in late 2016, and Hyrax is now available via GitHub for early beta testing. See the Timing section above for more information; we are planning to release 1.0.0 in February, 2017, shortly after Sufia 7.3.0 is released. Updates on this work will be provided regularly via e.g. Slack and Hydra Tech calls.

Even after Hyrax 1.0.0 is available for usage and testing, the Sufia and CurationConcerns repositories will hang around for some time while the community upgrades away from Sufia and CurationConcerns. Given past timelines, it is likely that the codebases for Sufia and CurationConcerns will remain available for many months, possibly more than a year, allowing the community to produce hotfixes and patch releases for both gems while they work through upgrade planning.

When should we start using Hyrax?

If you have the flexibility – meaning, if you have the time and the patience to be an early adopter – we'd suggest starting with Hyrax as soon as you can primarily because the new codebase could use more eyes on it. A more substantial answer to this question is going to be different for everyone based on where they are with their projects, what resources they have to throw at development, and what timelines they're committed to.

How does the consolidation relate to the Hydra Plugins Working Group?

The developers working on the consolidation will work with the Samvera Plugins Working Group to solicit a recommendation on a plugin architecture that can be used to extract social features (many of the ones marked with 🏛) into an optional plugin for Hyrax. We are also eager to determine how best to integrate concerns-based gems, using GeoConcerns as a testbed, with Hyrax, and would welcome a close collaboration with the Hydra Plugins Working Group if they have cycles to contribute to this work as another concrete way to test their recommendations.

What does this mean for the Hydra-in-a-Box repository application (Hyku)?
The Hydra-in-a-Box team is using its repository application, Hyku, as a testbed for the consolidation; indeed, it is already (as of early December, 2016) based on Hyrax.