Samvera Plugins Working Group

NOTE: The Call for Participation (CfP) for this working group was open from October 17 - November 1, 2016. Any further changes to this charter will be made only by consensus of the working group's active members.

NOTE: The Call for Participation (CfP) for this working group was open from October 17 - November 1, 2016. Any further changes to this charter will be made only by consensus of the working group's active members.

STATUS
This group has completed its work and has been retired.

Scope & Objectives

This working group is the result of discussions during the 2016 Hydra Connect in Boston - specifically the un-conference sessions that took place on Thursday, October 6th: “Building Apps with CurationConcerns” and “Architecture Working Group - Feedback on whitepapers”.

Several community members expressed use cases for a Hydra head that has all of the features of CurationConcerns, and some, but not all, of the features of Sufia.

Most of the features in Sufia are not easily turned off, and so some implementers of Sufia just keep the unused features and train their end-users to ignore them.

Conversely, there is no standardized way to port features from Sufia to CurationConcerns, and so implementers of CurationConcerns who wish to have some of the features from Sufia are left doing their own patches, which tends to require a fairly deep knowledge of the stack.

Also, developers who wish to create reusable plugins for CurationConcerns are left making their own decisions on how their plugin should interface with other layers of the Hydra stack, because those integration points and interfaces either do not yet exist, or are not yet well-documented.

Finally, there has been talk about a reunification of the Sufia and CurationConcerns code bases. If we had a more clearly defined plugin architecture, then the distinction between Sufia and CurationConcerns may become less important - i.e. “Sufia” may end up being “CurationConcerns” + “additional features”, which may be a combination of plugins and default configuration that “turns on” built-in features that would otherwise be turned off automatically.

Use Cases

From the discussions above, the following use cases have been created to better articulate specific outcomes we hope will result from this working group.

Nouns used in these use cases...

- **A plugin** is software that can be optionally installed into Hyrax.
- **The documentation** is the sum of documents, tutorials, etc. that explain the guidelines for developing plugins.
- **A developer** is one who is writing code for plugin.
- **An implementer** is one who installs and/or manages the installation of a product within the Hydra stack for production use.

1. The software expectations of plugins are well-documented for the developer.

   As a developer who is reading the documentation for developing plugins, I know the expectations that the CurationConcerns software will have of my plugin so that I do not develop a plugin that will break a Hyrax instance by violating those expectations.

2. The community expectations of plugins are well documented for the developer.

   As a developer who is reading the documentation for developing plugins, I know the expectations that the CurationConcerns software will have of my plugin so that my plugin is sufficiently documented, and can be installed, configured, and used by others with a known amount of effort.

3. The interfaces and conventions for plugins are well documented for the developer.

   As a developer who is reading the documentation for developing plugins, I know the interfaces and conventions within the Hydra stack that I must/should/may use when developing my plugin.

4. Installing and configuring a plugin is predictable and straightforward.
As an implementer of CurationConcerns, I can expect that the installation and configuration of a plugins will be similar to other plugins so that installing new plugins is a predictable process with minimal overhead.

**Deliverables & Timeframe**

*February 14, 2017 April 11, 2017* is the sunset date for this working group. By that time, we hope to have the following artifacts:

1. All the guidelines, tutorials, code changes, etc. that are needed to satisfy the use cases expressed in the Scope & Objectives section above.
2. A working plugin that adheres to the established guidelines.

**Meeting Times & Communication Channels**

- **When:** TBD after membership is determined (see below)
- **Where:** TBD after membership is determined (see below)
- **Mailing list:** samvera-tech@googlegroups.com
- **Chat:** Project Hydra on Slack ([project-hydra.slack.com](http://project-hydra.slack.com)). Use #dev or #general channels.

**Meeting Agenda and Notes**

- 2016-10-13 Meeting notes - Hydra Plugins Working Group
- 2016-11-08 Meeting notes - Hydra Plugins Working Group
- 2016-11-22 Meeting notes - Hydra Plugins Working Group
- 2016-12-06 Meeting notes - Hydra Plugins Working Group
- 2016-12-20 Meeting notes - Hydra Plugins Working Group
- 2017-01-10 Meeting Agenda and Notes
- 2017-01-17 Meeting Agenda and Notes
- 2017-01-31 Meeting Agenda and Notes
- 2017-02-14 Meeting Agenda and Notes
- 2017-02-28 Meeting Agenda and Notes
- 2017-03-14 Meeting Agenda and Notes
- 2017-04-11 Meeting Agenda and Notes

**Members**

1. Andrew Myers, facilitator (WGBH Educational Foundation)
2. Eliot Jordan (Princeton University Library)
3. Carolyn Cole (Penn State University)
4. Greg Kostin (University of Michigan)
5. Darren Hardy (Stanford University)
6. Noah Botimer (University of Michigan)
7. Katherine Lynch (University of Pennsylvania Libraries)
8. Matthew Barnett (University of Alberta)
9. Jenn Colt (Cornell University)
10. Chris Colvard (Indiana University)

Note that Working Groups must have participants from three different Partners. All members of a working group producing software must be licensed Hydra contributors covered by the appropriate CLAs. Other types of contributions such as requirements, design, best practices, documentation, etc. do not require CLAs but participants should accept that the materials to which they contribute may be released under a Creative Commons Attribution 4.0 International License.

**Resources**

- **Built-in Feature Checklist - Merge Working Document** A checklist of features between Sufia and CurationConcerns, which should be merged into the core code base as "optional features", and which should be maintained as plugins.