

## **Extending the Hydra Head to Create a Pluggable, Extensible Architecture: Diving into the Technology of Hydramata**

### **Abstract**

What makes for a flexible, extensible architecture? The complexity of our Digital Repository, Institutional Repository, and research data needs were pushing the limits of our existing systems. Faced with decreased resources and ever increasing demands for flexibility, interoperability, and extensibility, we realized we could not solve our problems alone. As members of the Hydra community, we were already addressing this problem to some degree. However, many Hydra partners were also utilizing aging Institutional Repository systems, or maintaining a variety of Digital Repository applications that do not interoperate well. As a result, six institutions within the Hydra community<sup>1</sup> banded together to form a cross-institutional team of subject matter experts, developers, librarians, project directors, and product owners focused on further extending Hydra to create a more sustainable and flexible digital platform that moves beyond the Institutional Repository, while not sacrificing sustainability for innovation. If you are a repository manager, developer, or anyone else faced with these challenges come for a deeper look at our architecture decisions and tools.

### **Transcending the Institutional Repository**

Our use cases and needs are pushing beyond the traditional definition of an Institutional Repository. This requires us to be more integrated and connected to other repository systems, information networks, and systems like CRIS's. We also need tighter integrations with our publishing platforms like journals and digital exhibits. Additionally, as research and scholarship increasingly involves mixing formats and disciplines, it was imperative that we support multiple formats coexisting while supporting the wide spectrum of use cases of our campus faculty, students, and other researchers ranging from research data, image collections, theses and dissertations, video, and library owned collections.

### **What Drives our Architectural Decisions?**

To really sustain each other it became clear that we needed an architecture that could provide a full working solution out of the box; be flexible, scalable, and modular; and be extensible for the future. The Hydra project has now reached a level of maturity that several robust Hydra Heads are available on the Partners and participants public github sites. But this does not mean that code sharing is always simple and straightforward, especially when trying to integrate code into an existing Hydra Head. Developers may find themselves installing a complete Hydra stack / application from another institution, and spending considerable time tearing it apart trying to find and pull out the code of interest for potential reuse and integration with their own project's code.

The Hydra Partners have made a strategic goal to address this issue with a process of 'gemification' to pull out our core code as separate Ruby on Rails gems that can be integrated into a different Hydra stack. Hydramata proposes taking this concept to the next level, so that our code does not depend on a single monolithic core gem, but our ingest, workflow and discovery processes are separated, by format, to make more discrete code sharing feasible.

---

<sup>1</sup> Indiana University, Northwestern University, the University of Cincinnati, the University of Notre Dame, the University of Virginia, Data Curation Experts

## Hydramata or Separate Hydra Heads?

This architectural design not only makes our Hydra code more sharable, but offers the benefit that a Hydramaton can either stand alone as a simple Hydra Head, be easily integrated into another Hydra Head, or utilize the out-of-the-box full solution. It should be possible, for example, for another Hydra participant to install our Hydramaton for data, and ignore the Hydramatons for images, ETDs, and audio-visual (time-based) media.

## Hydramata in Action

Figure 1 shows an example Hydramata Hydra Head, with separate Hydramatons by Content type (for example, ETDs, video or images), with code separated for the functions of metadata, viewers, workflow, discovery and dissemination.

We will demonstrate one or two example Hydramatons within our existing code base (for example, data, images and/or ETDs) and point to the areas in our github repository where the discrete code for these Hydramatons exists.

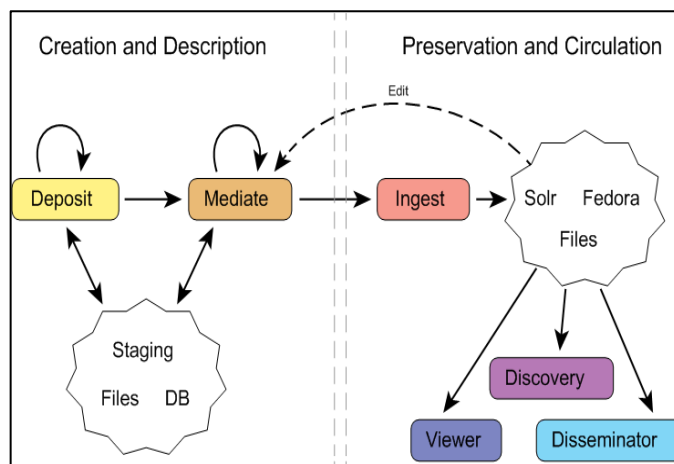
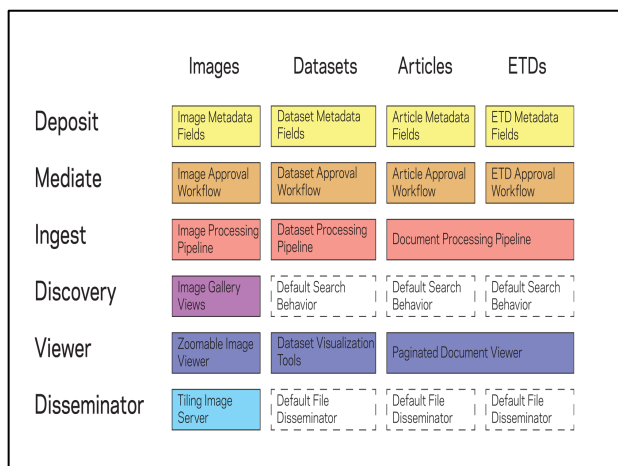


Figure 1. Sample Hydramatons in a Hydra Head

Figure 2. Discrete Components (Hydramata)

## Discrete Hydramata

Figure 2 shows the modular components of each Hydramaton, and the smaller discrete Hydramata to perform tasks like deposit, review, publish and display. Not all content types will require a Hydramata at each level, and they will certainly differ -- the discovery layer for images would include Gallery Views that don't apply to datasets or ETDs.

We will give an overview of discrete Hydramata that are included in this code, to cover processes like ingest, mediation, and discovery, and how they interplay with core Hydra code.

## Conclusion

Because we can never anticipate everything that will need to be done to advance, preserve the work of our colleagues, it is critical we continually invest in our community in order to sustain our efforts. In turn, it is vital our architecture and tools are flexible and nimble to enable us to move quickly to exploit the opportunities of tomorrow.