

Road Tripping to Valhalla

Samvera's Technical Move via Valkyrie to Storage Agnosticism

Jeremy Friesen

[2023-12-08 Fri 11:30]

Outline

- 1 Introduction
- 2 Object Relation Mapping (ORM) Patterns
- 3 Where To Now
- 4 Community Effort
- 5 Where We're At

- 1 Introduction
- 2 Object Relation Mapping (ORM) Patterns
- 3 Where To Now
- 4 Community Effort
- 5 Where We're At

About Me

Name Jeremy Friesen

Pronouns He/Him

Role Senior Lead Software Engineer at Software Services by
Scientist.com

Email jeremy@scientist.com

For over a decade, Samvera's technology has relied on Fedora for its persistence layer of digital objects (in particular metadata). It previously supported Fedora 3 and has current support for Fedora 4. But Fedora has moved past these versions, as of October 16, 2023 Fedora is on version 6, with native support for OCFL.

Plurality of Storage Needs

- There are members of Samvera committed to the continued use of Fedora; and are keen on migrating from the long-since end-of-life versions of Fedora 3 and Fedora 4.
- There are also members of Samvera that are looking to leverage Postgresql as the persistence layer for the metadata of their digital objects.
- The choice to favor/prefer either Postgresql or Fedora is based on the political and technical realities of each organization.

Fedora is not SQL

Were we looking at choosing between persistence layers of MySQL and Postgresql, the issue would be easy. SQL is a mostly standardized language and it is suitably implemented using the *Active Record pattern* (via the ActiveRecord gem that is part of Ruby on Rails).

A quick overview of two Object Relation Mapping patterns:

- Active Record¹
- Data Mapper

¹Made popular by Ruby on Rails .

Active Record pattern

The Active Record pattern is super convenient; but that convenience is at a cost. Tighter coupling of data modeling and persistence layer interactions.



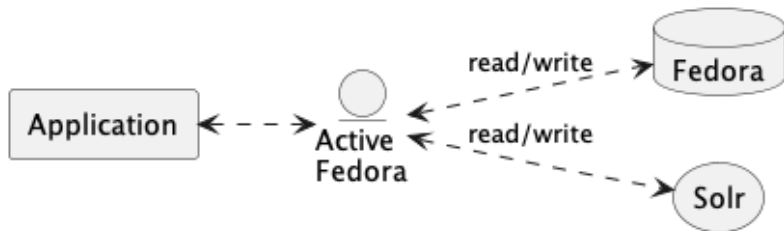
The Data Mapper pattern looks like...

In the *Data Mapper* pattern the Business Model does not know how to directly interact with the data store.



Existing Architecture


Samvera already broke the *Active Record* pattern in it's ActiveFedora implementation.



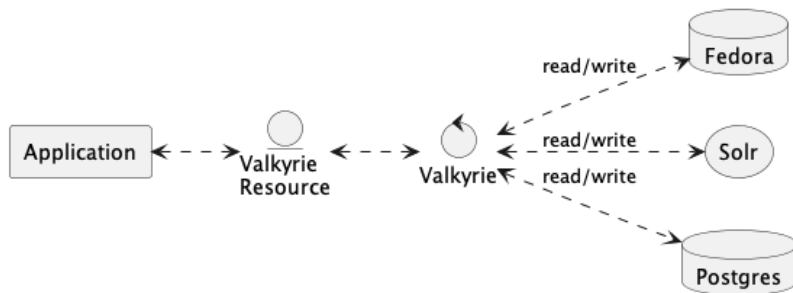
The community set about swapping out the *Active Record-esque pattern* (as implemented in the *ActiveFedora* gem) in favor of the *Data Mapper pattern*.

Avoid Long-Running Branch

The design decision being that we would **avoid a long-running feature branch**; one that is vulnerable to code drift and losing bug fixes. Instead we would do this against the `main` branch of Hyrax.²

²Credit to Tamsin Woo for architecting and envisioning this approach. 


Desired Architecture



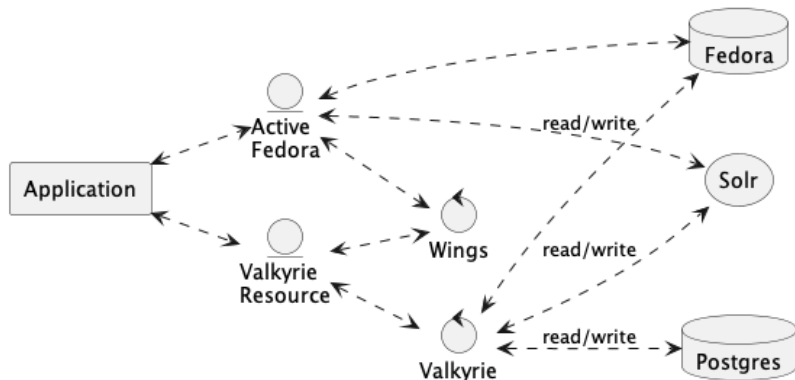
To quote an awful human being:

<expletive>, we're doing it live.

With our directional map in hand, we trod our bridging strategy named Wings.³

³Though perhaps we should've called it Bifröst after the Norse mythology bridge. 

Wings to Valkyrie




The application could use either a Valkyrie Resource or Active Fedora object; and we'd use Wings to perform the translations.

The primary purpose of the Wings adapter⁴ two fold:

- Demonstrate functional parity (e.g. we convert an ActiveFedora object to a Valkyrie Resource and back)
- Highlight inflection points for future migration.

If we could read and write to the same underlying persistence layer using the two different adapters, we could begin the slow and steady replacement of code.

⁴See Wings adapter as of Hyrax v5.0.0 release candidate 1 for implementation details. 

From the Wings documentation:

Wings is primarily an isolating namespace for code intended to be removed after a full transition to 'Valkyrie' as the persistence middleware for Hyrax. Applications may find it useful to depend directly on this code to facilitate a smooth code migration, much in the way it is being used in this engine. However, these dependencies should be considered temporary: this code will be deprecated for removal in a future release.

The Many Starts of the Marathon Relay

- In 2016 Trey Pendragon began work on Valkyrie.
- At Samvera Connect 2018 we talked about the existential crisis.
- In January 2019 we began work on the Wings adapter.
- In November 2023 we are minting release candidates for Hyrax 5, the first major version to directly leverage Valkyrie.
- At Samvera Connect 2023 Tamsin Woo ran a *Transitioning to Valkyrie Workshop*.⁵
- Sometime in the not to distant future, we'll release Hyrax 7 and remove the Wings adapter and the ActiveFedora dependency.

⁵The workshop's overview and activities are available at

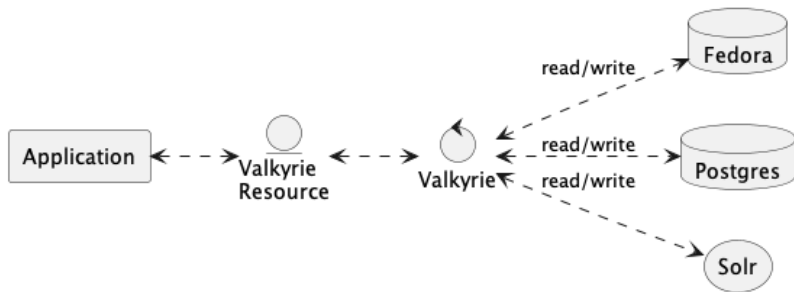
Many, Many Small Steps

The work between January 2019 and October 2023 has been consistently merged into the main code-base. We **incrementally shipped a significant architectural refactor** that was **invisible to the folks using the application**.

What Does It All Mean?

Through Valkyrie we've now performed data migrations within an application.⁶

SoftServ paired with the developers at GBH to *lazily* migrate Fedora 4 data to Postgresql. **No downtime. No major cut-over.**



⁶GBH's Archival Management System (<https://github.com/WGBH-MLA/ams>)

In moving to Valkyrie, we are increasing the durability of data by leveraging a storage agnostic interface enabling **lazy migrations within, into, and out of a Samvera-based application.**

Members of the Samvera and Fedora community know that there are many instances of Fedora 3 and 4 out there running. These are all long past their end of life⁷

At Software Service by Scientist.com (SoftServ) we are eager and ready to help folks move off of Fedora 3 and 4. We can help:

- Analyze current systems to help scope the project.
- Provide consultation support to bolster your team's efforts.
- Design and develop the software for egress.

⁷No security patches nor version updates.

Thank You

Name Jeremy Friesen

Pronouns He/Him

Role Senior Lead Software Engineer at Software Services by
Scientist.com (SoftServ)

Email jeremy@scientist.com