Earning Your Wings

Samvera Connect 2019

Tom Johnson
Hyrax Tech Lead
IT Architect/Senior Developer
Univ. of California Santa Barbara
What?
Diagram of Valkyrie’s Technical Stack
VALKYRIE IS MY BREAKABLE TOY
VALKYRIE IS MY BREAKABLE TOY
WINGS CAN BE OUR BREAKABLE TOY
What is Wings

What Wings is:

➢ A Valkyrie adapter with data compatibility with ActiveFedora
➢ Uses ActiveFedora and your existing models under the hood
➢ Provides round-tripping between Valkyrie::Resource and legacy models

What Wings isn’t:

➢ A Fedora killer.
➢ Permanent.
Where are we Now?
### Models Transformation

#### ActiveFedora to Valkyrie

```ruby
book = Hyrax::Test::Book.new(title: ['Comet in Moominland'])

valk_book = book.valkyrie_resource

valk_book.title
# => ['Comet in Moominland']
```

#### Valkyrie to ActiveFedora

```ruby
book = Hyrax::Test::BookResource.new(title: ['Comet in Moominland'])

af_book = Wings::ActiveFedoraConverter.convert(resource: book)

book.title = ['new title']

Hyrax.persister.save(resource: book)

af_book.reload.title
# => ['new title']
```
Native Models

class Book < Hyrax::Work
  include Hyrax::Schema(:basic_metadata)

  attribute :author, Valkyrie::Types::String
end
Native Models

Hydra Works Models as Valkyrie:
➢ Hyrax::Work
➢ Hyrax::FileSet
➢ Hyrax::PcdmCollection

Access Controls:
➢ Models
  ➢ AccessControl
  ➢ Permission
➢ Services
  ➢ Hyrax::PermissionManager
  ➢ Hyrax::AccessControlList (change set)

Schema Module Builder

include Hyrax::Schema(:book)

Model Registry

Wings::ModelRegistry
  .register(MyResource, LegacyModel)
What's Next

Wings is in Hyrax!

➢ 3.0.0 marks a commitment to what is already in the codebase.

➢ No longer marking Valkyrie issues/PRs as wings (except for wings adapter tickets)
  ➢ Valkyrie work in Hyrax is just Hyrax work.

Technical Next Steps

➢ Build out Valkyrie support in:
  ➢ Controllers
  ➢ Forms
  ➢ Presenters

➢ Define and document patterns for
  ➢ Versioning
  ➢ Indexing
  ➢ Business Transactions (actor stack)
What can you Do?

In your Hyrax application:

- Convert your application to Valkyrie with Wings!
  - Refactor isolated services
    - Cast to Valkyrie around the interface boundaries
  - Use Valkyrie native models

Explore the edges of support.
- Report issues
- Ask questions
- Provide feedback

In Hyrax itself:

- Review pull requests!
- Read the code!
  - Provide feedback on emerging patterns in Hyrax `app`
  - More eyes on Wings will mean more stability & better performance
- Work on Issues
  - Work on open tickets!
  - Find code that is dependent on ActiveFedora, open small tickets to rework.

Come to the code read/write breakout tomorrow(?)
Hyrax 3.0 is Valkyrie

Hyrax 3.0 is Hyrax