Work on Valkyrie
an Update
10/21/2020
Samvera Connect 2020

Trey Pendragon (Princeton University Library)
Three Audiences

1. People who have never heard of Valkyrie, or are confused about what it is.

1. People who know what it is, don’t really know why it exists.

1. People who have seen the above presentation a few times, are on board, and are about to check their email. I promise there’s new stuff! All new slides, 2020 edition.
Here’s What’s Gonna Happen

1. What is Valkyrie Anyways
2. Why is Valkyrie even a thing
3. What happened this year
4. What are we doing next / How you can help
What’s Valkyrie Anyways

1. Norse myth

2. Cool word Trey picked in an airport one day.

3. Data Mapper ORM with adapters and practices suited to digital repositories.
What’s Valkyrie Anyways

Let’s try that again.

1. Valkyrie is an ActiveFedora replacement with several adapters.

2. It splits persistence and queries from the object’s data.
   a. This means I can’t resource.save, Resource.find_by, or Resource.where(blabla).except(otherthing).includes(:stuff).destroy

3. Valkyrie lets you use Fedora, Solr, Postgres, Disk Storage, Cloud Storage, or write your own adapter quickly, but still hang out with your friends and share tips.
# What Will My Code Look Like

<table>
<thead>
<tr>
<th></th>
<th>ActiveFedora</th>
<th>Valkyrie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find a Record</td>
<td>Book.find(id)</td>
<td>adapter.query_service.find_by(id: id)</td>
</tr>
<tr>
<td>Save</td>
<td>book.save</td>
<td>adapter.persister.save(resource: book)</td>
</tr>
<tr>
<td>Delete</td>
<td>book.destroy</td>
<td>adapter.persister.delete(resource: book)</td>
</tr>
<tr>
<td>Save to Fedora/Solr</td>
<td>book.save</td>
<td>combined_adapter.persister.save(resource: book)</td>
</tr>
<tr>
<td>Save only to Fedora</td>
<td>Can’t Do It</td>
<td>fedora_adapter.persister.save(resource: book)</td>
</tr>
<tr>
<td>Save to Postgres</td>
<td>Write a New App</td>
<td>postgres_adapter.persister.save(resource: book)</td>
</tr>
</tbody>
</table>
If you want to go far, go together!
But Why

1. We can’t make up our minds.
2. Code is hard.
3. Friends are good
Can’t Make Up Our Minds

Things I heard or personally felt in 2016

1. Fedora is slow.
2. Fedora is fast enough.
3. Fedora 3 was better.
4. Elasticsearch is better.
5. Postgres is great.
6. Postgres is terrible.
7. Storing things in the cloud is hard.
8. Background workers have timeout problems.
9. We need transactions.
10. Triple stores are great.
11. Our triple store isn’t supported now.
12. We have to use MySQL.
13. Solr is great.
14. RethinkDB looks cool.
15. Why can’t my authors be sorted.
16. What about NoSQL.
17. I want to store my things differently.
18. Migrations are hard.
19. How can we use my tools instead.
Can’t Make Up Our Minds

Why can’t we just pick the right tool?

What if they’re all the right tool?

Maybe technology, institutions, and opinions are complex.

Maybe everyone’s making the best decision they can.

So let’s enable them to make that decision, test it, and share their results.

Principle 1: Make it easy to support your technology of choice.
Code is Hard

ActiveFedora works. Very few people know how. Even fewer of those are still working on it.

We use few external dependencies. Maintenance burden is all on us.

Our profession has some unique use cases not supported by wholesale usage of other libraries. (I’ll get back to this I promise)
Write Less Code

Principle 2: Use the smallest amount of maintainable code necessary to meet the requirements.

Valkyrie is 5,295 LOC. ActiveFedora is 9,039. (This is not a fair comparison. Feature set is not the same, on purpose.)
Write Less Code

Principle 3: Use external dependencies with communities whenever possible.

Valkyrie’s core is handled by dry-\* libraries, Reform, ActiveRecord, RSolr, and File. 1,240 LOC are the Fedora adapter, due to few external dependencies.
What about Hanami, ROM, etc?

I tried. Either writing a new adapter for these was too hard, they made too many assumptions about what you were doing, or they didn’t support an important use case.
What use cases make us special?

1. Our data is complicated, hierarchical, and ever-changing. Re-ordering happens constantly.

2. Several partners want a field to have two types of data. An Author might have two values - a URI representing an author, and also the string “Trey Pendragon.” Also, RDF.

3. We often change our minds about which fields go where, or how data should be represented or output. Our data needs to be flexible.

4. Order of metadata is important, but only sometimes.
Supporting Use Cases

Principle 4: Adapters can be verified as supporting the community’s needs via a suite of shared tests.

Valkyrie provides a set of shared tests to ensure that every adapter written for it supports the use cases this community has found to be important. They act as documentation and a guide to anyone wanting to make a new adapter.
The Principles

1. Make it easy to support your technology of choice.

2. Use the smallest amount of maintainable code necessary to meet the requirements.

3. Use external dependencies with communities whenever possible.

4. Adapters can be verified as supporting the community’s needs via a suite of shared tests.
Having Friends

If each of us can make our own decisions about where and how to store things, but still have the same code, we can still work together on shared solutions for our patrons.

We can still be a community, no matter what database we like.
State of Valkyrie

Software is stable!

v2.1.1

tpendragon released this on Mar 12 · 18 commits to master since this release

Changes since last release

- Add Ruby 2.7 support.
  tpendragon
- Make ID & String equality symmetrical
  no-reply
State of Valkyrie

Two stable production systems
State of Valkyrie - Hyrax Edition

Valkyrie is now part of Hyrax, headed towards full integration and needs your help! Talk to them if you want to do so.
State of Valkyrie - Thanks Friends!

1. Adam Wead
2. Anna Headley
3. Benjamin Armintor
4. Brendan Quinn
5. Carolyn Cole
6. Carrick Rogers
7. Chris Beer
8. Christina Chortaria
9. Chris Colvard
10. Chris Syversen
11. Daniel Pierce
12. David Cliff
13. Esme Cowles
14. James Griffin
15. Jeremy Friesen
16. Joe Atzberger
17. Josh Gum
18. Julie Allinson
19. Justin Coyne
20. Justin Gondron
21. Kate Lynch
22. Michael Klein
23. Michael Tribone
24. Mike Giarlo
25. Noah Botimer
26. Olli Lyytinen
27. Stuart Kenny
28. Tom Johnson
29. Institutional Administrations
30. Samvera Community!
What Happened This Year

1. Support for Fedora 6 (We didn’t do anything here, thanks for the consistent API Fedora!)

2. Released model constraints on some queries.

3. Duke released a fully Valkyrie-powered repository.

4. That’s it. It’s...been a long year.
What Next

Roadmap: https://github.com/samvera/valkyrie/projects/2

Highlights

1. Efficient solr indexing - no more duplicate fields and indexing everything under the sun.
2. Removing Draper
3. Hash support
4. Documentation
How you can help

We need a new product owner!

If your institution is implementing and wants to be a larger part of Valkyrie, get a hold of me.
How you can help

Keep your eyes out for a community sprint next year.
Contact Us

#valkyrie on Slack (https://slack.samvera.org)