



**IT'S
2019**

**DO YOU KNOW WHERE
YOUR METADATA IS?**

Oregon Digital and the Joys
of Migrating...Again

Cara Key, Oregon State University
Greg Luis Ramirez, Oregon State University
Linda Sato, University of Oregon
Ryan Wick, Oregon State University

About the repository - <https://oregondigital.org>

 UNIVERSITY OF OREGON Libraries

 Oregon State UNIVERSITY Libraries

OREGON DIGITAL

Search... 

[Advanced Search](#)

[Login](#)
[Bookmarks](#)
[History](#)

Limit your search

- [Creator](#) >
- [Contributor](#) >
- [Arranger](#) >
- [Artist](#) >
- [Author](#) >
- [Composer](#) >
- [Illustrator](#) >
- [Interviewee](#) >
- [Lyricist](#) >
- [Photographer](#) >
- [Recipient](#) >
- [Topic](#) >

OREGON DIGITAL

[Browse by Collection](#)

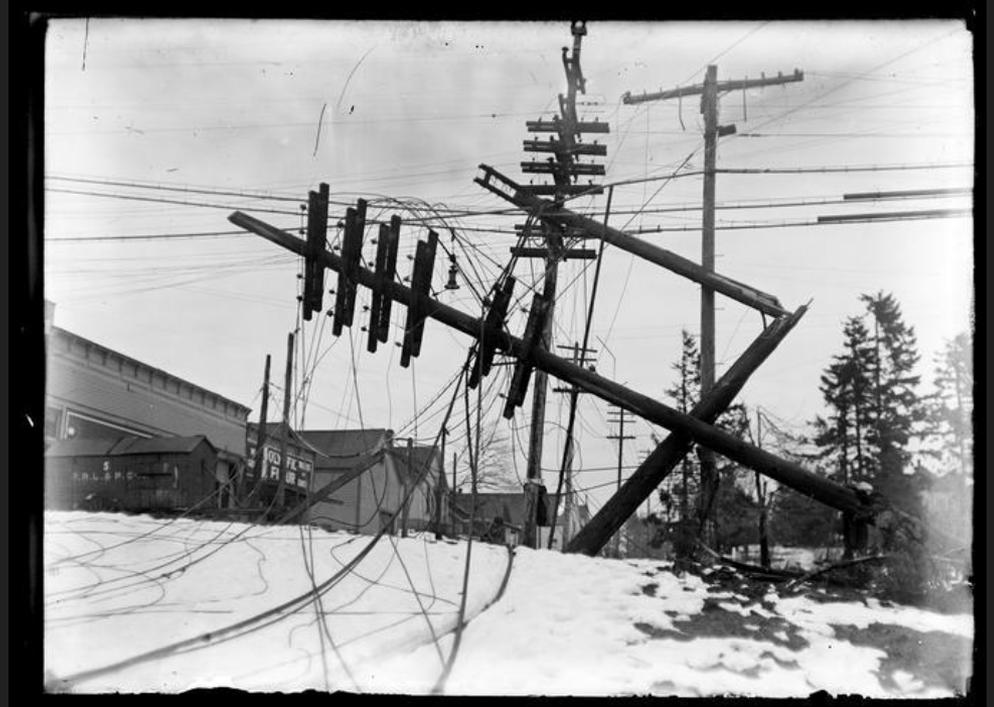
View a list of digital collections.

The collections of Oregon Digital are primarily created to support the teaching and research missions of the [University of Oregon](#) and [Oregon State University](#). The collections are comprised of unique digitized and born-digital materials including photographs, articles, sheet music, manuscripts, ephemera, and more. Oregon Digital is collaboratively managed by the [Digital Scholarship Center](#) of the [University of Oregon Libraries](#) and by [Oregon State University Libraries](#).



Decision to migrate/rebuild

- Not if but when/how
- Current stack is aging
- Balance work and resources needed with other projects
- How many features are in Hyrax now vs. what we want

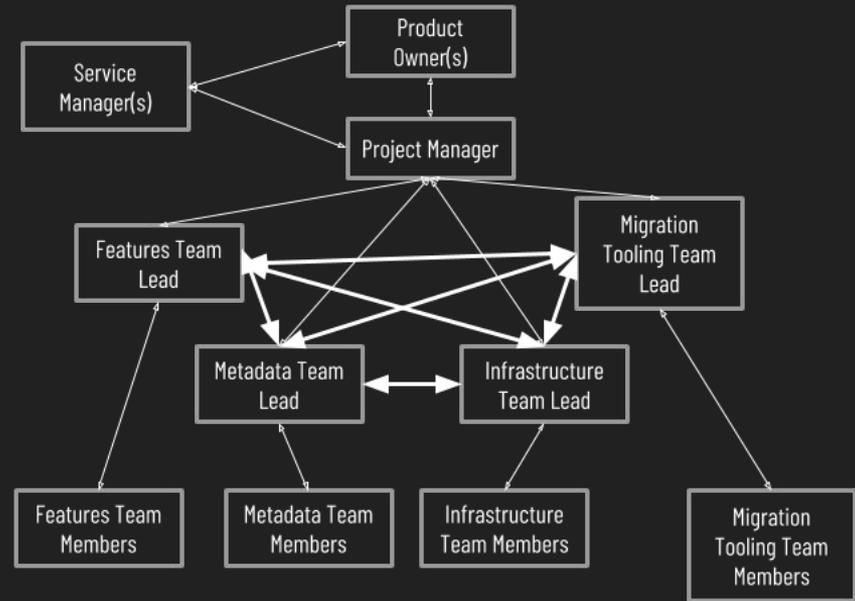


Project goals

- Rebuild Oregon Digital starting with latest Hyrax
- Less custom code, more manageable local customizations
- Better compound object support (yay PCDM)
- More metadata enhancements, better context (esp. archival items)
- Better and fewer media viewers
- Build a foundation for Spotlight exhibits
- Clearer display of Rights and License information, encourage re-use
- User Collections, 'My Shelf'
- Looking ahead to Valkyrie implemented in Hyrax

Project management

- Continue OSU and UO partnership
- Four technical teams: Features, Metadata, Infrastructure, Migration Tooling
- Product Owners and Service Managers determine requirements and priorities
- Workcycles are 6 weeks long, broken into 2-week sprints
- GitHub, Slack, Google Drive



Zooming in on migrating assets

- Since we're not upgrading-in-place: external migration
- Want as short a time of 'new content freeze' as possible
- Metadata Team and Migration Team working closely together
- Metadata updates covered first
- Second, introducing Hyrax-migrator, our migration gem

Metadata

Overview of metadata changes

- Creating an updated Metadata Application Profile (MAP) was a very early priority
- This effort was followed by crosswalking and remediation of existing OD metadata to fit new MAP
 - Updates to predicates, vocabularies, allowed values & data types
 - Cleanup of legacy problems
- Emphasis on where metadata impacts user experience, for example:
 - Streamlined facets for better browsing
 - Using metadata values to manage access restrictions

Property Name	Model	Work Type Where Defined	Obligation	Recommendation
Created	Collection; Work	Generic	{0,n}	SHOULD
Collected Date	Work	Generic	{0,n}	MAY
Date	Collection; Work	Generic	{0,n}	SHOULD
Earliest Date	r			
Issued	Work	Generic	{0,n}	SHOULD
Latest Date	r			
View Date	Work	Generic	{0,n}	MAY
Acquisition Date	Work	Generic	{0,n}	MAY
Identifiers				
Accession Number	Work	Generic	{0,n}	MAY
Barcode	Work	Generic	{0,n}	MAY
Hydrologic Unit Code (HUC)	Work	Generic	{0,n}	MAY
Identifier	Work	Generic	{1,n}	MUST
Item Locator	Work	Generic	{0,n}	MAY
LC Control Number				
Longitude Latitude Identification (LLID)	Work	Generic	{0,n}	MAY
Rights				
Copyright Claimant	Work	Generic	{0,n}	MAY
Rights Statement	Work	Generic	{1,1}	MUST
Rights Note	Work	Generic	{0,n}	SHOULD
Rights Holder	Work	Generic	{0,n}	SHOULD
License	Collection; Work	Generic	{0,1}	SHOULD
Use Restrictions	Work	Generic	{0,n}	MAY
Access Restrictions	Work	Generic	{0,1}	MAY
Source (Publishing)				
Box Name				
Box Number	Work	Generic	{0,n}	MAY
Citation	Work	Generic	{0,n}	MAY
Contained in Journal	Work	Document	{0,n}	MAY
Current Repository ID	Work	Generic	{0,n}	MAY
Folder Name	Work	Generic	{0,n}	MAY
Folder Number	Work	Generic	{0,n}	MAY
Has Number	Work	Document	{0,n}	MAY
Is Volume	Work	Document	{0,n}	MAY
Language	Collection; Work	Generic	{0,n}	SHOULD
Local Collection ID	Work	Generic	{0,n}	MAY

Metadata strategy

Step 1: Update MAP

Step 2: Update existing metadata to align with MAP

Step 3: Migrate perfect metadata to new system

Step 4: Party!



Metadata remediation

- We are doing the bulk of our metadata remediation in the current (old) Oregon Digital - before migration of the assets
 - Not fundamentally changing the nature of the metadata
 - Allows concurrent development and metadata work
 - Less complication within the migration process, vs remediating as part of migration pipeline
 - More “mature” product at delivery, vs remediating after migration
- Most changes happen in batches, either via export + CSV editing + scripted updates or directly with scripts
- Requires updating configuration of old system to accommodate changes

Metadata example: Rights statement update

- Changing all Rights values to RightsStatements.org URIs
 - Formerly a mix of OpaqueNamespace.org URIs and Creative Commons licenses
- Confirm all items have a Rights field value
- Map existing values to RightsStatement.org values
- Update OD1 to fetch & validate RightsStatement.org URIs
- Run batch processes to swap out URIs / move all Creative Commons URIs to License field
- Map RightsStatement.org values to proposed OD2 re-use categories for planned rights-based searching

Rights required

Apply 1 standardized rights statement. Explanations of the statements can be found at <https://rightsstatements.org/en/>

✓ In Copyright

- In Copyright - EU Orphan Work
- In Copyright - Educational Use Permitted
- In Copyright - Non-Commercial Use Permitted
- In Copyright - Rights-holder(s) Unlocatable or Unidentifiable
- No Copyright - Contractual Restrictions
- No Copyright - Non-Commercial Use Only

Hyrax-Migrator Gem

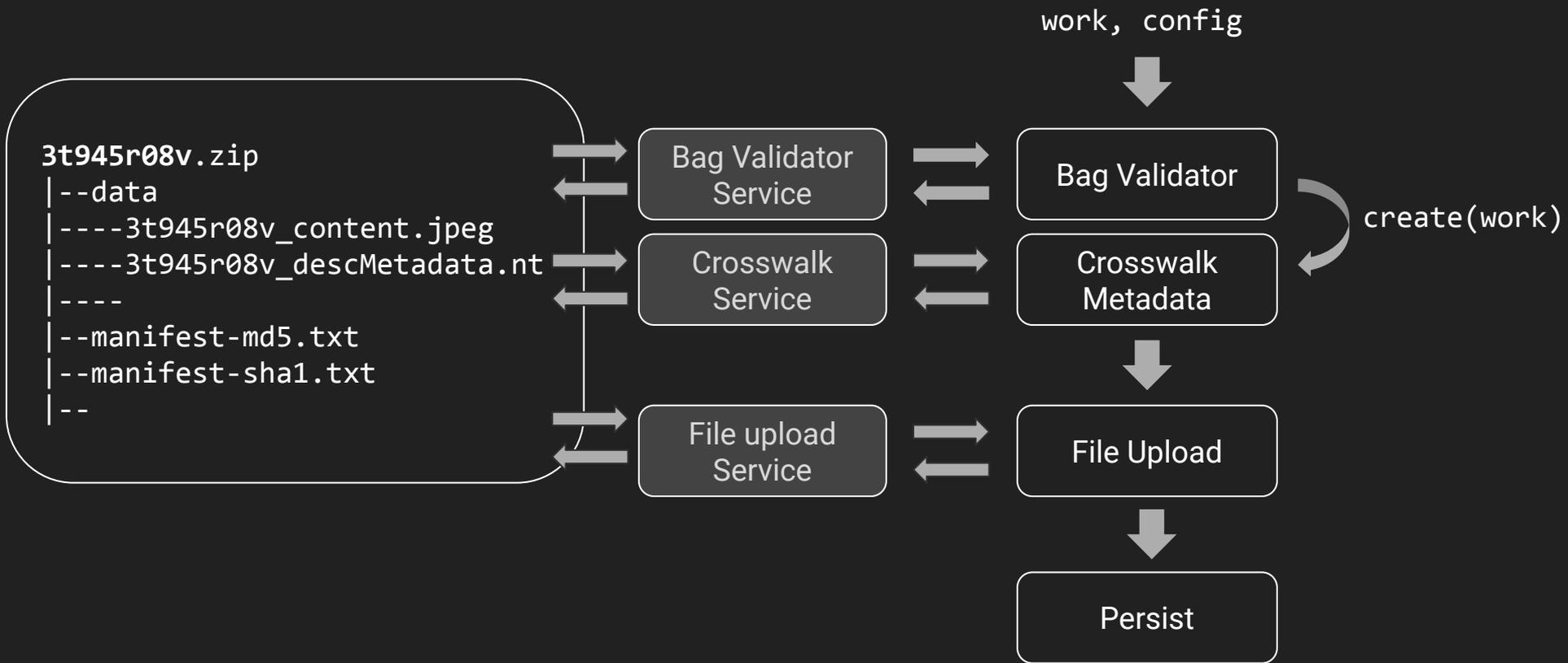
Hyrax-Migrator: Design

Inspired by Hyrax Actor Stack

Main components:

1. **Work** stores filepath of uploaded files, metadata, migration status
2. **Actors** update the work and launch services
3. **Services** do the work and are specific to our organization
4. **Middleware** can be modified by writing new actors/services and adding them to the configuration

Hyrax-Migrator: Ingest



Hyrax-Migrator: Strategies / Implementation

Crosswalk:

- property: title

 - predicate: <http://purl.org/dc/terms/title>

 - multiple: true

 - function:

- property: creator_attributes

 - predicate: <http://purl.org/dc/elements/1.1/creator>

 - multiple: true

 - function: attributes_data

Hyrax-Migrator: Running the stack

```
aasm do
  state :bag_validator_initial, initial: true
  state :bag_validator_succeeded, :bag_validator_failed
  event :bag_validator_initial do
    transitions from: %i[bag_validator_initial bag_validator_failed],
               to: :bag_validator_initial
  end
  event :bag_validator_failed, after: :post_fail do
    transitions from: :bag_validator_initial,
               to: :bag_validator_failed
  end
  event :bag_validator_succeeded, after: :post_success do
    transitions from: :bag_validator_initial,
               to: :bag_validator_succeeded
  end
end
```

Challenges



HyraX-Migrator: Bag Ingest Service

Configurations

```
c = Hyrax::Migrator::Configuration.new
c.upload_storage_service = :file_system
c.ingest_storage_service = :file_system
c.file_system_path = "/data/tmp"
c.ingest_local_path = "/data/tmp"
c.skip_field_mode = true
c.migration_user = 'admin@example.org'
c.crosswalk_overrides_file = "tmp/crosswalk_overrides.yml"
# Ingest service
i = Hyrax::Migrator::Services::BagIngestService.new(['batch_demo'], c)
i.ingest
```

Hyrax-Migrator: Batch demo

```
1. dev-tat (tmux)
/batch_demo c/i/m/collections.yml
=====
6 " Netrw Directory Listing (ne
5 " /Users/luisramg/Projects/OD2/tmp/batch_demo
4 " Sorted by name
3 " Sort sequence: [\/]$, \<core\%(\\. \\d\\+\\)\>=\\>, \\.h$, \\.c$, \\.cpp$, \~\>=
2 " Quick Help: <F1> help -go up dir D.delete R.rename s:sort-b
1 "
.. /
1 ./
2 3t945r08v.zip
3 4t64gn50h.zip
4 5138jf19t.zip
5 5t34sj883.zip
6 6t053g24j.zip

(0.5ms) BEGIN
server_1 |
(0.5ms) COMMIT
server_1 | Hyrax::Engine.after_initialize - persisting
server_1 | gistered roles!
server_1 | after_initialize, mounting hyrax-migrator at
/migrator'

(0.6ms) BEGIN
(0.4ms) COMMIT
Hyrax::Engine.after_initialize - persisting registered roles!
after_initialize, mounting hyrax-migrator at '/migrator'

d444e5c18455 healthcheck/postgres:alpine "docker-entr
oint.s.." 6 hours ago Up 6 hours (healthy) 0.0.0.0:54
->5432/tcp valkyrie-development_db_1
luisramg@Gregs-MacBook-Pro ~/Projects/OD2 master
exec -it c9941468dc50 bash
root@c9941468dc50:/data# bundle exec rails c
DEPRECATION WARNING: Sprockets method `register_engine` is deprec
ed.
Please register a mime type using `register_mime_type` then
use `register_compressor` or `register_transformer`.
https://github.com/rails/sprockets/blob/master/guides/extending_s
prockets.md#supporting-all-versions-of-sprockets-in-processors
(called from block in <top (required)> at /data/config/initializ
s/assets.rb:18)
Loading development environment (Rails 5.1.7)
[1] pry(main)>
```



Testing and Next Steps

Seed data - wide range of metadata and content types from both institutions

Testing locally and on staging

Validation

Finalize workflow

GUI



Image: Fruit Testing, Food Technology Department, Oregon State College, 1948.

OSU Special Collections & Archives Research Center

<https://oregondigital.org/catalog/oregondigital:df70d726p>

Did we really learn our lesson(s)?

Mostly. What are we doing different from last time?

- We have built up a lot of experience with our metadata/RDF/linked data, and how to do updates, so we're doing nearly all of it before the migration begins
- More development time spent on migration tooling
- Hyrax-migrator is stricter about what predicates to map to
- Compound objects are still complicated

{end}

- Questions?
- Thank you!

<https://oregondigital.org>

<https://github.com/oregondigital/hyrax-migrator>



ORNITHORHYNCHUS ANATINUS.
Illustrated by Gould, 1845. Original in the collection of the British Museum, London.

Image: John Gould - Richter, H. C. *The mammals of Australia.* by John Gould. (1845-1863) Volume 1, Plate 1, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=1172588>