

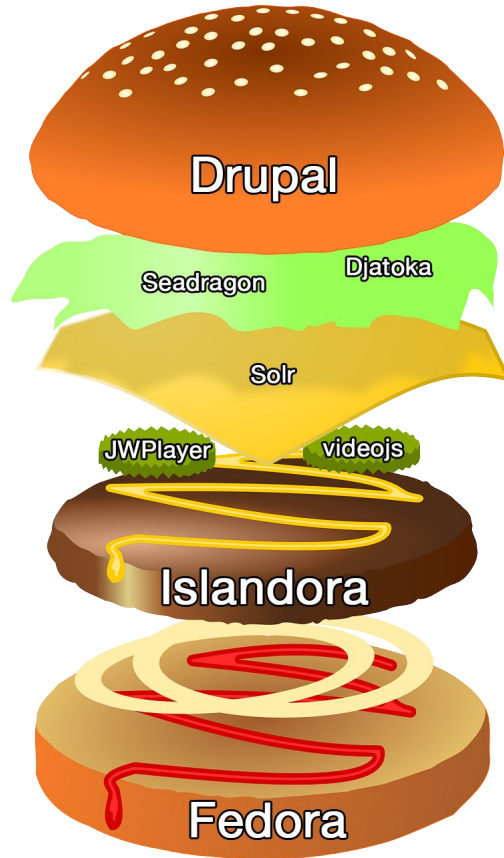
Islandora:

Creating and Sustaining an Open Source Community

Melissa Anez
Islandora Foundation

Nick Ruest
York University

Islandora



Where We've Been

History

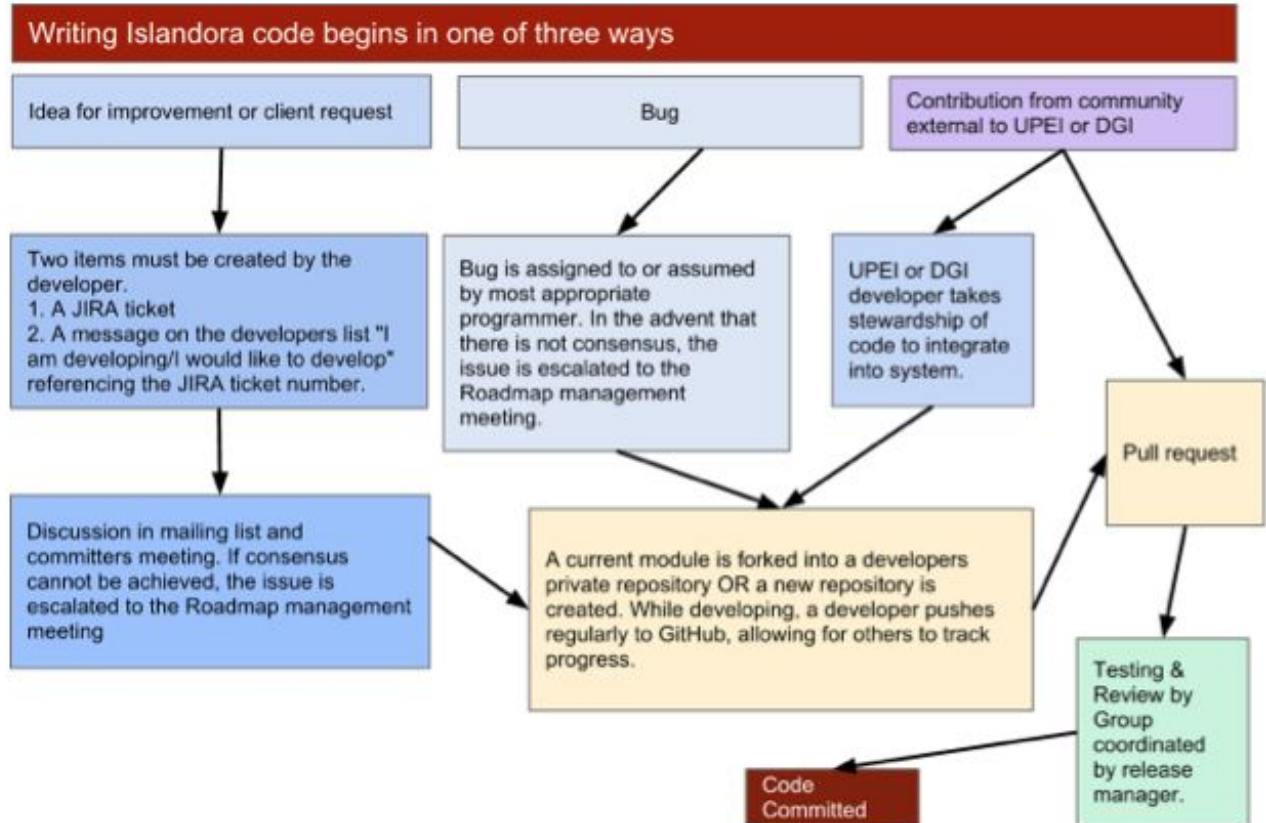
Developed at UPEI in 2006

AIF Grant 2010 - 2013

Islandora Foundation - July 2013



The old way...



The new way...

Coding Standards

The Islandora module follows the [Drupal coding standards](#). This will be tested time by time by the [coder module](#). In order to make things easier try to test your code w coder module before committing code to the Islandora GIT. There are also some [troubleshooting](#) tips.

Working With Fedora Objects Programmatically Via Tuque

qadan edited this page on May 17 - 12 revisions

Islandora introduces support for a Fedora repository to be connected to and manipulated using the Tuque PHP library. This library can be accessed using functions included with Islandora, available inside a properly-bootstrapped Drupal environment. It can also be accessed directly outside of an Islandora environment.

Tuque is an API, written and accessible via PHP, that connects with a Fedora repository and mirrors its functionality. Tuque can be used to work with objects inside a Fedora repository, accessing their properties, manipulating them, and working with datastreams.

This guide will highlight methods of working with Fedora and Fedora objects using Tuque both by itself and from a Drupal environment.

Table of Contents

Variables repeated often in this guide

Accessing the Fedora Repository

Connecting to Fedora

Accessing the repository

Working with existing objects

Loading an object

Properties



Documentation for developers

- Working With Fedora Objects Programmatically Via Tuque
- Working Programmatically With XML Forms
- Programming Solution Packs
- Coding Standards
- Git guidelines and best practices
- How Islandora Code Gets Made

Ins Welcome!

If you are reading this document then you are interested in contributing to Islandora, and that's awesome! All contributions are welcome: use-cases, documentation, code, patches, bug reports, feature requests, etc. You do not need to be a programmer to speak up!

We also have an irc channel -- #islandora -- on freenode.net. Feel free to hang out there, ask questions, and help others out if you can.

Git Guidelines Workflows

willtp87 edited Islandora Committers Calls happen every Thursday at 2PM EST. The meetings are generally held on Skype, and you can join the calls by sending a request to community@islandora.ca.

In general we **Documentation** sensible. They go into a lot of detail about line endings which we don't need to worry about as much at the moment. Specifically we should try and do the following, however:

Commit Messages

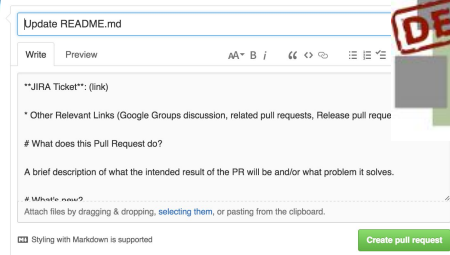
Commit messages should follow the guidelines [/2008/04/19/a-note-about-git-commit-messages](#)

In summary:

- First line: JIRA issue ID in all caps (if applicable)
- Second line: blank
- Following lines: more detailed description, multiple paragraphs, separated by blank lines. Use the present tense when writing messages, applied patch."

Using the git command line tool you can add multi-line commit messages by typing a \ at the end of a line and continuing to type. E.g.,

```
$ git commit -m "ISLANDORA-110 Cannot delete content model when there is only one"
Remove logic that prevents delete widget from being drawn."
```



How Islandora Code gets Made

ruobot edited this page on Mar 25 - 9 revisions

Contributing to Islandora

Islandora is committed to being a great open source project and to help improve that we are introducing new policies to improve our development process. We want to make our project a successful project and we want to make our development process more efficient through discussions.

All new features should be discussed in the [development](#) channel. We want to give the community the opportunity to make recommendations and discuss them before they are implemented.

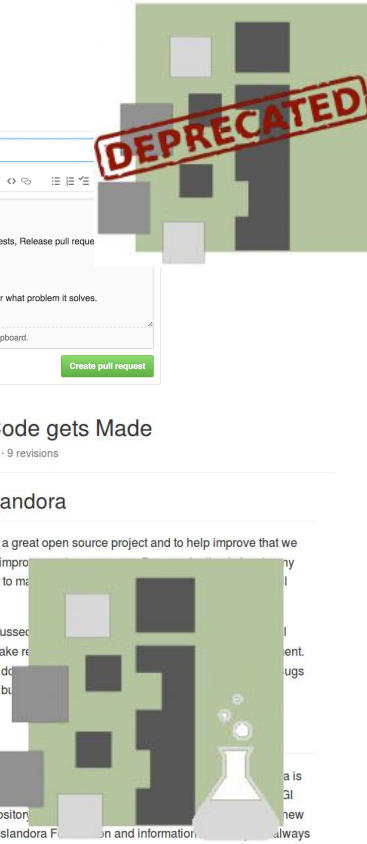
When a feature is broken down into smaller pieces, it can be implemented as a bug fix or a new feature.

Repository Structure

Islandora has a central repository for the code. This repository is used for all development. The master branch is used for production-ready code. The development branch is used for integration testing. The feature branch is used for developing new features. The branch naming convention is as follows: feature/branch-name, develop, master, production.

When a developer wants to contribute to the project, they must first create a pull request. The pull request should be created against the master branch of the Islandora repository. The pull request should include a description of the changes and a link to the JIRA issue that the pull request is addressing.

After the pull request is created, the Islandora committers will review the pull request. If the pull request is approved, the code will be merged into the master branch. The code will then be deployed to the production environment. The pull request should be signed off then the code will be merged into the project. We use Travis for unit testing and Jenkins for integration testing/docs. Tests do not supply 100% coverage and Jenkins is currently unreliable so smoke testing of all pulls by reviewers is advisable. It is also advisable to run the tests and style checks on a module that is being altered before submitting a pull request. The command Travis runs for style checking in Drupal 7: `./Programing-Solution-Packs`

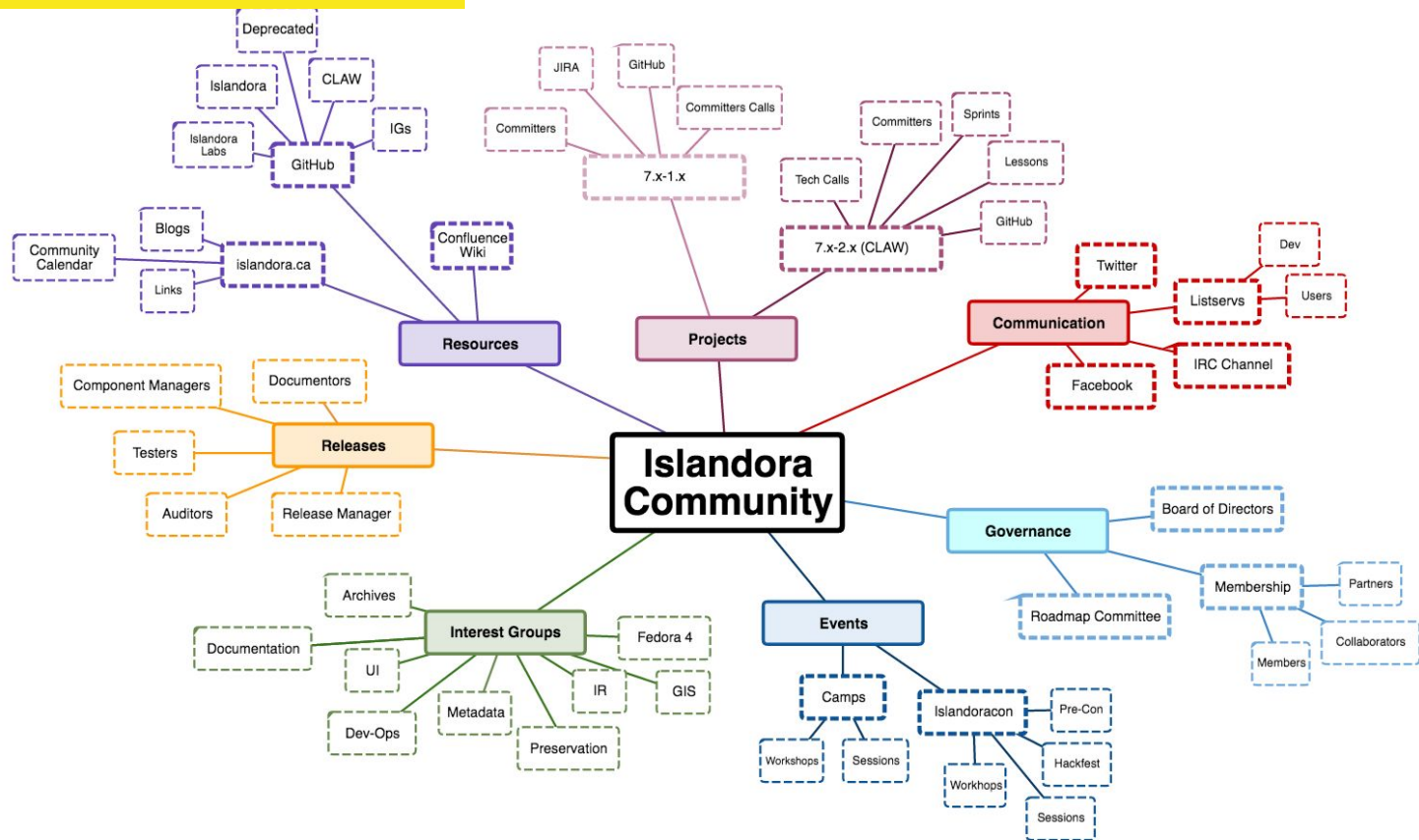


Where We Are

Islandora Community

- **150+** public sites,
- **1240** listserv members
- **104** GitHub members
- **20** Committers
- **23** Contributed Modules
- **26** Foundation members
- **10** Interest Groups
- **5** Community-led software releases
- **16** Camps
- **2** major versions
- **2** full-time employees

Islandora Community



Islandora Community



Islandora Foundation



Islandora Foundation



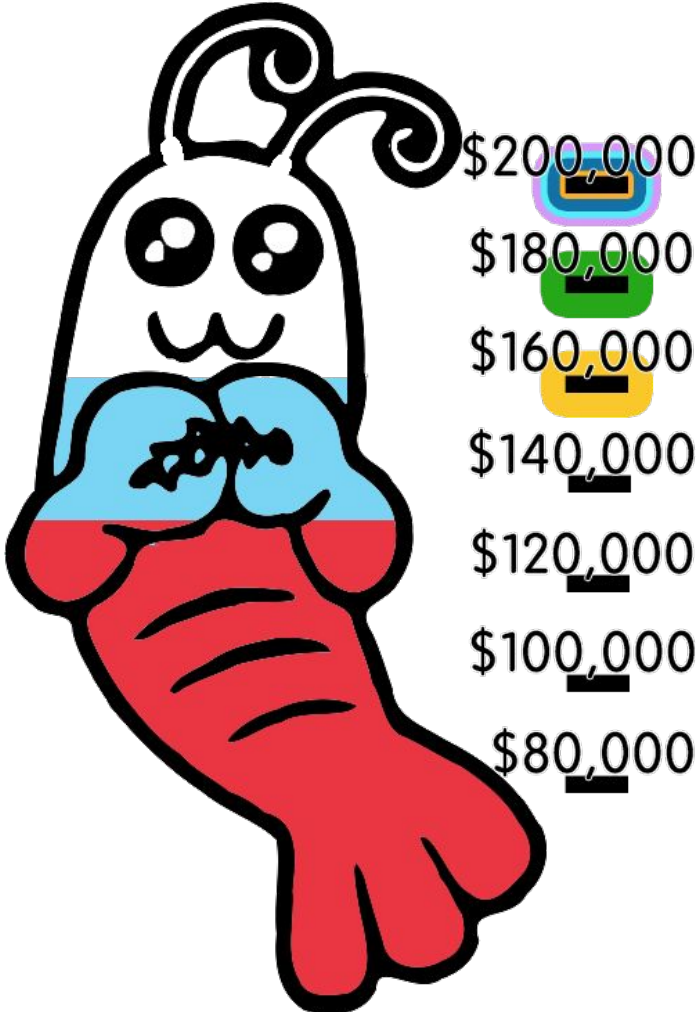
Collaborators:

- Andrews University
- University of Connecticut
- Common Media
- American Philosophical Society
- Florida State University
- Florida Virtual Campus
- Marmot Library Network
- Delft University

Members:

- California Institute of Technology
- PALS (Minnesota State Colleges and Universities)
- Technical University of Denmark
- Chinese University of Hong Kong
- BCELN
- Agile Humanities
- Berklee College of Music
- METRO

Membership Funding



What does membership get you?

- **A voice!**
 - Sustainability
 - Governance
 - Collaboration
- **Project & Community Manager**
- **Technical Lead**

Islandora is...

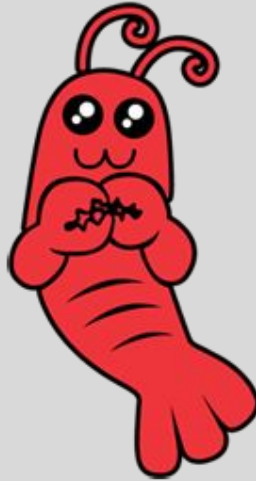
**...not static. It is ever
evolving, dynamic, and
requires organizational
commitment.**

**Contributing to open source
digital asset
management/preservation
systems and their
communities...**

**...is a core component of
digital preservation;
sustainability, and risk
mitigation at the very least.**

Governance

Board of Directors



Roadmap
Committee

Committers

Board of Directors

The Board is primarily concerned with legal and financial aspects of the Islandora community.

It is chiefly supportive of the Roadmap Committee and Islandora Committers, but directional in the sense that when the Roadmap Committee looks to the Board for direction and lays out the issues in a way that the Board can act on, the Board will provide direction.

Roadmap Committee provides the core directions of the community.

<http://islandora.ca/if/board>

Roadmap Committee

- Consults and liaises with the Board of Directors as needed, ex. fiscal or legal issues;
- Consults and liaises with the Committers as needed, ex. technical direction, policy and process changes;
- Review, consult with the Community, and vote on timelines for adoption of major versions of the main Islandora components, ex. next major version of Drupal;
- Review and vote on submissions made under the Licensed Software Acceptance Procedure;
- Facilitate the creation of Interest Groups, as well as review and dissolve Interest Groups, and appoint convenors when necessary;
- Solicit and receive reports, as well as review and approve requests from Interest Groups;
- Review and approve proposals for Islandora Camps, conferences, or other Islandora events; Propose, review, and approve policies and guidelines for community conduct as needed, ex. community practice, committers practice, community conduct.

Islandora Committers

Rights:

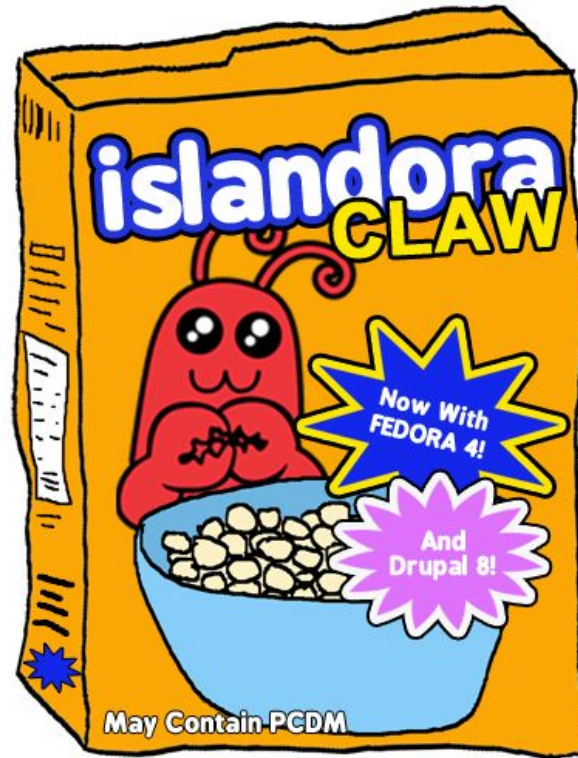
- Write access to the codebase
- Nomination privileges of new committers
- Release management privileges
- Binding votes on procedural, code modification, and release issues
- Access to the private Committers mailing list

Responsibilities:

- Monitor and respond to project mailing lists
- Attend project and technical meetings
- Monitor and vet bug-tracker issues
- Review and commit code contributions
- Ensure code contributions are properly licensed
- Guide and mentor new committers

Where We're Going

Future Development

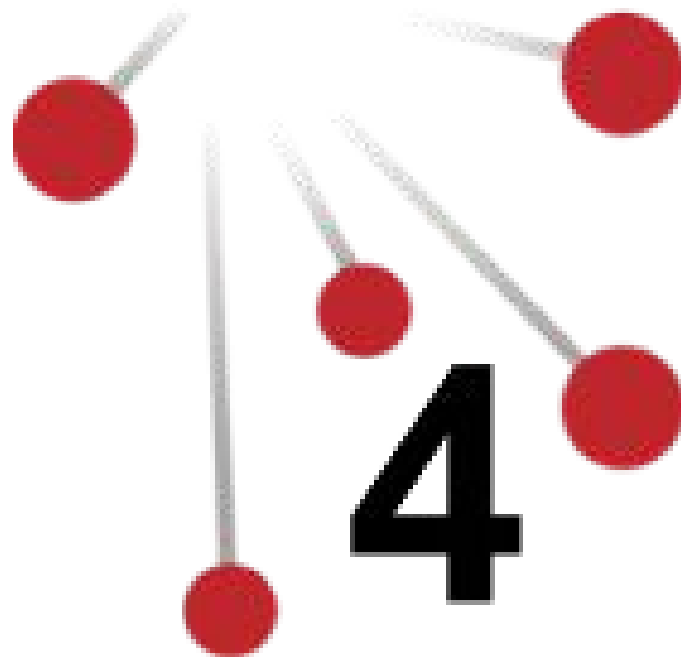


<http://islandora.ca/CLAW>

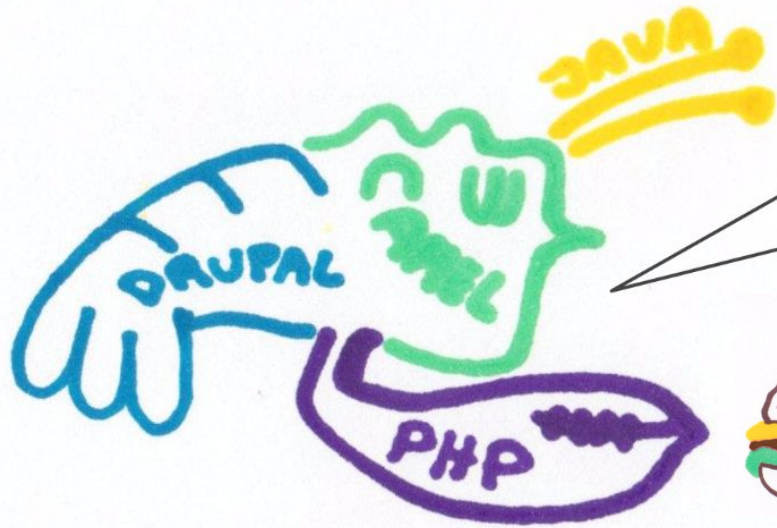
Islandora CLAW



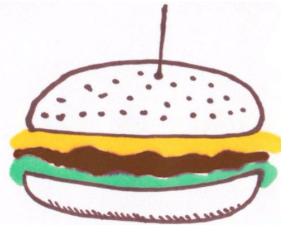
Drupal™



Islandora CLAW



I MAKE USE OF EXISTING TOOLS
INTEGRATING MYSELF INTO
EACH ONE. I'M NOT A LAYER, I
PLAY ROLES IN MULTIPLE ONES.



Community Centred

Islandora CLAW is
developed *by* the
Islandora community, *for*
the Islandora community.

Community Centred

“*All contributions* are welcome: use-cases, documentation, code, patches, bug reports, feature requests, etc. *You do not need to be a programmer to speak up!*”

**When is CLAW going
to be done?**

Minimum Viable Product; MVP

Clearly defined goals for community members to work towards, and present the point at which the CLAW team feels the software is ready to be tested out in small installations.

Needless to say, since this is a minimum product, a lot of parts of the previous prototype are either going to get replaced by an existing project to ease the maintenance load, or be removed altogether until time and resources allow other development.

https://islandora-claw.github.io/CLAW/mvp/mvp_doc/

CLAW Design Goals

- Big picture: Utilize Fedora 4/ Drupal 8 to their fullest
- Easier to install
- Easier to develop and contribute
- Expose Linked Data
- Easier to use
 - Enhanced user experience
 - Faster searching, display, and ingest
 - Expose more control of your repository through UI
- Easier to scale

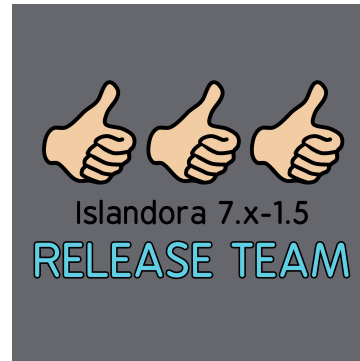
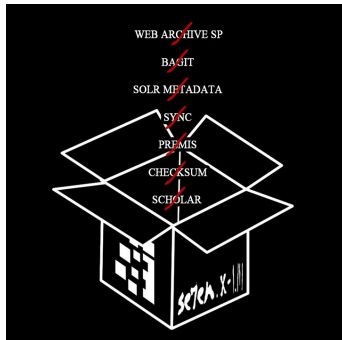
High Level Features

- Content modeled in Drupal as Entities using [PCDM](#) 1.0
- [REST](#) API exposed for Drupal Entities
- Support for collections, images, books, and pages
- The ability to control metadata mappings between Drupal and [RDF](#)
- Provide RDF based default descriptive metadata profile in Drupal
- The ability to export/import [JSON-LD](#)
- Automatic backup of Drupal content in Fedora 4
- Ability to restore/bootstrap a Drupal site from a properly structured Fedora 4 repository
- The ability to index and search resources with Apache Solr
- The ability to restrict access to collections and/or individual resources across all representations (Drupal, Fedora, Solr, etc...)
- Asynchronous derivative generation
- [Vagrant](#) environment for development purposes, which will serve as a starting point for more complicated, distributed installs

What about Islandora 7.x?

Islandora 7.x-1.x

- Still actively developed.
- Supported until CLAW is production ready (and, let's be honest, a little beyond).
- New releases twice yearly, managed by volunteers.





Islandoracon2017

Hamilton, ON

May 15 - 19

Questions?
