

A Distributed Microservices Framework Integrating Taverna

Donald Moses (UPEI), Paul Pound (UPEI)
OR 2014 - Helsinki, Finland

Microservices

... “a granular set of small, independent, but highly interoperable services”...

Our current implementation

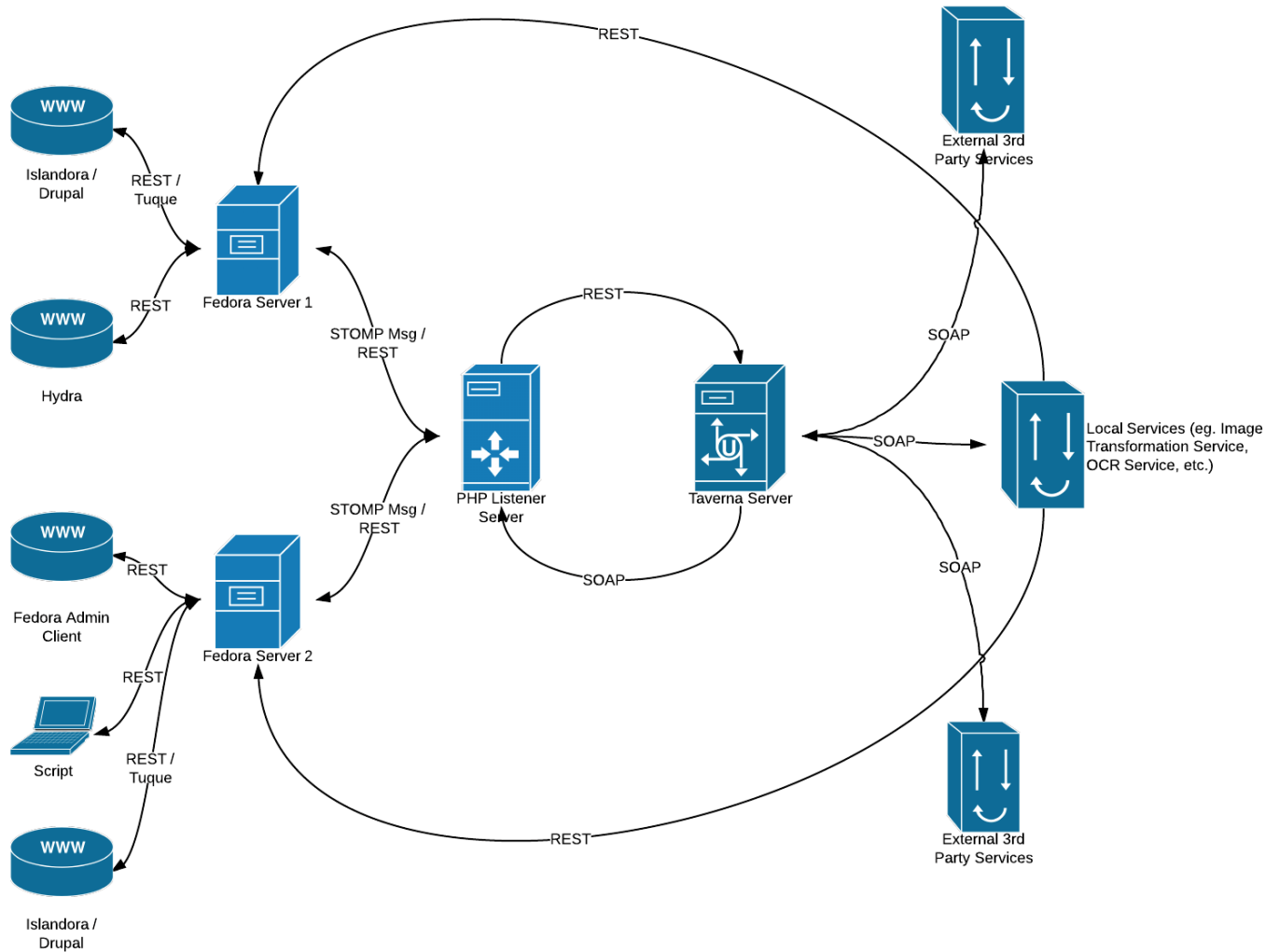
- Originally an in-house project utilizing a simple microservice framework
- 4th Year CompSci Capstone Course
 - Investigate integration of Taverna Workflow Server
 - Make services extensible/modular
 - Produced proof-of-concept
- Library completed work

Microservices Framework

- [Fedora / JMS](#)
- [Taverna Server](#)
 - [Taverna Workbench](#)
- [Tuque](#)
- [PHP Listeners](#)
- Local services
- External WSDL described services
 - including existing Taverna services

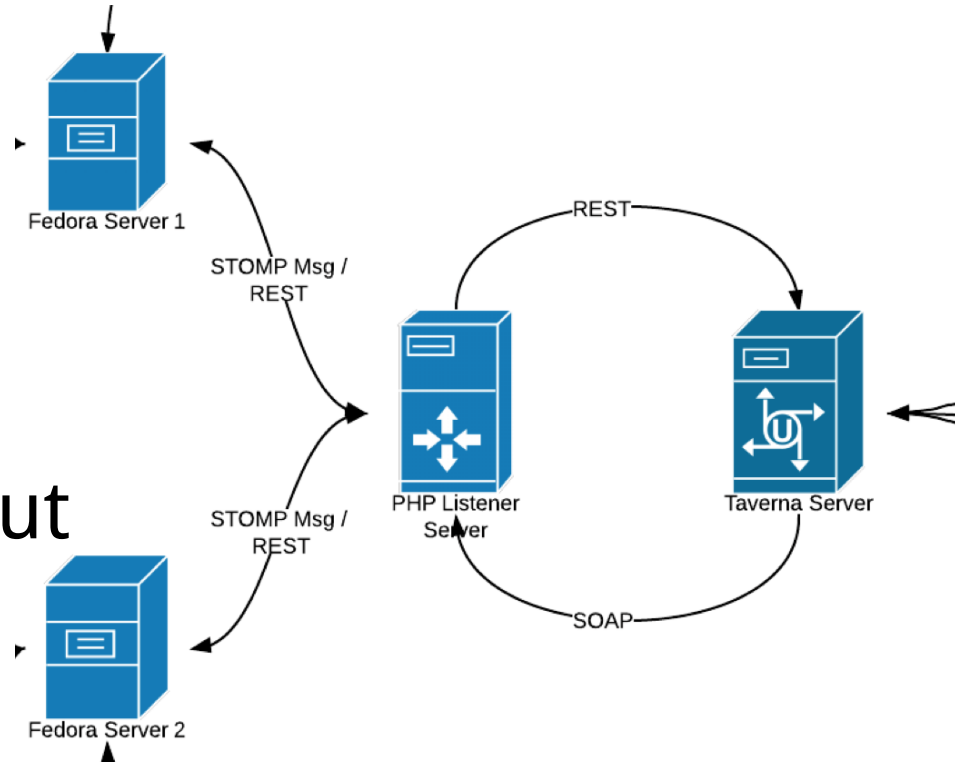
Microservices @ UPEI

- Object level or datastream level triggers
 - ingest, addDatastream, modifyDatastreamByValue, modifyDatastreamByReference, purgeObject, purgeDatastream
- Typical Services
 - creating derivatives
 - transforming data
 - enhancing metadata
 - migrating objects
 - creating/updating object relationships
 - updating XACML policies
 - identifier generation



PHP Listeners


- Listens for Fedora messages.
- Retrieves appropriate workflow.
- Posts workflow.
- Polls Taverna to find out when a workflow is finished.
- Sends ack to Fedora.



Extending Fedora's Content Models

- The Listeners inspect an object's CModel using Tuque to retrieve the appropriate workflow based on info in Trigger-datastreams.
- This workflow is then sent to the Taverna Server.

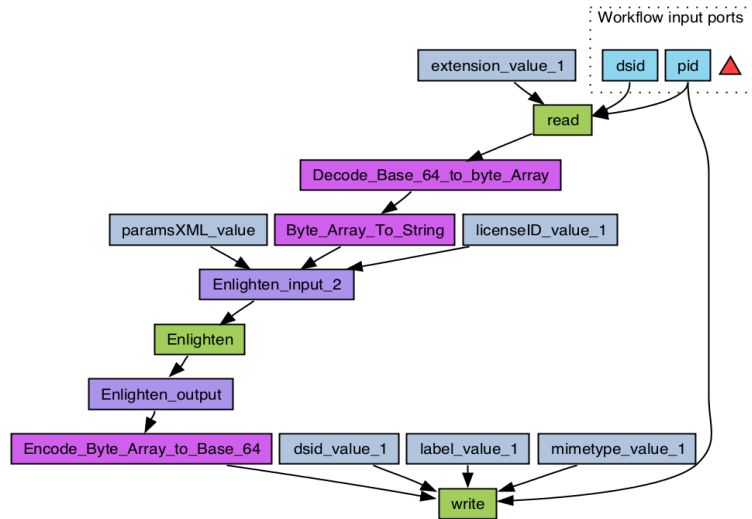
Newspaper Page CModel
DC
RELS-EXT
DS-COMPOSITE-MODEL
Trigger-datastreams
INGEST_WORKFLOW



```
<methods>
  <method type="addDatastream">
    <trigger id="OBJ">
      <t2flow id="INGEST_WORKFLOW"></t2flow>
    </trigger>
  </method>
  <method type="ingest">
    <t2flow id="INGEST_WORKFLOW"></t2flow>
  </method>
</methods>
```


Taverna Server

Taverna invokes the appropriate services as described by the t2flow.



```
<maxDelay>5000</maxDelay>
<maxRetries>0</maxRetries>
</net.sf.taverna.t2.workflowmodel.processor.dispatch.
layers.RetryConfig>
<wsdl>http://api.opencalais.com/enlighten/?wsdl</wsdl>
<operation>Enlighten</operation>
</net.sf.taverna.t2.activities.wsdl.
WSDLActivityConfigurationBean></configBean><annotatio
ns />
</activity></activities>
<maxJobs>1</maxJobs>
```

Roblib PHP Services

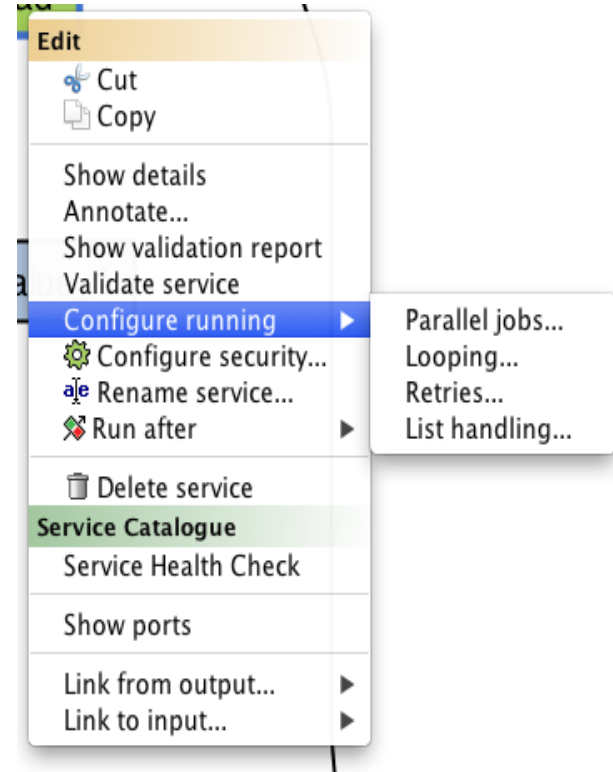
- Local services are written in PHP so we can leverage Tuque for communicating with Fedora.
- Utilize PHP-Soap to create a WSDL wrapper.
- Framework is pluggable so we can add new services without modifying existing code.
- Read and Write service so we can pass content to external services and back to Fedora.

Advantages

- Maintainability
- Consistency
- Leverage external services
- Distributed
 - less load on frontend server

Taverna Advantages

- Taverna Workbench for creating workflows.
 - Drag and drop GUI.
 - Import services via URLs.
 - Test workflows from within Workbench.
- Looping, run after, security.
- Taverna community.



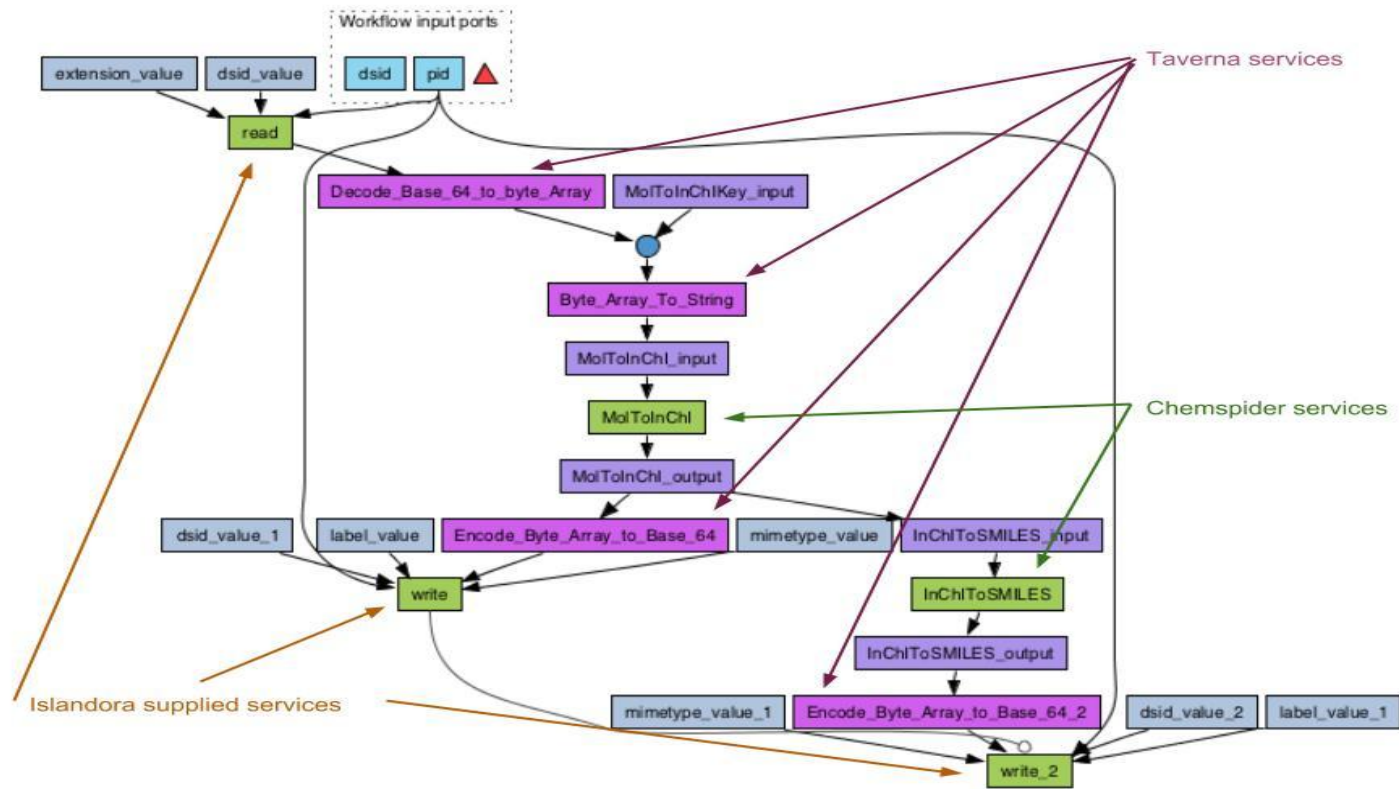
Scalable Framework

- Configurable number of threads in Listener.
- Taverna can distribute work to many servers.
- Can have many listeners running.
- Taverna Server can limit the number of active jobs on a per user basis.

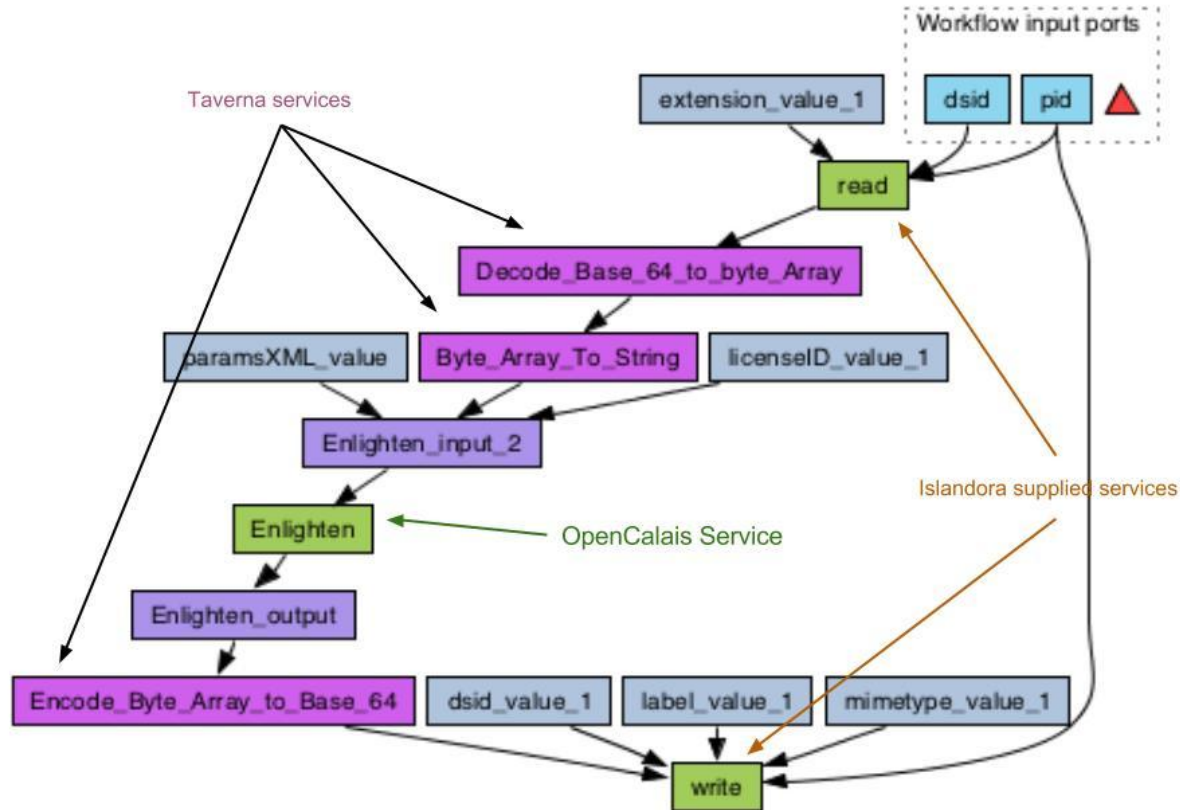
Sample Workflows

- ChemSpider
- OpenCalais
- Object Migration
- Derivative Generation

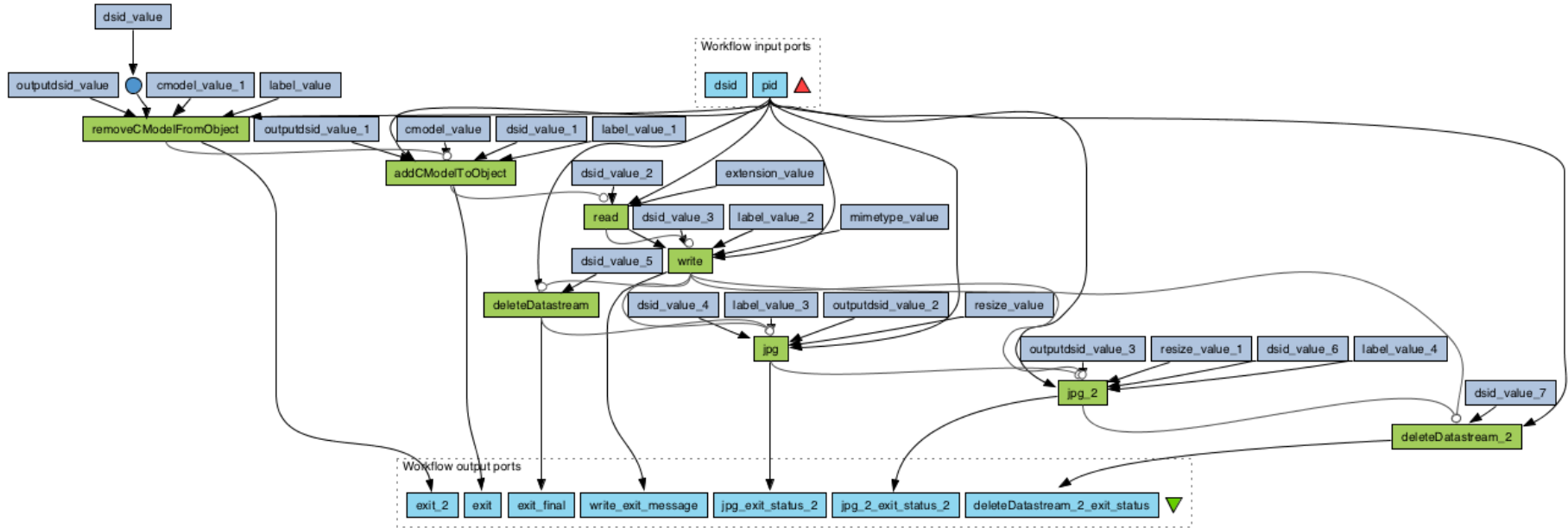
ChemSpider Workflow



OpenCalais Workflow



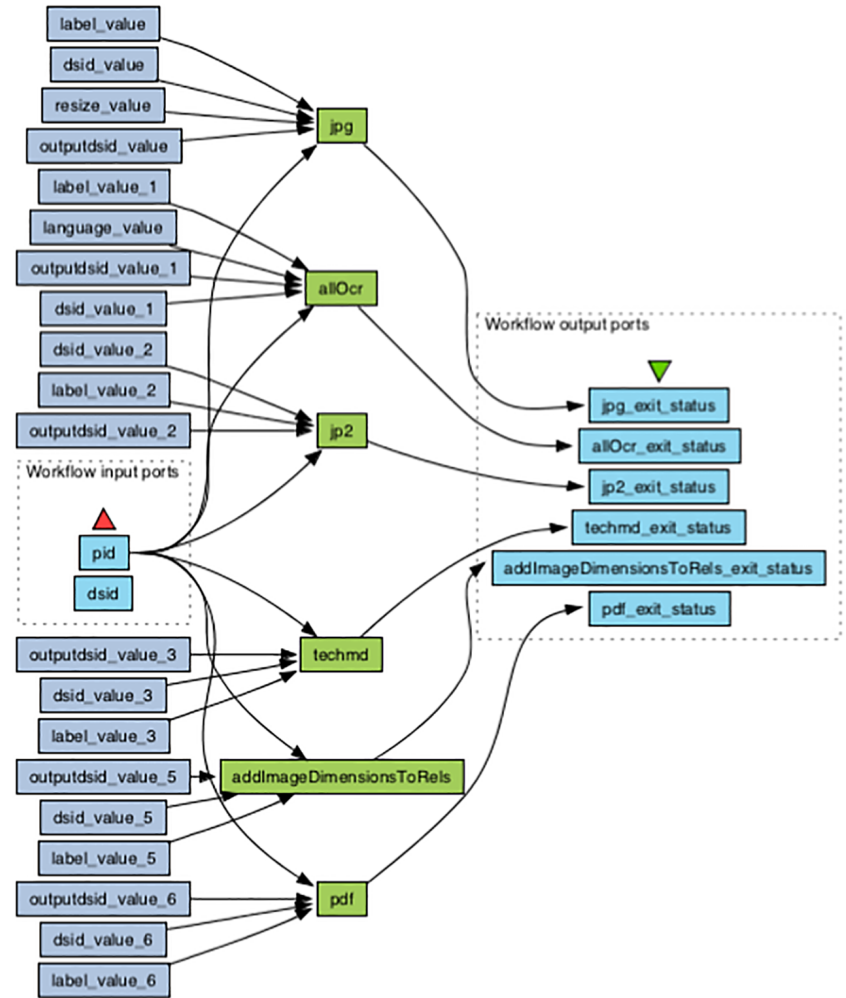
Object Migration Workflow



These are all local PHP services with some services configured to run after others

Newspaper workflow

All local PHP services, uses looping instead of running one service after another. If a service fails Taverna will run it again.



Future Work

- Taverna 2.5 testing underway.
- Move from PHP Listeners to Apache CAMEL?
- Cache Trigger-datastreams.
- Cache workflow datastreams.
- Workflow to index content in Solr.
- Explore in relation to Fedora4 sequencers.

Kiitos / Thank you

Don - dmoses@upei.ca

Paul - ppound@upei.ca

- [Github repo](#)
- [Documentation](#)



UNIVERSITY
of Prince Edward
ISLAND