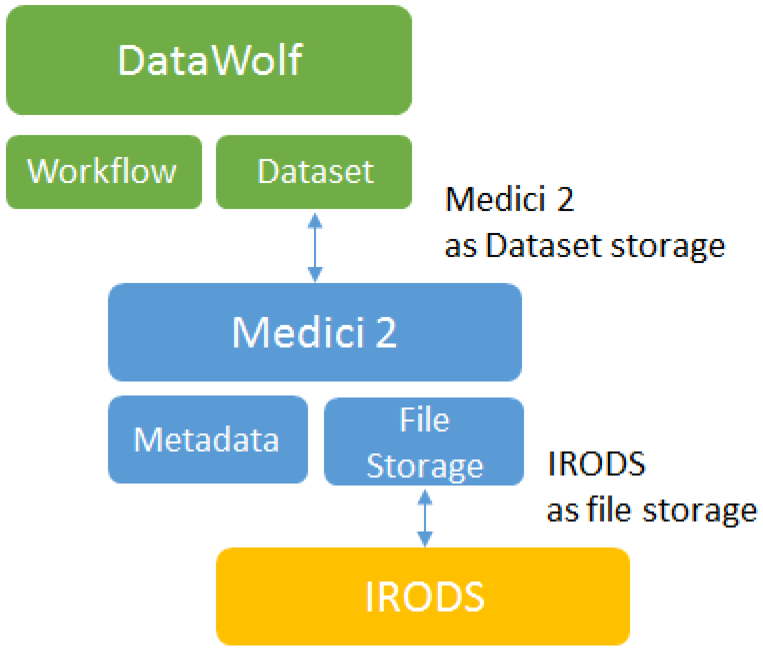
Integrating Medici and DataWolf with supporting IRODS

Note: requirement for CyberSEES project

- Access control: for example, there are LIDAR data from municipality and they don't want to share the data outside of the project.

This is the page to describe how to integrate Medici 2 and DataWolf with supporting IRODS. Through this integration/refactoring, we will achieve

- DataWolf will use Medici 2 as Dataset storage.
  - But, the design/refactoring will allow for DataWolf to store workflows in Medici 2
- Medici 2 will use IRODS as file storage
  - Medici 2 currently has two implementation for file storage: MongoDB GridFS and local file system
  - It is possible for us to store the part of the meatadata of datasets/files to IRODS, too. But it is a low priority and can be done later



Implementing IROD data storage in Medici 2

Medici 2 has the file storage API. It has two implementations: local file system and mongoDB GridFS. We need to have an implementation for IRODS. Since DataWolf already has an IRODS implementation for file storage using the IRODS java client library, Jargon, this can be leveraged for the Medici implementation.

Refactoring DataWolf to use Medici 2 as dataset storage

In order for DataWolf to use Medici 2 as dataset storage, the following tasks needs to be done

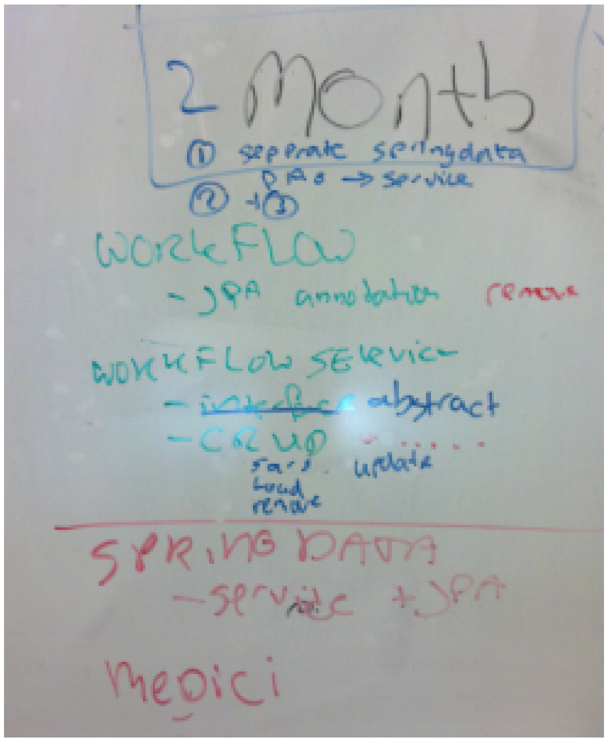
Researching on Spring Data vs Guice

Regarding eclipse (OSGI) support and properties configuration support.

- Spring Comparison: https://code.google.com/p/google-guice/wiki/SpringComparison

- Links about Guice OSGi support
  - Using Guice in an OSGi container: https://code.google.com/p/google-guice/wiki/OSGi
  - Peaberry Extension library: https://github.com/ops4j/peaberry
  - Eclipse RCP with Guice: http://www.itaware.eu/2007/11/27/rcp-with-guice/
  - Egap eclipse plugin for Guice: http://www.jaculon.de/egap/
- Guice Dynamic configuration by using properties file
  - https://code.google.com/p/google-guice/wiki/FrequentlyAskedQuestions#How_do_I_inject_configuration_parameters?
  - http://stackoverflow.com/questions/4805874/guice-and-general-application-configuration
  - http://stackoverflow.com/questions/9772557/using-google-guice-to-to-inject-java-properties
  - https://code.google.com/p/guice-xml-config/ (it has not been developed for a long time)
  - https://github.com/Netflix/governator
- Other links about Guice
  - JSR 330 (standard annotation for dependency injection) support of Guice: https://code.google.com/p/google-guice/wiki/JSR330
  - Guice JPA support: https://code.google.com/p/google-guice/wiki/GuicePersist

Separating the service layer and implementations



- Remove current JPA annotation from beans (separating spring data specifics)
  - workflow, workflow step
  - workflowtool
  - dataset
  - file
- Defining service API as abstract supporting (DAO -> service and implementation)
  - CRUD and extra capability (save, load, remove, update etc)
- Implementation layer for each service
  - JPA (without annotation in the bean but with xml definition)
  - Mongo
  - Medici 2