Open Core Data Status Update for ICDP

A review of Open Core Data opencoredata.org

Douglas Fils (@fils)

This presentation is at https://github.com/OpenCoreData/ocdDocumentation/tree/master/Talks/ICDP

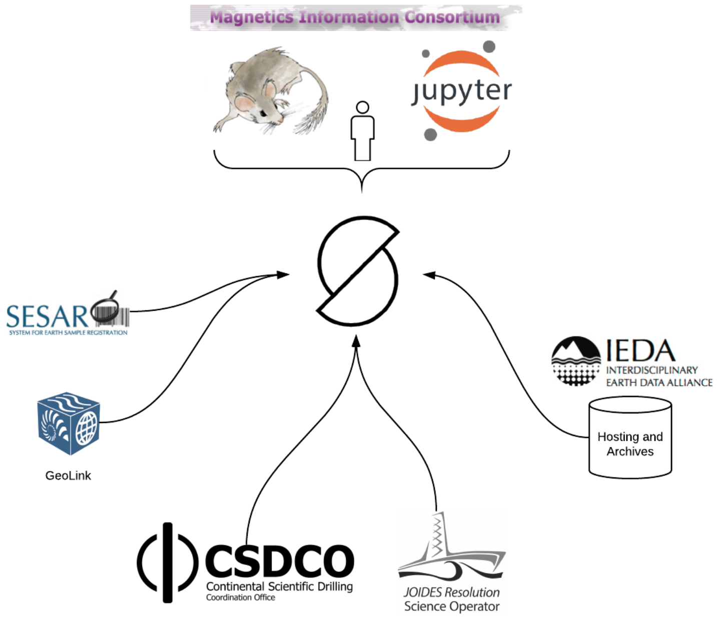
Outline

- Overivew of Open Core
- Review of Architecture
  - Docker based for easy deployment and migration
  - Simple dependancy approach
- Key components (Web, Semantic, Search, Document, API)
- Functional Goals
  - Semantic connection
  - Machine access
  - Cross site referencing

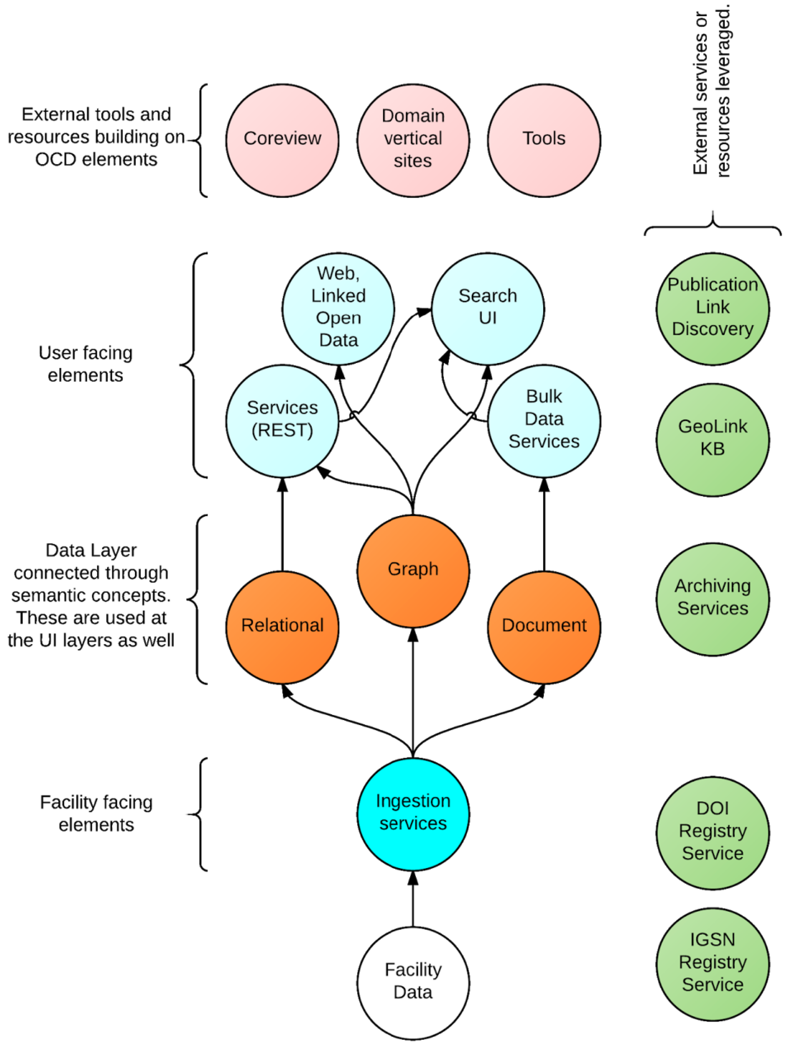
> *Note: for this talk I will try to move quickly to the functional goals section*

Over View

Open Core Data hosts metadata and optionaly raw data for CSDCO and JRSO. This data is semantically enhanced and connected using community vocabularies where possible and minimal self maintained vocabularies where required.

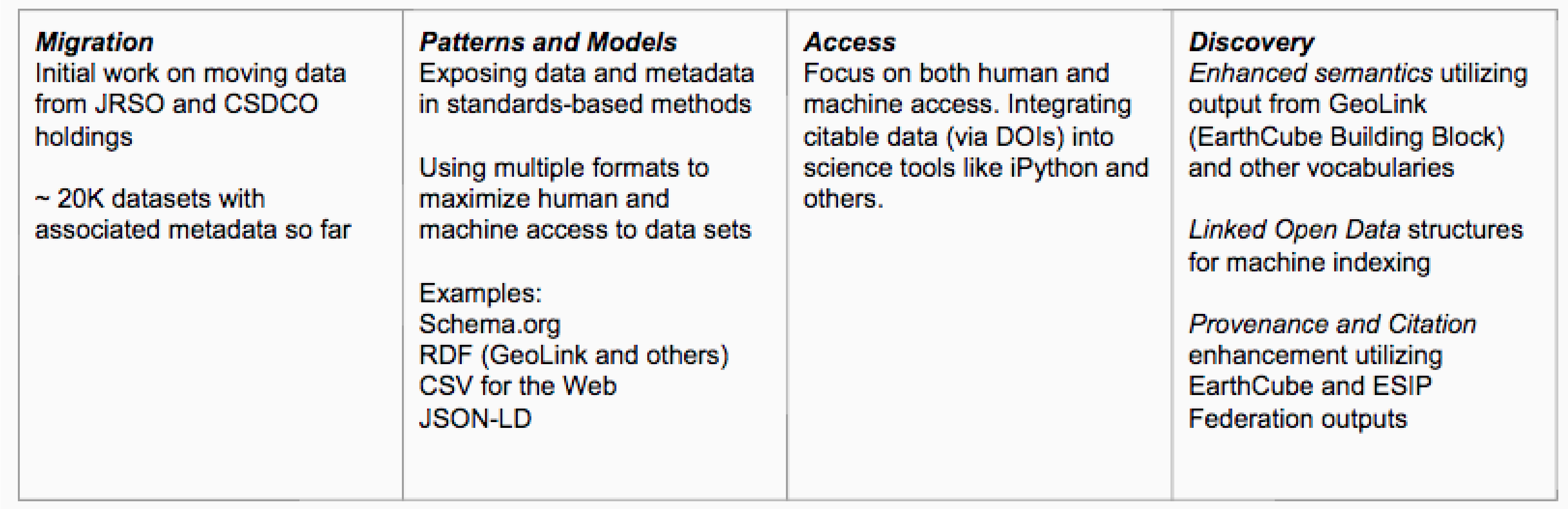

Magnetics Information Consortium
jupyter
SESAR
SYSTEM FOR EARTH SAMPLE REGISTRATION
GeoLink
IEDA
INTERDISCIPLINARY EARTH DATA ALLIANCE
Hosting and Archives
CSDCO
Continental Scientific Drilling Coordination Office
JOIDES Resolution Science Operator

The flow of interactions



This presentation is at https://github.com/OpenCoreData/ocdDocumentation/tree/master/Talks/ICDP

Table of functional activities



Migration	***Patterns and Models***	***Access***	***Discovery***
Initial work on moving data from JRSO and CSDCO holdings ~ 20K datasets with associated metadata so far	Exposing data and metadata in standards-based methods Using multiple formats to maximize human and machine access to data sets Examples: Schema.org RDF (GeoLink and others) CSV for the Web JSON-LD	Focus on both human and machine access. Integrating citable data (via DOIs) into science tools like iPython and others.	*Enhanced semantics* utilizing output from GeoLink (EarthCube Building Block) and other vocabularies *Linked Open Data* structures for machine indexing *Provenance and Citation* enhancement utilizing EarthCube and ESIP Federation outputs

Architecture

```
# Docker based
All elemented deploy from Docker files (most included
at github)
Use official containers for things like Mongo and others
we can

# Golang
Development done in a modern web centric language.

# Developed in the open; Github, Trello, Slack

# Polyglot Persistence  (all containerized)
- Triplestore (Blazegraph) for graphs
- MongoDB Mostly for core images, PDFs & otherblobs
  (some spatial use of Mongo as well)
- Relational (Janus) (and perhaps CSDCO and CHRONOS)
```

Key components

Web

```
- Linked Open Data
- HTML5 approaches including web components
- Responsive design for mobile access
- Leaflet (and leaflet components) for maps
- Polymer and other components for UI elements
```

Semantic

```
- Full stack semantics with use of RDF and connections to
ontologies and vocabularies
- Links to Geologic timescale URI's and GeoLink resources
- Links to DOIs Orcids and IGSNs under active integration
```

Key components

Search (free text)

```
- Based on Bleve (similar to Lucene)
- Allows mutli-index and faceted results.
- Low level, flexible to integrate but requires extra
effort in the UI area. (components being developed)
```

Document storage and API (and SPARQL)

```
Documents stored in MongoDB (GridFS)

API in Go with Swagger definitions
The APIs are still a quickly developing apsect.  Will use a
stable, dev, beta breakdown for them.

Access to all elements of OCD.  So external sites can call for
data, free text results, or any other call used in the
site UI/UX.
```

Function	Description

Functional Goals

Function	Description
Semantics	GeoLink, SKOS vocabularies connected to RDF graphs
Machine Access	Micro-data (schema.org), PIDs (DOI, IGSN, Orcid), CSV for the Web (JSON-LD)
Cross site connections	Working to leverage the PID's and Prov to allow data to migrate across sites like Neotoma, Magic and others

> *Note: Machine access is a product of the approach and the web architecture*

> *Note: "cross site connections" are under development*

Machine access

Example Landing Page

All pages have embedded Schema.org/Dataset and CSVW (CSV for the Web) metadata.

Access to datasets can be obtained both via SPARQL or Linked Open Data methods or via API calls.

Access to the CSDCO datasets is being developed under the ocdFX project. Data is being indexed and metadata and search index available in OCD. Links to files at CSDCO or IEDA are supported.

Cross site connections

As noted earlier in this document a key goal of Open Core Data is the exposure of OCD data holdings to 3rd parties.

This will be driven by a few keys elements

- Open data + Provenance
- Methods to identify data updates
- Two way connections and citation of data between partners

Early start will likely be 1 way flow out of OCD of data with provenance. From there we will build out the approach.

Access Methods

image here showing SPARQL, API (Jupyter and R) and machine and human web crawling

Ingest Methods

image here showing ETL process for ocdFX, transform from relational to DB + RDF, image loading and relations (note DOI, IGSN and potentially Orcid assignment here)

Thanks!

Douglas Fils

November, 2016 Douglas Fils