

# Using Semantic Web Technologies to Streamline the Implementation of the OGC Web Service Interface Specifications for Coverage and Feature Data within OPeNDAP

Daniel Holloway<sup>1</sup> (d.holloway@opendap.org), M. Benno Blumenthal<sup>2</sup> (benno@iri.columbia.edu), Nathan Potter<sup>2</sup> (ndp@opendap.org), Patrick West<sup>3</sup> (pwest@ucar.edu)

<sup>1</sup>OPeNDAP Inc., <sup>2</sup>IRI/Columbia University, <sup>3</sup>HAO/NCAR

## I. Abstract

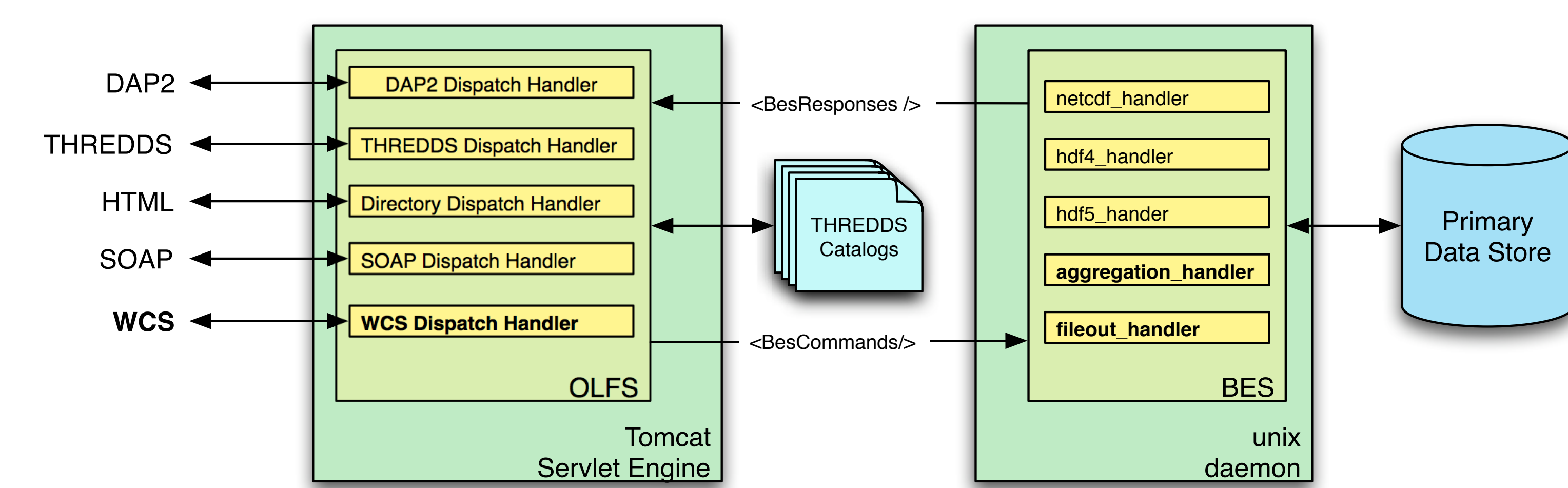
The OPeNDAP Data Access Protocol has seen widespread adoption within the science community. OPeNDAP servers are currently deployed by individual investigators, academic institutions, and at national and international data repositories to provide distributed data access for their respective user communities. Many of these data providers anticipate that there will be significant demand for data access by applications using the suite of OGC web service specifications. Supporting multiple data access protocols can be expensive, both in the initial acquisition and deployment cost for the software components as well as for the potentially redundant maintenance and security costs required when supporting multiple server implementations operationally. To provide a cost-effective solution for these data providers OPeNDAP is developing extensions to its data access protocol to enable the use of semantic web technologies for data and metadata transformations, and extensions to its server architecture to support request and response operations simultaneously for multiple data access protocols.

The OGC Web Coverage Service Interface Specification is the initial data access protocol to be layered onto the OPeNDAP server for this multi-protocol support. Supporting data access through the OGC service interfaces comprises operations that are both mechanical and semantic. The OPeNDAP server architecture (Hyrax) uses a Lightweight Front-End Server (OLFS) that is responsible for interacting with the requesting client application. The OLFS is extensible and in this project has been extended to support the OGC web service interface specifications. Coupled with the OLFS the Hyrax architecture uses a Back-End Server (BES) to provide data access, processing, and response generation that are then returned through the OLFS to the requesting client. Similar to the OLFS, the BES is extensible and for this project has been extended to support various mechanical actions required in support of the OGC service's request and response interface specification. In addition to the simpler, mechanical aspects required to support these multiple protocols, semantic operations are required in order to interpret request elements and for constructing well-formed OGC responses. To support these semantic operations we've developed ontological representations of the OGC, OPeNDAP, and NetCDF/CF data models, and the relationships between those models. The OLFS has been extended to support XSLT operations transforming OPeNDAP's XML data descriptor (DDX) to a Resource Description Framework (RDF) representation. Modules executing during server initialization ingest the RDF representations and use the ontologies to crosswalk the metadata elements between the protocols.

## II. Methodology

### Requirements

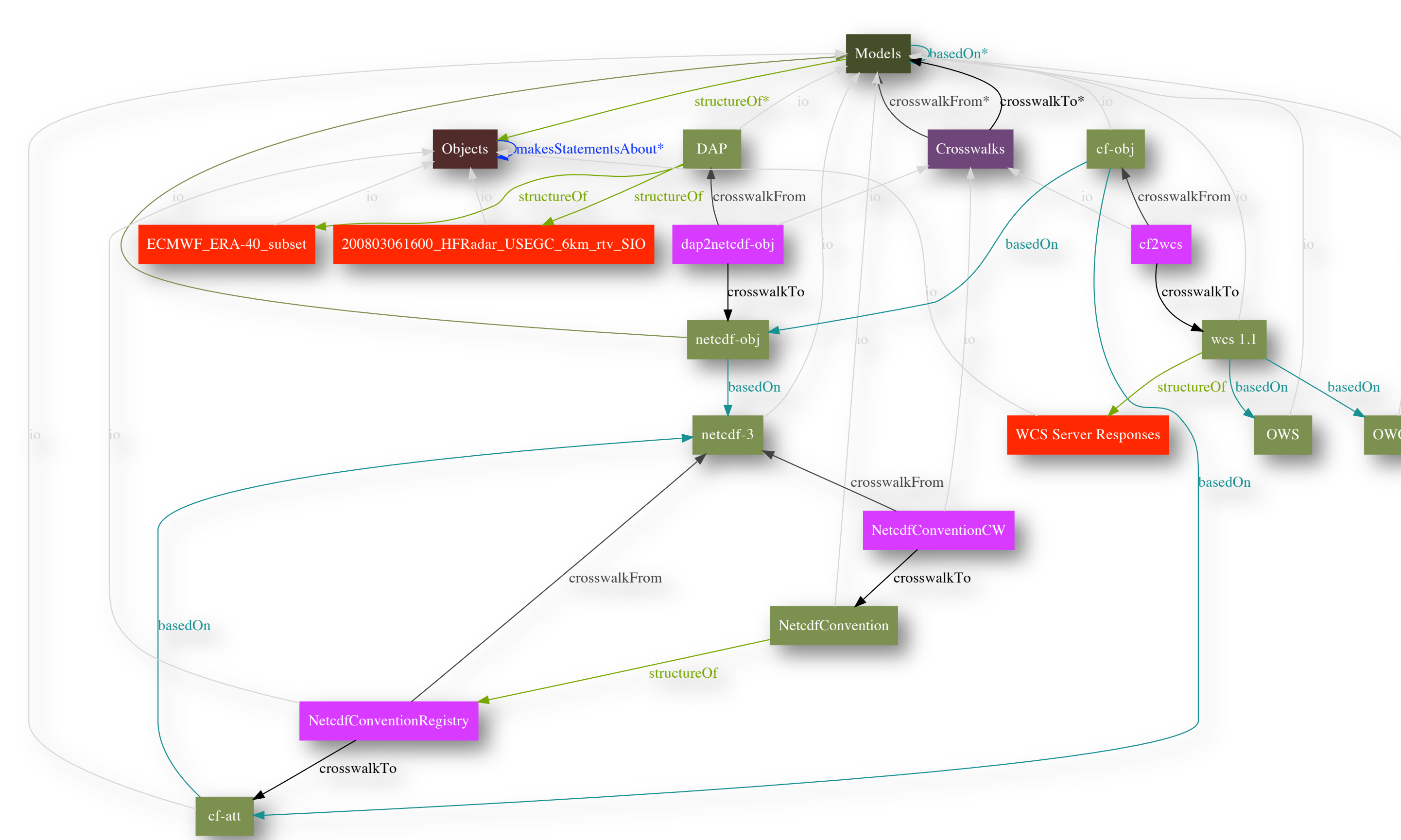
- Scalable – many DAP data providers have large holdings.
- Speed – semantic processing must not unduly increase transaction times.
- Cost – Easily installed solution for existing data provider base.



### Approach

- Utilize existing Hyrax architecture.
  - Write specific “plug-ins” to provide the WCS functionality to Hyrax.
- Use existing semantic web technologies to perform metadata mapping between geo-referenced data sources stored as NetCDF/CF files, accessible via OPeNDAP, onto OGC coverage responses.
  - Use XSLT to convert DDX semantics to RDF in the DAP3.2 namespace. Attribute metadata gets placed in “local” document namespace.
  - RDF/OWL inference/reasoning operations are used to identify CF-1.0 compliant Attributes and promote them to CF namespace and then crosswalk the appropriate items into elements in the WCS namespace.
- Semantic inference/reasoning operations happen asynchronously from client servicing activities.
- Utilize “cube-wise” aggregation to allow compact Coverage representations of large homogenous data holdings.
- Implementation of file-out service allows us to return NetCDF content to WCS clients.

## III. Semantic Mapping



RDF is sufficiently flexible to be able to hold both the information in a DAP DDX document, and the information in a WCS Capabilities XML document. In this diagram we show the semantic path to convert DAP DDX documents into WCS Capabilities.

We organize the ontology documents into three classes: *objects*, *models* and *crosswalks*. *Objects* are collections of entities, e.g. DAP datasets or WCS Coverages. *Models* are the structure (schema) for those entities, e.g. OPeNDAP or WCS. And *crosswalks* are the connections between those structures, containing rules for expressing objects in another model given the first model.

In the figure we have two sample datasets (ECMWF\_ERA-40\_subset and 200803061600\_HFRadar\_USEGC\_6km\_rt\_v\_SIO) which are structured according to the DAP Model. The goal is to serve them as WCS Server Responses in a WCS Service.

While these are DAP datasets, the interpretation of the data as geo-located depends on the DAP Attributes being interpreted according to the Climate Forecast (CF) convention. CF is based on netCDF, so the first mapping is from DAP to netCDF objects (netCDF-obj). In particular, DAP has Grid objects which have attributes, an *n*-dimensional array, and a set of *n* one-dimensional mapping vectors. NetCDF-3 has multi-dimensional arrays with attributes, one-dimensional arrays with attributes, and common dimension names, which together form the corresponding (netCDF-obj) objects. Traditionally, these objects remain implicit, but the netCDF-obj model gives explicit names/structure to these parts of the netCDF conventions, and it is these higher level objects which correspond to objects in other conventions.

In order to have geo-located objects, we need additional semantics on top of netCDF, e.g. the CF conventions. NetCDF has a special attribute “Conventions” to name the convention that governs the meaning of the attributes in a particular file, and the *crosswalk* NetcdfConventionCW connects that attribute to the NetcdfConventionRegistry, a list of conventions (described by ontology files) and their corresponding “Convention” attribute values. Thus the NetcdfConventionRegistry is a *crosswalk* between the netCDF-3 *model* and the corresponding convention *model*; for these example data, the CF convention (cf-att). The cf-att *model* describes the convention at the attribute level, i.e. which attributes are defined, and what are their possible values. However, these attributes imply higher-level objects (e.g. geo-located objects), which are given explicitly in the cf-obj *model*, along with the rules for setting these objects given the information in the cf-att *model*.

To repeat a bit: NetCDF-3 gives the explicit structure of netCDF files, and the cf-att *model* gives the explicit structure of the CF convention. NetCDF-obj gives the higher-level (formerly implicit) structure of netCDF objects (plus rules for setting these higher-level objects from netCDF-3), and cf-obj gives the higher-level (formerly implicit) structure of CF objects (plus rules for setting these objects from cf-att values). Thus both cf-att and netCDF-obj are based on netCDF-3, and cf-obj is in turn based on both cf-att and netCDF-obj.

Now we have geolocated objects in the cf-obj model. The cf2wcs *crosswalk* connects these geolocated objects to their WCS representation, here wcs1.1. Again the key element in the *crosswalk* is finding the corresponding objects. In this case WCS Fields correspond to geo-located netCDF variables, while WCS Coverages mostly correspond to netCDF datasets. This relationship is complicated by the fact that in order for a WCS Coverage to have an identifier (so that it can be requested), it must have a single Coordinate Reference System (imageCRS) -- this requires mapping datasets into multiple Coverages under some circumstances.

We can now generate WCS Server Responses.

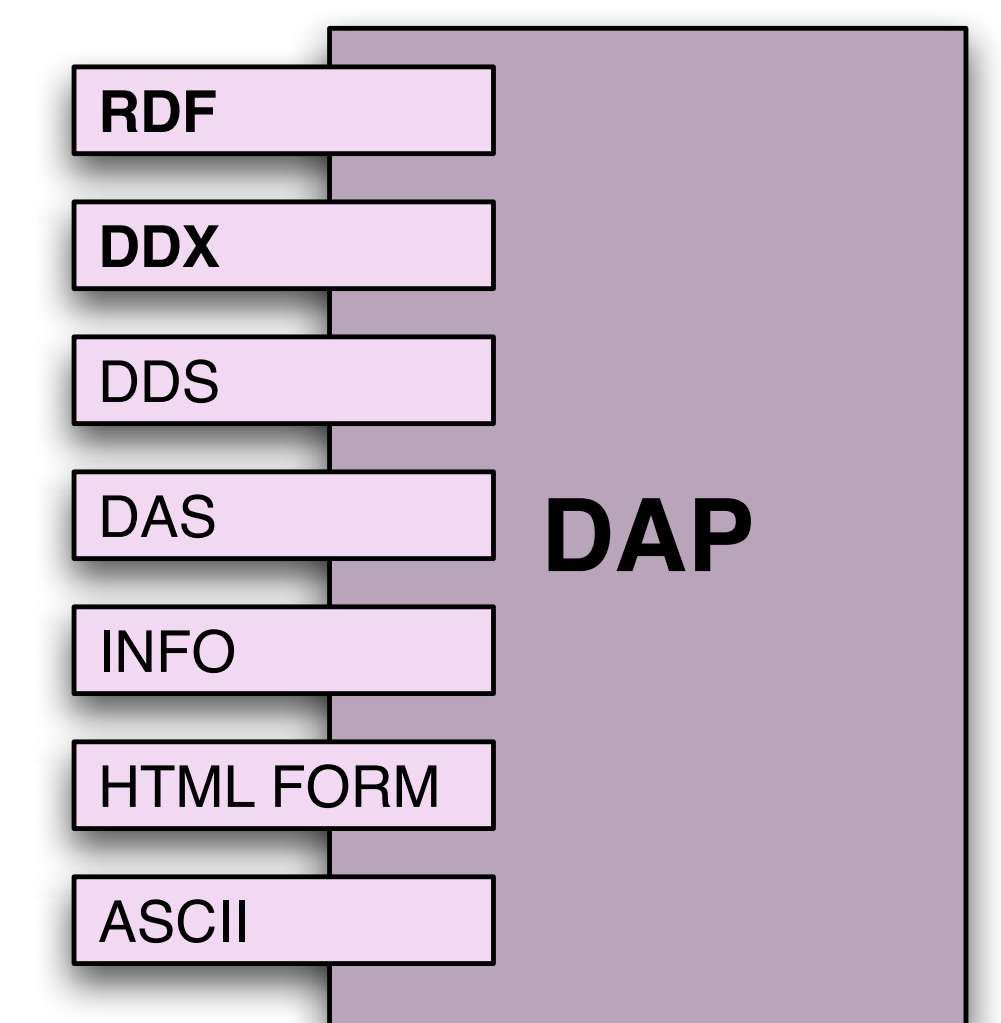
## IV. DAP Internals

### Changes:

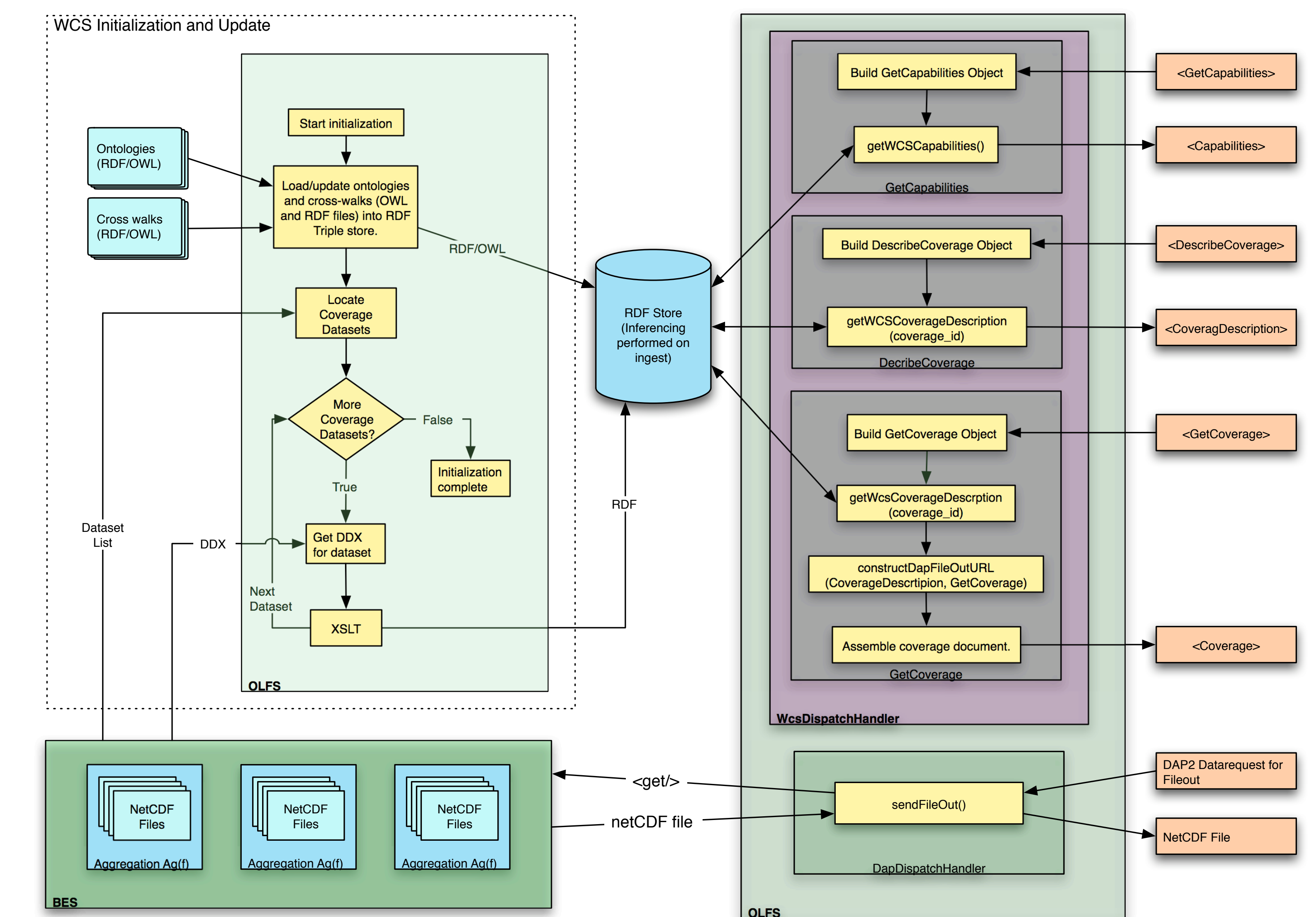
- RDF response:** A new DAP service, the RDF response, has been introduced. A request for this response will return an RDF representation of the data set.
- DDX response:** In support of the RDF representation of the data set, the DDX response (which has been available from Hyrax for several years) has been extended by adding both xml:base and dataset\_id attributes to the dap:Dataset element. This version of the DDX is only available to clients which indicate that they accept DAP version 3.2. The client does this by setting the HTTP header XDAP-Accept, “XDAP-Accept: 3.2”

### Proposed Changes:

- Allow XML content from other namespaces as semantic metadata in the DDX, essentially allowing the DDX to be porous. In order to maintain DDX usability some combination of rules may need to be created:
  - Elements of type dap:BaseType (data set variables) **may** contain elements from other namespaces.
  - Elements of type dap:BaseType (data set variables) **may not** be the children of elements in another namespace and still be seen as part of the top level dap:Dataset element's regular DAP semantics.



## V. Server Internals



## VI. Development Plan

- Currently semantic inference/reasoning steps are not integrated into the OLFS plug-in, but are running asynchronously.
  - Integrate semantic inference/reasoning into OLFS plug-in (probably as a service thread).
- Crosswalk and ontology development are ongoing. (Incomplete/invalid WCS schemas create significant extra work as, automated ontology builders cannot be utilized with the schemas.)
- Discovery and integration of absent geo-referenced metadata. For many of the datasets, the RDF ingestion step will require some degree of data set interrogation.

## Acknowledgements

Funding for this effort is provided by NOAA's National IOOS Development effort initiated in 2007. This effort initiated a competitive funding process to continue building capacity for regional ocean observing systems, and in particular this effort contributes toward developing applications for regional stakeholders, and establishing regional and national data management and communication protocols.