DataONE/DAP Server

James Gallagher*

*With lots of help from the DataONE developers
About DataONE

● Web servers provide access data
● They also provide a data location service
● And an 'upload' service
  ○ data are added using the servers
  ○ it's not a read-only system
More About DataONE...

- DataONE supports authorization and ...
- Anonymous access
- It also supports Data Replication
- Supports dataset versions and ...
- Old versions are not deleted
Foremost a Federation

Member Nodes (MNs)
- Heart of the federation
- Harness the power of local curation

Coordinating Nodes (CNs)
- Services to link Member Nodes

Investigator Toolkit (ITK)
- Tools for the whole data lifecycle

*DataONE in a Nutshell*

*Thanks to Matt Jones for this slide*
What DataONE* looks like

*Stolen from the DataONE web site
Software Components that make up DataONE*

*Stolen from the DataONE web site
The DataONE Member Node APIs

Tier 1: Core, Read, [Query, View, Package]
  Get Data and information about data

Tier 2: Authorization
  Limit access to the server or specific data sets

Tier 3: Storage
  Add data to the Member Node

Tier 4: Replication
  Instruction from a Coordinating Node to replicate data
What We Did

- Tier 1 Member Node
  - Core and Read APIs
- It will work with *any* DAP server that can return ISO 19115 docs and netCDF
- Open Source*
- Small: ~700 LOC; 8 source files
- Not 100% complete

*https://scm.opendap.org/svn/trunk/D1
How does it work?

- Servlet processes D1 requests
- A request is handled using information from the database and/or the DAP server
  - The DB holds 'D1 metadata'
  - The DAP server holds dataset data/metadata
- A separate program is used to add datasets to the database
Operation - setup, harvest

- Use the standalone program to add datasets to the database
  - This will use information from the DAP server to build D1 'system metadata'
- A D1 Coordinating Node (CN) asks the Servlet for information about its datasets (system metadata)
- The CN then reads the dataset metadata
Operation - clients

- Clients search the CN for datasets
- They find the URLs for data the server holds
- They get the Data and Metadata
Limitations - implementation

- Only ISO 19115 dataset metadata
  - but D1 supports many other formats
- All data returned in netCDF3 files
  - again, D1 supports other formats
  - more about this in a bit...
- Many features of DAP are not used
- Only D1 Tier 1 and only the required APIs
A few other limitations

- The program that adds datasets to the database is very limited
- Some very large datasets will fail
- The ping() function of the D1 Core API may return misleading (false positive) results - it tests only one DAP server
- However, these are not the most important limitations
System Differences

- DAP and DataONE solve different problems
- DAP is a protocol
  - format independence
  - subsetting
- DataONE is a system
  - persistence (replication, version management)
  - dataset location (CN/MN architecture, metadata harvesting)
- This makes combining them interesting!
The Hard Parts

● Subsetting
● Aggregation
● Persistence
● Cataloging
Subsetting

- DAP supports subsetting using a data model like most programming languages
- To use this, DAP's 'system metadata' describes a dataset's internal *structure*
- DataONE's metadata describes the internal *content*
- Solution: Add service APIs as a datatype to DataONE (but that's not simple)
Aggregation

● Some DAP datasets are large collections of files: $O(10k)$
  ○ Aggregation supported for a some of these
  ○ Access to the entire dataset not pragmatic - it must be subset

● DataONE uses ORE (Object Reuse and Exchange)
  ○ Aggregates more types of datasets
  ○ Cannot support large numbers of files
Persistence

- DAP knows nothing about versions - that idea is out of scope for the protocol
- DAP is an access protocol, not a system, so it cannot manage lifetimes
- We could (should?) combine DAP with other tools to add these capabilities
Cataloging

- DAP servers typically have catalogs that are hierarchical
  - ERDAP is a system that uses relational catalogs
- DAP catalogs do not support relational queries
- DataONE depends on relational queries to crawl servers and provide clients with a data location service - DAP has none
Conclusion: Subsetting and Aggregation

- DataONE could be extended to support service APIs - as a 'return type'
- But, this is a major change for the clients
- Aggregation can also be a service API
- But the return type will be a set of URLs that will then be accessed in succession
Conclusion: Cataloging and Persistence

- Existing DAP catalogs can be crawled and a RDB populated
- The RDB can support a query interface
- This RDB can support persistence
- There is no likely (short-term) solution for DAP datasets that are deleted
- Replication of large datasets is not pragmatic