DataONE – Virtual Data Center

2 – 4 June 2009 Technical Working Group Meeting Report

Submitted by:

Technical Working Group,
Virtual Data Center

7 August 2009

Revision History

Date	Version	Description	Author
15 June 2009	1.0	Original document draft.	M. Servilla
14 July 2009	1.0	Adding main content.	M. Servilla
20 July 2009	1.0	Completion of full first draft.	M. Servilla
3 August 2009	1.0	Added edits/comments from Ryan Scherle and Bruce Wilson.	M. Servilla
7 August 2009	1.0	Added edits/comments from Bob Sandusky and Matt Jones.	M. Servilla

Table of Contents

Attending 3
Summary 3
1 Detailed Discussions 4
1.1 Use Case Interaction Diagrams 4
1.2 New Use Case Scenarios 5
1.3 Coordinating Node Requirements 6
1.4 Identity Management, Authentication, and Authorization 7
1.5 Metadata Standards 8
1.6 Search Terms 10
1.7 Globally Unique Identifiers 11
1.8 Prototypes 14
2 Student Internship Presentations 15
2.1 Refactoring the EarthGrid SOAP API to REST style for Metacat – Akin 15
2.2 Generating Accurate Ranking Algorithms via Machine Learning – Dumoulin 15
2.3 Semantic Phyloinformatic Web Services using the EvoInfo Stack – Harney 16
2.4 Vocabulary Term Mapping - Lele 16

Attending

Investigators: Paul Allen, Duane Costa, Matt Jones, Jeff Horsburgh, Giri Palanisamy, Bob Sandusky, Ryan Scherle, Mark Servilla, Dave Vieglas, and Bruce Wilson

Students: Serhan Akin, Christine Dumoulin, John Harney, and Namrata Lele

Summary

A meeting of the DataONE-Virtual Data Center project Technical Working Group was held during 2 – 4 June 2009 at the University of New Mexico in Albuquerque, New Mexico. Topics discussed at this meeting include: (1) issues related to use case scenarios/diagrams developed during the January 2009 meeting in Santa Barbara, California; (2) development of new use case scenarios; (3) functional/non-functional requirements of Coordinating Nodes; (4) issues related to identity management, authentication, and authorization; (5) relevant metadata standards; (6) terms relevant to portal search interface fields; (7) issues related to globally unique identifiers; and (8) identifying and prioritizing prototype candidates. In addition, time was provided to interact with the Cyberinfrastructure Summer Traineeship students, including receiving presentations from each student and providing feedback on their individual projects.

1. Use Case Interaction Diagrams – A review of the current use case diagrams revealed inconsistencies in presentation and naming conventions, along with repetitive diagram components that may be abstracted to reduce clutter. Duane Costa will consolidate and unify existing and new use case diagrams. Level of effort is estimated at two weeks, which will take place during June 2009.

2. New Use Case Scenarios – Nineteen new use case scenarios were identified prior to and/or during the meeting. Duane Costa will draft new diagrams for each of the new use cases (see above effort).

3. Coordinating Node Requirements – Approximately twenty functional and non-functional requirements were identified for Coordinating Node services. This list is not complete, and more are expected to be specified in the ensuing months. In general, Coordinating Nodes should be robust, extensible, fault-tolerant with fail-over, secure, and simple to deploy.

4. Identity Management, Authentication, and Authorization – Six identity management/security models were identified for DataONE/VDC, along with methods for verifying identity, types of identity roles (including special roles), objects to be controlled, and models for expressing access control.

5. Metadata Standards - Twenty metadata standards were identified that are in use in the environmental communities (many more likely exist). Issues related to converting from one standard to another were discussed, most importantly the fact that most conversions result in “loss” of information. As such, it was decided that storing both key elements consistent across most standards and used primarily for

discovery purposes and the native format of the original metadata would be crucial for DataONE/VDC.

6. Search Terms – Eight search terms were deemed critical for a DataONE/VDC portal search interface, including (1) author(s)/principal investigator(s), (2) Keyword(s), (3) key concept, (4) spatial bounding box, (5) spatial windows, (6) named places, (7) temporal window, and (8) text search. An additional six terms were considered desirable and another eleven were identified as useful.

7. Globally Unique Identifiers – It was decided that DataONE/VDC will require that Member Nodes provide all data objects with a globally unique identifier (GUID) that meets a single criteria – the data object GUID must be unique across the DataONE/VDC domain space. As such, Member Nodes may use the GUID scheme of their choice. A Member Node may identifying a "replicated" data object with their local GUID scheme.

8. Prototypes – Thirteen prototyping tasks were identified; these were subsequently ordered based on priority. The highest priority tasks include (1) when a Member Node contributes a new data package (metadata and data) identified by a GUID to the Coordinating Node and (2) when a Coordinating Node initiates a data package replication from one Member Node to another Member Node.

1 Detailed Discussions

The following sections document elements of detailed discussions that occurred during the Technical Working Group meeting of the DataONE - Virtual Data Center project during 2 – 4 June 2009.

1.1 Use Case Interaction Diagrams

An initial set of fifteen use case scenarios, along with their interaction diagrams, were identified at the Technical Working Group meeting held in January 2009 at the National Center for Ecological Analysis and Synthesis (NCEAS) in Santa Barbara, California. Participating members volunteered to design diagrams for one or more of the scenarios. The resulting set of diagrams were inconsistent in presentation and terminology when viewed collectively. As such, it was decided that a single person should refactor all diagrams with an industry standard tool for consistency and simplification (where possible). Duane Costa was appointed to this task and estimated that the level of effort would require approximately two weeks. The list of original use case scenarios topics are:

1. Get object identified by GUID (authenticated or not, notify subscriber of access).
2. Get list of GUIDs from metadata search (anonymous and authenticated).
3. Registration of a new Member Node.
4. Create/Update/Delete metadata record in Member Node.
5. Create/Update/Delete data object in Member Node.
6. Replicate / synchronize metadata record between Member Node and Coordinating

Node.

7. Batch Operations - Coordinating Node requests metadata /data list from new Member Node and then batch upload (disable indexing for example to improve insert performance).
8. Communication of replication policy metadata among Member Nodes and Coordinating Nodes.
9. Replicate data from Member Node to Member Node -- (facilitated by Coordinating Node or independently but notifying Coordinating Node of operation).
10. Coordinating Node checks "liveness" of all Member Nodes - checks ping, service x, load, space, bandwidth, transaction rate, …
11. Create/update/delete/search workflow objects.
12. User Authentication - Person via client software authenticates against Identify Provider to establish session token.
13. User Authorization - Client requests service (get, put, query, delete, ...) using session token.
14. System Authentication/Authorization - Server authenticates and performs system operations (e.g., replication).
15. User Account Management - Create new user account on Identity Provider (also edit, delete, ...).

1.2 New Use Case Scenarios

There were nineteen new use case scenarios identified either before and/or during the meeting that will require the development of corresponding interaction diagrams. Duane Costa will attempt to generate diagrams for these new use cases. The new use case scenario topics are:

16. All CRUD operations on metadata and data are logged at each node.
17. All CRUD logs are aggregated at coordinating nodes.
18. Member nodes can request aggregated CRUD log for (time period/object id/userid) for all of 'their' objects.
19. General public can request aggregated download usage information by object id.
20. Data owners can request aggregated CRUD log for (time period/object id) for all of 'their' objects.
21. Data owners can subscribe to notification service for CRUD operations for objects they own.
22. User can get report of links/cites my data (also can view this as a referrer log).
23. User can find out where all copies of my data are in the system and can expunge them.
24. Transactions - Coordinating Nodes and Member Nodes should support transaction sets where operations all complete successfully or get rolled back (e.g., upload both data and metadata records).
25. System should scans for damaged/defaced data and metadata suing some validation process.

26. System performs data quality checks on data.
27. Coordinating Node should support forward migration of metadata documents from one version to another within a standard and to other standards.
28. Relationships/Versioning - Derived products should be linked to source objects so that notifications can be made to users of derived products when source products change.
29. Load Balancing - Requests to coordinating nodes are load balanced.
30. Member Node can notify Coordinating Node about pending outages, severity, and duration, and Coordinating Nodes may want to act on that knowledge to maintain seamless operation.
31. Client can specify access and replication restrictions for their data and metadata objects, and support timed embargoes.
32. User or organization takes over 'ownership' of a set of objects (write access for orphaned records).
33. Clients should be able to search for data using Coordinating Node metadata catalogs.
34. Coordinating Nodes publish metadata in formats for other discovery services like Google/Libraries/GCMD/etc.

In addition to the discussion regarding individual use cases, the concept of using a standard (such as the Interface Definition Language) specification for method signatures was explored as possible action for finalizing use case descriptions.

1.3 Coordinating Node Requirements

Because the Coordinating Nodes act as the central organizing entity within DataONE/VDC, it is critical to identify both functional and non-functional requirements early on. A “non-exhaustive” list of twenty-one requirements were identified during the TWG meeting:

1. Data packages are not discoverable through any public interface until all Coordinating Nodes have confirmed that they have a copy of the corresponding metadata document.
2. Metadata searches should return in a maximum of “xxx” seconds.
3. Coordinating Nodes can store and search greater than “xxx” metadata records.
4. Coordinating Nodes can store and search multiple metadata standards (see list of metadata formats of interest).
5. Coordinating Nodes can load-balance to maximize performance.
6. Any “xxx” number of Coordinating Nodes can be off-line without affecting system services.
7. A “xxx” number of transactions can be supported by the system (the actual number may vary depending on the type of transaction – e.g., delete versus insert).
8. Coordinating Nodes should be available an “xxx” percentage of time.
9. Coordinating Nodes should make available new metadata for discovery within “xxx” seconds of an insert by a Member Node.

10. Coordinating Node services should be designed to be independently scalable.
11. Coordinating Node services should be geographically replicated.
12. Coordinating Nodes should have complete metadata copies from all Member Nodes.
13. Collectively, Coordinating Nodes and Member Nodes (and therefore DataONE) should be compliant with criteria for trusted repositories per TRAC (Trustworthy Repositories Audit & Certification) and/or DRAMBORA (Digital Repository Audit Method Based on Risk Assessment).
14. Coordinating Nodes should respect replication policies of Member Nodes.
15. Coordinating Nodes should ensure that any given data set is available at any time (subject to policy restrictions) even when an "xxx" percentage of Member Nodes are off-line.
16. Coordinating Nodes must validate that (1) data are available at all replicating Member Nodes and (2) all replicas are valid.
17. Coordinating Nodes should ensure that data are available in current formats (i.e., provide support for data / metadata format migration).
18. Coordinating Nodes should be economical to run and maintain.
19. Coordinating Nodes should be able to be remotely administered.
20. Coordinating Nodes should be secure and deflect malicious intent.
21. Coordinating Nodes should provide services to detect rogue data (e.g., containing viruses, violate copyrights).

1.4 Identity Management, Authentication, and Authorization

The DataONE secutiy services provided by both Coordinating Nodes and Member Nodes are necessary to preserve and verify the integrity of all data packages held within the system, but also to prevent malicious intent or access to data packages that should not be available to the general public. The Technical Working Group identified six identity management/security models that are currently used in either industry or associated communities:

1. Centralized directories (LDAP),
2. Distributed directories (LDAP with referrals),
3. Distributed management and replication (LDAP with replication),
4. Grid Security Infrastructure (GSI) proxy certificates,
5. OpenID, and
6. Shibbileth + InCommon.

Of these, none were identified as the preferred model for DataONE/VDC. Further analysis and discussion is required to identify the model(s) of choice.

There were three different methods for verifying identity noted in discussions:

1. Self registered user (with valid email, response message, etc.),
2. A "real" person who can be verified by an external authority, and

3. A person who is a member of a verified “role”.

Further discussions identified issues about authorization and access control, including identifying principals who should be subject to some level of access control: (1) user, (2) group member, (3) public, (4) authenticated user, (5) site manager (for harvests, system operations, etc.), (6) change request approval workflow, and (7) groups who own intellectual property rights.

Items in need of protection include: (1) both data and metadata (e.g., read, write, change permissions), (2) Member Node Write, (3) the ability to make or execute certain system functions (e.g., register Member Node), and (4) logged information.

Only two methods of expressing access control permissions were discussed: (1) the Security Assertion Markup Language (SAML) and (2) the access control mechanism used by the Ecological Metadata Language (EML). No decisions were made regarding methods to express access control permissions.

1.5 Metadata Standards

Twenty different metadata standards that are commonly used in the ecological and environmental communities were identified during the meeting; it was noted that many more metadata standards exist, but were not identified, and that transforming from one standard to another standard is almost always “lossy”. No specific standard was adopted for DataONE/VDC, however, it was recognized that DataONE/VDC will have to store “key” elements that are consistent across the most commonly used standards. These “key” elements are likely usable only for discovery purposes. In addition, DataONE/VDC should store the native metadata along with any new or augmented content. Further investigation will be required to determine “key” elements. The list of noted metadata standards (in no particular order) is:

1. Dublin Core (DC) – A general purpose interoperability metadata standard used for a broad range business models (http://dublincore.org).
2. Darwin Core (DwC) – An XML grammar to facilitate the exchange of information about the geographic occurrence of organisms and the physical existence of biotic specimens in collections (http://www.tdwg.org/activities/darwincore).
3. Ecological Metadata Language (EML) – An XML grammar for describing and documenting ecological data (http://knb.ecoinformatics.org/software/eml).
4. Federal Geographic Data Committee (FGDC) Content Standard for Digital Geospatial Metadata (CSDGM) – A standard to provide a common set of terminology and definitions for the documentation of digital geospatial data (http://www.fgdc.gov/metadata/csdgm).
5. Global Change Master Directory (GCMD) Directory Interchange Format (DIF) – A metadata standard for describing and documenting Earth science data sets that are stored as part of the GCMD (http://www.esdswg.org/spg/docindexfolder/heritage/gcmd-dif).
6. International Standards Organization (ISO) 19137:2007 - Defines a core profile of the spatial schema specified in ISO 19107 that specifies, in accordance with ISO 19106, a

minimal set of geometric elements necessary for the efficient creation of application schemata.

7. NeXML (NEXUS file format in XML) – An XML grammar that is based on the NEXUS file format for use in describing and documenting phylogenetic data (http://www.nexml.org/)
8. Water Markup Language (WaterML) – An XML grammar that describes hydrologic-based data sets and originally defined by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (http://his.cuahsi.org/wofws.html).
9. Genbank Flat File Format – A structured text-based format used to document genetics data that has been submitted to GenBank (http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html).
10. International Standards Organization (ISO) 19115 - Defines the schema required for describing geographic information and services. It provides information about the identification, the extent, the quality, the spatial and temporal schema, spatial reference, and distribution of digital geographic data.
11. Dryad Application Profile (DAP) – An XML grammar, based on the Dublin Core Metadata Initiative Abstract Model, to describe datasets underlying published works in evolutionary biology and related fields (https://www.nescent.org/wg_dryad/Application_Profile_Development).
12. ADEPT/DLESE/NASA (ADN) - The purpose of the AND metadata framework is to describe resources typically used in learning environments (e.g., classroom activities, lesson plans, modules, visualizations, some datasets) for discovery by the Earth system education community (http://www.dlese.org/Metadata/adn-item).
13. Geography Markup Language (GML) Profiles – An XML grammar defined by the Open Geospatial Consortium (OGC) to express geographical features (http://www.opengeospatial.org/standards/gml).
14. Common Data Format (CDF) – A common data storage format standard, originally developed by NASA, that is used in a number of applications, including netCDF and HDF (http://cdf.gsfc.nasa.gov/).
15. Data Documentation Initiative (DDI) – An international standard for describing social science data that is defined by an XML grammar (http://www.ddialliance.org/).
16. Gene Expression Markup Language (GEML) – An XML grammar for storing DNA microarray and gene expression data.
17. Earth Science Markup Language (ESML) – An XML grammar that defines an data interchange format and used primarily in the Earth sciences community (http://esml.itsc.uah.edu/).
18. Cruise Summary Report (CSR) (formerly, Report of Observations/Samples collected by Oceanographic Programmes – ROSCOP) – An international standard for describing oceanographic data collected on research vessels and supported by the International Council for the Exploration of the Sea (https://www.ices.dk/ocean/roscop).
19. Earth System Grid (ESG) – A XML structure used for storing and distributing climate model data results.
20. Earth Observing System Clearinghouse (ECHO) – A second NASA-based metadata format, used primarily for remote sensing data, including field calibration and

validation data. ECHO differs from GCMD in that GCMD holds collection-level (data set) metadata, while ECHO is capable of holding file-level (granule) metadata.

Some of these formats (e.g. Darwin Core and EML) can be used in ways where the relevant data and metadata are in a coherent whole. A Darwin Core record for bird observational data, for example, can contain all of the metadata and all of the available data. This raises some challenges for metadata replications, since that replication may thereby also involve replication of the underlying data.

Note also that additional metadata formats exist for data services, such as a WSDL that can describe the data available through that service or an open geospatial consortium (OGC) capability description metadata record.

1.6 Search Terms

The DataONE/VDC TWG realized the importance of data discovery through a user interface and, therefore, discussed the need to narrow a set of descriptive metadata elements that should be used for the initial discovery interface for data held within the DataONE/VDC domain. As such, eight terms were labeled critical, with an additional six terms identified as important, and another eleven identified as potentially useful. The terms would be used on the DataONE/VDC website portal and focus specifically on data discovery.

Critical terms include:

1. Principal Investigator/Author – The TWG recognized the need for a "names authority" or "canonical naming" service where people's names are used consistently when referenced to a digital object. For searching, however, all agreed that simple text representation of names is critical in order to accommodate existing, widely-used non-canonical names, with the goal of becoming more controlled over time .
2. Keywords – Allowing free form or natural language searches without controlled vocabularies.
3. Key Concept – Key concepts drawn from a set of ontologies for both terms and different namespaces.
4. Spatial Bounding Box – Largest bounding rectangle.
5. Spatial Window – To consist of a series of spatial envelopes representative of the spatial locations where data is collected from/relevant to. For example, spatial features (points, centroids, lines, polygons, and bounding boxes) that may or may not be contained within a the largest bounding rectangle.
6. Named Places – Including terms, types, context (e.g., Columbus OH, Columbus GA), and gazetteer.
7. Temporal Window – Including date ranges, temporal coverage of the data set (e.g., during, before, and after), and relative terms (i.e., geologic-timescale).
8. Abstract/Full Text Search

Important or desirable terms include:

9. Title – Dataset or associated title.
10. Data Format – Including original data, summarized versions, processing methods (to generate format), resource type (e.g., spatial, models, observations, web service, etc.).
11. Scientific Variables – Should be from a controlled vocabulary for efficacy.
12. Subject Domain – For example, physics, geology, biology, etc...
13. Biological Taxonomic Extents
14. Associated Publications

Potentially useful terms include:

15. Data Source – The physical source of the data (e.g., instrument, application, model, etc.).
16. Related Data – Associated data products such as derived data and/or publications.
17. Data Quality/Level of Curation
18. Organizational Domains – Primary and/or secondary organizations associated with the data object.
19. Size of Data
20. Number/Location of Replicates
21. Data Dimensionality
22. Scientific Units
23. Globally Unique Identifier
24. Temporal Window (low priority) – Additional and/or associated temporal terms (e.g., publication date, creation date, last modified date, etc.).
25. Object Permissions

1.7 Globally Unique Identifiers

The DataONE/VDC TWG acknowledged great importance to the use of "globally unique identifiers" (GUIDs) (e.g., Digital Object Identifiers (DOIs), Handles, Life Science Identifiers (LSIDs), and Persistent URLs (PURLs)) for any and all data/metadata objects stored within the DataONE/VDC domain space and, therefore, take the position that Member Nodes will be required to use GUIDs for all data/metadata objects submitted to DataONE/VDC. The TWG also recognize that Member Nodes will likely adhere to their current GUID scheme and be unwilling to change or modify this scheme to participate in DataONE/VDC, particularly in the initial years of DataONE/VDC. As such, the TWG perspective is such that DataONE/VDC should accept data/metadata objects from Member Nodes that are correctly identified by a GUID assigned through a local scheme with the caveat that the applied GUID must be unique within the DataONE/VDC domain space. This particular caveat implies that

Coordinating Nodes must be (1) capable of determining GUID uniqueness and (2) reserve a specific GUID on behalf of a Member Node while the Member Node concludes processing steps used to insert a data/metadata object, but before the object is fully registered in DataONE/VDC.

A complication resulting from the TWG's indifference toward a specific GUID scheme surfaces with the need to replicate a data object to another Member Node that utilizes a different GUID scheme from the original scheme. In this case, the replicated data object may be stored in the GUID scheme of the replicating Member Node as long as the object may be resolved using the original GUID and/or a specific DataONE/VDC identifier. With regard to replicates of data object revisions, the TWG discussion concluded that one possible solution would be that the Member Node will not be required to store replicates of revisions, but they should be able to identify the existence of previous revisions. The alternative is that the Member Node must store all revisions regardless of the number of such revisions.

A number of additional questions for further discussion were considered:

- If a Member Node uses a DOI for a data set identifier, is it appropriate to include doi: in the identifier. For example, 10.3334/ORNLDAAC/840 is the DOI for a particular data set at the ORNL DAAC. Both doi:10.3334/ORNLDAAC/840 and 10.3334/ORNLDAAC/840 can be presumed to be unique identifiers. Which should be used? One option is to use the one with the resolution protocol included, which makes it a “smart identifier” (some members noted the potential for problems by using the full resolution protocol, but did not elaborate on the issues).

- Where an identifier has a mechanism to resolve to multiple locations (such as is possible with an LSID and some DOI mechanisms) and that object is replicated from one Member Node to another Member Node, this would suggest that the originating Member Node needs to be notified of the additional location and has the option of registering the new location with the handle registration authority. This also means that if a replication is removed, the original Member Node should have the option of being notified, so that the resolution points are updated. Ideally, this should happen before the replica is removed (where possible), so that we eliminate (or at least minimize) the amount of time that an invalid resolution point is in another system.

- Where an identifier (such as a Handle) has a URL resolution, what should that resolution be? An consensus among members of the TWG is that it is far better to resolve to a machine-interpretable version of an object, as it is easy to derive a human interpretable version from a machine interpretable one. It is very difficult and often impossible to go the other way -- from human interpretable to machine interpretable. There is, however, concern that a machine-based identifier is too ambiguous for a user interface, and one member opinion is that resolution to a human interpretable description of the object is more important than a machine interpretable resolution. An example is that of the ORNL DAAC, where DOI's resolve to a web page where a user (after logging in) can see and download the components of the data set. Some thought and guidance on this point for the overall

DataONE/VDC community of practice is desirable.

- Do we want/need a registry of name spaces? Where a Member Node uses a UUID (for example), there may not be a way to describe the name space for identifiers, unless the Member Node prefixes the UUID with some descriptor, which generally violates the general admonition about smart identifiers. It might, however, be helpful to have something like a set of "regular expressions" that describe the name space for a Member Node's identifiers, particularly if an automated way could be developed to look for potential collisions (non-null overlaps) between name spaces. One view is that this is far from an initial feature, but the desirability of this as a possible future feature could have implications on the way we do things from the start.

- Can the metadata standards support multiple globally unique identifiers? For example, what happens in the case that a Member Node starts down the DOI path and then switches to LSID's because of economic costs, for example, and goes back and assigns an LSID to historical data sets. Those data sets now have both an LSID and a DOI. Where is this in the metadata? Is there a mechanism for indicating the preferred ID and the alternate ID's? Likewise, how should things be handled when a Member Node decides to register an object with, for example, Global Change Master Directory (GCMD) and the namespace that GCMD allows for identifiers does not allow for the Member Node's preferred identifier? Can a Member Node update the metadata to show an alternate key with the GCMD identifier (data set is also known as)? What is the implication for the metadata identifier in such a case? This is an update operation to the metadata, which implies that the metadata identifier is changed. How would one update the old metadata record to indicate that it is a) deprecated and b) the id of the new metadata record? The above also relates to the issue of establishing predecessor-successor relationships between identifiers and the overall issue of provenance. How should this be done across the system?

- For identifiers, we may need to specify the character space. What happens when a Member Node stores unique identifiers in a database field that supports just ASCII, but a different Member Node does its unique identifiers in some other character set? PURL is a possible unique identifier, but we can get into cases now where URL's have characters from other language character sets (such as, Arabic, Kanji, etc.).

- What happens when a request for a replicated version of a data set comes to the replicate Member Node and the data set has been updated and the originating Member Node has not supplied the information about the update (e.g. they did an insert for the new version)?

- How do we assign ID's for a continuous data stream or for a subset calculated on the fly? Does this mean that every request for a continuous data stream gets its own data set identifier, which then gets stored in the DataONE/VDC system someplace? What is the value to the overall enterprise for storing the data set identifiers for each request, particularly in the context of something like a stream, where the on-the-fly processing is used to get a dynamic subset or dynamic re-projection? Examples of this sort of situation include the stream gauge data or the Atmospheric

Radiation Measurement (ARM) archive. Ameriflux Flux tower data is a simpler case, in that they work on the basis of a site-year as a unit of data. The World Oceanic DataBase (WODB), however, operates on a location (and possibly depth) as a unit of data. Many of these are updated quarterly. Each unit of data has an identifier, unique within WODB, and WODB publishes a data stream that indicates what data packages were updated at what point in time. It is possible to determine whether a particular data package changed between two points in time. The differences are human interpretable, but it is not possible (in any generally automated fashion) to recreate the data stream for a particular data package at an arbitrary point in prior time.

- Do the Coordinating Nodes need a method to determine the object type for an identifier?

- Should identifiers hold version information? Having different identifiers for each version of a data object (commonly known as "strict" or "forever-citable") makes it easier to tell the versions apart, and makes it much simpler to manage replication. But, creating a new ID for each version of an object could be technically difficult for many member nodes. Some software (e.g., Fedora, SVN) tracks versioning information internally, while the basic identifier represents only the most recent version (called a "non-strict" identifier scheme).

1.8 Prototypes

The DataONE/VDC TWG identified thirteen potential prototype scenarios/tasks that could be codified and evaluated as part of the VDC technical goals. The following prototype scenarios/tasks are prioritized in order of highest importance with the first four to be addressed prior to the November 2009 meeting (level of effort is estimated for some):

1. Member Node contributes metadata to Coordinating Node using GUID – a) for two or more underlying Member Node software systems (e.g., Metacat, Mercury, and/or Dryad) beginning with two similar systems to achieve working implementations, and then on two very dissimilar systems to uncover boundary condition complexities; b) for two or more underlying GUID schemes; c) for two or more metadata formats; d) define and implement a “READ” interface and provide a client application that can be integrated into a test module; estimated time approximately 3 months for 1 FTE.
2. Coordinating Node initiates replication of data object from one Member Node to another Member Node.
3. Logging for instrumentation and usage.
4. Update data object (revision) by Member Node; estimated time approximately 2 months for 1 FTE for items 2-4.
5. Replication of metadata and system information between Coordinating Nodes; address questions of operational requirements (e.g., bandwidth, response time, etc.) and latency for synchronization; estimated time approximately 1 month for 1 FTE.
6. Failover and load balancing between Coordinating Nodes.
7. Formalize all service API specifications identified by Use Case Interaction Diagrams

using a language agnostic Interface Description Language.

8. Comparison and evaluation of existing systems/standards/protocols used by prototype implementations.
9. Authentication and authorization using LDAP (Lightweight Directory Access Protocol) as an initial protocol (see Section 1.4).
10. Search portal user interface using Coordinating Node metadata content, including search indexing framework.
11. Heartbeat/state of health services (e.g., Hobbit, Nagios, Big Brother, Cacti, etc.).
12. Registry services using, perhaps, a simple list as an initial method.
13. Stress and load testing.

2 Student Internship Presentations

The DataONE/VDC internship program, modeled after the "Google Summer of Code" program, has enlisted four interns :

- Serhan Akin (Mentor – Matt Jones)
- Christine Dumoulin (Mentor – Bruce Wilson)
- John Harney (Mentor – Hilmar Lapp)
- Namrata Lele (Mentor – Dave Vieglas)

Each of the four interns has outlined and prepared project abstracts, detailed plans, and schedules that were presented to the TWG. The following are brief summaries of the presented projects.

2.1 Refactoring the EarthGrid SOAP API to REST style for Metacat – Akin

EarthGrid (EcoGrid) is a lightweight API which provides SOAP based communication of several types of client softwares with the data server applications. This project involves refactoring current SOAP based Earthgrid API to REST style that has certain benefits over SOAP. Then, this REST API will be implemented in the Metacat data management system. It will be a prototype for client software using the Earthgrid API such as Morpho and Kepler.

2.2 Generating Accurate Ranking Algorithms via Machine Learning – Dumoulin

This project aims to generate effective ranking algorithms using random combinations of functions from the Natural Language Toolkit (NLTK). A genetic algorithms approach will allow the testing and assessment of large numbers of combinations.

2.3 Semantic Phyloinformatic Web Services using the EvoInfo Stack – Harney

The evolutionary informatics working group at NESCent has produced a stack of deliverables to promote interoperability in evolutionary analysis. Among these are the comparative data analysis ontology and the NeXML exchange format. In addition, at a recent hackathon at NESCent, supporting tools were produced to transform NeXML to CDAO RDF/XML triples. Subsequently, at a hackathon in Japan, it was shown that these technologies can be leveraged for the development of semantic web services, which would open up a rich field of possibilities, including smarter service discovery and complex reasoning over web service workflows. This project brings these parts together to implement a proof-of-concept web service that accepts and emits NeXML in such a way that a standards-compliant client (e.g. SADI) can use this service in a larger semantic workflow.

2.4 Vocabulary Term Mapping - Lele

This project involves implementing a tool that could provide considerable assistance when attempting to map semantically similar terms between metadata, and thus semantically similar or equivalent content in data sets. This tool will return a ranked list of matching candidate terms from input of two sets of terms and their descriptions.