The Data Conservancy Blueprint for Data Management

DataConservancy

dataconservancy.org

The Data Conservancy Blueprint for Data Management

March 2012
Download the most recent version of this document at
dataconservancy.org/community/blueprint

Authors:
Matt Mayernik, Sayeed Choudhury, Tim DiLauro, Ruth Duerr, Elliot Metsger, Barbara Pralle, Mike Rippin

Publisher:
The Data Conservancy
dataconservancy.org

At the Sheridan Libraries
Johns Hopkins University
8489 North Charles Street
Baltimore, MD 21218

Contents

1 Executive Summary

Digital research data can only be managed and preserved over time through a sustained institutional commitment. Research data curation is a multi-faceted issue, requiring technologies, organizational structures, and human knowledge and skills to come together in complementary ways.

The Data Conservancy (DC) embraces a shared vision: scientific data curation is a means to collect, organize, validate and preserve data so that scientists can find new ways to address the grand research challenges that face society. DC is a community organized around data curation research, technology development and community building. Headquartered at the Sheridan Libraries at Johns Hopkins University, the DC community includes university libraries, national data centers, national research labs and information science research and education programs.

Libraries and data centers, as institutions devoted to providing access and preservation services for information resources, are in a key position to lead research data management and curation efforts within and across institutions and disciplines. Initiating data curation services from scratch, however, can be a daunting task. Infrastructures for research data curation are still in their infancy, as are service models for digital data resources.

The Data Conservancy helps fill the need for an institutional solution to the challenges of data curation across disciplines. This document provides a high-level description of the Data Conservancy Instance, which is an implementation of technical tools and organizational services for data collection, curation, management, storage, preservation, and sharing, in order to provide the means to manage, discover, and integrate data across disciplines..

The technical infrastructure for the DC Instance is the called the Data Conservancy software stack. The DC software stack is a digital curation system specifically developed for research data. While comparable to institutional repository systems and disciplinary data repositories in some aspects, the DC Instance has capabilities beyond what those can provide, including a data-centric architecture, discipline-agnostic data model, and a data integration framework that facilitates cross-cutting queries.

The institution that implements a Data Conservancy Instance will also need to organize and staff a data curation practice, including skilled staff for the installation, deployment, and ongoing upkeep of the Instance technology and services. Launching a DC Instance requires

- Expertise in data management,
- A flexible service development process,

- Policies that support service agreements,
- Marketing and outreach partners, and
- A sustainable financial model.

A DC Instance might be supported by an institutional library or academic computing unit, or a partnership between multiple organizational units. The DC Instance can fit into existing infrastructures, because it leverages established standards and tools, including the Open Archival Information System (OAIS) reference model and the Fedora Commons Repository Software, and provides Application Programming Interfaces (APIs) to allow interoperability with other technical tools.

The Data Conservancy website, dataconservancy.org, provides more information about the DC community, software, and educational resources. Please visit our Web site for more information or to contact us.
.

2 Introduction

Data management and curation feature prominently in the landscape of twenty-first century research. Digital technologies have become ubiquitous for the collection, analysis, and storage of research data in all disciplines. Digital research data, if curated and made broadly available, promise to enable researchers to ask new kinds of questions and use new kinds of analytical methods in the study of critical scientific and societal issues. Universities, research organizations, and federal funding agencies are all promoting the re-use potential of digital research data. Information institutions, such as libraries and data centers, are in the position to lead data management and curation efforts by developing tools and services that enable researchers to manage, preserve, find, access, and use data within and across institutions and disciplines.

The Data Conservancy, a growing community of research organizations, is devoted to developing institutional solutions to meet the challenges of curating data across disciplines. The Data Conservancy views data curation as a means to collect, organize, validate and preserve research data in order to enable researchers to find new ways to address the grand research challenges that face society. This document provides an overview of the Data Conservancy, and outlines the Data Conservancy Instance, a technical and organizational data curation solution for research institutions.

2.1 About this document

This document outlines the important requirements, functionalities, and features of a DC Instance. Section 2 provides an overview of the objectives and benefits of a Data Conservancy Instance. Section 3 details the software stack on which a DC Instance is based, as well as the kinds of hardware that are supported. Section 4 focuses on the human staffing and organizational structures necessary to install, run, and maintain a DC Instance. Section 5 discusses sustainability considerations for a DC Instance, outlining technological, financial, and human perspectives. The final section outlines the Johns Hopkins University Data Management Service as an example of how a DC Instance can play a primary role in enabling a university library to offer data curation services.

2.2 What is the Data Conservancy?

The Data Conservancy is a community organized around data curation research, technology development, and community building. Headquartered at the Sheridan Libraries, Johns Hopkins University, the Data Conservancy community includes university libraries, national data centers, national research labs, and information science research and education programs. Initially funded by the National Science Foundation's DataNet program, the Data Conservancy originally focused on four main activities:

1. **A focused research program** that examined research practices both broadly across multiple disciplines, as well as deeply in selected disciplines, in order to understand the data curation tools and services needed to support the interdisciplinary research community;
2. **An infrastructure development program** that developed a technical, community infrastructure ("cyberinfrastructure") on which these data management and curation services could be layered; and
3. **Data curation educational and professional development** programs aimed at developing a deeper understanding of data management within the research community as well as workforce development within the library community.
4. Development of **sustainability** models for long term data curation.

This work, which is ongoing, led to the development of the concept of a Data Conservancy Instance.

The Data Conservancy Community is driven by a common theme: recognizing the need for Institutional solutions to digital research data collection, curation and preservation challenges. Data Conservancy tools and services inherently incentivize scientists and researchers to participate in these institutional data curation efforts by adding value to existing data and allowing the full potential of data integration and discovery to be realized.

2.3 The Data Conservancy Instance

Data curation solutions for research institutions must address both technical and organizational challenges.

The Data Conservancy Instance (DC Instance) is an implementation of both technical tools and organizational services for data collection, curation, management, storage, preservation, and sharing.

To this end, a particular DC Instance is shaped by these considerations:

1. **Context**: DC Instances need to be developed for particular "contexts" to be addressed, such as a particular science domain or institutional domain.
2. **Software Infrastructure**: Each DC Instance has a technical infrastructure that utilizes the Data Conservancy software stack.
3. **Customized Services**: Within local institutional settings, the DC Instances provide particular sets of services, as defined by local needs, which leverage that software stack.
4. **Hardware Infrastructure**: Across DC Instances, there will be common software, but potentially different hardware.
5. **Organizational Infrastructure**: The principal organizational services of DC Instances are local policy frameworks for the system operation, and personnel to set up, manage, and support the continued operation of the Instance. DC Instances will require customized staffing configurations.
6. **Sustainability Strategy:** A sustainability/business plan must also be in place to ensure that the DC Instance (and the data held within it) can serve as a long-term data curation solution.

2.4 DC Instance service offerings and value

The Data Conservancy Instance has an extensive set of features; some are common to digital library systems like ingest, storage, and search/browse, and some are unique to the Data Conservancy Instance, like the feature extraction framework. Core features and values of a DC Instance include:

- **Preservation-ready system** – Facilitating preservation is a core element of the Data Conservancy Instance. Digital preservation requires a multi-faceted approach, involving technical approaches to capturing data, metadata, and provenance information, as well as preservation-specific organizational policies and practices.
- **Customizable user interfaces** – The DC Instance can be customized to meet the needs of specific deployment contexts.
- **Flexible data model** – The data model at the core of the DC software was developed to be conducive to managing and preserving diverse types of data resources.
- **Ingest and search-and-access Application Programming Interfaces (APIs)** – The DC Instance enables additional and external services to be built on top of the core software components via the Ingest and search-and-access APIs.
- **Feature Extraction Framework** - The DC Instance allows data from multiple projects to be brought together via the Feature Extraction Framework, a process through which ingested data resources are examined for particular characteristics, such as the presence of geo-spatial or temporal data structures. The particular kinds of features to be extracted are set up individually for each Instance by the Instance operators.
- **Scalable storage solution** – The underlying archival service of the DC Instance enables millions of digital objects to be ingested and managed.

2.5 Why set up a DC Instance?

The Data Conservancy Instance provides data curation infrastructure for data management, discovery, and integration across disciplines. While comparable to institutional repository systems and disciplinary data repositories in some aspects, the DC Instance has capabilities beyond what either institutional repositories or disciplinary data repositories provide:

2.5.1 Advantages over institutional repositories

The Data Conservancy Instance contains a number of unique features that set it apart from institutional repository systems:

1. **The DC Instance places importance on data over documents**, that is, the DC software system has been designed specifically as an archival repository and access system for data. As such, the DC Instance can provide functionalities

that institutional repositories cannot. The DC Instance Feature Extraction Framework allows data from multiple projects to be brought together through key integrators such as spatial, temporal and taxonomic queries. Additional data services, such as web-map services, sub-setting, and other features can be added to the system as needed.

2. **The DC Instance enables cross-organization and cross-discipline interoperability**. Data within a DC instance can be broadly advertised using a variety of discipline specific and industry standard mechanisms, and can appear in a variety of external catalogs and registries, greatly expanding the user base for data stored within the instance. External organizations can interact with the data within the DC Instance via the Application Programming Interfaces (APIs, discussed further in Section 3). In demonstrations of this interoperability, the Johns Hopkins University DC Instance provides data archiving and access services for the National Snow and Ice Data Center (NSIDC, Boulder, CO) glacier photo service, and data deposit, access, and archiving services for the arXiv.org pre-print repository.

With these features – data integration and external interoperability – the DC Instance is a tool that can lead to collaboration, by enabling researchers to find somebody else's data products and assess the applicability of those data to their own research.

2.5.2 Advantages over disciplinary repositories

The Data Conservancy Instance also contains features that set it apart from disciplinary repository systems:

1. **Discipline-specific Silos:** Discipline-specific data repositories exist for many different research specializations. These disciplinary repositories address the data curation requirements of particular data communities, but in doing so create the data "silo" problem. Each disciplinary repository is an independent silo of data with little ability to connect to other repositories. Determining data relations across discipline-specific repositories is a difficult task. Bringing data together from multiple repositories requires knowing that multiple potentially related repositories exist, searching each repository individually, and compiling data sets manually.
2. **Discipline-agnostic DCI**: The Data Conservancy Instance, in contrast, provides a discipline-agnostic infrastructure designed to meet the needs of diverse data communities. University research libraries, for example, must build data curation services for a broad disciplinary spectrum. The Data Conservancy software stack was designed to serve such multi-disciplinary settings. In addition, external organizations or systems can connect to a DC Instance via the built-in APIs in order to provide discipline-specific services as value-adding features if desired.

3 Data Conservancy Technical Framework

The technical infrastructure of a Data Conservancy Instance consists of the Data Conservancy software stack and appropriate web server and data storage hardware.

3.1 Software infrastructure

The Data Conservancy software stack is the core technical component of a DC Instance. The DC software provides a set of Application Programming Interfaces (APIs) which may be invoked by clients (a client being a human user or another program) of the DC Instance. The purpose of the APIs is to insulate the client from the complexities of the internal system; this allows the system to evolve without requiring the client to change their usage or invocation patterns of the Instance. However, Instance operators will necessarily be exposed to internal complexities of the software. This manifests itself in the selection, configuration, and installation of the components that make up the DC software stack.

The development of the DC software stack has used the Open Archival Information System (OAIS) reference model as a guide (CCSDC, 2002). The OAIS model gives definitions, semantics, considerations, responsibilities, requirements, functional entities, and important actors for an open archival system. The DC Instances largely conform to the OAIS model.

3.1.1 Software architecture

At present, the DC software stack must be installed as a full stack. There are four layers to the DC software architecture. Each layer can communicate with the layers above or below it, but, according to the design of the stack, communications cannot skip layers. Certain layers require external components to be installed, and are discussed below.

- **1st layer – Application layer** – The application layer consists of DC-owned and created applications that access specific services through the APIs. All DC system functions communicate with the DC software services through the APIs. There is no “back door” through which any component can send or receive data to/from the services. Examples of the application layer services include the user interfaces and the batch loading application. External entities, such as organizations and services that use the DC from outside of the stack, are considered to be applications, as they can invoke ingest and access layers via the API layer.
- **2nd layer – API layer** – The API layer provides the specifications for how ingest and search and access services are accessed and invoked. The APIs are invoked via HTTP requests, such as GET, POST, etc.

- **3rd layer – Services** – The services layer consists of services that are invoked as needed by the applications via the APIs. These services include ingest, indexing, and search and access. Users, whether DC Instance staff or researchers depositing and/or accessing data, will only interact with the services through the APIs. The services in this layer are designed to be modular. They potentially could be extracted and applied in another context. The DC services are distributed as a Java web application.
- **4th layer – Archiving** – The archival storage API is the interface to the archival services, and is used to deposit data into the archive and to bring data from the archive to the users. The recommended archival layer implementation uses the Fedora Commons Repository Software as the archival storage framework (http://fedora-commons.org). Fedora maintains an "audit trail" for files that are ingested into DC archival system, which tracks the provenance of individual items. This Fedora audit trail cannot be invoked via the DC software directly, but can be accessed via the Fedora implementation itself.

3.1.2 Software infrastructure components

The DC software components were developed within a modular framework. Each component provides a specific service.

- **Web User Interface (UI) -** The DC software provides a web-based UI. The UI is distributed as a Java web application which may be installed on the same server as the core DC software stack. It supports custom configuration to support branding by each Instance.

- **Index Services -** The DC index services requires an installation of Solr, the Apache web-based indexing server (http://lucene.apache.org/solr/). Due to the intensive nature of indexing, it is generally recommended that Solr be installed on a server separate from other DC software components.

- **Archive Services -** As indicated above, the recommended archive service implementation is based on the Fedora repository software. A Fedora-based implementation is considered to be production ready and is recommended for production installations of a DC Instance. Because Fedora is a separate software project, with its own requirements, installation components, and configurations, it is recommended that it be installed on a server separate from other DC software components.

- **Identity Services -** Identity services require a database to manage identifiers. The Data Conservancy development team recommends PostgreSQL, but any compliant SQL database can be used. The database may reside on a local Data Conservancy server, or it may reside on an institutional server.

3.1.3 Deployment configurations

As indicated in the previous section, when installing and running a DC Instance, some configuration is necessary. The Instance will need to:

- Set up three virtual machines,
- Download and install Solr and Fedora, install the DC software stack, and
- Configure the DC stack on a service machine.

Configuring the DC Instance requires a few steps. There is a single configuration file for DC services. The Fedora and Solr configurations, however, are independent from the DC configuration. They must be individually configured as part of the installation process.

A DC Instance may reside on a single server with proper component installation and configuration. For example, the DC Instance can be downloaded and installed on one machine to test the services. Solr and Fedora still need to be installed, but the Instance can then be run on a Java virtual machine. Practically, however, this single-machine approach is not recommended. The potential interactions between software components and resource constraints will ultimately reveal the shortcomings of a single-server installation.

The Data Conservancy development team recommends that three servers (virtual or physical) be used.

- **Services**: The first server hosts the Data Conservancy services, providing the execution environment for the DC web applications. This service server hosts all of the APIs and UIs, and receives the HTTP requests for data inputs or outputs.
- **Indexing**: The second server is the indexing server, and runs Solr.
- **Storage**: The third server is the Archival Storage server, and runs an instance of Fedora.

These servers should be running a Unix-like operation system (e.g. Solaris, Linux), and all require a Java runtime environment. The DC Services and Archival Storage servers must share a common storage pool. Commonly, the Network File System (NFS) will be used to share the storage, but alternate technologies may be employed, as long as each server references the common storage pool using the same mount path.

As an example of how data flows within this three-machine deployment configuration, the data ingest process makes use of all three servers. After receiving the HTTP ingest requests, the service machine sends the data to the archive machine, while also pushing that subset of the data that need to be indexed to the index server.

3.1.4 Interaction through public Web service APIs

There are two principal APIs providing the interface to the underlying services: Ingest and Search.

Data input and output requests submitted by users are fulfilled by the appropriate API transferring data to and from the archival services. Because the APIs work via HTTP requests, they also allow external services to connect to the DC Instance. Thus, both internal and external services interact with the DC ingest, search, and archival services in the same manner. This section provides a high-level overview of the APIs, please see the Data Conservancy web site for more detailed API documentation.

Ingest service and API

The Ingest service is focused on the acceptance, staging, acknowledgement, and notification associated with data submitted to the DC Instance.

Following the OAIS reference model, the DC Instance ingests data and associated metadata files as Submission Information Packages (SIPs). SIPs are collections of files that contain the content to archive and the properties of that content. SIPs also contain relationship information that describes the overall deposit. For example, the minimum required information that data depositors must supply about a SIP is a title for the deposit.

The Ingest API ingests data materials only if they are packaged as SIPs. When users submit data and associated files to a DC Instance via the UI, the DC UI software constructs a SIP on the back end. The user does not need to construct a SIP manually, but will be asked to provide metadata about the deposit as a whole. In contrast, when a system administrator is performing a batch upload of many data sets, the system administrator would need to construct the SIPs as part of the batch process for each set of materials being deposited.

SIPs are constructed in the Data Conservancy Package (DCP) XML format. A full discussion of DCP is out of the scope of this document, but DCP is an XML format developed for the DC software system that provides a structured format for expressing data resources in terms of a series of related entities with associated properties. A SIP in DCP form may contain embedded data, or it may reference external data. Once SIPs are deposited to the Ingest API, the API returns an acknowledgement in the form of a URI ingest ticket which may be used for polling the deposit's status. When polled for the status of a deposited SIP via its deposit ticket, the Ingest API returns a document containing information that contains status updates and final assigned identifiers for individual files, among other information.

Search service and API
The search API is implemented as a Java web application. User search queries are submitted via the user interface, and are then translated by the search services into the Solr syntax. Search results are returned in one of two formats:

- Data Conservancy Package (DCP) (application/xml) and
- JSON (application/json).

The output format is agreed on through HTTP content negotiation, so the desired output format can be specified by via the HTTP request. Results are returned as a list of resources from which the user can access individual data sets and collections.

External service integration
External entities can use the APIs to interact with the Data Conservancy Instance services. Initial examples that exhibited external use of DC Instance APIs included:

- An implementation that enables researchers to deposit data associated with articles in the arXiv pre-print repository, http://arxiv.org,
- A project that enables the National Snow and Ice Data Center (NSIDC, Boulder, CO) to access and deliver images of glaciers that are archived in the Johns Hopkins University DC Instance
- A pilot project integration integrate DC with the Sakai course management system.

For the arXiv pilot, authors submit a paper for publication, along with their data, to arXiv.org. Upon receipt, the article remains with the arXiv system, while the data are deposited to the JHU Data Conservancy Instance. A bi-directional link is established between the paper in arXiv and the data in Data Conservancy. The arXiv pilot uses the search, access, and ingest APIs.

For the NSIDC pilot, Data Conservancy deposited and curated a large number of glacier images. The NSIDC delivers these images via their Glacier Photo Collection, periodically harvesting the images from Data Conservancy, adding new images to their interface, or updating their system if any changes are made to the photos. The NSIDC uses the search and access APIs, which allow “read-only” functions.

3.2 Hardware infrastructure

The Data Conservancy software stack has been designed to be hardware agnostic. Hardware swapping has been demonstrated at JHU during the development of the software. The DC community is currently developing primarily Linux-based Instances, but the DC software stack can be installed on any hardware and operating system combination that supports a Java Runtime Environment. Fedora, the recommended digital archive underlying the DC Instance is also a Java-based system that is

hardware-agnostic. The Fedora specification sheet indicates that Fedora has been "tested under Linux, Solaris, Windows, and Mac OS X" (Duraspace, 2012).
The hardware requirements for a DC Instance will depend on the scope of the data curation services being implemented. For example, the data volumes collected from project-to-project in all disciplines is highly variable. Correspondingly, the amount of hard disk drive (HDD) and tape storage space needed for any particular instance will vary widely depending on the data to be managed and curated.

The Data Conservancy software has no explicit hardware requirements. DC Instances can be installed with varying amounts of RAM space, storage space, and processor speeds. Hardware requirements should be assessed prior to installing an Instance. Hardware needs can then be assessed on an ongoing basis as the Instance's performance is evaluated and any bottlenecks are identified.

As an example of a possible hardware configuration, the hardware being used for the current JHU DC Instance are shown below:

- Service machine – 2 GB RAM, the staging machine currently has 30 GB HDD to use in the ingest pipeline, using 1.7 TB mass storage for staging. Dual CPUs @ 2.40 GHz.
- Index machine – 8 GB RAM, single CPU @ 2.40 GHz, 30 GB HDD on local disk.
- Archive machine – 8 GB RAM, 30 GB HDD, 1.7TB mass storage. Dual CPUs @ 2.40 GHz.

3.3 System scalability

The Data Conservancy software system has not been subjected to structured scalability tests itself, but the scalability of the system can be evaluated based on the scalability of the underlying storage and indexing components: Fedora and Solr.

Storage - Fedora
According to the Fedora Specification Sheet, "Fedora is designed & tested to store 10+ million digital objects per instance" (Duraspace, 2012). As noted in the discussion of hardware requirements in Section 4, storage needs will vary from Instance to Instance, but Fedora's 10+ million digital object metric provides a good indication of how the DC Instance provides a scalable storage system for very large digital archiving implementations.

Indexing and query performance - Apache Solr
When looking at indexing and search query performance, the Solr search server also provides a very robust and scalable solution. The Apache Solr *Frequently Asked Questions* web page indicates that Solr users report typical indexing rates between 10 and 150 documents per second (reports provided at http://wiki.apache.org/solr/FAQ#How_fast_is_indexing.3F).

From a search and query perspective, large-scale Solr users have reported that Solr query response times are typically less than 0.2 seconds, even for systems that have indexed hundreds of thousands of documents and receive hundreds of thousands of queries per day (see examples of Solr performance reports at http://wiki.apache.org/solr/SolrPerformanceData).

4 Organizational Framework

A Data Conservancy Instance is designed to be set within an organizational structure, with skilled staff involved in the installation, deployment, and ongoing upkeep of the Instance technology and services. This section outlines staffing and skills, as well as organizational structures, that are necessary to the success of a DC Instance.

4.1 Staffing and skills

At the center of a Data Conservancy Instance are skilled staff. Staffing needs for a DC Instance may change over time depending on the kinds of service that are developed around the system, but a typical DC Instance will require particular types of staff and skill sets.

- **System Administrator***:* Installing a DC Instance, as described above in sections 3 and 4, requires configuring hardware and software environments. A DC Instance will need an administrator who is able to configure each of the software components (Fedora, Solr, and the DC software stack) in an appropriate hardware installation.
- **Software Developer:** A software developer is also necessary to perform any user interface enhancements that may be desired for a particular Instance, and to set up programmatic workflows for batch ingestion of large numbers of data files. In addition, because the DC software is open source, software developers can actually create new services or customize core DC services as required or desired by their individual Instances. New services that might be useful to other DC Instances should be contributed back to the broader DC community.
- **Instance Administrator***:* A DC Instance also requires an administrator who oversees the day-to-day activity within the system. The administrator grants and maintains user accounts, establishes and enforces any applicable quotas on user account size, and works with users on any customer service-related issues that occur as users deposit and access data from the Instance.
- **Data Management Consultants**: Depositing data into formal data archives is still a novel activity for many researchers, both student and faculty. As such, it is recommended that a DC Instance include staff who work as data management consultants. These consultants work with researchers to create data management plans and implement data management and archiving processes, including depositing data into the DC system and creating metadata.
- **Services Manager**: At a higher administrative level, a DC Instance requires a manager who ensures that the Instance components (technology, people, and services) all work together cohesively.

Depending on the situation in which a DC Instance is deployed, these roles might be consolidated into a single person, in particular the instance administrator and data management consultant roles, and the system administrator and software developer

roles. Many of the skills that these roles require are already part of library and computing departments. For example, the data management consultation process emulates a reference interview. Data management consultants need to gather information about a researcher's data management needs, identify gaps in current plans and practices, help the researchers understand their data management options, and then help researchers to prepare and iterate on their data management plan.

As with any reference work, consultants should adapt their recommendations and help to researcher timeframe and deadlines. In this way, the DC Instances will support better data management by encouraging systematic change of academic data management cultures through direct interactions with researchers.

4.2 Organizational structure

The Data Conservancy Instance has been developed as a research support service. Research libraries are the main type of institution that have expressed interest in Data Conservancy services, but an Instance is not restricted to being a library service. There might be another group within a research institution who supports an Instance, such as an academic computing group, or two or more units may work together within a single institution to support an Instance.

Each DC Instance needs to define its own collection policy. Collection policies will vary depending on the institutional context in which an Instance is situated, but should include considerations about issues such as the communities to be served by the Instance, the scope of the collections to be included, the data types that will be supported, the criteria for inclusion, how and when to reappraise data collections, and the levels of service that will be provided. In addition, an Instance should define data retention policies that outline levels of support and commitment to maintaining data over time. A data retention policy does not require that the Instance commit to open-ended support for data; a policy can specify that data retention procedures revolve around defined projects and time-period.

Collection policies and data retention policies will help to guide the development of agreements between the data archive and researchers who are depositing data. The DC Instance does not have a technical mechanism for encoding these agreements in machine actionable form, or enforcing collection policy. Instead, the collection policies and agreements should be used to educate the depositor and/or consumers as to the operation of the Instance, and the management and curation of the data resources held by the Instance.

5 Sustainability Strategies

Data can only be managed and preserved over time through a sustained institutional commitment. Ensuring the sustainability of data curation efforts is a multi-faceted issue, requiring sustainable technologies, sustainable financial structures, and sustainable ways of ensuring the continuity of human knowledge and skills.

5.1 Technical sustainability

Technical sustainability has been an important consideration throughout the DC software design and development process. Data Conservancy technical sustainability arises from:

- **The modular framework** and technology components outlined in Section 3. The DC software features a modular service-oriented architecture with interfaces and APIs that loosely couple layers and services. This approach facilitates seamless migration away from specific technologies that become deprecated and provides mechanisms for interoperability with other technical infrastructure.
- **The adoption of open-source** licensed technology ensures that DC software remains legally unencumbered and provides an open environment for technical collaborations moving forward.
- **Commitment to support future development**, both within JHU and within the wider community will ensure that the Data Conservancy not only remains available, but will also continue to support its users in that ways that they require.

5.1.1 Modular framework

DC demonstrated this modularity and flexibility through initial applications of its Archival Storage API. DC's original plan was to create a reference implementation based on the Fedora repository software, but the Fedora mapping from the API and DC data model could not be completed immediately. In the interim, DC implemented an approach based on a simpler file system. Once the Fedora modeling work was completed, DC replaced the file system implementation without changing any code that used the Archival Storage API. DC will take a similar approach for its more general Persistent Storage API, and is working with the cloud-based DuraCloud storage service and the University of Indiana Pervasive Computing Institute with regards to possible storage arrangements beyond the local storage recommended currently.

5.1.2 Open source licensing

The Data Conservancy software will be released under the Apache Software Foundation's Apache License, Version 2.0. This free software license allows users to use the software for any purpose, as well as to distribute, modify, and sub-license the

software. (See http://www.apache.org/licenses/LICENSE-2.0.html for the full details of the license.)

5.1.3 Development commitment

Revisions to the DC software will be managed on an ongoing basis by the DC development team. Development needs are assessed and prioritized in consort with the DC Instance product owners group. Each Instance has a designated "product owner" to serve as the prime contact and liaison to the broader DC community. The product owners group evaluates the current system and identifies development needs and functionalities to prioritize. The on-the-ground experience and varied disciplinary foci of the product owners enables the development team to be responsive to user needs while keeping the broader user base in mind. Product owners should also feed new or customized DC services, such as new search interfaces, feature extraction profiles, or other API-based services, back to the fuller DC community. As the DC community grows, new Instance product owners will be invited to join in the iterative feedback process. If, in the future, the size of the product owners group makes coordination among all parties difficult, the organization of the group will be re-assessed.

5.2 Financial sustainability

In order for data curation services to ensure that data resources are accessible and usable in the future, they must be backed-up by a sustainable financial model. Sustainable cost models for data curation services are not yet well understood, as different data curation institutions have different financial models. Financial sustainability is, however, obviously interconnected with technical and human sustainability issues.

Successful implementation and operation of a DC Instance will require a thorough analysis of all known or expected costs for the next several years, coupled with strategies for continuing to cover those costs in sustainable ways.

5.2.1 Cost categories

The three main costs of running a DC Instance, or other data curation solution, are:

- **Hardware.** Hardware costs include the costs of purchasing and running servers, storage media (disk and tape), and the maintenance and servicing costs that accumulate over time. Hardware costs might also include additional costs/equipment necessary to create off-site back-up copies of data collections.
- **Staffing**. Staffing costs include support for data management consultants, a software developer, a systems administrator, and the services manager. As noted in Section 7, one person might take on the responsibilities of more than one of these positions. Depending on the goals and needs of a particular

setting, an institution installing a DC Instance might also want or need a senior technical consultant, who would lead technical initiatives to extend or modify the Instance software, and/or a budget specialist, who would assist in financial planning and coordination. In addition, providing professional development opportunities for staff can be very beneficial to the individual staff members, and can help to improve and extend the services that an institution can provide, but must be accounted for in cost models. This is also true for routine training for new and current staff members.

- **Administrative costs**. As with any institutional information service, administrative and operational costs also need to be taken into consideration. These costs include computer equipment, furniture, supplies, phones, and physical space charges.

5.2.2 Financial models and strategies

Different institutions use different financial models to support data curation services. These models range from direct funding from national governments to fee-for-service models, with many institutions utilizing multiple sources of funds. The effectiveness and sustainability of any financial model must be continually evaluated and pro-actively managed so that adjustments can be made. Funding models include:

- **Government funding** – National governments fund data centers and repositories directly. These repositories often have a mandate to serve specific government agencies, as in the case of the NASA Earth Observing System Data and Information System (EOSDIS) data centers, but they can also be general purpose systems developed for national research communities, as is the case for the Australian National Data Service (ANDS, http://www.ands.org.au/). The ANDS is funded by the Australian government.
- **Institutional funding** – Many university libraries are developing data curation services through institutional funding means. Library-developed services might be funded through core library budgets, or via partnership with other campus units.
- **Community memberships** – Some repositories fund their activities through offering memberships. For example, the Interuniversity Consortium for Political and Social Research (ICPSR), a domain-focused repository for quantitative social science data, uses a membership model, in which other institutions (typically universities) pay membership dues to gain access to ICPSR services and resources (http://www.icpsr.umich.edu/icpsrweb/ICPSR/). ICPSR also receives grants from government agencies and private foundations to curate particular collections.
- **Fee for service** – Repositories can charge fees for data management and curation services. Fees might be specific to particular services, such as metadata creation, digitization, formatting or reformatting data files, or data presentation customization.

- **Grant funding** - Financial support for data management services can come from asking Principal Investigators to contribute funds from research grants. The National Science Foundation allows researchers to request money in grants for preparing research materials to be shared: “Costs of documenting, preparing, publishing, disseminating and sharing research findings and supporting material are allowable charges against the grant” (NSF 2012b, Section B.7.: Publication, Documentation and Dissemination).

5.2.3 Formalized agreements

When working with Principal Investigators to develop grant-based financial agreements, it is very important to draw these agreements as explicit documents. These agreements might be called Memoranda of Understanding (MOU) or, in the case of ICPSR, “deposit agreements”. These agreements should explicitly specify itemized costs, payment plans, and the kinds of support to be provided over a particular time period, such as a five-year grant period. Producer-archive agreements should also consider plans for addressing unexpected problems.

5.3 Human sustainability

Human sustainability is critical to ensuring continuity and consistency of data curation services over time. Staff develop knowledge and day-to-day practices for working with researchers, creating and implementing data management plans, and working effectively with technical data systems. Because data collections are so variable, the most effective data management and curation environments are those that allow for cross-pollination of expertise, practices, and skills among staff members.

5.4 Community building

Community building is an important part of the Data Conservancy mission. The DC Instance operators at the initial DC partners have been working together to sketch out requirements and system features for ongoing development by the Data Conservancy development team. As more Instances are deployed, this community will continue to grow, along with other sub-communities dedicated to technical or organizational issues surrounding the Instances.

6 Case Study: Johns Hopkins University Data Management Services

The Johns Hopkins University Data Management Services (JHU DMS) are a good example of how the DC Instance can be deployed. DMS has a full DC Instance running on a local server. The DMS went live in July of 2011, and the DC Instance itself went live in October of 2011. The JHU DMS website can be found at http://dmp.data.jhu.edu/.

6.1 DMS services offered and financial model

JHU DMS is supported by Deans of schools within JHU that receive NSF grants. The JHU DMS was proposed to the Deans of these schools as a research support service that filled a need newly opened up by the NSF data management planning requirement (NSF, 2012a). The proposal to the deans included a rationale for the DMS, the scope of the services the DMS would provide, and a budget for those services. As proposed (and now implemented), the DMS provides two services: 1) consultative services for researchers who are writing a data management plan for an NSF proposal, and 2) post-award services for researchers who receive a NSF award. Post-award services include working with researchers to detail data management procedures as the grant proceeds, and later helping researchers to actually deposit their data into the DMS DC Instance.

The pre- and post-award DMS services are financially distinct. The pre-award data management planning consultations are supported directly by the JHU deans, while post-award DMS services are written into NSF proposal budgets by the researchers who wish to work with DMS after the grant is received. Thus, post-award DMS fees are charged to the individual grants that are being supported. As these two distinct financial models suggest, researchers can work with DMS consultants pre-award (to create a data management plan for a proposal) without working with DMS consultants post-award (to actually archive their data in the DMS DC Instance).

6.2 DMS staffing

When proposing the DMS to the JHU Deans, the argument made was that the DC software and the existing hardware base within the JHU libraries provide the technical capabilities to manage and curate data at most scales that a research university will need. The missing component in developing data management services was a staff of consultants who could work with faculty to develop data management plans and deposit data. These data management consultant positions were part of the DMS proposal. Two data management consultant positions were filled after the deans endorsed the DMS and agreed to provide financial support for the pre-award data management planning services. The data management consultant position descriptions recognized the diversity of experiences and talents that can support such

a service, and were written to recruit people with skills that matched the organization's data curation objectives and services. The goal for the JHU DMS is to develop a broad set of domain expertise. Domain specificity was not part of the job search, but domain expertise was. The two initial data management consultants hired by DMS have graduate degrees in both library/information science and a particular science or social science domain. Depending on the scope of the services developed within any particular DC Instance deployment, such domain expertise may be more or less important. Within the JHU, this domain expertise has proven to be very valuable in rolling out the DMS services.

The DMS data management consultants also serve as the instance administrators. In this capacity, they help to administrate user accounts and provide consultation on the use of the Instance. Other DMS staff include a system administrator/software developer, and a services manager who oversees the overall operation of the DMS and serves as the JHU DMS representative in the DC Instance product owners group.

Sharing of expertise has played a central role within the JHU DMS. The DMS team has the advantage of being closely coordinated with the DC technology development team. One of the principal architects of the DC software stack is working half-time within the DMS team during the roll-out period, in effect serving as a "knowledge bridge" between the DC software development team and the DMS staff. In the long-term, data management consultants will develop their own expertise working with the DC Instance, and will be able to provide training when new people are hired, and/or the services expand or are offered to new constituencies.

6.3 Operational activities

When working with researchers, the JHU DMS consultants first have introductory meetings to discuss the proposed research project, its data management needs, and the DMS services that are most appropriate to the project. These meetings are followed by the consultants working with the researchers to write detailed data management plans that meet the NSF requirement and the recommendations of the individual NSF Directorates. If researchers wish to continue working with DMS consultants post-award, the DMS consultants work with the researchers to develop an agreement document that maps the DMS services onto the expected data and metadata products to be input to the data archive. This agreement is principally informed by the process of discussing the project and writing a data management plan with the researchers.

After a research award is received, data management consultants work with the researchers to do in-depth planning of data management activities. They also help the researchers to understand service levels within a DC Instance. As a research project proceeds, the consultants plan and prepare data for deposit, review data management plans, and identify solutions for the next phase of data management.

6.4 Initial challenges

Each DC Instance will encounter challenges unique their own environment, but the initial months of the DMS indicate some challenges that other Instances may face in the early stages of a DC Instance deployment. Because the DMS is a cross-disciplinary service, responding to widely ranging domains requires flexibility and awareness. One example is the difficulty of effectively timing consultative support in order to meet grant submission deadlines. Grant-writing and other deadlines vary widely from project to project, within and across disciplines. Data management consultants must be aware of the deadlines and prioritize work accordingly. Another cross-disciplinary challenge is the lack of common vocabulary for data management activities. One PI's "storage" is another PI's "archiving."

Navigating different data retention policies is another challenge, as different kinds of data have different data retention needs, and funding bodies vary in their retention policies. Working within the boundaries of the NSF data management planning policy is in-and-of-itself a challenge. Condensing key data management planning information into two pages regardless of the size of the project or expected data complexity is a notable constraint. Finally, marketing the DMS is necessary but difficult. Building a usage base requires building awareness of the value of data management and curation, and of the DMS itself.

7 References

Consultative Committee for Space Data Systems (CCSDC). (2002). *Reference Model for an Open Archival Information System (OAIS).* Recommendation for space data system standards, CCSDS 650.0-B-1. http://public.ccsds.org/publications/archive/650x0b1.PDF

Duraspace. (2012). *Fedora Repository Commons Software: Specsheet.* http://www.duraspace.org/fedora/repository/duraspace:23/OBJ/specsheet_Fedora.pdf

National Science Foundation (NSF). (2012a). *Chapter II - proposal preparation instructions: special information and supplementary documentation.* http://www.nsf.gov/pubs/policydocs/pappguide/nsf11001/gpg_2.jsp#IIC2j

National Science Foundation (NSF). (2012b). *Award and Administration Guide, Chapter V - Allowability of Costs.* http://www.nsf.gov/pubs/policydocs/pappguide/nsf11001/aag_5.jsp#VB7