The Data Conservancy Instance: Infrastructure and Organizational Services for Research Data Curation

This is a pre-print of an article published in 2012 in D-Lib Magazine, Volume 18, Number 9/10. http://dx.doi.org/10.1045/september2012-mayernik

Please do not quote from this version, as it differs slightly from the published version.

Authors:
Matthew S. Mayernik, G. Sayeed Choudhury, Tim DiLauro, Elliot Metsger, Barbara Pralle, Mike Rippin, Ruth Duerr

1 Abstract
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. This article provides a high-level description of the Data Conservancy Instance, an implementation of infrastructure and organizational services for data collection, storage, preservation, archiving, curation, and sharing. While comparable to institutional repository systems and disciplinary data repositories in some aspects, the DC Instance is distinguished by featuring a data-centric architecture, discipline-agnostic data model, and a data feature extraction framework that facilitates data integration and cross-disciplinary queries. The Data Conservancy Instance is intended to support, and be supported by, a skilled data curation staff, and to facilitate technical, financial, and human sustainability of organizational data curation services. The Johns Hopkins University Data Management Services (JHU DMS) are described as an example of how the Data Conservancy Instance can be deployed.

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 (Long-Lived Digital Data Collections, 2005; Changing the Conduct of Science in the Information Age, 2011).

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 (Walters & Skinner, 2011; UNC-CH, 2012).

This article provides an overview of Data Conservancy, and outlines the Data Conservancy Instance, an infrastructural and organizational data curation solution for research institutions. Data Conservancy is a community promoting data preservation and re-use across disciplines with tools and services.

2.1 What is the Data Conservancy?

The Data Conservancy is a community organized around data curation research, technology development, and community building (Treloar, Choudhury, & Michener, 2012). Initially funded by the National Science Foundation’s DataNet program, the Data Conservancy is headquartered at the Sheridan Libraries, Johns Hopkins University. Data Conservancy community members include university libraries, national data centers, national research labs, and information science research and education programs. The Data Conservancy community is driven by a common theme: the need for institutional solutions to digital research data collection, curation and preservation challenges. The four main activities of the Data Conservancy are:

1. **A focused research program** that examined research practices broadly across multiple disciplines, as well as deeply in selected disciplines, in order to understand the data curation tools and services needed to support interdisciplinary research (Renear, Sacchi, & Wickett, 2010; Lagoze & Patzke, 2011; Palmer, Weber, & Cragin, 2011; Wynholds, et al., 2011);
2. **An infrastructure development program** that developed a technical, community infrastructure (“cyberinfrastructure”) on which data management and curation services can be layered;
3. **Data curation educational and professional development** programs aimed at developing a deeper understanding of data management training needs within the research community as well as workforce development within the library community (Varvel, et al., 2011).
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.

2.2 The Data Conservancy Instance

Data curation solutions for research institutions must address both technical and organizational challenges. A Data Conservancy Instance is an installation of the Data
Conservancy technical infrastructure embedded within a particular organization. Individual DC Instances are shaped by these considerations:

1. **Context**: Data curation solutions must be developed for particular “contexts,” such as particular science or institutional domains (Agre, 2003).
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.
6. **Sustainability Strategy**: A sustainability/business plan must also be in place to ensure that the data curation solution (and the data held within it) can be maintained over the long-term.

### 2.3 DC Instance service offerings and value

The Data Conservancy Instance has a number of features that are common to digital library systems, such as ingest, storage, and search/browse, and some features that are unique to the Data Conservancy Instance. 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 (Li & Banach, 2011).

- **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 based on the PLANETS data model ([http://www.planets-project.eu/](http://www.planets-project.eu/)), and 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 by each Instance.
• **Scalable storage solution** – The underlying archival service of the DC Instance enables millions of digital objects to be ingested and managed.

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 development of the DC software stack has used the Open Archival Information System (OAIS) reference model as a guide (CCSDS, 2012). The OAIS model gives definitions, semantics, considerations, responsibilities, requirements, functional entities, and important actors for an open archival system.

3.1.1 Software architecture
The DC software architecture contains four layers. 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, as discussed below.

1st layer – Application layer – The application layer consists of DC-owned and created applications that access specific services through the APIs. 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 DC software provides a set of APIs that may be invoked by clients (either human users or other programs). All DC system functions communicate with the DC software services through the APIs. 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. The purpose of the APIs is to insulate the client from the complexities of the internal system, allowing the system to evolve without requiring the client to change the ways that they use or invoke the Instance features.

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. The services in this layer are designed to be modular. Potentially, they could be extracted and applied in another context. The DC services are distributed as a Java web application, while the search functionality uses the Apache Solr software ([http://lucene.apache.org/solr/](http://lucene.apache.org/solr/)).

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 archival layer can be implemented using a range of archival storage frameworks, but the implementation recommended by the DC development team uses the Fedora Commons Repository Software (http://fedora-commons.org).

3.1.2 Interaction through public Web service APIs
There are two principal APIs providing the interface to the underlying services: Ingest and Search-and-Access. Data requests 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-access, and archival services in the same manner. This section only 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. The DC software compiles the SIPs for the user when data are submitted through the DC user interface, or the data submitter can create their own SIPs for batch upload.

Search-and-access service and API
The search-and-access 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. The output format is agreed on through HTTP content negotiation, so the desired output format can be specified by 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. For example, in a pilot illustration of external use of DC Instance APIs, the arXiv pre-print repository (http://arxiv.org) enables researchers to deposit data associated with articles. Upon deposit, 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 the JHU DC Instance. The arXiv pilot uses the search-and-access and ingest APIs.

Another pilot project allowed the National Snow and Ice Data Center (NSIDC, Boulder, CO) to use the JHU DC Instance APIs to discover and include within their own system metadata about a number of glacier images held in a collection about volcanism in the Dry Valleys of Antarctica (http://nsidc.org/cgi-bin/glacier_photos/glacier_photo_search.pl?collection=dryvalley). The NSIDC Glacier Photo Collection periodically harvests the images from the DC Instance and updates their collection 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 is hardware agnostic. The DC community is 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 archival storage framework, is also a hardware-agnostic Java-based system.

The hardware requirements for a DC Instance will depend on the scope of the data curation services being implemented. For example, data volumes vary from project-to-project and disciplines-to-discipline. Correspondingly, the amount of hard disk drive (HDD) and tape storage space needed for any particular instance will vary depending on the data to be managed and curated. As such, 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, and can be assessed on an ongoing basis as the Instance’s performance is evaluated and any bottlenecks are identified.

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.

4.1 Staffing and skills
Skilled staff are central to any data curation service. 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 section 3, 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.
- **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, DC
Instances benefit from having staff who work as data management consultants. These consultants can work with researchers to create data management plans and implement data management and archiving processes, including creating metadata and organizing data for deposit into the DC system.

- **Services Manager**: At an 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 staffing and budget situation in which a DC Instance is deployed, multiple 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 (Garritano & Carlson, 2009). 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. The more direct interactions that data management consultants have with researchers, the more they can encourage systematic change of academic data management cultures.

### 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. Important considerations include 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-periods. Collection policies and data retention policies help to guide data deposition agreements, and the management and curation of the data resources over time.

### 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 (Lavoie, 2012).

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 feature a service-oriented architecture with interfaces and APIs that loosely couple layers and services. This approach facilitates seamless technology migration and provides mechanisms for interoperability with other technical infrastructure.
- **The adoption of open-source.** An alpha version of the DC software has been released under the Apache Software Foundation’s Apache License, Version 2.0 ([http://dataconservancy.org/software/downloads/](http://dataconservancy.org/software/downloads/)). 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](http://www.apache.org/licenses/LICENSE-2.0.html) for the full details of the license.)
- **Commitment to support future development,** both within JHU and within the wider community, will ensure that the DC software remains available and up-to-date. Revisions to the DC software are managed on an ongoing basis by the DC development team.

5.2 Financial sustainability
In order to ensure that data resources are accessible and usable in the future, data curation services must be backed-up by a sustainable financial model. Sustainable cost models for data curation services are not yet well understood; 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 any digital data curation service will require a thorough analysis of all known or expected costs for the immediate future, coupled with strategies for continuing to cover those costs in sustainable ways.

5.2.1 Cost categories
The three main costs of running a 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.** For a DC Instance, staffing costs include support for data management consultants, a software developer, a systems administrator, and the services manager. As noted in Section 4, one person might take on the responsibilities of more than one of these positions. 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.

- **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 organizations use different financial models to support data curation services. These models range from direct funding from national governments to fee-for-service models. Organizations commonly utilize multiple sources of funds (Lavoie, 2012). 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, such as the NASA Earth Observing System Data and Information System (EOSDIS) data centers (http://earthdata.nasa.gov/data/data-centers), and the Australian National Data Service (ANDS, http://www.ands.org.au/).

- **Institutional funding** – Many university libraries are investigating and/or developing data curation services through institutional funding means (Delserone, 2008; UCSD, 2009; UNC-CH, 2012; Witt, 2012). Library-developed services might be funded through core library budgets, or via partnership with other campus units.

- **Community memberships** – Some data 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/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). When working with Principal Investigators to develop grant-based financial agreements, it is very important to create explicit documents, typically called Memoranda of Understanding (MOU) or deposit
agreements, to specify itemized costs, payment plans, and the kinds of support to be provided over a specified time period, as well as 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 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. Sharing of expertise plays a central role within the ongoing operation and development of any data curation solution. Over time, data management consultants will develop their own expertise working with the DC Instance, and will be able to provide training to new users and staff.

Human sustainability feeds into technical sustainability. Each DC 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. 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.

6 Case Study: Johns Hopkins University Data Management Services
The Johns Hopkins University Data Management Services (JHU DMS) are the first example of how the DC Instance can be deployed. DMS has a full DC Instance installed, with the DC software stack running on a local server. The DMS went live in July of 2011. The JHU DMS website can be found at [http://dmp.data.jhu.edu/](http://dmp.data.jhu.edu/).

6.1 DMS services and financial model
JHU DMS is supported by Deans of schools within JHU that receive NSF research grants. The JHU DMS was proposed as a research support service to fill a need newly opened up by the NSF data management planning requirement (NSF, 2012a), but the scope of DMS has since expanded to other funding sponsors. The DMS proposal 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 research proposal, and 2) post-award services for researchers who receive a funding award, including developing data management procedures as a grant proceeds, and 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 all JHU schools actively seeking NSF funding, while post-award DMS services are written into 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 and operational activities

The DMS staff includes two data management consultants, a software developer, and a services manager who oversees the overall operation of the DMS and serves as the DMS representative in the DC Instance product owners group. A system administrator position is in the process of being filled. These positions were all part of the DMS proposal, and the two data management consultant positions were filled after the JHU Deans endorsed the DMS and agreed to provide financial support. Domain specificity was not part of the data management consultant 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 science or social science domain. This domain expertise has proven to be very valuable in rolling out the DMS services. The DMS data management consultants also serve as the DC Instance administrators, administering user accounts and consulting on the use of the Instance.

### 6.3 Initial DMS challenges

All DC Instances will encounter challenges unique to their own environment, but the initial months of the DMS indicate the types of challenges that other Instances may face. Because the DMS is a cross-disciplinary service, responding to widely ranging domains requires flexibility and awareness. For example, effectively timing consultative support in order to meet grant submission deadlines can be challenging. 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” (Choudhury, 2012).

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 Conclusion

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.

Discipline-specific and institutional repositories address the data curation requirements of particular data communities, but in doing so often create the data “silo” problem. Each disciplinary repository is an independent silo of data with little ability to connect to other repositories (Salo, 2010). Determining data relations across 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.

The Data Conservancy Instance contains a number of unique features that set it apart from discipline-specific or 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**. External organizations can interact with the data within the DC Instance via the APIs. 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.

3. **Discipline-agnostic infrastructure**: The Data Conservancy Instance 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.

With these features – data integration, external interoperability, and a discipline-agnostic infrastructure – the DC Instance is a tool that can lead to collaboration, by enabling researchers to find someone else’s data products and assess the applicability of those data to their own research.
8 Acknowledgements
The Data Conservancy is funded by the National Science Foundation under grant number OCI-0830976. Funding for the Data Conservancy and the Johns Hopkins University Data Management Services is provided by the JHU Sheridan Libraries. We acknowledge contributions from our Data Conservancy colleagues and their remarks on earlier versions of this paper.

9 References


https://www.youtube.com/watch?v=F6iYXNvCRO4&feature=plcp

http://public.ccsds.org/publications/archive/650x0m2.pdf

http://hdl.handle.net/2142/10670


http://www.istl.org/09-spring/refereed2.html

http://dx.doi.org/10.1145/1998076.1998145


http://dx.doi.org/10.1080/01930826.2012.655607