Abstract

Improving Fedora to Work with Web-scale Storage and Services

Memory institutions around the world face a rapidly expanding need for storage and access to digital objects and metadata. The Fedora Repository has long been at the forefront of their efforts, developing software to meet the challenge, including four major versions of the Fedora Repository software. Now the Fedora leadership have put forward a bold call to the community to create new implementations of Fedora to meet emerging needs, publishing a formal API that specifies the expectations of a Fedora repository. Through our research into computational archives and through prior Fedora involvements, we have learned that a new core need is scalability, by which we mean the ability to expand storage capacity without losing performance. We believe that institutions must be able to incrementally grow a fully-functional repository as collections grow, without the need for expensive enterprise storage plans, massive data migrations, and performance limits that stem from the vertical storage strategy of previous repository implementations.

The Digital Curation Innovation Center (DCIC) at the University of Maryland's College of Information Studies (Maryland's iSchool) intends to conduct a 2-year project to research, develop, and test software architectures to improve the performance and scalability of the Fedora Repository for the Fedora community. More specifically this project will apply the new Fedora 5 application programming interface (API) to the repository software stack called DRAS-TIC to create a new Fedora implementation we are calling *DRAS-TIC Fedora*. DRAS-TIC, which stands for Digital Repository at Scale that Invites Computation, was developed over the last two years through a collaboration between UK-based storage company, Archive Analytics, and the DCIC, through funding from an NSF DIBBs (Data Infrastructure Building Blocks) grant (NCSA "Brown Dog"). DRAS-TIC leverages NoSQL industry standard distributed database technology, in the form of Apache Cassandra, to provide near limitless scaling of storage without performance degradation. With Cassandra, *DRAS-TIC Fedora* can also hold redundant copies of data in datacenters around the world. Even if an entire datacenter is lost, access can remain uninterrupted, and data re-replicated to a new datacenter. Beyond institutional reliability, we think this creates the possibility for new reciprocal storage arrangements between Fedora institutions.

To meet with this potential, DRAS-TIC will first need to be adapted to the new Fedora API and then engineered and tested to meet the performance expectations of our Fedora community partners. We have identified a range of institutional partners in the Fedora community that will work with us to develop use cases and performance expectations. As we develop and test *DRAS-TIC Fedora*, their institutional needs will guide our efforts and become our measure of success. The proposal has received the endorsement of the Fedora Leadership Group: http://fedorarepository.org/leadership-group.

The proposed project will produce open-source software, tested cluster configurations, documentation, and best-practice guides that will enable institutions to manage Fedora repositories with Petabyte-scale collections and thus, contribute to big-data ready national software infrastructure.

Improving Fedora to Work with Web-scale Storage and Services

The Digital Curation Innovation Center (DCIC) at the University of Maryland's College of Information Studies (iSchool) requests $240,138 to research ways to improve the performance and scalability of the Fedora Repository for the Fedora user community. The work will be carried out over the 2-year period (October 1, 2017 to September 30, 2019) without cost sharing. This includes $157,986 in direct costs and $82,153 in indirect costs, calculated at UMD's negotiated rate. The direct cost is further broken down into $120,118 for salaries and wages, $27,868 for fringe benefits, $10,000 for travel.

1. Statement of National Need

This project addresses high priority challenges that face the Fedora user community. *"Fedora has a worldwide installed user base that includes academic and cultural heritage organizations, universities, research institutions, university libraries, national libraries, and government agencies."*[1] The National Digital Platform is interested in expanding the scalability and interoperability of major open source software used by libraries and archives. Our project aims to address current performance limits identified in Fedora 4.x through research and development of new ways to perform its operations to manage large-scale collections. For decades Fedora has consistently provided an object storage framework for repositories of content and data. Fedora developers and users have created hundreds of applications, a strong community, and a rich software ecosystem of shareable code. The Fedora 4 project, now in its seventh software release, worked to redefine Fedora's conventions, the application programming interface (API) and the data model, in the age of linked data.

Looking beyond Fedora 4, the Fedora community has placed significant effort towards formalizing and specifying the Fedora application programming interface[2], designated as the Fedora 5 API. This was done to clearly define the core services expected of a Fedora repository, to stabilize the technical contract for Fedora clients, and to enable new implementations of Fedora to enter the ecosystem. The Fedora 5 API, there is no implementation, is largely based on two open standards, the W3C Linked Data Platform[3] and IETF Memento[4]. The *DRAS-TIC Fedora* team will work with the Fedora community to develop an implementation of the Fedora 5 API to address a shortcoming in the existing Fedora 4 implementation. More details on DRAS-TIC are provided in Section 2, Project Design. We discussed this project with members of the Fedora Steering Group in the fall of 2016 and held a working meeting at the Dec. 2016 CNI conference to further refine the research and development ideas. The proposal has received the very important endorsement of the Chair of Fedora Leadership Group and Steering Committee (See: http://fedorarepository.org/leadership-group and http://fedorarepository.org/steering-group). We are including this letter of support. Since then we have reached out to key universities and partners on the Fedora Development Team and cultural institutions interested in management at scale of library collections. Initial community members include 5 partners: UMD Libraries,

[1] DURASPACE/Fedora home: https://wiki.duraspace.org/display/FF/Fedora+Repository+Home
[2] Fedora 5 API Specification: http://fedora.info/spec/
[3] Linked Data Platform 1.0 Specification: https://www.w3.org/TR/ldp/
[4] Memento Specification: https://tools.ietf.org/html/rfc7089

Smithsonian Institution (Office of Research Info. Services and National Museum of American History), UIUC National Center for Supercomputing Applications (NCSA), and Georgetown U. Library. All have expressed interest and a need and we expect to reach out to these partners during the course of the project for further feedback, use cases, and guidance.

Pursuing Fedora at Scale

The storage demands of Fedora users for objects and metadata are extreme when compared to typical content management applications. Many Fedora institutions store large data files in their repositories, such as disk images, unlike the typical media objects used in publishing. They may also store millions of objects with extensive metadata attached to each. It is for this reason that the user community has an innovative history of pursuing repository scalability. The Fedora 3.x design included a modular storage layer, which allowed for the development of custom, modern storage solutions by different institutions and coalitions. These included community development of the Akubra storage layer as well as the development of an iRODS-based Fedora storage module at UNC Chapel Hill. Jansen was principal architect of this iRODS storage module and Marciano was a project advisor and iRODS researcher. In addition UCSD at that time addressed their scalability needs by implementing the Fedora 3.x API over a custom storage system of their own design. Fedora institutions like these have been major proponents of the Fedora design, applications, and community, but they have had to create custom solutions to address repository scale.

While such scaling needs are becoming more typical, the Fedora 4 software implementation was still limited by an underlying software component, ModeShape, that was designed for more typical content management and could not meet all of these demands. Importantly, Fedora 4's storage layer is limited to large storage pools for both metadata and files, a performance bottleneck. The large storage pools also limit the total storage capacity of any repository. Institutions adjust their budgets to purchase expensive high-capacity storage arrays, so that their single storage pool can grow as large as possible. Fedora 4's storage pool is a single point of failure that impacts reliability and remains the limiting factor on repository growth for the Fedora community.

Since the first release of Fedora 4, the technical team has made many strides towards better performance at scale, including a major upgrade that included an improved ModeShape 5.1 storage system. There is also a Performance and Scale Working Group that has been measuring and attempting to address Fedora performance issues across six key metrics[5]. They have performed extensive testing of various storage systems and configurations of Fedora 4.x. They have produced metrics and analysis[6] for every available storage arrangement for Fedora 4. This testing has identified three main areas of concern:

- Single server or storage pool error conditions arise after a few million storage operations, such as "timeout", "too many open files", "out of memory", and barring these others, "disk full".

[5] Fedora Performance and Scalability Test Plans: https://wiki.duraspace.org/display/FF/Performance+and+Scalability+Test+Plans#PerformanceandScalabilityTestPlans-Test1-Sizeoffiles-large

[6] These metrics and analysis are available in separate folders by configuration in their Git repository: https://github.com/fcrepo4-labs/fcrepo_perf_analysis/tree/master/dist

- Performance degrades when the underlying ModeShape folders grow larger than 1000 items. This is described as the many members problem[7], and is an instance of the "supernode" problem, discussed later in research questions.
- Performance degrades when many Fedora clients are performing simultaneous tasks, due to transaction checking overhead.

While performance of Fedora 4 has been improved, these remain the most intractable issues.

Web-Scale Technologies from Industry

The Fedora user community's needs for storage are unique among content management applications, but not unique among web-scale companies. Storing millions of objects and their metadata is not unlike storing the data and preferences for millions of users. Large companies like NetFlix, Facebook, and Google have consistently succeeded in meeting this challenge by adopting a different strategy for storage. The Fedora 4 strategy for growth is one where you plan, budget, build, and finally migrate into a single storage system that is large enough to meet expected demand. In contrast, the web-scale companies meet demand by adding more, separate storage systems to a group of similar systems, a cluster, that work together to store the data. Each system in the cluster is relatively small and cheap, so you no longer have to plan and build cost-prohibitive enterprise storage arrays. Storage cluster technologies allow for growth while maintaining a smooth, predictable cost for increases in capacity. The performance of a storage cluster is not limited by a single bottleneck system.

One of the most widely adopted and successful industry solutions for distributed data storage is called Apache Cassandra[8] and it is a free and open source project. Apache Cassandra is the most popular distributed database available and it also ranks 8th among databases overall[9]. It is part of the family of so-called NoSQL databases and is second in use and popularity to MongoDB and is in use with CERN, eBay, GitHUB, Hulu, Instagram, Netflix, Twitter, and at 1500 other companies, and scales to petabytes of storage and billions of objects. An Apache Cassandra cluster can scale up to 75,000 separate servers (perhaps the largest so far, at Apple, Inc.) and beyond. This is the technology that DRAS-TIC uses for storing files and metadata and that we think will prove a good fit for Fedora community needs. A Fedora institution might begin with a cluster of just two or three servers, and then proceed to grow this cluster incrementally as their collections grow.

Avoiding "Big Bang" Storage Planning

Many Fedora institutions have trouble predicting costs and planning storage capacity for large new collections. When the current enterprise storage system is nearly full, adding new collections may be postponed for years, until the next, higher capacity, storage system can be designed, budgeted, and built. These long planning cycles make it difficult to estimate costs for collections, which stifles local collection efforts and innovation. In contrast, using a cluster approach allows one to simply count the number of new, separate servers required to

---

[7] See Fedora's wiki page: https://wiki.duraspace.org/display/FF/Many+Members+Performance+Testing
[8] Details about Apache Cassandra at: https://cassandra.apache.org/
[9] See relational vs. NoSQL database rankings at: https://db-engines.com/en/ranking

store the new collection. These marginal costs of new collections are simple to calculate, and servers may be purchased and added to the existing cluster as needed. The current Fedora storage strategy can have a serious impact on collection development, budgeting, and planning long before the total size of collections have an impact on performance.

2. Project Design

The University of Maryland iSchool's Digital Curation Innovation Center, DCIC, is in a unique position to deliver a cluster-based storage strategy for the Fedora community. We have been involved in developing a cluster-based archival repository software, called DRAS-TIC [10] (Digital Repository At Scale - That Invites Computation). This approach was designed from the ground up to leverage the Apache Cassandra cluster-based storage system and provide hierarchical storage of billions of digital objects and their metadata. DRAS-TIC currently implements an open-source industry standard API called the Cloud Data Management Interface[11] (CDMI) that defines the way in which applications can create, retrieve, update, and delete datasets, resources, and metadata. It has demonstrated dramatic improvements in performance and capacity without the liabilities of expensive storage, enterprise-style storage planning, and data loss that have plagued non-cluster storage strategies. We currently manage 100 Million files with metadata and 100 Terabytes of storage in the cluster at University of Maryland. This project will work closely with the Fedora community and Steering Group to further develop DRAS-TIC into an alternate implementation of the Fedora API. This means adding the Fedora API to the existing CDMI API, or replacing the CDMI API. The project will addresses an important shortcoming in the existing Fedora implementation, towards meeting the most demanding and extreme storage requirements.

Research Questions

- **RQ1:** *Is it possible and practical to maintain both the CDMI API and the Fedora API in one software platform?*
    - Can the CDMI API be used to edit basic Fedora metadata without data loss?
- **RQ2:** *How best to use the graph database available in* DRAS-TIC *to persist and maintain linked data of the form supplied through the Fedora API?*
    - Generalize approaches to storing RDF in a graph database
    - Develop asynchronous approaches to maintaining distributed linked data, such as a "garbage collection" strategy.
- **RQ3:** *Does* DRAS-TIC*'s distributed storage offer newly distributed or decentralized modes of building software with Fedora?*
    - New software architectures: We may further advance the notion, popularized by the Hydra community, of many applications sharing a common, multi-tenanted repository.
    - Practical backup strategies for distributed data

---

[10] DRAS-TIC scalable repository project: https://umd-drastic.github.io/about/

[11] Opensource CDMI API Industry standard for cloud storage: https://www.snia.org/cdmi

- **<u>RQ4:</u>** *Does* DRAS-TIC*'s distributed storage afford new forms of shared storage and federation across institutions?*
    - Reciprocal storage mirroring across datacenters
    - Loading dock clusters
- **<u>RQ5:</u>** *How can we address the problem of super-nodes?* "Super-nodes" are digital objects with many more connections than other objects, such as a folder that holds a million content items? This is a common Fedora pattern that creates multiple issues for the current implementation.

Research Methods

Given the inherent complexity of distributed storage solutions, we don't feel that cluster-based approaches can be recommended to institutions without strong guidelines on cluster configuration and operations that have been proven in full scale tests. While this project proposes a technical product as one outcome, it is primarily designed to maximally explore the software implementation and cluster configuration possibilities through rapid prototyping and iteration over potential *DRAS-TIC Fedora* architectures.

There are many ways to integrate the components we have identified and indeed myriad ways to configure and use a Cassandra cluster, all of which impact performance, consistency, and reliability. Our research will include the design of graph database schemas, Cassandra table schemas, cluster tunings, consistency rules, and replication rules. We will also explore multi-datacenter replication as a way to provide of geographic and datacenter redundancy. Each candidate configuration of the system will be subjected to uniform testing and measurement, including the running of test suites from the Fedora project itself. Guided by the measures we produce through tests, we will refine and tune the cluster configuration to meet the use cases and needs identified by our partners.

The tests that are run against candidate clusters will produce metrics that will be archived as part of the project, similar to the Fedora metrics produced for Fedora 4.x by the performance and scalability group. Each of the tested configurations will remain tagged and referenceable in a public git repository, alongside complete server deployment scripts that may be used to reproduce the cluster configuration at the time of the test. These reference points will be citable in discussions with partners and can provide configuration "fork points" any institution that find a metric suitable for their particular use case.

With a consistent method for producing and archiving metrics across potential implementations, we will be freed to explore research questions and optimize for different partner use cases, trying new things without losing site of the core need of performance and scalability. The *DRAS-TIC Fedora* metrics will be compared with the Fedora 4.x metrics to determine the recommended architecture choices and trade-offs for performance at various scales.

Project Leadership

- **Richard Marciano**, professor, will act as project director. He is the founder and director of the Digital Curation Innovation Center (DCIC), and director of the Sustainable Archives and Leveraging Technologies (SALT) lab. Marciano has been the Principal Investigator for grants from IMLS, NSF, the Library of Congress, and the National Archives and Records Administration. He is currently the UMD Principal Investigator on a joint NSF/DIBBs (UIUC/NCSA & UMD) software cyberinfrastructure grant called "Brown Dog"[12], the Principal Investigator on an IMLS digital curation professional training and virtual infrastructure grant called "CurateCloud", and a collaborator on a large NSF/IIS grant on pervasive ethics for computational research (PI: Katie Shilton at the UMD iSchool) where the DRAS-TIC repository will be used as the main repository. In 2009, Marciano was the Principal Investigator on an IMLS NLG Research grant with DuraSpace called Policy-Driven Repository Interoperability (PoDRI). The principal focus of PoDRI was to investigate the feasibility of interoperability mechanisms between repositories at the policy level. At the time the research focused on the integration of an object model and a policy-aware distributed data model with Fedora and iRODS as representative open source software for each model. With recent Fedora 4 API developments, there are now renewed opportunities for making a national impact.

- **Gregory Jansen** is the principal research software architect at the Digital Curation Innovation Center at the University of Maryland, where he is developing the DRAS-TIC repository software. He is a contributor to the NCSA Brown Dog project to develop digital curation web services, including the design of scalability and performance testing of the Brown Dog framework. He has been a contributor to the Fedora 4 Technical Team, and the creator of the Curator's Workbench archival packaging software. His past work includes being a technical advisor to the Academic Preservation Trust and the technical lead on the Carolina Digital Repository at UNC Chapel Hill.

Project Support

- **The Digital Curation Innovation Center (DCIC)** at the UMD iSchool will support major project activities. Its mission is to "Integrate Education and Research" in the field of Digital Curation using Big Records and Archival Analytics. It comprises 5 labs [see: http://dcic.umd.edu/infrastructure/] including the dataCave, a peta-scale storage facility powered by NetApp, providing commercial-grade storage to support the DRAS-TIC repository. DRAS-TIC was developed in collaboration with the DCIC and Archives Analytics Solutions Ltd., a software development firm in the United Kingdom, as the result of the $10.5 million NSF "Brown Dog" grant. The DCIC launched the DRAS-TIC repository at iPRES 2016 in Oct. 2016 after the University of Maryland concluded a significant license agreement that established its clear ownership and use.

---

12 NSF/DIBBs Brown Dog project: http://browndog.ncsa.illinois.edu/

- **Student Involvement:** The DCIC promotes both projects (focused on justice, human rights & cultural heritage) and cyberinfrastructure for the curation & management of digital assets at scale. Its approach has been to develop interdisciplinary student research teams to help develop new digital skills, conduct interdisciplinary research, and explore professional development opportunities at the intersection of archives, big data, and analytics through focused project themes which include: *Refugee Narratives, Community Displacement, Racial Zoning, Cyberinfrastructure for Digital Curation, Movement of People, Citizen Internment, and others.* We will create a student-centered team around this project for the two-years of the project and promote the development of repository testing and prototyping, and incorporate lessons learned in related classes.

Community Partners

In addition to the endorsement of the Chair of Fedora Leadership Group and Steering Committee, we have reached out to a number of institutions in research libraries, cultural institutions, and research organizations. We will recruit new use cases from these partners for the *DRAS-TIC Fedora* software. These institutions will provide diverse requirements, including repositories that have very large data files and repositories that store and access large amounts of linked data. We will provide assistance to these and other Fedora institutions so they can try *DRAS-TIC Fedora* for their collections and provide feedback. Initial partners include:

- *University of Maryland Libraries:*
  - Digital Systems and Stewardship
  - **Interest:** Managing Fedora in production and managing research data through the API specification.
- *Smithsonian Institution:*
  - Office of Research Info. Services (ORIS) within the Office of the CIO (OCIO)
    and
  - National Museum of American History
  - **Interest:** Curation of digital outputs using the SIdora Information Architecture which is based on Fedora (looking at genomics, hi-res multispectral 3D imaging, archaeology, museum, curation, etc.).
- *U. of Illinois at Urbana-Champaign:*
  - National Center for Supercomputing Applications (NCSA)
  - **Interest:** Establish Research Data Services (RDS) to preserve research data associated with research publications, by incorporating Brown Dog into repository technologies.
- *Georgetown University Library:*
  - Library Information Technology (LIT)
  - **Interest:** Currently using DSpace API. Interest in exploring repository scalability

Project Activities

The project will be roughly divided into three distinct phases for the purpose of start-up, exploration, and engagement respectively. Activities in earlier phases may continue in later phases as we iterate through architectures, testing apparatus, and partner use cases.

- **Phase One: Start Up (6 months)**

1. Create project structure and collaboration tools. Establish the agile software development process and related roles. Recruit student involvement.
2. Formalize communication with community partners and other stakeholders.
    - Continue to work with Fedora developers and Fedora Leadership Group and Steering Committee to ensure that this project aligns with Fedora conventions and the needs outlined above.
3. Develop the Fedora community partner relationships to identify the range of use cases and potential adopters for DRAS-TIC's capabilities.
   Explore research questions (3) and (4) with Fedora stakeholders:
    - **RQ3:** *Does* DRAS-TIC*'s distributed storage offer newly distributed or decentralized modes of building software with Fedora?*
    - **RQ4:** *Does* DRAS-TIC*'s distributed storage afford new forms of shared storage and federation across institutions?*
4. Create an easily installed *DRAS-TIC Fedora* Vagrant machine for use by all developers and to facilitate community evaluation.
5. Create a reconfigurable *DRAS-TIC Fedora* cluster for use as a performance testbed.
6. Develop server deployment and management scripts for *DRAS-TIC Fedora*, further addressing **RQ3** with regards to backup strategies.

- **Phase Two: Explore (one year)**

7. Extend and enhance Fedora's API and performance test suites to provide any missing metrics established by partner uses cases.
8. Create a first prototype of the Fedora API running on DRAS-TIC (*DRAS-TIC Fedora*) by implementing so-called CRUD operations (create, read, update and delete). *Address research question (1) and begin to evaluate (2).*
    - **RQ1:** *Is it possible and practical to maintain both the CDMI API and the Fedora API in one software platform?*
    - **RQ2:** *How best to use the graph database available in* DRAS-TIC *to persist and maintain linked data of the form supplied through the Fedora API?*
9. Setup a Continuous Integration workflow for iterative API and performance testing. This means that new *DRAS-TIC Fedora* code will automatically deploy to the testbed cluster and then trigger a fresh round of API and performance tests, capturing the results. With this test harness established, we can feel free to try new cluster and code strategies, knowing the performance will be uniformly captured.
10. ITERATE: Design, implement, and test in an iterative fashion to address partner needs (capturing outcomes through the continuous integration workflow):
    - Implement additional operations from the Fedora API as needed, suitable for a full repository and consistent with partner use cases. *Answer research questions (4) and (5):*
        - **RQ4:** *Does* DRAS-TIC*'s distributed storage afford new forms of shared storage and federation across institutions?*
        - **RQ5:** *How can we address the problem of super-nodes?*
    - Attempt software strategies made possible by DRAS-TIC's distributed storage, *in pursuit of research question (3)*:
        - **RQ3:** *Does* DRAS-TIC*'s distributed storage offer newly distributed or decentralized modes of*

building software with Fedora?

- Linked data grooming or "garbage collection" strategies that provide "eventual consistency" in the background. (Data visited and assessed outside of the moment of the API transaction.)
- Alternative persistent strategies based on the DataStax graph database or JanusGraph graph database.
- Alternative graph database schema for RDF data.

- **Phase Three: Outreach (6 months)**

11. Create an annotated Fedora API document. Clearly state which parts of the full Fedora API were implemented and why or why not. Identify any limitations to the standard Fedora API.
12. Develop a how-to guide for running a *DRAS-TIC Fedora* repository.
13. Publicize *DRAS-TIC Fedora* performance results with help from the Fedora Leaders Group. Make all metrics available to the community, alongside the code strategies and configurations that produced them.
14. Reach out to the broader Fedora community for adoption and support and present at DLF, CNI, and Open Repositories conferences.
15. If successful in finding community adoption, align *DRAS-TIC Fedora* with the Fedora project, in terms of sustainability and governance. Look to the broader Fedora community for software maintenance and future development.

3. National Impact

The proposed project will produce open source software, tested cluster configurations, documentation, and best-practice guides that will enable institutions to manage Fedora repositories with Petabyte scale collections and thus contribute to big-data ready national software infrastructure. Jansen and Marciano are participating in the IMLS-funded *"Always Already Computational: Library Collections as Data"* project (Principal Investigator: Thomas Padilla), and we will share research outcomes with the broad repository community at DLF, CNI, Open Repositories, and other upcoming venues.

Impact on Fedora Institutions

We expect that Fedora institutions and our community partners will make gains on several fronts through the *DRAS-TIC Fedora* project. Perhaps the most important gain will be the ability to smoothly increase the size of their repository collections without re-architecting storage. This smooth scaling of storage will lower costs and shorten planning cycles, allowing institutions to be more responsive to new and challenging collecting opportunities. Another impact can be better reliability, in terms of collection access and redundant datacenter storage. With Cassandra, *DRAS-TIC Fedora* can hold redundant copies of data in data centers around the world. Even if an entire data center is lost, access can remain uninterrupted, and data re-replicated to a new data center. Beyond institutional reliability, we think this opens up new possibilities for reciprocal storage arrangements between Fedora institutions.

Impact on Computational Archival Science

More broadly speaking we expect *DRAS-TIC Fedora*, by maintaining performance at scale and through distributed computation, to be a significant enabling contribution to the emerging field of Computational Archival Science (CAS).[13] Over the last year, we have convened key partners and led workshops to help shape emerging national conversations. In August 2017 we led an SAA Pop-up session on the theme of archival records in the age of big data.[14] Our working definition of CAS is:

> *Contributing to the development of the theoretical foundations of a new trans-discipline of computer and archival science*

Our current objectives are:

- Contributing to the development of the theoretical foundations of a new trans-discipline of computer and archival science
- Designing the educational foundations and delivering training in this emerging trans-discipline to support all industries and fields
- Developing a virtual and physical laboratory to test and apply scientific advances in a collaborative environment

We have recently authored:

- a Foundational Paper on **"Archives Records and Training in the Age of Big data"** to be published as a book chapter in 2017 in: *"Advances in Librarianship – Re-Envisioning the MLIS: Perspectives on the Future of Library and Information Science Education".*[15]
- a National Forum Position Paper on **"The Computational Turn in Archives & Libraries and the Notions of Levels of Computational Services,"** presented at the *"Always Already Computational: Library Collections as Data"* Forum in March 2017 at the UC Santa Barbara workshop.[16]

We will rally the CAS community as early as mid-December 2017 in Boston at our 2nd IEEE Big Data 2017 CAS Workshop[17], where the kinds of national infrastructure integration and interoperability questions explored in this proposal will be at the forefront. We will also propose a session on this theme at the 2018 Annual Meeting of the Society of American Archivists in Washington D.C.

---

[13] Computational Archival Science (CAS): http://dcicblog.umd.edu/cas/

[14] SAA 2016 pop-up session on CAS: https://archives2016.sched.com/event/7f9D

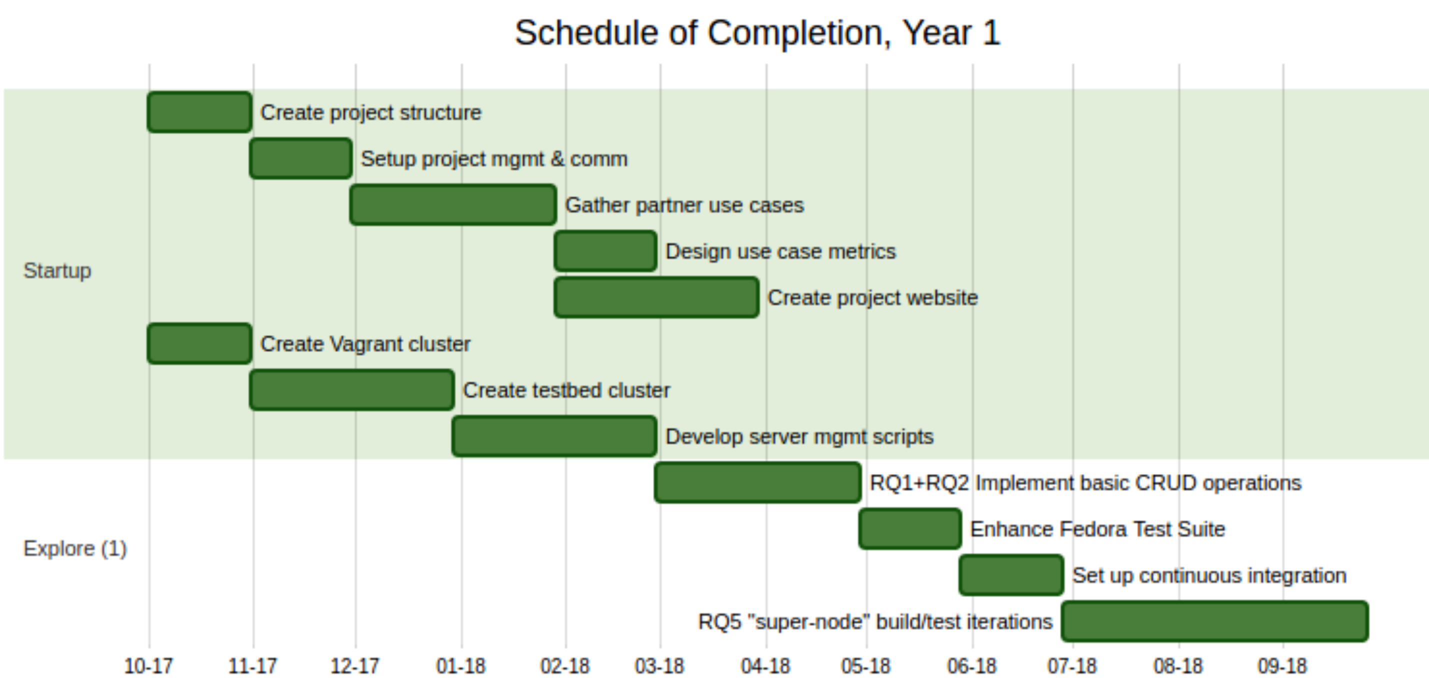
[15] Foundational Paper on "Archives Records and Training in the Age of Big data": http://dcicblog.umd.edu/cas/wp-content/uploads/sites/13/2016/05/submission_final_draft.pdf

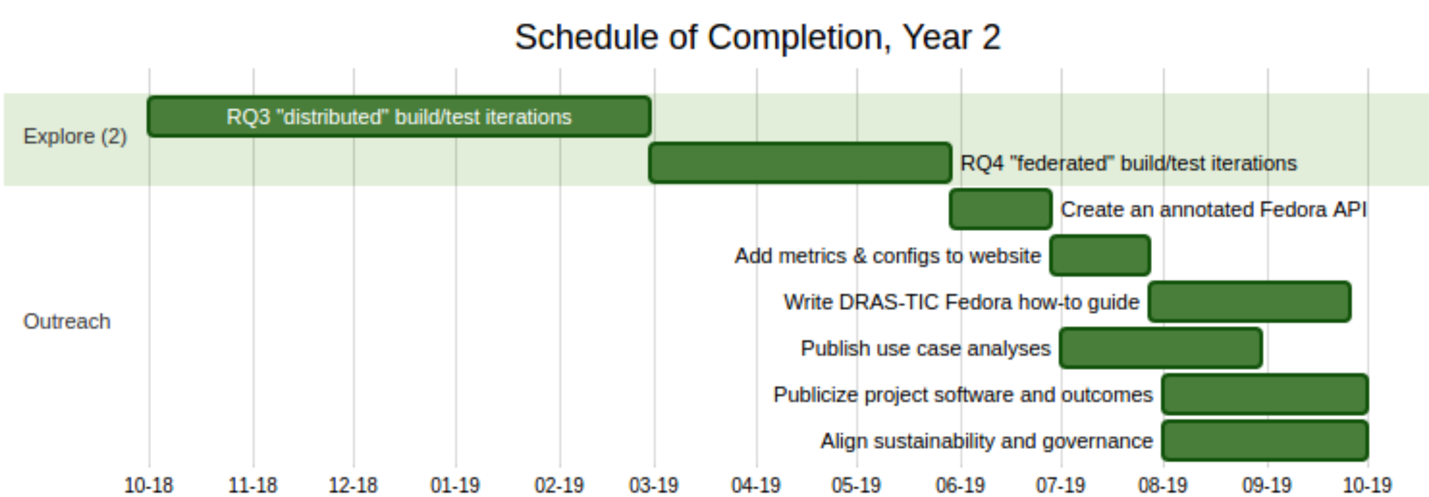
[16] National Forum Position Paper on "The Computational Turn in Archives & Libraries..." presented at the *"Always Already Computational: Library Collections as Data"* Forum in March 2017 at the UC Santa Barbara workshop: http://dcicblog.umd.edu/cas/wp-content/uploads/sites/13/2016/05/AlwaysAlreadyComputationalPositionStatement.pdf

[17] 2nd IEEE Big Data 2017 CAS Workshop: http://cci.drexel.edu/bigdata/bigdata2017/index.html

Schedule of Completion

Performance Period: October 2017 through September 2019
Showing three phases: Startup, Explore, and Outreach
and
the **embedding of research questions**: RQ1, RQ2, RQ3, RQ4, RQ5.





DIGITAL PRODUCT FORM

Introduction

The Institute of Museum and Library Services (IMLS) is committed to expanding public access to federally funded digital products (i.e., digital content, resources, assets, software, and datasets). The products you create with IMLS funding require careful stewardship to protect and enhance their value, and they should be freely and readily available for use and re-use by libraries, archives, museums, and the public. However, applying these principles to the development and management of digital products can be challenging. Because technology is dynamic and because we do not want to inhibit innovation, we do not want to prescribe set standards and practices that could become quickly outdated. Instead, we ask that you answer questions that address specific aspects of creating and managing digital products. Like all components of your IMLS application, your answers will be used by IMLS staff and by expert peer reviewers to evaluate your application, and they will be important in determining whether your project will be funded.

Instructions

You must provide answers to the questions in Part I. In addition, you must also complete at least one of the subsequent sections. If you intend to create or collect digital content, resources, or assets, complete Part II. If you intend to develop software, complete Part III. If you intend to create a dataset, complete Part IV.

PART I: Intellectual Property Rights and Permissions

A.1 What will be the intellectual property status of the digital products (content, resources, assets, software, or datasets) you intend to create? Who will hold the copyright(s)? How will you explain property rights and permissions to potential users (for example, by assigning a non-restrictive license such as BSD, GNU, MIT, or Creative Commons to the product)? Explain and justify your licensing selections.

This project will produce software code that we would like to make freely available via a public Git repository. The license to use this code will be featured prominently in the documentation and the full text of the license agreement will be published as a text file in every code repository. Unless the DRAS-TIC Fedora software license needs to align more closely with the Fedora project's own software license, which is to be determined, we will keep the same license that DRAS-TIC has now. This is the GNU Afferro Public License, version 3. This license ensures that the software will remain freely available to anyone who want to copy, distribute, or modify. However, the GNU APL requires that any commonly used modifications, even if they are publicly used only indirectly over a network, must also be made available as source code. This will keep various enterprises from simply forking the code for their own purposes, without contributing their work back to the original project.

A.2 What ownership rights will your organization assert over the new digital products and what conditions will you impose on access and use? Explain and justify any terms of access and conditions of use and detail how you will notify potential users about relevant terms or conditions.

We assert only copyright on the software products. All products will be freely available for access and use.

A.3 If you will create any products that may involve privacy concerns, require obtaining permissions or rights, or raise any cultural sensitivities, describe the issues and how you plan to address them.

We plan to use publicly available data for all testing of the DRAS-TIC Fedora platform. This data will be free of privacy concerns and will largely consist of government, cultural, and scientific data that is either publicly available or does not involve human subjects. If any data used in the project is found later to raise such privacy concerns, we will immediately remove it from the project.

Part II: Projects Creating or Collecting Digital Content, Resources, or Assets

A. Creating or Collecting New Digital Content, Resources, or Assets

A.1 Describe the digital content, resources, or assets you will create or collect, the quantities of each type, and format you will use.

Documentation will be produced as part of the project, including a how-to guide and an annotated API. Both of these will be published on a website, as well as available in a raw form as a public git repository. Documentation will be written either in Markdown text files or as HTML.

A.2 List the equipment, software, and supplies that you will use to create the content, resources, or assets, or the name of the service provider that will perform the work.

We will not restrict the software used to write the DRAS-TIC Fedora code. Developers will be free to use the software or editor of their choice, provided that it does not contain the resulting product. We will use GitHUB as a service provider for git repository hosting. The DRAS-TIC Fedora testing will be performed on a hardware cluster at the DCIC at Maryland's iSchool, within a co-location facility provided by the Division of Information Technology.

A.3 List all the digital file formats (e.g., XML, TIFF, MPEG) you plan to use, along with the relevant information about the appropriate quality standards (e.g., resolution, sampling rate, or pixel dimensions).

We plan to use every available file format for testing DRAS-TIC Fedora. We have a large set of born digital government records at the DCIC, which will be used as test inputs. In addition we may receive test data from community partners and other sites. These files will be of every conceivable format. Outside of test data, we will be using plain text formats for most things, in the form of Python code and markup languages. The only proprietary formats we can anticipate within the code project might be for Windows or OS X installer programs for client software.

B. Workflow and Asset Maintenance/Preservation

B.1 Describe your quality control plan (i.e., how you will monitor and evaluate your workflow and products).

Software quality control will be enforced by rigorous compliance and performance testing. We will employ LDP Test Suite (https://w3c.github.io/ldp-testsuite/) as we proceed to evaluate what parts of the LDP API are implemented and working. We will use the Fedora projects test suite to evaluate performance of the cluster under specific loads. Finally, we may also use the Gatling test framework to simulate specific user load patterns over time. Our project analyst will also be responsible for quality assurance for the website and all project documentation and procedures. We will solicit feedback from our community partners on the clarity and ease of use of these documents and the software itself.

B.2 Describe your plan for preserving and maintaining digital assets during and after the award period of performance. Your plan may address storage systems, shared repositories, technical documentation, migration planning, and commitment of organizational funding for these purposes. Please note: You may charge the federal award before closeout for the costs of publication or sharing of research results if the costs are not incurred during the period of performance of the federal award (see 2 C.F.R. § 200.461).

During the project our assets will largely remain on GitHUB as the site of collaboration. After the project period, we plan to leave these outputs on GitHUB, where they will continue to receive attention and modifications if we are successful in delivering value to our partners. A snapshot of the data at the end of the project period will remain available on GitHUB, as well as in an archival copy with the DCIC.

C. Metadata

C.1 Describe how you will produce any and all technical, descriptive, administrative, or preservation metadata. Specify which standards you will use for the metadata structure (e.g., MARC, Dublin Core, Encoded Archival Description, PBCore, PREMIS) and metadata content (e.g., thesauri).

This project will produce metadata throughout our testing and evaluation of software implementations. The metadata may include Dublin Code, MODS, EAD, or simple key/value terms. However, the metadata is not an output of the project and will not be preserved beyond generation and use. The metadata in this projects is part of the test data for software evaluation. Creating, reading, updating, and deleting metadata are all operations that we will be testing throughout the project. Much of the metadata involved will be created by software services, such as the Brown Dog API, from our partners at NCSA. Documentation will be produced with some limited metadata regarding authorship and contributions, software release dates and versions, and relevant API versions and references for Fedora, LDP, and Memento.

C.2 Explain your strategy for preserving and maintaining metadata created or collected during and after the award period of performance.

The metadata preserved after the period of performance will consist in the outcomes of software tests. Tests will need to be logged with precision, and may include these data elements: (1) Test date and duration, (2) Usage scenario tested, e.g. 25 folders created per second, for an hour, (3) Version of the DRAS-TIC Fedora code tested, a GIT commit reference, (4) Test suite and version, a GIT commit reference, (5) Configuration options tested, a GIT commit reference, and (6) Metrics obtained, likely a JSON file. These and more data elements will remain as references after the project is finished. This data will need to remain linked to specific code versions, specific configuration sets, and test outcomes. The data will have to remain navigable on the project website.

C.3 Explain what metadata sharing and/or other strategies you will use to facilitate widespread discovery and use of the digital content, resources, or assets created during your project (e.g., an API [Application Programming Interface], contributions to a digital platform, or other ways you might enable batch queries and retrieval of metadata).

We will be sharing all of the software source code, deploy scripts, and documentation, such that any institution can install DRAS-TIC Fedora locally and try it. At the same time, we will publish the test outcomes from various software iterations and make them available, so that any institution may reproduce the same test, using the same code and configuration.

D. Access and Use

D.1 Describe how you will make the digital content, resources, or assets available to the public. Include details such as the delivery strategy (e.g., openly available online, available to specified audiences) and underlying hardware/software platforms and infrastructure (e.g., specific digital repository software or leased services, accessibility via standard web browsers, requirements for special software tools in order to use the content).

We would like to make the documentation, software code, and test data freely available online, both through a website and via the git protocol.

D.2 Provide the name(s) and URL(s) (Uniform Resource Locator) for any examples of previous digital content, resources, or assets your organization has created.

DRAS-TIC Repository Website: https://umd-drastic.github.io/
International Research Portal: http://irp2.ehri-project.eu/
Mapping Inequality: http://dcicblog.umd.edu/mapping-inequality/

Part III. Projects Developing Software

A. General Information

A.1 Describe the software you intend to create, including a summary of the major functions it will perform and the intended primary audience(s) it will serve.

We intend to create a new version of the Fedora repository software that will be based upon an existing software called DRAS-TIC, called DRAS-TIC Fedora. The existing DRAS-TIC software implements a web interface and a set of functions call the cloud data management interface (CDMI). DRAS-TIC Fedora will implement a subset of the functions documented in the Fedora API specification (http://fedora.info/spec/). For example we intend to implement all of the basic create, read, update and delete function, but we may or may not implement Fedora's versioning strategy (Memento) and will not implement Fedora atomic batch feature. The primary audience will remain similar to the primary audience for Fedora, which is mainly galleries, libraries, archives, and museums. However, we also hope to overcome some of the limitations of the current Fedora implementation and attract some attention from more demanding cultural heritage institutions that do not use Fedora.

A.2 List other existing software that wholly or partially performs the same functions, and explain how the software you intend to create is different, and justify why those differences are significant and necessary.

This new software, DRAS-TIC Fedora, largely mimics the functions of Fedora 4, as newly described in an API specification that is now called Fedora 5. These are traditional digital repository functions for storing and accessing digital objects, but now designed for linked data representations. Indeed the Fedora 5 API is largely based on the Linked Data Platform specification. The key difference between Fedora 4 and DRAS-TIC Fedora, and the rationale for this project, is that we will implement these same function with very different performance characteristics and storage scalability. Please see our statement of national need and the statement of impact for the complete justification for the project. We, and our partners, feel that Fedora's existing implementation has significant limitations that constrain what collecting institutions may do with it. These limitations are largely due to a vertical storage strategy that puts all digital objects in one storage pool. We will instead embrace a horizontal (or distributed) storage pattern, which will spread storage across a cluster. A well designed storage cluster may grow very large without impacting performance and while maintaining a smooth cost curve for additional capacity.

B. Technical Information

B.1 List the programming languages, platforms, software, or other applications you will use to create your software and explain why you chose them.

This project will create source code for client and server software to run the DRAS-TIC Fedora repository. It will create several versions of that software, which will exist in several public git source code repositories. The format of these assets will be Python code and ancillary files. Python was chosen because it is the language that the original DRAS-TIC repository is written in and this project will extend that codebase to implement and test the Fedora API. Python is also a common skillset among web developers in libraries and other collecting institutions. Apache Cassandra supplies the underlying storage cluster for DRAS-TIC Fedora data. This is a well tested and widely used platform for web-scale companies, such as Apple, Netflix, Github and others. Cassandra is open source and written primarily in Java. It is a distributed database that offers replication and redundancy across a highly available cluster. It has unique performance characteristics that results from a feature called "tunable consistency". This means that you can ask for a level of data consistency that is appropriate for your application's storage request. The deployment scripts and setting may take various forms, but primarily we expect to use YAML (Yet Another Markup Language) to encode both deployment scripts (Ansible playbooks) and settings files. Ansible will be used to manage the cluster and application hosts. The test suites for this project, which are various, include a Fedora test client, written in Java; a LDP compliance test suite; and possibly a series of test scenarios implemented in the Gatling test framework and Scala. Gatling is being used at the DCIC currently for tests of the Brown Dog infrastructure for our NCSA partners.

B.2 Describe how the software you intend to create will extend or interoperate with relevant existing software.

DRAS-TIC Fedora will interoperate with all existing Fedora applications that make use of the Fedora API. This means that Fedora 4 institutions may consider a move to DRAS-TIC Fedora in the future with minimal adjustments to their own application code. In addition, the DRAS-TIC software as currently written provides a Cloud Data Management Interface (CDMI). This is a RESTful API that provide cloud storage operations, similar to Amazon S3 storage. DRAS-TIC Fedora may or may not continue to provide a CDMI endpoint. This depends primarily on how well or poorly we can use CDMI to edit Fedora metadata. It may prove too complicated to keep around. The DRAS-TIC repository has a messaging system that publishes notices as content items are changed. This messaging system is implemented as an MQTT broker and compatible with many languages. We intend to keep the messaging system running as we implement the Fedora API. Finally, the linked data we will persist in DRAS-TIC will either live in Cassandra tables, or in a graph database (layered over Cassandra tables). This means that the repository will be accessible in a new way, either via CQL Cassandra queries, or via Gremlin graph query language. Either of these will offer newly parallel modes of inspecting data. A graph database query may run on many nodes across the entire cluster to find relevant answers. Similarly, an Apache Spark query may run in parallel on many cluster nodes to perform calculations or aggregate data. These are new affordances for the Fedora repository world that do not exist today.

B.3 Describe any underlying additional software or system dependencies necessary to run the software you intend to create.

The DRAS-TIC Fedora software will be designed primary to run on a Linux/Debian operating system and may be run in virtual machine environments or the cloud. Python and Nginx are both required components to run the software.

B.4 Describe the processes you will use for development, documentation, and for maintaining and updating documentation for users of the software.

We intend to use a Markdown format for authoring documentation. This will be stored in a repository on the GitHUB service and published to a GitHUB.io website as HTML. This allows us to easily version and publish documents for public consumption. Markdown is a plain text format that can be edited in any text editor.

B.5 Provide the name(s) and URL(s) for examples of any previous software your organization has created.

Our organization, the DCIC, has created several software tools, some internal and some public. The closest effort to this one has been the DRAS-TIC repository software, which is the basis for DRAS-TIC Fedora. The DRAS-TIC code is freely available on GitHub (https://github.com/UMD-DRASTIC/). We are currently engaged in producing our first stable release, with documentation published on GitHub.io (https://umd-drastic.github.io/). Our organization has also created public-facing software for various digital humanities websites. Lastly we have created a virtual computer lab for teaching digital curation topics. This consists of a web dashboard that allows students and faculty to create and use remote desktops running in the Amazon cloud. These desktops are pre-configured from images to contain all the software needed for a class or activity.

C. Access and Use

C.1 We expect applicants seeking federal funds for software to develop and release these products under open-source licenses to maximize access and promote reuse. What ownership rights will your organization assert over the software you intend to create, and what conditions will you impose on its access and use? Identify and explain the license under which you will release source code for the software you develop (e.g., BSD, GNU, or MIT software licenses). Explain and justify any prohibitive terms or conditions of use or access and detail how you will notify potential users about relevant terms and conditions.

The license to use this code will be featured prominently in the documentation and the full text of the license agreement will be published as a text file in every code repository. Unless the DRAS-TIC Fedora software license needs to align more closely with the Fedora project's own software license, which is to be determined, we will keep the same license that DRAS-TIC has now. This is the GNU Afferro Public License, version 3. This license ensures that the software will remain freely available to anyone who want to copy, distribute, or modify. However, the GNU APL requires that any commonly used modifications, even if they are publicly used only indirectly over a network, must also be made available as source code. This will keep various enterprises from simply forking the code for their own purposes, without contributing their work back to the original project. If we do choose to align our license with that of Fedora, they use the Apache 2.0 for their software.

C.2 Describe how you will make the software and source code available to the public and/or its intended users.

This project will produce software code that will be freely available via a public Git repository. Major releases will be described and downloadable from a website or GitHUB.

C.3 Identify where you will deposit the source code for the software you intend to develop:

Name of publicly accessible source code repository: GitHUB Git Repository (as with DRAS-TIC currently)

URL: https://github.com/UMD-DRASTIC/ (or a new repository if we decide to fork DRAS-TIC for Fedora)

Part IV: Projects Creating Datasets

A.1 Identify the type of data you plan to collect or generate, and the purpose or intended use to which you expect it to be put. Describe the method(s) you will use and the approximate dates or intervals at which you will collect or generate it.

The datasets preserved after the period of performance will consist in the outcomes of periodic software tests. A complete battery of these tests will run often, as new code and configuration options are created. Our proposal includes a continuous integration workflow which will trigger performance testing automatically from our Git repository. Tests will be logged with precision, and may include these data elements: (1) Test date and duration, (2) Usage scenario tested, e.g. 25 folders created per second, for an hour, (3) Version of the DRAS-TIC Fedora code tested, a GIT commit reference, (4) Test suite and version, a GIT commit reference, (5) Configuration options tested, a GIT commit reference, (6) Metrics obtained, likely a JSON file These and more data elements will remain as references after the project is finished. This data will need to remain linked to specific code versions, specific configuration sets, and test outcomes. The data will have to remain navigable on the project website.

A.2 Does the proposed data collection or research activity require approval by any internal review panel or institutional review board (IRB)? If so, has the proposed research activity been approved? If not, what is your plan for securing approval?

Our data collection is related to the performance of computer systems and does not need institutional approval.

A.3 Will you collect any personally identifiable information (PII), confidential information (e.g., trade secrets), or proprietary information? If so, detail the specific steps you will take to protect such information while you prepare the data files for public release (e.g., data anonymization, data suppression PII, or synthetic data).

No PII will be collected.

A.4 If you will collect additional documentation, such as consent agreements, along with the data, describe plans for preserving the documentation and ensuring that its relationship to the collected data is maintained.

N.A.

A.5 What methods will you use to collect or generate the data? Provide details about any technical requirements or dependencies that would be necessary for understanding, retrieving, displaying, or processing the dataset(s).

Our test data will consist of logs and data files (JSON or CSV). These files are human readable in plain text, but are best consumed by log analysis software or statistical programs.

A.6 What documentation (e.g., data documentation, codebooks) will you capture or create along with the dataset(s)? Where will the documentation be stored and in what format(s)? How will you permanently associate and manage the documentation with the dataset(s) it describes?

Our process for producing the test datasets will be documented in code documentation and testing plans. The process for analyzing the data from each test will also be documented in code and in our project website. The permanent documentation will be stored in GitHUB after the project is complete. A snapshot of the entire project, including data, docs, and code, will be archived separately at the end of the project.

A.7 What is your plan for archiving, managing, and disseminating data after the completion of the award-funded project?

The data will be interpreted on the public project website and available for download.

A.8 Identify where you will deposit the dataset(s):

Name of repository: GitHUB Git Repository

URL: http://github.com (repository URL to be determined)

A.9 When and how frequently will you review this data management plan? How will the implementation be monitored?

This data management plan will be integral to the project. It will be reviewed at each phase of the project to support ongoing work and long term access.