A Systemic Approach to Facilitating Reproducibility via Federated, End-to-End Data Management

Dale Stansberry, Suhas Somnath, Gregory Shutt, and Mallikarjun Shankar

Oak Ridge National Laboratory, Oak Ridge TN 37831, USA,
stansberrydv@ornl.gov

Abstract. Advances in computing infrastructure and instrumentation have accelerated scientific discovery in addition to exploding the data volumes. Unfortunately, the unavailability of equally advanced data management infrastructure has led to ad hoc practices that diminish scientific productivity and exacerbate the reproducibility crisis. We discuss a system-wide solution that supports management needs at every stage of the data lifecycle. At the center of this system is DataFed - a general purpose, scientific data management system that addresses these challenges by federating data storage across facilities with central metadata and provenance management - providing simple and uniform data discovery, access, and collaboration capabilities. At the edge is a Data Gateway that captures raw data and context from experiments (even when performed on off-network instruments) into DataFed. DataFed can be integrated into analytics platforms to easily, correctly, and reliably work with datasets to improve reproducibility of such workloads. We believe that this system can significantly alleviate the burden of data management and improve compliance with the Findable Accessible Interoperable, Reusable (FAIR) data principles, thereby improving scientific productivity and rigor.

1 Introduction

Scientific research has been facing a reproducibility crisis [5,6,14]. One important and surmountable factor is the typical absence of sufficient information (data, metadata, provenance, workflow, software, etc.) associated with reports on scientific discoveries that are critically important for reproducing the research [20]. Software containers and modern workflow softwares have proven to be reasonably successful in facilitating reproducibility with respect to the software stack [19,8,7,22]. However, readily available, user-friendly, and comprehensive tools to access, search, share, organize, curate, publish, and otherwise manage scientific data remain a long-standing need. This is also an urgent need since the time spent on data management is projected to rise exponentially [17,11] due to the explosion in scientific data [9,15]. Despite the dearth of data management tools, increased globalization of scientific research, and the need to publicly share data [21], facilities and research groups are at best grappling with the data challenges

individually / independently or are typically resorting to ad-hoc methods. These ad-hoc practices not only result in loss / poor quality of data and metadata but also a substantial decrease in scientific productivity.

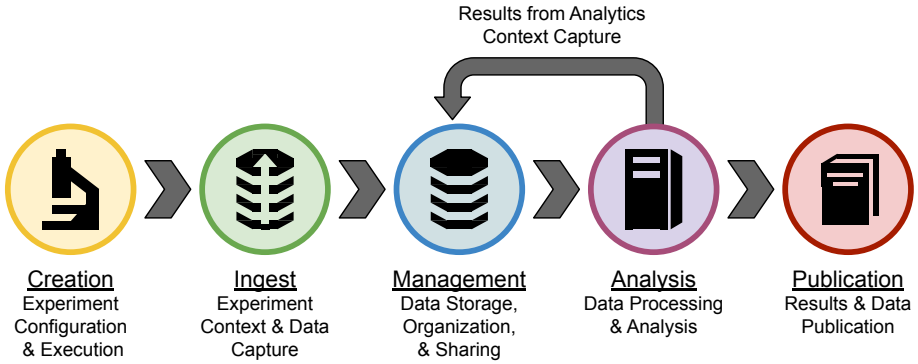

Fig. 1: Data Lifecycle for Reproducibility

Figure 1 illustrates the lifecycle of scientific data. Traditionally, sub-optimal and ad-hoc data management practices occur throughout the lifecycle. Research investigations start with the design, configuration, and execution of experiments which produce scientific data. Most experiments (simulations / observations, etc.) produce metadata that capture the context of the experiment in addition to the raw data itself. At the *ingest* step - since the context regarding experiments is often not comprehensively captured at the source (instrument, simulation module, etc.), researchers manually capture the remaining context (e.g. sample ID, etc.) in physical or electronic lab notebooks in a non-standardized, ad-hoc, and error-prone manner. However, these metadata are rarely collated and therefore do not support the data when necessary.

Moving on to the *management* step - when data is generated off-network (e.g. some scientific instruments), scientists resort to collecting and transporting measurement data using portable storage drives. The collected data and metadata are often stored in traditional file-systems which only provide primitive data sharing, search, and management capabilities. Since data in file-systems are discoverable largely based on file names and paths, most researchers resort to embedding key metadata into the file paths. Since each user stores and represents data and metadata in unique ways, such information collected by users is often usable only by the user who collected the data thereby exacerbating the reproducibility crisis. Desired data are still exchanged using emails, shared folders, and portable storage drives, each having their own set of limitations. Challenges in sharing and reusing data are further exacerbated by the diversity in the representation (schema and ontology for data and metadata), storage (file formats and data repositories), availability (proprietary / open), dimensionality

(1D signals to multidimensional hypercubes) and semantics of scientific data and metadata within and across scientific domains.

At the *analysis* step - results from data processing and analyses are stored back into the file-systems, often without capturing the complete context of the analyses, thereby inheriting many of the aforementioned problems. Finally, at the *publish* step - scientific discoveries are reported / published often without the supporting data. Even when data directly used in the publication are published, data deemed redundant or unimportant for the primary investigation are left untracked, unused, and unpublished despite their latent value [24] leading to the so-called "dark data" [13] problem. When data are published, they are often not discoverable since the scientific metadata associated with the data are not exposed to search engines. As a result of such practices and challenges, it is exceedingly challenging to comply with the Findable Accessible, Interoperable, and Reusable (FAIR) data principles, which were proposed to facilitate open, collaborative, and reproducible scientific research [27].

Improving reproducibility in science through better data practices therefore necessitates the use of comprehensive scientific data management tools that can effectively support scientific data throughout the data lifecycle from *ingest* to *publishing*. Revisiting Figure 1; using data management tools, researchers will be able to *ingest* - comprehensively capture context / metadata along with raw data from experiments, *manage* - intuitively and easily share, search for, organize, transport data, *analyze* - capture secondary data products from analyses and visualization along with context and provenance between products, and *publish* data for reuse in the broader scientific community. Importantly, other researchers should be able to easily find such published data and use the rich metadata and provenance associated with the data to reproduce the original results. Though there are several tools [3] that address specific data management challenges, there are very few flexible, system-wide solutions that support every stage of the data lifecycle for all scientific domains[23,18,4,1,12,25]. Limitations of existing solutions will be discussed later in appropriate sections.

2 Systemic Approach to Reproducibility

To facilitate reproducibility in science, we are proposing a systemic solution that will emphasize and directly support the critical data lifecycle phases of *ingest*, *management*, and *analysis*, shown in Figure 1, that are often overlooked or poorly executed. It is within these data lifecycle phases that full data provenance and rich domain-specific metadata can be captured and utilized to enhance the scientific context needed to ultimately reproduce experimental or computational results. The proposed solution includes components, services, and communication protocols that would be deployed across facilities in order to create a common, FAIR-principled "data federation" - enabling simple, uniform, and performant data access, management, analysis, and collaboration from anywhere within, or across, this federation.

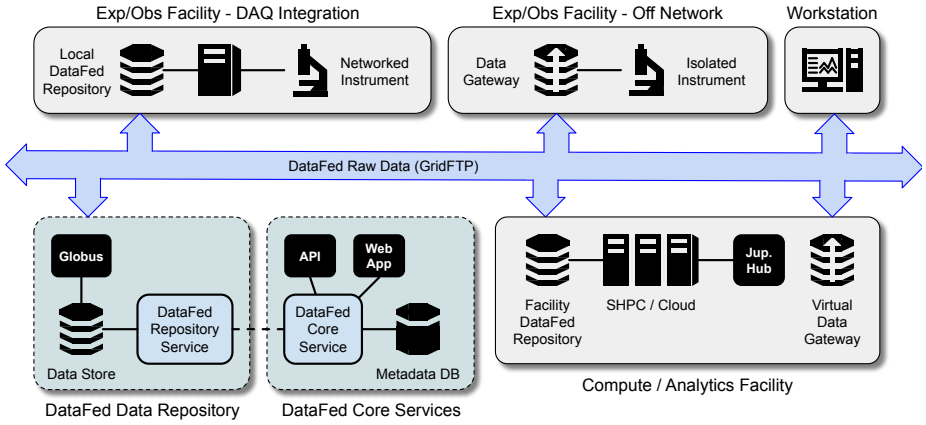

Fig. 2: Proposed Data System Architecture

Figure 2 shows a conceptual view of this system where experimental and/or observation facilities are connected to compute and/or analytics resources via the primary component of the system: a distributed scientific data management system (SDMS) called "DataFed" [26]. The key concepts of DataFed are distributed raw data storage, centralized metadata and provenance management, and performant data transfer. DataFed primarily addresses the needs of the *management* component of the data lifecycle phase; however, two additional components, the "Data Gateway" and "JupyterHub" (as an example), address the needs of the *ingest* and *analytics* phases of the data lifecycle respectively.

In addition to metadata and provenance management, the DataFed central, or "core", services, shown in Figure 2, provide system-wide command and control for raw data access-control and transfer. This is implemented using DataFed-specific application programming interfaces (APIs) and protocols that are used by other system components, such as the Data Gateway, integrated instrument data acquisition (DAQ) systems, or even user compute jobs at high performance computing (HPC) facilities, in order to ingest, locate, access, or share data, on behalf of scientific users. Upon data ingestion, raw data is transferred to DataFed "Data Repositories", which are managed data stores, and, unlike local file systems, these data repositories are connected to the DataFed data network and managed by DataFed core services. DataFed data repositories are not required to be collocated with instruments or facilities, and can be centrally located and/or shared by multiple facilities. An expanded view of a DataFed data repository is shown at the bottom left of Figure 2.

Experimental and/or observational facilities can be directly integrated with DataFed, such as through modification or extension of existing data acquisition or instrument control systems (top-left of Figure 2). For network-isolated instruments (top-right of Figure 2), the "Data Gateway" appliance is available to both provide network buffering as well as easy to use data ingest and context

capture services. The Data Gateway can also be deployed virtually, as shown in the "Compute / Analytics Facility" in Figure 2, to provide general data ingest support for users without access to an DataFed integrated facility.

Data analytics platforms deployed within Compute / Analytics facilities could provide data analytics and visualization capabilities for one or more facilities. We use Jupyter Notebooks [16] and JupyterHub [10] (multi-user) as an example since they capture context regarding analytics for reproducibility. Appropriate DataFed commands could be incorporated within analytics scripts to download / stage data, capture context regarding the analytics, and push results data back to DataFed for management later. By comprehensively capturing the software stack in containers, analytics related context within Jupyter Notebooks, data ingest operations via Data Gateway, and repeatable data operations using DataFed, analysis workloads can be more easily reproduced.

While the described system is intended to address specific aspects of the reproducibility crisis, it is vital that it also be easy for users to learn, adopt, and use. Moreover, use of this system should improve research productivity, not hinder it. The components of this system have been designed with this philosophy in mind - resulting in features and capabilities that directly reduce complexity, improve productivity, and help ensure correctness of data handling when compared to ad-hoc solutions. The individual components of this system are described in detail in sections 3, 4 and 5 below. For general use cases as well as examples of how this system would be useful for modeling, simulations, experiments, and data analytics, refer to Section 6.

2.1 Development and Deployment

The full system solution described above is currently in the design and prototyping stage of development. However, two of the components of the system, the Data Gateway and DataFed, have been partially implemented and deployed at ORNL within the Center for Nanophase Materials Science (CNMS) and the Compute and Data Environment for Science (CADES) facilities, respectively. DataFed is currently deployed as an alpha-release production service. One instance of the Data Gateway has been deployed for scanning probe microscopes at CNMS as a proof-of-concept and is currently capable of authenticating users at the instruments, capturing metadata and transmitting data and metadata to a remote data repository. A dedicated CNMS DataFed repository has been deployed within CADES, and a data repository within the OLCF is planned. In the future, integration with the SNS, and HFIR is anticipated, and JupyterHub services and a Virtual Data Gateway would be deployed within CADES. Additional funding is being actively pursued in order to complete development and deployment at ORNL.

3 Data Ingest

The need for DataFed to serve the broader scientific community in a domain-agnostic manner necessitates a tool that can ingest data and metadata while ac-

commodating the high heterogeneity in data generation sources and data types across scientific domains, especially from off-network data producers. Some solutions do indeed exist that purport to solve some of the above data infrastructure challenges [23,18,4]. However, these solutions are typically monolithic in nature and ingest data into a built-in SDMS with limited configurability / features with regards to data storage, data analytics / post-processing, and metadata capture and indexing. Furthermore, these capabilities are implemented using technologies that are not scalable to accommodate the needs of highly heterogeneous and large datasets. Importantly, these solutions result in disjoint silos of data that do not and cannot exchange data elsewhere in the world. Therefore, we are developing a "Data Gateway" to facilitate and streamline data ingest and metadata capture into DataFed.

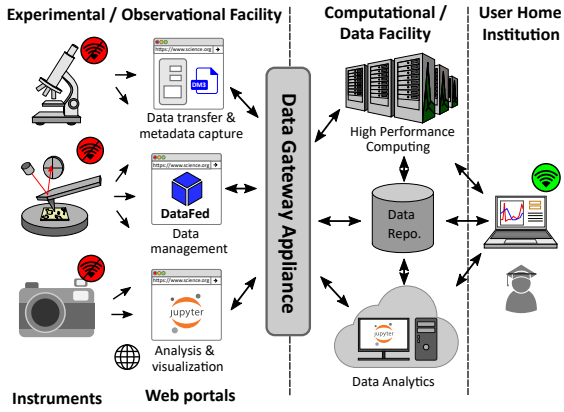

Fig. 3: Overview of the Data Gateway

Often, instrumentation software are incompatible with the latest security patches or operating system updates. Consequently, such instrumentation computers are often kept off the network to avoid security vulnerabilities. Yet, there is a need to capture data and metadata from such instruments. For such instruments, we would deploy a Data Gateway "appliance", as shown in Figure 3, that would consist of both a server (physical hardware) and a software stack (deployed within the server) that provides local data ingest services as well as configurable internet routing to expose remote web services such as the DataFed web portal and an analytics service such as JupyterHub. The Data Gateway consists of a suite of web-based data services pre-installed on a server, with local storage, that would be deployed within a given experiment facility and networked with the facility's individual scientific instrument control workstations. This configuration allows the scientific user at each instrument to access the Data Gateway services while maintaining general network isolation of the instruments (which may be required for IT security purposes). Due to this network isolation, scientists operating scientific instruments cannot directly access data stored in DataFed;

therefore, the Data Gateway acts as a data "buffer" between the instrument control workstation and DataFed, providing temporary data storage for both data uploads and downloads. While data upload is essential for the data ingest processes, data downloads may be needed in order to analyze data using proprietary software that may only be available on an instrument control workstation due to licensing or operating system constraints.

The Data Gateway's data services are configurable for both specific instruments and specific experiments and include data upload/download, metadata capture and extraction, and optional data preprocessing. The Data Gateway provides a graphical web-based "companion application" that can be used from an instrument control workstation while conducting an experiment or measurement - allowing users to easily upload resulting data files and capture associated metadata. Metadata can be captured using configurable input forms or by extracting metadata automatically from data files, or using a combination of both approaches. The API supporting the "companion application" could be exposed to allow instruments to push data and metadata from instruments without the need for humans in the loop.

Users may also opt to utilize available data preprocessing methods, such as file format translations or data reduction, prior to the transfer of the data into DataFed. Such data preprocessing code would be encapsulated in containers to simplify isolation, development and maintenance of the core Data Gateway software stack from the data preprocessing code. Additionally, the use of containers would provide freedom to for domain-scientists to write pre-processing codes in the language and using the software stack they are comfortable with. These metadata extraction and data preprocessing codes would be part of a centralized and vetted library of codes that could be shared across multiple physical and "virtual" Data Gateways. We are in the process of defining standards and an API that would be used for the containers to interact with the Data Gateway. Subsequently, we will start to populate and solicit such codes or references to containers in a public repository at https://github.com/ORNL/MD_Extractors. Additionally, we will provide documentation on the best practices for developing such data preprocessing codes that will lower the barrier for researchers to develop and provide their own codes. Domain scientists would need to develop these codes as they integrate new kinds of simulation codes / instruments with the Data Gateway and update codes only when they need to modify the data processing or account for changes in the simulation code / instrumentation.

For fully networked facilities, full automation of data and metadata capture can be achieved through DataFed's application programming interfaces (APIs) through a one-time integration into existing instrument control systems, data acquisition systems, data pipelines, job scripts, and/or workflows. Data preprocessing and metadata extraction codes from the library mentioned above could be reused optionally. Once this integration effort is complete, users need only authenticate prior to running an experiment or utilizing a resource, and data and metadata will be captured and ingested into DataFed with no further user interaction. Optionally, users may use DataFed to install local security cre-

dentials to avoid the need for subsequent authentication. For large user facilities, direct DataFed integration represents the ideal configuration as all relevant scientific context (instrument configuration, experiment/simulation parameters, run information, etc.) will be automatically captured and raw data will be ingested into DataFed with no additional burden on end-users.

Users outside such facilities that utilize the Data Gateway appliance or direct DataFed integration, such as those running simulations or analytics within a compute facility, can also utilize DataFed through one of two options: 1) users may use the DataFed command-line-interface (CLI) to add DataFed commands to their job scripts, or 2) a Data Gateway can be installed as a "virtual" service within a facility to provide generalized, web-based data ingest services to all users of the facility. Though, much of the software stack developed for the Data Gateway appliance can be readily deployed for "virtual" Data Gateways, users would need to develop metadata extraction and data-preprocessing codes specific to their needs if they are not available in the shared repository of vetted codes.

4 Data Management

A SDMS represents a type of laboratory informatics software for capturing, cataloging, and sharing heterogeneous scientific data. It is common to find products that combine SDMS features with other processing capabilities such as data distribution, workflow management, or even instrument interfacing and control. While there are many available SDMS or SDMS-like products available for use [12,25], these systems are based on older, non-scalable user authentication technologies and tend to be more applicable to the fixed data distribution needs of large-scale, domain-specific research efforts. Thus, there is still a need for scalable and user-friendly data management tools that work across scientific domains and profoundly empower scientists.

An SDMS suitable for use in open, cross-facility, and domain-agnostic scientific research contexts must be able to scale with the volumes and varieties of data being generated from research conducted at large scale experiment, compute, and analytics facilities. It must be able to function across organizational boundaries and efficiently cope with thousands of users, including both resident staff scientists and visiting researchers. It must be able to function within, and across, many different operating environments with varying security policies, ranging from individual scientific instruments to leadership class high-performance computing systems. And, importantly, it must offer simple and uniform interfaces to minimize the need for training and encourage adoption by non-technical users.

Based on these requirements and a lack of an appropriate existing solution, the decision was made to design and develop a new SDMS that would better match the needs of the scientific research community within DOE laboratories. This system is called "DataFed" with the name being derived from the approach of federating data management across existing organizations and facilities to provide flexibility, scalability, and cross-facility data access.

4.1 DataFed Overview

DataFed is a *federated* scientific data management system that differs from existing SDMS products by offering a scalable, cross-facility data management solution with decentralized raw data storage and high performance, secure, and reliable data movement. DataFed is able to scale-out through its ability to incorporate additional organizations/facilities, users, and shared storage resources without the typical burdens and bottlenecks associated with centrally administered systems that rely on virtual organizations (VO) and/or manually deployed user security credentials. Individual users, facilities, or entire organizations may join or leave the DataFed federation at any time without requiring any administrative actions on the part of other federation members. DataFed uses the scalable GridFTP protocol (via Globus[2,3]) for all raw data transfers and supports integration with high performance storage systems and networks. This ensures optimal and reliable handling of very large data files (up to petabyte scale).

DataFed provides a centralized orchestration service that integrates and manages remote raw data storage resources (aka "data repositories") physically housed within member facilities; however, while DataFed manages the raw data files in these repositories, individual facilities own the storage hardware and retain full administrative control over data policies and user/project allocations. DataFed data repositories may be configured to use most types of data storage systems including low-cost commodity disk-backed systems, fast SSD systems, and high-reliability archival storage systems. Facilities may opt to provide more data robustness by implementing periodic back-ups of these storage systems, or by utilizing data replication to prevent data loss from hardware failures. Ideally, facilities would integrate the management of DataFed allocations (assignment, capacity, durability, accounting, etc.) into existing user and project management systems and funding sources. The storage properties and policies of a facility's repositories are visible to users via DataFed, and users can easily migrate data between different facilities, or repositories within a facility, based on availability, locality, reliability, or performance requirements. Because DataFed utilizes Globus federated identity technology for user accounts and fine-grained access control, individual facilities no longer need to manually manage user security credentials or maintain complex and/or constantly changing cross-organizational access control lists.

When data is initially stored in a data repository, DataFed captures and retains any associated metadata and provenance (along with tracking information) in a centralized database. The use of a centralized metadata database does not significantly impact system scalability due to the relatively small storage requirements of metadata (on the order of 10's of kilobytes) when compared to raw data files (ranging from megabytes to terabytes, or more). Access to raw data stored in a DataFed data repository is controlled (managed) by DataFed - not the local storage system. By preventing users or processes from directly accessing or modifying raw files within a repository, DataFed ensures that associated tracking information and metadata remains synchronized with raw data and eliminates potential ambiguity regarding which file should be accessed (a

common problem when using unmanaged file sharing technologies for large collections of data). The central DataFed database would be deployed on a reliable and fast storage system (i.e. RAID) and would be regularly backed-up.

The raw data stored in a data repository is private and secure by default - meaning only the owner, or creator, of the data can access it, and data transfers are encrypted. Data owners may choose to share their data with other DataFed users or groups regardless of organizational affiliation through DataFed's own fine-grained access control system. Specific permissions such as read, write, create, or even administrative control can be granted; Moreover, by using DataFed's hierarchical data organization features, these permissions can be easily granted and managed for large collections of data. DataFed also provides a data project feature to facilitate teams of collaborators working with semi-private or collectively owned data. Due to the need for substantial compliance testing for higher-level data security policies, DataFed currently only supports open research.

DataFed creates a central database data record for each raw data file stored in a data repository in order to track and control access to the raw data and to store and index associated metadata and provenance relationships. A variety of built-in metadata fields are supported for data records (such as title, description, and keywords), but, importantly, domain-specific structured metadata may also be stored with a data record. Retaining and indexing all of this information within a central database enables powerful data organization, discovery, and dissemination capabilities that will be discussed later in this paper. DataFed does not support incremental versioning of metadata or raw data, but provenance-based, full-record versioning is supported by adding "deprecation" dependencies between new and old versions of a record.

4.2 FAIR Compliance

DataFed was designed to be as FAIR compliant as reasonably possible within the context of both pre-publication "working" data and "static" data that is published from DataFed. DataFed specifically addresses FAIR principles as follows:

- **Findable** - DataFed assigns persistent system-unique identifiers to every data record. DataFed also captures and indexes rich metadata that can be used to query for matching records.
- **Accessible** - DataFed identifiers can be used to locate and access associated data, and DataFed enforces authentication and authorization for all access. The protocol for access to data within DataFed is open and easily implementable (implementations are provided for Python and C++).
- **Interoperable** - DataFed utilizes a simple JSON representation for metadata with optional schema support; however, external metadata references are not directly supported.
- **Reusable** - DataFed represents domain-specific metadata and provenance in a uniform manner in addition to facilitating keywords and tags which would allow users to discover and reuse data shared by others for similar or other novel applications.

4.3 Data Organization, Sharing, and Dissemination

While FAIR compliance is an important aspect of DataFed, DataFed includes a number of features that extend beyond the scope of FAIR to more actively assist researchers in complex collaborative contexts. For example, DataFed can significantly assist with the challenges of managing and utilizing large volumes of data within the complex environments associated with high performance computing, cross-facility workflows, and data processing pipelines. In these situations, being able to locate a single data record is less important than being able to stage specific collections or subsets of data for processing within a compute environment. In addition, the ability for an upstream researcher (data producer) to automatically and precisely coordinate with and/or notify downstream collaborators (data consumers) is vital.

DataFed provides named data "collections" which provide a basic form of hierarchical data organization that resembles directories in a file system; however, unlike directories, data is only linked within collections rather than being "owned" by the collection. This allows data to be organized in multiple parallel collection hierarchies, if desired, without duplication of data. Both individual data records and collections can be shared by setting fine-grained permissions for specific users or groups of users. Collections can be assigned a topic and made public, which results in such collections being internally "published" as a DataFed catalog where they can be discovered and accessed by all DataFed users.

As an alternative to collections, DataFed also provides dynamic views of data records based on saved queries. The built-in data search capability allows users to search private, shared, and public data records by identifier, alias, keyword, words and phrases, tags, and arbitrary metadata expressions. For example, a view could be created to show only data records that were most recently created or updated by a collaborator, or records that include specific values or ranges in domain-specific metadata, such as sample type, temperature range, or experiment category.

As an aide in maintaining data awareness, users with appropriate access may opt to subscribe to specific data records and collections such that they will receive notifications whenever certain events occur, such as data or metadata updates, record creation, deprecation, and deletion, or changes in provenance information. If issues arise concerning specific shared data records or collections, users may choose to create linked annotations that will notify and convey additional information, warnings, and/or questions to all concerned parties (i.e. data producers and downstream data consumers via subscription or provenance links). These annotations function similarly to typical document review systems and are preferred over external methods (such as email) as they remain linked and visible on the subject record or collection within DataFed.

5 Data Analytics

Jupyter Notebooks have emerged as a popular framework for data processing and analytics workloads [16]. These notebooks not only contain the code to process information but can also contain rich markdown to provide contextual information such as equations, and provide a rich narrative using static or interactive visualizations in-line with code snippets. Users can add a preamble to the notebook to check and install necessary software or encapsulate the notebook, input data (when data is small) and necessary software stack in software containers [19] to facilitate reproducibility of data analytics workloads. A deployment of JupyterHub [10] would facilitate reproducible data analytics for several researchers. DataFed can further improve the reproducibility of analytics workloads through its ability to address specific datasets, stage multiple datasets (potentially located in multiple repositories) at specific file-systems, and capture the context (analytics algorithm parameters) and results (data) of data analytics runs systematically. Users could also share unpublished / private scripts or notebooks via DataFed.

6 Scientific Applications

The many features of the proposed system substantially alleviate data management burdens and improve scientific productivity. Many of the benefits of the system are shared for all modes of scientific discovery and are discussed below. Common use-cases and benefits specific to each modes of scientific discovery are discussed in dedicated subsections below.

DataFed facilitates capture of metadata and provenance, thereby obviating the need for scientists to embed selected metadata into file paths. Using DataFed, users could perform complex searches for data based on the rich domain-specific metadata over multiple repositories that span multiple facilities or organizations. By standardizing metadata representation, DataFed enables users to find and reuse data owned by themselves, others, or available publicly and also facilitates multi-disciplinary and multi-modal scientific (experiments, observations, simulation, analytics) collaborations. However, note that neither the Data Gateway, nor DataFed mandates the use of specific file formats for the raw data or schemas for metadata.

DataFed's use of Globus allows users to transport data quickly and seamlessly between repositories or facilities without concerning themselves about navigating complex security restrictions or the kind of file-system supporting these repositories. DataFed obviates the use portable storage drives. The barriers to publish data (downloading / uploading data, entering metadata again, repeating the process for multiple datasets) is also substantially mitigated since DataFed can integrate with data publishing services and repositories. Users would only need to switch a setting on the individual record or a large collection from private to published. Similarly, users can also accrue citations by publishing otherwise "dark data".

6.1 Modelling and Simulations

Researchers performing modelling or simulations could incorporate DataFed instructions within their scripts for reliable data staging and capture that:

1. Download input file from DataFed
2. Run modeling / simulation codes
3. Capture metadata
4. Put resulting data and metadata into DataFed repository

In step 1, researchers can use DataFed to unambiguously identify input files or other required files and reliably stage such files at the remote file-systems even if the data records are in repositories located in other institutions. In step 3, researchers can extract metadata from their input scripts and/or the results of the simulations by leveraging the repository of vetted codes for data preprocessing. Once the raw data (from the simulation) and metadata are available, researchers can push this information to a DataFed repository in step 4. Via 1-2 simple commands using the DataFed client, the researchers can create a DataFed record, add the metadata, and push the raw data. Optionally, links to related data records such as input files could be added to capture the complete provenance of the experiment. The same methodology would also accommodate common scenarios where several simulations are run as a function of one or more parameters. Once DataFed commands are integrated into the simulation script, the same / similar commands could be reused for a given type of simulation code.

Through consistent, correct, and careful handling of data, DataFed facilitates traceability and reproducibility of experiments. Once information from simulation runs is captured in DataFed, researchers can search for, share, organize, and move their data. Such consistent collection of data with rich metadata can enable scientists to build large collections of data that would be necessary to train surrogate models using machine learning (ML) or deep learning (DL). These surrogate models could replace expensive kernels of simulations, thereby accelerating the exploration of large and multidimensional parameter spaces.

6.2 Observations and Experiments

Unlike modeling and simulation workflows, the data handling processes for observational sciences are handled almost entirely by the Data Gateway. Researchers working on off-network scientific instruments could use the "companion web application" on the Data Gateway appliance to seamlessly capture the raw data and metadata from experiments and add them to a DataFed repository as experiments are being conducted. Scientific instruments used predominantly for conducting automated and long-running (days, weeks, or months) experiments / observations could instead be configured to automatically and periodically push data and metadata to DataFed repositories via the Data Gateway without the need for a human to manually upload data while at the instrument. This would allow researchers to analyze the data stream collecting in a DataFed

repository while working away from the instrument. Similarly, future iterations of the Data Gateway could potentially facilitate instrument control. The burden for extracting and standardizing metadata when pushing data into DataFed would also be diminished if researchers use the vetted set of codes for automated data-preprocessing at the Data Gateway.

Researchers could search, organize, share, and manage data with their collaborators via DataFed and use a data analytics platform like JupyterHub to analyze data in DataFed repositories even while operating the off-network instruments using the Data Gateway. Clearly, the proposed system dramatically simplifies the processes of capturing metadata, standardizing data formats, and collecting data in readily accessible and well connected data repositories. In addition, the data management capabilities offered by the proposed system are substantially superior to file explorers on personal computers.

6.3 Data Analytics

As discussed above, the proposed system is a conducive platform for researchers from multiple disciplines and working on disparate modes of scientific discovery to collaboratively assemble large collections of richly annotated datasets that are required for ML/DL applications. Similar to modeling and simulation workflows, data analytics applications could benefit immensely by incorporating a few DataFed commands into the scripts or Jupyter notebooks that:

1. Identify data records or collections of interest
2. Get datasets from DataFed repositories
3. Run data analytics application
4. Capture metadata context from analytics
5. Put resulting data and metadata into DataFed repository
6. Establish provenance

In step 1, researchers could optionally use the DataFed's search capability to identify collections and/or datasets of interest for the data analytics application. In step 2, researchers could stage large collections of datasets, that may potentially be spread over multiple institutions in multiple repositories, with a single 'get' command. After performing data analytics, researchers could capture metadata (analytics software version, algorithm identifier, algorithmic parameters, etc.) that are typically available within the data analytics script or notebook in step 4. In step 5, results such a weights for ML / DL models, model inference results, plots, etc. could all be captured as new data records as necessary and enriched with the collected metadata. Finally, the relationship between the results and the source dataset or collection could be captured via the provenance capability in DataFed in step 6. Thus, DataFed can facilitate traceability and reproducibility even in data analytics workflows through comprehensive and unambiguous data handling and management.

7 Conclusions

We presented a system architecture aimed at significantly alleviating the burden of data management, improving scientific productivity, facilitating compliance with FAIR data principles, lowering the barrier to cross-facility and collaborative research, and improving scientific rigor in general. Each component of the system is specifically designed to support the needs of each state of the data lifecycle past data acquisition. DataFed - a general purpose and domain-agnostic SDMS forms the backbone of this system and it is supported by the Data Gateway to capture raw data and context from experiments into DataFed. Optional components include a data analytics platform, such as a JupyterHub server, or other computational workflow software that can work with DataFed, software containers, and the Data Gateway to facilitate reproducible analytics workloads.

The Data Gateway's modular design allows it to be readily deployed for different scientific domains to comprehensively, swiftly, and seamlessly capture data and metadata, especially from off-network instruments, in a consistent, automated and repeatable manner. DataFed provides users with a logical view of data that abstracts routine nuances of data storage and facilitates capture and enrichment of scientific metadata and provenance associated with the raw data. DataFed users benefit from powerful data organization, search, sharing, and discovery capabilities. DataFed enables users to easily, correctly, repeatably, and reliably work with datasets within appropriate compute or analytic contexts to facilitate reproducible research. We are in the process of deploying the broader data management system described in this paper at select facilities at ORNL. We welcome interested readers to use DataFed at https://datafed.ornl.gov and get in touch with the authors for integrating the proposed system with their group / facility.

Acknowledgments

This research used resources of the Oak Ridge Leadership Computing Facility (OLCF) and of the Compute and Data Environment for Science (CADES) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

1. Chris Allan, Jean-Marie Burel, Josh Moore, Colin Blackburn, Melissa Linkert, Scott Loynton, Donald MacDonald, William J Moore, Carlos Neves, Andrew Patterson, et al. Omero: flexible, model-driven data management for experimental biology. *Nature methods*, 9(3):245, 2012.
2. William Allcock. Gridftp: Protocol extensions to ftp for the grid. *http://www. ggf. org/documents/GFD. 20. pdf*, 2003.
3. William Allcock, John Bresnahan, Rajkumar Kettimuthu, Michael Link, Catalin Dumitrescu, Ioan Raicu, and Ian Foster. The globus striped gridftp framework

and server. In *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, page 54. IEEE Computer Society, 2005.
4. Adam P Arkin, Rick L Stevens, Robert W Cottingham, Sergei Maslov, Christopher S Henry, Paramvir Dehal, Doreen Ware, Fernando Perez, Nomi L Harris, Shane Canon, et al. The doe systems biology knowledgebase (kbase). *BioRxiv*, page 096354, 2016.
5. Monya Baker. 1,500 scientists lift the lid on reproducibility. 2016.
6. Monya Baker. Biotech giant posts negative results. *Nature*, 530(7589):141–141, 2016.
7. F. Bartusch, M. Hanussek, J. Krüger, and O. Kohlbacher. Reproducible scientific workflows for high performance and cloud computing. In *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 161–164, 2019.
8. Brett K Beaulieu-Jones and Casey S Greene. Reproducibility of computational workflows is automated using continuous analysis. *Nature biotechnology*, 35(4):342–346, 2017.
9. Justin Blair, Richard S Canon, Jack Deslippe, Abdelilah Essiari, Alexander Hexemer, Alastair A MacDowell, Dilworth Y Parkinson, Simon J Patton, Lavanya Ramakrishnan, Nobumichi Tamura, et al. High performance data management and analysis for tomography. In *Developments in X-Ray Tomography IX*, volume 9212, page 92121G. International Society for Optics and Photonics, 2014.
10. Leandro Fernández, Hakan Hagenrud, Blaz Zupanc, Emanuele Laface, Timo Korhonen, and Riccard Andersson. Jupyterhub at the ess. an interactive python computing environment for scientists and engineers. 2016.
11. Tim Furche, Georg Gottlob, Leonid Libkin, Giorgio Orsi, and Norman W Paton. Data wrangling for big data: Challenges and opportunities. In *EDBT*, volume 16, pages 473–478, 2016.
12. Vincent Garonne, R Vigne, G Stewart, M Barisits, M Lassnig, C Serfon, L Goossens, A Nairz, Atlas Collaboration, et al. Rucio–the next generation of large scale distributed system for atlas data management. In *Journal of Physics: Conference Series*, volume 513, page 042021. IOP Publishing, 2014.
13. P Bryan Heidorn. Shedding light on the dark data in the long tail of science. *Library trends*, 57(2):280–299, 2008.
14. Matthew Hutson. Artificial intelligence faces reproducibility crisis, 2018.
15. Sergei V Kalinin, Evgheni Strelcov, Alex Belianinov, Suhas Somnath, Rama K Vasudevan, Eric J Lingerfelt, Richard K Archibald, Chaomei Chen, Roger Proksch, Nouamane Laanait, et al. Big, deep, and smart data in scanning probe microscopy. *ACS Nano*, pages 9068–9086, 2016.
16. Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90, 2016.
17. Marder, K, Patera, A, Astolfo A, Schneider, M, Weber, B, and Stampanoni, M. Investigating the microvessel architecture of the mouse brain: An approach for measuring, stitching, and analyzing 50 teravoxels of data. In *12th International Conference on Synchrotron Radiation Instrumentation*, page 73. AIP, July 2015.
18. Luigi Marini, Indira Gutierrez-Polo, Rob Kooper, Sandeep Puthanveetil Satheesan, Maxwell Burnette, Jong Lee, Todd Nicholson, Yan Zhao, and Kenton McHenry. Clowder: Open source data management for long tail data. In *Proceedings of the Practice and Experience on Advanced Research Computing*, page 40. ACM, 2018.

19. Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014.
20. Tsuyoshi Miyakawa. No raw data, no science: another possible source of the reproducibility crisis, 2020.
21. Brian A Nosek, George Alter, George C Banks, Denny Borsboom, Sara D Bowman, Steven J Breckler, Stuart Buck, Christopher D Chambers, Gilbert Chin, Garret Christensen, et al. Promoting an open research culture. *Science*, 348(6242):1422–1425, 2015.
22. Line Pouchard, Sterling Baldwin, Todd Elsethagen, Shantenu Jha, Bibi Raju, Eric Stephan, Li Tang, and Kerstin Kleese Van Dam. Computational reproducibility of scientific workflows at extreme scales. *The International Journal of High Performance Computing Applications*, 33(5):763–776, 2019.
23. Catherine Quintero, Kristen Tran, and Alexander A Szewczak. High-throughput quality control of dmso acoustic dispensing using photometric dye methods. *Journal of laboratory automation*, 18(4):296–305, 2013.
24. Paul Raccuglia, Katherine C Elbert, Philip DF Adler, Casey Falk, Malia B Wenny, Aurelio Mollo, Matthias Zeller, Sorelle A Friedler, Joshua Schrier, and Alexander J Norquist. Machine-learning-assisted materials discovery using failed experiments. *Nature*, 533(7601):73–76, 2016.
25. A Rajasekar, R Moore, and F Vernon. irods: A distributed data management cyberinfrastructure for observatories. In *AGU Fall Meeting Abstracts*, 2007.
26. Dale Stansberry, Suhas Somnath, Jessica Breet, Gregory Shutt, and Mallikarjun Shankar. Datafed: Towards reproducible research via federated data management. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1312–1317. IEEE, 2019.
27. Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3, 2016.