Future Generation Computer Systems 94 (2019) 854–867

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Computing environments for reproducibility: Capturing the "Whole Tale"

Adam Brinckman [f], Kyle Chard [b,c,*], Niall Gaffney [d], Mihael Hategan [b,c], Matthew B. Jones [e], Kacper Kowalik [g], Sivakumar Kulasekaran [d], Bertram Ludäscher [a,g], Bryce D. Mecum [e], Jarek Nabrzyski [f], Victoria Stodden [a,g], Ian J. Taylor [f,h], Matthew J. Turk [a,g], Kandace Turner [g]

[a] School of Information Sciences, University of Illinois at Urbana–Champaign, United States
[b] Computation Institute, University of Chicago, United States
[c] Argonne National Laboratory, United States
[d] Texas Advanced Computing Center, University of Texas at Austin, United States
[e] National Center for Ecological Analysis and Synthesis, University of California at Santa Barbara, United States
[f] Center for Research Computing, University of Notre Dame, United States
[g] National Center for Supercomputing Applications, University of Illinois at Urbana–Champaign, United States
[h] School of Computer Science, Cardiff University, Cardiff, UK

HIGHLIGHTS

- Presents an environment that facilitates linkage of data and code with publications.
- Automates the capture of provenance via hosted frontends in controlled environments.
- Focuses on the full spectrum of science, from the long tail to power users.

ARTICLE INFO

Article history:
Received 23 May 2017
Received in revised form 18 September 2017
Accepted 22 December 2017
Available online 10 February 2018

Keywords:
Living publications
Reproducibility
Provenance
Data sharing
Code sharing

ABSTRACT

The act of sharing scientific knowledge is rapidly evolving away from traditional articles and presentations to the delivery of executable objects that integrate the data and computational details (e.g., scripts and workflows) upon which the findings rely. This envisioned coupling of data and process is essential to advancing science but faces technical and institutional barriers. The Whole Tale project aims to address these barriers by connecting computational, data-intensive research efforts with the larger research process—transforming the knowledge discovery and dissemination process into one where data products are united with research articles to create "living publications" or *tales*. The Whole Tale focuses on the full spectrum of science, empowering users in the long tail of science, and power users with demands for access to big data and compute resources. We report here on the design, architecture, and implementation of the Whole Tale environment.

1. Introduction

The pervasive use of computation for scientific discovery has ushered in a new type of scientific research process. Researchers, irrespective of scientific domain, routinely rely on large amounts of data, specialized computational infrastructure, and sophisticated analysis processes from which to test hypotheses and derive results. While scholarly research has evolved significantly over the past decade, the same cannot be said for the methods by which research processes are captured and disseminated. In fact, the primary method for dissemination – the scholarly publication – is largely unchanged since the advent of the scientific journal in the 1660's. This disparity has led many to argue that the scholarly publication is no longer sufficient to verify, reproduce, and extend scientific results [1–6].

The challenges associated with rethinking the scholarly publication model are complicated by the pervasive increase in the collection and analysis of data, coupled with dramatic increases in computational power, and new methods for investigation such as data-driven discovery. The scientific landscape is now littered with a vast array of powerful cyberinfrastructure for acquiring,

* Correspondence to: 5735 S. Ellis Ave, Chicago IL 60637, United States.
E-mail address: chard@uchicago.edu (K. Chard).

https://doi.org/10.1016/j.future.2017.12.029

storing, analyzing, publishing, and archiving data. However, current approaches regarding the dissemination, validation, and verification of computationally based research outcomes do not yet accommodate this reality. Despite the increasing recognition of the need to share all aspects of the research process, scholarly publications today are often *disconnected* from the underlying data and code that produced the findings. While efforts have been made to support data publication [7–9], *"unfortunately, the vast majority of data submitted along with publications are in formats and forms of storage that makes discovery and reuse difficult or impossible"* [10]. Studies of published data have also shown that data availability decays with time, if the data are available at all [11].

To address these challenges we present Whole Tale [12], a research environment that captures and, at the time of publication, exposes salient details of the entire research process via access to persistent versions of the data and code used, provenance, and data lineage (including parameter settings, intermediate, and output data). The Whole Tale directly addresses the transformation of the scientific enterprise to deeply computational research by supporting the entire research pipeline, from pre-publication collaboration, through publication, and to post-publication access and re-use in the broader scientific community. We present here the design and current implementation of the Whole Tale environment (https://dashboard.wholetale.org).

The Whole Tale strengthens the three layers of scholarly publication: scholarly process, data, and computational analysis. Traditionally, the first layer of scholarly publication has been accomplished through the production and dissemination of research articles. As data become more open and transportable, a second layer of research output, linking publications to the associated data, has emerged [2]. This is now followed by the recognition of an important and new third layer: communicating the process of inquiry itself, i.e., a complete *computational narrative*, through the linking and sharing of methods, source code, and data, thereby introducing a new model of reproducible science and accelerated knowledge discovery [13]. The Whole Tale strengthens the second layer (linking data, code, and digital scholarly objects to publications) and also builds a robust third layer that integrates all parts of the research story into a computational *tale* (conveying the holistic experience of reproducible scientific inquiry, i.e., sharing the source code, data, and methods, along with the computational environment in which inquiry is conducted) and making both layers accessible from the scholarly publication. To the user it thus appears that *by sharing a paper as a tale, the narrative is shared together with an on-demand, virtual computer that is preloaded with all the relevant data, methods, software packages, and analysis frontends needed to reproduce, tinker with, or even extend the paper.*

The Whole Tale environment can also be seen as a form of *science gateway* [14]: it simplifies access to a vast array of cyberinfrastructure for a broad range of domain scientists. The architecture described herein builds upon advances made within the science gateways community to leverage external services for core functionality such as user authentication and authorization, data management, and management of computational resources. Finally, the Whole Tale architecture is based on extensible APIs that can be leveraged by other gateways for recording computational processes, importing and managing data, issuing identifiers, and sharing and publishing reproducible tales.

The remainder of this article is structured as follows. In Section 2 we first present examples from three scientific domains that highlight some of the challenges commonly faced by computational scientists. In Section 3 we describe prior and current efforts toward enabling reproducible research. We then describe high level requirements of the Whole Tale in Section 4 before presenting the architecture and current implementation in Section 5. In Section 6 we review related work. Finally, we summarize our contributions in Section 7.

2. Science narratives

We first describe three scientific domains that, like many others, have embraced computational and data-driven science. We focus specifically on examples that elucidate usage requirements for the Whole Tale.

2.1. Materials science

Materials scientists are now generating vast amounts of computational and experimental data from a wide set of user facilities (e.g., the APS, SNS, NSLS-II), from simulations at Leadership Computing Facilities, from individual research labs, and from high-throughput experiments. To address the deluge of high quality data there are now numerous data repositories designed to store and provide access to curated materials data including: the Materials Data Facility (MDF) [15], Materials Project [16], Citrination [17], and NoMaD (Novel Materials Design) [18]. With these rich materials data sources, opportunities are available to conduct new types of analysis and for researchers to supplement their own data to expand investigations.

Computational approaches are having a profound effect on materials science. Over the past several decades, concurrent advancements in physics-based simulation methods and computing power have made it possible to model material behavior on a large range of length and time scales [19]. Consequently, computational tools are starting to become an integral part of designing new materials [20]. For example, quantum-mechanics-based calculation tools are routinely used in the development of structural metals and semiconductors, among other materials [21]. Increasingly, these computational processes are based on new machine learning methods to construct rich models of materials properties [22–24]. With such changes, researchers are increasingly in need of new methods for publishing not only references to the data used to derive results but also the models and computational processes that underpin results.

As one example of these changes we describe the process undertaken by Ward et al. to design new metallic glasses using machine learning models [23]. The authors first assembled a collection of materials data [25]. In order to build a machine learning model, they used custom software to transform the raw data, text strings describing each material's composition and properties, to a form compatible with their models: finite length vectors of physically-meaningful inputs. They trained machine learning models using Weka [26] and employed the models to scan over several million compositions to identify novel glass-forming alloys. In an effort to make their methods verifiable and reproducible the authors published their workflows (as text input and data files) as supplementary information to the paper. Using Whole Tale these researchers could streamline this process via access to a large and varied amount of data, a platform for conducting their analyses using containerized frontends, and the ability to subsequently publish their entire method (including data and analyses) with a persistent identifier. Readers of their manuscript could then view their methods (as a tale) and reproduce the exact steps taken within the Whole Tale environment.

2.2. Astronomy

Detailed analysis and visualization of astronomical datasets, particularly those generated from computational simulations, requires both access to the original underlying data (or catalogs of reduced data products) and access to computing resources. One particular example being explored by the Whole Tale project is that of studying the formation of the first stars and galaxies in the universe (see, for instance, [27]), but another common use

case is that of galaxy formation [28]. These simulations are conducted on large-scale computing resources; typically, the analysis utilizes community packages such as `yt` [29] and, through the development of scripts and interactive analysis sessions, produces either publication-level plots or reduced data products that can be reanalyzed at a later date. In the case of observational astronomy, multiple datasets may need to be synthesized to create a unified understanding of either a particular class of object or a region on the sky; in many cases, this will require small "slices" of data from many different sources (e.g., from several public registries) to be combined.

For the specific case of analysis and visualization of simulations, Whole Tale will provide access to a collaborative environment, where scripts and analysis methods can not only be transplanted seamlessly between datasets, but where they can be collaborated on between individuals— such as an advisor and a student. Researchers will be able to conduct simulations, make available the results of those simulations inside the Whole Tale environment, and then conduct their analysis in that environment directly. The scripts that produce plots and analysis products for publication purposes will be combined with these datasets to form a tale, which can then be accessed, remixed, and modified for subsequent analysis either by those researchers or by others. This will open up new avenues for discovery, which at present is constrained both by the difficulty of providing access to data and by the difficulties inherent in collaborating on the specific methods of analysis and visualization.

2.3. Archaeology

A set of grand challenges in archaeology have been identified by the community through a crowd-sourced effort and synthesis workshop [30]. While archaeological data and research are essential to addressing fundamental questions, e.g., about the origin and trajectories of civilizations or the response of societies to climate change, the community lacks the capacity for acquiring, managing, analyzing, and synthesizing datasets needed to address such important questions. This in turn led to recommendations for computational infrastructure, tools, and scientific case studies to demonstrate archaeology's ability to contribute to transdisciplinary research on long-term social dynamics [31].

One such project is Synthesizing Knowledge of Past Environments (SKOPE) [32], which is developing an online resource and toolkit for paleoenvironmental data and models that will enable researchers to easily discover, explore, visualize, and synthesize knowledge of environmental factors most relevant to humans in the past. SKOPE's focus on transparent, reproducible research, facilitated by different forms of provenance, makes it an ideal partner project and science driver for Whole Tale. To address the complex, multi-stage workflows inherent in this domain, these researchers will employ YesWorkflow [33,34] to create graphical, queryable representations of each of the computational workflows enacted as part of the research. These workflows capture the *prospective* provenance of all data products generated during the study. Such workflows can be used within the Whole Tale environment to create hybrid forms of provenance [35,36] that combine prospective with retrospective provenance information of intermediate and final data products, complete with records of the specific program executions involved, the values of program arguments applied, and – where possible – the values of key variables within the programs themselves as exposed by YesWorkflow.

Finally, there are several community efforts such as "How To Do Archaeological Science Using R" [37] that aim to improve community practice: i.e., instead of sharing methods only via traditional publications, reproducibility and reuse are facilitated by authors communicating their methods also via open code repositories and using tools to package computational narratives as research compendia for R [38,39]. These efforts provide current, real-world science use cases that Whole Tale aims to support and enhance.

3. Toward reproducibility

As we noted at the outset, the complexity and details of the computational steps that gave rise to the scientific conclusions is typically impossible to capture in a traditional publication [40]. However, increasing the transparency of computational findings falls on several stakeholders. As researchers move toward greater reproducibility it is essential that funding agencies, publishers, and local incentives align to support this transition. Steps are being taken at all stakeholder levels, yet open questions remain.

Researchers were the first to implement new practices that encompassed reproducibility in computational research. The earliest to our knowledge was the introduction of "really reproducible research" in 1992 by the Stanford Exploration Project [41] which introduced reproducibility standards for electronic documents that contain computational results. Most recently [42] exhorted the community to ensure that the digital artifacts needed for verification (i.e. data, code, workflows) are made available to the community in a usable form with the publication. The recommendations were:

1. To facilitate reproducibility, share the data, software, workflows, and details of the computational environment in open trusted repositories.
2. To enable discoverability, persistent links should appear in the published article and include a permanent identifier for data, code, and digital artifacts upon which the results depend.
3. To enable credit for shared digital scholarly objects, citation should be standard practice [43,44].
4. To facilitate reuse, adequately document digital scholarly artifacts.
5. Journals should conduct a Reproducibility Check as part of the publication process and enact the Transparency and Openness Promotion (TOP) Standards at level 2 or 3 [45].
6. Use Open Licensing when publishing digital scholarly objects [46,47].
7. To better enable reproducibility across the scientific enterprise, funding agencies should instigate new research programs and pilot studies.

To date, more than 5000 journals have signed on to the TOP Guidelines [46,48]. Journals are progressively taking steps to encourage the submission and publication of reproducible computational research [49].

Funding agencies are also moving toward implementing reproducible research. The National Science Foundation (NSF) requires the disclosure of data and software created in the course of research they fund. The NSF Award & Administration Guide (AAG) Chapter VI.D.4. (October 2016) reads:

- b. Investigators are expected to share with other researchers, at no more than incremental cost and within a reasonable time, the primary data, samples, physical collections and other supporting materials created or gathered in the course of work under NSF grants. Grantees are expected to encourage and facilitate such sharing. [...]
- c. Investigators and grantees are encouraged to share software and inventions created under the grant or otherwise make them or their products widely available and usable.

The NSF held an agency-wide Director's Symposium "Robust and Reliable Science: The Path Forward" on September 10, 2015. More recently, on February 25–26, 2017, the NSF's Directorate on Mathematical and Physical Sciences held a workshop "Systematic Approaches to Robustness, Reliability, and Reproducibility in Scientific Research" fomenting a discussion around reproducibility [50]. In December of 2016 the Advisory Committee to the

Computer and Information Science and Engineering Directorate at NSF released a report "Realizing the Potential of Data Science" [51] which included recommendations on reproducibility:

- Recommendation 2: Invest in research into data science infrastructure that furthers effective data sharing, data use, and life cycle management: ... Research outcomes should ultimately be translatable to infrastructure that enables access to data in ways that: ... (iii) support reproducibility; (iv) support access, provenance, sustainability, and other life cycle challenges.
- Recommendation 3: Support research into effective reproducibility: Develop research programs that support computational reproducibility and computationally-enabled discovery, as well as cyberinfrastructure that supports reproducibility.

Scientific societies, in part in their role as publishers, are also taking steps toward reproducibility. The ACM has implemented a system of badging for publications that have digital artifacts available [52]. In November of 2016 IEEE held a workshop on publication practices for reproducibility, "The Future of Research Curation and Research Reproducibility" [53].

Finally, the National Academies of Sciences, Engineering, and Medicine, released a report in April 2017, Fostering Integrity in Research [54], which contained two recommendations regarding reproducible research:

- Recommendation 6: Through their policies and through the development of supporting infrastructure, research sponsors and science, engineering, technology, and medical journal and book publishers should ensure that information sufficient for a person knowledgeable about the field and its techniques to reproduce reported results is made available at the time of publication or as soon as possible after publication.
- Recommendation 7: Federal funding agencies and other research sponsors should allocate sufficient funds to enable the long-term storage, archiving, and access of datasets and code necessary for the replication of published findings.

There have been concurrent advances in European open access and open data policy. In 2003 the Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities was signed by nearly 300 stakeholder groups including research and educational institutions, libraries, museums, funding agencies, and governments from around the world to help establish the Internet as the primary medium of communication and dissemination of scientific knowledge [55]. EUDAT [56], a European-based effort to share and preserve data across international borders and across research disciplines was started in 2012 and continues actively today. OpenAIRE [57] is a European repository effort to, in part, link data to publications and was started in 2009. On the infrastructure side, EuroCloud [58] was launched in 2010 in part to support cloud based research and innovation in Europe.

The 2017 version of the European Code of Conduct for Research Integrity explicitly mentions data integrity [59]. Their list of "Good Research Practices" includes:

- Research institutions and organizations support proper infrastructure for the management and protection of data and research materials in all their forms (encompassing qualitative and quantitative data, protocols, processes, other research artifacts and associated metadata) that are necessary for reproducibility, traceability and accountability.

A Dagstuhl seminar on *Reproducibility of Data-Oriented Experiments* summarizes its findings as follows [60]:

- Transparency, openness, and reproducibility are vital features of science. Scientists embrace these features as disciplinary norms and values, and it follows that they should be integrated into daily research activities. These practices give confidence in the work; help research as a whole to be conducted at a higher standard and be undertaken more efficiently; provide verifiability and falsifiability; and encourage a community of mutual cooperation. They also lead to a valuable form of paper, namely, reports on evaluation and reproduction of prior work. Outcomes that others can build upon and use for their own research, whether a theoretical construct or a reproducible experimental result, form a foundation on which science can progress. Papers that are structured and presented in a manner that facilitates and encourages such post-publication evaluations benefit from increased impact, recognition, and citation rates. Experience in computing research has demonstrated that a range of straightforward mechanisms can be employed to encourage authors to produce reproducible work. These include: requiring an explicit commitment to an intended level of provision of reproducible materials as a routine part of each paper's structure; requiring a detailed methods section; separating the refereeing of the paper's scientific contribution and its technical process; and explicitly encouraging the creation and reuse of open resources (data, or code, or both).

As noted in several of the recommendations discussed above, new research and new technologies are needed to implement reproducible computational research, and the Whole Tale represents one initiative to address these gaps in our research and dissemination infrastructure.

4. Design requirements

The Whole Tale project is intended to support the lifecycle of data. This means that all parts of the lifecycle, from data ingest or creation through to publication of the resulting scholarly objects such as data, code, workflows, and manuscripts, should be managed within the Whole Tale environment. Our discussion of design and implementation therefore reflects an integrated view of the generation of computational scientific findings that includes all these research activities. This integrated approach to research is crucial to enable reproducibility and downstream re-use of scholarly objects.

To provide such support, Whole Tale incorporates data ingestion, identity management, data publication, and the deployment of user-facing "frontends." We use the term *frontend* to describe any environment in which data can be operated on, ranging from terminals with a command-line interface to specialized analysis programs. Examples of common frontends include interactive notebooks (e.g., Jupyter and RStudio), HTML5 web apps, and domain-specific GUIs (e.g., OpenRefine). We briefly describe the requirements in several core areas to motivate the architecture presented in the following section.

4.1. Data ingestion

Researchers now have access to an enormous amount of data from sources such as data repositories, instruments, and local storage. Researchers who want to act upon data, for example to test a hypothesis or reproduce a result, must first discover and then obtain access to data distributed across many possible locations. The Whole Tale environment aims to reduce these barriers by providing mechanisms by which researchers can ingest data from a wide variety of sources. We focus initially on four commonly used data sources:

- **Data Repositories:** There is an increasing number of domain-specific, institutional, project-centric, and publisher-owned data repositories. Many data repositories, including the Materials Data Facility (MDF) and DataONE [61] (a federation of repositories), support common interfaces for accessing published metadata and data. These interfaces include the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [62] as well as many custom REST interfaces.
- **Storage systems:** Research data is distributed across a range of local systems, from instruments to archival storage. Each storage system implements one of many interfaces for accessing that data (e.g., object storage, tape interfaces, cloud storage, high performance file systems, etc.)
- **Web accessible data:** There is a vast amount of data stored on web pages or web-based data repositories. In these cases, data can be discovered and downloaded using HTTP-based tools.
- **Local data:** Much research data exists on researchers' personal computers, shared clusters, or otherwise inaccessible (in terms of a common API) devices.

4.2. Analysis frontends

As mentioned above, we use the term *frontend* to describe any environment in which data is operated on, ranging from command-line terminals to specialized analysis programs. The choice of frontend used for a specific scientific analysis may be based on analysis requirements, data type, or user preference. One example frontend that is commonly used by researchers is the Jupyter notebook environment [63]. Jupyter notebooks support multiple language backends (Python, R, Julia, and many others), widget development for interactive exploration, file editing, and shell activity within a unified, web-based environment.

To address the needs of a wide range of users and use cases, the Whole Tale must support an extensible set of frontends. Users coming to Whole Tale should be able to search available frontends by the types of data they can support, the user interface offered (web, command-line, digital notebook, etc.), and which user provided them. Having discovered a frontend, users should then be able to rapidly deploy them on-demand and access data directly from within the frontend. The Whole Tale environment must manage the execution of a frontend while also capturing the steps followed by a user such that the entire frontend can be packaged, published, and shared with others.

4.3. Persistent identification

One of the primary goals of the Whole Tale is to enable publication and identification of scholarly objects, where the term scholarly object is used to describe not only a traditional publication but also data and computational processes. A flexible identification and resolution service is required to allow persistent identifiers (e.g., DOIs, ARKs, Handles) to be associated with these different objects. Furthermore, models are needed to allow researchers to organize their objects in different ways, both for their own purposes and also to simplify collaboration and discovery. As such, the Whole Tale must provide a way for researchers to organize their scholarly objects.

4.4. Authentication and authorization

The Whole Tale aims not to reinvent existing capabilities but rather to interoperate with existing services and cyberinfrastructure providers (e.g., repositories, compute environments, libraries). Each of these existing providers might be managed independently with proprietary identitity and authorization models. It is therefore important that Whole Tale adhere to existing authentication models and ensure that digital artifacts accessed and created during the exploration and publication of scholarly objects are correctly authorized.

When designing the authentication model it is desirable that researchers are able to sign in once, access a range of supported services, and have their identity and permissions be used securely across services. Given the rate of identity proliferation, the authentication system should allow researchers to authenticate with their preferred identity (e.g., campus identity, ORCID, Google account) and control authorization at a fine grained level (e.g., revoking access when needed). Rather than restrict the identities used in the system, we instead associate identities with actions and artifacts, and allow other users to determine trust based on knowledge of the identity used. To enable extensibility to new tools and services, Whole Tale must support standard authentication and authorization protocols through which external services and clients can easily integrate with the system. The Whole Tale focuses on acting upon research data, we therefore consider issues related to sensitive data (e.g., restricted medical or government data) outside the scope of this work.

4.5. Reproducibility: Defining a tale

The final, and perhaps most important, aim of the Whole Tale is to define a model for reproducibility by capturing the data, methods, metadata, and provenance of a particular research activity within the system. We refer to this entity as a *tale*. As has been observed time and again, successful adoption of new models is often related to the ease by which they can be used. As such, it is crucial that capturing, publishing, and replaying a tale is simple and unobtrusive: the relevant provenance of an analysis should be transparently recorded without requiring users to manage or record the data and computational process used in their work. Having created a tale, researchers should be able to simply share them with others, publish them to connected repositories, associate a persistent identifier, and link them to publications. Other researchers who access a tale should, just as simply, be able to instantiate a version of the tale and execute it in the same state as it was when published. Tales also contain Intellectual Property metadata with licensing information for its components (data, scripts, workflow information, etc.), which is crucial to enabling ease of re-use and reproducibility, as well as broad re-use, reproducibility, and broad access.

5. Architecture and implementation

The Whole Tale architecture uses a range of flexible APIs to enable users to ingest and manage data, manage frontends, and capture, replay, and extend tales. The general architecture of the platform is shown in Fig. 1. Our development philosophy follows open source principles to be consistent with our goals of research transparency, but more importantly to enable the re-use and extension of the project and encourage a community to grow around the Whole Tale, see https://github.com/whole-tale.

5.1. General architecture

At the heart of the Whole Tale infrastructure lies the Metadata Management System, which creates an abstraction layer between user-facing interfaces and the physical location of the data. For this purpose we utilize Girder [64]—a general purpose framework with a simple data model and REST interface. Using Girder, datasets can be organized into *Collections*, containing *Folders* and *Items*. *Folders*

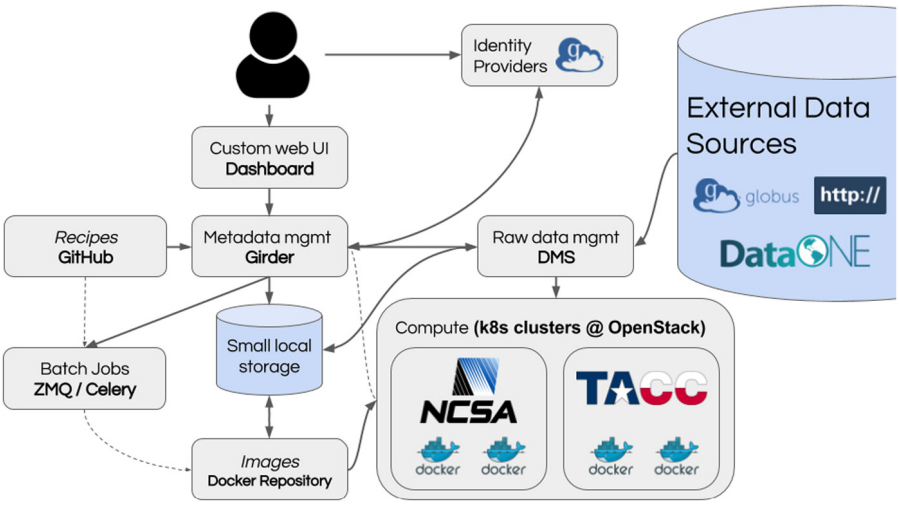

Fig. 1. Whole Tale architecture. Users connect via common web interfaces. Microservices manage data, users, frontends, and tales. The execution environment manages the execution of frontends in containers on different compute resources.

are a hierarchically nested organizational structure that consist of other *Folders* and *Items*. An *Item* is the basic unit of data in the system. *Items* live beneath *Folders* and contain zero or more *Files*, which represent raw data objects. Each organizational object, i.e., *Collection*, *Folder* and *Item* can be annotated with metadata. Additionally Girder provides models for user and group management (*Users*, *Groups*) and an access control model for resource management. Each of these objects is represented by a model with a RESTful interface, that can be used to create, store, and retrieve persistent records in an internal MongoDB. As a result, data is managed entirely by reference. That is, the data stored in external data repositories are completely decoupled from the data in Whole Tale, e.g., an *Item* representing an external object (e.g., an HTTP URL, or a file on a Globus endpoint) can be easily copied, renamed, and moved around Girder's *Folder* structure without performing any operations on the actual data. This approach is advantageous as it allows Whole Tale to scale to represent very large datasets by outsourcing data management tasks to external systems (e.g., Globus) and only copying the data when needed.

The Whole Tale builds upon Girder by providing plugins that:

- Introduce new models for objects specific to the project, such as *Recipe*, *Image*, *Tale*, *Instance*, *Repository* (see Section 5.2 for details).
- Allow users to execute tasks such as building *Images* from *Recipes* and creating *Instances* from *Tales*.
- Manage transfers of streams of data from remote *Repositories* into running *Instances* (see Section 5.5 for details).

5.2. The Whole Tale workflow

As mentioned in Section 4.1, Whole Tale's functionality includes the reuse of published scientific data. Users may **register** data located in an external repository which Whole Tale understands as native Girder objects, such as *Folders* and *Items*. Registration of data, say in preparation for conducting research in Whole Tale, is a two-step process. First, users provide a data identifier (e.g., DOI, data provider specific UUID, URL), which is passed to the external search engine of each supported data provider. Basic information about the dataset is obtained (e.g., name, size, provider; see Fig. 2). To obtain this information we define a new *Repository* endpoint[1] in Girder, which abstracts access to repository-specific interfaces. The registration procedure starts with the creation of a *Folder* object to group all the references related to the selected dataset (datasets may be comprised of many files and folders). For each of the files provided by the data provider an *Item* is created as a child object in the main *Folder*. Each *Item* stores the information about the original name of the file, its location, and the protocol to access it (e.g., HTTP, Globus, etc.).

In some cases datasets include references to other datasets. In this case, we create a sub-*Folder* for each reference and the procedure continues recursively. As this may be a time consuming process, registration occurs asynchronously and users are notified of progress. Once the registration is complete, users are allowed to modify the resulting data hierarchy, i.e. rename *Items*, move *Folders* etc. However, these modifications do not affect the provenance attributes of those objects (e.g., source repository, location, name, etc.). It is important to note that when data is registered from a remote source the system will provide a shallow copy of that entire dataset. As the user interacts with the imported dataset (e.g., in a tale), the raw data is copied on-the fly and cached thereafter to create a deep copy of the data. At present the Whole Tale supports repository access via DataONE and Globus [65] repositories. Additional repositories can be easily added by implementing a simple interface that provides the necessary information: name, size, location and access protocol; and embedding it within the *Repository* model.

The availability of the data and the fact that it can be freely composed into a dynamic dataset through the *Folder* and *Item* hierarchy, is a necessary ingredient of the most important artifact that comes out of the Whole Tale project, which is the *tale* itself. A tale **bundles a frontend and relevant data into a research environment**. The environment itself is based on a Docker image—a lightweight, stand-alone, executable package that includes everything needed to run a research environment for a tale. In order to ensure that the image can be reconstructed in exactly the same state we require a machine parseable description of all runtime dependencies. For this purpose we use a Dockerfile as a *Recipe* for constructing the environment (i.e., Docker image). For reproducibility purposes we treat each modification to a given Dockerfile as a prescription for a different frontend. We store these recipes using a combination of a Git repository alongside a log of changes and represent this object as a *Recipe* in Whole Tale.

Depending on the complexity of an image the process of building it can be lengthy and resource consuming. We utilize a Distributed Task Queue (Celery/ZMQ) that is integrated with Girder,

[1] https://github.com/whole-tale/girder_ythub.

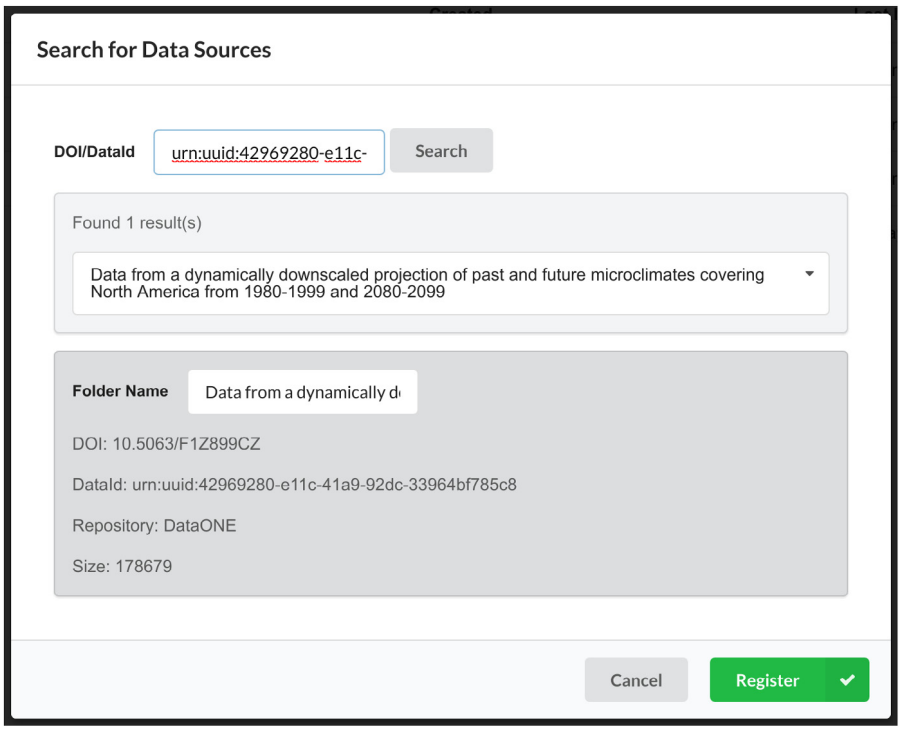

Fig. 2. Whole Tale data registration modal. Users provide the data unique identifier and register the external resources as native girder objects.

to asynchronously build, track status, and deposit Docker images in a local instance of a Docker registry—a server application that stores and distributes Docker images. The state of a given image is represented within the metadata management system through the *Image* model. Once a docker image is successfully built and deposited in the Docker registry, it becomes accessible on registered Whole Tale compute resources. At that point it is available to the user as an executable research environment. The *Image*, which represents the frontend, and the *Folder*, that represents the data, can then be at this point combined into a *tale* (see Fig. 3).

5.3. Web interface

The Whole Tale is designed to be easy to use and accessible to a wide range of users. Its primary interface is a web-based application that allows users to manage data; create, modify, and share frontends for analyzing data; and create, publish, and reproduce tales by linking together datasets and frontends.

The web interface supports the standard set of file and folder operations as one would expect in a desktop finder or file manager application (rename, remove, move, delete, etc.). Files or datasets can be registered from external data repositories (via a search workflow) or dragged and dropped from a user's desktop into the environment. Users can also view their registered files, as well as public datasets that may have been registered by other users.

The Whole Tale web interface[2] (shown in Fig. 4) is implemented using the Ember.js open-source JavaScript web framework [66]. Ember is based on the *Model View View Model* (MVVM) pattern, enabling developers to create single page applications (SPAs). Ember also provides front end data models, which provide seamless access to Web APIs. The Whole Tale interface is implemented using the Semantic UI development framework [67].

5.4. Authentication and authorization

We base the Whole Tale authentication and authorization model on Globus Auth [68]—a platform for identity and access management. By leveraging Globus Auth, we essentially outsource core authentication functionality to a highly reliable service provider and need not implement our own user management functionality (e.g., password management, user creation workflows, etc.)

Globus Auth provides a number of desirable properties for the Whole Tale. First, it allows researchers to authenticate using a range of identities, including those common in academia (e.g., campus credentials and ORCID). It also allows researchers to link together different identities such that presentation of one identity applies permissions granted to any identity in that set. Second, it supports standard web authentication and authorization protocols (e.g., OpenID Connect and OAuth 2) that simplify integration in Whole Tale services and also provides an extensible model by which other related services can leverage Whole Tale capabilities. Third, it provides an extensible delegated authorization model by which services (e.g., Whole Tale) can obtain delegated tokens to access other services (e.g., data repositories) on behalf of users. Conversely, the model also allows external services (e.g., publishers) to obtain tokens to access Whole Tale services on behalf of users.

We have implemented support for Globus Auth by extending Girder's OAuth plugin. This integration allows users to authenticate with Whole Tale using any of the supported identity providers. Whole Tale is configured to request access ("scopes") to various resources on behalf of users including their profile and linked identities, as well as being able to access other services including Globus transfer and MDF.

5.5. Data management

The Whole Tale Data Management System (DMS)[3] is responsible for managing the "bits" that make up the data used in tales.

[2] https://github.com/whole-tale/dashboard.

[3] https://github.com/whole-tale/girder_wt_data_manager/.

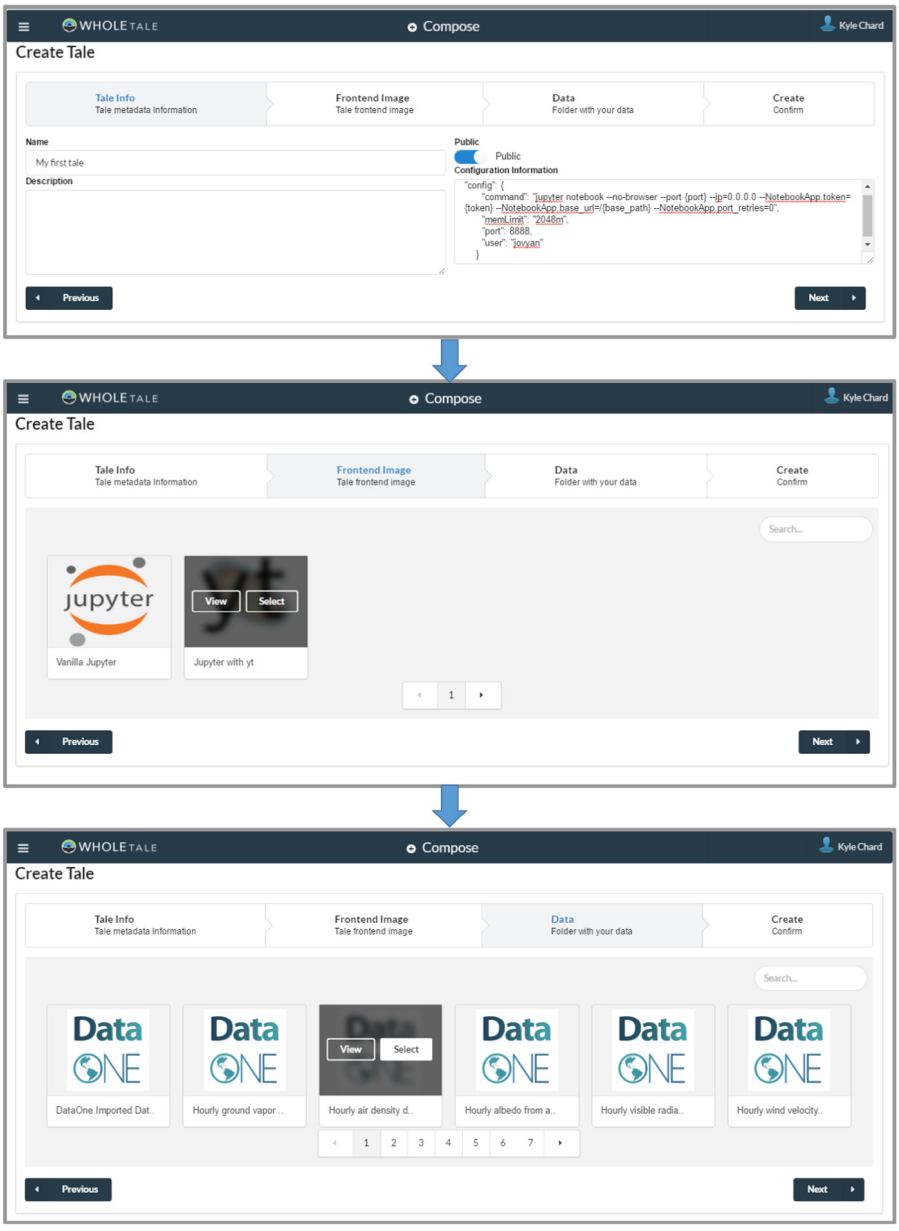

Fig. 3. Creating a tale via selection of a frontend and a folder containing data.

Primary data, which is data that is sourced from external services, does not, in general, come with a uniform access mechanism. Each external service is free to define its own rules and mechanisms of access. The DMS addresses this issue by providing a POSIX interface to primary data. This interface allows tales to act upon diverse, distributed data as if the data were local. A secondary goal of the DMS is to provide data locality. Primary data services are assumed to be geographically distributed, as such, there is significant latency when data is accessed directly. The DMS provides an abstraction layer that hides these differences. The main components of the DMS are:

- The **transfer subsystem** manages the movement of data from external data providers to a storage area local to the Whole Tale infrastructure. It does so through the use of transfer adapters, which are specific to each external provider.
- The **storage management system** controls the use of storage space by evicting data that is likely to be used infrequently. It acts as a data cache for external data.
- The **filesystem interface** allows tales to access cached external data through a POSIX interface.

From a user perspective, the process of consciously interacting with the DMS is limited to composing filesystem hierarchies to be used in tales. An initial process of ingestion described in Section 5.2 populates the Whole Tale backend with metadata about available external data collections. Many such collections are available and navigating them in a tale, through a filesystem interface, can be difficult. Users are, therefore, able to manage and organize data through the web interface (see Fig. 5) and to construct specialized subsets of the data that are accessible to a user or known to Whole Tale. These specialized subsets are termed "sessions." Each tale is associated with a session. This association is seen by the user as a filesystem that contains the data items composing the session. The filesystem is implemented as a FUSE [69] layer. The filesystem

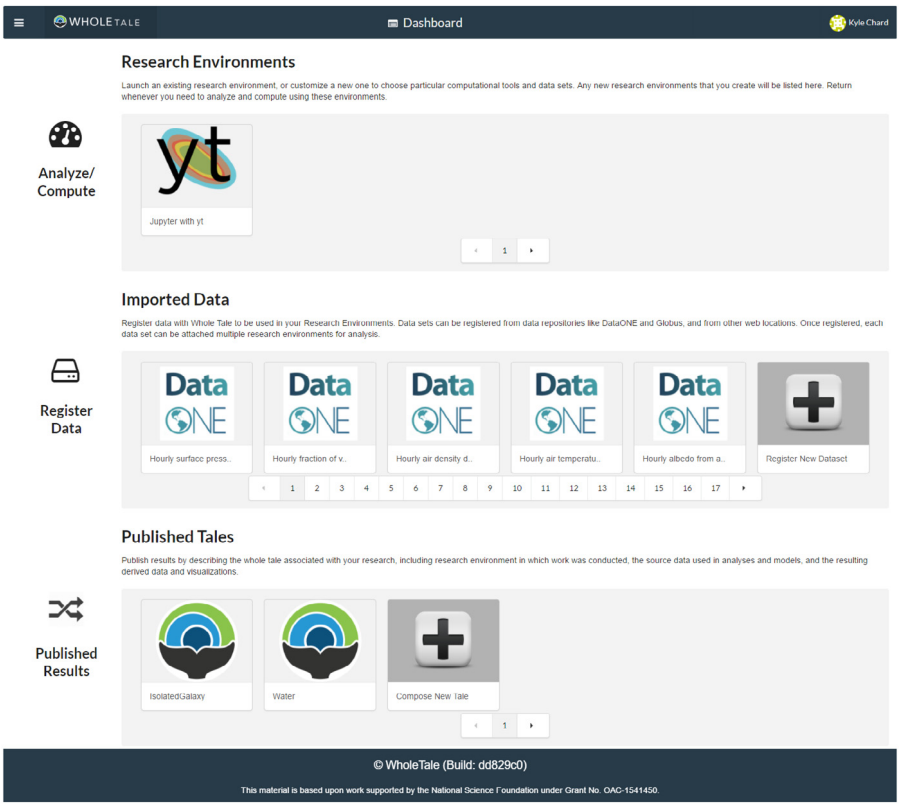

Fig. 4. Whole Tale front page. Showing accessible frontends, datasets, and tales. Users can select an existing entity or create a new one.

is currently stored using OpenStack's Cinder block storage. Direct access to files on this filesystem results in data transfers from external data sources, unless the data already exists locally. A locking mechanism ensures that data corresponding to files that are in active use by a tale cannot be removed to reclaim storage space.

Maintaining local copies of all external data available to Whole Tale is not feasible. Consequently, the storage management system acts as a cache and garbage collector, periodically traversing the local storage and purging data in a way that meets storage constraints as well as optimizing the latency of data access for tales. The exact optimization mechanism is flexible and involves sorting data based on an objective function that is calculated based on metadata generated by the DMS, such as usage count, usage frequency, and time of last access.

5.6. Tale execution and management

Once a *tale* has been created it can be executed (see Fig. 6). The only requirement for execution on a specific compute resource is the availability of a Docker Engine and two lightweight helper daemons: a reverse proxy that is responsible for routing all traffic in and out of a running tale instance (e.g., configurable-http-proxy or NGINX), and a tale management daemon (TMD)[4] that is responsible for managing the tale and its data dependencies. An instantiation of the tale is a multi-step process:

1. A request for a *tale* instance is sent from the Metadata Management System (MMS) to the TMD running on a computational cluster, along with credentials (access token).
2. The TMD creates a docker volume using the "local" driver, which is basically an empty POSIX directory inside the host's filesystem.
3. The TMD creates a docker instance using an *Image* referenced by the *tale* and the volume created in the previous step.
4. The TMD creates the FUSE layer using the *Folder* referenced by the *tale* and mounts it in the mountpoint corresponding to the docker volume.
5. The TMD starts the docker container and registers the internal port by which the container can be accessed with the reverse proxy.
6. The TMD returns basic information about the container (routing path, container id, host where it is running, etc.) to the MMS.
7. The MMS creates an *Instance* model to store the information provided by the TMD and exposes it to the web interface.

The *Instance* object created during the tale instantiation is a regular RESTful object. It allows the UI to query information about running tales, and to update, suspend, or delete them.

Tales can host *any* frontend as they are based on a generic Docker container, the only requirements of which are an open port for user access and a mountpoint for the Whole Tale FUSE filesystem. At present, pre-configured Jupyter and RStudio frontends are provided. These types of frontends were prioritized based user needs and popularity. While users can create their own frontends, we will continue to add base frontends to simplify use.

Whole Tale containers are executed on OpenStack virtual machines (running Container Linux). On each virtual machine we deploy Docker Swarm to manage the execution and scheduling of Docker containers on our resource cluster. We use cloud resources at the National Center for Supercomputing Applications (NCSA), Texas Advanced Computing Center (TACC), and San Diego

[4] https://github.com/whole-tale/girder_volman/.

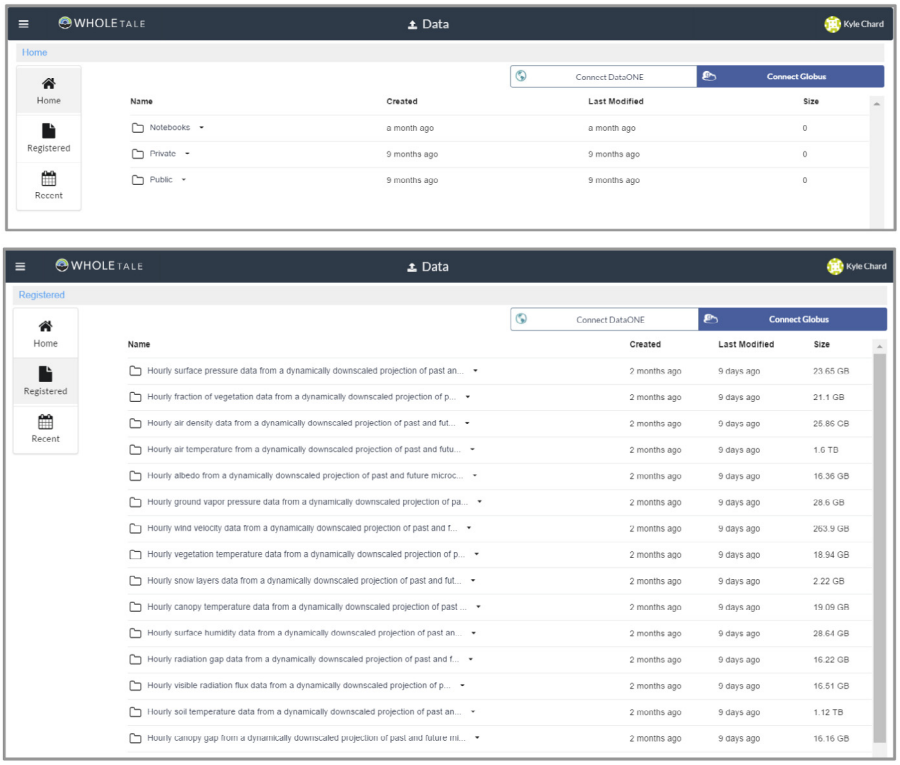

Fig. 5. Data management interface.

Supercomputer Center (SDSC). There are well-established security risks of running containers on shared infrastructure. To mitigate these risks we require that users upload source images to be built on-demand and we disable intra-container communication to limit possible interference between containers. In future work we intend to investigate methods for validating and certifying containers.

5.7. Tale representation

Tales are defined by their execution environment, the data used, and metadata related to that tale. The key elements of a tale are as follows:

The **environment** captures the active components of a tale. For this purpose we rely on a Docker image and container. The tale maintains a reference to a Git repository (including a hash to the specific commit version). This Git repository is used as the working directory to build the docker image. The environment includes a name, a Git repository URL, a commit ID that references specific version of the repository, and an optional configuration object that defines specific parameters passed to Docker when running the container.

Data is represented by a set of Girder objects (folders, items, files). Each object is described with several internal descriptors (e.g., name, size, child–parent relations, uuid, creation/modification time etc.). Those that are most important to capture in a tale are the source URL and access protocol (e.g., https://website/file1, globus://endpoint/file1), provider (e.g., DataONE, Globus, etc.), unique identifier (e.g., URN in DataONE, or DOI used by a publication repository), and in the future an optional checksum. Given these objects are mapped to a filesystem, each object must also include its size and a POSIX compatible name. The name includes the full path (with respect to the mount point) as the tale must recreate the entire directory structure.

Metadata represents information about the tale that is not specifically related to the environment or data. We expect that tale metadata will grow over time based on user needs and tale usage. At present, tales may include metadata that describes the title, authors, description, icon, illustration, category (e.g., tags), publication status, as well as licensing information for all artifacts [70].

6. Related work

Scientific reproducibility is becoming an increasingly widespread concern and stakeholders are exploring a range of approaches to address challenges. For example, data repositories now support data analysis [71,72], science gateways facilitate the capture of rich provenance information [73], and publishers enable verification of figures and computational results from within papers [74], and through third party offerings [75,76].

Science gateways allow users to conduct (generally domain-specific) analyses that exploit advanced computing infrastructure. They provide intuitive user interfaces that abstract the complexities of submitting jobs via queue submission systems or instantiating virtual computing environments for executing GUI-based tools [77]. Given the gateway's position at the center of all analysis it is possible to capture the steps performed by users (see e.g., [78]). Often these steps are recorded in the form of workflows [79] or in other standard formats [80]. Science gateways are generally focused on a specific domain, and on the analysis of data. Unlike the Whole Tale they do not provide a general model for capturing and sharing computational processes on arbitrary datasets and linking these artifacts with publications.

Scientific workflow systems, such as Galaxy [79] and Kepler [81], provide the ability for users to create flexible analysis routines comprising various processing steps. They typically provide extensible interfaces via which external data can be imported for analysis. While their goals overlap somewhat with the

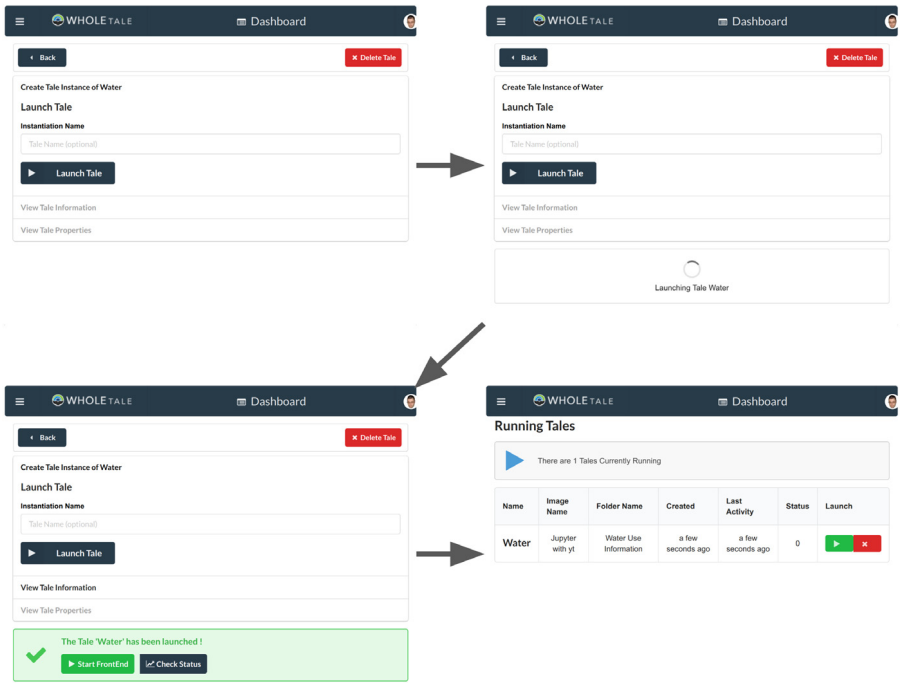

Fig. 6. An instantiation of a tale.

Whole Tale there are significant differences between these systems. Workflow systems prescribe a particular format and analysis model, they therefore require researchers to modify their computational processes to fit their model. They do not support the range of interactive and user-specific analyses enabled by the Whole Tale.

As data repositories grow in size and usage there is increasing interest in offering co-located analysis capabilities. Often these capabilities are offered as a set of tools that allow users to aggregate datasets and perform simple computations. However, recently several data repositories have added more advanced computing environments for processing managed data. For example, the Wolfram Data Repository [82] provides tight coupling with the Wolfram programming environment for analyzing and visualizing hosted data. The Cloud Kotta secure data enclave [83,84] provides a co-located analysis framework that supports interactive Jupyter notebooks and a batch submission system for analyzing sensitive data. These systems support only data stored within their respective repositories. They also do not provide a model for sharing analyses in a standard format nor are they capable of capturing complete provenance.

The widespread adoption of interactive programming environments (e.g., Jupyter) have lead to countless examples of multi-user, interactive analysis environments. For example, JupyterHub [85], supports multiple Jupyter notebook instances simultaneously via execution of notebook processes on a single server. Tmpnb [86] and Binder [87] provide multi-user environments by launching Docker containers for notebook instances. Tmpnb is used to provide temporary notebooks for replicating analyses published in Nature [74]. These systems provide similar analysis capabilities to the Whole Tale, however, they do not provide standard models for discovering and accessing data, capturing the computational process and the data used, or any form of linkage to publications.

Several platforms have emerged with the aim of hosting or linking digital scholarly objects to publications, including Zenodo [88], RunMyCode [89], ResearchCompendia [90], and SparseLab [91]. These platforms typically provide a web-based location for collecting data, code, and other information required for verification of the published claims, with a link to the article or the article itself.

Publishers are providing repository services either as standalone or in support of published claims, in addition to hosting supplementary materials. Springer-Nature for example provides the figshare service [8], and Elsevier provides Mendeley [92]—both host digital scholarly objects such as data and code and attach unique identifiers such as digital object identifiers (DOIs) to hosted items. Other projects exist to help close similar gaps in a variety of areas. For example, the journal Image Processing Online (http://ipol.im) provides reproducible publications for the image processing community, Code Ocean provides reproducibility functionality for IEEE publications, the Madagascar project extends the reproducibility functionality described by Claerbout and Karrenbach in 1992 (http://www.ahay.org), and the WaveLab project pioneered reproducibility in signal processing (http://statweb.stanford.edu/~wavelab/), just to name a few.

There are other general efforts aimed at aggregating digital resources and capturing the provenance of research artifacts. W3C PROV [93] defines a model for representing provenance using a data model that represents the entities (e.g., files), agents (e.g., people), and activities (e.g., computational processes) associated with data processing. Research Objects (RO) [94] provides a model for capturing a single unit of research including, for example, the datasets, analysis scripts, and derived results associated with a paper. RO provides a formal specification for encoding these objects, as well as associated attribution and provenance information. Both W3C PROV and ROs could be used as a basis for representing a tale. We are actively exploring interoperability between these models.

7. Summary

The widespread adoption of computational and data-driven science has significantly altered the discovery lifecycle. However, the methods by which scientific results are published have not kept pace with the drastic changes to the underlying processes used for discovery. The Whole Tale aims to redefine the model via which computational and data-driven science is conducted, published, verified, and reproduced. The Whole Tale builds upon a wide range of efforts to support data discovery and ingestion, analysis using

flexible frontends, scalable computation in isolated containers, and ultimately publication of verifiable and reproducible processes using these artifacts.

The Whole Tale architecture consists of a set of microservices (e.g., for data access, persistent identifier creation, etc.) and interoperability software that leverages, where possible, existing cyberinfrastructure. The resulting services not only provide value to end users through the Whole Tale web interface but also application developers through REST APIs. Through a number of Whole Tale working groups, we are actively engaging several science communities to pilot these capabilities and evaluate their use for enabling reproducible science.

Acknowledgments

This research was supported by NSF, United States grant 1541450 (CC*DNI DIBBS: Merging Science and Cyberinfrastructure Pathways: The Whole Tale).

References

[1] R.D. Peng, Reproducible research in computational science, Science 334 (6060) (2011) 1226–1227. http://dx.doi.org/10.1126/science.1213847.
[2] J. Kratz, C. Strasser, Data publication consensus and controversies, F1000Research 3 (94). http://dx.doi.org/10.12688/f1000research.3979.3.
[3] A.A. Alsheikh-Ali, W. Qureshi, M.H. Al-Mallah, J.P. Ioannidis, Public availability of published research data in high-impact journals, PLoS ONE 6 (9) (2011) e24357.
[4] V. Stodden, D.H. Bailey, J. Borwein, R.J. LeVeque, W. Rider, W. Stein, Setting the default to reproducible. Reproducibility in Computational and Experimental Mathematics, Tech. rep. http://icerm.brown.edu/tw12-5-rcem. (Last Accessed March 2017).
[5] D.L. Donoho, A. Maleki, I.U. Rahman, M. Shahram, V. Stodden, Reproducible research in computational harmonic analysis, Comput. Sci. Eng. 11 (1) (2009) 8–18.
[6] V. Stodden, F. Leisch, R.D. Peng, Implementing Reproducible Research, CRC Press, 2014.
[7] M. Crosas, The dataverse network: An open-source application for sharing, discovering and preserving data, D-Lib Magazine 17 (1/2).
[8] figshare, 2017. http://figshare.com, web site. (Accessed May 2017).
[9] K. Chard, J. Pruyne, B. Blaiszik, R. Ananthakrishnan, S. Tuecke, I. Foster, Globus data publication as a service: Lowering barriers to reproducible science, in: Proceedings of the 11th IEEE International Conference on e-Science, 2015, pp. 401–410. http://dx.doi.org/10.1109/eScience.2015.68.
[10] COPDESS, Statement of commitment from earth and space science publishers and data facilities, 2015. http://www.copdess.org/statement-of-commitment/.
[11] T.H. Vines, A. Albert, R. Andrew, F. Debarre, D. Bock, M. Franklin, K. Gilbert, J.-S. Moore, S. Renaut, D. Rennison, The availability of research data declines rapidly with article age, Curr. Biol. 24 (1) (2014) 94–97. http://dx.doi.org/10.1016/j.cub.2013.11.014.
[12] B. Ludäscher, K. Chard, N. Gaffney, M.B. Jones, J. Nabrzyski, V. Stodden, M. Turk, Capturing the "whole tale" of computational research: Reproducibility in computing environments. http://arxiv.org/abs/1610.09958.
[13] V. Stodden, M. McNutt, D.H. Bailey, E. Deelman, Y. Gil, B. Hanson, M.A. Heroux, J.P. Ioannidis, M. Taufer, Enhancing reproducibility for computational methods, Science 354 (6317) (2016) 1240–1241. http://dx.doi.org/10.1126/science.aah6168.
[14] N. Wilkins-Diehr, Special issue: Science gateways—common community interfaces to grid resources, Concurr. Comput.: Pract. Exper. 19 (6) (2007) 743–749. http://dx.doi.org/10.1002/cpe.1098.
[15] B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, I. Foster, The materials data facility: Data services to advance materials science research, J. Miner. Met. Mater. Soc. 68 (8) (2016) 2045–2052. http://dx.doi.org/10.1007/s11837-016-2001-3.
[16] A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K.A. Persson, Commentary: The materials project: A materials genome approach to accelerating materials innovation, APL Mater. 1 (1) (2013) 011002. http://dx.doi.org/10.1063/1.4812323.
[17] J. O'Mara, B. Meredig, K. Michel, Materials data infrastructure: A case study of the citrination platform to examine data import, storage, and access, J. Miner. Met. Mater. Soc. 68 (8) (2016) 2031–2034. http://dx.doi.org/10.1007/s11837-016-1984-0.
[18] K.S. Thygesen, K.W. Jacobsen, Making the most of materials computations, Science 354 (6309) (2016) 180–181. http://dx.doi.org/10.1126/science.aah4776.
[19] S. Yip, HandBook of Materials Modeling, Springer Netherlands, 2007.
[20] Committee on Accelerating Technology Transition, National Materials Advisory Board, Board on Manufacturing and Engineering Design, Division on Engineering and Physical Sciences, National Research Council of the National Academies, Accelerating Technology Transition: Bridging the Valley of Death for Materials and Processes in Defense Systems, National Academies Press, 2004. http://dx.doi.org/10.17226/11108.
[21] S. Curtarolo, G.L.W. Hart, M.B. Nardelli, N. Mingo, S. Sanvito, O. Levy, The high-throughput highway to computational materials design, Nature Mater. 12 (3) (2013) 1122–1476. http://dx.doi.org/10.1038/nmat3568.
[22] J. Hill, G. Mulholland, K. Persson, R. Seshadri, C. Wolverton, B. Meredig, Materials science with large-scale data and informatics: Unlocking new opportunities, MRS Bull. 41 (5) (2016) 399–409. http://dx.doi.org/10.1557/mrs.2016.93.
[23] L. Ward, A. Agrawal, A. Choudhary, C. Wolverton, A general-purpose machine learning framework for predicting properties of inorganic materials, NPJ Comput. Mater. 2. http://dx.doi.org/10.1038/npjcompumats.2016.28.
[24] K. Rajan, Materials informatics, Mater. Today 8 (10) (2005) 38–45. http://dx.doi.org/10.1016/S1369-7021(05)71123-8.
[25] Y. Kawazoe, J.-Z. Yu, A.-P. Tsai, T. Masumoto (Eds.), Phase diagrams and physical properties of nonequilibrium alloys, in: Nonequilibrium Phase Diagrams of Ternary Amorphous Alloys, Springer Berlin Heidelberg, 1997.
[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: An update, SIGKDD Explor. Newsl. 11 (1) (2009) 10–18. http://dx.doi.org/10.1145/1656274.1656278.
[27] B.D. Smith, J.H. Wise, B.W. O'Shea, M.L. Norman, S. Khochfar, The first Population II stars formed in externally enriched mini-haloes, Mon. Not. R. Astron. Soc. 452 (2015) 2822–2836. http://dx.doi.org/10.1093/mnras/stv1509. arXiv:1504.07639.
[28] J.-h. Kim, T. Abel, O. Agertz, G.L. Bryan, D. Ceverino, C. Christensen, C. Conroy, A. Dekel, N.Y. Gnedin, N.J. Goldbaum, J. Guedes, O. Hahn, A. Hobbs, P.F. Hopkins, C.B. Hummels, F. Iannuzzi, D. Keres, A. Klypin, A.V. Kravtsov, M.R. Krumholz, M. Kuhlen, S.N. Leitner, P. Madau, L. Mayer, C.E. Moody, K. Nagamine, M.L. Norman, J. Onorbe, B.W. O'Shea, A. Pillepich, J.R. Primack, T. Quinn, J.I. Read, B.E. Robertson, M. Rocha, D.H. Rudd, S. Shen, B.D. Smith, A.S. Szalay, R. Teyssier, R. Thompson, K. Todoroki, M.J. Turk, J.W. Wadsley, J.H. Wise, A. Zolotov, t. AGORA Collaboration29, The AGORA high-resolution galaxy simulations comparison project, Astrophys. J. Suppl. 210 (2014) 14. http://dx.doi.org/10.1088/0067-0049/210/1/14. arXiv:1308.2669.
[29] M.J. Turk, B.D. Smith, J.S. Oishi, S. Skory, S.W. Skillman, T. Abel, M.L. Norman, yt: A multi-code analysis toolkit for astrophysical simulation data, Astrophys. J. Suppl. 192 (2011) 9. http://dx.doi.org/10.1088/0067-0049/192/1/9. arXiv:1011.3514.
[30] K.W. Kintigh, J.H. Altschul, M.C. Beaudry, R.D. Drennan, A.P. Kinzig, T.A. Kohler, W.F. Limp, H.D.G. Maschner, W.K. Michener, T.R. Pauketat, P. Peregrine, J.A. Sabloff, T.J. Wilkinson, H.T. Wright, M.A. Zeder, Grand Challenges for Archaeology, American Antiquity.
[31] K.W. Kintigh, J.H. Altschul, A.P. Kinzig, W.F. Limp, W.K. Michener, J.A. Sabloff, E.J. Hackett, T.A. Kohler, B. Ludäscher, C.A. Lynch, Cultural dynamics, deep time, and data, Adv. Archaeol. Pract. 3 (1) (2015) 1–15.
[32] Synthesizing knowledge of past environments, https://www.openskope.org/. (Last Accessed March 2017).
[33] T. McPhillips, T. Song, T. Kolisnik, S. Aulenbach, K. Belhajjame, R.K. Bocinsky, Y. Cao, J. Cheney, F. Chirigati, S. Dey, J. Freire, C. Jones, J. Hanken, K.W. Kintigh, T.A. Kohler, D. Koop, J.A. Macklin, P. Missier, M. Schildhauer, C. Schwalm, Y. Wei, M. Bieda, B. Ludäscher, YesWorkflow: A user-oriented language-independent tool for recovering workflow information from scripts, Int. J. Digit. Curation 10 (1) (2015) 298–313. http://dx.doi.org/10.2218/ijdc.v10i1.370.
[34] T. McPhillips, S. Bowers, K. Belhajjame, B. Ludäscher, Retrospective provenance without a runtime provenance recorder, in: Proceedings of the 7th USENIX Conference on Theory and Practice of Provenance, USENIX Association, 2015.
[35] J.F. Pimentel, S. Dey, T. McPhillips, K. Belhajjame, D. Koop, L. Murta, V. Braganholo, B. Ludäscher, Yin & yang: demonstrating complementary provenance from noworkflow & yesworkflow, in: International Provenance and Annotation Workshop, Springer, 2016, pp. 161–165.
[36] Q. Zhang, Y. Cao, Q. Wang, D. Vu, P. Thavasimani, T. McPhillips, P. Missier, P. Slaughter, C. Jones, M.B. Jones, B. Ludäscher, Revealing the detailed lineage of script outputs using hybrid provenance, in: 12th International Digital Curation Conference, Edinburgh, Scotland, Int. J. Digit. Curation (2017).
[37] How to do archaeological science using R, https://benmarwick.github.io/How-To-Do-Archaeological-Science-Using-R/. (Last Accessed March 2017).
[38] B. Marwick, C. Boettiger, L. Mullen, Packaging data analytical work reproducibly using R (and friends), Tech. Rep. e3192v1, PeerJ Preprints, Aug. 2017.http://dx.doi.org/10.7287/peerj.preprints.3192v1, https://peerj.com/preprints/3192.
[39] K. Bocinsky, A. Budden, M. Jones, B. Ludäscher, D. Vieglais, Prov-a-thon: Practical tools for reproducible science, 2017. https://github.com/DataONEorg/provathon-2017.

[40] M. Shahram, V. Stodden, D.L. Donoho, A. Maleki, I.U. Rahman, Reproducible research in computational harmonic analysis, Comput. Sci. Eng. 11 (2009) 8–18. http://dx.doi.org/10.1109/MCSE.2009.15.

[41] J.F. Claerbout, M. Karrenbach, Electronic documents give reproducible research a new meaning, 1992, pp. 601–604. http//dx.doi.org/10.1190/1.1822162.

[42] V. Stodden, M. McNutt, D.H. Bailey, E. Deelman, Y. Gil, B. Hanson, M.A. Heroux, J.P. Ioannidis, M. Taufer, Enhancing reproducibility for computational methods, Science 354 (6317) (2016) 1240–1241. http://dx.doi.org/10.1126/science.aah6168.

[43] A.M. Smith, D.S. Katz, K.E. Niemeyer, FORCE11 Software Citation Working Group, Software citation principles, PeerJ Comput. Sci. 2 (2016) e86. http://dx.doi.org/10.7717/peerj-cs.86.

[44] M. Martone, Data citation synthesis group: Joint declaration of data citation principles, FORCE11. https://www.force11.org/datacitation.

[45] B.A. Nosek, G. Alter, G.C. Banks, D. Borsboom, S.D. Bowman, S.J. Breckler, S. Buck, C.D. Chambers, G. Chin, G. Christensen, M. Contestabile, A. Dafoe, E. Eich, J. Freese, R. Glennerster, D. Goroff, D.P. Green, B. Hesse, M. Humphreys, J. Ishiyama, D. Karlan, A. Kraut, A. Lupia, P. Mabry, T. Madon, N. Malhotra, E. Mayo-Wilson, M. McNutt, E. Miguel, E.L. Paluck, U. Simonsohn, C. Soderberg, B.A. Spellman, J. Turitto, G. VandenBos, S. Vazire, E.J. Wagenmakers, R. Wilson, T. Yarkoni, Promoting an open research culture, Science 348 (6242) (2015) 1422–1425. http://dx.doi.org/10.1126/science.aab2374.

[46] V. Stodden, The legal framework for reproducible scientific research: Licensing and copyright, Comput. Sci. Eng. 11 (1) (2009) 35–40. http://dx.doi.org/10.1109/MCSE.2009.19.

[47] V. Stodden, Intellectual property and computational science, in: Opening Science, Springer, 2014, pp. 225–235.

[48] B.A. Nosek, G. Alter, G.C. Banks, D. Borsboom, S.D. Bowman, S.J. Breckler, S. Buck, C.D. Chambers, G. Chin, G. Christensen, M. Contestabile, A. Dafoe, E. Eich, J. Freese, R. Glennerster, D. Goroff, D.P. Green, B. Hesse, M. Humphreys, J. Ishiyama, D. Karlan, A. Kraut, A. Lupia, P. Mabry, T. Madon, N. Malhotra, E. Mayo-Wilson, M. McNutt, E. Miguel, E.L. Paluck, U. Simonsohn, C. Soderberg, B.A. Spellman, J. Turitto, G. VandenBos, S. Vazire, E.J. Wagenmakers, R. Wilson, T. Yarkoni, Promoting an open research culture, Science 348 (6242) (2015) 1422–1425. http://dx.doi.org/10.1126/science.aab2374.

[49] V. Stodden, P. Guo, Z. Ma, Toward reproducible computational research: an empirical analysis of data and code policy adoption by journals, PLoS One 8 (6) (2013) e67111.

[50] NSF Workshop - Systematic Approach to Robustness, Reliability, and Reproducibility in Scientific Research, http://www.mrsec.harvard.edu/2017NSFReliability/.(Last Accessed March 2017).

[51] CISE AC Data Science Report, https://www.nsf.gov/cise/ac-data-science-report/CISEACDataScienceReport1.19.17.pdf.(Last Accessed March 2017).

[52] ACM - Artifact Review and Badging, https://www.acm.org/publications/policies/artifact-review-badging.(Last Accessed March 2017).

[53] IEEE Workshop - The Future of Research Curation and Research Reproducibility, http://www.ieee.org/publications_standards/publications/ieee_workshops/research_reproducibility.html.(Last Accessed March 2017).

[54] National Academies of Sciences, Engineering, and Medicine, Fostering Integrity in Research, The National Academies Press, Washington, DC, 2017. http://dx.doi.org/10.17226/21896.

[55] Berlin declaration on open access to knowledge in the sciences and humanities, https://openaccess.mpg.de/Berlin-Declaration. (Last Accessed March 2017).

[56] Eudat, https://eudat.eu/. (Last Accessed March 2017).

[57] Openaire, https://www.openaire.eu/. (Last Accessed March 2017).

[58] Eurocloud, https://eurocloud.org/. (Last Accessed March 2017).

[59] The european code of conduct for research integrity, http://ec.europa.eu/research/participants/data/ref/h2020/other/hi/h2020-ethics_code-of-conduct_en.pdf.(Last Accessed March 2017).

[60] J. Freire, N. Fuhr, A. Rauber, Reproducibility of data-oriented experiments in e-Science (Dagstuhl Seminar 16041), Dagstuhl Rep. 6 (1) (2016) 108–159. http://dx.doi.org/10.4230/DagRep.6.1.108.

[61] W.K. Michener, S. Allard, A. Budden, R.B. Cook, K. Douglass, M. Frame, S. Kelling, R. Koskela, C. Tenopir, D.A. Vieglais, Participatory design of DataONE –enabling cyberinfrastructure for the biological and environmental sciences, Ecol. Inform. 11 (2012) 5–15. http://dx.doi.org/10.1016/j.ecoinf.2011.08.007. Data platforms in integrative biodiversity research.

[62] C. Lagoze, van de Sompel Herbert, M. Nelson, S. Warner, The open archives initiative protocol for metadata harvesting, 2008. http://www.openarchives.org/OAI/openarchivesprotocol.html.

[63] Jupyter notebook, http://jupyter.org. (Last Accessed March 2017).

[64] Girder, 2017. https://girder.readthedocs.io/en/latest/ (Last Accessed March 2017).

[65] K. Chard, S. Tuecke, I. Foster, Efficient and secure transfer, synchronization, and sharing of big data, IEEE Cloud Comput. 1 (3) (2014) 46–55. http://dx.doi.org/10.1109/MCC.2014.52.

[66] EmberJS, http://emberjs.com/. (Last Accessed March 2017).

[67] Semantic UI, http://semantic-ui.com/. (Last Accessed March 2017).

[68] S. Tuecke, R. Ananthakrishnan, K. Chard, M. Lidman, B. McCollam, S. Rosen, I. Foster, Globus auth: A research identity and access management platform, in: 12th IEEE International Conference on e-Science (e-Science), 2016, pp. 203–212. http://dx.doi.org/10.1109/eScience.2016.7870901.

[69] M. Szeredi, File system in user space, http://fuse.sourceforge.net. (Last Accessed March 2017).

[70] V. Stodden, The legal framework for reproducible scientific research: Licensing and copyright, Comput. Sci. Eng. 11 (1) (2009) 35–40.

[71] J. Raddick, D. Medvedev, G. Lemson, B. Souter, SciServer compute brings analysis to big data in the cloud, in: American Astronomical Society Meeting Abstracts, in: American Astronomical Society Meeting Abstracts, vol. 228, 2016, p. 317.06.

[72] C. Willis, D. LeBauer, M. Lambert, M. Burnette, TERRA-REF analysis workbench: container-based environments for low-barrier access to research data, May 2017. http://dx.doi.org/10.5281/zenodo.580057.

[73] S. Gesing, R. Dooley, M. Pierce, J. Krüger, R. Grunzke, S. Herres-Pawlis, A. Hoffmann, Science gateways - leveraging modeling and simulations in hpc infrastructures via increased usability, in: International Conference on High Performance Computing Simulation, HPCS, 2015, pp. 19–26. http://dx.doi.org/10.1109/HPCSim.2015.7237017.

[74] H. Shen, Interactive notebooks: Sharing the code, Nature 515 (7525) (2014) 151–152.

[75] L. Zelnik-Manor, K. Rosenblum, Y.C. Eldar, Sensing matrix optimization for block-sparse decoding, IEEE Trans. Signal Process. 59 (9) (2011) 4300–4312. http://dx.doi.org/10.1109/TSP.2011.2159211.

[76] A. Gilinsky, L.Z. Manor, Siftpack: A compact representation for efficient sift matching, in: 2013 IEEE International Conference on Computer Vision, 2013, pp. 777–784. http://dx.doi.org/10.1109/ICCV.2013.101.

[77] M. McLennan, R. Kennell, Hubzero: A platform for dissemination and collaboration in computational science and engineering, Comput. Sci. Eng. 12 (2) (2010) 48–53. http://dx.doi.org/10.1109/MCSE.2010.41.

[78] D. James, N. Wilkins-Diehr, V. Stodden, D. Colbry, C. Rosales, M.R. Fahey, J. Shi, R.F. da Silva, K. Lee, R. Roskies, L. Loewe, S. Lindsey, R. Kooper, L. Barba, D.H. Bailey, J.M. Borwein, Ó. Corcho, E. Deelman, M.C. Dietze, B. Gilbert, J. Harkes, S. Keele, P. Kumar, J. Lee, E. Linke, R. Marciano, L. Marini, C. Mattmann, D. Mattson, K. McHenry, R.T. McLay, S. Miguez, B.S. Minsker, M.S. Pérez-Hernández, D. Ryan, M. Rynge, I.S. Pérez, M. Satyanarayanan, G.S. Clair, K. Webster, E. Hovig, D.S. Katz, S. Kay, G.K. Sandve, D. Skinner, G. Allen, J. Cazes, K.W. Cho, J. Fonseca, L. Hwang, L. Koesterke, P. Patel, L. Pouchard, E. Seidel, I. Suriarachchi, Standing together for reproducibility in large-scale computing: Report on reproducibility@xsede,.

[79] J. Goecks, A. Nekrutenko, J. Taylor, Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences, Genome Biol. 11 (8) (2010) R86. http://dx.doi.org/10.1186/gb-2010-11-8-r86.

[80] K. Belhajjame, R. B'Far, J. Cheney, S. Coppens, S. Cresswell, Y. Gil, P. Groth, G. Klyne, T. Lebo, J. McCusker, S. Miles, J. Myers, S. Sahoo, C. Tilmes, Prov-dm: The prov data model, Tech. rep., 2012. http://www.w3.org/TR/prov-dm/. (Last Accessed March 2017).

[81] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E.A. Lee, J. Tao, Y. Zhao, Scientific workflow management and the kepler system, Concurr. Comput.: Pract. Exper. 18 (10) (2006) 1039–1065. http://dx.doi.org/10.1002/cpe.994.

[82] Wolfram data repository, 2017. https://datarepository.wolframcloud.com/. (Last Accessed March 2017).

[83] Y.N. Babuji, K. Chard, A. Gerow, E. Duede, Cloud kotta: Enabling secure and scalable data analytics in the cloud, in: Proceedings of the IEEE International Conference on Big Data (Big Data), 2016, pp. 302–310. http://dx.doi.org/10.1109/BigData.2016.7840616.

[84] Y.N. Babuji, K. Chard, A. Gerow, E. Duede, A secure data enclave and analytics platform for social scientists, in: Proceedings of the 12th IEEE International Conference on e-Science (e-Science), 2016, pp. 337–342. http://dx.doi.org/10.1109/eScience.2016.7870917.

[85] Jupyterhub, 2017. https://github.com/jupyterhub/jupyterhub. (Last Accessed March 2017).

[86] tmpnb, the temporary notebook service, 2017. https://github.com/jupyter/tmpnb. (Last Accessed March 2017).

[87] Binder, 2017. http://mybinder.org//. (Last Accessed March 2017).

[88] Zenodo, https://zenodo.org/. (Last accessed March 2017).

[89] V. Stodden, C. Hurlin, C. Pérignon, Runmycode.org: A novel dissemination and collaboration platform for executing published computational results, in: 2012 IEEE 8th International Conference on E-Science, 2012, pp. 1–8. http://dx.doi.org/10.1109/eScience.2012.6404455.

[90] V. Stodden, S. Miguez, J. Seiler, Researchcompendia.org: Cyberinfrastructure for reproducibility and collaboration in computational science, Comput. Sci. Eng. 17 (1) (2015) 12–19. http://dx.doi.org/10.1109/MCSE.2015.18.

[91] D. Donoho, Sparselab, http://sparselab.stanford.edu/. (Last Accessed March 2017).

[92] Mendeley, https://www.mendeley.com/. (Last Accessed March 2017).

[93] Y. Gil, S. Miles, K. Belhajjame, H. Deus, D. Garijo, G. Klyne, P. Missier, S. Soiland-Reyes, S. Zednik, Prov model primer, Tech. rep. W3C, 2012. (Last Accessed March 2017).
[94] S. Bechhofer, D.D. Roure, M. Gamble, C. Goble, I. Buchan, Research Objects: Towards exchange and reuse of digital knowledge, in: Workshop on the Future of the Web for Collaborative Science, 2010. http://dx.doi.org/10.1038/npre.2010.4626.1.

Kyle Chard is a Senior Researcher and Fellow in the Computation Institute at the University of Chicago and Argonne National Laboratory. He received his Ph.D. in computer science from Victoria University of Wellington. His research interests include distributed meta-scheduling, cloud computing, economic resource allocation, social computing, and services computing.

Niall Gaffney's background largely revolves around the management and utilization of large inhomogeneous scientific datasets. He is currently the Director for Data Intensive Computing at the Texas Advanced Computing Center (TACC) where he uses his experience combining data, computation, and interfaces to create reproducible data driven science results. He joined TACC in May 2013. Prior to that he worked for 13 years in the role of designer and developer for the archives housed at the Space Telescope Science Institute (STScI), which hold the data from the Hubble Space Telescope, Kepler, and James Webb Space Telescope missions. He was also a leader in the development of the Hubble Legacy Archive, projects that harvested the 20+ years of Hubble Space Telescope data to create some of the most sensitive astronomical data products available for open research. Prior to his work at STScI, Niall was worked as "the friend of the telescope" for the Hobby Eberly Telescope (HET) project at the McDonald Observatory in west Texas where he started working to create systems to acquire and handle the storage and distribution of the data the HET produced.

Matthew B. Jones is the Director of Informatics at the National Center for Ecological Analysis and Synthesis at UC Santa Barbara, where he serves as PI for the NSF Arctic Data Center, co-PI for the DataONE federation of data repositories, and leads the KNB Data Repository. His research focuses on creating open science solutions to data and software challenges that have historically impeded large scale scientific synthesis. Recent projects have produced provenance tracking tools, metadata standards, data repository software, semantic search systems, and scientific workflow software. Jones has a B.A. degree in ecology from Dartmouth College, and a M.S. in Zoology from the University of Florida.

Kacper Kowalik received his undergraduate degree at Nicolaus Copernicus University (Torun, Poland) in 2008 and his Ph.D. in Astronomy also from Nicolaus Copernicus University in 2014. His research in astrophysics was primarily focused on early stages of protoplanetary formation and circumstellar disks' instabilities. He is interested in high performance computations — especially in the domain sciences where they haven't been widely adopted, developing new ways of sharing and interacting with large computational datasets, applying industrial IT solutions to scientific software.

Bertram Ludäscher is a professor at the School of Information Sciences at the University of Illinois, Urbana-Champaign, where he directs the Center for Informatics Research in Science and Scholarship. He is a faculty affiliate at the National Center for Supercomputing Applications (NCSA) and the Department of Computer Science. His research interests include scientific data and workflow management, provenance modeling and querying, and knowledge representation and reasoning. Prior to joining Illinois he was was a CS faculty and member of the Genome Center at UC Davis. He received his M.S. in computer science from the University of Karlsruhe (K.I.T.) and his Ph.D. from the University of Freiburg, both in Germany.

Bryce Mecum is a Science Software Engineer at the National Center for Ecological Analysis and Synthesis at UC Santa Barbara. His research interests include building software for open science, ecological forecasting, and fisheries management. He has a B.S. in Marine Biology from Western Washington University and an M.S. in Fisheries from the University of Alaska Fairbanks.

Jarek Nabrzyski is the Director of the Center for Research Computing and Concurrent Professor of Computer Science and Engineering at the University of Notre Dame. Before joining Notre Dame in 2009 Nabrzyski was the Executive Director of the LSU's Center for Computation and Technology, and before that he managed the Scientific Application Department at the Poznan Supercomputing and Networking Center in Poznań, Poland. Nabrzyski has received his M.Sc. and Ph.D. in Computer Science and Engineering from the Poznan University of Technology. His research interests cover scientific computing, distributed resource management and scheduling, cloud computing, decision support systems and broad aspects of reproducibility in science.

Victoria Stodden is an associate professor in the School of Information Sciences at the University of Illinois at Urbana-Champaign, with affiliate appointments in the School of Law, the Department of Computer Science, the Department of Statistics, the Coordinated Science Laboratory, and the National Center for Supercomputing Applications. She is also a faculty affiliate of the Center for Informatics Research in Science and Scholarship in the School of Information Sciences at the University of Illinois. Her research focuses on understanding the effect of big data and computation on scientific inference, including studying adequacy and robustness in replicated results, designing and implementing validation systems, developing standards of openness for data and code sharing, and resolving legal and policy barriers to disseminating reproducible research. She completed both her Ph.D. in statistics and her law degree at Stanford University, and graduated magna cum laude from the University of Ottawa. Her website is http://stodden.net.

Ian Taylor is a Research Professor of Computer Science and Engineering and a Computational Scientist at the CRC, Notre Dame, and a Reader in Cardiff University, UK. Ian has a degree in Computing Science and a Ph.D. researching and implementing artificial-neural-network types for the determination of musical pitch. Ian's research over the last 25 years has covered a broad range of distributed computing areas but he now specializes in Web interaction and APIs, big data applications, open data access, distributed scientific workflows and data analytics. He has managed over 15 research and industrial projects, published over 150 papers, 3 books, acted as guest editor for several special issues in journals and chairs the WORKS Workflow workshop yearly at Supercomputing. Ian has won the Naval Research Lab best research paper (ALAN Berman) prize in 2010, 2011 and 2015.

Matthew J. Turk is an Assistant Professor in the School of Information Sciences at University of Illinois Urbana-Champaign, with an appointment in the Astronomy department. His interests include developing systems for data analysis and visualization, understanding the formation of the first stars in the universe, and examining community-driven development in scholarly software. Turk has a B.A. in Physics from Northwestern University and a Ph.D. in Physics from Stanford.