FAIRSCAPE: A Framework for FAIR and Reproducible Biomedical Analytics

Maxwell Adam Levinson*, Justin Niestroy*, Sadnan Al Manir*, Karen Fairchild, Douglas E. Lake, J. Randall Moorman, Timothy Clark**

* Equal contributions ** Correspondence: Tim Clark, twclark@virginia.edu

Abstract

Results of computational analyses require transparent disclosure of their supporting resources, while the analyses themselves often can be very large scale and involve multiple processing steps separated in time. Evidence for the correctness of any analysis consists of accessible data and software with runtime parameters, environment, and personnel involved.

Evidence graphs - a derivation of argumentation frameworks adapted to biological science - can provide this disclosure as machine-readable metadata resolvable from persistent identifiers for computationally generated graphs, images, or tables, that can be archived and cited in a publication including a persistent ID.

We have built a cloud-based, computational research commons for predictive analytics on biomedical time series datasets with hundreds of algorithms and thousands of computations using a reusable computational framework we call FAIRSCAPE.

FAIRSCAPE computes a complete chain of evidence on every result, including software, computations, and datasets. An ontology for Evidence Graphs, EVI (https://w3id.org/EVI), supports inferential reasoning over the evidence. FAIRSCAPE can run nested or disjoint workflows and preserves the provenance graph across them. It can run Apache Spark jobs, scripts, workflows, or user-supplied containers. All objects are assigned persistent IDs, including software. All results are annotated with FAIR metadata using the evidence graph model for access, validation, reproducibility, and re-use of archived data and software.

FAIRSCAPE is a reusable computational framework, enabling simplified access to modern scalable cloud-based components. It fully implements the FAIR data principles and extends them to provide FAIR Evidence, including provenance of datasets, software and computations, as metadata for all computed results.

Keywords

FAIR data, Data Lake, Data Commons, Evidence graph, Argumentation framework, Apache Spark, Workflow

Introduction

Computation is an integral part of the preparation and content of modern biomedical scientific publications, and the findings they report. Computations can range in scale from simple statistical routines run in Excel spreadsheets to massive orchestrations of very large primary datasets, computational workflows, software, cloud environments, and services. They typically produce data and generate images as output. Scientific claims of the authors are supported both by reference to the existing domain literature, and to the experimental or observational data and its analysis represented in the figure or image.

The ideal recommended practice is now to archive and cite one's own experimental data (Cousijn et al. 2018; Data Citation Synthesis Group 2014; Fenner et al. 2019; Groth et al. 2020); to make it FAIR (Wilkinson et al. 2016); and to archive and cite software used in analysis (Smith et al. 2016). That is, increasingly strict requirements are demanded to leave a digital footprint of each preparation and analysis step in derivation of a finding to support reproducibility and reuse of both data and tools. This is a welcome development, now extended by many journals into the realm of critical research reagents (A. Bandrowski 2014; A. E. Bandrowski and Martone 2016; Prager et al. 2018). How do we facilitate it? And how do we make the recorded digital footprints most useful?

Our notion, inspired by a large body of work in abstract argumentation frameworks, and analysis of biomedical publications (Tim Clark et al. 2014; Greenberg 2009, 2011), is that the evidence for correctness of any finding can be represented as a directed acyclic support graph, an Evidence Graph. When combined with a graph of challenges to statements, or their evidence, this becomes a bipolar argument graph - or argumentation system (Cayrol and Lagasquie-Schiex 2009, 2010, 2013).

We have abstracted core elements of our micropublications model (Clark et al. 2014) to create EVI (http://w3id.org/EVI), an ontology of evidence relationships that extends the W3C Provenance ontology, PROV

(Gil et al. 2013; Lebo et al. 2013; Moreau et al. 2013), to support specific evidence types found in biomedical publications, reasoning across deep evidence graphs, and propagation of evidence challenges deep in the graph, such as: retractions, reagent contamination, errors detected in algorithms, disputed validity of methods, challenges to validity of animal models, and others. (Al Manir & Clark, in preparation; w3id.org/EVI#). EVI is based on the fundamental idea that scientific findings or claims are not facts, but assertions backed by some level of evidence, *i.e.*, they are defeasible components of argumentation. Therefore, EVI focuses on the structure of evidence chains that support or challenge a result, and on providing access to the resources identified in those chains. Evidence in a scientific article is in essence, a record of the provenance of the finding, result, or claim asserted as likely to be true.

If the data and software used in analysis are all registered and receive persistent identifiers (PIDs) with appropriate metadata, a provenance-aware computational data lake, *i.e.*, a data lake with provenance-tracking computational services, can be built that attaches evidence graphs to the output of each process. At some point, a citable object - a dataset, image, figure, or table will be produced as part of the research. If this, too, is archived with its evidence graph as part of the metadata and the final supporting object is either directly cited in the text, or in a figure caption, then the complete evidence graph may be retrieved as a validation of the object's derivation and as a set of URIs resolvable to reusable versions of the toolsets and data. Evidence graphs are themselves entities that can be consumed and extended at each transformation or computation.

A cogent use case for this treatment of evidence comes from the recent Surgisphere retractions in COVID-19 research (Mehra et al. 2020; Mehra, Mandeep R et al. 2020), and earlier, the Obokata “stimulus transitioned acquisition of pluripotency” (STAP) retractions (Aizawa 2016; Ishii et al. 2014; H. Obokata et al. 2014; Haruko Obokata, Wakayama, et al. 2014). Many more such cases could be cited, including the Wakefield paper in Lancert which claimed that MMR vaccination caused autism (Deer 2011; The Editors of The Lancet 2010; Wakefield et al. 1998). In these well-publicized cases, research that initially appeared to have groundbreaking promise, was shown to be invalid based on examination of the underlying data and methods. While the Obokata and Surgisphere retractions occurred relatively quickly, due no doubt to the egregiousness of the scientific misconduct involved, *it is reasonable to believe that less obtrusive, or more well-concealed errors, malfeasance, or simple hyped-up claims with a poor (or no) basis in evidence, is much more prevalent*.

We set out to construct a provenance-aware computational data lake, as described above, by significantly extending and refactoring the identifier and metadata services framework we and our colleagues developed in the NIH Data Commons Pilot Project Consortium (Timothy Clark et al. 2018; Fenner et al. 2018). This framework successfully demonstrated interoperability across several NIH "Data Commons" environments, providing the identifier, authN/authZ, and metadata management elements of Grossman's "data ecosystem" concept (Grossman 2019). We extended and re-engineered this framework over time to track and visualize computations and their evidence, to manage the computational objects (such as data and software) as well as their metadata, to analyze very large datasets with horizontal scale-out, to support neuroimaging workflows, and to make it generally more easy for scientists and computational analysts to use, by providing Binder and Notebook services (Jupyter et al. 2018; Kluyver et al. 2016), and a Python client.

End-users do not need to learn a new programming language to use services provided by FAIRSCAPE. They require no additional special expertise, other than basic familiarity with Python and the skillsets they already possess in statistics, computational biology, machine learning, or other data science techniques. FAIRSCAPE provides an environment that makes large-scale computational work easier and results FAIRer.

FAIRSCAPE is a reusable framework, suitable for installation in private, public, or hybrid cloud environments. We have also installed it on a high-end laptop. It focuses on ease of use for computational researchers, while capturing an extensible record of provenance, transparently to the user. FAIRSCAPE provenance is rendered as named evidence graphs. These provide a complete record of any series of computations, with FAIR access to every digital object in a series of computations and transformations, whether or not connected in a workflow, or done by different users, including both datasets and software source code.

The remainder of this article describes the approach, microservices architecture, and interaction model of the FAIRSCAPE framework in detail.

Materials and Methods

1. FAIRSCAPE Architectural Layers

FAIRSCAPE is built on a multi-layer set of components using a containerized microservice architecture (MSA) (Balalaie et al. 2016; Larrucea et al. 2018; Lewis and Fowler 2014; Wan et al. 2018) running under Kubernetes

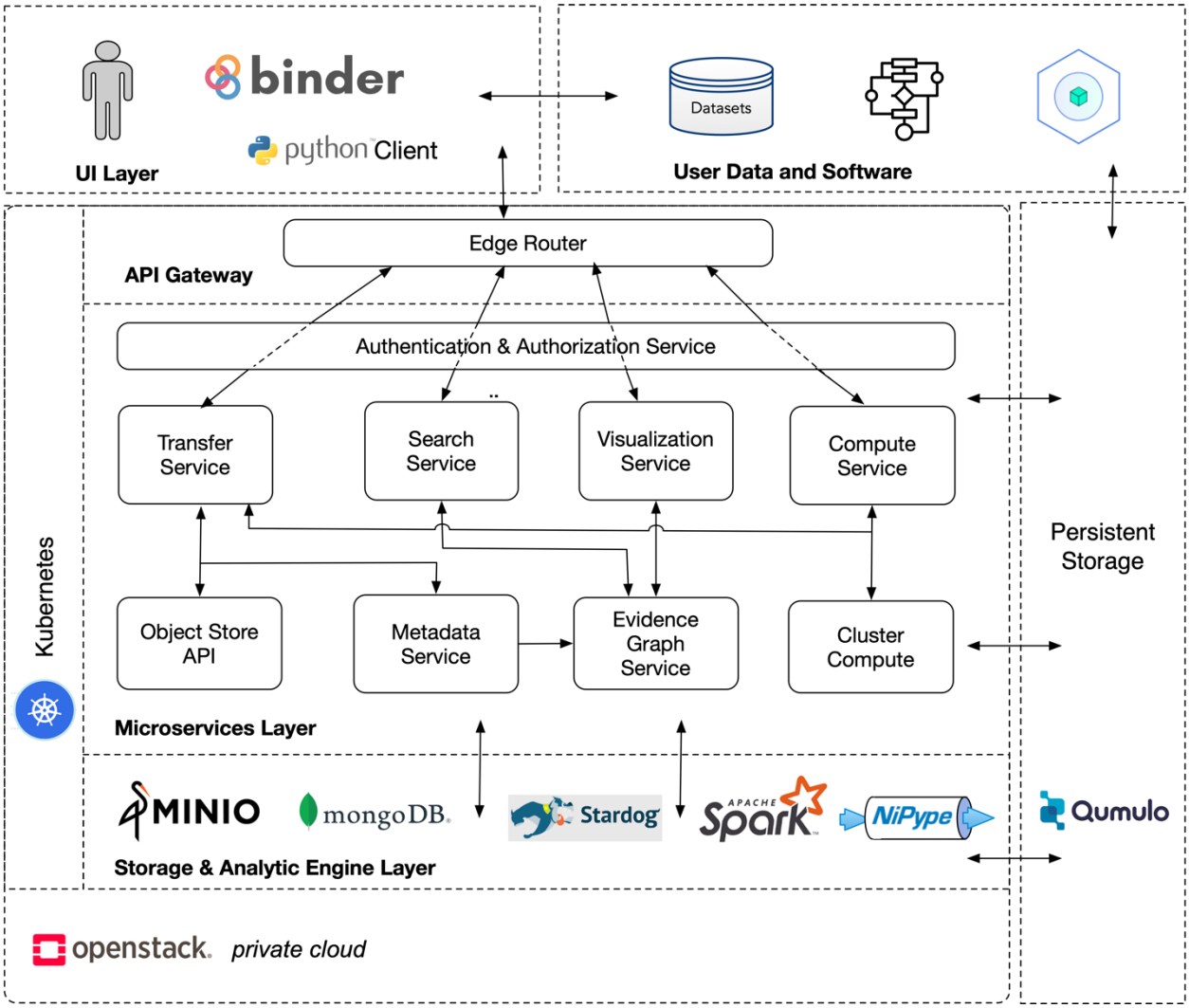


Figure 1. FAIRSCAPE architectural layers and components.

(Burns et al. 2016) in an OpenStack (Adkins 2016) private cloud environment, with a DevOps deployment model (Balalaie et al. 2016; Leite et al. 2020). An architectural sketch of this model is shown in Figure 1.

Ingress to microservices in the various layers is through a reverse proxy using an API gateway pattern. The top layer provides an interface to the end users with raw data and the associated metadata. The mid layer is a collection of tightly coupled services that allow end users with proper authorization to submit and view their data, metadata, and various types of computations performed on them. The bottom layer is built with special purpose storage and analytics platforms for storing and analyzing data, metadata and provenance information. All objects are assigned PIDs using local ARK assignment for speed, with global resolution for generality.

1.1 UI Layer

The User Interface layer in FAIRSCAPE offers end users various ways to utilize the functionalities in the framework. A reproducible interactive executable environment using Binders offers users with proper authorization the ability to use the features with ease. A Python client simplifies calls to the microservices. Data, metadata, software, scripts, workflows, containers, etc. are all are submitted and registered by the end users from the UI Layer.

1.2 API Gateway

Access to the FAIRSCAPE environment is through an API gateway, mediated by a reverse proxy. Our gateway is mediated by Traefik, which dispatches calls to the various microservices endpoints. Accessing the services

requires user authentication, which we implement using the Globus Auth authentication broker (Tuecke et al. 2016). Users of GlobusAuth may be authenticated via a number of permitted authentication services, and are issued a token which serves as an identity credential. In our current installation we require use of the CommonShare authenticator, with site-specific two-factor authentication necessary to obtain an identify token. This token is then used by the microservices to determine a user's permission to access various functionality.

1.3 Microservices Layer

The microservices layer is composed of seven services and two interfaces: Authentication, Authorization, Transfer, Metadata, Evidence, Computation, Search, and Visualization services; and the Object and Cluster Compute APIs to lower level services.

1.4 Storage and Analytic Engine Layer

FAIRSCAPE currently uses MinIO for object storage, MongoDB for basic metadata storage, and Stardog for graph storage. Computations are managed by Kubernetes, Apache SPARK, and the Nipype neuroinformatics workflow engine.

2. FAIRSCAPE Microservice Components

2.1 Transfer Service

This service transfers and registers digital research objects - datasets, software, etc., - and their associated metadata, to the Commons. These objects are sent to the transfer service as binary data streams, which are then stored in MinIO object storage. These objects may include structured or unstructured data, application software, workflow, scripts. The associated metadata contains essential descriptive information such as context, type, name, textual description, author, location, checksum, etc. about these objects. Metadata are expressed as JSON-LD and sent to the Metadata Service for further processing.

Hashing is used to verify correct transmission of the object – users are required to specify a hash which is then recomputed by MinIO after the object is stored. Hash computation is currently based on the SHA-256 secure cryptographic hash algorithm (Dang 2015). Upon successful execution, the service returns a PID of the object in the form of an ARK, which resolves to the metadata. The metadata includes, as is normal in PID architecture, a link to the actual data location.

An OpenAPI description of the interface is here:

https://app.swaggerhub.com/apis/FAIRSCAPE/Transfer/0.1

2.2 Metadata Service

The Metadata Service handles metadata registration and resolution including identifier minting in association with the object metadata. The metadata service takes user POSTed JSON-LD metadata and uploads the metadata to mongoDB and Stardog, and returns a PID. To retrieve metadata for an existing PID a user makes a GET call to the service. A PUT call to the service will update an existing PID with new metadata. While other services may read from mongoDB and Stardog directly, the Metadata Service handles all writes to mongoDB and Stardog.

An OpenAPI description of the interface is here:

https://app.swaggerhub.com/apis/FAIRSCAPE/Metadata-Service/0.1

2.3 Compute Service

This service executes user uploaded scripts, workflows, or containers, on uploaded data. It currently offers two compute engines (Spark, Nipype) in addition to native Kubernetes container execution, to meet a variety of computational needs. To complete jobs the service spawns specialized pods on kubernetes designed to perform domain specific computations that can be scaled to the size of the cluster. This service provides the essential ability to recreate computations based solely on identifiers. For data to be computed on it must first be uploaded via the Transfer Service and be issued an associated PID.

The service accepts a PID for a dataset, a script, software, or a container, as input and produces a PID representing the activity to be completed. The request, if successful, returns a job identifier from which job progress can be followed. Upon completion of a job all outputs are automatically uploaded and assigned new

PIDs, with provenance aware metadata. At job termination, the service performs a ‘cleanup’ operation, where a job is removed from the queue once it is completed.

An OpenAPI description of the interface is here:

https://app.swaggerhub.com/apis/FAIRSCAPE/Compute/0.1

2.4 Visualization Service

This service allows users to visualize Evidence Graphs interactively in the form of nodes and directed edges, offering a consolidated view of the entities and the activities supporting correctness of the computed result. Our current visualization engine is Cytoscape (Shannon 2003). Each node displays its relevant metadata information, including its type and PID, resolved in real-time.

The Visualization Service renders the graph on an HTML page.

An OpenAPI description of the interface is here:

https://app.swaggerhub.com/apis/fairscape/Visualization/0.1

2.5 Evidence Graph Service:

The Evidence Graph Service creates a JSON-LD Evidence Graph of all provenance related metadata to a PID of interest. The Evidence Graph documents all entities such as datasets, software, and workflows, and the activities performed involving these entities. The service accepts a PID as its input, runs a specialized PATH query built on top of the SPARQL query engine in Stardog with the PID as its source to retrieve all supporting nodes that can be reached. To retrieve an Evidence Graph for a PID a user may make a GET call to the service.

An OpenAPI description of the interface is here:

https://app.swaggerhub.com/apis/FAIRSCAPE/Evidence-Graph/0.1

2.6 Search Service

The Search Service allows users to search for object metadata containing strings of interest. It accepts a string as input and performs a search over all literals in the metadata for exact string matches and returns a list of all PIDs with a literal containing the query string. It is invoked via the GET method of API endpoint to the service with the search string as argrument.

An OpenAPI description of the interface is here:

https://app.swaggerhub.com/apis/FAIRSCAPE/Search/0.1

3. FAIRSCAPE Service Orchestration

FAIRSCAPE orchestrates a set of containers to provide the services in these layers, using Kubernetes. The services support a pattern composed of the following steps: (a) API ingress, (b) User Authentication and Authorization, (c) Service Dispatch, (d) Object Acquisition, (e) Computation, (f) Object Resolution and Access. These steps rely on further components (g) Identifier Minting and Resolution, (h) Object Access, (i) Object Verification, and (j) Evidence Graph Visualization.

3.1 User Authentication and Authorization

The AuthN/AuthZ service authenticates users and issues them a token, which is then used at the Service level to determine what permissions they have in that Service. Metadata access is authorized separately from data access, and separately from service execution. A user may be authorized to read an object’s metadata, but not its data. This is accomplished by preventing return of the *downloadUR*L term by the metadata service, and as a second level assurance, by blocking access to the object’s S3 bucket in MinIO.

3.2 Object Acquisition

The Transfer Service provides import of an object - software, container, or dataset - into FAIRSCAPE, documenting its origin, and enabling descriptive metadata to be attached. Once the object is stored robustly, it can be computed upon. Objects are automatically registered with a persistent identifier (PID) upon acquisition. These are currently limited to Archival Resource Keys (ARKs), generated locally. We plan to enable Datacite

DOI registration shortly. This was an original feature of the Object Registration System we developed in the NIH Data Commons Pilot, however since that time, changes have been made to the Datacite API which we need to review and address in our code.

3.3 Computation

The Compute Service executes computations using either a container specified by the user, or the Apache Spark service, or the Nipype workflow engine. Objects (again, datasets, software, containers) are passed to the compute service by their PID, retrieved from the Object Store, and acted upon using the facilities indicated. At end, the result is written back to the Object Store, the Metadata Store is updated, and the Evidence Graph updates the support graph. For Nipype jobs, the metadata includes all PROV records for each step of the workflow. For Spark jobs, data from the Object Store is written to the HFS file system, which maintains a direct interface with MinIO, separate from and below the level of the Compute Service, for efficiency.

3.4 Identifier Minting and Resolution

The Metadata Service mints PIDs using the appropriate internal or external service. In the current deployment, that is local ARK minting with global resolution. Multiple alternative PIDs may exist for any object, and DOI registration is a planned near-term feature. PIDs are resolved to their associated object level metadata, including the object's Evidence Graph and location, with appropriate permissions.

3.5 Object Resolution and Access

Objects are accessed by their location, after prior resolution of the object's PID to its metadata and authorization of the user's authentication token for data access on that object. Object access is either directly from the Object Store, or from wherever else the object may reside. Certain large objects residing in robust external archives, may not be acquired into local object storage, but remain in place, up to the point of computation.

3.6 Object Verification:

Objects are issued hashes when they are created, and these hashes are also required metadata on ingress. The original user-supplied hashes are verified whenever an object is ingested, and internally computed hashes are provided for re-verification when the object is accessed.

3.7 Evidence Graph Visualization

Evidence graphs of any object acquired by the system may be visualized at any point in this workflow using the Visualization Service. Nipype provides a chart of the workflows it executes using the Graphviz package. Our Evidence Graph Service is interactive, using the Cytoscape package (Shannon 2003), and allows Evidence graphs of multiple workflows in sequence to be displayed whether or not they have been combined into a single flow.

4. Continuous Integration / Continuous Deployment

Service testing and deployment is automated following modern continuous integration / continuous deployment (CI/CD) DevOps practices. When code is committed to the Github repository, unit and integration tests are automatically invoked. If the tests are passed, automated deployment of the microservice containers is invoked using Jenkins pipelines (Soni 2015) and Helm Charts. This allows for rapid evolution of the platform with reasonable integrity. We have installed FAIRSCAPE both in a large private cloud cluster computing environment at our university, and on laptops.

Results

1. Use Case Demonstration 1: Highly Comparative Time Series Analysis of NICU Data

We used FAIRSCAPE services to analyze ten years of neonatal ICU vital signs data from over 6,000 babies with over 100 different highly comparative time series analysis (HCTSA) methods taken from the literature (Fulcher et al. 2013; Fulcher and Jones 2017), recoding many of them from Matlab into Python. We analyzed the data with operations computed using several parameter sets amounting to > 2,000 separate computations (Niestroy et al., in preparation). One key step in the analysis was to cluster the algorithms by the similarity of the results. The results were represented in the heat map shown in Figure 2.

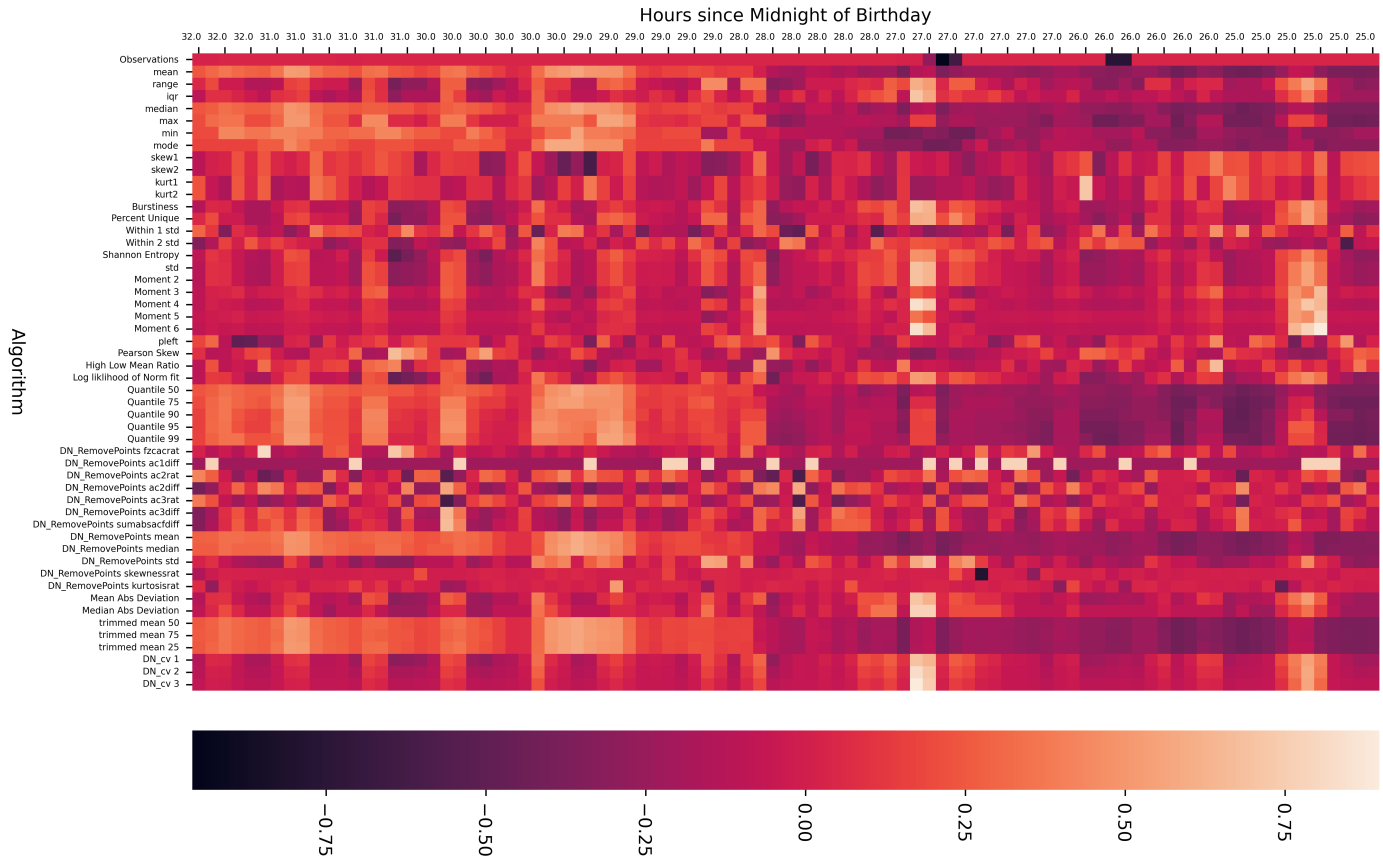


Figure 2 – NICU HCTSA computation heatmap. *X axis (top) is hours since birthday, Y axis is operations (algorithms using specific parameter sets), as representatives of result clusters.*

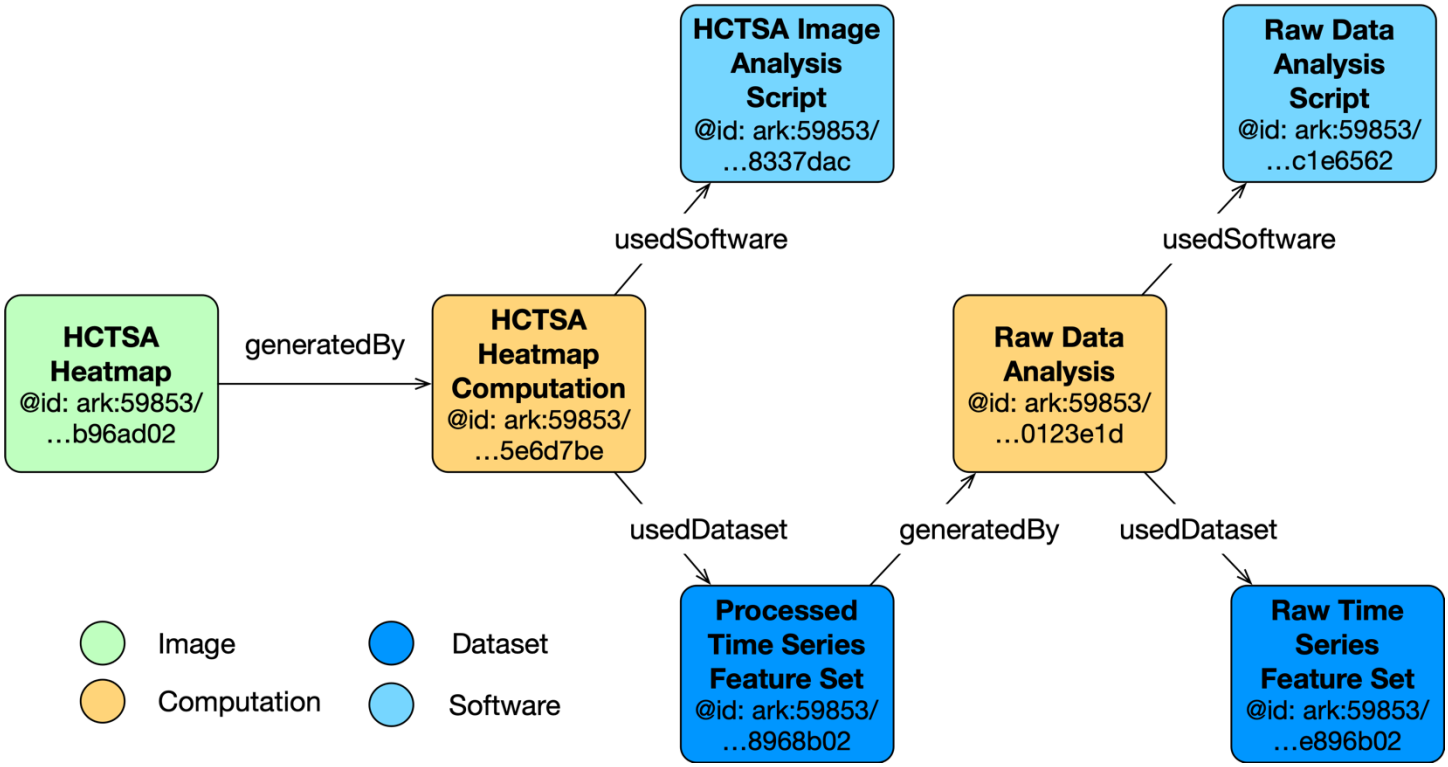


Figure 3 - Evidence graph for one patient's computations. *Vital signs = dark blue box bottom right; computations = yellow boxes; processed data = dark blue box in middle; green box = heatmap of correlations.*

The evidence graph for this result is quite large. A visualization of a section for one patient is shown in Figure 3. The full evidence graph for the clustering computation has 17,994 nodes. The JSON-LD for this Individual patient example is shown in Figure 4. Metadata for the archived image includes the JSON-LD evidence graph.


```
{'@context': {'@vocab': 'http://schema.org/', 'evi': 'http://w3id.org/EVI#'},
 '@id': 'ark:59853/2c94ffb3-8c5e-41d1-a305-d3bf6b96ad02',
 '@type': 'Dataset',
 'evi:generatedBy': {'@id': 'ark:59853/f13a2b45-17fd-45b2-8c5f-8a5275e6d7be',
  '@type': 'evi:Computation',
  'began': 'Saturday, August 01, 2020 11:33:38',
  'evi:usedDataset': {'@id': 'ark:59853/513033a6-1d31-4518-87b2-6b8693e165e6',
   '@type': 'Dataset',
   'evi:generatedBy': {'@id': 'ark:59853/1605848e-bdad-4842-850a-2b2180123e1d',
    '@type': 'evi:Computation',
    'began': 'Saturday, August 01, 2020 11:31:17',
    'evi:usedDataset': {'@id': 'ark:59853/d450914a-417e-44a5-9ef2-2741e8968b02',
     '@type': 'Dataset',
     'author': {'@id': 'https://orcid.org/0000-0002-1103-3882',
      '@type': 'Person',
      'name': 'Justin Niestroy'},
     'name': 'Raw Data'},
    'evi:usedSoftware': {'@id': 'ark:59853/f03ab4d2-e3d8-490f-8c70-5aade8337dac',
     '@type': 'SoftwareSourceCode',
     'author': {'@id': 'https://orcid.org/0000-0002-1103-3882',
      '@type': 'Person',
      'name': 'Justin Niestroy'},
     'name': 'Processing  Script'},
    'name': 'Computation'},
   'name': 'part-00000-4c704c95-fe5b-460c-b1e5-6bc4f1719cde-c000.csv'},
  'evi:usedSoftware': {'@id': 'ark:59853/13022344-010a-4add-a1f3-7a975c1e6562',
   '@type': 'SoftwareSourceCode',
   'author': {'@id': 'https://orcid.org/0000-0002-1103-3882',
    '@type': 'Person',
    'name': 'Justin Niestroy'},
   'name': 'Image Script'},
  'name': 'Computation'},
 'name': 'Histogram_Heatmap.png'}
```


Figure 4 – JSON-LD Evidence Graph for patient computations graphically illustrated in Figure 3.

In this set of computations, all steps required authentication and authorization within the University of Virginia computing infrastructure. We then used the following service calls to do the analysis:

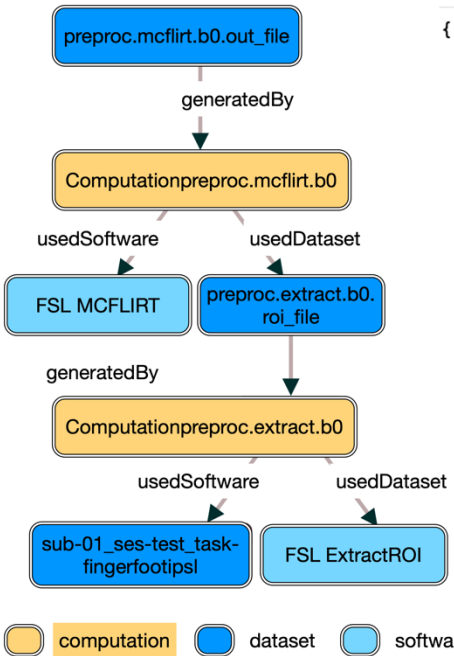
(a) Transfer Service to register all the objects with metadata and PIDs; (b) Compute Service to perform the individual computations, using Apache Spark; (c) Evidence Graph Service to compute and retrieve the Evidence Graph and create the visualization. Internally, services call each other in a more complex way, but this is masked from the user. For example, Transfer Service calls the Metadata Service to mint identifiers and register metadata, and it performs object verification against the inbound SHA256 hash.

2. Use Case Demonstration 2: Neuroimaging Analysis Using Nipype Workflow Engine

We ran neuroimaging workflows using test data provided for the Nipype workflow engine (Gorgolewski et al. 2011). Metadata for the archived computational result includes this evidence graph. A visualization of the evidence graph is shown in Figure 5. Intermediate results for such workflows have time-limited utility. Per Data Citation guidelines (Data Citation Synthesis Group 2014; Fenner et al. 2019; Starr et al. 2015), it is acceptable to clear this data if the useful metadata describing the procedure is preserved, which we do here. The service calls to perform this work were similar to those in Use Case 1 above, with the exception that the Compute Service was called using the Nipype option.

Discussion

Scientific rigor depends on the transparency of methods and materials. The historian of science Steven Shapin, described the approach developed with the first scientific journals as “virtual witnessing” (Shapin 1984), and this is still valid today. The typical scientific reader does not actually reproduce the experiment but is invited to review mentally every detail of how it was done to the extent that s/he becomes a “virtual witness”




```
{'@context': {'@vocab': 'http://schema.org/', 'evi': 'http://w3id.org/EVI#'},
 '@id': 'ark:99999/facac4e1-455a-4100-b7e6-8dc6e83980d3',
 '@type': 'Dataset',
 'evi:generatedBy': {'@id': 'ark:99999/8a9c39d2-b45c-4930-ad25-6a65f5c4b4d6',
  '@type': 'evi:Computation',
  'evi:usedDataset': {'@id': 'ark:99999/aae87d38-91be-4895-9131-ac355766825f',
   '@type': 'Dataset',
   'evi:generatedBy': {'@id': 'ark:99999/837e6dfd-de95-4bce-85d1-dab256feb6c3',
    '@type': 'evi:Computation',
    'evi:usedDataset': {'@id': 'ark:99999/79f96998-2b5d-48e6-8fd8-0b7dc0cacac1',
     '@type': 'Dataset',
     'author': 'Michael Notter',
     'name': 'sub-01_ses-test_task-fingerfootlips_bold.nii.gz'},
    'evi:usedSoftware': {'@id': 'ark:99999/2e574e91-bcdf-434f-bb82-cc734169c7b6',
     '@type': 'SoftwareSourceCode',
     'name': 'FSL ExtractROI'},
    'name': 'Computation preproc.extract.b0'},
   'name': 'preproc.extract.b0.roi_file'},
  'evi:usedSoftware': {'@id': 'ark:99999/86b6dce5-c085-4dc5-88b5-39f3276f49f0',
   '@type': 'SoftwareSourceCode',
   'name': 'FSL MCFLIRT'},
  'name': 'Computation preproc.mcflirt.b0'},
 'name': 'preproc.mcflirt.b0.out_file'}
```


Figure 5 - Evidence Graph visualization for the neuroimaging workflow execution.

to an envisioned live demonstration. That is clearly how most people read scientific papers - except perhaps when they are citing them, in which case less care is often taken. Scientists are not really incentivized to replicate experiments; their discipline rewards novelty. The ultimate validation of any claim once it has been accepted as reasonable on its face comes with support from multiple distinct angles, by different investigators, and with re-use of the materials and methods upon which it is based. If the materials and methods are sufficiently transparent and thoroughly disclosed as to be reusable, and they cannot be made to work, or give bad results, that debunks the original experiments - precisely the way in which the promising-sounding STAP phenomenon was discredited (Haruko Obokata, Sasai, et al. 2014), before the elaborate formal effort of Riken to replicate the experiments.

As a first step then, it is not only a matter of reproducing experiments but also of producing transparent evidence that the experiments have been done correctly. This permits challenges to the procedures to develop over time, especially through re-use of materials (including data) and methods - which today significantly include software and computing environments. We definitely view these methods as being extensible to materials such as reagents, using the RRID approach (Prager et al. 2018).

Conclusion

FAIRSCAPE is a reusable framework for biomedical computations that provides a simplified interface for research users to an array of modern, dynamically scalable, cloud-based componentry. Our goal in developing FAIRSCAPE was to provide an ease-of-use (and re-use) incentive for researchers, while rendering all the artifacts marshalled to produce a result, and the evidence supporting them, Findable, Accessible, Interoperable, and Reusable. FAIRSCAPE can be used to construct, as we have done, a provenance-aware computational data lake or Commons. It supports transparent disclosure of the Evidence Graphs of computed results, with access to the persistent identifiers of the cited data or software, and to their stored metadata.

We plan several enhancements in future research and development with this project, including support for DOI and Software Heritage identifier registration, metadata transfer to Dataverse instances, and integration of the Galaxy workflow engine for genomic analysis, for release later this year.

Many efforts involving overlapping groups have attempted to address parts of this problem, which is in large part an outcome of the transition of biomedical and other scientific research from print to digital, and our increasing ability to generate data and to compute on it at enormous scale. We make use of many of these in our FAIRSCAPE framework, providing an integrated model for FAIRness and reproducibility, with ease of use

incentives. We believe it will be a helpful tool for constructing provenance-aware computational data lakes, and data commons, as part of an interoperating model for reproducible biomedical science.

Information sharing statement

- Code for the microservices and the python client described in this paper are publicly available on GitHub under the MIT open source license at https://github.com/fairscape
- The MongoDB noSQL DB Community Version is available under MongoDB's license terms at https://www.mongodb.com/try/download/community .
- The Stardog knowledge graph DB is available under Stardog's academic license terms at https://www.stardog.com/academic-trial/ .
- The EVI ontology is available at https://w3id.org/EVI

REFERENCES

Adkins, S. (2016). *OpenStack: Cloud Application Development*. Indianapolis, IN: Wrox. http://RE5QY4SB7X.search.serialssolutions.com/?V=1.0&L=RE5QY4SB7X&S=JCs&C=TC0001588151&T=marc

Aizawa, S. (2016). Results of an attempt to reproduce the STAP phenomenon. *F1000Research*, *5*, 1056. https://doi.org/10.12688/f1000research.8731.2

Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture. *IEEE Software*, *33*(3), 42–52. https://doi.org/10.1109/MS.2016.64

Bandrowski, A. (2014). RRID's are in the wild! Thanks to JCN and PeerJ. *The NIF Blog: Neuroscience Information Framework*. http://blog.neuinfo.org/index.php/essays/rrids-are-in-the-wild-thanks-to-jcn-and-peerj

Bandrowski, A. E., & Martone, M. E. (2016). RRIDs: A Simple Step toward Improving Reproducibility through Rigor and Transparency of Experimental Methods. *Neuron*, *90*(3), 434–436. https://doi.org/10.1016/j.neuron.2016.04.030

Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes. *Communications of the ACM*, *59*(5), 50–57. https://doi.org/10.1145/2890784

Cayrol, C., & Lagasquie-Schiex, M.-C. (2009). Bipolar Abstract Argumentation Systems. In I. Rahwan & G. R. Simari (Eds.), *Argumentation in Artificial Intelligence*. Dordrecht: Springer.

Cayrol, C., & Lagasquie-Schiex, M.-C. (2010). Coalitions of arguments: A tool for handling bipolar argumentation frameworks. *International Journal of Intelligent Systems*, *25*(1), 83–109. https://doi.org/10.1002/int.20389

Cayrol, C., & Lagasquie-Schiex, M.-C. (2013). Bipolarity in argumentation graphs: Towards a better understanding. *International Journal of Approximate Reasoning*, *54*(7), 876–899. https://doi.org/10.1016/j.ijar.2013.03.001

Clark, Tim, Ciccarese, P., & Goble, C. (2014). Micropublications: a semantic model for claims, evidence, arguments and annotations in biomedical communications. *Journal of Biomedical Semantics*, *5*(1). http://www.jbiomedsem.com/content/5/1/28

Clark, Timothy, Katz, D. S., Bernal Llinares, M., Castillo, C., Chard, K., Crosas, M., et al. (2018, September 3). DCPPC DRAFT: KC2 Globally Unique Identifier Services. National Institutes of Health, Data Commons Pilot Phase Consortium. https://public.nihdatacommons.us/DCPPC-DRAFT-8_KC2/

Cousijn, H., Kenall, A., Ganley, E., Harrison, M., Kernohan, D., Lemberger, T., et al. (2018). A data citation roadmap for scientific publishers. *Scientific data*, *5*, 180259.

Dang, Q. H. (2015). *Secure Hash Standard* (No. NIST FIPS 180-4) (p. NIST FIPS 180-4). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.FIPS.180-4

Data Citation Synthesis Group. (2014). *Joint Declaration of Data Citation Principles*. San Diego CA: Future of Research Communication and e-Scholarship (FORCE11). https://doi.org/10.25490/a97f-egyk

Deer, B. (2011). How the case against the MMR vaccine was fixed. *BMJ*, *342*(jan05 1), c5347–c5347. https://doi.org/10.1136/bmj.c5347

Fenner, M. (2019). Tracking the Growth of the PID Graph. https://doi.org/10.5438/BV9Z-DC66

Fenner, M., Clark, T., Katz, D., Crosas, M., Cruse, P., Kunze, J., & Wimalaratne, S. (2018, July 23). Core Metadata for GUIDs. National Institutes of Health, Data Commons Pilot Phase Consortium. https://public.nihdatacommons.us/DCPPC-DRAFT-7_KC2/

Fenner, M., Crosas, M., Grethe, J. S., Kennedy, D., Hermjakob, H., Rocca-Serra, P., et al. (2019). A data citation roadmap for scholarly data repositories. *Scientific Data*, *6*(1), 28. https://doi.org/10.1038/s41597-019-0031-8

Fulcher, B. D., & Jones, N. S. (2017). hctsa : A Computational Framework for Automated Time-Series Phenotyping Using Massive Feature Extraction. *Cell Systems*, *5*(5), 527-531.e3. https://doi.org/10.1016/j.cels.2017.10.001

Fulcher, B. D., Little, M. A., & Jones, N. S. (2013). Highly comparative time-series analysis: the empirical structure of time series and their methods. *Journal of The Royal Society Interface*, *10*(83), 20130048. https://doi.org/10.1098/rsif.2013.0048

Gil, Y., Miles, S., Belhajjame, K., Deus, H., Garijo, D., Klyne, G., et al. (2013, April 30). PROV Model Primer: W3C Working Group Note 30 April 2013. World Wide Web Consortium (W3C). https://www.w3.org/TR/prov-primer/

Gorgolewski, K., Burns, C. D., Madison, C., Clark, D., Halchenko, Y. O., Waskom, M. L., & Ghosh, S. S. (2011). Nipype: A flexible, lightweight and extensible neuroimaging data processing framework. *Frontiers in Neuroinformatics*, *5*. https://doi.org/10.3389/fninf.2011.00013

Greenberg, S. A. (2009). How citation distortions create unfounded authority: analysis of a citation network. *British Medical Journal*, *339*, b2680. https://doi.org/10.1136/bmj.b2680

Greenberg, S. A. (2011). Understanding belief using citation networks. *Journal of Evaluation in Clinical Practice*, *17*(2), 389–393. https://doi.org/10.1111/j.1365-2753.2011.01646.x

Grossman, R. L. (2018). Progress Toward Cancer Data Ecosystems: *The Cancer Journal*, *24*(3), 126–130. https://doi.org/10.1097/PPO.0000000000000318

Grossman, R. L. (2019). Data Lakes, Clouds, and Commons: A Review of Platforms for Analyzing and Sharing Genomic Data. *Trends in Genetics*, *35*(3), 223–234. https://doi.org/10.1016/j.tig.2018.12.006

Groth, P., Cousijn, H., Clark, T., & Goble, C. (2020). FAIR Data Reuse – the Path through Data Citation. *Data Intelligence*, *2*(1–2), 78–86. https://doi.org/10.1162/dint_a_00030

Ishii, S., Iwama, A., Koseki, H., Shinkai, Y., Taga, T., & Watanabe, J. (2014). *Report on STAP Cell Research Paper Investigation* (p. 11). Saitama, JP: RIKEN. http://www3.riken.jp/stap/e/f1document1.pdf

Jensen, M. A., Ferretti, V., Grossman, R. L., & Staudt, L. M. (2017). The NCI Genomic Data Commons as an engine for precision medicine. *Blood*, *130*(4), 453–459. https://doi.org/10.1182/blood-2017-03-735654

Jupyter, P., Bussonnier, M., Forde, J., Freeman, J., Granger, B., Head, T., et al. (2018). Binder 2.0 - Reproducible, interactive, sharable environments for science at scale (pp. 113–120). Presented at the Python in Science Conference, Austin, Texas. https://doi.org/10.25080/Majora-4af1f417-011

Kluyver, T., Ragan-Kelley, B., Pérez, F., Bussonnier, M., Frederic, J., Hamrick, J., et al. (2016). Jupyter Notebooks—a publishing format for reproducible computational workflows. In *Positioning and Power in*

Academic Publishing: Players, Agents and Agendas (p. 4). IOS Press. https://ebooks.iospress.doi:nl/publication/42900

Larrucea, X., Santamaria, I., Colomo-Palacios, R., & Ebert, C. (2018). Microservices. *IEEE Software*, *35*(3), 96–100. https://doi.org/10.1109/MS.2018.2141030

Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., et al. (2013). PROV-O: The PROV Ontology W3C Recommendation 30 April 2013. http://www.w3.org/TR/prov-o/

Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2020). A Survey of DevOps Concepts and Challenges. *ACM Computing Surveys*, *52*(6), 1–35. https://doi.org/10.1145/3359981

Lewis, J., & Fowler, M. (2014, March 25). Microservices: a definition of this new architectural term. MartinFowler.com. https://martinfowler.com/articles/microservices.html#ProductsNotProjects

Mehra, M. R., Desai, S. S., Kuy, S., Henry, T. D., & Patel, A. N. (2020). Retraction: Cardiovascular Disease, Drug Therapy, and Mortality in Covid-19. N Engl J Med. DOI: 10.1056/NEJMoa2007621. *New England Journal of Medicine*, NEJMc2021225. https://doi.org/10.1056/NEJMc2021225

Mehra, Mandeep R, Ruschitzka, F., & Patel, A. N. (2020). Retraction—Hydroxychloroquine or chloroquine with or without a macrolide for treatment of COVID-19: a multinational registry analysis. *Lancet*, *published online*.

Moreau, L., Missier, P., Belhajjame, K., B'Far, R., Cheney, J., Coppens, S., et al. (2013). *PROV-DM: The PROV Data Model: W3C Recommendation 30 April 2013*. World Wide Web Consoritum. http://www.w3.org/TR/prov-dm/

Obokata, H., Sasai, Y., Niwa, H., Kadota, M., Andrabi, M., Takata, N., et al. (2014). Bidirectional developmental potential in reprogrammed cells with acquired pluripotency. *Nature*, *505*(7485), 676–80. https://doi.org/10.1038/nature12969

Obokata, Haruko, Sasai, Y., Niwa, H., Kadota, M., Andrabi, M., Takata, N., et al. (2014). Retraction Note: Bidirectional developmental potential in reprogrammed cells with acquired pluripotency. *Nature*, *511*(7507), 112–112. https://doi.org/10.1038/nature13599

Obokata, Haruko, Wakayama, T., Sasai, Y., Kojima, K., Vacanti, M. P., Niwa, H., et al. (2014). Stimulus-triggered fate conversion of somatic cells into pluripotency. *Nature*, *505*(7485), 641–647. https://doi.org/10.1038/nature12968

Prager, E. M., Chambers, K. E., Plotkin, J. L., McArthur, D. L., Bandrowski, A. E., Bansal, N., et al. (2018). Improving transparency and scientific rigor in academic publishing. *Brain and Behavior*, e01141. https://doi.org/10.1002/brb3.1141

Shannon, P. (2003). Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, *13*(11), 2498–2504. https://doi.org/10.1101/gr.1239303

Shapin, S. (1984). Pump and Circumstance: Robert Boyle's Literary Technology. *Social Studies of Science*, *14*(4), 481–520. http://sss.sagepub.com/content/14/4/481.abstractN2

Smith, A. M., Katz, D. S., Niemeyer, K. E., & FORCE11 Software Citation Working Group. (2016). Software citation principles. *PeerJ Computer Science*, *2*, e86. https://doi.org/10.7717/peerj-cs.86

Soni, M. (2015). *Jenkins Essentials: Continuous Integration, Setting Up the Stage for a DevOps Culture*. S.l.: Packt Publishing. http://RE5QY4SB7X.search.serialssolutions.com/?V=1.0&L=RE5QY4SB7X&S=JCs&C=TC0001567728&T=marc

Starr, J., Castro, E., Crosas, M., Dumontier, M., Downs, R. R., Duerr, R., et al. (2015). Achieving human and machine accessibility of cited data in scholarly publications. *PeerJ. Computer Science*, *1*. https://doi.org/10.7717/peerj-cs.1

The Editors of The Lancet. (2010). Retraction—Ileal-lymphoid-nodular hyperplasia, non-specific colitis, and pervasive developmental disorder in children. *The Lancet*, *375*(9713), 445. https://doi.org/10.1016/S0140-6736(10)60175-4

Tuecke, S., Ananthakrishnan, R., Chard, K., Lidman, M., McCollam, B., Rosen, S., & Foster, I. (2016). Globus auth: A research identity and access management platform. In *2016 IEEE 12th International Conference on e-Science (e-Science)* (pp. 203–212). Presented at the 2016 IEEE 12th International Conference on e-Science (e-Science), Baltimore, MD, USA: IEEE. https://doi.org/10.1109/eScience.2016.7870901

Wakefield, A., Murch, S., Anthony, A., Linnell, J., Casson, D., Malik, M., et al. (1998). RETRACTED: Ileal-lymphoid-nodular hyperplasia, non-specific colitis, and pervasive developmental disorder in children. *The Lancet*, *351*(9103), 637–641. https://doi.org/10.1016/S0140-6736(97)11096-0

Wan, X., Guan, X., Wang, T., Bai, G., & Choi, B.-Y. (2018). Application Deployment Using Microservice and Docker Containers: Framework and Optimization. *Journal of Network and Computer Applications*, *119*, 97–109. https://doi.org/10.1016/j.jnca.2018.07.003

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*, 160018. https://doi.org/10.1038/sdata.2016.18

Information Sharing Statement

All code developed for this framework is available at https://github.com/fairscape under MIT license (https://opensource.org/licenses/MIT).

Acknowledgements

We thank Satra Ghosh, Maryann Martone, John Kunze, Neal Magee, and Chris Baker, for several helpful discussions; and Neal Magee for technical assistance with the University of Virginia computing infrastructure. This work was supported in part by the U.S. National Institutes of Health, grants NIH OT3 OD025456-01 and NIH 1U01HG009452; and by a grant from the Coulter Foundation.

Author Information

University of Virginia School of Medicine
Department of Public Health Sciences (Biomedical Informatics)
Maxwell Adam Levinson, ORCiD: 0000-0003-0384-8499
Sadnan Al Manir, ORCiD: 0000-0003-4647-3877
Justin Niestroy, ORCiD: 0000-0002-1103-3882
Timothy Clark, ORCiD: 0000-0003-4060-7360

Department of Medicine
Douglas E. Lake, ORCiD: 0000-0001-6259-4850
J. Randall Moorman, ORCiD: 0000-0002-5772-1648

Department of Pediatrics
Karen Fairchild, ORCiD: 0000-0002-1081-8741

University of Virginia School of Data Science
Tim Clark, ORCiD: 0000-0003-4060-7360

University of Virginia College and Graduate School of Arts and Sciences
Department of Statistics
Douglas E. Lake, ORCiD: 0000-0001-6259-4850

Corresponding Author
correspondence to Tim Clark, twclark@virginia.edu

Ethics Declarations
Conflict of interests

The authors declare that they have no conflicts of interest.

Additional Information

- Data used in preparing this article was obtained from the University of Virginia Center for Advanced Medical Analytics and from OpenNeuro.org

Rights and Permissions

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate

credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third-party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.