[CP-02-009] Science Gateways

Abstract

Science gateways are a key driver of the democratization of access to computing and data resources in science and engineering research. Science gateway technology has evolved in tandem with modern web technologies and adopts several standard design principles that have accelerated their development. The evolution of science gateways and their most common components are discussed, followed by some examples of popular gateways adopted by the GIS community. The challenges faced by modern gateways in response to user needs and evolving technologies are presented to drive further discussion and development.

Keywords: geoportal, web services

Author & citation

Kalyanam, R. and Song, C. (2023). Science Gateways. The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2023 Edition), John P. Wilson (ed.).
DOI: 10.22224/gistbok/2023.4.1.

Explanation

1. Definitions
2. Introduction
3. Evolution of Gateways
4. Gateway Components
5. Role of Gateways in Research and Education
6. GIS Gateways
7. Modern Gateway Concerns

1. Definitions

1. Content Management System (CMS): a software system that standardizes, enables, and supports the management of user generated content
2. Application Programming Interface (API): a software interface for computer systems to communicate with one another to exchange data
3. Web Service: a service that serves content using web communication standards such as HTTP using standard formats such as JSON, XML, etc.
4. Cloud Computing: a service that provides on-demand access to compute and data resources without the user needing to manage these resources
5. Container: a means of virtualization of compute and data that is typically used to package software and its dependencies for ease of portability
6. Web 2.0: a web standard that focuses on user generated content and community-based computing
7. NoSQL: a database technology that does not use relational tables which traditionally support query via the structured query language (SQL) and is better suited to support a large amount of heterogeneous data types such as wide columns, key-value data,

documents, and graphs.
8. MVC (Model-View-Controller): a software design pattern that separates the data management business logic (model) from the web data presentation (view) and user control-based actions (controller)
9. Cloud-optimized: data formats that are conducive to computation on the cloud
10. FAIR: a set of guiding principles for data management that enable data reuse and reproducibility comprising four criteria: Findable, Accessible, Interoperable, and Reusable
11. DOI (Digital Object Identifier): a unique reference comprised of a string of letters and numbers for a digital object

2. Introduction

Science gateways allow science and engineering communities to access shared data, software, computing services, instruments, educational materials, and other resources specific to their disciplines via easy-to-use interfaces (Wilkins-Diehr 2007). Science gateways are also known by various terms such as virtual research environment (VRE) or portal or e-Science gateway or simply gateway, depending on the domain or region of the world. While specific terms may imply certain specific design patterns, for instance, a VRE or e-Science gateway may imply that networking is used to link together various labs or computational resources or that there is a focus on data sharing and interoperability at their core, these various terms refer to a means of enabling shared access to data, computation, software, and other resources.

3. Gateways Evolution

Gateways began as purpose-built websites designed using common web technologies that did not lend themselves to replication or re-usability. The first major step in the evolution of gateways was the rise in “portal” frameworks that enabled the development of standardized and modular portal components termed “portlets.” Portals essentially served as containers for one or more portlets that implemented various gateway features such as user management, data management, and computation management. While portal frameworks such as Gridsphere (Novotny, Russell, and Wehrens 2004) greatly simplified the development and deployment of science gateways, these were still entirely designed and developed by the gateway developer and did not support easy customization or resource contribution by the end user.

Around this time, the rise in web services and their standardization via the WSDL (Web Service Definition Language) format, and the SOAP (Simple Object Access Protocol) data transfer protocol led to the rise in portals that were designed around service consumption. Gateway developers no longer needed to develop all the gateway tools themselves, but could instead develop portlets that consumed web services which implemented the necessary business logic.

Gateways that enabled greater user customization and resource contribution grew out of the rise of content management systems (CMS) such as Joomla, Liferay, Symfony, etc. These primarily PHP-based frameworks could be deployed on a web-server such as Apache HTTP and enforced MVC (Model-View-Controller) design patterns that enabled modularity and reusability of key gateway components, as well as development of user-friendly views using a combination of modern web 2.0 technologies including HTML5, CSS, and Javascript.

Furthermore, these CMS fostered user engagement with the gateway through both ease of resource contribution as well as through the inclusion of community features such as message boards, wikis, user groups, etc. The popular HUBzero framework (McLennan and Kennell 2020) based on Joomla grew into a standalone science gateway framework by abstracting out the domain-specific functionality developed for the nanoHUB gateway (Klimeck et al. 2008) that supported research and education in nanotechnology. HUBzero quickly grew to support more than sixty gateways in diverse domains such as biosciences, natural hazards management, geosciences, pharmacy, and computational chemistry. A key innovation in HUBzero was the bridging together of the web CMS with scientific tools that could be built on Linux and served via the gateway web interface. This enabled researchers to carry out collaborative, end-to-end research workflows on the gateway platform, reducing the barrier to accessing compute, data, and software. From an implementation perspective however, CMS-based gateways still required significant effort towards deployment, configuration, and upkeep due to the various building blocks involved (web-server, databases, PHP), the various software versions available, and operating system compatibility.

The next major set of innovations that addressed some of these implementation difficulties while providing a new generation of capabilities was the rise in containers, container orchestration frameworks such as Kubernetes, and cloud computing. Gateway building blocks could now be encapsulated into containers or microservices and deployed in a scalable manner on a variety of commercial and on-premises cloud facilities. Gateways could also leverage other cyberinfrastructure capabilities that could be deployed alongside in the same cloud such as serverless computing, message queues, big data processing frameworks such as Hadoop and Spark, and interactive computing platforms such as Jupyter. At the same time, the rise in high level web frameworks such as Django, AngularJS, React, and NodeJS have greatly simplified API generation, web development, and portal development.

Alongside this technological evolution that simplified gateway development and deployment, a significant factor in the adoption of gateways was the establishment of a community of gateway developers and users. This was assisted through the efforts of the XSEDE (Towns et al. 2014) national computational ecosystem and the Science Gateways Community Institute (SGCI) (Gesing et al. 2019). XSEDE provided the computational resources to support resource-intensive computations from science gateways and also longer-term development and integration support through its Extended Collaborative Support Services (ECSS) program. SGCI was instrumental in understanding the needs of the research community through surveys and focus groups and also facilitated collaboration among the gateway developer community through its annual conferences. SGCI has also assisted with improving the usability of gateways through UX consulting, and conducted bootcamps that enabled teams of gateway users and developers to flesh out the value proposition and sustainability strategies for their gateways.

4. Gateway Components

Modern gateways comprise several distinct components (Figure 1): the web gateway interface, gateway middleware, and finally the data and computational resources that implement gateway operations and are made available to users through the gateway. The web interface as described previously has evolved through various technological phases and currently utilizes a combination of MVC frameworks and web 2.0 technologies. Modern

gateways also incorporate some means of authentication as well as resource authorization, access control, and delegation. Modern standards such as OAuth and federated authentication frameworks such as CILogon (Basney et al. 2019) are typically leveraged to enable users to reuse their institutional credentials and federate across multiple services using the same authorization infrastructure.

Figure 1.

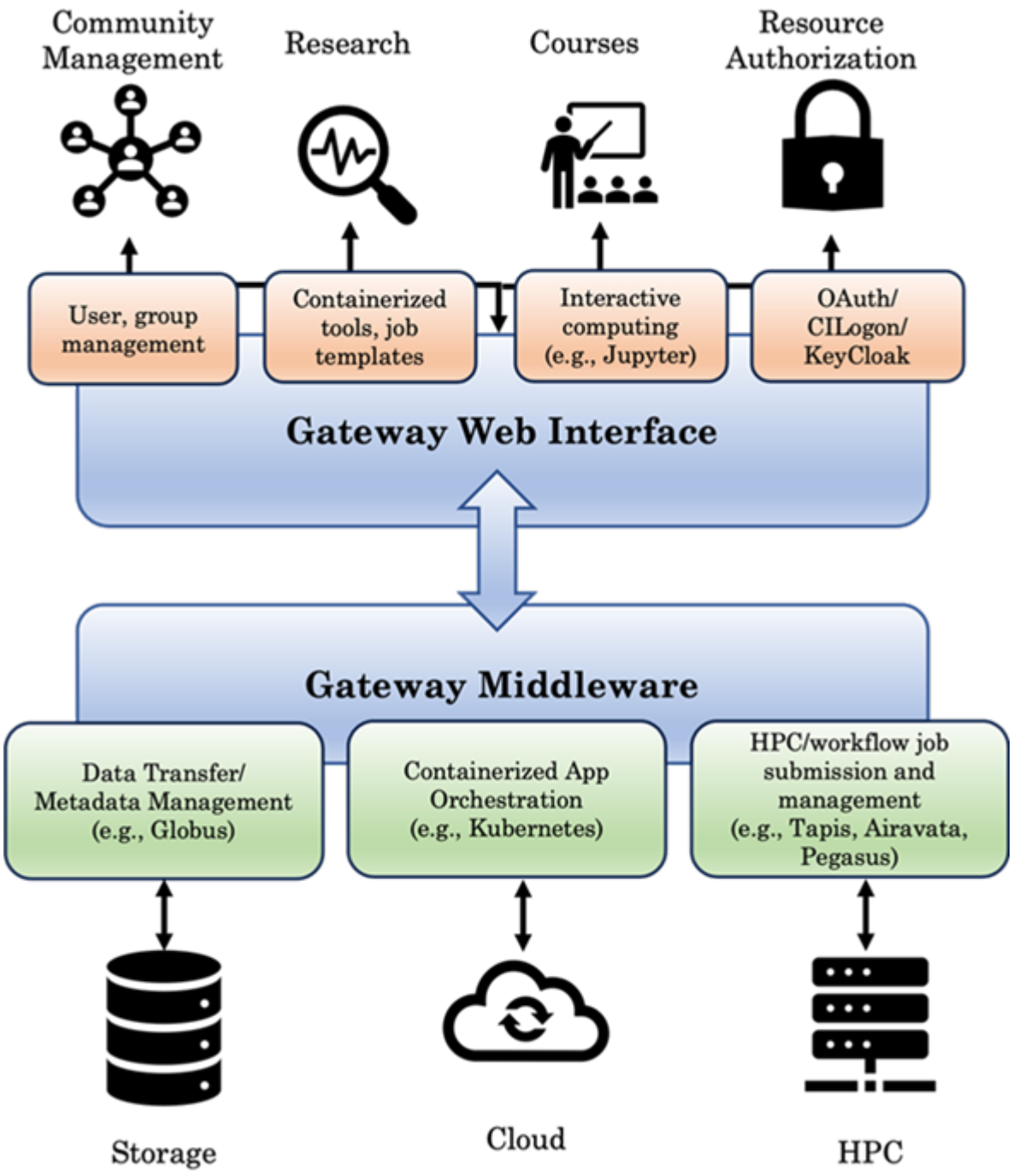


Figure 1. Image source: authors.

Gateway middleware enables gateways to easily integrate with external services that provide data management and computing capabilities. Data management typically includes

data transfer to and from remote shared data stores, high-performance computing (HPC) systems, and both relational and noSQL databases. Middleware such as Globus (Foster 2011) are used to efficiently transfer large datasets, while also managing metadata for future data discovery. Other gateway middleware such as Apache Airavata (Marru et al. 2011) or Tapis (Stubbs et al. 2021) are used to integrate gateways with computational resources including campus and the national HPC ecosystem (XSEDE, and its follow-on ACCESS (accessci.org), as well as commercial cloud. These middleware implement tasks such as job submission, monitoring, data transfer, and results retrieval. Several gateway platforms also enable users to compose together individual computational tasks into workflows. Workflow management systems such as Pegasus (Deelman et al. 2015), Kepler (Altintas et al. 2004), Taverna (Oinn et al. 2004), etc. enable high-level workflow composition as well as workflow submission and monitoring and integrate with various execution environments.

Most gateways employ a database as part of the MVC framework's model and to manage various gateway configurations. Most gateways also provide block and file storage to manage user uploaded datasets as well as other shared datasets from the specific domains. Some gateways might also provide other storage infrastructure such as object storage depending on the domain and supported applications. Finally, some gateways may provide big data stores and associated computational infrastructure such as Hadoop and Apache Spark. In the computational realm, gateway tools may simply utilize server-based computing or containers to execute domain-specific codes. Most modern gateways however provide some integration with HPC resources for resource intensive computations.

5. Role of Gateways in Research and Education

One measure for the role and impact of gateways in research and education is the growing number of gateways as well as the publications citing them. The Science Gateways Community Institute (SGCI) currently lists 539 gateways in its catalog split across various science and engineering domains, while a search for the term "science gateway" in Google Scholar yields nearly 17,500 results. Moreover, while there are nearly 6400 results for the term "science gateway" in publications between 2010 and 2017, there are 10,300 results for the period between 2018 and today, pointing to a steady increase in gateway-related publications. A review of the top relevant "science gateway" publications reported by Google Scholar since 2018 reveals that the gateways originate from diverse domains such as neuroscience, bioinformatics, agriculture, GIS, chemistry, and social sciences. Gateways have also had sustained impact in various domains as evidenced by long-running and popular gateways such as nanoHUB (Klimeck et al. 2008) and CIPRES (Miller, Pfeiffer, and Schwartz 2011) that originated in the early 2000's and now have close to 300 publications citing them.

Gateways today are being used for not just research but also educational and training activities. With the growth in distance and online education, gateways serve as an important resource for educators to host their learning materials in a web-accessible platform that can be accessed globally. Containerization, cloud computing, and interactive computing platforms such as Jupyter and R Studio further facilitate scalable, experiential learning that can improve educational outcomes compared to static web-based teaching materials (Hanc et al. 2020).

6. GIS Gateways

In the GIS domain, gateways are designed to enable researchers to carry out end-to-end workflows without having to resort to a mix of desktop GIS tools, gateway tools, and HPC resources. Consequently, GIS gateways are organized around geospatial data management, visualization, and analysis tools. Due to the prevalence of spatio-temporal data in a wide variety of domains such as hydrology, earth and atmospheric sciences, digital agriculture, forestry, and interdisciplinary studies such as agricultural economics, GIS gateways often have a mix of domain-independent as well as domain-specific tools for carrying out domain-specific simulation and model execution.

Following the popularity of R and Python as well as interactive computing and inline visualization (as enabled by platforms such as Jupyter), modern GIS gateways now include support for analysis using R and Python libraries and interactive computing platforms such as R Studio and Jupyter. With the rise in Python package and environment management tools such as Conda, most GIS gateways now include a variety of Conda environments, each designed for a specific GIS domain or set of processing capabilities. Separate Conda environments simplify their management and upgrade, while reducing the image size of containerized tools that may be built around them.

Some example of popular GIS gateways include CyberGISX (Yin et al. 2017), Hydroshare (Tarboton et al. 2014), and MyGeo-Hub (Kalyanam et al. 2019) each of which are based on different gateway technologies, but all implement geospatial data management, visualization, and analysis capabilities.

CyberGISX is based on JupyterHub and includes curated Jupyter notebooks demonstrating geospatial analysis and visualization, while also providing a large number of Conda environments for a variety of geospatial analysis including machine learning applications. CyberGISX also enables resource-intensive computations on the Roger cluster that is dedicated for GIS research. Hydroshare operated by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) enables researchers in hydrology to manage geospatial datasets and hydrologic models, and also provides the ability to publish these resources with DOIs. In addition, Hydroshare provides various web-based tools for interacting with and analyzing these datasets such as R Studio, Jupyter, Matlab, and customized tools such as CyberGIS for Water.

MyGeoHub based on the HUBzero gateway framework is operated via a shared-hosting model where several projects ranging from hydrology, agricultural economics, and sustainability analysis pay a fraction of the hosting costs. This enables MyGeoHub to continue to host the resources and tools developed by a project during gaps in funding, and archive the resources and tools during longer funding gaps. Like the other GIS gateways, MyGeoHub implements features specific to geospatial data on top of the HUBzero framework such as automated metadata extraction and visualization for geospatial files, and includes a variety of native-web and Jupyter-based GIS tools. MyGeoHub tools can submit resource-intensive jobs to a variety of campus and national HPC resources via the built-in HUBzero submit tool. MyGeoHub also hosts the GeoEDF plug-and-play workflow framework (Kalyanam et al. 2020) that addresses the challenge of wrangling data from different sources in geospatial workflows, making remote datasets directly usable in code while promoting research reproducibility.

7. Modern Gateway Concerns

Gateways have evolved through various technological phases and now play an important role in science and engineering research in various domains. With the growing reliance on gateways for end-to-end research and the rise in big data-driven methods propelled by advances in data-intensive machine learning, gateways now need to simplify the process of managing large datasets as well as computing on them. Consequently, gateways also need to simplify integrations with HPC as well as cloud resources, which have thus far relied on community accounts/credentials shared by all users on the gateway. Frameworks such as Apache Airavata and Tapis now provide users the capability to specify an allocation to use for these computations, but a generic framework is required for credential management and integration with job management libraries for ease of implementation on various gateways.

As gateways encourage users to carry out end-to-end research workflows, data acquisition and wrangling presents a significant challenge. In GIS particularly, researchers typically need to obtain and use data from a variety of sources such as NASA, USGS, USDA, etc. As the data sizes involved grow, it is no longer straightforward to simply transfer data from the source to the gateway. There has been a rise in cloud-optimized formats for geospatial data such as Zarr and Parquet that enable more efficient subsetting and analysis of parts of these large datasets. The ease of availability and seamless access of large datasets is still a challenge that most gateways need to address going forward.

With the growing emphasis on the FAIR principles (Wilkinson et al. 2016), both data and other resources need to be discoverable and re-usable by the broader community. The FAIR principles provide guidance on four key aspects that can help improve the “machine actionability” of digital assets: Findable, Accessible, Interoperable, and Reusable. Broadly, adhering to these four principles will ensure that humans and especially computer programs can easily find data and associated metadata, access them with authentication and appropriate authorization (if necessary), and gather sufficient information (via the metadata) to be able to use this data in combination with other data or computer programs. A related goal is to enable research reproducibility (Wilson et al. 2021; Kedron et al. 2021), which requires code and data contributed by users to have sufficient metadata and packaged in a way that simplifies its reuse. The rise in Jupyter notebooks solves some of these challenges by providing researchers with the ability to document the steps in a research workflow alongside the code in the notebook. However, packaging all the software dependencies as well as the datasets involved is still a challenge that most gateways need to solve. Containerization is a natural solution for these packaging challenges, but there is still a high barrier to entry for researchers to be able to create their own containers without assistance from the gateway.

Finally, sustaining a gateway is a significant challenge for many gateway providers. Keeping a gateway operational requires continuous software maintenance to ensure system security and functionality as the software stack upon which it is built changes over time. As the user needs evolve and new technology emerges, a gateway may need to adapt to support the increased demand for computational and data resources and new methods of research, all of which require sustained software engineering efforts. Multiple business models for sustaining gateways are in practice today, nonetheless, this will remain a challenge that any gateway development group ought to consider as early as possible.

References

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., and Mock, S. (2004). Kepler: an extensible system for design and execution of scientific workflows. In Proceedings, 16th International Conference on Scientific and Statistical Database Management, 423:424.

Basney, J., Flanagan, H., Fleury, T., Gaynor, J., Koranda, S., and Oshrin, B. (2019). CILogon: Enabling federated identify and access management for scientific collaborations. Proceedings of Science, 351: 031.

Deelman, E., Vahi, K., Juve, G., Rynge, M., Callaghan, S., Maechling, P. J., Mayani, R., Chen, W., Da Silva, R. F., Livny, M., et al. (2015). Pegasus, a workflow management system for science automation. Future Generation Computer Systems, 46:17-35.

Foster, I. (2011). Globus Online: Accelerating and democratizing science through could-based services. IEEE Internet Computing, 15(3): 70-73.

Gesing, S., Dahan, M., Zentner, M., Wilkins-Diehr, N., and Lawrence, K. (2019). The science gateways community institute: Collaborations and efforts on international scale. Future Generation Computer Systems, 101: 951-958.

Hanc, J., Strauch, P., Pankova, E., and Hancova, M. (2020). Teachers' perception of Jupyter and R Shiny as digital tools for open education and science. arXiv preprint.

Kalyanam, R., Zhao, L., Song, C., Biehl, L., Kearney, D., Kim, I. L., Shin, J., Villoria, N., and Merward, V. (2019). MyGeoHub—a sustainable and evolving geospatial science gateway. Future Generation Computer Systems 94:820–832.

Kalyanam, R., Zhao, L., Song, C., Merwade, V., Jin, J., Baldos, U., and Smith, J. (2020). GeoEDF: An extensible geospatial data framework for fair science, in Proceedings of the Practice and Experience in Advanced Research Computing 2020 (PEARC20) Conference, pp. 207–214.

Kedron, P., Li, W., Fotheringham, S., and Goodchild, M. (2020). Reproducibility and replicability: opportunities and challenges for geospatial research. International Journal of Geographical Information Science, 35(3): 427-445.

Klimeck, G., McLennan, M., Brophy, S. P., Adams III, G. B., and Lundstrom, M. S. (2008). nanoHUB.org: Advancing education and research in nanotechnology. Computing in Science & Engineering 10(5):17–23, 2008

Marru, S., Gunathilake, L., Herath, C., Tangchaisin, P., Pierce, M., Mattmann, C., Singh, R., Gunarathne, T., Chinthaka, E., Gardler, R., et al. (2011). Apache Airavata: a framework for distributed applications and computational workflows. In Proceedings of the 2011 ACM workshop on Gateway Computing Environments, 21-28.

McLennan, M., and Kennell, R. (2010). HUBzero: a platform for dissemination and

collaboration in computational science and engineering. Computing in Science & Engineering 12(2): 48-53.

Miller, M. A., Pfeiffer, W., and Schwartz, T. (2011). The CIPRES science gateway: A community resource for phylogenetic analyses. In Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery, 1-8.

Novotny, J., Russell, M., and Wehrens, O. (2004). GridSphere: a portal framework for building collaborations. Concurrency and Computation: Practice and Experience 16(5):503-513.

Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M. Carver, T., Glover, K., Pocock, M. R., Wipat, A., et al. (2004). Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics, 20(17): 3045-3054.

Stubbs, J., Cardone, R., Packard, M., Jamthe, A., Padhy, S., Terry, S., Looney, J., Meiring, J., Black, S., Dahan, M., et al. (2021). Tapis: An API platform for reproducible, distributed computational research. Future of Information and Communication Conference, 878-900.

Tarboton, D. G., Idaszak, R., Horsburgh, J. S., Heard, J., Ames, D., Goodall, J. L., Band, L., Merwade, V., Couch, A., Arrigo, J., et al. (2014). HydroShare: advancing collaboration through hydrologic data and model sharing," in Proceedings of International Congress on Environmental Modelling and Software.

Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., & Wilkins-Diehr, N. (2014). XSEDE: accelerating scientific discovery. Computing in Science & Engineering, 16(5), 62-74.

Wilkins-Diehr, N. (2007). Science gateways - common community interfaces to grid resources. Concurrency and Computation: Practice and Experience 19(6): 743-749.

Wilkinson, M., Dumontier, M., Aalbersberg, I., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data 3(1):1-9.

Wilson, J. P., Butler, K., Gao, S., Hu, Y., Li, W., and Wright, D. J. (2021). A Five-Star Guide for Achieving Replicability and Reproducibility When Working with GIS Software and Algorithms. Annals of the American Association of Geographers, 111(5): 1311-1317.

Yin, D., Liu, Y., Padmanabhan, A., Terstriep, J., Rush, J., and Wang, S. (2017). A CyberGIS-Jupyter framework for geosopatial analysis at scale. In Proceedings of the Practice and Experience in Advanced Research Computing 2017 (PEARC17) Conference, pp. 1–8.