Experiences from scaling scale Science Gateway operations

Suresh Marru
smarru@iu.edu
Science Gateway Research Center
Indiana University
Bloomington, IN

Marlon Piece
marpierc@iu.edu
Science Gateway Research Center
Indiana University
Bloomington, IN

Eroma Abeysinghe
eabeysin@iu.edu
Science Gateway Research Center
Indiana University
Bloomington, IN

Sudhakar Pamidighantam
pamidigs@iu.edu
Science Gateway Research Center
Indiana University
Bloomington, IN

Marcus Christie
machrist@iu.edu
Science Gateway Research Center
Indiana University
Bloomington, IN

Dimuthu Wannipurage
dwannipu@iu.edu
Science Gateways Research Center
Indiana University
Bloomington, IN

ABSTRACT

Science gateways are distributed computing systems that provide science-centric, end-user environments that simplify and expand the use of scientific software and data on diverse scientific software on backend resources. In this poster we describe the experiences of using a common software platform to host "Software as a Service" Science Gateways.

CCS CONCEPTS

• **Computing methodologies** → **Distributed programming languages**; • **Applied computing** → **Service-oriented architectures**; *IT architectures*; • **Software and its engineering** → **Software design tradeoffs**.

KEYWORDS

Distributed Computing, Science Gateways, Apache Airavata

ACM Reference Format:
Suresh Marru, Marlon Piece, Eroma Abeysinghe, Sudhakar Pamidighantam, Marcus Christie, and Dimuthu Wannipurage. 2019. Experiences from scaling scale Science Gateway operations. In *Practice and Experience in Advanced Research Computing (PEARC '19), July 28-August 1, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3332186.3333159

1 INTRODUCTION

Science Gateways provide a crucial user-centric and science-centric point of entry to the collection of computing, storage, and software that are used to support science and commonly referred to as cyberinfrastructure (CI) [3]. Over the past two decades, Science Gateways have dramatically increased cyberinfrastructure usage and accessibility for scientists and educators around the world. Gateways provide a federating bridge over cyberinfrastructure spanning campus resources, National Resources like Extreme Science and Engineering Discovery Environment XSEDE [10], commercial clouds and international computing resources.

Science Gateways develop and operate a domain-specific presentation layer as well as a generic gateway middleware layer that supports a common set of required functionalities. Repeating this development process for each new gateway is inefficient and wasteful of resources. In this poster we summarize our experiences, architectural and deployment choices made in within the open source and open community based Apache Airavata framework [4] to create a robust, sustainable Science Gateway Platform (SciGaP) [7] which minimizes the net operating cost of Science Gateways. Figure 1 illustrates the high level concept.

2 ANATOMY OF A SCIENCE GATEWAY

Science Gateways are typically a ecosystems of multiple software components integrated to operate as a unified service. These components typically include a user interfaces that are useful for end user communities; a data management systems to manage domain-specific data and metadata; identity and access management system to manage user identity, accounts, authorization and access for multiple, evolving available resources; an application catalog and resource catalog to record community applications installed, running, and integrated with cyberinfrastructure middleware on a wide range of resources from campus, national, and international Grid and cloud efforts; software components to reliably running jobs and returning results, supporting advanced execution scenarios, managing data; and instrumentation to providing job status feedback and easily understandable error reports.

As discussed in detail in [9] the goal of SciGaP project is to create a robust, sustainable infrastructure that can provide new gateway developers with the generic middleware functionalities required by all Science Gateways. The decreased overhead for operations can free resources for developing new capabilities, improving user interaction and support, and enhancing outreach efforts. As described in [9] SciGaP promotes sustainability through scaling: instead of having O(N) developers and operators for O(N) gateways, through consolidation to SciGaP aspired enable O(M) developers and operators to manage O(N) gateways, where M«N.

PEARC '19, July 28-August 1, 2019, Chicago, IL, USA

ACM ISBN 978-1-4503-7227-5/19/07…$15.00
https://doi.org/10.1145/3332186.3333159

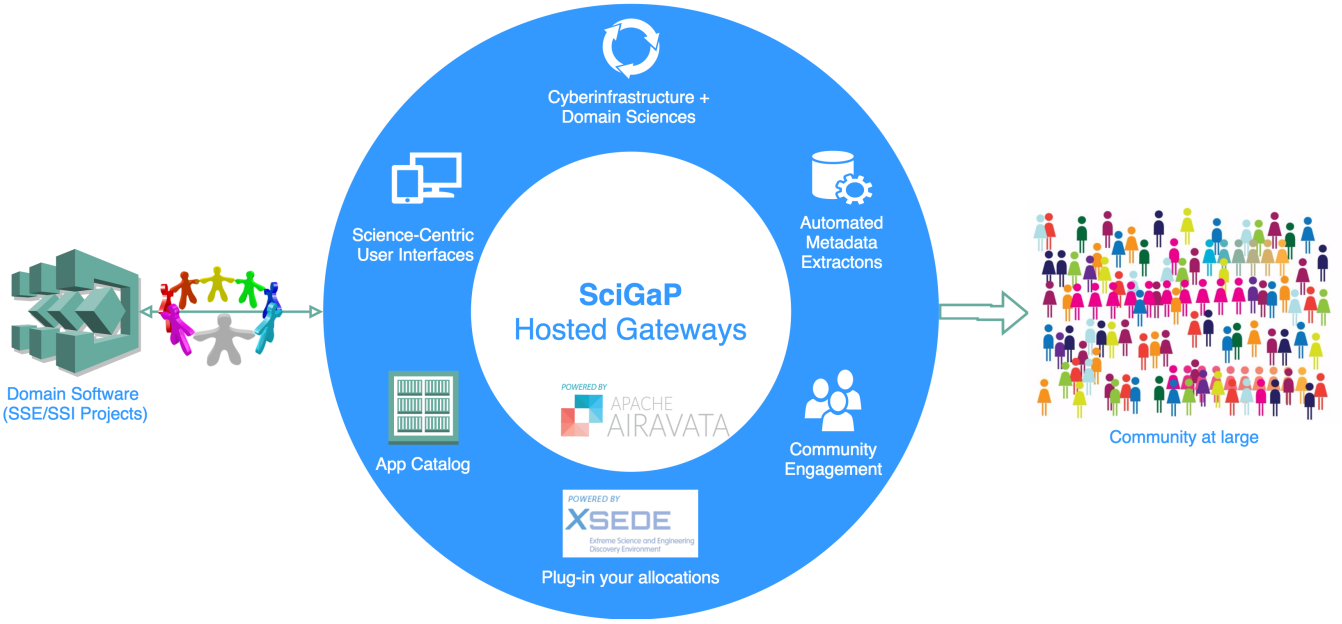

Figure 1: Overview of SciGateways Platform as a Service(SciGaP).

3 ARCHITECTURE CHOICES

As a first step to converging on a single set of hosted infrastructure services, we evaluated multiple architectural choices and choose multi-tenanted architectural pattern [5]. We experimented with multi-tenanted patterns of having identical data models for each hosted tenant. This options offers substantial scalability but limits configuration options for each gateway. On a contrary, multi-tenant with custom data models for each hosted gateway tenant supports custom capabilities but quickly multiplies to the operational cost. After further evaluation, we choose a hybrid approach of flexible and extendable data models. Individual gateways are logically segmented at the database level, complete with their own database models. This approach retain all the advantages of a highly scalable and secure multi-tenant model while still offering a highly configurable application powering diverse gateways.

3.1 Apache Thrift based flexible data models

Apache Airavata Application Programming Interface (API) [8] is developed using Apache Thrift's modular serialization framework [1]. Gateways define abstract data types in an Interface Definition Language (IDL). This IDL is then compiled into source code for any supported language. The generated code provides complete serialization and deserialization logic for all of the user's defined types. Apache Thrift ensures that types written by any language can be read by any other language. IDL creates a contract that gateway clients and Airavata Services can rely upon and that code generators can use to create working serialization operations, ensuring the contract is adhered to. Apache Thrift IDL supports a range of interface evolution features which, when used properly, allow fields to be added and removed, types to be changed, and more. Support for interface evolution greatly simplifies the task of ongoing software maintenance and extension. All Apache Airavata components illustrated in figure 2 describe interfaces using Thrift IDL's.

3.2 Airavata API

Airavata's public facing application programming interfaces (API's) are also based on Apache Thrift, which gives Airavata a strongly typed, programming language independent way of defining its interfaces. Based on API IDL, Airavata generates client packages in Java, PHP, Python and C++. Client gateways access Airavata through the API Server through a secure channel (SSL sockets or HTTPS). The API Server maps the client request into one or more calls to internal components described next.

3.3 Microservice Architecture

Airavata architecture packages the components based on functional areas and scalability and reliability needs referred to as microservices [6]. Microservice architectural pattern combined with Continuous Integration and Delivery (CI/CD) [2] enables the SciGaP system to support incremental improvements without impacting the rest of the platform. Use of Apache Thrift interfaces as described above facilitates feature evolution allow multiple interface versions to coexist seamlessly in a single operating environment. This makes incremental updates viable, enabling CI/CD pipelines and empowering individual gateways to deliver science value at their own cadence.

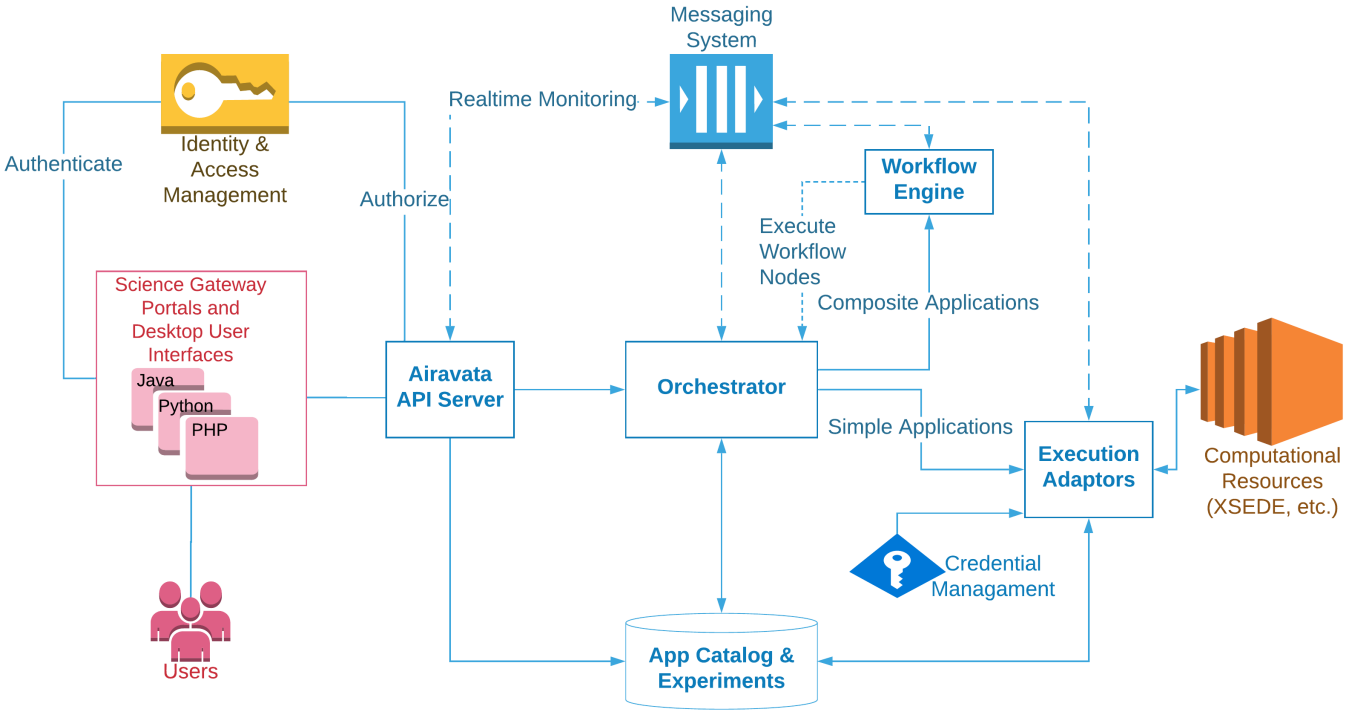

Figure 2: High Level architectural overview of Apache Airavata.

4 SCIENCE GATEWAYS INTEGRATION

Table 1 summarizes gateways using various computing resources (including Extreme Science and Engineering Discovery Environment XSEDE [10] brokered through a single set of hosted SciGaP Services.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation award number 1339774.

REFERENCES

[1] Randy Abernethy. 2018. *Programmer's Guide to Apache Thrift.* Manning Publications.
[2] Martin Fowler and Matthew Foemmel. 2006. Continuous integration. *Thought-Works) http://www. thoughtworks. com/Continuous Integration. pdf* 122 (2006), 14.
[3] Katherine A Lawrence, Michael Zentner, Nancy Wilkins-Diehr, Julie A Wernert, Marlon Pierce, Suresh Marru, and Scott Michael. 2015. Science gateways today and tomorrow: positive perspectives of nearly 5000 members of the research community. *Concurrency and Computation: Practice and Experience* 27, 16 (2015), 4252–4268.
[4] Suresh Marru, Lahiru Gunathilake, Chathura Herath, Patanachai Tangchaisin, Marlon Pierce, Chris Mattmann, Raminder Singh, Thilina Gunarathne, Eran Chinthaka, Ross Gardler, et al. 2011. Apache airavata: a framework for distributed applications and computational workflows. In *Proceedings of the 2011 ACM workshop on Gateway computing environments.* ACM, 21–28.
[5] Suresh Marru, Marlon Pierce, Sudhakar Pamidighantam, and Chathuri Wimalasena. 2015. Apache airavata as a laboratory: architecture and case study for component-based gateway middleware. In *Proceedings of the 1st Workshop on The Science of Cyberinfrastructure: Research, Experience, Applications and Models.* ACM, 19–26.
[6] Sam Newman. 2015. *Building microservices: designing fine-grained systems.* " O'Reilly Media, Inc.".
[7] Marlon Pierce, Suresh Marru, Eroma Abeysinghe, Sudhakar Pamidighantam, Marcus Christie, and Dimuthu Wannipurage. 2018. Supporting Science Gateways Using Apache Airavata and SciGaP Services. In *Proceedings of the Practice and Experience on Advanced Research Computing.* ACM, 99.
[8] Marlon Pierce, Suresh Marru, Borries Demeler, Raminderjeet Singh, and Gary Gorbet. 2014. The apache airavata application programming interface: overview and evaluation with the UltraScan science gateway. In *Proceedings of the 9th Gateway Computing Environments Workshop.* IEEE Press, 25–29.
[9] Marlon Pierce, Suresh Marru, Mark A Miller, Amit Majumdar, and Borries Demeler. 2013. *Science Gateway Operational Sustainability: Adopting a Platform-as-a Service Approach.* Technical Report.
[10] John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D Peterson, et al. 2014. XSEDE: accelerating scientific discovery. *Computing in Science & Engineering* 16, 5 (2014), 62–74.

Table 1: Science Gateways operated through multi-tenanted SciGaP Infrastructure

Gateway Name	Gateway URL	Field of Science	XSEDE Machines in Use
SEAGrid	https://seagrid.org/	Chemistry & Engineering	Comet, Stampede2, Bridges, Jetstream, Wrangler
Ultrascan	http://ultrascan.aucsolutions.com/	Biophysics	Comet, Stampede2, Jetstream
PGA	https://testdrive.airavata.org/	Computer & Information Science & Engineering	Comet, Stampede2, Jetstream
dREG	https://dreg.dnasequence.org/	Genetics & Nucleic Acids	Comet, Bridges, Jetstream(Gateway hosting)
PHASTA	https://phasta.scigap.org/	Mechanical Engineering	Comet, Stampede2
SimVascular	https://gateway.simvascular.org/	Cardiovascular Simulation	Comet
Searching SRA	https://www.searchsra.org/	Bio-informatics and Biology	Jetstream, Wrangler
InterACTWEL	http://interactwel.scigap.org/	Natural Resources Management Decisions Support	Jetstream
Next GEN Thermo DB	https://geochemsim.org/	Geochemistry & Environmental Science	Jetstream (Gateway hosting and job submission)
Atomic and Molecular Physics	https://ampgateway.org/	Atomic, Molecular, and Optical Physics	Comet, Stampede2, Bridges
Distant Reader	https://distantreader.scigap.org/	Library Science	Jetstream
Single Cell RNA Sequencing	https://singlecellgateway.wharton.upenn.edu/	Genetic Science	Bridges
Prostate Cancer Prediction	https://gemr.scigap.org/	Health Science	Comet