Experiences Providing a Multi-Tenanted Science Gateways Platform Service for Campus Cyberinfrastructure

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

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

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

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

KEYWORDS

1 INTRODUCTION

Science gateways make it easy to access and use cyberinfrastructure for scientific research, education and visualization [14]. Cyberinfrastructure consists of computing systems, data storage systems, advanced instruments, visualization environments, and people, linked together to improve research productivity [12]. Computational tools are routinely employed in research, and colleges and universities increasingly need to support diverse set of users. According to [6], at least 60% of respondents indicated that Web-based gateway applications were "somewhat" or "very" important for science research. This is supported by the fact that national Infrastructures such as XSEDE have more users accessing computational resources through Science Gateways than otherwise.

As the cost of computing resources has gone down, the number of universities offering centralized computing resources has increased [4]. As approximate measures of this breadth, we note that the XSEDE campus champions program includes over 220 members, and the Open Science Grid has over 160 member sites at the time of writing. An obvious benefit of providing university-level computing infrastructure is that these resources are dedicated to the university community. An important but still relatively emerging trend is the federation of statewide and regional cyberinfrastructure. The typical motivation for this is for the large, research-oriented universities in a state to support users at smaller colleges with less resources and expertise. The "OneOklahoma Cyberinfrastructre" Initiative (http://www.oneocii.okepscor.org/) is an example of this type of statewide cyberinfrastructure.

We thus identify some challenges and an opportunities. Gateways have been very successful at the national cyberinfrastructure level, and there are a large number of universities that now offer centralized computing facilities for their local and regional users. Can gateways be adopted by campus and regional cyberinfrastructure deployments just as they are at the national level?

Given the value proposition gateways provide in on boarding new users while simplifying access to well known applications, computing centers are increasingly deploying science gateways. Gateways can thus provide a force multiplying effect on computing center support staff: well designed, stable and supported gateways provide users with a self-service mechanism for interacting with campus resources, freeing support staff and giving them more time to devote to high priority users and projects.

Science gateways can be built using a number of technologies [8], [2], [7], [3], [5]. These fall into two general categories: gateway providers can download the gateway software and run it locally, or the providers can make their computing resources available to a centrally administered gateway platform as a service. Several gateway frameworks provide both options. There are tradeoffs for either approach. A key element in deciding which to choose is level of effort required from local computing cluster staff; if the gateway is taking too much support staff time to maintain, then it is not serving its purpose.

In the Science Gateways Platform as a service (https://scigap.org/) project, as the name indicates, we are focusing on the latter approach [10]. We believe that using a centralized but transparent service will ultimately provide more benefits to gateway providers than running software locally: services can be maintained at the latest, unified versions, the support cycle can be reduced, and the gateway software providers can devote more resources to development and enhancement.

Such benefits do not simplify follow logically from having a platform service. This extended abstract describes the technical challenges for and experiences gained by providing a science gateways platform as a service for campus computing resources, and how these differ from XSEDE science gateways. We will discuss how we dealt with the challenges of enabling individual user accounts, multiple authentication mechanisms, legacy resources, and

disparate functional requirements of science researchers and non-uniform software stacks installed in resources. Finally the paper briefly discusses our future work in these areas.

2 PROVIDING A HOSTED GATEWAY PLATFORM

At the most basic level, hosting a gateway using an existing instance of multi-tenant supporting middleware is a two-step process: 1) deployment and 2) configuration. However, in a university environment, there are situations that are going to add a layer of complexity to the deployment. In order to host a gateway successfully we obtain the services from gateway providers, gateway administrators/PIs and cluster technical staff.

Figure 1 illustrates a set of hosted gateways using Apache Airavata hosted middleware. One of the goals of the SciGaP project is to expand beyond our initial collaborating gateways (CIPRES, Neuroscience Gateway, and UltraScan) to provide a general purpose infrastructure. The Identity Server [9] within Apache Airavata eco system integrates with Campus authentication via CILogon [1] service.

When deploying these science gateways we identified three main categories: general purpose gateways that need to support a diverse application set, domain-centric gateways that focus on a particular field of science, and application-centric gateways that focus on making a particular application or set of applications easier to use. Campus gateways fall into the general purpose category and are dedicated to a single campus and its collaborative research work. They may wish to offer up a very diverse set of applications. Domain-centric gateways target certain fields, but are more generic in providing applications to their users and are not tied to specific resources. That is, the gateway provider and the resource provider are distinct. In both cases, users will find a mixed combination of published and licensed applications, as well as privately developed applications and softwares. Finally, application centric gateways (such as PHASTA and dREG) focus on providing only a single application or set of closely related applications in order to increase the usability and adoption of a particular code. Like domain-specific gateways, application gateways are not tied to specific resources.

Figure 2 illustrates the way a potential gateway administrator would go about requesting an Apache Airavata gateway through the SciGaP platform. Requests are made through our SciGaP platform, and once received they are evaluated by our SciGaP team. When the request is evaluated and approved, gateway requested person (typically the potential gateway administrator) is informed and together gateway is configured by providing gateway preferences, typical applications, storage resources, etc. After this is completed, the initial group of users are invited to submit jobs through the gateway. Although this sounds straightforward, there are consistent challenges we have faced as we have worked through this process with various institutions. Below we share some of the most common challenges as well as some strategies we have adopted to face them.

## 3 CHALLENGES PROVIDING CAMPUS GATEWAYS

As we extend the SciGaP platform services to include campus resources, we encountered several challenges described below.

No Community Accounts: Most of the institutions and campuses we have worked with have not wanted to distribute group or community accounts [13]. This is understandable given the need to account for computing resources, the need for security, and the possibility of audit. For this reason, we encourage users to use their own cluster allocation for job submissions. This leads to users having to take necessary steps to add their public keys to their cluster allocation, which in turn leads to more work for cluster administrators. This is currently one of the primary issues we are working on. One solution involves automating the process of adding the public key to user allocation in the cluster. For example, when the user logs in to the gateway for the first time, a generated key will be added for ssh communications.

Diverse User Management: Campuses follow different protocols and mechanisms for user authentications. Gateway deployments for each of these campuses have to adapt to these diverse mechanisms. Gateway administrators also differ in defining policies for onboarding users onto a gateway. Almost all campuses prefer to enable their campus identity systems to access the portal, thus avoiding a manual user management within the gateway. To meet these requirements, Airavata integrated with InCommon Federation and CILogon Identity brokering. Some campuses have LDAP servers dedicated to the research infrastructure and the gateway has to be integrated with these systems. A few gateways prefer to maintain a stand alone user management where users operate the gateway upon account validation (without administrative approval). These variations are enabled within Airavata using Industry standard security protocols and the use of Open Source Identity Servers. Airavata currently uses WSO2 Identity Server [15] and is evaluating the feasibility of migrating to Redhat's KeyClock Identity Server [11].

Legacy Systems: While working with campus clusters we have encountered diverse job-schedulers and multiple versions of software applications. Clusters are designed to be refreshed every three years, but often they are not, and financial constraints and familiarity both contribute to the use of legacy software. In situations where older software versions are being run on an outdated infrastructure, technical issues with the gateway are going to occur. Specifically, the lack of proper module commands forces significant collaboration between gateway consultants and gateway administrators. This challenge can only be mitigated through timely updates and proper maintenance of both software and hardware. When gateway cluster-based issues arise, technical support from someone familiar with the cluster is essential. Many science gateway clusters do not have a dedicated technical staff, and this complicates solving problems and increases wait times for doing so. It is expected that there is a gateway administrator who would champion the gateway from campus user support organizations ably supported by their own cluster technical staff.

For gateways to enable job submissions to clusters, a programmatic way to monitor progress is required. Airavata currently relies on email-based monitoring as all job managers reliably send emails upon job status changes. In gateway clusters this is initially challenging because the email formats and the content are not automatically identified with the job status. Multiple rounds of testing need to be done in order to finalize email formats and content
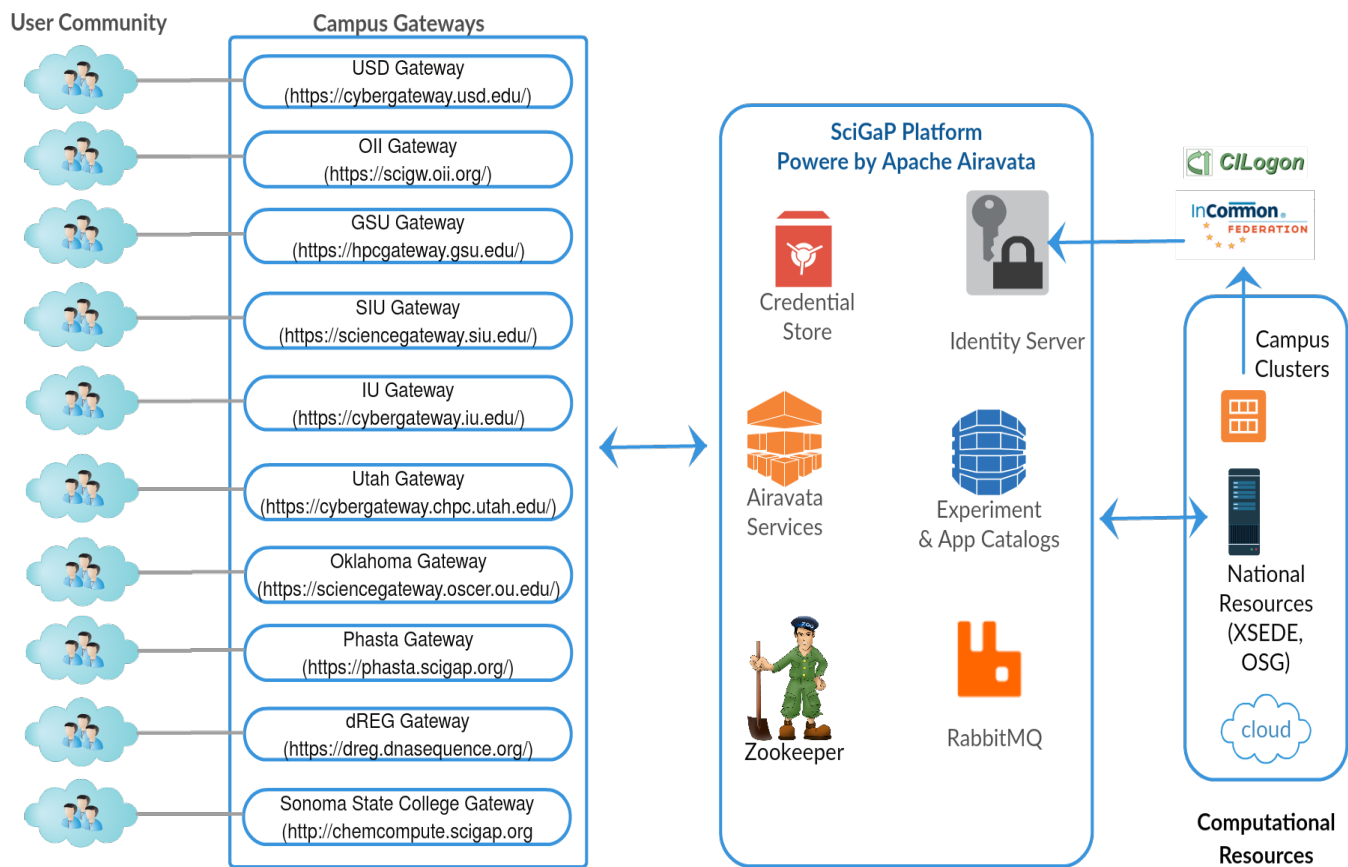
Figure 1: Science gateways with Airavata middleware and computational resources
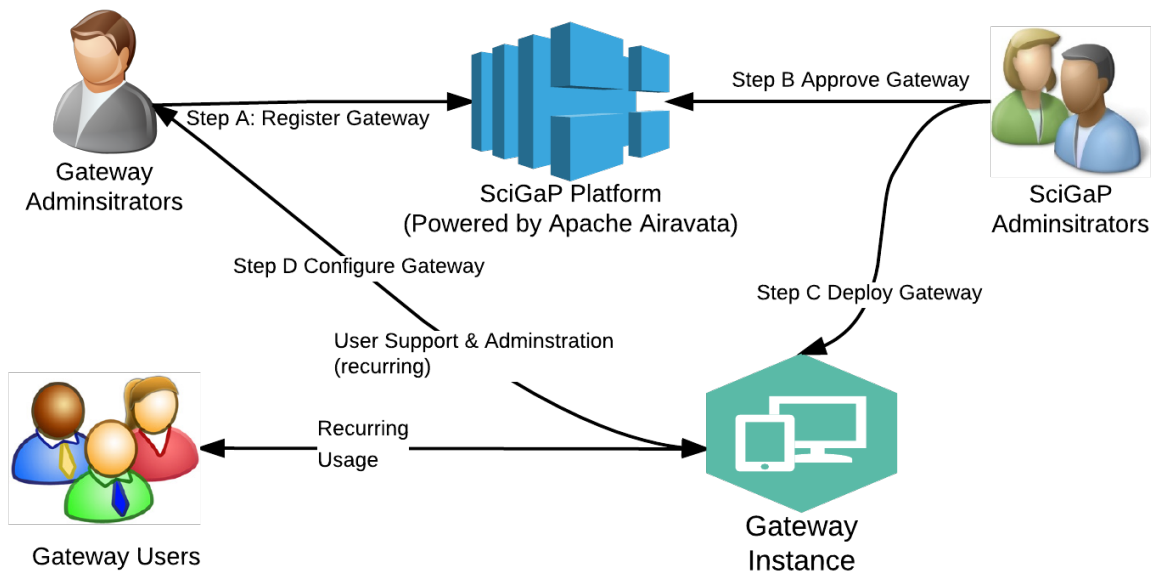
Figure 2: Workflow illustrating steps from requesting a gateway to using and providing support

with correct job status updates. This requires time and energy from gateway providers, gateway administrators and cluster technical staff. In most cases the technical assistance is also, provided by the gateway administrators. In the future we plan to explore alternative job monitoring mechanisms so it is not required to solely rely on email-based monitoring systems.

Feature Support: Apache Airavata is open source software to which anyone can contribute. Apache Airavata supports the integration of arbitrary gateways through its Application Programming Interface (API), which can be used by any gateway developer. We provide also a reference implementation, the PHP Gateway for Airavata (PGA), which can be simply taken and used with minimal theming.

The key advantage of a centralized platform gateway is that it allows campus support staff to outsource their gateway operations. In our experience, campus gateway providers are less likely to want to make extensive customizations to their gateway when compared to application and domain-centric gateway providers. Campuses prefer to make requests for features and changes from the gateway providers. When managing multiple gateways, gateway providers can end up with a pipeline of feature requests. While this may result in delays from the gateway tenant's point of view, it does help us as platform providers to identify particularly important, cross-cutting features that need to be implemented.

4 FUTURE WORK

First, we must reduce the number of human interactions needed to deploy a gateway. We are working to reduce the time for gateway deployment and configuration and to automate the process as much as possible. Second, at this time, campus clusters need to be added to the gateway by the SciGaP administrators. In the future this will be a task for the gateway administrators themselves because we are actively assembling publicly-available resources on deploying gateways and will soon have a catalogue of working examples. Third, we must unify the user provisioning process between the gateway and the cluster. Currently this requires the manual step of adding public keys generated by the gateway to a user's authorized_keys file in clusters. Fourth, the diversity of deployments of schedulers and resource managers on campuses has challenged our current solutions for generating queuing scripts because of the variety of applications and softwares deployed in various campus clusters. Currently, we generate individual solutions foreach new deployment; moving forward we intend to develop a more refined and more generic application property configuration. Fifth, we need to revisit our permission models to make it easier for "power" users to have access to administrative methods in the API in order to control their deployments within a gateway. These operations are currently restricted to administrators. Gateway providers will have to decide who should have this privilege.

ACKNOWLEDGMENTS

REFERENCES

[1] Jim Basney, Terry Fleury, and Jeff Gaynor. 2014. CILogon: A federated X. 509 certification authority for cyberinfrastructure logon. *Concurrency and Computation: Practice and Experience* 26, 13 (2014), 2225–2239.

[2] Rion Dooley, Matthew Vaughn, Dan Stanzione, Steve Terry, and Edwin Skidmore. 2012. Software-as-a-service: the iPlant foundation API. In *5th IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS)*.

[3] Jeremy Goecks, Anton Nekrutenko, and James Taylor. 2010. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology* 11, 8 (2010), R86.

[4] Thomas J Hacker and Bradley C Wheeler. 2007. Making research cyberinfrastructure a strategic choice. *Educause Quarterly* 30, 1 (2007), 21.

[5] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, and others. 2016. Jupyter Notebooks–a publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (2016), 87.

[6] 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.

[7] Suresh Marru, Lahiru Gunathilake, Chathura Herath, Patanachai Tangchaisin, Marlon Pierce, Chris Mattmann, Raminder Singh, Thilina Gunarathne, Eran Chinthaka, Ross Gardler, and others. 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.

[8] Michael McLennan and Rick Kennell. 2010. HUBzero: a platform for dissemination and collaboration in computational science and engineering. *Computing in Science & Engineering* 12, 2 (2010).

[9] Supun Nakandala, Hasini Gunasinghe, Suresh Marru, and Marlon Pierce. 2016. Apache Airavata Security Manager: Authentication and Authorization Implementations for a Multi-Tenant eScience Framework. (2016).

[10] Marlon E Pierce, Suresh Marru, and Chris Mattmann. 2015. Patching It Up, Pulling It Forward. *Journal of Open Research Software* 3, 1 (2015).

[11] Redhat. 2017. Key Cloak Identity and Access Management. (2017). http://www.keycloak.org/

[12] Craig A Stewart, Stephen Simms, Beth Plale, Matthew Link, David Y Hancock, and Geoffrey C Fox. 2010. What is cyberinfrastructure. In *Proceedings of the 38th annual ACM SIGUCCS fall conference: navigation and discovery*. ACM, 37–44.

[13] Von Welch, Jim Barlow, James Basney, Doru Marcusiu, and Nancy Wilkins-Diehr. 2007. A AAAA model to support science gateways with community accounts. *Concurrency and Computation: Practice and Experience* 19, 6 (2007), 893–904.

[14] Nancy Wilkins-Diehr, Dennis Gannon, Gerhard Klimeck, Scott Oster, and Sudhakar Pamidighantam. 2008. TeraGrid science gateways and their impact on science. *Computer* 41, 11 (2008).

[15] WSO2. 2017. WSO2 Identity Server. (2017). https://wso2.com/identity-and-access-management