CTSC

CENTER FOR TRUSTWORTHY
SCIENTIFIC CYBERINFRASTRUCTURE
The NSF Cybersecurity Center of Excellence

SciGaP-CTSC Engagement: Final Technical Recommendations

April 29, 2016

For Public Distribution

Randy Heiland, Scott Koranda, and Von Welch

About CTSC

The mission of the Center for Trustworthy Scientific Cyberinfrastructure (CTSC, trustedci.org) is to improve the cybersecurity of NSF science and engineering projects, while allowing those projects to focus on their science endeavors. This mission is accomplished through one-on-one engagements with projects to solve their specific problems, broad education, outreach and training to raise the practice-of-security across the community, and looking for opportunities for improvement to bring in research to raise the state-of-practice.

Acknowledgments

CTSC's engagements are inherently collaborative; the authors would like to thank the SciGaP team, especially Marlon Pierce and Suresh Marru, for the collaborative effort that made this document possible.

This document is a product of the Center for Trustworthy Scientific Cyberinfrastructure (CTSC). CTSC is supported by the National Science Foundation under Grant Numbers OCI-1234408 and ACI-1547272. For more information about the Center for Trustworthy Scientific Cyberinfrastructure please visit: http://trustedci.org/. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Using & Citing this Work

Cite this work using the following information:
R. Heiland, S. Koranda, and V. Welch, "SciGaP-CTSC Engagement: Final Technical Recommendations," Center for Trustworthy Scientific Cyberinfrastructure, trustedci.org, April 2016. Available: http://hdl.handle.net/2022/20927

This work is available on the web at the following URL: http://trustedci.org/scigap

Table of Contents

Executive Summary

The Science Gateway Platform as a service (SciGaP) project provides middleware services for science communities. SciGaP has several cybersecurity challenges as it integrates web, campus cyberinfrastructure, and cloud technologies. These challenges cover a broad range of topics: levels of trust between multiple entities, identity management, authentication and authorization, software assurance, and more. The CTSC-SciGaP engagement has been quite unique. Unlike most every other CTSC engagement which have had relatively short durations (few months), very targeted goal(s), and very concentrated effort; the SciGaP engagement has been more open-ended, with a longer duration (about 18 months), but very infrequent meetings. One reason for choosing this consulting-style engagement model was that the SciGaP project had only recently begun when the engagement started, so CTSC staff made themselves available over time as the SciGaP project started up. The engagement has clarified security challenges, generated actionable advice, and produced multiple reports that should be useful for general security issues for the broader NSF science community.

1 Introduction

A high-level diagram of a science gateway [1], [2] architecture is shown in Figure 1. In a typical use case, a user/scientist accesses a gateway from a browser, authenticates, and creates/submits a computational/data task or workflow. The gateway middleware orchestrates the execution of the task/workflow on one or more resources that the user is authorized to use, hiding much of the complexity from the user.

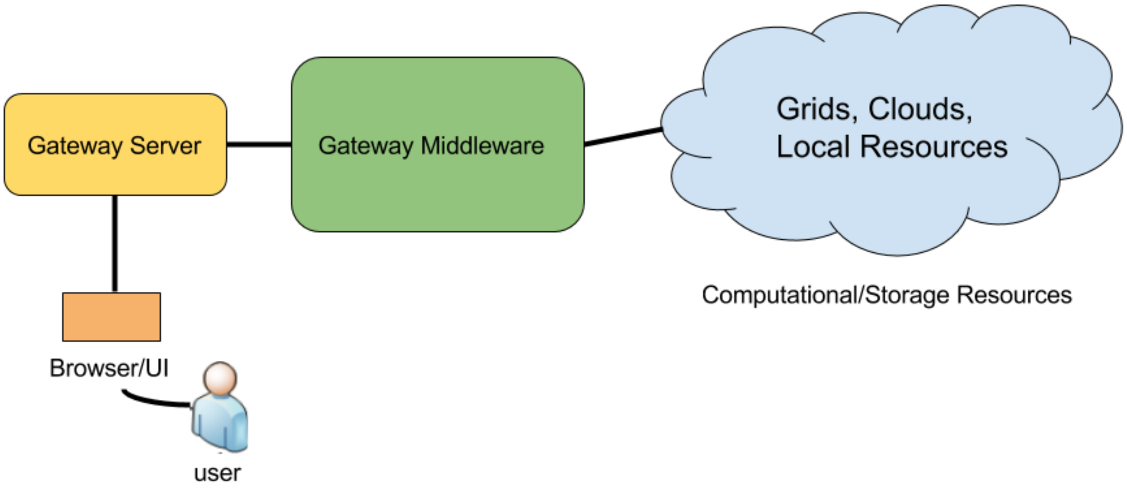

Gateway Server
Gateway Middleware
Grids, Clouds, Local Resources
Computational/Storage Resources
Browser/UI
user

Figure 1. Science gateway diagram.

The Science Gateway Platform as a service (SciGaP, scigap.org) project, depicted in Figure 2, is a hosted service that provides common middleware functionality for multiple gateway servers and "thick" clients [3] simultaneously. SciGaP provides software development kits (SDKs; in multiple languages) that help gateway developers use its application programming interface (API).

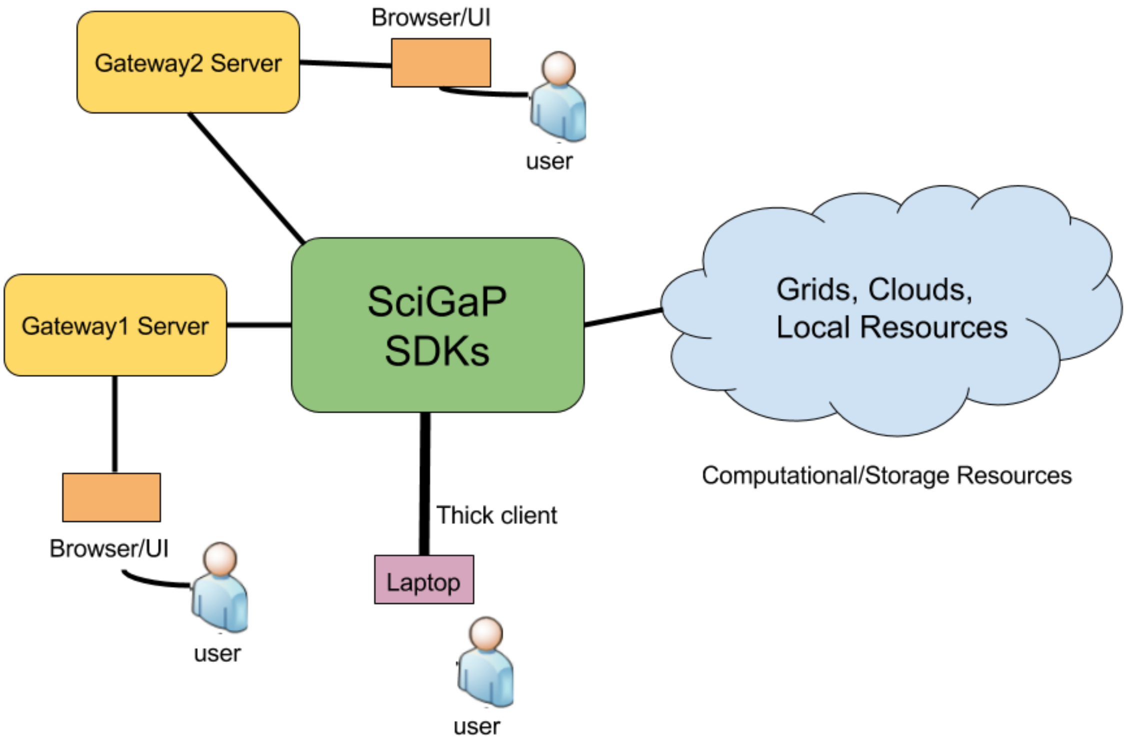


Figure 2. SciGaP diagram

There will be multiple entities that need to interact in SciGaP: individual users, system administrators (for the servers and resources), domain science communities, other services (e.g. for authentication, data movement), etc. From a security perspective, this brings us to the next topic: trust.

2 Trust Models

Trust models "describe ways in which organizations can obtain the levels of trust needed to form partnerships, collaborate with other organizations, share information, or receive information system/security services." (p. G1-G2 of [4]). In the context of science gateways, they describe how the science gateway and the resource provider establish trust such that the resource provider is willing to provide services to the science gateway on behalf of that science gateway's user community.

The NIST 800-39 report [4] goes on to describe a variety of trust models; in this document we will focus on two trust models: *brokered trust* and *transitive trust* as described in [5]. Brokered trust (Figure 3) is when a trust relationship is created through a trusted third party. In the context of XSEDE, an example of brokered trust is when a project PI requests account creation for members of their project team [6]. XSEDE grants these requests because of their relationship with a principal investigator (PI), and then establishes a direct relationship with the project member.

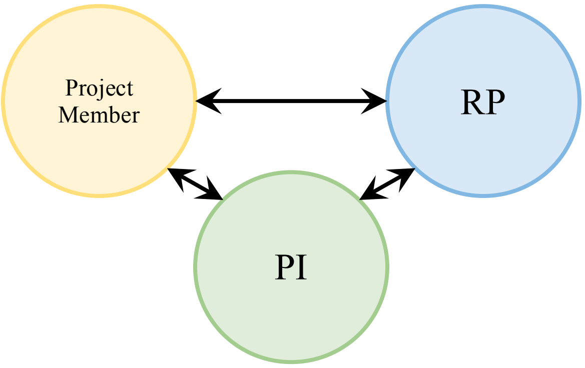


Figure 3. Brokered trust model with resource provider establishing relationships with members of a PI's project through the PI who acts as a broker to establish the relationship.

Brokered trust in science gateways comes about when a gateway user has a relationship (an account typically) with a resource provider (RP) and utilizes the science gateway as an interface to access that account. While the gateway is trusted by both parties in that circumstance (e.g., it may pass through and have access to credentials), the primary relationship is between the user and the resource provider[1].

Transitive trust (Figure 4) is the relationship between communities, science gateways and resource providers as described in [7]. In this case, the science gateways serve as intermediaries in the relationship such that there are really three trust relationships: one between the community served by the science gateway and the gateway, one between the gateway and the resource provider, and then the resulting relationship between the community and the resource provider that results from the combination of the first two relationships.

[1] Technically, this would be what [4] describes as a Hybrid Model.

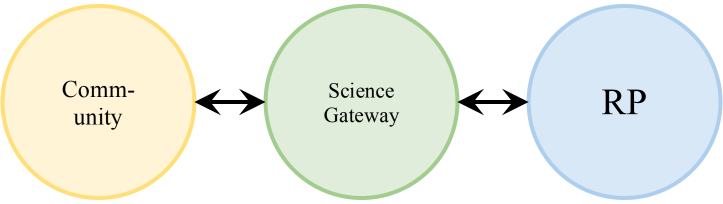


Figure 4. Transitive trust approach with the science gateway managing its community and resource providers trusting it to do so. No direct trust relationship between community and the resource provider exists.

As described in [8], there are a variety of reasons why transitive trust may be preferred for a given set of partners, including having dynamic community membership with complicated roles, and community that span multiple resource providers. The same reference describes that transitive trust tends to be more readily achieved when the missions of the resource provider and community align, they have an established history and relationship, and when the community has suitable experience and resources in managing their community and services to give the resource provider confidence in their ability to do so.

3 Identity and Access Management

Much of our engagement has dealt with the topic of identity and access management (IAM). SciGaP will be responsible for managing user and community identities, authentication and authorization to resources, and account management for those resources [9]. IAM has been a challenging, evolving topic for science gateways ever since they began over ten years ago. The IAM ecosystem involves a variety of authentication concepts, e.g., username/password, SSH, SSL, X.509 certificates, SAML, OAuth, OpenID Connect, etc. [10] and there are a variety of software packages and services that have provided IAM solutions for gateways [7], [11]–[13]. During our engagement, CTSC cautioned the SciGaP team about the challenges of writing and maintaining their own "credential store" [9]. Partly based on this advice, SciGaP researched available options and determined the WSO2 Identity Server (WSO2 IS) (http://wso2.com/products/identity-server/) would be a good choice. CTSC and SciGaP jointly reviewed the documentation for WSO2 IS and determined that it would likely be an adequate solution for their identity management needs. In addition to the IAM functionality it provides, other benefits include: 1) it is open source, and 2) the SciGaP team is acquainted with the WSO2 team and therefore it is likely SciGaP can get customized functionality in WSO2 IS.

We note that CTSC makes available several resources that address IAM (http://trustedci.org/iam/).

4 OAuth Best Practices

Because SciGaP considers OAuth to be highly desirable for authentication - from a usability perspective, we include some background and best practices that have been only partially in our other reports [14][3][10].

Resource Owner Password Credentials Grant, as defined by Section 4.3 of [15], is intended for situations where a OAUTH client is not capable of other, more secure workflows described in RFC 6749. It is in effect a compromise to allow legacy clients, or other situations where the normal OAUTH workflows cannot be used, to still operate and do so more securely than persistently storing credentials. As the security considerations section (10.7) of RFC 6749 describes, the client must be trusted to not misuse the credentials (e.g. ask for more authority than is intended) and not to intentionally or accidentally persist them in some manner (e.g. in a log).

Whether or not the use of Resource Owner Password Credentials Grant is suitable for a particular situation will depend on the details of the situation: the importance of the client being enabled, the level of trust in the client, and the risk tolerance of the parties involved. In the authors' experience, situations exist where its use would be warranted. Its inclusion in RFC 6749 would imply that the RFC authors (and associated working group members) agree. (We note the security section states that its use SHOULD be minimized, which is a not as strong a statement as could have been made to avoid its use - e.g. they could have said it "MUST be avoided.")

SciGaP's two use cases for wanting this type of OAuth grant were: 1) desktop applications, i.e. *thick* clients, and 2) gateways that wish to avoid having users be redirected to a (unknown) site for credentials. A specific example of the second use case is:

> The SEAGrid gateway (seagrid.org) has an existing community of users who know and trust SEAGrid services. When SEAGrid adopts SciGaP for IAM, they would like to avoid the OAuth "Authorization Code Grant" that would redirect them to the SciGaP identity management page. SEAGrid users may not know/trust the SciGaP service and might be reluctant to use it.

If this grant type is used, one potential risk mitigation that could be applied is to use short-lived (or even single use) passwords to reduce the trust in the client to not store or lose confidentiality of those passwords (e.g. as described in [16]).

We recommend establishing a written agreement with the client that:

1. Describes the intended use of credentials it receives (or how that intended use will be conveyed);

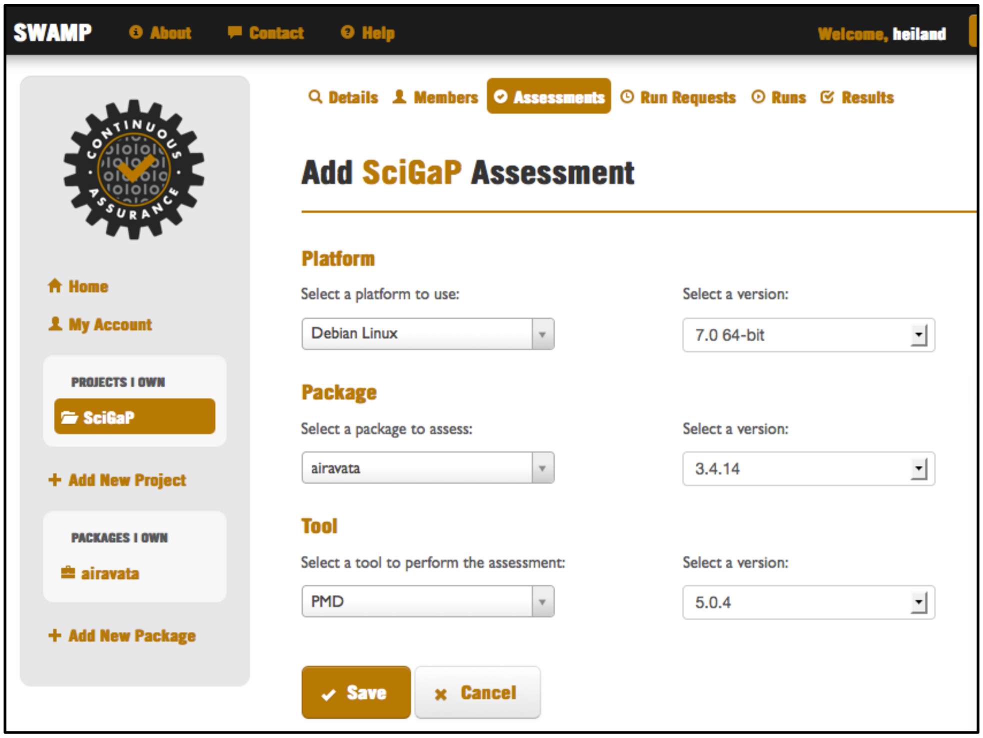
2. Specifies the client will not use the credentials other than they are intended or otherwise disclose them; and
3. Specifies the client will delete credentials once its immediate need for them has been satisfied.

Trust in the client could also be augmented with an audit of its implementation (granted, this is a labor intensive process), or ongoing or occasional checking that it is adhering to an agreed to policy (e.g. the client will demonstrate logs that show how credentials are being used and deleted).

5 Software Assurance

SciGaP is comprised of multiple software packages. Apache Airavata (https://airavata.apache.org/) [17] and Apache Thrift (https://thrift.apache.org/) are two of the most relevant. Airavata is the middleware that orchestrates the workflows from the gateways. Thrift is a software framework that uses an interface definition language (IDL) to generate code for cross-language client-server communication. During our engagement, CTSC analyzed Thrift and created a best practices report, using the Evernote (https://evernote.com) service as a case study [14].

As in any software development project, software assurance is vital to improve security. In addition to developers being mindful of best practices for software security engineering as they write their code, it is also possible to perform programmatic static analysis on large code bases (multiple files). During our engagement, CTSC demonstrated one free online service that does this: the Software Assurance Marketplace (SWAMP; https://continuousassurance.org/). SWAMP can analyze code in several different languages, using multiple static analysis tools. Figure 5 shows how one begins an assessment of a snapshot of the Airavata code. Essentially, a user uploads a software package, specifies the build process, and selects a tool to perform the static analysis. When the analysis is complete, results can either be viewed in the browser or downloaded for further processing.



Add SciGaP Assessment

Platform

Select a platform to use: Debian Linux

Select a version: 7.0 64-bit

Package

Select a package to assess: airavata

Select a version: 3.4.14

Tool

Select a tool to perform the assessment: PMD

Select a version: 5.0.4

Figure 5. Setting up an assessment (static analysis) of Airavata in SWAMP.

Results (using an early version of the Airavata code) listed several "weaknesses" (http://cwe.mitre.org/) that included:

CWE-398 : Indicator of Poor Code Quality
CWE-547: Use of Hard-coded, Security-relevant Constants
CWE-252: Unchecked Return Value
CWE-571: Expression is Always True
CWE-581: Object Model Violation: Just One of Equals and Hashcode Defined
CWE-584: Return Inside Finally Block
CWE-563: Assignment to Variable without Use ('Unused Variable')
CWE-478: Missing Default Case in Switch Statement
CWE-495: Private Array-Typed Field Returned From A Public Method

The point of this demonstration was not to suggest that SciGaP developers fix every single weakness in the code. The point was to make them aware that such tools do exist, are relatively easy to use, and can indeed help improve the security of the project.

6 Conclusion

The SciGaP-CTSC engagement was unique in many ways: the duration of the engagement (unusually long; about 18 months), the frequency of meetings (very *infrequent*, but when we did meet it was in-depth), and the breadth of the security topics that were covered. Through face-to-face meetings, email correspondence, formal reports, and a peer-reviewed paper, CTSC was able to provide the SciGaP project with a better understanding and guidance on relevant security topics, including: trust models, authentication and authorization, identity and access management, and software assurance. In addition, we provided best practices for their use of the Apache Thrift software, demonstrated static analysis for the Apache Airavata software, and commented on the pros/cons of using the WSO2 Identity Server software. We firmly believe our engagement will lead to a more secure SciGaP service.

References

[1] D. Gannon, B. Plale, M. Christie, L. Fang, Y. Huang, S. Jensen, G. Kandaswamy, S. Marru, S. L. Pallickara, S. Shirasuna, Y. Simmhan, A. Slominski, and Y. Sun, "Service Oriented Architectures for Science Gateways on Grid Systems," in *Service-Oriented Computing - ICSOC 2005*, Springer Berlin Heidelberg, 2005, pp. 21–32 [Online]. Available: http://link.springer.com/chapter/10.1007/11596141_3. [Accessed: 27-Apr-2016]

[2] N. Wilkins-Diehr, D. Gannon, G. Klimeck, S. Oster, and S. Pamidighantam, "TeraGrid Science Gateways and Their Impact on Science," *Computer* , vol. 41, no. 11, pp. 32–41, Nov. 2008 [Online]. Available: http://dx.doi.org/10.1109/MC.2008.470

[3] R. Heiland, J. Basney, and V. Welch, "Suggested Security Practices for SciGaP: A Preliminary Report," Jun. 2014 [Online]. Available: http://hdl.handle.net/2022/20811. [Accessed: 23-Apr-2016]

[4] R. Ross, "Managing Information Security Risk: Organization, Mission, and Information System View. NIST Special Publication 800-39," National Institute of Standards & Technology, Gaithersburg, MD, United States, 2011 [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-39/SP800-39-final.pdf

[5] R. Cowles and V. Welch, "XSIM OSG IdM Guidance," Open Science Grid, OSG Document 1199-v2, Jan. 2015 [Online]. Available: http://osg-docdb.opensciencegrid.org/cgi-bin/ShowDocument?docid=1199

[6] "XSEDE User Portal | Manage Allocation." [Online]. Available: https://portal.xsede.org/manage-allocation. [Accessed: 29-Apr-2016]

[7] J. Basney, V. Welch, and N. Wilkins-Diehr, "TeraGrid Science Gateway AAAA Model: implementation and lessons learned," in *Proceedings of the 2010 TeraGrid Conference*, 2010, p. 2 [Online]. Available: http://portal.acm.org/citation.cfm?doid=1838574.1838576. [Accessed: 29-Apr-2016]

[8] R. Cowles, C. Jackson, V. Welch, and S. Cholia, "A Model for Identity Management in Future Scientific Collaboratories," in *International Symposium on Grids and Clouds (ISGC)*, 2014 [Online]. Available: http://www.vonwelch.com/papers/ISGC-XSIM-revised.pdf

[9] T. A. Kanewala, S. Marru, J. Basney, and M. Pierce, "A Credential Store for Multi-tenant Science Gateways," in *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*, 2014, pp. 445–454 [Online]. Available:

http://dx.doi.org/10.1109/CCGrid.2014.95

[10] R. Heiland, S. Koranda, S. Marru, M. Pierce, and V. Welch, “Authentication and Authorization Considerations for a Multi-tenant Service,” in *Proceedings of the 1st Workshop on The Science of Cyberinfrastructure: Research, Experience, Applications and Models*, 2015, pp. 29–35 [Online]. Available: http://doi.acm.org/10.1145/2753524.2753534

[11] V. Welch, J. Barlow, J. Basney, D. Marcusiu, and N. Wilkins-Diehr, “A AAAA model to support science gateways with community accounts,” *Concurr. Comput.*, vol. 19, no. 6, pp. 893–904, 2007 [Online]. Available: http://dx.doi.org/10.1002/cpe.1081

[12] J. Basney and J. Gaynor, “An OAuth Service for Issuing Certificates to Science Gateways for TeraGrid Users,” in *Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery*, 2011, pp. 32:1–32:6 [Online]. Available: http://doi.acm.org/10.1145/2016741.2016776

[13] J. Basney, J. Gaynor, S. Marru, M. Pierce, T. A. Kanewala, R. Dooley, and J. Stubbs, “Integrating Science Gateways with XSEDE Security: A Survey of Credential Management Approaches,” in *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment*, 2014, p. 58.

[14] R. Heiland, S. Marru, M. Pierce, and V. Welch, “CTSC Recommended Security Practices for Thrift Clients: Case Study - Evernote,” May 2014 [Online]. Available: http://hdl.handle.net/2022/20620. [Accessed: 29-Apr-2016]

[15] D. Hardt, “The OAuth 2.0 Authorization Framework,” RFC Editor, Oct. 2012 [Online]. Available: http://www.rfc-editor.org/rfc/rfc6749.txt

[16] T. Fleury, J. Basney, and V. Welch, “Single sign-on for java web start applications using myproxy,” in *Proceedings of the 3rd ACM workshop on Secure web services*, 2006, pp. 95–102 [Online]. Available: http://portal.acm.org/citation.cfm?doid=1180367.1180384. [Accessed: 28-Apr-2016]

[17] S. Marru, L. Gunathilake, C. Herath, P. Tangchaisin, M. Pierce, C. Mattmann, R. Singh, T. Gunarathne, E. Chinthaka, R. Gardler, A. Slominski, A. Douma, S. Perera, and S. Weerawarana, “Apache Airavata: A Framework for Distributed Applications and Computational Workflows,” in *Proceedings of the 2011 ACM Workshop on Gateway Computing Environments*, 2011, pp. 21–28 [Online]. Available: http://doi.acm.org/10.1145/2110486.2110490