

Appears in the Proceedings of the 2005 European Grid Conference (EGC2005), February 14-16, 2005. Science Park Amsterdam., The Netherlands.

The University of Virginia Campus Grid: Integrating Grid Technologies with the Campus Information Infrastructure*

Marty Humphrey and Glenn Wasson

Department of Computer Science, University of Virginia, Charlottesville, VA 22904
{humphrey, gsw2c}@cs.virginia.edu

Abstract. Grid software often unfortunately requires significant changes in existing infrastructure, both in terms of policy and mechanism, instead of accommodating and leveraging existing information servers such as enterprise LDAP servers and enterprise authentication infrastructures. The University of Virginia Campus Grid (UVaCG) has been designed explicitly to re-use as much existing infrastructure in the campus environment as possible in creating a Grid based on the Web Services Resource Framework (WSRF), specifically the Globus Toolkit v4 and WSRF.NET. We report on the design and the current status of the UVaCG, with particular emphasis on the challenge of creating explicit policy expression, negotiation, and enforcement. When fully operational, campus researchers will be able to seamlessly utilize resources within the campus enterprise and expand on-demand to larger Grids such as the TeraGrid and the Open Science Grid.

1 Introduction

In many ways, the core challenge of Grid computing is the ability to seamlessly integrate legacy systems, both in terms of policy and mechanism. That is, while new protocols are often needed for such Grid-specific activities as remote execution and third-party data transfer, the goal of the Grid is to create a virtual computing platform from *existing* heterogeneous systems without requiring the imposition of new software mechanisms and policies that are radically different than those already in place. For example, it is unrealistic to expect sites to abandon an existing Kerberos authentication infrastructure [1] in lieu of some Grid-specific authentication such as the PKI-based GSI [2]. Similarly, resource owners who wish to participate in Grid activities must absolutely be allowed to retain control over their resources in a manner in which

* This work is supported in part by the US National Science Foundation under grants ACI-0203960 (Next Generation Software program), SCI-0438263 (NSF Middleware Initiative), SCI-0123937 (through a subcontract to SURA), the US Department of Energy through an Early Career Grant (Humphrey), and Microsoft Research.

they have previously established. Without this support for local autonomy, the vision of the Grid as vital-yet-commonplace infrastructure will never be achieved.

The “campus Grid” is a compelling environment for Grid activities, because of the inherent opportunity for sharing of resources between campus researchers. Campus researchers should be able to utilize unused campus-wide resources such as the PCs available in general-purpose labs. However, mechanisms are lacking by which to integrate campus IT infrastructure and policies—both explicit and implicit—with the current Grid tools, particularly those based on the Web Services Resource Framework (WSRF [3][4]). Maintaining student security and privacy (as defined by FERPA [5] and increasingly HIPAA [6]) is paramount. While Grid software is certainly being deployed on machines on campuses as part of national and international Grid efforts, these deployments are generally “side-by-side” with the main campus IT infrastructure, particularly that for AAA (authentication, authorization, and accounting). In short, the campus is an excellent candidate for Grids, but only if the campus is committed to it, the Grid software taps into existing campus IT infrastructures, and the Grid deployment is especially careful not to impede pre-existing use patterns by campus researchers and students.

This paper describes the design and current status of the University of Virginia Campus Grid (UVaCG), whose goal is to both create an intra-campus environment of sharing and as such facilitate the broader inter-campus sharing via future participation in Grids such as the Open Science Grid [7] and the TeraGrid [8]. This will be the first Grid in which the Windows machine (via our own WSRF.NET) participates side-by-side with the Linux machine (via the Globus Toolkit v.4). The key challenges will be for the Grid infrastructure to recognize and leverage the pre-existing campus trust fabric and campus policies. We describe the technologies being utilized, the unique challenges being faced as we attempt to integrate a substantial campus IT with the Grid technologies particularly with regard to policy management, and present a status report of the UVaCG. We believe that the UVaCG will be a prototypical example of a new class of Grids -- the enterprise Grid that is relatively self-contained but which interoperates and extends into other Grids on demand. As described in this paper, the key will be expressible, manageable, and negotiable policies on the part of resource owners, service deployers, service invokers, and the virtual organizations themselves.

This paper is organized as follows. In Section 2, we present the technologies being used and deployed for the UVaCG. Section 3 enumerates the challenges being encountered and addressed. Section 4 contains the conclusions.

2 Core Technologies

In this section, we describe the core technologies that are the basis of the University of Virginia Campus Grid (UVaCG). Section 2.1 discusses the Grid-related technologies, Section 2.2 discusses the key components of the Campus IT, and Section 2.3 illustrates how everything fits together.

2.1 Grid Technologies

The foundation of the University of Virginia Campus Grid (UVaCG) is the Web Services Resource Framework (WSRF). WSRF is a set of specifications that describe the relationship between “stateful resources” and web services. This relationship is defined in terms of WS-Resources, an abstraction for modeling/discovering state manipulated by web services. A WS-Resource is a “composition of a web service and a stateful resource” [3] described by an XML document (with known schema) that is associated with the web service’s port type and addressed by a WS-Addressing EndpointReference [9]. WSRF defines functions that allow interactions with WS-Resources such as querying, lifetime management and grouping. WSRF is based on the OGSi specification [10] and can be thought of as expressing the OGSi concepts in terms that are compatible with today’s web service standards [11]. Arguably and simply, it is sometimes convenient when contrasting OGSi and WSRF to think of OGSi as “distributed objects that conform to many Web Services concepts (XML, SOAP, a modified version of WSDL)”, while WSRF is fundamentally “vanilla” Web Services with more explicit handling of state. One artifact of this is that OGSi did not really support interacting with these base Web Services and instead only interacted with “Grid Services” (by definition these were OGSi-compliant); WSRF fully supports interacting with these base Web Services (although the argument is made that client interactions with WSRF-compliant are richer and easier to manage).

Currently, there are 4 specifications [4] in the WS-ResourceFramework with a small number yet to be officially released. WS-ResourceProperties defines how WS-Resources are described by ResourceProperty (XML) documents that can be queried and modified. WS-ResourceLifetime defines mechanisms for destroying WS-Resources (there is no defined creation mechanism). WS-ServiceGroups describe how collections of services can be represented and managed. WS-BaseFaults defines a standard exception reporting format. WS-RenewableReference (unreleased) will define how a WS-Resource’s EndpointReference, which has become invalid, can be refreshed. There are also 3 WS-Notification specifications (WS-BaseNotification, WS-Topics and WS-BrokeredNotification) that although not part of WSRF, build on it to describe asynchronous notification.

UVaCG will rely on two packages that support WSRF and WS-Notification: the Globus Toolkit v4 [12] and our own WSRF.NET [13][14]. We have been working closely with the Globus Toolkit developers to ensure that the two systems are interoperable, both with regard to the core WSRF and WS-Notification specifications and with regard to the Globus protocols.

Many of the existing Windows boxes on campus that are to be included in the UVaCG are already controlled by a Windows domain (username/password) that is maintained by the UVa Campus IT department. To support single sign-on and re-use existing authentication infrastructures, we have developed CredEx [15], which is a general-purpose credential exchange mechanism. The main authentication mechanism will be the campus PKI (described in the next section), and these PKI credentials will be dynamically exchanged to obtain the appropriate username and password for the target (Windows) machine. We would have liked to use MyProxy [16], but MyProxy is not capable of speaking SOAP and WS-Security. By using CredEx, campus re-

searchers can still rely on the Campus IT department to create and maintain accounts; in the UVaCG we are selectively and securely re-using this legacy infrastructure.

2.2 Campus Technologies

In addition to leading a well-run campus infrastructure consisting of network management, user-help desk, account creation, etc., the University of Virginia is very active in the Internet2 community, helping to develop and promote common schemas (such as EduPerson and LDAP Recipe) and PKI procedures for inter-campus interoperability.

Our main focus with the UVaCG is the re-use of campus authentication techniques. Recently, we have pursued the Bridge Certificate Authority (CA) as the basis for scalable Grid PKIs [18]. The Bridge CA is a compromise between a strictly hierarchical PKI and a mesh PKI and achieves many of the benefits of the hierarchical PKI and mesh PKI without with single point of failure of the hierarchical PKI and without the path validation complexity of the mesh PKI. This work is in part based on our previous work contributing to the Internet2/EDUCAUSE HEPKI-TAG group [19] and the EDUCAUSE effort to build and deploy a Higher Education Bridge CA (HEBCA) [20] in the US. Because the University of Virginia runs the PubCookie [21] infrastructure, in which a cookie-based authentication can be used to protect content on web servers (e.g., as an alternative for password-protected course content), we are in the final stages of completing a prototype in which a valid PubCookie can be exchanged (via CredEx discussed in the previous section) for either a username/password for a target machine or for a valid X.509 proxy credential for the Grid.

Our second major focus is the re-use of campus account creation and management. We are currently considering the use of Walden [17] to manage the campus-wide Gridmap file as well as allocate temporary identities and accounts for non-local researchers.

2.3 University of Virginia Campus Grid (UVaCG)

By re-using campus authentication and campus account procedures, the WSRF implementations of the Globus Toolkit and WSRF.NET will be deployed with the minimal amount of disruption for existing campus resource users. The campus Grid resources—Windows machines *and* Linux compute clusters—will be seamlessly available to University of Virginia researchers and their collaborators as appropriate. Our goal is to have campus researchers that acquire new machines want to have these resources included in the campus grid, because they know that they are *not*, in fact, “giving their cycles away” but are instead being given access to an instantaneous pool of resources that they could never acquire merely in their campus lab. Figure 1 shows the UVaCG, illustrating the resources involved and the Grid users. UVaCG initially, will contain five separately-owned Linux clusters and two separately-owned instructional labs containing Windows XP machines.

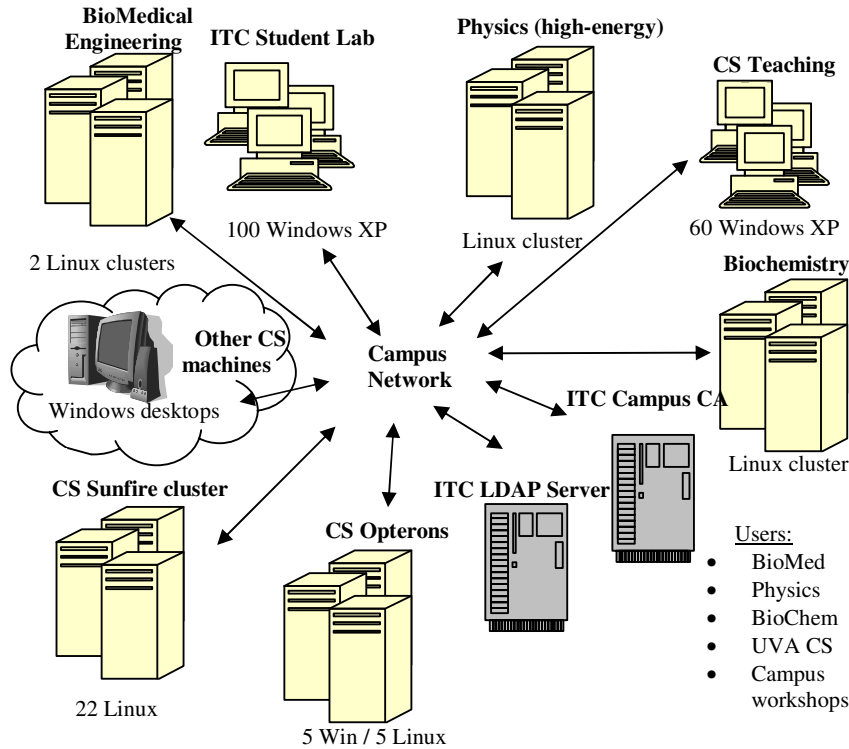


Figure 1: University of Virginia Campus Grid (UVaCG)

3 Explicit Policy Expression, Negotiation, and Enforcement for the Campus Grid

As we make progress on the interoperability between WSRF.NET and the Globus Toolkit v4, and we successfully integrate with the campus IT environment, we are finding it necessary to be able to express policies on these services, both from the viewpoint of the service author and/or deployer, the administrator of the domain in which the service is deployed, as well as the hierarchical policy makers of the Grid (Grid users will also have policies). These policies must be expressed and obeyed. We believe that in today's Grid deployments, policies of Grid resource owners and Grid users are unnecessarily simplistic and homogeneous. For example, machines will often be purchased as part of a funded project and dedicated to "The Grid". Resources will be hand-selected or hard-coded by the Grid user because he/she has had a good experience with that particular resource in the past. We believe that individual Grid users can have individual policies, and they want the Grid automation tools to adhere

to their policies. Grid resource owners also want to explicitly and concisely express their policies, *if even only to make it possible for Grid users (or tools acting on their behalf) to begin to recognize these policies and adhere to them*. Today, this is not possible because policy languages have not been evaluated, policy negotiation tools don't exist, and policy enforcement points are ad hoc, if they exist at all.

We have begun investigating approaches by which Grid users, resource owners, and the Grid as a whole can begin to express these policies and have them adhered to and enforced [22]. We focused only on the *resource provisioning policy*, which describes the Grid-wide distribution of resource utilization. In our approach, we made policy operational by describing actions that will be taken to maintain the desired Grid state. For example, a policy such as “if any site is performing less than 25% of the work of the other sites, all new work will be scheduled on that site until the work load is equalized” describes an explicit *trigger condition* and an *action* that will be taken if that condition is met. Note that while such statements may allow for automated policy enforcement, the Grid's response need not be “fully automated”. That is, the appropriate administrator could be alerted to the relevant condition and/or a set of corrective actions might be suggested, allowing the human to make the final decision.

We have focused our preliminary explorations for the UVaCG on three representative Grid-wide resource provisioning policies:

- Each physical organization opportunistically gives what it can to the Grid [the you-give-what-you-can policy]
- Resource utilization is divided equally among member resources [the 1/N policy]
- Each physical organization receives Grid utilization credit for the resource utilization their physical organization provides to other Grid users outside their physical organization [the you-get-what-you-give policy]

Our prototypical software simulation implemented the “you-get-what-you-give” policy and consisted of three types of software components: *Gatekeeper services*, *Bank services*, and *Enforcement services*. The Gatekeeper service was based on the operations of the Globus gatekeeper and controlled access to the resources. The bank service recorded the “contribution” the physical organizations made to the Grid (based on a hypothetical exchange rate). The enforcement services implemented two kinds of enforcement actions in the Grid, one punitive (“cutoff”) and one corrective (“redirect”). The “cutoff” action is when the enforcer contacts one or more Gatekeepers and instructs them to deny access to their resources to the enforcer's associated user. The “redirect” action involves the enforcer contacting one or more Gatekeepers to ask that resource requests they receive be re-directed to the enforcer's user's resource. This has the effect of providing more credit to a needy user.

While arguably simplistic, we were able to conduct experiments in which we refined the issues and approach and began to quantify the costs and benefits of attempting to enforce Grid policy in this manner. In our experiments, we were able to dynamically shift the workload to meet the resource-sharing policy, but only under highly-constrained conditions. *We believe that policy expression, negotiation, and enforcement is an extremely difficult problem that must be addressed comprehensively in Grids*. Specifically, we are extending this work in the following ways for the UVaCG:

- **Expressible policies.** We will make representative policies for resource owners and Grid users using a combination of WS-Policy, WS-SecurityPolicy, and XACML. Our previous policies were in our own ad-hoc language.
- **Policy enforcement in WSRF.NET.** While we plan to utilize the Microsoft WSE as much as possible for policy enforcement, and we suspect that much of the policy enforcement may only be performed via application-specific logic, we believe that the WSRF.NET hosting environment that we are constructing can perform policy enforcement for the entire PC. An important first application of this is to enforce the resource owner’s policy (such as “If a local user is interacting with this PC, no Grid activities can execute on this PC.”) We will create comprehensive, integrated logging for post-facto analysis.
- **Prototype tools that understand explicit policies.** The first tool that we will develop will be a scheduler that understands explicit policies. A core capability of this tool will be the hypothetical exploration of the Grid via “What if...?” questions, such as “What if I tried to execute my parameter-space job at 5pm? Where could I be allowed to execute?” A related tool will be used to understand the effects of policy on the Grid. For example, we anticipate situations in which policies are so restrictive (e.g., by a resource owner) that *no* services are engaged or *no* jobs are executed. Breaking the “inertia” of first Grid deployment, as well as determining the steady-state performance/behavior of the Grid will be very challenging problems. We need tools to resolve these policy conflicts and make suggested modifications.

We are leveraging existing work to express, negotiate, and enforce the policies of Campus Grid users and resource providers. To a certain extent, security authorization policy has been treated explicitly in Grid contexts in Akenti [23], CAS [24] and the work by Keahey and Welch [25]. These systems make permit/deny decisions based on (potentially multiple) access control policies and the accessor’s identity / group membership. While this project will leverage these excellent projects, this existing work does not immediately satisfy the requirements of the UVaCG because all focus on authorization policy, whereas we need to address policies more broadly and in the context of Web Services (e.g., “I prefer predictable Grid behavior rather than having 9 out of 10 jobs finish very quickly and the last job finish incredibly slowly”). Also, the referenced work is generally applicable to a *particular service*, whereas the problem we are addressing encompasses *per-Grid-user*, *per-Service*, and *Grid-wide*. For example, the campus IT administrators may impose requirements on the “holistic” behavior of the Grid that is not merely satisfied through the individual actions of the services and the Grid users.

The UVaCG leverages research into explicit policy management that has been performed outside of the context of the Grid. Preconditions and obligation in policy [26] provide a compact means of representing what a user must do *before* performing a specific action as well as what they must do *after* an action. The Generic Authentication, Authorization, and Accounting (AAA) Architecture (RFC 2903[27]) builds an

architecture in terms of Generic AAA servers, policy and event repositories, and Application Specific Modules (ASMs). It is not clear how this architecture can be applied to policies broader than AAA. The specification is also not prescriptive in how to best *implement* such an architecture based on a particular technology such as Web Services.

Within the context of OGSA, the WS-Agreement specification proposes support for service management, which is “the ability to create Grid services and adjust their policies and behaviors based on organizational goals and application requirements”[28]. This is a form of Service-Level Agreement (SLA) for OGSA. WS-Agreement will be leveraged as much as possible to frame the work described in this proposal. While WS-Agreement does not suggest actual policies for Grid users, Grid Services, and the Grid as a whole (which is a contribution of this work), we anticipate using aspects of WS-Agreement.

4 Conclusion

The campus Grid presents a unique set of challenges and opportunities with regard to Grid Computing. Explicit policy management and negotiation is needed in order to support the dynamic environment. We are currently continuing to advance the interoperability of WSRF.NET and the Globus Toolkit v4 and have begun initial deployment on the UVaCG.

We are also separately working with the San Diego Supercomputing Center (SDSC) to develop support for the campus researcher to more easily handle the stringent security requirements of SDSC, NCSA, and the TeraGrid. We are hardening our Bridge CA work, whereby campuses that meet the security requirements of the TeraGrid can “cross-certify” with other CAs that the TeraGrid recognizes. This will, in principle, allow campuses to utilize their Campus authentication infrastructure as the basis of a single sign-on capability.

References

- [1] B.C. Neuman, and T. Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications Magazine*, 32(9):33-38, September 1994.
- [2] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. A Security Architecture for Computational Grids. *Proc. 5th ACM Conference on Computer and Communications Security Conference*, pg. 83-92, 1998.
- [3] K. Czajkowski., Ferguson, D., Foster, I., Frey, J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S., Vambenepe, W. 2004. The WS-Resource Framework. <http://www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf>
- [4] WS-ResourceFramework and WS-Notification Specifications. <http://devresource.hp.com/drc/specifications/wsrf/index.jsp>
- [5] Family Educational Rights and Privacy Act (FERPA). US Department of Education. <http://www.ed.gov/policy/gen/guid/fpco/ferpa/index.html>

- [6] United States Department of Health and Human Services. Office of Civil Rights – HIPAA. <http://www.hhs.gov/ocr/hipaa/>
- [7] Open Science Grid. <http://www.opensciencegrid.org/>
- [8] TeraGrid. <http://www.teragrid.org>
- [9] IBM, BEA, and Microsoft. WS-Addressing. 2004. <http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnglobspec/html/ws-addressing.asp>
- [10] S. Tuecke et. al. Open Grid Services Infrastructure (OGSI) Version 1.0. Global Grid Forum. GFD-R-P.15. Version as of June 27, 2003.
- [11] K. Czajkowski, Ferguson, D., Foster, I., Frey, J., Graham, S., Snelling, D., Tuecke, S., From Open Grid Services Infrastructure to Web Services Resource Framework: Refactoring and Evolution, 2004. <http://www-106.ibm.com/developerworks/webservices/library/ws-resource/grogsitowsrf.html>
- [12] Globus Toolkit v. 4. <http://www.globus.org/wsrfl/>
- [13] WSRF.NET: The Web Services Resource Framework on the .NET Framework. <http://www.ws-rf.net>
- [14] Humphrey, M., G. Wasson, M. Morgan, and N. Beekwilder (2004). An Early Evaluation of WSRF and WS-Notification via WSRF.NET. *2004 Grid Computing Workshop (associated with Supercomputing 2004)*. Nov 8 2004, Pittsburgh, PA.
- [15] D. Del Vecchio, J. Basney, N. Nagaratnam, and M. Humphrey. CredEx: User-Centric Credential Selection and Management for Grids. University of Virginia Computer Science Technical Report. November, 2004.
- [16] J. Novotny, S. Tuecke, and V. Welch. An Online Credential Repository for the Grid: MyProxy. Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.
- [17] Kirschner, B., Adamson, W., Hacker, T. , Athey, B., Walden: A Scalable Solution for Grid Account Management, *2004 Grid Computing Workshop (associated with Supercomputing 2004)*. Nov 8 2004, Pittsburgh, PA.
- [18] J. Jokl, J. Basney, and M. Humphrey. Experiences using Bridge CAs for Grids. UK Workshop on Grid Security Experiences, Oxford 8th and 9th July 2004.
- [19] Higher Education PKI Technical Activities Group (HEPKI-TAG). <http://middleware.internet2.edu/hepki-tag/>
- [20] Higher Education Bridge Certificate Authority (HEBCA). <http://www.educause.edu/hebca/>
- [21] Pubcookie. <http://www.pubcookie.org>
- [22] G. Wasson and M. Humphrey. Policy and Enforcement in Virtual Organizations. In *4th International Workshop on Grid Computing (Grid2003)* (associated with Supercomputing 2003). Phoenix, AZ. Nov 17, 2003.
- [23] M. Thompson. 2001. Akenti Policy Language. <http://www-itg.lbl.gov/security/Akenti/Papers/PolicyLanguage.html>
- [24] L. Pearlman, V. Welch, I. Foster, C. Kesselman, S. Tuecke. A Community Authorization Service for Group Collaboration. *Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, 2002.
- [25] K. Keahey, V. Welch. Fine-Grain Authorization for Resource Management in the Grid Environment. *Proceedings of Grid2002 Workshop*, 2002.
- [26] N. Dulay, Lupu, E., Sloman, M. and Damianou, N. 2001. A Policy Deployment Model for the Ponder Language. *Proc. IEEE/IFIP International Symposium on Integrated Network Management (IM'2001)*

- [27] C. de Laat et. al. Generic AAA Architecture. RFC 2903. Available at: <http://www.faqs.org/rfcs/rfc2903.html>
- [28] K. Czajkowski, A. Dan, J. Rofrano, S. Tuecke, and M. Xu. Agreement-based Service Management (WS-Agreement). Global Grid Forum draft-ggf-graap-agreement-1. Version as of Feb 8, 2004.