Toward SVOPME, a Scalable Virtual Organization Privileges Management Environment

Nanbor Wang[1], Gabriele Garzoglio[2], Balamurali Ananthan[1], Steven Timm[2], Tanya Levshina[2]

[1] Tech-X Corporation, 5621 Arapahoe Ave, Suite A, Boulder, CO 80303, USA

[2] Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL, 60510, USA

Abstract Grids enable uniform access to resources by implementing standard interfaces to resource gateways. In the Open Science Grid (OSG), privileges are granted on the basis of the user's membership to a Virtual Organization (VO). However, individual Grid sites are solely responsible to determine and control access privileges to resources. While this guarantees that the sites retain full control on access rights, it often leads to heterogeneous VO privileges throughout the Grid and hardly fits with the Grid paradigm of uniform access to resources. To address these challenges, we developed the Scalable Virtual Organization Privileges Management Environment (SVOPME), which provides tools for VOs to define, publish, and verify desired privileges. Moreover, SVOPME provides tools for grid sites to analyze site access policies for various resources, verify compliance with preferred VO policies, and generate directives for site administrators on how the local access policies can be amended to achieve such compliance without taking control of local configurations away from site administrators. This paper describes how SVOPME implements privilege management tools for the OSG and our experiences in deploying and running the tools in a test bed. Finally, we outline our plan to continue to improve SVOPME and have it included as part of the standard Grid software distributions.

Acknowledgments The SVOPME project is partially funded by the Office of Advanced Scientific Computing Research, Office of Science, United States Dept. of Energy under contracts DE-FG02-07ER84733 and DE-AC02-06CH11357, the Fermi National Accelerator Laboratory, and the Tech-X Corporation. Fermilab is operated by Fermi Research Alliance, LLC under Contract DE-AC02-07CH11359 with the U.S. Department of Energy.

1 Introduction

The Grid computing environment has emerged as the leading technology for coordinated resource sharing among participating institutions and individuals. It enables execution of large-scale computation jobs by providing uniform access to distributed resources such as computational cycles and data storage, shared among participating institutes. The Virtual Organization (VO) is a key concept in grid computing. A VO manages members from different home institutes with common interests. Multiple VO's can coexist and share a common set of resources in a Grid. Meanwhile, the structure and membership of a VO are dynamic, as groups and individuals may join and leave VO's based on their interests and needs.

Grid middleware aims to provide uniform access to all the resources made available at various distributed sites for members of a VO. The Grid Security Infrastructure (GSI) [5] provides the core security capabilities such as secure communication and mutual authentication between users and resources, using X.509 certificates. A VO is responsible for managing user memberships to groups and roles according to its organizational structure [7]. Individual sites enforce the resource usage privileges according to VO specifications through underlying OS' access control mechanisms.

1.1 Challenges in Reconciling VO and Site Policies

Within a Grid body such as the Open Science Grid (OSG) [1] or European Grid for E-sciencE (EGEE) [2], a VO establishes resource-usage agreements with Grid resource providers to grant access of site resources to group of users within a VO. Modern Grid middleware provides both the mechanisms and tools to enable the fine-grained, role-based access control. However, it comes up short in providing a streamlined and consistent distributed user privilege management across VO's and sites. Currently, this lack of automatic policy instantiation/reconciliation mechanism is handled manually via verbal discussions between VO administrators and site administrators. Such manual propagation of VO policies is a brittle and time-consuming process. As privilege policies change dynamically, which is becoming more common for large VO's, and new VO's are onboard, Grid utilization suffers as legitimate users may not be able to access resources which are otherwise perfectly usable.

1.2 The Need for Managing VO User Privileges

To realize the vision of providing uniform access to distributed resources in Grid Computing, there is an urgent need to bridge the gap between VO privilege policy

specifications and local Grid site configurations. VO user roles and privilege policies must be able to propagate to Grid site automatically, yet allowing site administrators to retain full control over site policies. Furthermore, with the ever changing numbers of VO's, organizations, and privilege policies, there need to be tools to help VO and site administrators alike to verify that policies at VO's and sites are consistent with one another.

2 Related Works

SVOPME project is synergistic to many projects on authorization management. For example, the GPBox [13] project is a policy management framework for the Grid environment to globally modify the execution priorities of jobs submitted from VOs at sites. Compared to GPBox, SVOPME project does not attempt to configure site policies directly. Instead, SVOPME produces compliance reports about local configurations that hint on how the configurations could be modified for the site to provide better support for VO's. We believe that leaving local site administrators in full control of site configuration will give them peace of mind and reduce their resistance toward the eventual adoption of SVOPME.

Another synergistic effort to the SVOPME project is the EGEE Authorization Service [12]. Similar to SVOPME, the EGEE Authorization Service aims to provide consistent authorization decisions for distributed services over the Grid. It provides software components for defining privilege policies at services. These policies are then used to answer queries about whether a particular action is permissible by certain users. Although the EGEE Authorization Service also aims at providing a set of consistent authorization policies over the Grid, unlike SVOPME, the new Authorization Service does not focus on the VO policies. The two projects will be able to leverage the work done by each other.

Another effort closely related to SVOPME is the Authorization Interoperability project [3]. This project defines an attribute and obligation profile for authorization interoperability across Grids as described in Section 3. We will leverage the efforts from this project to integrate SVOPME into OSG and other Grid infrastructure.

3 VO Policies and Grid Sites

Fig 1 illustrates the security model in OSG. Other modern Grid software stacks, such as EGEE, also adopt similar security models [4]. The figure depicts the procedures for performing authentication on the VO side and authorization on Grid sites. The Authorization Interoperability project standardizes the terms and formats used in the authorization process between security components, specifically Policy-Enforcement-Point (PEP), such as gLexec, and Policy-Decision-Point

(PDP) [10], such as GUMS [11]. This profile is based on the eXtensible Access Control Markup Language (XACML) [8] and the Security Assertion Markup Language (SAML) [9]. However, as highlighted by **Fig. 1**, the existing Grid security model does not provide support for a policy-administration-point (PAP), i.e., how a VO can define its privilege policies. The SVOPME project, therefore, fills this gap and utilizes XACML as its VO privilege policy definition language for administering policies over the Grid.

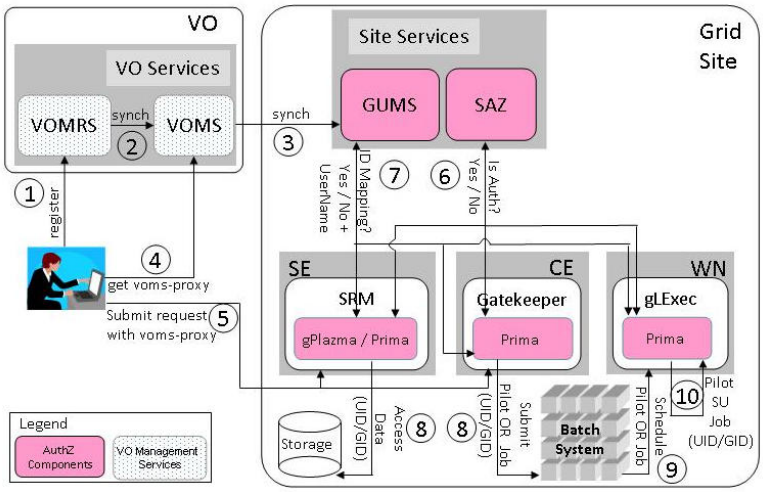

Fig. 1 The OSG Security Model.

4 The SVOPME Architecture

We develop the Scalable Virtual Organization Privilege Management Environment (SVOPME) to address the challenges in administering and maintaining user privileges over multiple VOs and grid sites. Built on top of the existing OSG privilege infrastructure, the SVOPME project is an extension to the OSG VO Services project [3]. The work fills a common gap that many Grid privilege management infrastructures share, as we pointed out in Section **Error! Reference source not found.**. **Fig. 2** illustrates the main data entities and operations performed in SVOPME that help VO administrators and grid site administrators maintain a consistent view of privileges across the Grid. Furthermore, SVOPME identifies inconsistencies in user privileges and help transform them into appropriate site configurations. These key data entities include:

- **VO Privilege Policies:** A VO administrator defines a set of VO's organizational privilege policies for users of different groups and roles.

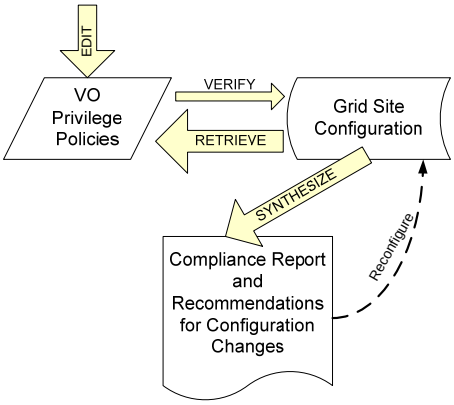

Fig. 2 SVOPME Helps Distribute and Realize VO Privilege Policies.

- **Site Configurations:** A collection of all relevant Grid site system and software configurations that enforce the local privilege policies. These configurations include the local user account settings, local user group setting, identity mapping (GUMS) configurations, batch system configuration, storage element configurations, etc. Like the VO Privilege Policies, SVOPME will also need to codify and transform these configurations into a document that can be reasoned on and compared against VO policies.
- **Compliance Report and Recommendations for Site Configurations:** SVOPME generates this document on each Grid site to provide
  - detailed evaluation of how many of the VO policies are supported and,
  - recommendations on how to modify the local configurations to fully support VO privilege policies for site administrators.

  Note that the site administrators maintain full control over their site configurations and need to reconfigure the site manually.

Error! Reference source not found. illustrates the architecture of the SVOPME tools that we developed. Four key components provide the core functionality of SVOPME. They include support for VO administrators to define and generate VO privilege policies, and for site administrators to automatically generate local site privilege policies based on existing site configurations.

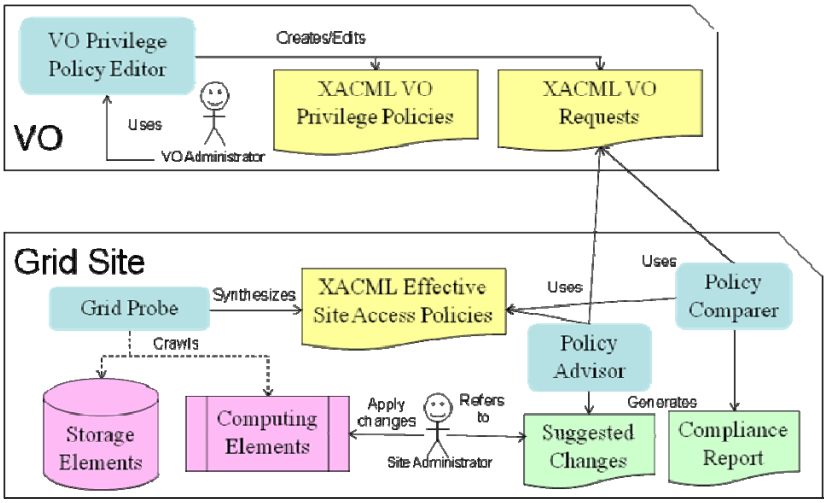

Fig. 3 Overall SVOPME Architecture.

To allow easy extension of the policies supported, we design the tools to use policy templates. The use of policy templates allows us to support new policy types by simply adding new templates into the tools chain without modifying the core implementations. SVOPME currently supports the following common VO privilege policies:

- Account mapping policies allow a VO to define how users of a particular group or role should be mapped to local accounts. This mechanism grants members of the same group the inherent OS-level protections at Grid sites. One policy is for all users of a VO group or role to be mapped to a single shared local "group account". Alternatively, a pool account policy maps users of a specific VO group or role into one of a pool of local accounts.
- Relative priority policies allow a VO to specify that jobs submitted by a particular VO group or role should be executed with higher priority than those submitted by another group or role. For example, a VO may want to grant the highest execution priority to jobs from members of the production team to ensure highest throughput in producing science.
- Pre-emption policies define if jobs submitted from a group or role should be allowed to run for consecutive hours without pre-emption to ensure quick turnaround time of these tasks.
- Permission policies define if users from a specific group or role are allowed to access certain storage areas. For example, a VO may want to grant only users playing the "software administrator" role the permission to install software into the $OSG_APP area for the VO.
- UNIX group sharing policies allow a VO to define finer grained permission management. For example, a VO may want to specify that local accounts of

two groups share the same group ID on Grid sites to ensure that they can freely exchange data, if needed.

- Job suspension policies let a VO to specify if jobs submitted from a particular group or role are not to be suspended.
- Data privacy policies allow a VO to specify that files created by a group of users should be accessible only by that group by default.
- Disk quota policies allow a VO to specify the maximum disk usage by a group of users.

The remainder of this Section describes these tools in more details.

4.1 VO Components

Fig. 4 illustrates all the utilities SVOPME provides for VO administrators. In the core of VO supports is the **XACML VO Policy Editor**. As we mentioned earlier, the VO Policy Editor uses XACML as the internal representation of privilege policies. This provides a generic mechanism for describing, combining, and reasoning with policies. However, XACML as a language is too complex and verbose for VO administrators to express their policies. Furthermore, XACML is a generic language for defining privilege policies and defines a very limited set of standard attributes about resources, actions, etc. Each community, therefore, needs to define a vocabulary (i.e., XACML profile) to frame the concepts in its domain and VO administrators may not be expert in it.

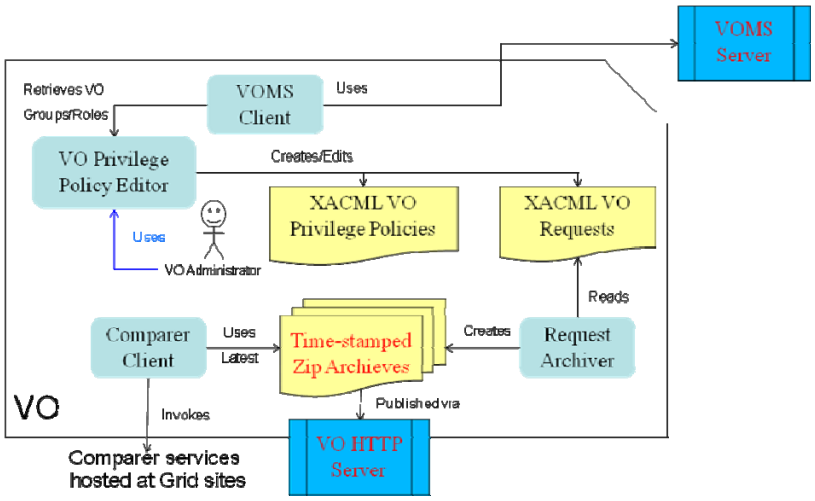

Fig. 4 VO Tool interactions.

In order to address these issues, we develop a “domain-specific” GUI-based VO policy editor. The editor enables its users to create individual privilege policies as separate XACML files. As opposed to a generic GUI-based editor for arbitrary XACML documents, our VO Policy Editor predefined a set of VO policy types and attributes that users can generate and edit.

VO administrators can then use the editor to create new VO privilege policies by selecting from a list of pre-defined policy types and filling in the key information for the type of policy being created. For example, when defining an account mapping policy, an administrator only needs to specify the kind of users (their group and role attributes) to which the policy will be applied and the kind of account to which these users should be mapped (group or pool account). Furthermore, VO Policy Editor is now capable of detecting and prevent user from redundant and contracting policies. A utility tool called ``VOMS Client'' [6] in **Fig. 4** contacts the VO's VOMS server to retrieve the VO structural information and offers the information to the editor. This alleviates users from having to remember and type out all the group and role combinations.

For every VO policy defined by the VO Policy Editor, a matching XACML verification request is generated. These requests define the operations that *must be permitted* at a Grid site supporting the VO, and are used to test the compliance of Grid sites. A “Request Archiver” enables a VO to package all these test queries into a time-stamped archive, which can then be published on the VO’s web site. Timestamps allow other tools to retrieve only the latest set of test queries.

The “Comparer Client” provides VO administrators and users access to the policy comparer Web Services hosted on Grid sites. Users can use the comparer client contact to verify the degree of VO privilege support at individual sites using a set of test requests.

4.2 Site Components

This Section describes the 3 site-specific components in SVOPME, namely, Grid Probe, Policies Advisor, and Policies Comparer.

4.2.1 Grid Probe

Mechanisms for enforcing Grid site privilege policies currently are scattered at different locations on a Grid site. There is no one centralized entity to manage and configure existing Grid middleware infrastructure. For example, the VO group and role to local user ID mapping is managed by the Grid site's GUMS database. However, setting up local user ID and group ID, which are the basic subject entities for enforcing local policies, has to be done via OS tools. Batch system and storage elements, too, have their own configuration points to control privileges such as priority and permissions.

To try to compare and reason on VO policies directly with all these configuration points will result in *ad hoc* software tools that are very complicated and hard to maintain and expand. To address this issue, we developed the Grid Probe tool that scans and gathers configuration information from various tools and mechanisms. The Grid Probe then analyzes this information and generates the *effective local privilege policies* in XACML.

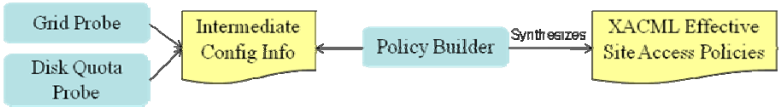

Fig. 5 SVOPME's site mechanisms for synthesizing effective site policies.

One contentious issue in deploying tools like SVOPME is the need to run the probing with special privileges. For example, to scan the disk quota configurations for all relevant local user accounts, the probe needs to have root privilege. As a result, local site administrators may become very suspicious about the adoption of SVOPME site tools. Moreover, different sites may have different configuration strategies. To address these concerns, SVOPME separates the actual probing components into multiple, small scripts/programs that are easy for administrators to examine. These probes can be scheduled to run periodically using accounts with minimal privileges required. The probes collect the site configurations into intermediate files which are used by the Policy Builder component to synthesize the actual equivalent site XACML policies. The approach of using customized probes also allows SVOPME to easily adapt to the differences in site configuration strategies.

Current Grid Probe implementations support the scanning of the following configuration points:

- GUMS configurations provide mappings from user identity and VO's group and role membership to a specific local GID and UID. GUMS mappings are used to generate account type and mapping policies. Many other policies also require information of local user identification.
- Unix group memberships of various users are needed to determine if they can share information.
- Unix directory permissions are needed to determine a series of privilege policies such as software installation and user data privacy.
- Condor configurations are used to determine site policies in job execution priority, pre-emption, and suspension/resumption privileges.
- Disk quota of all local users.

4.2.2 Analyzing Site Configurations

SVOPME uses the verification queries generated by the VO policy editor to verify if policies defined by a VO are supported on a site. Using the VO Request Retriever, the site checks periodically if a VO has published a new set of verification queries, and downloads them when necessary. As shown in **Fig. 6**, the requests for VOs are cached locally at Grid sites and are used to verify the site configurations as described in the remainder of this Section.

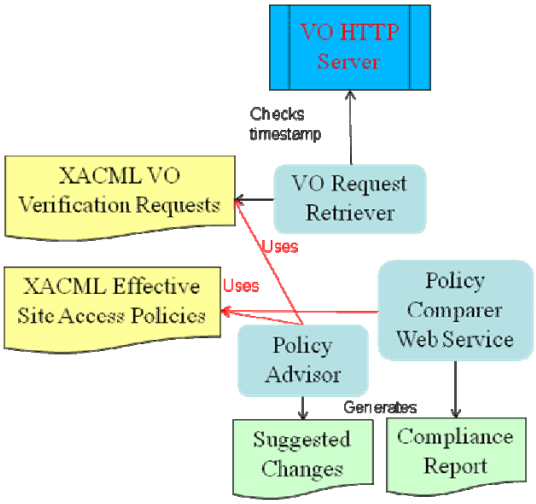

Fig. 6 VO requests are used to verify site configurations.

4.2.2.1 Policy Advisor

The Policy Advisor runs on Grid sites. It verifies if the VO privilege policies are supported at the site by issuing VO verification requests against the local privilege policies, generated by the Grid probe. If the site fails to comply with a specific policy, the Policy Advisor will analyze and recommend a way to modify the site configuration to correct the problem.

4.2.2.1 Policy Comparer

The Policy Comparer is a Web/Grid Service invoked by a VO administrator or any VO user to verify the degree of support of the site of their VO privilege policies. Site compliance test can be done by using the Policy Comparer Client VO tool with a set of test queries to check if the VO policies are supported at the site. By providing the Policy Comparer Web Service, we avoid publishing the site configurations information in the form of site equivalent policies while allowing the

VO users to verify site compliance by producing a pass/fail response to each query.

5 Experimental Deployments

Since our last report in iSGC 2009, we continued to enhance and harden the SVOPME implementation. In particular, we defined and implemented how verification requests flow among many VO's and grid sites through standardized mechanisms. Furthermore, we deployed the SVOPME tools in a realistic, large-scale Grid environment using FermiGrid's integrated test bed (ITB). To evaluate the effectiveness of SVOPME, we gathered and defined the VO policies for the DZero and the Engage VO's of OSG as our target VOs. The experiment motivated several enhancements in Grid tools. Furthermore, we were able to identify some inconsistent and unconventional site configurations in our target site environment.

Some of the differences are due to legacy site configurations known to the administrators. More importantly, SVOPME was able to identify some inconsistencies in the ITB that were unknown to the site administrators. This experiment demonstrated the potential benefits brought by SVOPME to managing a Grid. If deployed in large scale, we believe that we will be able to further demonstrate how the SVOPME project addresses the scalability issues in providing consistent resource usage over the Grid.

6 Project Outlook and Conclusions

To address the scalability issues in providing consistent access to Grid resources, we develop a set of tools and services to realize a Scalable Virtual Organization Privilege Management Environment. We have demonstrated the feasibility and the effectiveness of this project in a test bed environment to allow VOs and sites to communicate the VO privilege policy needs and to verify the degree of site support automatically. Fully deployed, the SVOPME project can greatly reduce the costs in running and maintaining VO's and sites alike. Similarly, SVOPME allows sites to advertise and prove their degree of support for a VO. For a site to support a new VO and its privilege policies, there are now semi-automatic mechanisms to amend site configurations. Equally important is that Grid sites do not relinquish the privilege enforcement to the VOs. Rather, SVOPME informs the site administrators with a formal VO policy assessment.

We are currently soliciting VO's and sites interested in testing out SVOPME in a production environment. We will continue to enhance and harden SVOPME tools based on the feedbacks and experiences from these early adopters. For example, we may implement the following policies previously discussed for future extension based on the needs:

- Policies on job resubmission semantics instruct the underlying batch system to execute jobs from a specific group or role at most once. This is particularly important in some high energy physics data processing, where the workflow must guarantee that output data is not duplicated.
- Network policies allow a VO to request outbound network connections for jobs submitted by a group of users.
- File retention policies defines for how long files owned by users of a specific group or role should be kept in the storage.

We intend to have SVOPME incorporated in the Virtual Data Toolkit (VDT), the *de facto* standard Grid middleware distribution.

References

[1] Pordes R et al. 2007 The Open Science Grid *Journal of Physics: Conference Series* 78 15

[2] Laure E et al. 2004 Middleware for the next generation Grid infrastructure Proceedings of Computing in High Energy Physics and Nuclear Physics 2004, Interlaken, Switzerland 826

[3] Garzoglio G et al. 2009 Definition and Implementation of a SAML-XACML Profile for Authorization Interoperability across Grid Middleware in OSG and EGEE *Journal of Grid Computing* DOI: 10.1007/s10723-009-9117-4

[4] Garzoglio G et al. 2009 An XACML profile and implementation for Authorization Interoperability between OSG and EGEE *Proceedings of Computing in High Energy Physics and Nuclear Physics 2009*, Prague, Czech Republic

[5] Foster I and Kasselman C 1997 Globus: A Metacomputing Infrastructure Toolkit *International Journal of Supercomputer Applications*, 11(2) 115-128

[6] Alfieri R et al. 2004 VOMS, an authorization system for virtual organizations *Proceedings of European across Grids conference No1, Santiago De Compostela, Spain* 2970 33-40

[7] Ceccanti A, Ciaschini V, Dimou M, Garzoglio G, Levshina T, Traylen S, Venturi V 2009 VOMS/VOMRS Utilization patterns and convergence plan *Proceedings of Computing in High Energy Physics and Nuclear Physics 2009*, Prague, Czech Republic

[8] Moses T et al. 2005 Extensible access control markup language (xacml) version 2.0 *Oasis Standard*

[9] Cantor S, Kemp J, Philpott R, Maler R 2005 Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2. 0 *OASIS SSTC*

[10] Garzoglio G et al. 2008 An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids *Fremilab White Paper* CD-doc-2952-v2

[11] Lorch M, Kafura D, Fisk I, Keahey K, Carcassi G, Freeman T, Peremutov T, Rana A S 2005 Authorization and account management in the Open Science Grid *The 6th IEEE/ACM International Workshop on Grid Computing*, 2005

[12] The EGEE Authorization Service: `http://twiki.cern.ch/twiki/bin/view/EGEE/AuthorizationFramework` Accessed on May 13, 2009

[13] Cesini D, Ciaschini V, Dongiovanni D, Ferraro A, Forti A, Ghiselli A, Italiano A, Salomoni D 2008 Enabling a priority-based fair share in the EGEE infrastructure *Journal of Physics: Conference Series* 119 062023 DOI:10.1088/1742-6596/119/6/062023