Streamlining Grid Operations:
Definition and Deployment of a Portal-based User Registration Service

Ian Foster[1,2]   Veronika Nefedova[1]   Lee Liming[1]   Rachana Ananthakrishnan[1]
Ravi Madduri[1]   Laura Pearlman[3]   Olle Mulmo[4]   Mehran Ahsant[4]

[1] Argonne National Laboratory, Argonne, IL, 60439

[2] University of Chicago, Chicago, IL, 60637

[3] Information Sciences Institute, University of South California, Marina del Rey, CA 90292

[4] Kungliga Tekniska Högskolan, Nada, 100 44 Stockholm, Sweden

Abstract

Manual management of public key credentials can be a significant and often off-putting obstacle to Grid use, particularly for casual users. We describe the Portal-based User Registration Service (PURSE), a set of tools for automating user registration, credential creation, and credential management tasks. PURSE provides the system developer with a set of customizable components, suitable for portal integration, that can be used to address the full lifecycle of Grid credential management. We describe the PURSE design and describe how it has been used within portals for two different systems, the Earth System Grid data access system and the Swegrid computational grid. In both cases, the user is entirely freed from the need to create or manage public key credentials, thus simplifying their Grid experience and reducing opportunities for error. We argue that this capturing of common use cases in a reusable "solution" can be a model for how Grid ease-of-use can be addressed in other domains as well.

1   Introduction

A typical Grid application requires that a set of users share resources of various kinds in some controlled manner. To this end, many extant Grid deployments use the public-key infrastructure (PKI)-based Grid Security Infrastructure (GSI) [10] as a basis for secure user single sign on and subsequent authentication of users and resources prior to authorization. GSI defines and implements useful algorithms for authentication and delegation. However, the tasks of creating and managing the PKI credentials used by GSI can be significant sources of complexity, user difficulty, and even error (and thus insecurity) in Grid deployments.

These considerations motivate our design of the Portal-based User Registration Service (PURSE), a set of tools for developing portal-based systems that automate user registration, the creation of PKI credentials, and subsequent credential management. A typical PURSE-based portal allows a user to register via a Web page, follow which a credential is created and managed on their behalf, with subsequent access provided via a username and password. A separate administrator interface allows a portal administrator to approve requests, revoke credentials, and so forth. By streamlining and codifying these various steps, PURSE-based systems can significantly reduce barriers to the integration of new users, overheads associated with credential management, and opportunities for error—and thus simplify the development of usable Grid applications.

An important PURSE design goal was to support the creation and use of PKI credentials of varying "quality." It is often the case that different access control policies are associated with

different resources and operations. For example, some operations and resources (e.g., write access to archival storage) may require stringent verification of the identity and/or attributes of a requestor, while others (e.g., read access to Web pages) require only audit of a weakly authenticated identity. The definition and enforcement of such policies can be a significant source of complexity in Grid application deployments, due to the need not only to implement policies correctly but also to achieve appropriate tradeoffs between operational security and ease of use. Thus, PURSE mechanisms allow for the automatic creation of credentials following either simple online registration or stringent identity verification, and for the upload of existing credentials.

The PURSE implementation is not particularly complex, being based on an integration of a number of existing components, including GSI libraries, the MyProxy online credential repository, the SimpleCA credential generator, and portal tools. This implementation approach of integrating existing components to construct a reusable "solution" that addresses an important set of use cases is one that we hope will be pursued by many other Grid developers.

We have recently become aware of the Grid Account Management Architecture (GAMA) project [1, 9], which has produced similar mechanisms in parallel with our PURSE development. GAMA differs from PURSE in various mostly minor respects: for example, it is hosted on GridSphere rather than Axis, and does not support uploading of existing credentials, a critical requirement for all PURSE users to date. We view this parallel evolution as demonstrating the importance of this technology.

The rest of this paper describes in turn the PURSE system (Section 2), two different PURSE-based portals (Section 3), and the sample registration portal distributed with PURSE (Section 4). We conclude in Section 5.

2   System Description

The PURSE user registration system is a collection of Java APIs designed to work as a backend for a front-end user interface, typically a web portal, to ease registration and credential management. Driven by user requests through the interface, this Java code stores user contact information, generates and stores new credentials for users, and allows for subsequent use of those credentials to access Grid resources. The system has functionality to support credential renewal and revocation. This functionality can be accessed through a well-defined API and is easily configurable.

The system is built upon some common tools, as follows:

- A JDBC-compliant database is used to persist user data. (MySQL is currently used.)

- A Certification Authority is used to generate and sign user credentials. Depending on application requirements, either SimpleCA [6] or an external CA can be used for generating and signing users credentials.

- The MyProxy server [3, 11] is used to store user credentials

- JavaMail [2] is used to send and receive notifications to the user and CA operator.

2.1   Typical Usage Scenarios

A PURSE user must first register with the PURSE system. This is a one-time event that must precede any other use of the system. Registration involves three principal steps, as follows.

1. The user accesses the registration page on the portal and enters relevant information (e.g., contact information, desired user name, desired password).

2. PURSE persists the user information and, using the provided contact information, sends an email back to the requesting user requesting that they confirm the request. This email typically provides a link that the user can click to confirm the request. This step helps to prevent registration errors and to verify the legitimacy of the email address.

3. Upon confirmation, the submitted request is sent to the certificate authority (CA) configured in the PURSE system. The CA operator reviews the information provided by the user, checks the contact information and decides whether to approve or reject the request based on criteria of their choosing. If the request is rejected, an email is sent to the user notifying them of the decision. If the request is approved, then PURSE generates and stores long-term user credential in the MyProxy server. An email is then sent to the user notifying them that registration has completed successfully.

In a variant of this scenario, the user may instead supply an existing credential during the registration process. The same registration and approval process is followed, but following approval by the CA operator, the user is instructed to upload their existing certificate into the PURSE MyProxy.

Following successful registration, the user can use the username and password requested during registration to log in to the portal. The portal then retrieve a short-term credential for the user from the MyProxy service and uses that credential on behalf of the user to access VO resources as directed by VO-specific logic in the portal.

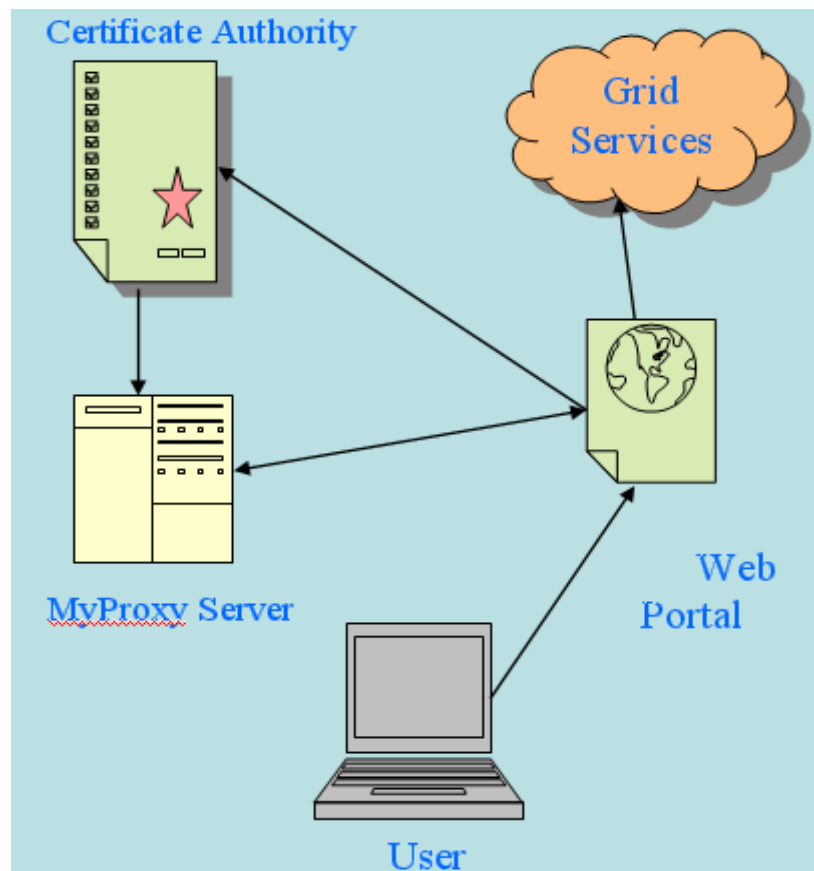The overall system architecture is presented on Figure 1.

Figure 1: Sample Registration Portal Architecture

2.2 Overview of Registration System APIs

PURSE is structured as a set of building blocks that can be used to create a fully functional web-based portal for accessing the Grid. The modules are available as "jar" files and can be plugged into any front-end interface such as an existing portal. We describe the high-level functionality and APIs for these building blocks in the following.

New user registration

Register user: This step initiates user registration by storing relevant user information, including requested username and user email address in the backend database. Once the information is stored, an email is sent to the user requesting confirmation of request.

Process user request: This step is triggered by the user's confirmation of the request to the registration system. An email is sent to a configured CA operator email address with instructions for the CA operator to access the user details.

Accept user: This module is invoked when a CA operator accepts a particular user's request. The following steps are performed.

- If the user wishes to use their own credentials (from an outside CA), the user is sent an email with a link that, when clicked by the user, downloads a simple java MyProxy client using Java Webstart that the user can use to upload their credential to the PURSE MyProxy server.

- If the user does not have their own credentials:

    - Either SimpleCA is used to generate a certificate for the user or a certificate request is sent to an external CA, depending on application requirements.

    - Either the configured SimpleCA certificate is used to sign the certificate or a signed certificate is received from the external CA.

    - The resulting long-term credentials are loaded onto a MyProxy server.

    - The database is updated to set the user's request status to "accepted."

- In both cases, an email is sent to the user indicating that registration is complete.

Reject user: If the CA chooses to reject the user, this module is invoked. It sends an email to the user and updates the user request status to "rejected."

Managing registered user

Revoke user: This module deletes the user from registration system. The user's credentials are removed from the MyProxy server and the user's status in the database is set to "revoked."

Renewal notice: This operation can be run as a periodic task to send mail to all users whose credentials are due to expire in some configured timeframe.

Renew user: This operation is triggered by a user attempting to renew membership and sets the user status in the database to "renew." If the renewal request is granted, an API to generate new long term credentials for the user and store them in the MyProxy server is provided.

Tools for registered users

Change password: Allows a registered user to change their password.

2.3 PURSE Setup

Establishing a PURSE-based portal involves two steps. In the first, we develop the portal code or alternatively integrate PURSE calls into an existing portal. In the second step, we set up the

backend database used to maintain user information (e.g., MySQL), a SimpleCA certificate authority (or alternatively configure PURSE to access an existing CA), and the MyProxy server used to store user credentials. Complete instructions for PURSE installation and testing are on the PURSE web side [5].

3 Deployment Use Cases

We describe two production deployments that have served both to drive PURSE requirements and to validate PURSE functionality.

3.1 Earth Systems Grid

PURSE was initially developed for the Earth System Grid (ESG) [8], a U.S. Department of Energy project to provide online access to climate data. We describe here the ESG production deployment as an example of how the registration system can be used. The following details are specific to deploying the Registration System for ESG.

The ESG portal needs to support two different classes of users: a small number of "privileged" users who can access all ESG data , including the newest data produced by the climate models, and all other users, who can access only publicly available, previously published data. Privileged users must be strongly authenticated, while for all other users, the requirement is to have some weak verification of their identity for the purpose of tracking ESG usage. At the same time, all users must have valid GSI credentials in order to access the data stored on the ESG various storage systems, which include NCAR MSS, NERSC HPSS, and GridFTP servers throughout the ESG grid.

This combination of authentication and authorization requirements spurred the development of PURSE. The user registration process plus email verification provides sufficient verification of user identity to satisfy requirements for tracking ESG usage. The ability to upload an existing credential supports the stronger authentication required for privileged users, who can obtain that credential from a CA with a stronger authentication policy. Users are then assigned to the appropriate user groups during registration, based on ESG policy. When a user would like to access data via the ESG Portal, their request is validated by the portal, which bases its decision on the user's group assignment. The number of user groups is configurable and depends on ESG policy. ESG uses the standard workflow for user registration, described in Section 2.1. ESG has 700 registered users as of May 2005.

Users who wish to see PURSE in action can register with the ESG portal by following the Registration link from the main ESG site (https://www.earthsystemgrid.org). At the registration web page, specify in the "Statement of Work" that you are interested in seeing PURSE in action. Access will be granted with limited access to ESG data.

3.2 Swegrid

Swegrid [7], a distributed computational resource in Sweden, uses the PURSE libraries to provides a registration system for its users. This system uses PURSE to meet Swegrid requirements for providing users with a certificate signed by an external (real) CA.

The main difference between the Swegrid and ESG registration system is the workflow for issuing certificates. In contrast to ESG, the Swegrid portal after registering the user in its local database sends a notification to the Swegrid registration authority which contains a link, which can be used by the RA to validate the user's information and to verify their identity against the papers signed and sent by that user. Upon approving the identity of the user, the RA sends the confirmation message to the Swegrid portal by replying to the notification email. The portal then accepts the user and generates a certificate request, which will be sent to the configured external and trusted certification authority. The CA may also use a similar link to access the local

information saved on portal database in order to verify the user's identity. Upon approval, CA signs the certificate and sends it back to the Swegrid portal. The portal receives the signed certificate from the CA and uploads this to the MyProxy server.

4   Sample Portal User Registration Interface

The PURSE distribution includes code for a simple Sample Registration Portal that may be adapted to meet specific application requirements. The Sample Registration Portal solicits basic data from the user, generates a certificate request to the VO operator, (following approval) generates a certificate and stores it in the MyProxy server, and gives the user an identifier and password for MyProxy access. A separate administrator interface allows a CA operator to accept or reject user requests and also to revoke issued certificates.

User registration involves the following steps.

1. The user fills in the Sample Registration Portal's entry page, shown in Figure 2, to submit their registration request.

2. The Sample Registration Portal verifies the user's email by sending the mail in Figure 3(a) to the provided email address.

3. Following user acknowledgement, the CA operator receives an email notification when a new account is being requested, as in Figure 3(b).

4. After receiving this notification, the CA operator logs in to a secure web site (Figure 4) and views the request.

5. After the user's credentials are generated and uploaded into MyProxy the user receives an email notification, as in Figure 3(c).

Figure 2: Screenshot of the PURSE sample user registration interface

(a) Email confirmation step: message sent to user

Date: Thu, 1 Jul 2004 14:25:47 -0600 (MDT)
From: esgport@ucar.edu
To: john_smart@ucar.edu
Subject: ESG Registration

The Earth System Grid (ESG) Portal received a request for a new user account that uses your email address. Click on the link below to confirm your request (NOTE: you will not be able to login until you receive an email from the portal administrator indicating your request has been approved):

Hhttp://www.earthsystemgrid.org/security/confirmRequest.do?token=000000fd-7c62-605c-ffffdea0-766ad9819840H

If you did not request this account, please inform us at esg-admin@earthsystemgrid.org.

Thank you,

ESG System Administrator

(b) Email sent to CA operator for approval

From: esgport@ucar.edu
Date: July 1, 2004 12:17:07 AM MDT
To: esg-ca@ucar.edu
Subject: ESG Registration

A request has been made for user account on the ESG Portal. You may access the details of the request by clicking on the following link.

Hhttp://www.earthsystemgrid.org/administration/accountRequestData.do?token=000000fd-2e0e-5d33-00006ac0-8387f64897beH

(c) Registration confirmation email sent to user

Date: Thu, 1 Jul 2004 14:34:52 -0600 (MDT)
From: esgport@ucar.edu
To:john_smart@ucar.edu
Subject: ESG Registration

Your request for an account with the ESG portal has been approved.

Figure 3: The three emails sent during user registration (based on ESG operational system)
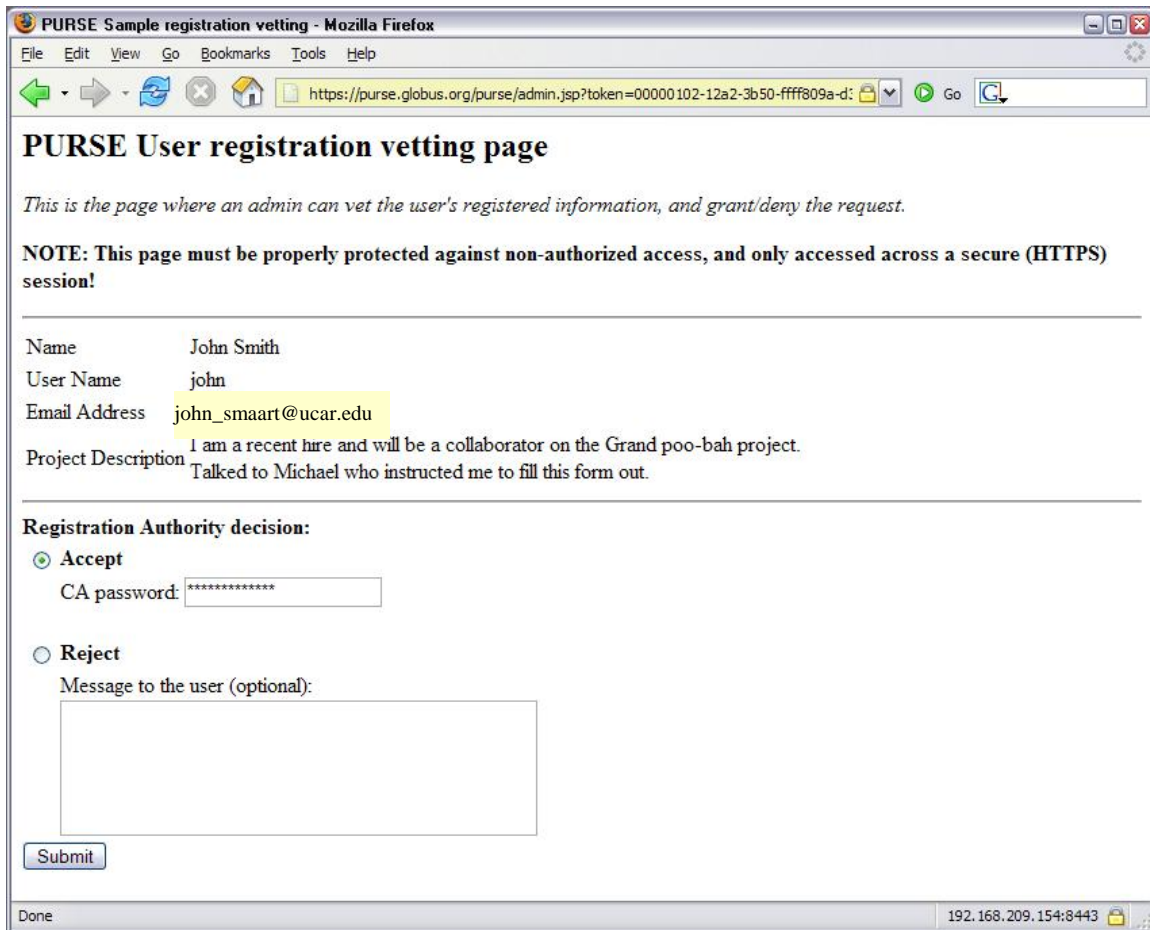
Figure 4: Screenshot of the PURSE sample administrative interface

5 Summary and Next Steps

PURSE provides a set of tools that can be used to construct Web-based user and administrative interfaces for user registration, credential management, and Grid access. PURSE automates the process of obtaining PKI credentials for users; provides for the secure storage of credentials; allows users to use existing Grid credentials, if available; and provides for Grid access via Web portals and secure username-password authentication.

In future releases, we plan to work towards simplifying PURSE installation by creating an easy packaging solution for this system. In addition, we need to adapt the current implementation to separate the credential repository from the rest of the portal logic, so as to permit hosting of the credential repository on a secure system.

Acknowledgements

References

1. Grid Account Management Architecture (GAMA), 2005. http://grid-devel.sdsc.edu/gamaT.
2. JavaMail, 2005. http://java.sun.com/products/javamail.
3. MyProxy, 2005. http://grid.ncsa.uiuc.edu/myproxy.
4. NSF Middleware Initiative (NMI) Grid Research Integration Development and Support (GRIDS) Center, 2005. www.grids-center.org.
5. Portal-based User Registration Service (PURSE), 2005. www.grids-center.org/solutions/purse.
6. SimpleCA, 2005. www.globus.org/security/simple-ca.html.
7. Swegrid, 2005. www.swegrid.se.
8. Bernholdt, D., Bharathi, S., Brown, D., Chanchio, K., Chen, M., Chervenak, A., Cinquini, L., Drach, B., Foster, I., Fox, P., Garcia, J., Kesselman, C., Markel, R., Middleton, D., Nefedova, V., Pouchard, L., Shoshani, A., Sim, A., Strand, G. and Williams, D. The Earth System Grid: Supporting the Next Generation of Climate Modeling Research. *Proceedings of the IEEE*, *93* (3). 485-495. 2005.
9. Bhatia, K., Lin, A., Link, B., Mueller, K. and Chandra, S. Geon/Telescience Security Infrastructure. San Diego Supercomputer Center, Technical Report TR-2004-5, 2004.
10. Foster, I., Kesselman, C., Tsudik, G. and Tuecke, S., A Security Architecture for Computational Grids. *5th ACM Conference on Computer and Communications Security*, 1998, 83-91.
11. Novotny, J., Tuecke, S. and Welch, V., An Online Credential Repository for the Grid: MyProxy. *10th IEEE International Symposium on High Performance Distributed Computing*, San Francisco, 2001, IEEE Computer Society Press.