J Grid Computing (2006) 4: 135–144
DOI 10.1007/s10723-006-9047-3

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

**Ian Foster · Veronika Nefedova · Mehran Ahsant ·
Rachana Ananthakrishnan · Lee Liming · Ravi Madduri ·
Olle Mulmo · Laura Pearlman · Frank Siebenlist**

Received: 8 September 2005 / Accepted: 24 April 2006

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 integration with portals, that can be used to address the full lifecycle of Grid credential management. We describe the PURSE design and its use in portals for two 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 the 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.

Key Words certificate · certificate authority · credentials · Grid · Grid security · portal security · user registration · public key

1. Introduction

A typical Grid application requires that a set of users share resources of various kinds in a controlled manner. To this end, many existing 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

The submitted manuscript has been created by the University of Chicago as Operator of Argonne National Laboratory ("Argonne") under Contract No.W-31-109-ENG-38 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

I. Foster · V. Nefedova (✉) · R. Ananthakrishnan ·
L. Liming · R. Madduri · F. Siebenlist
Argonne National Laboratory,
9700 S. Cass Ave.,
Argonne, IL 60439, USA
e-mail: nefedova@mcs.anl.gov

L. Pearlman
Information Sciences Institute,
University of Southern California,
4676 Admiralty Way,
Marina del Rey, CA 90292, USA

M. Ahsant · O. Mulmo
KTH, Teknikringen 14,
100 44 Stockholm, Sweden

and delegation. However, the tasks of creating and managing the PKI credentials [14] 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 users to register via a Web page, triggering a vetting process, upon successful completion of which a credential is created and managed on their behalf. Subsequent access is provided via a username and password, an authentication scheme that is familiar to most users The system also allows a user access to short term credential, if needed. It is important to note that while the authentication scheme looks to the user like a username/password scheme, it is in fact not. After the user authenticates to the portal with his username/password, the short-term PKI credential is generated for the user and the portal will use that credential to further authenticate the user to the remote Grid sites. 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.’ Different access control policies often are associated with different resources and operations. For example, write access to archival storage may require stringent verification of the identity or attributes of a requestor, while read access to Web pages may require only audit of a (by comparison) weakly authenticated identity. The definition and enforcement of such policies can be a significant source of complexity in Grid application deployments, because of 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.

The rest of this paper describes in turn the PURSE system (Section 2), two PURSE-based portals (Section 3), the sample registration portal distributed with PURSE (Section 4), and a security analysis of our system (Section 5). We conclude in Section 6 with a brief look at future work.

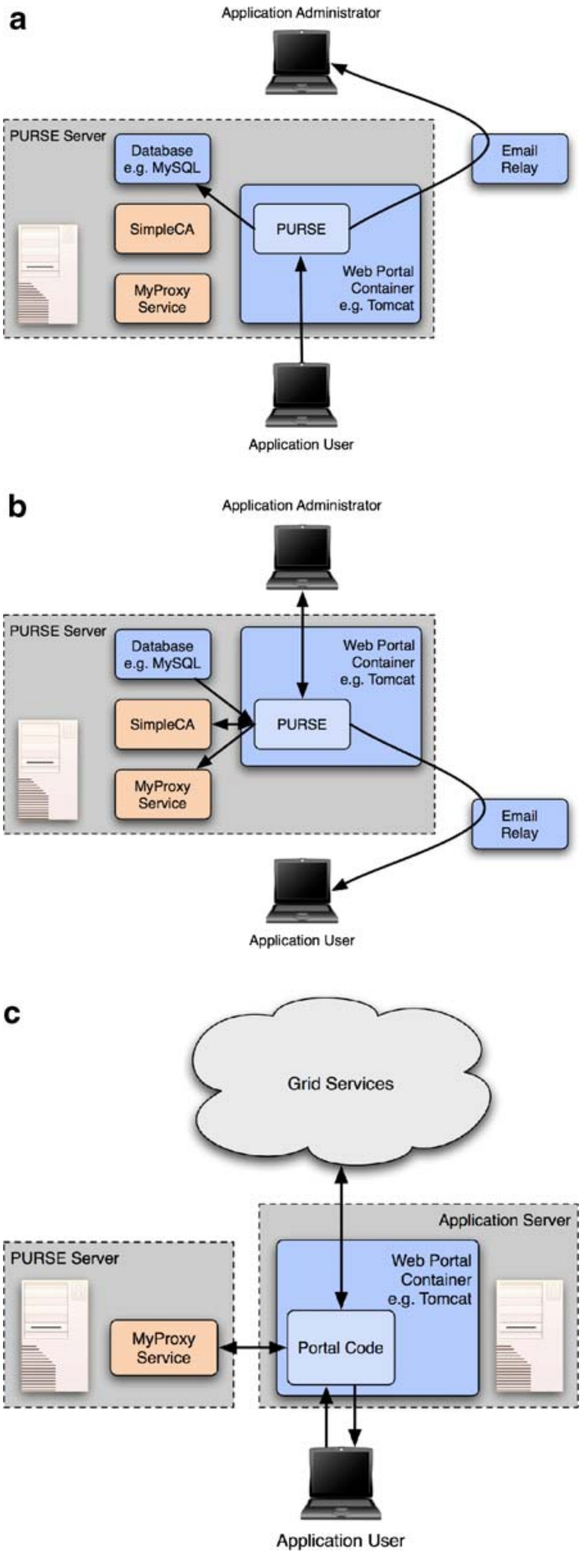
2. System Description

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 on some common tools, as follows:

- A JDBC-compliant database to persist user data. (MySQL is currently used.)
- A certification authority 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] to store user credentials

Figure 1 **a** User registration request. **b** User registration approval. **c** Web portal obtains a credential for a registered user.



- JavaMail [2] to send and receive notifications to the user and CA operator. In addition, an operator can perform certain administrative tasks remotely by sending cryptographically secured emails using S/MIME technology [12].

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.

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 stores the user information in a database and, using the contact information provided, sends the user an email message requesting confirmation of the request. The message provides a link that the user can click to confirm the request. This step (Figure 1a) helps to prevent registration errors and to verify the legitimacy of the email address.
3. Upon confirmation, the submitted request is sent via an email message to the registration authority (RA) configured in the PURSE system. The RA operator reviews the information provided by the user, and decides whether to approve or reject the request based on criteria of their choosing. If the request is rejected, an email is sent notifying the user of the decision. If the request is approved, PURSE generates long-term user credentials, signed by a certification authority using SimpleCA, and stores them in the MyProxy server (Figure 1b).
4. 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 RA, the user is instructed to upload his existing certificate into the PURSE MyProxy.

In a second variant of this scenario, an external (possibly offline) CA may be utilized instead of a local SimpleCA instance. To allow for a wide range of deployment constraints, this communication may either be performed out of band by the RA or in a more automated fashion by PURSE itself via email notifications, secured by S/MIME. While slightly more complex to deploy, it provides for a clean separation of the registration and user credential management from the operations of the CA, which is a highly tenable goal from a security point of view. It also provides for a backwards-compatible integration with an existing CA infrastructure.

Following successful registration, the user can use the username and password to log in to the portal. The portal then retrieves a short-term credential (a proxy certificate [15]) 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. Moreover, experienced users can interact with the MyProxy server directly to obtain short-lived credentials locally to their desktop, should they require that. This interaction is presented in Figure 1c.

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 Java archive ('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.

2.2.1. *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 RA email address with instructions for the RA to access the user details.

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

- If the user wishes to use his own credentials, the user is sent an email with a link to a simple Java MyProxy client that uses Webstart. The user can use the client to upload his credential to the PURSE MyProxy server.
- If the user does not have his own credentials:
  - The PURSE code base generates a user certificate request.
  - The request can be signed either by a local CA configured in the portal (via SimpleCA) or by some external CA, depending on application requirements.
  - 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 RA rejects the user, this module is invoked. It sends an email to the user and updates the user request status to 'rejected.'

2.2.2. *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.' This module does not currently create or publish a certificate revocation list (CRL), though that is clearly a desirable feature.

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 is provided to generate new long-term credentials for the user and store them in the MyProxy server.

2.2.3. *Tools for Registered Users*

Change password: This operation allows a registered user to change his password.

2.3. PURSE Setup

Establishing a PURSE-based portal involves two sets of steps. The first set involves developing the portal code (or integrate PURSE calls into an existing portal). The second set includes establishing the backend database used to maintain user information (e.g., MySQL), a SimpleCA certificate authority (or configuring 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 site [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 System 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 the data on ESG, including the newest data produced (by CCSM model [13]), and the rest of the users who can access only the publicly available, previously published data. All users must have valid GSI credentials in order to access the data stored on various storage systems (NCAR MSS, NERSC HPSS, GridFTP servers) throughout ESG. This combination of authentication and authorization requirements motivated the development of PURSE. All users are assigned to the specific user groups during the registration process based on ESG policy. When a user wants to access some data via ESG portal, the request is validated by the portal using the user's group assignment. The number of user groups is configurable and depends on ESG policy. ESG is using the standard workflow for user registration, described in Section 2.1.

ESG has more then 1,800 registered users as of January 2006. New users can register with the ESG portal by following the Registration link from the main ESG site (https://www.earthsystemgrid.org). To allow for PURSE trials, the ESG registration system provides limited access to ESG data to anyone if the 'Statement of Work' is filled to express interest in seeing PURSE in action. The functionality is available today and can be tried out by anyone.

3.2. Swegrid

Swegrid [7], a distributed computational resource in Sweden, uses the PURSE libraries to provide a registration system for its users. This system uses PURSE to meet the Swegrid requirements for providing users with a certificate signed by an external certification authority. While a SweGrid-local CA is currently used, the goal is to switch to an external CA that is a member of, and operates in accordance with policies defined by, the European Grid PMA (http://www.eugridpma.org). That way, the user's credentials will be honored in all of Europe (and beyond), should the user want to access non-Swegrid resources.

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 containing a link that 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 the RA's approval of the identity of the user, the portal then accepts the user and generates a certificate request that is sent to the configured external CA, using a cryptographically signed email. 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. A confirmation email then is sent to the user.

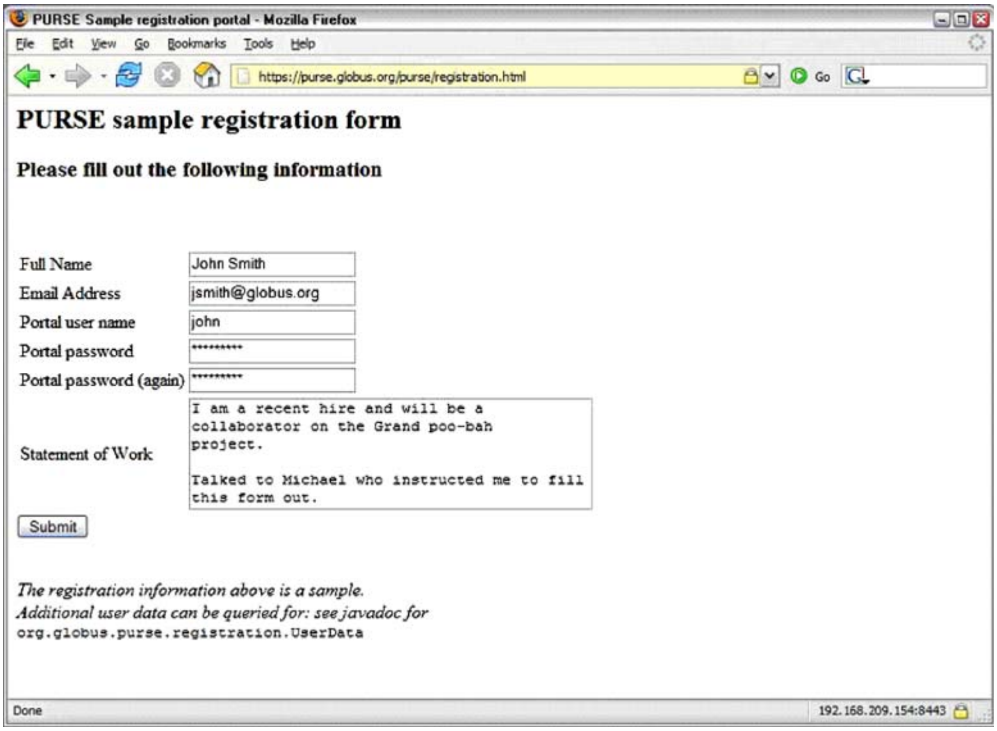
4. Sample Portal User Registration Interface

The PURSE distribution includes code for a sample registration portal that may be adapted to meet specific application requirements. Figure 1 shows the architecture of this system.

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 his registration request. The password will be used later to log into the portal application.
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 acknowledgment, 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).


PURSE Sample registration portal - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
https://purse.globus.org/purse/registration.html
Go

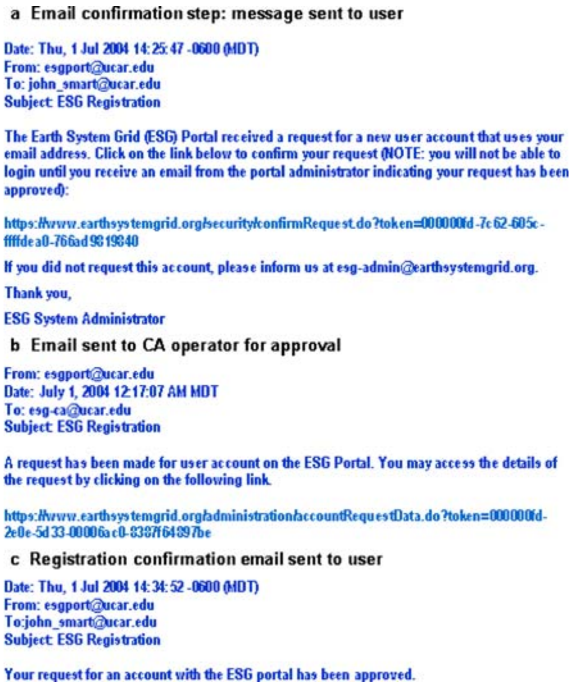
PURSE sample registration form

Please fill out the following information

Full Name	John Smith
Email Address	jsmith@globus.org
Portal user name	john
Portal password	**********
Portal password (again)	**********
Statement of Work	I am a recent hire and will be a collaborator on the Grand poo-bah project. Talked to Michael who instructed me to fill this form out.

The registration information above is a sample.
Additional user data can be queried for: see javadoc for
`org.globus.purse.registration.UserData`

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):

https://www.earthsystemgrid.org/security/confirmRequest.do?token=000000fd-7c62-605c-ffffdea0-766ad9819840

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.

https://www.earthsystemgrid.org/administration/accountRequestData.do?token=000000fd-2e0e-5d33-00006ac0-8387f64897be

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).

5. Related Work

While a number of efforts solve the various issues PURSE addresses, we view PURSE as an *integration* of solutions to provide an end-to-end system for user registration and credential management. Here we discuss some of the point solutions available:

5.1. CACL

Tools like CACL and MyProxy Online CA address the need for a simple and automatic way of issuing certificates to users. CACL provides a server daemon that can be contacted to get certificates. The PURSE system accomplishes that by providing a web interface to a SimpleCA backend.

5.2. MyProxy CA

The MyProxy service has recently been enhanced with the addition of an integrated online CA, which allows users to obtain short-lived End Entity Certificates. As PURSE is already fully integrated with MyProxy, we are currently investigating how and in what form we can make this additional functionality available to PURSE deployments in the future.

5.3. GAMA

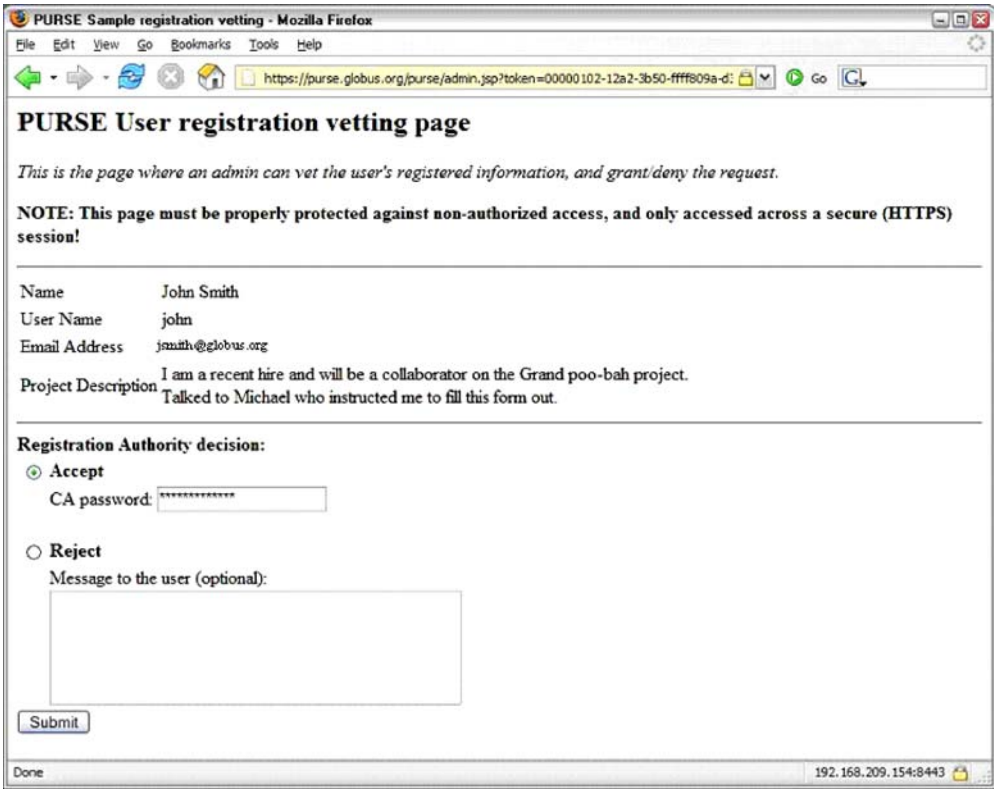
The Grid Account Management Architecture (GAMA) activity [1, 9] has produced a similar system in parallel with our PURSE development, using roughly the same architecture. GAMA differs from PURSE primarily in implementation details. For example, GAMA is implemented as a server and a set of portlets that communicate with that server, while PURSE is implemented as a Java library that can be called from servlets, portlets, JSP pages, etc. GAMA uses CACL for generating and storing credentials while PURSE uses SimpleCA and MyProxy. GAMA does not support uploading existing credentials nor does it support an email driven notification system to trigger the various steps in user registration and management. We view this parallel evolution as a demonstration of the importance of this technology.

5.4. GridShib

GridShib is an NSF-funded project at Argonne National Laboratory, NCSA and the University of Chicago [16]. Its goal is to leverage the Shibboleth [17] attribute services for the webservices clients and Grid application services. The GridShib and PURSE technologies are very complementary, and many GridShib deployment scenarios involve MyProxy services to bootstrap the initial authentication where PURSE could potentially be used for the initial user registration and credentials creation.

6. Security Considerations

PURSE provides a complete life-cycle solution for end-user credential management. It is targeted


PURSE Sample registration vetting - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
https://purse.globus.org/purse/admin.jsp?token=00000102-12a2-3b50-ffff809a-d:
Go
Submit
Done
192.168.209.154:8443

PURSE User registration vetting page

This is the page where an admin can vet the user's registered information, and grant/deny the request.

NOTE: This page must be properly protected against non-authorized access, and only accessed across a secure (HTTPS) session!

Name	John Smith
User Name	john
Email Address	jsmith@globus.org
Project Description	I am a recent hire and will be a collaborator on the Grand poo-bah project. Talked to Michael who instructed me to fill this form out.

Registration Authority decision:

● **Accept**

CA password: **************

○ **Reject**

Message to the user (optional):

Figure 4 Screenshot of the PURSE sample administrative interface.

primarily at solving several operational problems often associated with PKI:

- Automate as much of the registration process as possible, making it more useful. Access to the registration pages are secured using HTTPS.
- Ease the burden on the users and eliminate the security risks associated with users managing their own credentials. PURSE provides a controlled and auditable environment where the long-term credentials are stored, secured by well-known and trusted technologies.
- Secure remote access to the credentials for the mobile computer user. This has been one of the principal problems with Grid portals in the past.
- Control credential renewal and removal.

The long-term credentials are stored on a MyProxy server running on the same machine as the portal, which in turn must be considered the ‘weakest link’ from a security perspective. In order to mitigate this risk, the portal can be isolated and given a limited and controlled access to the SimpleCA and MyProxy installations by using privilege separation techniques. To illustrate this, we have successfully experimented with hardening the Swegrid installation by running the registration portal and the MyProxy software in different local user accounts. We then use the sudo software, configured to allow the user account of the portal the additional privilege of executing a few well-defined administrative commands in the MyProxy user account, for upload, renewal, and removal of long-term credentials. With this approach, the long-term credentials cannot easily be stolen or copied from the system in the case that the portal account (which is the most exposed) gets compromised. In addition, we have strengthened the security of PURSE such that a local privilege escalation attack (the attacker has to obtain root access) is required in order to obtain physical access to the (still password-protected) long-term credentials.

In analogy with the MyProxy and portal isolation approach above, it should be a deployment option to completely separate the CA installation from the other components. As mentioned, the Swegrid installation will be configured to use an external CA in the near future, and such a configuration will be fully supported by the PURSE credential processing commands.

Another attack vector is to target the MyProxy server or the operating system directly (the SimpleCA and database components do not expose any external interfaces that an attacker can utilize). The MyProxy technology has existed for five years and is widely used and trusted in the Grid community. MyProxy has a well-known threat model with well-established principles on operational policies and management. We recommend that PURSE be installed in line with these guidelines: For example, PURSE should run on dedicated hardware with limited administrative access (console login only).

We emphasize that the PURSE system does not remember or store any passwords. The user supplies a password when he/she registers with the portal application, and this password is used to generate a key pair and encrypt the private key. The password is not stored. When the user logs into the portal, he/she supplies the password again so that the private key can be decrypted. Thus, in order to obtain access to any long-term credential (including the SimpleCA signing key), the password has to be provided (or can be guessed/cracked through some time-consuming brute-force method). This situation holds true even in the case of stolen hardware.

When PURSE generates new credentials for users at registration time, the credentials are 'signed' by the portal application's SimpleCA service. The portal administrator established SimpleCA when PURSE was installed. Typically, the signature identifies the portal application, e.g., 'Earth System Grid Certificate Authority.' In order for these credentials to be recognized and trusted by Grid services operated by third parties, the operators of those services must configure their code to trust the portal's certificate authority. If this is not acceptable, users may obtain credentials from a trusted CA using other mechanisms and supply those existing credentials to PURSE at registration time.

7. 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 toward 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 and provide support for multiple RA authorities for varied vetting strength requirements. The Open Grid Collaboration Environment (OGCE) project has produced a set of JSR-168 compliant portlets to simplify the process of incorporating PURSE into a web portal.

Communities currently using PURSE include the Earth System Grid, SweGrid, and the Large Synoptic Survey Telescope (LSST) community.

Acknowledgments PURSE was first developed at Argonne National Laboratory in collaboration with the Earth System Grid, with support from the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-Eng-38. Staff at NCSA and the University of Chicago supported by NSF Middleware Initiative's GRIDS Center [4] contributed to the extensive pre-release testing. KTH made a detailed security analysis of the original software, and provided the sample portal interface and added several features to the API, as part of a project funded by the Swedish Research Council.

References

1. Grid Account Management Architecture (GAMA), 2005. http://Grid-devel.sdsc.edu/gama
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., Williams, D.: The earth system Grid: Supporting the next generation of climate modeling research. Proc. I.E.E.E. **93**(3), 485–495 (2005)
9. Bhatia, K., Lin, A., Link, B., Mueller, K., Chandra, S.: Geon/Telescience Security Infrastructure. Technical Report TR-2004-5, San Diego Supercomputer Center, 2004
10. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A security architecture for computational Grids. In: 5th ACM Conf. on Computer and Communications Security, pp. 83–91, 1998
11. Novotny, J., Tuecke, S., Welch, V.: An online credential repository for the Grid: MyProxy. 10th IEEE International Symposium on High Performance Distributed Computing, San Francisco, IEEE Computer Society, 2001
12. Ramsdell, B. (ed.): S/MIME Version 3.1 Message Specification. RFC 3851, July 2004
13. CCSM – Community Climate System Model. http://www.ccsm.ucar.edu/
14. Housley, R., Polk, W., Ford, W., Solo, D.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, April 2002
15. Tuecke, S., Welch, V., Engert, D., Pearlman, L., Thompson, M.: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC 3820, June 2004
16. Welch, V., Barton, T., Keahey, K., Siebenlist, F.: Attributes, anonymity, and access: Shibboleth and globus integration to facilitate Grid collaboration. 4th Annual PKI R&D Workshop, April 2005
17. Shibboleth Project Web Page http://shibboleth.internet2.edu/