A Portal-based User Registration Service for Grids (PURSE)

Draft: July 28, 2004 version

Abstract

This document describes a software module that can be used to build a registration system for grid users for use with a web-based portal. It provides functionality to store user information, generate credentials for the user and store them, renew credentials and allow their use by the portal to access standard Grid resources. The Earth Systems Grid and Swegrid have developed portal interfaces that use PURSE. The Earth Systems Grid portal is described.

Table of Contents

1 Introduction and Motivation

Resources on Grids often need to be accessed with varying levels of security. Often a virtual organization (VO) will have a subset of its resources, typically data on web pages, which are public, their access perhaps only subject to audit of a weakly authenticated identity, while other resources will be strictly accessed controlled based on the strongly verified identity and/or attributes of a requestor.

The Grid Security Infrastructure (GSI) provides a mechanism for secure single sign on and access control to resources. GSI is based on public key infrastructure (PKI), with credentials issued by a Certificate Authority. GSI has proven itself to be a viable mechanism for resources that need strong access control, however it can be overly burdensome for accessing resources that only require weak authentication of a requestor.

The portal-based user registration system described in this document, PURSE, can be used to construct a system that integrates registration and credential management, by generating credential for the user and storing the long-term credential at the time of registration associated with a web portal. Accesses to portal are then authenticated with a typical username and password, which grant the user access to Grid resources via the portal using their credentials held by the portal.

PURSE provides the following functionality:

- It automates the process of obtaining Grid credentials for the user, shielding them from a process that is cumbersome and often error-prone.
- Secure storage of credentials, again often a cumbersome and error-prone process, is migrated from the user to the registration service.
- It allows for users to use a common web browser as their method to access Grid resources, which is ideal for occasional users of those resources who do not have the motivation to learn a new set of tools.
- The user's access to the portal is authentication with a common and familiar security mechanism, that is, username and password.

2 Description of the User Registration System

The PURSE user registration system is a collection of JAVA applets 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, it 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 the renewal of credentials and revocation of credentials. This functionality can be accessed through a well-defined API and is easily configurable.

The system is built upon some common tools:

- JDBC compliant database : used to persist user data
- SimpleCA (http://www.globus.org/security/simple-ca.html) : used to generate and sign user credentials
- MyProxy server (http://grid.ncsa.uiuc.edu/myproxy/): used to store user credentials
- JavaMail : used to send and receive notifications to user and CA

2.1 Typical Usage

This section describes the typical usage of a web portal using PURSE.

The user must first be registered into the PURSE system. This is a one-time event that must precede any other use of the system. The registration process has the following 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 persists the user information and using the provided contact information, sends an email back to the requesting user to confirm of the request. This typically would be a link the user can click to confirm the request. This step is to help prevent errors in registration and verify the legitimacy of the email address.
3. Upon receiving the 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 the request or reject it 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 CA approves the request, the following happens:
        - PURSE generates credentials for the user and places the long-term credential in the MyProxy server.
        - An email is sent to the user notifying them that their registration process has been completed successfully.

The user is now registered and log onto the portal using the username and password requested during the registration process. The portal would then retrieve a short-term credential for the user from the MyProxy service and use it on behalf of the user to accessing VO Grid resources as directed by VO-specific logic in the portal.

2.2 Overview of Registration System APIs

The registration system is 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 jars and can be plugged into any front-end interface such as existing portal. This section describes the high-level functionality and APIs for these building blocks:

New user registration:

- Register user: This step initiates the 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 and an email is sent to a configured CA email address with instructions for the CA to access the user details.
- Accept user: This module is invoked when a CA accepts a particular user's request and the following is done:
    - SimpleCA is used to generate a certificate for the user.
    - The configured CA certificate is used to sign the certificate.
    - The user's long-term credentials are loaded onto a MyProxy server.
    - The database is updated to set the user's request status to "accepted".
    - An email is sent to the user indicating that the registration has been completed.
- Reject user: If the CA chooses to reject the user, this module is invoked. It sends am 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. It involves removing the user's credentials from the MyProxy server and setting the user's status in the database to "revoked".
- Renewal notice: This can be run as a periodic task and sends mail to all users whose credentials are due to expire in some configured timeframe.
- Renew user: This 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 for a registered user to change the password that is used to access the MyProxy server.

2.3 Downloading PURSE

A packaged version is still under development.

A snapshot of the current code can be obtained via anonymous CVS:

% setenv CVSROOT :pserver:anonymous@cvs.globus.org:/home/globdev/CVS/esg

% cvs checkout esg

If you get errors about a missing .cvspass file or missing password, run "cvs login" before running the cvs checkout command.

2.4 Deploying PURSE

The following is a high-level overview of how to set up the Registration System

1. Setup MyProxy server for online storage of user's Grid credentials. It is required that the MyProxy sever be installed on the same machine as the registration service since some commands use direct file manipulation. The installation instructions could be found here: http://grid.ncsa.uiuc.edu/myproxy/
2. Setup a backend database (can be any JDBC complaint database) to store user information. MySQL has been used to test the software.
3. Setup and configure the Certificate Authority. This can be accomplished using the SimpleCA package (http://www.globus.org/security/simple-ca.html), but the system as such does not require that the CA certificates be generated using this tool. The registration system may be configured to use any CA.
4. Deploy the PURSE system. Full documentation is still under development. The file "notes" in the distribution contains draft directions.

3 A Deployment Use Case: Earth Systems Grid

The registration system was initially developed for the Earth Systems Grid (ESG) project. In this section we describe their deployment as an example of how the registration system can be used, providing screen shots of their deployed system.

The following details are specific to deploying the Registration System for ESG

1. All components are installed on dataportal.ucar.edu at NCAR

   This is how the ESG has configured its Web portal for RS use. The portal consists of several software components running on a multi-processor Sun system running Solaris 9. The first of these is the Tomcat application server, which provides the execution environment for the portal software. The portal itself has been constructed using a combination of custom and pre-built Java-based software components. The Apache Struts framework is incorporated to provide a Model

View Controller (MVC) infrastructure upon which individual portal applications are built. The Model layer of the MVC framework consists of Java classes which implement the business logic necessary for the various portal applications. These classes also interface with, call, or execute externally developed software components developed by the collaboration which enable the Grid infrastructure, such as MyProxy, SimpleCA, and the various Databases in which user data and metadata are stored. The Controller layer is implemented using the Apache Struts default ActionServlet class, and an XML configuration file (struts-config.xml) which determines how requests are threaded through the system. The View layer consists of JSP files and the Tiles framework which construct the HTML presentation sent to clients using data produced by the Model layer. MyProxy repository is also owned by the portal user (the MyProxy repository can only be accessed by the repository owner), and is located in the default location /var/myproxy and uses the host certificate DN for authentication. The portal user has its local credentials mapped to the host certificate files. The MySQL database has a schema which is distributed with the registration component. The table names and fields are configurable in the event a pre-existing database is to be used through use of a properties file, which is also included in the distribution. All registration components are configurable by editing entries in the portal's web.xml file, including MySQL database connection information, email addresses, hosts and protocols for notification of new registrations and sending errors, locations of template files for email to users and administrators, URLs for creating links in email notifications which execute actions through the portal, the location of binaries and directories for creating and signing certificates, and the locations of the MyProxy server and repository.

2. New users could register with the ESG portal by following the Registration link from the main ESG link: https://www.earthsystemgrid.org

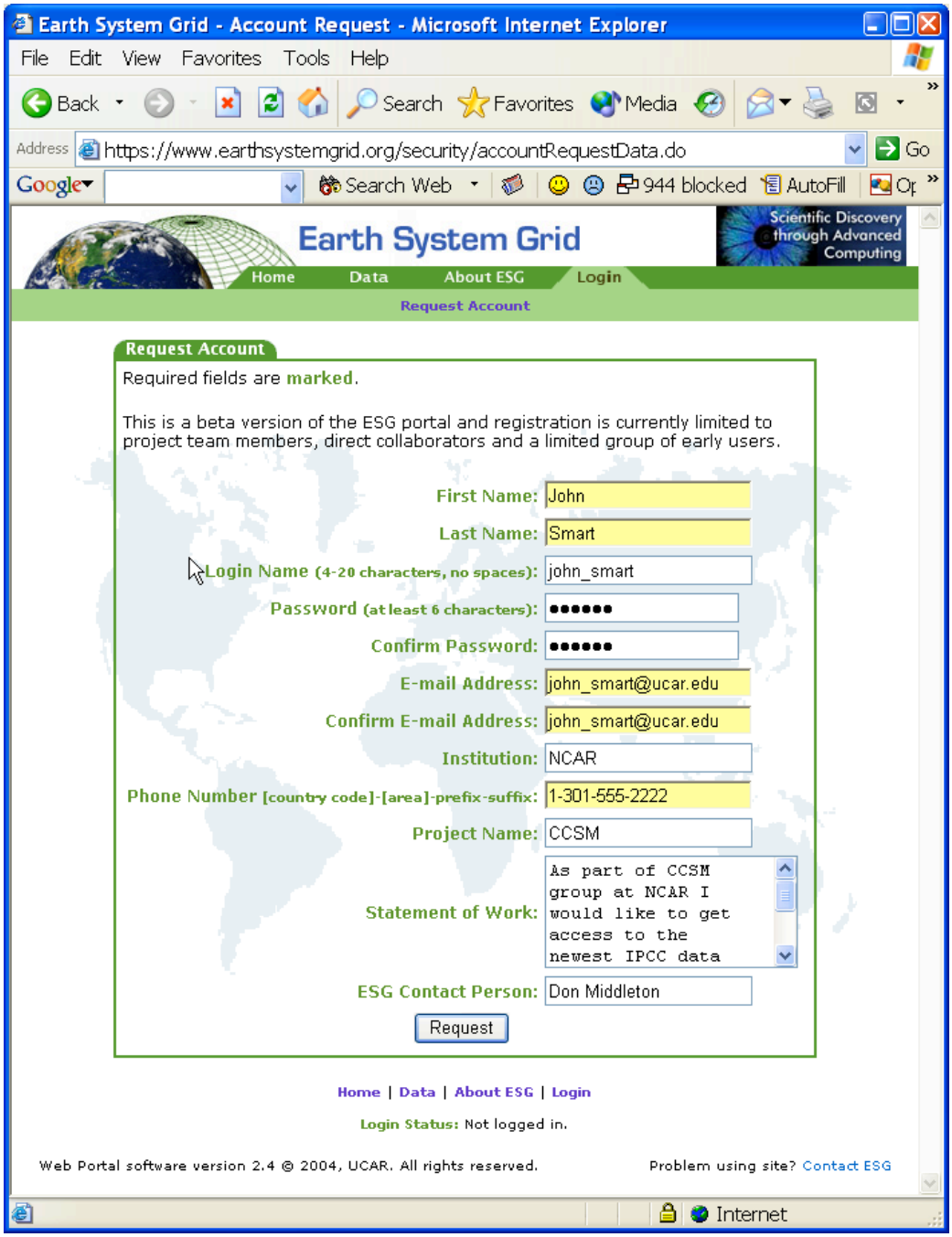
3.1 ESG User Registration Interface

ESG has developed a Registration System that allows users to register easily with the ESG portal. The focus of this system is ease of use. The portal extensions (CGI scripts) have been developed to automate user registration requests. The system solicits basic data from the user, generates a certificate request for the ESG CA, if approved, generates a certificate and stores it in the MyProxy server, and gives the user an ID/password for MyProxy access. The approval/rejection of the request is based on ESG policy: everybody with the valid e-mail address and relevant/verifiable project description would be approved. The Registration System also has an Administrator's interface that allows CA administrator to accept/reject user's request as well as to revoke the already issued certificates. Another major part of new user registration process is actually making sure that user's credentials are known everywhere on ESG Grid. It has been accomplished by establishing the ESG accounts everywhere on ESG grid and mapping all the users with ESG CA-issued certificates to these accounts. With this approach when the new user is being registered with ESG he automatically is being granted the access to ESG resources. The most important benefits of this system are that users never have to deal with certificates and that the portal can get a user certificate from the MyProxy server when

needed on behalf of the user. Figure 1 shows the Registration System architecture.

This is the 'entry page' for submitting the initial request.

Figure 2: Screenshot of the ESG User Registration Interface



3.2 E-Mail Confirmation Step

Before approving User's request, the system verifies the user's email by sending the following mail to the provided email address.

Figure 3: Screenshot of the E-mail Confirmation Step

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):

http://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

3.3 Registration Confirmation E-Mail

After the user's credentials are generated and uploaded into MyProxy the user receives the following notification by e-mail:

Figure 4: Screenshot of the Registration Confirmation Step

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.

3.4 Administrative Interface

The ESG Certificate Authority receives an email notification when a new account is being requested:

Figure 4: Screenshot of email sent to the CA for the 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.

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

The CA logs in via secure web site and views the request

Figure 5: Screenshot of user request in Administrative Interface

3.5 Post conditions

As a result of this deployment, ESG now has the following:

- An easy-to-use web user interface for new Grid users.
- An easy-to-use web administrative interface for managing user credentials.

4 Future Work

- Packaging of the PURSE system and generalization of ESG-specific functionality (underway at KTH).
- Incorporating the Community Authorization Service (CAS) with Registration System
- Adding possibility for users to use their GSI credentials obtained from a different CA (DOE SG CA) in ESG portal.

5 Acknowledgements

PURSE was developed by Argonne National Laboratory in collaboration with the Earth Systems Grid. NCSA contributed to the writing of this document.