A Credential Renewal Service for Long-Running Jobs

Daniel Kouřil
CESNET z.s.p.o.,
Zikova 4, 160 00 Praha 6,
Czech Republic
Email: kouril@ics.muni.cz

Jim Basney
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
USA
Email: jbasney@ncsa.uiuc.edu

***Abstract*— Jobs on the Grid require security credentials throughout their run for accessing secure Grid resources, such as GridFTP data repositories. However, delegating long-lived credentials to long-running jobs brings an increased risk that a credential will be compromised and misused. Additionally, it is often difficult to predict the run-time of jobs on the Grid, due to changes in application performance and resource load, making it difficult to set the lifetime of the delegated credential in advance. We have developed a solution to this problem for the EU DataGrid project using the MyProxy online credential repository and have further evolved it during the EGEE project. Users store their long-lived credentials in a dedicated MyProxy server and delegate short-lived credentials to their jobs. When a job's credential nears expiration, the Workload Management System retrieves a new short-lived credential from the MyProxy server on the user's behalf and uses it to refresh the job's credential. The MyProxy server's policy specifies which services may obtain credentials on the user's behalf, and all operations are logged at the MyProxy server, where access to credentials may be restricted if a compromise is detected or suspected. This system has been used for credential renewal in Grids in Europe for over three years. In this paper, we present the system design, describe our experiences, and discuss the security implications of this approach.**

I. Introduction

From 2001 to 2003, the European DataGrid (EDG) project [1] developed a Grid computing infrastructure for data intensive applications in the areas of high energy physics, biology and medical image processing, and earth observations. In high energy physics, EDG developed solutions for managing the large amount of data to be produced by CERN's Large Hadron Collider starting in 2007. The EDG testbed included over 1,000 computers sharing more than 15 Terabytes of data at 25 sites across Europe, Russia, and Taiwan, serving over 500 scientists. Many of the products of the EU DataGrid project are now being carried over to the two year Enabling Grids for E-sciencE (EGEE) project [2] in Europe to build a production Grid facility.

One of the challenges addressed in the EDG project was credential management for long-running jobs. It is good practice to limit credential lifetimes, to minimize the vulnerability of stolen credentials. However, jobs require valid credentials for the duration of their run, so they can access authenticated services, such as metadata catalogs and GridFTP servers [3]. Additionally, the Workload Management System requires credentials for submitting jobs to compute elements and resubmitting jobs when failures occur. Providing long-lived credentials to long-running jobs is a simple solution, but given the difficulty of predicting job run-times in practice, this leads to over-estimation of required credential lifetime and increased vulnerability. Thus, EDG identified the need for a credential renewal service to enable long-running jobs to use short-lived credentials.

This paper describes the credential renewal service, which has been in use now for over three years. We first give an overview of Grid security in Section II followed by an overview of EGEE job management services in Section III. We then describe the design of the renewal service in Section IV, discuss related work in Section V and security implications in Section VI, and document experiences with the renewal service in Section VII. We end the paper with a discussion of future work and conclusions.

II. Grid Security Overview

The X.509 Public Key Infrastructure (PKI) is the basis for Grid security, chosen for its support of cross-domain, decentralized trust establishment. The EDG PKI, now the EGEE PKI, includes multiple Certification Authorities (CAs), typically one per country, for issuing certificates, typically valid for one year, to Grid users in that country. The user's private key, associated with the certificate, must be well-protected to prevent unauthorized use of the certificate.

A. Proxy Certificates

To minimize exposure of their long-lived private key, users can create session credentials using X.509 proxy certificates [4]. At the start of a session, the user runs a program to create a new private key and proxy certificate, signed by the long-lived key. The user can then authenticate with the proxy certificate and key during the session, without requiring further use of the long-lived private key.

The need to limit the lifetime of proxy certificates to reduce exposure to potential compromise is an important principal of the Grid Security Infrastructure [5], [6]. This technique of limited-lifetime session credentials is also used by Kerberos [7] to limit the usefulness of stolen tickets. Organization

policies typically limit proxy certificate lifetime to one day or less [8].

Grid security protocols support delegating credentials to jobs and services running on the user's behalf using proxy certificates [9]. The job or service creates a new private key and proxy certificate, and the user signs the proxy certificate with his or her current proxy key. The job or service can then use the new private key and certificate to authenticate as the user, because of the user's valid signature on the proxy certificate. Delegating a proxy credential over the network requires the delegatee to send the proxy certificate to the delegator for signature, but no private keys are exchanged.

A Grid credential is a private key with multiple certificates, which form a chain of signatures back to the signature of a trusted Certification Authority. The credential is valid only if all of the certificates are valid, so the lifetime of the credential is the intersection of the lifetimes of each certificate. Thus, a job on the Grid will typically have a credential with lifetime of one day or less.

B. MyProxy Online Credential Repository

MyProxy [10], [11] allows users to store their credentials in an online repository for later retrieval. A MyProxy server is typically run by Grid administrators on a well-secured machine, to provide maximum protection for the stored credentials. Management of proxy credentials in the repository is entirely under the user's control. Users may store, update, and remove their credentials in the repository at will, after first authenticating to prove ownership of the credentials. Users also specify the access control policies that protect the credentials. MyProxy supports a variety of authentication methods, including password, certificate, and Kerberos. Authenticated clients retrieve proxy credentials from MyProxy using proxy delegation, so it is not necessary to transfer any keys over the network.

In developing the credential renewal service, EDG extended the functionality of the MyProxy software to support the certificate-based authentication described later in Section IV. The EDG extensions to MyProxy maintained backward compatibility in the MyProxy protocol and have been included in the baseline MyProxy software distribution since 2001.

C. Authorization Assertions

It is also common to include authorization assertions, signed by trusted servers, in the proxy certificate, to grant additional rights to users. For example, the Virtual Organization Management Service (VOMS) [12], used in the EGEE project, issues signed attributes indicating group membership roles or capabilities, to enable fine-grained authorization for Grid resources, removing the burden of managing accounts for every user across all resources in the Grid. Like proxy certificates, VOMS attributes have limited lifetimes and so must be managed by the credential renewal service.

III. EGEE JOB MANAGEMENT

Efficient management of resources in a dynamic Grid environment is a crucial factor for satisfying utilization of the Grid. The EGEE project uses a Workload Management System (WMS) [13], which is based on a solution developed within EDG [14] that was further enhanced with new features.

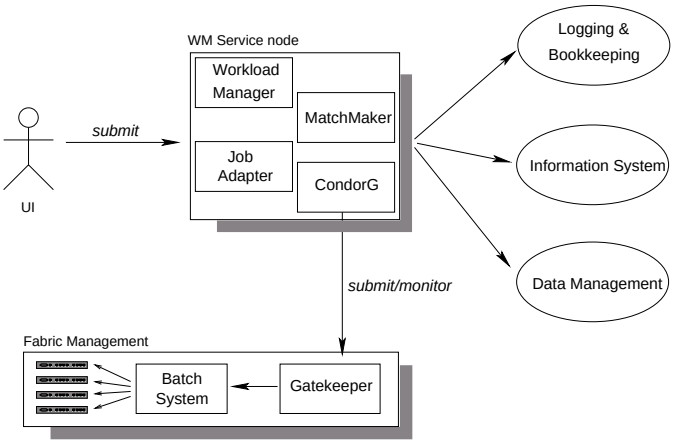

Fig. 1. WMS Architecture

The overall architecture of the Workload Management System and how it fits into the rest of Grid environment is given in Figure 1. The WMS system is composed of multiple components that are responsible for job management and also communicate with other services on the Grid.

The User Interface (UI) provides users with an interface to access the WMS, to submit a new job or possibly cancel running jobs and also to retrieve the result data produced by the job after the job finishes. In order to submit a job to the WMS the user has to specify the description of the job, along with its needs. The EGEE architecture uses the Job Description Language (JDL) [15], which provides a very general means of expressing the requirements. The JDL allows users to specify a broad range of parameters, both characterizing the job (executable name, parameters, etc.) and specifying requested resource parameters (CPUs, network, storage, etc.). The user sends the JDL to the WMS as part of the job submission protocol.

Upon receiving the JDL, the WMS processes it and tries to find the computing resource that best fits the requirements specified. Choosing the most appropriate resource is the goal of the matchmaking process. In order to be able to make its decision, the WMS must interact with external components (the Information System and Data Management services) that provide the current status of the resources and the Grid environment. The WMS can use different scheduling strategies, which can be added easily to the WMS in the form of dynamic plug-ins.

Once the resource is chosen, the WMS prepares the job for submission and then submits it. In order to submit and monitor jobs on Compute Elements (CEs), the WMS uses the Condor-G [16] system. Condor-G contacts the remote resource using the Globus GRAM protocol [17] and sends the specification of the job to the Globus gatekeeper. After authenticating and authorizing the request, the gatekeeper ensures via the standard Globus mechanism that the job is

put into a queue of the local batch system.

When the job starts computing, a special wrapper added by the WMS retrieves an input sandbox from the WMS that contains input data for the job. The input sandbox is defined by the user in the job's JDL and specifies files to be copied from the UI to the end computing resource, such as the job's binary or input data. Upon completion of the job the output sandbox with results is saved on the WMS. The contents of the output sandbox is also defined by the JDL and contains the output produced by the job, such as the standard output and standard error output streams. Sandboxes are transfered using the GridFTP protocol [3].

When the user finds out that her jobs have finished, she uses the UI to retrieve the output sandbox from the WMS to her local machine to analyze the result data.

The WMS has access to a lot of information and manages jobs of many users and often processes sensitive data. Therefore the WMS was designed and implemented with special care to possible security issues.

All network connections to the WMS are authenticated using TLS [18], and users must always prove possession of a valid credential during each job submission. During the job submission the user's proxy is also delegated to the WMS. The WMS will subsequently use this proxy when acting on behalf of the job owner, e.g., when submitting the job to the compute resource

Even though each submitted job must have a valid proxy certificate delegated to the WMS, the overall job lifetime may easily exceed the lifetime of the credential. Proxy certificates are usually valid for a few hours, which may not be sufficient for a job to complete. Even if this lifetime is sufficient for the job's computation, the overall job lifetime may be much longer and unpredictable. For example if the resource where the job is being processed stops working, the job must be rescheduled and resubmitted by the WMS, possibly restarting the computation from the beginning. The jobs can also spend some time in queues waiting for required resources. It is not possible to estimate these times in advance.

A simple solution for providing jobs with valid certificates would be submissions with long-lifetime proxy certificates. However, this would certainly lead to overestimation of the lifetime of certificates, which would break the meaning of short-lifetime proxy certificates. The Proxy renewal service is designed to address problems concerning support for long-running jobs. The initial version of the service was designed and implemented in the EDG project, and further modification and development has been done in EGEE.

IV. A Proxy Renewal Service

The proxy renewal service utilizes the functionality offered by MyProxy and extends its possibilities to support long-lived jobs. From a logical point of view, the service can be seen as a module of the WMS, which registers and manages the proxy certificates of all submitted jobs that request proxy renewal. All jobs maintained by the service are kept valid by periodically retrieving newer proxies from the MyProxy repository and replacing proxies near expiration with the new ones. Besides the renewal of the proxy certificate itself, the service also supports renewal of VOMS certificates, which are embedded in the proxy as X.509 extensions and also have a limited lifetime. Details on how the VOMS certificates are handled are given in Subsection IV-C. Renewed certificates are detected by Condor-G [16], which ensures that the proxy is transfered to the CE, if needed. The architecture of the renewal service is illustrated in Figure 2.

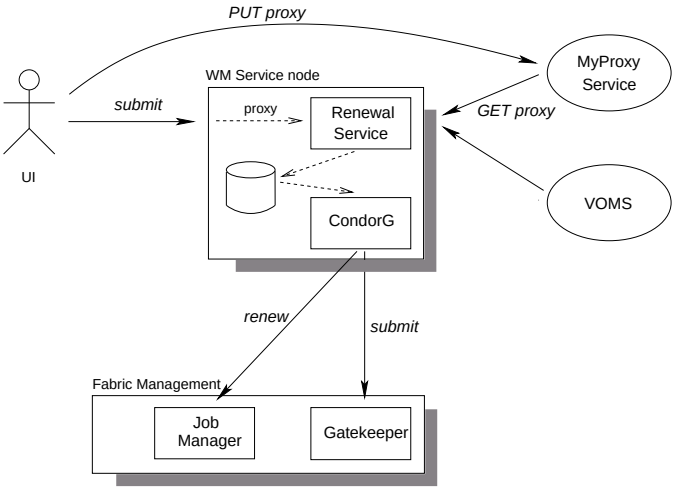

Fig. 2. Renewal Architecture

The proxy renewal service is a separate application executing on the same host as the WMS using the same account. It communicates with its clients via a local Unix socket using a simple text protocol. The renewal service does not listen on any network interface so it is not exposed directly to any kind of remote attacks. Access to the socket (and hence to the renewal service) is only allowed from the local system and is secured by the local operating system. However, being part of the whole WMS system, the renewal service (and proxies maintained by the service) can be compromised if an attacker gains control over the WMS. Thus, the WMS should always be installed on a well protected and monitored machine.

A. Registering a Proxy with the Renewal Service

During the submission of a job to the WMS, the job owner's proxy is delegated to the WMS as part of the submission protocol. The proxy is stored in a file on a local filesystem and is used by the WMS to act on the user's behalf, especially to submit the job to the chosen computing resource. When the users expect the job submitted will last longer than the proxy lifetime, they can specify a special option (called MyProxy) in the job description (JDL) containing a hostname of the MyProxy server containing long-lived credentials for the user. If the WMS encounters this option in the JDL, it contacts the renewal service to register the job proxy certificate for the renewal mechanism.

In the registration request the WMS includes the filename of the file containing the proxy to register. Upon receiving the

request, the renewal service reads the proxy to verify that it contains a valid credential. It then creates a new file in its repository directory and copies the contents of the file to the new location. At the same time the renewal service updates its local database to add information about the new registration. In particular, it computes the time when the proxy should be renewed and stores this information in the database. Both the database and the registered proxy certificates are stored on disk so the service can recover easily from sudden reboots or crashes of the machine. To finish the registration, the service returns to the caller (WMS) the filename in the repository containing the registered proxy. This filename is used by the WMS instead of the original one, and the renewal service ensures that it always contain a valid credential. Since the WMS does not modify the proxy certificates, and the renewal service ensures that all updates to proxy files are atomic, the proxy file does not get corrupted and can be used by multiple processes or threads. The proxy registered with the service is identified by the jobid, a unique identifier which is generated for each job submitted to the WMS.

In order to ease management of proxies owned by a single user and minimize network communications during renewal, the proxy renewal service avoids managing duplicate proxies when possible. If a registration request arrives and the proxy renewal service already has an equivalent proxy for the same user, the jobid is added to the list and no new file is created. However, if the new proxy is not equivalent to any stored proxy, (for example, it may contain different VOMS attributes), then the new proxy will also be stored for the user.

B. Renewing a Proxy Credential

The renewal service checks its database and list of registered proxies and if a proxy is nearing expiration ($1/4$ of lifetime remaining), it attempts to contact the MyProxy server asking for a new credential. The connection to the MyProxy server is secured using TLS with mutual authentication, and the renewal service uses its service certificate to authenticate. During the communication the service also has to prove it is in possession of the appropriate proxy certificate that is still valid before retrieving a new proxy. If the renewal service successfully retrieves a new proxy, it updates the registered proxy file with the new proxy. Otherwise, it computes a new time to attempt renewal again, using the method of bisecting intervals of remaining lifetime, with a minimum interval of five minutes. If all the renewal attempts fail and the proxy expires, it is removed from the repository and Condor-G stops the job.

When a proxy is renewed and the file is updated, in some cases the proxy must be distributed further to update proxy certificates currently in use. If the job has not yet been submitted to a computing resource, there is no need to distribute any file, since the newer proxy will be used automatically by the WMS when the renewal service updates the file. However, if the job was already submitted to a computing resource, the proxy must be transported to that resource to allow the job to continue to perform authenticated actions.

The renewal service does not contact the resources itself but utilizes the ability of the Condor-G service, used for job control by the WMS. Condor-G components running on the WMS maintain a list of submitted jobs and their proxy files. Whenever any of the files changes and the corresponding proxy is renewed, Condor-G contacts the corresponding job running on the resource and delegates a renewed credential there. Condor-G uses the GRAM protocol [17], which supports a special command for credential renewal. Using this command it is possible to delegate a new proxy to the GRAM jobmanager that manages the running job on the resource. EDG and Condor developers designed and implemented initial support for this command, and the modification was accepted to the standard Globus toolkit.

After a job finishes, the WMS sends to the renewal service an unregistration request containing the jobid of the finished job. If the jobid is the last one in the list for a proxy file, the renewal service removes the file from the repository.

Certificates may be revoked any time during their lifetime by the issuing CA, for example, if the credentials have been compromised. The renewal service does not itself perform revocation checks on registered certificates, but when they are used to authenticate to the MyProxy server or the CE, those services will verify that the user's certificate has not been revoked, to ensure that invalid credentials cannot be misused.

C. Renewing VOMS Attributes

The authorization framework of EGEE is based on VOMS [12], which issues attribute certificates to users that also have lifetime limited to a few hours. Since many EGEE services use VOMS attributes for making access control decisions, the job must also posses a valid VOMS certificate in addition to the proxy certificate. Therefore, it is not sufficient to renew only the proxy certificate, but VOMS attributes must be renewed as well.

We contemplated two possible approaches to enable renewal of the VOMS attributes. The first possibility is to store the VOMS attributes in the proxy certificate on the MyProxy server. This possibility would ensure that valid VOMS attributes are part of each credential retrieved from the MyProxy server. Though simple, this approach has several limitations. The lifetime of VOMS attributes should not be set to very long values, since there is no means to revoke issued VOMS certificates. The credentials stored in the MyProxy server would have to either be too short or contain different proxy certificate and VOMS lifetimes. Both these scenarios would cause problems for long-running jobs. Also, the proxy renewal service uses the user's certificate name (X.509 subject name) as the identifier of the proxy requested from the repository, so the users could only use a single set of VOMS attributes, which also would not suit the users.

In order to support VOMS credentials we chose to adapt the proxy renewal service so it is able to contact the VOMS servers. Whenever the proxy renewal service renews a proxy that contains VOMS attributes, it parses the VOMS data to find out which VOMS server issued it. Then it contacts the

VOMS server asking for the same set of VOMS attributes as was present in the proxy to be renewed. As the result of this process a new proxy certificate is created that contains the same set of VOMS attributes.

V. Related Work

The problem of credential expiration for long-running jobs motivated the development of renewable tickets in Kerberos version 5 [7]. Users can request tickets from the Key Distribution Center (KDC) with a renewable lifetime greater than the ticket lifetime. Before the ticket lifetime is reached, the ticket holder can contact the KDC to renew the ticket. This process can be repeated until the renewable lifetime is reached. Policies at the KDC set the maximum ticket lifetime and who can request a renewable ticket. The Kerberized version of the Grid Engine scheduler supports automatic renewal of Kerberos tickets for long-running jobs.

Unlike proxy renewal, Kerberos renewable tickets require users to specify the maximum renewable lifetime of the ticket in advance, which can be difficult to predict. Also, though it is possible in the Kerberos protocol for a user to tell the KDC to stop renewing a ticket, this functionality is not provided in standard Kerberos implementations.

Since the development of the EDG renewal service, Condor-G [16] has also added support for credential renewal using MyProxy. Users can include MyProxy information in the job description file when submitting jobs to Condor-G, and Condor-G will contact MyProxy to retrieve new credentials for jobs before they expire. Condor-G will then delegate the new credentials to the jobs via GRAM proxy refresh, as described in Section IV.

VI. Security Implications

MyProxy server policies control access to credentials in the repository. To renew a credential, the WMS must have a credential with distinguished name matching the renewal policy in the MyProxy server. The renewal policy is the combination of a server-wide policy set by the MyProxy administrator and a per-credential policy set by the credential owner. Both policies, written as regular expressions, must be met for the MyProxy server to grant credential access.

Additionally, the WMS must have a valid proxy credential to be renewed. This provides an additional level of protection, so the WMS can only obtain new credentials if the user has already submitted a job to it with a delegated credential.

Thus, the MyProxy server requires the WMS to authenticate twice: first with its service credential and then with the credential it wants to renew. The first authentication using the service credential is performed during the initial TLS handshake in the MyProxy protocol. Then, the MyProxy server sends an “Authorization” message to the client, containing a random challenge that the client must sign with the proxy key to prove possession of the proxy.

The MyProxy server’s credential repository, holding a large number of keys, is an attractive target for attack and must be well-protected, similar to a Kerberos KDC. It should be noted, however, that a professionally administered MyProxy server can provide a more secure storage location for user credentials than typical desktop systems, and all operations on the MyProxy server are logged to the system log for monitoring to help detect possible misuse.

VII. Experiences

The EDG infrastructure has enabled a wide range of computational science activities in Europe. The WMS, including the proxy renewal service, was adapted and deployed in many development and production environments. The widest deployment is the LCG Grid that currently comprises about 140 sites and 13000 CPUs. During various data challenges performed during the first half of 2004, it was shown that the WMS can process a vast number of jobs [14]. The execution time of these jobs often exceed the default proxy lifetime, requiring use of the proxy renewal service.

After starting the EGEE project, the WMS was also used as the main cornerstone for job management. The current EGEE environment where the WMS is deployed contains 28 instances of the WMS and serves 12 virtual organizations. Current data received from the Quality Assurance JRA2 group [19] of EGEE which also monitors the usage of the resources shows that since January to the beginning of May, 1,090,849 successful jobs have been computed in this environment. 104,218 of these jobs took more than 12 hours (which is the default proxy lifetime), requiring the use of the proxy renewal service.

When we initially deployed the proxy renewal service, we experienced problems with the sudden reboot of WMS machines, since for performance and security reasons the service kept all data in memory. Subsequently we redesigned and reimplemented the service to store all data on disk for persistence, and we have not experienced performance problems. We also divided the services into two separate processes. One of the processes (the master) serves the client requests for registration and unregistration of proxies, while the second one (the slave) performs the renewal process. This separation allows us to accumulate all calls to the Globus and MyProxy libraries in the slave. The slave is completely stateless and can be restarted at any time; it is automatically restarted after finishing 1,000 renewals. Such a change allows us to deal with multiple memory leaks that were present in the external libraries.

VIII. Future Work

The area of job management is not the only one dealing with limited lifetimes of proxy certificates. Similar issues were also encountered by the Data Management activity group in EGEE, which is responsible for providing and supporting the data management related middleware of EGEE. One of the corner stones of the data management infrastructure is Transfer Scheduling [20]. This service allows clients to specify requested data operations and submits them to a specialized service which is responsible for scheduling the requests and executing the file transfers. Since the data to transfer are

usually very large and also due to the dynamic character of the Grid environment, it is difficult to predict the duration of the file transfers, so this environment has similar credential management issues as the area of job management.

To address these issues we are working with the data management group on augmenting the proxy renewal service to also support long-lived file transfers.

IX. Conclusion

The proxy renewal mechanism ensures that jobs submitted to EGEE have valid certificates throughout their lifetime without violating the meaning of short-time proxy certificates. All the proxy renewal process is performed in a controlled way, with only a small set of trusted services allowed to renew proxies. All transactions concerning proxy renewal are logged and can be analyzed for possible misuse. Users may choose which trusted proxy repository server to use for proxy renewal. Since management of proxy certificates in the repository is completely under the user's control, they may stop renewal at any time by removing their proxy from the repository.

The proxy renewal capability has proven useful in European Grids over the past few years. While credential expiration has been long recognized as a problem for long-running jobs on the Grid, the EDG proxy renewal service is the first example of a solution that has seen widespread use.

Acknowledgment

EGEE is a project funded by the European Union under contract INFSO-RI-508833. We also acknowledge the national funding agencies participating in EGEE for their support of this work.

The MyProxy project is supported by the U.S. National Science Foundation Middleware Initiative under contract ANI–02–22571.

References

[1] "Home page of the EU DataGrid project." [Online]. Available: http://www.edg.org/

[2] "Home page of the EGEE project." [Online]. Available: http://www.eu-egee.org/

[3] "GridFTP Protocol Specification," Global Grid Forum GFD.20, March 2003.

[4] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson, "Internet X.509 Public Key Infrastructure (PKI) proxy certificate profile," IETF RFC 3820, June 2004.

[5] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A Security Architecture for Computational Grids," in *Proceedings of the 5th ACM Conference on Computer and Communications Security Conference*, 1998, pp. 83–92.

[6] M. Humphrey and M. Thompson, "Security Implications of Typical Grid Computing Usage Scenarios," in *Proceedings of the 10th International Symposium on High Performance Distributed Computing (HPDC)*, August 2001.

[7] B. C. Neuman and T. Ts'o, "Kerberos: An Authentication Service for Computer Networks," *IEEE Communications*, vol. 32, no. 9, pp. 33–38, September 1994.

[8] S. Mullen, M. Crawford, M. Lorch, and D. Skow, "Site Requirements for Grid Authentication, Authorization and Accounting," Global Grid Forum GFD.32, 2004. [Online]. Available: http://www.ggf.org/documents/GFD.32.txt

[9] V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, J. Gawor, S. Meder, and F. Siebenlist, "X.509 proxy certificates for dynamic delegation," in *Proceedings of the 3rd Annual PKI R&D Workshop*, April 2004.

[10] J. Novotny, S. Tuecke, and V. Welch, "An Online Credential Repository for the Grid: MyProxy," in *Proceedings of the Tenth IEEE Symposium on High Performance Distributed Computing (HPDC10)*, August 2001.

[11] J. Basney, M. Humphrey, and V. Welch, "The MyProxy Online Credential Repository," *Software: Practice and Experience*, 2005.

[12] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Frohner, A. Gianoli, K. Lőrentey, and F. Spataro, "VOMS, an Authorization System for Virtual Organizations," in *Grid Computing: First European Across Grids Conference*, 2004.

[13] P. Andreetto, S. Borgia, A. Dorigo, A. Gianelle, M. Mordacchini, M. Sgaravatto, L. Zangrando, S. Andreozzi, V. Ciaschini, C. D. Giusto, F. Giacomini, V. Medici, E. Ronchieri, V. Venturi, G. Avellino, S. Beco, A. Maraschini, F. Pacini, A. Guarise, G. Patania, D. Kouřil, A. Křenek, L. Matyska, M. Mulač, J. Pospíšil, M. Ruda, Z. Salvet, J. Sitera, J. Škrabal, M. Voců, V. Martelli, M. Mezzadri, F. Prelz, D. R. S. Monforte, and M. Pappalardo, "Pratical approaches to Grid workload and resource management in the EGEE project," Computing in High Energy and Nuclear Physics (CHEP04), 2004.

[14] G. Avellino, S. Beco, B. Cantalupo, A. Maraschini, F. Pacini, M. Sottilaro, A. Terracina, D. Colling, F. Giacomini, E. Ronchieri, A. Gianelle, M. Mazzucato, R. Peluso, M. Sgaravatto, A. Guarise, R. Piro, A. Werbrouck, D. Kouřil, A. Křenek, L. Matyska, M. Mulač, J. Pospíšil, M. Ruda, Z. Salvet, J. Sitera, J.Škrabal, M. Voců, M. Mezzadri, F. Prelz, S. Monforte, and M. Pappalardo, "The DataGrid Workload Management System: Challenges and Results," *Journal of Grid Computing*, 2004.

[15] F. Pacini, "JDL Attributes," DataGrid-01-TEN-0142, 2003, http://www.infn.it/workload-grid/documents.html.

[16] J. Frey, T. Tannenbaum, I. Foster, M. Livny, and S. Tuecke, "Condor-G: A Computation Management Agent for Multi-Institutional Grids," in *Proceedings of the Tenth IEEE Symposium on High Performance Distributed Computing (HPDC10)*, August 2001.

[17] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke, "A Resource Management Architecture for Metacomputing Systems," in *Proceedings of the IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing*, 1998, pp. 62–82.

[18] T. Dierks and C. Allen, "The TLS Protocol Version 1.0," IETF RFC 2246 (Standards Track), January 1999.

[19] "Home page of the EGEE QA group." [Online]. Available: http://egee-jra2.web.cern.ch/EGEE-JRA2/index.html

[20] "EGEE gLite User Guide — Overview of gLite Data Management," EGEE-TECH-570643-v1.0, March 2005. [Online]. Available: https://edms.cern.ch/document/570643