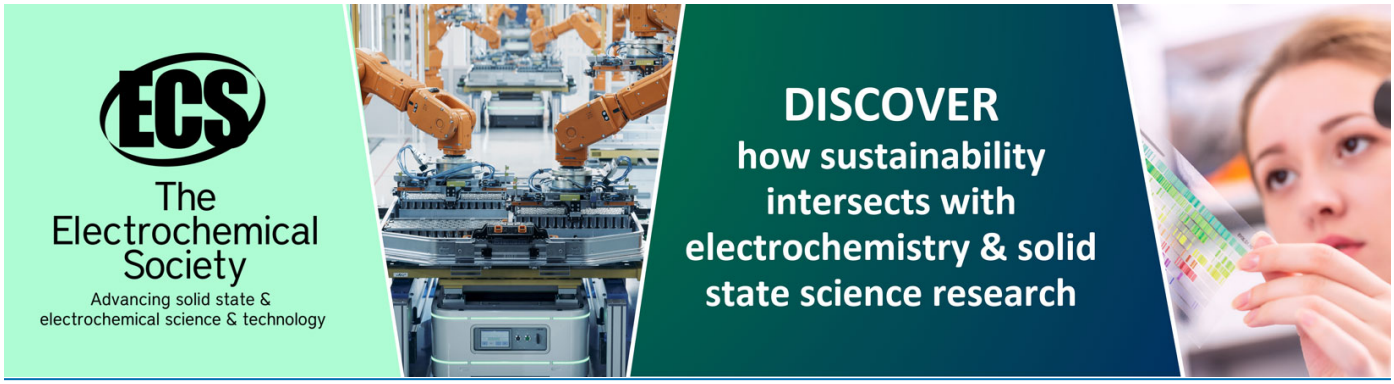
An Authentication Gateway for Integrated Grid and Cloud Access



Journal of Physics: Conference Series **331** (2011) 062021 doi:10.1088/1742-6596/331/6/062021

An Authentication Gateway for Integrated Grid and Cloud Access

V Ciaschini, D Salomoni

INFN CNAF, Viale Berti Pichat 6/2, Bologna, 40127, Italy

E-mail: vincenzo.ciaschini@cnaf.infn.it, davide.salomoni@cnaf.infn.it

Abstract. The WNoDeS architecture, providing distributed, integrated access to both Cloud and Grid resources through virtualization technologies, makes use of an Authentication Gateway to support diverse authentication mechanisms. Three main use cases are foreseen, covering access via X.509 digital certificates, federated services like Shibboleth or Kerberos, and credit-based access.
In this paper, we describe the structure of the WNoDeS authentication gateway.

1. Introduction

High-Energy Physics experiments and other collaborations have come to be highly reliant on the availability of abundant computation and storage resources, organized in Grids and, more recently, in Clouds. While these resources are indeed necessary for peak computation periods, they could be shared among other experiments or for other uses when the needs for them are less urgent.

Such a sharing, however, meets with a very large access barrier. All Grid resources, and several cloud resources as well, still require the users to obtain and maintain a personal X.509 digital certificate [1]. This task has proven to be quite complex and time consuming if done correctly, and requires setting up a complex structure of Registration Authorities (RA) for the Certification Authorities (CA) in all organizations who desire access to the resources. Furthermore, these RAs are not even unique, since if an organization has multiple sites, then a separate RA has to be created in each site.

This paper suggests a setup that would ease this task for both users and administrators by leveraging multiple existing technologies that will be configured to work together, with the objective of an easier, and possibly invisible to the user, certificate management.

This setup is the basis of the WNoDeS (Worker Nodes on Demand Service) authentication layer [2, 3], of which it forms the Authentication gateway to seamlessly allow access to virtual resources via Grid or Cloud interfaces to users authenticated through multiple methods.

This paper is organized as follows. After a review of the state of the art in authentication management in Section 2, the core of the proposal is described in Section 3. Finally, Section 4 presents a description of the state of the actual implementation, followed by concluding remarks and future directions.

2. State of the Art

As has been mentioned in Section 1, the traditional way of obtaining X.509 certificates for Grid access to resources requires setting up a network of RAs on all sites belonging to the organizations wishing to utilize those resources.

To avoid or lessen this burden, several solutions already exist in production, and indeed we will describe the main ones here.

2.1. KCA

Kerberos [4] is a common authentication mechanism, widely deployed in many organizations. It is well trusted and well understood, and indeed the administrative procedures necessary to maintain coherency between the list of Kerberos users and authorization members are already in place. Since this part is already there, a reasonable argument could be that there is no reason to duplicate another time this infrastructure to setup an RA, when all the necessary information is already registered into Kerberos. This naturally leads to the idea of using a Kerberos ticket as the source of trust for the user identity when issuing a user certificate.

Indeed, this idea is well known as the Kerberized CA, or KCA [5] for short.

The main idea of the Kerberized CA is that when the user has already authenticated with Kerberos and obtained his ticket, we already know who the user is, and there is therefore no reason to re-authenticate the user again with a different method. The KCA works by using a specialized protocol called 'kx509' to establish a communication, mutually authenticated by Kerberos, between the user and a specialized service.

This protocol will then pass a public key to the KCA service, which will wrap it into a certificate and send it back. All of this communication happens in the clear, but since a certificate is supposed to be public anyway there is no much reason to encrypt it.

The client, upon receipt of the certificate, stores it into a credential service, from which it can be retrieved via a specialized command.

From the user's point of view, this boils down to the following three commands:

kinit authenticate with Kerberos and obtain a ticket. This should be done anyway.

kx509 connect with the online CA and store a certificate in the credential storage.

kxlist -p extract the certificate from the credential storage and save it into the file system.

As can be seen, this whole procedure can be easily scripted and is secure, because the tools mentioned above already secure communication with a CA, at least as long as the system hosting the tools does not get compromised. Indeed a KCA is already among the CAs distributed by the International Grid Trust Federation (IGTF) [6], the main organization which evaluates Certification Authorities for usage in a Grid environment.

2.2. SLCS

Section 2.1 described Kerberos. But there is another very common authentication mechanism, namely, Shibboleth [7].

Shibboleth is a widely used protocol that can be used to authenticate users with web resources, and federate together several of them, with each organization or site responsible only for authenticating its own users via their own Identity Providers (IdP). All the resources trust successful authentication from the IdPs belonging to the federation.

This structure is remarkably similar to a RA infrastructure, and one could actually say identical if "resource" is substituted with "CA." Given this, organizations are again justifiably reluctant to essentially throw it away and duplicate all the infrastructure and information to setup RAs, when they could instead reuse their existing information. Indeed, such an effort does not make sense and ways to reuse it have been devised.

The following will describe one such mechanism, the Short Lived Credential Service (SLCS) [8] but, *mutatis mutandis*, it could be applied to any other similar methods.

SLCS once again relies on having a separate service, belonging to the federation, which signs certificates for its members.

A user who wants to obtain one such certificate does so by issuing a special command, `slcs-init`. This command starts an agent that contacts the online CA, and is by it redirected towards the user's IdP to authenticate. Once the authentication step is successfully completed, the online CA issues a DN, and *only* to the agent.

It is then the agent's job to create a private/public key pair and to create a Certificate Signing Request (CSR) binding the public key to the DN. Such a CSR is then sent to the service again, where it is signed and therefore transformed into a certificate, and sent back to the user.

From the user's point of view, this boils down to issuing the following command:

```
slcs-init
```

and then completing the authentication procedure. Again, this process can be easily scripted and is secure, because the tool mentioned above already secures communication with a CA, at least as long as the system hosting the tool does not get compromised. Indeed a SLCS is already among the CAs distributed by IGTF.

2.3. Summary of existing implementations

As can be seen from Sections 2.1 and 2.2, tools and procedures for transforming other kinds of credentials into X.509 certificates are already commonplace and widely used. What these techniques lack, in our opinion, is the proper scaffolding to make them easily accessible from a portal, thus forcing users to always drop down "to the metal" for the initial setup of all their sessions.

The rest of this article will propose a deployment setup meant to ameliorate, and possibly eliminate, this issue.

3. An authentication gateway

The National Computing Center (INFN Tier-1) of the Italian National Institute for Nuclear Physics (INFN) provides computing and storage facilities to the International High-Energy Physics community and to several multi-disciplinary experiments. The INFN Tier-1, hosted at CNAF, Bologna, is running a software called WNoDeS [2] (Worker Nodes on Demand Service), developed by INFN to virtualize computing resources and to make them available through local, Grid or Cloud interfaces.

Since WNoDeS internally uses X.509 certificates, one possibility could be to request that users obtain their own X.509 certificates in order to access resources using one of the interfaces made available by WNoDeS (local, Grid, or Cloud). But not all users have, want, or are able to easily get an X.509 certificate, which also assumes the presence of an explicit RA infrastructure.

Here we describe a deployment setup which leverages both KCA and SLCS to provide users with X.509 certificates on-the-fly. The architecture, which will be applied to WNoDeS once development is mature, is pictorially described in Figure 1.

The local access case, where a user has direct access to a command line interface, is simple. Nothing forbids the installation of clients for both KCA and SCLS services on the same UI, at which point it all becomes just a matter of calling the right tool. We, however, want all of this to be available from a web interface.

One important thing that should be noted is that, since both these services generate their own certificates, and more specifically their private keys, on the local machine on which the client part is running, direct transmission of the certificate to the user is forbidden by all the rules regarding the management of private keys. Therefore, only two setups are possible: either the underlying

machine qualifies as a UI, or the credentials, once generated, are immediately registered on a `myproxy` [9] server. Note that both these choices require a password, since a private key on a UI must be encrypted, while `myproxy` requires a password upon registration. Such a password cannot be saved by the infrastructure, but must be kept in memory if at all. This approach is explicitly allowed by the CAs that are part of the IGTF. Indeed, the requirements that a site must obey to be permitted to receive the actual credentials are described in [13]. Our proposed usage is covered by section 2.2, case 4.

A possible setup would therefore be the following:

A simple web interface would ask the user if he means to authenticate with his own certificate, if he possesses one, or with his own Kerberos or Shibboleth credentials. In any of the latter choices, the user would be required to identify its own source of trust, *i.e.* its own IdP or Kerberos Ticket Granting Service (TGS) among those supported by the federation and subsequently input the corresponding username and password, all in a form protected by HTTPS.

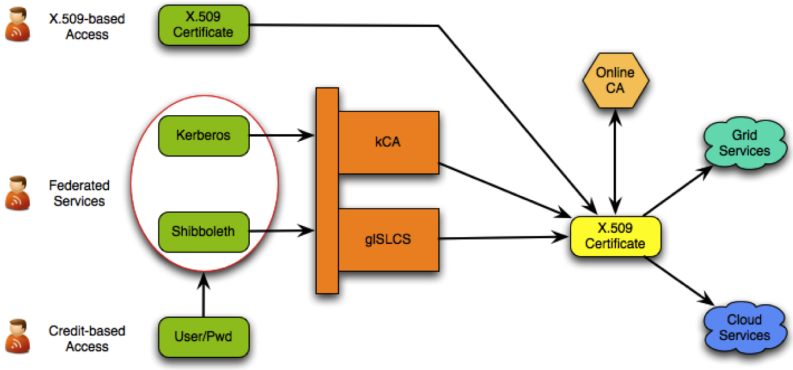


Figure 1. The WNoDeS authentication gateway architecture.

This information would then be used to pass the right parameters to the underlying tool, and obtain a certificate which will be registered with a `myproxy` server and immediately deleted afterwards. Also, the information necessary to successfully authenticate must not be kept in memory, and immediately overwritten past use. This is fully equivalent to use, for example, a `slcs-init` command on a UI, where the authentication information are entered directly, and therefore should not constitute a vulnerability, provided that they are deleted when no longer used.

After the certificate is properly registered in a `myproxy` server, there is a plethora of already well-known techniques to use these certificates for job submission or resource instantiation, so from this point of view the problem could be classified as solved.

All of the above assumes that *two* different online CAs are active at any time, namely, the KCA and the SLCS. Both these CAs need to be certified by the IGTF, so that short-lived certificates can be trusted across multiple organizations.

There is also the case, called *credit-based access* in Figure 1, where users are authenticated via a local username/password pair. In practice, in this case an IdP (Kerberos or Shibboleth) needs to be setup; we therefore fall into one of the previous scenarios.

The use cases that will then be covered by an authentication gateway such as the one described here can be divided as follows:

3.1. Non-Grid users accessing Grid resources

Usually, such a user would not have an X.509 certificate. Without the gateway, this user would be required to obtain an X.509 certificate (possibly duplicating his credentials, since it is possible he possesses other federated identities) and to register into a Grid Virtual Organization (VO). If the VO does not exist yet, it must be set up, which may be a time and resource consuming process; this is especially true for individual users and for small collaborations, who do not often

have the critical mass to support these tasks. Finally, sites need to allow the new user and possibly the new VO so that he can use their resources.

With the gateway, the user gets authenticated through a federated identity. The gateway then provides the user with a short-lived X.509 certificate and registers the user into a dedicated VO. It is to be noted that this method works because DN persistency is guaranteed across credential re-creation.

Sites then need to accept the VO the user has been made part of. It is possible to only accept subgroups of this VO, if this is desired. In practice, one could have a catch-all VO, or set-up multiple VOs for this purpose, depending on operational or business considerations.

In the end, users have gained access to Grid resources (being able to *e.g.* get VOMS proxies) with minimal overhead for Grid sites and for the users themselves.

3.2. Grid users accessing non-Grid resources

In this case a user already has an X.509 certificate and there is therefore no need to get another, short-lived one. In the current WNoDeS gateway implementation, the same services used in Grid computing for authentication and authorization are also used for the access to non-Grid (*e.g.*, Cloud) resources.

In practice, the current implementation of the gateway supports the Virtual Organization Membership Service (VOMS) [10] for VO membership, and gLite Argus [11] for authorization policies. This allows us to automatically support existing Grid certificates and VOs, so that existing Grid users are able to access Cloud resources, just using their Grid credentials.

4. State and Conclusions

A KCA and a Shibboleth IdP are up and running and an *internal* online CA issuing short-lived certificates is also available. Internal here means that the certificates issued by this CA are only valid within INFN; we are currently working on having this CA recognized by the IGTF, which is a process that takes some time. Nevertheless, the current set-up proves that the idea of a generic authentication gateway is a viable one. In the final implementation, the test Shibboleth IdP at INFN will be replaced by the IdP of the Italian Federation of Universities and Research Institutes (IDEM) [12]. A WNoDeS web application is also being written to interface to the gateway, so that users may directly exploit the cases described in Sections 3.1 and 3.2.

This gateway, once deployed, will give users without traditional X.509 digital certificates the possibility to access distributed Grid computing resources. Conversely, it will let Grid users to access Cloud-type resources without the need of additional credentials.

Future developments will center on providing a complete implementation of the gateway; on the writing of a Web application fully exploiting the gateway capabilities; and on the study of the possibility of using such a gateway to access distributed Cloud resources.

References

[1] ITU-T Recommendation X.509, `http://www.itu.int/rec/T-REC-X.509`
[2] Salomoni D, Italiano A, Ronchieri E, WNoDeS, a tool for integrated Grid and Cloud access and computing farm virtualization, International Conference on Computing in High Energy and Nuclear Physics (CHEP'10), To be published, 2011
[3] WNoDeS Web Site, `http://web.infn.it/wnodes`
[4] Clifford Neuman B, Ts'o T, Kerberos: An Authentication Service for Computer Networks, IEEE Communications 32 (9): 338, 1994
[5] Doster W, Watts M, Hyde D, The KX.509 Protocol, `http://www.citi.umich.edu/techreports/reports/citi-tr-01-2.ps.gz`
[6] The International Grid Trust Federation, `http://www.igtf.net/`
[7] Shibboleth, `http://shibboleth.internet2.edu/`
[8] Witzig C, Shibboleth Interoperability Through a Short-Lived Credential Service (SLCS), `http://www.switch.ch/grid/slcs/index.html`

[9] MyProxy Credential Management Service, http://grid.ncsa.illinois.edu/myproxy/
[10] Virtual Organization Membership Service, https://twiki.cnaf.infn.it/twiki/bin/view/VOMS/WebHome
[11] gLite Argus, https://twiki.cern.ch/twiki/bin/view/EGEE/AuthorizationFramework
[12] Federazione IDEM, https://www.idem.garr.it/
[13] Protection of private key data for end-users in local and remote systems, https://www.eugridpma.org/guidelines/pkp/pk-protection-1.2-20110322.pdf