Identity Federation in Cloud Computing

Valentina Casola  
Dipartimento di Informatica e Sistemistica  
Università degli Studi di Napoli Federico II  
casolav@unina.it

Massimiliano Rak  
Dipartimento di Ingegneria dell’Informazione  
Seconda Università di Napoli  
massimiliano.rak@unina2.it

Umberto Villano  
Dipartimento di Ingegneria  
Università del Sannio  
villano@unisannio.it

Abstract—Both cloud and GRID are computing paradigms for the large-scale management of distributed resources. Even if the first is usually oriented to transaction-based applications, and the latter to High Performance Computation, there is a lot of interest in their integration. This is typically obtained through the Infrastructure-as-a-Service cloud model, which is exploited in the GRID context to offer machine with full administration rights to users. In this paper the focus is on the security problems linked to the integration of cloud and GRID computing. It is proposed the adoption of identity federation between different security domains to manage the relationship between the user machines and the standard GRID infrastructure. This solution is experimented within PerfCloud, a cloud implementation that exploits an underlying GRID platform.

I. INTRODUCTION

According to the definition by NIST, cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service), three service models (IaaS, PaaS, SaaS), and four deployment models (Private, Community, Public, Hybrid) [1]. This can rely on a very high number of different technologies, and has an incredibly large set of possible use cases [2].

In this paper the focus will be on the security problems linked to the integration of cloud computing and GRID computing. The latter is a computing paradigm partially similar to cloud computing, whose most relevant implementations, Globus [3] and gLite [4], are largely diffused in the scientific community. As computing GRIDs enable access to high performance distributed resources in a simple and standard way, they are widely exploited in the e-science world for high performance computing tasks.

As mentioned above, GRID and clouds have many points in common, not to mention the use of similar underlying technologies. However, they are typically exploited for different purposes by different classes of users. In short, clouds are for users that are prone to buy computing resources to get their results as soon as possible. On the other hand, GRID users wish to exploit the optimum number of resources that solve the problem, overcoming the boundaries of a single enterprise. In fact, the two technologies complement gracefully each other, and currently their integration is actively investigated. At the state of the art, the two principal approaches used are the following:

• GRID-on-Cloud: a cloud IaaS (Infrastructure as a Service) approach is exploited to build up and to manage a flexible GRID system [5]. As the GRID middleware runs on cloud-managed virtual machines, the main drawback of this approach is performance. Virtualization inevitably entails performance losses as compared to the direct use of physical resources.

• Cloud-on-GRID: the well-known and stable GRID infrastructure is exploited to build up a cloud environment. This is usually the preferred solution [6], because the cloud approach mitigates the inherent complexity of the GRID. The use of Globus workspaces [6], along with a set of GRID services for the Globus Toolkit 4 is the prominent solution, as in the Nimbus project [7].

The cloud-on-GRID approach has gained large interest in the scientific community, as it helps to manage some of the most common problems with parallel programming: the incredible variety of different softwares (and software versions), configurations, operating systems and hardware layers that often have to coexist, but are not mutually compatible. Thanks to the adoption of clouds, and their underlying virtualization techniques [8], [9], [10] it is possible to provide GRID users and parallel application developers with a “clean” environment, freely and completely customizable. Current cloud-on-GRID systems [7], [11], [12] offer services to manage (create, destroy, modify, . . .) virtual clusters (i.e., clusters composed of virtual computing resources) on the underlying GRID infrastructure.

In a “pure” cloud-on-GRID system, the virtualized environments assigned to users are mutually isolated, and do not know of each other. Given two scientific communities working on different, but related problems, the users of each group have access to a freely-customizable computing platform, but cannot invoke the software services developed by the other group. Even if their computing environments exploit physically the same GRID, they are not interopera-
In PerfCloud, an existing cloud-on-GRID infrastructure with provision for predictive performance evaluation [13], [12], this problem is solved by integrating the virtual resources offered by the cloud into the underlying GRID. In other words, given an existing computing GRID, users can gain access to virtualized resources (e.g., virtual clusters) through a cloud interface, and these virtual resources are integrated in the existing GRID and can cooperate with its physical (and virtual) component systems. Stated another way, the virtual clusters provided by the cloud are automatically part of the underlying GRID infrastructure. This approach, which we call cloudgrid, achieves full cloud and GRID integration. It is a sort of three-layered “GRID-on-cloud-on-GRID” (Figure 1). If one looks only at the two upper layers, it can be regarded as a GRID-on-cloud system. Looking at the two layers at the bottom, it is instead a cloud-on-GRID system.

Cloudgrid is an almost obvious approach to solve the interoperability problem discussed above, but it entails big security issues. As a matter of fact, any virtual cluster administrator, who is for the GRID a standard user with no physical resource management grants, can act as a GRID administrator once the virtual cluster is integrated in the GRID. Furthermore, in the cloud environment, the cooperation among virtual resources is desirable and the extension of trust to untrusted users through identity federation is still an open issue.

The objective of this paper is to tackle these cloudgrid security issues. We propose a solution where any virtual cluster administrator configures a new and independent security domain (and a new Certification Authority), granting its access to the underlying GRID infrastructure and to virtual clusters through an Identity Federation mechanism. In other word, the virtual cluster will build up a completely new security domain, independently managed by the cloud user. Successively, a federated security domain will be built between the virtual cluster domains or with the underlying GRID. In accordance with the basic principles of the cloud-oriented approach (and most notably, with the on-demand self-service characteristic), the identity federation problem, usually managed manually, will be completely automatized thanks to the introduction of an Interoperability System that does not require human intervention. The proposed architectural solution has been implemented for PerfCloud and will be presented as a case study; nevertheless, it has a wider validity, as it can be adopted for every GRID-on-cloud implementation.

The remainder of this paper is organized as follows. The next section briefly introduces PerfCloud, the proposed cloudgrid infrastructure and its security issues. In Section 3 the identity federation problem in GRIDs is dealt with, and it is described an interoperability system that allows cooperation among untrusted domains. Section 4 presents an implementation of the interoperability system for Identity Federation in clouds. The paper closes with the conclusions and the presentation of future work.

II. PERFCloud AND SECURITY ISSUES

PerfCloud [12] is a cloudgrid implementation, offering services for the creation and management of virtual computing resources (virtual clusters) on the top of a computing GRID. The cluster-on-demand functionalities are integrated with a simulation environment able to predict user application performance on the newly instantiated virtual clusters, which are automatically integrated in the existing GRID. PerfCloud builds a IaaS (Infrastructure-as-a-Service) cloud environment upon a Globus GT4 GRID infrastructure. The resources offered to users by the PerfCloud system are virtual clusters, i.e., sets of preconfigured virtual machines connected by a (virtualized) network. The cloud user owns such systems and has administration rights on them, but he cannot manage the physical resources.

Figure 2 describes the overall architecture of PerfCloud. The PerfCloud application client resides on a user machine (which has access to the GRID environment) and interacts with the PerfCloud system by invoking GRID services. Furthermore, it manages GRID connections, also providing utilities for end-users (notably, the above-mentioned performance analysis services). The architecture provides different GRID services that enable the user to build up a new cluster as a GRID Virtual Workspace [6] with full access rights. In light of the above, the PerfCloud architecture can be subdivided into three main components, as is shown in the figure:

- **Services**, which offer the PerfCloud functionalities to the GRID environment.
- **Images**, which are the Virtual Cluster (VC) Node images, containing all the software needed to integrate the VC into the GRID environment (a GT container) and to offer services to the final user (a set of GS deployed on the VC container), along with other software needed
for application development and execution (compilers, messaging libraries and run-time support, . . .

- **Client**, which allows the final user to interact with the Cloud environment.

As shown in the figure, the virtual clusters host a Globus container on the virtualized cluster front-end, together with a set of predefined services. Further details on the PerfCloud implementation can be found in [13], [12].

In previous paper [14] we have enhanced PerfCloud with a role-based access control mechanism to authorize access to the resources offered by the framework. In particular, we identified the following set of roles for the users of the cloudgrid system:

- **System Administrator** manages the physical machines from HW up to the operating system level. He is also responsible for installing, configuring and starting the Globus platform and for managing identities and accounts (issues digital certificates, enables users);
- **GRID User** can access and use generic GRID resources;
- **Cloud User** is a GRID user with special rights for accessing and using Virtual Cluster and Virtual Machines GRID resources;
- **Cloud Administrator** is a GRID User with special rights for creating new Virtual Clusters and managing the rights of the cloud users.

It is interesting to point out that the cloud administrator does not have administrator rights on the virtual cluster, which are owned by the cloud user. A cloud administrator is instead able to create a new set of virtual clusters and to offer them to different cloud users. It should be noted that once a virtual cluster set exists and is assigned to a cloud user, the data and configuration of the VC are his private property.

Even if offering full rights on the virtual cluster is one of the aims of clouds, this can have a side effect on the cloudgrid approach. In fact, the VC administrator has full right access to the VC and he can also manage the new GRID site that is hosted by the physical GRID site. This represents a big security issue. An user of an hosted site could access physical resources if the cloud administrator does not enforce proper security policies in issuing credential or delegating rights, or if it wants to abuse of his role on the physical resources. Furthermore, to fully enable the cloud approach, it is desirable to grant cooperation among virtual resources even when they are offered by potentially untrusted domains.

This problem cannot be solved only by an access control mechanism. At the state of the art, when a GRID infrastructure has to accept a new virtual site as part of the infrastructure, usually a complex procedure takes place. This involves human interaction and manual evaluation of the security policies adopted by the site to enable cooperation and to trust each other. But, as one of the objectives of any cloud system is to automatize all procedures (on-demand self-service, according to NIST document [1]), the safe “manual” GRID admission procedure is not a reasonable solution.

In the rest of this paper, we will describe the proposal of an interoperability system that provides an automatic way to implement the Identity Federation concept and to assign specific roles to “federated” users.

### III. Identity Federation in GRIDs

Our approach to manage the security issues that arise from the integration of the Virtual Cluster in the underlying GRID layer is based on two key points: (i) a Cloud administrator builds up a new Virtual Organization for each Virtual Cluster as an independent security domain (i.e., with its own Certification Authority), (ii) an identity federation is implemented to enable an automatic interoperability among a Virtual Cluster and the physical Grid site or between two Virtual Clusters.

A Cloud user will be able to customize the Certification Authority of its Virtual Cluster, or even to setup a completely new one.

In order to describe clearly the problem, we summarize here how the authentication process takes place in a GRID environment (or in a cloud-on-GRID, which is the same). In a cloud environment, users possess a set of Grid credentials consisting of a X.509 v3 digital certificate [15] and a private key. The certificate is digitally signed by a Certification Authority that guarantees for the binding of the entity distinguished name (DN) to its private key. The authentication mechanism involves the presentation of the certificate and the possession of the corresponding private key. A known problem in such a protocol is the protection of the private key. At this aim, two strategies are commonly adopted: i) the key is protected with encryption or by storing it on a hardware token (e.g., a smart card); ii) the
private key has limited lifetime, after which it is no longer valid.

The Globus Toolkit security implementation, known as the GRID Security Infrastructure (GSI) [16], follows the second strategy, using Proxy Certificates [17]. Short-term credentials created by a user can successively be used in the place of traditional long-term credentials to authenticate him. The proxy certificate has its own private key and certificate, and is signed using the user long-term credential. A typical session with the GSI would involve the GRID user (End-Entity) using its passphrase and the GSI command grid-proxy-init creating a proxy certificate from its long-term credential. The user could then use a GSI-enabled application to invoke a service operation from a Globus Toolkit GRID Services Container [16].

From the GRID resource point of view, to fully perform the authentication process, a certificate validation service interface should be defined and used within the Open GRID Services Architecture (OGSA) [18] implementation to:

1) parse a certificate and to verify desired attribute values, as the validity period, the Distinguished Name and so on,
2) perform path validation (basic and extended) [15] on a certificate chain and verify the revocation status according to updated certificate revocation lists (CRLs) or through an on-line Certificate Status Protocol,
3) return attribute information for different usage.

Available GRID implementations, as the Globus Toolkit (GT4) [1], provide static mechanisms to perform a basic certificate path validation process, i.e.:

1) Cryptographic verifications over the certificate path (verifying the digital signature of each certificate).
2) Verification of each certificate validity period.
3) Verification that the root certificate in the chain is trusted.
4) Verification of the certificates status to ensure that they have not been revoked or suspended.

In particular, in GT4 the first certificate in the chain is considered a Trust Anchor if it has been stored into the GRID node /etc/grid-security/certificates/ directory, while the certificate status is retrieved from a locally-stored Certificate Revocation List (CRL). To validate a digital certificate issued by any other CA, an explicit cross-certification process is needed and the so-called “extended path validation” [15] must be enforced. The main idea behind the extended path validation mechanism is to define an approach that enables any GRID relying-party to validate in real-time a digital certificate issued by any other CA, even if they do not belong to the same trusted domain. So, there is also the need to evaluate and to extend trust to the authority that issued such certificates. In the GRID environment, if we are able to automatically perform the extended path validation, we are able to perform Identity Federation since we can accept and validate credentials from any domain. Our approach is to build a dynamic federation of CAs by automatically evaluating their certificate policies (automatic cross-certification) [19].

In Figure 3 the logical blocks of the Interoperability System (IS) are shown. It acts as an intermediary between the certificate verifiers (relying parties) and the issuing CAs by managing (retrieving, elaborating and updating) the information needed to perform the extended path validation: the list of accredited CAs, the list of revocation sources and the Certificate Policies.

Figure 3. Interoperability System (IS) main components

The IS may be allocated within the Trusted Third Party domain. In our case, it will be offered as a service by the Physical GRID site, and must perform two main tasks: 1. on-line validation of the certificate status, 2. evaluation of the issuing CA security level.

For the first task we will use a GRID Validation System able to retrieve the status of a digital certificate through the OCSP protocol in a CA federation. This feature has been implemented as an extension of Globus Toolkit, and it is named OGRO [20], [21]. As regards the second task, for evaluating a CA security level we have adopted the Reference Evaluation Methodology (REM), a policy-based evaluation technique that was primarily proposed for evaluating Certification Authorities in the cross-certification process [19].

The REM approach is based on the formalization of a GRID-CA Certificate Policy i) to determine if this Authority is compliant with another CA Certificate Policy and ii) to quantitatively evaluate the Global Security Level (GSL) of this CA. The GSL is a quantitative measure of the CA trust degree, and it will be compared with the other CA level to
decide to extend or not the trust to an incoming user request. The GSL will be retrieved and evaluated by a GRID Service before extending trust (to federate an identity).

In [22] we proposed a Interoperability System based on Policy evaluation and on the OCSP protocol (POIS), in next section we will illustrate how it can be adopted in a Cloud-on-GRID environment to enable Identity Federation.

IV. Identity Federation in PerfCloud

In this section we illustrate the use of the Policy and OCSP based Interoperability System (POIS) to enable Identity Federation among cloud users. Figure 4 shows the main actors and the system components that have an active role within the PerfCloud framework during the invocation of a service offered by a Grid container that supports federation (Federated Grid Container).

At a coarse view, POIS offers the following features:

1) Manages (retrieves, updates) the list of Virtual Organizations CAs (accredited by the root-VO).
2) Manages (retrieves, updates) the accredited CAs Certificate Policies.
3) Manages (retrieves, updates) the accredited CAs CRLs.
4) Communicates validation information to relying parties over OCSP.

As for the features 1 and 2, we assume that the database with the list of accredited CA and their Certificate Policies is manually and off-line managed by system administrators (in particular by the GRID Administrator). The CRLs (feature 3) from each accredited CA are retrieved by using CertiVeR CRL Updater module (described in [23]), so they can be used in the Extended Path Validation process. The Policy Evaluator subsystem implements the REM evaluation technique to evaluate the GSL on any VO CA Policy. The GSL is then communicated to relying parties by the OCSP protocol (feature 4).

To summarize, POIS is able to perform the Extended Path Validation thanks to a specific client (like OGRO [24]) which provides the following two enhancements to the GT4 basic path validation algorithm: i) certificate status is extracted from the OCSP response and ii) the GSL evaluated by POIS is compared against the GSL value required by the Grid Service for trusting the request from a different Virtual Organization.

We wish to point out that it is up to the Grid Container to interact with the Interoperability System. It can perform such an operation if and only if a suitable client has been added (possibly OGRO). We will denote such container as a Federated Grid Container (and, consequently, we will refer to Federated GRID Services and Federated Resources). When a customer asks for a Virtual Cluster, the Cloud Administrator will create a new Virtual Cluster and configure images and services with or without federated facilities according to the customer desiderata.

Once federated services are available, different scenarios may occur. They are strictly related to the relations between the different Virtual Organizations. In particular:

1) the new Virtual Cluster has a pre-configured Certification Authority that has a hierarchical relation with the Root Certification Authority of the Physical Cluster;
2) the new Virtual Cluster CA has not any relations with the other Virtual Clusters or with the Physical Cluster.

In the first case, basic path validation is enough to extend trust to cloud users whose certificates have been issued within the hierarchy (the Certification Path validation process does not fail as the Root CA acts as the Trust Anchor). Nevertheless, the adoption of a hierarchical approach should be discouraged (i) to enable Cloud Administrators to configure their own Certification Authorities and (ii) to avoid that a Cloud Administrator issues to its users digital credentials that can be directly validated and accepted by all physical grid resources.

In the second case, an explicit federation mechanism is needed to cooperate. Identity Federation does not automatically mean to grant access to a federated resource. In fact, once a user has been authenticated by an external Virtual Cluster, its identity will be mapped to a specific (federated) role and a role-based access control policy will be enforced at resource level [14]. For example, a user from the Virtual Cluster 1 that has a role of Cloud Administrator can access specific resources of Virtual Cluster 2 but, possibly, with a non-administrator role.

In the following, we will describe in detail the second scenario, including the role mapping and authorization aspects.
A. A case study scenario of access to Federated Resources

Figure 5 shows a typical scenario, where a cloud user belonging to the Virtual Cluster 1 (denoted user@VC1 from Virtual Organization 1) requests a GRID service offered by the physical cluster (FGC@Root within the Root Virtual Organization). The interoperability is achieved because the GRID Services Container has been federated.

To let POIS work in an off-line manner, any CA from the different Virtual Organizations submits its Certificate Policy (CP) to the POIS. Then, the Policy Evaluator subsystem evaluates the corresponding GSL that is stored into OCSP Responder DB along with the CA data (this step is not shown in the diagram).

The main steps of this scenario are:

1) When a Cloud user from the Virtual Cluster 1 (user@VC1) requests a GRID service offered by the Root Virtual Organization, the Federated Grid Container (FGC@Root) performs extended path validation:
   - basic path validation on the proxy certificate is performed;
   - the digital certificate status is evaluated on-line through the OCSP Responder;
   - the GSL value is directly retrieved from the POIS (that holds a database with all pre-evaluated Certification Authorities of the accredited Virtual Clusters).
   - the GSL of the Cloud user’s CA is compared against the minimum required-GSL defined by the Federated Grid Container to extend trust, and if $GSL_{VC1} > GSL_{GC}$, the validation is successful.

2) If the extended path validation is successful, the cloud user is mapped to a “federated user”. Before accessing the requested resource, the corresponding role is retrieved by a role repository and a role-based access control mechanism is enforced by using an external Policy Decision Point (PDP) configured at Resource Level [16].

In Table I, we report different cases of federation and the corresponding “federated role” that is assigned considering who requests a resource and where the resource is hosted.

We can note that if an user from VC1 requests a service of the Root VO, it is recognized as a federated limited user and a limited set of rights will be granted in the rbac policy (for example, we could grant computing resources, but not administration ones). If the same user from the VC1 requests a service of the VC2, it will be recognized as a federated user and the rbac policy will grant services according to the agreements between parties. If a user from the root VO (or a user from VC1) requests a service of the Root (or of the VC1) - this is a trivial case - basic path validation is performed and the user role does not change being the requester not federated but “internal” and this not requires the extended validation mechanism. The last case is when a user from the Root VO requests services offered by a VC. Once again, extended path validation is performed and the assigned role and the policy will depend on the agreement between the parties.

In conclusion, thanks to the Interoperability System we are able to extend trust to cloud users even when they come from untrusted domains. The whole mechanism works because we are able to automatically validate any digital certificate and to assign it a degree of trust (the Global Security Level). On the basis of this trust degree, we are also able to assign a dynamic profile to users coming from the outside world, to let them access to computing or even administration resources. We think that this is a very important point in a cloud environment, as identity federation is an open issue that still limits the wide adoption of these infrastructures.

V. CONCLUSIONS AND FUTURE WORK

One of the main security issues for the cloud community is identity federation. Many cloud solutions leverage several services without integrating identity management and access control mechanisms.

In this paper we have presented a solution to some open security issues in cloud architectures and, in particular, in PerfCloud, a framework which offers cloudgrid functionalities. We have focused our attention on the need of users to access both virtual and physical grid sites without abusing the privileges linked to their role in their own domain. The proposed Identity Federation approach is based on
the definition of an Interoperability System that validates certificates by dynamically performing a Cross Certification among untrusted Certification Authorities (we adopted the REM methodology to evaluate and compare the trust level of a Certification Authority) and by evaluating on-line the status of a digital certificate (we proposed the adoption of the OCSP protocol and a Grid OCSP client named OGRO). To illustrate in detail the proposed architecture we have presented a case study and some typical scenarios where users from a virtual cluster request resources hosted by other clusters (virtual or physical), illustrating the federation process and the enforcement of access control policies that are specific of “federated users”.

As regards the future evolution of our work, we intend to integrate the Interoperability System with different authorization services (as XACML, PERMIS or ShibGRID) to evaluate the trade-off between the security level provided and overall system performance. This will make it possible for the cloud to offer guarantees on the quality of service and to negotiate Service Level Agreements in an automatic way.

REFERENCES


