The CloudGrid approach: Security analysis and performance evaluation

Valentina Casola\textsuperscript{a}, Antonio Cuomo\textsuperscript{b}, Massimiliano Rak\textsuperscript{c,\ast}, Umberto Villano\textsuperscript{b}

\textsuperscript{a} Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Italy
\textsuperscript{b} Dipartimento di Ingegneria, Università degli Studi del Sannio, Italy
\textsuperscript{c} Dipartimento di Ingegneria dell’Informazione, Seconda Università di Napoli, Italy

\begin{abstract}
Both cloud and grid are computing paradigms that manage large sets of distributed resources, and the scientific community would benefit from their convergence. This paper proposes a novel computing model, \textit{cloudgrid}, able to achieve full cloud and grid integration. After presenting its three-layer architecture, the security issues involved are analyzed, proposing a solution based on fine-grained access control mechanisms and identity federation that allows cooperation and interoperability among untrusted cloud resources. The overhead introduced by the multiple-layer architecture and by the security system are measured by extensive testing on a prototype implementation, and a trade-off analysis between security and performance is presented.
\end{abstract}

1. Introduction

Today the term “cloud” is widely adopted by the Internet community with many different meanings, from that of “well-managed” data-center, to that of hardware–software structure supporting the concepts of \textit{utility computing} and \textit{elastic computing} introduced by Amazon [1]. A recent definition by Foster says that cloud is a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet [2]. A formal definition is provided by NIST, which describes cloud computing as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The cloud model promotes availability, and is characterized by five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service), three service models (IaaS, PaaS and SaaS), and four deployment models (Private, Community, Public and Hybrid) [3]. It can rely on a high number of alternative technologies, and has an incredibly large set of possible applications [4]. Whether cloud computing will be the next “big thing” or not, it is a fact that this new computational model is having great impact on the worldwide computing scenario.

On the other hand, grid computing [5] basically aims at enabling access to high performance distributed resources in a simple and standard way. In grids, users can compose stateful services to build up complex and computation-intensive tasks. This is obtained by means of a middleware paradigm: every host has a grid interface, and developers adopt middleware-dependent APIs for building up their applications. Currently the Globus Toolkit [6] and gLite [7] are the most relevant and widely used implementations available.

Grids 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, grid users wish to exploit the optimum number of resources that solve their problem, overcoming the boundaries of a single enterprise. On the other hand, clouds are for users that are prone to buy computing resources to get their results computed as soon as possible. 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 for grid and cloud integration are the following:

- \textbf{Grid-on-cloud}: a cloud IaaS (Infrastructure as a Service) approach is exploited to build up and to manage a flexible grid system [8]. As the grid middleware runs on cloud-managed virtual machines, the main drawback of this approach is
low 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. Usually this is the preferred solution [9], because the cloud approach mitigates the inherent complexity of the grid.

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 of their underlying virtualization techniques [10–12] it is possible to provide grid users and parallel application developers with a “clean” environment, freely manageable and fully customizable. Current cloud-on-grid systems [13–15] 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 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 interoperable.

The authors of this paper are involved in a project (PerfCloud) that aims at joining the cloud-on-grid and grid-on-cloud approaches, by implementing a cloud on the top of a grid, and integrating the cloud resources leased to users in the existing grid. In PerfCloud, given an existing computing grid, users can gain access to virtualized resources (e.g., to 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 have named cloudgrid, makes it possible to achieve full cloud and grid integration (Fig. 1).

The cloudgrid approach has clear advantages, as it offers at the same time the grid and the cloud environments, and these are also integrated. However, it may have drawbacks in terms of performance and security, as discussed in the following. Companion papers sketch the security requirements for a cloudgrid [16], propose a possible solution based on identity federation [17], and present a sample of the measurements of the overheads involved in the provisioning of a security infrastructure [18]. On the one hand, this paper summarizes our previous work, in that it presents in an organic way the cloudgrid approach and its existing implementation. On the other hand, it includes novel contributions, as it presents for the first time a working authentication infrastructure able to tackle the cloudgrid security problems, with a fairly complete quantitative evaluation of the overheads introduced. It will be shown that the overhead introduced by the additional layers and the security mechanisms adopted, almost surprisingly, have negligible impact on the overall performance of the grid.

The remainder of this paper is organized as follows. The next section details the cloudgrid approach and describes the PerfCloud prototype. Section 3 analyzes the cloudgrid security issues, pointing out the need for a role-based access control (RBAC) model and an identity federation system. Section 4 proposes the technique
that can be adopted as solution, whose implementation details are described in Section 5. Section 6 presents a detailed performance analysis of our cloudgrid implementation, PerfCloud. Finally, related work is discussed and the conclusions are drawn, outlining the future research.

2. The cloudgrid approach

As anticipated in the introduction, the work described in this paper aims at joining the cloud-on-grid and grid-on-cloud approaches, by implementing a cloud on the top of a grid, and integrating the cloud resources leased to users in the existing grid. In our prototype implementation (PerfCloud, described below), given an existing computing grid, users can gain access to virtualized resources (virtual clusters) through a cloud interface. 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 have called cloudgrid, makes it possible to obtain full cloud and grid integration. It is a sort of three-layered “grid-on-cloud-on-grid” (see again Fig. 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.

The motivation for the integration of grid and cloud models is twofold. The cloud-on-grid model offers cloud services, but the leased computing resources (e.g., the virtual clusters) are mutually isolated and cannot inter-operate. On the other hand, the grid-on-cloud model makes it possible to build inter-operable systems (after all, this is the main grid advantage), but these systems are built from resources leased from clouds and cannot exploit the great potential of existing scientific grids. Cloudgrid gets the advantages of both cloud-on-grid and grid-on-cloud models (manageability, flexibility and interoperability, respectively), without their disadvantages.

On the down side, this architecture has potential performance and security drawbacks. As far as performance is concerned, a three-layer architecture should have higher overheads as compared to a two-layered one. In the following sections, extending the results presented in a previous paper [18], we will show that it is not so, since unexpectedly performance (in terms of service response time) is not affected by the addition of the third layer. The security issues are related to the protection of virtual and physical resources, and to the desire to share resources among untrusted domains. In fact, these represent access control and identity federation security requirements, respectively, and they are among the most discussed open security issues in the literature [19,20]. 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, and this potentially is very dangerous. On the other hand, any virtual resource can be shared with other trusted users if proper federation rules are enforced.

In the next subsection, we will present the structure of PerfCloud, our cloudgrid implementation, introducing its main features. The security requirements for cloudgrids, the solution we propose and its implementation are dealt with in Sections 3–5.

2.1. PerfCloud: the cloudgrid prototype

Our work on cloudgrids has been performed with a prototype environment, PerfCloud [21,15]. PerfCloud builds an IaaS (Infrastructure as a Service) cloud environment upon a grid infrastructure. IaaS cloud environments usually adopt virtualization engines in order to manage (start, stop) the virtual machines offered to the final user. Virtual Clusters (VCs) are composed of two or more virtual machines connected through a (virtual) dedicated network. PerfCloud leases virtual computing resources that usually (but not mandatorily) are organized in virtual clusters. These clusters are automatically included in the underlying grid, and therefore can offer their computational resources to the grid users. The Virtual Clusters (VCs) provided by PerfCloud have a master-server structure, with a FrontEnd (FE) that has public IP and is the only machine externally accessible from the network. They are equipped with typical cluster software (like MPI, PVM, Tentakel and others) and job management tools (SGE or PBS).

The PerfCloud architecture can be subdivided into three main components, shown (left to right) in Fig. 2, which describes the process of leasing a virtual cluster “loaded” with predefined disk images through a service-oriented interface. In the figure, the presence of a (physical) grid and the automatic enrollment of the VC in this grid are neglected for the sake of clarity. The three components are as follows:

Client, an extensible client which allows the final users to interact with the cloud environment;
Service Oriented Interface, the set of (grid) services that offer the PerfCloud functionalities to the grid environment. The offered services range from Cloud IaaS Services (offering Virtual Machine (VM) and VC management services) to added-value services for performance evaluation;
Images, the set of image files used to start up the virtual machines and clusters (virtual engines start up a machine by means of a file that represents the VM file system and contains all the software of the VM, operating system included).

PerfCloud provides a service-oriented interface on top of existing virtual machine engines (like Xen [10] and KVM [22]) and schedulers (among the others, OpenNebula [23] and Nimbus [13]). It offers a resource abstraction (in terms of WSRF [24]) that enables services (and service invokers) to manage directly the functionalities offered by virtualization engines and VM schedulers. PerfCloud was initially implemented on the top of Virtual Workspaces (see Section 7), but its current implementation is self-contained. As shown in Fig. 3, PerfCloud resource representation (hexagons in the picture) can be Virtual Workspaces or an abstraction of VMs and VCs. The invocation of suitable PerfCloud services lets users read and control the state of VMs. When cloud schedulers as OpenNebula or Nimbus are used, it is also possible to perform more complex operations, such as virtual machine migration between different physical machines.

The PerfCloud API offers services to set up the virtual clusters, to measure the performance of the virtual cluster, to predict the performance of HPC applications using as input high-level descriptions of the software. The interested reader is referred to our previous papers [21,15,25] for further details on PerfCloud services.

3. Cloudgrid security requirements

As a matter of fact, in a cloud platform resource administration is completely up to the user, according to the self-service principle. This is reasonable, due to the adoption of the pay-per-use paradigm: every user is induced to manage resources well, in order to keep costs low. On the other hand, in a grid environment resources are often provided for free. Thus it is necessary to differentiate users, introducing figures with higher privileges to perform administration tasks. It is important to point out that this role cannot be played by physical machine administrators, since resources to be administered are location-transparent grid resources. The only solution is to perform these tasks through suitable grid services accessible only to users with
enough privileges. Hence fine-grain access control mechanisms are needed to grant different privileges to service administrators, system administrators and end-users.

In a cloudgrid architecture, we can identify the following roles and security policies:

- **System Administrator**: The System Administrator can manage the physical machines from the hardware up to the operating system level. He is responsible for installing, configuring and starting the grid platform and its Certification Authority, for managing grid identities and accounts, for updating the security policies on the system;
- **Grid User**: The Grid User can create and use grid resources;
- **Cloud Administrator**: The Cloud Administrator is a Grid User with additional rights. He can supervise the cloud environment, creating/maintaining new Virtual Clusters and managing Cloud User rights. In particular, he can enable/disable a Cloud User for the access to one or several Virtual Clusters;
- **Cloud User**: The Cloud User is a Grid User with additional rights. He can turn on/off, access, use and configure the Virtual Clusters previously assigned to him by the Cloud Administrator.

It is interesting to point out that the cloud administrator is able to create a new set of virtual clusters and assign them to cloud users, but does not have administration rights over them, which are owned by the cloud users. Stated another way, once a virtual cluster exists and is assigned to a cloud user, the data and configuration of the virtual cluster are his own private property.

Even if offering to users full rights on the virtual cluster is one of the aims of clouds, this can have a side effect in a cloudgrid: in fact, a VC administrator has full access rights to the VC but he can also manage the new grid site that is hosted by the physical grid.
site. This represents a big security issue: a user of a hosted cloud site could access physical resources if the cloud administrator does not enforce proper security policies in issuing credential or delegating rights, or if he abuses of his role on the physical resources. As a consequence, a cloudgrid requires a powerful role-based authorization mechanism.

Furthermore, to fully enable the cloud approach, it is desirable to grant cooperation among users and virtual resources even when they are offered by potentially untrusted domains. This problem is strictly related to identity federation among distributed domains, and cannot be solved by traditional 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, a complex evaluation procedure usually 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 [3]), the safe “manual” grid admission procedure is not a reasonable solution, and a federated approach to security and, more specifically, to identity management is required [19].

Summarizing, the access control requirement can be seen from two different points of view. If we look at the vertical layered structure (grid-cloud-grid-physical resource), we need to enforce proper security access control rules to protect the virtual resources that are owned by a cloud user and the physical environment that is shared by all virtual resources. If instead we look at the cloud layer, which includes all virtual resources, we need a secure way to allow access to all users and to enable resource cooperation, even if they belong to reciprocally untrusted cloud sites.

Data integrity, confidentiality and privacy are not secondary requirements: suitable security mechanisms must be enforced not only at application level, but also in the other layers and components of the architecture (e.g., network, transport, service communication, internal application and databases). As we will discuss later, the adoption of secure protocols and encryption techniques has proven to be very useful in such distributed architectures.

In conclusion, cloudgrids sum up the security problems of cloud-on-grid and grid-on-cloud solutions. These problems, to the best of the author’s knowledge, have never been addressed in the literature, either separately, or (a fortiori) in a unified way.

4. An Interoperability System for the identity federation in cloudgrid

Our solution 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), and (ii) an identity federation is implemented to enable an automatic interoperability between a virtual cluster and the physical grid site or between two virtual clusters. Moreover, a cloud user can customize the Certification Authority of its virtual cluster, or even set up a completely new one.

In order to describe clearly the problem, it is worth summarizing here how the authentication process takes place in a grid environment (or in a cloud-on-grid, which is the same). In a grid environment, users possess a set of grid credentials consisting of a X.509 v3 digital certificate [26] 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 issue 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 Grid Security Infrastructure (GSI) [27], follows the second strategy, using Proxy Certificates [28]. 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.

From the grid resource and service 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) [29] implementation:

1. to parse a certificate and to verify the desired attribute values, as the validity period and the distinguished name,
2. to perform path validation (basic and extended) [26] on a certificate and to verify its revocation status, according to updated Certificate Revocation Lists (CRLs) or through an online Certificate Status Protocol,
3. to return security attribute information.

Available grid implementations, as the Globus Toolkit (GT4) [1], provide static mechanisms to perform a basic certificate path validation process. Basic path validation means to be able to verify identity and authentication of users whose certificate has been issued by a Trusted Certification Authority. It consists of the following steps:

1. cryptographic verifications over the certificate path (verification of 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 certificate 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” [26] must be enforced. The main idea behind the extended path validation mechanism is to define a procedure 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. The first step is to evaluate and to extend trust to the authority that issued such certificates.

Our approach is to build a dynamic federation of CAs by automatically evaluating their certificate policies (automatic cross-certification) [30]. In a grid environment, the ability to perform automatically the extended path validation, makes it possible to obtain identity federation, since it is possible to identify users and to validate credentials from any domain.

As shown in Fig. 4, the Interoperability System (IS) 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.

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 primary tasks: (i) on-line validation of the certificate status, to check for revocation or suspension, (ii) evaluation of the CA security level.

For the first task, we will use an OCSP Responder that is part of the Interoperability System. It is invoked by an OCSP client
encapsulated in the “Federated” Grid Service that is 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 the Globus Toolkit, and is named OGRO [31,32].

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 [30]. The REM approach is based on the formalization of a 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. The IS client will compare this level with its CA to decide whether to extend or not the trust to an incoming user request, thus authenticating a federated identity.

An implementation of a Interoperability System was proposed in [33]. It was 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 fully enable identity federation.

5. PerfCloud security infrastructure

PerfCloud relies on Globus Toolkit 4 (GT4) as grid middleware. So we customized and enriched the default Globus configurations in order to meet the security requirements described in the previous sections. As regards access control, GT4 offers a flexible but poor mechanism to enforce authentication, authorization and data confidentiality [34]. In particular, GT4 uses the concept of Security Descriptors as standard method for configuring the security requirements and policies of clients and services. The Security Descriptors are XML files which are deployed together with the services. They can be enforced to protect different resources, such as a Globus container or a service.

GT4 authentication is based on PKI and the adoption of X.509 digital certificates with basic path authentication. The standard GT4 implementation is not configurable, and it does not allow to supply additional parameters for the authentication procedure. However, the Security Descriptor (SD) makes it possible to specify the communication protocols that an authenticated user must adopt in order to access a resource. It is possible to specify four (non-exclusive) options:

- **None**, in which no secure communication is enforced;
- **Secure Message**, which provides per-message secure communication in accordance with the WS-Security specification [35];
- **Secure Conversation**, which provides a secure session in accordance with the WS-SecureConversation specification;
- **SecureTransport**, in which a secure communication channel is adopted (i.e., SSL/TSL transport protocols are used).

For example, if the SD of a service specifies the adoption of SecureTransport and SecureMessage, the only service invocations accepted will be those with a SOAP message encrypted according to the SecureMessage WS-security standard over a TLS channel. Everything else will be rejected.

As for authorization, by default GSI offers only simple mechanisms:

- **Mapfile**, in which no authorization is performed, except for the container-wide grid mapfile;
- **Embedded PDP**, in which it is possible to define a per-container, a per-service or per-resource mechanism handled by a Local Policy Decision Point (PDP).

However, GSI also offers a set of APIs to integrate an external PDP, as XACML [36] or shibboleth [37], to support more expressive authorization policies. Finally, the default security solutions offered by Globus do not meet all the security requirement we have outlined in the previous section.

To implement the PerfCloud security infrastructure, we first extended the authorization mechanism in order to support XACML and, consequently, role-based policies [38]. Then, we forced the Security Descriptor to adopt secure communication channels for all the cloud-related service and resources [16,18]. In the following there are two simple examples of XACML rules. The first allows only the owner to access his own resources (enforced thanks to the **Condition** tag in the XACML syntax). The latter allows any user with an administrator role to access a resource (enforced thanks to the role attribute of the **Subject** tag in the XACML syntax).

**EXAMPLE 1:**

```
<Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
    <SubjectAttributeDesignatorDataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"/>
  </Apply>
</Condition>
```

**EXAMPLE 2:**

```
<Subject>
  <SubjectMatchMatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
  <AttributeValueDataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"/>
</Subject>
```

To complete our proposal, we enriched this solution with the POIS Interoperability System [33]. This is based on policy evaluation and on the OCSP protocol to support the extended path validation and enable the federated approach [17]. In the next subsection, we will illustrate its architecture and its integration in PerfCloud.

5.1. Identity federation in PerfCloud

As already mentioned, we adopted a federation approach to solve the security problems due to the interoperability between virtual and physical resources. The leased VCs are not incorporated in the VO of the underlying grid, and do not use the same
certificates and CA. They instead belong to a federated grid; they have their own CA and their certificates are accepted through an extended path validation procedure and dynamically-verified certificates.

Fig. 5 shows the main system components that are invoked within the PerfCloud framework at the call of a service offered by a grid container that supports federation (Federated Grid Container). Interoperability is dynamically granted, as it is possible to perform in an automatic way the cross certification via the REM methodology. As already said, the REM methodology is able to quantitatively evaluate a Certificate Policy and it is useful to understand if a CA is trusted or not. Furthermore, the OCSP Responder is able to evaluate the certificate status of any certificate issued by any Certification Authority.

We wish to point out that it is up to the Grid Container to interact with the Interoperability System. It can perform such interaction if it has been federated and has a suitable client (e.g., OGRO [39]) that provides the following two enhancements to the GT4 basic path validation algorithm: (i) the 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 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, configuring images and services with or without federated facilities according to the customer request.

Once federated services are available, different scenarios are possible. They are strictly related to the relations between the different grids (both Virtual and Physical). In particular, the possibilities are:

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

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 Grid, his identity will be mapped to a specific (federated) role and the defined role-based access control policy will be enforced at resource level [16]. For example, a user from the Virtual Grid 1 that has a role of Cloud Administrator can access to specific resources of Virtual Grid 2 but, possibly, with a non-administrator role.

Fig. 6 shows the whole security infrastructure of PerfCloud.

In the following, we will describe in detail the second scenario, including the role mapping and the authorization issues.

5.2. Accessing federated resources: an example scenario

Fig. 7 shows a typical scenario where a cloud user belonging to the virtual cluster 1 (denoted user@VC1 from Virtual Grid 1) requests a grid service offered by the physical cluster (denoted FGC@Root within the Root Physical Grid). The interoperability is achieved because the Grid Service Container has been federated.

In order to simplify the description, we suppose that POIS can also work in an off-line manner: the Certificate Policies from any Virtual Grids have been previously submitted to the POIS, they have been evaluated by the Policy Evaluator subsystem and the corresponding GSL has been stored into OCSP Responder DB along with the CA data (this step is not shown in the diagram).

The steps of this scenario are:

1. When a cloud user from the Virtual Grid 1 (user@VC1) requests a grid service offered by the Physical Grid, its 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 maintains 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 the XACML Policy Decision Point (PDP) configured at Resource Level [27].

In Table 1, 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.
It is possible to note that if an user from the Virtual Grid 1 (denoted User@VC1) requests a service of the Physical Grid (denoted Root in the table), 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 Virtual Grid 1 (User@VC1) requests a service of the Virtual Grid 2 (denoted 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 (or a user from Virtual Grid 1) 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 as the requester is not federated but “internal” and so does not require the extended validation mechanism. The last case is when a user from the Physical Grid requests services offered by a Virtual Grid. 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 trust the security mechanisms 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 computing or even administration resources. We think that this is a very important issue in a cloud environment, as identity federation is an open issue that still limits the wide adoption of these infrastructures. On the down side, the main drawback of this approach is an inevitable performance penalty, as complex authorization mechanisms or secure channels may heavily increase the platform overhead. The next section will evaluate quantitatively their actual performance impact.

6. PerfCloud performance evaluation

The goal of the following analysis is to evaluate the overhead introduced by the multilayered cloudgrid architecture by performing measurements on our implementation (PerfCloud). We have developed a synthetic PerfCloud NULL service, which just sends a reply message when invoked. The response time measured in several working conditions (corresponding to the different security configurations discussed in the previous section) will provide insight on the delay introduced by the PerfCloud middleware, unrelated to the service invoked and to the actual tasks that in normal conditions are be performed on the target environment. In order to understand how the security mechanisms in PerfCloud and the cloudgrid architecture affect performance, we measured the response time of the above described target service on physical, virtual and federated virtual clusters. The comparison between the figures obtained will make it possible to ascertain if the use of a federated grid made up of both physical and virtual resources is a reasonable solution, or if the performance penalties incurred are too high.

In [18] we presented an analysis, focused only on the security mechanisms in physical and virtual clusters. This analysis is presented, slightly expanded, in Section 6.2. Leveraging on these results, we present unpublished work regarding the federated scenario in Section 6.3.

6.1. Testing environment

PerfCloud is currently deployed on the grid system shared between the University of Sannio (Powercost cluster) and the Second University of Naples (Vega cluster).

The Powercost cluster is a 40-nodes cluster made of Intel Xeon-based nodes with 2 GB RAM, interconnected through a Gigabit Ethernet (a Myrinet network also exists, but it was not used for the test). The O.S. is Rocks Cluster version 5.1; the Globus Toolkit used is version 4.0.8. The Vega cluster is a system with a very similar configuration. The presented results were measured on the Powercost cluster. Analogous results (not presented here for the sake of brevity) were obtained for the Vega cluster.

We set up a custom client that performs I invocations of the (NULL) service and, at the end of the test, records in a database the I response times, on which to perform the analysis. It should be explicitly pointed out that this measurement technique does not allow to test system behavior under real workloads, as it corresponds to an ideal condition, where only the target service uses the target resource. However, this solution has the advantage to be easily reproducible, and so to allows us to perform all the tests under exactly the same conditions.

All the tests were performed in a dedicated environment, where the only source of load is the NULL service client. To collect a statistically significant number of measurements, I was set to 50. Every I-invocations session was repeated three times, and we chose the session with the lowest standard deviation, discarding the remainder. For the chosen session the final output consisted of the mean, the standard deviation and the confidence interval for the experimental points.

In order to understand the performance behavior of the PerfCloud NULL service with different security mechanisms, we set up three testbeds as shown in Fig. 8. The PerfCloud client is directly connected on the physical LAN and the virtual clusters are instantiated only on the cluster nodes. This choice aims at reducing the effects of the network on the measured overhead. For further details, the interested reader is referred to the complete TR of the experiments performed, available at the URL http://goo.gl/3VFus.

6.2. Performance analysis for the non-federated scenario

This subsection presents the first set of experiments concerning the performance analysis of the security mechanisms in physical and virtual clusters. According to the well-known experimental methodology presented in [40], the steps for performing such an analysis are:

- **Experimental design.** In this phase a design of experiments (DOE) is realized such that the maximum information can be obtained with the minimum number of experiments. This will be described in Section 6.2.1.
- **Collection of experimental results,** in which the outcome of the experiments is gathered with associated basic statistical parameters (averages and variations). Our results are reported in Section 6.2.2.
- **Experimental analysis,** which is the object of Section 6.2.3, made up of two steps:
  - Model construction, in which a mathematical model is developed that best represents the data obtained.
  - Model validation, in which diagnostic tests (visual or mathematical) are performed to verify that the model does not contain errors or polarizations.

Section 6.2.2 presents detailed considerations about the insights gathered from the analysis.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Roles mapping.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requester</td>
<td>Req. resource</td>
</tr>
<tr>
<td>User@VC1</td>
<td>Root</td>
</tr>
<tr>
<td>User@VC1</td>
<td>VC2</td>
</tr>
<tr>
<td>User@Root</td>
<td>Root</td>
</tr>
<tr>
<td>User@VC1</td>
<td>VC1</td>
</tr>
<tr>
<td>User@Root</td>
<td>VC1</td>
</tr>
</tbody>
</table>
6.2.1. Design of experiments

To quantitatively evaluate such a complex system as the one described in the previous sections, we first isolated the main factors (security and architectural components or configurations) that contribute to its performance. Our analysis identified the following factors and their corresponding N levels:

- Resource (N = 2) can be Physical or Virtual
- Channel (transport) (N = 2) can be HTTP or HTTPS
- Auth (N = 3) can be None, Conversation or Message
- Authz (N = 3) can be None, MapFile or XACML.

Since the number of different factors and levels is tractable, we decided to perform a full factorial design of experiments [40], in which every possible combination at all levels of all factors is examined. This allows us to find the effect of every factor including the secondary factors and their interactions.

6.2.2. Experimental results

Table 2 summarizes the performance results obtained for all the possible configurations. Each cell in the table reports the mean value and the confidence interval (in square parenthesis) of the response time, in milliseconds. The figures in the table are “raw” data, which cannot be easily interpreted and discussed. Our considerations on the obtained results will be based on the model of the system presented in the following subsection.

6.2.3. Model evaluation

The next step of the analysis is the definition of a model taking in consideration the main effects of each factor (the contributions that come from a single factor). Then the interaction effects (the contributions that come from the mutual interaction of factors) are gradually taken in consideration, until the evaluated error (meant as the difference between the measured value and the predicted value) can be considered negligible. We derived the following prediction model that corresponds to the DOE used:

\[ y = \mu + R_i + C_j + Auth_k + AuthZ_h + Auth - C_{jk} + e_{ijkh} \]

where \( i = \text{Physical, Virtual} \), \( j = \text{HTTP, HTTPS} \), \( k = \text{None, Conversation, Message} \), \( h = \text{None, MapFile, XACML} \), and \( \mu \) is the global mean (i.e., the average of all values). \( R, C, Auth \) and \( AuthZ \) (respectively Resource, Channel transport protocol, Authentication security protocol and Authorization mechanism) are the independent factors that we decided to take into account in the model. They appear in the equation for \( y \), according to the factor-level under evaluation. We will consider all the possible configurations described in the previous section with \( R_{\text{physical}} \) and \( R_{\text{virtual}} \), \( C_{\text{HTTP}} \) or \( C_{\text{HTTPS}} \), and so on. These factors are all independent, and represent the main effects of the prediction model.

\( Auth - C \) is the contribution to the model that comes from the interaction between the authentication security protocol and the transport protocol, and represents the way in which such coupling affects the mean. So, we will evaluate the different combinations \( Auth - C_{\text{None,HTTP}}, Auth - C_{\text{Conversation,HTTP}} \) and so on. These factors are all independent, and represent the main effects of the prediction model.

\( Auth - R \) was computed, but not included in the model, because of their negligible impact. In fact, the model that includes
Table 2
Response times for the different configurations (ms).

<table>
<thead>
<tr>
<th>AuthZ</th>
<th>Transport None</th>
<th>Conversation Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical cluster</td>
<td></td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>HTTP 106 [102.61, 109.39]</td>
<td>370 [358.51, 381.49]</td>
</tr>
<tr>
<td></td>
<td>HTTPS 263 [256.50, 269.50]</td>
<td>853 [837.60, 868.40]</td>
</tr>
<tr>
<td></td>
<td>HTTPS 262 [255.94, 268.06]</td>
<td>859 [843.31, 874.69]</td>
</tr>
<tr>
<td>XACML</td>
<td>HTTP 108 [104.81, 111.19]</td>
<td>369 [358.87, 379.13]</td>
</tr>
<tr>
<td></td>
<td>HTTPS 266 [259.67, 272.33]</td>
<td>866 [850.20, 881.80]</td>
</tr>
<tr>
<td>Virtual cluster</td>
<td></td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>HTTP 85 [81.86, 88.14]</td>
<td>328 [317.06, 338.94]</td>
</tr>
<tr>
<td></td>
<td>HTTPS 238 [231.96, 244.04]</td>
<td>797 [780.52, 813.48]</td>
</tr>
<tr>
<td>MapFile</td>
<td>HTTP 86 [82.48, 89.52]</td>
<td>322 [311.42, 332.58]</td>
</tr>
<tr>
<td></td>
<td>HTTPS 239 [232.88, 245.12]</td>
<td>797 [780.35, 813.65]</td>
</tr>
<tr>
<td>XACML</td>
<td>HTTP 86 [82.88, 89.12]</td>
<td>321 [310.09, 331.91]</td>
</tr>
<tr>
<td></td>
<td>HTTPS 245 [238.58, 251.42]</td>
<td>799 [782.67, 815.33]</td>
</tr>
</tbody>
</table>

Table 3
ANOVA table.

<table>
<thead>
<tr>
<th>Component</th>
<th>Sum of squares</th>
<th>Variation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource (R)</td>
<td>927,068.06</td>
<td>0.91</td>
</tr>
<tr>
<td>Channel (C)</td>
<td>27,565,312.50</td>
<td>26.99</td>
</tr>
<tr>
<td>Authentication (Auth)</td>
<td>60,886,636.11</td>
<td>59.60</td>
</tr>
<tr>
<td>Authorization (AuthZ)</td>
<td>6,211.11</td>
<td>0.01</td>
</tr>
<tr>
<td>Auth − C interaction</td>
<td>12,606,925.00</td>
<td>12.34</td>
</tr>
<tr>
<td>Errors</td>
<td>149,862.50</td>
<td>0.15</td>
</tr>
</tbody>
</table>

only Auth − C explains for the 99.85% of the measured values, as shown in Table 3.

For the sake of brevity, we present below just the results of the analysis of variance (ANOVA) of the obtained model (Table 3). In this table, we have reported for every effect the sum of the squares of all values (second column) and the variation of y as it is explained by the different effects. For example, in the considered model the main effect channel explains the model for the 26.99% while the factor authentication (Auth) explains the model for the 59.60%.

The correctness of the prevision model is supported by the value of the error variation, which is under 1%. To confirm the absence of polarization in the results, we performed a visual analysis of the residuals, which, as clearly shown in Fig. 9, are uniformly distributed around zero and have homogeneous variance.

6.2.4. Performance considerations

The above model points out clearly the fundamental sources of overhead in a fully functional cloud grid solution without identity federation support. In this section, we present the considerations that can be drawn from the obtained results.

Virtualization overhead. The first result obtained, which was not completely unexpected, is the low impact of virtualization on overhead. In practice, there is no difference (in terms of security overhead) between a service hosted on a physical cluster and on a virtual one. In fact, the ratio of variation, explained by the factor R, is 0.91%. Moreover, looking at the results in Table 2, we can note that the invocation of a service on a virtual resource performs even better than the corresponding request on a physical one. This counterintuitive behavior is due to the fact that modern virtual engines introduce very small overhead (in our test, we used Xen 3, which declares less than 5% overhead). This can be compensated and sometimes overcome by the performance gain obtained by OS noise reduction. The virtual images, being targeted to a specific appliance, can exclude generic OS management services, which are instead needed in a general-purpose physical environment. It should noted, however, that we have implemented a service that does not stress so much the system and the virtualization environment. For other kind of services, e.g., bandwidth-intensive ones, virtualization could become a bottleneck.

Authorization mechanisms. The introduction of advanced authorization mechanisms as XACML, needed to deal with the different roles of the users, surprisingly does not have a great effect on the overall system performance. The AuthZ factor simply does not affect the overhead, as shown by the ANOVA Table (less than 0.01% of variation explained). This can be explained by considering that even the basic GSI authorization mechanism needs to instantiate Java objects to take an authorization decision. As a result, the time required to invoke external authorization services is dwarfed by the time necessary to load the chain of GSI objects.
Role of channel and authentication systems. Unlike the previous factors, the channel transport and the authentication security protocol affect heavily system performance, both on virtual and physical resources. The model analysis shows that the introduced overhead depends on two factors: the adoption of security protocols at transport layer, i.e., the choice between HTTP and HTTPS, and the security protocols adopted at message layer, i.e., the cryptography applied to the XML-based SOAP messages. Both factors have a great impact (with a variation on the model of about 30% and 60%, respectively). Also their interaction has a high impact (about 10%).

Considered that the measured overhead shows a minimum value of 85 ms and a maximum value of 866 ms (a value about 10 times higher), the parameters set up in the Security Descriptors should be performed with great care. From a security point of view, we can note that security at transport and security at message layer are independent of one another. Both of them aim at granting confidentiality and integrity of information, and they can reach the same result in different ways. The secure transport layer (HTTPS) has a lower impact on performance and, in terms of security, is considered equivalent to the message layer one. So the best solution is to disable completely the message layer security. When this is not possible, it should be avoided to enable both the security layers.

Some interesting considerations can also be done on the interaction of the two factors. Analyzing the result table (Table 2), it is possible to note that, enabling the transport layer, the performance of Secure Conversation (which offers a lower security level) becomes worse than the performance offered by the Secure Message approach. So, if for any reason both transport and message security layers must be enabled, the lowest overhead can be obtained by enabling the Secure Message security protocol, which also offers the highest security level.

6.3. Performance analysis for the federated scenario

This section presents the analysis of the overhead due to the use of federated authentication at service invocation. As explained in Section 4, the federation mechanism is commonly built up by extending some existing authentication mechanism (as HTTPS, Secure Message or Secure Conversation) with augmented path validation. In the previous section, we have presented results that show that the most effective combinations of security and performance levels for the authentication layer are those based on the HTTPS protocol or the SecureMessage over HTTP protocol. So it is a good choice to extend one of these with federative controls. In order to simplify the analysis, we chose the latter configuration (SecureMessage over HTTP) for our tests on the federated scenario. This allows us to instantiate a simple HTTP container and to impose security overheads only on the services that need the security infrastructure. In designing these experiments, another choice was to allocate the Interoperability System in a dedicated virtual machine, not belonging to any Grid, to reflect the more general case in which its tasks are performed by a Trusted Third Party.

6.3.1. Experimental results

Table 4 summarizes the performance results obtained for the experiments involving the use of identity federation. As in the previous case, each cell in the table reports the mean value and the confidence interval (in square parenthesis) of the response time, in milliseconds.

Table 5 compares the performance obtained in the presence of federation with that of the corresponding configuration from the previous section, pointing out the slowdown (meant as the ratio between the highest and the lowest response time) and the corresponding percent overhead due to the need of extra messages for completing the authentication process.

6.3.2. The impact of identity federation

From a performance point of view, the introduction of identity federation poses new challenges. It is easy to observe from Table 4 that in the presence of federation the authorization mechanisms continue to play a small role in determining the response time. However, the identity federation system or, in general, a trusted Interoperability System has a significant impact on performance, even greater than the one due to authentication protocols and authorization mechanisms. This result is of interest for both the System Administrator and the Cloud Administrator, since our toolkit offers them the possibility to manage their resources and to obtain the performance they need. The analysis tells us that, whenever a Cloud Administrator is trusted, a System Administrator should prefer to grant him the rights on a Certification Authority hierarchically related to the one of the Physical Grid, avoiding the use of identity federation. This has the cost of widening the number of actors which can grant rights on the global infrastructure, which includes the physical one, and possibly of weakening security. On the other hand, when the Cloud Administrator is not trusted, the System Administrator should prefer to maintain the rights on the global Certification Authority and to provide virtual clusters with federated facilities to enable cooperation. From the Cloud Administrator perspective, the proposed methodology offers the possibility to obtain a quantitative measure of overhead in each possible configuration, making it possible to choose the preferred solution on the basis of his own security and performance requirements.

7. Related work

A detailed comparison of the cloud and grid computing models and of the security architectures proposed for the cloud is out of the scope of this paper. In the following, we will focus on scientific
work that has addressed the problem of cloud and grid integration, also presenting briefly several cloud security solutions that address the identity federation problem.

The roots of cloud-on-grid solutions can be traced down to research preceding in time the cloud model diffusion, and targeted to the use of virtualization in grid or cluster systems [41]. It is worth mentioning the Virtual Workspaces (VW) project [42], the In-VIGO system [43] and Duke University Cluster-on-Demand (CoD) [44]. The VW project evolved from the Nimbus Cloud Toolkit [13] and is still active, whereas In-VIGO is now closed, even if the software developed is still used inside the research group. PerfCloud, Nimbus and In-VIGO can all be regarded as cloud-on-grid systems, in that they offer cloud-like services on the top of an existing grid. However, it is worth pointing out the differences among them.

PerfCloud, as described in Section 2.1, works on the top of an existing VM engine (as Xen [10], VMWare [11], VirtualBox [45]) and schedulers/resource managers (as OpenNebula [23] or CLEVER [46]) and provides a service-oriented interface to its users, who can be both final users or administrators. Moreover, it provides performance management services (e.g., services for performance prediction and autobenchmarking).

Nimbus [13] is the most well-known cloud-on-grid solution. It is based on the Virtual Workspace resource abstraction [42] and its underlying software was adopted in many other cloud-on-grid projects as Wispy [14]. Nimbus provides a set of grid services that implement a fully-featured virtual machine scheduler and management system, which supports the principal virtual engines (Xen, VMWare, KVM). Moreover, Nimbus offers also an Amazon EC2-compatible interface.

The In-VIGO [43] project anticipated many of the ideas and solutions developed successively in the research on cloud computing. The In-VIGO system builds a set of services for VM management and offers services to automatically install user software on the VM. It uses the GSI as basis for the security infrastructure, enriching the authorization mechanism with an RBAC-based component.

Another interesting cloud-on-grid precursor is the Cluster-On-Demand system at Duke University [47], which is a composition of two different projects: the CoD system itself, which offers the Cluster-on-Demand functionalities (much alike a Cloud IaaS system) and Shirako [48], which is a Java-based resource manager.

All these solutions are motivated by the need to offer easily-customizable environments to grid users. In fact, very often scientific software and libraries suffer from dependences on legacy software, and the possibility to lease a fully administrable and configurable cluster is indeed a big advantage.

Once the cloud paradigm imposed itself (thanks to the diffusion of Amazon EC2 and the blast of other cloud providers), the grid-on-cloud approach turned out to be one of the most attractive solution for acquiring huge computational (and storage) power. The rationale is clear: to add dynamically nodes leased by cloud providers to existing grid infrastructures. There are many different examples of this solution [49–54], whose main advantages are the ease to scale up and to meet the changing requirements of applications (by acquiring new computational resources as needed), the flexibility of the business model (moving the capital expenditure of acquiring a full-fledged data center to the operational expenditure of renting only the machines needed at a certain moment in time), and the assurance of a certain level of delivered performance. The last point was the main limitation of the first generation of cloud systems, which showed that cloud-based VCs were characterized by lower performance than the equivalent physical systems, mainly because of the poor cloud internal interconnect [55]. It is worth noting that recently Amazon EC2 has started providing a commercial HPC service, offering cluster-compute instances, which are granted to be connected through a (relatively) high performance network.

In order to compare the cloud-on-grid, grid-on-cloud and cloudgrid approaches we can consider for the three paradigms the following properties:

- **Reuse**: defined as the capability of using already-available resources (as the incredible number of academic clusters in actual Grids).
- **Software independence**: defined as the capability of developing services and applications using any kind of environment.
- **Extensibility**: defined as the capability of adding dynamically new resources to acquired ones.
- **Interoperability**: defined as the capability of offering the developed application as service to others and/or use services offered from other systems.

As shown in Table 6, the cloud-on-grid solutions, as Nimbus or In-VIGO, usually grant Reuse, Software Independence and Extensibility, but cannot provide Interoperability. On the other hand, grid-on-cloud solutions, as the ones cited above, grant Software Independence, Extensibility and Interoperability, but cannot provide Reuse. Only the cloudgrid approach can provide all the desirable properties at the same time. Furthermore, to the best of the authors’ knowledge, no grid-on-cloud solution offers proper security solutions (as the PerfCloud federation approach) to face the security issues that arise in these systems.

Even if security is a hot topic in cloud computing, the literature studying the security problems linked to the integration of cloud nodes into an existing infrastructure is not so wide. On the other hand, in many commercial cloud solutions great effort has been put in more specific security issues, as network protection against malicious attacks, data integrity, access control, disaster recovery strategies and security organizational aspects. Available cloud solutions offer a large set of different solutions for access control. For example Google, which is essentially a PaaS provider, adopts a number of industry-standard mechanisms to authenticate users (including LDAP, Active Directory-based authentication and single sign-on systems (SSO)). It requires the use of a unique User ID (based on standards like OpenID [56] or OAUTH [57]). Besides being used to identify the activity of each person on the Google network, this ID is used to control access to every system at Google. As for authorization, access rights are based on user’s role. A fine grain access control can be enforced by a central authorization service compliant to SAML standards (this solves interoperability among authorization services), or by existing access control policies (i.e. already in place in just-added systems).

Amazon enforces access control through a centralized Identity and Access Management (IAM) system that lets administrators to manage users, groups of users, and access permissions for services and resources. The authentication protocols are mainly based on the adoption of Public Key Certificates. On the other hand, the cloud platforms that integrate grid infrastructures mainly enforce security mechanisms that are provided by the underlying security system (i.e., grid CSI). As a matter of fact, in these architectures, public key infrastructures are configured and used in different ways to provide security features (e.g., user certificates, grid proxy certificates, encryption/decryption functions, secure protocols, etc.).

<table>
<thead>
<tr>
<th>Approach</th>
<th>Reuse</th>
<th>Software Independence</th>
<th>Extensibility</th>
<th>Interoperability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid-on-cloud</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Cloud-on-grid</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Cloudgrid</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 6: Comparison of different approaches for cloud and grid integration.
8. Conclusions and future work

Currently there is great interest in the integration of grid and cloud computing. In this paper, we have proposed a novel computing model taking the best of both worlds. Unfortunately, our cloudgrid has security and performance issues. The security issues have been pointed out, showing that they can be tackled from two different perspectives. On the one hand, we outlined the need to protect physical and virtual resources, due to the lack of sophisticated authentication and authorization security mechanisms in the underlying grid infrastructure. At this aim, we proposed a role-based access control mechanism and an extension of the grid security infrastructure (GSI).

On the other hand, as the cloudgrid “contains” a cloud environment, we need to let virtual resources cooperate and interoperate with potentially untrusted domains. This last problem is referred to in the literature as identity federation. The identity federation approach proposed here is based on the definition of an Interoperability System that validates certificates by performing dynamically a Cross Certification among untrusted Certification Authorities and by evaluating on-line the status of a digital certificate.

In our previous papers, we outlined the importance to understand if and how the introduction of such mechanisms would affect the overall system performance, along with the adoption of the three-layer cloudgrid architecture. We showed through extensive experimentation that, in contrast with intuition, the cloudgrid approach does not introduce a significant overhead, even if it entails the use of virtualization and requires complex authorization mechanisms. On the contrary, the introduction of an Interoperability System imposes heavy performance penalties. On the basis of these observations, we presented a trade-off analysis between performance and security, whose final results shed light on how the factors that affect the service performance quality (namely, the message and transport level authentication protocols, the authorization mechanism and the adoption of a federated approach) should be tuned in order to optimize the system while maintaining suitable security warranties. We think that this quantitative analysis makes it possible for the cloud to offer guarantees on the quality of service and to negotiate Service Level Agreements in an automatic way.

As regards the future evolutions of our work, we intend to integrate the Interoperability System with authorization services of widely adopted architectures (as PERMIS or ShibGrid). At the same time, we wish to move toward emerging standards for managing trust (WS-trust, WS-federation). This will require a new evaluation of the trade-off between the security level provided and the overall system performance.

Further work is also necessary to ensure that the introduction of interoperability among different open cloud infrastructures will not affect the global security and the quality of the provided services. To enable the full deployment of the proposed infrastructure in a public environment, it is necessary to solve not only access control and identity federation security problems, but also issues linked to asset classification, privacy and data integrity, disaster recovery, business continuity, and last but not least, personnel and organizational security. Finally, we intend to define quantitative benchmarks to compare different cloud solutions. This will make it possible to enrich our trade-off analysis, and to provide useful guidelines for designing and deploying cloud infrastructures.

References

Valentina Casola is an assistant professor at the University of Naples Federico II. She received the Laurea degree in Electronic Engineering cum laude from the University of Naples in 2001 and she received her Ph.D. in Electronic and Computer Engineering from the Second University of Naples in 2004. Her research activities are both theoretical and experimental and are focused on security methodologies to design and evaluate distributed and secure infrastructures.

Antonio Cuomo is a Ph.D. student at the University of Sannio, Benevento, Italy. His research activities focus on performance evaluation and prediction of parallel and distributed computer systems, including Cloud Computing, GRID Computing and their integration. He is currently taking part in an Italian project concerning the development of a volunteer-based cloud computing platform with an emphasis on performance management. He received the laurea specialistica degree cum laude in Computer Science Engineering from the University of Sannio in 2009.

Massimiliano Rak is an assistant professor at Second University of Naples. He received his degree in Computer Science Engineering at the University of Naples Federico II in 1999. In November 2002 he received Ph.D. in Electronic and Computer Engineering at Second University of Naples. His scientific activity is mainly focused on the analysis and design of High Performance System Architectures and on methodologies and techniques for Distributed Software Development.

Umberto Villano is full professor at the University of Sannio, Benevento, Italy, where he is Director of the Department of Engineering. His major research interests concern performance prediction and analysis of parallel and distributed computer architectures, tools and environments for parallel programming and distributed algorithms. He received the Laurea degree in Electronic Engineering cum laude from the University of Naples in 1983.