FCFA: a Semantic-based Federated Cloud Framework Architecture

Giuliano Manno1, Waleed W. Smart2, Luca Spalazzi3

1Oculus AI Technologies AB
Stockholm, Sweden
giuliano@oculusai.com

2 Ball Aerospace &Technologies Corp.
Fairborn, OH, USA
smart@arys.org

3Università Politecnica delle Marche
Ancona, Italy
spalazzi@diiga.univpm.it

Abstract—Cloud Computing is a paradigm that applies a service model on infrastructures, platforms and software. In the last few years, this new idea has been showing its potentials and how, in the long run, it will affect Information Technology and the act of interfacing to computation and storage. This article proposes an architecture for an ontology-based resource life-cycle management and provisioning in a federated Cloud Computing environment. Federated Clouds are presumably the first step toward a Cloud 2.0 scenario when different providers will be able to share their assets in order to create a free and open Cloud Computing marketplace. The idea of creating a network of Clouds has addressed a number of issues that in the last two years have motivated new research efforts. The contribution of this article is an innovative redesign of a Cloud Computing architecture from the ground-up, leveraging semantic web technologies and natively supporting a federated resource provisioning.

Index Terms—Cloud Computing, Federated Clouds, IaaS, Federated Infrastructure, Cloud Architectures, Semantic Web Technologies

1. INTRODUCTION

Cloud Computing [1] enables the idea of utility on hardware and software so that computation, storage, as well as software components and high-level services can be outsourced and used on-demand by the end users. On top of the infrastructure, the utility paradigm is supported by a set of technologies that can exploit the underlying resources through a service-oriented approach. Key players in the public Cloud industry, such as Google AppEngine [2], Amazon EC2 [3], Rackspace [4], Terremark [5], IBM SmartCloud [6] and Joyent Smart Datacenter [7], have implemented solutions that are mainly focused on centralization and localization. At the present time, they do not support an elastic scale-out to avoid resource demands outside the limits of the datacenters. Such limitation can be overcome by means of the notion of cloud federation. Cloud federation can be described through the concept of community, where each Cloud is a different and independent domain. The community is a collection of interacting Clouds whose purpose is to fulfil a specified objective. This objective is defined by a contract that describes the economic and technical nature of the commitment along with the policies that define the constraints of the interaction. Each domain is independent. It retains autonomy over its shareable resources and is autonomous with respect to entering or leaving the federation. The controlling service involves several components that manage domain naming, identification, authorization and trust, access control, and enables search and retrieval services. A contract establishes the agreement between the parties, and carries with it the obligations of each domain, the duration of the agreement, and possibly an agreement about the resolution of failures or other types of exceptions.

This article presents a semantic-based architecture for federated cloud computing in order to address certain challenges of interoperability among heterogeneous and autonomous clouds. The article is organized as follows. First of all, the initial applicative domain will be presented in Section 2 alongside with a comprehensive description of the research background in Section 3. Sections 4 and 5 will describe the use of semantics as the foundational element of the proposed architecture, thus outlining how resource description and consistency have been addressed. Section 6 describes the research direction, the implementation currently in process, and the next tasks.

2. MOTIVATING SCENARIO

Starting from a general definition of federation [8], cloud federation could be defined as: “Multiple geographically distributed subsets of resources, organized in an independent manner, belonging to the same or different organizations, managed by a dynamic chain of authority established between various levels of responsibility.” The first scenario that drove the initial design of the proposed architecture conceived the creation of a high-level service deployment methodology to support different applications as in the case of Emergency Response Systems (ERS). An Emergency Response System [9-11] is typically based on the analysis of data generated by sensors that are geographically distributed in specific places. Therefore, a Federated Cloud could be employed to perform a deployment of high-level services, run by the independent organization that retains the ownership of the sensors and the sensed data. As depicted in Figure 1, a possible multi-organization structure is formed by three independent organizations that are geographically distributed in three macro-areas. Every area is covered by a set of offices that have an independent IT department with a certain amount of resources provisioned in a cloud fashion. Every organization has one headquarter that...
manages the scaling out of resources from one cloud to another. The headquarter is also responsible for the information shared with other organizations and also the provisioning of cloud resources outside its hierarchical structure. The main advantage of this scenario is the possibility to tackle new situations and provide a fast response running ad-hoc virtual machines that can operate the services taken from a common repository. The management of services can be achieved through the federation, by executing proper commands to single or general areas. The management also employs the same resources for large scale and distributed data analysis [12-13]. The results of the analysis usually are under the responsibility of the organization that decides the filtering and publication policies. The data will be shared with other organizations through web mesh-ups.

3. RELATED WORK

The subjects discussed in this paper have been active areas of research for some time. In this Section, we review some of the main ideas and solutions proposed in the literature, especially related to clouds, federation, and the use of web semantics.

3.1. Cloud Federation

In recent years, several groups of researchers have started initiatives to analyse and solve problems that involve one or more clouds working together on a certain operational level to deliver promised outcomes. The Reservoir Project [14] is probably the first federated cloud computing research initiative that has addressed the challenges of the infrastructure integration (IaaS) between nodes in a geographically distributed Cloud Computing environment. The Reservoir project has proposed an architecture for the dynamic partnership between infrastructure providers in order to create a seemingly infinite pool of IT resources, preserving the business and technological autonomy of involved parties. The approach of the Reservoir project points out the lack of interoperability between Clouds and its consequences. The design of such an architecture needs a Service Level Agreement (SLA) centric approach. The goal of the Reservoir project is the creation of a federated cloud computing system that focuses on the clear separation between the service provider and infrastructure provider. In order to exchange the knowledge of a computational unit, called VEE (Virtual Execution Environment), a Service Manifest is shared between the two or more sites. A service manifest will be one of the key elements that permit the interoperability between Reservoir sites. The work also describes the VEE and all its components and extends an industry standard, the Open Virtualization Format (OVF), used for the description of virtual machines. Regarding monitoring, which is a crucial activity in cloud computing management, the same authors have created another project called Lattice [15], a monitoring framework designed around the concept of producers and consumers of monitored data, generated by probes or a group of probes that are tightly coupled with their point of control, thus having the ability to migrate.

Other publications have covered specific issues related to the Reservoir architecture. Larsson et al. [16] propose a scheduling methodology for virtual machine migration. Their approach aims to minimize SLA exceptions and maximize performance and resource usage through the use of a heuristic method that can take fine grained descriptors into account. The authors also describe a semantic-based monitoring facility that collects data from the pre-existent monitors and creates metadata annotations. This methodology gives the possibility to search in a semantic database and obtain detailed structured information for accounting and exception handling.
Another interesting federation project is the so-called Inter-Cloud. Its authors, Berstein et al. [17], highlight goals and challenges that have to be addressed, alongside with a set of suitable open protocols. They, furthermore, propose an architecture based on three main components. The first is the Inter-Cloud Root, a set of physically redundant systems that enables naming, discovery, brokering in the Cloud network, managing the involved resources through a Resource Catalogue. The second entity is the Inter-Cloud Exchange, which enables the communication between Clouds. The third piece is a set of Inter-Cloud Gateways that, similarly to Internet Gateways, support all the technologies for the Inter-Cloud low-level communication. In another set of articles [18-20], researchers describe an effective network viewpoint involving the use of an open peer-to-peer communication protocol named XMPP [21] as the foundation of a lightweight service framework alternative to SOAP [22] or ReST [23] that are based on synchronous communication, inefficient for data intensive, and time consuming tasks.

Additional works that studied specific aspects of cloud federation were taken into account during the design of the proposed architecture. In particular, how the problem of static addressing is solvable through the adoption of a location independent technology called Networking of Information (NetInf) [24]. Unlike URLs that are location-dependent, Net-inf implements an abstraction layer that can assign a name to an object with a location-independent model, exposing an API that hides the dynamics of the physical locations and network topologies.

Other important publications worth mentioning are those that address the design of an architecture specifically regarding the semantic viewpoint [25]. The use of ontologies as an abstraction layer on top of the physical and virtualized resources has been proved to be a breakthrough for an interoperability-centric cloud design. Furthermore, semantic technologies are nowadays ready for production systems that require reliability and performance for defining, collecting, and querying resource descriptions and monitored data.

3.2. Semantics

In a cloud computing environment, in spite of all the advantages mentioned already, the interoperability among heterogeneous resources could be difficult or even impossible when there are different schemas and different names for the same entities. It should be noted that Berners-Lee et al. proposed the Semantic Web when they faced similar problem for the Web [26]. Nevertheless, most of the work related to semantics and cloud computing focused on how semantics can benefit from cloud computing [27-28]. Only recently, it was proposed to exploit semantics for cloud computing [16]. This paper presents a federated cloud architecture based on semantics in order to integrate heterogeneous resources deployed as services in different clouds.

The semantic web relies on the notion of concept-based knowledge base. A knowledge base is formed by two components [29]: an ontology (also called Terminological Box or TBox) and a set of facts (also called Assertional Box or ABox). An ontology (TBox) is a structured collection of concepts, relations between concepts, and a set of inference rules [30]. Even if there are several slightly different definitions of the notion of ontology, most authors agree that an ontology should be defined in a formal language. Well known languages for ontologies are OWL [31] (based on the Description Logics SHIOQ(D)[32]), RDF [33], and Topic Maps [34]. An ABox is a set of individuals and relation between individuals. Indeed, each concept can be thought of as a set of individuals.

4. THE PROPOSED ARCHITECTURE: AN OVERVIEW

The architecture proposed in this article has been designed with heterogeneity in mind. In fact, the ontological description of the system allows for platform independence. Through the instantiation of concept individuals, it is possible to represent different contexts, different types of resources and related commands belonging to different virtualization platforms, as will be demonstrated.

4.1. An Holistic View

The process of knowledge engineering applied to this complex environment divides the whole set of actors in five nested sub-domains. The bird’s-eye view of the overall system is depicted in Figure 2. The central activity of a Cloud infrastructure takes place in the hardware node (HNode), which is the rack-mounted server that runs the hypervisor. An entity linked to the hardware node is the virtual node (VNode), which represents the virtual machine executed by the hypervisor in the hardware node. The cluster of hardware nodes is managed at the Datacenter level by an entity called cloud controller. The concept of Cloud federation is established in the upper layer of the hierarchy and it is managed by the federation controller. It is important to emphasize the independence of each Cloud provider, which can act autonomously and is responsible of its resources. The federation is established when more than one datacenter leases a set of resources belonging to one or more virtual nodes. The overlay network and the whole set of resources that create the federation are identified by the term “Super Cloud”, since it is a Cloud of Clouds. The federation is established through an ex-ante agreement by the involved datacenters, which can receive the federation requests by another datacenter. The whole federation process is carried out by the federation controller, which discovers the available resources and defines a contract between the datacenters. The upper scope is the network of Super Clouds, which can be called Meta Cloud. The Meta-
Cloud resembles the hierarchy of the Internet, which is a network of networks, representing the connection between Federated Clouds. The overlay network is established through a set of services that connect together the services that belong to each scope. Each scope represents also an area of responsibility, segmenting the network between the sub-components.

### 4.2. The Semantic View

The management of a heterogeneous environment creates several issues. In order to manage them, the information about the cloud federation has been segmented in four ontologies as depicted in Figure 2. The characteristics of each node, such as resources and their Quality-of-Service (QoS), are represented by the HNode Ontology. The Cloud Ontology, managed by the Cloud Controller, has knowledge about the instantiated hardware and virtual nodes, enabling to each leased resource a pricing model. The application of the pay-per-use model is a role carried out by the datacenter, not the hardware node controller, which has to act only on the instances of its physical and virtual resources. The Federation Ontology, managed by the Federation Controller, contains the information about the established community of Clouds. Another clear division between responsibility areas, as depicted in Figure 3, is established between the TBox (terminological component) and ABox (assertion component) of any ontology, since the instances of every concept can be shared with an external entity only through accepted commands in a trusted environment. Following a bottom-up paradigm, the overall ontology can be considered as a basic set of concepts defined in the HNode schema and extended in the higher levels. The HNode Ontology deals with key characteristics such as the state of the resource and its description. This issue is connected with the fact that a resource provisioning needs the definition of an abstraction layer able to describe the physical resources and specifically in a Cloud environment, also able to describe subsets of virtualized resources that can be temporary leased to users. Taking into account knowledge engineering methodologies, it is possible to

---

**Figure 2**: Holistic View of the Entities Organized in the 5 Scopes

**Figure 3**: Semantic Intersection between Domains
define a general model that is able to capture the features of Cloud resources, not only on the infrastructure level, but also on the platform and software ones. The description of the resources can be created using ontologies placed in every Control Service Layer of the Cloud-stack [35] as shown Figure 4. The ontologies have been implemented using the Web Ontology Language (OWL).

The Figure depicts also a methodology that maintains the consistency between the actual state of the resources and their ontological descriptions. The act of keeping the ontology consistent is based on the command applied on the single resource whose state is updated with the command response provided by the virtualization system.

The essential resource description contains four basic concepts, as shown in Figure 5. The first concept is called Node and represents the basic entity of the infrastructure. This concept will be placed at the top of a taxonomy that describes other types of nodes as its sub-concepts. The second concept is the most important one and represents the notion of Resource, which permits a general high-level description that can capture the characteristics of the infrastructure. The third concept represents the notion of Command, which is the logic action that has the resource as its target. In the complete ontology, the Command concept will be part of a multilevel hierarchy, where the instantiation of the sub-concepts will permit the system to keep track of the commands that have been sent to the resource.

In order to apply a pay-per-use business model, the system must be metric measurable. QoSMetric is a concept and the instantiation of its sub-concept will capture the metrics of any type of resource. Four object properties will create relationships between these concepts. Since a node exposes a set of resources that can be manipulated, the two concepts will be connected with the “hasResource” property. Any resource may have a command that manipulates its state. Therefore the “hasCommand” object property will connect the two concepts. Regarding the pay-per-use model, the metric concept will be the object of two properties: the first one from the resource (hasMetric) and the second one from the node (hasMetrics).

5. THE PROPOSED ARCHITECTURE: A DETAILED VIEW

Next, we describe in details each of the five nested sub-domains.

5.1. Scope 1: Virtual Nodes (VNode)

The first domain is created by a set of virtual resources abstracted from the hardware resources. Figure 6 describes in synthesis how the ontology represents the characteristics of a virtual node. The concept VNode is a concept of the HNode Ontology (i.e., the namespace NODE). The conceptual connection between the node and the resources is established by an object property called “hasVirtualNode”, whose domain is the VNode itself.

---

**Figure 4:** Control Service and the Use of the Node Ontology

**Figure 5:** Fragment of the Node Ontology for the Atomic Resource Representation

---
and the range is a set of concepts that define the basic resources of the infrastructure. These resources are: vCPU, vMemory, vStorage and vNetwork. Each resource has a set of datatype properties that provide their quantitative description. When the HNode controller receives a Virtual Node instantiation command, its internal logic will extract the proper command from the ontology and send it to the hypervisor. If the command successfully instantiates the virtual machine, the internal logic will create an instance of the VNode concept with the requirements sent by the Datacenter Controller. The same instance code will be the payload of the response message.

that the HNode concept is a sub-concept of the Node concept. The VNode concept, described in the previous Section, is ontologically a sub-concept of the HNode, but in this case, for pragmatic reasons, is related to it with an object property (hasVNode). The HNode ontology contains the main part of the information that describes a Cloud infrastructure since all the virtual nodes will be included in it. Viewing the concepts of the node namespace from left to right, it’s possible to identify the description of the resources, which is similar regarding their physical and virtual representations. The concepts that describe the physical and virtual resources are respectively the sub-concepts of PhysicalResource and VirtualResource. The subconcepts of VirtualResource are also sub-concept of the single PhysicalResources, the multiple inheritance keeps the ontology consistent since the inherited concepts are siblings. Another important aspect, captured by the ontology, is the concept of Command, depicted on the right side of the Figure. The Command concept has two datatype properties that identify its type and the actual CLI terminal command. The Hnode and VNode concepts are connected to the concept of command, since the HNode itself has a set of physical resources that can be manipulated. An instance of the Command concept can refer, for example, to the instantiation of a new virtual machine. The Datacenter Controller sends the command type to the proper HNode, which will execute it, extracting the string from the Terminal datatype. The use of another hypervisor will not need any re-implementation but only the proper instantiation of the Command concept during the setup process. Every executed command can be monitored through the instantiation of the Monitor concept. The Monitor concept is a generic container that has two datatypes, which keep track of

5.2. Scope 2: Hardware Nodes (HNode)

The Hardware Node is the key infrastructural component of a Cloud datacenter. As stated before, the scalability exploited by the cloud computing paradigm permits the use of the same types of resources. Thus, the capability of the overall system is raised through the instantiation of further physical resources, instead of raising vertically the capability of the pre-existing ones. The HNode concept represents a single physical rack-mounted server that is installed in the datacenter. A datacenter can have a large number of hardware nodes, usually with the same hardware configuration. Each hardware node runs an operating system that is able, at kernel level, to instantiate and run virtual subsets of its physical resources, managed by a virtual operating system on a virtual machine. The virtualization monitor (hypervisor) is responsible for the management and monitoring of the virtual machines and their virtual resources. Figure 6 shows the model of the HNode ontology on the right, surrounded by the dotted frame. Every coloured frame describes a different namespace. The Figure shows

![Figure 6 : A Simplified Version of the Overall Ontology](image-url)
5.3. Scope 3: Clouds/Datacenters

The datacenter is the enabler of the Cloud in terms of service provisioning, accounting and business model. Every datacenter manages the resource it leases and keeps track of the business activity in the Cloud, represented by service contracts. In the proposed architecture all the activities are managed by a component called Datacenter Controller. From the infrastructural perspective, a datacenter can be described as the entire set of physical resources (HNodes) and virtual resources (VNodes). The virtual resources, exposed “as a service,” represent the core business of the Cloud and the datacenter is the central coordination entity that has access to the internal overlay network and can send commands to the HNodes. The infrastructure of a datacenter is physically characterized by a variable number of cabinets in which the hardware nodes are rack mounted. Inside the datacenters, the physical resources of the hardware nodes are usually the same, primarily to take advantage of the economies of scale in capital expenditure, and second because the same hardware can be easily replaced, reducing the lead-time of substituting hardware parts and the mean time to repair (MTTR).

The semantic viewpoint of the datacenter includes the previous descriptions of the hardware and virtual nodes, whose schema definition represents them as physical subsystems of the datacenter. Referring to Figure 6 again, the HNode schema is referenced in the datacenter ontology through its namespace “hnode:” and URI. The ontology extends the HNode description with further concepts that belong to the responsibility area of the datacenter, related to its specific namespace (“datacenter:”) and URI. The first concept is the Datacenter, which, similarly to the Node concept in the HNode, represents the highest descriptive level of the considered domain. The second fundamental concept is the Contract, whose instantiation represents the establishment of a resource leasing. The Contract concept is related to the resources in two different ways, the first one is through the concept of Bundle, which describes the template of the various Cloud offerings. The Bundle concept is connected with the virtual resources with an object property called “hasBundleResource” and its datatype properties describe the name and the price of each offering. The instantiation of the virtual resources can be done starting from the templates or other specific requirements.

The second relationship between the contract and the virtual resources is made through the VNode concept and the “leasesNode” object property. Each datacenter has a list of the active users that have one or more instantiated contracts; the User concept is related to the Datacenter concept itself with the “hasUser” object property and with the Contract concept through the “hasContract” property. When a contract has been established, the user can authenticate and have access to the User Control Panel (UCP), a web interface that shows a dashboard with the state of the resources related to the contracts, allowing the user to manage them. The users do not have direct access to the overlay network for security reasons. The Datacenter Controller will forward the user’s requests after the authentication. Especially in case of Infrastructure-as-a-Service, the leased resources have their Internet endpoints that can be reached by the user, for example, when a Virtual Machine (VNode) is instantiated and exposes an SSH server for the direct management. Every task that is performed on the resources and the contracts will be stored in the ontology in order to keep track of the user’s activities. This mechanism is established through the concept of command, which can be applied not only to the leased resources, but also to the contracts and the nodes. The contract itself can be seen as a resource that accepts commands. The act of executing a command on the contract will change its state. For instance, when the user will need to know how much he is spending, the list of the instantiated commands on his contracts will provide a correct billing information. The Cloud Controller (CC) is not the endpoint of the overlay network. In fact, it exposes a service that interfaces with the Federation Controller and of course with the other datacenters that belong to the Federated Network.

5.4. Scope 4: Super Cloud, Federated Clouds

In the proposed architecture, the Federated Cloud is the entity that enables the Cloud federation. A Cloud federation is established when two or more domains designate shareable assets that can be remotely managed. The act of federating extends the resources of an initiator domain by the resource of another domain after a mutual agreement that creates the contract. As discussed before, the ontology inherits the schemas of the other ontologies in order to let the Federation Controller (FC) have the correct knowledge about the environment that it manages. The mechanism of keeping the ontologies separated is useful for several reasons. First it allows for the division of responsibility areas. Second, it provides a way to include other service descriptions that can be exploited to enhance platform independence, for example, in the case of an external Meta Cloud network. The federation ontology contains the concepts that extend the chain of schemas that describe the datacenter and the HNode, by defining a specific namespace (“federation:”) and URI.

In this scope, the ontology extension defines the Organization as the highest descriptive level. The Organization concept is related to the datacenter instances with the object property called “hasDatacenter” in order to have a complete list of the Clouds that belong to the Super Cloud. It’s important
that the Federation Controller has the list of all the Clouds because it represents the physical organization with all the affiliated entities that can be contacted in case of service discovery. The federation, described with the Community concept, is a logical subset of datacenters that joins it. These subsets can be overlapped since several communities can federate resources belonging to the same or different Clouds.

### 5.4.1. Federated Contracts

In order to federate, each domain needs to explicitly describe the characteristics of the lease. One of the most important concepts behind the establishment of a federation is the Contract. The federation contract needs to describe the details of the interactions among federating datacenters, especially regarding how failure will be handled and how Quality of Service will be managed across the domains. In the ontology, the FederationContract concept has an instance for every established community and it is a kind of super concept of the contracts that describe the leasing in the single datacenters. In order to respect the responsibility areas, the creation of a federation contract is reflected by one contract in every datacenter. The instantiated Contracts will also be written in the ABox of the federation ontology, in order to allow the Federation Controller to identify the endpoints. In the datacenter schema, the Contract concept has two datatypes (datacenter:CID, datacenter:Type) that can help the Cloud Controller to identify which VNode belongs to the federation, not to a simple local user. Inside the datacenter, the description of a contract will remain the same if a resource is located in another Cloud, except for the Type datatype that will specify the federation. The same exception is present in the User concept that describes the type of the user with the same specifications. Each local command request to that contract will be redirected to the Federation Controller that will send the right command to the right datacenter, indicating the Contract ID and the Command Type. The receiving datacenter will be able to identify the correct resource because it will have a similar instantiation of the Contract concept with the description of the leased resources. The Federation Controller will act as a mediator, just for the commands needed to manage the federation. All other activities, especially data intensive ones, will take place outside the overlay network. The communication channels will establish an out-of-band data stream on the lower overlay network (Internet), whose result state will be sent back to the controllers.

### 5.4.2. Command Execution

In order to change the state of a resource, the system provides a number of commands that can be invoked locally by the users of the Cloud provider or remotely if that resource has been leased in a federation. The following Figure 7 describes the UML sequence diagram of a command execution between two datacenters. After the first phase of authentication, Cloud Controller 1 sends an asynchronous request to the Federation Controller, describing in the payload of the message what kind of command is requested. The Federation Controller, after a further authentication phase, will send the command to the Cloud endpoint. Once the command is received by the datacenter service, the credentials are checked against the contract instances and the command is sent to the HNode service that converts the logic command into a real system command compatible with the local implementation. The execution of the command changes the state of the resource. The new state is sent back to every entity involved in the command invocation. This mechanism allows the distributed

![Diagram of Federated Command Execution](image-url)
ontologies to remain consistent.

5.4.3. Resource Discovery

Service discovery is a fundamental activity in a federated Cloud. This paragraph describes an algorithm for resource discovery in the proposed architecture. Since each Cloud is independent and has control over its resources, a mechanism for resource discovery must be designed to allow an internal or external entity to query the availability of a service and what kinds of services are offered. Some authors have proposed loosely coupled resource catalogs that could be queried from other systems [16]. However, this approach seems unrealistic in large-scale systems, essentially because the various ontologies should be updated and kept consistent. Hence, the proposed architecture has followed an opposite direction, strictly coupling the ontologies with the related resources. Furthermore this choice has been based on a preliminary analysis that indicated that most of the main activities, their frequency, and their the network and computation loads, try not to violate the principle of independence between systems. Figure 8 describes the algorithm developed to support this activity in the proposed architecture. The datacenter is the entity that provides the input and receives the output from the Federation Controller service. It is for this reason that a semantic based solution is useful. Indeed, in a scenario.

When the service that runs on the Federation Controller validates the input, it will query the ontology and obtain the set of Clouds belonging to the organization. The Federation Controller will forward the resource description to every datacenter, except the caller. The datacenters will receive the request, query the ontology and randomly select an available hardware node.

The datacenters that have the availability will respond describing the nature of the available resources. The answer to the requesting entity will contain the characteristics of the offering in terms of resources and billing. The Federation Controller will collect every answer sent by the datacenters (with a certain timeout restriction) and the requestor will receive the list of available datacenters ordered by a specified criterion. The identified criteria are several, the most important are related to non-functional requirements (e.g., SLAs) but another one could be related to a proximity measurement made by the Federation Controller considering the geo-location of the involved nodes. Further developments will include an algorithm for the optimal automatic selection once the list of available clouds has been obtained. This achievement will enable a high-level SaaS selection in the proposed ERS scenario.

5.5. Scope 5: Meta Cloud and P2P Federations

The Meta Cloud is the highest layer of the architecture. As seen in the introductory paragraph, like the Internet is a network of networks, the Meta Cloud can be considered a network of Federated Clouds. It’s

<table>
<thead>
<tr>
<th>Algorithm 1: Resource discovery over a Federated environment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong>: Resource descriptors ( r ), QoS descriptors ( q ), Sort Criterion ( s )</td>
</tr>
<tr>
<td><strong>Output</strong>: Set of endpoints with available resources and QoS metrics ( E )</td>
</tr>
<tr>
<td><strong>Definitions</strong>: Hardware Node ( n ), Datacenter ( d ), Ontology ( o ), End ( E ), Node ( n ), Cloud ( c ), Federation ( f )</td>
</tr>
<tr>
<td><strong>Procedures</strong></td>
</tr>
<tr>
<td><strong>Discovery</strong> ((r,q,s))**: Lists the datacenters that have the specified resources</td>
</tr>
<tr>
<td><strong>Initialization</strong> (E \leftarrow {\phi})**: E (\leftarrow d_i);</td>
</tr>
<tr>
<td><strong>Body</strong></td>
</tr>
<tr>
<td>foreach ((d_i \in o^{\phi})) do</td>
</tr>
<tr>
<td>if call_service ((d_i, check_resourceAvail(r,q)) \neq \phi) return (E);</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td><strong>checkResAvail</strong> ((r,q))**: Checks if a Cloud has the specified resources available</td>
</tr>
<tr>
<td><strong>Input</strong>: ( r,q )</td>
</tr>
<tr>
<td><strong>Initialization</strong> : Boolean availability of an hardware node ( a )</td>
</tr>
<tr>
<td><strong>Body</strong></td>
</tr>
<tr>
<td>foreach ((n_i \in o^{\phi})) if query ((n_i, q) \neq \phi) return (TRUE);</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>return (TRUE);</td>
</tr>
</tbody>
</table>

**Figure 8**: Federated Resource Discovery Algorithm
possible to describe the concept of the Meta Cloud as the whole set of activities in a future scenario where Clouds will be interconnected in a global wide network of Cloud providers that manage groups of distributed datacenters. The Meta Cloud is the result of an explorative scenario analysis that tries to establish a causality chain of events in the development of the Cloud Computing paradigm. The evaluation of the driving forces responsible of the current trends shows that the business of Cloud Computing, in the next few years, will encounter the end of an initial battle for supremacy, which can be called Cloud 1.0. The first phase of the Cloud 2.0 Era will bring to the market all the technologies that will be standardized de-facto by the winning players.

The idea of the Federated Cloud tries to represent a scenario where an organization manages different Cloud providers but also a future where different Cloud providers will implement different protocols and architectures. The use of ontologies that describe the characteristics of a Cloud service is nowadays considered a reasonable solution for new implementations. Some Cloud providers are already starting to describe semantically their infrastructure with new languages, and probably in the future will be possible to conform to a standardized way to describe every kind of service, up to the Platform and the Software layer. Furthermore, in the proposed architecture there’s a clean topologic separation between the Super Cloud and the Meta Cloud. A hybrid topology like this means that there should be a precise intersection point where two different network topologies must be able to communicate. The Federation Controller needs to manage a centralized network where the endpoints are the joining datacenters, but in order to allow the Federated Network to communicate with the Meta Cloud it’s necessary to implement a bridging entity. Furthermore the use of ontologies for the bridge can be useful to intercept the differences between requests and responses and create a static or dynamic translation mapping.

6. CONCLUSIONS AND FUTURE DEVELOPMENTS
The era of cloud computing is developing rapidly. Many research groups and companies are focusing on this paradigm in an attempt to exploit the capabilities it offers and to solve the issues and challenges involved. Surely, there’s a long journey ahead in a world where the quest for highly scalable and robust systems and services will always necessitate more and continuous flow of data moving throughout the Internet and among its users. The Cloud represents a significant development for next generation computing if it delivers on its promises. Yet, the shift in paradigm will not be painless and will require addressing several challenges in architecture, performance, security, interoperability and resilience. Federation will allow for the establishment of distributed peer-to-peer Clouds without the intervention of a centralized entity. Providing extended and scalable services will prove to be key aspect to focus on. In order to tackle some of the issues above, this work has investigated the new paradigm of Distributed Federated Clouds. It proposed a comprehensive architecture that supports federated clouds in the context of scalable multi-organizational distributed system. Such architecture will permit the efficient exploitation and sharing of resources and data that belong to independent organizations cooperating on joint tasks. The approach employed to implement the proposed architecture leverages semantic web technologies and solutions. Web semantics are considered in order to overcome several interoperability issues that exist between heterogeneous virtualization systems, data and resources. This implementation will preserve the independence of services and clouds providers. The paper motivates the reasons behind the work, introduces the terminology used in distributed cloud federation, discusses some of the related work, and gives an overview of this proposed architecture. It briefly describes the various components of the architecture and the purpose behind them. It then delves into low-level details of the semantic implementation of each of the five levels in the architecture. It identifies the concepts and components used, and illustrates how various parts will interact via commands and contracts. It also shows how to share and exchange while maintaining privacy and independence in the respective providers domains. Lastly, it briefly explains how to establish distributed resource discovery. We believe that the use of a description logic semantics for the systems and subsystems in this architecture should not be deemed solely for academic purposes since many production and real-world systems in recent years have been using this technology and the related tools to solve complex issues effectively. Hence, we foresee extended benefits in pursuing our proposed approach to the next level. In the growing literature in the area of cloud computing, there is no indication about the characteristics of hypothetical platform-as-a-service (PaaS) or software-as-a-service (SaaS) models in a federated environment. The next efforts will concentrate on investigations in this research direction, aiming at the development of a common platform that could span over different cloud implementations.

This work has focused on the infrastructure-as-a-service (IaaS) of a federated Cloud architecture. An implementation of the FCFA has been developed to prove the effective use of semantic technologies in this context. The service framework includes a set of APIs that will be used by the upper layer of the platform in order to provide a programming interface to the infrastructure layer beneath. The next steps will present the characteristics of the implementation, the case studies and the performance tests. Additionally, future work will also provide a semantic-based solution for resource monitoring and federated exception handling. Due to space limitations, this paper did not get into specific scenarios or case studies. However, we already have developed such examples using this architecture, which we intend to disseminate in forthcoming publication.
REFERENCES


