A Requirements Analysis for IaaS Cloud Federation

Alfonso Panarello, Antonio Celesti, Maria Fazio, Massimo Villari and Antonio Puliafito
DICIEAMA, Università degli Studi di Messina, Contrada Di Dio, S. Agata 98166, Messina, Italia

Keywords: Cloud Computing, Federation, Requirements Analysis, XMPP, SAML, XACML.

Abstract: The advent of Cloud computing offers different ways both to sell and buy resources and services according to a pay-per-use model. Thanks to virtualization technology, different Cloud providers supplying cost-effective services provided in form of Infrastructure as a Service (IaaS) have been rising. Currently, there is another perspective which represents a further business opportunity for small/medium providers known as Cloud Federation. In fact, the Cloud ecosystem includes hundreds of independent and heterogeneous cloud providers, and a possible future alternative scenario is represented by the promotion of cooperation among them, thus enabling the sharing of computational and storage resources. In this paper, we specifically discuss an analysis of the requirements for the establishment of an IaaS Cloud Federation.

1 INTRODUCTION

Cloud computing is a new paradigm able to provide on-demand servicesin a transparent way to usersaccording to given pay-per-use constrains.These services are arranged by providers using distributed and virtualized computing resources. This approach avoids to small and medium enterprises to make large investments of capital for purchasing their hardware/software equipments. The flexible and dynamic use of services takes place by means of thevirtualization technology that allows to decouple applications from the physical machine on which they run by means of Virtual Machines (VMs). Furthermore, VM migration gives the opportunity to guarantee a particular degree of Quality of Service (QoS).Thus, aggregating and mapping VMs, a Cloud provider is able to supply different levels of service: Infrastructure as a Service (IaaS), Platform as a service (PaaS) and Software as a Service (SaaS). Considering the cloud computing ecosystem, besides large cloud providers, smaller ones are also becoming popular even though their own virtualization infrastructures (i.e., deployed in their datacenters) cannot directly compete with the bigger market leaders. The result is that often small/medium cloud providers have to exploit the services of mega-providers in order to develop services. Thus, a possible future alternative scenario is represented by the promotion of a cooperation among small/medium providers, enabling the sharing of computational and storage resources. IaaS Cloud Federation is a partnership between providers for the borrowing/lending of virtual equipments including VMs, virtual clusters, virtual networks, and so on.However, guidelines regarding the design and implementation of the functionalities enabling IaaS Cloud Federation are not precisely defined.

In this paper, we discuss a requirement analysis for the establishment of an IaaS Cloud Federation. In particular, considering the "three-phase model" (A. Celesti, et al, 2010a), we identify the involved actors, also discussing automatism, scalability, versatility, and security features.Furthermore, we will analyze the major standards and protocols useful for the establishment of an IaaS Cloud Federation.

The paper is organized as follows. In Section II, we summarize several initiatives regarding Cloud Federation. An overview on the major IaaS pieces of middleware is discussed in Section III. In Section IV, we provide a detailed analysis of the requirements regarding the IaaS Cloud Federation. In Section V, we discuss a case of study considering the three-phase model and several emerging technologies. Section VI concludes the paper.

2 RELATED WORKS

Several architecture and models for Cloud Federation have been proposed recently with

Panarello A., Celesti A., Fazio M., Villari M. and Puliafito A..
A Requirements Analysis for IaaS Cloud Federation.
DOI: 10.5220/0004945705840589
In *Proceedings of the 4th International Conference on Cloud Computing and Services Science* (CLOSER-2014), pages 584-589
ISBN: 978-989-758-019-2

different motivations. A three-phase cross-cloud model was presented in (Celesti et al., 2010b) In this work, the federation process takes place according to three subsequent phases, i.e., discovery, match-making, and authentication. The Reservoir FP7 European project (Rochwerger et al., 2010) introduced a modular and extensible Cloud supporting the management of business services and cloud federation. Claudia (Rodero-Merino et al,. 2010) provides an abstraction layer that allows the execution of services on top of a transparent federation of Cloud providers. Open Cirrus (Avetisyan et al., 2010) is a federation initiative between universities and research centers promoting the research in design, provisioning,and management of services in scale of multi data centers. Sky Computing (Keahey et al., 2009) introduces a virtual site layer on dynamically provisioned distributed resources provided by several data centers and a closed federation model, where the sharing of resources is based on cooperation basis like in a grid. OPTIMIS (Ferrera et al., 2012) is a platform for Cloud service provisioning that manages the whole lifecycle of the service and that also addresses issues such as risk, trust management, energy efficiency, and legislation. However, OPTIMIS does not address negotiation and a marketplace for discovery of resources. Other work which focus on low-level aspects of federation, such as security, networking, and disk image management is (Bernstein et al., 2009). Regarding scalable applications across multiple independent Cloud data centers, a market-based trading mechanism is required. The *InterCloud project* (Buyya et al., 2010) focuses on Cloud data centers and brokers that dynamically negotiate resources. Key component of *InterCloud* is the *Cloud coordinator*, whose architecture is described in (Calheiros et al., 2012). Other recent works that focus on other aspects of CoudFederation brokering are discussed in (Garge et al., 2011). A Cloud federation architecture based on satellite communications is discussed in (Celesti et al., 2012c). In (Celesti et al., 2012), the authors discuss how the data web can support Cloud federation. Furthermore, a mathematical programming technique to minimize the cost of leased VMs in investigated in (Chaisiri et al., 2009). In (Celesti et al., 2013), an architecture to build Platform as a Services (PaaS) exploiting different IaaS Cloud providers is proposed. These works focus on howto minimize the cost of external provisioning in hybrid Cloud providers. In (Breitgand et al., 2011), it is proposed a integer programming formulations for placement of VM workloads within as well as across multiple Cloud providers collaborating in a federation.

3 A SURVEY ON IaaS MIDDLEWARE

In this Section, we providean overview of the major existing open source IaaS Cloud pieces of middleware, evaluating their main features.

Nimbus (Nimbus, 2013) is an open source toolkit that allows turning a set of computing resources into an IaaS cloud. Nimbus comes with a component called workspace control, installed on each node, used to start, stop,and suspend VMs. It implements disk-image reconstruction and management, and securely connects the VMs to the network, and delivers contextualization. Nimbus's workspace control tools work with Xen and KVM but only the Xen version is distributed. Nimbus provides interfaces to VM management functions based on the WSRF set of protocols.

Eucalyptus (Eucalyptus, 2013) is an open-source framework that uses the computational and storage infrastructures commonly available at academic research groups to provide a platform that is modular and open to experimental instrumentation and study. Eucalyptus addresses several crucial Cloud computing questions, including VM instance scheduling, administrative interfaces, construction of virtual networks, definition and execution of service level agreements (cloud-to-user and cloud-to-cloud), and cloud computing user interfaces.

OpenQRM (OpenQRM, 2013) is an open-source platform for enabling flexible management of computing infrastructures. Thanks to its pluggable architecture, OpenQRM is able to implement a cloud with several features that allows the automatic deployment of services. It supports different virtualization technologies managing Xen, KVM and Linux-VServer. It also supports P2V (physical to virtual), V2P (virtual to physical) and V2V (virtual to virtual) migration. This means Virtual Environments (VEs) (appliances in the OpenQRM terminology) cannot only easily move from physical to virtual (and back), but that they can also be migrated from different virtualization technologies.

OpenNebula (OpenNebula, 2013) is an open and flexible tool that fits into existing data center environments to build a Cloud computing environment. OpenNebula can be primarily used as a virtualization tool to manage virtual infrastructures

in the data center or cluster, which is usually referred as private Cloud. OpenNebula also supports public Clouds by providing Cloud interfaces to expose its functionalities for VM, storage, and network management.

OpenStack (OpenStack, 2013) is IaaS cloud computing project that is a free open source software released under the terms of the Apache License. The project is managed by the OpenStack Foundation, a non-profit corporate organization. The technology consists of a series of interrelated projects that controls large pools of processing, storage, and networking resources throughout a datacenter, all managed through a dashboard.

CLEVER (CLoud-Enabled Virtual EnviRonment) (Celesti et al., 2012) is modular and pluggable middleware that specifically aims at the administration of private Cloud infrastructures. CLEVER is able to manage cluster of nodes each containing a host level management module (Host Manager). A single node may also include a cluster level management module (Cluster Manager). All these entities interact exchanging information by means of the Communication System based on the *Extensible Messaging and Presence Protocol (XMPP)*. The set of data required to enable the middleware functionalities is stored within a specific database deployed in a distributed fashion. CLEVER offers security and fault-tolerance.

4 FEDERATION REQUIREMENTS ANALYSIS

Despite of the obvious advantages that Cloud Federation offers, its implementation is not at all trivial. The main reason is that Cloud providers are more complicated than traditional systems and the existing federation models are not applicable. In fact, while Cloud providers are typically heterogeneous and dynamic, the existing federation models are designed for static environments where it is needed an a priori agreement among the parties. With regard to IaaS Cloud providers, we think federation needs to meet the following requirements:

a) *Identification of the Actors in a Federated System*: it is necessary to define which are the entities that cooperate each other in a federated environment. In particular, it is necessary to exactly know who plays the role of the cloud system provider and what is the role of Cloud system consumer.

b) *Automatism and Scalability*: In a federatedenvironment, a Cloud provider that accomplishautomated decisions has to determine which external provider has to be used for a particular workload, because not all Clouds are equal in terms of warranty, cost, reliability, QoS and resource availability. For example, a particular Cloud provider may be cheaper, but it could not provide guarantees of availability, making it unsuitable for mission-critical workloads. Another Cloud provider, however, could provide "five nines" availabilitybut be more expensive. For this reason, a Cloud provider requiring additional resources, should be able to pick out the right Clouds provider which satisfies its requirements reacting also to sudden environmental changes;

c) *Versatility and Security*: In a heterogeneous Cloud federation scenario, interoperability is a key concept. A Cloud provider requiring additional resources must be able to work with multiple Cloud providers based on different pieces of middleware. The ability of a Cloud consumer to be able to integrate different external Cloud providers in a transparent manner allows to integrate computing, storage, and network services across multiple operators simultaneously.

In order to achieve federation, it is fundamental the integration of different security technologies, for example, permitting a Cloud provider to be able to join the federation without changing its security policies. The Cloud systems require, generally, a simple, secure access to resources that they make available. The access to Cloud services is often achieved through web interfaces. Regardless of the model that will be implemented, an authentication system that allows a Cloud provider to access the resources offered by other operators maintaining its own identity, it is a necessary element. Hence, a Single Sign-On (SSO) authentication system is required.

d) *Authorization:* In a federated environment it is necessary that the internal management policies of the Cloud operators coexist without interfering with each other. Nowadays, it is necessary to apply access policies to user profiles and resources, because in a distributed system the access to a particular resource must be controlled. Typically, these policies establish which are the entities that may access a specific resource in a distributed system according to a matching criterion that allows to check whether the requirements of the subject correspond to the requirements of the system.

5 CASE OF STUDY

Currently, there are not standards that determine how the process of federation should be accomplished. In this Section, in order to describe how the *IaaS Cloud Federation* can take place, we will consider the three-phase model and several emerging technologies for its implementation.

A – The Three-phase Model. The three-phase model implies three subsequent phases for federation establishment: discovery, match-making, and authentication. During the discovery phase, all the available Cloud providers have to be discovered. As this environment cannot be a priori known, but it is pretty flexible and dynamic, the discovery process should be implemented in a totally distributed fashion with a peer-to-peer (p2p) approach. After that, the match-making phase has to choose the more convenient cloud providers with which to establish federation. All the available (discovered) providers have to be associated to several policies, describing the offered resources according to particular conditions. In addition, these policies have to match the policies of the provider requiring external resources. During the authentication phase, after that the Cloud providers have been selected for federation, a mechanism for creating a security context among the involved Cloud providers should be accomplished by means of Single Sign-On mechanisms.

B - XMPP or Web Services for Discovery Phase. As depicted in Figure 1, possible solutions to the discovery problem consist in using XMPP or Web Services. The architecture of the XMPP (Ejabberd, 2013) network is quite similar to the approach used by the e-mail service. Every user on the network is associated to a Jabber ID (JID) and communicate each other using a chat room. Scenario A shows a Cloud consumer that in order to gain knowledge about all the available Cloud providers, retrieves other providers' availability information merely querying the "shared location" on which it is published. More specifically, in a XMPP based scenario, the "location" is identified with a specific chat room, where the operators aiming to take part the federation hold a subscription.

Scenario B shows an alternative solution based on Web Services. This scenario includes a UDDI web server that is a xml-based registry (a sorted , indexed database) which allows Cloud providers to publish their WSDL document. A WSDL document give e formal description of the public interface of a web service. Therefore, a Cloud consumer can "read" the WSDL document related to a Web Service to determinate how the offered service can be used.

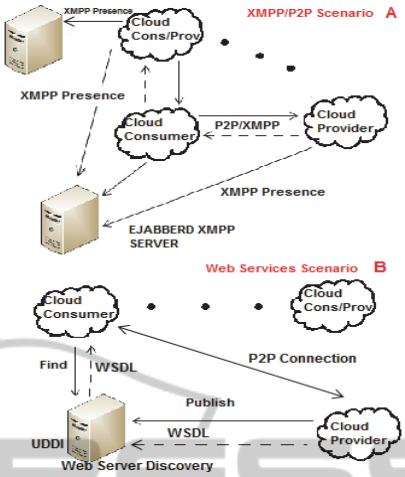

Figure 1: XMPP/P2P/Web Services for Discovery.

C - XACML for Match-Making Phase. During the Match-Making phase, it is needed to pick out which of the discovered operators. More specifically, the policies of the involved Cloud providers must coexist without affecting each other. Nowadays, in large-scale distributed systems there is the need to enforce policies in many different enforcement points. Typically such policies govern which subjects can access to a target resource of a distributed system according to a policy matching task performed in order to check if the requirements of the subject match the requirements of the system. To this end, a Cloud provider should be able to choose whether to accept or not a given Cloud consumer and it must also be able to restrict such access as appropriate. One of the best solutions which is able to address the aforementioned scenarios is the *eXtensible Access Control Markup Language* (XACML) technology (XACML, 2013). XACML allows to express policies by means of four major components: attributes, rules, policies, and policy set.

Attributes are characteristics of subjects, resources, actions, or environments that can be used to define a restriction. A *Rule* is the basic element of a policy. It identifies a complete and atomic authorization constraint which exists in isolation respecting to policy in which it has been created. A rule is composed by a Target, to identify the set of requests the rule is intended to restrict, an Effect, which is either "Permitted" or "Denied. *Policies*. A *Policy* is a combination of one or more rules. A policy contains a Target (specified using the same

components as the rule Target), a set of rules, and a rule combination algorithm.

Due to its nature, the XACML technology can be successfully used toachieve the match-making phase. Figure 2 shows how XACML allows to accomplish the match-making phase considering two Cloud operators. The actors of the above scenario are: a Cloud Consumer, a Cloud Provider, a server PEP (Policy Enforcement Point), a server PDP (Policy Decision Point) and a Policy Store. In the first step of the authorization process, the Cloud Consumer sends a Resources Request to PEP.

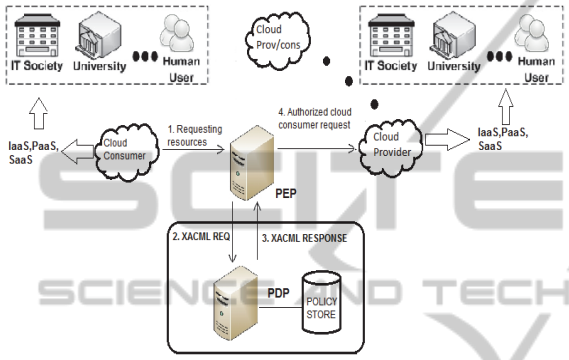

Figure 2: XACML Authorization Scenario.

The second step shows that the PEP translates the resource of the Cloud consumer in a XACML request. PEP forwards XACML request to PDP and asks whether the cloud consumers is authorized to access the requested resources. PEP evaluates the policies in the Policy Store and takes the decision according to the defined policies. At the third step PDP returns to PEP a XACML response message which contains the taken authorization decision. At the fourth step, the PEP depending on the XACML message received grants or denies access to resources.

D – SAML, OpenID and Shibboleth for Authentication Phase. During the Authentication phase it is needed to establish a trust context between different operators by means of SSO authentication mechanisms. In order to achieve such a goal, one of the major technologies using the Identity Provider/Service Provider (IdP/SP) model is *the Security Assertion Markup Language* (SAML) (SAML, 2013). It is an open standard based on XML for the exchange of authentication assertions between different parties. Figure 3 shows how the IdP/SP model can be used to establish a secure context between two operators. The Cloud consumer tries to access a Cloud provider (step 1). The Cloud provider requires a SAML assertion type (step 2). Therefore, the Cloud provider redirects the user to the IdP (step 3). The IdP authenticates the Cloud consumer that responses with authentication information provided by the IdP (step 4). The Cloud consumer forwards to Cloud provider a resources request attaching to it the authentication token obtained by IdP (step 5).

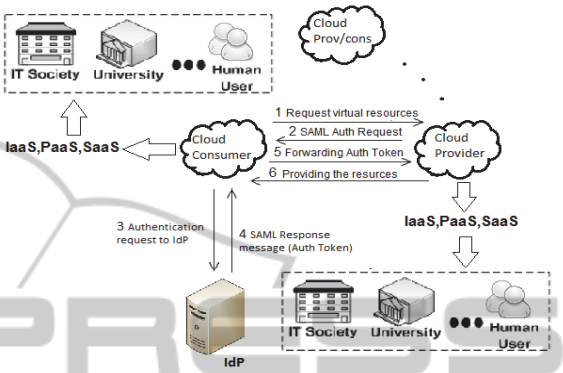

Figure 3: SAML Authentication Scenario.

Cloud provider unpacks the request obtaining the token and verifies its correctness. Finally, the Cloud provider contacts the cloud consumer notifying where and how to access the requested resources (step 6).

The Cloud consumer, subsequently, can request access the resources of other Cloud providers relying on IdP without further authentication tasks.

Other valuable technologies used to accomplish the SSO authentication are OpenID (OpenID, 2013) and Shibboleth (Shibboleth, 2013) .

6 CONCLUSIONS

Cloud federation is an emerging topic. Currently there are not so many standards and guidelines to accomplish a federation. In this paper, firstly, we performed a requirement analysis for the establishment of the IaaS Cloud federation, analyzing different architecture and models. Secondly, we discuss a case of study adopting the three-phase model and considering several technologies. We hope, our analysis will be useful for Cloud architects facing federation issues.

REFERENCES

A. Celesti, F. Tusa, M. Villari and A. Puliafito, "How to Enhance Cloud Architectures to Enable Cross-

Federation", *IEEE 3rd International Conference on Cloud Computing (CLOUD)*, pp.337-345, 2010.

B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas,I.Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P. Massonet, H. Munoz and G. Toffetti, "Reservoir - when one cloud is not enough", in *Computer*, pp. 44-11, 2010.

L. Rodero-Merino, L. M. Vaquero, V. Gil, F. Galán, J. Fontán, R.S. Montero and I. Llorente, "From infrastructure delivery to service management in clouds" in *Future Generation Computer Systems*, 1226-1240, 2010.

A. Avetisyan , R. H. Campbell et al, "Open Cirrus: A Global Cloud Computing Testbed", in *Computer*, vol. 43, no. 4, pp. 35-43, 2010.

K. Keahey and M. Tsugawa and A. Matsunaga and J. A. B. Fortes, "Sky Computing", in *IEEE Internet Computing*, pp. 43-51, 2009.

A. J. Ferrera, F. Hernándezb et al, "OPTIMIS: A holistic approach to cloud service provisioning", in *Future Generation Computer Systems*, 66-77, 2012.

D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond and M. Morrow, "Blueprint for the InterCloud - protocols and formats for cloud computing interoperability", in *Fourth International Conference on Internet and Web Applications and Services (ICIW '09)*, pp. 328-336, 2009.

R. Buyya , R. Ranjan, R. N. Calheiros, InterCloud: utility-oriented federation of cloud computing environments for scaling of application services, *Algorithms and Architectures for Parallel Processing, ICA3PP'10*, pp. 13-31, 2010.

R. N. Calheiros, A. N. Toosi, C. Vecchiola, R. Buyya, A coordinator for scaling elastic applications across multiple clouds, *Future Generation Computer Systems,* 1350-1362, 2012.

S. K. Garg, C. Vecchiola, R. Buyya and Mandi, June 2011, A market exchange for trading utility and cloud computing services, *The Journal of Supercomputing*, pp. 1153-1174.

A. Celesti, M. Fazio, M. Villari, A. Puliafito, Virtual machine provisioning through satellite communications in federated Cloud environments, *Future Generation Computer Systems*, Volume 28, Issue 1, pp. 85-93, 2012.

A. Celesti, F. Tusa, M. Villari and A. Puliafito, "How the Dataweb Can Support Cloud Federation: Service Representation and Secure Data Exchange", *Second Symposium on Network Cloud Computing and Applications (NCCA)*, pp.73-79, 2012.

S. Chaisiri, B. Lee and D. Niyato, "Optimal virtual machine placement across multiple cloud providers", in *IEEE Asia-Pacific Services Computing Conference (APSCC 2009)*, pp. 103-110, 2009.

A. Celesti, N. Peditto, F. Verboso, M. Villari, A. Puliafito, "DRACO PaaS: A Distributed Resilient Adaptable Cloud Oriented Platform", in *IEEE 27th Internationa Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, pp.1490,1497, 2013.

D. Breitgand, A. Marashini and J. Tordsson, "Policy-driven service placement optimization in federated clouds", in *Computer Science Technical Report*, IBM Haifa Labs, 2011.

Nimbus, "Cloud Computing for Science". http://www.nimbusproject.org, 2013

Eucalyptus, "Open Source Aws Compatible Private Clouds", http://eucalyptus.cs.ucsb.edu, 2013.

OpenQRM Enterprise, http://www.openqrm-enterprise.com/, 2013.

OpenNebula, "Flexible Enterprise Cloud Made Simple, opennebula.org", 2013

OpenStack, "Open source software for building", 2013
private and public clouds", http://www.openstack.org.

A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "Integration of CLEVER clouds with third party software systems through a REST web service interface", in *IEEE Symposium on Computers and Communications (ISCC '12),* pp. 827-832, 2012.

Ejabberd, "The Erlang Jabber/XMPP Deamon", http://www.ejabberd.im, 2013.

XACML, https://www.oasis-open.org/, 2013

SAML,https://www.oasis-open.org/committees/, 2013

OpenID, http://openid.net/developers/specs/ 2013,

Shibboleth, https://shibboleth.net, 2013.