Developing Interoperable and Federated Cloud Architecture

Gabor Kecskemeti
University of Miskolc, Hungary

Attila Kertesz
University of Szeged, Hungary

Zsolt Nemeth
MTA SZTAKI, Hungary

A volume in the Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series

Information Science REFERENCE
An Imprint of IGI Global

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Names: Kecskemeti, Gabor, 1981- editor. | Kertesz, Attila, 1980- editor. | Nemeth, Zsolt, 1971- editor.
Title: Developing interoperable and federated cloud architecture / Gabor Kecskemeti, Attila Kertesz, and Zsolt Nemeth, editors.
Description: Hershey, PA : Information Science Reference, 2016. | Includes bibliographical references and index.
Identifiers: LCCN 2015051296| ISBN 9781522501534 (hardcover) | ISBN 9781522501541 (ebook)
Subjects: LCSH: Cloud computing. | Computer network architectures.
Classification: LCC QA76.585 .D488 2016 | DDC 004.67/82--dc23 LC record available at http://lccn.loc.gov/2015051296

This book is published in the IGI Global book series Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) (ISSN: 2327-3453; eISSN: 2327-3461)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 1
Cloud Federations: Requirements, Properties, and Architectures

Marcio R. M. Assis
University of Campinas, Brazil

Rafael Tolosana-Calasanz
Universidad of Zaragoza, Spain

Luiz Fernando Bittencourt
University of Campinas, Brazil

Craig A. Lee
The Aerospace Corporation, USA

ABSTRACT

With the maturation of the Cloud Computing, the eyes of the scientific community and specialized commercial institutions have turned to research related to the use of multiple clouds. The main reason for this interest is the limitations that many cloud providers individually face to meet all the inherent characteristics of this paradigm. Therefore, using multiple cloud organizations opens the opportunity for the providers to consume resources with more attractive prices, increase the resilience as well as to monetize their own idle resources. When considering customers, problems as interruption of services, lack of interoperability that lead to lock-in and loss of quality of services due to locality are presented as limiting to the adoption of Cloud Computing. This chapter presents an introduction to conceptual characterization of Cloud Federation. Moreover, it presents the challenges in implementing federation architectures, requirements for the development of this type of organization and the relevant architecture proposals.

INTRODUCTION

Cloud Computing (Mell & Grance, 2011) has emerged as a *vedette* in information technology in the 21st century, presenting a paradigm shift on how computing capacity is acquired by consumers. In this paradigm, computing resources of various kinds are offered as a service in the form of utilities, where users pay according to their necessity for computing power. Computing services in clouds can be offered at three different levels, according to the computing object being offered: (i) Infrastructure as a Service (IaaS), offered to infrastructure management clients; (ii) Platform as a Service, offered to application

DOI: 10.4018/978-1-5225-0153-4.ch001

development clients; and (iii) Software as a Service (SaaS), offered to the application's final users. The most prominent characteristic that makes cloud computing attractive is the elasticity, which allows management of computing power, increasing or decreasing it, according to the workload. For cloud clients, elasticity allows cost reduction and avoidance of upfront investments in computing infrastructure. On the other hand, providing elasticity is a challenging technical issue that must be tackled by cloud providers.

Elasticity provisioning is inherent to the amount of physical resources (e.g., CPU) that each cloud provider has on its datacenter(s). Therefore, resource exhaustion can compromise service offering to cloud clients, as well as hamper the quality of services already running, especially in small- and medium-sized cloud providers. Other limiting factors of monolithic clouds (where a provider is a single, isolated, domain) include the business continuity problems in case of unexpected faults that cause service disruption (Toosi, Calheiros, & Buyya, 2014; Grozev & Buyya, 2012); the challenging issues related to lack of geographical dispersion, which can affect quality of service; and lack of interoperability with other providers (Grozev & Buyya, 2012; Assis, Bittencourt, & Tolosana-Calasanz, 2014). In face of such limitations, a need for evolution of this technology arises, where solutions of multiple clouds started to be designed and deployed.

Along with the multiple clouds solutions recently proposed, such as Multi-Clouds (Kurze et al., 2011; Grozev & Buyya, 2012; Toosi et al., 2014) and Sky-Computing (Keahey, Tsugawa, Matsunaga, & Fortes, 2009), the Cloud Federation can be highlighted as a voluntary association of clouds subject to a federative contract that defines the behavior (duties and penalties) of participating entities. In this chapter we define and discuss properties, opportunities, challenges, current research state and development of Cloud Federations.

The remainder of this chapter is organized as follows: the Background section presents the characteristics of Cloud Computing in detail. The definition of Cloud Federation, the motivations to the emergence of this kind of association, the open challenges in Federations, and the identified characteristics are presented in Cloud Federations: Motivations and Challenges section. The Cloud Federations Properties section describes the properties identified in cloud federations. In Architectural Specifications, Blueprint and Existing System section, some of the main federation architectures available in the literature are presented, followed by the concluding section.

BACKGROUND

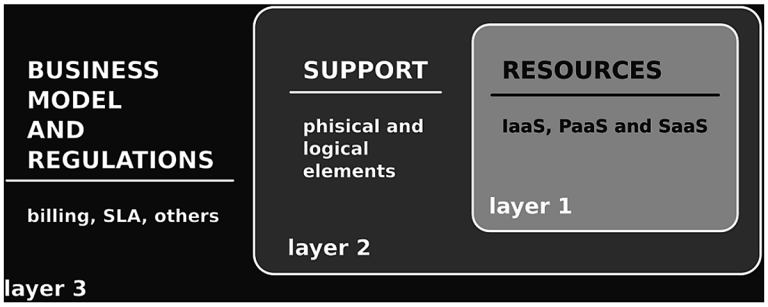
This section aims to provide insight for understanding the rest of the document. It presents the Cloud Computing paradigm, exploring its main properties, delivery and deployment models, as well as covering their characteristics and key elements.

Cloud Computing

The term Cloud Computing has frequently been used as a synonym for technological advancement. However, there is not a uniform understanding of its meaning, which is mainly due to the overloading of multiple related concepts behind the term Cloud.

As quoted in the compilation of cloud-related work performed by Vaquero et al. (Vaquero, Rodero-Merino, Caceres, & Lindner, 2008), some authors as Watson et al. (Watson, Lord, Gibson, Periorellis, & Pitsilis, 2008) and Geelan et al. (Geelan et al., 2008) define Cloud Computing as a novel computing

Figure 1. Three-layer model of Cloud Computing



paradigm providing resources through a new business model, which allows the reduction of capital used to purchase resources (Zhang, Cheng, & Boutaba, 2010). Antonopoulos and Lee (Antonopoulos & Gillam, 2010) describe Cloud Computing as the natural evolution of existing technologies offered on a new business model in which consumers only pay for usage resources of interest. In another seminal work, the technical report produced by the UC Berkley Adaptive Distributed Systems Laboratory (Armbrust et al., 2009) states that Cloud Computing refers to the applications offered as a service (SLA@SOI, 2011) over the Internet and all hardware and software used to provide the services. The institute National Institute of Standards and Technology (NIST) (Mell & Grance, 2011) describes Cloud Computing as a model to conveniently activate a set of computational resources that can be rapidly provisioned and released with minimal effort or interaction.

Synthesizing the definitions above, Cloud Computing can be described as a set of several existing technologies working in a symbiosis to provide computing resources to interested parties on a new paradigm of computing utility and marketing, where customers sometimes pay for usage according to restrictions and duties defined in a contract. This description allows Cloud Computing to be organized in a three-layer model, as shown in Figure 1. At the first layer are the resources for the provisioning in the form of services; at the second layer there are physical and logical elements that enable the operation of the cloud; and at the peripheral layer the business model and items that regulate how services will be offered and charged can be found.

Cloud Computing Paradigm Properties

Although Cloud Computing shares characteristics with previous types of distributed systems, there is a number of properties for Cloud Computing that differentiate it from previous paradigms.

Physical Resource Sharing

Recent advances in virtualization technologies have enabled an efficient sharing, or multi-tenancy, of physical resources (i.e. networking communications, processors, memory, and storage) among various users (Zhang et al., 2010). Virtualization is a critical aspect in Cloud Computing as it supports the on-demand computation by allowing stakeholders to adjust customized resources to the run-time requirements. This is made considering the isolation between consumers (the resources allocated to a consumer cannot be accessed by others without authorization), and it offers consumers control over the acquired virtualized resources (Foster, Zhao, Raicu, & Lu, 2008).

Elasticity

Elasticity is the ability of a system to adapt to workload changes by on-demand provisioning and de-provisioning of resources, such that at each point in time the available resources match the current demand as closely as possible (Herbst, Kounev, & Reussner, 2013). As discussed by Herbst et al. in (Herbst et al., 2013), elasticity differs from scalability in which the latter is the ability of the system to sustain increasing workloads by making use of additional resources at run-time and without requiring human intervention.

Self-Provisioning

Self-provisioning is the property that allows Cloud consumers to manage contracted resources in an affordable and agile way, reducing bureaucracy and increasing the dynamism of use. This is accomplished mostly often by means of a dashboard, which makes the interface more user-friendly to the environment. This dashboard enables consumers to perform management and monitoring tasks such as instantiation and consumption of resources agreed in the Service Level Agreement (SLA), as well as reporting without requiring previous, advanced knowledge in administration of these types of activities.

Business Model

The marketing model in this technology differs from conservative systems that determine a fixed monthly fee for consumption of a quantity of resources within a time period (month, year etc.). Called pay-as-you-go (Mell & Grance, 2009; Foster et al., 2008), this model is to charge only what the customer consumes. For the values to be charged, metrics are used to account the amount of resources consumed (e.g. processing cycles/hour), and basic values (Zhang et al., 2010) by unit use (e.g. the value of a processing cycle). Both items are specified and pre-agreed in the SLA.

Delivery Models

In Cloud Computing, resources are offered in the form of services (Gang & Mingchuan, 2014). There is no explicit limitation on the type of resource that can be offered as a service within this paradigm, ant it is common to find in the literature (Banerjee et al., 2011; Armbrust et al., 2009) the term Everything as a Service (XaaS). Within this universe of features, the ones that act on resources can also be found and may also be available as services, such as high availability and monitoring (Al-Hazmi, Campowsky, & Magedanz, 2012).

There is a canonical organization of services in Cloud Computing (Zhang et al., 2010). This organization consists of three distinct delivery models that are now a reference to all the services offered in this computing model. It is possible to characterize these classes by the target audience (infrastructure manager, developer, and end user) of their services. Considering this relationship we have: i) Infrastructure as a Service (IaaS) – focuses on providing resources such as processor, storage or communication networks, ii) Platform as a Service (PaaS) – focuses on mapping applications onto the infrastructure, and iii) Software as a Service (SaaS) – focuses on end-users and providing them with applications. Figure 2 organizes these three models of service. These service models were originally used to denote how much of the system stack is “owned” by the provider and the user, as illustrated in Figure 3.

Figure 2. Canonical organization of cloud delivery services

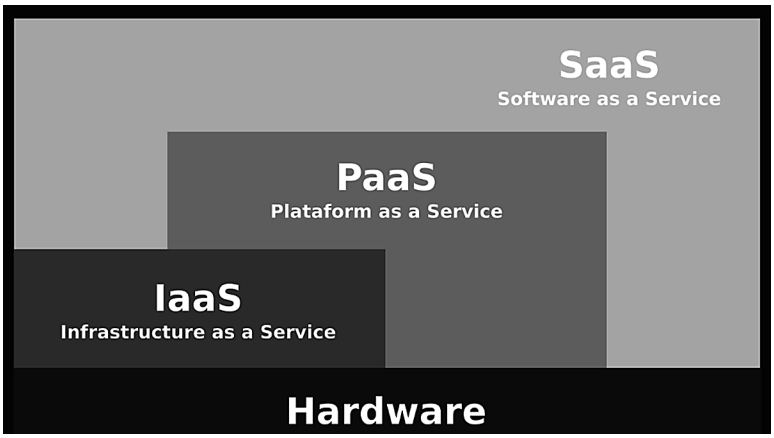
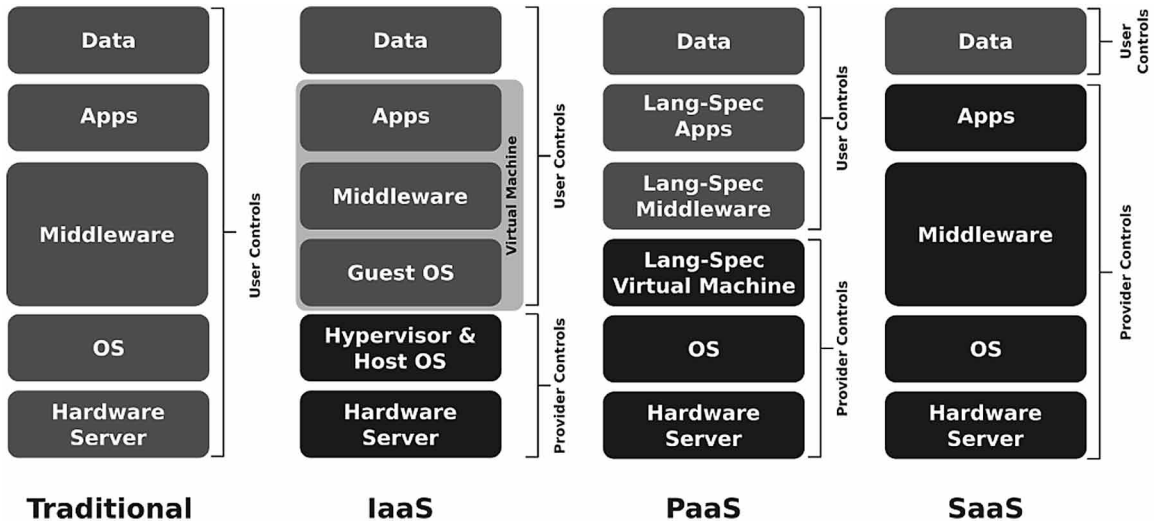


Figure 3. System stack ownership in cloud delivery service



Infrastructure as a Service

Infrastructure as a Service (Bhardwaj, Jain, & Jain, 2010) is the basis of Cloud Computing. It is the delivery of computing infrastructure assets that can be offered (processors, storage areas, etc.) as a service to customers. These assets are made available through virtual partitions of physical resources (some exceptions (Campbell et al., 2009) do not use virtualization). However, customers see these partitions as completely isolated and independent resources. There are several IaaS providers available in the market. Amazon stood out for popularizing Cloud Computing to the general public and has a portfolio of IaaS products, including EC2 (EC2, 2015). Microsoft also offers infrastructure as a service through Microsoft Azure (Azure, 2015). Adding to them, Google and Rackspace companies that, respectively, provide Google Compute Engine (Google Compute Engine, 2015) and the Rackspace Open Cloud (Rackspace, 2015). A study by Li et al. (Li, Yang, Kandula, & Zhang, 2010) compares the features and functionalities of the leading IaaS providers that were available in 2010.

Platform as a Service

Platform as a Service (Tolosana-Calasanz, Bañares, & Colom, 2015) abstracts the underlying computing infrastructure and provides the developer with a language interface, so that both the program logic and the SLAs can be specified. It is of paramount importance that such specifications are infrastructure-agnostic, that is, without referring to specific details of a particular infrastructure. PaaS aims at application developing and subsequent deployment. Hence, it typically provides a complete set of tools and programming models and interfaces for processing the logic and automatically deploying and executing them into de underlying infrastructures (Tolosana-Calasanz et al., 2015).

Software as a Service

The Software as a Service model provides software applications as a service, thereby end-users do not have to install them in their computers, but they can access them through the network. This delivery model allows end-users to pay for the usage of the software and save costs of management and maintenance of hardware. On the other hand, SaaS providers can also benefit by offering the same software instance to a multiple clients, adopting the so-called SaaS Multi-tenancy architecture.

Canonical Deployment Models

According to NIST, there are four established deployment models for Cloud computing, namely Private, Public, Hybrid, and Community clouds. Moreover, both Hybrid and Community clouds can actually involve the interconnection and usage of multiple cloud infrastructures.

In a *Private cloud*, the infrastructure is provisioned for exclusive use by a single organization. In turn, the organization comprises a number of users. It may be managed by the organization itself or by a third-party (Mell & Grance, 2011). In *Public clouds*, the infrastructure is provided for open use by the general public (Mell & Grance, 2011). It may be managed by a number of organizations, and typically users pay per the usage, while the provider agrees to enforce the Quality of Service (QoS) in accordance with a previously negotiated SLA. Amazon EC2 or SoftLayer (Softlayer, 2015) are examples of this type of deployment model. A *Hybrid cloud* is a composition of two or more distinct cloud infrastructures (i.e. private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds) (Mell & Grance, 2011). Finally, *a Community cloud* is a collaborative effort among several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.). The organizations share their infrastructures among the users (Mell & Grance, 2011).

CLOUD FEDERATIONS: MOTIVATIONS AND CHALLENGES

The study of Cloud Federations is a relatively new topic in the broader Cloud Computing subject. Therefore, some concepts are new and diffuse. One of the pioneers in research in this area is Dr. Rajkumar Buyya in his work with Grozev (Grozev & Buyya, 2012), which discussed aspects of the formalization of various concepts related to multiple cloud organizations. The same authors addressed Cloud Federations as a voluntary grouping of different clouds that work together to exchange resources[1] when needed.

In another study, Buyya et al. (Buyya, Ranjan, & Calheiros, 2010) report that a Cloud Federation must have at least three characteristics in order to be effective: to be able to dynamically expand or resize the present resources to meet the demand that may arise; operate as part of a market directed to loan resources; and, finally, deliver reliable services to customers, with effective costs and respecting QoS predetermined by a contract.

In works by Manno et al. (Manno, Smari, & Spalazzi, 2012) and Celesti et al. (Celeti et al., 2010), a Cloud Federation is described as a geographically dispersed community, where several heterogeneous and autonomous clouds cooperate sharing computer resources to achieve a common goal described in a contract. This agreement also defines the economic and technical aspects of the federation: charging model, quality of service, policies, use restrictions and penalties that may arise when the restrictions are violated. Manno et al. (Manno et al., 2012) also report that every cloud belonging to a federation is interpreted as an independent domain, with autonomy over their native computing assets and the free will to, at any time, leave the community.

Govil et al. (Chaurasiya, Srinivasan, Thyagarajan, Govil, & Das, 2012) argue that this organization arose from the union of service providers to make more resources available for their clients, and thereby to reduce problems related to non-compliance with SLAs. From the point of view of the authors, clouds organized in associations provide several advantages, among which: i) performance guarantee – through the use of resources "borrowed" from other clouds, the performance of services can be maintained; ii) guaranteed availability – the diversity of locations where clouds infrastructure are located allows the migration of services from areas that may be affected by outages (Amazon, 2011, 2012), maintaining the availability of consumer services; iii) convenience – the federation provides convenience for consumers in relation to contracted services, and they may see their various services in a unified manner; and iv) dynamic distribution of workload – due to geographical dispersion, it is possible to redirect workloads to clouds closer to customers.

This work uses a synthesis of the above definitions, also exploiting the fact that there is no impediment for specialized clouds from the same institution to constitute a federation. We propose the following definition:

Cloud Federation is a multiple cloud organization with a voluntary character. It should have a maximum geographical dispersion, a well-defined marketing system, and be regulated in terms of the Federative Agreement that determines the behavior of heterogeneous and autonomous clouds. This organization has to be able to provide effective resource scalability, ensure the performance of services, perform a dynamic allocation of resources present in the environment, and honor end-to-end SLA to consumers.

In the remainder of this section, we list the main reasons to the emergence of federations and challenges identified by several authors to design a final architecture of a Cloud Federation. Based on this, are describes so the functional and usage properties that must be present in this organization.

Motivation

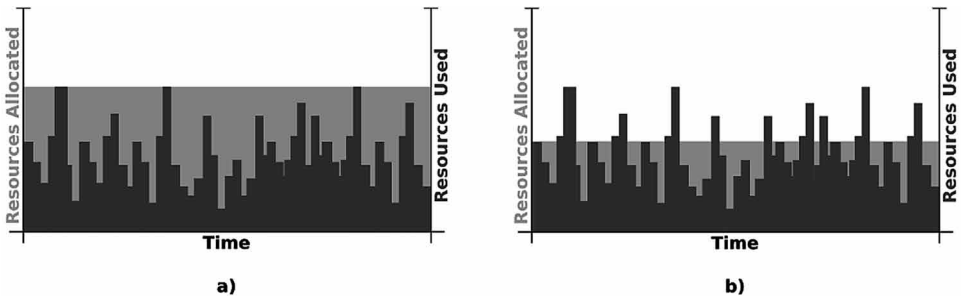
As discussed above, individual cloud providers cannot achieve certain properties from Cloud Computing paradigm. Even some multiple clouds solutions, such as Hybrid Clouds (Bittencourt & Madeira, 2011; Mell & Grance, 2009), are not able to meet all identified needs. This section aims to describe four major limitations that are presented as reasons for the emergence of Cloud Federations.

Resource Provisioning

The predicted amount of physical resources needed to support the applications is subject to variable behavior. This dynamicity is one of the main difficulties faced by resource management in this environment. There are two conservative approaches (Toosi, Calheiros, Thulasiram, & Buyya, 2011) used by providers to handle variable resource consumption: oversizing, or overprovisioning, and undersizing, or underprovisioning. In oversizing, the environment allocates beforehand a significant amount of resources to handle peak workload. Since demand can be variable, there may be a waste of resources most of the time (light gray area in Figure 4a) because many resources will be idle unnecessarily consuming energy (Armbrust et al., 2009). The second approach underestimates consumption, pre-allocating resources according to the average usage over time, thus not considering potential unexpected variations in the workload. In undersizing (Figure 4b) there is less waste of computational resources, but peak demand for resources can lead to degradation of service until the extra resources can be reactively allocated. This lack of prompt provisioning can lead to eminent losses that go beyond economic performance, such as loss of confidence in the service provision or in the provider itself.

One of the objectives in Cloud Computing is to dynamically solve (proactively or reactively) the resource provisioning problem (Zhang et al., 2010), offering the property called elasticity. However, the elasticity is performed on physical assets which in turn are finite. Small and medium providers are more sensitive because they have fewer assets, which can cause resource contention to new requests and hamper elasticity (Toosi et al., 2014). Providing mechanisms to mitigate this limitation is one of the motivations of the Cloud Federation (Hassan, Abdullah-Al-Wadud, & Fortino, 2015). Clouds organized in associations can offer their idle resources and/or request additional resources as needed to other members of the organization (Gomes, Vo, & Kowalczyk, 2012), which transcends the limits of local physical resources. Being governed by a contract, this sharing behaves obeying certain pre-established rules. Solutions such as the categorization of supply/resource consumption by clouds within the federation are proposed by Kecskemeti et al. (Kecskemeti et al., 2012; Marosi, Kecskemeti, Kert´esz, & Kacsuk, 2011). Gomes et al. (Gomes et al., 2012) propose the use of economic tools to provide context to resource sharing. Other authors (Mihailescu & Teo, 2010; Flake, Tacken, & Zoth, 2012) dynamically define resource prices to regulate supply and demand within the federation and the use of policies (Petri et al., 2014) to define sharing (Rochwerger et al., 2009). A reputation system can also be used to moderate sharing on multiple cloud organizations, especially in Cloud Federation. Hassan, Abdullah-Al-Wadud and Fortino (2015)

Figure 4. Chart depicting the variation from resource allocation to meet variable demands in workload: (a) represents oversizing of resources and (b), undersizing



interpret the provisioning of resources as a financial incentive problem, and they proposed a dynamic distribution mechanism of the profit earned in the federation to cloud providers.

Regional Workloads

The worldwide network of computers named as Internet is scattered globally. Services such as social networks and search engines run over it, benefitting from its reachability to aggregate the largest number of potential customers. In this scenario, the quality of services is affected by technical factors (e.g. latency) and usage factors, such as those based on consumer culture of certain regions. Popular services that have global range (users worldwide) are subject to local workload variations determined by regional events (in addition to the global ones). For example, the famous day of discounts in the US department stores, called Black Friday, can generate a significant increase in workload in the search services (e.g. Google) coming from the American consumers, while having little or no impact in the rest of the world. Another example is the possibility that some social networks offer to users to construct small applications (Buyya et al., 2010). Therefore, some may become very popular and used in specific regions, so it is better to have those applications running geographically close to users.

Due to technical factors (e.g. network latency), this phenomenon can lead to loss of quality in provisioning for services that do not have infrastructure close to consumers. Some providers use other strategies, such as the Amazon CloudFront (Amazon, 2014), which uses Points of Interest (PoI) distributed across the globe to assist the demand for regional workloads with low latency without the need to maintain their own infrastructure scattered. These POIs are implemented in leased data centers in a similar behavior to an organization of multiple clouds. Services such as Content Delivery Networks (CDNs) (Pathan, Buyya, & Vakali, 2008; Canali, Cardellini, Colajanni, & Lancellotti, 2008) also offer a similar solution to POIs, bringing the application closer to consumers.

Cloud Federations are designed taking geographic dispersion into account. This dispersion and other goals, such as resilience to natural disasters (Aoyama & Sakai, 2011) and costs mitigation (Le et al., 2011), focus on tackling with the regional workloads. As the federation is a well-behaved organization, defined in terms of a contract, the location of each cloud that composes the federation is known, which makes it possible to redirect workloads to partner clouds closer to regions of interest.

Economic Barriers

The services market forces institutions within the same niche to implement techniques and properties that differentiate them from others, pursuing a competitive advantage (Buyya, Pandey, & Vecchiola, 2012). Service providers, due to lack of standards, implement their own storage and operating mechanisms, as well as interfaces and access protocols. Such characteristic increases the data migration cost from one provider to another, limiting the freedom of customers by the lack of portability and creating a phenomenon called “lock-in” (Armbrust et al., 2009; Ardagna et al., 2012; Petcu, 2011; Petri et al., 2014). This is clear in the case where, due to economic opportunities offered by other providers, a customer cannot migrate her services because the cost to perform the conversion between data types is so high that migration becomes inviable (Kurze et al., 2011).

It is possible to discuss the lock-in phenomenon from the perspective of customers and that of the service providers. For customers, it is extremely important to mitigate the lock-in, since it forces them to come under the tutelage of a single provider technology and, therefore, be exposed to the costs and

directions that such provider thinks as appropriate. Moreover, this phenomenon contributes to increased caution of many institutions in adopting Cloud Computing. On the other hand, for the service providers the lock-in can be attractive, since the data implementation schemes, techniques, and standards themselves may create advantage over competitors (Toosi et al., 2014).

Kurze et al. (Kurze et al., 2011) describe a situation related to the lock-in that triggers another economic phenomenon in the context of isolated clouds. In this situation, an institution can establish a contract with a cloud provider to develop applications related to it. Hence, this institution needs to make an investment in technical knowledge and technological directives for that infrastructure in order to enable the development of the application. However, during the establishment of the contract not all aspects of the relationship institution-provider are raised. Given this situation, the provider can take advantage benefiting from previous investments made by the institution in relation to the technological direction for the development of its applications. In order to avoid such a scenario, interested parties may come into hold-up situations, generating underinvestment and consequently inefficient results for both sides.

Two properties inherent to the Cloud Federation enable stakeholders to mitigate the exposed economic barriers. Volunteering and Federation Level Agreement (FLA) enable the maintenance of interoperability within the organization. These properties lead the members of the federation to be willing to participate in the organization and to submit to pre-established interoperability standards.

Legal Issues

In the Cloud Computing paradigm, customer data can be stored in or travel to a location far from the user – a customer in Africa may be using a cloud allocated in the US, for example. After the terrorist attacks that occurred on the fateful 11 September 2011, several countries have imposed strict laws related to break of data confidentiality in their territory. The United States approved the US Patriot Act (Us Patriotic Act, 2001) and began to monitor cloud providers infrastructure located in its territory, as denounced the US National Security Agency (NSA, 2015) former contractor Edward Snowden. The possibility of monitoring and break of confidentiality motivated providers to migrate to countries where laws were milder in order to keep their business and to protect consumer data (Buyya, Broberg, & Goscinski, 2011).

On the customer side, some institutions, especially government entities, are subject to laws restricting the use of commercial Cloud Computing providers (Jeffery & Neidecker-Lutz, 2010) due to the characteristics presented in the previous paragraph. In Brazil, for instance, these features restrict the use of public clouds because the law prevents the Brazilian government data from leaving the country.

In federations, localities where data is or will travel can be defined in advance. This behavior can be used to overcome legal restrictions faced by some institutions. Returning to the case of Brazil, it would be possible to create a federation containing countries in Mercosul (Mercosur, 2015), where the data storage and traffic from Brazilian government would be limited to those clouds located within national boundaries.

Open Challenges

Several authors, as Toosi et al. (Toosi et al., 2014) describe a set of challenges inherent to the establishment of multiple cloud organizations. This section discusses these challenges, focusing solely on Cloud Federations.

Interoperability

Interoperability (Bernstein et al., 2009; Assis et al., 2014), the ability for systems to interact with each other, is a crucial aspect for forming a Cloud Federation. It involves the development of interaction protocols and interfaces that must be known in advance for all the interacting parties. The formation of a federation is possible only with the presence of such a property, as the federation is formed by a set of heterogeneous domains that can exchange information, resources, and services. According to Toosi et al. (Toosi et al., 2014) and Chen and Doumengts (Chen & Doumeingts, 2003), interoperability in multiple cloud organizations such as federations, can be implemented by using ontologies, brokers, or interfaces, which in Computing Cloud paradigm are exposed in most cases in the form of Application Programming Interfaces (APIs).

Ontologies (Manno et al., 2012) are representations of knowledge that can be used to provide interoperability without the need to explicitly implement the technologies used by delegating this responsibility to local contexts of the participants of the federation. Brokers (Kurze et al., 2011; Buyya et al., 2010; Villegas et al., 2012; Makkes et al., 2013; Marosi et al., 2011), in this context, are responsible for the intermediation of interactions between customers and providers, performing the translation of messages originating in and destined to providers. This mitigates the need of a single communication language/protocol, allowing the diversity within the federation. The interfaces provide a direct and controlled way of communication between different entities. Standard interfaces make the process of improvement and introduction of new features faster if compared to other approaches, because in most cases there is an exclusive working group to develop standard interfaces.

As to the limitations that make interoperability an open challenge, brokers add an extra layer in the composition of the federation, since all interactions with the environment are made through it. This type of interaction mechanism generates an overhead, being also a Single Point of Failure (SPOF). Therefore, if no replication strategy is implemented and the broker fails, the whole environment can stop working. On the other hand, the standard interfaces, even if bypassing the problems faced by brokers, are more difficult to be adopted by providers (Rochwerger et al., 2009), who are mostly interested in maintaining their own interfaces that reflect the implementation of various technologies (Petcu, Craciun, & Rak, 2011). Among the initiatives to implement standard interfaces, the Open Cloud Computing Interface (OCCI) (Nyr´een, Edmonds, Papaspyrou, & Metsch, 2011) is focused on remote management services (IaaS, PaaS and SaaS). The Distributed Management Task Force (DMTF, 2015) is an organization that also works for publication of standards and interface specifications in the cloud context, such as Cloud Management Working Group (CMWG) and the Cloud Infrastructure Management interface (CIMI) (CIMI, 2012), which respectively describe patterns of interaction between customers and providers, and IaaS resources management.

Portability is also important in the context of interoperability. In Federations of Clouds, constant migration of applications and customer data must be supported by other elements of the organization. In order to generate the portability of applications, it is necessary to list the types of services requested and the granularity of the elements that support the applications: Containers – e.g. LXC (LXC, 2015), Docker (Docker, 2015) –, virtual machines etc. Another element that enhances the difficulties of portability refers to the origin and destination domain. Items such as the local network settings and security constraints present in each domain can hamper migration between clouds. In his work about Sky Computing, Keahey et al. (Keahey et al., 2009) attack the problem of migration of virtual machines, highlighting that the main limitations to this migration are the diversity of VMs representation formats, which can cause

incompatibility during the change of context, which they are subject to. The authors propose the use of Virtual Appliances and context descriptors (e.g. metadata) to mitigate such difficulties. As an initiative to migration problems, the DMTF keeps the Open Virtualization Format (OVF, 2015), which describes how to package software to run in virtual environments.

Federated Identity Management

Identities management (IdMs) are entities present in the domain of a cloud provider (Toosi et al., 2014; Celesti, Tusa, Villari, & Puliafito, 2010; Dreo, Golling, Hommel, & Tietze, 2013). IdMs are responsible for the authentication process, and to enable the authorization process for access to resources. On multiple cloud organizations, such as federations, IdMs may need prepared to treat visitors (customers) that were not originally registered in their user base. This treatment consists in authentication to the visitor and to allow him access to resources of interest with certain policies that state the scope of access. In some multiple cloud organizations, the complexity to allow these actions is accentuated (Toosi et al., 2014) due to different IdMs implementations and standards of the authentication information representation. For example, X.509 certificates – RFC 4158 (RFC 4158, 2015) and RFC 5280 (RFC 5280, 2015) – and the Security Assertion Markup Language (SAML, 2015) depend on the local context and may compromise interoperability. The Organization for the Advancements of Structure Information Standard (OASIS) maintains a working group to standardize identity management in Cloud Computing. As a result, this group published the Identity in the Cloud Use Cases (OASIS, 2012), presenting use cases that describe the identity management between clouds.

As described in (Toosi et al., 2014), federated identity management can enable Single Sign-On (SSO) whereby after a single authentication, a user can gain access to multiple systems. Different approaches are possible for the implementation of SSO: global user and identity provides (Makkes et al., 2013). The first approach is the simplest: the existence of a standard user registered in all user bases of the clouds in the federation, and any action related to the consumption of a foreign resource[2] is performed by this user. Although this approach is the simplest to implement, using a single user can add a single point of security failure, since with the acquisition of authentication properties of this user, a malicious user has access to all clouds in the federation. The use of global users also introduces difficulties in tracing the origin of the consumer of foreign resources, and encumbers the process of accounting and charging the use of foreign resources between clouds.

Another approach is to delegate authentication to Identity Provides (IdPs) (Celesti, Tusa, Villari, & Puliafito, 2011). IdPs are specialized institutions in IdM and can be public or private. Public IdPs are open to all interested parties, implement open authentication standard protocols – OpenID (OpenID, 2015), Oauth (Oauth, 2015) etc. –, and are a solution to normalize the authentication process in environments such as Cloud Federations. This normalization is achieved by the delegation of the authentication process by the federation to the selected IdP. However, IdPs may also be focus of malicious attacks, since they are centralized and with sensitive information. Moreover, they can result in an increase in cost to the authentication process, as well as introduce authentication lock-in, because the authentication data will be delegated to a service provider that in turn can direct its own policies. A second model is targeted to specific clients, which have a well-defined niche, such as those formed by commercial institutions that do not expect their databases to be exposed, or by classes of workers within a certain niche with compatible trade relations. In addition to the lock-in, private external IdPs add to the federation a non-negligible cost.

Services Management

Service management comprises the discovery and mapping of services offered by the clouds in the federation and the presentation thereof to stakeholders. Manno et al. (Manno, Smari, & Spalazzi, 2012) reported that in a federative environment is essential to have a mechanism for service discovery, which should allow internal or external entities to request the availability and types of service provided. Also, it should be considered that each Cloud Federation is independent and have control of their own resources, therefore some clouds from a federation may be not interested in publishing all the services they offer. Toosi et al. (Toosi et al., 2014) report that there is a lack of publishing methods and standards designed for multiple cloud organizations. This arises from the heterogeneity of publishing methods and the lack of expressiveness of these methods. However, in environments such as federations, where the clouds are well behaved considering the methods they implement (because they are subject to an FLA), it is possible to mitigate these problems. Thus, the question shifts from implementation methods to choosing the most efficient of them, because all the clouds will be subject to the chosen method.

Data communication that occurs outside of a single domain can be a bottleneck, since exchange of messages among multiple clouds mainly uses shared/untrusted networks with high latency, e.g. the Internet. Some solutions, such as the use of repositories and catalogs, are presented as centralized approaches where clouds proactively or reactively perform the publication of the resources made available to interested parties, as in a planned approach in publishing protocol of the Service Oriented Architecture (SOA) (The Open Group, 2009).

Contract Maintenance

Among domains, the contract between the federation and the clouds that compose it is the main item that distinguishes this organization from other multiple clouds. Extending the initial concept of the FLA defined by Toosi et al. (Toosi et al., 2011), such document actually acts as an internal SLA, stipulating how clouds should behave in the environment. The FLA should describe the mode of provisioning of resources from the clouds to the federation, that is, if every cloud reserves a fixed amount of resources to the environment or if the resources are dynamically offered without pre-fixed quantities. This contract should also define items that determine the quality of the use of the offered resources. This is necessary because clouds are heterogeneous, and each one has a certain amount and types of computing assets, in most cases diverse, and different procedures to access them.

As within SLAs, penalties should also be contained in the FLA. These penalties can occur if the clouds do not comply with contract items such as the provision of predetermined features. These penalties may reflect the orchestration held in the federation. Clouds with penalties history can be discarded in the feature selection process and, if penalties are recurrent, such clouds can even be disassociated from the federation. Bernsmed et al. (Bernsmed, Jaatun, Meland, & Undheim, 2011) state that the SLA must present mechanisms to the user describing criteria for security items, i.e., introducing the concept of Quality of Protection (QoP), which can also be extended to the Federation Level Agreement.

There are few studies about contracts in the Federations of Clouds, as there is a lack of research in this area in monolithic Cloud Computing environments. As a result of this gap, the contracts are not yet mature (Cloud Standards Customer Council Workgroup, 2012) to express the Service Level Specifications (service delivery restrictions, QoS, and penalties) (Bernsmed et al., 2011) at the time of translation of textual description (semantics) to logical description that can be interpreted automatically within

the environment. Some initiatives, such as Patel et al. (Patel, Ranabahu, & Sheth, 2009), propose the utilization of specifications created for other contexts, such as WS-Agreement (Andrieux et al., 2005) and Web Service Level Agreement (WSLA) (Keller & Ludwig, 2003) implemented in representative languages – (JavaScript Object Notation (JSON, 2015) and eXtensible Markup Language (XML, 2015) –, to manage SLAs in clouds. Such initiatives can be adapted to perform the translation and representation of the FLA at a level that the federation is able to understand and execute. Another challenge is to maintain a rigorous policing (Emeakaroha et al., 2012) of the FLA. This policing is only possible through efficient capture of items of interest by the monitoring system of the federation. Comuzzi et al. (Comuzzi, Kotsokalis, Spanoudakis, & Yahyapour, 2009) propose an architecture for SLA monitoring in monolithic clouds that can be used to monitor FLAs.

Providers Behavior

As described by Toosi et al. (Toosi et al., 2011), each cloud from the providers that form a federation are autonomous and can have a customer portfolio that has nothing in common with the others. Attending these customers can restrict the provision of computational resources that each cloud can provision to the federation. Moreover, apart from institutions with scientific focus, the associations may be established by private institutions aimed at maximizing the profitability by selling their idle resources and using foreign resources that are offered at more attractive prices for the internal members of the organization (Toosi et al., 2011). Maximizing profitability may lead some clouds to only request resources, and not offer them to the federation.

Finding ways to avoid this problem leads to the analysis of some challenges, particularly service management. The supply management of resources by constituent clouds is critical in organizations such as federations, since mismanagement can result in environmental degradation. In this context, three scenarios arise which show the main behaviors that require uplifting the association of regulatory mechanisms.

The first scenario is related to setting in the FLA some explicit limits for the provision of resources from each Cloud Federation. The definition of such limits can lead to problems such as resource idleness; thereby the respective clouds of origin mostly supply the amount of workload. For example, suppose that in a hypothetical Cloud Federation each clouds must provide 25% of their resources to the organization, but each cloud is operating at 70% of its capacity. If a cloud cannot have its own management policy to freely turn off its resources because of the FLA, this cloud would have 25% idleness of computing assets. This may generate unnecessary expenses (power, cooling, maintenance) and consequently a lack of interest in using this type of organization. Furthermore, in certain situations to meet its own consumers' workloads, some clouds may be required to request resources from the federation to meet the local workload demand even if it has idle resources in their organizations. These resources could not be used directly as they are reserved to the federation. This problem is described in the example where a cloud belonging to a federation has 30% of its local resources reserved to the federation environment, and at some point it faces a sudden increase in workload that consumes 80% of its assets. In this situation, the cloud will be required to request to the federation more resources even though it has sufficient resources to meet the demand within its own domain.

A second scenario considers the extinction of such a explicit designation of a minimum amount of resources to be offered for the federation. This strategy alleviates the idle problem from the previous scenario, but can provide the possibility of some of the clouds not to offer resources proportionally to the environment, consuming much more than it offers. When there is no explicit determination in the FLA

of the resources that every cloud should offer to the federation, a cloud with a high number of customers could demand an increasing amount of resources from the federation to serve them taking advantage of economic factors (attractive prices, for example) or technical factors (specialized resources, optimized communication channels etc.). Due to its high workload, this cloud could deny the supply of resources to any request from the federation. The recurrence of this behavior from a cloud provider could make it undesirable in the environment, contributing to the de-characterization of the organization, since strict consumption has similar behavior to Hybrid Clouds, for example, and not a Cloud Federation.

Another scenario, which is in the scope of resource management in the federation, refers to the administration of clouds life cycle in the organization. In order to try to mitigate problems related to the withholding of resources, a federation should be able to delete or quarantine[3] clouds that do not honor the basic principles of supply and demand of resources.

There are no works in the current literature that clearly specify methods for the acquisition or exclusion of clouds within the federation environment, as well as the management of available resources. This management ends up being done manually through relationships between institutions. Such methods can cause waste of human resources and loss of resources when clouds are excluded. In the latter, the amount of resources lost is directly linked to the temporality of detection of those clouds that are only consuming and are not contributing to the environment.

Monitoring

The monitoring in a distributed environment, such as the Cloud Federation, is extremely important because it will provide information for the maintenance of the organization. As described by Toosi et al. (Toosi et al., 2014), in environments such as federations, diversity of components (e.g. the Orchestrator) and features (elasticity, high availability, etc.) may increase the complexity and the need to maintain a monitoring system. In federations, monitoring can be divided into two distinct groups considering the object of interest (Al-Hazmi et al., 2012): monitoring the federation and monitoring applications. The monitoring of the federation, as the Lattice (Clayman et al., 2010), should be able to check the status and the alignment of each cloud to the FLA, as well as the components that are part of the federation infrastructure (communication channel, resources utilization etc.) with maximum temporality (Flake et al., 2012) and with minimal impact on the communication network. For the federation infrastructure elements this task is trivial from the point of view of tools and implementation, but it becomes difficult when the clouds belonging to the organization are considered. Providers are independent and heterogeneous and may have systems to monitor their own infrastructure, so it is necessary the monitoring of the federation to be able to perform communication and the exchange of information between the various monitoring systems that can coexist in the environment. Faced with this challenge, some solutions can be implemented, such as the development of interfaces for interoperability to perform data exchange between different monitoring systems, and the implementation of an appropriate mechanism to communicate with the most popular proprietary monitoring tools – HP Openview (HP, 2015) for example – or free monitoring tools–Ganglia (Ganglia, 2015), Nagios (Nagios, 2015), Zabbix (Zabbix, 2015)–with low level intrusion. A more restrictive approach can also be assumed, creating a compatibility matrix in the FLA with monitoring systems supported by the federation.

When considering applications in the environment, especially those implemented in multi-tier, the monitoring of the properties related to them can be offered to customers as a service – Monitoring as a Service (MaaS). AlHazmi et al. proposed the BonFIRE (Al-Hazmi et al., 2012), which provides global

monitoring of the federation and also offers MaaS to interested parties. MaaS can use the federation monitoring information to tracking of resource utilization by customers and their applications. Seo, Kim, Cui, Seo & Lee (2015) proposed a solution based on aggregation to assist administrators in monitoring resources and services used by users in a federated environment. As SLAs may be binding between customers and certain clouds of the organization (transparent federation) or directly to the federation, the two modes of supplying Monitoring as a Service must be considered. In the first way, the customer does not know the existence of the federation and hires the services of the specific provider. Considering MaaS, in this scenario, the provider, through the FLA, knows the details of the global monitoring system which in turn can be used to trace the execution of the customer application and if it will use other service providers from the federation. The second way comprises the establishment of a contract between the customer and the Cloud Federation, where the Monitoring as a Service can be delegated directly to a central component of the federation and not to the clouds that compose it, which should be a global monitoring system.

Three characteristics directly impact in the efficiency and effectiveness of monitoring systems in a federation: temporality, diversity, and scalability. The first two influences on the amount of messages generated in the network: decreasing the time window to obtain information and increasing the diversity of components and monitoring systems in the network generates more messages to the shared federation network. Such messages can be influenced by network latency as well as contribute to the increase of this latency. Solutions such as implementing more efficient protocols (pooling or publishing), segmented networks only for monitoring and calculation periods, or collection/publication of data can be useful in this context. About scalability, in an environment such as the one considered in this work, association and disassociation of clouds belonging to the federation are expected to occur. Thus, the global monitoring system must be able to efficiently scale according to the fluctuation of these occurrences. If this is not possible, the monitoring system may use outdated information, which can compromise other systems that depend on this information, such as scheduling or orchestration of components that need to know the actual state of the Cloud Federation, as for example when building dynamic workflows[4]. From the point of view of MaaS, there may also be fluctuations in relation to systems that each cloud uses within its domains to monitor their environments. Both the global monitoring and the MaaS should be able to provide support to new monitoring system when necessary. Solutions based on the Advanced Message Queuing Protocol (AMQP) (Godfrey, Ingham, & Schloming, 2012) – Apache Qpid (Qpid, 2015), zeroMQ (zeroMQ) etc. – can be used to automate the insertion of new monitoring systems within the federation.

Business Model

In an organization of multiple clouds as the federation, it is possible to create new marketing models as well as to adapt existing models used in other contexts. The framework proposed by Buyya et al. (Buyya et al., 2010) uses a market-oriented model (Buyya, Yeo, & Venugopal, 2009; Petri et al., 2015) as resource trading system in the Federation. In this marketing model, a central entity acts as a point of concentration and through the supply and demand of the federation's resources in terms of a product market. Another marketing model is the offering of specialized resources by certain providers within the federation. In this model, some clouds within the federation are specialized in certain types of resources, and thus the customers can hire differentiated resources to run their applications or parts thereof. Among the benefits of this model, are: the high degree of specialization of certain resources, resources differentiated costs, explicit separation of responsibilities, and improvement of customer experience by

optimizing the operation of their applications. Moreover, in this model, applications should be prepared (as a TIER model, for example) to use segmented features on more than one cloud.

Considering only the customers, it is also possible to abstract the Cloud Federation from the commercial context and use it only as an improved infrastructure provider for elasticity and geographical dispersion to the interested parties. Thus, customers could use the marketing model of monolithic clouds and negotiate directly with them as in a decentralized architecture. In this approach customers are unaware that the cloud they interact is part of a federation, and the clouds would hold negotiations with others transparently to the allocation of foreign resources, if they are necessary. Thus, only clouds in the organization would benefit from the lucrative aspect of the federation, offering their idle resources and obtaining resources with more attractive prices as compared to the public cloud model, as exemplified by Petri et al. (Petri et al., 2014). Similarly to a resource market model, it is also possible in a centralized architecture to make the supply and demand concentration, omitting the party responsible by the auction.

In the contract present in the federation, it should be specified the marketing of services model, i.e., charging and service models, of the services offered by the clouds. In SLAs between customers and providers, the business model should be described because it directly impacts on how resources in the federation are hired and charged with customers. In the FLA, the business model specification influences how orchestration systems perform their tasks, such as resource allocation and accounting of consumption.

The monetary value of the resources contained in the clouds are subject to market variations and the costs of the infrastructure maintenance and management where they are allocated, as well as variable according to utility and local supplies prices (electricity, water for cooling etc.). Given these two regulators, it is necessary to define economic criteria regarding price fluctuations within the federative environment because large variations may impact the integrity of the Cloud Federation. This is clear in two extremes, when a particular provider A has a high maintenance cost of its own cloud and dilutes this value in the final price of the offered resources, while another provider B has a very low supplies cost due to a privileged location, and can reduce resource prices. In this scenario, and considering a business model, there would be a very high demand from the federation clients for resources offered by B and lower demand for resources available in A, which consequently unbalances the federation.

Obtaining an embracing business model in all contexts as well as polices for the availability and prices of resources within the federation remains an open challenge.

Orchestration

In multiple cloud organizations, orchestration consists in receiving an application and distributing it over the environment considering pre-established criteria by customers and/or providers. Performing this distribution includes the selection and allocation of the best service providers available that meet customer/application needs. Moreover, from the providers' point of view, they want to optimize the use of resources and reduce costs of their own infrastructure (Toosi et al., 2014). In organizations like Cloud Federations, factors such as diversity of quality of services, price of resources, geographic dispersion and network latency between providers directly impact the selection and allocation of services required by customers. Toosi et al. (Toosi et al., 2014) mention that it is necessary to implement automatic methods for deploying applications that optimize the various dynamic factors to which federations are exposed, as the latency and throughput in data transfers, in addition to considering constants like legal and security constraints. Le et al. (Le et al., 2011) propose a solution where through policies; workloads are migrated from region to region within the federation, according to the cost of use (cost of energy, cooling etc.).

The orchestration is associated with other properties of the Cloud Federation, mainly related to aspects of interaction (centralized or peer-to-peer) and visibility (translucent and transparent) to the customer. In federations, translucent centralized selection of services and resources can be performed directly by customers or automatically by the organization through a set of criteria established in the SLA. Concentrator service offerings, as the market model, enable customers to analyze the providers who fulfill their momentary needs. Customers then become responsible for the selection and distribution of their applications over the chosen providers. In another federation architecture, a service of the organization itself performs this task automatically. Regarding the automatic process, there are solutions that use monitoring to perform data collection and analysis on which providers can meet SLAs required by customers (Cuomo et al., 2013).

In the orchestration topic, the Topology and Orchestration Specification for Cloud Application (TOSCA) (TOSCA, 2013) is an OASIS working group focused on developing a document called Server Template, which describes the topologies and deployment procedures, implementation, and management services. This document can be used to soften the orchestration task in federations, where the support of Server Template abstracts cloud providers and the technologies used for the implementation and execution of services.

Use Cases

Clouds Federation associations can be utilized in different situations. This flexibility is one of the characteristics that make them attractive when compared with other Inter-Clouds organizations. Among the scenarios that federations can be used, we highlight the following ones:

- **To Increase the Profitability of Cloud Providers:** federations can be used to increase the profitability of cloud providers (Gori, Guilart, & Torres, 2010). In this situation providers can obtain resources with more attractive prices, as well as market their own idle resources to the other members of the federation, thus increasing revenue.
- **To Maintain the Resilience of Services:** Aoyama and Sakai (2011) apply the federation of clouds to implement a monitoring system and responses to natural disasters. Benefiting from the geographic distribution of the cloud that a federation provides, disrupted services from locations affected by disasters are supplied by other providers in non-affected locations.
- **To Avoid Lock-In:** a cloud federation allows customers to migrate to other providers when convenient with minimal financial impact and technical difficulty.
- **To Lower the Cost to Customers:** through federation that are directed to the resource market (Buyya et al., 2010), customers can select the resources and services that best suit their needs with the cost they are prepared to pay.
- **To Process Hard Problems:** applications that require enhanced computing capacity can use the cloud federations to scatter application components through various providers, with the same SLA, to improve response time.

CLOUD FEDERATIONS PROPERTIES

There exist a variety of definitions related to concepts and their semantics in Cloud Computing. Many concepts are defined more than once, which impairs their understanding and can hamper attractiveness of this technology for the general user. This lack of definition is also true to organization of clouds as the Cloud Federations. Based on concepts learnt and extracted from the state-of-art research on Cloud Federation, such as the proposal by Toosi et al. (Toosi et al., 2014) for generic multiple cloud organizations, this section aims at describing properties specifically from Cloud Federations, separating them into *Functional Properties* and *Usage Properties*, detailed below.

Functional Properties

In this work, functional properties proposed in GICTF 2010 (GICT, 2010) are highlighted. They are presented in Table 1, and are detailed in the next sections.

Authentication

In Cloud Federations, there is a frequent consumption of foreign resources, i.e., resources from other participants in the federation. In order to enable such a utilization of foreign resources, users need to obtain access credentials to the relevant foreign domains, which usually do not have his/her authentication information. In this context, some solutions are highlighted in section *Global Identity Management: Global User and IdPs.*

Commercialization

Federation models must foresee how services will be commercialized with their peers. This kind of organization supports the adoption of a fixed commercial model, defined as a contract, or provides free choice to the federated providers. In this first mode (fixed), there is a consensus among organization

Table 1. Functional properties related to Cloud Federations derived from those defined for multiple clouds and presented in GICTF 2010

FUNCTIONAL PROPERTIES
Authentication – authentication mechanisms for users of organizations participating in the federation.
Contracts – support for service contracts as well as environment contracts (rules to participate in the federation).
Commercialization Model – commercialization model of services in the federation.
Integrity – integrity maintenance regarding resource offer and demand.
Interoperability – data exchange among clouds in the federation.
Monitoring – environment monitoring, including offered services and contracts.
Object – what is the object of commercialization.
Provisioning – resources provisioning, considering consumers and federation environment requirements.
Service Management – management of services offered in the environment.

members on how commercialization is performed, since it is defined in the FLA. This commercialization can be marked-driven, where providers publish their offers in a central entity, and consumers interact with this entity to check prices and post proposals/requests. In this scenario, the auctioneer may or may not be present, who is responsible for matching offers and proposals. When no auctioneer is present, the client himself verifies prices and providers that better fits his needs. In the second mode (free choice), providers have the autonomy to decide their own commercialization model, different (or not) from the other participants in the federation. In this case, the federation acts more as an extension of each provider's infrastructure. The RESERVOIR proposal, described in the next section, utilizes this model.

Contracts

SLAs are contracts between providers and consumers that act as a guarantee of service fulfillment to the users (Buyya et al., 2012). These contracts contain technical and administrative details regarding contracted services. The technical details section of the contract, called Service Level Specification (SLS), describes the quality of service, penalties for violations of contract terms, and how services are delivered to consumers.

In Cloud Federations, in addition to customers SLAs, there is also the federation level agreement, which regulates and maintains the integrity of the federation. Details of functional and usage properties are described in this document, which serves as a basis for the federation activities.

Integrity

Integrity is the functional property that describes the consistency of the environment in what regards offer and demand of resources by providers in the federation. As mentioned earlier, environments where no such regulation mechanism exists are prone to suffer from lack of resources and, at the end of the day, the federation can have its purpose questioned by its participants. This situation can be illustrated in the scenario where a provider offers low-cost resources, but other providers only take advantage of those resources without offering their own share of services.

In order to maintain the federative organization characterization, a management process is needed by the providers. This process can be executed manually by a designated federation administrative board, or by an automatic process that triggers administrative actions/sanctions when anomalies are detected. In both cases, indicators that help in observing cloud participants performance within the FLA are needed to back up administrative decisions. Common indicators are rankings, rewards, and reputation. Ranking uses a score to classify providers according to the relation offer/demand; reward systems is an incentive mechanism to enhance resource offering, which in general attaches resource offering with advantages when using foreign resources; and reputation considers a history of a provider to generate an index that reflects its behavior in the federation.

Interoperability

In a distributed heterogeneous system, interoperability mechanisms are of paramount importance to perform data exchange between different domains. According to solutions presented in section *Cloud Federations: Open Challenges: Interoperability*, there are three strategies to achieve interoperability in

a federation: broker, ontology, and standard interfaces. The first one can be easier to implement, but introduces a new layer into the federation. The second is at a conceptual level, and delegates the implementation mechanisms to third parties. Lastly, the standard interfaces can provide better performance, reducing overheads, but may be harder to implement in commercial federations.

Monitoring

As highlighted in section *Cloud Federations: Open Challenges: Monitoring*, two types of monitoring can co-exist: global and MaaS. Global monitoring is focused on the maintenance of the federative organization, while MaaS aims at providing consumers with information to track contracted services. MaaS can rely of global monitoring services in order to reduce implementation efforts.

Objects

The marketing object is the smallest unity of service that a service provider can offer. This object will pass through the federation when resource consumption is needed. Thus, migration mechanisms must be taken into account. This objects can be organized into delivery models: Infrastructure, Platform, and Software as a Service.

Infrastructure as a Service (IaaS) clouds can offer physical resources (bare metal), virtual machines, virtual appliances, and containers. Platform as a Service (PaaS) development frameworks and tools can be delivered through virtual machines as well as using their packing/distribution methods, such as the cartridges utilized in the OpenShift solution. In the Software as a Service class (SaaS), the commercialized service is the access to that specific software.

Provisioning

Provisioning consists of the distribution of application coordinators (or part of them) to consumers through federation providers. This provisioning considers the installation and migration of application components and can be performed in two modes: automatically or manually. In the automatic way, an entity within the federation chooses the best providers for application installation or migration, according to SLA requirements. This can also be done manually, where a system operator performs provider selection and application installation.

Service Management

Service management is responsible for discovery and publishing of services offered by the federation members. Service discovery can be performed by the federation through pooling mechanisms, where consults to the federated clouds are performed regularly. The pooling frequency must be defined to reduce network traffic, but also to capture the dynamicity of the federation.

Service management is also responsible for publishing services throughout the federation members. This service publication can be done by any federation component after running a service discovery, resulting into an indirect publication. If the provider who offers the service publishes it, then we have a direct publication.

Table 2. Six main usage properties of a federation: visibility, interaction, centric, volunteer, practice niche and expansion

USAGE PROPERTIES
Centric – the focus of the Cloud Federation.
Expansion – the way federations expand on in relation to the services offered.
Interaction – the interaction architecture within the federation.
Practice Niche – acting area of the federation.
Visibility – the way customers see and use the Cloud Federation.
Volunteer – voluntary level of clouds contained in the organization.

Usage Properties

In this section, the properties that must be present in federations related to environmental usage from the customers' perspective are presented. Table 2 lists six properties that were highlighted in the literature.

Centric

Recently, proposed approaches in Cloud Federation (next section) have the focus on implementation and usability in certain elements of their respective architectures. This feature is called centric and in this work four centrics were identified: customer, business, provider and service. In the first, all the architecture and federation support mechanisms are designed prioritizing the customers leaving the others in the background actors. Business Centric makes the federation focuses on monetization thus federations focused on eScience not fit into this classification. In the Provider Centric the use of resources and services providers are maximized in detriment of other authors. Finally, in Service Centric the types of services and their specialities are treated as references to the architecture. Architectures with this focus may be prepared to offer homogeneous services considering IaaS, PaaS, and SaaS or heterogeneous, differentiated by diversity and specialty in certain features.

Expansion

Expansion property reflects how a federation uses the resources and services available in the environment. This property is based on the work by Celesti et al. (Celeti et al., 2010), which describes the possibility of expanding a Cloud Federation horizontally and vertically (in addition to those expanding in both modes – hybrid), and working of Bermbach et al. (Bermbach, Kurze, & Tai, 2013). The horizontal federations expand in relation to the same class of service (IaaS, PaaS and SaaS). When considering IaaS, the expansion can be used to provide redundancy and parallelism. When the class is SaaS the expansion can be used to improve the Quality of Experience (QoE) of use or encourage independence from providers (e.g. mitigate the lock-in). The vertical expansion uses all service classes in the environment. In this scenario a SaaS service can use the infrastructure of other federation providers, for example. Finally, the hybrid federations perform horizontal and vertical expansion in accordance with the interest of both customers and providers.

Interaction Architecture

In the Federations of Clouds the interaction of customers (providers of clients) with the organization can be performed centrally (Manno et al., 2012; Buyya et al., 2010) through a single access point, or decentralized (Petri et al., 2014), where every cloud belonging to the federation is a gateway. In centralized interaction architecture, a common approach is to use brokers to mediate the interaction of stakeholders with the rest of the organization. In the decentralized architecture (peer-to-peer), where users interact directly with the clouds, standards or ontologies are often used.

In both architectures actors that interact with the federation are the providers and customers. Considering this last, as the visibility property is inherent to them, the relationship visibility and interaction architecture affects how customers interact with the organization. In transparent federation customers are unaware of the existence of the Cloud Federation, so they interact indirectly with the federation through those providers with whom they have contact. However, providers that are part of the organization are aware of the existence of the federation and interact directly with it. In the translucent federation customers interact directly with the organization because they are aware of their existence, whether organized centrally or decentralized.

Practice Niche

Federations of Clouds operate in different niches, among them stand out from the commercial and non-commercial. In the first group, private, public or hybrid clouds with commercial nature can be found, where the clouds intend to use the organization to increase their respective revenue by selling idle resources as well as to acquire foreign resources with more attractive prices and conditions. As for the non-commercial associations, those where there is no explicit monetary profitability within the organization can be found. In this group, government federations are formed to mitigate, among other factors, legal restrictions, whereas scientific federations consists of research institutions that aim at sharing of computational resources for research.

Visibility

The visibility of the federations is a usage property that determines how customers interpret the organization. It is possible to view the federations translucently as an organization of multiple clouds that can be seen by the user, or transparently as an independent monolithic cloud. In the first mode (translucent), customers explicitly use the benefits of the infrastructure of the federations (e.g. real elasticity) and the economic potential they provide with knowledge of the federation components, such as in a services market. In transparent mode, customers interact with the elements of the federation as a monolithic cloud, whereas cloud components interact among themselves, without user knowledge, to make use of the benefits that the federation infrastructure offers.

Volunteer

Organizations like the federations proposed by Grozev and Buyya (Grozev & Buyya, 2012) must be voluntary, i.e. all the elements involved must have some level of knowledge that they are part of the organization (Assis & Bittencourt, 2015). Moreover, clouds should be able to leave the federation as

soon as they want, according to the established agreement (i.e. the FLA). Volunteer is an abstract property and its exposition can vary from solution to solution, as illustrated by the architectures presented in next section. This abstraction makes it difficult to, at a first sight, identify when an architecture has this voluntary characteristic or not.

Finally, in order to highlight and summarize the properties described above and in order to foster their readability and understanding, we provide a diagram, depicted in Figure 5, which covers both the functional and usage properties.

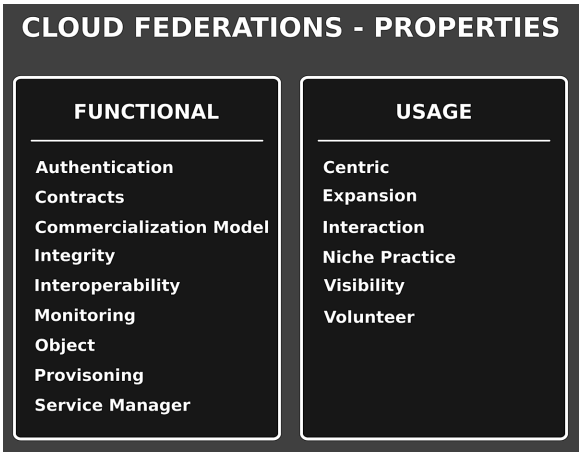
ARCHITECTURAL SPECIFICATIONS, BLUEPRINTS, AND EXISTING SYSTEMS

Recently, several architectures have been proposed aiming at the creation and formalization of cloud federations. This section presents prominent architectures published in the literature representing the state of art of federation models.

Contrail

The Contrail (Contrail, 2015) project was a European Union funded project executed from 2010 to 2014. Contrail was designed to support the integration of a number of independent clouds, forming an integrated federation, a combination of a number of independent clouds into one integrated federated cloud (Harsh et al., 2011; Copolla et al., 2012). The main objective of Contrail was to bring different cloud providers under the umbrella of a true federation, where users have the option to deploy services of multiple providers transparently and homogeneously. In order to achieve such an objective, a Contrail federation manages users identities, coordinates application deployment and the SLA management conducted by single cloud providers. Unique API, billing and monitoring capabilities were provided regardless of the nature of the heterogeneous infrastructures of the federation. Moreover, a Contrail federation is said to be horizontal and vertical. Horizontal, as different IaaS providers (for instance, both public and private

Figure 5. Functional and usage properties of Cloud Federations



clouds) can become part of the federation. Whereas, vertical integration is achieved by developing both the Infrastructure- and the Platform-as-a-Service architectural tiers.

mOSAIC

The Open-source API and Platform for Multiple Clouds (mOSAIC) is also a European initiative that tried to solve the challenges of cloud federations (executed from 2010 to 2013). The focus os mOSAIC was to offer a solution for application portability and interoperability across multiple clouds (Petcu et al., 2013). The emphasis of mOSAIC was on data intensive applications, though other objectives also include governance and security. The system architecture was designed with a language- and vendor-agnostic application-programming interface for accessing multiple clouds homogenously. In order to enforce the SLA of applications, it also incorporates user-centric service level agreements, a cloud ontology, and mechanisms for dynamic negotiation of resources based on multi-agent technologies and semantic data processing.

IEEE P2302

Cloud Federation has many areas that can benefit from standardization. Hence, the IEEE initiated the P2302 Standard for Intercloud Interoperability and Federation project. (P2302, 2015) The initial goal for this effort was to define topologies, functions, and governance for interoperability among Cloud Federations. To facilitate the development of a deeper understanding of practical federation challenges and approaches, the IEEE Intercloud Testbed project was concomitantly started. (Intercloud Testbed, 2015). From the many federation challenges, the IEEE Intercloud Testbed has focused on federated network communications management (Bernstein & Vij, 2104). Possible federated relationships are identified through a signaling protocol. Once this is established, a bearer network can be set up. This architecture leverages software-defined networks and virtual private networks through an Intercloud Federation API.

OpenStack Keystone

OpenStack is a large, open source software project that is creating a suite of services for Cloud Computing (OpenStack, 2015). The core services include computing (Nova), object storage (Swift), image storage (Glance), and identity (Keystone). To support the business models of cloud-bursting (Hybrid Clouds) and cloud brokering, the Keystone team has been developing critical capabilities for federation management. Concerning federated identity management, Keystone has implemented an attribute mapping capability whereby identity credentials from different identity providers can be mapped to attributes that are locally understood (Chadwick et al., 2013). Keystone has also added two calls to the API whereby trusted Identity Providers and trusted Service Providers can be explicitly specified. This is called “federating in” and “federating out”, respectively. While this relies on trust relationships to be managed by cloud administrators out-of-band, it nonetheless enables fundamental pair-wise federation management. Such federations can be symmetric or asymmetric. In a symmetric federation, users from both sites can use the other’s resources. In an asymmetric federation, as in a Hybrid Cloud scenario, users from only one site can use the other’s resources.

Massachusetts Open Cloud

Notably the Massachusetts Open Cloud project (MOC, 2015) is deploying this capability to manage a collaborative federation initially among Boston University, Harvard, UMass Amherst, MIT and Northeastern University. Users from different institutions will be able to instantiate VMs at one site that can access storage containers at a different site, in support of "big data" science projects. Ultimately, though, the goal is to achieve an Open Cloud eXchange (OCX) where cloud consumers can discover and use resources from multiple providers (Bestravros & Krieger, 2014).

EGI

The European Grid Infrastructure (EGI) is also deploying a set of federated cloud resources (Sipos et al. 2013). While EGI and grid community originally endeavored to deploy a global infrastructure to support "big science", EGI is deploying these cloud resources for all the same reasons as industry, e.g., ease of access to elastic compute resources. EGI can act as a cloud broker for resources from many of the member nations. Users can also instantiate VMs with the traditional grid software stack to support legacy grid applications. It is important to realize, however, that federation was central to the original grid concept. Hence, EGI is able to leverage many existing tools and capabilities to support Cloud Federation. For example, core EGI services for service discovery, monitoring and accounting using standards developed in the Open Grid Forum are all applicable. The Virtual Organization (VO) concept was also developed in grid computing as a way to manage federations. VO membership was managed by obtaining a PKI proxy certificate that augmented a user's identity with VO-specific attributes used to manage authorizations within a VO. This existing VO capability has been integrated with the OpenStack Keystone service (Garcia & Puel, 2013).

Broker Multi-Clouds

In the work by Kurze et al. (Kurze et al., 2011) the authors described and discussed various concepts (redundancy, migration etc.) as well as scenarios related to the use of multiple clouds associations to supply mainly emerging economic problems of Cloud Computing paradigm such as Lock-in and Hold-up. The main contribution was the presentation of a centralized reference architecture based on open source cloud management systems and multi-cloud libraries (Kurze et al., 2011; Grozev & Buyya, 2012; Toosi et al., 2014), where it is highlighted a federal layer represented by a broker. This broker is assigned to perform actions on the resources of multiple domains and is located between the business logic and computing assets contained in the fields it has visibility.

Oriented to Computing Service

The architecture proposed by Buyya et al. (Buyya et al., 2010) focused on IaaS services. The authors were motivated mainly by two challenges of the inherent heterogeneity of both the quantity and the types of services that run on Clouds: the prediction of workload variability and the contracted QoS. It is a centralized federation approach that indirectly interacts with customers through a Broker. One of the highlights of this proposal is the business model called driven market (Buyya et al., 2012). In this model

the clouds composing the federation publish their services and their values in a component responsible for offering it to the interested parties. One can also offer bids for services in a scenario similar to an auction.

Oriented to Service Layer

Villegas et al. (Villegas et al., 2012) explore the canonical services of Cloud Computing and the interrelationships between them to model a federation centered in service layers. The highlight of this architecture is the isolation between layers of services and the restriction of expansion between clouds in the federation being restricted to layers at the same level. These two properties allow the existence of heterogeneous clouds regarding computing assets, business models, and types of layers. The proposal provides two feature request modes that are selected at runtime: Delegation and Federation. The first performs the request and allocation of resources to subsequent layers of the cloud (IaaS – PaaS). The Federation is the act of requesting the allocation of resources between the same layers of different clouds. To accomplish this task, the authors highlight the difficulty of the definition and adoption of protocols and policies for interoperability between layers of clouds to expose or to share services during the composition of federation. The choice between performing a Delegation or Federation, as stated previously, is a process that must be done in real time during a request. It is complex and involves cost analysis and availability of resources within the clouds themselves. Besides the architecture, the authors describe the flow of information and processing requirements between the same cloud layers.

RESERVOIR

RESERVOIR (Rochwerger et al., 2009) is a project funded by the European Commission (EC, 2015) and sponsored by the Seventh Framework Programme (FP7, 2015). This architecture was developed with the aim of providing a federal environment for SaaS offering service providers. Its model was focused on loosely coupling and in the absence in the literature of support to Business Service Management[5] (BSM). The functional requirements set by the proposal include the fast and automatic installation of applications and services, dynamic elasticity, automated continuous optimization, and independence virtualization technologies. This organization is "well behaved" as it keeps the same arrangement of the layered components in the federation elements. Consequently, interoperability is maximized because the structural elements are known and communicate over a set of protocols optimized for this purpose. Moreover, RESERVOIR does not have an FLA, since the clouds of the federation already behave following certain rules defined at the implementation level. Another feature is that every cloud has autonomy to choose its marketing model (fixed price per use period or pay-per-use, for example). These features provide the federation sites freedom to adapt to different scenarios and niches (commercial and non-commercial).

Federated Cloud Framework Architecture

Manno et al. (Manno et al., 2012) use semantics to model a federation called Federated Cloud Framework Architecture (FCFA) aiming at IaaS level. The authors use ontology to provide interoperability between distinct and autonomous clouds. The representation language chosen to describe the ontology was the Web Ontology Language (OWL, 2015), which has been used in the implementation of semantic aspects

of Web 2.0[6]. The proposal focuses on the utilization of the federation as an execution environment for distributed and complex applications, treating resources as the main feature of a cloud. Around the resources are ontologies used to provide interoperability between different technologies and environments. The FCFA treats the Cloud Federation from the aspect of infrastructure, linking components with the physical elements, and a semantic level related to dynamic operation where there are three actors and four ontologies. The actors and their ontologies are: virtual environments – hNode Ontology, physical server – hNode Ontology, datacenter – Cloud Ontology and Federation – Federation Ontology.

Inter-Cloud Federation Framework

Inter-Cloud Federation Framework architecture (ICFF) is part of an Inter-Cloud framework proposed by Makkes et al. (Makkes et al., 2013). The main objective of this proposal is the creation of a set of solutions capable of performing centralized allocation and coordination of distributed services to appear to the user as single set of services. The ICFF provides a federated environment of heterogeneous clouds, also adds to the organization features that are not allocated in clouds but in other administrative domains providing the implementation and migration of legacy applications. The component responsible for providing interoperability within the ICFF is the Gateway, which performs the translation of requests, protocols, and data formats between clouds that participate in the federation. In addition to the Gateway, there are also components in the architecture capable of providing security features: identification and trust management; and those related to the services themselves: discovery, record, and brokerage services.

Federated Cloud Management

Marosi et al. (Marosi et al., 2011) propose a service-oriented architecture for Federation of IaaS Clouds. Services are made available as Virtual Appliances (VAs) (Sapuntzakis et al., 2003), which meet the requests and are initially available in a central repository called FCM Repository. For the VAs to be effectively used, they must be locally present, so their replication by local repositories is necessary. The replication of VAs contained in the FCM Repository to local repositories is performed thought the segmentation of the VA of interest into small pieces and reconstruction of the Virtual Appliance in the local destination repository. Two classes of Brokers perform all actions within the federation: Generic Meta-Broker, performing the interface with consumers and with other Brokers of the environment; and the Cloud Brokers who are responsible for managing the virtual machine instances of a VA provider located in a specific cloud.

CometCloud Federation

The federation proposed by Petri et al. (Petri et al., 2014) has a fully decentralized profile and focuses in providers' revenue maximization by outsourcing tasks. Petri et al. used the CometCloud as the infrastructure to implement a federation using a Master-Workers methodology. This autonomic computing engine (Huebscher & McCann, 2008) implemented over Comet (Li & Parashar, 2007) supports the integration of public and private clouds by generating spaces (sharing and manager) and also a set of services that assists building of multiple clouds environments, e.g. service discovery. In the CometCloud Federation, each cloud present in the organization interacts directly with one or more clouds without performing broadcast or using a centralized entity. The authors formalize a service execution as a set of tasks that

could be outsourced. To decide between outsourcing a task or running it in-house, the architecture uses policy sets: the time to conclude the task, local computing power, and the cost to run it locally.

SUMMARY OF APPROACHES

In order to foster a deeper understanding of the concepts described previously on Cloud Federations, we propose the following summary and comparison of the existing Cloud Federation systems and how they addressed the functional properties described in Section III. Based on such functional properties, a Cloud Federation can accommodate or adopt multiple uses for any of the usage properties also from Section III, with the only constraints imposed by the functionality implemented. For instance, any of the Cloud Federation systems can implement policies to prioritize any participating actor (namely, customer, business, provider or service) of the centric property. Moreover, regarding the expansion of a Federation, it can be accomplished in a number of ways depending on the functional mechanisms adopted, the approaches that decided to adopt a broker-based interoperability can perhaps be more flexible than purely standard-based interoperability approaches, as the former can expand on vertically and horizontally easier.

As it can be seen in Table 3, all of the approaches adopted automated provisioning of resources and this requires specifications to express contracts (SLAs) and monitoring mechanisms for supervising the state of resources as well. Authentication is also a mandatory requirement for accessing the federations, most of the approaches adopted them or referred to consolidated authentication / security policies and mechanisms. Regarding the interoperability approach, it seems that the broker is the most adopted one, as it provides with certain degrees of isolation between consumers and providers and, therefore, more flexibility (less coupling) in the interactions. Nevertheless, the broker is also enhanced in many cases by means of standardization and ontologies for describing the services / resources available. On the other hand, many of the existing approaches are dealing with IaaS, PaaS and SaaS. Finally, the integrity property, as defined above, is the one that opens more research challenges and opportunities. It aims at regulating the behavior of cloud providers in the federation and it establishes and regulates the consumption and provision of resources by the participants. OpenStack / keystone with their Virtual Organization (VO) concept is the only system, to be best of our knowledge, that worked in defining the role played by each participant and the rules that govern the offering / consumption of resources. However, this concept of VO, as discussed within the ICFF project in (Makkes et al. 2013), comes from Grid computing and may need to be adapted to the Cloud Computing domain.

EMERGING PRACTICAL APPLICATIONS OF CLOUD FEDERATION

While much of the concepts defined and work cited here may be more research-oriented, the fact that cloud federation has such tremendous potential to enable large-scale collaborations among institutions, corporation and even governments means that there are definite emerging practical applications of cloud federation. The importance of cloud federation has been recognized by NIST as “Requirement 5: Frameworks to Support Federated Community Clouds”, in the NIST US Government Cloud Computing Technology Roadmap (NIST, 2011). The NIST cloud deployment models are widely accepted definitions, but both *hybrid* and *community* clouds are fundamental examples where federation management is necessary to be done in a secure manner.

Table 3. Functional properties

Federation Approach	Authentication	Contracts	Comm. Model	Integrity
Contrail	N/A initially, certificates eventually	Cloud SLAs (SLA@SOI)	Market-driven	N/A
mOSAIC	Keystone	User-centric SLAs (Cloud SLAs)	Market-driven	N/A
OpenStack	X509 certificates: Global Identities	Virtual Organization + Cloud SLAs	Market-driven	VOs as a FLA
RESERVOIR	Certificates	No global SLA, rules for each service (Cloud SLAs).	Free-choice	N/A
FCFA	Federated Id Management: 3rd party IDP	Cloud SLAs + Federated Contract	Market-driven	N/A
ICFF	N/A initially, certificates eventually	Cloud SLAs (SLA@SOI)	Market-driven	N/A
FCM	Keystone	User-centric SLAs (Cloud SLAs)	Market-driven	N/A
CometCloud	X509 certificates: Global Identities	Virtual Organization + Cloud SLAs	Market-driven	N/A

Federation Approach	Broker	Monitoring	Object	Provisioning	Service Mngmnt.
Contrail	Broker + ontology	✓	IaaS, PaaS, SaaS	Automatic	✓
mOSAIC	Broker	✓	PaaS	Automatic	✓
OpenStack	Standards	✓	IaaS, PaaS, SaaS	Automatic	✓
RESERVOIR	Broker + ontology	✓	IaaS	Automatic	✓
FCFA	Broker	✓	IaaS	Automatic	✓
ICFF	Broker	✓	IaaS, PaaS, SaaS	Automatic	✓
CommetCloud	Broker	✓	IaaS	Automatic	✓

Federation can be managed at any level in the system stack. As we have noted above, the OpenStack Keystone project is building out basic support for cloud federation whereby multiple OpenStack deployments can peer to one another at the IaaS level (OpenStack, 2015). That is to say, different deployments can offer resources, e.g., compute and storage, to users within the federation. This is being done to support the hybrid cloud business model where one cloud can cloud-burst into another. Offering resources from a provider to a consumer is a form of *asymmetric* federation. The OpenStack corporate sponsors that are major contributors to the Keystone project clearly view this as enabling a cloud marketplace.

As a fully commercialized example of IaaS federation, Cisco offers the Cisco *Intercloud Fabric* that can establish a hybrid cloud between private and public clouds (Cisco, 2015). This provides an environment where consistent network configurations and security policies can be enforced. The hybrid resources can be managed by either the consuming enterprise IT department or the public Service Provider. Enterprises can use the fabric to access different cloud service providers. Likewise, service providers can use the fabric to make their resources available to consumers. Regardless of who's managing the environment, users see a uniform set of resources and workloads.

Besides enabling a marketplace for cloud infrastructure services, federation can enable a marketplace for business-level services at the SaaS-level. Here, one corporation can contract with a commercial SaaS provider for web-based services. As an example, the training courses in many businesses are provided by external organizations. These can be offered as SaaS-level services where the consuming organization acts as the Identity Provider to the SaaS-level Service Provider. Employees log into web-based training exercises that have been customized for their corporation using their corporate identity credentials. These corporate identity credentials are then validated by the SaaS-provider with the home organization. Again, this is an example of an asymmetric federation to sell commercial services.

Such commercial SaaS-level federations are supported by tooling from corporations such as PingIdentity (PingIdentity, 2015). PingIdentity's PingFederate tool enables federated identity management for SaaS-level access using established standards, such as OpenID Connect, OAuth and SAML. The actual integration of user services with secure communication is enabled by PingAccess. We note that other corporations such as Amazon Web Services, Microsoft, F5, and CA Technologies, all support different forms of SaaS federation.

Clearly these systems do not support federation in the most general sense, but rather these corporations are tailoring the use of federation technologies to build an economically viable marketplace. As use of federation technology widens, we can expect to see broader and more general uses. As an important precedent, the Interoperable Global Trust Federation (IGTF) is an operational organization that is critical to enabling the global collaboration of "big science" groups (IGTF, 2015). IGTF essentially defines standards for the operation of PKI Certificate Authorities (CAs). Once an institution demonstrates that it's CAs are compliant, other institutions will trust certificates signed by their CA. While this was originally developed to support the sharing of data from the high-energy sensors at CERN, it is now used by a wide variety of science user groups, including chemistry, biology, and environmental monitoring, on five continents. While IGTF provides trust management in a very specific context, it nonetheless demonstrates the possibilities of global-scale federations.

CONCLUSION

The Cloud Computing paradigm has emerged as an answer to pursue computing as a utility. However, as its adoption and usage are widespread, a number of difficulties and limitations arise. Such limitations are related to the idea of *unlimited* computation on-demand, which can be constrained in cases where the amount of available resources is exhausted (e.g. a small- or medium-sized Cloud provider with heavy workload conditions). The solution for clients in order to overcome such limitations is the capability of interacting with multiple Cloud providers simultaneously. However, due to the lack of standardized methodological approaches and mechanisms, Clouds have been built as monolithic systems, where Cloud providers act as single and isolated domains. In this paper, we review the literature and existing Cloud technologies with different purposes and profiles in order to present a survey of the existing organizations involving multiple Clouds and Federation of Clouds. For each organization, its architectural blueprints were described and the potential applications and current limitations were discussed for each organization, as well as the research challenges and opportunities. A number of different organizations were proposed integrating various methods and techniques of associations. This chapter is suitable as a reference material for researchers interested in the topic to guide their research.

ACKNOWLEDGMENT

This work was co-financed by the Industry and Innovation Department of the Aragonese Government and European Social Funds (COSMOS research group, ref. T93); and by the Spanish Ministry of Economy under the program "Programa de I+D+i Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad", project id TIN2013-40809-R. Also, M.R.M.A would like to thank CAPES and L.F.B. would like to thank CNPq for the financial support.

REFERENCES

P2302 interoperability and federation (SIIF). (2015). *P2302*. Retrieved from https://standards.ieee.org/develop/project/2302.html

Al-Hazmi, Y., Campowsky, K., & Magedanz, T. (2012). A monitoring system for federated clouds. *Proceedings of 1st International Conference on Cloud Networking (CLOUDNET)*, Paris, France: IEEE.

Amazon AWS Cloudfront. (2014). *Amazon*. Retrieved from http://aws.amazon.com/cloudfront/

Amazon EC2. (2015). Retrieved from http://aws.amazon.com/ec2/

Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., . . . Xu, M. (2005). *Web services agreement specification (WS-Agreement)* (Tech. Rep.). Global Grid Forum, Grid Resource Allocation Agreement Protocol (GRAAP) WG.

Antonopoulos, N., & Gillam, L. (2010). *Cloud computing: principles, systems and applications*. London, UK: Springer Publishing Company, Incorporated. doi:10.1007/978-1-84996-241-4

Aoyama, T., & Sakai, H. (2011). Inter-cloud computing. *Business & Information Systems Engineering*, *3*(3), 173–177. doi:10.1007/s12599-011-0158-4

Apache Qpid. (2015). *Qpid*. Retrieved from https://qpid.apache.org/

Ardagna, D., di Nitto, E., Mohagheghi, P., Mosser, S., Ballagny, C., D'Andria, F., & Sheridan, C. et al. (2012). Modaclouds: A model-driven approach for the design and execution of applications on multiple clouds. *Proceedings of Workshop on Modeling in Software Engineering (MiSE) in Internation Conference on Software Engineering (ICSE),* Zurich, Switzerland (pp. 50-56). doi:10.1109/MISE.2012.6226014

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., ... Zaharia, M. (2009). *Above the clouds: A Berkeley view of cloud computing* (Tech. Rep. No. UCB/EECS-2009-28). EECS Department, University of California, Berkeley.

Assis, M. R. M., & Bittencourt, L. F. (2015). An Analysis of the Voluntary Aspect in Cloud Federations. *Proceedings of the 4rd International Workshop on Clouds and (eScience) Applications Management (CLOUDAM 2015)*. Limassol, Cyprus.

Assis, M. R. M., Bittencourt, L. F., & Tolosana-Calasanz, R. (2014). Cloud federation: Characterisation and conceptual model. *Proceedings of 3rd International Workshop on Clouds and (eScience) Applications Management (CLOUDAM 2014).* London, UK.

Banerjee, P., Friedrich, R., Bash, C., Goldsack, P., Huberman, B., Manley, J., & Veitch, A. et al. (2011). Everything as a service: Powering the new information economy. *Computer*, *44*(3), 36–43. doi:10.1109/MC.2011.67

Bermbach, D., Kurze, T., & Tai, S. (2013). Cloud federation: Effects of federated compute resources on quality of service and cost. *Proceedings of the IEEE International Conference on Cloud Engineering (IC2E),* San Francisco, California, USA (pp. 31–37). IEEE Computer Society. doi:10.1109/IC2E.2013.24

Bernsmed, K., Jaatun, M. G., Meland, P. H., & Undheim, A. (2011). Security SLAs for federated cloud services. *Proceedings of IEEE International Conference on Availability, Reliability and Security (ARES),* Vienna: Austria (pp. 202-209). IEEE Computer Society.

Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S., & Morrow, M. (2009). Blueprint for the intercloud - protocols and formats for cloud computing interoperability. *Proceedings of the Fourth International Conference on Internet and Web Applications and Services (ICIW)*, Washington, DC, USA. IEEE Computer Society. doi:10.1109/ICIW.2009.55

Bernstein, D., & Vij, D. (2104) Intercloud federation using via semantic resource federation API and dynamic SDN Provisioning. *Proceedings of International Conference and Workshop on the Network of the Future (NOF),* Paris, France (pp.1–8). IEEE Computer Society.

Bestavros, A., & Krieger, O. (2014). Toward an open cloud marketplace: Vision and first steps. *IEEE Internet Computing*, *18*(1), 72–77. doi:10.1109/MIC.2014.17

Bhardwaj, S., Jain, L., & Jain, S. (2010). Cloud computing: A study of infrastructure as a service (IaaS). *International Journal of Engineering and Information Technology*, *2*(1), 60–63.

Bittencourt, L. F., & Madeira, E. R. M. (2011). HCOC: A cost optimization algorithm for workflow scheduling in hybrid clouds. *Journal of Internet Services and Applications*, *2*(3), 207–227. doi:10.1007/s13174-011-0032-0

Buyya, R., Broberg, J., & Goscinski, A. M. (2011). *Cloud computing principles and paradigms. Hoboken*. New Jersey, USA: John Wiley & Sons, Inc. doi:10.1002/9780470940105

Buyya, R., Pandey, S., & Vecchiola, C. (2012). Market-oriented cloud computing and the cloudbus toolkit. In S. Azad & H. Zomaya (Eds.), *Large Scale Network-Centric Distributed Systems*. Wiley-IEEE Press.

Buyya, R., Ranjan, R., & Calheiros, R. N. (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. *Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing,* Busan, Korea (Vol. part I pp. 13–31). Springer-Verlag Berlin, Heidelberg. doi:10.1007/978-3-642-13119-6_2

Buyya, R., Yeo, C. S., & Venugopal, S. (2009). Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility. *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID),* Shanghai, China (p. 1). IEEE Computer Society. doi:10.1109/CCGRID.2009.97

Calheiros, R. N., Toosi, A. N., Vecchiola, C., & Buyya, R. (2012, October). A coordinator for scaling elastic applications across multiple clouds. *Future Generation Computer Systems*, *28*(8), 1350–1362. doi:10.1016/j.future.2012.03.010

Campbell, R., Gupta, I., Heath, M., Ko, S. Y., Kozuch, M., Kunze, M., & Soh, Y. C. et al. (2009). Open CIRRUS™ cloud computing testbed: Federated data centers for open source systems and services research. *Proceedings of the 2009 Conference on Hot Topics in Cloud Computing (HotCloud)*, San Diego, CA, USA. USENIX Association.

Canali, C., Cardellini, V., Colajanni, M., & Lancellotti, R. (2008). Content delivery and management. In R. Buyya, M. Pathan, & A. Vakali (Eds.), *Content Delivery Networks* (Vol. 9, pp. 105–126). Springer Berlin Heidelberg. doi:10.1007/978-3-540-77887-5_4

Celesti, A., Tusa, F., Villari, M., & Puliafito, A. (2010). Security and cloud computing: Intercloud identity management infrastructure. In S. Reddy (Ed.), *Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE)* (pp. 263–265). IEEE Computer Society. doi:10.1109/WETICE.2010.49

Celesti, A., Tusa, F., Villari, M., & Puliafito, A. (2011). Evaluating a distributed identity provider trusted network with delegated authentications for cloud federation. *Proceedings of the 2th International Conference on Cloud Computing, Grids and Virtualization (CLOUD COMPUTING 2011),* Rome, Italy (pp. 79–85). International Academy, Research, and Industry Association.

Chadwick, D. K., Siu, K., Lee, C., Fouillat, Y., & Germonville, D. (2013). Adding Federated Identity Management to OpenStack. *Journal of Grid Computing*, *12*(1), 3–27. doi:10.1007/s10723-013-9283-2

Chaurasiya, V. K., Srinivasan, K., Thyagarajan, K., Govil, S. B., & Das, S. (2012). An approach to identify the optimal cloud in cloud federation. *International Journal of Cloud Computing and Services Science*, 1(1), 35–44.

Chen, D., & Doumeingts, G. (2003). European initiatives to develop interoperability of enterprise applications - basic concepts, framework and roadmap. *Annual Reviews in Control*, *27*(2), 153–162. doi:10.1016/j.arcontrol.2003.09.001

Cisco Intercloud Fabric: Hybrid Cloud with Choice, Consistency, Control and Compliance. (2015). *Cisco*. Retrieved from http://www.cisco.com/c/en/us/td/docs/solutions/Hybrid_Cloud/Intercloud/Intercloud_Fabric.pdf

Clayman, S., Galis, A., Chapman, C., Toffetti, G., Rodero-Merino, L., Vaquero, L. M., & Rochwerger, B. (2010). Monitoring service clouds in the future internet. In G. Tselentis, A. Galis, A. Gavras, S. Krco, & T. Zahariadis et al. (Eds.), *Towards the Future Internet - Emerging Trends from European Research,* Amsterdam, Netherlands (pp. 115–126). IOS Press.

Cloud infrastructure management interface (CIMI) model and restful http based protocol an interface for managing cloud infrastructure (Standard No. DSP0263). (2012). *Distributed Management Task Force*.

Comuzzi, M., Kotsokalis, C., Spanoudakis, G., & Yahyapour, R. (2009). Establishing and monitoring SLAs in complex service based systems. *Proceedings of the IEEE International Conference of Web services (ICWS),* Miami, FL, USA (pp. 783–790). IEEE Computer Society. doi:10.1109/ICWS.2009.47

Cuomo, A., Modica, G., Distefano, S., Puliafito, A., Rak, M., Tomarchio, O., & Villano, U. et al. (2013). An SLA-based broker for cloud infrastructures. *Journal of Grid Computing*, *11*(1), 1–25. doi:10.1007/s10723-012-9241-4

Distributed management task force. (2015). *DMTF*. Retrieved from http://www.dmtf.org/

Docker: open platform for developers and sysadmins of distributed applications. (2015). *Docker*. Retrieved from http://www.docker.com/

Dreo, G., Golling, M., Hommel, W., & Tietze, F. (2013). ICEMAN: An architecture for secure federated inter-cloud identity management. *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM),* Ghent, Belgium (p. 1207–1210).

Emeakaroha, V. C., Netto, M. A., Calheiros, R. N., Brandic, I., Buyya, R., & De Rose, C. A. F. (2012). Towards autonomic detection of SLA violations in cloud infrastructures. *Future Generation Computer Systems*, *28*(7), 1017–1029. doi:10.1016/j.future.2011.08.018

European Commission. (2015). Retrieved from http://ec.europa.eu/index/

Flake, S., Tacken, J., & Zoth, C. (2012). Real-time rating and charging in federated cloud environments. *Proceedings of the IEEE 17th Conference of Emerging Technologies Factory Automation (ETFA),* Krakón, Poland (pp. 1–6). IEEE Computer Society. doi:10.1109/ETFA.2012.6489791

Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008). Cloud computing and grid computing 360-degree compared. *Proceedings of the Grid Computing Environments Workshop (GCE),* Austin, TX (pp. 1–10). USA: IEEE Computer Society. doi:10.1109/GCE.2008.4738445

Gang, L., & Mingchuan, W. (2014). Everything-as-a-service platform for on-demand virtual enterprises. *Information Systems Frontiers*, *16*(3), 435–452. doi:10.1007/s10796-012-9351-3

Ganglia monitoring system. (2015). *Ganglia*. Retrieved from http/ganglia.sourceforge.net/

Garcia, A. L. C., & Puel, M. (2013). Identity Federation with VOMS in Cloud Infrastructures. *Proceedings of the 5th IEEE International Conference on Cloud Computing Technology and Science (CLOUDCOM),* Bristol, UK (pp. 42–48). IEEE Computer Society. doi:10.1109/CloudCom.2013.13

Geelan, J., Klems, M., Cohen, R., Kaplan, J., Gourlay, D., Gaw, P., . . . Berger, I. W. (2008). *Twenty-one experts define cloud computing*. Retrieved from http://virtualization.sys-con.com/node/612375

Godfrey, R., Ingham, D., & Schloming, R. (2012). *OASIS advanced message queuing protocol (AMQP) version 1.0*. Retrieved from http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf

Goiri, I., Guitart, J., & Torres, J. (2010). Characterizing cloud federation for enhancing providers' profit. *Proceedings of the 3rd International Conference on Cloud Computing (CLOUD),* Miami, FL, USA (pp. 123–130). IEEE Computer Society. doi:10.1109/CLOUD.2010.32

Gomes, E. R., Vo, Q. B., & Kowalczyk, R. (2012). Pure exchange markets for resource sharing in federated clouds. *Concurrency and Computation*, *24*(9), 977–991. doi:10.1002/cpe.1659

Google apps for works. (2015). *Google Apps*. Retrieved from https://www.google.com/intx/pt-BR/work/apps/business/

Google Compute Engine. (2015). *Google.* Retrieved from https://cloud.google.com/products/compute-engine/

Grozev, N., & Buyya, R. (2012). Inter-Cloud Architectures and Application Brokering: Taxonomy and Survey. ACM Computing Surveys, 47(1), 7:1–7:47.

Harsh, P., Jegou, Y., Cascella, R. G., & Morin, C. (2011, October 26-28). Contrail virtual execution platform challenges in being part of a cloud federation - (invited paper). In Abramowicz, W., Llorente, I. M., Surridge, M., Zisman, A., and Vayssière, J., (Eds.), *Towards a Service-Based Internet – Proceedings of the 4th European Conference, ServiceWave 2011*, Poznan, Poland, LNCS (V*ol. 6994,* pp. 50–61). Springer.

Hassan, M. M., Abdullah-Al-Wadud, M., & Fortino, G. (2015). A socially optimal resource and revenue sharing mechanism in cloud federations. *Proceedings of the 19th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD),* Calabria, Italy (pp. 620–625). IEEE Computer Society. doi:10.1109/CSCWD.2015.7231029

Herbst, N. R., Kounev, S., & Reussner, R. (2013). Elasticity in cloud computing: What it is, and what it is not. *Proceedings of the 10th International Conference on Autonomic Computing (ICAC),* San Jose, CA, USA (pp. 23–27). USENIX.

HP Openview. (2015). *HP*. Retrieved http://www.hp.com/

Huebscher, M. C., & McCann, J. A. (2008). A survey of autonomic computing—degrees, models, and applications. *ACM Computing Surveys*, *40*(3), 7. doi:10.1145/1380584.1380585

An open, global, cloud interoperability project. (2015). Intercloud Testbed. Retrieved from http://www.intercloudtestbed.org

Interoperable Global Trust Federation. (2015). Retrieved from http://www.igtf.net

Jeffery, K., & Neidecker-Lutz, B. (Eds.), (2010). The future of cloud computing: Opportunities for European cloud computing beyond 2010 (Tech. Rep). European Commission, Information Society and Media.

Javascript object notation. (2015). *JSON*. Retrieved from http://www.json.org/

Keahey, K., Tsugawa, M., Matsunaga, A., & Fortes, J. (2009). Sky computing. *IEEE Internet Computing*, *13*(5), 43–51. doi:10.1109/MIC.2009.94

Kecskemeti, G., Kertesz, A., Marosi, A., & Kacsuk, P. (2012). Interoperable Resource Management for Establishing Federated Clouds. In M. Villari, I. Brandic, & F. Tusa (Eds.), *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice* (pp. 18–35). Hershey, PA, USA: Business Science Reference. doi:10.4018/978-1-4666-1631-8.ch002

Keller, A., & Ludwig, H. (2003). The WSLA framework: Specifying and monitoring service level agreements for web services. *Network and System Management*, *11*(1), 57–81. doi:10.1023/A:1022445108617

Kurze, T., Klems, M., Bermbach, D., Lenk, A., Tai, S., & Kunze, M. (2011). Cloud federation. *Proceedings of the 2nd International Conference on Cloud Computing, Grids, and Virtualization (Cloud Computing 2011),* Rome, Italy (pp. 32–38). International Academy, Research, and Industry Association.

Le, K., Bianchini, R., Zhang, J., Jaluria, Y., Meng, J., & Nguyen, T. D. (2011). Reducing electricity cost through virtual machine placement in high performance computing clouds. *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis,* Seattle, Washington (pp. 22:1–22:12). ACM. doi:10.1145/2063384.2063413

Li, A., Yang, X., Kandula, S., & Zhang, M. (2010). CloudCmp: Comparing public cloud providers. *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement,* Melbourne, Australia (pp. 1–14). ACM.

Li, Z., & Parashar, M. (2007). A computational infrastructure for grid-based asynchronous parallel applications. *Proceedings of the 16th International Symposium on High Performance Distributed Computing,* Monterey, California, USA (pp. 229– 230). ACM. doi:10.1145/1272366.1272404

LXC. (2015). *Linux container*. Retrivied from https://linuxcontainer.org/

Makkes, M. X., Ngo, C., Demchenko, Y., Stijkers, R., Meijer, R., & Laat, C. d. (2013). Defining intercloud federation framework for multi-provider cloud services integration. *Proceeding of the 4th International Conference on Cloud Computing, Grids, and Virtualization (CLOUD COMPUTING 2013),* Valencia, Spain (pp. 185–190). International Academy, Research, and Industry Association.

Manno, G., Smari, W. W., & Spalazzi, L. (2012). FCFA: A semantic-based federated cloud framework architecture. *Proceedings of the International Conference on High Performance Computing & Simulation (HPCS),* Madrid, Spain (p. 42-52). IEEE Computer Society. doi:10.1109/HPCSim.2012.6266889

Marosi, A., Kecskemeti, G., Kertesz, A., & Kacsuk, P. (2011). FCM: An architecture for integrating iaas cloud systems. *Proceedings of the 2th International Conference on Cloud Computing, Grids, and Virtualization (CLOUD COMPUTING 2011),* Rome, Italy (pp. 7–12). International Academy, Research, and Industry Association.

Marshall, P., Keahey, K., & Freeman, T. (2010). Elastic site: Using clouds to elastically extend site resources. *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid),* Washington, DC, USA (pp. 43–52). IEEE Computer Society. doi:10.1109/CCGRID.2010.80

Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing (Technical Report). National Institute of Standards and Technology*.

Mercosur. (2015). Retrieved from http://www.mercosur.int/

Microsoft Azure cloud computing platform and services. (2015). *Azure*. Retrieved from https://azure.microsoft.com/

Mihailescu, M., & Teo, Y. M. (2010). Dynamic resource pricing on federated clouds. *Proceedings of the 10th IEEE/ACEM International Conference on Cluster, Cloud and Grid Computing (CCGrid),* Washington, DC, USA (pp. 513–517). IEEE Computer Society.

Massachusetts open cloud. (2015). *MOC*. Retrieved from http://www.bu.edu/hic/research/massachusetts-open-cloud

National security agency. (2015). Retrieved from https://www.google.com/intx/pt-BR/work/apps/business/

NIST US Government Cloud Computing Technology Roadmap. (2011) *NIST*. Retrieved from http://www.nist.gov/itl/cloud/upload/SP_500_293_volumeI-2.pdf

Nyréen, R., Edmonds, A., Papaspyrou, A., & Metsch, T. (2011). O*pen cloud computing interface (OCCI) – core*. Retrieved from http://www.ogf.org/documents/GFD

Identity in the cloud use cases version 1.0. (2012). *OASIS*. Retrieved from http://docs.oasis-open.org/id-cloud/IDCloudusecases/v1.0/cn01/IDCloud-usecases-v1.0-cn01.pdf

Open computing infrastructure for elastic service. (2015). *Contrail*. Retrieved from http://www.contrail-project.eu/

Open standard for authorization. (2015). *oAuth*. Retrieved from http://oauth.net/

OpenId. (2015). Retrieved from http://openid.net/

OpenStack. (2015). Retrieved from http://www.openstack.org/

Open virtualization format. (2015). *OVF*. Retrieved from http://www.dmtf.org/standards/ovf

Web ontology language. (2015). *OWL*. Retrieved from http://www.w3.org/TR/owl-ref/

Patel, P., Ranabahu, A., & Sheth, A. (2009). Service level agreement in cloud computing. *Proceedings of the Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA)*, Orlando, FL, USA.

Pathan, M., Buyya, R., & Vakali, A. (2008). Content delivery networks: State of the art, insights, and imperatives. In R. Buyya, M. Pathan, & A. Vakali (Eds.), *Content Delivery Networks* (Vol. 9, pp. 3–32). Springer Berlin Heidelberg. doi:10.1007/978-3-540-77887-5_1

Petcu, D. (2011). Portability and interoperability between clouds: Challenges and case study. *Proceedings of the 4th European Conference on Towards a Service-based Internet,* Poznan, Poland (pp. 62–74). Springer Berlin Heidelberg. doi:10.1007/978-3-642-24755-2_6

Petcu, D., Craciun, C., & Rak, M. (2011). Towards a cross platform cloud API. *Proceedings of the International Conference on Cloud Computing and Services Science (CLOSER),* Noordwijkerhout, The Netherlands (pp. 166–169).

Petcu, D., Di Martino, B., Venticinque, S., Rak, M., Máhr, T., Lopez, G. E., & Stankovski, V. et al. (2013). Experiences in building a mOSAIC of clouds. *Journal of Cloud Computing*, *2*(1), 1–22.

Petri, I., Beach, T., Zou, M., Montes, J., Rana, O., & Parashar, M. (2014). Exploring models and mechanisms for exchanging resources in a federated cloud. *Proceedings of the IEEE International Conference on Cloud Engineering (IC2E)*, Boston, Massachusetts (pp. 215–224). USA: IEEE Computer Society. doi:10.1109/IC2E.2014.9

Petri, I., Montes, J. D., Zou, M., Beach, T., Rana, O. F., & Parashar, M. (2015). Market models for federated clouds. *IEEE Transactions on Cloud Computing*, *3*(3), 398–410. doi:10.1109/TCC.2015.2415792

PingIdentity. (2015). *Ping Identity.* Retrieved from http://www.pingidentity.com

Practical guide to cloud service level agreements version 1.0 (Tech. Rep.). (2012Cloud Standards Customer Council Workgroup. CSCC.

Rackspace open cloud. (2015). *Rackspace*. Retrieved from http://www.rackspace.com/cloud/

Rebai, S., Hadji, M., & Zeghlache, D. (2015). Improving profit through cloud federation. *Proceedings of the 12th Annual IEEE Consumer Communications and Networking Conference (CCNC),* Las Vegas, Nevada, USA (pp. 732–739). IEEE Computer Society. doi:10.1109/CCNC.2015.7158069

Reference architecture for an SLA management framework (Standard). (2011). EU FP7 project SLA@SOI.

RFC 4158. (2015). *Request for comment*. Retrieved from http://tools.ietf.org/html/rfc4158

RFC 5280. (2015). *Request for comment*. Retrieved from http://tools.ietf.org/html/rfc5280

Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., & Galán, F. et al. (2009). The RESERVOIR model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, *53*(4), 535–545. doi:10.1147/JRD.2009.5429058

Sapuntzakis, C., Brumley, D., Chandra, R., Zeldovich, N., Chow, J., Lam, M. S., & Rosenblum, M. (2003). Virtual appliances for deploying and maintaining software. *Proceedings of the 17th USENIX Conference on System Administration,* Berkeley, CA, USA (pp. 181–194). USENIX Association.

Seo, S., Kim, M., Cui, Y., Seo, S., & Lee, H. (2015). SFA-based cloud federation monitoring system for integrating physical resources. *Proceedings of the International Conference on Big Data and Smart Computing (BIGCOMP)* Jeju Island, Korea (pp. 55–58). IEEE Computer Society. doi:10.1109/35021BIGCOMP.2015.7072851

Seventh framework programmer. (2015). Retrieved from http://ec.europa.eu/research/fp7/index/

Sipos, G., Turilli, M., Newhouse, S., & Kacsuk, P. (2013, April). A European Federated Cloud: Innovative distributed computing solutions by EGI. Proceedings of the EGU General Assembly Conference Abstracts (Vol. 15, p. 8690).

Softlayer cloud built to perform. (2015). *Softlayer*. Retrieved from http://www.softlayer.com/

Summary of the Amazon EC2 and Amazon RDS service disruption. (2011). *Amazon*. Retrieved from http://aws.amazon.com/message/65648/

Summary of the aws service event in the US east region. (2012). *Amazon*. Retrieved from http://aws.amazon.com/message/67457/

The enterprise-class monitoring solution for everyone. (2015). *Zabbix*. Retrieved from http://www.zabbix.com/

SOA source book. (2009). *The Open Group*. Retrieved from http://books.google.com.br/books?id=SbZfhkdqbagC

Thomas, M. V., Dhole, A., & Chandrasekaran, K. (2015). Single sign-on in cloud federation using cloudsim. *International Journal of Computer Network and Information Security*, *7*(6), 50–58. doi:10.5815/ijcnis.2015.06.06

Tolosana-Calasanz, R., Bañares, J. A., & Colom, J.-M. (2015). On autonomic platform-as-a-service: Characterisation and conceptual model. *Proceedings of the Agent and Multi-Agent Systems: Technologies and Applications – 9th KES International Conference (KES-AMSTA),* Sorrento, Italy (Vol.. 38. pp. 217–226).

Toosi, A. N., Calheiros, R. N., & Buyya, R. (2014, May). Interconnected cloud computing environments: Challenges, taxonomy and survey. *ACM Computing Surveys*, 47(1), 7:1–7:47.

Toosi, A. N., Calheiros, R. N., Thulasiram, R. K., & Buyya, R. (2011). Resource provisioning policies to increase IaaS provider's profit in a federated cloud environment. *Proceedings of the 13th IEEE International Conference on High Performance Computing and Communications (HPCC),* Washington, DC, USA (pp. 279–287). IEEE Computer Society.

Topology and orchestration specification for cloud applications (TOSCA) version 1.0. (2013). *TOSCA*. Retrieved from http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.pdf

US Patriotic Act. (2001). Retrieved from http://www.gpo.gov/fdsys/pkg/PLAW-107publ56/pdf/PLAW-107publ56.pdf

Use cases and functional requirements for inter-cloud computing (Tech. Rep.). (2010). Global Inter-Cloud Technology Forum.

Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008, December). A break in the clouds: Towards a cloud definition. *SIGCOMM Computer Communication Review*, *39*(1), 50–55. doi:10.1145/1496091.1496100

Villegas, D., Bobroff, N., Rodero, I., Delgado, J., Liu, Y., Devarakonda, A., & Parashar, M. et al. (2012). Cloud federation in a layered service model. *Journal of Computer and System Sciences*, *78*(5), 1330–1344. doi:10.1016/j.jcss.2011.12.017

Watson, P., Lord, P., Gibson, F., Periorellis, P., & Pitsilis, G. (2008). Cloud computing for e-science with CARMEN. *Proceedings of the 2nd IBERIAN Grid Infrastructure Conference,* Porto, Portugal (pp. 3–14). NETBIBLO.

Extensible markup language. XML. (2015). Retrieved from http://www.w3.org/standards/xml/

ZeroMQ enterprise messaging broker. (2015). *ZeroMQ*. Retrieved from http://zeromq.org/

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*, *1*(1), 7–18. doi:10.1007/s13174-010-0007-6

ENDNOTES

1 The term resource is associated with the assets used by customer service.

2 Resources from other providers which are marketed to interested parties.

3 Quarantine this context means a trial period where the suspicion cloud is monitored more closely to determine whether it will return or not to be part of the federation. In this period, the consumption of the resources from the federation may be restricted.

4 In this context, workflow is a sequence of procedures for the execution of workloads considering the application characteristics.

5 Business methodology that aligns the management of information technology companies with their strategic business goals.

6 Term popularized by O'Reilly Media Company designating a second generation of communities and services based on the Web as a platform.