Interconnected Cloud Computing Environments: Challenges, Taxonomy, and Survey

ADEL NADJARAN TOOSI, RODRIGO N. CALHEIROS, and RAJKUMAR BUYYA,
The University of Melbourne, Australia

A brief review of the Internet history reveals the fact that the Internet evolved after the formation of primarily independent networks. Similarly, interconnected clouds, also called *Inter-cloud*, can be viewed as a natural evolution of cloud computing. Recent studies show the benefits in utilizing multiple clouds and present attempts for the realization of an Inter-cloud or federated cloud environment. However, cloud vendors have not taken into account cloud interoperability issues, and each cloud comes with its own solution and interfaces for services. This survey initially discusses all the relevant aspects motivating cloud interoperability. Furthermore, it categorizes and identifies possible cloud interoperability scenarios and architectures. The spectrum of challenges and obstacles that the Inter-cloud realization is faced with are covered, a taxonomy of them is provided, and fitting enablers that tackle each challenge are identified. All these aspects require a comprehensive review of the state of the art, including ongoing projects and studies in the area. We conclude by discussing future directions and trends toward the holistic approach in this regard.

Categories and Subject Descriptors: D.4.7 [**Operating Systems**]: Organization and Design—*Distributed systems*; D.2.12 [**Software Engineering**]: Interoperability; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms: Design, Standardization

Additional Key Words and Phrases: Cloud computing, cloud federation, Inter-cloud, multi-cloud, cross-clouds, utility computing

ACM Reference Format:
Adel Nadjaran Toosi, Rodrigo N. Calheiros, and Rajkumar Buyya. 2014. Interconnected cloud computing environments: Challenges, taxonomy, and survey. ACM Comput. Surv. 47, 1, Article 7 (April 2014), 47 pages.
DOI: http://dx.doi.org/10.1145/2593512

1. INTRODUCTION

Cloud computing is a term used to describe a paradigm for delivery of computing services to users on a pay-as-you-go basis. In this paradigm, users utilize the Internet and remote data centers to run applications and store data. The cloud technology allows more efficient computing by removing most of the upfront costs of setting up an IT infrastructure. It allows organizations to expand or reduce their computing facilities

Authors' addresses: A. N. Toosi, R. N. Calheiros, and R. Buyya, **Clou**d Computing and **D**istributed **S**ystems (CLOUDS) Laboratory, Department of Computing and Information Systems, The University of Melbourne, Parkville Campus, Melbourne, VIC 3010, Australia. R. Buyya is also associated as a visiting professor for the University of Hyderabad, India; King Abdulaziz University, Saudi Arabia; and Tsinghua University, China.

DOI: http://dx.doi.org/10.1145/2593512

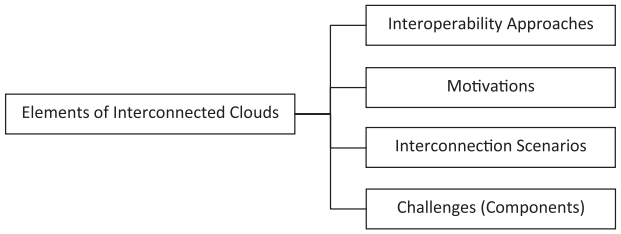

Fig. 1. Elements of interconnected cloud environments.

very quickly. There is an increasingly perceived vision that the birth of cloud computing is a big step toward the long-held dream of computing as a utility [Buyya et al. 2010].

Over the years, several technologies such as virtualization, grid computing, and service-oriented architecture (SOA) have matured and significantly contributed to make cloud computing viable. However, cloud computing is still in its early stage and suffers from lack of standardization. What actually happens is that most new cloud providers propose their own solutions and proprietary interfaces for access to resources and services. This heterogeneity is a crucial problem as it raises barriers to the path of the ubiquitous cloud realization. The main barrier is *vendor lock-in,* which is unavoidable at this stage [Rochwerger et al. 2009; Petcu 2011]; customers applying cloud solutions need to tailor their applications to fit the models and interfaces of the cloud provider, which makes future relocation costly and difficult. Furthermore, cloud computing, as a novel utility, requires ubiquitously interconnected infrastructure like other utilities such as electricity and telephony. Accordingly, interoperability and portability across clouds are important not only for protection of the user investments but also for realization of computing as a utility.

In this article, we aim to cover various aspects of interconnected cloud computing environments. Key elements of interconnected clouds are provided, and each one of them is classified in depth. Figure 1 shows the key elements of interconnected clouds from our point of view.

First, we consider interoperability approaches. Cloud interoperability in practice can be obtained through either *brokering* or *standard interfaces*. By using a service broker, which translates messages between different cloud interfaces, customers are able to switch between different clouds and cloud providers can interoperate. Standardization of interfaces is another common method for realization of interoperability. Part of this study covers different standards and initiatives related to technologies to regulate the Inter-cloud environment. However, one comprehensive set of standards is difficult to develop and hard to be adopted by all providers. A combination of the aforementioned approaches often occurs in practice.

If cloud interoperability happens, both cloud providers and customers benefit from different possible cloud scenarios as shown in Figure 2. Benefits of an interconnected cloud environment for both cloud providers and their clients are numerous, and there are essential motivations for cloud interoperability such as avoiding vendor lock-in, scalability, availability, low-access latency, and energy efficiency.

Cloud interoperability requires cloud providers to adopt and implement standard interfaces, protocols, formats, and architectural components that facilitate collaboration. Without these *provider-centric* changes, cloud interoperability is hard to achieve. Among different provider-centric approaches, *Hybrid Cloud*, *Cloud Federation,* and *Inter-cloud* are the most prominent scenarios. A hybrid cloud allows a private cloud to form a partnership with a public cloud, enabling the *cloud bursting* application deployment model. Cloud bursting allows an application to run in a private data center and to

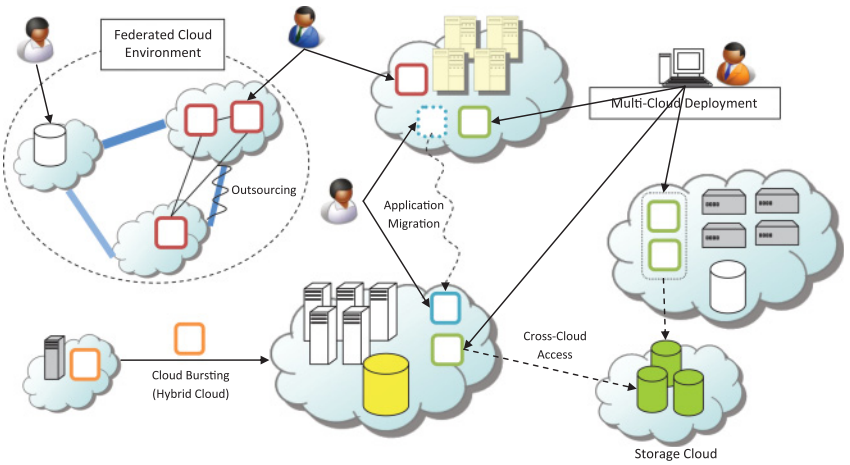

Fig. 2. Cloud interoperability and Inter-cloud.

burst into a public cloud when the demand for computing capacity spikes. Cloud federation allows providers to share their resources through federation regulations. In this paradigm, providers aim to overcome resource limitation in their local infrastructure, which may result in rejection of customer requests, by outsourcing requests to other members of the federation. Moreover, cloud federation allows providers operating at low utilization to lease part of their resources to other federation members in order to avoid wasting their nonstorable compute resources. Last but not least is Inter-cloud, in which all clouds are globally interconnected, forming a worldwide cloud federation. Inter-cloud removes difficulties related to migration and supports dynamic scaling of applications across multiple clouds.

Even if cloud interoperability is not supported by cloud providers, cloud customers are still able to benefit from *client-centric* interoperability facilitated by user-side libraries or third-party brokers. *Multicloud* application deployment using adapter layer provides the flexibility to run applications on several clouds and reduces the difficulty in migrating applications across clouds. *Aggregated service by broker*, a third-party solution in this regard, offers an integrated service to users by coordinating access and utilization of multiple cloud resources.

Cloud interoperability is a challenging issue and requires substantial efforts to overcome the existing obstacles. These include both functional and nonfunctional aspects. This study covers the spectrum of challenges in cloud interoperability. These challenges broadly cover security, service-level agreement (SLA), monitoring, virtualization, economy, networking, provisioning, and autonomics.

There are various projects and studies to propose and prototype approaches to enable interconnected cloud environments. They vary in terms of architecture, facilities they provide, and challenges they address. Another aim of this study is to survey and classify these projects.

In summary, the contributions of this article are:

—It introduces and analyzes the relevant aspects motivating cloud interoperability.
—It explores the spectrum of challenges and obstacles for Inter-cloud realization and identifies open challenges.
—It proposes a taxonomy of such challenges and obstacles and identifies enablers that tackle each of them.

—It presents a comprehensive review of the state of the art, including ongoing projects and research in the area.
—It discusses future directions and trends in the area.

This survey is organized as follows: In Section 2, we define and discuss several different terms and descriptions in the area of interconnected cloud environments to clarify the positioning of this work. In Section 3, we explore the motivation and benefits of interoperability between clouds. Then, we introduce different possible architectures for multiple cloud scenarios in Section 4. Classification of the challenges and potential enablers with respect to each challenge is provided in Section 5. Successful realization of cloud interoperation requires standards, so in Section 6, we review the current standard protocols and interfaces that facilitate the realization of cloud interoperation, including organizations and initiatives dealing with these standard technologies. In Section 7, we survey the state-of-the-art projects and developments in the area and fit them into taxonomies based on the characteristics and the challenges they address. We discuss different tools and frameworks supporting interoperable cloud infrastructure in Section 8. Finally, we conclude the study and provide a basis for future developments in this area.

2. TERMS, QUOTES, AND DEFINITIONS

In view of the fact that integration and aggregation of cloud services to achieve a seamless computing infrastructure have recently received attention and it is in the early stage of development, several different terms and descriptions have been used in the scientific community to define it. Precise understanding of these terms and definitions, including differences, similarities, and experts' comments, clarifies the current study and helps in future directions.

Inter-cloud has been introduced by Cisco [Bernstein et al. 2009] as an interconnected global "cloud of clouds" that mimics the known term Internet, "network of networks." The Inter-cloud refers to a mesh of clouds that are unified based on open standard protocols to provide a cloud interoperability [Bernstein et al. 2009]. The main objective of the Inter-cloud is similar to the Internet model and telephone system, where everything is ubiquitously connected together in a multiple-provider infrastructure.

According Vint Cerf, vice president and chief Internet evangelist at Google, who is recognized as one of "the fathers of the Internet":[1]

> *. . . It's time to start working on Inter-cloud standards and protocols so your data doesn't get trapped in one of the problems with cloud computing . . . [and these standards and protocols] allow people to manage assets in multiple clouds, and for clouds to interact with each other.*

According to Gregory Ness, chief marketing officer at Vantage Data Centers, we can:[2]

> *think of the Inter-cloud as an elastic mesh of on demand processing power deployed across multiple data centers. The payoff is massive scale, efficiency and flexibility.*

The Global Inter-cloud Technology Forum (GICTF), a Japanese organization that aims at promoting standardization of network protocols and interfaces through which cloud systems interwork with each other, defines Inter-cloud computing as:[3]

> *a cloud model that, for the purpose of guaranteeing service quality, such as the performance and availability of each service, allows on-demand reassignment of resources and transfer of workload through a*

[1]http://www.guardian.co.uk/technology/blog/2010/feb/05/google-Cloud-computing-interCloud-cerf.
[2]http://Cloudcomputing.sys-con.com/node/1009227.
[3]Use Cases and Functional Requirements for Inter-Cloud Computing: A white paper by Global Inter-Cloud Technology Forum (GICTF), http://www.gictf.jp/doc/GICTF_Whitepaper_20100809.pdf.

[sic] interworking of cloud systems of different cloud providers based on coordination of each consumer's requirements for service quality with each provider's SLA and use of standard interfaces.

The term cloud federation, on the other hand, implies the creation of a group of aggregated providers that are mutually collaborating to share their resources in order to improve each other's services [Kurze et al. 2011]. One definition of cloud federation can be deduced from Rochwerger et al. [2011], where the authors talk about the basic principles of cloud computing:

. . . federations of providers such that they can collaborate and share their resources. . . . Any federation of cloud computing providers should allow virtual applications to be deployed across federated sites. Furthermore, virtual applications need to be completely location free and allowed to migrate in part or as a whole between sites. At the same time, the security privacy and independence of the federation members must be maintained to allow computing providers to federate.

Reuven Cohen, founder and CTO of Enomaly Inc.,[4] defines cloud federation as follows:

Cloud federation manages consistency and access controls when two or more independent geographically distinct clouds share either authentication, files, computing resources, command and control or access to storage resources.

Cloud federation and Inter-cloud are relatively new in the cloud computing area. According to the aforementioned arguments, different standards are first required before it can be defined and subsequently realized.

The terms Inter-cloud and cloud federation are often used interchangeably in the literature. Similarly, we consider these as synonyms in this survey and contemplate Inter-cloud as a worldwide federation of the clouds. However, some practitioners and experts prefer to give different definitions to these terms. According to Ellen Rubin, founder and VP of Products at CloudSwitch,[5] there are some key differences between Inter-cloud and cloud federation. The primary difference between the Inter-cloud and federation is that the Inter-cloud is based on the future standards and open interfaces, while federation uses a provider version of the interfaces. According to the discussion, cloud federation can be considered as a prerequisite toward the ultimate goal of the Inter-cloud. With the Inter-cloud vision, clouds must federate and interoperate and all would have a common perceptive of how applications should be deployed. In this model, interoperability of different cloud platforms is achieved without explicit referencing by user.

According to Chen and Doumeingts [2003], there are two distinguished approaches to obtain interoperability in practice:

(1) Adhering to published interface standards
(2) Developing a broker of services that can convert one product's interface into another product's interface "on the fly"

A relevant example of the first approach is TCP/IP and of the second kind of approach is enterprise application integration approaches, the CORBA architecture, and its object request broker (ORB) [Chen and Doumeingts 2003]. Current cloud federation and Inter-cloud projects, as expressed in Figure 3, follow either of the two methods or a combination of them. However, it seems that the Inter-cloud concept is typically close to the former approach, while the cloud federation idea is mostly inspired by the latter.

[4]Enomaly, http://www.enomaly.com/.
[5]http://www.Cloudswitch.com/page/Cloud-federation-and-the-interCloud.

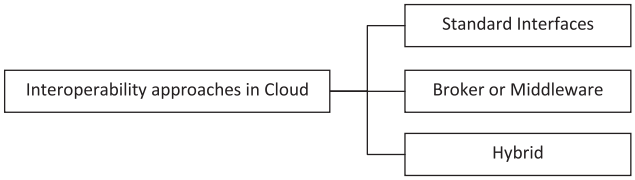

Fig. 3. Cloud interoperability approaches.

There is another group of professionals who believe that the aforementioned form of interoperability is unlikely to happen. They think of cloud integration as a completely separated layer that handles all the issues regarding aggregation and integration of the clouds as entirely detached from vendors and providers. We call this form of integration *multiple cloud*, *multicloud*, or *cross-cloud*. Aligning to this thought, Joe Skorupa, Gartner's[6] vice president, said:[7]

> *Even if an open cloud standard should come to pass, every provider would still continue to implement its own proprietary enhancements to differentiate its wares from the competition. . . . Vendors do not want clouds to become commodity products because they do not want to compete on price alone.*

Interested readers can find a thorough survey and taxonomy of architectures and application brokering mechanisms across multiple clouds in an article by Grozev and Buyya [2012]. Their work is mostly focused on application-specific brokering mechanisms and can be positioned as a part of a broader view of interconnected cloud environments presented in this article.

In this section, we provided different terms and definitions including experts' quotes on interconnected clouds. Although there are different terms and titles, interoperability between clouds is the main requirement for the realization of these scenarios. Therefore, in this study, our aim is to cover all aspects in this regard. However, some discussions are just applicable for special cases, for example, cloud federation or multiclouds, which are clearly stated. In summary, we believe that transition toward Inter-cloud has already started and it is an inevitable need for the future of cloud computing. In this regard, we present supporting arguments and motivations for cloud federation and interconnected clouds in the next section.

3. MOTIVATIONS FOR CLOUD INTEROPERABILITY

Cloud computing has already provided considerable capabilities for scalable, highly reliable, and easy-to-deploy environment for its clients. Nevertheless, the benefits of interconnected cloud environments for both cloud providers and their clients are numerous and there are essential motivations for cloud interoperability, which will eventually lead to the Inter-cloud. In this section, key benefits of an Inter-cloud environment have been summarized as demonstrated in Figure 4.

3.1. Scalability and Wider Resource Availability

Even though one of the key features of cloud computing is the illusion of infinite resources, capacity in cloud providers' data centers is limited and eventually can be fully utilized [Calheiros et al. 2012a; Aoyama and Sakai 2011]. Growth in the scale of existing applications or surge in demand for a service may result in immediate need for additional capacity in the data center. Current service providers handle this issue by overprovisioning data center capacity. That is, the average demand of the system is

[6]Gartner, http://www.gartner.com/.
[7]http://www.computerworld.com/s/article/9217158.

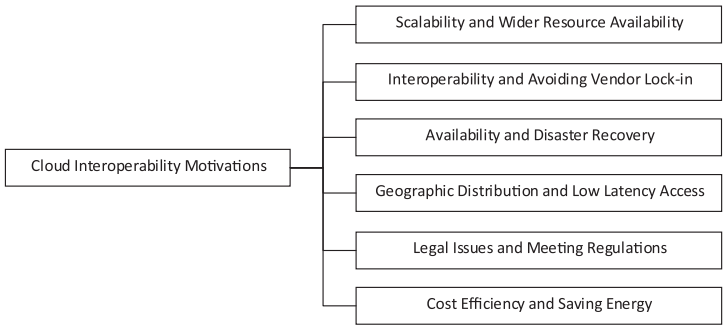

Fig. 4. Cloud interoperability motivations.

several times smaller than the capacity of its computing infrastructure. This strategy and the cost of its operation constitute a large expense for cloud owners. Actual usage patterns of many real-world application services vary with time and most of the time in unpredictable ways. Therefore, as we stated earlier, unexpected loads can potentially overburden a single cloud provider and lead to unreliable and interrupted services. It is overly restrictive in terms of small-size or private clouds. If cloud providers were able to dynamically scale up or down their data center capacity, they could save a substantial amount of money and overcome this issue. Scalable provisioning of application services under variable workload, resource, and network conditions is facilitated by interoperation of the clouds [Buyya et al. 2010]. Cloud federation helps the peak-load handling capacity of every enterprise cloud by resource sharing, without having the need to maintain or administer any additional computing nodes or servers [Rochwerger et al. 2009].

One may argue that public cloud providers are outstandingly elastic, with the perception of unlimited resources, so providers never need immediate additional capacity in their data center and they never fit into the aforementioned scenario. However, this claim does not obviate the need for additional capacity by small-size private clouds and for those applications requiring expansion across geographically distributed resources to meet quality of service (QoS) requirements of their users [Buyya et al. 2010].

3.2. Interoperability and Avoiding Vendor Lock-In

In economics, vendor lock-in is a situation where a customer becomes dependent on a vendor for its products or services and cannot move to another vendor without considerable cost and technical effort. It is also perceived as one of the current drawbacks of cloud computing [Armbrust et al. 2010]. With respect to cloud computing, vendor lock-in is the direct result of the current difference between the individual vendor paradigms based on noncompatible underlying technologies and the implicit lack of interoperability. Contemporary cloud technologies have not considered interoperability in design [Rochwerger et al. 2009; Bernstein et al. 2009]; hence, applications are usually restricted to a particular enterprise cloud or a cloud service provider. By means of cloud interoperability, cloud application deployment no longer needs to be customized. Cloud interoperability makes cloud services capable of working together and also develops the ability of multiple clouds to support cross-cloud applications [Bernstein et al. 2009].

3.3. Availability and Disaster Recovery

Although high availability is one of the fundamental design features for every cloud service, failure is inevitable. For instance, recently Amazon Web Services suffered

an outage, and as a result, a group of large customers dependent on Amazon were affected seriously.[8] Unexpected failures can easily impose service interruption on a single cloud system. Aoyama and Sakai [2011] look into an instance of a service failure in which a cloud system witnesses a natural disaster. They identify the most important requirements for disaster recovery through cloud federation. In order to enable cloud providers to continue the delivery of guaranteed service levels even in such cases, a flexible mechanism is needed to relocate resources among the multiple cloud systems. Moreover, highly available cloud applications can be constructed by multiple cloud deployments to guarantee the required service quality, such as service availability and performance. Thus, cloud systems complement each other by mutually requesting required resources from their peers.

3.4. Geographic Distribution and Low-Latency Access

It is highly unlikely that a single cloud provider owns data centers in all geographic locations of the world to meet the low-latency access requirement of applications. Moreover, existing systems do not support mechanisms to dynamically coordinate load distribution among different cloud data centers. Since predicting geographic distribution of users consuming a cloud provider's services is not trivial, the load coordination must happen automatically, and distribution of services must change in response to changes in the load [Buyya et al. 2010]. Utilizing multiple clouds at the same time is the only solution for satisfying the requirements of the geographically dispersed service consumers who require fast response time. Construction of a federated cloud computing environment is necessary to facilitate provisioning of such application services. Consistently meeting the QoS targets of applications under variable load, resource, and network conditions is possible in such an environment.

3.5. Legal Issues and Meeting Regulations

Many cloud customers have specific restrictions about the legal boundaries in which their data or application can be hosted [Schubert et al. 2010]. Supplying resources in specific geographic locations to meet regulations in the places of those customers is an essential issue for a provider who wants to serve them. These regulations may be legal (e.g., an existing legislation specifying that public data must be in the geographic boundaries of a state or country) or defined by companies' internal policies [Calheiros et al. 2012a]. Cloud interoperability provides an opportunity for the provider to identify another provider able to meet the regulations due to the location of its data center.

3.6. Cost Efficiency and Saving Energy

The pay-as-you-go" feature of cloud computing directly awards economic benefits for customers by removing the cost of acquiring, provisioning, and operating their own infrastructures [Armbrust et al. 2010]. On the other hand, cloud computing providers should avoid the problem of the *idle capacity* (where their in-house hardware is not fully utilized all the time) and the problem of *peaks in demand* (where their own systems would be overloaded for a period). As the average demand of the system is several times smaller than the peak demand [Armbrust et al. 2010], providers are able to lease part of their resources to others, in order to avoid wasting their unused resources. Moreover, they can manage peaks in demand by purchasing resources from other underutilized providers. Both strategies help them to gain economies of scale, an efficient use of their assets, and enlargement of their capabilities through enhanced resources utilization [Celesti et al. 2010a]. Furthermore, this cooperation among cloud providers lowers the energy usage by promoting efficient utilization of the computing infrastructure.

[8]https://cloudcomputing.sys-con.com/node/2416841.

In a recent study done by Le et al. [2011], the plausibility of reducing cost and energy consumption by interconnecting cloud data centers has been investigated. They present a scenario in which a provider is able to save money by placing and migrating load across multiple geographically distributed data centers to take advantage of time-based differences in electricity prices. In addition, their policies reduce the required cooling power, considering data centers located in areas with widely different outside temperatures. In general, a unified interface that provides federated interoperation between clouds would help providers save costs and reduce their carbon footprint by energy-efficient utilization of physical resources.

4. CLOUD INTEROPERABILITY SCENARIOS

"Cloud computing refers to both the applications delivered as services over the internet and the hardware and systems software in the data centers that provide those services" [Armbrust et al. 2010]. In this definition, data center hardware and software are called *cloud* and the services can be sold in low-level abstraction like Amazon EC2[9] or at a higher level like Google AppEngine.[10] When a cloud is available to the public in a pay-as-you-go manner, it is called *public* cloud, and when a cloud belongs to a business or an organization and not made available to the public, it is called *private* cloud.

Cloud environments include a multitude of independent, heterogeneous, private, and public clouds. Based on Celesti et al. [2010a], the evolution of cloud computing can be hypothesized in three subsequent phases: *monolithic*, *vertical supply chain,* and *horizontal federation*. In the monolithic stage, cloud providers are based on their own proprietary architectures that create islands of cloud. Cloud services are delivered by different providers in this stage. In the vertical supply chain stage, some cloud providers leverage services from other providers. For example, an SaaS provider deploys services of an IaaS provider to serve its own customers. In horizontal federation, different-size cloud providers federate themselves to gain benefits of a cloud federation. For example, a fully utilized IaaS provider may use resources in an underutilized provider to accommodate more VM requests.

The main stakeholders in cloud computing scenarios are *cloud users* and *cloud providers* (CPs). Cloud users can be either software/application *service providers* (SPs) who have their service consumers or *end-users* who use the cloud computing services directly. SPs offer economically efficient services using hardware resources provisioned by CPs; that is, CPs offer utility computing service required by other parties. Different combinations of CPs and cloud users (SPs or end-users) give rise to a number of plausible scenarios between clouds [Ferrer et al. 2012].

According to this discussion, if interconnection happens between clouds at different levels of cloud stack layers (Figure 5), for example, a PaaS and IaaS provider, we call it *delegation* or *vertical federation*. But if interconnection between clouds happens at the same layer (e.g., IaaS to IaaS), we call it *horizontal federation*. Since the adoption of the latter faces many more hurdles, in this article we are mostly interested in the horizontal federation. Villegas et al. [2012] consider that a federated cloud structure can be viewed as a vertical stack analogous to the layered cloud service model. At each layer, a service request can be served either through local resources using delegation or by a partner cloud provider through federation.

If cloud interoperability requires cloud providers to adopt and implement standard interfaces, protocols, formats, and architectural components that facilitate collaboration, we call that *provider-centric* interoperability. Provider-centric scenarios are categorized as *hybrid* and *federated* cloud scenarios. In *client-centric* interoperability,

[9]Amazon EC2, http://aws.amazon.com/ec2/.
[10]Google AppEngine, https://developers.google.com/appengine/.

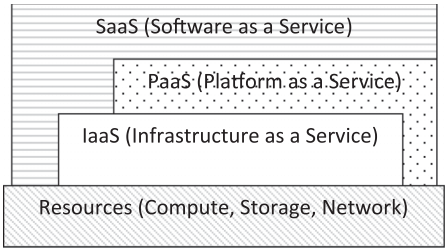

Fig. 5. Cloud service stack.

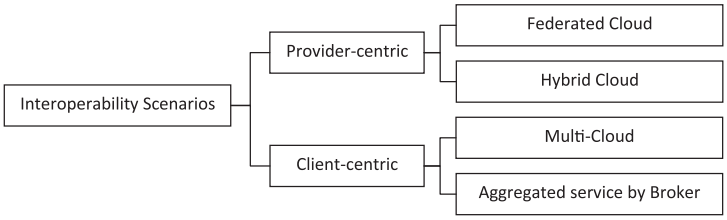

Fig. 6. Cloud interoperability scenarios.

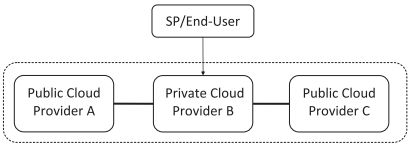

Fig. 7. Federated cloud scenario.

interoperability is not supported by cloud providers and cloud customers are required to initiate it by themselves or via third-party brokers. This kind of interoperability does not require prior business agreement among cloud providers and allows multiple cloud scenarios without adoption of common interfaces and protocols or with minimal adoption of the same. We consider *multicloud* and *aggregated service by broker* as client-centric interoperability scenarios. Figure 6 depicts the discussed classification, and the remaining parts of this section describe each scenario in detail.

4.1. Federated Scenario

In this scenario, SP establishes a contract with CP that itself is a member of a federation. A group of cloud providers are federated and trade their surplus resources among each other to gain economies of scale, efficient use of their assets, and expansion of their capabilities [Celesti et al. 2010a], for example, to overcome resource limitation during spike in demands. In this model, the computing utility service is delivered to SPs using resources of either one CP or a combination of different cloud providers. In such a scenario, the SP might be unaware of the federation and its contract is with a single cloud provider (Figure 7).

4.2. Hybrid Cloud Scenario

In hybrid cloud architecture, an organization that owns its private cloud moves part of its operations to external CPs. The organization can also sell idle capacity to other providers during periods of low load. This extension of a private cloud to combine local resources with resources from remote CPs is called hybrid cloud. In this scenario, an

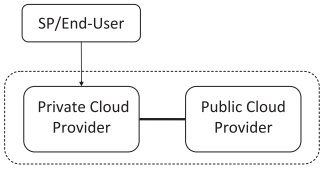

Fig. 8. Hybrid cloud scenario.

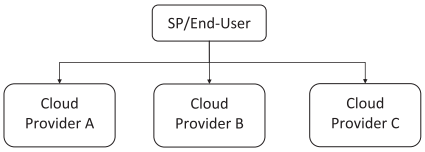

Fig. 9. Multicloud scenario.

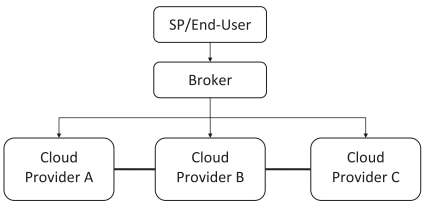

Fig. 10. Aggregated service broker scenario.

SP/end-user application can scale out through both private and public clouds when the local infrastructure is insufficient. Furthermore, this scenario can be extended if the organization offers capacity from its private cloud to others when that capacity is not needed for internal operations (Figure 8).

4.3. Multicloud Scenario

In this scenario, SPs or end-users are responsible to manage resources across multiple clouds. Service deployment, negotiating with each CP, and monitoring each CP during service operation are performed by the SP or end-user applications. In this case, the SP may require using an adapter layer with different APIs to run services on different clouds, or similarly, an end-user application may need a proper abstraction library. The important point about this scenario is that a separated layer handles all the issues regarding aggregation and integration of the clouds that is entirely apart from vendors and providers (Figure 9).

4.4. Aggregated Service by Broker

A new stakeholder, the broker, aggregates services from multiple CPs and offers an integrated service to the SPs or end-users. The deployment and management of components have been abstracted by the third-party broker. SPs or end-users benefit greatly from this model as the broker can provide a single entry point to multiple clouds. In this model, providers may also be required to install some internal components to support aggregated services by a trusted broker (Figure 10).

5. INTER-CLOUD CHALLENGES AND ENABLING APPROACHES

Inter-cloud raises many more challenges than cloud computing. In this section, the main challenges of cloud interoperability will be examined as listed in Figure 11.

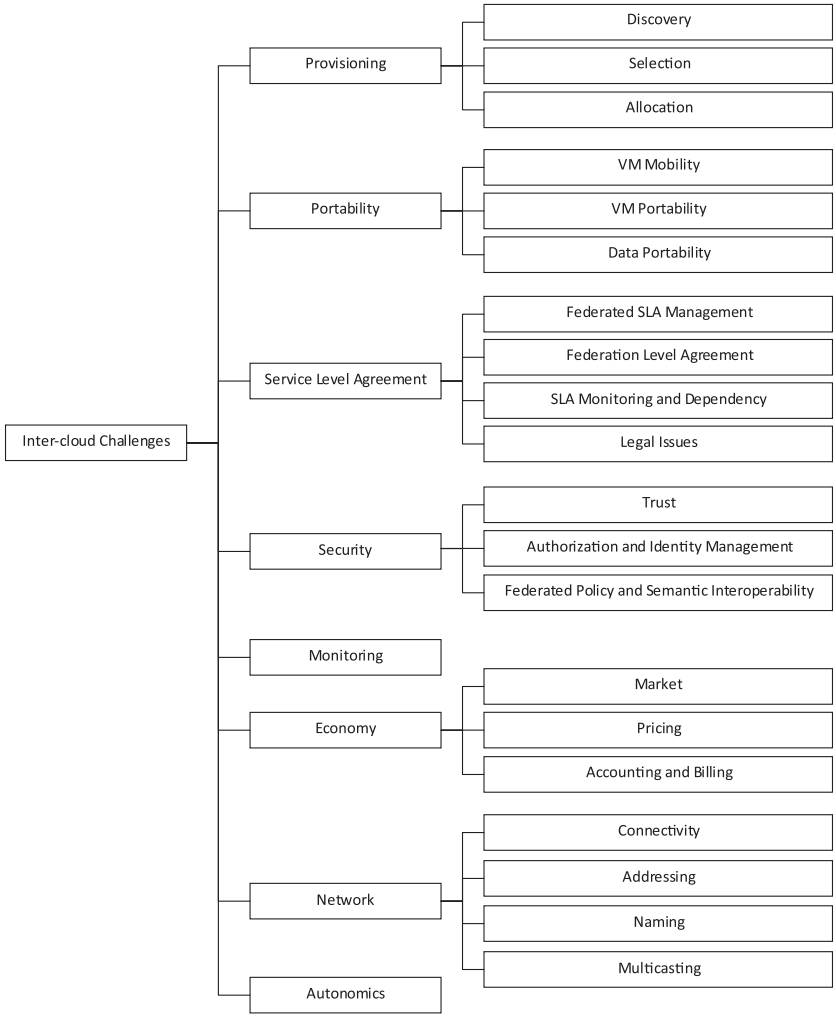

Fig. 11. Taxonomy of Inter-cloud challenges.

5.1. Provisioning

5.1.1. Discovery. Cloud service discovery allows automatic detection of services and resources offered by cloud providers on the Internet. Since cloud providers offer a variety of services and use different ways to describe them, a way to provide a common access to cloud services and to discover and deploy them is necessary. Cloud customers require selection of the best possible application deployments in the cloud according to their objectives and constraints for QoS. Achieving this goal requires effective discovery of the available services and their characteristics. In general, even though there are various forms of cloud services, discovery of services hosted in clouds has not received enough attention yet. For instance, Google App Engine and Amazon EC2 do not offer discovery services, and Microsoft Azure and Force.com offer limited discovery capabilities [Goscinski and Brock 2010].

One of the main issues regarding service discovery in a multiple-cloud deployment is the lack of an integrated repository of cloud services. Ranjan and Zhao [2011] believe

that centralized approaches for an integrated service catalog are not appropriate due to concerns of scalability, performance, and reliability arising from a large volume of service requests. They present a peer-to-peer cloud service discovery over a distributed hash table (DHT) overlay network. In contrast, Bernstein and Vij [2010a] argue that a point-to-point discovery architecture results in the n^2 complexity problem, and they propose *Intercloud Root Instances* and *Intercloud Exchanges* to solve the problem. In fact, Intercloud Root provides a service catalog, which is an abstracted view of the resources across disparate cloud environments.

Another issue is that cloud providers describe their services with diverse languages, terms, and names. Moreover, there is not a common understanding regarding service functionalities, their QoS, and metrics among providers and customers. In a heterogeneous environment such as Inter-cloud, it is difficult to enforce a standard syntax on service description or common metrics. Therefore, the use of syntactic-based approaches such as Universal Description, Discovery and Integration (UDDI)[11] is not applicable. Moreover, the Web Service Description Language (WSDL), which is used by UDDI, does not support modeling of QoS properties and it is difficult to add them. According to several recent studies [Bernstein and Vij 2010a; Moscato et al. 2010], the solution that covers mentioned drawback is a semantic web service that can increase expressiveness, flexibility, and accuracy of the service discovery by the application of ontologies for representation of the service properties. However, a detailed discussion about ontology-based approaches in this regard falls outside the scope of this survey.

Finally, states of a large part of services in clouds change constantly and are dynamic in nature. The situation is even worse in interconnected cloud environments. Consequently, dynamic attributes should be added to cloud services and a web service–based resource. A general framework for service and resource publication, discovery, and selection using dynamic attributes that expose current state and characteristics via web services has been proposed by Goscinski and Brock [2010].

5.1.2. Selection. Optimal application deployment in the cloud requires an effective selection strategy that works based on QoS criteria such as reliability, cost, and security and returns the set of the most suitable cloud services for end-customers. Cloud service selection did not receive much attention in the literature mostly due to the lack of reliable data on cloud services' QoS criteria.

Currently, selection is performed manually by cloud customers based on their requirements or through consultant companies. In multiple cloud application deployment scenarios, selection is not a trivial task due to the diversity of cloud services' characteristics and QoS. However, application deployment across multiple providers benefits from significant features such as the range of geographical locations, lower latency, higher reliability, lower deployment cost, higher failure resistance, and so forth. Consequently, an automated selection approach for application deployment is well motivated to optimize different aspects such as latency, reliability, throughput, data transfer, and cost. In addition, such a selection approach must take into account different constraints such as legal issues or security concerns.

The selection process can be performed either based on static information on the service quality provided by cloud providers or through dynamic negotiation of SLAs. Limited support is currently available for dynamic negotiation of SLAs [Petcu et al. 2011]. In order to design an infrastructure for negotiation and management of SLA in the cloud, several issues need to be addressed. A few cross-cloud projects (e.g., mOSAIC [Petcu et al. 2011]) focus on agent-based dynamic negotiation for cloud services.

[11]A web-based distributed directory that enables providers to list their services and discover each other.

OPTIMIS, a toolkit proposed by Ferrer et al. [2012], focuses on cloud service and infrastructure optimization throughout three phases of the service life cycle: construction, deployment, and execution. In its deployment engine component, the OPTIMIS risk assessor provides multiple-criteria evaluation of cloud services using the Dempster-Shafer Analytical Hierarchy Process (DS-AHP).[12] The criteria that are used for evaluation can be generally classified as past performance, maintenance, security, customer support, and legal aspects. A value between 0 and 1 is computed for each of the criteria by evaluating a provider. These values are used as the basis for the final selection.

Another framework called CloudGenius has been introduced by Manzel and Ranjan [2012]. The framework provides a web application migration process and decision support. Cloud customers are able to migrate web applications to the cloud along a process that suggests cloud VM images and cloud infrastructure services according to requirements and goals of the cloud customer.

Han et al. [2009] present a cloud service selection framework in the cloud market that recommends best services from different cloud providers that match user requirements. Their framework ranks different services with providers and presents it to users so that they can select the appropriate or optimal services.

Resources and services in clouds can be represented by web services. There are considerable works in the context of SOA and grid web service selection. As a result, their contribution can be shared to tackle selection problem in clouds.

5.1.3. Allocation. Service selection on the customer's side leads to resource allocation on the provider's side. Resource allocation is a challenging issue from the cloud provider's perspective. Cloud providers usually offer their virtualized resources based on different QoS levels (e.g., best effort and reserved). Physical resources in clouds are shared between cloud users. Therefore, allocation strategies are needed to allocate resources to the requests in a profitable manner while fulfilling requests' QoS requirements.

As the number of resource consumers is increasing, clouds need to share their resources with each other to improve their quality of service. In general, such a collaborative cloud computing system (e.g., cloud federation) is prone to contention between user requests for accessing resources [Salehi et al. 2012]. Contention happens when a user request cannot be admitted or cannot acquire sufficient resources because resources are occupied by other requests (e.g., requests from the federated cloud provider). We call this issue "resource contention" from here onward.

Resource contention is not a new issue in federated environments. Various solutions have been proposed for the resource contention problem in federated cloud environments and other interconnected distributed computing systems [Salehi et al. 2012; Rochwerger et al. 2009; Toosi et al. 2011]. There is growing interest in the adoption of market-based approaches for allocation of shared resources in computational systems [Mihailescu and Teo 2010b]. Mihailescu and Teo [2010b] propose a dynamic pricing scheme for federated sharing of computing resources, where federation participants provide and use resources. They show that in their proposed dynamic scheme, the user welfare, the percentage of successful requests, and the percentage of allocated resources increase in comparison to the fixed pricing [Mihailescu and Teo 2010d]. Toosi et al. [2011] propose a model for trading of cloud services based on competitive economic models. They consider circumstances in which cloud providers offer on-demand and spot VMs while they participate in a federation.[13] The resource manager unit

[12]The AHP is a structured technique for organizing and analyzing complex decisions and helps decision makers to find one that best suits their goal and their understanding of the problem. The DS theory is a mathematical theory that allows combining evidence from different sources to achieve a degree of belief.

[13]Spot VMs are VMs that can be terminated by providers whenever the current value for running such VMs (defined by the provider) exceeds the value that the client is willing to pay for using such resource.

evaluates the cost-benefit of outsourcing an on-demand request to a third party or allocating resources via termination of spot VMs. Their ultimate objective is to decrease the rejection rate and have access to seemingly unlimited resources for on-demand requests. Gomes et al. [2012] propose and investigate the application of market-oriented mechanisms based on the general equilibrium theory to coordinate the sharing of resources between clouds in the federated cloud. Goiri et al. [2011] propose an economic model that characterizes situations that assist decisions in a federated cloud, namely, when to outsource resources to other providers, when to admit requests from other providers, and how much capacity to contribute to the federation.

5.2. Portability

5.2.1. VM Mobility. The challenges regarding live virtual machine (VM) migration between physical nodes under the same administrative domain has been addressed previously in both industry and academia. The main challenge in this regard is that VMs require storage and network services from their hosts, and once a VM is live migrated from a host to another host, it still requires access to the storage and network services of the source host. Traditional support for live VM migration resolved this issue by a shared storage device and hosts, which are connected to the same local area network (LAN) [Nagin et al. 2011]. However, storage and network environments of different clouds are generally independent and are separated by firewalls.

VM Mobility is defined as the ability to move a running VM from one host to another without stopping it [Dowell et al. 2011]. In the Inter-cloud scenario, the cloud application might require VM Mobility. Moreover, Inter-cloud VM Mobility should not violate the independence of the respective clouds in terms of autonomy, privacy, and security. VM migration, from a source cloud to a destination one over a wide area network (WAN), constitutes transferring memory, status, and storage of the VM. According to Nagin et al. [2011], cross-cloud VM migration requires the following:

(1) Memory and state transfer between hosts residing in different data centers
(2) Same LAN access by VMs at the destination host, without two sites sharing LAN
(3) Same storage access by VMs at the destination host, without two sites sharing storage

VM Mobility requires a VM transfer from one physical machine to another without disrupting the network traffic flow. Some hypervisors allow a running VM to migrate from one physical machine when they are connected to the same local area network (e.g., XEN and KVM). However, long-distance VM Mobility between sites with separate LANs in a federated cloud environment, as we stated earlier, must ensure that migrated VM will have access to the same LAN at the destination host, without a sharing LAN between two sites.

The suggested approach is extending LANs between sites using WAN encapsulation technologies, like F5, a global traffic manager, by VMware[14] or VMware and Cisco Migration Solution.[15] However, such LAN extensions may violate providers' IT infrastructure autonomy, security, and privacy requirements. A virtual networking technology is needed that will allow VMs connected to the same virtual network to communicate with each other over a private and isolated virtual network within and across clouds, systems like Vine [Keahey et al. 2009] and VNET [Sundararaj et al. 2004]. Regarding this issue, Nagin et al. [2011] propose proxy servers at the source and destination clouds that communicate with each other while hiding the details of the source and

[14]http://www.f5.com/pdf/solution-center/f5-for-virtualized-it-environments.pdf.
[15]http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns836/white_paper_c11-557822.pdf.

destination hosts. In Section 5.6, we discuss challenges regarding VM Mobility in more detail.

5.2.2. Data Portability. Users or applications that store data in the cloud, especially for SaaS and PaaS applications, often require access to the data so that it can be used by services of other cloud providers. Giving users control over their data is an important part of establishing trust and is paramount for creating interconnected cloud environments allowing users to easily move their data from one cloud to another [Fitzpatrick and Lueck 2010]. If a cloud provider stores data in their own proprietary format, then users cannot move their data to other vendors without considerable cost and technical effort. Therefore, industry standards and exporting tools, or at the very least formats that are publicly documented, are required to avoid data lock-in. Nowadays, data portability is hindered by the lack of proper technology and standards and nonportability of the applications and data, which is exploited by cloud service providers for their own benefits [Petcu et al. 2013].

According to Hill and Humphrey [2010], solutions for avoiding data lock-in can be classified into the following categories:

(1) Using APIs that have multiple independent implementations, for example, Amazon EC2 APIs, which are used by several others such as Eucalyptus [Nurmi et al. 2009];
(2) Choosing a particular API that can run on multiple Clouds, for example, MapReduce and Hadoop;
(3) Manually decoupling the cloud-specific code of the application designed for each cloud provider from the application logic layer;
(4) Creation of widespread standards and APIs; and
(5) Utilization of vendor-independent cloud abstraction layers such as jclouds[16] and libcloud.[17]

In fact, the main obstacle for data movement in interconnected cloud environments is the lack of standard metadata and data formats. To deal with this issue, platform-independent data representation and standardization of data import and export functionality between providers are needed [Petcu et al. 2013].

Google [Fitzpatrick and Lueck 2010] attempts to address this problem through its *Data Liberation Front,*[18] whose goal is to make it easier to move data in and out of Google products. This is a step toward what is called *data liberation* by Google and provides freedom of data movement between clouds. The data liberation effort focuses specifically on tools and methods that allow users to export any data they create and import into another service or competing products.

CSAL [Hill and Humphrey 2010] is a *Cloud Storage Abstraction Layer* to enable portable cloud applications and supports three storage abstractions: *Blobs*, *Tables*, and *Queues*. CSAL implementation provides a single unified view of cloud storage across platforms and manages the metadata necessary for utilizing storage services across multiple clouds.

Bernstein and Vij [2010c] proposed the Simple Storage Replication Protocol (SSRP) for a federated cloud environment that facilitates distributed unstructured storage (e.g., Blobs) connectivity between clouds in a point-to-point manner.

Petcu et al. [2013] proposed APIs for the mOSAIC project [Petcu et al. 2011] that provides portable cloud application development solutions.

[16]jclouds, http://code.google.com/p/jclouds/.
[17]libcloud, http://libcloud.apache.org/.
[18]Data Liberation Front, http://www.dataliberation.org/.

Bermbach et al. [2011] present MetaStorage, a federated cloud storage system that replicates data on top of diverse storage services using scalable distributed hash tables.

5.3. Service-Level Agreement

5.3.1. Federated SLA Management. Cloud providers define (or negotiate with customers) a service-level agreement (SLA) to specify what they guarantee. In a simple definition, SLA is a contract that describes a service and, most importantly, sets the expected service-level objectives (QoS expectations). It can even encompass more details such as penalties applied to the provider if it does not deliver services according to the service-level objectives. Implementation of SLA mechanisms on top of federated resources is still an open question. Since it is an area whose volume of ongoing works enables a survey of its own, we are not intended to cover all issues related to the SLA management in this article.

In federated cloud environments, it is expected that each participant cloud provider has its own SLA management mechanisms. Since user applications in such an environment exploit services and resources from different providers, one role of the federation is to set up and enforce a global SLA. By global SLA, we mean comprehensive SLAs between the user and the federation including all SLAs for each cloud provider. The federation should monitor the application in order to verify that the SLA is met by providers and should react to SLA violations.

In federated cloud environments, the entity that acts as a mediator between the cloud consumer and interoperable cloud providers must select services from different providers that better meet the user requirements. In such a dynamic environment, cloud providers can offer or meet guarantees according to their resource situation at the time the service is requested. Moreover, the service provided for the user might be composed of services from different providers. Hence, methods and protocols for negotiation of dynamic and flexible SLAs is a must for dynamic environments such as Inter-cloud. This demands that agreements are established dynamically at the time the service is requested, rather than advertised as an invariant property of the service.

A similar scenario happens when a broker of service acting on behalf of the user selects services among multiple cloud providers. There are several proposals addressing the negotiation of dynamic and flexible SLAs in service-oriented environments including *WS-Agreement* [Andrieux et al. 2004] and *WS-Agreement Negotiation* [Battré et al. 2010]. WS-Agreement provides language and protocols for creation agreements based on offers and for monitoring of agreement compliance at runtime. WS-Agreement supports a one-round negotiation process, whereas WS-Agreement Negotiation can be used when a more flexible and dynamic negotiation process is required. WS-Agreement Negotiation provides renegotiation capabilities on top of the WS-Agreement specification.

Another important issue in the federated cloud environment is how SLAs can be enforced in a federation where there are conflicting policies and goals of different members versus the objectives of the federation as a whole. For example, the federation layer can offer an SLA that promises highly reliable service, while none of the federation members, which are self-interested parties trying to maximize their revenue, are willing to offer such a service, which is costly.

There are a few works in the literature that addressed SLA management in the context of the federated cloud environment. Contrail [Carlini et al. 2012] is a project that proposes a federated and integrated cloud architecture. They provide extended SLA management functionalities by integrating the SLA management approach of the SLA@SOI[19] project in the federation architecture. The Contrail federation coordinates the SLA support of different cloud providers. As cloud providers in the cloud federation

[19]SLA@SOI, http://sla-at-soi.eu/.

have their own SLA management mechanisms, Contrail tries to set up, coordinate, and enforce a federation-level SLA. In the context of the mOSAIC project [Petcu et al. 2011], there also exists facilities in order to offer user-oriented SLA services to final users [Amato et al. 2012]. Cuomo et al. [2013] propose an SLA-based broker operating in a volunteer computing environment to discover the most suitable resources to a user when resources are not reliable and can provide QoS comparable to those offered by commercial cloud providers.

5.3.2. Federation-Level Agreement. In addition to SLA, there can be a contract—so-called Federation-Level Agreement (FLA)—that includes the set of rules and conditions that has to be signed by new providers once they join the federation [Toosi et al. 2011]. For example, a federation can set rules for minimum resources contributed to the federation or the set of QoS such as minimum expected availability. In a more complex scenario, the federation can set different rules to have multiple pools of resources with different QoS guarantees.

5.3.3. SLA Monitoring and SLA Dependency. In federated cloud environments where a provider outsources its load to another provider, it expects a set of guaranteed QoS that is compatible to the promised QoS to end-users. Therefore, either the federation has to provide a service to match end-user QoS requirements or cloud providers have to do it on their own. In addition, in the cloud federation environment, there might be dependencies between performances of services provided by different providers. As explained in the "Practical Guide to Cloud Service Level Agreements by Cloud Standards Customer Council,"[20] there can be an environment where a cloud user has an SLA with a provider, and the cloud provider by itself has SLAs with two cloud providers and utilizes their resources to provide services to the end-user. Therefore, quality of a service can be affected by external services. It means that if one of the lower-layer services (e.g., infrastructure layer) is not functioning properly, it can affect the performance of higher-layer services (e.g., software layer). A practical approach is required to model the dependencies among services.

Winkler et al. [2010] propose an approach for automated management of SLAs to support composite services. In that approach, the explicit knowledge about a set of dependencies to automate the tasks of negotiation and renegotiation of SLAs and the handling of service-level objective (SLO) violations are taken into account.

Bodenstaff et al. [2008] propose an approach called MoDe4SLA to monitor dependencies between SLAs when managing composite services. In that case, different types of dependencies between services and the impact that services have on each other are analyzed during the development phase. The approach has not provided an Inter-Cloud language to share common understanding regarding the QoS criteria and their measurement units.

Dastjerdi et al. [2012] show how dependency knowledge can be modeled using semantic technology and how that knowledge can be used in discovery of monitoring services and SLA failure detection. The major contributions of the work are modeling services' performance interdependencies and elimination of SLA failure cascading effects on violation detection.

5.3.4. Legal Issues. Interconnected cloud computing environments extricate applications from being confined to a single data center and open opportunities for globalizing and integrating services from multiple and disparate geographies. Besides technical complexities of interconnecting clouds, legal issues might arise with the realization of

[20]Practical Guide to Cloud Service Level Agreements: A white paper by the Cloud Standards Customer Council, http://www.cloudstandardscustomercouncil.org/2012_Practical_Guide_to_Cloud_SLAs.pdf.

Inter-cloud. Cloud computing by itself requires consideration of a broad set of legal issues such as privacy and security, contracting issues, issues related to location and ownership of data, and business considerations [Bowen 2010]. Inter-cloud makes existing legal issues more complicated and also introduces new ones. A major part of these issues requires defining new laws and regulations beyond technical innovations and falls out of the scope of this survey. Readers who are interested in where legal issues might arise in cloud federation are referred to Kertesz and Varadi [2014].

When different organizations are involved in providing services for customers, one major issue is that it is difficult to guarantee confidentiality and privacy on data, especially when data is located in different countries with different laws. Cloud providers in interconnected cloud environments must provide mechanisms to guarantee the security and privacy of sensitive data within legal borders. For example, in a federated scenario in which a cloud provider leverages another provider's services and customers might not generally have control of where the service they are leasing is operating, in case of failures in the service delivery it will be difficult for the cloud user to identify the real causes. Moreover, legislation and laws concerning the privacy and security of data are different among different countries, and even among different states within the same country. For instance, based on the European Union (EU) directive, any personal data generated within the EU is subject to the European law and data cannot leave the EU unless it goes to a country that provides an adequate level of protection [Bowen 2010]. The US PATRIOT Act[21] allows the U.S. government to gain access to personal financial information and student information stored in electronic systems just by providing a governmental certificate that the information might be relevant to criminal activities [Bowen 2010].

From a technical point of view, in interconnected cloud environments, application deployment requires that juridical and legislative restrictions be considered. Therefore, geo-location and legislation awareness policies must be imposed into the entity that acts as a mediator between the cloud consumer and interoperable Cloud providers (e.g., broker or federation layer) [Grozev and Buyya 2012], and compliance with such policies and agreement must be enforced. For instance, as part of the SLA management system, services of specific vendors can be avoided or placing the data outside a given country can be prohibited.

Management of legal issues for Inter-cloud requires defining a comprehensive body of laws compliant with all the legislation of the countries involved in possible transactions among cloud providers and users. This falls beyond technical aspects and constitutes efforts by legislatures to facilitate Inter-cloud by defining proper laws and regulations.

5.4. Security

5.4.1. Trust. In a social context, trust typically refers to a situation where one party is willing to rely on the actions of another party. The control over the actions is abandoned by the former to the latter. Ultimately, there is uncertainty as to whether the trusted party will behave or deliver as promised.

In the cloud computing environment, customers must trust in a cloud provider for the privacy and security of their assets (i.e., their data and processes). The degree of lost control over the data and processes depends on the cloud service model. In cloud computing, the risk of losing data confidentiality, integrity, and availability for customers is triggered by the lack of control over the data and processes [Khan and Malluhi 2010].

[21]Uniting and Strengthening America by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism Act of 2001 (USA PATRIOT Act).

In interconnected cloud environments, establishment of trust is even more important and complex [Bernstein and Vij 2010b], as besides the customers, cloud providers must trust each other. In the Inter-cloud scenario, the trust and reputation of a cloud provider affect other cloud providers. Specifically, as Inter-cloud computing constitutes collaboration between independently owned autonomous clouds, and these providers peer with each other to outsource requests or data, there is a need for mechanisms to evaluate the trustworthiness of a federation member [Abawajy 2009].

Apart from reputation-based trust, because of the criticality of many computing tasks and diversity and vastness of services in Inter-cloud environments, formal trust mechanisms are required to help Inter-cloud entities (i.e., providers and users) to trust each other. A formal process for assessment of cloud services and their providers in such a highly dynamic system requires independent third parties that are acceptable to all entities. The concept of *trust federations* can be used in this regard. Trust federation is a combination of technology and policy infrastructure that allows organizations to trust each other's verified users to enable the sharing of information, resources, and services in a secure and distributed way.

The International Grid Trust Federation (IGTF)[22] is a prominent example of a trust federation that can be leveraged for the Inter-cloud scheme as well. The IGTF is an organization that fosters harmonization and synchronization policies with the goal of enhancing establishment of cross-domain trust relationships for intergrid participants. Grid entities use X.509 certificates for authentication and authorization. These certificates are issued by the Certificate Authorities (CAs) that are part of the IGTF. In order to ensure compliance with established policies and guidelines, these CAs should be externally audited periodically. The IGTF has established such sets of policies and guidelines and ensures compliance to them between its members.

If we assume that the challenges regarding the trust evaluation of an Inter-cloud provider has been overcome and a provider is able to find a trustable party to peer with, we still face the challenge of building a trusted context for interconnected providers. In this trusted context, providers must be able to access each other's services while they still adhere to their internal security policies.

Currently, public key infrastructure (PKI) is the common trust model in cloud environments [Bernstein and Vij 2010b]. A PKI is a system that verifies a particular public key belongs to a certain entity. The process is done through the creation, storage, and distribution of digital certificates. The PKI creates digital certificates that map public keys to entities, stores these certificates in a central repository securely, and revokes them if needed.

The current PKI certificates-based trust model only checks if the entity is either trusted or nontrusted. However, an all-or-nothing trust model is not appropriate for the Inter-cloud environment in which cloud providers may have different levels of trust in each other [Bernstein and Vij 2010b]. Thus, they suggest a dynamic trust-level model layered on top of the PKI certificate-based trust model. Trusted context has been previously investigated in other collaborative environments. However, customized enabling technologies such as XACML[23] and SAML[24] were also proposed to build a trusted context for cross-cloud federation [Celesti et al. 2010a]. Abawajy [2009] proposes a distributed framework that enables parties to determine the trustworthiness of other entities. The proposed trust framework is a reputation-based trust management system that enables a service requester to obtain the trustworthiness of the service.

[22]International Grid Trust Federation (IGTF), http://www.igtf.net/.

[23]eXtensible Access Control Markup Language (XACML), OASIS, https://www.oasis-open.org/committees/xacml/.

[24]Security Assertion Markup Language (SAML), OASIS, https://www.oasis-open.org/committees/security.

5.4.2. Authorization and Identity Management. Identity Management (IdM) is an administrative task that deals with authentication of individuals in a system and authorization to access resources of the system based on the associated rights and restrictions. In cloud computing environments, identity management services are mainly responsible for supporting access control to services based on user attributes (e.g., IP address, user and group name) and resource attributes (e.g., availability periods).

Identity management systems, in federated cloud environments, should allow identification of users and resources in addition to support interoperability across multiple identity domains [Núñez et al. 2011]. In such a scenario, users should be able to access various resources and services offered by different service providers once they are successfully authenticated in the Inter-cloud interface [Núñez et al. 2011]. One of the problems related to this issue is how to free users from the burden of authenticating with resources from multiple cloud providers. In other words, since each cloud has its own authentication mechanism, a standard method that provides Single Sign-On (SSO) authentication within Inter-cloud environments should be deployed. This must be applied both for customer–provider and provider–provider interactions.

In a federated environment, the SSO issue can be achieved through the *delegation of trust* that allows an entity to act on another entity's behalf. This is especially important when resources and services of different service providers are involved in serving an Inter-cloud application and it might be redundant or very expensive to authenticate each and every time a user or application has to access the resource. Utilizing proxy certificates is a common way of delegating trust that is successfully used in grid computing and service computing. This method allows entities of a federated system to securely interact and might require multiple levels of delegation by establishing a *chain of trust* of proxy certificates. SSO can also be achieved through the use of a trusted third party who will certify credentials on behalf of all parties in the federation. In fact, instead of knowing all possible entities, it is enough to be able to verify claims from the trusted third party.

Effective identity management in Inter-cloud environments requires support for established standards such as X.509 certificates, SAML, and WS-Federation [Bernstein and Vij 2010b]. These standards use different "languages" to express the identity information. A thorough solution is required to deal with these incompatibilities. Moreover, to solve the problem of different formats, names, and meanings for identity attributes, identification of common attributes or use of ontology is also suggested [Núñez et al. 2011]. Consequently, interoperability of identity management systems is a key issue that has to be taken into account.

Another issue that must be considered is how to manage the life cycle of identities. Typically, in the Inter-cloud environment, digital identity information is required in many directories and data stores, but it is hard to keep them synchronized with each other and remove or disable entries when required. In this direction, Service Provisioning Markup Language (SPML) proposed by OASIS is a possible solution [Núñez et al. 2011].

Celesti et al. summarize the requirements of Inter-cloud Identity Management in two categories [Celesti et al. 2010a]:

(1) Single Sign-On (SSO) authentication, where a cloud must be able to authenticate itself to gain access to the resources provided by federated foreign clouds belonging to the same trust context without further identity checks
(2) Digital identities and third parties, where a cloud has to be considered as a subject distinctively identified by credentials and each cloud must be able to authenticate itself with foreign clouds using its digital identity guaranteed by a third party

They propose an Inter-cloud Identity Management Infrastructure according to the selected technologies ranging from XMPP and XACML to SAML [Celesti et al. 2010c]. Support of XACML-compliant entitlement management is highly desirable for the Inter-cloud environment [Bernstein and Vij 2010b]. XACML provides a standardized language and method of access control and policy enforcement.

Another main problem that must be taken into account for Identity Management in large interconnected cloud environments is *scalability*. Performance of any IdM needs to be scalable and operation must be agile and quick. With the current technologies for IdMs, security must be compromised in favor of scalability. Methods such as PKI perform based on the *top-down* approach, where each entity starts out knowing the Root Certificate Authority (CA) and retrieves all the certificates from the Root down to its own key. Root CA is the only trusted body to certify name-to-key mappings. However, for scalability purposes, we require a hierarchy of CAs to be used. Approaches such as *Friend-of-a-Friend (FoaF)*[25] that do not rely on the root of trust can be helpful in this regard. FoaF provides machine-readable ontology for describing persons, their activities, and their relations to other people and objects and commonly used in social networks. Although methods like FoaF provide less strong security, they are highly scalable and obviate the need for a centralized database.

All in all, Federated Identity Management (FIM) is a critical step toward the realization of Inter-cloud. Identity federation can be accomplished in a number ways, from use of formal Internet standards, such as SAML and XACML specifications, to open-source technologies or other openly published specifications, such as OpenID,[26] OAuth,[27] and WebID.[28]

Cloud computing provides on-demand resource provisioning using vast amounts of data and computing resources in centralized data centers; it was not designed based on the idea of federating distributed computing resources in geographically dispersed locations like a grid computing environment. Most of the security concerns in interconnected cloud environments underlie the coordinated resource sharing and problem solving in dynamic multiple administrative domains. The concept of *Virtual Organizations (VOs)*[29] defined in grid computing is highly relevant in this regard. For instance, Makkes et al. [2013] present the Inter-cloud Federation Framework that attempts to reuse successful experiences of VOs within grids.

5.4.3. Federated Policy and Semantic Interoperability. Various cloud providers may collaborate in Inter-cloud environments to provide aggregated services for clients. In other words, providing service for the application might consist of multiple services from different providers. These cloud providers can have different privacy mechanisms and security approaches. This heterogeneity must be addressed, and mechanisms are necessary to securely handle such a dynamic collaboration and authorization to use resources (data, services, etc.) during the interoperation process. Hence, providers should carefully manage access control policies and should agree upon some well-understood common *federated policy* to ensure that integration does not lead to any security breaches [Takabi et al. 2010].

Achieving such agreements requires some degree of *semantic understanding* or *semantic interoperability* due to the domain heterogeneity and different access policies each service provider offers. Solutions like Shibboleth [Needleman 2004], VOMS

[25]Friend-of-a-Friend (FoaF) project, http://www.foaf-project.org/.
[26]OpenID, http://openid.net/.
[27]OAuth, http://oauth.net/.
[28]WedID, http://webid.info/.
[29]Virtual Organization (VO) is defined as a collection of individuals and institutions that access and share resources for the purpose of one or more identified goals within the grid.

[Alfieri et al. 2004], or XACML provide different methods to enable administrators to apply authorization policies; however, these approaches are designed to describe security policies and they are not able to handle the semantics associated with the elements they are protecting. Moreover, they do not properly cope with policy conflicts nor detect risky security situations as a result of errors in definition of complex policies. Semantic approaches for policy representation provide the ability to analyze policies related to entities described at different levels of abstraction.

In order to achieve a high level of expressiveness for policy definitions, Pérez et al. [2011] present an authorization architecture that relies on semantic web technologies. Heterogeneity of multiple organization domains and support for different policies have been taken into account by providing the capability to describe the semantics of the resources that are to be protected. Hu et al. [2009] propose a semantic access control approach applying semantic web technology to access control in cloud computing environments. Singhal et al. [2013] present a generic cloud collaboration framework that allows cloud user applications to use services from multiple clouds without prior business agreements among cloud providers and without adoption of common standards.

In summary, security challenges related to interconnected cloud environments are numerous, and it is not in the scope of this article to cover all the security-related issues of this area. There is a wealth of literature dealing with security aspects in distributed environments (e.g., grid), which are closely connected to interconnected cloud environments. Hence, many Inter-cloud security-related issues can be addressed based on the experiences in grid computing. Interested readers are referred to Bernstein and Vij [2010b], Chakrabarti et al. [2008], and Singhal et al. [2013] for more details regarding security concerns and challenges in interconnected cloud environments.

5.5. Monitoring

Cloud monitoring is a broad term that means monitoring of various aspects of the service, from VM performance to a very complicated monitoring of mutually dependent services in the cloud. Monitoring systems are required to monitor performance of physical and virtual resources and running cloud applications. A monitoring system can audit and aggregate data to help an administrator or a service manager to make sure the applications and contents are performing properly. In other words, the monitoring system gathers data from all the components within the cloud architecture and provides the data for the infrastructure and service management. Monitoring data is used for different purposes such as enforcing SLAs, enabling elasticity, ensuring QoS, and so forth.

Monitoring of cloud-based applications can be achieved in two separate levels of the *infrastructure* and the *application*. Infrastructure-level resource monitoring aims at measuring and reporting system parameters related to the infrastructure services offered to the user such as CPU, RAM, or data storage parameters. According to Aoyama and Sakai [2011], in a federation of clouds, infrastructure-level resource monitoring data can be collected about the usage status and dead or alive status of computing, storage, and network resources of a cloud system. This monitored data is required to determine the need for load distribution or even disaster recovery. On the application level, the monitored parameters and the way their values should be retrieved depend on the application and not on the cloud infrastructure it is running on. In an integrated cloud environment like a cloud federation, a general monitoring infrastructure is required to collect and process the information provided by the monitored components regardless of the level of parameters being monitored [Rak et al. 2011].

Existing monitoring systems such as Ganglia [Massie et al. 2004], Nimsoft,[30] Nagios,[31] and GridICE [Andreozzi et al. 2005] addressed monitoring of large distributed systems, but the rapidly changing and dynamic nature of services in clouds cannot be addressed thoroughly by these systems. In a federated cloud environment, monitoring is a more complicated task because of the diversity of the clouds and different domains they exist in. Resources may reside across different cloud infrastructures, so the monitoring system must collect and aggregate data from heterogeneous cloud environments. As a result, standardized interfaces and formats are required to enable the federation monitoring.

In such a federated environment, when virtual resources are migrated from one site to a remote site, the monitoring data from the remote site still needs to be collected by the service manager in the origin destination. Moreover, by migrating to a remote site, the originating site loses direct control of the virtual resource and underlying hardware. In order to ensure a continuous monitoring capability, each of the clouds needs federation components and objects to be created and managed to support remote monitoring [Clayman et al. 2010].

As cloud applications get larger and larger and are scattered across clouds, the need for an autonomic monitoring framework that works without intervention and reconfiguration arises. Monitoring tools must be designed to support autonomic federated cloud monitoring. In addition to basic monitoring, in a large environment such as Inter-cloud, the monitoring system requires mechanisms that allow a service to receive messages when events occur in other services and applications. Services interested in receiving such messages are often unknown in advance or will change over time. Therefore, services must be able to register (subscribe) interest for receiving events from the event sources. Experiences with WS-Eventing[32] can be helpful in this regard.

Rak et al. [2011] present monitoring components that facilitate the development of custom monitoring systems for cloud applications using the mOSAIC project [Petcu et al. 2011] APIs. Clayman et al. [2010] present the Lattice[33] framework, which has been specially designed for monitoring resources and services in virtualized environments such as the RESERVOIR project [Rochwerger et al. 2009].

5.6. Economy

5.6.1. Market. Interoperability between different providers allows cloud customers to use the service across clouds to improve scalability and reliability [Mihailescu and Teo 2010c]. Computing as a utility can be considered as one of the main goals in federated cloud computing where resources in multiple cloud platforms are integrated in a single resource pool. A key challenge in this regard is how cloud providers interact with each other to realize collaboration [Zhang and Zhang 2012].

A cloud provider is able to meet the peak in resource requirements by buying resources from other cloud providers. Similarly, when a cloud provider has idle resources, it can sell these resources to the federated cloud market. In order to enable such resource sharing and collaboration among cloud providers, there is a need for a marketplace with exchange facilities that helps providers in trading resources among each other [Toosi et al. 2011].

Buyya et al. [2010] propose a federated network of clouds mediated by a cloud exchange as a market maker to bring together cloud providers and customers. It supports trading of cloud services based on competitive economic models such as commodity

[30]Nimsoft, http://www.nimsoft.com/index.html.
[31]Nagios, http://www.nagios.org/.
[32]Web Service Eventing (WS-Eventing), http://www.w3.org/Submission/WS-Eventing/.
[33]Lattice framework, http://clayfour.ee.ucl.ac.uk/lattice/.

markets and auctions. Federated cloud providers require a clear understanding of the ramifications of each decision they make regarding selling/buying resources to/from other providers. Goiri et al. [2011] present a plausible characterization of providers decisions operating in a federated cloud including outsourcing requests or renting idle resources to other providers.

Market-based approaches for allocation of shared resources have proven their potential in computational systems [Mihailescu and Teo 2010a]. To address the market-based resource allocation mechanism design problem, Mihailescu and Teo [2010b] propose a reverse auction-based mechanism. The market maker selects the sellers for allocation, based on the published price, such that the underlying resource costs are minimized. Afterward, the actual payments for the winning sellers are determined based on the market supply.

5.6.2. Pricing. Pricing and profit are two important factors for cloud providers to remain in the business [Toosi et al. 2011]. Cloud federation allows providers to trade their resources under federation regulations. Strategies regarding selling and buying of resources in federated cloud environments are important issues that should be considered by providers. How providers price their services in the federated cloud market requires profound considerations to ensure the profitability of the providers. To be precise, it is important that the provider has a clear understanding of the potential of each federation decision, and providers should answer questions such as to what extent they want to contribute to the federation or how much they should charge other providers for their service.

Goiri et al. [2011] present a profit-driven policy for decisions related to outsourcing or selling idle resources. They characterize these decisions as a function of several parameters and implement a federated provider that uses this characterization to exploit federation. Toosi et al. [2011] propose similar policies and a way to price resources in the federated cloud. Furthermore, they proposed a financial option-based cloud resource pricing model to help providers in the management of reserved resources [Toosi et al. 2012].

Dynamic resource pricing is a necessity in interconnected cloud environments where distributed cloud providers seek to accommodate more customers while they compete with each other. Mihailescu and Teo [2010b] argue that dynamic pricing is more suitable for federated sharing of computing resources, where participants may both provide and use resources. They present an auction framework that uses dynamic pricing to allocate shared resources. They show that using their proposed dynamic pricing scheme, the user welfare, the percentage of accepted requests, and the percentage of allocated resources increase in comparison to fixed pricing.

5.6.3. Accounting and Billing. In a federated cloud environment, accounting and billing must be carried out in a way that meets the requirements of the cross-cloud scenario. Some identified challenges may affect the design of the accounting and billing in this environment; the actual placement of the resources may not be known to the entire system and may also change during the service lifetime. Moreover, the number of required resources composing a service can dynamically go up and down to cope with a change in demand [Elmroth et al. 2009]. Primarily, it is required that resource usage be monitored for billing and accounting purposes. Additionally, in federated cloud environments, cloud providers expect the federation to be honest in its accounting and billing practices [Harsh et al. 2011].

Any accounting and billing approach must be performed in a fair and standardized way both (a) between cloud customers and Cloud provider and (b) between cloud providers [Elmroth et al. 2009]. Moreover, for billing, those approaches must take into account the postpaid and prepaid payment schemes for capacity that varies over time

in response to customer requirements. Elmroth et al. [2009] present a solution for accounting and billing in a federated cloud environment. The focus of the work is in the design of the accounting and billing system, utilizing existing alternatives, for the RESERVOIR [Rochwerger et al. 2009] project. They focused on accounting and billing between a cloud provider and consumer. However, Inter-cloud accounting and billing still remain as an open issue and need further considerations.

5.7. Network

5.7.1. Network Virtualization and Connectivity. Network connectivity over distributed resources, for example, deployment of a "virtual cluster" spanning resources in different providers, is a challenging issue for both users and providers. Providers are not willing to give users privileged access to the core network equipments because of security risks. Furthermore, creating APIs that reconfigure the network infrastructure based on the user requirements is also difficult. Consequently, creation of a trusted network environment faces challenges in terms of connectivity, performance, and management [Keahey et al. 2009].

One common method of addressing the connectivity problem involving resources in multiple sites is creation of a *virtual network* based on *network virtualization* technologies. A virtual network is an overlay network that consists of virtual resources, such as virtual network interface cards, rather than physical resources and is implemented using methods of network virtualization. The concept of network virtualization has appeared in the networking literature in the past and can be applied to the interconnected cloud environment.

EUCALYPTUS [Nurmi et al. 2009], an open-source cloud platform, provides support for VLANs across multiphysical hosts and requires the VLANs capable managed switch. To support applications that are required to be deployed and migrated across clouds, RESERVOIR [Rochwerger et al. 2009] employs an overlay network between hypervisors. These overlays are called virtual application networks (VANs). Keahey et al. [2009] propose ViNe for Sky Computing that offers end-to-end connectivity among nodes on the overlay network even if nodes are in private networks or protected by firewalls. ViNe supports multiple, mutually isolated virtual networks, which providers can dynamically configure and manage. Regarding this issue, Nagin et al. [2011] propose proxy servers at the source and destination clouds that communicate with each other while hiding the details of the source and destination hosts. VNET [Sundararaj et al. 2004] is a virtual private network that implements a virtual local area network spread over a wide area using layer 2 tunneling. VNET is an adaptive overlay network for virtual machines and is not designed for specific applications.

5.7.2. Addressing. One main challenge in the implementation of long-distance VM migration is the addressing issue. In a virtualized environment like cloud, in the near future the IPv4 address space will not be sufficient as millions of VMs will be running and each one could have a handful of IP addresses associated with it. Therefore, many cloud operators are considering switching to IPv6, which provides much larger local address space. The fact that some cloud builders will use IPv4 and some will use IPv6 is not far-fetched. As a result, a common IP mobility scheme between these two is required [Bernstein et al. 2009].

When a running VM migrates from one location to another, the IP address goes with the running VM and any application hosted by that VM. Both *location* and *identity* are embodied by IP addresses. That is, routers and switches of the network not only identify the endpoint but also infer the location of the endpoint from the IP address. In federated cloud environments, where VMs migrate between geographically distributed

sites, mobility of IP addresses is a challenge. Nevertheless, mobile IP mechanisms[34] can be used in this case. However, they are not common between IPv4 and IPv6. The new scheme is called Location Identity Separation Protocol (LISP) and has been proposed by Bernstein et al. [2009] to operate with both IPv4- and IPv6-based networks. LISP facilitates the IP mobility by decoupling location and identity. In summary, any addressing scheme should consider the mobility aspects of VMs in federated cloud environments [Bernstein et al. 2009].

5.7.3. Naming. In federated cloud environments, services, workloads, and applications are distributed across multiple locations and those locations may change on a frequent basis. Finding those services and scaling the rate of change effectively need an efficient dynamic cloud naming system. The Domain Name System (DNS) is designed to locate and address hosts, services, or any resource connected to the Internet. But the flat name space and a simple name lookup that DNS represents is not sufficient for cloud computing. In fact, clouds are not endpoints in the way servers or clients on the Internet are [Bernstein et al. 2009]. Cloud computing environments are endlessly changing environments. In order to enable effective identification of the required service, its capabilities, and required features, audit capabilities are required in the design of a cloud naming system [Bernstein et al. 2009]. Moreover, in federated environments, a cloud naming system should be able to manage frequent name alteration and name space integration. A cloud entity being part of a virtual cloud application could later become part of another cloud application [Núñez et al. 2011].

Clouds include many entities that need to be identified [Celesti et al. 2010d]. In order to enable cloud platforms to manage and control their resources, they need to name, identify, and locate them [Celesti et al. 2010d]. The nature of the resources involved in the cloud computing paradigm varies from physical components (servers, storage units, etc.) to abstract elements (virtual machines, data repositories, applications, etc.). All these real or abstracted entities are offered to users and can be seen as entities of the cloud [Núñez et al. 2011]. In an Inter-cloud scenario, clouds themselves could be seen as potential resources to be exploited, in the form of a high-level component capable of offering computation, storage, and networking.

5.7.4. Multicasting. Cloud computing is a suitable platform for applications with a large number of users and data like multimedia-enabled applications such as Facebook and MySpace. Multicasting is effectively exploited in these massive-scale, real-time, multipoint applications. Cloud providers mostly do not allow IP multicasting[35] on their networks, as it imposes a high load on their routers. However, it is a crucial element for the aforementioned applications where multicasting must be supported for their implementation. More significantly, for these types of applications to work in the Inter-cloud context, IP multicast between clouds must be supported. Consequently, interdomain multicasting is required [Bernstein et al. 2009]. This becomes further complicated if a location-agnostic addressing scheme has been adopted, as discussed earlier. Cisco has been actively involved in work in this area.[36]

5.8. Autonomics

With the growing complexity of interconnected systems such as Inter-cloud, system management duties become too complex to be carried out only with human intervention

[34]IP Mobility Support for IPv4, revised, at http://www.ietf.org/rfc/rfc3344.txt, IP Mobility Support in IPv6, at http://www.ietf.org/rfc/rfc3775.txt.
[35]IP multicast is a method of sending Internet Protocol (IP) datagrams to a group of interested receivers in a single transmission.
[36]LISP for Multicast Environments, http://tools.ietf.org/html/draft-farinacci-lisp-multicast-01.

and manual administration. Hence, to overcome the issue, the need for *autonomic computing* becomes more and more tangible. Autonomic computing refers to the self-managing principles of computer-based systems while hiding the intrinsic complexity of the system. Using the holistic techniques provided by autonomic computing, we can handle to a large extent different system requirements such as performance, fault tolerance, reliability, security, QoS, and so forth without manual intervention.

In heterogeneous and dynamic interconnected cloud environments, the system must continuously adapt itself to the current state of the system. The result must be an integrated solution capable of a wide range of autonomic management tasks including *self-configuration* (i.e., automatic configuration of components), *self-healing* (i.e., automatic discovery and correction of faults), *self-optimization* (i.e., automatic optimization of resource allocation), and *self-protecting* (i.e, automatic system security and integrity). Self-management of cloud services minimizes user interactions with the system and represents challenging research issues. There is big overlap between autonomics and other challenges and related issues we discussed in this article. That is, autonomic systems can be utilized for different aspects of interconnected cloud environments such as SLA management, provisioning, security, market, and so forth. This requires a detailed investigation of autonomic computing for each aspect.

For example, autonomic computing principles can be applied for provisioning. This is because in interconnected cloud environments, user applications might require expanding their resources by scaling out onto another cloud, and may have preference for a particular cloud or may want to combine multiple clouds. Such integration and interoperability must be done without manual intervention. In addition, in a federated cloud environment, a small or private cloud might be required to expand its capacity or computational resources by integrating or bursting into other cloud platforms on demand. Such dynamic and scalable provisioning must be done autonomously based on the workload, spikes in demands, and other extreme requirements.

Self-manageable interconnected cloud infrastructures on one hand are required to achieve a high level of flexibility and on the other hand to comply with users' requirements specified by SLAs. Matching desired user SLAs with cloud providers' service offerings is a challenge. That is, due to a large variety of services and offerings in cloud environments, matching between user requirements and services is a difficult task. Flexible and adaptive SLA attainment strategies are needed to overcome this issue. Such flexible and dynamic SLAs cannot be generated manually due to the high number of services and consumers and requires tomust be done autonomously.

Relevant progress in the field of autonomic cloud computing has been achieved by Kim and Parashar in the CometCloud project [Kim and Parashar 2011]. CometCloud is an autonomic computing engine developed for cloud and grid environments that supports highly heterogeneous infrastructures, integration of public and private clouds, and dynamic application scale-out. The service layer of CometCloud provides a range of services to support autonomics at the programming and application levels, including features such as deadline-, budget-, and workflow-based autonomic scheduling of applications on the cloud, fault tolerance, and load balance.

Other approaches apply autonomic computing principles for specific aspects of cloud computing such as market management [Breskovic et al. 2011], cloud networking [Choi et al. 2011], VM configuration [Xu et al. 2012], and workflow execution [Papuzzo and Spezzano 2011].

6. STANDARDS

Successful realization of Inter-cloud requires different standards. In particular, standards for interoperability, security, and legal issues must be taken into account when a platform for interoperability among different cloud vendors is created. Stepping toward

a commonly and widely adopted solution requires a considerable amount of work and research to overcome existing diversities. There are several barriers and problems in applying standard APIs and protocols in cloud environments [Petcu 2011]:

(1) Vendors usually prefer to lock in their customers with their facilities to avoid losing them to competitors.
(2) Cloud providers offer differentiated services and desire to have their own particular services to attract more customers.
(3) Cloud providers often do not easily agree on certain standards.
(4) It takes years to fully develop a standard and apply it globally.
(5) There are numerous standards being developed simultaneously, and agreement on which one to adopt may be difficult and sometimes impossible to attain.
(6) In cloud computing, substantially different standards are required for diverse cloud models (e.g., IaaS, PaaS, SaaS). Accordingly, one comprehensive set of standards is hard to develop.

There are many groups and initiatives that are working on cloud computing standards. We identified the main active groups and their activities and summarized them in Table I. These groups and organizations can be categorized into two main categories:

—*Standards developing organization* (SDO), when they are technically involved in developing and publishing standards for cloud computing and cloud interoperability; and
—*Industrial or scientific consortia and standards-setting organization (SSO)*, when they work toward promoting the adoption of emerging technologies, typically without the intention of developing their own standards. Consortia bring organizations, companies, academia, and governmental institutes together to cooperate toward the purpose of wider adoption and development of cloud computing technologies and they are interested in achieving a consensus to address technical problems in cloud computing. The efforts of consortia and SSOs expedite the standard development process or even in the case of being widely accepted by cloud computing stakeholders are converted to standards.

Bold entries in Table I represent SDOs. A more detailed review of the current standard protocols and interfaces facilitating Inter-cloud realization can be found in the supplementary Appendix B. Interested readers will also find an inventory of standards relevant to cloud computing[37] compiled by the National Institute of Standards and Technologies (NIST) Cloud Computing Standards Roadmap Working Group (CCSRWG). The Working Group actively updates the inventory as and when more standards are created.

7. INTER-CLOUD PROJECTS

7.1. RESERVOIR

The RESERVOIR [Rochwerger et al. 2009] project introduces modular, extensible, and open cloud architecture that supports business-driven cloud federation. It enables cloud infrastructure providers, from different administrative domains, to collaborate with each other in order to create a vast pool of resources while technological and business management decisions on each domain are made autonomously by the provider.

In the RESERVOIR model, service and infrastructure providers play different functional roles. Service providers offer service to their customers' applications based on

[37]Inventory of Standards Relevant to Cloud Computing, http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/StandardsInventory.

Table I. Standardization Activities Regarding Inter-Cloud Challenges: Provisioning (Pr), Portability (Po), Service-Level Agreement (SLA), Security (S), Monitoring (M), Economy (E), Network (N), Autonomics (A)
Bold entries are standards developing organizations.

	Pr	Po	SLA	S	M	E	N	A
DMTF[a]	✓	✓	✓	✓	✓			✓
OGF[b]	✓		✓	✓	✓			✓
CSA[c]				✓	✓			
OCM[d]		✓	✓					
NIST[e]	✓	✓		✓				
CCIF[f]	✓			✓	✓			
OCC[g]							✓	
OASIS[h]	✓		✓	✓				
GICTF[i]							✓	✓
ETSI[j]		✓					✓	✓
CWG[k]						✓		
OMG[l]			✓	✓		✓		
ODCA[m]	✓		✓	✓				
IEEE P2302[n]	✓	✓		✓	✓	✓	✓	
SNIA CSI[o]	✓	✓						
ISO JTC 1/SC 38[p]	✓	✓	✓		✓	✓	✓	
ITU-T FG[q]	✓	✓	✓	✓	✓		✓	
SIENA[r]	✓	✓		✓				✓

[a]Distributed Management Task Force (DMTF), http://www.dmtf.org/.
[b]Open Grid Forum (OGF), http://www.gridforum.org/.
[c]Cloud Security Alliance (CSA), https://cloudsecurityalliance.org/.
[d]Open Cloud Manifesto, http://www.opencloudmanifesto.org/.
[e]National Institute of Standards and Technologies (NIST), http://www.nist.gov/.
[f]Cloud Computing Interoperability Forum (CCIF), http://www.cloudforum.org/.
[g]Open Cloud Consortium (OCC), http://opencloudconsortium.org/.
[h]Organization for the Advancement of Structured Information Standards (OASIS), https://www.oasis-open.org/.
[i]InterCloud Technology Forum (GICTF), http://www.gictf.jp/index_e.html.
[j]The European Telecommunications Standards Institute (ETSI), http://www.etsi.org/.
[k]The Open Group Cloud Computing Work Group, http://www.opengroup.org/getinvolved/workgroups/cloud-computing.
[l]Object Management Group (OMG), http://www.omg.org/.
[m]Open Data Center Alliance (ODCA), http://www.opendatacenteralliance.org/.
[n]IEEE P2302 Working Group (Intercloud), http://grouper.ieee.org/groups/2302/.
[o]Storage Networking Industry Association (SNIA) Cloud Storage Initiative, http://www.snia.org/forums/csi.
[p]ISO JTC 1/SC 38, http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees/jtc1_home/jtc1_sc38_home.htm.
[q]ITU-T Focus Group on Cloud Computing (FG Cloud), http://www.itu.int/en/ITU-T/focusgroups/cloud/.
[r]Standards and Interoperability for eInfrastructure implemeNtation initiAtive (SIENA), http://www.sienainitiative.eu/.

leased resources of infrastructure providers. Infrastructure providers provide a seemingly infinite pool of virtualized computational, network, and storage resources. These virtualized resources are offered in the form of fully isolated runtime environments called virtual execution environments (VEEs). VEEs abstract away the physical characteristics of the resources and enable resource sharing.

Every RESERVOIR site includes three different abstract layers: Service Manager, Virtual Execution Environment Manager (VEEM), and Virtual Execution Environment Host (VEEH). The Service Manager, the highest level, receives a service manifest from the service provider. Service Manager handles several tasks such as deploying and provisioning VEEs, billing, accounting, and monitoring SLA compliance. VEEM, the second layer, is responsible for managing VEEs and interacting with VEEM on remote sites allowing federation of infrastructures. VEEM is also responsible for optimal placement of VEEs into VEE hosts according to constraints determined by Service

Manager. The lowest level, VEEH, supports different virtualization platforms for control and monitoring of VEEs. Moreover, transparent VEE migration within the federated cloud is supported by VEEH.

7.2. mOSAIC

mOSAIC [Petcu et al. 2011] is a multicloud solution for cloud application developers to help them to see the cloud resources as abstract building blocks in their application. It deals with the cloud issues by focusing on the application layer rather than the IaaS layer. mOSAIC enables application developers to obtain the desired application characteristics such as scalability, fault tolerance, and QoS.

One of the main goals of mOSAIC is to allow transparent and simple access to heterogeneous cloud resources and to avoid vendor lock-in. It fulfills this goal by its cloud ontology that describes services and their interfaces. Moreover, a unified cross-platform API that is platform and language independent is provided by mOSAIC.

The mOSAIC platform is targeted mainly toward cloud application developers. Therefore, mOSAIC intends to offer them an SLA-oriented resource management based on agent technologies. Inside its platform, several different agents are provided to support resource-related services such as resource discovery, negotiation, brokering, monitoring, tuning, and scalability. It also provides an event-driven approach to adapt the cloud configuration according to changes in application requirements. All these capabilities establish a framework for dynamically interconnecting and provisioning services from multiple clouds.

7.3. Contrail

The Contrail project [Carlini et al. 2012] proposes a federated and integrated cloud approach. Contrail tries to create an environment that allows cloud customers to exploit resources belonging to different cloud providers through a homogeneous secure interface regardless of the technology the providers use. In addition, it promotes adoption of a fully open-source approach toward this goal.

Contrail integration can be categorized in two parts: vertical and horizontal integration. In the vertical integration, a unified platform for accessing different resources is provided, while in the horizontal integration, the interaction between different cloud providers has been provided. Contrail works based on the broker services (*federation support*) that act as mediators between cloud users and providers. The federation support offers resources belonging to different cloud providers to users in a uniform fashion.

The federation architecture is composed of three layers, namely, *interface*, *core*, and *adapters*. The interface layer gathers requests from users as well as other Contrail components that rely on the federation functionality and facilities. The interface layer includes a Command-line interface and a web interface, from which it is possible to access REST services. The core layer contains modules for identity management, application deployment, and SLA coordination.

The identity management provides a federation-level account to each user. By using this account, the user can have access to all the resources owned by the federated cloud providers. Single Sign-On (SSO) has been provided by federation support; that is, once a user is authenticated and gains access to the federated cloud providers, the user is not prompted again to log in at each of them.

The Federation Runtime Manager (FRM) component in the core layer is responsible for application deployment. FRM provides discovery and selection to minimize economical costs and to maximize performance levels. Moreover, FRM is responsible for the application life cycle management. The Image manager and provider watcher are two

other components in the core, which are in charge of managing images and monitoring processes, respectively.

One of the main components in the core layer is the SLA Organizer. It extends the SLA management functionalities of the SLA@SOI[38] project. The SLA Organizer is a collection of three modules: SLA Coordination, SLA Negotiation, and SLA Template Repository.

The adapters layer contains the internal and external modules that enable access to infrastructural services for both the Contrail cloud and external clouds, respectively. Internal adapters provide components for network (Virtual Infrastructure Network [VIN]), storage (Global Autonomous File System [GAFS]), and computing (Virtual Execution Platform [VEP]). External adapters supply provider-specific adapters for non-Contrail providers by translating requests from the federation support into requests that are understood by the provider.

7.4. Cloudbus InterCloud

InterCloud [Buyya et al. 2010] promotes interconnected cloud computing environments that facilitate scalable provisioning of application services based on QoS constraints under variable workload, resource, and network conditions. It supports scaling of applications across multiple clouds according to required resources for cloud applications (VMs, services, storage, and database) in order to handle sudden variations in service demands.

InterCloud is composed of a set of elements that interact via a market-oriented system to enable trading of cloud resources such as computing power, storage, and execution of applications. The Inter-cloud model comprises two main elements: *Cloud Exchange* and *Cloud Coordinator*.

The *Cloud Exchange* component offers services regarding the information system directory and market making that allow providers to find each other and directly trade cloud resources. In the former case, it implements a web-service-based interface that allows providers to join and leave the federation. In the latter case, with the aim of finding available resources, providers send requests for resources to the Cloud Exchange. Providers who are interested in selling resources publish their offers to the Cloud Exchange as well. The Cloud Exchange generates a list of providers with corresponding service prices that can handle requests according to the market mechanism. In this way, buyers are able to locate potential sellers for the required resources.

The *Cloud Coordinator* component is responsible for domain-Â-specific issues related to the federation. Every provider in the federation contains this component. The Cloud Coordinator has two main parts: front-end, which is responsible for interaction with the federation, and back-end, which interacts with the associated provider. Front-end components interact with the Cloud Exchange and other coordinators. The former allows data centers to publish their offers and requests for resources, whereas the latter allows the Coordinator to acquire the current state of the provider to decide about allocation of additional resources from the federation or the amount of offering resources with other members. Hence, wherever the Coordinator detects that additional resources are required, it sends requests to the federation to discover potential seller providers. Once potential providers are discovered and the preferred one is selected, the Coordinator contacts the remote Coordinator and they start the resource exchange process. Similarly, when the Cloud Coordinator notices that local resources are underutilized, it can publish an offer for idle resources in the Cloud Exchange in order to find potential buyers.

[38]SLA@SOI, http://sla-at-soi.eu/.

Inter-cloud acts at high levels of cloud interoperability, and issues related to security, VM images, and networking are not handled by the framework. However, existing approaches or even new solutions for them can be applied in Inter-cloud without modifying its architecture.

7.5. OPTIMIS

OPTIMIS [Ferrer et al. 2012] is a toolkit that enables flexible and dynamic provisioning of cloud services targeting multicloud architectures. The focus of the toolkit is on cloud service and infrastructure optimization throughout the construction, deployment, and operation phases of the service life cycle. It provides a platform for consumers to employ cloud services with requirements regarding allocation of data and VMs such as elasticity, energy consumption, risk, cost, and trust. In terms of provisioning models for cloud computing, OPTIMIS facilitates cloud bursting, multicloud provisioning, and federation of clouds. Multicloud architectures and federated cloud environment in OPTIMIS enable transparent, interoperable, and an architecture-independent fashion of utilizing resources from multiple providers.

The toolkit consists of a set of fundamental components in order to realize different multiple cloud scenarios. The main components of the toolkit are the Service Builder, the Basic Toolkit, the Admission Controller, the Deployment Engine, the Service Optimizer, and the Cloud Optimizer. *Service Builder* allows a service programmer to access an integrated development environment. It simplifies both the development and configuration of the service using a novel programming model for service development. The *Basic Toolkit* provides functionalities common to components that are used during service deployment and execution (e.g., monitoring and security).

Providers receiving a deployment request perform an admission control to decide whether to admit the request. Using the Basic Toolkit, in order to choose the most suitable provider, the *Deployment Engine* evaluates the providers' offers to run the service. Afterward, allocation of resources for the service is performed by the *Cloud Optimizer* with help from components for management of VMs and data using functionalities in the Basic Toolkit. The *Service Optimizer* is notified once the deployment process is completed. According to the agreed-upon SLAs, the *Service Optimizer* continuously checks the service, and if it is required, it can migrate the service to another provider.

7.6. Open Cirrus

Open Cirrus [Avetisyan et al. 2010] is a federation-based cloud computing testbed sponsored by companies such as HP, Yahoo!, and Intel and supported by academic institutions in the United States, Germany, and Singapore. It is composed of data centers located in the United States, Europe, and Asia. It offers researchers access to the federated infrastructure. Each data center (site) is organized as a stack of services.

At the lowest layer, referred to as the foundation layer, a service called *Zoni* is offered. Zoni offers services at the IaaS layer and is responsible for activities such as management of physical resources and services such as resource allocation, node isolation, software provisioning, logging, and networking. The next layer, called *primary domain services,* offers services at the PaaS level. It enables users to use application frameworks such as Hadoop and to execute MPI applications without having to interact with the foundation layer. This layer also offers cloud storage (via the Hadoop File System) and management of virtual machines deployed in multiple sites via an AWS-compatible API. Finally, the *utility services* offer additional noncritical services such as monitoring, network file system, and accounting for resource utilization.

Allocation of resources belonging to multiple sites is a user-initiated task. It is facilitated via services such as a global sign-on,[39] global monitoring tools, and user directories remotely mountable, and global storage. It is worth noting that, even though allocation of resources at the federation level is a user-initiated task, the final decision on allocation is made by individual remote sites. Therefore, it is possible that a particular user will not have access to a particular site of the federation. As users are assigned to a particular site, utilization of remote resources (i.e., resources that belong to other sites rather than the one the user belongs to) incur credits being transferred from the local to the remote site for compensation for resource utilization.

7.7. Claudia

Claudia [Rodero-Merino et al. 2010] is a user-side system for cloud abstraction that aims at providing a single interface for cloud service management and auto-scaling abilities for cloud service providers (the intended users of the system). Claudia enables different services to be deployed independently via service description files (SDFs). This enables service providers to describe service dependencies, which are handled by Claudia adequately: dependencies are identified and deployed before dependent services are deployed. Claudia's SDF language is an extension of the OVF.

An important feature of Claudia is the ability to perform automated auto-scaling based on user-defined scalability rules. Such rules define the trigger for the auto-scaling feature (either service metrics such as response time or hardware metrics such as CPU utilization) and the scalability action to be carried out by the system. When the trigger conditions are detected, the action is automatically performed by the system. Scalability rules encompass both scaling resources up/down (i.e., increasing or decreasing the number of resources from a specific provider) and scaling resources in/out (i.e., consolidating or spreading resources across different providers).

Access to the underlying cloud infrastructure is performed by independent cloud infrastructure managers that must be able to interact with multiple cloud providers. In this sense, client-side libraries, described in Section 8.2, could be adopted for this purpose. Alternatively, systems such as OpenNebula [Moreno-Vozmediano et al. 2012], which are also able to interact with multiple clouds, can also be used for this purpose. Claudia is developed as part of the RESERVOIR [Rochwerger et al. 2009] project.

7.8. Intercloud by Bernstein et al.

Bernstein et al. [2009] propose a blueprint for interconnection of cloud data centers. The blueprint focuses on challenges in low levels, such as virtual machine mobility and interoperability, storage interoperability, network addressing, and addressing mobility, security (mainly identity and trust), and messaging. In this direction, this research has been targeting protocols and mechanisms for Inter-cloud. Advanced features not addressed by other initiatives but considered in the blueprint are multicasting, time synchronization, VM formats, reliable transport, events sequencing, and storage replication [Bernstein and Vij 2010c].

The establishment of Intercloud starts with elements called *Intercloud Root Instances* [Bernstein et al. 2009] that are responsible for mediating connection between different cloud providers. This avoids the necessity of each provider having to mediate the access to other providers in Intercloud, which helps in increasing the system scalability.

Besides InterCloud Root Instances, the architecture also contains *Intercloud Exchange Providers* that are responsible for aiding with the negotiation between providers for utilization of resources. Finally, a Catalog component stores information necessary

[39]Global sign-on enables users to access any federated site via a single set of credentials.

for providers to locate other members of the Intercloud and the offered services. Catalogs are locally made available by providers, and the Exchange is able to aggregate information from these multiple catalogs in order to handle complex queries from other members.

Most of the work regarding the blueprint is directed toward concepts, architectures, and standards rather than actual system developments. It also includes protocols necessary for enabling different parts of the Intercloud interactions [Bernstein et al. 2009].

7.9. Federated Cloud Management (FCM)

Federated Cloud Management (FCM) [Marosi et al. 2011] is an architecture that enables the integration of multiple IaaSs to execute applications from multiple users. It is composed of a number of elements. The core element of the architecture is the *Generic Meta Brokering Service* (GMBS) component that receives requests from multiple users and can direct them to multiple IaaS clouds for execution. The GMBS performs the matchmaking between user requests and resources from clouds to determine where the request should be scheduled.

A specific *Cloud-Broker* for each available IaaS provider interacts with the GMBS to receive the user request and to execute it in its infrastructure. IaaS brokers are also responsible for managing resources and keep QoS and resource utilization metrics updated so it is used by the GMBS during the matchmaking process. Virtual appliances that contain the necessary software to support user requests are kept in a virtual appliance repository that is part of the FCM architecture. These virtual appliances are reconstructed on each cloud provider, when required by user requests, by a *VM Handler* component that is executed on each IaaS cloud.

7.10. Sky Computing

The Sky Computing project [Keahey et al. 2009] aims at enabling aggregation of multiple virtualized sites in order to enhance availability of resources. This project addresses issues such as trust, VM portability, and connectivity of geographically spread resources. The latter is achieved via overlay networks that enable remote resources to access each other as if they were connected in a local area network. The different features from the project that enable interoperability are achieved via utilization of middleware. Demonstrations of the feasibility of the approach were performed with leverage of different existing open-source tools such as Xen, ViNE (for overlay networking), Nimbus, and Hadoop.

7.11. STRATOS

STRATOS [Pawluk et al. 2012] proposes a broker enabling allocation of cloud resources from multiple providers. The decision of the providers to be utilized for a specific allocation is defined at runtime. The proposed architecture performs selection and monitoring and is able to consider SLA for decision making. Among the use cases proposed by the authors, the goals of the broker were avoiding lock-in and minimizing deployment cost. One of the main differences of this work in relation to competing approaches is the consideration of service measurements and KPIs for the research selection via the SMI (Service Measurement Index) framework.[40]

8. TOOLS AND FRAMEWORKS

Apart from the Inter-cloud projects we discussed in the previous section, there are tools and frameworks that play a crucial role in enabling cloud interoperability. In

[40]http://www.cloudcommons.com/about-smi.

this section, we initially discuss open-source cloud platforms that facilitate cloud interoperability and portability. Then, we discuss client-side libraries and distributed programming languages that provide abstract programming facilities for clients to build their own multicloud solutions. Finally, we discuss projects and tools providing interconnected cloud platforms for scientific applications and research purposes.

8.1. Open-Source Cloud Management Platforms

Open-source cloud platforms (OCPs) are important for cloud interoperability not only because of the benefits of being open source but also because they are able to mitigate the risk of vendor lock-in by providing interoperable cloud environments. As OCPs mostly support standard interfaces such as Open Grid Forum (OGF)[41] and Open Cloud Computing Interface (OCCI),[42] applications deployed on OCPs can be easily moved from one IaaS provider to another one implementing these APIs, without having to be modified. Considering the fact that OCPs facilitate cloud interoperability and portability, we study them among Inter-cloud solutions. We briefly explore four main OCPs and their architectures: OpenNebula[43] [Moreno-Vozmediano et al. 2012], OpenStack,[44] CloudStack,[45] and Eucalyptus[46] [Nurmi et al. 2009].

8.1.1. OpenNebula. OpenNebula [Moreno-Vozmediano et al. 2012] is an open-source platform for management of virtualized data centers to enable IaaS clouds. OpenNebula's main application is as a tool to manage a virtualized infrastructure in private, public, or hybrid clouds. OpenNebula is not only designed for cloud interoperability but also for comprehensive management of virtualized data centers. Interoperability and portability, leveraging and implementing standards, adaptability to manage any hardware and software, and scalability of large-scale infrastructures are among the main principles considered in the design of OpenNebula.

The OpenNebula architecture consists of three layers:

—*Tools*: Contains OpenNebula's command line interface (CLI), the scheduler and interfaces for communication with the *Core* layer. The scheduler is an independent entity that uses an XML-RPC interface to invoke actions on virtual machines. The *Haizea* lease manager [Sotomayor et al. 2009] can also be used as a scheduling module in OpenNebula. Haizea allows OpenNebula to lease resources as VMs, with a variety of lease terms supported, including advance reservation of resources and best-effort requests.
—*Core*: Consists of components to control and monitor virtual machines, virtual networks, storage, and hosts. The core layer performs its actions by invoking a suitable driver.
—*Drivers*: Contains drivers for virtualization, storage, monitoring, and authorization and connects to the underlying physical infrastructure.

OpenNebula provides a higher level of interoperability for private clouds by supporting the most common hypervisors, such as KVM, VMware, and Xen and its libvirt plug-in. In the public cloud, interoperability is provided by supporting the most common cloud interfaces, such as VMware vCloud, OCCI, and open libraries, such as libcloud[47]

[41]Open Grid Forum, http://www.ogf.org/.
[42]Open Cloud Computing Interface, http://occi-wg.org/.
[43]OpenNebula, http://opennebula.org/.
[44]OpenStack, http://www.openstack.org/.
[45]CloudStack, http://Cloudstack.apache.org/.
[46]Eucalyptus, http://www.eucalyptus.com/.
[47]libcloud, http://libcloud.apache.org/.

and δ-Cloud.[48] Interoperability and portability in the hybrid cloud are also enabled by supporting the combination of local private infrastructure with Amazon EC2 and ElasticHosts, Rackspace, GoGrid, or Terremark through the RedHat's δ-Cloud APIs.

8.1.2. OpenStack. OpenStack is an open-source IaaS Cloud management platform, released under the terms of the Apache License, designed to control large pools of compute, storage, and networking resources in a data center. OpenStack is not specifically designed for either interoperability or portability; nevertheless, it is very close to being a standard in the cloud ecosystem.

OpenStack provides a web interface (dashboard) and Amazon EC2-compatible APIs that can be used by users to provision resources. Similar to OpenNebula, OpenStack also supports OCCI. Since OpenStack APIs are compatible with Amazon EC2 and Amazon S3, applications designed for Amazon Web Services can be used with OpenStack with minimal modification effort.

To maximize interoperability and deployment flexibility and to reduce risks of lock-in associated with proprietary platforms, OpenStack is designed as a series of loosely coupled components that are easy to integrate with a variety of solutions and hardware platforms. The main components are:

—*OpenStack Compute (Nova)*: It manages the life cycle of VM instances from scheduling and resource provisioning to live migration and security rules.
—*OpenStack Storage (Swift)*: Swift is a scalable redundant storage system responsible for enabling data replication and ensuring integrity.
—*Block Storage (Cinder)*: The block storage system allows users to create block-level storage devices that can be attached to or detached from VM instances.
—*OpenStack Networking (Neutron)*: Neutron is a system for managing networks and IP addresses. The system allows users to create their own networks and assign IP addresses to VM instances.
—*OpenStack Dashboard (Horizon)*: It provides users and administrators with management capabilities via a web interface. Management actions enabled by this component include VM image management, VM instance life cycle management, and storage management.
—*OpenStack Identity (Keystone)*: Keystone is an account management service that acts as an authentication and access control system.
—*OpenStack Image (Glance)*: It supplies a range of VM image management capabilities from discovery and registration to delivery services for disk and server images.

8.1.3. CloudStack. CloudStack is an open-source IaaS platform originally developed by Cloud.com and later purchased by Citrix. The source code was later denoted by Citrix to the Apache Software Foundation and was released under Apache license in April 2012. CloudStack was designed to support the deployment and management of large networks of virtual machines as an IaaS cloud computing platform. CloudStack supports both the Amazon EC2 and vCloud APIs, in addition to its own API. CloudStack has a hierarchical structure that enables management of large networks of virtual machines. The CloudStack structure includes the following components:

—*Hosts*: Physical machines onto which virtual machines are provisioned
—*Cluster*: A group of physical machines that utilize the same type of hypervisor
—*Pod*: A rack in a data center containing one or more clusters and a switch shared by all clusters in that pod
—*Zone*: Collection of pods and secondary storage shared by all pods in the zone

[48] δ-Cloud, http://deltacloud.apache.org/.

Table II. Comparison of Open-Source Cloud Management Platforms

	OpenNebula	OpenStack	CloudStack	Eucalyptus
License	Apache v2.0	Apache v2.0	Apache v2.0	GPL v3
API Compatibility	AWS, OCCI	AWS, OCCI	AWS	AWS
Hypervisor Support	Xen, KVM, VMware	Xen, KVM, VMware	KVM, Xen, VMware, Oracle VM	Xen, KVM, VMware
Architecture	Loosely Coupled	Component Based	Tightly Coupled	Tightly Coupled
Hybrid Cloud	Yes	No	Yes	Yes

—*Primary Storage*: Shared storage across a cluster used to host the guest virtual machines
—*Secondary Storage*: Shared storage in a single zone used to store virtual machine templates, ISO images, and snapshots

8.1.4. Eucalyptus. Eucalyptus[49] [Nurmi et al. 2009] is an open-source platform, compatible with Amazon Web Services' (AWS) APIs, for building private and hybrid cloud computing environments. Eucalyptus provides a platform for managing pools of compute, storage, and network resources that can be dynamically provisioned based on the application requirements. In order to maintain compatibility, Eucalyptus Systems announced a formal agreement with AWS in March 2012. Eucalyptus enables workload migration and deployment of hybrid cloud environments.

The Eucalyptus platform is composed of the following high-level components:

—*Cloud Controller*: Manages the underlying virtualized compute, network, and storage resources and provides an Amazon EC2-compatible web interface and APIs. Moreover, it handles authentication and accounting.
—*Cluster Controller*: Communicates with the Storage Controller and Node Controller and manages virtual machines' execution and SLAs
—*Storage Controller*: Provides block storage equivalent to AWS Elastic Block Storage (EBS) that can be dynamically attached to VMs
—*Node Controller*: Hosts the virtual machine instances and manages the virtual network endpoints during the VM life cycle using the functionality provided by the hypervisor
—*Walrus*: Provides persistent storage service compatible with Amazon S3
—*VMware Broker*: Is an optional component that offers an AWS-compatible interface for VMware and runs on the Cluster Controller

At present, Amazon[50] is one of the dominant players in the IaaS cloud market and its service APIs are becoming de facto standards for operation on cloud resources; accordingly, other vendors offer AWS-compatible APIs for their services. Not surprisingly, all open-source platforms we discussed in this section support AWS-compatible APIs, what makes multicloud deployment scenarios further attainable. A comparison of the discussed cloud platforms is presented in Table II.

[49]Eucalyptus, an acronym for "Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems."
[50]Amazon Web Services, http://aws.amazon.com/.

8.2. Client-Side Libraries

Different projects are focusing on the development of libraries that enable clients to build their own Inter-cloud solutions at different abstraction levels. These libraries offer high-level APIs that exempt developers to decide about interaction with specific cloud providers at application development time. It brings extra flexibility to public cloud utilization, because it facilitates both consideration of new cloud providers for application deployment and changes in the types of instances to be used with the application. It also helps in avoiding vendor lock-in by forcing developers to utilize library-defined services rather than utilizing specialized features that are specific to a given provider.

The jclouds library[51] provides programmers with abstractions representing typical compute elements (called *ComputeService*) and key-value data stores (called *BlobStore*). As the APIs defining these two elements are vendor agnostic, deployment of cloud-ready applications is made independent from the underlying IaaS provider. jclouds supports development of applications in both Java and Clojure and offers extra features such as ability for unit testing, load balancing, location-aware abstractions, and mappers that eliminate the need for programmers to interact with web services-based APIs. Providers supported by jclouds include Amazon AWS, GoGrid,[52] Windows Azure,[53] CloudSigma,[54] and Ninefold,[55] among others.

LibCloud[56] is a Python-based library that, like jclouds, offers abstractions for compute elements, storage, and load balancing. LibCloud also offers an API for interaction with IaaS-provided DNS. This project supports over 26 different cloud providers.

The δ-Cloud[57] library is a Ruby-based library for interaction with public cloud providers. Its features are similar to those offered by LibCloud and jclouds and it supports over 15 different cloud providers.

The main difference between the aforementioned projects are the programming language chosen for the library development and the maturity level of each project, whether in terms of cloud providers supported or in terms of cloud services that can be abstracted via the library's API.

8.3. Distributed Programming Languages

Inter-cloud interoperability can be achieved not only from client-side libraries but also with the use of distributed programming languages such as Nomadic Pict [Sewell et al. 2010] and SALSA [Varela and Agha 2001]. These programming languages allow the algorithms or applications to be executed independent of their locations and transparent to migrations. For instance, SALSA, an actor-based language for Internet computing, provides facilities for applications composed of SALSA actors to be easily reconfigured at runtime by using actor migration. This only requires cloud applications to contain SALSA migratable components and does not impose any further restrictions on workloads. By using this kind of application-level migration, it is possible to achieve load balancing, elasticity, and scalability with finer granularity (in the scale of application entities instead of VM-level coarse granularity). With the same objectives, Imai et al. [2012] proposed a middleware framework to support autonomous workload elasticity

[51]jclouds, http://code.google.com/p/jclouds/.
[52]GoGrid, http://www.gogrid.com/.
[53]Windows Azure, http://www.windowsazure.com/.
[54]http://www.cloudsigma.com/.
[55]Ninefold, http://ninefold.com/.
[56]LibCloud, http://libcloud.apache.org/.
[57]δ-Cloud, http://deltacloud.apache.org/.

and scalability based on application-level migration to cloud computing, also targeting hybrid clouds.

8.4. Interoperable Cloud Infrastructure Projects Supporting e-Science

While all previously discussed projects try to build interconnected cloud environments that are independent of the application type, there are projects that focus on forming distributed multiple cloud infrastructures for scientific applications and for research purposes. In this section, we briefly cover these projects and we categorize them as interoperable cloud infrastructures supporting e-science.

The Open Science Data Cloud (OSDC)[58] is a distributed cloud-based infrastructure that provides platforms for users to compute over large scientific datasets. The OSDC operates one storage cloud named *Root* and two main compute clouds named *Adler* and *Sullivan*. Adler is an Eucalyptus-based cloud and Sullivan is an OpenStack-based cloud. Root is a repository of various public scientific datasets that can be accessed from the OSDC clouds. The European Grid Initiative is also looking into how to make a grid of academic private clouds and virtualized resources (federate clouds) while focusing on the requirements of the scientific community.[59] Their goal is to provide an e-infrastructure for research based on the federated operations services.

Aneka [Calheiros et al. 2012b] is a cloud platform that supports creation and deployment of scientific applications across multiple IaaS clouds including a private cloud (one created using Aneka) and public clouds such as Amazon EC2 and Microsoft Azure. Parashar et al. [2013] comprehensively explore benefits, limitations, and research challenges of executing high-performance computing (HPC) scientific workloads across a federation of multiple resources including clouds. Vázquez et al. [2009] present an architecture to build a grid infrastructure with a unified point of access hiring compute resources from public cloud providers with potentially different interfaces to execute HPC applications. They use available technologies rather than developing new standards for future use. The proposed architecture is able to dynamically expand and use resources from cloud providers to react to peak demands. In contrast, Bittencourt et al. [2010] propose an infrastructure able to manage the execution of workflows on a hybrid system composed of both grid and cloud technologies. Vöckler et al. [2011] leverage Pegasus and Condor to execute an astronomy workflow on virtual machine resources provisioned from multiple clouds. Similarly, Gorton et al. present [2010] a federated cloud-based architecture for modeling, simulation, and experimentation of bioinformatics applications.

9. SUMMARY, DISCUSSION, AND FUTURE DIRECTIONS

As the adoption of cloud as the main technology for provisioning of infrastructure, platform, and service for users grows continually, the need to aggregate services and functionalities from different providers arises. This aggregation can happen in any of the delivery models (IaaS, PaaS, or SaaS) and can be enabled by different approaches and technologies.

In this article, we surveyed the relevant aspects that motivate cloud interoperability and the mechanisms and technologies enabling it. We discussed why aspects such as scalability, resource limitations, vendor lock-in, availability, disaster recovery, geographic distribution, latency reduction, regulation and legislation, cost efficiency, and energy savings play a vital role in pushing technologies for cloud interoperability.

[58]Open Science Data Cloud (OSDC), https://www.opensciencedatacloud.org/.
[59]European Grid Community, http://www.egi.eu/infrastructure/cloud/.

Besides specific motivation, interoperability can be achieved via one or more standard interfaces, brokers, or middlewares. Any of these approaches can be applied to the surveyed interoperability scenarios, which are:

—*Federation*, when interoperation is enabled by direct agreements between cloud providers and is transparent to end-users;
—*Hybrid clouds*, where local resources of an organization that owns a private cloud is complemented with public cloud resources to meet spikes in resource demand;
—*Multicloud*, when the end-user coordinates access and utilization of different cloud providers to meet his or her requirements; and
—*Aggregated service by broker*, when a third-party (the broker) coordinates the access and utilization of multiple cloud resources on behalf of a user.

We also discussed standardization initiatives that are under development in the area of cloud interoperability and presented a comprehensive survey of research projects in this direction. As the summary Tables III and IV in online Appendix A indicate, even though there is a significant number of works in development, a few works address all the motivation scenarios and challenges we discussed. Therefore, it is possible that comprehensive and holistic approaches to cloud interoperability will be a result of the combination of one or more of the ongoing initiatives, either directly or via an extra abstraction layer hiding the complexities from end-users.

Moreover, from the summary of the projects, it is possible to notice that there are currently only a few cloud federation projects and they are mostly brokering technologies for multicloud and aggregated service scenarios. This is because so far, cloud services have been designed without considering cloud interoperability issues. We will see more of federated and hybrid cloud environments in the future, when more cloud providers will emerge with standard interfaces for their services.

Our summary also identified that, apart from scalability, avoidance of vendor lock-in is the most common motivation for Inter-cloud projects (Table III). This is because cloud computing users are vulnerable to rises in prices, decreases in availability, and even the cloud provider's bankruptcy and consequent loss of access to data stored on the provider. As a consequence, most current Inter-cloud projects are motivated by interoperability and avoidance of vendor lock-in. However, lock-in might be attractive to cloud providers as it enables them to retain their customers with little effort in having competitive products. Researchers and professionals who are working in the interconnected cloud area must take into account that, although avoidance of vendor lock-in is the great motivation for customers, it does not provide enough incentive for providers to boost up clouds' integration. This is why a large group of professionals believe that clouds' integration must be enabled on a separated layer detached from both vendors and providers.

After the analysis of ongoing projects, we analyzed the state of the art and the trends in the area of integrated clouds, where we identified that legal issues and meeting regulations are major concerns that are not well studied by the current projects. Therefore, appropriate application brokering that honors legal issues in terms of SLA is necessary.

Apart from interfaces, we also concluded that issues regarding economic aspects of cloud interoperability have received little attention, even though interoperation cannot be achieved without the resolution of these economic aspects. Once cloud vendors are convinced that adoption of cloud interoperability awards them financial and economical benefits, the goal of ubiquitously interconnected clouds is more likely to be achieved. This requires addressing issues regarding billing and accounting, novel methods of pricing suitable for interconnected cloud environments, and finally formation of Inter-cloud marketplaces.

To summarize, our broad and deep analysis of challenges and issues regarding cloud interoperability shows a limited trace of agreement and completeness among existing projects. The current study is intended to pave the way for further research and development activities by identifying weaknesses and deriving guidelines toward the holistic approach for interconnected clouds.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

ACKNOWLEDGMENTS

The authors would like to thank Amir Vahid Dastjerdi, Mohsen Amini Salehi, Deepak Poola, and Nikolay Grozev for their constructive comments and suggestions on improving this survey. They also wish to acknowledge the comments and contributions provided by three anonymous reviewers that greatly strengthened the manuscript.

REFERENCES

Jamal Abawajy. 2009. Determining service trustworthiness in intercloud computing environments. In *Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN'09)*. 784–788.

Roberto Alfieri, Roberto Cecchini, Vincenzo Ciaschini, Luca dell'Agnello, Akos Frohner, Alberto Gianoli, Karoly Lorentey, and Fabio Spataro. 2004. VOMS, an authorization system for virtual organizations. In *Grid Computing*, Francisco Fernández Rivera, Marian Bubak, Andrés Gómez Tato, and Ramón Doallo (Eds.). Lecture Notes in Computer Science, Vol. 2970. Springer, Berlin, 33–40.

Alba Amato, Loredana Liccardo, Massimiliano Rak, and Salvatore Venticinque. 2012. SLA negotiation and brokering for sky computing. In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science (CLOSER'12)*. SciTePress, Porto, Portugal, 611–620.

Sergio Andreozzi, Natascia De Bortoli, Sergio Fantinel, Antonia Ghiselli, Gian Luca Rubini, Gennaro Tortone, and Maria Cristina Vistoli. 2005. GridICE: A monitoring service for grid systems. *Future Generation Computer Systems* 21, 4 (2005), 559–571.

Alain Andrieux, Karl Czajkowski, Asit Dan, Kate Keahey, Heiko Ludwig, Toshiyuki Nakata, Jim Pruyne, John Rofrano, Steve Tuecke, and Ming Xu. 2004. *Web Services Agreement Specification (WS-Agreement)*. Technical Report.

Tomonori Aoyama and Hiroshi Sakai. 2011. Inter-cloud computing. *Business & Information Systems Engineering* 3, 3 (2011), 173–177.

Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. 2010. A view of cloud computing. *Commun. ACM* 53, 4 (2010), 50–58.

Arutyun I. Avetisyan, Roy Campbel, Indranil Gupta, Michael T. Heath, Steven Y. Ko, Gregory R. Ganger, Michael A. Kozuch, David O'Hallaron, Marcel Kunze, Thomas T. Kwan, Kevin Lai, Martha Lyons, Dejan S. Milojicic, Hing Yan Lee, Yeng Chai Soh, Ng Kwang Ming, Jing-Yuan Luke, and Han Namgoong. 2010. Open Cirrus: A global cloud computing testbed. *Computer* 43, 4 (Apr. 2010), 35–43.

Dominic Battré, Frances M. T. Brazier, Kassidy P. Clark, Michael Oey, Alexander Papaspyrou, Oliver Wäldrich, Philipp Wieder, and Wolfgang Ziegler. 2010. A proposal for WS-agreement negotiation. In *Proceedings of the 11th IEEE/ACM International Conference on Grid Computing (GRID'10)*. 233–241.

David Bermbach, Markus Klems, Stefan Tai, and Michael Menzel. 2011. MetaStorage: A federated cloud storage system to manage consistency-latency tradeoffs. In *Proceedings of IEEE International Conference on Cloud Computing (CLOUD'11)*. Washington, DC, 452–459.

David Bernstein, Erik Ludvigson, Krishna Sankar, Steve Diamond, and Monique Morrow. 2009. Blueprint for the InterCloud—protocols and formats for cloud computing interoperability. In *Proceedings of the 4th International Conference on Internet and Web Applications and Services*. 328–336.

David Bernstein and Deepak Vij. 2010a. Intercloud directory and exchange protocol detail using XMPP and RDF. In *Proceedings of the 6th World Congress on Services (SERVICES'10)*. Miami, FL, 431–438.

David Bernstein and Deepak Vij. 2010b. Intercloud security considerations. In *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom'10)*. Indianapolis, IN, 537–544.

David Bernstein and Deepak Vij. 2010c. Simple storage replication protocol (SSRP) for intercloud. In *Proceedings of the 2nd International Conference on Emerging Network Intelligence (EMERGING'10)*. 30–37.

David Bernstein, Deepak Vij, and Stephen Diamond. 2011. An intercloud cloud computing economy - technology, governance, and market blueprints. In *Proceedings of 2011 Annual SRII Global Conference (SRII)*. IEEE, San Jose, CA, 293–299.

Luiz F. Bittencourt, Carlos R. Senna, and Edmundo R. M. Madeira. 2010. Enabling execution of service workflows in grid/cloud hybrid systems. In *IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksps)*. 343–349.

Lianne Bodenstaff, Andreas Wombacher, Manfred Reichert, and Micheal C. Jaeger. 2008. Monitoring dependencies for SLAs: The MoDe4SLA approach. In *Proceedings of IEEE International Conference on Services Computing (SCC'08)*, Vol. 1. Miami, FL, 21–29.

Janine Anthony Bowen. 2010. *Cloud computing: Principles and paradigms*. Vol. 87. Wiley, Chapter Legal Issues in Cloud Computing, 593–613.

Ivan Breskovic, Michael Maurer, Vincent C. Emeakaroha, Ivona Brandic, and Jörn Altmann. 2011. Towards autonomic market management in cloud computing infrastructures. In *Proceedings of the International Conference on Cloud Computing and Services Science (CLOSER'11)*. 24–34.

Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N. Calheiros. 2010. InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'10)*, Vol. 6081. Busan, South Korea, 13–31.

Rodrigo N. Calheiros, Adel Nadjaran Toosi, Christian Vecchiola, and Rajkumar Buyya. 2012a. A coordinator for scaling elastic applications across multiple clouds. *Future Generation Computer Systems* 28, 8 (2012), 1350–1362.

Rodrigo N. Calheiros, Christian Vecchiola, Dileban Karunamoorthy, and Rajkumar Buyya. 2012b. The aneka platform and QoS-driven resource provisioning for elastic applications on hybrid clouds. *Future Generation Computer Systems* 28, 6 (2012), 861–870.

Emanuele Carlini, Massimo Coppola, Patrizio Dazzi, Laura Ricci, and Giacomo Righetti. 2012. Cloud federations in contrail. In *Euro-Par 2011: Parallel Processing Workshops*. Lecture Notes in Computer Science, Vol. 7155. Springer, Berlin, 159–168.

Antonio Celesti, Francesco Tusa, Massimo Villari, and Antonio Puliafito. 2010a. How to enhance cloud architectures to enable cross-federation. In *Proceedings of the 3rd International Conference on Cloud Computing (Cloud'10)*. Miami, FL, 337–345.

Antonio Celesti, Francesco Tusa, Massimo Villari, and Antonio Puliafito. 2010b. Improving virtual machine migration in federated cloud environments. In *Proceedings of the 2nd International Conference on Evolving Internet (INTERNET'10)*. 61–67.

Antonio Celesti, Francesco Tusa, Massimo Villari, and Antonio Puliafito. 2010c. Security and cloud computing: InterCloud identity management infrastructure. In *Proceedings of the 19th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'10)*. 263–265.

Antonio Celesti, Massimo Villari, and Antonio Puliafito. 2010d. A naming system applied to a RESERVOIR cloud. In *Proceedings of the 6th International Conference on Information Assurance and Security (IAS'10)*. Atlanta, GA, 247–252.

Anirban Chakrabarti, Anish Damodaran, and Shubhashis Sengupta. 2008. Grid computing security: A taxonomy. *IEEE Security & Privacy* 6, 1 (2008), 44–51.

David Chen and Guy Doumeingts. 2003. European initiatives to develop interoperability of enterprise applications - basic concepts, framework and roadmap. *Annual Reviews in Control* 27, 2 (2003), 153–162.

Taesang Choi, Nodir Kodirov, Tae-Ho Lee, Doyeon Kim, and Jaegi Lee. 2011. Autonomic management framework for cloud-based virtual networks. In *Proceedings of the 13th Asia-Pacific Network Operations and Management Symposium (APNOMS'11)*. 1–7.

Stuart Clayman, Alex Galis, Clovis Chapman, Giovanni Toffetti, Luis Rodero-Merino, Luis M. Vaquero, and Kenneth Naginand Benny Rochwerger. 2010. Monitoring service clouds in the future internet. In *Towards the Future Internet - Emerging Trends from European Research*. Amsterdam, Netherlands, 1–12.

Antonio Cuomo, Giuseppe Modica, Salvatore Distefano, Antonio Puliafito, Massimiliano Rak, Orazio Tomarchio, Salvatore Venticinque, and Umberto Villano. 2013. An SLA-based broker for cloud infrastructures. *Journal of Grid Computing* 11, 1 (2013), 1–25.

Amir Vahid Dastjerdi, Sayed Gholam Hassan Tabatabaei, and Rajkumar Buyya. 2012. A dependency-aware ontology-based approach for deploying service level agreement monitoring services in Cloud. *Software: Practice and Experience* 42, 4 (2012), 501–508.

Scott Dowell, Albert Barreto, James Bret Michael, and Man-Tak Shing. 2011. Cloud to cloud interoperability. In *Proceedings of the 6th International Conference on System of Systems Engineering (SoSE'11)*. 258–263.

Erik Elmroth and Lars Larsson. 2009. Interfaces for placement, migration, and monitoring of virtual machines in federated clouds. In *Proceedings of the 8th International Conference on Grid and Cooperative Computing (GCC'09)*. Lanzhou, China, 253–260.

Erik Elmroth, Fermn Galan Marquez, Daniel Henriksson, and David P. Ferrera. 2009. Accounting and billing for federated cloud infrastructures. In *Proceedings of the 8th International Conference on Grid and Cooperative Computing (GCC'09)*. Lanzhou, China, 268–275.

Ana Juan Ferrer, Francisco Hernández, Johan Tordsson, Erik Elmroth, Ahmed Ali-Eldin, Csilla Zsigri, Raül Sirvent, Jordi Guitart, Rosa M. Badia, Karim Djemame, Wolfgang Ziegler, Theo Dimitrakos, Srijith K. Nair, George Kousiouris, Kleopatra Konstanteli, Theodora Varvarigou, Benoit Hudzia, Alexander Kipp, Stefan Wesner, Marcelo Corrales, Nikolaus Forgó, Tabassum Sharif, and Craig Sheridan. 2012. OPTIMIS: A holistic approach to cloud service provisioning. *Future Generation Computer Systems* 28, 1 (2012), 66–77.

Brian W. Fitzpatrick and J. J. Lueck. 2010. The case against data lock-in. *Queue* 8, 10 (2010), 20:20–20:26.

Íngo Goiri, Jordi Guitart, and Jordi Torres. 2011. Economic model of a cloud provider operating in a federated Cloud. *Information Systems Frontiers* 14, 4 (2011), 827–843.

Eduardo R. Gomes, Quoc Bao Vo, and Ryszard Kowalczyk. 2012. Pure exchange markets for resource sharing in federated Clouds. *Concurrency and Computation: Practice and Experience* 23, 9 (2012), 977–991.

Ian Gorton, Yan Liu, and Jian Yin. 2010. Exploring architecture options for a federated, cloud-based system biology knowledgebase. In *IEEE 2nd International Conference on Cloud Computing Technology and Science (CloudCom'10)*. 218–225.

Andrzej Goscinski and Michael Brock. 2010. Toward dynamic and attribute based publication, discovery and selection for cloud computing. *Future Generation Computer Systems* 26, 7 (2010), 947—970.

Nikolay Grozev and Rajkumar Buyya. 2012. Inter-cloud architectures and application brokering: Taxonomy and survey. *Software: Practice and Experience* (2012), n/a–n/a. DOI:http://dx.doi.org/10.1002/spe.2168

Seung-Min Han, Mohammad Mehedi Hassan, Chang-Woo Yoon, and Eui-Nam Huh. 2009. Efficient service recommendation system for cloud computing market. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human (ICIS'09)*. 839–845.

Piyush Harsh, Yvon Jegou, Roberto Cascella, and Christine Morin. 2011. Contrail virtual execution platform challenges in being part of a cloud federation. In *Towards a Service-Based Internet*, Witold Abramowicz, Ignacio Llorente, Mike Surridge, Andrea Zisman, and Julien Vayssire (Eds.). Lecture Notes in Computer Science, Vol. 6994. Springer, Berlin, 50–61.

Zach Hill and Marty Humphrey. 2010. CSAL: A cloud storage abstraction layer to enable portable cloud applications. In *Proceedings of 2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom'10)*. Indianapolis, IN, 504–511.

Luokai Hu, Shi Ying, Xiangyang Jia Kai, and Zhao. 2009. Towards an approach of semantic access control for cloud computing. In *Cloud Computing*, MartinGilje Jaatun, Gansen Zhao, and Chunming Rong (Eds.). Lecture Notes in Computer Science, Vol. 5931. Springer, Berlin, 145–156.

Shigeru Imai, Thomas Chestna, and Carlos A. Varela. 2012. Elastic scalable cloud computing using application-level migration. In *Proceedings of the 5th IEEE/ACM International Conference on Utility and Cloud Computing (UCC'12)*. IEEE/ACM, 91–98.

Katarzyna Keahey, Maurício Tsugawa, Andréa Matsunaga, and José A. B. Fortes. 2009. Sky computing. *IEEE Internet Computing* 13, 5 (2009), 43–51.

Gabor Kecskemeti, Michael Maurer, Ivona Brandic, Attila Kertesz, Zsolt Nemeth, and Schahram Dustdar. 2012. Facilitating self-adaptable inter-cloud management. In *Proceedings of the 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP'12)*. Garching, Germany, 575–582.

Attila Kertesz and Szilvia Varadi. 2014. Legal aspects of data protection in cloud federations. In *Security, Privacy and Trust in Cloud Systems*, Surya Nepal and Mukaddim Pathan (Eds.). Springer, Berlin, 433–455.

Khaled M. Khan and Qutaibah Malluhi. 2010. Establishing trust in cloud computing. *IT Professional* 12, 5 (2010), 20–27.

Hyunjoo Kim and Manish Parashar. 2011. CometCloud: An autonomic cloud engine. In *Cloud Computing: Principles and Paradigms*, Rajkumar Buyya, James Broberg, and Andrzej Goscinski (Eds.). Wiley, 275–298.

Tobias Kurze, Markus Klemsy, David Bermbachy, Alexander Lenkz, Stefan Taiy, and Marcel Kunze. 2011. Cloud federation. In *Proceedings of the 2nd International Conference on Cloud Computing, GRIDs, and Virtualization*. 32–38.

Kien Le, Richardo Bianchini, Jingru Zhang, Yogesh Jaluria, Jiandong Meng, and Thu D. Nguyen. 2011. Reducing electricity cost through virtual machine placement in high performance computing clouds. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. Seattle, WA, 22:1–22:12.

Maik Lindner, Fermin Galan, Clovis Chapman, Stuart Calyman, Daneil Henriksson, and Erik Elmroth. 2010. The cloud supply chain: A framework for information, monitoring, accounting and billing. In *Proceedings of the 2nd International ICST Conference on Cloud Computing (CloudComp'10)*. Springer Verlag, Barcelona, Spain.

Marc X. Makkes, Canh Ngo, Yuri Demchenko, Rudolf Stijkers, Robert Meijer, and Cees de Laat. 2013. Defining intercloud federation framework for multi-provider cloud services integration. In *Proceedings of the 4th International Conference on Cloud Computing, GRIDs, and Virtualization*. Valencia, Spain, 185–190.

Attila Csaba Marosi, Gabor Kecskemeti, Attila Kertesz, and Peter Kacsuk. 2011. FCM: An architecture for integrating IaaS cloud systems. In *Proceedings of the 2nd International Conference on Cloud Computing, GRIDs, and Virtualization*. IARIA, Rome, Italy, 7–12.

Matthew L. Massie, Brent N. Chun, and David E. Culler. 2004. The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Comput.* 30, 7 (2004), 817–840.

Michael Menzel and Rajiv Ranjan. 2012. CloudGenius: Decision support for web server cloud migration. In *Proceedings of the 21st International Conference on World Wide Web (WWW'12)*. 979–988.

Marian Mihailescu and Yong Teo. 2010a. A distributed market framework for large-scale resource sharing. In *Euro-Par 2010—Parallel Processing*, Pasqua DAmbra, Mario Guarracino, and Domenico Talia (Eds.). Lecture Notes in Computer Science, Vol. 6271. Springer, Berlin, 418–430.

Marian Mihailescu and Yong Teo. 2010b. Strategy-proof dynamic resource pricing of multiple resource types on federated clouds. In *Algorithms and Architectures for Parallel Processing*. Vol. 6081. Springer, Berlin, 337–350.

Marian Mihailescu and Yong Meng Teo. 2010c. Dynamic resource pricing on federated clouds. In *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid'10)*. Melbourne, Australia, 513–517.

Marian Mihailescu and Yong Meng Teo. 2010d. On economic and computational-efficient resource pricing in large distributed systems. In *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid'10)*. 838–843.

Rafael Moreno-Vozmediano, Rubén S. Montero, and Ignacio M. Llorente. 2012. IaaS cloud architecture: From virtualized datacenters to federated cloud infrastructures. *Computer* 45, 12 (2012), 65–72.

Francesco Moscato, Rocco Aversa, Beniamino Di Martino, Teodor-Florin Fortis, and Victor Munteanu. 2011. An analysis of mOSAIC ontology for cloud resources annotation. In *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS'11)*. 973–980.

Francesco Moscato, Rocco Aversa, Beniamino Di Martino, Massimiliano Rak, Salvatore Venticinque, and Dana Petcu. 2010. An ontology for the cloud in mOSAIC. In *Cloud Computing*. CRC Press, Chapter 20, 467–485.

Kenneth Nagin, David Hadas, Zvi Dubitzky, Alex Glikson, Irit Loy, Benny Rochwerger, and Liran Schour. 2011. Inter-cloud mobility of virtual machines. In *Proceedings of the 4th Annual International Conference on Systems and Storage (SYSTOR'11)*. Haifa, Israel, 3:1–3:12.

Mark Needleman. 2004. The shibboleth authentication/authorization system. *Serials Review* 30, 3 (2004), 252–253.

David Núñez, Isaac Agudo, Prokopios Drogkaris, and Stefanos Gritzalis. 2011. Identity management challenges for intercloud applications. In *Secure and Trust Computing, Data Management, and Applications*, Changhoon Lee, Jean-Marc Seigneur, James J. Park, and Roland R. Wagner (Eds.). Communications in Computer and Information Science, Vol. 187. Springer, Berlin, 198–204.

Daniel Nurmi, Richard Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov. 2009. The eucalyptus open-source cloud-computing system. In *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid'09)*. 124–131.

Giuseppe Papuzzo and Giandomenico Spezzano. 2011. Autonomic management of workflows on hybrid grid-cloud infrastructure. In *Proceedings of the 7th International Conference on Network and Services Management (CNSM'11)*. IFIP, Paris, France, 230–233.

Manish Parashar, Moustafa AbdelBaky, Ivan Rodero, and Aditya Devarakonda. 2013. Cloud paradigms and practices for computational and data-enabled science and engineering. *Computing in Science & Engineering* 15, 4 (2013), 10–18.

Przemyslaw Pawluk, Bradley Simmons, Michael Smit, Marin Litoiu, and Serge Mankovski. 2012. Introducing STRATOS: A cloud broker service. In *Proceedings of the 5th IEEE International Conference on Cloud Computing (CLOUD'12)*. 891–898.

Juan M. Marín Pérez, Jorge Bernal Bernabé, Jose M. Alcaraz Calero, Felix J. Garcia Clemente, Gregorio Martínez Pérez, and Antonio F. Gómez Skarmeta. 2011. Semantic-based authorization architecture for Grid. *Future Generation Computer Systems* 27, 1 (2011), 40–55.

Dana Petcu. 2011. Portability and interoperability between clouds: Challenges and case study. In *Towards a Service-Based Internet*. Vol. 6994. Springer, Berlin, 62–74.

Dana Petcu, Ciprian Craciun, Marian Neagul, Silviu Panica, Beniamino Di Martino, Salvatore Venticinque, Massimiliano Rak, and Rocco Aversa. 2011. Architecturing a sky computing platform. In *Towards a Service-Based Internet. ServiceWave 2010 Workshops*. Vol. 6569. Springer, Berlin, 1–13.

Dana Petcu, Georgiana Macariu, Silviu Panica, and Ciprian Crăciun. 2013. Portable cloud applications-from theory to practice. *Future Generation Computer Systems* 29, 6 (2013), 1417–1430.

Massimiliano Rak, Salvatore Venticinque, Tamas Mahr, Gorka Echevarria, and Gorka Esnal. 2011. Cloud application monitoring: The mOSAIC approach. In *Proceedings of the 3rd IEEE International Conference onCloud Computing Technology and Science (CloudCom'11)*. 758–763.

Rajiv Ranjan and Liang Zhao. 2011. Peer-to-peer service provisioning in cloud computing environments. *Journal of Supercomputing* 65, 1 (2011), 1–31.

B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Galan. 2009. The Reservoir model and architecture for open federated Cloud computing. *IBM Journal of Research and Development* 53, 4 (2009), 1–11.

Benny Rochwerger, Constantino Vázquez, David Breitgand, David Hadas, Massimo Villari, Philippe Massonet, Eliezer Levy, Alex Galis, Ignacio M. Llorente, Rubén S. Montero, Yaron Wolfsthal, Kenneth Nagin, Lars Larsson, and Fermín Galán. 2011. An architecture for federated cloud computing. In *Cloud Computing*. John Wiley & Sons, 391–411.

Luis Rodero-Merino, Luis M. Vaquero, Victor Gil, Fermín Galán, Javier Fontán, Rubén S. Montero, and Ignacio M. Llorente. 2010. From infrastructure delivery to service management in clouds. *Future Generation Computer Systems* 26, 8 (2010), 1226–1240.

Mohsen Amini Salehi, Bahman Javadi, and Rajkumar Buyya. 2012. QoS and preemption aware scheduling in federated and virtualized Grid computing environments. *Journal of Parallel and Distributed Computing* 72, 2 (2012), 231–245.

Lutz Schubert, Keith Jeffery, and Burkhard Neidecker-Lutz. 2010. *The Future for Cloud Computing: Opportunities for European Cloud Computing Beyond 2010*. Technical Report.

Peter Sewell, Pawel T. Wojciechowski, and Asis Unyapoth. 2010. Nomadic pict: Programming languages, communication infrastructure overlays, and semantics for mobile computation. *ACM Transactions on Programming Languages and Systems* 32, 4 (2010), 12:1–12:63.

Mukesh Singhal, Santosh Chandrasekhar, Tingjian Ge, Ravi Sandhu, Ram Krishnan, Gail-Joon Ahn, and Elisa Bertino. 2013. Collaboration in multicloud computing environments: Framework and security issues. *Computer* 46, 2 (2013), 76–84.

Borja Sotomayor, Rubén S. Montero, Ignacio M. Llorente, and Ian Foster. 2009. Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing* 13, 5 (2009), 14–22.

Ananth I. Sundararaj, Ashish Gupta, and Peter A. Dinda. 2004. Dynamic topology adaptation of virtual networks of virtual machines. In *Proceedings of the 7th Workshop on Languages, Compilers, and Run-Time Support for Scalable Systems (LCR'04)*. ACM, 1–8.

Hassan Takabi, James B. D. Joshi, and Gail-Joon Ahn. 2010. Security and privacy challenges in cloud computing environments. *Security Privacy* 8, 6 (2010), 24–31.

Adel Nadjaran Toosi, Rodrigo N. Calheiros, Ruppa K. Thulasiram, and Rajkumar Buyya. 2011. Resource provisioning policies to increase IaaS provider's profit in a federated cloud environment. In *Proceedings of the 13th IEEE International Conference on High Performance Computing and Communications (HPCC'11)*. 279–287.

Adel Nadjaran Toosi, Ruppa K. Thulasiram, and Rajkumar Buyya. 2012. Financial option market model for federated cloud environments. In *Proceedings of the 5th IEEE/ACM International Conference on Utility and Cloud Computing (UCC'12)*. IEEE/ACM, 3–12.

Carlos A. Varela and Gul Agha. 2001. Programming dynamically reconfigurable open systems with SALSA. *SIGPLAN Notices* 36, 12 (2001), 20–34.

Constantino Vázquez, Eduardo Huedo, Rubén S. Montero, and Ignacio Martín. Llorente. 2009. Dynamic provision of computing resources from grid infrastructures and cloud providers. In *Workshops at the Grid and Pervasive Computing Conference (GPC'09)*. IEEE, Geneva, Switzerland, 113–120.

Salvatore Venticinque, Rocco Aversa, Beniamino Martino, Massimilano Rak, and Dana Petcu. 2011. A cloud agency for SLA negotiation and management. In *Euro-Par 2010 Parallel Processing Workshops*, MarioR.

Guarracino, Frédéric Vivien, JesperLarsson Träff, Mario Cannatoro, Marco Danelutto, Anders Hast, Francesca Perla, Andreas Knüpfer, Beniamino Martino, and Michael Alexander (Eds.). Lecture Notes in Computer Science, Vol. 6586. Springer, Berlin, 587–594.

David Villegas, Norman Bobroff, Ivan Rodero, Javier Delgado, Yanbin Liu, Aditya Devarakonda, Liana Fong, S. Masoud Sadjadi, and Manish Parashar. 2012. Cloud federation in a layered service model. *Journal of Computer and System Sciences* 78, 5 (2012), 1330–1344.

Jens-Sönke Vöckler, Gideon Juve, Ewa Deelman, Mats Rynge, and Bruce Berriman. 2011. Experiences using cloud computing for a scientific workflow application. In *Proceedings of the 2nd International Workshop on Scientific Cloud Computing (ScienceCloud'11)*. ACM, 15–24.

Matthias Winkler, Thomas Springer, and Alexander Schill. 2010. Automating composite SLA management tasks by exploiting service dependency information. In *Proceedings of IEEE 8th European Conference onWeb Services (ECOWS'10)*. 59–66.

Cheng-Zhong Xu, Jia Rao, and Xiangping Bu. 2012. URL: A unified reinforcement learning approach for autonomic cloud management. *Journal of Parallel and Distributed Computing* 72, 2 (2012), 95–105.

Zehua Zhang and Xuejie Zhang. 2012. An economic model for the evaluation of the economic value of cloud computing federation. In *Future Communication, Computing, Control and Management*, Ying Zhang (Ed.). Lecture Notes in Electrical Engineering, Vol. 141. Springer, Berlin, 571–577.

Received March 2013; revised December 2013; accepted February 2014

Online Appendix to: Interconnected Cloud Computing Environments: Challenges, Taxonomy, and Survey

ADEL NADJARAN TOOSI, RODRIGO N. CALHEIROS, and RAJKUMAR BUYYA,
The University of Melbourne, Australia

A. SUMMARY OF PROJECTS

In this article, we reviewed the main projects related to interconnected clouds. These projects are summarized in Table III. As can be seen in the table, most of the projects exclusively follow client-centric approaches using aggregated service by a broker. Exceptions are RESERVOIR [Rochwerger et al. 2009] and Inter-cloud [Bernstein et al. 2009], which are provider-centric cloud federation approaches. This is because provider-centric approaches require more standard interfaces and more components to be installed by providers, which is hard to be achieved with the significant diversity existing at current cloud providers.

We also identified challenges and obstacles for the Inter-cloud realization and proposed potential enablers for each challenge. Table IV outlines surveyed projects according to the challenges they address. References for articles regarding a specific challenge are provided in the table when there is a specific publication in the literature on that regard. Even though there is a significant number of works, none of them addresses all the challenges and we realized that issues regarding economic, networking, SLA, and security aspects received less attention from the community compared to other challenges.

Table III. Summary of Projects

Project Name	Motivations	Interoperability	Scenario
Contrail	Scalability and Wider Resource Availability, Interoperability and Avoiding Vendor Lock-in	Hybrid (Emphasis on Standards)	Client-centric, Aggregated Service (federation-support)
RESERVOIR	Scalability and Wider Resource Availability, Interoperability and Avoiding Vendor Lock-in	Hybrid	Provider-centric, Federation
Cloudbus Intercloud	Scalability and Wider Resource Availability, Geographic distribution and low latency access	Hybrid (Emphasis on Broker)	Client-centric, Aggregated Service (Cloud Exchange and Cloud Coordinators)
mOSAIC	Interoperability and Avoiding Vendor Lock-in, Geographic Distribution and Low Latency Access	Broker	Client-centric, Multi-Cloud
Open Cirrus	Research testbed	Borker (Middleware)	Client-centric, Aggregated Service
OPTIMIS	Scalability and Wider Resource Availability, Legal Issues and Meeting Regulations	Hybrid	Client-centric and provider-centric, Multi Cloud, Federation, Aggregated Service, Hybrid
Claudia	Scalability and Wider Resource Availability, Interoperability and Avoiding Vendor Lock-in	Hybrid	Client-centric, Aggregated Service

(Continued)

DOI: http://dx.doi.org/10.1145/2593512

Table III. Continued

Project Name	Motivations	Interoperability	Scenario
Intercloud by Bernstein et al.	Interoperability and Avoiding Vendor Lock-in	Standards	Provider-centric, Federation (Inter-cloud)
FCM	Interoperability and Avoiding Vendor Lock-in	Broker	Client-centric, Multi-Cloud
Sky Computing	Scalability and Wider Resource Availability, Interoperability and Avoiding Vendor Lock-in	Broker (Middleware)	Client-centric, Multi-Cloud, Aggregated Service
STRATOS	Interoperability and Avoiding Vendor Lock-in, Cost Efficiency and Saving Energy	Broker	Client-centric, Aggregated Service

B. STANDARDIZATION ACTIVITIES

B.1. Distributed Management Task Force (DMTF)

The Distributed Management Task Force[60] is an association involving 160 member companies and organizations and more than 4,000 active members spread across 43 countries that develops, maintains, and promotes standards for interoperable IT systems management. With specific reference to cloud computing, the DMTF has introduced standards and promoted several initiatives for the endorsement of interoperable cloud technologies. In the following, our aim is to briefly explain these standards and initiatives that help in enabling cloud interoperability.

Open Virtualization Format (OVF). OVF[61] is an open, secure, portable, efficient, and extensible format for the packaging of software to be run in virtual machines. OVF is vendor independent and has been designed to facilitate the portability and deployment of virtual appliances (ready-to-run certified applications packaged as virtual machines) across different virtualization platforms (e.g., Virtual Box, Xen, VMware Workstation, and Parallels Workstation). The OVF specification has been successfully accepted by companies and the open-service community. Several open-source initiatives and commercial products can import software appliances distributed as OVF packages.

Open Cloud Standards Incubator (OCSI). OCSI focuses on standardization of interactions between cloud computing environments by developing cloud management use cases and defining interactions to facilitate interoperability. The activity of the incubator has resulted in a collection of white papers[62] guiding development of interoperable cloud systems. The work has been addressed in the Cloud Management Work Group (CMWG)[63] and the Cloud Auditing Data Federation (CADF) Work Group.[64]

[60]Distributed Management Task Force (DMTF), http://www.dmtf.org/.

[61]Open Virtualization Format Specification: A white paper by the Distributed Management Task Force (DMTF), http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.0.0.pdf.

[62]Interoperable Clouds: http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0101_1.0.0.pdf, Architecture for Managing Clouds: http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0102_1.0.0.pdf, Use Cases and Interactions for Managing Clouds: http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0103_1.0.0.pdf.

[63]The CMWG is developing a set of prescriptive specifications that deliver architectural semantics as well as implementation details to achieve interoperable management of clouds between service requesters/developers and providers. This WG is proposing a resource model that captures the key artifacts identified in the "Use Cases and Interactions for Managing Clouds" document produced by the Open Cloud Incubator.

[64]The CADF will develop specifications for federating audit event data including interface definitions and a compatible interaction model that will describe interactions between IT resources for cloud deployment models. The CADF is also working closely with the DMTF Cloud Management Working Group (CMWG) to reference its resource model and interface protocol works.

Table IV. Summary of Projects and Challenges They Address

Project Name	Security	SLA	Monitoring	Portability	Network	Economy	Provisioning	Autonomics
Contrail	-Identity Management [Harsh et al. 2011] -Trust [Harsh et al. 2011]	-Federated SLA Management -SLA Monitoring and dependency [Harsh et al. 2011]	-Monitoring	-VM Portability Data Portability	-Connectivity	-N/A	-Discovery -Selection	-Autonomics
RESERVOIR	-Identity Management [Celesti et al. 2010a, 2010c] -Trust [Celesti et al. 2010a]	-Legal Issues	-Monitoring [Lindner et al. 2010]	-VM Mobility [Elmroth and Larsson 2009] -VM Portability [Celesti et al. 2010b]	-Naming [Celesti et al. 2010d] -Connectivity [Elmroth and Larsson 2009] -VM mobility and Addressing [Elmroth and Larsson 2009]	-Accounting and billing [Lindner et al. 2010; Elmroth et al. 2009]) -Pricing [Goiri et al. 2011]	-Allocation [Goiri et al. 2011]	-Autonomics
Cloudbus InterCloud	-N/A	-Federation Level Agreement [Toosi et al. 2011; Calheiros et al. 2012a]	-N/A	-VM Mobility	-N/A	-Market -Pricing [Toosi et al. 2011] [Toosi et al. 2012] -Accounting and Billing	-Allocation [Calheiros et al. 2012a] -Discovery -Selection [Calheiros et al. 2012a]	-Autonomics
OPTIMIS	-Trust	-Legal Issues	-Monitoring	-N/A	-N/A	-N/A	-Allocation -Selection	-Autonomics
Sky Computing	-Trust	-N/A	-N/A	-VM portability	-Connectivity	-N/A	-N/A	-N/A
STRATOS	-N/A	-SLA Monitoring	-Monitoring	-N/A	-Connectivity	-N/A	-Selection	-Autonomics
mOSAIC	-N/A	-Federated SLA Management SLA Monitoring and dependency [Venticinque et al. 2011]	-Monitoring [Rak et al. 2011]	-VM Portability [Petcu 2011] Data Portability [Petcu et al. 2013]	-N/A	-N/A	-Discovery [Moscato et al. 2011] -Selection [Moscato et al. 2011]	-Autonomics

(Continued)

Table IV. Continued

Project Name	Security	SLA	Monitoring	Portability	Network	Economy	Provisioning	Autonomics
Open Cirrus	-Identity Management -Key Management	-N/A	-Monitoring	-N/A	-N/A	-N/A	-Allocation	-N/A
Claudia	-N/A	-N/A	-Monitoring	-VM portability	-N/A	-N/A	-Selection	-Autonomics
FCM	-N/A	-Federated SLA Management SLA Monitoring and Dependency Lagal Issues	-Federated Cloud Monitoring	-VM portability -Data portability	-N/A	-N/A	-Selection	-Autonomics [Kecskemeti et al. 2012]
Intercloud by Bernstein et al.	-Identity Management [Bernstein and Vij 2010b] -Trust [Bernstein and Vij 2010b] -Key management [Bernstein and Vij 2010b]	-N/A	-N/A	-VM mobility [Bernstein et al. 2009] -VM portability [Bernstein et al. 2009] Data Portability [Bernstein and Vij 2010c]	-Connectivity [Bernstein et al. 2009] -Addressing [Bernstein et al. 2009] -Naming [Bernstein et al. 2009] -Multicast [Bernstein et al. 2009]	-Market [Bernstein et al. 2011] -Pricing	-Discovery [Bernstein and Vij 2010a]	-Autonomics

B.2. Open Grid Forum (OGF)

The OGF[65] is an open community committed to driving the rapid evolution and adoption of applied distributed computing such as grids and clouds. The OGF community develops standards through an open process for development, creation, and promotion of relevant specifications and use cases.

Open Cloud Computing Interface Working Group (OCCI-WG). The Open Cloud Computing Interface (OCCI)[66] includes a set of open APIs and protocols delivered by the OGF. Initially, the OCCI was proposed as a remote management API for services of the IaaS cloud model. Later on, it evolved into a flexible API with a focus on integration, portability, and interoperability for all cloud models including IaaS, PaaS, and SaaS.

Current OCCI specifications are released as three documents consisting of the following:

—OCCI Core:[67] Describes the formal definition of the OCCI core model
—OCCI Renderings:[68] Defines how to interact with the OCCI Core Model using the RESTful OCCI API
—OCCI Extensions:[69] Contains the definition of the OCCI Infrastructure extension for the IaaS domain

In order to be OCCI compliant, a cloud resource provider has to

(1) define the services and resources it offers, according to the OCCI core model, and
(2) provide a RESTful interface allowing clients to discover the set of resources it exposes according to the OCCI HTTP rendering model.

The OCCI is a promising step toward the definition of cloud interoperability standards for a cloud federation scenario. Currently, different open-source initiatives such as jclouds,[70] libvirt,[71] OpenNebula [Moreno-Vozmediano et al. 2012], and OpenStack;[72] research projects like RESERVOIR [Rochwerger et al. 2009] and Claudia [Rodero-Merino et al. 2010]; and consortia like SLA@SOI[73] are offering OCCI interfaces to their services.

B.3. Cloud Security Alliance (CSA)

The Cloud Security Alliance (CSA)[74] is a nonprofit organization whose mission is to address cloud security aspects in the cloud. It acts as a standardizing body by offering a context in which to discuss security practices and provide guidance for the development of reliable and secure cloud computing systems.

Exploration of all the categories of CSA initiatives and research falls outside the scope of this article. We briefly discuss one of the most relevant initiatives; however, there are other active initiatives. Interested readers are referred to the CSA's website for more information.

Cloud Controls Matrix (CCM). The CCM is a matrix designed to provide basic security principles for guiding cloud vendors and assist prospective cloud service consumers

[65]Open Grid Forum (OGF), http://www.gridforum.org/.
[66]Open Cloud Computing Interface (OCCI), http://occi-wg.org/.
[67]http://ogf.org/documents/GFD.183.pdf.
[68]http://ogf.org/documents/GFD.185.pdf.
[69]http://ogf.org/documents/GFD.184.pdf.
[70]jClouds, http://code.google.com/p/jclouds/.
[71]libvirt, http://libvirt.org/.
[72]OpenStack, http://www.openstack.org/.
[73]SLA@SOI, http://sla-at-soi.eu/.
[74]Cloud Security Alliance (CSA), https://cloudsecurityalliance.org/.

in assessing the overall risks implied in leveraging a cloud service provider. The CCM reduces and identifies consistent security threats and vulnerabilities in the cloud, provides standardized security and operational risk management, and seeks to normalize security expectations and security measures implemented in the cloud.

The relevance of the Cloud Controls Matrix within a cloud federation scenario is evident. It provides a standard way for assessing the security measures of each cloud service provider and helps to define a minimum security profile within a federated cloud scenario, thus increasing trust in the concept of federation.

B.4. Open Cloud Manifesto

Open Cloud Manifesto,[75] a public declaration of principles and intentions of a group of cloud service providers, constitutes the first step toward the realization of a cloud interoperability platform. As a result of this coordinated activity of cloud vendors, the manifesto was drafted in 2009. Instead of proposing standards, the document is a declaration of intent to establish a core set of principles to bring together the emerging cloud computing community. The Open Cloud Manifesto, in fact, is dedicated to the belief that the cloud should be open.

The manifesto declares the goals of an open cloud platform and admits that as an open cloud becomes a reality, the cloud community will benefit in several ways that can be summarized into the following:

Choice: IT consumers are able to select different providers, architectures, or usage models as the business environment changes with the use of an open cloud technology.

Flexibility: An open cloud makes it easy for cloud customers to interoperate between different cloud providers. Moreover, when different vendors do not use a closed proprietary technology, change between one provider and another is facilitated and considerable switching costs is diminished.

Speed and Agility: The use of open interfaces facilitates the integration of public clouds, private clouds, and current IT systems. The promise of on-demand scaling of hardware and software with speed and agility is realized.

Skills: The possibility of finding someone with appropriate skills by an organization using cloud services increases with an open cloud, because there will be a smaller set of new technologies to learn by professionals.

By confirmation of the advantages of an open cloud platform, the manifesto lays out the conceptual foundations for a cloud federation scenario. In fact, the use of open technologies will create a more flexible environment where cloud consumers will more comfortably choose cloud computing technologies, without feeling the menace of the vendor lock-in. The concept of the cloud federation constitutes an evolution of this initial vision, which implies a more structured and explicit collaboration.

B.5. National Institute of Standards and Technologies (NIST)

The National Institute of Standards and Technologies (NIST)[76] proposed a widely accepted definition for important aspects of cloud computing. The NIST's activities are mostly related to the assessment of existing standards in cloud computing, to actively contribute to the creation of open standards and to the identification of gaps in existing standards.

B.5.1. NIST Cloud Computing Standards Roadmap Working Group (CCSRWG). The role of the NIST Cloud Computing Standards Roadmap Working Group is to study the security, portability, and interoperability standards, models, and use cases to support U.S.

[75]Open Cloud Manifesto, http://www.opencloudmanifesto.org/.
[76]NIST, http://www.nist.gov/.

government (USG)-wide use of cloud computing. High-priority strategic and tactical requirements for USG cloud adoption including current standards, standards gaps, and standardization priorities are identified and reported by the group. Outcomes of the group include the USG cloud computing technology roadmap document,[77] which is designed to foster adoption of cloud computing by USG federal agencies and the private sector, to improve the information for the decision makers, and to facilitate more development of cloud computing. The roadmap document identifies 10 high-priority requirements that must be met for further adaptation of the cloud computing model by the USG. Among these requirements, the importance of frameworks to support federated community cloud environments is clearly identified and the importance of interclouds and cloud federation is recognized as a high priority. CCSRWG has also compiled an Inventory of Standards relevant to cloud computing[78] to review the state of standardization supporting cloud computing.

B.6. Cloud Computing Interoperability Forum (CCIF)

The CCIF[79] is an industry forum formed in order to enable a global cloud computing ecosystem whereby organizations can seamlessly cooperate for the purposes of wider adoption of the cloud computing technology. It focused on building community consensus, exploring emerging trends, and advocating best practices and reference architectures for the purposes of standardized cloud computing.

The activity of the forum has led to the proposal of the Unified Cloud Interface (UCI). The UCI is an attempt to provide a unified interface to the various APIs exposed by different vendors. Functional implementations exist for Amazon EC2 and Enomaly ECP.

B.7. Open Cloud Consortium (OCC)

The OCC[80] is a nonprofit organization that manages and operates cloud computing infrastructures to support scientific research. The OCC presents itself as a Science as a Service, or precisely the infrastructure, platform, and services to support science as a service. The OCC manages cloud computing test-beds, such as the Open Cloud Testbed, to improve cloud computing software and services. It also develops reference implementations, benchmarks, and standards for cloud technologies.

B.8. Organization for the Advancement of Structured Information Standards (OASIS)

OASIS[81] is a consortium that supports the development, convergence, and adoption of open standards for the global information society. It focuses on improving the standards development process, improving the quality of standards, and advising on improvements to community and collaboration processes and technologies. OASIS sees cloud computing as a natural extension of SOA and network management models. We review

[77]NIST Special Publication 500-293, US Government Cloud Computing Technology Roadmap, Volume I: High-Priority Requirements to Further USG Agency Cloud Computing Adoption, http://www.nist.gov/itl/cloud/upload/SP_500_293_volumeI-2.pdf, and Volume II: Useful Information for Cloud Adopters, http://www.nist.gov/itl/cloud/upload/SP_500_293_volumeII.pdf.

[78]NIST Inventory of Standards Relevant to Cloud Computing, http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/StandardsInventory.

[79]Cloud Computing Interoperability Forum (CCIF), http://www.cloudforum.org/.

[80]Open Cloud Consortium (OCC), http://opencloudconsortium.org/.

[81]Organization for the Advancement of Structured Information Standards (OASIS), https://www.oasis-open.org/.

a number of the OASIS Cloud-Specific or Technical Committees (TCs) that are deeply committed to building cloud models, profiles, and extensions on existing standards.

OASIS Identity in the Cloud (IDCloud). The OASIS IDCloud TC develops profiles of open standards for identity management in cloud computing. It identifies gaps in existing identity management standards and investigates the need for profiles to achieve interoperability within current standards. The IDCloud performs risk and threat analyses on collected use cases and produces guidelines for mitigation of vulnerabilities.

OASIS Symptoms Automation Framework (SAF). Since it is difficult for different enterprises and domains to cooperate with each other to fix issues and recognize and respond to their customers' needs, the SAF provides a catalog-based XML collaborative knowledge framework that enables diverse enterprises and domains to address these issues through automation at lower cost and more effectively. It facilitates knowledge sharing between cloud consumers and providers by publishing information about conditions and appropriate responses that helps them to optimize their business relationship.

OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA). The OASIS TOSCA TC's goal is to enhance the portability of cloud applications and services. It facilitates this goal by enabling an interoperable description of application and infrastructure cloud services, the operational behavior of these services, and the relationships between parts of the service, independent of the supplier of service and any particular cloud provider or hosting technology.

TOSCA enhances service and application portability in a vendor-neutral ecosystem by enabling portable deployment to the compliant cloud, smooth migration of existing applications to the cloud, flexible bursting, and dynamic multicloud provider applications.

B.9. Inter-Cloud Technology Forum (GICTF)

The GICTF[82] is a voluntary Japanese organization whose aim is to bring together the knowledge developed within industry, academia, and government to support research and development on the technologies related to cloud interoperability. More precisely, it proposes standard interfaces and network protocols that allow cloud system interoperation to happen.

B.10. The European Telecommunications Standards Institute (ETSI)

ETSI[83] is an independent, nonprofit standardization organization in the telecommunications industry in Europe, with worldwide projection. ETSI has more than 700 member organizations drawn from 62 countries.

TC CLOUD. The goal of ETSI TC CLOUD is to address issues associated with the convergence between IT and telecommunications. The focus is on interoperable solutions in situations where connectivity goes beyond the local network. TC CLOUD has particular interest in the Infrastructure as a Service (IaaS) delivery model. TC CLOUD focuses on global standards and validation tools to support these standards. They promote standards toward a coherent and consistent general purpose cloud infrastructure.

B.11. The Open Group Cloud Computing Work Group

This is an initiative of the Open Group aimed at creating a common understanding between buyers and suppliers of how enterprises of all sizes and scales can leverage

[82]InterCloud Technology Forum (GICTF), http://www.gictf.jp/index_e.html.
[83]The European Telecommunications Standards Institute (ETSI), http://www.etsi.org/.

cloud computing technologies.[84] The group has established several activities to enhance business understanding, analysis, and uptake of cloud computing technologies.

B.12. Object Management Group (OMG)

This is a consortium originally aimed at setting standards for distributed object-oriented systems; it is now focused on modeling and model-based standards.[85] It proposes cloud-related specifications focusing on modeling deployment of applications and services on clouds for portability, interoperability, and reuse.

B.13. Open Data Center Alliance (ODCA)

The Open Data Center Alliance[86] was formed in 2010 as a unique consortium of leading global IT organizations. Its mission is to speed up the migration to cloud computing by enabling solutions for service ecosystem and addressing IT requirements with the highest level of interoperability and standards. It focused on open, interoperable, standard solutions for a secure cloud federation, automation of cloud infrastructure, common management, and transparency of cloud service delivery.

B.14. IEEE P2302 Working Group (Intercloud)

The IEEE P2302 Working Group[87] established six subgroups responsible for different aspects of *Intercloud Interoperability and Federation (SIIF)*. Its aim is to prepare a draft for the IEEE Standard. It defines topology, functions, collaboration protocols, security, ontology, and governance for cloud-to-cloud interoperability and federation.

B.15. Storage Networking Industry Association (SNIA) Cloud Storage Initiative (CSI)

The SNIA[88] promotes IT technologies, standards, and education programs for IT professionals. The SNIA Cloud Storage Initiative (CSI)[89] is a standardization organization for DaaS (Data Storage as a Service) and provides the standard API for managing data storage services called Cloud Data Management Interface (CDMI). The CDMI is an interface that cloud applications use to create, retrieve, update, and delete data elements from the cloud. As part of this interface, clients are able to discover services of the cloud storage and use this interface to manage the data that is placed in the cloud. In addition, the CDMI allows cloud users to tag their data with special metadata that tells the cloud storage provider what data services (such as backup, archive, and encryption, among others) to provide for that data. By implementing CDMI, cloud users are free to move data between cloud providers without the burden of recoding to different interfaces.

B.16. ISO JTC 1/SC 38

The ISO JTC 1 is a technical committee of the International Organization for Standardization (ISO). The purpose of the committee is to develop, maintain, promote, and facilitate standards in the fields of IT and information and communications technology (ICT). The ISO JTC 1/SC 38 is a subcommittee that works on Distributed Application Platform and Services (DAPS). The subcommittee includes three main working groups

[84]The Open Group Cloud Computing Work Group, http://www.opengroup.org/getinvolved/workgroups/cloudcomputing.
[85]Object Management Group (OMG), http://www.omg.org/.
[86]Open Data Center Alliance (ODCA), http://www.opendatacenteralliance.org/.
[87]IEEE P2302 Working Group (Intercloud), http://grouper.ieee.org/groups/2302/.
[88]Storage Networking Industry Association (SNIA), http://www.snia.org/.
[89]Cloud Storage Initiative (CSI), http://www.snia.org/forums/csi.

that are acting in the following areas: web services, service-oriented architecture (SOA), and cloud computing.

Both the OGF and SNIA have a Category A liaison with ISO JTC1 SC/38 on cloud computing and are working with the ISO on joint activities. They cooperate to promote OGF OCCI and SNIA CDMI into ISO standards. The DMTF OVF has also been promoted to an ISO standard by ISO JTC 1/SC 38.

B.17. ITU-T Focus Group on Cloud Computing

The International Telecommunication Union's Telecommunication Standardization Sector (ITU-T) brings experts together to develop standards known as ITU-T recommendations that define key aspects in the ICTs. These standards are helpful to the interoperability of ICTs. The ITU-T Focus Group on Cloud Computing (FG Cloud)'s objective is to collect and document information and concepts that are helpful for developing recommendations to support cloud computing from a telecommunication perspective. FG Cloud includes two working groups that are involved in two main areas:

—Cloud computing benefits and requirements
—Gap analysis and roadmap on cloud computing standards developments

B.18. Standards and Interoperability for eInfrastructure implemeNtation initiAtive (SIENA)

SIENA[90] is a European-funded initiative that brings together experts from research centers, academic institutions, and major enterprise companies to accelerate and coordinate the adoption of interoperable distributed computing infrastructures (DCIs). SIENA's objective is to produce a roadmap focusing on interoperability and standards, similar to NIST's roadmap, on the adoption of cloud technologies that help the rapid spread of services. The roadmap assesses the situation, recognizes trends, identifies issues, and delivers insights and recommendations on adoption and evolution of grid and cloud standards shaping current and future development and deployment of cloud for e-science in Europe and globally.

[90]Standards and Interoperability for eInfrastructure implemeNtation initiAtive (SIENA), http://www.sienainitiative.eu/.