

REVIEW

Open Access



Issues on developing interoperable cloud applications: definitions, concepts, approaches, requirements, characteristics and evaluation models

Elias Nogueira^{1*}, Ana Moreira², Daniel Lucrédio³, Vinícius Garcia⁴ and Renata Fortes¹

*Correspondence:

eliasadriano2010@gmail.com

¹ICMC-USP, São Carlos, Brasil

Full list of author information is available at the end of the article

Abstract

Among research opportunities in software engineering for cloud computing model, interoperability stands out. We found that the dynamic nature of cloud technologies and the battle for market domination make cloud applications locked-id, i.e, proprietary, non-portable and non-interoperable. In general context of cloud computing, interoperability goes beyond communication between systems like in other fields, it goes in direction of more dynamic, heterogeneous, complex and composed applications that take advantage of best features from different providers and services simultaneously. Interoperability in cloud constitutes a great challenge that must be overcome for that, in the future, software be more dynamic and improved.

Objective: This paper aims at identifying how interoperability in cloud computing has been addressed in the existing literature, offering an up-to-date view of concepts relate to how to develop interoperable software that takes advantage of different cloud models. Thus, providing a basis for further research in the field and consolidating e better exploring existing concepts.

Method: To fulfill this objective, we surveyed literature. We defined six research questions and conducted the study according to a protocol that included planning, and execution.

Results: A first result of the review is that there is no well established definition for cloud interoperability. This study also identified cloud interoperability concepts (e.g., cloud brokers, multi-cloud and cloud federation), requirements for interoperable applications and existing cloud interoperability solutions, showing that these are either too specific for particular situations. Finally, the survey found no evaluation models for cloud interoperability solutions. We also present a discussion on the findings of this study.

Conclusion: Since the study observed that there are no well-established cloud interoperability solutions yet, we conclude that the issues raised by lack of interoperability persist. Selecting one interoperable solution or even a cloud standard can free the system from the underlying providers, but it would still be locked into the selected particular solution.

Keywords: Cloud computing, Software engineering, Lock-in, Interoperability, Multi-cloud, Federation, Inter-cloud

1 Introduction

In a report, the *IEEE Computer Society*¹ highlights 22 technologies with potential to change the scenario of computer science and its role in industry (Alkhatib et al. 2014). Apart from foreseeing relevant research paths up to 2022, it also discusses a vision of a possible future for each one of those technologies. One such technology is cloud computing.

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction (Mell and Grance 2010). Having gained notoriety in industry, cloud computing is also capturing the attention of the academic community. The number of scientific publications presenting methods, processes and tools for the cloud is constantly increasing as researchers investigate how cloud computing can be used to promote advances in several areas of computer science (da Silva and Lucrecio 2012).

In a previous paper (da Silva and Lucrecio 2012) we performed a systematic literature review and identified how researchers from the software engineering field are viewing the cloud computing paradigm. We identified ten software engineering research opportunities focused specifically on cloud computing, which are presented and discussed in terms of related work. We also presented a discussion on some practical issues related to the development of software for the cloud, trying to make some obscure points clearer and aiming to facilitate the work of software engineering researchers and practitioners. We found that the dynamic nature of cloud technologies and the battle for market domination make cloud applications locked-id, i.e, proprietary, non-portable and non-interoperable. Interoperability in cloud is a great challenge and needs to be better explored.

Developers have a multitude of cloud computing providers available, each one with different technical characteristics, costs and benefits. In this scenario, developers may want to build an application to be deployed over several different clouds, choosing the best features of each provider, to deliver a final product with higher quality or lower cost. A small case that illustrates this scenario could be an application that uses Windows Azure PaaS to store non-structured data as videos and images while, at the same time, uses free features from the Google App engine platform.

With the existing body of knowledge and technology, building such systems is still a complex task (Alkhatib et al. 2014). Standards are divergent and scarce, causing solutions to be often proprietary and non-interoperable. Many technologies can be used to enable development for multiple clouds (Petcu 2013), such as cloud brokers and federated clouds. However, the challenge remains, mainly due to the so-called **lock-in** problem. Lock-in is caused by specific programming styles and data management policies imposed by the providers to offer augmented elasticity and flexibility of their cloud architectures (Armbrust et al. 2009; da Silva et al. 2013). The consequence is the development of proprietary-like solutions (e.g. data and programs) that are non-interchangeable between platforms (Armbrust et al. 2009).

In single cloud development, lock-in is also caused by the dynamic nature of information technology and the constant rise of new cloud platforms and services. In the actual stages of the cloud technology development, cloud platforms and APIs might become rapidly obsolete, forcing developers to spend effort in maintenance and re-engineering to

move their systems to new providers or to adapt them to new demands. In multiple cloud development, the complexity of combining multiple resources from different platforms and/or services exacerbates the problem. Hence, developing interoperable multi-cloud systems is a challenge (Delgado 2014; Loutas et al. 2011; Petcu 2014; 2013). As the IEEE report (Alkhatib et al. 2014) suggests, an effort to produce a cloud infrastructure that allows industry players to cooperate in the development of interoperable modules is needed.

Until this becomes concrete, we need to understand the existing body of work and the choices available to developers and researchers. In direction of this objective, we surveyed literature aiming at **identifying what are the definitions, concepts, approaches, requirements, characteristics, evaluation models and the key concepts regarding interoperability** specifically in context of cloud computing. We defined six research questions and conducted a literature survey. We wanted to share our observations because, after a considerable review work, we believe to have reached a point where we should compile the information.

In summary our conclusion is that, although cloud interoperability is available today, lock-in caused issues and lack of standards demand from developers difficult decisions when time arrives to choose one or more providers. Even in contexts that exist standards, choosing for one may lead to Standard-Lock-In, i.e, the adopter becomes locked-in in chosen standard. Thus, future research should focus on creating platform-independent mechanisms to support transparent, locked-in-free software for the cloud, so that the underlying details can be more easily replaced as the cloud technologies evolve, a broader discussion and remaining results of our research are presented throughout this work.

The rest of this paper is organized as follows. Section 2 presents methodology that we followed. Section 3 presents main concepts related to cloud computing in general. Section 4 presents research objectives and literature survey questions. Section 5 presents the evaluation of the research questions. Section 6 presents a discussion about each issue and the process of performing this work. Section 7 presents directions for future research in the field. Section 8 present some works related to this. Section 9 presents some threats to validity of this research and, finally, Section 10 presents concluding remarks and acknowledgments respectively.

2 Review

Hence, this study may serve as a prelude to investigate a research topic and identify further research activities. We surveyed literature aiming at **identifying what are the definitions, concepts, approaches, requirements, characteristics, evaluation models and the key concepts regarding interoperability** specifically in context of cloud computing. We defined six research questions and conducted a literature survey. We planned our review, defined research questions, selected studies and extracted data before, finally, to summarize extracted content and to answer research questions.

Additionally, we performed searches using well known search engines. Each step is described in detail in the following sections.

2.1 Search for similar studies

Before starting such a study, it is wise to check if a similar study already exists. We performed an ad-hoc search for cloud interoperability and cloud computing in general.

2.2 Search strategy

To define our search strategy, we identified the keywords and synonyms related to the subject of our research, the source selection criteria, accepted languages, search methods, study selection criteria, definition of study types, initial and selection process. We defined, tested and consequently used several strings. We also used snow bowling technique and included many references cited by studies we found in our searches. A specialist about the subject informed us that Armbrust et al.'s paper is among most cited (Armbrust et al. 2009), thus we considered to verify manually all papers that cited their job in order to manually include relevant ones in our research.

The **keywords and synonyms** considered were “cloud computing”, “interoperability”, “interoperable”, and the **selection criteria** involved the following three criterion:

- *Digital libraries*: we considered the most relevant digital libraries in software engineering and technology (Dyba et al. 2007), particularly: Scopus², IEEEExplore Digital Library³, ACM Digital Library⁴ and Google scholar⁵.
- *Related work*: we considered studies cited as related work in both primary and secondary studies found in the search engines. This is also known as *snowballing* analysis.
- *Selected conferences and journals*: in addition to the studies retrieved from the automatic search engines and related work analysis, we performed a manual search in well-known conferences and journals in cloud computing. This increased the chances of finding additional relevant studies.
- *Selected languages*: the **accepted language** was English, hence excluding studies published in any other language.

In what concerns the **search methods**, we first defined the search string based on the identified keywords from the Scopus database. This string was then translated to the formats accepted by each library. Our generic search string was: TITLE-ABS-KEY(("Cloud Computing") AND ("interoperability" OR "interoperable")). This string is quite comprehensive as our goal is to retrieve the maximum number of potentially relevant studies. For the two other sources (related work and selected conferences and journals) we performed manual searches by browsing their respective proceedings and websites.

After performing the searches, we proceeded to apply the **study selection criteria** to select the studies to be analysed, adopting the following inclusion and exclusion criteria:

- *Inclusion criteria*: (i) studies that report approaches related to interoperability in cloud computing; (ii) studies addressing interoperability in cloud computing.
- *Exclusion criteria*: (ii) papers which full-text was not available; (iii) previous versions of more complete studies; (iv) studies that are not related to the subject of this research.

Regarding the **definition of study types**, we considered both primary and secondary studies, including technical reports⁶. The **initial selection process** included only papers retrieved automatically from the digital libraries. The decision about inclusion and exclusion of each retrieved work was first based on their titles and abstracts, and, when necessary, we also read the concluding remarks and related work sections. During the **second selection process**, each study retrieved from the initial selection was fully read and analysed according to the inclusion and exclusion criteria. Notice that we did not

evaluate the studies against any **quality evaluation criteria**, basically being interested in any relevant study.

After the initial study selection, we initiated the extraction process. The **data extraction form fields** were:

Contextualization (aiming at extracting the context of the proposed solution), interoperability definition, problem definition, solution classification, evaluation method, metrics and observations. (The observations field contains general impressions and relevant concepts found in the study.)

The **approach for summarization of results** although we performed a qualitative analysis on the extracted data, we did not use any form of meta-analysis or quantitative methods, because the research questions did not require such analysis.

2.3 Execution

The execution consists in applying the research plan, which includes selecting the studies, extracting the data, and answering the research questions. This section presents the most relevant information about the execution of this review.

This step searches for studies by following the search method defined in the protocol. Regarding manual searches, we first checked the list of papers published in each selected conference and journal and then used Google Scholar to retrieve the studies. We retrieved 401 studies from Scopus, 65 from IEEE, and 3 from ACM. Additionally, 14 were found manually from selected conferences and journals.

As predicted, most results come from the Scopus database, since it searches other databases such as IEEE and ACM. Thus, using Scopus together with databases it indexes increases the amount of studies retrieved, as well as duplicates.

We performed searches between 2015/October and 2016/May. In the first selection, from the 483 studies initially retrieved, 386 were excluded. In the second selection, from the remaining 97 studies, 51 were accepted, 38 were rejected and 8 were found duplicated.

The full text of the all papers selected were analysed and the data extracted from them was stored in the extraction forms. To summarize results we performed a qualitative analysis of the results (Cruzes and Dyba 2011), as our research questions did not require a quantitative analysis. Following sections discuss the main findings of our study, by answering defined questions.

3 Cloud computing concepts

Cloud computing is not a new technology model, but an integration of technologies from the past (Chen et al. 2011) that resulted in a new way to make computing services available through the Internet. The main idea behind this concept is to allow companies to acquire computational resources on demand, and pay according to the utilization volume (Armbrust et al. 2009). Many definitions for cloud computing are available in the literature. Vaquero et al. (2008) performed a rigorous study of many definitions, presenting them under different perspectives. In this work, the NIST⁷ definition is adopted, as it is one of the most commonly accepted definitions (Hogan et al. 2011):

“Cloud computing is a model for enabling **ubiquitous, convenient, on-demand** network access to a **shared** pool of **configurable computing resources** (e.g.,

networks, servers, storage, applications, and services) that can be **rapidly provisioned and released** with **minimal management effort** or service provider interaction.”

Some of the keywords are highlighted. We discuss them next, as they refer to the concepts that are most commonly associated with cloud computing:

“ubiquitous, convenient, on-demand”: this refers to the fact that the customer does not need to spend too much effort to setup an environment in order to use some cloud resource. This is also known as “utility computing”, which means treating computing resources as other common utilities, such as water and electricity. In these services, the customer pays for what he/she uses and does not need to care about where these resources are coming from;

“configurable computing resources”: processing time and storage are the most common resources being used in the cloud, but the definition mentions “resources”, in a generic way, to highlight the fact that any kind of computing resource can be used as well;

“shared”: resources can be shared between customers, which means the provider can move resources from one customer to another, to minimize the amount of idle resources, thus reducing the overall investment made in building a pool of resources. However in some cases, if a customer does not want to share its resources, it is possible do not share them. Depends on the kind of agreement between the customer and the provider;

“rapidly provisioned and released”: this is also known as “elasticity”, and refers to the ability of a cloud provider to quickly respond to very dynamic demands. If a customer needs more resources, the provider can increase the customer’s share on the pool. As soon as the customer no longer needs them, they can return to the pool. Elasticity is only viable if minimal management effort is possible, which is described in the next item; and

“minimal management effort”: this means that cloud technologies should be made as automatic as possible, so that elasticity can be indeed a reality in the dynamic world of real business needs. To put it simply, provisioning and releasing resources should be made as easy as some mouse clicks.

Regarding cloud service offerings, different taxonomies have been suggested. Most are business-oriented, as they describe technologies that support a particular way to negotiate computational demand (Rimal et al. 2009). There are many common terms, such as HaaS (Hardware as a Service), SaaS (Software as a Service), PaaS (Platform as a Service), DaaS ([Development, Database, Desktop] as a Service), IaaS (Infrastructure as a Service), BaaS (Business as a Service), FaaS (Framework as a Service), or even XaaS (Everything as a Service), which refers to a generic service. The definitions behind these terms are very wide and often misunderstood (Armbrust et al. 2009). However, by observing the literature it is possible to classify the solutions in the following categories:

- **Software as a Service – SaaS**: refers to applications delivered as a service to consumers through the Internet/Intranet (Armbrust et al. 2009; Mell and Grance 2010). The applications are accessible from various client devices through a thin client interface such as a Web browser (e.g., Web-based email). The consumer does not manage or control the underlying cloud infrastructure including network,

servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user specific application and configuration settings (Hogan et al. 2011). Examples of SaaS include: Google Drive, Flickr and Picasa, among others. The target audience of this kind of offering is the end user, who does not have to worry about acquiring licences or deploying the application himself.

- **Platform as a Service – PaaS:** the idea is to provide a complete development platform as a service, over a cloud infrastructure (Rimal et al. 2009; Mell and Grance 2010). It includes application hosting, testing and delivery. Examples of PaaS include: Google App Engine, Microsoft Azure, among others. The target audience of PaaS is the developer, since the capability provided to the consumer is to deploy, onto the cloud infrastructure, consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations (Hogan et al. 2011). PaaS is suitable for the development of new applications, because most configuration details are managed by the provider (Esparza and Escóí 2011).
- **Infrastructure as a Service – IaaS:** consists of the delivery of computational infrastructure as a service (Mell and Grance 2010; Rimal et al. 2009). The target audience of this model are normally infrastructure architects. These architects do not need to worry about issues such as dynamic demands, elasticity and scalability, which are handled by the provider. The capability provided to the consumer consists of processing, storage, networks, and other fundamental computing resources, upon which the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (Hogan et al. 2011). In other words, customers can rent a virtual computer and/or some services in the provider's infrastructure. Examples of IaaS include: Amazon, GoGrid, IBM SmartCloud, among others.

Many are the technological requirements for these models to work properly, including virtualization, standards and interfaces that allow shared access to virtual servers and APIs. This section presented most known concepts related to cloud computing. An interoperable solution consider the combination of multiple services from diverse cloud service and deployments models presented. Next section presents research objectives and questions related to this research.

4 Research objectives and questions

This study may serve as a prelude to investigate the research topic and identify further research activities. We wanted to identify the existing body of knowledge about cloud computing interoperability, to find how interoperability has been addressed in the literature, and to present an updated view on the topic. The idea is to compile the main concepts presented in previous studies and produce a synthesis to help researchers and developers gaining a better understanding of the field, as well as increase the discussion about the theme. To achieve this objective we surveyed literature to answer the following six research questions:

- **RQ1:** What are the definitions of “interoperability” in the context of cloud computing?
- **RQ2:** What are the concepts related to cloud interoperability?
- **RQ3:** Is there a classification of interoperability approaches?
- **RQ4:** What are the requirements and possible usage scenarios for a cloud interoperability solution?
- **RQ5:** What are the solutions for cloud interoperability and what are their characteristics?
- **RQ6:** Are there any evaluation models for cloud interoperability solutions?

The execution of this survey consisted in applying a research plan, which includes selecting studies, extracting data, and answering the research questions. We followed a well organized research plan and performed searches in most known databases as ACM dl⁸, Scopus⁹, IEEE¹⁰ and Google Scholar.

5 Evaluation of the research questions

This section discusses the main findings of our study, by answering the six questions presented in section 4.

5.1 RQ1: What are the definitions of “interoperability” in the context of cloud computing?

Several research papers we found in literature present definitions for interoperability (Dowell et al. 2011; Harsh et al. 2012; Loutas et al. 2011; Petcu 2013; 2014; Petcu et al. 2013; Petcu 2011; Rashidi et al. 2013; Rezaei et al. 2014), we present them next.

Definition 1 *The IEEE Glossary (2013) defines interoperability as “the ability of a system or a product to work with other systems or products without special effort on the part of the customer.” According to this definition, interoperability is made possible by the implementation of standards.*

Definition 2 *A discussion group on cloud computing use cases¹¹ defines interoperability in cloud computing as the ability to write code that works with one or more cloud providers simultaneously, regardless of the differences between platforms.*

Definition 3 *Rashidi et al. (2013) define interoperability in cloud computing as the ability of a service to interact with other homogeneous or heterogeneous services to improve its provided resources. The authors state that interoperability can be implemented under one or more domains, and that cloud services implemented in different layers¹² are heterogeneous while those implemented in the same layer are homogeneous.*

Definition 4 *Dowel et al. (2011) state that cloud interoperability refers to the ease of migration and integration of applications and data between different cloud providers.*

Definition 5 *ISA (Interoperability Solutions for European Public Administrations)¹³ states that interoperability is the ability of systems and business processes to exchange data and to enable the distribution of information and knowledge.*

Definition 6 Harsh et al. (2012) define interoperability as the ability of different heterogeneous systems to interact and function together. In cloud computing, interoperability could be defined as the ability of different applications to understand each other's formats, Service Level Agreement (SLA) templates, authentication and authorization token formats and attribute data.

Definition 7 Rezaei et al. (2014) state that interoperability is a multidimensional concept and can be viewed from various perspectives and in different directions. Regarding interoperability in general (i.e., not only cloud interoperability) the authors identified four levels of interoperability issues, each one built on the top of the previous: Technical interoperability (Level 1), typically associated with hardware/software components, systems, and platforms, enabling machine-to-machine communication; Syntactical interoperability (Level 2), defined as the ability to exchange data. It is associated with data formats, and the messages exchanged in the communication must have a well-defined syntax; Semantic interoperability (Level 3), also related to data exchange, involves the interpretation of contents to give them meaning; and Organizational interoperability (Level 4), more related to the capability of organizations to exchange information regardless of the differences between the information systems being used. The success of the first three levels of interoperability forms the basis for organizational interoperability. They cite the definition used by Koussouris et al. (2011) that seem cloud interoperability as the ability of cloud services to work with different cloud services and providers, and other applications or platforms that are not cloud dependant. According Rezaei et al. (2014) cloud interoperability fits in level 3.

Definition 8 Petcu (2011) defines interoperability as a general property that refers to the ability of many systems or organizations to work together (inter-operate). From a general computing perspective, this means that data can be exchanged and used by two or more systems or components. However, for the cloud, Petcu goes beyond this simpler perspective, listing different abilities that are normally associated to cloud interoperability:

- “the ability to abstract the programmatic differences from one cloud to another”;
- “the ability to translate between the abstractions supported by different clouds”;
- “the ability to flexibly run applications locally or in the cloud, or in a combination”;
- “the ability to move applications from one environment to another or run in multiple clouds”;
- “the ability to move services, processes, workload and data between clouds”;
- “the ability to use the same management tools, server images and software in multiple clouds”;
- “the ability to communicate between providers, porting applications and data between them”;
- “the ability to federate multiple clouds to support a single application”.

Thus, there are many distinct definitions of interoperability, but they cannot always be applied to the same context. In general, we observed that interoperability is often associated with many of the advantages of the cloud model. Therefore, we need a more precise definition, a more specific one that is aligned to how interoperability has been implemented in practice.

Some of the reviewed definitions emphasise communication (e.g., Definitions 5 and 6), focusing on data and information exchange between systems. Other definitions mention properties related to platform independence, emphasizing ease of use (Definition 1), programming for different platforms (Definition 2), ease of migration (Definition 4) and support to different service levels (Definitions 3 and 8). However, in our view, this platform independence is more related to portability. As pointed out by Petcu (2011), while interoperability is the successful communication between or among systems, portability is the ability to use components or systems lying on multiple hardware or software environments. From Definition 7, interoperability in cloud computing is at level 3, and traditional systems interoperability is at level 1. One final observation about these definitions is that interoperability can also mean the integration between cloud providers. This is suggested by Petcu (Definition 8), who mentions the abilities to communicate between providers, to use the same management tools and software in multiple clouds, and to federate multiple clouds to support a single application. Derived from these observations, we propose the following definition: **cloud interoperability refers to the ability to develop applications that combine resources that can inter-operate, or work together from multiple cloud providers, hence taking advantage of specific features provided by each provider.** As example of this definition could be an application that part of it uses resources provided by Google App Engine, part uses resources provided by Windows Azure and part uses resources provided by any third provider.

Our definition is simple and enjoys a good abstraction level, but it is quite restrictive, it refers to cloud applications that use specific services from many providers and not to the communication between the infrastructure services between two cloud provider in agreement to share resources. However, we believe that cloud interoperability is not restricted to a particular strategy. More importantly than using standardized data exchange formats, integration of services/providers in different levels or federated clouds¹⁴, is the final goal of building applications that can benefit from many cloud providers, without requiring too much effort from the developer. Therefore, our definition will probably evolve, as research in cloud interoperability is still far from mature. Many definitions, approaches, and, consequently, requirements and applications, may appear as research in the field advances.

5.2 RQ2: What are the concepts related to cloud interoperability?

In this review, and based on (Dowell et al. 2011; Grozev and Buyya 2014; Kostoska et al. 2012; Liu et al. 2011; Loutas et al. 2011; Petcu 2014; Rashidi et al. 2013; Zhang et al. 2013), we identified four concepts that are normally associated with cloud interoperability: Inter-Cloud computing, Cloud Federation, Multi-Cloud Environment, and Cloud Brokering.

Inter-Cloud computing is a “*cloud model that, for the purpose of guaranteeing service quality, such as the evaluation and availability of each service, allows on-demand reassignment of resources and transfer of workload through a 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*” (Grozev and Buyya 2014; Petcu 2014). This means that applications, or parts of an application, can be deployed into, and take advantage of, more than one cloud system. Benefits include increased support for diverse geographical locations, and giving developers a fine-grained control as to where resources are positioned. Inter-cloud also results in better application resilience,

since with multiple supporting clouds there is no single point of possible service outage. Also, by using multiple clouds, a customer can easily avoid vendor lock-in (Grozev and Buyya 2014).

If individual cloud providers voluntarily participate to form an Inter-Cloud environment, the result is called a **Cloud Federation** (Grozev and Buyya 2014). This means that the providers knowingly collaborate to interconnect their infrastructure in order to allow resources to be reassigned among them. This kind of Inter-Cloud is normally adopted by governmental or non-profit organizations, since in these cases the individual providers share the common interest of increasing overall service quality and availability.

If providers are unaware of the connection, i.e. clients and/or services are the sole responsible for managing resource provisioning over multiple providers, the environment is a **Multi-Cloud** environment (Grozev and Buyya 2014). This kind of Inter-Cloud is normally adopted with independent providers, such as large commercial companies like Amazon and Google. These providers normally have no interest in sharing their resources, preferring customer exclusivity.

In order to make Inter-Cloud computing possible, a service is required for dynamic provisioning resources, deployment of application components into these resources, and redirecting incoming requests to the allocated resources (Grozev and Buyya 2014). This service is called **cloud brokering**. According to NIST (Liu et al. 2011), a cloud broker is a cloud computing actor that manages use, performance and delivery of cloud services, and negotiates relationships between many providers and cloud customers. In a multiple cloud setup, a broker could help find and manage cloud services.

Similarly to the state of cloud interoperability definitions, cloud interoperability concepts are still evolving. Petcu (2014) presents a list containing “top 25” categories of multiple clouds. However, we believe that, in essence, most of the concepts presented in her work, raise from the four highlighted here.

5.3 RQ3: Is there a classification of interoperability approaches?

Petcu (2011) identified various criteria to classify the types of cloud interoperability. These criteria refer to a classification of the solution itself, depending on the implemented interoperability strategy. They are:

1. **Agreement Level:** this level refers to the contract between two systems that need to interoperate. There are two types of agreement:
 - Syntactic interoperability: when systems can communicate and exchange data with a specific data format and communication protocols;
 - Semantic Interoperability: when there is a significant and accurate automatic interpretation of data exchanged to produce useful results (usually using a common reference model in this communication).
2. **Adoption Level:** when two or more cloud systems need to interoperate, a decision must be taken about which technology, standard or message format should be used:
 - Interoperability by design: when there is a standard documentation to develop interoperable products;
 - Post-facto interoperability: when, after releasing, a technology becomes dominant in the market. Remaining products that need to communicate with it need to adopt the same format already used in the dominant technology.

3. **Deployment Level:** this type of interoperability refers to how service integration occurs regarding the different cloud levels (IaaS, PaaS, SaaS). It can be:
 - Horizontal interoperability: when services work together at the same deployment level, e.g. SaaS-SaaS, PaaS-PaaS and IaaS-IaaS;
 - Vertical interoperability: when services work together in different deployment levels, e.g. PaaS-SaaS and SaaS-IaaS.
4. **Patterns of interactions between clouds:** this level refers to how interaction between cloud applications occur. It can be:
 - Synchronous interoperability: direct calls only for very specific real-time applications where the response time is critical;
 - Asynchronous interoperability: applications are loosely coupled so that applications do not wait for a response.

Besides these four levels of agreement identified by Petcu, we identified a fifth one, defined as follows:

5. **End-user level:** this level considers the end user of the interoperability solution, and it can be:
 - Provider-centric interoperability: when the interoperability strategy focuses on making the providers' systems interoperable (Zhang et al. 2013). Examples of this type of interoperability are solutions based on standards like Cloud Data Management Interface (CDMI), Open Cloud Computing Interface (OCCI), and Open Virtualization Format (OVF).
 - User-centric interoperability: when the focus is on the interoperability of applications, from the perspective of the users (Zhang et al. 2013). These solutions help users of different cloud services find a common denominator between the services (Zeginis et al. 2013). Examples of user-centric interoperability approaches (Zeginis et al. 2013) are mOSAIC (Rak et al. 2011) and Cloud4SOA¹⁵.

Another possible classification is according to the technology used to implement the interoperability solution (Petcu 2011). The most common implementation choices (Petcu 2011) are: Open APIs, Open protocols, Standards, Layers of abstraction, Semantic repositories, and Domain-Specific Languages. We do not describe these in detail because they are all well-known concepts, and their exploration is not in the scope of this work.

In this survey, we also found a category of cloud interoperability solutions based on MDE (Model-Driven Engineering) (Brunelière et al. 2010; Ferry et al. 2013; Sharma et al. 2011; Sharma and Sood 2011) and related technologies. These use platform-independent models and code generation for the creation of interoperable software. We also identified some SOA-based approaches (Mei et al. 2009; Papazoglou et al. 2007; Sharma et al. 2011), which are solutions that use SOA (Service-Oriented Architecture) technologies to allow different systems to interoperate regardless of where they are deployed. It is possible that some other categorization can raise as research and interoperability technologies advances.

5.4 RQ4: What are the requirements and possible usage scenarios for a cloud interoperability solution?

Petcu (2013) identified many requirements that should be implemented in solutions and consequently need to be satisfied by developers to build Inter-Cloud environments. She classifies the requirements in three groups: development, with requirements related to the development of services and applications, deployment, with requirements related to how applications can be deployed over multiple clouds, and execution, with requirements related to how applications execute and are monitored in multiple clouds. Additionally, she identifies a set of usage scenarios. These require solutions more focused in migration strategies between clouds to avoid lock-in and interface across multiple clouds to manage services from different clouds (brokerage service). Some of the usage scenarios include:

- optimizing costs and improving quality of services (QoS);
- reacting to changes in technologies and services offered by providers;
- adapting to rules and laws according to each country and enterprises;
- replicating applications or consuming services from different providers to ensure availability on the Web;
- avoiding dependence on specific providers;
- ensuring backups;
- acting as a broker mechanism; and
- combining different services offered by different providers to build a new improved service.

These usage scenarios and requirements reinforce the importance and necessity of developing cloud interoperability solutions. As technologies related to interoperable solutions advances, many other usage scenarios can raise and become relevant. Some scenarios that were not mentioned in surveyed studies are mobile cloud applications and cloudlets. As cloud computing in general advances, many other research fields and technologies can benefit from it. Among main advantages of cloud model is the combination of its services with spread computing potential as in mobile devices.

Benefits of interoperable solutions are many and directly aligned with above listed requirements and scenarios. (A full list can be seen in (Petcu 2013), although no detailed discussion is included.) Analyzing which solutions are aligned with each of those scenarios and requirements is out of the scope of this work.

5.5 RQ5: What solutions for cloud interoperability exist and what are their characteristics?

Cloud interoperability is an extensive subject. While many solutions in the literature exist, new ones are appearing as technology becomes better established. Despite the careful conduction of this literature survey, and the rigour in searching, extracting and analysing data, its results are limited to research papers we found. It is not always that all available papers are found. In practice, a research group always tries to find as much studies as possible, but after publishing the paper, the results and catalog may rapidly become obsolete. In an ideal scenario, it would be much more useful to keep a Wiki, like the one maintained by Cloud Standards¹⁶ and allow authors of new solutions to update the list and present their approaches to the community.

Despite that observation, in this work it would be impractical to describe all the solutions found. So we only included those that were more referenced in previous research

papers we found (Grozev (2014) and Petcu (2014) also present catalog of solutions). We tried to categorize the solutions according to the observations discussed in RQ3.

Standardization: according to Armbrust et al. (2009), one way to solve interoperability is through standardization. If enough cloud service/providers successfully adhere to the standards, applications could then be more easily ported and interoperability is facilitated¹⁷. Standardization efforts include: Distributed Management Task Force (DMTF), Cloud Standards Coordination (ETSI), Advancing Storage and Information Technology (SNIA), CloudAudit, Cloud Security Alliance (CSA), TC CLOUD, Object Management Group (OMG) and IEEE. However, although standardization is important, its success depends on adoption. This means that cloud providers must all agree and follow a standard. As discussed in Section 5.3, while public and governmental cloud providers share a common interest in collaboration, most independent commercial vendors have no interest in allowing their customers to easily migrate away from them. For this reason, in determined contexts, standards may not be the best solution. Adopting a standard can also lead the consumer to become locked-in in selected one. There are so many standards that even choosing one may become a challenge. Thus, the effort required to choose one standard should not be disregarded. Also, there are no guarantees that the chosen standard will become a success, therefore increasing the risk that needs to be managed by cloud adopters/providers.

Abstraction Layers: some initiatives are trying to achieve cloud interoperability through abstraction layers. This consists in abstracting some common feature of each cloud provider and offering a high level layer to make the development more platform-dependent. Projects RESERVOIR, SLA@SOI and CSAL¹⁸ are examples of initiatives following this idea.

Open protocols: this strategy is more focused on a communication protocol between the platforms. Examples of open cloud protocols are OCCI and DeltaCloud. Examples of proprietary protocols are Amazon EC2 or VMware vCloud.

Open APIs: an API is a set of tools and protocols for building applications. In cloud computing, there are APIs that target interoperability, such as Jclouds, GoGrid, OpenStack, Simple Cloud and CloudLoop.

Model-driven approaches: despite the advances of software development techniques, concerns about reuse, productivity, maintenance, documentation, validation, optimization, portability and interoperability are still under discussion. MDE aims at solving some of these issues (Kleppe et al. 2003), shifting the focus of modern development methodologies from implementation to conceptual modeling. Thus, models are now first-class citizens, and transformation mechanisms are used to generate code from them, reducing development effort, increasing portability and interoperability of software systems (France and Rumpe 2007).

Sharma and Sood (2011) present a model-driven approach for interoperability in SaaS (Software-as-a-Service). They define models at different abstraction levels, based on separation of concerns, particularly Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM) of the MDA¹⁹. Each level can be composed by one or more models to specify the structural, functional and behavioral features of a system. For PIMs, a formal definition of the operations offered by the service is used, which can be accessed through an interface that must later be composed with other services to build a complete system. Business rules are specified through

the declaration of restrictions, pre-conditions and post-conditions, and invariants in Object Constraint Language (OCL). After some transformations the PIM is converted to a Web-Service Description Language (WSDL) PSM, which consists of a specific model description language for web services. The final step is the transformation of the WSDL PSM into WSDL specifications.

The ModelDriven Approach for the design and execution of applications on multiple Clouds (MODACLOUDS) aims at supporting system developers and operators in exploiting multiple clouds and in migrating their applications from cloud to cloud, as needed (Ardagna et al. 2012). MODACLOUDS main objective is to provide methods, a decision support system, an open source IDE and runtime environment for the high-level design, early prototyping, semi-automatic code generation, and automatic deployment of applications on multiple clouds. It also helps administrators to monitor the services and measure their quality. By the time this paper was written, the framework was still under development²⁰, but it intends to provide one of the first implementation of OASIS TOSCA emerging standard. MODACLOUDS also includes the development of CloudML, a domain-specific modeling language and runtime environment that facilitate the specification of cloud application provisioning, deployment, and adaptation.

SOA-based solutions: there are some strategies based on Service-Oriented Architecture for cloud interoperability. For example, Cloud4SOA presents a broker-based architecture to address interoperability in PaaS model (Zeginis et al. 2013). The system enables management, monitoring and migration through the interconnection between different PaaS providers that share the same technology. The main idea behind this solution is to facilitate developers during search, deployment and governance of PaaS applications to better meet their demands. Semantic algorithms implemented in Cloud4SOA allow management of data and applications. They enable portability between PaaS that use same technologies but have different APIs and data models (Zeginis et al. 2013). In practice, Cloud4SOA is more focused on portability concepts.

mOSAIC is a project aiming at offering a simple and clear access to heterogeneous resources from multiple clouds. It is a vertical interoperability strategy targeting hybrid clouds (Petcu 2011). Under a high-level perspective, mOSAIC is composed of an API and a PaaS that allow to deploy and manage applications. It offers an abstraction layer and applies web semantic to the implementation its interfaces and components (Petcu 2014). In short, this solution is a PaaS that allows development, configuration and management of applications running on IaaS (Petcu 2014).

5.6 RQ6: are there any evaluation models for cloud interoperability solutions?

We found a previously published work on evaluation models for interoperability (Rezaei et al. 2014) which describes existing models and offers a comparative analysis highlighting major similarities and differences between them. However, the authors conclude that these existing interoperability evaluation models are not applicable to cloud computing. We too, could not find any evaluation model for cloud interoperability in the selected studies. This may suggest an open challenge to be addressed in the future. Such an evaluation model for cloud interoperability could be a good contribution to this area of research. We believe Rezaei et al. (2014) work can serve as basis to propose some evaluation model. Due heterogeneity of interoperable cloud services, their usage scenarios

and requirements, we believe only one model for evaluating is not enough. It is needed to propose many models, each one aligned with the context of interoperability strategy.

6 Discussion of the results

It may be the case that some evidence was left out of this research. Databases such as Scopus, ACM or IEEE are focused on academic results, and may not reflect industry practices or technologies. Industry initiatives often experience real advancements in the field without necessarily bothering about publishing papers or reports. The following paragraphs highlight the major findings of this study.

Definitions In this work, we found that many definitions have been proposed for cloud interoperability, but none is yet established and well accepted by the community. Indeed, there is no consensus about its scope, as they focus on different perspectives of interoperability. We observed that interoperability is often associated with many of the advantages of the cloud model. Therefore, we need a more precise and specific definition, aligned to how interoperability has been implemented in practice. Our definition embraces the idea of multiple cloud providers to allow their combination in building improved solutions, taking advantage of each provider offerings.

Concepts We believe that most of the identified concepts emerged from four key ones: inter-cloud, cloud federation, multi-cloud and cloud broker. These four key concepts reflect the evolution of interoperability, forming the basis for all the others (and eventually new one) as the list of 25 concepts presented in Petcu's catalog (Petcu 2014).

Classification Although some solid concepts and strategies have been implemented in practice, there is no well-established categorization. We presented the categorizations and their perspectives found in the literature. We believe it would be useful to (i) define such classifications more rigorously and (ii) offer a mapping between the classification and the solution.

Requirements and usage scenarios Many requirements for interoperable solutions were identified by Petcu (2013). Although many of them have been implemented in interoperable solutions in practice, we believe this topic needs to be further explored, as she does not inform about the origin of those requirements nor does she discuss the methodology followed to perform the study. Mapping requirements, implementations and possible solutions would be valuable for the software engineering community. A more solid empirical work about requirements for cloud and cloud interoperability is needed.

Solutions We found many research papers related to this, some of them cataloguing solutions for cloud interoperability. Some of these papers, discussed in Section 8, highlight the most referenced solutions. While there are many solutions in the literature, there is no evidence that these solutions are being highly adopted in industry.

In general, industry is still in the adoption phase, as suggested by many of the works studied here. Given that adoption and consequent migration is still a challenge, interoperability has not yet received the deserved attention, as it is a concern that will be felt

further down the road. Nonetheless, new solutions, proposals, case studies and experience reports should be kept on the research agenda. The more case studies available, the more confident researchers and practitioners will become regarding the benefits of cloud interoperability, and the validity of the research.

Evaluation models As solutions are still under investigation, it is not surprising that evaluation models do not yet exist. Some solutions were evaluated in one way or other (e.g., evaluation, communication, data exchange), but a systematic evaluation approach, or simply a specific evaluation model, for interoperable cloud solutions was not found.

All the research topics highlighted here must be further explored and research needs to be stimulated to promote the advancement of the field. At the time this paper was written, the choice of one solution (the consumer becomes locked-in in the provider/solution) or even of a standard (the consumer becomes locked-in in the standard) may lead to lock-in. Many solutions and standards diverge from each other, while others are characterized as scientific proposals aiming at solving the problem. However, in their majority, they are not straightforward and stable enough to be accepted by industry. In conclusion, the lock-in problem remains.

7 Directions for future research

As we presented throughout this paper, interoperability in cloud computing constitutes a research area that needs to be more explored. This paper presented definitions, concepts, approaches and a further discussion about our findings. Research on Cloud Computing is increasingly growing and interoperability is one of main issues related to the field. Additionally, the combined use of multiple cloud technologies is becoming increasingly common. Solutions for cloud interoperability found in the literature cannot yet be regarded as established. Following list presents some open issues and directions for future research, in no way we believe the results of our observations are the only and definitive list. The community is welcome to continue the discussion and to increase this list as further research takes place.

- **Approaches to avoid Lock-In:** Lock-in is still a big problem in cloud computing in general. Approaches for this problem need to be more explored and presented. This problem and dynamic nature of cloud technologies lead to solutions to be non-interoperable. Even if we choose a solution/pattern trying to avoid lock-in, we become locked-into that solution. There is no established standard, in practice many enterprises have a conservative profile and adopt technologies only when they become established and highly adopted in the market.
- **Interoperability evolution models:** as we presented in question **RQ6**, existing evaluation models are not applicable to cloud computing. We believe one unique interoperability evaluation model that fits in Cloud computing in general is not possible. As we presented in **RQ6**, interoperability can be viewed in many perspectives. Thus, an interoperability evaluation model according each one is needed. Future research in this direction should consider each context of interoperability, including different service and deployment models.
- **Case studies on Cloud interoperability:** Organizations are normally afraid to adopt new technologies until relevant success cases are presented. In this sense, to reinforce the benefits of the cloud interoperability and presented solutions, to show

the importance of this paradigm, and its potential to revolutionize the market, it would be interesting to have a large number of case studies. The literature already has many examples of such efforts. Community should continue to publish such type of results, and case studies and experience reports should always be on the agenda of software engineering researchers. The more case studies available, the more confident researchers and practitioners will become regarding the benefits of the cloud, and the validity of the research.

- **Comparative study of the existing solutions:** we performed this study as a first effort to define the body of knowledge about cloud interoperability. Among our objectives was to identify existing solutions and their characteristics. However, a comparative study between solutions found was not in the scope of this research. We believe an additional study comparing the solutions, their usage scenarios, and their benefits would be very useful for the community.
- **More solutions for cloud interoperability:** as we presented before, solutions for cloud interoperability are not established yet. The community is welcome to increase the body of knowledge about the field and present more solutions that will certainly advance the state-of-the-art of the field.
- **Approaches for combining cloud services:** many solutions for combining services in context of software-oriented architecture (SOA) can be found in the literature (da Silva and Lucrecio 2012). However the term “services” in context of Cloud Computing refers to different concept than SOA (da Silva and Lucrecio 2012). Thus, approaches for combining cloud services and providing multi-cloud and federated solutions should be better explored. According North Bridge Survey 2015²¹, 71% of survey’s answers use more than one cloud vendor. This indicates that in near future multi-cloud applications will be the rule, not the exception.
- **A process for building multi-cloud applications:** most solutions are focused in technology itself. However, we believe a conceptual outline about building multi-cloud solutions would be a very important contribution for community. A process for building multi-cloud applications could guide developers to conceive this kind of applications, helping them to achieve heterogeneous and improved solutions that take advantage of features from different providers.
- **Definition of requirements and usage scenarios for interoperable solutions:** as we presented before, requirements and usage scenarios of interoperable solutions should be better explored. Mapping requirements, implementations and possible solutions is not in the scope of this paper and would be valuable for the software engineering community. A more solid empirical work about requirements for cloud and cloud interoperability is still needed.
- **Investigate how cloud interoperability field can benefit from reuse techniques:** reuse techniques as frameworks, crosscutting framework (Coelho et al. 2006), software product lines (Clements and Northrop 2002), model-driven development (France and Rumpe 2007), among others, have presented many advancements in development of systems in general. Research community needs to investigate how these techniques can benefit development of multi-cloud applications, identifying main issues regarding each one when applied to this domain.
- **Data interoperability:** in practice an interoperable solution or a solution that provides interoperability needs to deal with heterogeneous data formats. Community

needs to better explore this field identifying best strategies and solutions for this issue. We believe that, in terms of data, there should not exist any standard data formats, but a way that each cloud service composition define their terms of communication and format.

8 Related work

Petcu (2013) presents some gaps to be addressed in near future, proposes requirements for interoperability solutions, highlights the technological barriers and some well-known solutions for Multi-Cloud environments. The author has been studying this topic for many years and many presented concepts seem to be empirical results. We analyzed the requirements presented and agree that interoperable solutions should implement them, mainly to (i) avoid lock-in, (ii) keep providers' specificity for strategic reasons and (iii) facilitate development of solutions that take advantage of the best features and services from adopted providers. Despite that, the author did not present the origin of these requirements or in which solutions they were implemented. The paper does not justify the solutions classification provided, the source of these solutions or the methodology used for cataloging.

Rashidi et al. (2013) performed a literature survey on interoperability. The authors do not state the questions they wanted to answer, but their work offers some definitions for cloud interoperability, some solutions and some challenges regarding the implementation of components involved in the composition of cloud services, such as technological tools and decision making. They do not consider all types of interoperability, since the presented solutions and concepts are focused on interaction between IaaS.

Zhang et al. (2013) state that lock-in has seriously limited the flexibility of what cloud end users would like to process, when it comes to deploying applications over different infrastructures in different geographic locations, or to migrate a service from one provider's cloud to another. The authors performed a survey on the state-of-the-art on cloud interoperability, investigating the existing efforts on the subject. They also proposed some challenges that must be overcome to advance further the research topic. But, like Rashidi et al. (2013), their work is also focused on IaaS interoperability, offering a map on the coverage of some cloud standards.

Grozev and Buyya (2014) performed a survey on inter-cloud architectures and application brokering. They present the architectural taxonomy of Inter-Cloud computing and brokering mechanisms, and catalogue many solutions for Inter-Cloud, but mainly focusing on broker-based strategies. The authors compile many concepts regarding interoperability, more specifically Inter-Cloud, Multi-Cloud and Cloud Federation. They focused on how Inter-Cloud projects facilitate the brokering of different applications across multiple clouds. They state that cloud interoperability and related technologies form a novel area of intensive research and development whose body of knowledge has not been well established yet. Therefore, more effort is needed to improve existing concepts, solutions and projects.

Rezaei et al. (2014) performed a Systematic Literature Review (Kitchenham et al. 2009) on interoperability focusing on interoperability evaluation models, but did not focus on cloud interoperability. They searched for evaluation models for interoperability in general and included cloud interoperability in their results. The authors also offer a comparative analysis between their findings to determine similarities and differences between the

interoperability evaluation models found. As we mentioned before, they did not find any cloud interoperability model.

While Petcu (2013) focuses on Multi-Cloud environments and requirements, Rashidi et al. (2013) and Zhang et al. (2013) focused only on IaaS solutions. Grozev and Buyya (2014) focused on strategies related to brokering mechanisms and Rezaei et al. (2014) spent their efforts on interoperability evaluation models. Although we consider all mentioned papers relevant, we still thought it would be important to perform a broader search for further works on the field. We had several research questions that none of these studies answered individually. Furthermore, we decided to cover not only one type of cloud interoperability as previous works did, but to compile all relevant evidence and concepts. As the body of knowledge is still being established, we aimed at compiling relevant concepts about cloud interoperability through this work.

9 Threats to validity

The main threat to validity of our work is the summarization process. We defined a protocol to guide our review work, but our main objective in performing this protocol was only increasing rigor on conducting the review. It was not our intention to perform a Systematic Literature Review as proposed by Kitchenham, B. (2004). In our view, a systematic literature review is still more rigorous and should be applied in specific contexts for summarizing specific evidences about a subject.

We present our own position about the findings in Section 6. As it is ours, it can be biased. We are from software engineering field and tried viewing the technologies and concepts considering how we can contribute to our research field. Thus, that discussion can bring biased perspective of the field. A systematic literature review perhaps can reduce the research bias, but in this first moment, it was not our objective to perform one. We wanted to identify research field to enable further research in other opportunities, this include conducting a Systematic Literature Review on a specific subject when appropriate.

10 Conclusions

As we presented before, in this work we aimed at identifying how interoperability problem has been addressed in the literature and present an updated view of the subject by answering six research questions on the field. We also compiled the main concepts presented in previous reviews in a new one that can better help researchers and developers to understand the subject and show a high-level view that is not possible by only seen in an individual study.

As additional contribution of this research we present a new definition for cloud interoperability more related to software development perspective; a discussion that exposes our position on the findings and 10 (ten) research directions for future research. We performed this work as an attempt to better define the research topic to enable further research on the subject. In no way we believe the results of this work are unique and definitive, and the community is welcome to continue the discussion.

In the near future we plan to investigate five main challenges: (i) provide an open wiki to catalog as many solutions as possible, in all different contexts of cloud interoperability; (ii) perform a mapping of solutions being proposed by industry and consequently their applicability context; (iii) perform a relation between requirements and solutions; (iv) perform

more systematic study and, since our research group is highly specialized in model-driven approaches, (v) study how MDE can be used to leverage the level of abstraction when developing applications for the cloud, providing increased interoperability.

Another interesting point of view is that only the advantages of the cloud interoperability are described in papers we found and in this one. In a first moment, we believe there are many disadvantages involving development and usage of interoperable applications. Empirically, we can cite lots of them. However, we think would be better if we conduct an additional and more rigorous study regarding this point. This can be very interesting for the community and can give researchers even more subsidies for better decision making on the process of developing cloud interoperable applications.

Endnotes

¹ <http://www.computer.org/>

² <http://www.scopus.com/>

³ <http://www.ieeexplore.ieee.org/>

⁴ <http://www.portal.acm.org/>

⁵ <http://scholar.google.com/>

⁶ Although technical reports do not usually go through a rigorous scientific review process, we decided to include them to avoid missing relevant studies such the one from Armbrust et al. (2009), which is simply the most cited work in the field of cloud computing.

⁷ National Institute of Standards and Technology - www.nist.gov

⁸ <http://dl.acm.org/>

⁹ <http://www.scopus.com/>

¹⁰ <http://ieeexplore.ieee.org/>

¹¹ <http://cloudusecases.org/>

¹² Each layer means a type of service: IaaS (Infrastructure-as-a-Service), PaaS (Platform-as-a-Service) and SaaS (Software-as-a-Service); these are well-known cloud computing terms (Liu et al. 2011) that refer to the different levels of resources offered in a cloud.

¹³ http://www.informatizacia.sk/extdok-european_interoperability_framework/9319c

¹⁴ Each developer or software company may want build its own interoperability strategy more aligned with their business rules and choose not to follow a standard.

¹⁵ <http://www.modaclouds.eu/software/cloud4soa/>

¹⁶ http://cloud-standards.org/wiki/index.php?title=Main_Page

¹⁷ Some enterprises and research groups maintains a catalog of latest cloud standard activities. For more information see http://cloud-standards.org/wiki/index.php?title=Main_Page

¹⁸ <http://www.cs.virginia.edu/~humphrey/papers/CSAL.pdf>

¹⁹ <http://www.omg.org/mda/>

²⁰ MODACLOUD's research papers list are available at <http://www.modaclouds.eu/publications/papers/>

²¹ <http://www.northbridge.com/2015-future-cloud-computing-survey>

Acknowledgments

We would like to thank FAPESP (processes 2012/24487-3 and 2012/04549-4) and Brazil-Europe Erasmus Mundus project (process BM13DM0002) for partially funding this research. We acknowledge the help from UFSCar's LAPES lab (Laboratory for software engineering research at Federal University of São Carlos), more specifically Sandra Fabbri, for the important contributions to this research. We also would like to thank João Araújo from Universidade NOVA de Lisboa and Journal of Software Engineering Research and Development reviewers for important reviewing work and commenting this paper.

Authors' contributions

EN designed and conducted the study, wrote first version of the paper and addressed reviewer's issues; AM designed the study, reviewed all sections and improved summarization; DL designed and reviewed the study, improved summarization and addressed reviewer's issues; VG reviewed and improved summarization; and RF designed the study, reviewed, improved summarization and addressed reviewer's issues. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

¹ICMC-USP, São Carlos, Brasil. ²NOVA LINES FCT-UNL, Almada, Portugal. ³DC-UFSCar, São Carlos, Brazil. ⁴CIN-UFPE, Recife, Brazil.

Received: 6 May 2016 Accepted: 7 November 2016

Published online: 07 December 2016

References

- Alkhatib H, Faraboschi P, Frachtenberg E, Kasahara H, Lange D, Laplante P, Merchant A, Milojicic D, Schwan K (2014) IEEE CS 2022 Report Technical report. <https://www.computer.org/cms/Computer.org/ComputingNow/2022Report.pdf>
- Ardagna D, Di Nitto E, Mohagheghi P, Mosser S, Ballagny C, D'Andria F, Casale G, Matthews P, Nechifor CS, Petcu D, Gericke A, Sheridan C (2012) MODAClouds: A model-driven approach for the design and execution of applications on multiple Clouds. In: 2012 4th International Workshop on Modeling in Software Engineering (MISE). IEEE. pp 50–56. doi:10.1109/MISE.2012.6226014
- Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M (2009) Above the Clouds: A Berkeley View of Cloud Computing, Vol. 28. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>
- Brunelière H, Cabot J, Jouault F (2010) Combining Model-Driven Engineering and Cloud Computing. http://hal.archives-ouvertes.fr/docs/00/53/91/68/PDF/CombiningMDE-CloudComputing_2010.pdf
- Chen Y, Li X, Chen F (2011) Overview and Analysis of Cloud Computing Research and Application:1–4. doi:10.1109/ICEBEG.2011.5881819
- Clements P, Northrop L (2002) Software Product Lines. Addison-Wesley. <http://dl.acm.org/citation.cfm?id=501065>, <http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=30731>, <https://scholar.google.com.br/scholar?hl=pt-BR&q=Software+product+lines+clements+paul&btnG=&lr=>
- Coelho R, Alves V, Kulesza U, Costa Neto A, Garcia A, Staa AV, Lucena C, Borba P (2006) A Verification Approach for Crosscutting Features Based on Extension Join Points. https://scholar.google.com.br/scholar?cluster=3366721263192567151&hl=pt-BR&as_sdt=0,5
- Cruzes DS, Dyba T (2011) Recommended Steps for Thematic Synthesis in Software Engineering. In: 2011 International Symposium on Empirical Software Engineering and Measurement. IEEE. pp 275–284. doi:10.1109/ESEM.2011.36
- da Silva EAN, Lucredio D (2012) Software Engineering for the Cloud: A Research Roadmap. IEEE, Brazil, UFSCar. doi:10.1109/SBES.2012.12
- da Silva EAN, Fortes RPM, Lucrédio D (2013) A Model-driven Approach for Promoting Cloud PaaS Portability. CASCON '13. IBM Corp., Riverton
- Delgado JCM (2014) A Framework for Cloud Interoperability Based on Compliance and Conformance. *Computer Communications and Networks*. Springer. doi:10.1007/978-1-4471-6452-4_9
- Dowell S, Barreto A, Michael JB, Shing MT (2011) Cloud to cloud interoperability. In: System of Systems Engineering (SoSE), 2011 6th International Conference on. IEEE. pp 258–263. doi:10.1109/SYSE.2011.5966607
- Dyba T, Dingsoyr T, Hanssen GK (2007) Applying Systematic Reviews to Diverse Study Types: An Experience Report. doi:10.1109/ESEM.2007.59
- Esparza J, Escof M (2011) Towards the Next Generation of Model Driven Cloud Platforms:494–500. https://scholar.google.com.br/scholar?cluster=137254110196252998&hl=pt-BR&as_sdt=0,5
- Ferry N, Rossini A, Chauvel F, Morin B, Solberg A (2013) Towards Model-driven Provisioning, Deployment, Monitoring, and Adaptation of Multi-cloud Systems. In: 2013 IEEE Sixth International Conference on Cloud Computing. pp 887–894. doi:10.1109/CLOUD.2013.133
- France R, Rumpe B (2007) Model-driven Development of Complex Software: A Research Roadmap. FOSE '07. IEEE Computer Society, doi:10.1109/FOSE.2007.14
- Grozev N, Buyya R (2014) Inter-Cloud Architectures and Application Brokering: Taxonomy and Survey, Vol. 44. Wiley Online Library. doi:10.1002/spe.2168
- Harsh P, Dudouet F, Cascella RG, Jegou Y, Morin C (2012) Using open standards for interoperability issues, solutions, and challenges facing cloud computing. In: 2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualization management (svm). pp 435–440
- Hogan M, Liu F, Sokol A, Tong J (2011) Nist Cloud Computing Standards Roadmap. https://scholar.google.com.br/scholar?cluster=328777146437379106&hl=pt-BR&as_sdt=0,5
- IEEEGLOSSARY (2013). http://www.ieee.org/education_careers/education/standards/standards_glossary.html. Accessed 20 Apr 2016
- Kitchenham B (2004) Procedures for performing systematic reviews. Technical report, Keele University Technical Report TR/SE-0401 ISSN:1353-7776
- Kitchenham B, Brereton OP, Budgen D, Turner M, Bailey J, Linkman S (2009) Systematic Literature Reviews in Software Engineering: A Systematic Literature Review, Vol. 51. doi:10.1016/j.infsof.2008.09.009
- Kleppe A, Jos W, Wim B (2003) MDA Explained, The Model-Driven Architecture: Practice and Promise. Addison-Wesley

- Kostoska M, Gusev M, Ristov S, Kiroski K (2012) Cloud Computing Interoperability Approaches-Possibilities and Challenges:30–34. https://scholar.google.com.br/scholar?cluster=8723810457458895632&hl=pt-BR&as_sdt=0,5
- Koussouris S, Lampathaki F, Mouzakitis S, Charalabidis Y, Psarras J (2011) Digging Into the Real-life Enterprise Interoperability Areas Definition and Overview of the Main Research Areas:19–22. https://scholar.google.com.br/scholar?cluster=6143984506274935631&hl=pt-BR&as_sdt=0,5
- Liu F, Tong J, Mao J, Bohn RB, Messina J, Badger L, Leaf D (2011) NIST Cloud Computing Reference Architecture. In: 2011 IEEE World Congress on Services. pp 594–596. doi:10.1109/SERVICES.2011.105
- Loutas N, Kamateri E, Tarabanis K (2011) A Semantic Interoperability Framework for Cloud Platform as a Service:280–287. doi:10.1109/CloudCom.2011.45
- Mei L, Zhang Z, Chan WK (2009) More Tales of Clouds: Software Engineering Research Issues from the Cloud Application Perspective 1:525–530. doi:10.1109/COMPASAC.2009.76
- Mell P, Grance T (2010) The NIST Definition of Cloud Computing 53:50. <http://dl.acm.org/citation.cfm?id=2206223>
- Papazoglou MP, Traverso P, Dustdar S, Leymann F (2007) Service-Oriented Computing: State of the Art and Research Challenges 40:38–45. doi:10.1109/MC.2007.400
- Petcu D (2011) Portability and interoperability between clouds: challenges and case study. In: Abramowicz W, Llorente IM, Surridge M, Zisman A, Vayssi re J (eds). Towards a Service-Based Internet: 4th European Conference, ServiceWave 2011, Poznan, Poland, October 26–28, 2011. Proceedings. Springer, Berlin, Heidelberg. pp 62–74. doi:10.1007/978-3-642-24755-2_6
- Petcu, D (2013) Multi-Cloud: Expectations and Current Approaches. In: Proceedings of the 2013 International Workshop on Multi-cloud Applications and Federated Clouds. ACM, New York. pp 1–6. doi:10.1145/2462326.2462328
- Petcu D (2014) Consuming Resources and Services from Multiple Clouds 12:321–345. doi:10.1007/s10723-013-9290-3
- Petcu D, Macariu G, Panica S, Cr  ciun C (2013) Portable Cloud applications—From Theory to Practice 29:1417–1430. doi:10.1016/j.future.2012.01.009
- Rak M, Venticinque S, Mahr T, Echevarria G, Esnal G (2011) Cloud Application Monitoring: The mOSAIC Approach:758–763. doi:10.1109/CloudCom.2011.117
- Rashidi B, Sharifi M, Jafari T (2013) A Survey on Interoperability in the Cloud Computing Environments 5:17–23. doi:10.5815/ijmecs.2013.06.03
- Rezaei R, Chiew TK, Lee SP, Aliee ZS (2014) Interoperability Evaluation Models: A Systematic Review 65:1–23. doi:10.1016/j.compind.2013.09.001
- Rimal BP, Choi E, Lumb I (2009) A Taxonomy and Survey of Cloud Computing Systems:44–51. doi:10.1109/NCM.2009.218
- Sharma R, Sood M (2011) A Model Driven Approach to Cloud SaaS Interoperability 30:1–8. https://scholar.google.com.br/scholar?cluster=10634166007824275691&hl=pt-BR&as_sdt=0,5
- Sharma R, Sood M, Sharma D (2011) Modeling Cloud SaaS with SOA and MDA. In: Abraham A, Lloret Mauri J, Buford JF, Suzuki J, Thampi SM (eds). Advances in Computing and Communications: First International Conference, ACC 2011, Kochi, India, July 22–24, 2011. Proceedings, Part I. Springer Berlin Heidelberg, Berlin. pp 511–518. ISBN:978-3-642-22709-7, doi:10.1007/978-3-642-22709-7_50
- Vaquero LM, Rodero-Merino L, Caceres J, Lindner M (2008) A Break in the Clouds 39:50. doi:10.1145/1496091.1496100
- Zeginis D, D'Andria F, Bocconi S, Gorrongoitia Cruz J, Collell Martin O, Gouvas P, Ledakis G, Tarabanis KA (2013) A User-centric multi-PaaS Application Management Solution for Hybrid multi-Cloud Scenarios 14. https://scholar.google.com.br/scholar?cluster=15905955653602311204&hl=pt-BR&as_sdt=0,5
- Zhang Z, Wu C, Cheung DWL (2013) A Survey on Cloud Interoperability: Taxonomies, Standards, and Practice. SIGMETRICS Perform. Eval. Rev 40(4):13–22. doi:10.1145/2479942.2479945

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
