A Systematic Review of Cloud Lock-in Solutions

Gabriel Costa Silva, Louis M. Rose, Radu Calinescu
Department of Computer Science, University of York
Deramore Lane, York YO10 5GH, UK
{gcs507, louis.rose, radu.calinescu}@york.ac.uk

***Abstract*—The heterogeneity of cloud semantics, technology and interfaces limits application and platform portability and interoperability, and can easily lead to vendor lock-in. We identify, analyse and classify existing solutions to cloud vendor lock-in, and highlight unresolved challenges. Our survey is based on a systematic review of 721 primary studies that describe the state-of-the-art in managing cloud lock-in, portability and interoperability. 78 of these primary studies were selected and used for a thorough analysis of cloud standards, commercial products and academic work related to cloud lock-in. Our review shows that most solutions proposed so far are platforms, APIs or architectures addressing infrastructure-as-a-service (IaaS) interoperability. From our review, we identify a need for: (i) exploiting established solutions from areas that are closely related to cloud computing; (ii) increasing empirical evidence to raise confidence in existing solutions; and (iii) addressing the socio-technical and business challenges related to cloud lock-in.**

Keywords— cloud lock-in; portability; interoperability; review

I. INTRODUCTION

Vendor, lock-in, or *vendor dependency*, is widely recognized as one of the greatest obstacles to cloud adoption [1]. Cloud users are often locked-in to a specific cloud provider due to the significant differences in the semantics [2], technologies [3], and interfaces [4] adopted by different providers. These differences hinder cloud *portability* and *interoperability* [5]. Although the definition of these concepts may vary over service models [6], [7], we consider portability as the ability to move assets across clouds [8], and interoperability as the ability of cloud platforms, from different cloud providers, to seamlessly exchange assets [1], [9].

As cloud portability and interoperability are critical to cloud adoption, many approaches to enabling and enhancing them have been proposed. Practitioners, researchers and policy makers from industry [10], academia [11], standards organisations [12] and governments [3] have all contributed to this significant effort. The large number of resulting solutions gave rise to a new challenge, i.e. identifying and understanding the contributions and limitations of each solution. Without this, it is hard to identify areas in which further research would be beneficial. Yet, existing surveys of cloud lock-in solutions pay only limited attention to key questions such as: How can solutions be classified? How are they related or complementary? Our study aims to address this limitation, by presenting a systematic review carried out to identify, analyse and classify existing solutions to vendor lock-in, portability and interoperability problems in the cloud.

We start with a brief overview of the main concepts from this area of cloud computing (Section II). The methodology used by our systematic review is then described (Section III), followed by the key contributions of our paper: (i) a systematic review and bibliography of primary sources for a broad range of approaches that address cloud lock-in (Section IV-A); (ii) a classification of cloud lock-in solutions, and a discussion of areas requiring further research (Section IV-B and IV-C); (iii) a critical analysis of the progress made towards addressing vendor lock-in for cloud computing (Section V); and (iv) a review of existing surveys (Section VI) on cloud lock-in. The paper concludes with an analysis of the threats to the validity of our review (Section VII), and a discussion of potential directions for further work (Section VIII).

II. BACKGROUND

Irrespective of their underpinning cloud service model – *infrastructure-as-a-service* (IaaS), *platform-as-a-service* (PaaS) or *software-as-a-service* (SaaS) – cloud deployments are affected by the semantics [2], technologies [3] and interfaces [4] adopted by the providers of the cloud services.

Cloud semantics refers to the description of a cloud service by its provider [13]. As an example, the Amazon S3 storage service (aws.amazon.com/s3/) is described in terms of *buckets* and *availability zones*, where a bucket is the storage root element, and an availability zone represents o datacentre in one region. To store a file, Amazon S3 users must create a bucket in one availability zone. In contrast, the Dropbox storage service (www.dropbox.com) is specified using concepts similar to those used to describe a traditional file system. Dropbox users cannot identify the region in which their files are stored.

Cloud technologies comprise the middleware and applications used to support a cloud service. In the context of the PaaS service model, for instance, this technology refers to the types of application server(s) and programming language(s) supported by the provider [14]. Thus, a PHP application that can be deployed on Amazon Beanstalk (aws.amazon.com/ elasticbeanstalk) will not work when deployed on a cloud that lacks native PHP support, such as Heroku (www.heroku.com).

Cloud interfaces are the common APIs that provide programmatic access to the services offered by a cloud vendor. These APIs expose the semantics and technologies used by a provider, and play a key role in providing access to the service management functionality. As a result, their heterogeneity across different clouds is deemed a major barrier to cloud portability and interoperability [4], [15], [16]. Even API

This work was funded in part by CNPq - Brazil and by UK EPSRC grant EP/H042644/1

components that implement simple actions, such as creating a virtual machine (VM), vary considerably across providers.

In summary, applications deployed on a cloud become locked in to the cloud provider, due to the way in which the application uses the semantics, technologies and APIs of the provider [17]. Vendor lock-in happens even between cloud providers offering the same service model [18]. These differences render the "utility computing" view of cloud unrealistic. In a public utility such as electricity a device connected to a power supply works unmodified, regardless of the source of electricity (e.g., hydro, wind, or solar power) [5].

Cloud portability, or the ability to migrate a cloud-deployed asset to a different provider [19], is a direct benefit of overcoming vendor lock-in. Given the different characteristics of each cloud service model, the concept of portability depends on the service model adopted. According to Govindarajan & Lakshmanan [6], IaaS portability is the migration of virtual machines, PaaS portability is the migration of code and data, and SaaS portability is the migration of data and content. Portability is desirable for mitigating cloud outages [20] and for pursuing new business opportunities [1] (e.g., better price).

Cloud interoperability, according to Petcu [7] is "a property referring to the ability of diverse systems and organizations to work together (inter-operate)". The interest in it is largely due to the demand for *hybrid clouds*, in which the business-critical or sensitive data and applications are kept on a private cloud, while the other assets are deployed on a public cloud [21]. Other common scenarios that require interoperability include *accident recovery*, in which a secondary cloud can be used during an outage of the main provider; and *bursting*, in which different clouds cooperate to cope with excessive demand [22].

Cloud standards are regularly proposed as a way to mitigate vendor lock-in and achieve portability and interoperability in cloud computing [6]. However, many providers are concerned about the loss of customers that may come with standardization, and do not regard this solution favourably [7]. Since a partially adopted standard would represent a poor solution [23], numerous other approaches to addressing cloud lock-in have been proposed in recent years, as explained in the remainder of our review.

III. Systematic Review Methodology

We adopted the guidelines for systematic reviews in Software Engineering proposed by Kitchenham & Charters [24] and by Biolchini *et al.* [25], and ensured that the study was driven by a set of exploratory research questions, grouped into three categories:

1. *Solution identification, analysis and classification*
   a. What are the existing solutions addressing vendor lock-in, portability and/or interoperability in cloud?
   b. Which service model are these solutions targeting?
   c. What is the maturity level of current solutions?
   d. How were the solutions evaluated?
   e. How are the solutions related to industrial use?
2. *Identification and analysis of research areas*
   a. Which venues are the solutions published in?
   b. Who are the authors of these solutions?
   c. How do these solution relate to other work from the same venues?
3. *Gap analysis*
   a. What Computer Science areas do the solutions focus on?
   b. What issues are not addressed by the existing solutions?

The first set of questions was used to build a thorough understanding of the work carried out to solve known cloud lock-in, portability and interoperability problems. The second question set was used to assemble a detailed map of current research in the area. Finally, the last set of questions underpinned the identification of gaps in the current solutions.

To ensure that the study was as comprehensive as possible, we based it on a wide range of sources obtained from: (i) Six widely used digital libraries, comprising four libraries of high relevance to Computer Science – IEEE, ACM, SpringerLink and Science@Direct, and two libraries with comprehensive databases – Scopus and ISI Web of Science; (ii) Web searches, performed using the Google search engine; (iii) References from the material in (i); and (iv) The personal knowledge of the authors and their academic and industrial collaborators. The searches in (i)-(ii) were carried out using query strings comprising a first phrase that narrowed down the search to cloud computing, and a second phrase related to the objective of the study. The following first phrases were used: "cloud computing", "infrastructure as a service", "platform as a service", "software as a service", and the associated acronyms "IaaS", "PaaS" and "SaaS". The second phrases used to select relevant solutions were: "interoperability", "portability", "vendor lock" and "lock-in". The search engines from (i) retrieved all primary studies whose title, abstract and/or keywords matched any of the query strings.

From the 721 primary studies identified through sources (i)-(iv), we eliminated duplicates, documents with fewer than four pages [1], and solutions not directly related to cloud computing. The remaining 348 studies underwent a rigorous selection process. For studies published in peer-reviewed journals and conference proceedings, the selection involved the analysis of the study: (a) title and abstract; (b) conclusion; and (c) section titles and figures/diagrams. For other sources, such as websites and white papers, the selection involved a careful analysis of their analogous attributes. In both cases, the purpose of the analysis was to identify whether the study described a: (i) practical solution, e.g. an architecture, framework or API, or (ii) theoretical solution, e.g. a set of guidelines, requirements, or a conceptual framework. The 78 studies identified as practical or theoretical solutions were read in full, and were used to answer our research questions. To enable this process, a data collection form was used to summarise five characteristics of each solution studied:

1. *General information*: title, authors, DOI/URL and abstract;
2. *Source information*: data about the origin of the source, e.g. publication venue and type of research project;
3. *Taxonomy*: classifies the solution based on targeted cloud service model, level of maturity, and addressed issue(s);
4. *Level of industrial adoption*; and
5. *Work analysis:* an analysis of the solution goals, motivation and experiments carried out.

[1] Due to the limited information they provide.

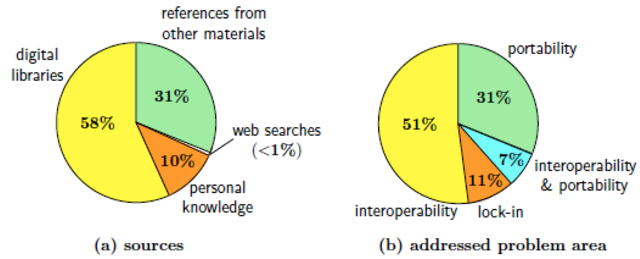

Fig. 1. Sources and addressed problem areas for the 78 solutions.

IV. Results

A. Solution Identification, Analysis & Classification

From a total of 721 materials retrieved, 78 (11%) were identified as providing solutions to problems related to vendor lock-in, portability and/or operability in the cloud. From these solutions, 55 (71%) were published in peer-reviewed journals and conference proceedings, and 23 (29%) were obtained from alternative sources such as web sites and white papers. Fig. 1 shows the partition of the solutions across the four types of sources described in the previous section, and the problem areas targeted by these solutions.

Each solution proposes a way to address one or more issues of vendor lock-in, portability or interoperability. The solutions were grouped according to their similarities, resulting in 26 different kinds of solutions (Fig. 2). Platforms followed by API and architectures are the most common types of solution. Frameworks, approaches, and tools are also often proposed. It is important to highlight that, unlike related work (see Section VI), we focus on classifying the solutions according to the type of artefact that they propose instead of the nature of the solution (standard, commercial, and so forth). It is valuable to highlight that, in this study, we identified the solutions according to the artefact that they devise. Thus, although Fig. 2 does not show an entry for standards, they are present, such as the interfaces proposed in [26] and [27], the packaging format devised in [28], and the specification presented in [12].

Some types of solution have been used to address a combination of portability, interoperability and vendor lock-in issues. For instance, the solution in [29] focuses on interoperability, whereas the solution proposed in [8] focuses on portability, and [30] focuses on vendor lock-in, although all of them are APIs. Although some solutions were grouped because they propose the same type of solution, they can deal with different parts of a problem. For Architectures, for instance, the solutions in [31] and [11] are general architectures aiming to support interoperability, while the solutions in [32], [33], and [18] propose architectures, but have more specific goals, such as an architecture for a broker that supports interoperability [33]. Similarly, some APIs are focused on supporting the management of resources ([34] and [35]) whereas others aim to abstract over the differences between different providers [8]. In contrast with some solutions that can be directly understood in context, such as an API or proxy, others are unclear, such as a framework or an approach. This is the case with the RESERVOIR project [3], which is presented in its official website as a framework. Unless the report that presents the solution makes absolutely clear the type of solution proposed, we used the name adopted by its author.

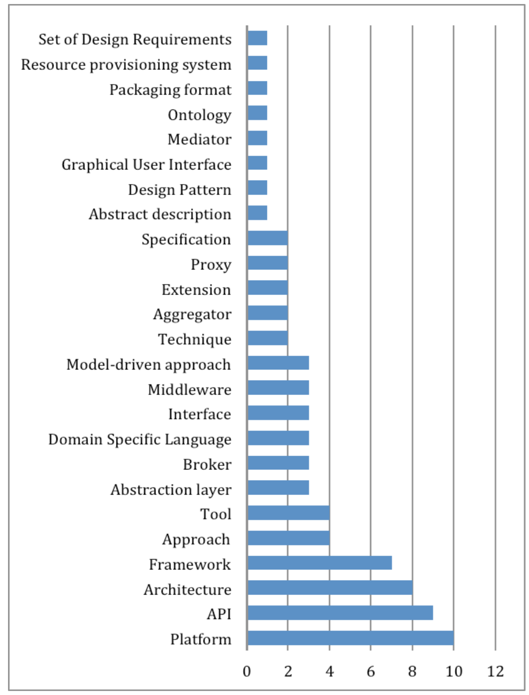

Fig. 2. Types of solution, and their occurrences.

Whereas 47 solutions (60%) present a developed artefact, 15 solutions (19%) do not present any artefact. 16 solutions (21%) evaluate their work. Although this classification might clarify the status of current approaches, we should not consider it as an indication that the solution is ready to be applied. In general, solutions proposed by community, industry and consortiums offer functional versions of their product.

Most evaluations consider performance analysis, such as measuring throughput and latency [8], timestamp [36], inference and query time [13], schedule time [32], overhead in a translation process [37], and time to responds to user requests [38]. We also found solutions that evaluate the financial cost to the cloud user when using their solutions, namely [39], [11] and [17]. The most common form of evaluation involves using prototypes, or simulations. Regarding industrial usage, 5 solutions (6%) are commercial, 11 (14%) have industrial partners or receive some support from industry, 8 (10%) have one or more authors from industry, and 5 (6%) reported having their product running in a company. The remaining 49 solutions (63%) do not clearly state their relation with industry.

Regarding the service model, 26 solutions (33%) are intended for IaaS, whereas 7 (9%) for PaaS and 1 (1%) for SaaS. There are also 12 solutions (15%) which are focused on more than one service model, and 8 solutions (10%) which are intended to other service models, such as Data-as-a-Service. It is worth noting that 24 solutions (31%) do not specify a target service model. Some solutions, such as [36] and [16], show examples related to SaaS and IaaS (respectively), but the authors did not make clear if it is the target. Other solutions, such as [31] and [40], have some relation with well-known solutions. However, we cannot infer the intended service model from the description of the solution. One interpretation might be that these solutions aim to be generic approaches that target all service models.

TABLE I. THE COMMON SOURCES

Name	# studies
IEEE International Conference on Cloud Computing Technology and Science	5
International Conference on Cloud Computing and Services Science	5
Lecture Notes in Computer Science (Journal)	4
IEEE Intl. Conference on Cloud Computing	3
Intl. Conf. on Recent Trends In IT	2
Future Generation Computer Systems (Journal)	2

B. *Identification and Analysis of Research Area*

It is critical to any researcher in a new area to identify the most relevant sources of material. In addition, writing and publishing results are commons task in research. To support these activities, we show in the Table I the journals and conferences most targeted by researchers to publish their results. We also analysed the relationships between studies, by examining citations. Our analysis allows us to reason about the impact of a solution. The OCCI project, for example, is cited as related work [2], [37], as an example [8] and as part of a solution [15]. Table II shows the most cited solutions. Using well-established solutions is a potentially promising approach to yield sound solution, as evidenced by recent work [41].

C. *Gap Analysis*

To assist in identifying gaps in existing work, the solutions were classified within a Computer Science area, according to the devised artefact, using the 10 of the 12 categories of the first level of the ACM Computing Classification System (Table III). Two general categories have been disregarded: *General and reference* and *Proper nouns*. Existing solutions cover most of the ACM categories, except for areas such as *Applied Computing* and *Theory of computation*. For instance, we did not find any solution dealing with the impact of changes in the business process due to the use of one specific product, or working on issues related to the cross-organizational business in an interoperability scenario. Similarly, of the work that we have considered, no work proposes algorithms to balance load between providers in an interoperability case.

V. ANALYSIS

By performing a thorough analysis of the solutions identified in our systematic review, we made three key observations. Firstly, regardless of the solution, a common approach is creating an abstraction layer that seeks to hide the differences between cloud providers and users. In turn, adapters are used to support the interaction between the abstraction layer and the target. Secondly, existing solutions are complementary rather than contradictory. This is the case for APIs and ontologies, for instance. A prerequisite for proposing an API is the identification of entities that the API will deal with [8]. Likewise, architecture is a precondition to propose a platform in [42] and [38]. However, reusing or extending existing solutions is not common practice, although we have identified a strong relation between some approaches. An exception is the use of standards. For example, the OCCI standard [12] is used by other solutions (e.g., [43] and [44]). Finally, solutions of a same type are very similar, with a slight few differences. For example, analysing three platforms for cloud management ([45], [44], [43]), we could not identify any

TABLE II. THE MOST CITED SOLUTIONS

Name	# citations
Eucalyptus [44]	22
Open Virtualization Format [28]	18
Open Cloud Computing Interface [12]	17
OpenNebula [43]	15
Reservoir Project [3]	9
Nimbus [46]	8
Libcloud [35]	7
Delta-Cloud [34]	7
InterCloud [11]	7
OpenStack [45]	6

TABLE III. SOLUTIONS DISTRIBUTIONS ACCORDING TO CS AREA

Computer Science Area	(%)
Software and its engineering	72%
Information systems	18%
Computing methodologies	5%
Social and professional topics	3%
Human-centered computing	1%
Security and privacy	1%

particular difference which could benefit one over another. The main difference in APIs, used to programmatically access cloud services, is the technology supported: Java [7], Python [35], and REST [34].

Even though some solutions are mature enough to be used, they still face some challenges. From a research point of view, an initial challenge is to conduct more rigorous studies. The quality assessment form shows that only a few studies meet the established requirements. From 55 studies, only one study clearly presents the research question whereas no studies present the method for conducting the study. This statistic considers only research papers as such information is expected from this kind of document. In addition, less than one third of studies included any form of evaluation. Another challenge is increasing the use of empirical evidence to support the studies. It is common that a solution is proposed based on hypothetical scenarios or requirements derived from a brief analysis of current cloud providers. Empirical evidence is critical to evidencing the problem as well as to supporting the solution.

Cloud computing is built on well-known and established concepts, such as distributed computing and virtualization. However, we identified that solutions do not take advantage of approaches previously devised in areas related to cloud computing. Analysing the issues and solutions for vendor lock-in, portability, and interoperability in areas related to cloud computing might save time and effort, and also support the cloud solutions providing a sound foundation.

Considering that cloud providers, such as Amazon and Microsoft, are also concerned about portability and interoperability, much may be gained by increasing collaboration with industry. In our systematic review, we have identified that 37% of all proposed solutions have some relation with the industry. IBM and HP have demonstrated their concerns recently announcing their support to OpenStack, whereas Amazon and Microsoft have developed their own strategies to enable virtual machine portability.

Finally, a further challenge is that solutions should reflect – and be adaptable to – the social and technical contexts in which they are applied. We identified that in recent studies to migrate an application (not necessarily to make use of cloud), companies conduct a series of activities, such as analysis, test, and training. The process is cumbersome, usually taking a long time to be completed. We believe that the migration process is similar when migrating to the cloud, or migrating to a different cloud provider. Thus, we argue that there is a need for solutions to portability, interoperability and vendor lock-in issues to address socio-technical and business challenges.

VI. Related Work

The surveys analysed in this section were selected because they share the same concern as our review: to identify and analyse solutions for vendor lock-in, portability, or interoperability. Loutas et al. [47] carried out a literature review to identify solutions and derive requirements to support semantic interoperability in cloud computing. Like our review, their study covered research databases, standardization bodies, industry solutions and funded projects, resulting in 48 studies and 18 solutions, which were analysed and classified over four issues: definitions, standardization solutions, views and interoperability solutions. In addition, a second type of classification was made, based on the requirements that authors identified. As the result of their analysis, the authors identified two main requirements: standardized models to describe cloud entities, and standardized APIs supported by different vendors.

Govindarajan & Lakshmanan [6] explored standardization solutions covering interoperability, security, portability and governance in cloud, resulting in 11 solutions. For each issue covered, the authors conducted an analysis in four issues: (i) purpose, expectations and challenges; (ii) solutions; (iii) market adoption; and (iv) gaps or areas for improvement. For interoperability, the authors classified the approaches in three groups, referring to approaches from standardization bodies, industry solutions, and solutions for brokering and management. In addition, they briefly presented the role of standardization for portability and interoperability.

Petcu [7] identified and classified several issues of cloud portability and interoperability. Highlighting the growing number of approaches to address these issues, Petcu considered that it would be useful to survey concepts such as requirements, evolution stages, and types of portability and interoperability. Petcu presented a comprehensive list of approaches, classifying the approaches according to its type (e.g., APIs, standards, and so forth). This analysis was used as the basis to support the mOSAIC approach, also presented in the article. The work is the most comprehensive and detailed analysis of available solutions for vendor lock-in so far.

VII. Threats to Validity

Although the assessment questions were set up to assess the quality of material selected for reading, indications of the quality of a publication were not used as inclusion or exclusion criteria. This does not differ from common practice in this kind of study [48], and we decided to follow this approach due to the relative immaturity of the area. Thus, using quality assessment to limit inclusion would have significantly reduced the number of solutions to be analysed.

Inconsistent use of terminology in the primary studies might also harm the results. Although there is some agreement regarding definitions for vendor lock-in, portability and interoperability in cloud computing, some studies use some terms differently by, for example, identifying portability and interoperability as the same concept. To prevent misclassification, we adopted a widely accepted definition for vendor lock-in, portability and interoperability at the early stages of this work, and they were used throughout this work.

Finally, some valuable solutions might not have been included due to the date of their publication. Our review concluded in March 2013 and since then new (and important) solutions might have been published.

VIII. Conclusion

We have presented a systematic review of approaches that address issues relating to vendor lock-in, portability and interoperability for cloud computing. Our review has been conducted to identify, analyse and classify existing solutions. In contrast with related work, our work: (i) extends the scope beyond one specific issue, addressing vendor lock-in, portability and interoperability; (ii) identifies the solutions according to the devised artefact; (iii) provides an overview of the research area; (iv) highlights areas in which solutions have not been proposed; in addition, we identified the gaps of research regardless the interest in one specific solution; (v) identifies the relations amongst solutions; and (vi) summarises the most relevant work done so far addressing vendor lock-in, portability, and interoperability in cloud computing.

In future work, we will: (i) perform a further systematic review to identify possible solutions for vendor lock-in, portability and interoperability in five areas closely related to cloud computing, namely: grid computing, virtualization, operating systems, programming, and telecommunications; (ii) devise and carry out an experiment migrating several applications between different cloud providers and service models, to collect empirical evidence of issues arisen during application migration; and (iii) perform a survey of companies to identify their existing practices for performing application and migration and their requirements for portability and interoperability in cloud computing.

Acknowledgment

The authors are grateful to Shinji Kikuchi for valuable feedback on an earlier version of the manuscript.

References

[1] T. Dillon, C. Wu, and E. Chang, “Cloud Computing: Issues and Challenges,” in 2010 24th IEEE International Conference on Advanced Information Networking and Applications, 2010, pp. 27–33.

[2] N. Loutas, V. Peristeras, T. Bouras, E. Kamateri, D. Zeginis, and K. Tarabanis, “Towards a Reference Architecture for Semantically Interoperable Clouds,” in 2010 IEEE Second International Conference on Cloud Computing Technology and Science, 2010, pp. 143–150.

[3] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. S. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Galán, “The Reservoir model and

architecture for open federated cloud computing," IBM Journal of Research and Development, vol. 53, no. 4, pp. 4:1–4:11, Jul. 2009.

[4] L. Rodero-Merino, L. M. Vaquero, V. Gil, F. Galán, J. Fontán, R. S. Montero, and I. M. Llorente, "From infrastructure delivery to service management in clouds," Future Generation Computer Systems, vol. 26, no. 8, pp. 1226–1240, Oct. 2010.

[5] P. Hofmann and D. Woods, "Cloud Computing: The Limits of Public Clouds for Business Applications," IEEE Internet Computing, vol. 14, no. 6, pp. 90–93, Nov. 2010.

[6] A. Govindarajan and Lakshmanan, "Overview of Cloud Standards," in in Cloud Computing, London: Springer London, 2010, pp. 77–89.

[7] D. Petcu, "Portability and interoperability between clouds: Challenges and case study," in in Towards a Service-Based Internet, vol. 6994 LNCS, Poland: Springer Berlin Heidelberg, 2011, pp. 62–74.

[8] D. Petcu, G. Macariu, S. Panica, and C. Crăciun, "Portable Cloud applications—From theory to practice," Future Generation Computer Systems, Jan. 2012.

[9] N. Ferry, A. Rossini, F. Chauvel, B. Morin, and A. Solberg, "Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems," in IEEE 6th Intl. Conf. on Cloud Computing, 2013, pp. 887–894.

[10] VMWare, "vCloud." [Online]. Available: http://vcloud.vmware.com/. [Accessed: 25-May-2013].

[11] R. Buyya, R. Ranjan, and R. Calheiros, "InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services," in in Algorithms and Architectures for Parallel Processing, vol. 6081, Korea: Springer, 2010, pp. 13–31.

[12] O. G. Forum, "Open Cloud Computing Interface (OCCI)." [Online]. Available: http://occi-wg.org/. [Accessed: 21-Feb-2013].

[13] V. Nelson and V. Uma, "Semantic based Resource Provisioning and scheduling in inter-cloud environment," in International Conference on Recent Trends in Information Technology, 2012, pp. 250–254.

[14] S. . b Yangui and S. . Tata, "PaaS elements for hosting service-based applications," in CLOSER 2012, 2012, pp. 476–479.

[15] A. Sampaio and N. Mendonça, "Uni4Cloud," in 2nd Intl. workshop on Software engineering for cloud computing, 2011, pp. 15–21.

[16] J. Tao, H. Marten, D. Kramer, and W. Karl, "An Intuitive Framework for Accessing Computing Clouds," Procedia Computer Science, vol. 4, no. 1, pp. 2049–2057, Jan. 2011.

[17] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, "RACS: a case for cloud storage diversity," in 1st ACM SoCC '10, 2010, pp. 229–240.

[18] W.-T. Tsai, X. Sun, and J. Balasooriya, "Service-Oriented Cloud Computing Architecture," in 2010 Seventh International Conference on Information Technology: New Generations, 2010, pp. 684–689.

[19] P. Mell and T. Grance, "NIST Definition of Cloud Computing," Gaithersburg, 2011.

[20] B. Satzger, W. Hummer, C. Inzinger, P. Leitner, and S. Dustdar, "Winds of Change: From Vendor Lock-In to the Meta Cloud," IEEE Internet Computing, vol. 17, no. 1, pp. 69–73, Jan. 2013.

[21] "Cloud Computing Use Cases," Cloud Computing Use Cases group, 2010. [Online]. Available: http://cloudusecases.org/. [Accessed: 11-Jun-2013].

[22] "Use Cases and Functional Requirements for Inter-Cloud Computing," Global Inter-Cloud Technology Forum, 2010. [Online]. Available: http://www.gictf.jp/doc/GICTF_Whitepaper_20100809.pdf. [Accessed: 10-Jun-2013].

[23] C. Shan, C. Heng, and Z. Xianjun, "Inter-cloud operations via NGSON," IEEE Communications Magazine, vol. 50, no. 1, pp. 82–89, Jan. 2012.

[24] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," Keele, 2007.

[25] J. Biolchini, P. G. Mian, A. C. C. Natali, and G. H. Travassos, "Systematic Review in Software Engineering," Rio de Janeiro, 2005.

[26] SNIA, "Cloud Data Management Interface (CDMI)." [Online]. Available: http://www.snia.org/cdmi. [Accessed: 21-Feb-2013].

[27] "Unified Cloud Interface Project," Cloud Computing Interoperability Forum. [Online]. Available: http://code.google.com/p/unifiedcloud/. [Accessed: 15-May-2013].

[28] DTMF, "Open Virtualization Format (OVF)." [Online]. Available: http://www.dmtf.org/standards/ovf. [Accessed: 21-Feb-2013].

[29] A. Merzky, K. Stamou, and S. Jha, "Application Level Interoperability between Clouds and Grids," in 2009 Workshops at the Grid and Pervasive Computing Conference, 2009, pp. 143–150.

[30] The Apache Software Foundation, "Apache Nuvem." [Online]. Available: http://incubator.apache.org/nuvem/.[Accessed: 21-Feb-2013].

[31] T.-F. Fortis, V. I. Munteanu, and V. Negru, "Towards an Ontology for Cloud Services," in 6th International Conference on Complex, Intelligent, and Software Intensive Systems, 2012, pp. 787–792.

[32] T. S. Somasundaram, K. Govindarajan, M. . Rajagopalan, and S. M. Rao, "An architectural framework to solve the interoperability issue between private clouds using semantic technology," in Intl. Conf. on Recent Trends in Information Technology, 2012, pp. 162–167.

[33] F. Jrad, J. Tao, and A. Streit, "SLA based service brokering in intercloud environments," in CLOSER 2012, 2012, pp. 76–81.

[34] The Apache Software Foundation, "Delta-Cloud." [Online]. Available: http://deltacloud.apache.org/. [Accessed: 21-Feb-2013].

[35] T. A. S. Foundation, "Libcloud." [Online]. Available: http://libcloud.apache.org/. [Accessed: 21-Feb-2013].

[36] H. Flores, S. N. Srirama, and C. Paniagua, "A generic middleware framework for handling process intensive hybrid cloud services from mobiles," in 9th MoMM '11, 2011, pp. 87–94.

[37] J. Ejarque, J. Alvarez, R. Sirvent, and R. M. Badia, "A Rule-based Approach for Infrastructure Providers' Interoperability,"in IEEE 3rd Intl. Conf. on Cloud Computing Technology and Science,2011,pp. 272–279.

[38] A. Wolke and G. Meixner, "TwoSpot: A Cloud Platform for Scaling Out Web Applications Dynamically," in in Towards a Service-Based Internet, vol. 6481, Belgium: Springer, 2010, pp. 13–24.

[39] P. Pawluk, B. Simmons, M. Smit, M. Litoiu, and S. Mankovski, "Introducing STRATOS: A Cloud Broker Service," in 2012 IEEE Fifth International Conference on Cloud Computing, 2012, pp. 891–898.

[40] D. Ardagna, E. Di Nitto, P. Mohagheghi, S. Mosser, C. Ballagny, F. D'Andria, G. Casale, P. Matthews, C.-S. Nechifor, D. Petcu, A. Gericke, and C. Sheridan, "MODAClouds: A model-driven approach for the design and execution of applications on multiple Clouds," in 4th Int.l Workshop on Modeling in Software Engineering, 2012, pp. 50–56.

[41] G. Vernik, A. Shulman-Peleg, S. Dippl, C. Formisano, M. C. Jaeger, E. K. Kolodner, and M. Villari, "Data On-boarding in Federated Storage Clouds," in IEEE 6th Intl. Conf. on Cloud Computing,2013,pp.244–251.

[42] F. Paraiso, N. Haderer, P. Merle, R. Rouvoy, and L. Seinturier, "A Federated Multi-cloud PaaS Infrastructure," in 2012 IEEE Fifth International Conference on Cloud Computing, 2012, pp. 392–399.

[43] O. P. (OpenNebula.org), "OpenNebula." [Online]. Available: http://opennebula.org/. [Accessed: 21-Feb-2013].

[44] I. Eucalyptus Systems, "Eucalyptus." [Online]. Available: http://www.eucalyptus.com/. [Accessed: 21-Feb-2013].

[45] T. O. Foundation, "openstack," 2013. [Online]. Available: http://www.openstack.org/. [Accessed: 24-Apr-2013].

[46] U. of Chicago, "Nimbus." [Online]. Available: http://www.nimbusproject.org/. [Accessed: 21-Feb-2013].

[47] N. Loutas, E. Kamateri, F. Bosi, and K. Tarabanis, "Cloud Computing Interoperability: The State of Play," in IEEE 3rd Intl, Conf. on Cloud Computing Technology and Science, 2011, pp. 752–757.

[48] D. I. K. Sjoberg, T. Dyba, and M. Jorgensen, "The Future of Empirical Methods in Software Engineering Research,"in FOSE,2007,pp.358–378.