Multi-Cloud: Expectations and Current Approaches

Dana Petcu
Institute e-Austria Timişoara and West University of Timişoara, Romania
petcu@info.uvt.ro

ABSTRACT

Using resources and services from multiple Clouds is a natural evolution from consuming the ones from in-silo Clouds. Technological and administrative barriers are however slowing the process. Fortunately the recent years are marked by the appearance of several solutions that are partially overpassing them. However, the approaches are quite various and not adopted at large scale. This paper intends to offer a snapshot of the current state-of-the-art and to identify the future steps in building Multi-Clouds. A list of basic requirements for a Multi-Cloud is proposed.

Categories and Subject Descriptors

C2.4 [**Computer Communication Networks**]: Distributed Systems—*Cloud Computing*

Keywords

Cloud computing; Multi-Cloud; interoperability; portability; brokering

1. INTRODUCTION

Nowadays the usage of services and resources from multiple Clouds is driven by the needs of their consumers expressed in simple requirements, like service quality and cost. However the guidance through the variety of the offers, based on monitoring tools for the quality of services, is in its early stages.

This paper intends to identify the state of the art in building Multi-Cloud and which enhancements need to be done to the current solutions to comply with the needs of software developers. First the need of Multi-Cloud is stated. Secondly, the position of Multi-Cloud versus other multiple Cloud models is identified. Thirdly, the available software support from commercial and academic communities is revised. After this analysis, the requirements of a Multi-Cloud are stated. The barriers to provision technical solutions for these requirements are then identified. Finally, the future steps are referring to a near future.

The paper is not presenting an innovative approach. Instead it intends to point to the gaps to be filled in the near future by Multi-Cloud developers.

MultiCloud'13, April 22, 2013, Prague, Czech Republic.

2. THE NEED FOR MULTIPLE CLOUDS

As NIST report [17] has recently stated the Clouds can be used serially, when moved from one Cloud to another, or simultaneous, when using services from different Clouds. The simple scenarios are the migration from a Private Cloud to a Public Cloud (for the serial case), respectively the Hybrid Cloud, when some services are lying on the Private Cloud, while other services are lying on a Public Cloud (for the simultaneous use).

The reasons for which the services and resources from multiple Clouds are needed are various. As being spread in various research or business papers, we considered useful to gather them here:

- *N1* deal with the peaks in service and resource requests using external ones, on demand basis;
- *N2* optimize costs or improve quality of services;
- *N3* react to changes of the offers by the providers;
- *N4* follow the constraints, like new locations or laws;
- *N5* replicate the applications or services consuming resources or services from different Cloud providers to ensure their high availability;
- *N6* avoid the dependence on only one external provider;
- *N7* ensure backup-ups to deal with disasters or scheduled inactivity;
- *N8* act as intermediary;
- *N9* enhance own Cloud resource and service offers, based on agreements with other providers;
- *N10* consume different services for their particularities not provided elsewhere.

Different actors can be interested in different scenarios. The users can be individuals with interest in *N2, N3, N7* or *N10.* Companies which are consuming Cloud services can be interested in *N1-N2, N4-N7* or *N10.* Cloud providers can be interested in *N1, N7-N9.*

3. THE POSITION OF MULTI-CLOUD VERSUS OTHER MULTIPLE CLOUDS

Despite the fact that multiple Clouds have only a quinquennial history, several terms were coined to depict their various forms. We name here only few: Multi-Cloud, Cloud Federation, Inter-Cloud, Hybrid Cloud, Cloud-of-Clouds, Sky Computing, Aggregated Clouds, Multi-tier Clouds, Cross-Cloud, Cloud Blueprint, Cloud Merge, Fog Computing, Hierarchical Clouds, Distributed Clouds and so on.

While several Cloud ontologies are already available (for example in [15]), the taxonomy of multiple Clouds is not well established and the border between different terms is still cloudy. Therefore we consider useful to delimit the Multi-Cloud from other multiple Clouds models.

We agree with the statement that there are two types of delivery models in multiple Clouds: Federated Cloud and Multi-Cloud. The difference is made, according to [6], by the degree of collaborations between the Clouds involved and by the way by which the user interacts with the Clouds. In the first model there is an agreement between the different Cloud providers to share their resources, while in the second model there is no such agreement. Similarly, in [7] the two models are classified based on the collaboration type: volunteer, in Federation, or not, in Multi-Cloud. In the first model the user interacts with one Cloud and is not aware that the resources or services that are consumed are from another Cloud. In the second model the user is aware of the different Clouds and is responsible, or a third party is responsible, to deal with the provisioning of the services or resources.

Following the above mentioned classification and the proposal from [7], the term Multi-Cloud can denote the usage of multiple and independent Clouds by a client or a service.

The Cloud provider needs are served by the Federated Cloud. The main drive is the need *N9*, especially to acquire new resources due to limitation of the own ones. However, *N2* and *N4* can be also the reasons of its usage, especially in the case of the geographical location restrictions or own cost reduction politics.

The Cloud clients needs are served by the Multi-Cloud model. The main drive is the need *N2*, as relying upon own or third party capacity to identify the appropriate service or resource for a concrete application or new service. However, any reason from *N1* to *N8* can be also motivating the usage of Multi-Cloud.

In the academic communities, the Federated Cloud is following the Grid path by establishing Community Clouds, like Open Cirrus (opencirrus.org). However agreements between large Cloud providers are difficult to be achieved, especially due to the need to control the resources and services that the external provider is expected to obtain for its benefit. Hopefully, the small Cloud providers will be interested in such agreements to enhance their offer and to avoid the cases when potential users are not served due to the lack of sufficient resources; however, few cases are registered until now (especially on the vertically of the delivery model stack, i.e. SaaS built with an external PaaS, PaaS that are using an external IaaS). In this context, we expect that the Multi-Cloud is more appealing than the Federated Cloud at this stage of development of the multiple Clouds.

Sky Computing, Aggregated Clouds, Multi-tier Clouds, or Cross-Cloud are particular cases of Federated Clouds and therefore are not discussed here. One of the main problem is the interoperability between different Clouds, while in Multi-Cloud the main one is the portability of applications between Clouds [20].

The most common Multi-Cloud is the Hybrid Cloud that involves two or more Clouds, e.g. a Private and a Public Cloud. Often Hybrid Clouds are used for Cloud Busting, the usage of Public Cloud in cases of peaks registered in the Private Cloud. The migration from one Cloud to another, on-time or real-time, is another example (one-time usage of a Multi Cloud). A good example of an application of a Multi-Cloud is presented in [19], related to the monitoring of multiple Clouds.

According to [7] there are two categories of Multi-Clouds: service based or library based. In the first case, a special service is offering brokerage between multiple Clouds based on clients' service level agreements or provisioning rules and performs deployment, execution and monitoring. In the second case, a library facilitates a uniform way to access multiple services and resources, as well as the provisioning of services and resources from multiple Clouds.

In the NIST Cloud Computing Reference Architecture [17], beyond the Cloud provider and Cloud consumer, another major actor is the Cloud broker. A Broker is expected to intermediate the relationship between the providers and consumers. In the case of a Federation, it is part of a centralized entity or appears to each provider. In the case of Multi-Cloud, it is part of the special service or library. In each model, it can take important roles (extension of the ones from [13]): optimizer in finding the best match between requirement and offer; adapter by offering a unique management interface; extension of existing services; aggregation with indexing; splitter of user requests to multiple providers; arbitrage between Clouds.

An Inter-Cloud is a Federated Cloud or a Multi-Cloud that includes at least one broker and offers dynamic service provisioning. According [3], it is expected to provide an opportunistic and scalable application service provisioning environment. Moreover, it goes beyond the management of resources and service from different Clouds, by including Cloud governance or marketplace.

4. MANAGEMENT SOFTWARE FOR MULTI-CLOUD

An important functionality of the Multi-Cloud is to manage the deployments on various Clouds. Therefore, its classification is depending on the approach taken in the implementation of such management software: as identified in [7] and mentioned in the previous section, the Multi-Cloud can be library-based or service-based.

The most known library-based approaches are:

jclouds is an open source Java library designed to support the portability of Java applications, which allows the uniform access to the resources of AWS, vCloud, CloudServers, ElasticHosts, Eucalyptus, GoGrid, OpenHosting, Rackspace, DeltaCloud and so on (complete list at www. jclouds.org).

libcloud is a Python library that abstract the differences among the programming interfaces of various services offered by OpenNebula, GoGrid, Enomaly, SliceHost, Elastic Hosts, RackSpace, Eucalyptus, AWS, Joyent, vCloud and so on (details at libcloud.apache.org)

δ-cloud is a REST-based API written in Ruby which allows also the connections to various Cloud resources of AWS, Eucalyptus, GoGrid, OpenNebula, Eucalyptus, Rackspace, OpenStack and so on (details at delta-cloud.apache.org)

SimpleCloud is a PHP library offering uniform interfaces for file and document storage, queues and infrastructure services of AWS, RackSpace, Azure, Nirvanix (details at www.simplecloud.org).

The service-based approach for Multi-Cloud can also be classified in two categories: hosted or deployable.

The most known hosted services are the commercial ones:

RightScale is offering a management platform for control and administration of deployments in different Clouds, based on AWS, Eucalytus, GoGrid, vCloud and Flexiant (details at www.rightscale.com). Its Multi-Cloud Engine is able to broker capabilities related to virtual machine placement in Public Clouds;

enStratus allows the management, monitoring, automation and governance of resource consumption based on the services from AWS, GoGrid, Joyent, OpenStack, CloudStack, CloudSigma, vCloud, Azure and so on (details at enstratus.com);

Kaavo allows the management of distributed applications and workloads in various Clouds based on AWS, Rackspace, Terremark, Eucalyptus (details at www.kaavo.com).

Deployable services are results of open-source projects like:

Aeolus is an open-source cloud management software written in Ruby and provided for Linux systems by RedHat. It is based on the δ-cloud library (details at www.aeolusproject.org);

mOSAIC is an open-source API and a deployable Platform as a Service allowing the deployment and the life-cycle control of applications consuming infrastructure services. Services that can be consumed are provided by AWS, GoGrid, vCloud, Eucalytus, RackSpace, CloudSigma, DeltaCloud, Flexiscale and so on (details at bitbucket.org/mosaic and [23]);

Optimis is a deployable Platform as a Service that allows Cloud service provisioning and the management of the life-cycle of the services. Services that are currently connected are from AWS, OpenNebula, Eucalyptus, Emotive, Flexiant (details at www.optimis-project.eu/Toolkit_v2).

The Cloud brokers are playing an important role in both Multi-Cloud and Inter-Cloud. The most known independent Cloud brokers (without offering a complete solution for a Multi-Cloud) are:

SpotCloud provides a marketplace for infrastructure service and a matching service with the client requirements (details at spotcloud.com);

Scalr provides deployment of virtual machines in various Clouds and includes automated triggers to scale up and down (details at scalr.net);

Stratos offers single sign-on and monitors resource consumption and the fulfillment of service level agreements and offers auto-scaling mechanisms (details at wso2.com/cloud/stratos).

Unfortunately none of the above mentioned solutions are dealing with all the requirements of the Multi-Cloud as exposed in section 7 that follows.

5. RECENT RESEARCH PROTOTYPES

Beyond the production-ready management tools, there are significant efforts of the research community to support the development of innovative solutions for the Multi-Clouds. In this section we mention only few of them. A more comprehensive study is done in [21].

Most of these efforts are intending to help in eliminating the barrier of vendor lock-in. The reasons of vendor lock-in are various: services that are subject of high investment protection, lack of wide acceptance of standards, proprietary APIs of the services, and so on. The problem is not necessarily a consequence of the vendor wills, but can be a reflection of the large option space in what concerns the set of hardware and software stacks needed to build a Cloud service.

Migrations from one Cloud to another poses multiple problems that are solved today only in particular cases. For example, Shrinker [24] is a hypervisor modification based on the detection of inter-virtual-machines data similarities. PSIF [12] models and tries to resolve semantic interoperability conflicts raised during the deployment or the migration of an application between PaaSs.

Several Broker prototypes were build also recently. SORMA [16] use bidders and sellers to represent the beneficiaries of the brokering system. SERA [5] uses agents to represent the beneficiaries of the brokering system, to schedule and control the resources, and to enable monitoring, registering or recovery. CCFM [4] is for example a manager with discovery, match-making and authentication features. Zeel/i [8] allows single-sign and the selection of Cloud resources according to specific requirements. An extension of OpenFlow reported in [9] is solving the selection problem by a mixed integer program. Cloudbus [3] is a platform that incorporate several brokers. mOSAIC approach is based on a Cloud Agency gathering client and provider agents in a brokerage process working with service level agreements [26].

Search engines and benchmarks are often used in Multi-Cloud. Cloudle [10] is an example of a Cloud service search engine based on a specific Cloud ontology. CloudCmp [11] is a set of benchmarking tools for comparing services from different point of views, e.g. elasticity, persistence of storage, intra-cloud.

The advances in network technologies are expected to facilitate the transition from the Cloud to Multi-Cloud. Network virtualization techniques for distributed resources in different administrative domains, like TinyViNe [25], were built recently. Making a step forward, ORCA [14] enables computational and network resources from multiple clouds and network substrates to be aggregated into a single virtual resource.

6. EUROPEAN COLLABORATIVE RESEARCH EFFORTS TO SUPPORT MULTIPLE CLOUDS

The European interest in the subject is highly motivated by the fact that the market dominating Cloud providers are non-European and the European Cloud providers are striving to attract customers with special services, and worldwide e-markets of Cloud services are emerging. Therefore several research and development collaborative on-going projects that are dealing with multiple Clouds are funded by European Commission.

We mentioned already the projects mOSAIC and Optimis. Other projects started in the same time are: Cloud4SOA (www.cloud4soa.eu) that is dealing with portability of applications between PaaSs by relying upon semantic technologies, 4CaaST that is building a Cloud blueprint (4caast.morfeo-project.org), REMICS that is migrating applications from one Cloud to another using model-driven engineering, code introspection and rewriting (www.remics.eu), Contrail that is allowing a centralized management of web application deployments (www. contrail-project.eu), Vision Cloud that is dealing with vendor-agnostic data management services (www.visioncloud.eu), TClouds that is investigating the trust management in multiple Clouds (www.tclouds-project.eu). Details can be found in [2] or [22]. Model-driven engineering is proposed to be used in Multi-Cloud by the more recent projects, like MODAClouds (www.modaclouds.eu), ARTIST (www.artist-project.eu) or PaaSage (www.paasage.eu). Started at the same time, Broker@Cloud (www.broker-cloud.eu) develops methods and mechanisms for quality assurance and optimization of software-based services, while Cloudspaces (www.cloudspaces.eu) is looking to interoperability mechanisms between Personal Clouds.

The concept of Cloud Blueprint [18] was introduced recently as an enhanced Cloud delivery model. It is a reference architecture for Integration-as-a-Service for pre-built, preconfigured, pre-optimized application components deployable in various Clouds. It is also an example on a model-driven approach to represent and manage Cloud services and resources: templates are used by providers to specify the service features, while the blueprints are used by clients to assemble their applications. Such approach is expected to be considered in the MODAClouds project that intends to provide methods, a decision support system, an open source IDE and a run-time environment for the high-level design, early prototyping, semi-automatic code generation, and automatic deployment of applications on Multi-Cloud with guaranteed quality of service [1].

7. THE TECHNICAL REQUIREMENTS FOR A MULTI-CLOUD

In this section we propose a list of requirements to be fulfilled by the Multi-Cloud.

1. Development group

R1.1 Offer a resource and service (meta-)management software (portal, service or interface);

R1.2 Offer services that are Cloud vendor agnostic;

R1.3 Abstract service control interfaces of multiple Clouds;

R1.4 Offer an interface for describing functional and non-functional requirements of the clients;

R1.5 Support the application portability between the connected Clouds;

R1.6 Offer an Integration-as-a-Service or service aggregators to combine services from different Clouds*;

R1.7 Comply with the current standards and protocols in Cloud resource management;

2. Deployment group

R2.1 Offer a facility for the selection of consumable Cloud services and resources;

R2.2 Offer support for the deployment of components of applications in multiple Clouds;

R2.3 Preserve the particularities of various Clouds;

R2.4 Do not impose any constrains to the connected Clouds;

R2.5 Allow seamless join by new Cloud without changing local policies (both of the Cloud and the Multi-Cloud);

R2.6 Offer a broker or match-matching service;

R2.7 Implement a search engine based on a specific taxonomy or using semantic processing;

R2.8 Offer authentication services for single sign-on or Cloud credentials repositories;

R2.9 Support the connection with the top Cloud providers;

R2.10 Support the application relocation between Clouds**;

R2.11 Allow the deployment on Private Clouds to enable testing, debugging, or privacy;

R2.12 Implement network overlay technologies to overcome limited connectivity;

R2.13 Implement (meta-) scheduling, load-balancing or auto-scaling mechanisms;

R2.14 Use automated procedures for deployments***;

R2.15 Implement a recommendation system, a trust management system or a reputation management system;

R2.16 Enable a fair marketplace for Cloud services and updates with the latest offers of the connected Clouds;

3. Execution group

R3.1 Offer support for application component execution simultaneous in multiple Clouds;

R3.2 Offer a (meta-)monitoring service for the deployed applications;

R3.3 Offer a (meta-)monitoring facility of the Cloud resource consumptions;

R3.4 Allow the control of the full life-cycle of the deployed applications;

R3.5 Allows the measurement of the degree of fulfillment of the service level agreements;

R3.6 Allow dynamic allocation of resources or mechanisms for self-adaptation;

R3.7 Introduce only a small overhead in comparison with a direct connection to each supported Cloud.

* Like dashboards or smashups.
** At least of the simple case, of stopping the application in the current Cloud, and restarting it entirely and from the beginning in another (off-line). A more complex case is that in which the relocated application is decomposed and relocated over a new set of Clouds; or the on-line case when the status (VM, databases) need to be moved almost instantaneously from one Cloud to another (as expected in Federated Clouds, where is named migration).
*** Use deployment description languages or intelligent management systems.

The above described requirements are either served currently by the existing enabling software for Multi-Clouds or expressed in research and development activities related to

Multi-Clouds. Unfortunately none of the software solutions that were mentioned in the previous two sections is complying with all the above requirements. An analysis of the degree of fulfillment of the requirements by each solution is out of the scope of this paper. However, the sum of all the approaches that were mentioned is almost offering a complete solution.

8. THE TECHNOLOGICAL BARRIERS IN ENABLING THE MULTI-CLOUD

Following the wish list from the previous section, a Multi-Cloud enabler is invited to follow the current Cloud standards (*R1.7*). While the current Cloud standards are few (OCCI, CDMI, CIMI being the most relevant in the Multi-Cloud context), they are still not adopted on large scale. One reason is their limited scope. Another is the reluctance of the Cloud providers which do not see them as business needs or priority (or even the contrary, as a danger for innovation and market advantage).

As mentioned in Section 4 there are several libraries abstracting the Cloud APIs that are more widely adopted than the standards. However these services are offering the common denominator of the underlying services, and are loosing their individuality, i.e. entering in contradiction with *R2.3*. Moreover, while they are compliant with *R1.5*, they do not contain any concepts or mechanisms to ensure *R2.10*.

The diversity of services is a challenge for the service selection (*R2.1*). A methodology to compare Cloud service based on multiple criteria and for various user profiles is needed. Comparison criteria can vary from cost, policies, performance and so on. Best matching in a certain context instead an optimal matching is expected in most the cases due to the complexity of the problem. Moreover, for the performance measurements, independent observer services need to be built, in the context that the Cloud providers are reluctant in providing monitoring services or data about their services' performances. Furthermore, the few current monitoring services are heterogeneous and a meta-monitoring service (*R3.3*) is difficult to be build.

Heterogeneity is encountered to both low and high levels, from virtualization technologies, to programming environments. It is expected that the Multi-Cloud is hiding this heterogeneity. If this is happening at the Cloud provider level to a certain level, the meta-level is still lacking a complete offer beyond the research prototypes (i.e. fulfillment of the requirements *R1.1*-*R1.7*).

The hosted services are the most common services and therefore the common understanding of Cloud services is referring to this group. Their interfaces conceived by the providers are unique entry points to complex processes and heterogeneous resources. In the condition of ignoring the Cloud standards and protocols, a common understanding on a certain action or feature is hard to be reached. The Multi-Cloud developers needs to understand each particular interface in order to connect each service, fact that is not compliant with *R2.3* and *R2.5*. In particular, the achievement of portability or relocation (*R1.5* or *R2.10*) is considered a moving target, difficult to reach, if the set of APIs to comply with is constantly increased.

The development of a Multi-Cloud requires to offer solutions to multiple levels:

(a) business by establishing strategies, regulations, or mode of use, e.g. to fulfill *R2.3, R2.5, R2.8, R3.5*;

(b) semantic by establishing a taxonomy for calls, responses, functionality, e.g. to fulfill *R1.3, R1.4, R2.7*;

(c) application and services by enabling automation or configuration, e.g. to fulfill *R2.14, R3.4*;

(d) management by using rules, protocols, standards in deployment or relocation, e.g. to fulfill *R1.1, R1.7, R2.2, R3.1*

(e) image and data by using the specificities of each Cloud that is connected, e.g. to fulfill *R2.3, R2.9, R3.7*;

(f) network by allocation and admission procedures, e.g. to fulfill *R2.12, R3.6*.

In order to tackle with all the above enumerated levels, a large team with various expertises is needed, not necessarily available to one research team or one company. Therefore the horizon of a Multi-Cloud complete enabling solution is far at this moment. The adoption and improvement of the current existing partial solutions (like mOSAIC deployable PaaS or OPTIMIS toolkit) can speed the process.

The usage of services from multiple Clouds has been introduced first with the idea of Hybrid Cloud, where a Private Cloud and a Public Cloud are building a transitory Multi-Cloud. However the outages of Public Clouds and the security breaks have brought into discussions the trustfulness of the Hybrid Cloud. A solution to *R2.15* is therefore needed in Multi-Cloud, and very few prototypes are currently available as the trust management problem is more than a technical one.

9. CONCLUSIONS AND FUTURE DIRECTIONS

Following the current achievements in sustaining multiple Cloud usage scenarios, a concrete image of what a Multi-Cloud is and is not can be formed. However such a concrete image has not been exposed yet, and we considered useful to point which are the latest solutions enabling Multi-Clouds and to identify the gaps that are needed to be filled in the near future.

Instead claiming to offer a perfect image, our proposal in terms of Multi-Cloud requirements is considered as a starting point for latest improvements. A first step can be to identify of the degree of fulfillment of the requirements by the current theoretical approaches and technical solutions, and second one, to study and experiment how can they complement each other in order to build a more complete enabler for a Multi-Cloud.

10. ACKNOWLEDGMENTS

The research reported in this article is partially supported by the European Commission grant no. FP7-ICT-2011-8-318484 (MODAClouds).

11. REFERENCES

[1] D. Ardagna, E. Di Nitto, G. Casale, D. Petcu, P. Mohagheghi, S. Mosser, P. Matthews, A. Gericke, C. Ballagny, F. DâĂŹAndria, C.S. Nechifor, and C. Sheridan. Modaclouds: A model-driven approach for the design and execution of applications on multiple clouds. In *Procs. MISE 2012*, pages 50–56, 2012.

[2] A. Bessani, R. Kapitza, D. Petcu, P. Romano, S.V. Gogouvitis, D. Kyriazis, and R.G. Cascella. A look to the old-world sky: Eu-funded dependability cloud computing research. *SIGOPS Oper. Syst. Rev.*, 46:43–56, 2012.

[3] R. Buyya, R. Ranjan, and R. Calheiros. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *LNCS 6081*, pages 13–31, 2010.

[4] A. Celesti, F. Tusa, M. Villari, and A. Puliafito. How to enhance cloud architectures to enable cross-federation. In *Procs. 3rd IEEE Cloud*, pages 337–345, 2010.

[5] J. Ejarque, R. Sirvent, and R. Badia. A multi-agent approach for semantic resource allocation. In *Procs. 2nd CloudCom*, pages 335–342, 2010.

[6] A. J. Ferrer, F. Hernandez, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. M. Badia, K. Djemame, W. Ziegler, T. Dimitrakos, S. K. Nair, G. Kousiouris, K. Konstanteli, T. Varvarigou, B. Hudzia, A. Kipp, S. Wesner, M. Corrales, N. Forgo, T. Sharif, and C. Sheridan. Optimis: A holistic approach to cloud service provisioning. *Future Generation Computer Systems*, 28(1):66 – 77, 2012.

[7] N. Grozev and R. Buyya. Inter-cloud architectures and application brokering: Taxonomy and survey. *Software Practice and Experience*, pages in print, http://dx.doi.org/10.1002/spe.2168, 2012.

[8] T. Harmer, P. Wright, C. Cunningham, and R. Perrott. Provider-independent use of the cloud. In *Procs. Euro-Par'09, LNCS 5704*, pages 454–465, 2009.

[9] I. Houidi, M. Mechtri, W. Louati, and D. Zeghlache. Cloud service delivery across multiple cloud platforms. In *Procs. SCC'11*, pages 741–742, 2011.

[10] J. Kang and K.M. Sim. Cloudle: a multi-criteria cloud service search engine. In *Procs. APSCC '10*, pages 339–346, 2010.

[11] A. Li, X. Yang, S. Kandula, and M. Zhang. Cloudcmp: Comparing public cloud provider. In *Procs. 10th Conf. Internet measurement*, pages 1–14, 2010.

[12] N. Loutas, E. Kamateri, and K. Tarabanis. A semantic interoperability framework for cloud platform as a service. In *Procs. 3rd IEEE CloudCom*, pages 380–387, 2011.

[13] J. L. Lucas-Simarro, R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente. Scheduling strategies for optimal service deployment across multiple cloud. *Future Generation Computer System*, pages in print, http://dx.doi.org/10.1016/j.future.2012.01.007, 2012.

[14] A. Mandal, Y. Xin, I. Baldine, P. Ruth, C. Heerman, J. Chase, V. Orlikowski, and A. Yumerefendi. Provisioning and evaluating multi-domain networked clouds for hadoop-based applications. In *Procs. 3rd IEEE CloudCom*, pages 690–697, 2011.

[15] F. Moscato, R.Aversa, B. Di Martino, D. Petcu, M. Rak, and S. Venticinque. An ontology for the cloud in mosaic. In *Cloud Computing: Methodology, Systems, and Application*, pages 467–486. CRC Press, 2011.

[16] J. Nimis, A. Anandasivam, N. Borissov, G. Smith, D. Neumann, N. Wirstrom, E. Rosenberg, and M. Villa. Sorma - business cases for an open grid market: Concept and implementation. In *LNCS 5206*, pages 173–184, 2008.

[17] NIST. Cloud computing standards roadmap-version 1.0. In *Special 9 Publication 500-291*, 2011.

[18] M. P. Papazoglou. Cloud blueprints for integrating and managing cloud federations. In *LNCS 7365*, pages 102–119, 2012.

[19] F. Paraiso, N. Haderer, P. Merle, R. Rouvo, and L. Seinturier. A federated multi-cloud paas infrastructure. In *IEEE 5th International Conference on Cloud Computing*, pages 392–399, 2012.

[20] D. Petcu. Portability and interoperability between clouds: Challenges and case study. In *ServiceWave 2011, LNCS 6994*, pages 62–74, 2011.

[21] D. Petcu. A panorama of cloud services. *Scalable Computing: Practice and Experience*, 13:303–314, 2012.

[22] D. Petcu and J.L. Vazquez-Poletti (eds). *European Research Activities in Cloud Computing*. Cambridge Scholars Publishing, Cambridge, UK, 2012.

[23] D. Petcu, G. Macariu, S. Panica, and C. Craciun. Portable cloud applications - from theory to practice. *Future Generation Computer System*, pages in print, http://dx.doi.org/10.1016/j.future.2012.01.009, 2012.

[24] P. Riteau. Building dynamic computing infrastructures over distributed clouds. In *Procs. IPDPS'11*, pages 2097–2100, 2011.

[25] M. Tsugawa, A. Matsunaga, and J. Fortes. User-level virtual network support for sky computing. In *Procs. 5th IEEE e-Science*, pages 72–79, 2009.

[26] S. Venticinque, R. Aversa, B. Di Martino, and D. Petcu. Agent based cloud provisioning and management. design and prototypal implementation. In *Procs. CLOSER 2011*, pages 184–191, 2011.