On the Interoperability in Multiple Clouds

Dana Petcu
Computer Science Department, West University of Timişoara, Timişoara, Romania

Keywords: Interoperability, Multiple Clouds.

Abstract: Interoperability between Clouds is a desire of the last half decade to fulfill the dream of apparently unlimited capacity of the interconnected e-infrastructures as well as the elimination of the vendor lock-in. In this paper we try to point towards the missing pieces by identifying the requirements and the degree of their coverage by the existing solutions. Moreover, a particular case, of application interoperability, is studied using a concrete and recent deployable and open-source platform as a service.

1 INTRODUCTION

The need of using multiple Clouds has appear with the proliferation of various Cloud services. The simplest case, the Hybrid Cloud is motivated by the need to deal with the peaks in e-infrastructure usage. The migration for one Cloud to another is motivated by economic reasons. The usage of services from a partner Cloud is appealing also for the Cloud provider in order to deal with an unexpected increase of requests. Using simultaneously the services from multiple Clouds (in opposition with the previous two cases that are sequentially using different Clouds) is done due to the particularities of each service not encountered elsewhere. Such scenarios are pushing the technical developments from nowadays and we see already several improvements of the state-of-the-art in the last two years in what concerns the tools supporting the multiple Clouds usage.

In Spring 2009, OpenManifesto group (www.openmanifesto.org) has identified the five main challenges for the Cloud: (1) data and application interoperability; (2) data and application portability; (3) governance and management; (4) metering and monitoring; (5) security. Despite the considerable efforts in the latest four years in the field of Cloud computing, both in industry and research these five challenges are still persisting. Partially solutions or early prototypes were fortunately build in the last two years. However, complete solutions are not yet expected in the near future as several technical barriers have not been overcome. In this context, is the aim of this paper to point where the missing pieces are. To do this, we need first to clearly identify the requirements and the current solutions. Therefore the first sections are dedicated to fulfill this aim.

Interoperability is an issue for both Cloud provider (to extend its capacity and services) as well as for the Cloud consumer (to allow the freedom of movement between various Clouds). In the second part of the paper we take the position of the Cloud application developer (whose application will consume the Cloud services). In particular we discuss the case of the application interoperability, having in mind that application portability is partially solved by various solutions.

The contributions of this paper are the followings:

1. identify the key elements of the interoperability in multiple Cloud usage scenarios;
2. identify the current gaps in fulfilling the interoperability requirements;
3. provide a concrete example related to a recent open-source platform as a service for multiple Clouds, designed for portability reasons.

2 MULTIPLE CLOUDS AND INTEROPERABILITY NEEDS

Multiple Clouds are classified nowadays (e.g. in (Ferrer et al, 2012)) in Federated Clouds and Multi-Clouds.

In the first case, of Federated Clouds, agreements or contracts between Cloud providers are established in order to increase their services, to ensure the possibility to deal with peaks, to survive in case of dis-

Petcu D..
On the Interoperability in Multiple Clouds.
DOI: 10.5220/0004503105810590
In *Proceedings of the 3rd International Conference on Cloud Computing and Services Science* (CI-2013), pages 581-590
ISBN: 978-989-8565-52-5

asters. There are very few examples of such agreements at this moment in the case of Horizontal Federations (between peer Clouds), and multiple in the case of Vertical Federations (if a SaaS needs a PaaS, or a PaaS relies on a IaaS). The reduced number of such agreements are due to the request of a certain degree of control by a Cloud provider over the resources or services of another Cloud provider in agreement with the first. The degree of control is providing also a criteria for further classify the Federated Clouds. Beyond the barrier of the agreements, the next barriers are the technical ones, including interoperability.

The intensive studied topics for interoperability at Federated Cloud level are related to VM migration, API for communications and requests, automation, standards, and so on. These subjects are related to *run-time* stage of the life-cycle of services and applications: interoperability is expected to be applied during the execution of services or applications. Moreover, interoperability is a subject for Cloud provider, as the Cloud consumer should not be aware of the subcontracting.

In the second case, of Multiple Clouds, no agreement needs to be established. Instead a third party (beyond the Cloud provider and Cloud consumer) is offering services that are build on top of the several Cloud services, offering a unique entry point for several Clouds. As underlined in (Grozev and Buyya, 2012) the available tools to support Multi-Clouds can be classified in library-based or service-based. In the case of the libraries, uniform access to infrastructure (-as-a) services is ensured by considering the common denominator of the existing libraries. In the case of the services, according to the applications needs, a matching between the needs of the applications and the special offers of various Cloud services is done (an incipient form of a Cloud broker, not necessarily performing auditing or single entry point, trust measurements or monitoring).

The intensive studied topics for interoperability at Multi-Cloud level are related to automation of deployments, configuration of services, semantic processing and so on. These subjects are related to *design-time* stage of the life-cycle of services and applications: interoperability is expected to be supported by portability solutions. Moreover, interoperability is a subject for the third party (representing and even identified with the Cloud consumer).

An evolving step from the Federated Cloud or Multi-Cloud is considered to be the Inter-Cloud, an Federated Cloud or Multi-Cloud that includes a Cloud broker and offers dynamic service provisioning. Therefore it inherits the issues already mentioned above in what concern interoperability at design and run-time. The Inter-Cloud goal is according (Bernstein et al, 2009) to create an environment that supports dynamic expansion or contraction of capabilities for handling variations in demands (dynamic workload migration is possible). A high level architecture of the Inter-Cloud was recently proposed in (Demchenko et al, 2012).

The interaction between the Clouds can be synchronous (e.g. in the case of vertical federations) if direct calls are made, or asynchronous (in case of loosely coupling as in emergency scenarios). The interaction can be also take a synthetic form, when the communication and exchanges are using specific formats or protocols, or a semantic form, when the information exchanged is interpreted using a common information exchange reference model.

3 INTEROPERABILITY REQUIREMENTS

In a previous paper (Petcu, 2011) we have elaborate on the dimensions, levels and technological requirements of interoperability. We remind them here and elaborate further according to the last achievements in the field.

Interoperability is needed for at least for the following three reasons: (a) protection of the end user investments in developments; (b) development of a Cloud eco-system and market; (c) exploit full advantage of elasticity and pay-as-you concept.

The evolution of interoperability issues has take three stages, according (Williams, 2009): (1) migration, referring to the portability of VMs; (2) federation, targeting networking; (3) burst, targeting APIs.

Figure 1 is suggesting the three dimensions of the interoperability problem. Two are technical and encountered as pointed earlier, at run-time in Federated Clouds and at design time in Multi-Cloud. The third dimension is non-technical, more human oriented, as being related to policy, i.e. establishing agreements

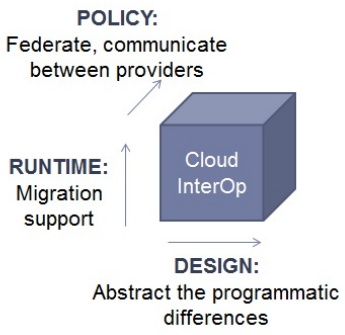

Figure 1: Interoperability dimensions.

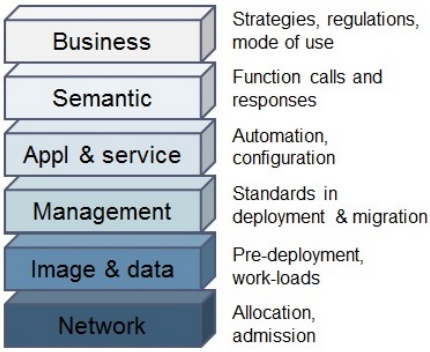

Figure 2: Levels of interoperability.

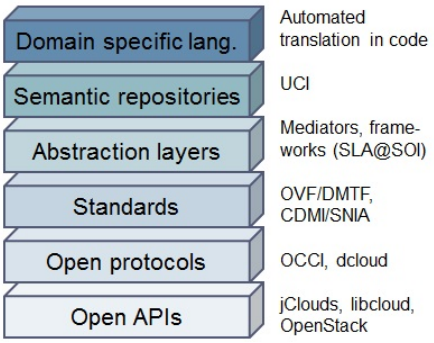

Figure 3: Technical solutions for interoperability.

and contracts between providers, as well as standards elaborations and promotion.

Figure 2 is proposing a split of the requirements in different categories (based on the initial classification proposed in (Khattak et al, 2010)). The top level is associated with the policy dimension; the next one with the design; the last three with the run-time.

The most complex level is the one related to application and services, that is covering both design and run-time. The requirements to this level were first discussed in (Merzky et al, 2009), in the context application-level interoperability versus the service-level interoperability: the first is considered to ensure a strong interoperability while the second one, a weak interoperability (note that the discussion was related to Grids and Cloud interoperability).

Table 1 is pointing to some of the requirements of the interoperability, associated with a certain level or dimension as exposed in the previous figures.

4 COVERAGE AND GAPS

Several technical solutions are available currently to support interoperability. Part of them are represented in Figure 3 and we have review them in (Petcu, 2011). The closest and recent snapshot of the interoperability solution was provided in (Loutas et al, 2011). We insist in this section more on gaps that should be filled, using few examples.

For the run-time dimension, at the infrastructure levels (last two in the table and image), several partial solutions exists to migrate virtual machines, virtual storage or services. Despite the presence of standard OVF format, the VMs are not yet ready for interoperability. For example, Amazon is one of the few Cloud providers who are allowing to export VMs; however, their related resources (e.g. network, storage) cannot be exported too. In general, VMs cannot be transferred from one hypervisor to another. VMs can be converted today with tools like qemu-image, but this requires to stop the VM and to apply the adaptation off-line.

We should mention the application perspective: also that once the application is deployed and adapted to a certain Cloud, in order to move it in another Cloud, an inspection of the source code is needed to identify the specific API calls or to build a model or representation of the code. Tools that can do that are only in early stage of prototyping (in European projects like Cloud4SOA, REMICS or MODAClouds) and not yet integrated in the PaaSs.

It should be also mentioned that the diversity of the APIs is natural, as each providers intends to offer something new or unique compared with other offers, in order to attract customers. The interoperability issue is therefore an issue for the management and governance levels where automation should be achieved as much as possible. The mix-in of services from different providers can be a strong argument in using such entry level instead a direct connection to only one provider. Therefore the management and service automation levels in multiple Clouds are the hot-spots of the development activities in the last two years (commercial solutions like RightScale, enStratus or Kaavo have emerge that are able to deploy applications in various Clouds, but not yet migrate the running ones).

For the design time dimension, several prototypes are available, from frameworks like SLA@SOI or the more recent (Di Modica et al, 2012), as well as APIs like jclouds, libcloud or OpenStack (a more complete list at (Lee, 2009)), or emerging standards like CloudML, but there is no wide acceptance of one or another proposal. Nor a Cloud specific programming model has emerge, despite the high-potential of the concept of e-infrastructure programming (Petcu et al, 2012b).

At the business level, a unified policy of the contractual terms was not yet established at national or international levels, while several proposals are on the

Table 1: Minimal requirements of interoperability at various levels.

Dimension	Level	No	Requirements
Policy	Public administration	P1	Regulations on contracts
		P2	Regulations on services level agreements
	Business	B1	Business strategies
		B2	Regulations on mode of use
		B3	Economic model driven optimization mechanisms
		B4	License flexibility
Design	Application/service abstractions	A1	Execution-unaware application support
		A2	Infrastructure independent programming
		A3	Common set of interfaces, standard APIs
		A4	High level modeling and programming models
	Semantic	S1	Message content and communication protocols
		S2	Service calls and answers
		S3	Cloud ontology and semantic-based discovery of services
Runtime	Service automation	R1	Re-configuration at run-time
		R2	Scaling in and out on multiple sites
		R3	Workflow management
		R4	Service and resource discovery, reservation and setup
		R5	Automatically provision resources in multiple Clouds
		R6	Optimal mapping of services/applications on resources
	Management	M1	Control the life-cycle of the application and its components
		M2	Standardized functionality for deployment in multiple Clouds
		M3	Migration support for application components
		M4	Plug-ins for the working environment
		M5	SLA and performance monitoring
		M6	Standards for allocation and admission control
		M7	Single sign-on for users accessing multiple Clouds
		M8	Auditing and trust mechanisms
	VM images, data and workloads	V1	Import and share VMs, data and workloads
		V2	VM, data and workloads migration support
		V3	Use standards for accessing compute and storage capacities
		V4	Support for multiple hypervisor technologies
	Network	N1	Uniform access to individual resources
		N2	Routing optimization based on monitoring
		N3	Software-defined networking

way. One of the hottest topic is the privacy and data protection compliance, for which no general accepted proposal is available yet.

A classical way to bust the interoperability is the adoption of standards and open source. Surely they are important mostly for the Cloud providers and service developers, not for the consumers. A classification of the standards at IaaS level was done in (Teckelmann et al, 2011) and refers to access mechanism, virtual appliance, storage, network, security and SLA; the paper also analysis the three oldest standard proposals, OVF, OCCI and CDMI from these point of views, identifying their gaps. Note that the three nominated standards have partially failed to be adopted on large scale by the providers, but are implemented in few Cloud management middlewares. New ones are emerging nowadays, like CIMI or CloudML, and several working groups are working to elaborate other proposals. Tables 2 and 3 are pointing to some of the standards respectively standard initiatives that are relevant for interoperability. An earlier list is provided in (Loutas et al, 2010). Comparing them, the tremendous changes from the last two years are evident. However there several gaps

Table 2: Standards, reference architectures, open group proposals.

Type	Standard	Link	Usefulness for interoperability
Standard	OCCI	occi-wg.org	It is designed to be a management API for IaaSs allowing the development of interoperable tools for common tasks including deployment, autonomic scaling and monitoring. OCCI is currently evolving into a general and complex system to support the multiple Clouds
	OVF	www.dmtf.org/standards/ovf	Describes virtual appliances for deployment across heterogeneous virtualization platforms, like the ones using different hypervisors, and allowing the users to deploy their virtual appliances in various Clouds
	CDMI	cdmi.sniacloud.com/	Used for data management and specifying how applications create, retrieve, update, and delete data from the cloud
	CIMI	dmtf.org/sites/default/files/standards/documents/DSP0263_1.0.1.pdf	It addresses the management of the lifecycle of Cloud infrastructure services. Basic resources of IaaS are modeled in order to provide to the consumer a management access to an implementation of IaaS. It focus on the portability between cloud implementations that support the specification. A REST-style protocol is proposed.
Architecture	IETF	tools.ietf.org/html/draft-khasnabish-cloud-reference-framework-01	Proposed a draft of intra-cloud and inter-cloud reference framework, documenting basic functions or layers to support the general requirements of Cloud services. The framework can be used to standardize the interfaces between the functions/layers.
	DMTF	www.dmtf.org/sites/default/files/standards/documents/DSP-IS0102_1.0.0.pdf	Cloud Service Reference Architecture describes key components, such as actors, interfaces, data artifacts, profiles and the interrelationships among these components.
Open groups proposal	The Open group	www.opengroup.org/getinvolved/workgroups/cloudcomputing	The Cloud Work Group within Open group collaborate on standard models and frameworks aimed at eliminating vendor lock-in for enterprises and looking to benefit from cloud products and services.
	CSCC	www.cloud-council.org	Has released a guide for SLAs that highlights the critical elements of a SLA for Cloud and provides guidance on what to expect and what to be aware of when negotiating an SLA
	OCC	opencloudconsortium.org	Has develop benchmarks for cloud computing with a particular focus in large data clouds: MalStone Benchmark is targeting large data clouds
	GICTF	www.gictf.jp	Promotes standardization of network protocols and the interfaces through which Cloud systems inter-work with each other. Has published: (1) Intercloud Interface Specification Draft (Cloud Resource Data Model); (2) Intercloud Interface Specification Draft (Intercloud Protocol); (3) Technical Requirements for Supporting the Intercloud Networking; (4) Use Cases and Functional Requirements for Inter-Cloud Computing

in the collections of available standards, like proposals for Cloud metrics and real-time monitoring, interfaces for security(-as-a-)services, accountability associated with transparency and responsibility. Furthermore, there are several barriers to the standard that were already pointed in (Machado et al, 2009).

As the multiple Clouds are based on the values of the Cloud services of various providers, thin layers build on top to ensure the functionality of Multi-Cloud of Federated Cloud are often open-source. Ta-

Table 3: Standard initiatives.

Organization	Link	Usefulness for interoperability
ETSI	www.etsi.org/technologies-clusters/technologies/grid-and-cloud-computing	Technical Committee for CLOUD is looking to interoperable solutions involving IT and Telecom industries, whit emphasis on IaaS. It focuses on interoperable applications and services based on global standards and the validation tools to support these standards.
IEEE	standards.ieee.org/develop/project/2301.html and 2302.html	IEEE started two development projects related to cloud interoperability, reflected in two working groups, P2301 and P2302. First group is working on standardizing Cloud portability and management, using a number of file formats and interfaces. The second group is focusing on Cloud-to-Cloud interoperability and federation.
ITU	www.itu.int/en/ITU-T/focusgroups/cloud/Pages/	Study Group 13 works on standards for next generation networks and future networks and in conjunction with Clouds (a Focus Group on Cloud computing has published a report).
NIST	www.nist.gov/itl/cloud/	Program to develop a set of cloud computing standards. First results were published as NIST Cloud Computing Program. NIST' special publications cover Cloud architectures, security, and deployment strategies for the US federal government
OASIS	www.oasis-open.org/committees/tc_home.php?wg_abbrev=id-cloud, www.oasis-open.org/news/announcements/symptoms-automation-framework-saf-v1-0-cs02-published, docs.oasis-open.org/tosca/TOSCA/v1.0/TOSCA-v1.0.html	Three cloud specific or extended technical committees have been founded: (1) Identity in the Cloud, looking to the security challenges posed by identity management in Cloud, identifying the gaps in current identity management standards, and investigating the need for profiles to achieve interoperability within current standards; (2) Symptoms Automation Framework, facilitating knowledge sharing and allowing consumer and provider to work cooperatively together to ensure adequate capacity, maximize quality of service, and reduce cost; (3) Topology and Orchestration Specification for Cloud Applications looking to enhance the portability of Cloud applications and services, and enabling the interoperable description of application and infrastructure Cloud services, the relationships between parts of the service, and the operational behavior of these services.
SNIA	www.snia.org/tech_activities/work/twgs	Cloud Storage Technical Work Group is developing an architecture related to system implementations of cloud storage technology
TM Forum	www.tmforum.org/DigitalServices/13907/home.html	Cloud Services Initiative intends to stimulate the growth of an open marketplace for Cloud services by promoting to large eco-system of enterprise customers, Cloud and technology providers the adoption of standards

ble 4 is pointing to the recent open-source solutions that are supporting multiple Clouds.

5 EXAMPLE

We consider in this section an example: how one of the latest open-source middleware for multiple Clouds is supporting the interoperability. We have pick mOSAIC, as being recent contributor to it. mOSAIC is providing an open-source and deployable Platform-as-a-Service that is supporting multi-Cloud scenario of re-deployment. The initial cycle of its development was supported through a collaborative and multi-national project, funded by the European Commission and ended in Spring 2013. Targeting open-source, its codes are available at the public repository https://bitbucket.org/mosaic, under Apache License.

Figure 4 indicates the main components of the latest stack of mOSAIC's software:

– red components, for the design phase: API libraries (for Java, Python and Erlang) with examples; application tools including plug-ins for Eclipse, templates and workflows; semantic support for developing applications, using domain specific and Cloud ontologies; a service-level agreement framework based on concepts introduced by SLA@SOI;

Table 4: Open-source middleware, framework, library, tool or service that is working with multiple Clouds and support a certain degree of interoperability.

Acronym	Link	Short description
Aoleus	www.aeolusproject.org	Cloud management software sponsored written in Ruby, open source, and provided by Red Hat on Linux systems. It allows users to choose between private, public or hybrid clouds, using δ-Cloud library
BonFIRE	portal.integration.bonfire.grid5000.fr/	Operates a Multi-site Cloud-based facility on top of infrastructure testbeds. Its Enactor shields the implementation details of how to interact with various testbeds from a resource manager.
ConPaaS	contrail-project.eu/conpaas	Provides a federation layer support for bringing Cloud providers together. It allows Cloud bursting. The Contrail Virtual Execution Platform interfaces with the IaaS layer of Cloud providers and upgrades the supported Cloud providers by adding SLAs features
δ-Cloud	deltacloud.apache.org	It is REST-based API written in Ruby necessary to connect to various Cloud providers (Amazon EC2, Eucalytus, SmartCloud, GoGrid, OpenNebula, RackSpace, OpenStack and others)
JClouds	www.jclouds.org	It is an open source Java library that introduces abstractions aiming the portability of applications. It support more than thirty Cloud providers and software stacks including AWS, GoGrid, vCloud, OpenStack, Azure.
LibCloud	libcloud.apache.org	Apache Libcloud is a standard Python library that abstracts away differences among multiple Cloud provider APIs
mOSAIC	bitbucket.org/mosaic	An API and a deployable PaaS that allows deployment, configuration management, and control of the life-cycle of applications or services consuming IaaS. Supports more than ten providers
Nimbus	www.nimbusproject.org	Introduces a virtual site layer for dynamically provisioned distributed resources of multiple data centers and a closed federation model where resources are shared based on cooperation
OPTIMIS	www.optimis-project.eu/Toolkit_v2	Is a platform for Cloud service provisioning that manages the lifecycle of the service and addresses issues like risk and trust management
SAGA	saga-project.github.com	API for managing e-infrastructures, from Grids to Clouds. Implementation of MapReduce using SAGA is proving its support for interoperability across Clouds and Grids.
SimpleCloud	www.simplecloud.org	It is a PHP library providing common interfaces for file and document storage services, queue services and infrastructure services
StratusLab	stratuslab.eu	It is an open source IaaS distribution that can be used for Cloud bursting

- green components, for the deployment phase: the brokerage system including the Cloud Agency that assists in the selection of the Cloud provider according to the application needs specified in a particular descriptor and generating the service level agreement, as well as vendor agents;

- blue components, for the runtime phase: support for a Personal Cloud (on a desktop using the Portable Testbed Cluster), for provider resource allocation based on existing credentials, the minimal kernel of the platform (mOS) to run remotely on providers' virtual machines, a naming service that offers an application virtual domain, and the execution engine that allows the control of the deployed applications; customized versions of open source Cloud technologies (for message queues, key value stores, distributed file systems) are available to be deployed (COTS) with corresponding drivers; special web-interfaces and console interfaces are allowing the control of the application life-cycle at the level of their components; benckmark sets are supporting the testing of applications and infrastructure services;

- purple components, the proof-of-the-concept applications.

Drivers and vendor agents are currently available

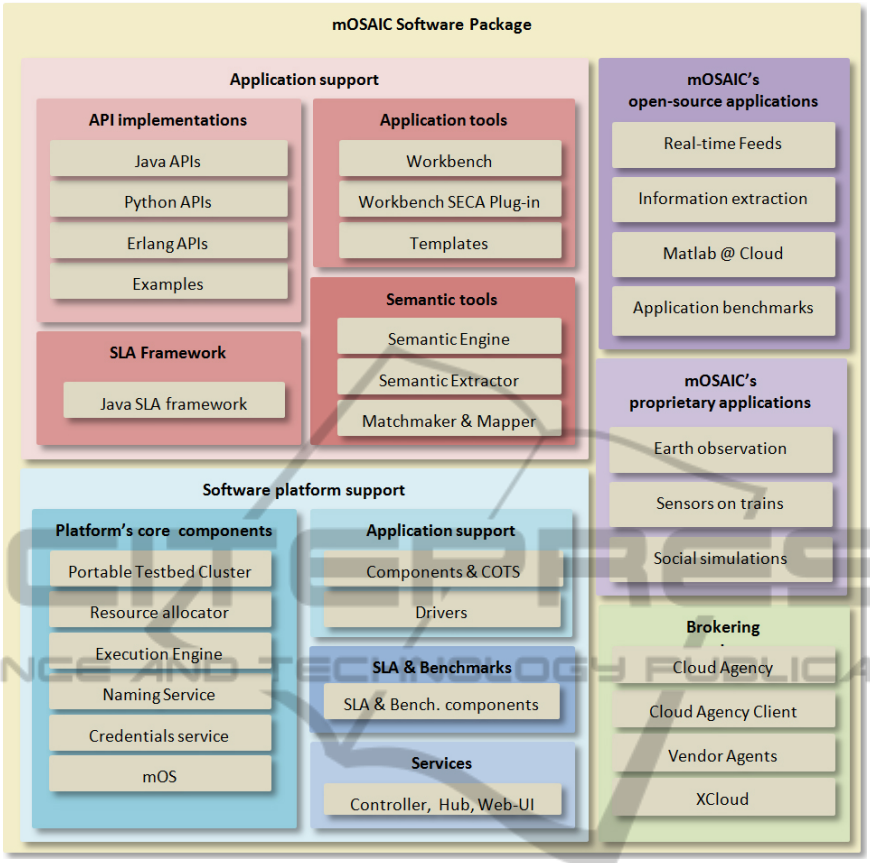

Figure 4: mOSAIC's software package: main components.

for various Public Cloud providers (Amazon, Flexiant, CloudSigma, GoGrid, OnApp, NIIFI) as well as for Private Cloud support technologies (Eucalyptus, OpenNebula, VMware, DeltaCloud, OpenStack, CloudStack).

mOSAIC is targeting mainly Cloud-enabled application portability – details can be found in (Petcu et al, 2012a; Sandru et al, 2012). The targeted applications are based on loosely coupled components that can be written in different languages and which are can be elastic (can be multiplied or reduced in the number of instances, at run-time).

The interoperability in multiple Cloud scenarios is only partially supported by mOSAIC, and is the aim of this section to detect the degree of coverage of the interoperability requirements as identified in the previous section. Table 5 is presenting the result of the analysis. Most of the requirements enumerated in the previous section (more than three quarters) are fulfilled. Not accidentally the PaaS is not able to offer solutions to several problems: auditing or trust mechanisms, optimized or inter-Cloud routing, sharing data between Clouds. These are subject of complex prototype platforms or intensive studies of other teams and projects (like TClouds for trust mechanisms).

Targeting to support the developers, less the providers, mOSAIC is oriented more towards interoperability solutions at design phase (application interoperability) and part of the run-time requirements related to the infrastructure (e.g. VMs and data migrations, or routing) are not yet served. However this status can be changed in the future in the further development stages, including the current on in the frame of MODAClouds project, in which support for migration of services and data is expected to be added (as well as support for the emerging standard CloudML).

Developed based on a strategy established three years ago, mOSAIC has not take advantage of the new standardization initiatives, incorporating only those that were available at its development moment. A refactoring of its codes as well as of the other close related platforms or resource management tools according to the new emerging standards specifications can raise the interest of the application developer community.

Table 5: mOSAIC support for interoperability on the technical dimensions (design and run-time)

No	Requirement	Support
A1	Execution-unaware application support	At the development phase the code is independent from the specific Cloud APIs; at the deployment phase the platform is entitled to use drivers to match various APIs of the selected Cloud provider
A2	Infrastructure independent programming	At the development phase, the code does not include any VM reference; the operations on the storage are referring only to a general type (e.g. key-value store)
A3	Common set of interfaces, standard APIs	Resources of same type have one interface in development stage
A4	High level modeling and programming models	Templates are used
S1	Message content and communication protocols	Asynchronous interaction, based on message queues, a standard protocol (AMQP) is used for communications
S2	Service calls and answers	In the case services are translated into the platform specific messages that are transmitted via message queues
S3	Cloud ontology and semantic-based discovery of services	A Cloud ontology has been build and new Cloud services can be detected and aligned
R1	Re-configuration at run-time	According to the behavior of the components, they can be multiplied/stopped during the execution
R2	Scaling in and out on multiple sites	Scaling is done by the platform at one deployment site (no cross-Cloud border mechanisms)
R3	Workflow management	Start the application components in a certain order
R4	Service and resource discovery, reservation and setup	A service discovery mechanism has been designed, the resources are configured automatically by the drivers
R5	Automatically provision services in multiple Clouds	The provisioning is done in the same way for various Cloud providers
R6	Optimal mapping of services/applications on resources	Cost optimality is targeted by the brokerage system
M1	Control the life-cycle of the application and its components	The application components can be started, stopped, multiplied during the application execution
M2	Standardized functionality for deployment in multiple Clouds	The deployment is done by a particular component of the platform, that is not Cloud dependent
M3	Migration support for application components	The components needs to be stopped and started remotely
M4	Plug-ins for the working environment	Eclipse plug-ins are available
M5	SLA and performance monitoring	An SLA framework was build, that is involving SLA brokerage and monitoring
M6	Standards for allocation and admission control	Use standards to logging
M7	Single sign-on for users accessing multiple Clouds	A credential service has been build to facilitate the user access to various Clouds (however it does not ensure single sign-in)
M8	Auditing and trust mechanisms	No support for auditing or trust mechanisms
V1	Import and share VMs, data and workloads	A special image at Cloud providers with the platform is available. No support for sharing workloads or data
V2	VM, data and workloads migration support	No support for migration
V3	Use standards for accessing compute and storage capacities	OCCI can be used in the brokerage and the resource control
V4	Support for multiple hypervisor technologies	The platform was tested on three hypervisors (no modification needed)
N1	Uniform access to individual resources	From the programming point of view, similar resources are addressed in the same way, only the types (e.g. key-value store, distributed file system, message queues) are different
N2	Routing optimization based on monitoring	No special support for routing optimization
N3	Software-defined networking	No special support for inter-Cloud routing

6 CONCLUSIONS

Interoperability in multiple Clouds is still an open problem. Considerable improvements where done in the last two years. However the subject is far from being closed. We have try to provide in this paper a snapshot of the state-of-the-art. A spot light was put on the current gaps in ensuring interoperability through technical solutions. Moreover, we proposed a list of technical requirements of the interoperability and in order to check the list we have pick one example of a current open-source and deployable platform-as-a-service for multiple Clouds.

ACKNOWLEDGEMENTS

This work was supported by the grant of Romanian National Authority for Scientific Research, CNCS UEFISCDI, PN-II-ID-PCE-2011-3-0260 (AMICAS).

REFERENCES

Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S., Morrow, M.: Blueprint for the Intercloud - protocols and formats for cloud computing interoperability. Procs. ICIW '09, IEEE Computer Press, 328–336 (2009)

Demchenko, Y., Makkes, M.X., Strijkers, R., de Laat, C.: Intercloud Architecture for interoperability and integration. Procs. CloudCom '12, IEEE Computer Press, 666-674 (2012)

Ferrer, A.J. et al, OPTIMIS: A holistic approach to cloud service provisioning, Future Generation Computer Systems 28, 66–77 (2012)

Grozev, N., Buyya, R.. Inter-Cloud Architectures and Application Brokering: Taxonomy and Survey. Software Practice and Experience, doi: 10.1002/spe.2168 (2012)

Khattak, A.M., Pervez, Z., Sarkar, A.M.J. Lee, Y.-K.: Service level semantic interoperability. Procs. IEEE SAINT 2010, IEEE CSP, 387–390 (2010)

Lee, C.A., An open cloud computing interface status update (and roadmap dartboard) Cloud interoperability roadmaps sessions, OMG Technical meeting (2009)

Loutas, N., Peristeras, V., Bouras, T., Kamateri, E., Zeginis, D., Tarabanis, K.: Towards a reference architecture for semantically interoperable clouds. Procs. CloudCom 2010, IEEE CSP, 143–150 (2010)

Loutas, N., Kamateri, E., Bosi, F., Tarabanis, K.: Cloud Computing Interoperability: The State of Play. Procs. CloudCom 2011, IEEE CSP, 752–757 (2011)

Machado, G.S., Hausheer, D., Stiller, B.: Considerations on the interoperability of and between cloud computing standards. Procs. OGF27: G2C-Net (2009)

Merzky, A., Stamou, K., Jha, S.: Application level interoperability between clouds and grids. Procs. GPC workshops 2009, IEEE CSP, 143–150 (2009)

Di Modica, G., Petralia, G.;, Tomarchio, O.: A Semantic Framework to Support Cloud Markets in Interoperable Scenarios. Procs. UCC 2012, IEEE CSP, 211–214 (2012)

Petcu, D.: Portability and Interoperability between Clouds: Challenges and Case Study. Procs. ServiceWave 2011, LNCS 6994, 62–74 (2011)

Petcu, D., Macariu, G., Panica, S., Craciun, C.: Portable Cloud Applications - from Theory to Practice, Future Generation Computer Systems, doi:10.1016/j.future.2012.01.009 (2012)

Petcu, D., Frincu, M.E., Panica, S., Neagul M.: Towards Programmatic Management of Services from Multiple Clouds. Procs. InCoS 2012, 487–488 (2012)

Sandru C., Petcu, D., Munteanu, V.: Building an Open-source Platform-as-a-Service with Intelligent Management of Multiple Cloud Resources. Procs. UCC 2012, 333-338 (2012)

Teckelmann, R., Reich, C., Sulistio, A.: Mapping of Cloud Standards to the Taxonomy of Interoperability in IaaS. Procs. CloudCom 2011, 522–526 (2011)

Williams, J.L.: An implementor's perspective on interoperable cloud APIs. Cloud interoperability roadmaps sessions, OMG Technical meeting (2009)