Semantic Interoperability in Context to Cloud Computing

A Technical Seminar Report
submitted for fulfilment of
the requirements for the
Degree of Bachelor of Technology
Under Biju Pattnaik University of Technology

Submitted By

Karisma Pradhan **Roll No. 201011269**

March- 2012

Under the guidance
of
Prof.Jaydip Sen

NATIONAL INSTITUTE OF SCIENCE & TECHNOLOG
PALUR HILLS, BERHAMPUR, ORISSA – 761008, INDIA

ABSTRACT

Cloud computing is a promising IT paradigm which enables the Internet's evolution into a global market of collaborating services. Cloud computing semantic interoperability plays a key role in making this a reality. Towards this direction, a comprehensive and systematic report of Cloud computing interoperability efforts by standardization groups, industry and research community is carried out. The main objective of this report is to derive an initial set of semantic interoperability requirements to be supported by existing as well as next generation Cloud systems. This report motivates and encourages the Cloud community to adopt a common Cloud computing interoperability framework with core dimensions the creation of a common data model and a standardized Cloud interface (API), which will constitute the base for the development of a semantically interoperable Cloud environment.

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **PROF. JAYDIP SEN** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards **Mr. Ranjit Behera** Faculty of **NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind co-operation and encouragement which help me in completion of this project.

April 10, 2013

Berhampur, Odisha, India

Karisma Pradhan

Roll No-201011269

B. Tech- CSE

Semester - VI

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Cloud computing affects and transforms current computer infrastructure and services. In just a few years, it managed to evolve into one of the most influential IT trends, reserving for second successive year the first place in Gartner's 2011 list of top strategic technologies . The novelty behind Cloud computing is that distributed physical resources, such as storage, CPU and networking (Infrastructure), programming frameworks and libraries (Platform) and services (Software) can now be traded as economic goods, integrated and offered on-demand through the Internet in a "pay-as-you-go" manner. Cloud computing interoperability problems arise when different Cloud providers try to co-operate exchanging data, applications and VMs. These incompatibilities can be either technical, e.g. incompatible virtualization implementations (VMware, Xen and KVM) or incompatible programming code (Java-based, PHP-based), or semantic. For instance, different Cloud providers use different modeling and notation for exposing the same features. There are several research initiatives focusing on Cloud computing interoperability. However, a systematic review of Cloud computing with emphasis on semantic interoperability is still essential. This paper provides a comprehensive report of the Cloud computing interoperability literature focusing mainly on the semantic aspects. We thus identify a set of primary requirements that should be first addressed by a common Cloud computing interoperability framework for solving semantic conflicts between heterogeneous Cloud systems. Motivation describes the methodology used to carry out this work illustrates the problem presenting definitions of Cloud computing semantic interoperability [1]. The key requirements for addressing semantic interoperability in Cloud computing are identified by analyzing the standardization initiatives and the frameworks working towards Cloud interoperability. Furthermore, several semantic interoperable Cloud solutions are investigated validating the requirements derived. Since 2009, the business world is concerned about the competition between Google, Microsoft and Apple on who will take the lead in the Cloud. This dominance becomes problematic if combined with the familiar risk of vendor lock-in, as the three major rival companies promote their own, mutually incompatible, Cloud standards and formats. Currently, many Cloud Platform as a Service (PaaS) providers make it very hard for software created by their customers to be moved off their platforms. According to the European Network and Information Security (ENISA)1, Cloud platform lock-in occurs at both the API (i.e. platform specific API calls) and the component level (i.e. a PaaS provider may offer a higher efficient back-end data store). Therefore, even if a compatible API is offered, the data may not be portable across PaaS offerings, as different data access models may exist [2].

2. MOTIVATION

Existing Cloud computing solutions have not been built with interoperability in mind. They usually lock customers into a single Cloud infrastructure, platform or service preventing the portability of data or software created by them [3]. Moreover, the battle for dominance between the big vendors, like Amazon, Google and SalesForce, makes them reluctant to agree on widely accepted standards promoting their own, incompatible formats . This dominance increases the lock-in effect and affects the competition since Small and Medium Enterprises (SMEs) deter from entering the Cloud market. The European Network and Information Security Agency (ENISA) and European Commission have recognized the vendor lock-in problem as a high risk that Cloud infrastructures entail . Interoperability is the missing element that will remedy this situation and benefit both Cloud customers and Cloud providers. In particular, in an interoperable Cloud environment customers will be able to compare and choose among Cloud offerings with different characteristics while they will switch between Cloud providers whenever needed without setting data and applications at risk. Moreover, an interoperable Cloud market will open up the IT industry to SMEs and strengthen their market position. They will interoperate and cooperate creating new business models according to demand without conflicts due to interoperability problems [4]. However, a danger lurks. Different (semantic) interoperability standards and frameworks can possibly lead to different interoperability solutions which are not interoperable between each other. Therefore, standardization bodies and researchers need to sit together and agree on a set of common principles that all interoperability solutions will adhere to.

3. METHODOLOGY

In order to conduct this report, related literature was reviewed using a systematic methodology . We started by searching the major research databases of computer science, i.e. ACM Digital Library, IEEE Xplore, SpringerLink, ScienceDirect and Google Scholar using keywords such as Cloud computing, semantic, interoperability, abstraction, architecture, etc. Initiatives coming from standardization bodies, industry, researching organizations/institutes and funded projects were also included in the report [4]. After briefly reviewing them we selected 48 publications as the most relevant. Afterwards these were organized depending on their content in four main strands:

1. Definitions of Cloud computing interoperability.
2. Standardization initiatives on Cloud computing interoperability.
3. Cloud computing interoperability frameworks
4. Semantically interoperable Cloud solutions.

4. CLOUD COMPUTING INTEROPERABILITY

This part reports the current State of the Art in Cloud computing interoperability. Section A presents the attempts to define Cloud computing interoperability focusing on semantic aspects. Initiatives from standardization bodies are discussed in sections 4.2. Section 4.3 describes Cloud computing interoperability frameworks which investigate the core directions in achieving Cloud computing interoperability and constitute the theoretical background of interoperable Cloud architectures and solutions. Such architectures and solutions are discussed in Section 4.4.

4.1 Definition

There are several attempts in the literature to scope, address and define Cloud computing interoperability. However, limited work is available in the literature defining Cloud computing semantic interoperability. Nevertheless, the semantic aspects of Cloud interoperability can be identified in several works. Hence, this section reports approaches working towards this direction. Interoperability in the area of Cloud computing means *"the ability to write code that works with more than one Cloud provider simultaneously, regardless of the differences between the providers"* As regards semantic interoperability , the differences pertain to the way two Cloud systems express and understand the same information. A common Cloud taxonomy and ontology can be the base for the development of such a common understanding between different Cloud systems according to Cohen, providing a common framework for describing Cloud "components" and their relationships [5].

According to Cerf , Tim Berners-Lee argues that semantically linking data may be *"the missing part of the vocabulary needed to interconnect computing Clouds"* and therefore solving Cloud interoperability problem. In particular, semantics of data and of the actions one take on the data and the vocabulary in which these actions are expressed appear to constitute the beginning of an InterCloud computing language. Moreover, the APIs will allow Cloud operation, management,\ assurance and governance. The creation of a common information space for data analysis, processing and exchange can be facilitated if the Cloud computing model is developed following semantic approach that focuses on semantic interoperability, modularity, statelessness and low coupling. Reviewing Cloud computing literature, we can deduce that there are several initiatives trying to address semantic interoperability through standardized data models and APIs. The semantic interconnection of Cloud systems is expected to significantly contribute to the elimination of the vendor lock-in problem and consequently to facilitate application approaches. Cloud computing interoperability and portability are closely related terms and often confused. Cohen clarifies these concepts saying that: "Cloud computing interoperability is the ability for multiple Cloud providers to work together or interoperate, whereas Cloud portability is the ability of data and application components to be easily moved and reused regardless of the provider, operating system, storage, format or API".

Interoperability between different Cloud PaaS providers involves the transparent portability of "the code", including data and applications. Moreover, interoperability may entail several additional features such as the management compatibility or the capability of seamless integration. However, the current definitions do not make a clear distinction between the different interoperability dimensions, i.e. technical referring to incompatible code or protocols, and semantic referring to incompatible data models and APIs' specifications.

4. 2. Standardization Initiatives

Standardization bodies, not-for-profit groups and member operated organizations are working on advancing Cloud computing interoperability standards, with the collaboration of academia and researchers, governments and vendors. The Distributed Management Task Force (DMTF) has introduced the Open Cloud Standards Incubator1, recently by the Cloud Management Working Group (CMWG)2, which aims at standardizing the interactions among Cloud environments. Moreover, the Open Virtualization Format (OVF)3, released by DMTF, describes a portable and efficient format for packaging and distribution of software to be run across multiple virtual machines. IEEE4 has introduced two working groups, named P2301 and

P2302. The first is responsible for developing a standard that will enable portability, whereas the second concentrates on allowing a system in one Cloud to work with a system in another. The Cloud Computing Interoperability Forum (CCIF)5 is planning to come up with a global Cloud computing ecosystem, where two or more Cloud platforms will be able to work together seamlessly. Key factors are the standardization of Cloud interfaces and the unified description of semantic Cloud data models. Meanwhile, the Organization for the Advancement of Structured Information Standards (OASIS)6 sees Cloud computing as an extension of Service-Oriented Architecture (SOA) and plans to develop Cloud models, profiles and extensions on existing standards to support Cloud security and interoperability. The Open Group Cloud Work Group7 aims to create a common understanding among buyers and suppliers, eliminating the vendor lock-in problem. The Open Cloud Consortium (OCC)8 will support the development of reference implementations, benchmarks and frameworks for interoperation between different Cloud providers. Moreover, a consortium of businesses launched by Intel, called the Open Data Center Alliance9, scopes to specify the future hardware and software requirements that lead to more open and interoperable Cloud and data center solutions. The Cloud Industry Forum (CIF)10 scopes to advance and advocate the adoption and use of Cloud-based services by businesses and individuals creating a marketplace while the growth of a marketplace is also the primary objective of TM Forum’s Cloud Services Initiative11. The Open Cloud Computing Interface (OCCI) from OGF is an example of a standard IaaS resource management interface interfacing IaaS Cloud computing facilities and allowing users interoperate using the same context. Similarly, Storage Networking Industry Association (SNIA) has produced the Cloud Data Management Interface (CDMI)13, an interface standard that enables interoperation with storage

Clouds and provides a standardized way to access all such services. The aforementioned standardization initiatives seem to rotate around three key enablers for tackling Cloud computing interoperability summarized in Table I. These are the adoption of (i) a standardized API/interface and a common management model,

(ii) a common data model and

(iii) the utilization of a marketplace/broker.

TABLE I. STANDARDIZATION ACTIVITIES IN CLOUD COMPUTING

Standardized API	**Cloude Model**	**Broker**
OGF/OCCI, CDMI/SNIA	DMTF/Opencloudstandards Incubator,GMTF/CMWG,DMTF/OVF,OASIS,open group cloud work group,IEEE P2301,IEEE P2302,CCIF,OCC,Data Center Alliance,OGF/OCCI	TM Forum/Cloud services Initiative,CCIF CIF

The aforementioned standardization initiatives agree that the creation of a common framework that will enable different Cloud platforms interoperate is of primary importance. To succeed this, two standardization approaches seem to gain popularity: the adoption of a standardized Cloud interface for the unification of multiple Cloud APIs and a common Cloud data model for describing with a similar way different Cloud entities.

4. 3 Cloud Computing Interoperability Frameworks

This section presents several Cloud computing interoperability frameworks which define the main directions for the investigation and advancement of Cloud computing inter-operability [6].

Hoff states that the Cloud community rotates around three ways in addressing Cloud computing interoperability: *service brokers, semantics and APIs*. Similarly, Cohen argues that Cloud providers can interoperate when they share *a common set of APIs* as well as a *consensus on the terminology/taxonomies* that describe them. Govindarajan and Lakshmanan report that besides APIs and brokers, interoperability should be investigated trough control, data and other additional issues, such as *policy management, security management* and *deployment /provisioning* aspects. Moreover, they propose to build relevant layers of *abstraction* to help interoperability and portability. Bozman and Chen present three key enablers towards seamlessly movement of a workload from one Cloud to another: *standardized APIs, layers of abstraction* and *management capabilities* that will enable manage physical and virtual servers in a unified way. Urquhart and Sambyal claim that there are only two interface points that PaaS and IaaS services need to standardize: the *management interfaces* and the *unit of delivery*. The management interfaces enable a wide variety of tools to monitor and manipulate the resources and services being offered while the "unit of delivery" includes the software to be hosted and any required supporting data, configuration, and policy required to allow that software to work. In the same context, Llorente identifies two interoperability levels: *management* and *service* interoperability. Jha et al. propose to summarize the exposed core capabilities and achieve interoperability with standardization at two levels: *at the infrastructure level (core capabilities) and at the Cloud interface level*. The semantics of a resource/service constitute the core capabilities along with the ability to manage these capabilities (provisioning, availability, QoS, security). Services expose their capabilities via interfaces and these interfaces can be accessed through APIs [7].

Mell and Grance propose a standardization model to address Cloud computing interoperability where Cloud capabilities fall into two categories: *core and advanced capabilities*. Core refers to portable features while advanced to proprietary capabilities. Furthermore, they argue that each Cloud model (IaaS, PaaS, SaaS) has its own specifications and need to be focused separately.

Endo et al. claim that APIs include *basic/core and additional functionalities*. The basic functions range from control of VM to programming primitives used to develop distributed applications in the Cloud, while the additional functions can include service quality assessment, load balance, elastic application growth and backup strategies. Cloud computing interoperability frameworks share a set of common characteristics and directions for achieving interoperability mainly utilizing semantic approaches. More specifically, state that a common, standardized API is of high importance to Cloud interoperability. This involves the standardization of core functionalities that will meet the basic needs of different Cloud provider and unify their heterogeneous APIs. Furthermore, Cloud 754 interoperability frameworks agree that a common understanding of Cloud stack is needed including common semantics for describing Cloud artifacts and the actions one take on these. Finally, a service broker and a layer of abstraction are also techniques used for tackling Cloud interoperability [8].

NIST has recently released the Cloud Computing Standards Roadmap which proposes the standardization of two main interfaces: the *self-service management API* and the *functional interface*. The management interface is used to control the usage of the Cloud services by starting, stopping, manipulating the virtual machine images and the associated resources. On the other hand, the functional interface refers to the content of the Cloud services enclosing the primary function of a Cloud offering, e.g. for an IaaS Cloud offering the functional interface is a virtualized CPU, the memory and input/output (I/O) space used by an operating system. Standardization gaps are identified in both interfaces and standards are needed to address application data and metadata, resource description and discovery as well as service management in both interfaces.

4. 4 Semantically Interoperable Cloud Solutions

This section presents architectures and solutions proposed to resolve Cloud computing interoperability. A mediation mechanism enabling connectivity among disparate Cloud providers is proposed in . The mediation mechanism captures the capabilities available from a Cloud provider infrastructure, logically groups and exposes them as standardized units. The mediation utilizes a resources catalog approach, defined using RDF and a common Ontology of Cloud Computing Resources. InterCloud Root providers act as brokers and host the Cloud Computing Resource Catalogs. The Infrastructure-as-a-Service Aggregator (IaaSA) enables different IaaS providers subscribe their resources, whereas it serves as a common interface for IaaS users to manage resources. IaaSA description model, based on DMTF Common Information Model (CIM)14, is generic enough in order to cater to all IaaS providers and provide operations to fulfill the IaaS user's requirements as well. In , the authors envisage a Cloud-agnostic middleware that can serve as broker for any Cloud client, without being tied to any Cloud provider. There are three main requirements to be addressed: management, quality concerns and agnostic interoperability (referring to frameworks, languages, and target). To achieve the requirements and to realize the architectural goals a meta-model applicable to any Cloud platform was created. This meta-model serves as an abstract representation of the APIs and all Cloud functionalities. A Blueprint concept is proposed in that provides a uniform abstract description of Cloud offerings. The Blueprint has 6 primary elements: the basic properties, the offering which contains the description of functional and non-functional offerings, the implementation Artifacts, the resource requirements, the architectural requirements and the invariants which are constraints that must not be violated by all elements of blueprint. By using templates for designing the blueprints, XaaS (IaaS, PaaS and SaaS) providers can seamlessly interoperate and collaborate. Sheth and Ranabahu propose a model in order to address the semantic conflicts of Cloud computing interoperability . Their semantic model has four dimensions: *functional, non-functional, data and system*. *Functional semantics* pertain to the core functionality (commonly referred to as the business logic) of the application while *non-functional semantics* refer to information which is not-directly relevant to the business logic such as access control and logging. The *data semantics* address the data aspects of an application such as definitions and relationships of data structures, while *system semantics* govern the system related concerns of the application which 14 http://www.dmtf.org/standards/cim become important when the application starts running.Virtual Machine Contracts is a platform designed to provide automated control and unified management of VMs by adding management metadata to the package in which VMs are stored and communicated . The authors describe how VMCs can be expressed as an extension to OVF. Furthermore, standardization of VMCs can offer uniform management even across environments with heterogeneous elements for enforcement. The interaction between service and infrastructure provider has also bee studied in the literature. The most common problem of this interaction is that each Cloud has its own proprietary mechanism for services definition. The authors propose to base the service definition on open standards, utilizing OVF as a service definition language. However, the OVF is not designed for Clouds so important issues for IaaS Clouds such as self-configuration, custom automatic elasticity and performance monitoring cannot be met. Therefore, they propose OVF extensions to achieve these goals. The Service-Oriented Cloud Computing Architecture (SOCCA) combines SOA and Cloud computing technologies in order to enhance Cloud interoperability. The Individual\ Cloud Provider Layer is the bottom layer where each Cloud provider builds its own data centers and provides its services. The innovation of SOCCA's resources is that they are componentized into independent services and accessed by the means of open-standardized interfaces. By this way, they can be combined with services from other Cloud providers. The Cloud

Ontology Mapping Layer is used in case extra features, not included in the standards, have to be implemented. The Cloud Broker Layer serves as the agent between individual Cloud providers and the SOA Layer which is responsible for publishing services in deployable packages. Some authors claim that it is difficult for heterogeneous Clouds to enforce semantics of virtual machine descriptions and user's requirements. In order to tackle this problem, the authors introduce a matchmaker architecture where end-users can discover the desired virtual machine from a range of providers and dynamically deploy it on different IaaS providers. The approach includes interesting novelties such as: user requirements' conversion to OVF to be a standard package format for Cloud development, an advertisement approach for IaaS providers and application of an ontology-based discovery to find the best suited providers. The aim of SITIO15 project is to create a new platform oriented towards interoperability and cost reduction. The SITIO architecture facilitates access to business and Cloud services from the perspective of a broker, based on a combination of the emerging IT trends, such as SaaS, Semantics, Process Modeling and Cloud computing . Moreover, it provides ontologies in order to semantically describe the available services and facilitate customers locating the desired applications [9].

5. FINDINGS AND DISCUSSION

In this report we studied the existing literature on Cloud computing interoperability focusing mostly on the semantic aspects. Thus, an initial set of key requirements that should be first included by a Cloud computing semantic interoperability framework derived. We observed that the Cloud community sets semantic interoperability between heterogeneous Clouds as a high priority for Cloud computing and utilizes semantic models and techniques for addressing Cloud interoperability. Moreover, the Cloud community agrees on the need to develop and adopt standardized methods and directions towards addressing Cloud computing semantic interoperability. Specifically, the standardization initiatives and the interoperability frameworks discussed in the previous section employ similar strategies for addressing semantic interoperability as the solutions and architectures discussed in the section IV.D. This is a rather interesting finding as it proves that more theoretical and more practical works seem to agree on the steps to be taken in order to develop semantic interoperable Clouds. There are two fundamental requirements that any newly developed or existing Cloud system should satisfy in order to resolve compatibility conflicts and cooperate with another Cloud system. A set of standardized Cloud models should be developed for describing the fundamental Cloud entities, e.g. service, computing resource, SLA etc. A standardized Cloud API should be created and supported by the Cloud vendors. The message to be conveyed is that re-inventing the wheel should be avoided, especially when it comes to the development of Cloud models and APIs. To facilitate interoperability, any Cloud system should describe its components including resources, services and APIs, based on a widely-accepted Cloud data model. Moreover, interactions and management of resources and services should be performed by means of a standardized API that will also be co-agreed by the community. However, in real life persuading the whole community to agree on and adopt common models may sound unrealistic or at least hard to achieve. Competition and the fact that differentiation is often considered a competitive advantage, organizational reasons and technical barriers prevent existing Cloud vendors from such decisions that can bring about a radical restructuring in their way of working. Thus, a more flexible approach is needed, namely a Cloud broker. The Cloud broker acts as a mediation mechanism that enables connectivity among heterogeneous Cloud systems reducing the need for reforming of the Cloud systems' establishments that would be required if the new common models were inherently supported. However, the Cloud broker still has to be based on a common Cloud data model and to provide a standardized API that will unify existing ones [10].

Table 2: CLOUD SEMANTIC INTEROPERABILITY APPROACHES

Requirements	
Standardized Description/Common Data model	Cloud Resource/Service model
	Cloud Meta model
Unified Management/Standardized API	
Cloud Broker	

The review of the related literature found that the Cloud community mainly utilizes semantic techniques for addressing Cloud interoperability, setting semantic interoperability as a high priority in addressing Cloud PaaS interoperability. In particular, there are two prevailing standardization approaches to achieve semantic interoperability at the PaaS layer: i) the standardization of a common data model for semantically describing any PaaS entity and ii) the standardization of a common management model for semantically annotating how PaaS entities will be managed. This also involves the adoption of a standardized API that will unify all different APIs (an API for all APIs) enabling any existing Cloud API to be mapped to this common API. To succeed this, the Cloud community should first agree on a common PaaS data model that will comprise of the basic PaaS concepts and their properties structured in an ontological schema. Any PaaS entity will be described based on this data model. This common understanding will ensure consistency and interoperability among different Cloud PaaS providers.

6. THE PaaS SEMANTIC INTEROPERABILITY FRAMEWORK

This section begins with the provision of a working definition for PaaS semantic interoperability. Hence, we define PaaS semantic interoperability as *the ability of heterogeneous Cloud PaaS systems and their offerings to overcome the semantic incompatibilities and communicate*. This refers to the ability of applications and their data to be seamlessly deployed on and/or migrated between Cloud PaaS offerings that are using the same technological background but different data (information) models and management interfaces (APIs). Next, we introduce a three-dimensional PaaS Semantic Interoperability Framework (PSIF) that aims to capture and represent any type of semantic interoperability conflict arising at the PaaS layer. At the same time it enables every semantic conflict to be mapped to the appropriate PaaS entity and the type of semantics. PSIF has been implemented by the Cloud4SOA project to resolve the semantic incompatibilities arisen both within the same as well as across different Cloud PaaS systems and enable Cloud-based application development, deployment and migration across heterogeneous PaaS offerings. In particular, PSIF is structured according to the following core dimensions.

i. **Fundamental PaaS Entities,** i.e. PaaS system, PaaS offering, management interface, software component, IaaS system and application.
ii. **Types of Semantics,** i.e. functional, non-functional and execution.
iii. **Levels of Semantic Conflicts**, i.e. information model and data.

A semantic interoperability conflict is raised when during the deployment of an application on a PaaS offering or during the migration of an application from one PaaS offering to another, the semantic models of any of the fundamental PaaS entities are incompatible. A semantic interoperability conflict may also be raised when two different PaaS systems try to exchange information. The first dimension (i.e. Fundamental PaaS Entities) allows us to locate where a semantic conflict is raised, i.e.

between which fundamental entities. The second dimension (i.e. Types of Semantics) allows us to identify the type of the semantic conflict that has occurred. Finally, the third dimension (i.e. Levels of Semantic Conflicts) allows us to identify the nature of the semantic conflict that has occurred. Information about all three dimensions needs to be collected in order to concretely define, fully understand and effectively treat a specific semantic conflict. Levels of Semantic Conflicts Fundamental PaaS Entities PaaS Offering ,PaaS System ,Data Level ,Types of Semantics ,Information-model Level ,Management Interface ,Software Component Application ,IaaS System Functional Execution.

6.1 Fundamental PaaS Entities

PSIF adopts a basic PaaS system model. The objective of this model is to recognize the fundamental PaaS entities and consequently investigate the semantic interoperability problems that relate to each of them. The PaaS model defines the following fundamental entities: the PaaS system, the PaaS offering, the management interface, the software component, the IaaS system and the application (Error! Reference source not found.). A PaaS system allows developers to develop and deploy their applications on one of the PaaS offerings that it offers using the management interface and the software components provided. Hence, a PaaS system may offer different offerings allowing developers select the one that best matches their needs, e.g. a Java-based, a PHP-based and a .NET-based offering. The semantic description of a PaaS system contains general information about the PaaS provider, such as name, location and contact details, as well as a list of the available PaaS offerings. A PaaS system possesses and publishes a number of PaaS offerings, i.e. platforms or programming platforms that Cloudbased application developers can use in order to develop, deploy and manage their applications. Indeed, they constitute subsystems of a PaaS system consisting of a management interface and a set of software components, while they use infrastructure provided by one or more IaaS systems. Moreover, a PaaS offering can host, depending on its capacity, more than one application. The semantic description of a PaaS offering contain basic information about the offering such as the name, the programming language, information about what it offers such a list with the software components and the IaaS systems and the management interface, as well as business information, e.g. details about the pricing.\ The functionalities of a PaaS offering are made available through a management interface. The management interface constitutes the point that enables a developer to communicate with a PaaS offering to manage the lifecycle of an application. It can range from a Web interface to an API. Thus, it contains a number of operations and for each operation the management interface specifies the information that should be sent to the system, the information that the system will send back and any error conditions that might occur.

6.2 Types of Semantics

PSIF defines three types of semantics that can be used for describing the fundamental PaaS entities: (i). The functional semantics capture the capabilities of a fundamental PaaS entity that is what an entity can offer or can do when it is invoked, e.g. operations, software components, management interfaces and development environment. According to literature, the functional semantics can also model data inputs and outputs, preconditions and effects of an entity. (ii) The non-functional semantics model the non-functional aspects of a fundamental PaaS entity including security, pricing, performance and other QoS related information. Moreover, they entail metadata that will allow the description of the basic properties of a PaaS entity like name, version, release data, etc. (iii) The execution semantics are used for specifying runtime and governance-related information of a fundamental PaaS entity (usually an application) that become important when it starts running or when it is invoked, such as information about deployment, management, discovery and orchestration, etc.

1. The **functional semantics** describe the core functionality of an application or else what an application can do including the operations, inputs, outputs and effects that an operation has. Moreover, they contain preconditions for software components, the management interface and the infrastructure provided by the IaaS systems.

2. The **non-functional semantics** refer to non-functional characteristics of an application such as basic properties e.g. the name, the developer, the release

date, the version, the programming language used, a short description of the application, etc. as well as security rights, performance and scalability considerations that should be met from the PaaS offering.

3. The **execution semantics** model information related to the application's deployment and governance, including deployment description, load balancing, dependency management and elasticity considerations (rules and automatic resource allocation). These semantics will allow an application to be seamlessly executed.

7. CLOUD COMPUTING SEMANTIC INTEROPERABILITY USE CASES

The Cloud Computing Use Case group has released a white paper highlighting the requirements that need to be standardized to ensure interoperability in most typical scenarios of interaction in Cloud computing. Based on these scenarios, a set of use cases have been produced emphasizing at PaaS layer and, especially, the interoperation between PaaS providers and Cloud-based application developers.

End-User to Cloud: Application Deployment on a PaaS offering : This scenario involves a developer who implements and deploys his application on a specific PaaS offering. Semantic conflicts might be raised if the developer attempts to deploy on a PaaS offering an application that he has developed in a different environment which used different models for describing the PaaS entities than those supported by the specific PaaS offering. Differences may exist in any of the functional, non-functional and execution semantics either at the information model or at the data level.

Changing Cloud vendors: Migrating an application to a different PaaS offering: This use case is realized when an application migrates to a different PaaS offering within the same PaaS system or across different PaaS systems. Moreover, it can entail the case of an additional PaaS offering supporting part of the application's running instance. For example, a developer, who has used in the code of his application a function, e.g. the 'connect a db' call to bind a database, is not able to migrate his application and seamlessly deploy it on another PaaS offering that is using a different management interface, e.g. the new interface is using a function called 'insert a db' to bind a database.

Hybrid Clouds: PaaS systems/offerings interoperation: This scenario refers to multiple Clouds working together for a common purpose. A hybrid Cloud can be delivered by (i) a federated Cloud provider that combines its own resources with those of other providers, and (ii) a broker with the difference that it does not have any

Cloud resources of its own. Focusing on the PaaS layer, this scenario encloses multiple PaaS systems working together as a federation for delivering a joint purpose. For xample, two PaaS systems decide to cooperate. However, their owners realize that the models that they use for describing the same PaaS offerings are incompatible. Assume that the first one describes as 'programming language' both the specific language supported as well as its version, e.g. Java Version 1.6, while the second uses 'programming language' and 'version' respectively. This semantic conflict is raised due to differences in the semantic models of the two PaaS offerings; more specifically to the way that their non-functional semantics are modelled. It is thus raised at the information model level, as it is caused by different logical representation for the same information. This semantic conflict can be resolved by means of a common PaaS offering model.

Enterprise to Cloud to Enterprise: Interoperation between: applications deployed on heterogeneous PaaS offerings: This scenario involves two enterprises located in different Clouds and, specifically, two Cloud applications which are running in different Clouds and need to interoperate for a joint purpose. In the PaaS context, this is translated to the ability of an application which is developed in a specific PaaS offering to interoperate with an application developed in a different offering. For example, two applications (A and B) running on two heterogeneous PaaS offerings need to collaborate for a common purpose. In particular, application A is initiated and application B is deployed when the performance of A exceeds a predefined value. To this end, application A records constantly its usage experience including the current performance, while application B constantly reads these data. The interoperation cannot be established in case the two applications use different representations of exposing the information related to their performance, e.g. application A refers to it as 'performance' while application B refers to it as 'CPU speed'. This semantic conflict is raised due to differences in the semantic descriptions of the two applications; more specifically to the way that their non-functional semantics are modelled.

8. GOOD PRACTICES FOR SEMANTICALLY INTEROPERABLE CLOUD PAAS

This section summarizes good practices for building semantically interoperable PaaS systems. Towards this direction, most opinions converge on the important role of common models, standardized management interfaces and standardized core and advance PaaS functionalities. To achieve semantic interoperability, PaaS architectures should support a semantic layer which will house all the semantic artifacts and the common models for describing PaaS offerings and applications hosted in a PaaS system. Moreover, PaaS architectures should structure their management interfaces' functions and calls based on a standardized interface adopted by the entire Cloud community. Towards this goal, the National Institute of Standards and Technology (NIST), proposes a standardization model where PaaS capabilities fall into two categories: core and advanced capabilities. Core refers to portable features e.g. deployment, management, monitoring, reporting, SLA management, metering/billing and provisioning services, while advanced to proprietary capabilities . More specifically, the detailed literature coupled with the experience gained during the development of PSIF, helped us compile the following list of recommendations:

i. Develop and use common PaaS models and standards as well as standardized management interfaces
ii. Use widely-accepted methodologies, technologies, development paradigms and best practices.
iii. Support and facilitate application portability. Developers should be able write applications once and deploy them on any PaaS offering.
iv. Support and facilitate data portability. Customers must be able to export their data in semantically interoperable formats that can be seamlessly migrated to other offerings.
v. The PaaS offering should provide the ability to include/embed/integrate other applications built on the same platform or others.
vi. Develop an application architecture with layers of abstraction to minimize access to proprietary models and capabilities.

vii. Develop ways to test and validate semantic interoperability and ensure that all elements can work together and interoperate as required and in conformance with known standards and certification.

9. CONCLUSION

This report provides insights on the essential aspects of semantic interoperability in Cloud PaaS environments. Given the rapid uptake and the great diversity of PaaS offerings, understanding semantic interoperability at the PaaS level is essential for supporting inter-Cloud cooperation and seamless information exchange. Achieving semantic interoperability is a fundamental requirement and a necessary precondition to be fulfilled in order to enable portability, i.e. the seamless migration of applications across different Cloud PaaS offerings. PSIF studies, models and contributes to resolving semantic interoperability conflicts that may be raised during the deployment or the migration of an application by defining a three-dimensional space comprising of: Fundamental PaaS Entities, Types of Semantics, and Levels of Semantic Conflicts. A set of guidelines and good practices were also collected and discussed, thus providing hints on how semantic interoperability can be tackled. The development of common PaaS models and standardized management interfaces is raised as a primary requirement in this context. PaaS architectures could then be augmented with a semantic layer that would host the common models and would be the link between heterogeneous PaaS offerings.

10.REFERENCES

[1] Gartner, "Gartner Identifies the Top 10 Strategic Technologies for 2011," in DOI: http://www.gartner.com/it/page.jsp?id=1454221, 2010.

[2] A. Sheth and A. Ranabahu, "Semantic Modeling for Cloud Computing, Part I & II," IEEE Internet Computing Magazine, vol. 14, pp. 81-83, 2010.

[3] J. McKendrick, "Does Platform as a Service have interoperability issues?," in DOI: http://www.zdnet.com/blog/service-oriented/doesplatform- as-a-service-have-interoperability-issues/4890, 2010.

[4] Economist, "Battle of the Clouds," in DOI. URL:

http://www.economist.com/opinion/displaystory.cfm?story_id=1464439 3, 2009.

[5] D. Catteddu and G. Hogben, "Cloud Computing-Benefits, risks and recommendations for information security," in DOI: http://www.enisa.europa.eu/act/rm/files/deliverables/cloud-computingrisk-assessment/, 2009.

[6] J. Bozman and G. Chen. Cloud Computing: The Need for Portability and Interoperability. IDC Analyze the Future. 2010.

[7] S. Jha, A. Merzky, and G. Fox. Using Clouds to provide Grids Higher-Levels of Abstraction and Explicit Support for Usage Modes. Concurrency abnd Computatioons: Practice and Experinec, Vol 21, pp. 1087-1108, 2009.

[8] R. Cohen. Examing Cloud Compatibility, Portability and Interoperability. Elastic Vapor: Life in the Cloud. DOI: http://www.elasticvapor.com/2009/02/examining-cloud-compatibility.html, 2009.

[9] N. Loutas, V. Peristeras, T. Bouras, E. Kamateri, D. Zeginis, and K. Tarabanis. Towards a Reference Architecture for Semantically Interoperable Clouds. In Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2010), Indianapolis, IN, USA, 2010, pp. 143-150.

[10] P. Mell and T. Grance. A Roadmap for Cloud Standards. In DOI: http://www.slideshare.net/kvjacksn/peter-mell-cloud-standards-20090915, 2009.