ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 5, October 2014

An Analysis of Cloud Interoperability Standards onVarious Service Models

SuchethaVijayakumar[1], LenitaVelezaSerrao[2], Hancel Anacletus D'Souza[3]

Assistant Professor, Department of IT, AIMIT, St Aloysius College, Beeri, Mangalore, India[1]

M.Sc (ST) I Semester, AIMIT, St. Aloysius College, Mangalore, India[2]

M.Sc (ST) I Semester, AIMIT, St. Aloysius College, Mangalore, India[3]

ABSTRACT: The cloud-computing community typically uses the term interoperability to refer to the ability to easily move workloads and the data from one cloud provider to another or between private and public clouds. Interoperability requires standard data models and communication technologies compatible with existing internet infrastructure. Typically, cloud of computing consists of the three cloud layers infrastructure (IaaS), platform (PaaS) and software (SaaS) as a service .In addition we can differentiate between (hardware or software) resources provided in a traditional way, and as a server version of them, which considers virtualization, multitenancy and elasticity as the main properties. However the lack of standards seems to be a drawback related to the interoperability. Convenient actions like changing clouds may be an arduous work for its customers .Therefore, this paper presents major consideration regarding the lack of cloud standards and pointing why this is considered to be a problem.This leads to a set of important observation towards a solution solving the interoperability and standardization problem. The paper also explores the role of standards in cloud computing interoperability. The goal of the paper is to provide basic insight into areas of cloud computing in which standards would be useful for interoperability and areas in which standards would not help or would need to mature to any value

KEYWORDS: Cloud computing, Service Models (IaaS, PaaS, SaaS), Interoperability, Standards

I. INTRODUCTION

Cloud computing has recently emerged as a new computing paradigm for organizing a shared pool of servers in data centers , which can provide on-demand computing resources (e.g., CPU, storage, network, database, applications and services) to users anywhere anytime, in a pay-as-you-go manner. Cloud computing provides scalable and cost-effective computing solutions,by rapidly provisioning and releasing resources withminimal user management effort and cloud provider interaction.According to the NIST definition of cloud computing , the cloud computing model is composed of five essential characteristics, on-demand self-service, broad networkaccess, resource pooling, rapid elasticity and measured service,and three service models, SaaS (Software as a Service),PaaS (Platform as a Service) and IaaS (Software asa Service). A cloud also supports variousapplications, including computation-intensive applications, storage-intensive applications , and bandwidthdemandingWeb applications .Every vendor promotes their own cloud infrastructure, and incompatible standards and formats to access the cloud, preventing them from agreeing upon a widely accepted, standardized way to support cloud applications. Therefore, the need for multiple clouds to beable to work seamlessly together, i.e., cloud interoperability, is rising.The importance of cloud interoperability has been highlighted by both the industry and the academiaIn this paper, we try to conduct a comprehensive survey on the state-of-the-art efforts to enable cloud interoperability, with a focus on interoperability among different IaaS (infrastructureas a service) cloud platforms. We investigate the existingtaxonomies, standards and implementation of

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 5, October 2014

cloudinteroperability, as well as practical cloud technologies (e.g.,nested virtualization) to enable interoperation. We also present issues and challenges in the topic area, and hope to make a way for the forthcoming research.

1. Cloud Service Models

Cloud provides three different types of Service Models as follows:

a)Infrastructure as a Service (IaaS): This cloud Infrastructure service deliver computer infrastructure such as platform, storage and networking. IaaS is the service model that would most benefit from standardization because the main buildingblocks of IaaS are workloads represented as virtual-machine images and storage units that vary from typed data to raw data

b) Platform as a Service (PaaS): This is the most complex of the three which delivers computational resources through a platform. The PaaS service model benefits less from standardization than IaaS. The platform provides many capabilities out of the box, such as managed application environments, user authentication, data storage, reliable messaging, and other functionality in the form of libraries that can be integrated into applications.

c) Software as a Service (SaaS): Saas is the most popular form of cloud computing and are easy to use. It uses the web to deliver the applications that are managed by a third party vendor and whose interface is accessed on the clients side. SaaS is a somewhat different model than IaaS and PaaS because it is a licensing agreement to third-party software instead of a different deployment model for existing resources that range from data storage to applications.Benefits of standardization for SaaS are even more limited than for PaaS.

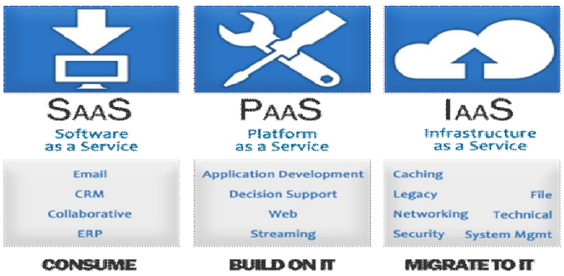

Fig 1: Various Cloud Service Models

II. WHY STANDARDIZATION?

Interoperability is needed when using multiple Clouds for at least three reasons:[9]

(a) protection of the end user investments in developments;

(b) development of a Cloud eco-system and market;

(c) take full advantage of elasticity and pay-as-you-go concept.

The problem is multi dimensional at design time and run time. The standardization bodies are taking individualor coordinatedactions. But the current solutions are solving the problem partially and the take-up level is low. One of the key characteristics that makes clouds different from usual enterprise computing is that the infrastructure can be programmable

.In enterprise computing environments, physical resources like servers, storage, and network connections should be deployed using human efforts most of times. Nevertheless, in the cloud computing manner developers can describe how these same components are virtually configured or interconnected, how the virtual infrastructure topology are formed, and how they interact to each other, depending on developing needs. In order to accomplish this virtual deployment and management, developers should manipulate an API that is implemented by the cloud provider.

However, cloud APIs are not yet standardized, and what happens nowadays is that each cloud provider has its own specific API for deploying and managing its services. In addition, each cloud provider has its own solution that tends to lock users into a specific technology.

Taking into consideration that aspects like agility, efficiency, and low associated cost are also key characteristics in cloud computing , the lack of standards definitely run against them. The reasons are that customers may change providers or combine them in a optimum manner – depending on its needs. In this case, the lack of standardization may bring disadvantages, when the moving, integration, or exchange of resources is required. The main negative aspect is the necessity of refactoring applications to comply with other cloud APIs, which can possibly lead to higher costs, delay of different natures, and risks – thus opposing agility, efficiency, and low costs.

An additional problem covers the necessity of integrated clouds, in the sense thatusers would move information/applications/servers from one cloud to another without losing functionality. Even for the most prominent problem inside the cloud interoperability subject is the lack of a standardized API, in this case it is also needed to plan for common mechanisms for pricing, accounting, and measuring various parameters across cloud boundaries.

2. Standard-Related Work for Cloud Computing

Many organizations are involved in various standardization efforts on the common theme of clouds. Notable among them are the working groups operating under the Open Grid Forum (OGF) umbrella. Other prominent industry consortiums active in cloud standardization are Distributed Management Task Force Inc. (DMTF) Storage Networking Industry Association (SNIA). In this section, we discuss three major open standards on cloud interoperability,which are widely adopted. These open standards help build bridges towards the goal of achieving user application and cloud provider interoperability. A significant progress has been made for pivotal elements such as storage, infrastructure management, and application description formats in the interoperability taxonomy, but much work remains to reach the final destination

OVF[6]

The Open Virtualization Format (OVF) is a packaging standard initiated by DMTF for deployment ofvirtual appliances. By abstracting a virtual appliance as a single atomic unit, it enables simplified and error-free deployment and migration of virtual appliances across multiple virtualization platforms, including IBM, Microsoft, Jump- Box, VirtualBox, XenServer, AbiCloud, OpenNode Cloud, SUSE Studio, Morfeo Claudia, and OpenStack. An OVF package contains multiple files in a single directory. The directory always contains an XML file called OVF descriptor with the VA name, hardware specifications and references to other files in the package. In addition, the OVF package typically contains a network description, a list of virtual hardware, virtual disks, certificate files, information about the operating system. DMTF advertises this format as vendor-neutral as it contains no reference to any vendor-specific information.

The OVF describes an open, portable, efficient and flexible format for packaging and distributing virtual appliances, with the following key features: (1) Enabling optimized distribution of virtual appliances; (2) Providing a simple, automated user experience, as it offers a robust and userfriendly approach to streamline the VA installation process. (3) Supporting both single and multi virtual machine configurations. (4) Enabling portable VM packaging, as OVF does not rely on any specific host platform, virtualization platform, or guest operating system. OVF addresses the following issues of Virtual Appliance in the cloud interoperability taxonomy.

• Life Cycle: With a standard format of VA, the five stages of VA life cycle can now be handled freely by the developers, due to the flexibility of the XML descriptor employed in OVF, which contains all the metadata of the VA. Cloud providers are also capable to build pre-configured VM images based on OVF, therefore allowing cloud users to better automate their service deployment.

• Virtualization Platform: The extensibility of OVF allows different virtualization platforms to be defined within a VA, whether it be OS-level virtualization, paravirtualization, or hardware-assisted virtualization, thus allowing cloud vendors with different virtualization platforms (e.g., AWS, Microsoft Azure) to interoperate

together.

• Virtualization Manager: OVF also provides a simple and unified interface for the virtualization managers to perform migration tasks within or across data centersof different cloud providers (e.g., libvirt.

CDMI

The Cloud Data Management Interface (CDMI) , proposed by the Storage Networking Industry Association (SNIA), defines the functional interface that applications can use to create, retrieve, update and delete data elements from a cloud. It is the storage backbone in cloud interoperability. CDMI features functions that (1) allow clients to discover the capabilities available in the cloud storage offering, (2) manage containers and the data that are placed in them, and (3) allow metadata to be associated with containers and the objects they contain. By abstracting the storage as containers which contain data objects, CDMI addresses the following issues on Storage of the cloud interoperability taxonomy.

• Backup: Backup is the most common operation for cloud storage, which is enabled by CDMI as a simple interface. Cloud users can schedule the backup or perform backup operations manually via the interface.

• Replication: CDMI containers and objects are mapped to a mounted file system's directories and files. This mapping allows their flexible replication in the lowlevel data storage cloud, even across different data centers.

• Snapshots: CDMI defines a snapshot as a point-intime copy (image) of a container and all of its content. A snapshot operation is requested using the container update operation. A snapshot may be accessed in the same way that any other CDMI object is accessed. An important use of a snapshot is to allow the contents of a source container to be restored to their values at a previous point in time, even across different cloud providers, using a CDMI copy operation.

OCCI[3]

Open Cloud Computing Interface (OCCI) is a boundary API that acts as a service front-end to

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 5, October 2014

anIaaS provider's internal infrastructure management framework. The OCCI community is an open community under the umbrella of the

Open Grid Forum (OGF), with contributing members from both the industry and the academia. OCCI was originally initiated to create a remote management API for IaaS model-based services, allowing for the development of interoperable tools for common tasks including deployment, autonomic scaling and monitoring. It has since evolved into a flexible API with a strong focus on interoperability while still offering a high degree of extensibility. It is compatible with the existing standards such as the OVF and the CDMI, and serves as an integration point for standardization efforts among DMTF, IETF and SNIA. The current OCCI specifications define a set of common standard interfaces of management for IaaS cloud interoperability. The documents are divided into three categories consisting of the OCCI Core, the OCCI Renderings and the OCCI Extensions.

• The OCCI Core defines a representation of instance types which can be manipulated through an OCCI rendering implementation. It is an abstraction of realworld resources, including the means to identify, classify, associate and extend those resources. It interacts with the renderings (including their associated behaviors) and can be expanded through extensions.

• The OCCI Renderings each describe a particular rendering of the OCCI Core model, i.e., how each OCCI Core model is rendered over the HTTP protocol thatleads to a RESTful API implementation. Multiple renderings can interact with the same instance of the OCCI Core model and will automatically support anyadditions to the model which follow the extension rules defined in the OCCI Core.

• The OCCI Extension each describe a particular addition to the OCCI Core model.

OCCI addresses the following issues on Access Mechanism, Network, Security and SLA of the cloud interoperability taxonomy.

What is the Solution?

Stepping towards a commonly and widely adapted solution means that there is a considerable amount of work and research to be performed. However, initially there are key observations collected, which cannot be ignored simply by an approach for developing a new interoperability standard.

• All cloud service providers, will possibly not agreewith an easy and standardized manner to export/import such cloud configurations. Obviously that cloud providers are not interested to give total freedom to their clients in an straight-forward manner in changing providers, and also do not want to open direct "competition" with other companies.

• Each cloud provider offers differentiated services, and want to have special "products" or services to attract more customers. A common standard may regulate them. However, the standard ensures under European regulation a core basic requirement for future software systems, since frameworks need to be accessible from competing vendors as well as customers.

Based on these inquiries, it can be derived that cloud computing needs a common and more basic approach of interacting with its offered services and resources. It does not mean that proprietary solutions will vanish or be limited. Basically, the first standardization shall address cloud core capabilities or cloud core features.These basic functions inside a cloud, like the management and provision of virtual resources, storage, or network components, reach a central importance level. Thus, on one hand, an extensive research among actual cloud providers has to be done to extract key and more interesting core functions. On the other hand, cloud providers will still have the opportunity to add proprietary functions, but not touching these cloud core capabilities and making it clear what are those proprietary values added on its APIs.

REFERENCES

[1]. Grace A. Lewis, The Role of Standards in Cloud-Computing Interoperability, October 2012
[2]. Claus Pahl, Li Zhang, Frank Fowley-Interoperability Standards for Cloud Architecture
[3]. OCCI Open Cloud Computing Interface (2012). Retrieved from: http://occi-wg.org/
[4]. OMG Object Management Group (2009). Cloud Interoperability Roadmaps Session. Retrieved from: http:// www.omg.org / news/ meetings/ tc/ ca/specialevents / Cloud Interop Roadmaps.htm

[5]. Open CSA - Open Composite Services Architecture (2012). Retrieved from: http://www.oasis-opencsa.org/
[6]. OVF Open Virtualization Format (2012.) Retrievedfrom: http://www.dmtf.org/sites/default/files/ standards/ documents/DSP0243 1.1.0.pdf
[7]. Considerations on the Interoperability of and between Cloud Computing Standards GuilhermeSperb Machado, David Hausheer, Burkhard Stiller-
[8]. SUN Microsystems: Introduction to Cloud Computing Architecture, White Paper, 1st Edition, June 2009. Available at: http://www.sun.com/featured-articles/CloudComputing.pdf.
[9]. The Need of Interoperability between Clouds, Dana Petcu, YvonJegou, Francesco D'Andria, Thomas Edwall,Alessandro Bassi, SilvanaMuscella, Beniamino Di Martino, Springer-Verlag Berlin Heidelberg 2011