State of the Art and Research Challenges of new services architecture technologies: Virtualization, SOA and Cloud Computing

Asma BEN LETAIFA, Amel HAJI, Maha JEBALIA, Sami TABBANE
Higher School of Communication of Tunis, SUP'COM, Tunisia
7th November University at Carthage
{ asma.benletaifa | amel.haji | maha Jebalia | sami.tabbane}@supcom.rnu.tn

Abstract

The telecommunication world is evolving towards networks and different services. It is necessary to ensure interoperability between different networks to provide seamless and on-demand services. As the number of services and users in Internet Protocol Multimedia Subsystem (IMS) keeps increasing, network virtualization and cloud computing technologies seem to be a good alternative for Mobile Virtual Network Operators (MVNOs) in order to provide better services to customers and save cost and time. Cloud computing known as an IT environment that includes all elements of the IT and network stack, enabling the development, delivery, and consumption of Cloud Services. In this paper, we will present the challenges and issues of these emerging technologies. The first part of this paper describes Cloud computing as the networks of the future. It presents an overview of some works in this area. Some concepts like cloud services and Service oriented Architecture designed to facilitate rapid prototyping and deployment of on demand services that enhance flexibility, communication performance, robustness, and scalability are detailed. The second part exposes SOA and its concept, the third one deals with virtualization.

Keywords- *cloud computing; services; SOA architecture; virtualization, IMS*

1. Introduction

With new generation networks, the principal element is IMS. Internet Protocol Multimedia Subsystem (IMS) is continuing to provide high quality solutions with new technologies for a rapidly changing service world. With the enhancement of on-demand services and systems management growth that the newer environments and cost measures require, better performances in terms of capacity, usability, and manageability are needed and seem to be necessary.

That's why, virtualization was proposed for cost and time saving [1]. In fact, it allows multiples MVNOs to share the same hardware through virtual machines (VM). However, the MVNO can evolve considerably and number of network users may become important.

As a consequence, the capacities of the VM that handle the MVNO components become insufficient to manage all users demand [2]. Thus, live migration seems to be a promising solution to deal with the lack of resources. Therefore, this paper aims at depicting the overall live migration techniques as well as the related motivations.

The current network architectures make difficult reusing and interoperability which provide considerable costs in managing the flow. These architectures do not take into account the changing functional requirements at the application development. Faced to costly development, redundant interconnections (point to point), a big complexity and difficulty to maintain connections, the notion of service and integration of services are recommended. Cloud computing and service oriented architecture are a very effective response to these issues in terms of reusability, interoperability and reduce coupling between different systems to ensure their cooperation. Among the many limitations that the current cloud platforms are facing today, the heterogeneity of resources and networks that form the clouds and the dynamic deployment and ondemand application of "cloud applications". The architecture of clouds networks must be flexible, scalable and expressive to ensure interoperability and a flexible and scalable coupling between different networks and heterogeneous services. To ensure flexibility, interoperability and scalability of the future's clouds, it is necessary to rely on methodologies and tools for dynamic composition of networks as is done today with dynamic composition of applications and autonomic service users. SOA (Service-Oriented Architecture - SOA) emerged recently, may represent a method to design generic enough for cloud networks.

Internet Protocol Multimedia Subsystem (IMS) is continuing to provide high quality solutions with new technologies for a rapidly changing service world. With the enhancement of on-demand services and systems management growth that the newer environments and cost measures require, better performances in terms of capacity, usability, and manageability are needed and seem to be necessary. That's why, virtualization was proposed for cost and time saving [1]. In fact, it allows multiples MVNOs to share the same hardware through virtual machines (VM). However, the MVNO can evolve considerably and number of network users may become important. As a consequence, the capacities of the VM that handle the MVNO components become insufficient to manage all users demand [2]. Thus, live migration seems to be a promising solution to deal with the lack of resources. Therefore, this paper aims at depicting the overall live migration techniques as well as the related motivations.

The reminder of this paper is organized as follow. In section 2, we will try to introduce IMS, its basic principles and purposes. Then an overview of Cloud Computing will be explained. The forth section deals with SOA architecture. Network Virtualization technology will be presented in section 5. After that, the live migration will be described in this section. Here, we will introduce issues and challenges of these concepts and gives comparative table for cloud technologies and some ideas about mixing together two or more of these concepts. To the best of our knowledge, no other paper summarizes all these technologies.

2. IMS Overview

The IP multimedia Subsystem (IMS) is a network functional architecture that facilitates multimedia service creation and deployment in Next Generation Networks (NGN), as well as supporting interoperability and network convergence [3]. It provides a common platform for different access technologies like WiFi, Wimax, and DSL and aims to supply an open, standard based network that grants a wide range of multimedia services. In [2], they emphasize the fact that the IMS architecture has been designed to clearly separate the connectivity, control and service plane. This separation allows a fast evolution of the architecture by handling different access networks. Figure 1 shows that an IMS system serves as a congruent component to allow different access technologies reach a common application service platform.

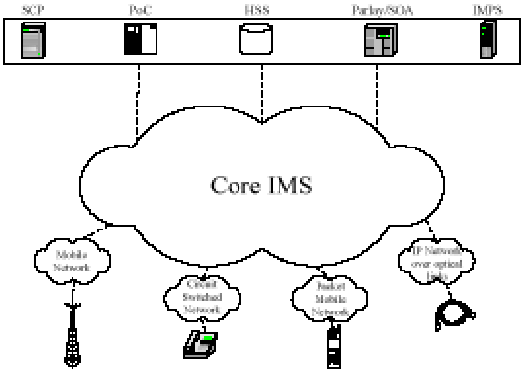

Figure 1. IMS: A common platform for multimedia applications

The IMS is being developed as the next generation core architecture for converged voice and data services. While it was originally intended for 3GPP wireless, IMS has evolved to include multiple access technologies and is the basis for planned converged network services including both wireless and wired access.

The three primary goals most often cited for the IMS are Quality of Service (QoS), Charging and Billing, and Integration of Services. In short, the first two goals derive from the fact that modern networks are primarily based on the packet switched (e.g. IP) protocols that provide only a best

effort service. Thus, most VoIP applications cannot provide any guarantees as to the user experience nor can they provide fine-grain charging and billing interfaces for the network provider. An important objective of the IMS standards is to create an architecture for deploying VoIP applications that provides for both QoS and charging. The third goal of IMS is to provide an architecture for efficiently integrating multiple different services, that can be easily mixed and matched to meet the user's needs. The IMS is designed to eliminate the need to create the traditional stovepipe applications that must include all features in a single application and do not easily integrate with other applications. Instead, the IMS supports the notion of combinational services such as Presence, Location, and Push-to-Talk that can be leveraged for new applications.

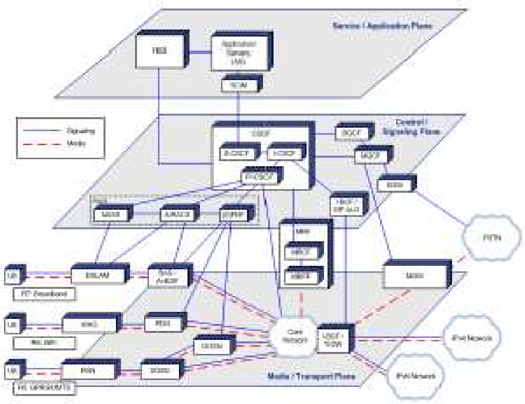

Figure 2. IMS architecture

The IMS architecture is depicted in Figure 2. It is divided into three layers: transport layer, session control layer, and application layer. The application layer includes application servers and data bases and provides service logic for users.

The session layer includes such functional entities that provide connectivity between users and applications. This layer provides policy decision, routing, and subscriber management functions. SIP servers and connectivity gateways are the main entity of this layer. The transport layer (or connectivity layer) is responsible for the end user's basic access functions and rules as a connector between IMS core and users [4].

3. The Cloud computing solution

Cloud Computing is a paradigm that focuses on sharing data and computations over a scalable network of nodes. Examples of such nodes include end user computers, data centers and web services. We term such a network of nodes as a cloud [9]. It is an emerging computing paradigm. It aims to share data, calculations, and services transparently among users of a massive grid.

Cloud computing is the next generation platform that provides dynamic resources, virtualization and high availability. Cloud computing is not associated with a particular technology, protocol or provider. In practice, applications and data are no longer on the local computer but in a "cloud" composed by a number of remote servers and left interconnected using bandwidth. Cloud computing describes a new supplement, consumption and delivery model for IT services based on Internet, and it typically involves the provision of dynamically scalable and often virtualized resources as a service.

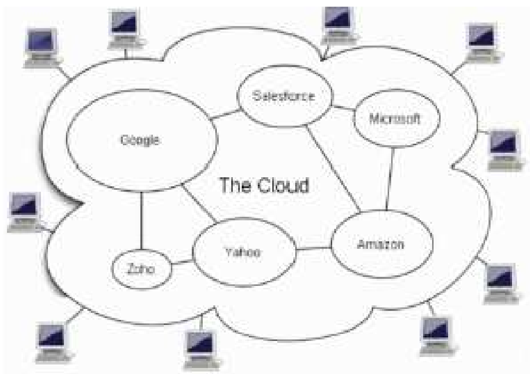

Figure 3. Cloud Computing Logical Diagram

The goal of Cloud Computing is to share resources among the cloud service consumers, cloud partners, and cloud vendors in the cloud value chain. The resource sharing at various levels results in various cloud offerings such as infrastructure cloud (e.g. hardware, IT infrastructure management), software cloud (e.g. SaaS focusing on middleware as a service, or traditional CRM as a service), application cloud (e.g. Application as a Service, UML modeling tools as a service, social network as a service), and business cloud (e.g. business process as a service). The character of cloud computing is in the virtualization, distribution and dynamically extendibility. Virtualization is the main character. Most software and hardware have provided support to virtualization. We can virtualize many factors such as IT resource, software, hardware, operating system and net storage, and manage them in the cloud computing platform; every environment has nothing to do with the physical platform. Carries on the management, the expansion, the migration, the backup through the hypothesized platform, all sorts of operations will be completed through the virtualization level.

Cloud computing offer:

- Ubiquitous network access
- Location independent resource pooling
- Rapid elasticity
- Self Service and Instant-On
- Elasticity and Pay-as-you

3.1 Cloud Computing: Type and Services

3.1.1 Types of cloud

Three types of cloud could be presented: Public cloud, Private cloud and hybrid cloud.

Public cloud: the services are delivered to the client via the Internet from a third party service provider

Private Cloud: these services are managed and provided within the organization. There are less restriction on network bandwidth, fewer security exposures and other legal requirements compared to the public Cloud.

Hybrid Cloud: there is a combination of services provided from public and private Clouds[20].

3.1.2 Models of Service

We find in littrature Evrything as a Service (EaaS). EaaS is the concept of reusable component called across network. It's a subset of cloud computing. "as a Service" was been associated with others functions such as communication (CaaS) or data (DaaS). Three models of service are the most used.

Infrastructure as a Service (IaaS): This model is a modern form of utility computing and outsourcing. IaaS can manage computer resources (networking, storage, virtualized servers). This model allows consumers to deploy and manage assets or leased server instances, while the own service providers govern the underlying infrastructure.

Platform as a service (PaaS): It facilitates the development and deployment of applications without the management of the underlying infrastructure, by providing all necessary equipment to support the entire life cycle of construction and delivery of Web applications and services. This platform consists of software infrastructure, and typically includes a database, middleware and development tools. This type of service typically operates at a high level of abstraction. Users can manage and control resources that they deploy in these environments. Service providers maintain and govern application's environments, server instances, and the underlying infrastructure.

Software as a Service (SaaS): The hosted software or applications are consumed directly by users. Consumers control only the way in which they use cloud services while service providers maintain and manage software, data and the underlying infrastructure [10].

3.1.3 Actors of cloud computing

Cloud computing [22] can involve 3 actors: vendors, developers and end-users like illustrated in the figure below:

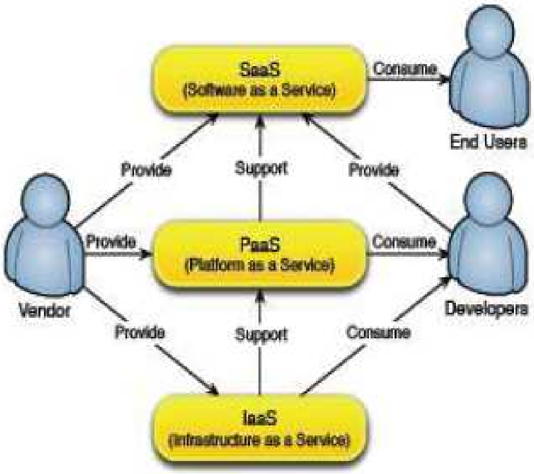

Figure 4. Actors of cloud computing

We include the vendor as a resource provider. Developers use the resources provided, the strengthening of services for end users. This separation of roles defines the divergent interests of stakeholders. However, actors could play multiple roles; providers can develop services to end users and developers can use the services of others to build their own services. However, every cloud in the role of provider, and therefore controller, can be occupied by the vendor providing the Cloud.

3.1.4 Layers of cloud computing

Andy Mulholland et al. detail the architecture of cloud computing [21] as shown in Figure 5.

Figure 5: Cloud Computing Architecture by GeoffreyRaines

Authors divide the cloud into 3 layers:

Cloud computing power: the computing power and resources that enterprises need to deliver a service or application, including storage, processing, networks and servers

Cloud Platform: Platform as a Service. It's a Platform that allows adding specific services.

Cloud services: includes the services of clouds themselves.

Other works [23] propose to represent Cloud Computing Architecture in 6 layers as shown in Figure 6.

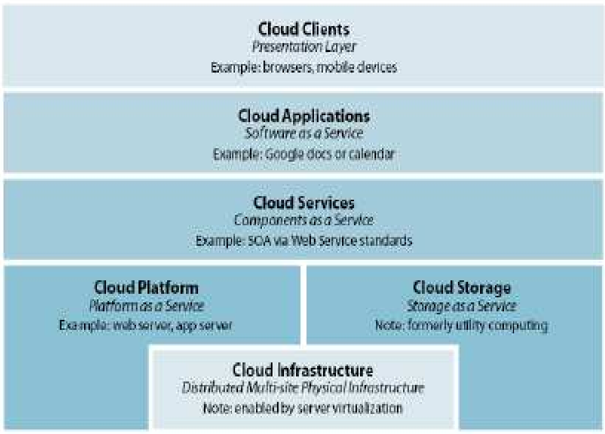

Figure 6. Cloud Computing Architecture

So, authors divided the cloud computing power layer into Cloud Infrastructure and Cloud Storage and they add two layers of upper level: Cloud Applications and Clients which are usually combined in layer Cloud Services [23].

Cloud infrastructure: Distributed physical components to support cloud computing, (storage, processing resources)

Cloud storage: Incremental renting of storage on the Internet (Utility Computing)

Cloud platform: Platform as a Service; infrastructure for developping and operating web-based software applications

Cloud services: Definition of software components, managed in a distributed fashion, across the commercial Internet. It's like SOA, with defined service interfaces as a basis for system-to-system integration.

Cloud applications: Software As A Service; hosted a set of services and software on the web Cloud clients: Distribution of business and personal data across servers on the Internet.

3.1.5 Models of services

Software as a Service (SaaS): Applications are available and delivered as services accessible via the Internet [21].

Platform as a Service (PaaS): Hosted application environment for building and deploying cloud applications.

Infrastructure as a Service (IaaS): Utility computing data center providing on demand server resources.

These models must be deployed at Cloud infrastructure which has the main characteristics as shown in the figure below:

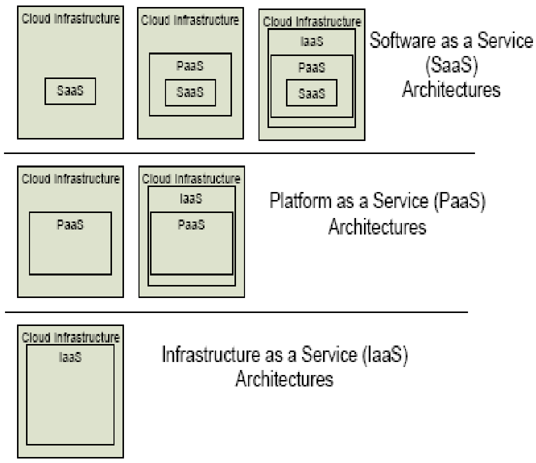

Figure 7. Models of services

We find in literature [24] Evrything as a Service (EAAS, ASA *) is the concept of a reusable component called across a network. It is a subset of cloud computing. "As a service" has also been associated with other functions such as communication, infrastructure and data. Network as a Service, Data as a Service, Communication as a Service. Network as a Service Naas: provides the overall system capacity.

Data as a Service Daas: softaware as a service or web service that provides customers access and analysis of a set of aggregated data.

Communication as a Service Caas: type of outsourced solution for business communication where a third party (known as the seller) is responsible for managing the hardware and software required for delivering voice over IP (Voice as a Service), instant messaging, and video conferencing applications using fixed and mobile devices.

3.1.6 Security in the cloud

Nuno Santos et al. concerned with issues of confidentiality and integrity of data provided to customers. To make the treatment of confidential virtual machines, they propose designing a platform of trust cloud computing: Cloud Trusted Computing Platform (TCCP). TCCP activates IaaS to provide a closed box environment to ensure confidentiality and integrity of the VM to a user, and allows a user to determine whether or not the IaaS applies these properties [25]. This software sollution allows a trusted VMM ensures that VMS only run on nodes with Trusted VMM and a security perimeter. The proposed trust will prevent consideration of a VM guest and guard featuring a small TCB interface close to the launch and migration is to a destination trusted.

3.1.7 Services delivred in the cloud

Jyoti Namjoshi et al. propose the design of a software solution for the provision "as a service” shared on Cloud. They describe an approach to book travel business based on Service oriented used by business travelers. They focuse their solution on Architecture Software-as-a-Service and Cloud Computing Paradigms for benefits of these concepts. To meet the requirements of their scenario, they suggest a SaaS and Cloud-based online, on request, Travel Reservation (Software as a Service TRSA) charged by monthly subscription in addition to use pay-per-employee basis of undertaking. TRSA provides a service abstraction, modularity, high cohesion, low coupling, efficient booking process, accessible at any time from multiple locations and heterogeneous technology environments [26] P.K. McKinley et al propose a distributed infrastructure designed to facilitate rapid prototyping and deployment of services. This infrastructure supports the dynamic instantiation and reconfiguration of services (by combining middleware functionality and a scalable overlay network substrate). Their architecture is based on two modules developed before by same authors: Application-Middelware eXchange (AMX) provides interfaces and encapsulates the logic high level to drive the services of various overlay and Kernel-Middleware eXchange (KMX): A set of interfaces and services that facilitate collaboration between the middleware and OS [27].

3.2 COMPARISION OF SEVERAL CLOUD’S SOLUTIONS

In this section we describe some platforms of cloud, and then we summarize in a comparative table some characteristics as shown in Table 1.

3.2.1 Eucalyptus

Eucalyptus for "Elastic Utility Computing Architecture for LinkingYour Programs To Useful Systems" is an open source software to implement Infrastructure as a Service in the cloud. The architecture of eucalyptus is simple, flexible and modular with a design hierarchy as shown in Figure 8. Three essential components of eucalyptus [11]:

Cloud controller queries information about resources from node managers, makes the scheduling decisions and executes them by using the cluster controllers.

Cluster Controller collects information about a set of virtual machines and schedules their execution on specified node controllers.

Node controller is running on each node that is designated to host virtual machine. It manages the implementation, inspection, and the termination of the VM on the host where it runs. Users have the ability to execute and monitor virtual machines deployed throughout the physical resources in a flexible, portable, modular and easy manner.

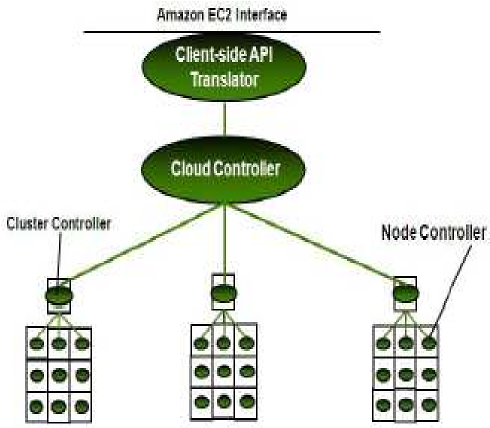

Figure 8. Eucalyptus's Architecture

The design of eucalyptus gives users the flexibility to seamlessly move applications to Eucalyptus on-premise on the public cloud, and vice versa. Eucalyptus also makes easy the deployment on the hybrid cloud, using resources from public and private clouds for the unique advantages of each [12]. The disadvantage is the lack of an interface to manage virtual machine and an advanced monitoring.

3.2.2 Open Nebula

OpenNebula is an open source manager of virtual infrastructure [13], able to build private, public and hybrid clouds. OpenNebula offers flexible architecture, interfaces and components that could be integrated into any data center. This tool supports Xen, KVM and VMware and access to Amazon EC2s.

OpenNebula was designed to be integrated into any network and storage solution. OpenNebula manages the storage, networking and virtualization technologies to enable the establishment of dynamic multi-level services (groups of interconnected virtual machines) on the distributed infrastructure, combining the resources of physical machines and cloud distance, based allocation policies OpenNebula consists of three components [14]: core (Virtual Infrastructure Manager): Manages the lifecycle of the virtual machine by running the basic operations (deployment, monitoring, migration).

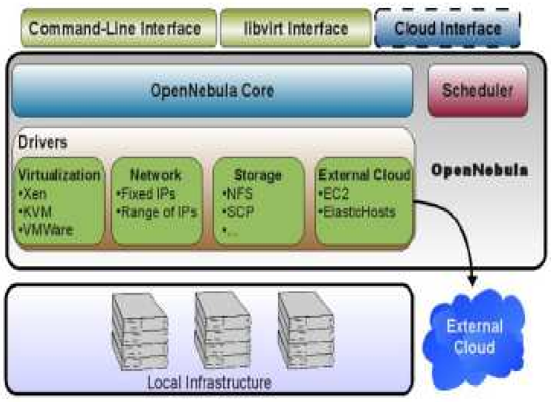

Figure 9. OpenNebula's architecture

Capacity Manager (scheduleur) module that governs the functionality provided by the core of OpenNebula: workloads balancing in virtual machines.
Virtual Access Drivers: virtualization layer. The drawback of openNebula is the lack of GUI (Graphic User Interface) OpenNebula provides the "load balancing" across nginx as shown in figure 10.

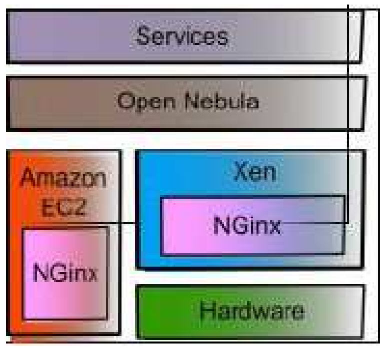

Figure 10. Load balancing in Open nebula

Some advantages of OpenNebula [15] like its is shown in figure 10:
- Centralized management of the balance of the workload "load balancing"
- Server Consolidation
- Resizing dynamic infrastructure
- Partitioning Dynamic Clustering
- Support for heterogeneous workloads
- Supply virtual machines on demand

3.2.3 Nimbus

Nimbus is an open source toolkit that provides "infrastructure-as-a-Service."It allows a client to lease resources in distance by deploying virtual machines (VMs), for building a desired environment [16]. Nimbus Cloud Storage offers the model pay-as-you-go and scalability (scale up and scale down as needed without adding expensive infrastructure). It allows customers to reduce costs and eliminate the task of management, giving them the freedom to focus on their business.
Nimbus allows providers to build clouds: Private clouds (Workspace Service: Open source implementation EC2) and developers to experiment with clouds: research or the use / performance improvements and contributions. It requires certain dependencies are installed first. On the service node: Java (1.5 +) and bash. On the nodes of the hypervisor: Python, bash, ebtables, dhcpd, and KVM or Xen libvirt. It supports both interfaces EC2 (Elactic Computing Cloud) and WSRF (Web Service Resource Framework).

3.2.4 Abicloud

Abicloud is an open source infrastructure for building and managing public and private clouds based on heterogeneous environments. The tool offers users the ability primarily for scaling, managing, providing automatic and immediate servers and networks [17]. AbiCloud is autoscale: We can change the number of virtual servers, storage and memory. Therefore allows the platform to scale up or down as needed. The platform of AbiCloud is modular because it try to improve the scalability of the system. The architecture is represented by the figure 12.

abiCloud_Server: contains the business logic of the global platform of clouds and interacts with the database.
abiCloud_WS: This virtual assembly line of the platform. interacts with various virtualization technologies to manage virtual machines. AbiCloud_VMS (Virtual Monitor System) is the component developed to monitor the virtual infrastructure to learn about events or states. AbiCloud Appliance Manager: This component enables the management, distribution and scaling (scalability), allowing the import of external applications to the cloud platform. This component is under development.

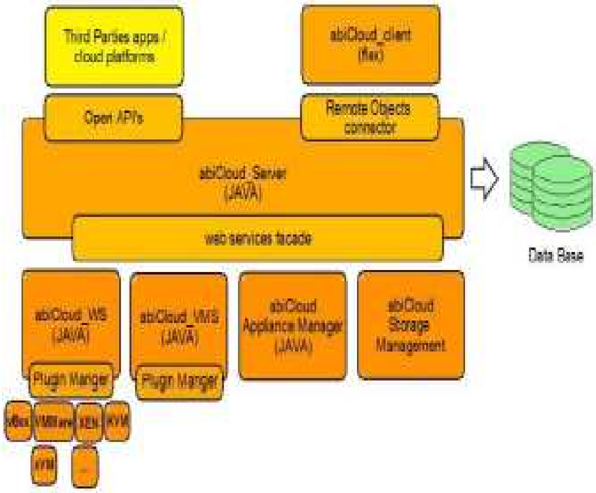

Figure 11. Abicloud's architecture

abiCloud Storage Management: This component is currently being formulated and will be dedicated to the integration of storage platform systems. abiCloud_client: this web application RIA developed in Flex enables users to manage their private Cloud [18].

3.2.5 Windows Azure

Windows Azure is Microsoft's offer on Cloud Computing. This is an application platform providing services, accommodation and administration tools. Windows Azure is an operating system for cloud services that serves the development, service hosting and service management environment for the Windows platform Azure [19]. Windows Azure provides developers ability to do computing and storage and also the management of Web applications on the Internet through on demand data center.
Windows Azure is a flexible platform that supports multiple languages and could be integrate with the existing environment. In addition, Windows Azure supports popular standards and protocols, including SOAP, REST, XML and PHP. Accommodation Azure will provide a set of scalability features of operating on demand. It is thus possible to obtain and allocate additional processors if the scalability of an application requires it.

3.2.6 Google appEngine

AppEngine is intended solely for conventional web
applications, the application is structured with a clear separation between the third load and the storage. In addition, AppEngine applications should be requestresponse. AppEngine provides automatic scaling and high availability. For example, AppEngine is not suited for general computing. He admits a fixed topology structure to accommodate the 3-tier application [20].

4. SOA: Service Oriented Architecture

Service oriented Architecture is an abstract paradigm, based distributed architecture without any reference to implementation. It's a new platform to develop distributed solutions [28]. SOA is an open architecture, extensible, composed and promises guidance service. SOA concepts are often seen as built upon and evolving from older concepts of distributed computing and modular programming (component based software engineering). A focus of this service-oriented approach is on the definition of service interfaces and predictable service behaviors. In SOA, a service is a contractually defined behavior that can be produced and provided by any component to be used by any component, solely on the basis of contract. Service should be autonomous, interoperable, reusable and independent from vendors. In fact services in SOA should verify these characteristics [29]:

Service contract: A defined schema that identifies the message structure, and operations and endpoints.
Loose-coupling. Consumers of the service shouldn't be aware of any implementation details. Related to this is service
abstraction: which aims to make the service a "black-box" to the consumer.
Statelessness. Ideally, a service is completely stand-alone, and does not require previous or follow-up calls for processing. This dramatically improves scalability.
Discoverability. A service registry should exist that identifies available services and their contracts

Benefits of SOA are:
- Facilitates reuse of software assets.
- Enables creation of composite services and BPMS
solutions
- Technology bridge between different software platforms
- Focus shifts from applications to common business services.
- Facilitates creation of distributed solutions that can be easily run in the cloud
- Substitutability: alternate implementations can be easily inserted

The architecture of SOA [29] is presented in the figure bellow:
SOA provides a reduction of dependency but there is interdependence between the various components as shown in the figure: The first step in building an SOA is to identify the various independent components that comprise a business process. Each component is a logical set of people, technologies and resources that provide specific business value, with the potential to operate independently. The components have well defined interfaces; each receives input, processes relevant tasks, and returns the results to other components. These components are considered as services that can be consumed by people or other IT components. Each service is bound to another to communicate between services,
regardless of platform and location [29].

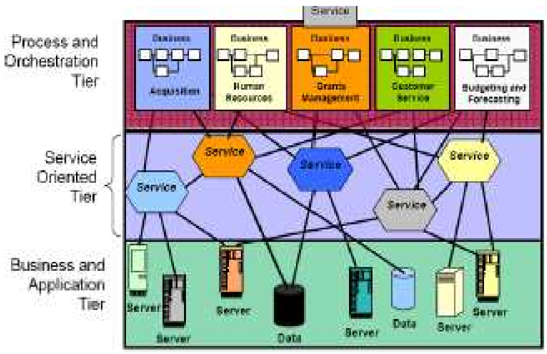

Figure 12. SOA Architecture

The figure 12 shows a clear distinction between business processes, relevant services, and technical implementation of these services. Implementation Services is a separate process. The same services can be implemented with reusable services, detained or services the loan [29]. The first dimension of services is the establishment of clear relationships between business processes and services (between layer 2 and 3). The second dimension is the implementation of services with the available resources (between layer 1 and 2).

4.1 Infrastructure using SOA

M. Agni et al. [30] make an analogy between self-organizing systems and the dynamic characteristic of the SOA in the way they are able to organize items (services in the case of SOA) to change their functions or create new functions.
For them, the dynamic nature of SOA can benefit from the use of primitive self-organization in nature and propose architecture for adapting self-organizing systems found in nature in the SOA to ensure the robustness of SOA. G.P. Kumari et al. [31] attempt to adopt and use the SOA. The need for intelligent infrastructure services leads and simplifies the reuse of services and provides a reliable integration across a heterogeneous and multiple vendors in the computing landscape. They present a solution based on open standards redesigned using principles of SOA and deployed an SOA infrastructure for service delivery and business processes of a major telecom operator.

4.2 Bringing Cloud and SOA together

We can say that SOA provides loose coupling between applications and cloud computing provides a loose coupling between applications and hardware as shown in Figure 13.

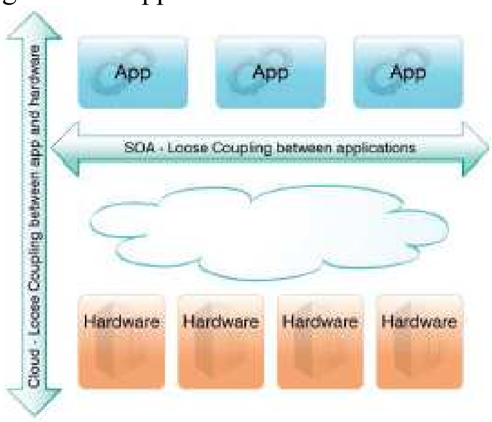

Figure 13. Cloud computing and SOA

David Linthicum details in its paper “Cloud Computing: The Value of Application Service Governance for Cloud Computing” [32] how SOA governance is advantageous when using the cloud Computing specially in the location, length, monitoring and security services.

In SOA, the tipping point is the point when becomes impossible to manage services without a governance model because of large number of services in a cloud. Author proposes to add Process layer around services in the cloud to notify the production of any event and react to it. They try to

Allow technology to self-correct; afford to take corrective action using "trust, but verify" model.

5. Virtualization

5.1 NETWORK VIRTUALIZATION OVERVIEW

Network virtualization provides a powerful way to run multiple networks, each customized to a specific purpose, at the same time over a shared substrate. In fact, it is the technology that allows the simultaneous operation of multiple logical networks (also known as overlays) on a single physical platform. Network virtualization permits distributed

participants to create almost instantly their own network with application-specific naming, routing, and resource management mechanisms such as server virtualization enables users to use even a whole computing center arbitrarily as their own personal computer. In [5] and as shown in figure 15, the authors report that Network virtualization is could be defined by decoupling the roles of the traditional Internet service providers (ISPs) into two independent entities: infrastructure providers (InPs), who manage the physical infrastructure, and service providers (SPs), who create virtual networks (VNs) by aggregating resources from multiple InPs and offer end-to-end services. Such an environment will proliferate deployment of coexisting heterogeneous network architectures free of the inherent limitations of the existing Internet.

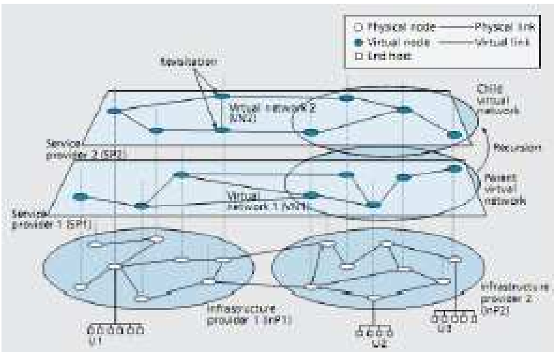

Figure 14. Network Virtualization Environment

5.2 Live Migration Issues and Challenges

5.2.1 Live migration overview

With the extensive use of IMS as a common infrastructure by the almost all the 3G operators, live migration is considered as an interesting and very promising alternative and solution to do with overloads which could affect the components of the Call Session Control Function (CSCF). The technique of live migration involves the transfer of a virtual machine from a physical host to another while ensuring continuity of service. It allows the administrator to reduce the impact of overload without subjecting systems to downtime. Thus, the live migration ensures high availability with service continuity: in the case of resource shortage, the administrator performs a live VM migration to a physical machine more powerful and more efficient to provide more resources in order to meet needs of customers and reduce the unpleasant effects of potential overloads.

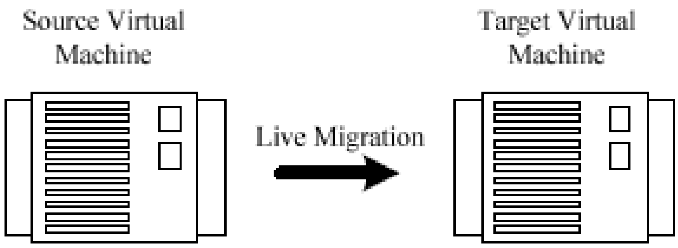

Figure 15. Live migration scenario

In fact, and as mentioned in [8], live OS migration is an extremely powerful tool for cluster administrators, allowing separation of hardware and software considerations, and consolidating clustered hardware into a single coherent management domain. The authors underline that an
administrator may migrate OS instances including the applications that they are running to alternative machine(s) when a physical machine needs to be freed for maintenance. They also mentioned that OS instances may be rearranged across machines in a cluster to reduce load on congested hosts.

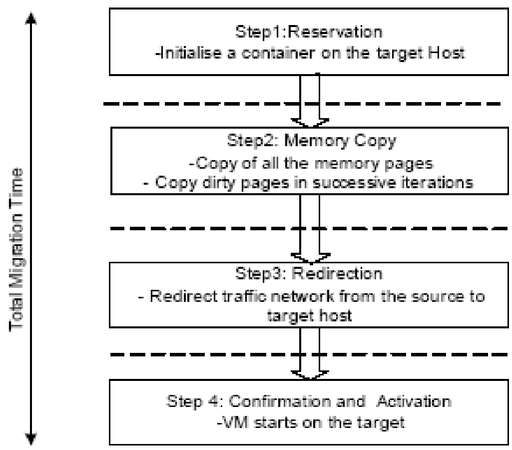

Figure 16. Live migration steps

In [2] and as shows in figure 16, the authors presented the logical steps of live migration when migrating an OS. First of all, we have to select a target host that provides the required resources in order to migrate the overloaded VM. Secondly, a memory copy is made to redirect it from the source to the target host in third step. Finally, the destination machine confirms the reception of the migrated OS image. The memory copy at the destination machine becomes primary and the migrated VM is now activated on the target host.

5.2.2 Live migration challenges

Live migration of virtual machines has to guarantees load balancing, fault tolerance, and power saving, especially in clusters or data centers. In addition, and as highlighted in [8], when a VM is running a live service it is important that this transfer occurs in a manner that balances the requirements of minimizing both downtime and total migration time. In [6], the authors present the design and implementation of a novel memory-compression-based VM migration approach (MECOM) that first uses memory compression to provide fast, stable virtual machine migration, while guaranteeing the virtual machine services to be slightly affected.

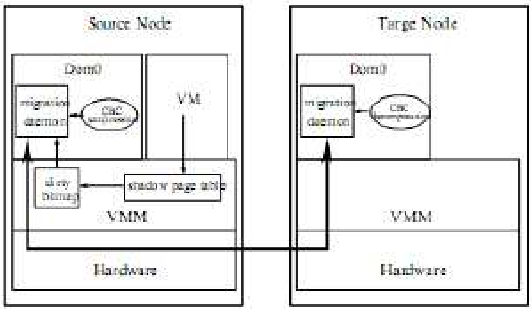

Figure 17. The MECOM architecture

In [7], a novel approach called Hypervisor controlled Mobile IP was proposed to support live migration of Virtual Machine across networks, which enables virtual machine live migration over distributed computing resources.

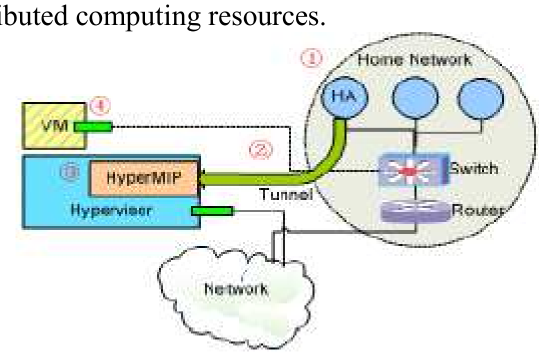

Figure 18. Hypervisor controlled Mobile IP

6. Conclusion and Future Works

In this paper, we tried to review the IMS, Cloud Computing, Network Virtualization, and Live Migration techniques. We presented also some concepts and related work concerning SOA and Cloud Computing. We aimed to focus on the presentation and description of the emerging trends on technology used for cloud-based services. Several researches were made in this field in order to offer the on-demand services for the users and minimize downtime while migrating VM's memory from one physical host to another. We will try in the future to bring them together to develop a viable methodology to describe, discover, compose and manage computing resources and network components constituting the clouds based on the SOA concept. This methodology will provide the flexibility and scalability needed to ensure interoperability between networks and heterogeneous resources forming the clouds. We will focuses on the dynamic composition of networks forming clouds and allow the discovery, decomposition and execution services "cloud services" on demand. The organization and orchestration of these services will be managed optimally under SOA.

References

[1] I. Limam Bedhiaf, O. Cherkaoui, G. Pujolle "Third-generation virtualized architecture for the MVNO context", Annals of Telecommunication, June 2009.

[2] M. Fakhfakh, O. Cherkaoui, I. Limam Bedhiaf, M. Frikha, "High Availability in IMS Virtualized Network", First International Conference on Communications and Networking (ComNet2009), Nov 2009.

[3] G. Bertrand, "The IP Multimedia Subsystem in Next Generation Networks", May 2007.

[4] R. Farahbakhsh, M. Vaposhti, N. Movahhedinia, "Transmission Delay Reduction in IMS by Re-registration procedure modification", The second International Conference on Next Generation Mobile Applications, Services, and Technologies, 2008.

[5] N. M. Mosharaf Kabir Chowdhury and Raouf Boutaba, "Network Virtualization: State of the Art and Research Challenges", IEEE Communications Magazine, July 2009.

[6] Hai Jin, Li Deng, Song Wu, Xuanhua Shi, Xiaodong Pan, "Live Virtual Machine Migration with Adaptive Memory Compression".

[7] Qin Li, Jinpeng Huai, Jianxin Li, Tianyu Wo, Minxiong Wen, "HyperMIP: Hypervisor controlled Mobile IP for Virtual Machine Live Migration across Networks", 11th IEEE High Assurance Systems Engineering Symposium, 2008.

[8] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, Andrew Warfield, "Live Migration of Virtual Machines", Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2, pp 273-286, 2005.

[9] Lijun Mei, W.K. Chan, T.H. Tse, "A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues", Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference (APSCC 2008), IEEE Computer Society Press, Los Alamitos, CA.

[10] David Chou, "Understanding Cloud Computing and Cloud-Based Security", SOA Magazine, 2010

[11] Eucalyptus systems, Eucalyptus Open-Source Cloud Computing Infrastructure - An Overview, August 2009

[12]http://www.ubuntu.com/system/files/Ubuntu%209%2010%20Eucalyptus-Press%20Release%20-%2026Oct09_1.pdf

[13] WWW.opennebula.org

[14] Borja Sotomayor et al., « capacity leasing in cloud Systems using the Open Nebula Engine », Facultad de Informaticà, Universidad Complutense de Madrid.

[15] Constantino Vázquez Blanco, "The OpenNebula Virtual Infrastructure Engine", Distributed Systems Architecture Research Group, Universidad Complutense de Madrid, June 27, 2009

[16]http://www.nimbusproject.org/docs/current/faq.html#nimbus-maincomponents

[17] http://www.abicloud.org/display/abiCloud/Home ABICLOUD TECHNICAL OVERVIEW,2009.

[18] Nicolas CLERC , " Windows Azure – Présentation technologique, Expertise des ecosystems".

[19] Michael Armbrust et al."Above the Clouds: A Berkeley View of Cloud Computing", Electrical Engineering and Computer Sciences, University of California at Berkeley, 2009

[20] Steve Bennett, Mans Bhuller, Robert Covington, Architectural Strategies for Cloud Computing, Oracle corporation, 2009.

[21] Andy Mulholland, Russ Daniels, Tim Hall, Mary Johnson, Pete Chargin, "The Cloud and SOA : Creating an Architecture for Today and for the Future", Capgemini in collaboration with HP invent, 2008.

[22] Alexandros Marinos, Gerard Briscoe, "Cloud community", Department of Computing University of Surrey, Department of Media and Communications London School of Economics and Political Science, United Kingdom, 2009.

[23] Goeffrey Raines, "Cloud Computing and SOA", Systems Engineering at MITRE, technical paper SERVICE-ORIENTED ARCHITECTURE (SOA) SERIES,2009.

[24] Mell, Tim Grance, "Effectively and Securely Using the Cloud Computing Paradigm", NIST, Information Technology Laboratory, 2009.

[25] Nuno Santos, Krishna P. Gummadi, Rodrigo Rodrigues, " Towards Trusted Cloud Computing", MPI-SWS, workshop on hot Topics in Cloud Computing, San Diego, 2009.

[26] Jyoti Namjoshi, Archana Gupte, "Service Oriented Architecture for Cloud based Travel Reservation Software as a

Service", Product and Technology Initiatives, Patni Computer System Ltd, IEEE International Conference on Cloud Computing India, 2009.

[27] Philip K. McKinley, Farshad A. Samimi, Jonathan K. Shapiro, Chiping Tang, "Service Clouds: A Distributed Infrastructure for Constructing Autonomic Communication Services'', department of Computer Science and Engineering, Michigan State University, Dependable, Autonomic and Secure Computing, 2nd IEEE International Symposium, 2006.

[28] K. Ben Driss, "SOA : pour une interopérabilité intrinsèque du Système d'Information", WIA Workshop in Informatique et Applications, Tozeur, 2008.

[29] K. Kanchanavipu, "An Integrated Model for SOA Governance: An Enterprise Perspective", Master Thesis in Software Engineering and Management , IT University of Göteborg Chalmers University of Technology and University of Gothenburg Göteborg, Sweden 2008.

[30] M. Agni C. Bhakti, Azween B. Abdullah, "Towards Self-Organizing Service Oriented Architecture", CIS Department, University Teknologi PETRONAS, 2009.

[31] Geetha Presena Kumari, Balaji Kandan, Asish Kumar Mishra, « Experience sharing on SOA based Heterogeneous Systems Integration » SOA Competency Center, Hewlett-Packard Bangalore, India, 2008.

[32] David Linthicum, "Cloud Computing : The Value of Application Service Governance for Cloud Computing", Layer 7 Technologies, 2009

Table 1 : Comparaison of Several Cloud's solution

	Eucalyptus	OpenNebula	Nimbus	AbiCloud	Azure	Appengine
Service	IaaS	IaaS	IaaS	IaaS	PaaS	PaaS
Type of cloud	Public / Private	Private	Public	Public/ Private	-	-
Scalability	Not Scalable	Scalable dynamique	Scalable	Scalable	Scalable	Scalable
Compatibility	Not support EC2	Multi-plateforme	Support EC2, WRSF	Support EC2	-	-
VM support	VMware, Xen, KVM	Xen, VMware	Xen	virtualBox, Xen, VMware	XEN Hypervisor	Multitenant architecture
Structure	Module	Module	Component	Module	-	-
Provisioning Model	Immediate	Best effort + haizea: advance reservation, immediate, best-effort + reservoir: Immediate, Best-effort	Immediate	Immediate	-	-
Load balancing	Simple load balancing cloud controller	Nginx configured as load balancing	auto configuration of virtual clusters		Should install software	Automatic