Cloud Computing Architectures

Huaglory Tianfield
School of Engineering and Built Environment, Glasgow Caledonian University
Cowcaddens Road, Glasgow G4 0BA, United Kingdom
E-mail: h.tianfield@gcu.ac.uk

***Abstract*—In this paper, we put forward a basic taxonomy of cloud computing architectures. By this taxonomy, cloud computing architectures are essentially subdivided into Cloud Platform Architecture (CAA) and Cloud Application Architecture (CAA) which are linked via the cloud services available in the marketplace of Information Technology (IT) capabilities. We elaborate the constructs of CPA and CAA, respectively. Such a division between CPA and CAA is fundamental for cloud computing to serve as a potential foundation for delivering IT services as utilities over the Internet.**

Keywords-cloud computing; cloud computing architecture; cloud platform architecture; cloud application architecture; service-oriented architecture; utility computing

I. INTRODUCTION

A cloud pools together large numbers of physically distributed compute resources, e.g., processors, memory, network bandwidth and storage, which can be organized on demand into services that can grow or shrink in real-time. [1]

NIST defines cloud computing as "… a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [2]

Cloud computing is the convergence of several concepts from resource pooling, virtualization, dynamic provisioning, utility computing, on-demand deployment, Internet delivery of services, to enable a more flexible approach to deploying and scaling applications.

Through cloud computing, applications can rapidly be deployed where the underlying technology components can expand and contract with the natural ebb and flow of the business life cycle. [3]

Instead of requiring a long-term contract for services with an Information Technology (IT) organization or a service provider, clouds work on a pay-by-use, pay-per-cycle or pay-by-the-sip model where an application may exist to run a job for a few minutes or hours, or to provide services to customers on a long-term basis. Compute clouds are built as if applications are temporary, and billing is based on resource consumption: CPU hours used, volumes of data moved, or gigabytes of data stored.

Most of the current work on cloud computing focuses on its concepts and the analysis of business opportunities, benefits and deployment modes. However, very little looks at the inherent architectures of cloud computing. In this paper, we will put forward a basic taxonomy of architectures for cloud computing. We essentially subdivide cloud computing architectures into Cloud Platform Architecture (CPA) and Cloud Application Architecture (CAA). Such a division between CPA and CAA is fundamental for cloud computing to serve as a potential foundation for delivering IT services as utilities over the Internet.

II. FUNDAMENTALS OF CLOUD COMPUTING ARCHITECTURES

To reach the essence of cloud computing, we revisit the basic concepts as follows.

Definition 1. A Cloud is an Internet-Centric Marketplace of IT Capabilities.

Definition 2. Cloud Computing is a paradigm of computing that operates on the resources which are made available via cloud services.

As evident by Definition 1, the basic mechanism that governs the cloud would be the demand and supply relations in the cloud marketplace. Thus, cloud computing architectures should basically involve CPA and CAA. Based on such an understanding, we put forward a basic taxonomy of cloud computing architectures, as illustrated in Fig. 1.

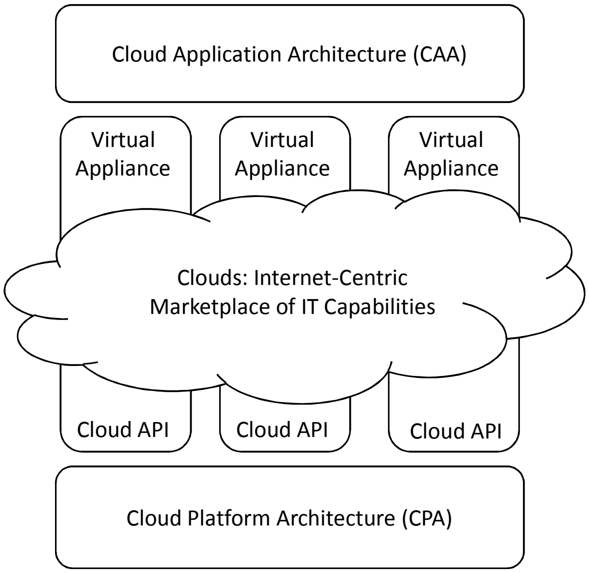
Cloud Application Architecture (CAA)
Virtual Appliance
Virtual Appliance
Virtual Appliance
Clouds: Internet-Centric Marketplace of IT Capabilities
Cloud API
Cloud API
Cloud API
Cloud Platform Architecture (CPA)

Figure 1. Taxonomy of cloud computing architectures

978-1-4577-0653-0/11/$26.00 ©2011 IEEE

There is a clear separation between the functional roles of service providers and infrastructure providers. Service providers are the entities that understand the needs of a particular business and offer service applications to address those needs. Service providers do not own the computational resources needed for these service applications; instead, they lease resources from infrastructure providers, which provide them with a seemingly infinite pool of computational, network, and storage resources.

Infrastructure providers operate host sites that own and manage the physical infrastructure on which service applications execute. The federation of collaborating sites forms a cloud marketplace. To optimize resource utilization, the computational resources within a site are partitioned by a virtualization layer into Virtual Execution Environments (VEEs), namely fully isolated runtime environments that abstract away the physical characteristics of the resource and enable sharing. The virtualized computational resources, alongside the virtualization layer and all the management enablement components, are referred to as service provider. [4]

III. Cloud Services

In a marketplace of IT utilities, a wide range of cloud services may be offered. Cloud services are encapsulated, have Application Programming Interfaces (APIs), and are available over the network.

Cloud Services represent any type of IT capability that is provided by Cloud Service Provider (CSP) to Cloud Service Customers (CSCs). Typical categories of cloud services are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), or Business Process as a Service, as depicted in Fig. 2. In contrast to traditional IT services, cloud services have attributes associated with cloud computing, such as a pay-per-use model, self-service usage, flexible scaling, and shared underlying IT resources.

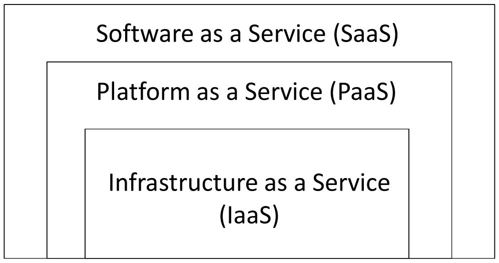
Software as a Service (SaaS)
Platform as a Service (PaaS)
Infrastructure as a Service (IaaS)

Figure 2. Cloud service stack

Let icvoda denote “internet-centric virtualization and on-demand access”, DF “distributed IT facilities”, | “and/or”, then the service delivery models of Cloud Computing can be expressed as follows.

$$\begin{aligned} \text{IaaS} \Leftarrow\ & \text{icvoda}^{(I)}\{\text{DF}\}, \\ \text{PaaS} \Leftarrow\ & \text{icvoda}^{(P)}\{\text{DF}\} \mid \text{icvoda}^{(P,I)}\{\text{IaaS}\}, \\ \text{SaaS} \Leftarrow\ & \text{icvoda}^{(S)}\{\text{DF}\} \mid \text{icvoda}^{(S,P)}\{\text{PaaS}\} \mid \\ & \text{icvoda}^{(S,P)}\{\text{icvoda}^{(P,I)}\{\text{IaaS}\}\} \mid \\ & \text{icvoda}^{(S,I)}\{\text{IaaS}\}. \end{aligned} \quad (1)$$

IV. Cloud Platform Architecture (CPA)

In a cloud platform, which can offer IaaS, PaaS, SaaS, etc., large resource pools based on virtualized infrastructure provide greater flexibility and efficiency. Resources of each physical host are virtualized and presented as multiple Virtual Machines (VMs) to run multiple operating systems and application instances. Cloud platform provides pools of virtualized resources (compute, memory, storage, bandwidth) spanning multiple hosts and storage frames. Multi-tenancy (different resource pools for different customers) is on shared physical infrastructure.

To achieve higher levels of resource utilization within each pool, techniques such as workload balancing across physical servers and storage frames can be used. Workload balancing is achieved with VM live migration, which migrates virtualized applications between physical resources within a resource pool in a way that is transparent to users and does not interrupt the service provided by the cloud platform.

We put forward a CPA as illustrated in Fig. 3.

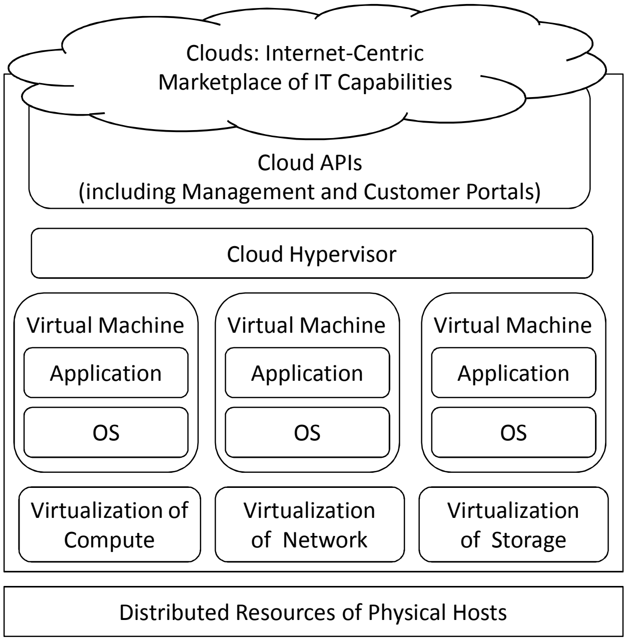
Clouds: Internet-Centric Marketplace of IT Capabilities
Cloud APIs
(including Management and Customer Portals)
Cloud Hypervisor
Virtual Machine
Application
OS
Virtual Machine
Application
OS
Virtual Machine
Application
OS
Virtualization of Compute
Virtualization of Network
Virtualization of Storage
Distributed Resources of Physical Hosts

Figure 3. Cloud platform architecture (CPA)

Computing, storage and network resource are three basic resources in a cloud platform. Compute clouds are usually complemented by storage clouds that provide virtualized storage through APIs that facilitate storing VM images, source files for components such as Web servers, application state data, and general business data.

A. Virtualization Technologies

Virtualization has re-emerged in recent years as a compelling approach to increasing resource utilization and reducing IT service costs. The common theme of all virtualization technologies is hiding the underlying infrastructure by introducing a logical layer between the physical infrastructure and the computational processes.

Virtualization technologies are one of the important building blocks in CPA. The dynamic infrastructure enabled by technologies such as virtualization aligns well with the dynamic on-demand nature of clouds.

At a fundamental level, virtualization technology enables the abstraction or decoupling of the application payload from the underlying physical resource [5]. What this typically means is that the physical resource can then be carved up into logical or virtual resources as needed. This is known as provisioning. By introducing a suitable management infrastructure on top of this virtualization functionality, the provisioning of these logical resources could be made dynamic, i.e., the logical resource could be made bigger or smaller in accordance with demand. This is known as dynamic provisioning. To enable a true "cloud" computer, every single computing element or resource should be capable of being dynamically provisioned and managed in real-time. [1]

Virtualization takes many forms. System virtualization [6], also commonly referred to as server virtualization, is the ability to run multiple heterogeneous operating systems on the same physical server [7]. With server virtualization, a control program (commonly known as "hypervisor" or "VM monitor") is run on a given hardware platform, simulating one or more other computer environments (VMs). Each of these VMs, in turn, runs its respective "guest" software, typically an operating system, which runs just as if it were installed on the stand-alone hardware platform. Other forms of virtualization include storage virtualization and network virtualization, namely logical representations of the physical storage and network resources. [4]

Virtualization further enhances flexibility because it abstracts the hardware to the point where software stacks can be deployed and redeployed without being tied to a specific physical server. Virtualization enables a dynamic datacenter where servers provide a pool of resources that are harnessed as needed, and where the relationship of applications to compute, storage, and network resources changes dynamically in order to meet both workload and business demands. With application deployment decoupled from server deployment, applications can be deployed and scaled rapidly, without having to first procure physical servers. [8]

Virtualization dynamically overlays VMs over physical resources. In general, these efforts try to extend the benefits of virtualization from a single resource to a pool of resources, decoupling the VM not only from physical infrastructure but also from physical location. [4]

Virtual appliances, namely VMs that include software that is partially or fully configured to perform a specific task such as a Web or database server, further enhance the ability to create and deploy applications rapidly. The combination of VMs and virtual appliances as standard deployment objects is one of the key features of cloud computing.

Distributed VM management in hypervisors enables live migration and suspend/resume mechanisms that allow moving a VM from one host to another, stopping the VM and starting it again later. To have a dynamic virtualized, multi-tenant environment, key requirements include optimal runtime placement of virtualized workloads and comprehensive VM performance monitoring and diagnostics.

B. Scaling and Elasticity

Approaches to scaling infrastructures to meet the demand can be classified as physical investment type and run-time horizontal scaling type.

Scale-up (i.e., physical investment) approach is not concerned with scalable architecture, but invests heavily in larger and more powerful computers (vertical scaling) to accommodate the demand.

The traditional scale-out (component based) approach creates an architecture that scales horizontally and invests in infrastructure in increments. Most of the businesses and large-scale web applications follow this approach by distributing their application components, federating their datasets and employing a service-oriented pattern. This approach, often more effective than a scale-up one though, still requires predicting the demand at regular intervals and then deploying infrastructure in increments to meet the demand.

In the context of the cloud, decoupling your components, building asynchronous systems and scaling horizontally become very important. It will not only allow you to scale out by adding more instances of the same component, but also allow you to design hybrid models in which a few components continue to run in on-premise resources while other components can take advantage of the cloud marketplace and use the cloud services for additional compute-power and bandwidth. By this way, you can "overflow" excess workload to the cloud via load balancing tactics.

Applications taking advantage of horizontal scaling should focus on overall application availability with the assumption that individual components may fail. Most cloud platforms are built on a virtual pool of server resources where, if any one physical server fails, the VMs that it was hosting are simply restarted on a different physical server. The combination of stateless and loose-coupled application components with horizontal scaling promotes a fail-in-place strategy that does not depend on the reliability of any one component. [8]

Horizontal scaling does not have to be limited to a single cloud. Depending on the size and location of application data, "surge computing" can be used to extend a cloud's capability to accommodate temporary increases in workload. In surge computing, an application running in a private cloud might recruit additional resources from a public cloud as the need arises, i.e., to overflow excess workload to a public cloud. [8]

Horizontal scaling basically calls for Service-Oriented Architectures (SOAs). The cloud reinforces the SOA design principle that the more loosely coupled the components of the system, the bigger and better it scales. [9]

Elasticity is the power to scale computing resources up and down easily and with minimal friction. Elasticity should be one of the architectural design requirements or a system property. [9]

Automated elasticity of cloud computing enables the infrastructure to be closely aligned (as it expands and contracts) with the actual demand, thereby increasing overall utilization and reducing cost. The elastic aspect of cloud computing allows applications to scale and grow without needing traditional 'fork-lift' upgrades.

Elasticity can be achieved through auto-scaling based on demand. Auto-scaling means you can scale your applications up and down to match your unexpected demand without any human intervention. By using a monitoring tool, your system can send triggers to take appropriate actions so that it scales up or down based on metrics (utilization of the servers or network I/O, for instance). [9]

A cloud platform can be monitored using data analysis tools in order to gain visibility into resource utilization, operational performance, and overall demand patterns (including metrics such as CPU utilization, disk reads and writes, and network traffic). Auto-scaling can automatically scale your capacity on certain conditions based on metrics that data analysis tools collect, e.g., historical consumption and purchasing information, performance and utilization trends, summaries of alerts and security-related events, etc.

Within each host site, the resource utilization is monitored and the placement of VEEs is constantly updated to achieve optimal utilization with minimal cost.

V. CLOUD APPLICATION ARCHITECTURE (CAA)

Cloud computing takes further concepts such as utility computing and virtualization by allowing self-service, metered usage and more automated dynamic resource and workload management. As services became more and more distributed, SOAs have emerged as a methodology to integrate and orchestrate distributed business services. [3]

From an enterprise perspective, the on-demand nature of cloud computing helps realize the performance and capacity aspects of Service-Level Objectives (SLOs). The self-service nature of cloud computing allows organizations to create elastic environments that expand and contract based on the workload and target performance parameters. The pay-by-use attribute of cloud computing may take the form of equipment leases that guarantee a minimum level of service from a CSP.

The key is to build components that do not have tight dependencies on each other, so that if one component were to die (fail), sleep (not respond) or remain busy (slow to respond) for some reason, the other components in the system are built so as to continue to work as if no failure is happening. In essence, loose coupling isolates the various layers and components of your application so that each component interacts asynchronously with the others and treats them as a "black box". [9]

Cloud computing does not replace SOA, or the use of distributed software components, as an integration technology. [10] Rather, SOA and cloud computing are related. Specifically, SOA is an architectural pattern that guides business solutions to create, organize and reuse its computing components, while cloud computing is a set of enabling technologies that services a bigger, more flexible platform for enterprise to build their SOA solutions.

Only through federation and interoperability can infrastructure providers take advantage of their aggregated capabilities to provide a seemingly infinite service computing utility. [4]

We put forward a CAA, as illustrated in Fig. 4, which is itself wholly a SOA. In fact, it is by means of SOA that cloud services are able to be organized in CAA more effectively.

CAA is basically comprised of three layers, namely, the virtual appliances which run with the APIs of various CSPs/platforms, the cloud brokers which work with the associated cloud ontologies, and the Business Service and Process (BSP) layer which performs Business Service Management (BSM), Service Level Agreement (SLA), service orchestrations and process management. BSP and cloud broker layer jointly implement service-oriented processes, including cloud service discovery, matching, dynamic SLA negotiation, on-demand provision, etc.

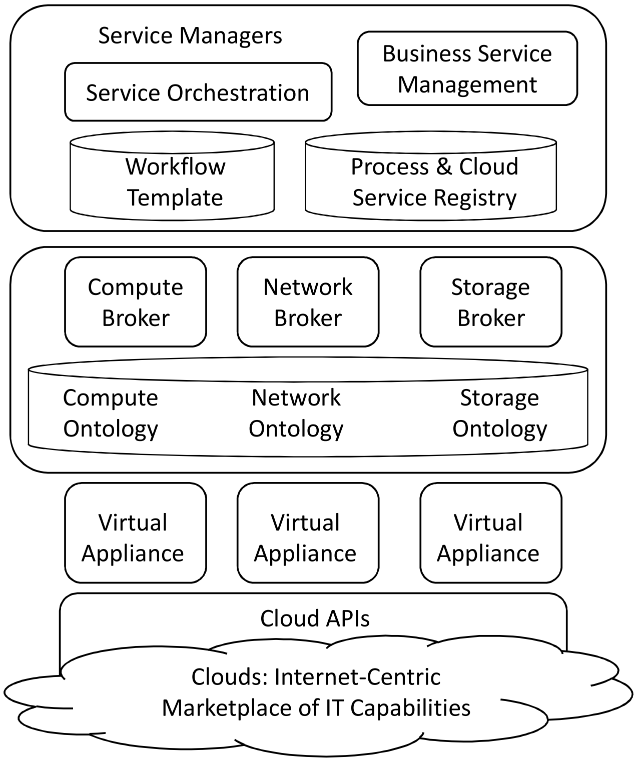

Figure 4. Cloud Application Architecture (CAA)

CAA embodies service management framework and is overall a SOA. CAA has unified the service-oriented cloud computing artchctatuere in [11] and the federated cloud architecture in [4]. The concept of federation of clouds [4] essentially is about management of cloud services from heterogeneous CSPs. The essence of federated clouds is that a future enterprise computing in a cloud environment has to resort to IT utilities that are distributed and heterogeneous in the cloud marketplace.

A service application is a set of software components that work collectively to achieve a common goal. Each component of such service applications executes in a dedicated VEE.

These VEEs are placed on the same or different CSPs within the site, or even on different sites. A service application is deployed on the CAA using a service manifest that formally defines the contract and SLA between the service provider and the infrastructure provider.

The execution of the service applications is monitored and the capacity is constantly adjusted to meet the requirements and SLA specified in the service manifest.

A. BSP Layer

In BSP Layer, not only services but also many other artifacts can be published and shared, such as workflow templates, collaboration templates and test cases.

BSP layer handles the full lifecycle of virtualized resources and provides additional common infrastructure elements for service level management, metered usage, policy management, license management, and disaster recovery. Mature cloud service management software allows dynamic provisioning and resource allocation to allow applications to scale on demand and minimize the waste associated with underutilized and static computing resources. [3]

A key aspect of BSM is SLA management. New SLA management challenges arise due to the dynamic federation of cloud infrastructure providers.

Cloud computing must support for BSM, specifically for business-aligned SLA management. While specific cloud computing solutions can be enhanced with some aspects of BSM, the provisioning of complex services across a federated network of possibly disparate datacenters is a difficult problem. A service may be a composition of numerous distributed resources, including computing, storage, and network elements. Provisioning such a service consumes physical resources, but should not cause an SLA violation of any other running application with a probability larger than some predefined threshold.

Functionalities of BSP layer are represented by service manager. Service manager interacts with CSPs to receive their service manifests, negotiate pricing, and handle billing. Two of its most complex tasks are deploying and provisioning VEEs based on the service manifest, and monitoring and enforcing SLA compliance by throttling a service application's capacity.

Service manager receives service manifests from CSPs. Based on information in the manifests, it deploys and provisions the service application by interacting with cloud brokers to allocate VEEs and their associated resources. From the service requirements in the manifests (i.e., SLOs, elasticity rules, etc.), service manager derives a list of required resources and their configuration, as well as placement constraints based on cost, licensing, confidentiality, etc. For unsized service applications, service manager is responsible for generating explicit rules based on site policy. Deployment and provisioning decisions are based on performance and SLA compliance and adjusted according to business considerations (e.g., costs, security, offers, etc.). [4]

Service manager is also responsible for monitoring the deployed services and adjusting their capacity, i.e., the number of VEE instances as well as their resource allocation (memory, CPU, etc.), to ensure SLA compliance and alignment with high-level business goals (e.g., cost-effectiveness). [4]

B. Cloud Broker Layer

Cloud brokers serve as the agents between individual CSPs and BSP layer. Each major cloud service has an associated service broker type.

Cloud broker is responsible for the optimal placement of VEEs into CSPs subject to constraints determined by service manager. The continuous optimization process is driven by a site-specific programmable utility function. Cloud broker is free to place and move VEEs anywhere, even on the remote sites (subject to overall cross-site agreements), as long as the placement satisfies the constraints. Thus, in addition to serving local requests (from the local service manager), cloud broker is responsible for the federation of remote sites. [4]

CSPs might not conform to the standards rigidly; they might also have implemented extra features that are not included in the standards. Cloud ontologies exist to mask the differences among the different individual CSPs and can help the migration of cloud application from one cloud to another. Each cloud broker has the associated cloud ontology, i.e., storage ontology, compute ontology, network ontology. [11]

At cloud broker level a service is realized as a set of inter-related VEEs (a VEE Group), and hence it should be managed as a whole. For example, the service manifest may define a specific deployment order, placement constraints (i.e., affinity rules), or rollback policies. Cloud broker also provides the functionality needed to handle the dynamic nature of the service workload, such as the ability to add and remove VEEs from an existing VEE Group, or to change the capacity of a single VEE. [4]

C. CSP Layer

CSP layer resembles the normal cloud platforms. Each CSP builds its own datacenters that power the cloud services it provides. Each cloud may have its own proprietary virtualization technology or utilize open source virtualization technology, such as Eucalyptus [12].

Deploying cloud applications as virtual appliances makes management significantly easier. The virtual appliances should bring with them all of the software they need for their entire lifecycle in the cloud. More importantly, they should be built in a systematic way, akin to an assembly line production effort as opposed to a hand crafted approach. The reason for this systematic approach is the consistency of creating and re-creating images. [13]

A virtual appliance is an application that is bundled with all the components that it needs to run, along with a streamlined operating system. In a cloud computing environment, a virtual appliance can be instantly provisioned and decommissioned as needed, without complex configuration of the operating environment. [13]

When building virtual appliances, it is obvious that they should contain the operating system and any middleware components they need. A virtual appliance is an instance run in a VEE. Less obvious are the software packages that allow them to automatically configure themselves, monitor and report their state back to a management system, and update themselves in an automated fashion. Automating the virtual appliance configuration and updates means that as the application grows in the cloud, the management overhead does not grow in proportion. In this way virtual appliances can live inside the cloud for any length of time with minimal management overheads.

When virtual appliances are instantiated in the cloud, they should also plug into a monitoring and management system. This system will allow you to track application instances running in the cloud, migrate or shutdown instances as needed, and gather logs and other system information necessary for troubleshooting or auditing. Without a management system to handle the virtual appliances, it is likely that the application will slowly sprawl across the cloud, wasting resources and money.

By automating the creation and management of these virtual appliances, you are tackling one of the most difficult and expensive problems in software today: variability. By producing a consistent virtual appliance image and managing it effectively, you are removing variability from the release management and deployment process. Reducing the variability reduces the chances of mistakes.

One of the key characteristics that distinguish cloud computing from standard enterprise computing is that the infrastructure itself is programmable. Instead of physically deploying servers, storage, and network resources to support applications, developers specify how the same virtual components are configured and interconnected, including how VM images and application data are stored and retrieved from a storage cloud. They specify how and when components are deployed through an API that is specified by CSP. [8]

Effective development tools can leverage cloud's distributed computing capabilities. These tools not only facilitate service orchestration that can leverage dynamic provisioning, but also enable business processes to be developed that can harness the parallel processing capabilities available to clouds. The development tools must support dynamic provisioning and not rely on hard coded dependencies such as servers and network resources. [3]

Service providers of traditional SOA develop the logic of a service and provide its running environment. In CAA, services are published as re-deployable packages, namely service package. If CSPs only use the standard APIs and protocols, a single version of complied code is enough; if CSPs optimize the performance of their services by utilizing some platform unique APIs and features, complied code for each platform is needed. [11]

CSP is responsible for the basic control and monitoring of VEEs and their resources (e.g., creating a VEE, allocating additional resources to a VEE, monitoring a VEE, migrating a VEE, creating a virtual network and storage pool, etc.). Each CSP type encapsulates a particular type of virtualization technology, and all CSP types expose a common interface such that cloud broker can issue generic commands to manage the life-cycle of VEEs. The receiving CSP is responsible for translating these commands into commands specific to the virtualization platform. [4]

VI. Conclusions

We have put forward a basic taxonomy of architectures for cloud computing. Cloud computing architectures are essentially subdivided into Cloud Platform Architecture (CPA) and Cloud Application Architecture (CAA) which are linked via the cloud services available on the marketplace of IT utilities. Such a division between CPA and CAA is fundamental for cloud computing to serve as a potential foundation for delivering IT services as utilities over the Internet, because by this way, the concerns of CSPs and CSCs are profoundly separated. Our elaborations on the constructs of CPA and CAA have manifested that while the focus of CPA lies at Internet-centric virtualization of IT capabilities and the elasticity, the focus of CAA is at service management and SOAs, which will be able to provide a robust cloud computing environment despite heterogeneity and dynamic changes of CSPs.

References

[1] V. Sarathy et al, "Next generation cloud computing architecture -- enabling real-time dynamism for shared distributed physical infrastructure", 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'10), Larissa, Greece, 28-30 June 2010, pp. 48-53.

[2] P. Mell et al, "NIST definition of cloud computing", vol. 15, October 2009.

[3] S. Bennett et al, "Architectural strategies for cloud computing", Oracle White Paper in Enterprise Architecture, August 2009.

[4] B. Rochwerger et al, "The RESERVOIR model and architecture for open federated cloud computing", IBM Journal of Research and Development, vol. 53, no. 4, 2009, pp. 1-11.

[5] R. Buyyaa et al, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, vol. 25, no. 6, June 2009, pp. 599-616.

[6] G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures", Communications of ACM, vol. 17, no. 7, 1974, pp. 412-421.

[7] P. Barham et al, "Xen and the art of virtualization", in Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP'03), New York, USA, 19-22 October 2003, pp. 164-177.

[8] SUN Microsystems, "Introduction to cloud computing architecture", White Paper, 1st Edition, June 2009

[9] J. Varia, "Architecting for the cloud: Best practices", May 2010.

[10] G. Raines, "Cloud computing and SOA", Service-Oriented Architectures (SOA) series, The MITRE Corporation, Case Number: 09-0743, Document Number: MTR090026.

[11] W.-T. Tsai et al, "Service-oriented cloud computing architecture", Proceedings of 7th International Conference on Information Technology: New Generations (ITNG'10), 12-14 April 2010, Las Vegas, Nevada, USA, pp. 684-689.

[12] Eucalyptus System, http://www.eucalyptus.com/

[13] J. Barr et al, "Application architecture for cloud computing", White Paper, rPath.