RESEARCH PAPER | Engineering | Volume : 2 | Issue : 2 | November 2012 | ISSN - 2249-555X

Improving Utilization of Infrastructure Cloud Computing Reference Architecture

KEYWORDS: CCRA, SOA, Hybrid, Ontology, Open Group

D.Pramodh — PG Student, LIET, Hyderabad, India,

Abdul Mazid — Assoc. Professor, LIET, Hyderabad, India,

ABSTRACT *Reference Architecture (RA) provides a blueprint of a to-be-model with a well-defined range, requirements it satisfies, and architectural decisions it understands. By delivering best practices in a standardized, methodical way, an RA ensures consistency and quality across development and delivery. The architecture overview is intended to provide a general, coherent architectural structure which should be used as a base for any cloud computing task. This allows representing the architecture of any cloud project in a consistent manner. Existing "legacy" products and technologies as well as new cloud technologies can be mapped on the Arichecture Overview Diagram (AOD) to show integration points amongst the new cloud technologies and integration points between the cloud technologies and already existing ones. The architectural principles define the fundamental principles which need to be followed when realizing a cloud. These principles need to be followed on all implementation stages (architecture, design, and implementation) and have implications across all work products.*

I.INTRODUCTION

The mission of the Cloud Computing Reference Architecture (CCRA) is to enable cloud-scale finances in delivering cloud services while optimizing resource and labor utilization and delivering a design blueprint for

- Cloud services, which are offered to clients
- Private, public or hybrid cloud projects
- Workload-optimized systems
- Enabling the management of multiple cloud services based on the same, general management platform for enabling economies of scale.

In order to understand the CCRA it is important to understand the relationship between SOA and Cloud, not only at an architectural level, but also at a solution and service level. Service oriented architecture (SOA) is defined by The Open Group to be "an architectural style that supports service orientation. Service orientation is a way of thinking in terms of services and service-based development and the outcomes of services [1].

Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g.,

networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models. The Service models are Cloud Infrastructure as a Service, Cloud Platform as a Service, and Cloud Software as a Service. The service models, and the fact that cloud computing is discussed in terms of the creation, delivery and consumption of cloud services, means cloud computing supports service orientation. Enterprises expose infrastructure, platforms and software as

services as part of SOA solutions today. Certainly Software as a Service is not new and has been a popular topic for years [4].

The cloud deployment models are Private, Community, Public, and Hybrid. These deployment models define the scope of the cloud architecture and solution, does the cloud solution exist strictly within the organization boundaries, across organization boundaries or a combination. Certainly these scopes have been seen in SOA solutions before Cloud, there are SOA solutions that exist strictly within an enterprise, or between businesses across enterprise boundaries. In fact one of the key values of SOA was to develop SOA solutions with services that are integrate between business partners, enabling outsourcing, simplifying integration and increasing agility, much like the Hybrid model. Cloud computing enables this paradigm by adding cloud-characteristics to the services being delivered & consumed.

The essential characteristics for Cloud Computing are on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured Service. These characteristics can be found in requirements and SOA solutions in various organizations today, although these characteristics are optional for SOA and mandatory for cloud. Usually, a single SOA solution does not have all of these are characteristics simultaneously, unless it is a very mature organization leveraging SOA. For these organizations, each of these solutions had to be built in its entirety for that organization. The SOA solution and management of it must be built from scratch, and is not generally shared amongst organizations. Reuse is generally within an organization or an industry, not between organizations and entire communities. Service delivery and consumption aspects are a small part of the requirements for the SOA solution. What is new about cloud is that instead of supporting these requirements per solution, the industry is trying to 'standardize' how these requirements are being met to enable cloud computing. Cloud architectures require a set of capabilities. This discussion shows that Cloud computing architectures are Service Oriented architectures and adhere to architectural style that supports service orientation. Cloud solutions are SOA solutions.

The Open Group defines a service:
"A service:

- Is a logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit; provide weather data, consolidate drilling reports)
- Is self-contained
- May be composed of other services
- Is a "black box" to consumers of the service "

Cloud services, according to The Open Group definition, are SOA services. However, not all SOA services are Cloud service because they require automated deployment and management as well as offering support in order to support the Cloud characteristics. On the architecture continuum, Cloud architectures are more concrete than The Open Group's SOA reference architecture domain architecture scoped to service delivery and management. Principles and architectural decisions have been pre-made

Volume : 2 | Issue : 2 | November 2012 | ISSN - 2249-555X

already to enable the Cloud computing architecture to be self service, network accessible, and scalable. Architectural building blocks have already been identified for Cloud solution architects to use for operational and business support. In some cases, Cloud service providers may provide well defined, maybe even standardized, management and security support and services.

For cloud, some service identification has been done and implementations of services may be available from an existing services ecosystem. The existence of this services ecosystem and concrete architecture makes SOA via cloud simpler for service consumers to adopt because the designs and implementations have been provided [2]. The benefit of recognizing the heritage of Cloud from SOA is that the existing experience over the last 5 years and

standards already available for SOA and SOA solutions can be applied to Cloud Computing and Cloud solutions [5].

- The Open Service Integration Maturity Model, this model helps determine the level of service use in an organization; these levels apply to the use of cloud services. Cloud computing can be seen as the 'Virtualized" and "Dynamically reconfigurable" levels.
- The SOA Ontology defines service and SOA concepts which can be used as a basis for describing cloud services, though extension Ontology's should be developed for cloud..
- The SOA Governance Framework defines a governance reference model and method that applies to the development of cloud services and solution portfolio and lifecycle management. Best practices for governance of Cloud solutions will need to be developed in addition to this standard.
- Security for Cloud and SOA, a joint workgroup between SOA and Cloud Workgroups in The Open Group, defines security considerations for both Cloud and SOA.
- SOCCI, another joint SOA and Cloud Workgroup in the Open Group define the architecture for exposing infrastructure as a service for both SOA and Cloud solutions.

Certainly functions that were optional for SOA solutions are now required for Cloud solutions, like virtualization, security across business boundaries, and service management automation. New functions and requirements are getting in focus with cloud driving experiences from the SOA world to the next level.

II. Cloud Computing Reference Architecture

The CCRA defines the fundamental architectural elements constituting a cloud computing environment. The CCRA is structured in a modular fashion in a sense that on its highest level of abstraction, the main roles and the corresponding architectural elements are defined allowing to drill down for each of these elements as needed. Later sections of this document will describe detailed versions of this CCRA Overview diagram, which provide a more fine-grain view of the high-level architectural elements of the overall CCRA are shown in Figure 1. As a result of specifying the CCRA, a base terminology is established, which should be used as a reference for any other cloud computing related effort.

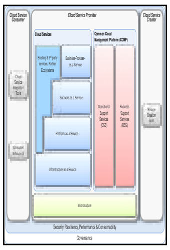

Figure 1: Cloud Computing Reference Architecture Overview

2.1. Roles

The CCRA defines three main roles: Cloud Service Consumer, Cloud Service Provider and Cloud Service Creator. Each role can be fulfilled by a single person or can be fulfilled by a group of people or an organization. The roles defined here intend to capture the common set of roles typically encountered in any cloud computing environment. Therefore it is important to note that depending on a particular cloud computing scenario or specific cloud implementation, there may be project-specific sub-roles defined.

2.1.1. Cloud Service Consumer

A cloud service consumer is an organization, a human being or an IT system that consumes (i.e., requests, uses and manages, e.g. changes quotas for users, changes CPU capacity assigned to a VM, increases maximum number of seats for a web conferencing cloud service) service instances delivered by a particular cloud service. The service consumer may be billed for all (or a subset of) its interactions with cloud service and the provisioned service instance(s).

A service consumer can also be viewed as a kind of super-role representing the party consuming services. For example, in case a credit card company is using some cloud services, the company as a whole is a service consumer relative to the provider. Within the service consumer role more specific roles may exist, such as a technical role responsible for making service consumption work from a technical perspective; and there might be a business person on the consumer side who is responsible for the financial aspects. Of course, in more simplified public cloud scenarios all of these consumer-centric roles could be collapsed into a single person, but the roles still exist.

2.1.2. Cloud Service Provider

The Cloud Service Provider has the responsibility of providing cloud services to Cloud Service Consumers. A cloud service provider is defined by the ownership of a common cloud management platform (CCMP). This ownership can either be realized by truly running a CCMP by himself or consuming one as a service. People acting in the role of a Cloud Service Provider and a Cloud Service Consumer at the same time would be a partner of another cloud service provider reselling cloud services or consuming cloud services and adding value add functionality on top, which would in turn be provided as a cloud service. Although defined as a separate role, it would also be possible that a Cloud Service Provider has Cloud Service Creators in the same organization, i.e. it is not necessary that Cloud Service Provider and Cloud Service Creator are in separate organizations.

2.1.3. Cloud Service Creator

The Cloud Service Creator is responsible for creating a cloud service, which can be run by a Cloud Service Provider and by that exposed to Cloud Service Consumers. Typically, Cloud Service Creators build their cloud services by leveraging functionality which is exposed by a Cloud Service Provider. Management functionality which is commonly needed by Cloud Service Creators is defined by the CCMP architecture. A Cloud Service Creator designs, implements and maintains runtime and management artifacts specific to a cloud service. Just like the Cloud Service Consumer and the Cloud Service Provider, the Cloud Service Creator can be an organization or a human being. For example, an ISV company developing a cloud service is a Cloud Service Creator, whereas obviously there could be 100's of employees within that particular Cloud Service Creator incarnation, each of them taking on a specific business or technically focused sub-role.

It is also typical that that operations staff responsible for operating a cloud service is closely integrated with the development organization developing the service This is an important aspect to achieve the delivery efficiency expected from cloud services as it allows a very short feedback loop to implement changes in the cloud service which benefit the operational efficiency of the cloud service.

Volume : 2 | Issue : 2 | November 2012 | ISSN - 2249-555X

III. Infrastructure

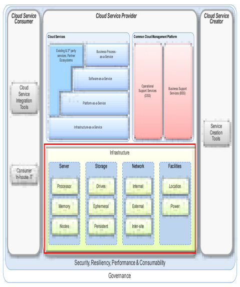

Figure 2: CCRA – Infrastructure Details

"Infrastructure" represents all infrastructure elements needed on the cloud service provider side, which are needed to provide cloud services. This includes facilities, server, storage, and network resources, how these resources are wired up, placed within a data center, etc [3]. The infrastructure element is purely scoped to the hardware infrastructure; therefore it does not include software running on top of it such as hypervisors. Consequently it also does not include any virtualization management software. The decision whether the infrastructure is virtualized or not depends on the actual workload characteristics to be run on the respective infrastructures: For many workloads (e.g. compute & storage as-a-Service) it is very convenient to virtualizes the underlying infrastructure, especially since virtualization enables some use cases which can basically not be realized with a physical infrastructure (e.g. all use cases related to image management or dynamic scaling of CPU capacity as needed). For other workloads (e.g. analytics/search) it is required to have maximum compute capacity and use 100's or 1000's of nodes to run a single specialized workload. In such cases a non-virtualized infrastructure is more appropriate.

3.1 CCRA Architectural Principles

The following top-level architectural principles guide the definition of any cloud implementation, with a focus on delivery & management of cloud services. In this work they are focused on the CCMP element of the overall architecture as this element is required consistently, independent of which cloud service is implemented, delivered & managed:

1. Design for Cloud-scale Efficiencies: When realizing cloud characteristics such as elasticity, self-service access, and flexible sourcing, the cloud design is strictly oriented to high cloud scale efficiencies and short time-to-delivery/time-to-change. ("Efficiency Principle")
2. Support Lean Service Management: The Common Cloud Management Platform fosters lean and lightweight service management policies, processes, and technologies. ("Lightweight ness Principle")
3. Identify and Leverage Commonalities: All commonalities are identified and leveraged in cloud service design. ("Economies-of-scale principle")

IV. CONCLUSIONS

To really implement a cloud following this principle with that high level of efficiency and flexibility, a very high degree of standardization (i.e. minimal variety in the data center with respect to numbers of server, storage, network technologies, operating systems & versions, middleware products, applications, etc.) is required to enable high degrees of automation. The higher the degree of standardization / minimization of variety are, the better automation can be realized Obviously, in a highly homogeneous public cloud data center this can be achieved in a better way compared to private cloud enterprise data centers running a variety of workloads each of them having different requirements, so there is typically a trade-off between degree of standardization and level of efficiency.

IT services have to be defined with the highest degree of standardization and documented in a comprehensive IT service catalogue. For each cloud service every service request has to be described and implemented to be run on an automated basis. A cloud management platform has to be designed and implemented that is able to manage the status of every operational task and all operational deviations.

Examples

Infrastructure costs can be significantly reduced if their utilization is increased. That is quite easy to be done by leveraging the degree of standardization of the infrastructure components as well as the whole IT services. The IT services are run on a standardized and shared IT environment that keeps IT costs low. Deliveries as well as change tasks for example today are too often based on manual work and coordination. In a cloud environment most of these typical tasks can be automated; manual work is only necessary in case of exceptional situations. Keeping manual effort and therefore labor costs low is achieved by a high level of automation.

REFERENCE 1] Open Group, The Open Group Architecture Framework (TOGAF), Definition of the term "Architectural Principle", http://www.opengroup.org/architecture/togaf8-doc/arch/chap29.html | [2] Barroso A., Hölzle U., the Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines | [3] Amazon, AWS and EC2 resources | [4] Meyer, B., Object-Oriented Software Construction, 2nd edition. Prentice Hall, 2000. | [5] National Institute of Standards and Technologies; Draft NIST Working Definition of Cloud Computing, May 14, 2009 | |