

# Semantic Cloud Service Lifecycle for it Services

M. Shanmugapriya<sup>1</sup>, T. Ramesh<sup>2</sup>

1. PG Scholar (CSE)

2. Assistant Professor (CSE)

Department of Computer Science and  
Engineering  
RMK Engineering College

**Abstract** - Cloud computing is the new technology that can access the data and applications from anywhere. The service is non-inheritable on associate as-needed basis and might be termed as service on demand. Typically, the service is hosted on a cloud or a computing grid and is delivered to the organization via the net or mobile devices. Maintaining a conventional IT infrastructure is progressively unable to accommodate the growing range of private devices such as tablets, good phones, and laptops that students bring into the field. To resolve this drawback the projected system presents a brand new integrated methodology for the life cycle of IT services delivered on the cloud and repair necessities then automatism, service acquisition and consumption from the cloud. The IT services life cycle is divided into five phases as requirements, discovery, negotiation, composition, and consumption. The IT service lifecycle is developed and demonstrated by using a semantic web technology which includes a cloud storage service prototype. The archetype allows cloud consumer to discover and acquire disk storage on the cloud by specifying the service attributes, security policies and compliance policies. The proposed methodology will enable the practitioners to plan, create and deploy virtualized services successfully. The key reason to have a semantically rich approach is to automate the process of acquisition and consumption of cloud services.

**Keywords**- *Semantic Web, life cycle, ontology design, web-based services*

## 1. INTRODUCTION

Cloud computing springs up as a new computing paradigm which aims to provide reliable, modifiable and QOS guaranteed computing fledging environment for end users. Distributed processing, parallel processing and grid computing combined together to form the cloud computing. In cloud computing, the user data is not stored locally instead it is stored in the data center of internet.

The companies which provide cloud computing service could carry off and stabilize the operations of these data centers.

NIST [1] characterizes cloud computing in terms of on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. The cloud service models are classified into "Software as a service" (SaaS), "Platform as a service" (PaaS) and "Infrastructure as a service" (IaaS). In SaaS, the consumer can use the provider's

application which is stored in the cloud infrastructure. In PaaS, the consumer gets the platform that is running in the cloud infrastructure for creating or deploying the applications. In IaaS, the consumer gets the virtual resources in which an execution environment can be deployed to run an application.

Grid computing [2] is the bureau of computer resources from multiple administrative disciplines to solve scientific applications involving high-performance computing (HPC). It consists of a eclectic, large-scale and fledging environment, and it aims to up cycle the usage of underutilized resources and the parallel processing capability of HPC applications. However, the major snag of grid is its inability to provide the required execution environment due to its firm nature. Thus, it leaves certain user application requirements unsatisfied and this result in the failure or rejection of some jobs. This snag has been collocated through integrating virtualization technology with grid and cloud computing, and provisioning resources in an on-demand manner.

The management of grid and cloud resources is a serious and complex issue. The opacity of the resource management is due to the dynamic nature of scaling and shrinking resources in a grid and cloud environment. The main intention of resource management is to set a mutual agreement between a resource provider and a resource consumer. When the rely grid resources are not accessible, cloud resources are used to serve virtual resources dynamically. And also the prevailing grid resource broker has to bring off both the grid and cloud resources in an interoperable and a worthwhile manner.

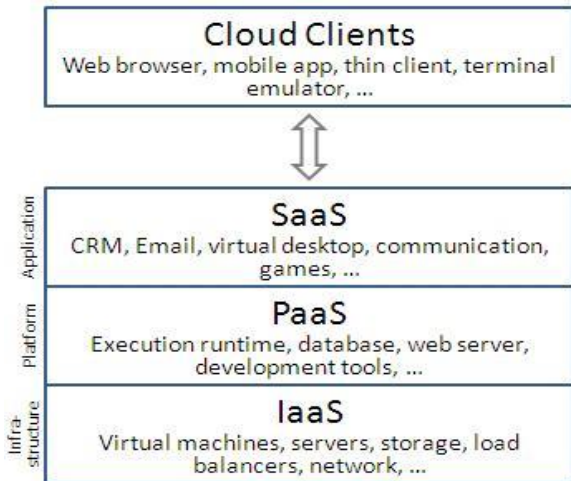


Fig.1. Cloud service models

Moreover, these resources may inhabit in different geographical locations, managed by different organizations, and thus, there are no stabilized interfaces, protocols or commands. To systemize the usage of resources and to solve interoperability issue between different grid and cloud resources, there is a requirement to assimilate semantic technology in the resource broker.

According to Berners-Lee et al [3] a semantic web is a cluster or group of information fixed in a hierarchical structure and it is easily accepted by the computer. In addition, it is an efficient way of supersede data on the World Wide Web (www). Ontology [4] is the language mainly used for representing the available information. Semantic grid [5] is a recent platform to reveal semantically rich information associated with grid resources to assemble more intelligent grid services. Semantic web and semantic grid use a resource description framework [6] to create an ontology knowledge base. It is used as a standard model for interchanging data on the web and grid. It is able to amalgamate data even when the hidden schemas are diverse in nature. Web ontology language (OWL) [7] is one of the top-rated ontology language recommended by the world wide consortium (w3c). It is a combination of the classes, properties and their instances.

## 2. PROPOSED METHOD

The proposed methodology will enable practitioners to plan, create, and deploy virtualized services successfully. The key reason to have a semantically rich approach to describe cloud attributes and service-level agreements (SLA) is to permit distributed clients and cloud service providers to automate the process of acquisition and consumption of services. The service attributes are the storage size, backup rules, service availability, and service costs. Specifications

also list acceptable security levels, data quality, and performance levels of the service software. Proposed system used W3C standard Semantic web technologies, such as Web Ontology Language (OWL), Resource Description Framework (RDF), and SPARQL, to develop our prototype system since they enable us to build the vocabulary (or ontology) of our service life cycle using standardized languages that support our design requirements, which include interoperability, sound semantics, web integration, and the availability of tools and system components. The OWL language has a well-defined semantics that is grounded in first-order logic and model theory. It is possible to embed RDF and OWL knowledge in HTML pages and several search engines (including Google) will find and process some embedded RDF.

### 2.1 System Modules:

- Service Requirement
- Service Discovery
- Service Negotiation
- Service Composition
- Service Consumption

#### Implementation Description

1. Planning a workflow of individual service types.
2. Locating services from a service registry, i.e., finding service instances.
3. Selecting the best candidate services based service agent for deployment and execution.
4. Executing the selected services. When an exception occurs during execution, services might have to be retried or the planning and selection stages might have to be redone.

##### 2.1.1 Service Requirement:

In the service requirement phase, the consumer details the technical and functional specifications that a service needs to full fill. Service compliance details like certifications needed, standards to be adhered to, and so on, are also identified. The technical specifications lay down the hardware, software, application standards, and language support policies to which a service should adhere. Once the consumers have identified and classified their service needs, they issue a Request for Service (RFS). This RFS can be generated in a machine readable format using Semantic Web technologies.

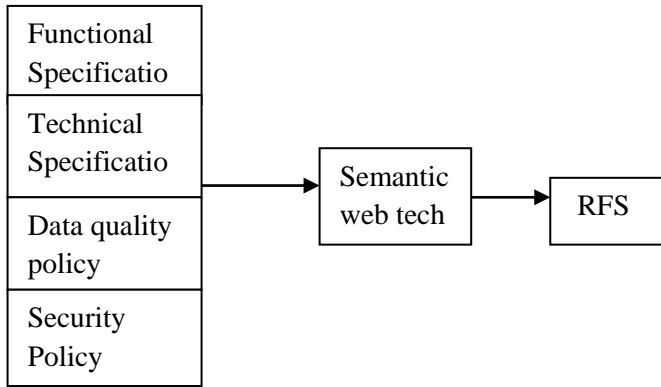


Fig.2. Service requirement

2.1.2 Service Discovery:

In the service discovery phase, providers are discovered by comparing the specifications listed in the RFS with service descriptions. The discovery is constrained by functional and technical attributes defined, and also by the budgetary, security, compliance, data quality, and agent policies of the consumer. An organization can release the RFS to a limited pre-approved set of providers. Alternatively, it can search for all possible vendors on the Internet. While searching the provider, service search engines or cloud brokers can be employed. A “cloud broker” role has been identified by cloud server. This cloud broker runs a query against the services registered with a central registry or governing body and matches the service layer, domain, data type, compliance needs, and functional and technical specifications and returns the result with the service providers matching the maximum number of requirements listed at the top.

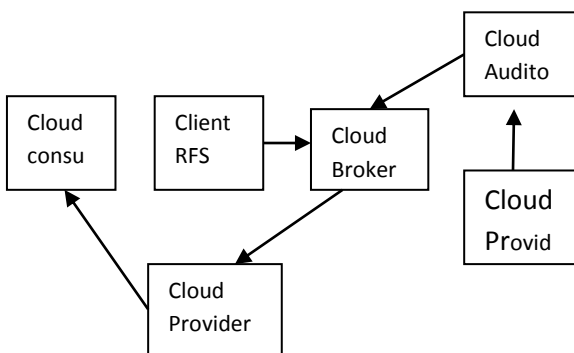


Fig.3. Service discovery

2.1.3 Service Negotiation:

In this module the consumer sends an RFS to the provider specifying the functional and non functional requirements. The provider responds to the RFS in one of three ways:

- Informs the consumer that it cannot provide the service, terminating negotiation.
- Indicates that a service matching all the requirements exists and sends the quote with SLAs.
- Indicates that there is a partial match of requirements and sends the quote with SLA file listing matching constraints.

The consumer receives and considers the quote. Then the consumer responds to the quote in one of three ways:

- If the quote is a partial match, the consumer relaxes the service constraints and/or functionality and resends the RFS to the provider. The provider repeats the actions in step 2.
- If the response is a full match and the consumer is satisfied with the offer then negotiation is regarded complete. The consumer signs this offer and returns it as an SLA.
- The consumer can decline the service, terminating the negotiation.

The provider responds to the RFS in one of two ways:

- The provider can no longer provide the service, and rejects the agreement, terminating negotiation.
- The provider agrees with the constraints, and the same RDF file consisting of the SLA now exists with both parties.

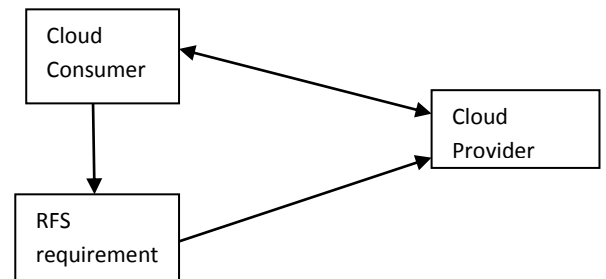


Fig.4. Service negotiation

2.1.4 Service Composition

In this phase, one or more components provided by one or more providers are combined and delivered as a single service to the service consumer. Service orchestration determines the sequence of the service components. Service Composition (SC) requires a computer program to automatically select, integrate, and invoke multiple web services in order to achieve a user-defined objective the composite service is composed of human agents providing the service, the service software, and dependent service components. All the three elements, agents, software, and

dependent services, must be monitored to manage the overall service quality. The service class takes inputs from the specification, service contracts, and service-level agreement classes defined in the earlier phases to determine the orchestration of the various components.

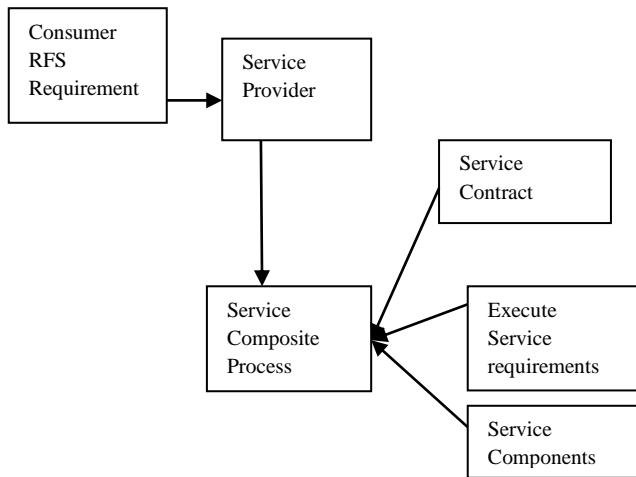
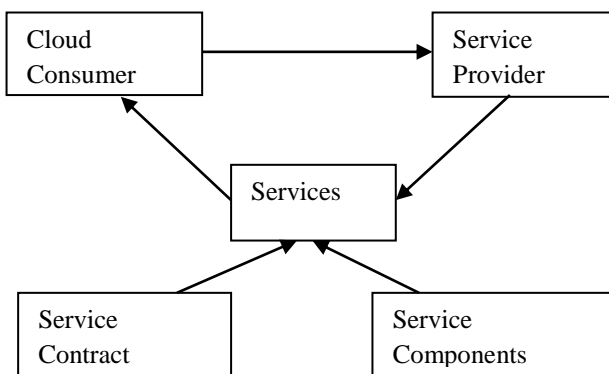


Fig.5. Service composition

2.1.5 Service Consumption

The service is delivered to the consumer based on the delivery mode agreed upon in the negotiation phase. After the service is delivered to the consumer, payment is made for the same based on the pricing model agreed to in the SLA. The consumer then begins consuming the service. In a cloud environment, the service usually resides on remote machines managed by the service providers. In this phase, consumer will require tools that enable service quality monitoring and service termination if needed. This will involve alerts to humans or automatic termination based on policies defined using the quality-related ontologies. The service monitor measures the service quality and compares it with the quality levels defined in the SLA. This phase spans both the consumer and cloud areas as performance monitoring is a joint responsibility. If the consumer is not satisfied with the service quality, she/he should have the option to terminate the service and stop service payment.



3. SYSTEM IMPLEMENTATION

Semantic Web technologies to build the front end of our prototype as they are platform independent and interoperable. Proposed system used SPARQL, Jena Semantic Web framework which is an HTTP engine that supports the SPARQL Protocol and the SPARQL RDF Query language, to develop the prototype. After defining our service, this system created a SPARQL endpoint using Sesame framework to simulate a service provider providing the service. Since the Sesame server allows multiple service definitions, we used it to simulate both multiple services provided by the provider as well as multiple instances of a same service. The Sesame service database contained the service description along with the provider policies endpoint. System is using our service life-cycle ontology that we described in the previous section and the OWL-S ontology to develop the tool. In addition to these two ontologies, system also created OWL ontology to describe the technical and security policies for our prototype.

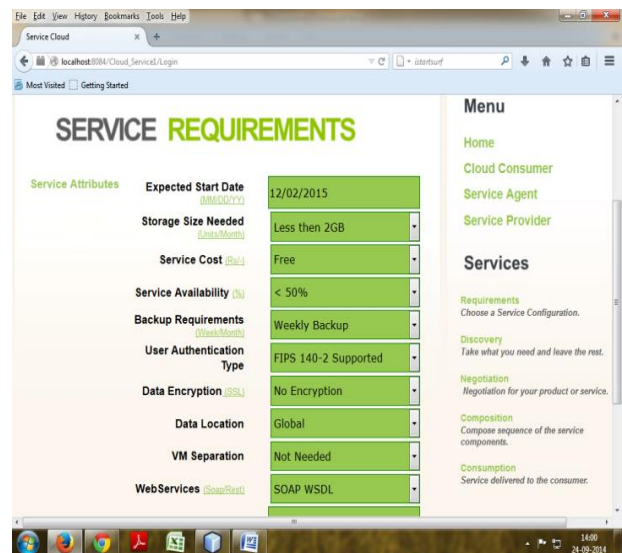


Fig.6. Requirements

ADVANTAGES

- The biggest benefit of resource allocation is that user neither has to install software nor hardware to access the applications and to develop the application over the internet.
- Cloud providers can share their resources with the users over the internet even in case of resource scarcity
- The user does not need to expend on hardware and software systems.

- The next major benefit is that there is no limitation of place and medium. We can use our applications and data from anywhere in the world and also using any system.

#### LIMITATIONS

- The users only rent the resources from remote servers for their use, so they don't have control over their resources.
- Migration problem occurs, when the users wants to switch to some other provider for the better storage of their data. It is difficult to transfer huge data from one provider to the other.
- In public cloud, the clients' data can be susceptible to hacking. Since the servers on cloud are interconnected, so it is easy to spread any infections.
- For allocating and managing resources in cloud, the cloud admin must have high knowledge since all knowledge about the working of the cloud mainly depends upon the cloud service provider.

#### 4. CONCLUSION

In cloud paradigm, an effective resource allocation strategy is required for achieving user satisfaction and maximizing the profit for cloud service providers. The proposed system will demonstrate how our methodology can be used to significantly automate the acquisition and consumption of cloud-based services thereby reducing the large time required by companies to discover and procure cloud-based services. The proposed system will demonstrate how our methodology can be used to significantly automate the acquisition and consumption of cloud-based services thereby reducing the large time required by companies to discover and procure cloud-based services.

#### REFERENCES

- [1]. National Institute of Standards and Technology (NIST).[http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145\\_Cloud-definition.pdf](http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_Cloud-definition.pdf). Accessed Jan 2013
- [2]. Price Water House Coopers (2002) Powerful technology trends continue despite downturn. PWC Global Technology Center, Menlo Park.
- [3]. Berners-Lee T, Hendler J, Lassila O (2001). The semantic web. *Sci Am* 284(5):34-43
- [4]. D. Bianchini, V. De Antonellis, B. Pernici, and P. Plebani, "Ontology-Based Methodology for E-Service Discovery," *Int'l J. Information Systems, Special Issue: The Semantic Web and Web Services*, vol. 31, nos. 4/5, pp. 361-380, June/July 2006.
- [5]. J. Black et al., "An Integration Model for Organizing IT Service Management," *IBM Systems J.*, vol. 46, no. 3, pp. 405-422, 2007.
- [6] Apache Software Foundation, "Jena - A Semantic Web Framework for Java," <http://incubator.apache.org/jena/>, Mar. 2012.
- [7]. K. Joshi, T. Finin, and Y. Yesha, "Integrated Lifecycle of IT Services in a Cloud Environment," *Proc. Third Int'l Conf. Virtual Computing Initiative (ICVCI '09)*, Oct. 2009.
- [8]. Xen. <http://www.xen.org/>. Accessed Mar 2013.
- [9]. VMware. <http://www.vmware.com/>. Accessed Mar 2013.
- [10]. Kernel-based Virtual Machine (KVM). <http://www.linux-kvm.org/>. Accessed Mar 2013.
- [11]. Amazon EC2 Cloud. <http://www.amazon.org/ec2>. Accessed Mar 2013.
- [12]. Ma YB, Jang SH, Lee JS (2011) Ontology-based resource management for cloud computing. In: *Intelligent information and database systems. Lecture Notes in computer science*, vol 6592. Springer, Berlin, pp 343-352