Integrated Lifecycle of IT Services in a Cloud Environment

Karuna P Joshi, Tim Finin, Yelena Yesha,
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250, USA
{kjoshi1,finin, yeyesha }@cs.umbc.edu

Abstract*— Virtualized Service models are now emerging and redefining the way information technology is delivered to end users. Current research is focused on specific pieces like service discovery, composition etc. There is no holistic view of what would constitute a lifecycle of virtualized services delivered on a cloud environment. In this paper, we propose an integrated methodology that covers the entire service lifecycle. We have divided the IT service lifecycle on the cloud into five phases of requirements, discovery, negotiation, composition, and consumption. We describe each phase and it's sub-phases in detail along with the information that needs to flow between them. We also propose some key metrics that should be tracked for each phase.

Keywords-component; Services; Methodology; lifecycle

I. Introduction and Background

The development and maintenance of Information Technology (IT) was in the past regarded as an integral part of any organization. It has now been outsourced by most companies to external consulting/staffing companies or service providers. This outsourcing model initially led to off-shoring of tasks to providers due to their specialized expertise and/or labor arbitrage. It is now being replaced by a new delivery model where businesses purchase IT components like software, hardware or network bandwidth as services from providers who can be based anywhere in the world. The service is acquired "on demand". Increasingly it does not even reside within the organization, but remotely on the third party servers and delivered to the organization via the Internet on fixed or mobile end devices.

In such scenarios, multiple providers often collaborate to create a single service for an organization. In many cases, a business may seemingly utilize a single provider, who in turn utilizes the services of other providers for parts of the service. In some other cases, businesses utilize multiple service providers to mitigate risks that may be associated with a single provider. In either case, the delivery of IT service is moving away from a single provider mode, and is increasingly based on the composition of multiple services and assets (technological, human, or process) that may be supplied by one or more service providers distributed across the network. Often, a single provider in the cloud seemingly provides the service, and each component service might be virtualized and participate in many composite service orchestrations. The service, in effect, is *virtualized on the cloud*. This virtualized model of service delivery [1] allows easier customization, better utilization and greater responsiveness. It is becoming the preferred method to deliver services ranging from helpdesk and back-office functions to Infrastructure as a Service (IaaS). Indeed, the virtualized model of service delivery even extends to IT Enabled Services (ITeS) which typically include a large human element.

A key barrier preventing organizations from successfully managing services on the cloud is the lack of an integrated methodology for service creation and deployment that would provide a holistic view of the service lifecycle on a cloud. In this paper we present a methodology to address the lifecycle issue for virtualized services delivered from the cloud. We use semantically rich descriptions of the requirements, constraints, and capabilities that are needed by each phase of the lifecycle. These can be reasoned over to automate the phases guided by high level policy constraints provided by consumers, service customers, or service providers. To judge if the lifecycle is progressing successfully, we have proposed metrics for each phase of the lifecycle. This methodology will enable practitioners to create and deploy virtualized services, and measure their success using the associated metrics.

In Section 2 we review the existing methodologies and their limitations when applied to virtualized IT services. In section 3 we detail our methodology which integrates the entire service lifecycle along with the associated metrics. We conclude with our ongoing work in this field.

II. Related Work

At present there is no integrated methodology for the entire service lifecycle covering service planning, development and deployment in virtualized environments. Most approaches are limited to exploring a single aspect of the lifecycle like service discovery, service composition or service quality. In addition, most of the work is limited to the software component of the service and does not cover the service processes or human agents which are a critical component of IT services.

Papazoglou and Heuvel [2] have proposed a methodology for developing and deploying web services using service oriented architectures. Their approach, however, is limited to the creation and deployment of web services and does not account for virtualized environment where services are composed on demand. Providers may need to combine their services with other resources or providers' services to meet consumer needs. Other methodologies, like that proposed by Bianchini et al. [3], do not provide this flexibility and are limited to cases where a single service provider provides one service. Zeng et al. [4] address the quality based selection of composite services via a global planning approach but do not cover the human factors in quality metrics used for selecting the components. Maximilien and Singh [5] propose an ontology to capture quality of a web service so that quality attributes can be used while selecting a service. While their ontology can serve as a key building block in our system, it is limited by the fact that it considers single web services, rather than service compositions. Black et al. [15] have proposed an integrated model for IT service management. Their model is limited to managing the service from the service provider's perspective.

Information Technology Infrastructure Library (ITIL) is a set of concepts and policies for managing IT infrastructure, development and operations that has wide acceptance in the industry today. The latest version of ITIL lists policies for managing IT services [9] that cover aspects of service strategy, service design, service transition, service operation and continual service improvement. However, it is limited to interpreting 'IT services' as products and applications that are offered by in-house IT department or IT consulting companies to an organization. This framework in its present form does not extend to the service cloud or a virtualized environment that consists of one or more composite services generated on demand.

A. Service Ontologies

In a virtualized service-oriented environment, consumers and providers need to be able to exchange information, queries, and requests with some assurance that they share a common meaning. This is critical not only for the data but also for the policies that consumers or providers of the service have. The handling of heterogeneous policies is usually not present in a closed and/or centralized environment, but is an issue in the open cloud. The interoperability requirement is not just for the data itself, but even for the describing services, their quality related measures and attributes, and their policies for sharing data.

One possible approach to this issue, that we have used, is to employ Semantic Web techniques for modeling and reasoning about services related information. *Semantic Web* is an enhancement of the World Wide Web that deals primarily with data instead of documents. It enables data to be annotated with machine understandable meta-data, allowing the automation of their retrieval and their usage in correct contexts. Semantic Web technologies include languages such as Web Ontology Language (OWL) [11] and Resource Description Framework (RDF) [13] for defining ontologies and describing meta-data using these ontologies as well as tools for reasoning over these descriptions. These technologies can be used to provide common semantics of service information and policies enabling all agents who understand basic Semantic Web technologies to communicate and use each other's data and services effectively.

The Web Ontology Language OWL is a family of knowledge representation languages based on Description Logic (DL) [14] with a representation in RDF. OWL supports the specification and use of ontologies that consist of terms representing individuals, classes of individuals, properties, and axioms that assert constraints over them. The axioms can be realized as simple assertions (e.g., *TelephoneRequest is a sub class of SupportRequest, Tier 1 and Tier2 are subclasses of Agents, Tier1 and Tier2 are disjoint classes, assignRequest is a property from SupportRequest to Agent, escalateRequest is a property from Agent to Agent*) and also as simple rules.

The Semantic Annotations for WSDL and XML Schema (SAWSDL) [10] W3C Recommendation defines mechanisms using which semantic annotations can be added to WSDL components. WSDL 2.0 does not include semantics in the description of web services. This can lead to a situation where two services with similar descriptions can mean totally different things, or they can have very different descriptions yet similar meaning. Eliminating semantic ambiguities in web services descriptions is a critical requirement in automating the discovery and composition of web services. SAWSDL doesn't specify a language for representing the semantic models, but provides mechanisms by which concepts from the semantic models that are defined either within or outside the WSDL document can be referenced from within WSDL components as annotations. These semantics when expressed in formal languages can help disambiguate the description of web services during automatic discovery and composition of the web services. We use the SAWSDL infrastructure to add additional descriptions and constraints to service descriptions as listed in the next section. We have developed ontology [16] to capture service quality elements and their relationships.

The Semantic Web has also been used to express policies that govern services or information access and usage policies. Semantic Web-based policy languages promote common understanding of quality requirements among participants who might not use the same information or data model. The use of these languages to define policies has several very important advantages that become critical in cloud environments involving coordination across multiple organizations. First, most policy languages define constraints over classes of targets, objects, actions and other constraints (e.g., time or location). A substantial part of the development of a policy is often devoted to the precise specification of these classes, e.g., the definition of what counts as a *CMM Level 5 provider* or being a *Sarbanes-Oxley Compliant provider*. Using Semantic Web languages means that not only are we using W3C standards, we can reuse the work done by others in defining ontologies by importing or extending them. The second advantage is that the grounding of the languages in (description) logic facilitates the translation of policies expressed in them to other formalisms, either for analysis or for execution.

III. Proposed Methodology

Extant methodologies for service development do not account for a cloud environment, which includes services composed on demand at short notice. Currently, the service providers decide how the services are bundled together and delivered to service consumers. This is typically done statically by a manual process. There is a need to develop reusable, user-centric mechanisms that will allow the service consumer to specify their desired security or quality related constraints, and have automatic systems at the providers end control the selection, configuration and composition of services. This should be without requiring the consumer to understand the technical aspects of services and service composition.

For our methodology, we divide the IT service lifecycle on the cloud into five phases. In sequential order of execution they are requirements, discovery, negotiation, composition, and consumption. Figure 1 illustrates our proposed service lifecycle. Metrics should be tracked for each phase of the lifecycle to ensure that the phase is successfully completed. In the following sections we define the phases in detail and also identify some of the key metrics that must be monitored for each phase. We have also developed ontology in OWL for our proposed lifecycle which can be accessed at [16].

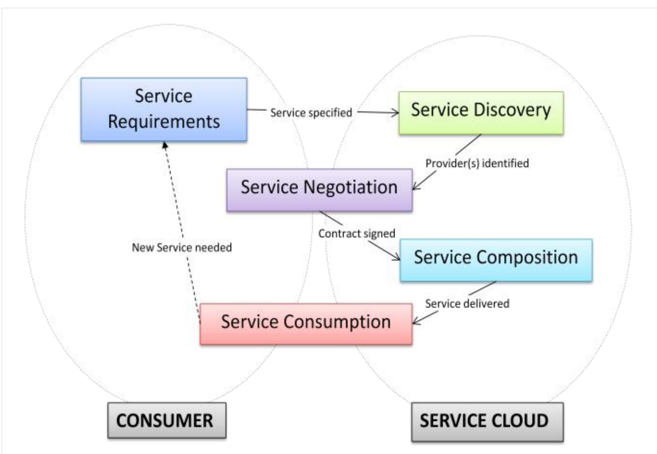

Figure 1: The IT service lifecycle on a virtualized cloud comprises five phases: requirements, discovery, negotiation, composition and consumption.

A. Service Requirements

In the service requirements phase the consumer details the technical and functional specifications that a service needs to fulfill. While defining the service requirements, the consumer specifies not just the functionality, but also non-functional attributes such as characteristics of the providing agent, constraints and preferences on data quality, service compliance and required security policies for the service. Depending on the service cost and availability, a consumer may be amenable to compromise on the service quality. For example, a simple service providing stock prices might deliver data of varying quality, with high quality data tracking instantaneous price and low quality capturing the share price at the close of the business the previous day. Depending on their requirements, service consumers may be interested in the high quality data (e.g. hedge funds, day traders etc.) or might be fine with low quality (e.g. Mutual Fund buyers).

Such explicit descriptions are of use not just for the consumer of the service, but also the provider. For instance, the cost of maintaining the service data quality can be optimized depending on the type of data quality requested in the service. The advantage for the service provider is that they will not need to maintain the lower quality data with the same efficiency as the higher quality data; but they would still be able to find consumers for the data. They can separate the data into various databases and make those databases available on demand. As maintainability is a key measure of quality, low maintenance need of the service data will result in improved quality of the service.

The service requirement phase consists of two main sub-phases that are listed below and illustrated in Figure 2.

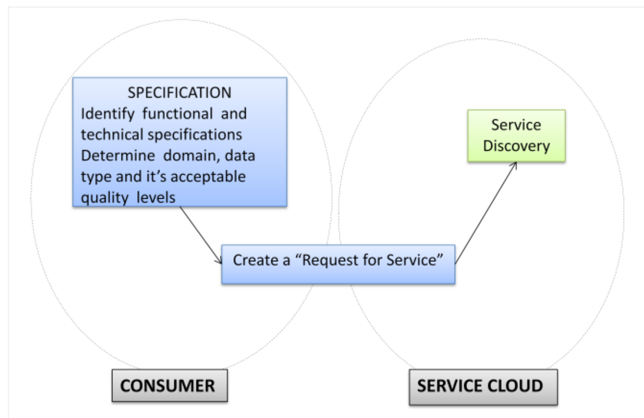

Figure 2: The service requirements phase consists of generating a specification of the functional and non-functional requirements and sending request for services to potential providers or service brokers.

Service Specification: In this sub-phase, the consumer identifies the domain (for instance banking, airline, etc.) of the service needed and details the functional and technical specifications for the same. Functional specification describe in detail what functions/tasks should a service help automate and the acceptable performance levels of the service agent and the service software. The technical specifications lay down the hardware, operating system, application standards and language support policies that a service should adhere to. Specifications also list acceptable security levels, data quality and performance levels of the service agent and the service software. Service compliance details like required certifications, standards to be adhered to etc. are also identified. For instance a simple service specification could be *"Service should provide hotel reservation functionality manned by one agent conversant in English and French on a 24x7 basis. The service application should be accessible using* *a mobile phone or a Laptop and compatible with MS Windows and Oracle database platform."* Depending on the requirements, specifications can be as short or as detailed as needed. To achieve this, service description languages such as WSDL [10] are expanded to include domain, quality, compliance and security related descriptors describing acceptable levels.

Request for Service: Once the consumers have identified and classified their service needs, they will issue a "Request for Service" (RFS). This request could be made by directly contacting the service providers. The consumer can send the RFS to a few service providers that s/he is comfortable with and get quotes from them for the services. Alternatively, consumers can utilize a service search engine on the cloud to procure the desired service. Contract net [17] type approaches can be used for the RFS process.

The key metrics that will be tracked in this phase are shown in Table 1. In our ontology [16] we have defined the specifications and RFS classes. The specification class gets input from other classes that define the functional specs, technical specs, human agent specs, security policies, service compliance policies and data quality policies.

Table 1: Key metrics for requirements phase must ensure that the specifications are comprehensive

Quality Metrics	Definition
Functional specs review	Functional specifications should be approved by the business team in the consumer organization.
Technical specs review	Technical specifications should be approved by the IT team in the consumer organization
Data quality	The quality of the data delivered by the service.
Cost	Cost of the service to the consumer - can be measured as a fixed price cost or on a time and materials basis.
Security	Required security/permission levels for the service

Service requirement is a critical phase in service lifecycle as it defines the "what" of the service. It is a combination of the "planning" and "requirements gathering" phases in a traditional software development lifecycle. The consumers will spend the maximum effort in this phase and so it has been depicted entirely in the consumer's area in our lifecycle diagrams. The consumer could outsource compilation of technical and functional specifications to another vendor, but the responsibility of successful completion of this phase resides with the consumer and not the service cloud. Once the RFS has been issued, we enter the discovery phase of the service lifecycle.

B. Service Discovery

In this phase, service providers that offer the services matching the specifications detailed in the RFS are searched (or discovered) in the cloud. The discovery is constrained by functional and technical attributes defined, and also by the budgetary, security, data quality and agent policies of the consumer.

If the consumer elects the option to search the cloud instead of sending the RFS to a limited set of providers, then the discovery of services is done by using a **services search/discovery engine**. This engine runs a query against the services registered with a central registry or governing body and matches the domain, data type, functional and technical specifications and returns the result with the service providers matching the maximum number of requirements listed at the top. The service search engine would be similar to the ones available currently for web services like Seekda (http://seekda.com/). The OASIS consortium is also working towards standardizing such a search engine (http://www.dlib.org/dlib/january09 /denenberg/01denenberg.html).

One critical sub-phase is **service certification,** in which the consumers will contact a central registry, like UDDI [7], to get verification for providers that they narrow down to.

The discovery phase, detailed in Figure 3, may also not provide successful results to the consumers and so they will need to either change the specifications or alter their in-house processes to be able to consume a service that meets their needs the most.

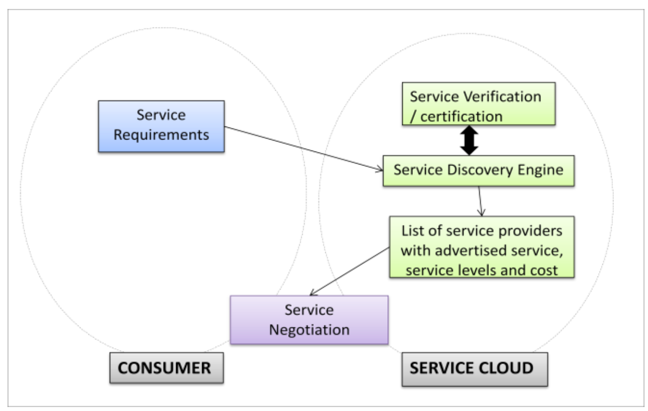

Figure 3: Service providers are searched and verified in the service discovery phase

Some of the key metrics that should be tracked in the discovery phase are defined in Table 2. The main deliverable of this face is to identify the services that that fulfill most of the requirements and also identify the gaps that exist between the consumer's specifications and the service capabilities.

If the consumers find the exact service within the budget that they are looking for, they can begin consuming the service. However, often the consumers will get a list of providers who will need to compose a service to meet the consumer's specifications. The consumer will then have to begun negotiations with the service providers which is the next phase of the lifecycle. Each search result will also return the primary provider who will be negotiating with the consumer. It will usually be the provider whose service meets most of the requirement specifications.

Table 2: Key Metrics for discovery phase help identify the services that match the requirements and their gap.

Quality Metrics	Definition
Service Gap	The gaps that exist between the consumer's requirements and functionality of the services available off the shelf.
Data quality	The quality of the data delivered by the service.
Cost	Cost of the service to the consumer. While discovering a service, there may be a cost limitation imposed by the consumer.
Certificate	Certification of the service provider to be able to meet service requirements and constraints.

C. Service Negotiation

Service negotiation phase, illustrated in figure 4, covers the discussion and agreement that the service provider and consumer have regarding the service delivered and its acceptance criteria. The service delivered is determined by the specifications laid down in the RFS. Service acceptance is usually guided by the **Service Level Agreements** (SLA) [12] that the service provider and consumer agree upon. SLAs define the service data, delivery mode, agent details, compliance policy, quality and cost of the service. While negotiating the service level with potential service providers, consumers can explicitly specify service quality constraints (data quality, cost, security, response time, etc.) that they require.

A consumer may compromise on data quality if it ensures cost saving or may be agreeable to have provider advertisements displayed on the screen if the service is delivered at reduced/no cost. SLAs will help in determining all such constraints and preferences and will be part of the service contract between the service provider and consumer.

At times, the service provider will need to combine a set of services or compose a service from various components delivered by distinct service providers in order to meet the consumer's requirements. The negotiation phase also includes the discussions that the main service provider has with the other component providers. When the services are provided by multiple providers (composite service), the primary provider interfacing with the consumer is responsible for composition of the service. The primary provider will also have to negotiate the **Quality of Service** (QoS) with the secondary providers to ensure that SLA metrics are met.

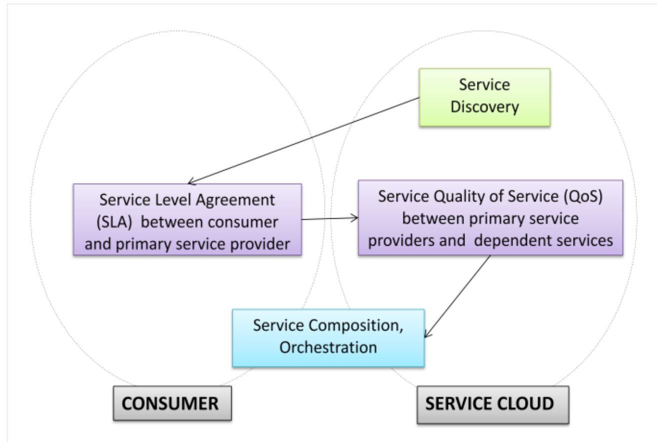

Figure 4: Service negotiation involves discussions between consumers and providers to determine the SLA and QoS

Table 3 lists the key metrics that should be tracked during the negotiation phase to ensure a successful contract negotiation.

Table 3: Key metrics of negotiation phase help define the service contract

Quality Metrics	Definition
Service Gap	The gaps that exist between the consumer's requirements and functionality of the services available off the shelf.
SLA	Service Level Agreement between consumer and primary provider. Includes security policy and data quality policy.
QoS	Quality of service agreement between primary provider and component providers.
Delivery mode	Service delivered in real-time, batch mode or as a one-time service.
Payment options	Service payment will be up-front or on a periodic basis (monthly, quarterly, annual etc.). Depending on the option selected, the service will be delivered before or after payment.
Certificate	Service certificate for the service will be issued by an independent body.

Thus the negotiation phase comprises of two critical sub phases, the negotiation of SLAs and negotiation of QoS. If there is a need for composite service, lots of iterative discussions takes place between the consumer and primary provider and the primary provider and component providers. The final product of the negotiation phase is the service contract between the consumer and primary provider and between the primary provider and the component (or secondary) providers. Once the service contract is approved, the lifecycle goes to the composition phase where the service software is compiled and assembled.

D. Service Composition, Orchestration

In this phase, illustrated in figure 5, one or more services provided by one or more providers are combined and delivered as a single service. Service orchestration determines the sequence of the service components.

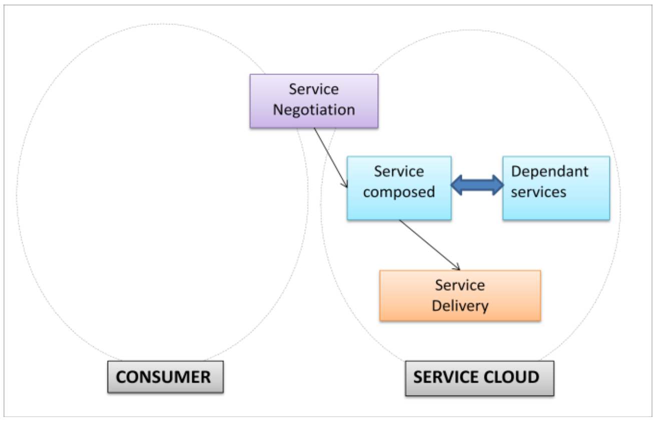

Figure 5: One or more services are combined and delivered as a single service in the service composition phase

Many times what is advertised as a single service by a provider could in turn be a virtualized composed service consisting of various components delivered by different providers. The consumer neither knows that the service is composite, nor needs to care. The provider will have to monitor all the other services that it is dependent on (like database services, network services etc.) to ensure that the SLAs defined in the previous phase are adhered to.

To illustrate a composite service, consider a collaboration tool service that is provided via the Internet. Figure 6 illustrates this service. The service provides capability to its consumers to collaborate online allowing them to conduct meetings, simultaneously work on documents, chat as a group, etc. As the service is completely automated, it has no human agents who provide the service. The service providers have a high degree of separation from the consumers who only contact them for technical assistance or if the tool performance is below par. The service is highly dependent or tightly coupled with the underlying core services, like databases and network, for its successful delivery.

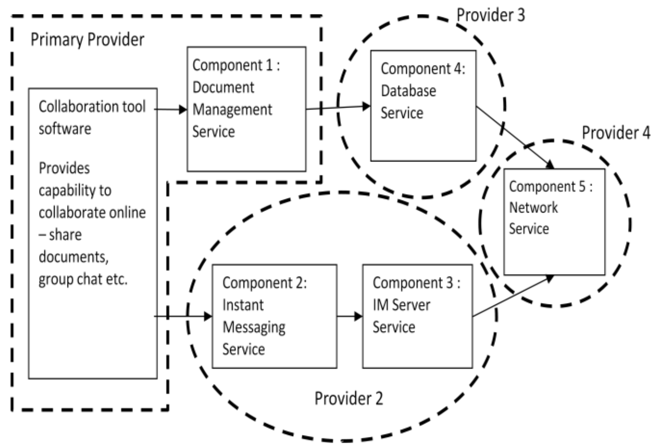

Figure 6: Composite Service consist of one or more components provided by one or more providers that are combined on demand as a single service for the consumers.

While the consumers will view the collaboration tool as a single service, it is actually composed of small components provided by different providers. The primary provider will be supplying the user interface and some of the features, and would in turn be procuring the "group chat" functionality as a service from a different provider. Similarly it could be relying on other providers for the database and network components of the service. QoS levels agreed upon by the Primary providers and component providers will be critical in maintaining high performance of the service.

Table 4 lists some of the key metrics that should be tracked in the composition phase to ensure that the components are combined optimally. .

Table 4: Metrics for composition phase ensure that the service components are optimally combined

Quality Metrics	Definition
Coupling	Coupling determines how dependent the service is on other services or resources for its delivery [8]
Cohesion	Cohesion measures the extent to which related aspects of a requirement are kept together in the same service, and unrelated aspects are kept out.
Security	Required security/permission levels for the service

Once the service is composed, it is ready to be delivered to the consumer. The lifecycle then enters the final phase of service consumption.

E. Service Consumption and Monitoring

The service is delivered to the consumer based on the delivery mode (synchronous/asynchronous, real-time, batch mode etc.) agreed upon in the negotiation phase. After the service is delivered to the consumer, payment is made for the same. The consumer then begins consuming the service. An important part of the consumption phase includes performance monitoring using automated tools.

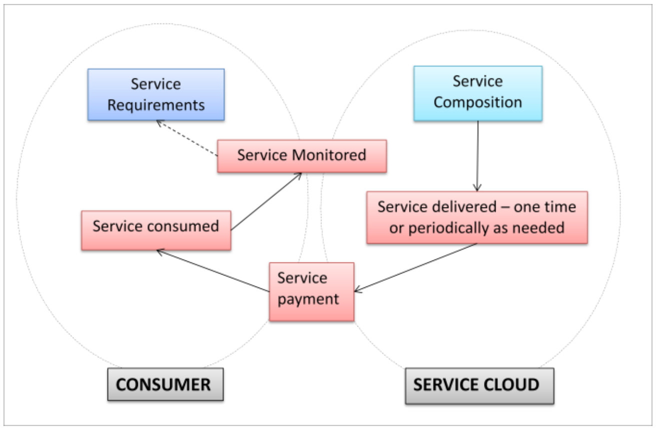

Figure 7: In the service consumption phase, consumers pay and use the service; Service quality is also monitored.

In this phase, consumer will require tools that enable 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 that need to be developed. The **service monitor** sub-phase 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, s/he should have the option to terminate the service and stop service payment. If the service is terminated, the consumer will have to restart the service lifecycle by again defining the requirements and issuing a RFS.

As expected, there are more metrics to be managed in this phase than in other phases. Table 5 lists some of the key metrics that should be tracked to ensure high service quality. There will be more domain specific metrics that the consumers/providers might track. A framework [8] that enables service administrators to automatically track service quality in this phase will be very beneficial.

Figure 8 illustrates the detailed lifecycle and provides a single view that also covers the sub-phases. It also shows the flow of information between them.

Table 5: Metrics for consumption phase monitor the performance of the service to ensure SLAs are met.

Quality Metrics	**Definition**
Delivery	Monitor if service is delivered on time and in mode agreed upon in the Negotiation phase.
Security	Monitor service security to ensure it adheres to the security policies agreed upon in the Negotiation phase
Cost	Monitor cost of the service to ensure it stays within budget
Service payment	Track if payment was done per the payment options agreed to in the Negotiation phase
Reliability	Reliability tracks the service quality to ensure the service functionality and data accuracy is maintained
Performance	Track the service performance that includes throughput, latency and Response Time.
SLA	Track Service Level Agreements defined in Negotiation Phase.
QoS	Track dependent services to ensure Quality of Service defined in Negotiation Phase is being met.
Consumer Satisfaction	Periodically the provider tracks (via survey, opinion poll etc.) if consumers are satisfied with the service.

IV. ONGOING WORK

In this paper we have defined the integrated lifecycle for IT services on the cloud. To the best of our knowledge, this is the first such effort, and it is critical as it provides a "big" picture of what steps are involved in deploying IT services. This methodology can be referenced by organizations to determine what key deliverables they can expect at any stage of the process. We also hope that it will enable the academia and the industry to be in the "same page" when they speak about IT services on the cloud.

In our ongoing work, we are developing the ontology to capture the steps and metrics we have identified in the lifecycle using semantic web languages.

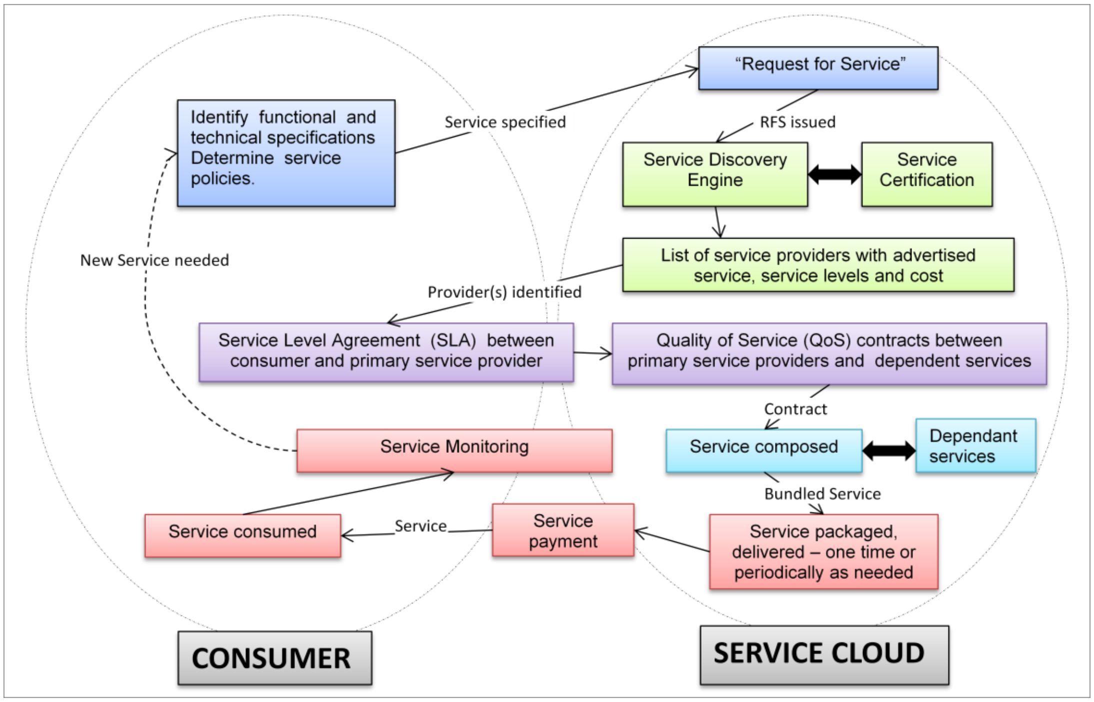

Figure 8: Detailed service lifecycle illustrates all the sub phases and the flow of information between them.

REFERENCES

[1] M Xu, Z Hu, W Long, W Liu, Service virtualization: Infrastructure and applications - The Grid: Blueprint for a New Computing Infrastructure By Ian Foster, Carl Kesselman, Morgan Kaufman, 2004

[2] M. Papazoglou and W. Van Den Heuvel, Service-oriented design and development methodology, International Journal of Web Engineering and Technology, Volume 2, Number 4, 2006, pp. 412 – 442

[3] D. Bianchini, V. De Antonellis, B. Pernici, P. Plebani, Ontology-based methodology for e-service discovery, International Journal of Information Systems, The Semantic Web and Web Services, Volume 31, Issues 4-5, June-July 2006, pp 361-380

[4] L Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, Q. Sheng, Quality driven web ser-vices composition, Proceedings of the 12th international conference on World Wide Web, 2003, pp 411 - 421

[5] E. M. Maximilien, M.Singh, A Framework and Ontology for Dynamic Web Services Se-lection, IEEE Internet Computing, vol. 8, no. 5, pp. 84-93, Sep./Oct. 2004

[6] J. Cardoso, A. Sheth, J. Miller, J. Arnold and K. Kochut, Quality of service for workflows and web service processes, Web Semantics: Science, Services and Agents on the World Wide Web, Volume 1, Issue 3, April 2004, pp 281-308

[7] S Ran, A model for web services discovery with QoS, ACM SIGecom Exchanges, Vol 4, Issue 1, 2003, pp 1-10, 2003

[8] K. Joshi, A. Joshi, Y. Yesha, R. Kothari, A Framework for Relating Frontstage and Backstage Quality in Virtualized Services, UMBC Tech Report TR-CS-09-01, May 2009

[9] J Van Bon et. al., Foundations of IT service management based on ITIL V3 , Van Haten Publishing, 2008

[10] J. Kopecky, T. Vitvar, C. Bournez and J. Farrell, SAWSDL: Semantic annotations for WSDL and XML schema, IEEE Internet Computing, v11n6, pp. 60-67, 2007.

[11] D. McGuinness, F. Van Harmelen, et al., OWL web ontology language overview, W3C recommendation, World Wide Web Consortium, 2004.

[12] 'Whats in a Service Level Agreement?', SLA@SOI, http://sla-at-soi.eu/?p=356, retrieved on July 30, 2009

[13] O. Lassila, R. Swick and others, Resource Description Framework (RDF) Model and Syntax Specification, World Wide Web Consortium, 1999

[14] F. Baader, D. Calvanese, D. McGuinness, P. Patel-Schneider and D. Nardi, The description logic handbook: theory, implementation, and applications, Cambridge University Press, 2003

[15] J. Black et al, An integration model for organizing IT serviceManagement, IBM Systems Journal, VOL 46, NO 3, 2007

[16] K. Joshi, OWL Ontology for Lifecycle of IT Services on the Cloud, http://www.cs.umbc.edu/~kjoshi1/IT_ Service_Ontology.owl

[17] R. Smith, The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, IEEE Transactions on computers, Volume C-29, Issue 12, 1980, pp 1104-1113