Cloud Based Next Generation Service and Key Challenges

Santonu Sarkar, Rajeshwari Ganesan, Manish Srivastava
Infosys Labs. Bangalore, India
Email: {santonu_sarkar01,rajeshwari_ganesan,manishsv}@infosys.com

Sharada Dharmasankar
University of Chicago, USA
Email: sharada@uchicago.edu

***Abstract*—The National Institute of Standards and Technology (NIST) defined cloud computing as a model that enables on-demand access of computing resources. These resources are provisioned rapidly, without unnecessary client-provider interaction or management. Consumers of cloud get everything, such as software, platform and even the computing infrastructure as a service, rather than an on-premise deployed product. The cloud is perceived to be the game-changer of this decade as it offers a flexible, on-demand and elastic delivery method for the business especially services.**

To realize the potential and promise of the cloud, we need to step back and revisit the entire lifecycle of a service from six dimensions, namely the way a service is i) designed, ii) engineered, iii) deployed, iv) measured, v) managed and vi) experienced. This will help us identify the key concerns and barriers to its wide scale adoption. In this paper, we propose the notion of "next generation service" that embraces cloud computing as its backbone. Then we describe each of these six aspects and identify a set of important technical challenges that hinders the complete realization of the next generation service. These challenges are still unresolved to a large extent. We believe that a satisfactory solution to these challenge will largely help in making the benefits of the next generation service available for a wider consumption.

***Keywords*-Cloud Computing, Software as a Service, Service Definition, Engineering Challenges**

I. Introduction

The National Institute of Standards and Technology (NIST) [1] defined cloud computing as a mechanism that enables on-demand access of computing resources without any service provider management. NIST further specifies five critical characteristics of cloud computing:

1) The consumer can provision the computing resource without provider's intervention whenever required, i.e. it is *on-demand* and *self-serviced*.
2) The resource should have a broad network access– i.e. it is highly *available* through any client platform
3) Internally, hardware resources are pooled and shared among the consumers– giving rise to the multi-tenant model for *better resource consumption*.
4) It is possible to rapidly scale up or down the resource *elastically*, to meet the just in-time computing demand.
5) The service usage can be measured based on monitoring– resulting in an extremely *affordable* mode of payment of the service.

What consumers get from cloud is a service. NIST describes three types of service, namely i) Software as a Service (SaaS), ii) Platform as a Service (PaaS) allowing a consumer to deploy and configure his/her own middleware, applications on a cloud, and iii) Computing Infrastructure as a Service (IaaS).

What is it about cloud that makes it a game-changer? It is reported that the business at large find cloud's affordable, flexible, on-demand, elastic delivery method, to be extremely beneficial[2], [3]. These reports highlight significant reduction of IT budget (infrastructure, software, maintenance cost) due to cloud computing. Due to the on-demand nature of the service without any management and operational overheads, the subscriber consumes the service as and when required, and spend their time more usefully on the subject of their core interest.

There are several concerns about using cloud computing, despite the advantages. A survey of over 500 IT executives worldwide showed that they were all concerned with data security, and would rather keep data in their own infrastructure rather than resorting to public clouds. Studies and experiments have shown that it is possible to breach secure cloud environments, like EC2 [4] despite traditional security measures like strong encryptions.

In spite of these concerns, many top IT corporations such as Microsoft, Google, IBM, SalesForce.com, and Amazon have started offering their products as service in the cloud due to its strong value proposition. For example, Microsoft Azure[1] is not just a cloud based operating system, but a platform to offer Microsoft's products as SaaS applications. We further believe that for a global IT services company, cloud computing is going to be a game changing phenomenon. With the advent of cloud, the notion of a service offered by an IT service company will be transformed to what we propose as the "next generation service" in this paper. This next generation of services, offered on a cloud platform, will have characteristics such as self-service, personalized, and ubiquitously available. Furthermore, these services will be measurable, and highly affordable by offering extremely granular usage. The service is not going to be merely transactional like a web-service or SOA. There will a dynamic, social ecosystem around the service which will

[1] http://www.windowsazure.com

978-0-7695-4937-8/12 $26.00 © 2012 IEEE
DOI 10.1109/ICSEM.2012.11

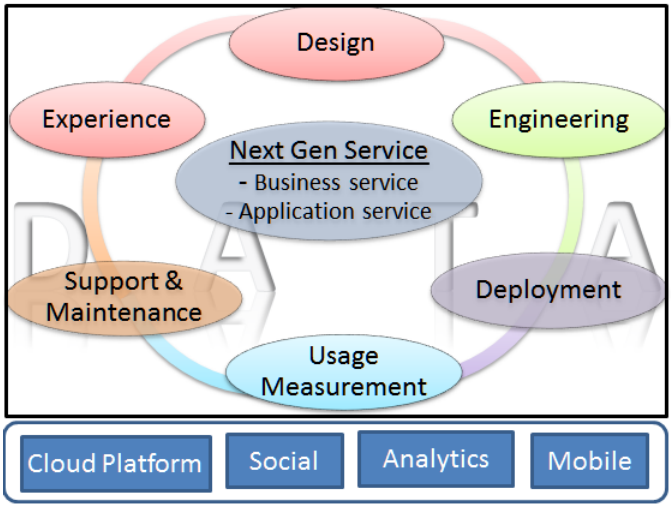

Figure 1. Differnt Aspects of Next Generation Service

foster an environment of co-creation of service capabilities. Such a service will be extremely agile and adaptable to change. Lastly, the service will learn, adapt, and make just-in-time decision by analyzing data (possibly massive) from its ecosystem. In this paper we describe the characteristics of such a service and then elaborate the set of challenges that need to solved in order to achieve such a service.

The paper is organized as follows. Section II introduces the notion of next generation services and its key characteristics from the service provider's perspective. Next from Section III till Section VIII we describe important challenges that the next generation service provider needs to address. Finally we conclude in Section X.

II. Cloud Based Service and Its Lifecycle

The infrastructure behind cloud computing is based on four main components namely i) virtualization, which allows sharing of a physical resource, ii) grid computing architecture, where large number of computers working in parallel are pooled, iii) elastic computing model that allows provisioning of computing resource on-demand, and iv) utility computing- a business pricing model in which users pay only the resources they use. This backbone in turn helped a software to evolve from a shrink-wrapped product form to a service (SaaS). Historically, the first successful SaaS was the early version of Salesforce.com where a CRM application with a set of functionalities was exposed as a paid service. Eventually, the notion of service has expanded from the concept of an application with a set of functionalities being exposed as a service. As a case in point, consider services such as iTunes[2], Coursera[3], healthdata.gov, UIDAI[4] and so on. As depicted in Figure 1, we refer to such a service as the "Next Generation Service". Such a service not only exploits the cloud platform, it also utilizes three other infrastructural elements i) social ii) analytics and iii) mobile as shown in Figure 1.

By social, we mean that a next generation service is becoming more "inclusive", where customers of the service not only use the service but also take part in evolving the service capabilities. Next, by "analytics", what we emphasize is that a next generation service is going to use data analytics as its intrinsic component as it learns from the massive amount of information available on a continuous basis, takes informed decision, and be more adaptive. Let us explain this with several data points.

In the last few years we have seen a phenomenal growth in generation, and assimilation of data in every sphere of computing. Business process, decision making, scientific experiments, process of buying and selling are increasingly being driven by data, and it is being perceived as the competitive advantage across industries. As described in a popular open-source information sharing sites [5] [5] it is reported that Facebook processes more than 30 petabytes of user content [6], amazon uses user-recommendations and browsing patterns to recommend a product, and Akamai analyzes nearly 75 million events per day for better target advertisement. There are growing evidences of processing massive data in scientific [6], and business [7] to derive the cruicial insight that was never possible before. In view of such startling information, we believe that massive data generated from the social ecosystem around the service and insights derived from this data through statistical modeling and machine learning techniques will be a core component of the next generation service.

Finally, we predict that the service is going to be more and more pervasive due to the inroad of mobile devices into the mainstream[7], [8], [3]. These three infrastructural components alongwith the cloud platform are going to have a profound impact on the next generation service. In this paper we delve deep into this "next generation service", which we refer to as "service" in short, from now on. In the next section we identify different aspects of such a service, from the service provider's perspective.

A. *Different Aspects of "Next Generation Service"*

As cloud computing becomes more prevalent, the next generation services encompass capabilities any user would want to see in an IT solution. We have identified six important

[2]http://www.apple.com/itunes
[3]https://www.coursera.org/
[4]http://uidai.gov.in

[5]http://wikibon.org/wiki/v/Big_Data, http://wikibon.org/wiki/v/Big_Data_Market_Size_and_Vendor_Revenues, http://wikibon.org/blog/big-data-statistics/, http://www.forbes.com/sites/louiscolumbus/2012/08/16/roundup-of-big-data-forecasts-and-market-estimates-2012/
[6]http://blog.cloudera.com/blog/2012/05/treato-analyzes-health-related-big-data-with-hadoop/
[7]http://www.lexisnexis.com/risk/downloads/casestudy/HPCCSystems-HealthCareCaseStudy.pdf, http://www.oracle.com/technetwork/topics/entarch/articles/oea-big-data-guide-1522052.pdf

aspects of the lifecycle of the next generation service, as shown in Figure 1 from the service provider's perspective.

- *Design:* The service provider conceptualizes the requirement, and designs the service. As discussed in the previous section, a next generation service will have to embrace massive amount of data relevant to the service as its integral part. This demands a new service design paradigm where the service functionality is aware and make use of the data that's going to be available around the service once it is live.
- *Engineering:* During the engineering phase, the service providers develops, tests and operationalizes the service. Once again, data plays a vital role during the engineering phase. First of all, service engineering must take care of creating the ecosystem around the service that can generate meaningful data related to the service once it is commissioned. Furthermore, it is imperative that the engineering phase needs to take care of information acquisition, processing, and communication.
- *Deployment:* The service deployment happens over a cloud infrastructure on a virtualized platform where the data center may or may not be owned by the service provider.
- *Usage Measurement:* The service needs to be designed and engineered keeping the monitoring and metering the usage in mind. After the service roll out, the service provider needs to manage the subsequent metering and billing for different customers.
- *Support and Maintenance:* The service provider provides support to the customer from training to issue resolution. The service provider also needs to constantly monitor various service quality parameters for better assurance. Proactive maintenance is an important activity for the service provider to assure near 100% availability and reliability. This information is also used to evolve the service.
- *Experience:* One of the important factors to increase the consumer stickiness to a particular service is to improve the experience of the service usage. The service provider needs to plan for the best-in-class service experience during the service design and implementation.

B. *Technical Challenges*

How easy is it to realize a next generation service? In order to build such a service, it is absolutely essential that the service provider understands key challenges along the above mentioned aspects of a service. Figure 2 shows a set of challenges that we believe to be critical to realize a service.

In the following sections we will take each aspect of the next generation service at a time and elaborate key challenges associated with this aspect that limit the realization of the service.

III. SERVICE DESIGN

When in application moves to the cloud, what are some key design considerations that should be kept in mind, in order to allow the elasticity and dynamic usage that cloud is known for?

A. *Service Composition*

Next generation services will not be designed in a traditional bespoke fashion; rather it will be built by composing other services; possibly in a dynamic manner. Evidences of such extreme form of system integration can be seen in Facebook[9], Amazon Simple Queue Service[10]. What are the challenges associated with system integration?

1) Sometimes, services from multiple clouds need to be composed, which results in resources being redistributed. Such a composition will be enormously difficult when underlying components of a service are tightly coupled [11]. A composition, when expected to be carried out dynamically, will be even more challenging.
2) How can one ensure that there is fault tolerance when there are multiple modules or components to a process?

B. *Interoperability*

It has been a pressing concern to make a software interoperable since distributed processing came to the mainstream [12]. Since 1990, middleware platforms like CORBA, DCOM, .NET and J2EE have been built to make a software interoperable. With the advent of cloud computing, major players like Amazon, Google, Microsoft, VMWare[8], Salesforce.com[9] etc. have started offering an integrated environment for developing, testing, hosting and monitoring an application on the cloud. A service developer can use such a platform to design a service. Furthermore, composition of services will be the preferred way of service design. Interoperability is especially important in the deployment and maintenance stages. This is because a software's interoperability is critical after it has been put on the cloud, and this interoperability needs to be retained no matter what happens to the cloud environment.

Given this context, what are the new challenges?

1) Interoperability becomes a major issue if a service has to run on a private, or a public cloud, without compromising security[13].
2) We need to ensure that when a service undergoes upgradation, or the cloud configuration changes, or the network availability is changed, the service must remain interoperable[14]. Likewise, the storage configuration, or its security settings can change. What are the design considerations that one need to keep in mind so that the service remains interoperable under all these circumstances?

[8]http://www.cloudfoundry.com/
[9]http://www.heroku.com/

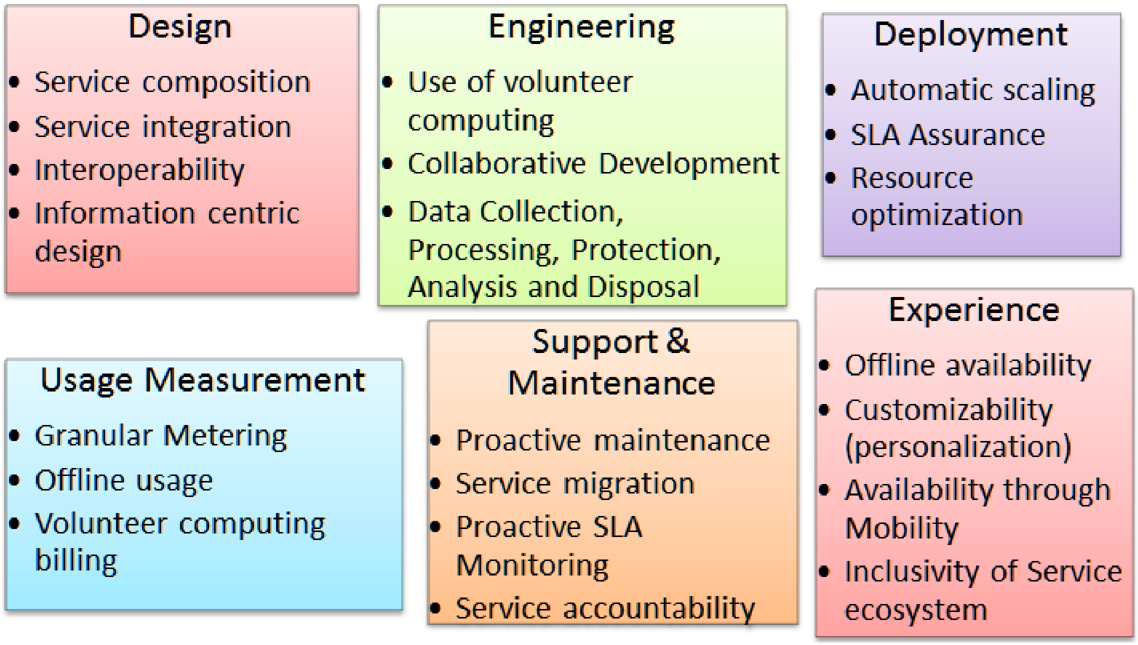

Figure 2. Challenges

C. Information Centric Design

The next generation digital consumer will significantly influence the capabilities that service ought to have, as they interact with the service and provide feedback[15]. The service in turn is expected to analyze the information generated within and available outside (which will be massive, as Section IV-B evidences) and continuously learn while it is used. Such an use of information is drastically different from a conventional business application design where the data is typically used to generate various reports, rather than influencing its decision making capability.

1) What should be the design principle in creating a such a service so that it can analyze the data, and use the outcome of the analyze to take the decision on-time?
2) As the service learns, it should be able to adapt and evolve itself over time. The learning happens as service consumers use the service, also provide feedback to the system. This implies that the service designer needs to incorporate a service feedback collection and analysis mechanism while designing the service. Furthermore, the design of the service should be adaptive, such that the service can adapt itself based on the feedback analysis after the service is deployed. What should be design strategy in this case?
3) Sometimes, a consumer may not be very serious with the feedback s/he gives. If that is the case, is there a way to automatically ensure that low-quality data is removed, and not incorporated into design modifications?

IV. Service Engineering

As discussed earlier, by service engineering we mean the construction and maintenance of a service. In this section discuss a few critical challenges that the provider needs to address to engineer the service.

A. Use of Volunteer Computing

Volunteer computing such as Amazon's Mechanical Turk has become an attractive proposition where human volunteers can be employed to do small human intelligence tasks (HITS) [16], [17], [18] that are easy for a human to perform but extremely difficult for a computer. much more potential to perform computations than a computer does. What are difficulties in exploiting this in engineering a service?

1) The provider of a next generation service can harness the power of volunteer computing as a means of gathering data or testing the service. The question is how one can ensure the quality of the activities (that is, ensuring the quality of data gathered, or quality of tests performed) performed by volunteers?
2) While using a Mechanical Turk for testing, how do we ensure that the testing environment is secure?
3) How do you identify that a task is suitably a HIT, as it delves into the realm of subjectivity?
4) How do you exploit the parallelism optimally in executing a task by multiple "turks" since they won't respond to your instruction the way a machine would? How would you manage and schedule tasks, given that you are assigning them to human beings instead of machines?

B. Service Data Management

More and more data is being produced every moment, at exponential rates, as Gartner predicts that by 2016, 36% of consumer content will be stored on the cloud, and that cloud storage in general will grow at aggressive rates [19]. We have predicted in Section III that next generation services must embrace data as a part of the service design. The question is, are we equipped to handle the "Big Data" phenomenon?

1) Data Upload and Transfer: A viewpoint from Berkeley [20] describes the challenges in even transferring and uploading petabytes of data, where an upload of 10 TB of data at 20 Mbits/sec would take over 45 days! Specific questions are:

1) How can data be packaged so as to decrease the time it takes to upload or transfer large datasets?
2) What sort of security and trust mechanism we need to employ while uploading the data through a public network so that only the authorized people can see and work with it will be able to access it when it is in transit?
3) How can you help mobile clients upload large amounts of data?

2) Data Processing: Data is being produced more rapidly and in greater volumes than it has been prior, and processing large data sets is an increasingly common and immensely important task [21], [22], [23]. Specific issues are:

1) How do you develop the capabilities to quickly and securely capture, store, and analyze vast amounts of information?
2) The distributed environment common to cloud computing causes complications when doing big data analytics. If one part of the data is not analyzed due to a node failure, then the entire operation is a failure. How do you mitigate such a risk when large amount of potentially sensitive data is distributed among different servers?
3) How do you ensure that a data processor is capable of handling so much information, which are of different types, such as streamed, accumulated [23], and in many different formats? This is extended to data coming from mobile devices as well- where the data is a stream of sensed form [24].
4) Data processing also extends to mobile clients, and data uploaded from a mobile should be able to be processed. This data is usually a stream of sensed data, meaning that mobile devices produce data differently than computers do . How does the service developer process data from different clients, and in this case, what are the challenges of receiving sensed data?

3) Data Security: A survey of over 500 IT executives worldwide showed that most of them would rather keep data on their own servers, than a public cloud. Trials have shown that it is still possible to hack secure cloud environments [25], like EC2, and traditional security measures are no longer enough to withstand such attacks. Data security is, therefore, still of paramount concern [4].

1) What kind of new security measures need to be in place, so that the data in public cloud is as safe as a private setup and how do you instill the sense of assurance? What should be the proper security measures and authentication processes?
2) How do you ensure that data is protected from being wiped out on accident[26]? How can it be removed in a timely manner?

4) Data and Network Isolation: While there is a growing demand for cloud services is increasing, customers have concerns regarding sharing infrastructure.

1) Microsoft had an incident where an unauthorized user downloaded other tenant data because it's cloud-hosted Business Productivity Online Suite was misconfigured. How do you ensure that all applications are correctly configured in order to avoid internal or outside party attacks [27]?
2) Use of the network to access applications on the cloud by one client should not impede the other client's accessibility to cloud resources. What needs to be done in order to make sure that one client's use of network capabilities do not affect another's ability to run an application or use the same network?

5) Data Analysis: As data analysis is a vague term, it shall encompass the computations that are done with large datasets that are uploaded in the cloud, as well as the steps that need to be taken in order for it to happen, in addition to the complications and challenges that one encounters in doing this in short, the culmination of processing big data. As tera and petabytes of data are uploaded the cloud, it needs to be processed and analyzed, a task which involves compressing the data into small, easy-to-read packages and distributed across servers

1) One of the biggest challenges an IT services company faces in the context of "Big Data" is the ability to reduce the amount of time it takes to perform computations on and analyze data. How do you decrease the amount of time it takes to perform computation on large datasets?
2) If a dataset needs to be distributed, how do you find the optimal distribution among several servers? How can all of this data be properly consolidated after each packet is analyzed?

6) Data Disposal: There are evidences [26] where massive data meant for deletion, isn't done reliably as the infrastructure containing the data was lost. As such, there should, first of all, be another way to back up data, and secondly, an automatic way that this information is removed after the required amount of time.

1) What are other ways that data can be backed up, but still be disposed of, with 100% assurance?
2) The way data is disposed should be thorough and automatic, so that all of the information a client no longer wants on the cloud is hidden. This is especially critical with regard to sensitive data. How do you develop an automated mechanism by which data can be truly removed from all possible sources?

V. Service Deployment

When a service is deployment, it is obviously going to be executed on a virtualized platform. Gartner predicts that the IaaS market will grow to $10.5 billion by 2014 [28], as more and more traditional application development and maintenance (ADM) businesses will evolve towards the SaaS as the service delivery platform. With that, it is extremely critical the SaaS provider achieves maturity at the infrastructure level.

A. Automatic Scaling

One of the tenets of cloud computing is the on-demand provisioning of computing resources [20], or automatic scaling by provisioning more servers automatically. This is essential to meet the response time SLA [29] of the service as the number of requests grows dynamically. What are the impediments in achieving automatic scaling?

1) As the infrastructure scales while the application is running, there is risk of the application failing to execute. The challenge is what are the necessary steps and precautions that should be in place should such a failure occur?
2) Alongwith computing resource scaling, how do we ensure network resource scalability? This is an important open question today [30].
3) For upward and downward scaling, it is necessary to perform a lot of workload analysis in order to discern any patterns in it. How do you analyze a workload in light of the fact that the number of submitted loads are always changing, in addition to their size? Because the variability is so high, this makes it difficult to find stable workload distribution pattern.
4) How do you determine the optimal way to distribute a workload among different clouds, or different virtual servers located on the same cloud? There will always be risks of fault, security, and data corruption as a result of this, as well, which make the solving of this challenge difficult.

B. Resource optimization

There is a fine line between conserving cloud computing resources without compromising on the SLA. How does the service provider ensure that the underlying infrastructure is utilized most optimally without violating the SLA of the service, specifically when the service workload varies dynamically?

C. SLA Assurance

A problem related to dynamic scaling and resource optimization is the assurance of service SLA to the customer. In the virtualized environment, there may be many collocated client workloads on a given hardware. Performance, or security vulnerability of one client workload can impact the other collocated one. The service provider may not have control over the workload of other workloads that the provider does not manage. Under this circumstances, it becomes extremely challenging to assure the service SLA.

VI. Usage Measurement

An important characteristics of a cloud based service is the ability to measure the usage and charge the consumer only on the amount of usage. This feature leads to several important design challenges that the service provider needs to address.

A. Granular Metering

An important aspect of the next generation service is to make it extremely affordable (for example one can buy a song from the iTunes service in less than a dollar) by making the usage of service as granular as possible. An important question is how can a service be designed in a higher modular manner, and its usage can be monitored at an extremely granular level?

B. Offline Usage

A next generation service is going to be pervasive where the consumer can use it even when the consumer is not connected to the network. This has an impact to the metering and billing. How do you charge someone for use of a service if you have no evidence of the network being used?

C. Volunteer Computing

When the service provider uses volunteers to execute a part of the service, how can one account for the volunteer computing effort in the service metering model?

VII. Service Experience

In order to garner a large share of the SaaS market, one needs to make its services more universally accessible to clients, which means it will need to be more flexible with client connectivity, platform, and specifications (according to business type), personalization and self-service. This will dramatically improve the client's ability to work with the service despite any intrinsic circumstances, such as client/-company type, location, resources, etc.

A. Offline Availability

One facet of cloud computing [31], is the availability of a variety of files, applications, and computing resources as long as you are connected. But what is to occur if someone does not have this reliable internet connection as the person is in transit [32]?

1) In order to access and use a service, a fast, reliable internet connection may be necessary whereas the client has slow connectivity. How do you design the service experience so that the client can use a "light" versions of service when the connectivity is poor?
2) It may so happen that client can be offline at times. How do you synchronize work done offline and update it once the client has an internet connection once again?

B. Customization, Personalization and Self-Service

A SaaS application is meant to cater en masse, which means that there is little ability to customize applications for different clients, specifically when the application allows multi-tenancy [33]. As a result, for a customized requirement, one has to wait until the next release of the application[34]. Such a rigidity is going to be a deterrent for quick on-boarding of new clients.

1) How do we retain multi-tenancy, yet make the service flexible enough so that sufficient customizations can be done at run-time, without impacting the configuration of an existing client?
2) How do we design a service to be completely self-serviceable, so that a client can set up, personalize and use the service without any third party intervention?

C. Availability through Mobility

Gartner predicted that by 2013, mobile devices like phones would outnumber PCs as the prevalent web-access devices. Therefore, it is obvious that our proposed next generation service will be used more and more through mobile devices. In many cases the mobile devices will have a handheld version of the service counterpart, or will be accessed remotely through the device[7], [8], [3]. What should the strategy be in offering a rich user experience through a handheld device?

1) How do you ensure that a mobile application runs as quickly as a non-mobile counterpart, without compromising output quality?
2) As mobile devices increase in their computing capabilities (for example, having more memory or processing power), how will this affect application design?
3) Additionally, there are many different mobile platforms with varying capabilities; how can an application be customized so that it works on all of these devices?
4) To what extent will security be broken? What new security measures will need to be added to a mobile application?

VIII. Service Support

Gartner predicts that SaaS is predicted to grow at 15.8% through 2014, but this is contingent upon how the service is supported once it goes live. The service may go live by migrating an existing enterprise application. Once it is live, the SaaS provider will have to assure that the service will remain reliable, available and will meet SLAs all the time.

A. Proactive Maintenance

Proactive maintenance is about having mechanisms in place that prevents the service from failing in future. This in turn requires a proactive analysis of service behavior under various faulty and vulnerable scenarios, and making sure that the service is designed to handle such situations gracefully. This is often referred to as service dependability [35] that deals with improving reliability, protection of vulnerabilities, security, availability, and fault tolerance.

1) A next generation service, comprising of multiple components, is bound to be distributed over multiple cores and servers. How should the service respond if even one server fails? Furthermore, if there is high network latency, fault tolerance incurs a large cost. As such, how can one reduce network latency and optimally distribute processes over different networks?
2) How do you integrate a checkpoint mechanism with all components [36] without impeding the application's ability to run? An efficient checkpoint mechanism will have the ability to realize if there is a crash, and restore the system to a pre-crash state, rather than re-running the process from the beginning. This problem is complex as one needs to find out exactly what pre-crash state to restore the application to.
3) In general, multi-core VM environments are neglected with regard to fault tolerance [37]. How would you standardize incorporation of fault tolerance mechanisms in such environments?

B. Service Migration

With growing demand for a service over cloud, more and more existing enterprise applications will be migrated to a cloud [38], [39]. Here we illustrate some of the challenges related to service migration.

1) Seamlessly move legacy software applications from old physical computing environments to the virtualized environment is not easy, even when you keep the same OS infrastructure. Besides technical intricacies, what are the repercussions in terms of licensing and pay-per usage policy as the original software wasn't meant for pay-per-use?
2) A major concern is whether or not application isolation will still be maintained, even after software moves from legacy systems to the cloud [40].

3) How do you migrate applications to the cloud while ensuring that the service runs just as quickly, efficiently, and securely as it would on the old environment?

C. Proactive SLA Monitoring

Service Level Agreements (SLA), particularly those concerning performance variance of applications are like contracts that represent what guarantees are made between service providers and cloud consumers. As a part of continuous support, SLA parameters need to be continuously monitored by the SaaS provider in order to ensure that SLAs are met.

1) In order to meet SLAs, an organization would have to be aware of workload variability and fast resource scaling. How can we monitor and predict such variability and decide when more resources are needed?
2) Currently, the specifications of existing SLAs for cloud services are not designed to be flexible. What needs to be achieved in order to make SLA specifications more flexible to adapt to client needs [41]?
3) Many providers only guarantee the availability of services, rather than performance. This means that it becomes harder to provide performance guarantees to clients [41], which is a large concern for the clients. How can you automatically evaluate the performance of their services and be able to provide adequate SLAs as a result?

D. Service Accountability

Accountability raises the question of how one can decide, between the infrastructure provider and the application provider, who is to be held accountable for when an application demonstrates faulty behavior It also encompasses the idea of being able to detect and contain faults.

1) One challenge with accountability is gauging whether or not it is useful in distributed systems. As control is decentralized, how are you determine who to hold accountable for disparate components of a system?
2) With regard to accountability, how do you reliably detect faulty nodes or components in a distributed system, and record the fault? This will detect who or what is to be held accountable for any occurrence of a fault.

IX. Next Generation Service: Summary of Challenges

We summarize the set of challenges that a next generation service needs to address in Figure 3.

The challenges described in this paper and summarized in this figure, are being addressed by the academic and practitioner community with great interests. Needless to say, service providers will have to resolve these challenges at different levels of depth, depending on business needs.

X. Conclusion

In this paper we proposed the concept of the next generation service. By leveraging the cloud computing platform, this service is going to be pervasive in the near future. In this paper we characterized the desired capabilities and defined the life cycle of such a service. In order to achieve the capabilities of such a next generation service it is essential to resolve a set of technical challenges which are particularly prominent as the service is cloud based. In this paper we have identified and discussed these challenges that are present in various life cycle phases of a service. Each of these challenges are quite difficult in itself, though workable solutions exist in several cases. We haven't discussed any state of the art solutions for these challenges in this paper.

In this paper we have described the concept of the next generation service from the service provider's viewpoint. However, it is equally important to think about such a service from the consumer's viewpoint and link these two viewpoints. We believe that by linking these two viewpoints we can possibly categorize the needs and articulate the associated challenges in a holistic way.

References

[1] P. Mell and T. Grance, *The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology*, United States Department of Commerce Std. Special Publication 800-145, September 2011. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[2] *Gartner Says 90 Percent of E-Commerce Sites Will Use At Least One SaaS-Based Service by 2013*, Gartner, 2008, gartner Press Release. [Online]. Available: http://www.gartner.com/it/page.jsp?id=734612

[3] *Gartner Highlights Key Predictions for IT Organizations and Users in 2010 and Beyond*, ser. Gartner Press Release, no. G00173482, Gartner, January 2010. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1278413

[4] S. Ko, K. Jeon, and R. Morales, "The hybrex model for condentiality and privacy in cloud computing," in *HotCloud*. University of Buffalo, State University of New York, and Xerox Center Webster, 2011.

[5] *Hype Cycle for Big Data, 2012*, no. G00235042, July 2012. [Online]. Available: http://www.gartner.com/id=2100215

[6] J. Vijayan. (2011) Facebook moves 30-petabyte Hadoop cluster to new data center. ComputerWorld. [Online]. Available: http://www.computerworld.com/s/article/9218752/Facebook_moves_30_petabyte_Hadoop_cluster_to_new_data_center

[7] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso, "Calling the cloud: enabling mobile phones as interfaces to cloud applications," in *10th International Conference on Middleware*, 2009, pp. 83–102. [Online]. Available: http://dl.acm.org/citation.cfm?id=1813362

[8] D. Huang, "Mobile cloud computing," in *IEEE COMSCO MMTC E-Letter*, 2011. [Online]. Available: http://snac.eas.asu.edu/snac/document/E-Letter_PositionPaper_Mobicloud%20Dijiang%20Huang.pdf

[9] K. Haselden, *Microsoft SQL Server 2008 Integration Services Unleashed*. Pearson, 2009. [Online]. Available: http://www.informit.com/articles/article.aspx?p=1324340

Service Design	Service Composition	How efficiently one can compose services dynamically? How to ensure fault-tolerance after composition?
	Interoperability	How to ensure service interoperability in public, private, hybrid cloud, and ensure interoperability after service is modified?
	Information Centric Design	How to incorporate on-time decision based on data analysis? How to evolve based on learning from environmental feedback? How to ensure data cleanliness?
Service Engineering	Volunteer Computing	How to ensure quality of output, security of testing environment? How to identify a suitable task? How to exploit parallelism in computation?
	(Big) Data Handling	How to optimize massive data upload? How to efficiently perform massive data processing? How to ensure data security? How to efficiently perform (big) data analysis and data distribution? How to dispose off data after usage?
Service Deployment	Infrastructure Scaling	How to dynamically scale (infrastructure and network) without impacting the service? How to decide the scaling based on dynamic workload analysis? How to ensure security, fault-tolerance during scaling?
	Resource optimization	How to ensure service SLA and optimize the resource usage?
Usage Measurement	Granular metering	How to design a service in a modular way so that it's usage can be measured at an extremely granular level?
	Offline usage	How to monitor and measure the usage when the consumer is not always connected?
	Volunteer computing	How to account for volunteer computing effort while metering a service?
Service Experience	Offline availability	How to design a service that tunes itself when there is a poor connectivity? How to efficiently sync the data and computation state when the connection is established?
	Self-service	How to make a service flexible enough so that it can be customized at run time without impacting another customer? How to design a service totally self-serviceable ?
	Ubiquitous access	How to design services that can be accessed through diverse portable devices? How to harness the ever increasing computing power of these devices? How to handle security?
Service Support	Proactive maintenance	How to introduce fault-tolerance in the infrastructure? How to integrate efficient checkpoint mechanism with various components of a service?
	Service migration	How to efficiently migrate an on-premise service onto a virtualized environment in the public/hybrid cloud? How to efficiently make the software ready for pay-per-use?
	Proactive SLA monitoring	How to build a monitor that is lightweight and can monitor various SLA parameters of a service with sufficient granularity? How to make such a monitoring activity generic?
	Accountability	When a service relies on other services, and runs on a virtual hardware, how to accurately held a component, or a part of the infrastructure accountable in case of a failure?

Figure 3. Summary of Challenges from Service Provider's Perspective

[10] J. Murty, *Programming Amazon Web Services: S3, EC2, SQS, FPS, and SimpleDB*, M. Loukides, Ed. O'Reilly Media Inc., 2008.

[11] M. A. Vouk, "Cloud computing - issues, research and implementations," *Journal of Computing and Information Technology*, vol. 4, pp. 235–246, 2008.

[12] C. T. Howie, J. Kunz, and K. H. Law, "Software interoperability," Stanford University, Tech. Rep. 117, December 1997. [Online]. Available: http://www.stanford.edu/group/CIFE/online. publications/TR117.pdf

[13] M. Becker, "Interoperability case study: Cloud computing," *Berkman Center Research Publications*, vol. 2012-11, pp. 1–25, April 2012. [Online]. Available: http://cyber.law.harvard.edu/node/7887

[14] J. Urquhart. (2012, August) The great challenge - and opportunity - of cloud: Interoperability. GigaOM. [Online]. Available: http://gigaom.com/cloud/the-great-challenge-and-opportunity-of-cloud-interoperability/

[15] A. Mhaiskar and S. Kasthuri, "Converged digital devices and strategies for hi-tech companies," *SETLabs Briefings*, vol. 8, no. 5, pp. 49–60, 2010. [Online]. Available: http://www.infosys.com/infosys-labs/publications/Documents/digital-convergence.pdf#page=51

[16] G. Paolacci, J. Chandler, and P. Ipeirotis, "Running experiments on amazon mechanical turk," *Judgement and Decision Making*, vol. 5, no. 5, pp. 411–419, August 2010. [Online]. Available: http://repub.eur.nl/res/pub/31983/jdm10630a[1].pdf

[17] P. Heymann and H. Garcia-Molina, "Turkalytics: Analytics for Human Computation," in *WWW-2011*. ACM, 2011, pp. 477–486.

[18] P. Ipeirotis, "Crowdsourcing using Mechanical Turk: Quality Management and Scalability," in *Proceedings of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011*. ACM, 2011.

[19] *Gartner Says That Consumers Will Store More Than a Third of Their Digital Content in the Cloud by 2016*, ser. Gartner Newsroom, Gartner, June 2012. [Online]. Available: http://www.gartner.com/it/page.jsp?id=2060215

[20] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications

of the ACM, vol. 53, no. 4, pp. 50–58, April 2010.
[21] B. Rimal, "A taxonomy and survey of cloud computing systems," in *Fifth International Joint Conference on INC, IMS and IDC*, 2009.
[22] P. Bhatotia, A. Wieder, I. Ekin, Akkus, R. Rodrigues, and U. A. Acar, "Large-scale incremental data processing with change propagation," in *HotCloud*. Max Planck Institute for Software Systems, 2011. [Online]. Available: http://static.usenix.org/events/hotcloud11/tech/final_files/Bhatotia.pdf
[23] S. Tsuchiya, Y. Sakamoto, Y. Tsuchimoto, and V. Lee, "Big data processing in cloud environments," *FUJITSU Sci Tech*, vol. 48, no. 4, pp. 159–168, 2012. [Online]. Available: http://www.fujitsu.com/downloads/MAG/vol48-2/paper09.pdf
[24] D. Huang, Z. Zhou, L. Xu, T. Xing, and Y. Zhong, "Secure data processing framework for mobile cloud computing," in *IEEE INFOCOM 2011 Workshop on Cloud Computing*. Arizona State University, 2011. [Online]. Available: http://cse.unl.edu/ byrav/INFOCOM2011/workshops/papers/p620-huang.pdf
[25] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds," in *ACM CCS*, 2009, pp. 199–212.
[26] J. Mangalindan, "Is user data safe in the cloud?" CNN Money, September 2012. [Online]. Available: http://tech.fortune.cnn.com/2010/09/24/is-user-data-safe-in-the-cloud/
[27] Y. Mundada, A. Ramachandran, and N. Feamster, "Silverline: Data and network isolation for cloud services," in *HotCloud*. School of Computer Science, Georgia Tech, 2011. [Online]. Available: http://static.usenix.org/events/hotcloud11/tech/final_files/Mundada6-1-11.pdf
[28] *Gartner Maps Out the Rapidly Evolving Market for Cloud Infrastructure as a Service*, Gartner Newsroom, April 2011. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1622514
[29] S. Datta, S. Geera, A. Verma, and B. Viswanathan, "Smartscale: Automatic application scaling in enterprise clouds," in *IEEE Fifth International Conference on Cloud Computing*, 2012, pp. 221–228. [Online]. Available: http://www.computer.org/portal/web/csdl/doi/10.1109/CLOUD.2012.12
[30] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, "Dynamically scaling applications in the cloud," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 45–52, January 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1925869&bnc=1
[31] M. Miller, *Cloud Computing: Web-based Applications that Change the Way You work and Collaborate Online*. Que Publishing, 2009.
[32] T. Peck, "Going offline with gwt an gears," 2008, appEngines, LLC. [Online]. Available: http://www.slideshare.net/tom.peck/going-offline-with-gears-and-gwt
[33] M. Kapuruge, "Defining customizable business processes without compromising the maintainability in multi-tenant saas applications," in *IEEE International Conference on Cloud Computing (CLOUD)*, September 2011, pp. 748–749.
[34] R. Mietzner and F. Leymann, "Generation of bpel customization processes for saas applications from variability descriptors," *SCC '08 Proceedings of the 2008 IEEE International Conference on Services Computing*, vol. 2, pp. 359–366, 2008, this is reference 43 in the original document.
[35] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, January 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1026488.1026492
[36] Y. Teo, B. Luong, Y. Song, and T. Nam, "Cost-performance of fault tolerance in distributed systems," *Journal of Science and Technology*, vol. 49, no. 4A, pp. 61–73, 2011. [Online]. Available: http://www.comp.nus.edu.sg/ teoym/pub/11/ACOMP-Oct-2011.pdf
[37] W. Huang, M. J. Koop, Q. Gao, and D. K. Panda, "Virtual machine aware communication libraries for high performance computing," in *ACM/IEEE Conference on Supercomputing*, no. 9. The Ohio State University, Columbus, 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1362635
[38] G. J. Weiss and M. Chuba, "Key issues for data center, server and software/operating system infrastructure, 2010, part 2," Gartner, Inc., Tech. Rep., 2010. [Online]. Available: http://www.gartner.com/id=1324121
[39] M. Hajjat, X. Sun, Y.-W. E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, and M. Tawarmalani, "Cloudward bound: Planning for benecial migration of enterprise applications to the cloud," in *SIGCOMM*, Purdue University and Microsoft Research and IBM TJ Watson Research Center. Association for Computing Machinery, Inc., 2010. [Online]. Available: http://research.microsoft.com/pubs/132805/CloudwardBound-sigcomm2010.pdf
[40] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousi, "Black-box and gray-box strategies for virtual machine migration," in *4th USENIX Symposium on Networked Systems Design & Implementation*, 2007, pp. 229–242.
[41] S. Sakr and A. Liu, "Sla-based and consumer-centric dynamic provisioning for cloud databases," in *IEEE Fifth International Conference on Cloud Computing*, 2012. [Online]. Available: http://www.nicta.com.au/pub?id=5899