ISSN No. 0976-5697

Volume 5, No. 8, Nov-Dec 2014

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Survey on an Approach to Enhance QoS and QoE by Migrating Services in Cloud Based Mobile Environments

Ms. Mayuri R. Gawande
M.E Second Year CSE, P.R.Patil COET
Amravati, Maharashtra, INDIA

Ms. Parnal P. Pawade
Asst. Prof. IT Dept. P.R.Patil COET
Amravati, Maharashtra, INDIA

Abstract: Mobile networks currently play a key role in the evolution of the Internet due to exponential increase in demand for Internet-enabled mobile devices and applications. As intelligent mobile phones and wireless networks become more and more popular, network services for users are no longer limited to the home. Multimedia information can be obtained easily using mobile devices; allowing users to enjoy everywhere network services. So if user moves around then they can access cloud services without any disadvantages. When user is moving and accessing services then the service from private cloud get migrate to the near geographical area public cloud and user get service more fastly. But now days, there is a major problem of hacking of cloud data. In this paper we provide more security for cloud data using OAuth protocol.

Keywords: cloud computing, QoS, QoE, service population, OAuth protocol

I. INTRODUCTION

Cloud computing is a relatively new trend in Information Technology that involves the provision of services over a network such as the Internet. The cloud services offered are divided in three categories: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) as illustrated in Fig.1.SaaS delivers software applications such as word processing over the network. PaaS delivers a host operating system and development tools that come installed on virtualized resources. Such Cloud services are now being used to support Video-on-Demand (VoD) services which have much more demanding Quality of Service (QoS) constraints. Finally, IaaS offers raw resources such as a number of virtual machines or processors and storage space and leaves it up to the user to select how these resources are used [1]. Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications [2]. Mobile computing also becomes more popular due to smartphones and tablet pcs.

It provides center based resources and those devices require decenter based pool of resources. It creates traffic congestion problem on internet due to user mobility and high bandwidth services. It affects QoS and QoE factors in mobile services. This paper consists of framework which overcomes the problem by service populating technique and also provides security to cloud data.

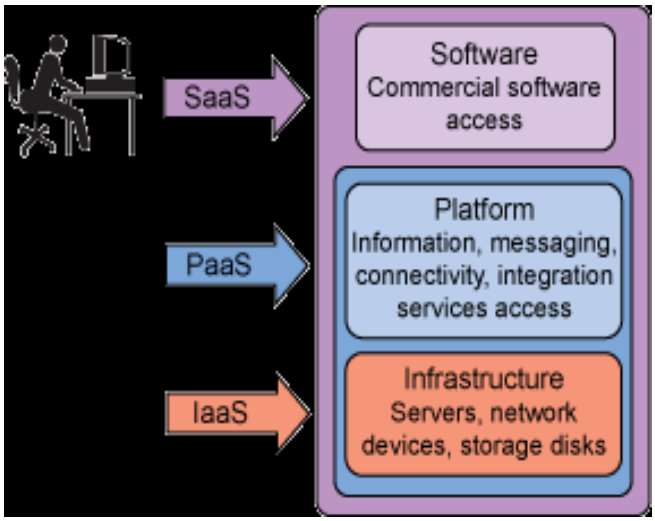

Figure 1. cloud service layers

II. LITERATURE SURVEY

A. Mobile Multimedia QoS Provision Architecture:

The provision of QoS for mobile multimedia applications requires the support of the architectures, protocols, and applications so that the mobile devices can access the multimedia data ubiquitously: anytime and anywhere. Multimedia transmission needs to meet the following requirements, namely high bandwidth, low error rate, low delay, and very small delay variance. As mentioned in the current research effort cannot provide solutions to fulfill all of these requirements for even the wired media. It is thus well-acknowledged that it is even more challenging to meet these requirements for high-quality multimedia transmission over wireless connections. The QoS of multimedia applications are not limited to bandwidth, delay, and jitter. Furthermore, the services provided to the mobile devices should be personalized. There are two ways of emphasizing Region of Interest (ROI), zooming in and enhancing the quality to optimize the overall user experience of viewing sports videos on mobile phones [3].It found out the overall user experience is closely

related to the acceptance of video quality and the interest in video content.

The current state of QoS provision in architectural frameworks can be summarized as follows [4]:

a. ***Incompleteness*****:** current interfaces (e.g., application programming interfaces such as Berkeley Sockets) are generally not QoS configurable and provide only a small subset of the facilities needed for control and management of multimedia flows;

b. ***Lack of mechanisms to support QoS guarantees*****:** research is needed in distributed control, monitoring and maintenance QoS mechanisms so that contracted levels of service can be predictable and assured; and

c. ***Lack of an overall framework*****:** it is necessary to develop an overall architectural framework to build upon and reconcile the existing notion of quality of service at different system levels and among different network architectures.

End-to-End QoS Scenario In recognition of the above limitations, a number of research teams have proposed a systems architectural approach to QoS provision. In this paper these are referred to as QoS architectures. The intention of QoS architecture research is to define a set of quality of service configurable interfaces that formalize quality of service in the end-system and network, providing a framework for the integration of quality of service control and management mechanisms.

Another recently proposed architecture aimed at improving the performance of Cloud technologies is called Media-Edge Cloud (MEC). It is an architecture that aims to improve the QoS and Quality of Experience (QoE) for multimedia applications [5]. This is achieved by a "Cloudlet" of servers running at the edge of a bigger Cloud. The aim of this is to handle requests closer to the edge of the Cloud and thus reduce latency. If further processing is needed, then requests are sent to the inner Cloud, so the "Cloudlets" are reserved for QoS sensitive multimedia applications. In essence, the aim is to divide the network hierarchy within the Cloud, in such a way that physical machines that are closer to the Cloud's outer boundaries will handle QoS sensitive services. Since these machines reside on the border of the Cloud, the data has to travel less distance within the Cloud before it is sent out to the clients. This not only improves QoE for clients but it also reduces network congestion within the Cloud.

However, these new concepts and research into improving Cloud performance, do not take into account user mobility. Media delivery on mobile clients is the new trend in computing and mobile devices are the most likely to make use of Cloud resources in the future. Furthermore, all the research at present assumes that only one entity (the provider) is in control of a Cloud and as a result different providers cannot "share" resources in a manner that can improve the utilization efficiency of their hardware. This can potentially lead to problems in the future as mobility and multimedia-rich content

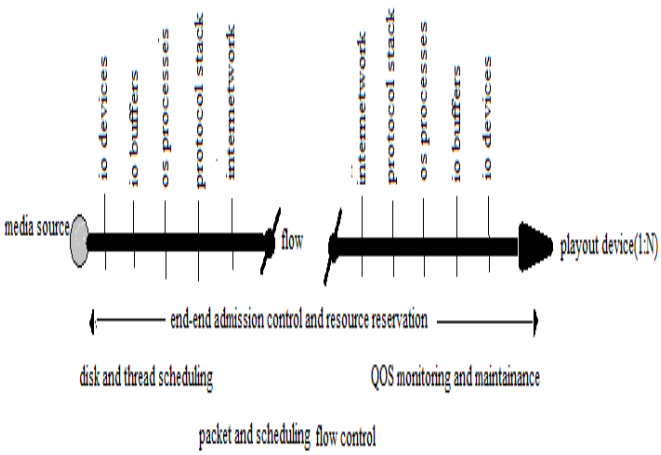

Figure 2. End-to-End QoS Scenario

Becomes more popular and high bandwidth data streams will have to travel great distances and reach moving targets. Cloud providers may find themselves in situations where their hardware resources are not adequate and they may have to create more Clouds to handle the load and relieve network congestion.

B. Experiments and Results (Comparison and Testing):

For the comparison and testing purposes in which is shown in Fig. 2, is implemented in which the same input is given to both the traditional and Usage Pattern selection mechanisms. Both the mechanisms also share the same test data so that there are no unfair circumstances. A knowledge base is created which is a set of 600 test data of cloud services divided in 4 groups. Each group is having 150 cloud services, of 2, 3, 4 and 5 numbers of Limitations. Each test data is tested with 200 randomly and automatically generated input values, which are supposedly provided by the consumer. Finally, the results are generated by averaging 100 such iterations.

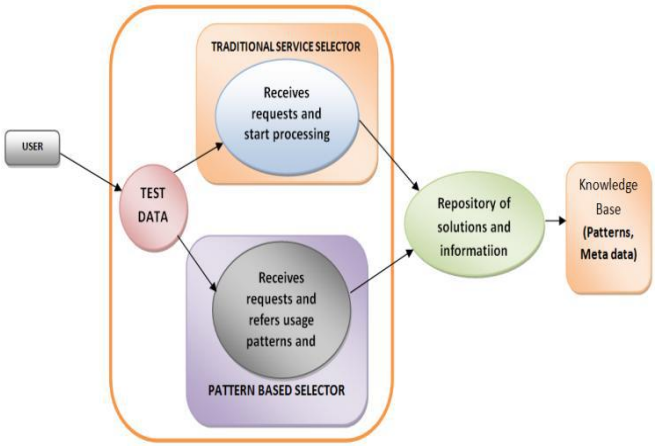

Figure 3. Comparisons of Service Selection Mechanisms

The results of the experiment show a significant improvement in the consumer satisfaction as we move from hard to soft limitation. The results are divided into Response time, Query Processing time and Cost. The support perspective is based on average number of services selected by each mechanism. The Usage Pattern mechanisms performance is significantly better than the traditional mechanism and can be easily noticed in the graph. However, it is observed that as we increase the number of the number

of requests, the Usage Pattern mechanism performs even better than the traditional one. We call „increasing the number of requests" as „moving towards the real world" , as in real web services the number of non-functional and QoS limitations can be much more than ones in this paper [6].Following comparisons clearly reveal than Traditional Service Selector is no longer beneficial in cloud environment. When number of requests increases the Response Time, Query Processing Time and Cost increase in case of Traditional Service Selector.

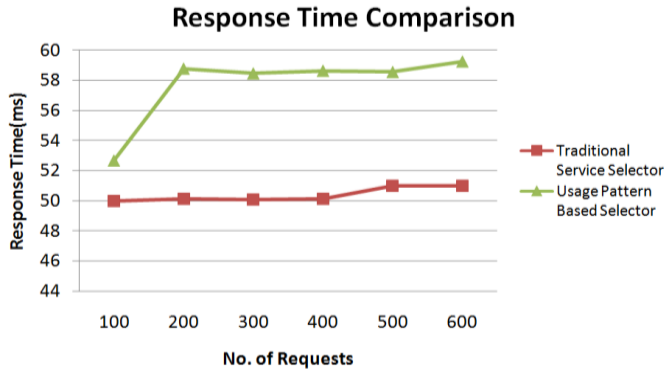

Figure 4. Comparison of User Request Response Time

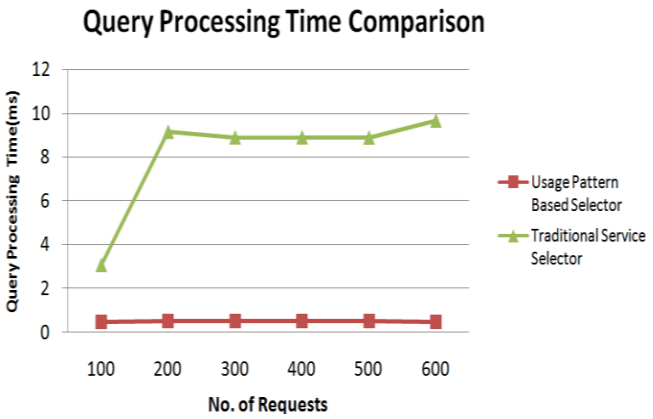

Figure 5. Comparison of Query Processing Time

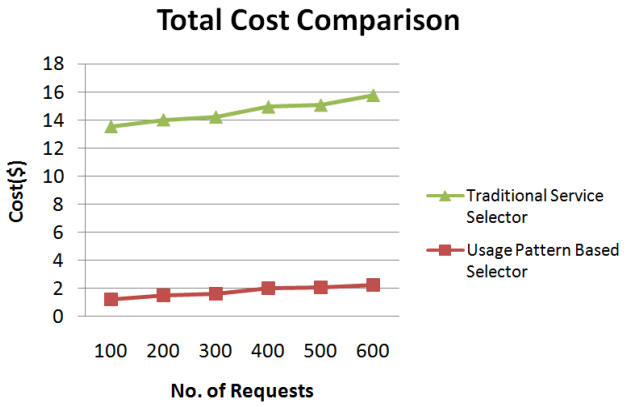

Figure 6. Comparison of Total Cost

III. PROPOSED SYSTEM

A. *Overview:*

As we know now a day's data hacking is increasing rapidly specially clod data. For this in the proposed system, we are providing more security to the cloud data using OAuth protocol. OAuth is an open standard to authorization. OAuth provides client applications a 'secure delegated access' to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner, or end-user. The client then uses the access token to access the protected resources hosted by the resource server. OAuth is commonly used as a way for web surfers to log into third party web sites using their Google, Facebook or Twitter accounts, without worrying about their access credentials being compromised. Here we are using OAuth 3.0 protocol. The main goals are openness and simplicity.

a. ***OAuth 3.0: The New Open Protocol Paradigm*:** OAuth 1.0 is a protocol but OAuth 2.0 was considered to be too loose and has been reclassified as a simple framework. A lot has been said about this issue: OAuth 2.0 has been criticized to its compromises. However, the real problem is that protocols are too strict for today's internet. It was possible to agree on very strong rules when the discussion was limited to a small circle of scholars. Nowadays, Stand ford, the MIT and the CERN no longer rule the internet. Hence, we need to open protocols and make their rules fork able by anyone.One could argue that protocols are meant to provide fixed rules, but we disagree. In fact, choosing your own protocol rules is de facto the new standard. For example, the emergence of multiple crypto-currencies based on the Bitcoin protocol proves that openness is the new standard. This is the new paradigm underlying the OAuth 3.0 protocol. Our draft will be studied by the Internet Task Force in the coming weeks. We hope other standards will also tolerate flexible rules in the future. We believe HTTP and SMTP are next.

b. ***OAuth 3.0 Overview:*** *3 Key Features:* To understand the OAuth 3.0 revolution, you need to be familiar with the core architectural evolutions.0Token is a logical evolution. OAuth 1.0 started with 2 tokens, which was a heavy process. OAuth 2.0 brought this number down to 1, but we are now taking it to 0.This 0 token architecture allows a very efficient authorization flow much less verbose. Plain-Text Encryption OAuth 1.0 imposed a complicated "dance" with digital signatures. OAuth 2.0 removed this need by relying on HTTPS rather than digital signatures, but again, we are taking it one step further. In a groundbreaking paper we are about to publish, we will provide all the details about our new security revolution. Cryptography didn't change much since 1976, when the idea of public-key cryptosystems was first introduced. Whitfield Diffie and Martin Hellman did only half of the work: we have discovered that by making public both the private and the public keys, we would remove the need for digital signatures or TLS. Several government agencies support our work in the domain. Any Grant You Like OAuth 2.0 went from proposing 1 type of grant to 4. Again, we improve the existing by providing unlimited flexibility [7].

B. *Cloud Based Service Framework:*

Cloud based service Layered Framework: We relate the layers of the architecture with the OSI model. The proposed

framework and the OSI model share the same level of abstraction in terms of network technologies and protocols and this makes it easy to use the OSI as a reference to our model as opposed to using the TCP/IP model. The service architecture is not meant to map directly to some of the OSI layers. Some of the functions performed in the proposed layers can interact with OSI layers to perform network-level operations while other layers do not present any functions that directly interface with the OSI and are therefore considered extra layers.

a. The Service Management Layer (SML): Deals with how services are registered in a Cloud. This also includes the overall Service and Security Level Agreement (SSLA) between the Cloud providers and the service providers and the unique Service ID. The SML can be considered as part of the Application Layer in the OSI since it defines the applications themselves and how they use resources.

a) The Service Subscription Layer (SSL): Deals with the subscription of clients to the service and holds information that handles the subscriptions such as User IDs, the list of services subscribed to by individual client and the associated client SLAs between clients and services. This layer can give instructions to the Presentation Layer in the OSI in order to handle user specific service parameters such as encryption or CODECs in video streams. The SSL can be considered as part of the Application Layer in the OSI.

b. The Service Delivery Layer (SDL): Is responsible for the delivery of services to individual clients. The layers below receive instructions from this layer with regard to connecting to individual clients as well as populating Clouds.

a) The Service Migration Layer (SMiL): Is responsible for the Migration of services between Clouds. It deals with resource allocation across Clouds to facilitate service population. It also holds the mechanism that performs the handover of client connections between services. The SSL can be considered as part of the Application Layer in the OSI.

c. The Service Connection Layer (SCL): Monitors connections between clients and services. Some of this layer's functions map directly to the Session Layer in the OSI model.

d. Service Network Abstraction Layer (SNAL): Makes the network technology transparent to the upper layers in order to simplify and unify the process of migration. The function of this layer is to act as a common interface between the service delivery framework and the underlying network architecture such as IP overlay network or new technologies which divide the Internet into a Core network surrounded by Peripheral wireless networks.

e. Abstraction of service layer: In SML when a service provider wishes to publish a service, they have to define security and QoS parameters [5]. In SDL, the logic that processes all the data regarding QoS characteristics and user mobility resides in this layer. It uses data from the overall SSLA and the client SLA and checks if the requirements are met by using network QoS data given by the layer below. Such data can be fed to this layer by the mobile devices themselves either in the form of a process running separately or through a QoS-aware protocol that can report latency and bandwidth between two end points. The Cloud that fulfills all the parameters in the SSLA list and can provide better QoS than the others can then proceed to the Migration process in the layer below. In SCL the SCL is also responsible for the network handover between clients and services after a service moves. This is done by gathering QoS data from the network and from client devices.

f. Implementation mechanism: In order to gather QoS data and know the network conditions in a specific area, we are using another mechanism that we call the QoS Monitor. It is considered to be part of the SCL and acquires such data by querying the clients for network conditions. The mechanism that we are assuming here that can resolve human-friendly service names to unique Service IDs. In the SDL we need mechanisms that will connect service subscribers to the correct instance of a service for service delivery purposes. A record of Service IDs and in which Clouds their instances are running and also uses input by the QoS Tracking are maintained by the Service Tracking and Resolution or STAR. STAR will make a decision on which Cloud is better suited to service a client request based on the location of the client, using this information.

STAR achieve this functionality is by look up routing tables in order to identify which Cloud is closer to a user. Service delivery mechanism using STAR is shown in Fig. 3 Service to reject the new client and forward them to another Cloud if possible. This gives control to service providers and also becomes a contingency mechanism in case STAR makes a wrong decision. The STAR server can be scaled similarly to the DNS [8] system since it is essentially the same type of service albeit with some extra parameters. Once a Cloud ID is found, then the ID is resolved into the IP addresses of the Cloud controllers that the client can contact to access the service. The process is shown in the Fig.7. It should be noted that alternatively the Cloud ID can be returned to the client, at which point, the client will have a choice of which DNS to use to find the IP addresses [9].

In order to gather QoS data and know the network conditions in a specific area, we are using another mechanism that we call the QoS Monitor. It is considered to be part of the SCL and acquires such data by querying the clients for network conditions. At this point we are assuming a mechanism that can resolve human-friendly service names to unique Service IDs. For service delivery purposes in the SDL we need mechanisms that will connect service subscribers to the correct instance of a service. Service Tracking and Resolution or STAR keeps a record of Service IDs and in which Clouds their instances are running and also uses input by the QoS Tracking. Using this information, STAR will make a decision on which Cloud is better suited to service a client request based on the location of the client. To achieve this functionality, STAR can look up routing tables in order to identify which Cloud is closer to a user. A choice is always given to a service to reject the new client and forward them to another Cloud if possible. This gives control to service providers and also becomes a contingency mechanism in case STAR

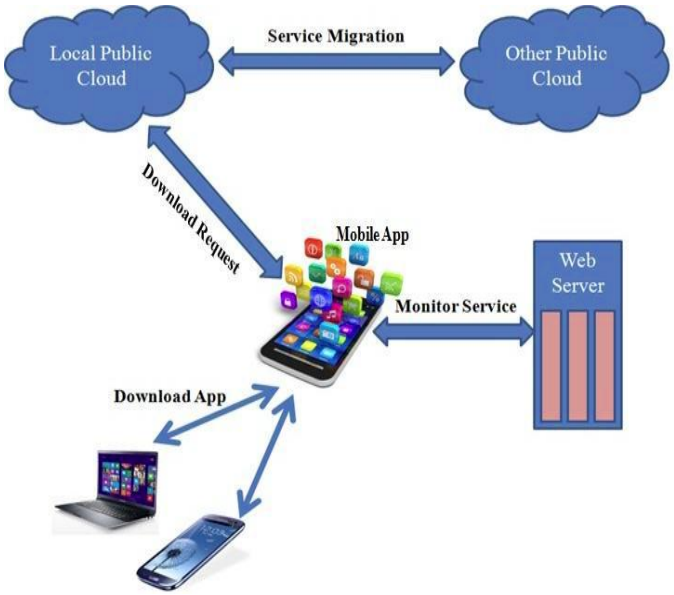

Figure 7. System Architecture

Makes a wrong decision. The STAR server can be scaled similarly to the DNS system since it is essentially the same type of service albeit with some extra parameters. Once a Cloud ID is found, then the ID is resolved into the IP addresses of the Cloud controllers that the client can contact to access the service. The process is shown in the Fig.8. It should be noted that alternatively the Cloud ID can be returned to the client, at which point, the client will have a choice of which DNS to use to find the IP addresses.

Finally, Fig.9 illustrates a simplified global infrastructure for user mobility and service population. Global Service Population Authority (GSPA) also performs SDL functions and makes decisions on when to populate a Cloud based on all the factors given by the aforementioned mechanisms. The instruction to move a service will be given after the target cloud has agreed with the SSLA of the service at which point the next function of GSPA is to update STAR records with new instances of services.

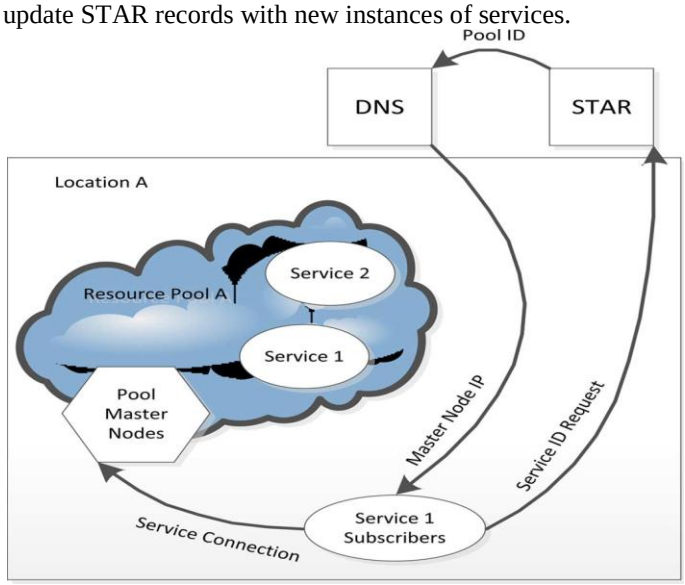

Figure 8. Service request and resolution.

We should note at this point that the GSPA can also be implemented as part of each Cloud so that each Cloud will manage QoS statistics for its own clients. Using this method we can leave it up to individual Clouds to negotiate service migrations instead of receiving instructions from a global mechanism. This allows for a more self-managed design but lacks the central management capability of the GSPA.

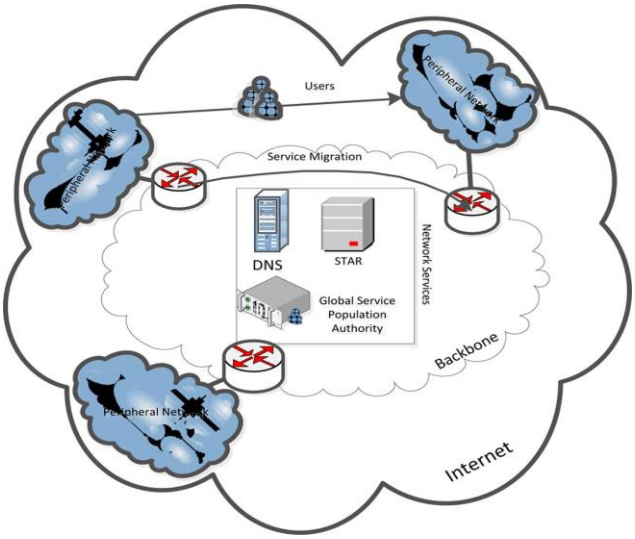

Figure 9. Global service population infrastructure.

Fig.10 shows a handshake diagram on how a client requests a service and how all the layers work to deliver it. The first step is for the client to request a service ID from STAR. This service request includes the location of the client as well as the level of QoS required. STAR will then forward the client to a Cloud ID that hosts the requested service can honor the QoS level. While the connection is active, the client sends QoS metrics to the GSPA. If the GSPA detects that a QoS drops below a threshold, it will signal the Cloud to perform a service migration. When the service migration is performed successfully, the Cloud will also register the new instance of the service to the STAR.

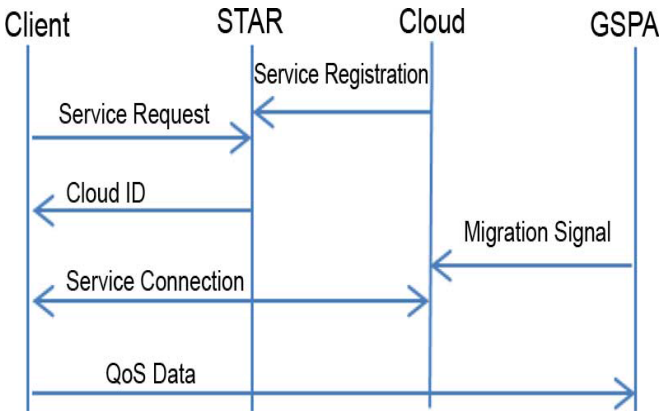

Figure 10. Service delivery handshake diagram.

We have identified however, that moving a service can cause a large overhead on the network. The amount of traffic generated by the migration of a service depends on the size of the service itself and the user files it needs to copy. This means that aside from QoS criteria, any services that migrate gratuitously for unnecessary or minimal QoS gains can cause excessive congestion. A potential solution can be to prevent a service that migrated recently, from migrating again in a short time period. Such behavior would congest a network with more traffic than letting clients connect over a large distance. This is currently an open issue in our research.

IV. CONCLUSION

This paper gives the powerful solution to the problem of security of cloud data during accessing services from cloud. This paper also introduces a new concept of OAuth level 3 protocol which overcome previous problem of cloud data security.

V. REFERENCES

[1] Fragkiskos Sardis, Glenford Mapp, Jonathan Loo, "On the Investigation of Cloud Based Mobile Media Environment with Service Populating QoS Aware Mechanisms," IEEE trasaction on multimedia, vol.15.

[2] http://www.scaledb.com/DBaaS-Database-aa-a Service. php

[3] Hongli Luo**,**Mei-Ling Shyu, **"**Quality of service provision in mobile multimedia - a survey," Human-centric Computing and Information Sciences *2011*

[4] Hutchison, D., Coulson G., Campbell, A., and G. Blair , "Quality of Service Management in Distributed Systems," M. Slomaned., Network and Distributed Systems Management, Addison Wesley, Chapter 11, 1994.

[5] D. Gupta, S. Lee, M. Vrable, S. Savage, A. C. Snoeren, G. Varghese, G. M.Voelker, and A.Vahdat, "Difference engine: Harnessing memory redundancy in virtual machines," in Proc. OSDI, 2008.

[6] Mandeep Devgan, Kanwalvir Singh Dhindsa, "QoS and Cost Aware Service Brokering Using Pattern Based Service Selection in Cloud Computing," International Journal of Soft Computing and Engineering, Volume-3, Issue-2, May 2013

[7] http://blog.oauth.io/oauth-3-0/

[8] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," IEEE Signal Process. Mag., vol. 28, no. 3, pp. 5969, May 2011.

[9] Swati Vani, Bhagyashri Bhosale, Ganesh Shinde, Rajni Shinde, Prof.Manoj Pawar, "Cloud-Based Multimedia Storage System with QoS Provision," International Journal of Computer Science and Information Technologies, Vol. 5, 2014, 1173-1176

