International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 3, May-June-2020, ISSN (Online): 2395-566X

A Concise Study of Cloud Computing Web Servicesin the Real World

Ph.D. Scholar Momin M. Nadeem, M. Y., Assistant Professor Dr. Gyanendra Kumar Gupta
IT&CS Department
Kalinga University Raipur, Chhattisgarh, India

Abstract – **Cloud Computing makes the dream of computing real as a tool and in the form of service. This internet - based ongoing technology which has brought flexibility, capacity and power of processing has realized service-oriented idea and has created a new ecosystem in the computing world with its great power and benefits.Cloud computing is fast creating a revolution in the way information technology is used and procured by organizations and by individuals.Cloud Computing can be described as web serviceoriented computing that provides an environment whichacts as a service in delivering software and information management in a way that would have typically only been available in product format. This is done throughPersonal devices – such as a laptop – that would access the services available through the network of servers that is called the "cloud". In this paper we have discussed on study of the motivation factors of adopting cloud computing, web services and their types. It also explores SOAP and REST full messaging styles are considered as adoption reasons for cloud computing environment. We also include web services security, designing web services and WSDL.**

Keywords– Cloud Computing, web Services,SOAP web services, REST full web services, web services security.

I. INTRODUCTION

The Cloud Computing was established in late of 2007, currently emerges as a hot topic due to its abilities to provide flexible dynamic IT infrastructures, Quality of Services guaranteed computing environments and configurable software services. In short period, Cloud Computing is completely real and will affect almost everyone. In this day and age, we have all become stakeholders in the computing movement, and we are all affected when major changes occur. Now, Cloud Computing is shifting that computing power back to hosts again. Only this time things are different, since those hosts have become abstract, and are spread all over the Internetand world. That is to say that computing power is being shifted to the ―cloud‖. Such a shift to Cloud Computing would not have been possible until now.

Broadband connectivity now makes Cloud Computing a realistic possibility for larger companies, small businesses, and individual consumers. Now these users have chance to access the cloud, and they also have access now toapplications and services. Few years ago they couldn't begin to access or afford applications and services. Cloud Computing guarantee to deliver all the functionality of existing information technology services even as it dramatically reduces the upfront costs of computing that prevent many organizations from deploying many cutting edge IT services. All such guarantee has led to superior expectations.

Fig.1 connectivity makes Cloud Computing .

II.TYPES OF CLOUD SERVICES

These services are broadly divided into three categories as shown in above figure.

1. Platform as a Service (PaaS): In the cloud PaaS is a collection of software and development tools hosted on the provider's infrastructure. Developers build and deploy their applications on the provider's platform. The customer may use API‟s and software installed on a PaaS provider. The aim is to design and deploy the application in a quick and efficient manner.

2. Infrastructure as a service (IaaS): It is a preliminary point for anyone looking to receive aIaaS cloud services. The Iaa Smarket, has proven to be one of the most exciting ones in the cloud space, and there have been several important factors, such as changes in pricingstrategies, Pricing information, Compatibility operating Systems and languages, supporting services and many other have to be considered in order to choose particular service provider.

3. Software as a Service (SaaS): This is where cloud service providers enable their clients to get access of infrastructure, software's etc. From productivity applications and CRM app's suites to software programs which manage cloud applications and deployments and even enable the creation of hybrid clouds, software as a service is exceptionally broad and runs the scope. Here we consider cloud software and application providers that are performing in different way, something new or something predominantly well.

III. CHARACTERISTICS OF CLOUD COMPUTING SERVICE

There are many different cloud computing services present ,but they all are characterize according to type of services offered .Now we examine the common and specifice characteristics of Iaas, Paas, and SaaS cloud services. It is likely that the list can be expanded further,however, the selected characteristics allow more clear distinctions at each level.

1. Common characteristics
The shared characteristics are the license type, the intended user group, the security offered, formal agreements between the provider and the customer. In the following sections, each of these features will be discussed.

2. License type
A lot of cloud monitoring software is open-source based, as well as smaller cloud computing services, since small players often lack the power and influence to push proprietary software on the market. License types also play a role when offering infrastructure- and platform-level services. IaaS providers do not suffer from software licensing issues when renting out their virtual servers without operating systems installed. However, when including operating systems and software packages, this can cause potential problems as to how the customer should be billed when using the service for a limited time period. Often additional fees for the software use need to be paid.

3. Intended user group
Cloud computing services offered are varied according to the users (private, public, corporate). Mainly IaaS and PaaS offerings are intended for companies, whereas SaaS offerings exist for corporations, private individuals or both, such as the Google Apps. However, this does not imply that services aimed at companies cannot be purchased by individuals.

A further distinction can be made between mobile and fixed users. Mobile users access their cloud computing services from anywhere such as office, home and from laptop or desktop. Fixed users are static and always use the same device to connect to the service. Once cloud computing services intended to support smart phones and other lower source devices are available an additional group, based on this hardware type, can be considered.

4. Security and privacy
Security and privacy are major aspects, because we know very useful and important data resides on cloud's data server. Any kind of data leakage and losses cause overall profit ofcompany. There has certain rules are implemented to ensure data protection from any loss.

5. Formal Agreements
Service level arguments (formal arguments) define which level of service, quality of service is given to client. The SLAs include technical specifications of measures, such as uptime or turnaround time. Most SLAs also describe on failure what compensation provided to client. Due to the lack of standards most cloud service providers use SLA agreements to convince potential customers to use clouds "even for mission-critical industrial services.

6. IaaS -Specific Characteristics
The main IaaS specific characteristics are the available operating system, software frameworks and applications. Mostly IaaS providers support Open-Solaris, Linux systems, and Windows .Widely supported applications include the Apache HTTP Server and the MySQL database software. Another characteristic that is important for developers is whether and what kind of development tools the provider supplies. This could include an API or special command line tools. Services comprising virtual instances can be further differentiated based on the virtualization technology used.

7. PaaS-specific characteristics
An important platform-level characteristic is related to which programming languages and environments are supported. Google's App Engine, for example, currently only supports Python and Java environments. The supported operating systems and applications can also be a relevant feature.

8. SaaS-specific characteristics
Software cloud services vary a lot. A characteristic to be considered is the customer/application domain of the offered service. This domain could be customer relations or other business management areas, office applications, social networking, and data exchange.

IV. INTRODUCING WEB SERVICES

1. Understanding Web Services

A Web service is a software application that can be accessed remotely using different XML-based languages. Normally, a Web service is identified by a URL, just like any other Web site. What makes Web services different from ordinary Web sites is the type of interaction that they can provide. Most Web sites are designed to provide a response to a request from a person. The person either types in the URL of the site or clicks on a hyperlink to create the request. This request takes the form of a text document that contains some fairly simple instructions for the server. These instructions are limited to the name of a document to be returned or a call to a server-side program, along with a few parameters. Figure-2 shows this process.

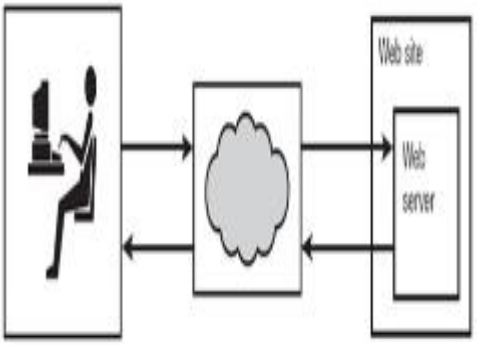

Fig.2. A browser interacts with a Web server to make requests.

2. The Key Components

Web services transactions take place between components. You can either program these components yourself, download them from open source software foundation's such as Apache, or purchase them from commercial vendors such as Microsoft or IBM. There is no requirement that you obtain all the components that you use from a single vendor; you can write some, download others, and purchase still more. At this writing, the following are considered the core Web services standards

3. SOAP

SOAP originally stood for Simple Object Access Protocol. But SOAP is now considered a specification name and not an acronym. SOAP is a specification that defines an XML grammar for both sending messages and responding to messages that you receive from other parties.

4. Extensible Markup Language (XML)

Extensible arkup Language (XML) is the language that all the Web services Languages are built on. XML is a tool for constructing self-describing documents. In fact, XML is more of a meta-language than a language in that it is used to create grammars.

5. Hypertext Transport Protocol (HTTP)

Hypertext Transport Protocol (HTTP) is a standard that precedes the advent of Web services. It was developed to facilitate the transfer of requests from a browser to a Web server.

6. Web Services Description Language (WSDL)

Web services Description Language (WSDL) is a specification that tells us how to describe a piece of software in terms of the method calls that it responds to. The WSDL also contains a concrete section in which the various details of how to actually make a connection to the service are stored.

7. Universal Discovery Description Integration (UDDI)

The Universal Discovery, Description, and Integration (UDDI) specification describes how a potential customer of a Web service could learn about its capabilities and obtain the basic information needed to make the initial contact with the site.

7.1 Advantages of Web Services:

There are many advantages to using the Web services architecture over any other. In fact, some Web services applications would be expensive or impossible to duplicate using any other technology.The advantages of using Web services are

- Integrating legacy systems
- Lowering operational costs
- Lowering software development costs
- Getting systems done faster
- Interfacing with customers
- Integrating with external business partners
- Generating new revenue
- Supporting new business models

V. EXCHANGING MESSAGES WITH SOAP

SOAP messages are XML documents that are embedded in the transport's request and response. In this case, we will learn how SOAP works. We will first learn what SOAP is and where it came from. Next, we will learn theanatomy of SOAP documents so that we will be able to understand them (at least in rough outline form) when we look at them.

1. What SOAP Is

Many of the standard definitions of SOAP sound like buzzwords strung together. Oneparticularly good one is that SOAP is a specification for a ubiquitous XML-based distributedcomputing infrastructure. If we translate these words, we can get a better feel forwhat SOAP really is:

2. Specification

SOAP is not a product that was created and sold by a vendor. Rather, it is a document that describes the

characteristics of a piece of software. The basic idea is that if two parties create programs to the same specifications, these programs will be able to interoperate seamlessly.

3. Ubiquitous

SOAP is defined at a high enough level of abstractions that any operating system and programming language combination could be used to create SOAP-compliant programs.

4. XML-based

SOAP is built on top of XML, which means that SOAP documents are XML documents constructed to a tighter set of specifications.

5. Infrastructure

SOAP does not specify what data can be moved or what function calls can take place over it. An analogy could be made to a railroad car. The car is capable of moving any item that will fit in it from point A to point B. In the same way, software products that are constructed to the SOAP specification can move data from computer A to computer B and hand it to another program written to the same specification. The actual real-world meaning of the data is outside the scope of the SOAP specification. So, a SOAP message is an XML document. Using SOAP can be thought of as a set of layers, as shown in Figure 3

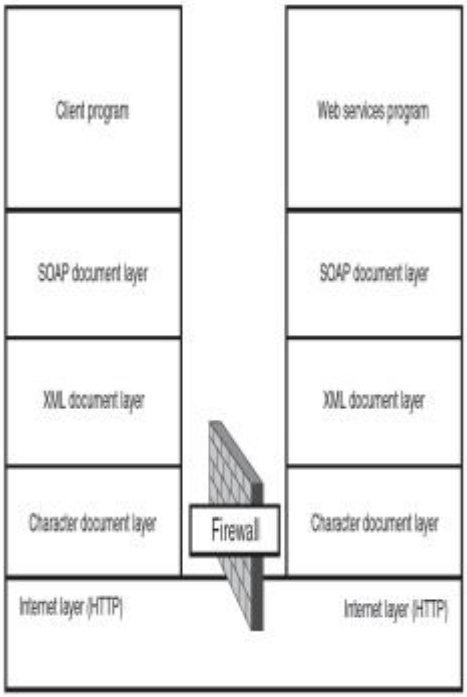

Fig.3. Looking at the layers makes it easier to understand SOAP's multiple personalities.

6. Why SOAP Is Different

You might be wondering how this specification differs from a traditional DCOM or CORBA application. Like a CORBA application, a call is made and a response is returned. At that point, the similarity mostly ends. "Comparing Web Services to Other Technologies," a CORBA client actually makes calls to the object on the server in a tightly coupled way. The SOAP client just formats a text file and transfers it to the other machine. Perhaps the biggest difference between the SOAP and CORBA or DCOM approach comes from the casual nature of the relationship between the two computers.

You set up CORBA programs by generating special files and placing some of them on each computer. SOAP, in theory, doesn't need the name of anything. A client could use UDDI to find a service, upload the WSDL, generate the client, and make the call without knowing anything in advance. In theory, a Web service could publish its existence on a special type of directory called a repository. Clients could then discover it and connect to it with no human intervention. If this kind of casual relationship becomes popular, it could usher in a whole new wave of applications that are based on just-in-time peer-to-peer discovery.

7. SOAP Data Types

One of the most difficult problems for inter computer communication concerns the representation of data types. Declaring data to be of a certain type is fundamental to getting a computer program to work correctly. Data typing **8. Provides three primary advantages:**

Strongly typed data is more efficient to store and process than untyped data. Untyped data processing requires additional processing to determine whether a requested action is allowed on this type of data. For typed data, this is a simple task, but for untyped data, many cycles must be consumed to make this determination.

- Typed data can be combined with other data with a higher degree of confidence than non-typed data. There is also less potential for confusion when doing conversions. The processor decides whether an operation can be performed on a piece of data. Because it is inferring the true type of the data, it sometimes behaves differently than the programmer thought it would. Typed data makes it easier for the language compiler to recognize and reject nonsensical operations such as multiplying your social security number, which is stored in a string variable number, by the number 3. SOAP does allow us to pass data without data type information. If we specify no type,
- The default type of string is used. This default can be overridden by the inclusion of a string that indicates the type of the data along with the string version of the data itself, as shown here: <accountNumberxsi:type="xsd:int">123456780</accountNumber> The simplicity of this scheme is impressive.

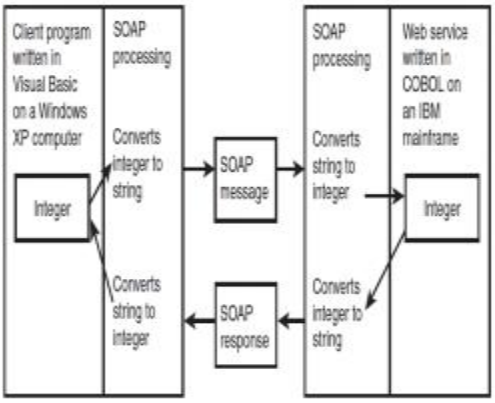

Fig.4. Data type conversion is the responsibility of the SOAP message creators and consumers.

VI. RESTFUL STYLE WEB SERVICE

1. What is REST?

The term REST comes from Roy Fielding's PhD dissertation, published in 2000, and it stands for REpresentational State Transfer. REST by itself is not an architecture; REST is a set of constraints that, when applied to the design of a system, creates a software architectural style. If we implement all the REST guidelines outlined in Fielding's work, we end up with a system that has specific roles for data, components, hyperlinks, communication protocols, and data consumers.

The following constraints that define a Restful system:

- It must be a client-server system
- It has to be stateless—there should be no need for the service to keep users' sessions; in other words, each request should be independent of others
- It has to support a caching system—the network infrastructure should support cache at different levels
- It has to be uniformly accessible—each resource must have a unique address and a valid point of access
- It has to be layered—it must support scalability
- It should provide code on demand—although this is an optional constraint, applications can be extendable at runtime by allowing the downloading of code on demand, for example, Java Appletswe can now look at the abstractions that make a RESTful system, namely resources, representations, URIs, and the HTTP request types that make up the uniform interface used for client/server data transfers.

2. Resources

A RESTful resource is anything that is addressable over the Web. By addressable, we mean resources that can be accessed and transferred between clients and servers. Subsequently, a resource is a logical, temporal mapping to a concept in the problem domain for which we are implementing a solution.

These are some examples of REST resources:

- A news story
- The temperature in NY at 4:00 p.m. EST
- A tax return stored in IRS databases
- A list of code revisions history in a repository like SVN or CVS
- A student in some classroom in some school
- A search result for a particular item in a web index, such as Google

3. Representation

The representation of resources is what is sent back and forth between clients and servers. A representation is a temporal state of the actual data located in some storage device at the time of a request. In general terms, it's a binary stream together with its metadata that describes how the stream is to be consumed by either the client or the server (metadata can also contain extra information about the resource, for example, validation, encryption information, or extra code to be executed at runtime).

For human-generated requests through a web browser, a representation is typicallyin the form of an HTML page. For automated requests from other web services,readability is not as important and a more efficient representation can be used suchas XML.

4. URI

A Uniform Resource Identifier, or URI, in a RESTful web service is a hyperlink to a resource, and it's the only means for clients and servers to exchange representations. In a RESTful system, the URI is not meant to change over time, as the architecture's implementation is what manages the services, locates the resources, negotiates the representations, and then sends back responses with the requested resources. More important, if we were to change the structure of the storage device at the server level (swapping database servers, for example), our URIs will remain the same and be valid for as long the web service is online or the context of a resource is not change.

5. Restful Web Services: The Service Side

Java has options for implementing and publishing RESTful web services. On the publishing side, the choices range from very basic, command-line servers that are well suited for development, testing, and even low-volume production; through lightweight, Java centric web servers such as Tomcat and Jetty; and up to full-blown Java Application Servers (JAS) such as GlassFish, JBoss, Oracle WebLogic, and WebSphere. There is also variety among the APIs and their implementations for RESTful services. Here is the list of APIs covered as shown below:

- The HttpServlet and JSP APIs
- The JAX-RS (Java API for XML-RESTful Services) API
- The third-party Restlet API, which is very similar to JAX-RS in look and feel
- The JAX-WS (Java API for XML-Web Services) API, the @WebServiceProviderinterface in particular

6. A RESTful Service as an HttpServlet

The JSP-based service supported only GET requests. This section revises the example to provide an HttpServlet implementation with support for the four CRUD operations:

- A new Prediction can be created with a POST request whose body should have two key/value pairs: who's key whose value is the name of the predictor and what key whose value is the prediction.
- The Prediction objects can be read one at a time or all together with a GET request.If the GET request has a query string with an id key, then the correspondingPrediction, if any, is returned. If the GET request has no query string, then the fulllist of Predictions is returned. On any GET request, the client can indicate a preferencefor JSON rather than the default XML format.
- A specified Prediction can be updated with a PUT request that provides the identifierfor the Prediction and either a new who or a new what.
- A specified Prediction can be deleted.

VII. WEB SERVICES SECURITY

1. Defining Web Services Security

The openness of the Internet is one of its greatest assets. From the standpoint of security, however, this openness is its greatest liability. It is great to be able to connect and order an authentic South Korean Soccer Team home jersey from a Web site in Hong Kong. If you think about the number of wires that your request (that includes your credit card number) has to travel over and the number of different hubs that it must pass through to get there, you will be nervous. It you add to that fear the fact that these transactions are very attractive to criminals; you would have a justifiable panic. Securing Web services is a conceptually simple topic.

Although the details of how to program different techniques and implement different technologies can seem complex, the issues are easy to understand. For an XML transaction between your computer and mine to complete securely, you have to be able to say the following:

1.1 Authenticity

we are certain that the Web services transaction took place between my client and your server. Another client could impersonate me, or another server could pose as you.

1.2 Privacy

we are sure that our messages and responses were not "overheard" by any unauthorized person.

1.3 Integrity

we are both sure that every message that the client sent to the server and that the server sent to the client arrived unaltered.

1.4 Nonrepudiation

Both the client and the server can prove to everyone's satisfaction (including each other) that the transaction actually took place (if a question were to arise) and that the parties involved were indeed who they said they were.

2. Attacks in Web Services

Any attempt by an unauthorized person to access a target is called an attack. Every attack has unique characteristics, but in general they can be grouped in the following categories:

2.1 Theft of Information

some information is very valuable to a thief. Credit card numbers and personal identification numbers (PINs) for automated teller machines (ATMs) are very popular because they can provide quick access to your money.

2.2 Unauthorized Modification

many messages could contain information thatmust arrive unaltered. Intercepting, altering, and forwarding a message could cause serious problems.

2.3 Impersonation

many attacks occur when your system thinks that it is communicating with an authorized user when, in reality, it is communicating with a criminal.

Some Web services have characteristics that make them more likely to be attacked in one way than in another. A banking system might be more likely to suffer from a theft of information than from unauthorized modification. Your security strategy for each Web service will need to be tailored based on these characteristics.

3. The Web's Security Infrastructure

Fortunately for us, the Internet's security infrastructure was created long before Web servicescame into being. This infrastructure was created to facilitate the transfer of private information across the Internet from browsers to Web sites. The establishment of online stores such as Amazon.com and Buy.com drove much of the demand for this.A key piece of this infrastructure is the public key certificate. A person or organization who wants to engage in secure Internet communication can obtain a public key certificate that identifies him to actual and potential Web services trading partners. Certificate authorities such as VeriSign issue these certificates.

Certificate authorities are trusted companies who create certificates in accordance with the ITU-T X.509 Certificate Standard. They normally gather quite a bit of informationabout an applicant before granting him a certificate. Much of this information is placedinside the certificate before it is encoded with the authorities' private key. Two companieswho provide these certificates are VeriSign, www.verisign.com, and Thawte, www.thawte.com.

For example, a person armed with your public key can encrypt messages that only you can decrypt because you

are the only one who possesses the private key. Public keys have two features to make them useful. It is very difficult to use only a public key to decrypt a document that was encoded with that key. Next, it must be extremely difficult to figure out the private key by examining the public key. Otherwise, you might as well send clear text. Figure 5 shows how public and private keys work to encrypt amessage.

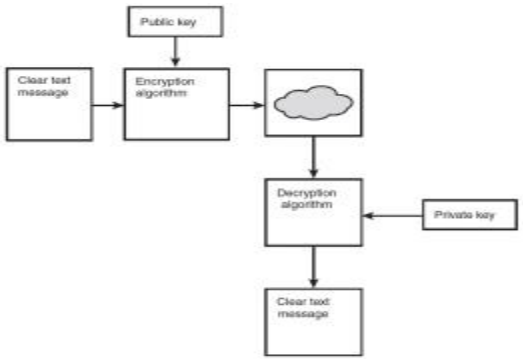

Fig. 6. SSL allows a client to send a secure message to a server..

Notice that everything that goes over the Internet gets encrypted, not just the sensitive parts.

4. Limitations of SSL:

SSL is like a tired workhorse. For all the good work that it has done over the years, it lacks the pep to go to the next level. The reasons for this are

- The fact that it provides encryption during transport ignores the fact that encryption is also needed while some documents are stored on intermediate servers.
- The whole message is encrypted and decrypted together. This keeps an intermediate end point from adding to a message without discovering its contents.
- SSL, by itself, cannot provide authentication, data integrity, and nonrepudiation for the life of the message if it is routed across more than one Web server.

VII. CONCLUSION

We may look back on the latter portion of this first decade of the new millennium as a trueturning point in the history of computing. The transition, however, will take years, perhaps evendecades, and as Nicholas Carr observed, “We’re not going to wake up tomorrow and get all ourcomputing requirements through a socket in the wall”. Pick your weather analogy - between “the perfect storm,” a “cloudy day ahead,” a “cloudburst,” or the like—to represent the vast possibilities that are being brought about by the adoption of cloud computing.

Cloud computing is the cost, time and performance effective technology. Of course the usage of cloud computing will surely will increase more in next few years. In this paper we have discussed basic of cloud computing and web services basics in the cloud computing. Some security issues are the key concern in web services of the cloud computing. Especially privacy and integrity of data are the key concern security issues. In the cloud as data is stored publically and we really don’t know where the data is being stored, we don’t know the exact location of the data, due to this data stored in the cloud has a higher risk of being accessed by un- theorized person during storage as well as transmission.

REFERENCES

[1]. G. Eason, B. Noble, and I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (references)

[2]. AmitGoyal and Sara Dadizadeh, “A Survey on Cloud Computing” University of British Columbia, Technical Report for CS 508, December 2009.

[3]. The NIST Definition of Cloud Computing.Version 15 10-7-09.

[4]. “software as a service strategic Backgrounder “http://www.siia.net/estore/ssb-01.pdf.

[5]. An overview of cloud market vendor landscape http://www.tanejagroup.com

[6]. Hilley D (2009), “Cloud computing: a taxonomy of platform and infrastructure-level offerings,”. Tech Rep GIT-CERCS-09-13,CERCS, Georgia Institute of Technology.

[7]. Google Apps. http://www.google.com/apps

[8]. Murray P (2009), “Enterprise grade cloud computing,”. In: WDDM‟09: proceedings of the third workshop on dependable distributeddata management. ACM, New York, pp 1–1, 2009

[9]. Lin G, Fu D, Zhu J, Dasmalchi G (2009) Cloud computing: IT as a service. IT Prof 11(2):10–13

[10]. Tippit Inc. (2008) WebHostingUnleashed: Cloud-computing services comparison guide. http://www.itsj.com/Resources/cloudcomputing-comparison.pdf. Visited: 2010, Feb 3

[11]. C.N. Höfer G. Karagiannis, “Cloud computing services: taxonomy and comparison,” 19 June 2011 Springerlink.com

[12]. Torry Harris A Cloud computing services -a comparison

[13]. HUANG, W., GANJALI, A., KIM, B., OH, S. and LIE, D. (2015). "The State of Public Infrastructure-as-a-Service Cloud Security. "ACM Comput.Surv.47, 4, Article 68 (June 2015), 31 pages.Salih, A. (2016). A survey of Cloud Computing Security challenges and solutions.Najran University.

[14]. CdrNimitKaura, W. and Col AbhishekLal, L. (2017). "Survey Paper On Cloud Computing Security."International Conference on Innovations in Information, Embedded and Communication Systems.

[15]. Li, W. and Lu, D. (2012). "Study on Cloud Computing." Applied Mechanics and Materials, 263-266, pp.2020-2023.
[16]. http://www.academia.edu/760613/Survey_of_Virtual_Machine_Migration_Techniqueswebcite
[17]. Ranjith P, Chandran P, Kaleeswaran S (2012) On covert channels between virtual machines. Journal in Computer Virology Springer 8:85-97 Publisher Full Text
[18]. Cloud Security Alliance (2010) Top Threats to Cloud Computing V1.0.
[19]. Available:https://cloudsecurityalliance.org/research/top-threats website.
[20]. ENISA (2009) Cloud Computing: benefits, risks and recommendations for information Security.
[21]. http://www.enisa.europa.eu/activities/riskmanagement/files/deliverables/cloud-computing-riskassessment webcite
[22]. Subashini S, Kavitha V (2011) A survey on Security issues in service delivery models of Cloud Computing.
[23]. F. Gens., New IDC IT Cloud Services Survey: Top Benefits and Challenges, 2009.
[24]. A Platform Computing Whitepaper, Enterprise Cloud Compt.: Transforming IT, pp6,2010.
[25]. Dooley B, Architectural Requirement of The Hybrid Cloud, Information Management Online, 2010.
[26]. Global Netoptex Incorporated, Demystifying The Cloud. Important Opportunities, pp 4-14, 2009.
[27]. Kortchinsky, K. , Cloudburst: A VMWare Guest to Host Escape Story, BlackHat, USA, 2009.
[28]. Google App Engine. http://code.google.com/appengine
[29]. Manish Pokharel, Young Hyun Yoon, Jong Sou Park, Cloud Computing in System Architecture, 2009.
[30]. Zissis D, Lekkas D (2012) Addressing Cloud Computing Security issues.
[31]. Jansen W, Grance T (2011) Guidelines on Security and privacy in public Cloud Computing.
[32]. Mell P, Grance T (2011) The NIST definition of Cloud Computing. Gaithersburg, MD: NIST, Special Publication 800–145