



## **A SURVEY ON CLOUD COMPUTING FROM BANKING SYSTEM PERSPECTIVE**

**Samridhi Gupta<sup>1</sup>, Satwik Sabharwal<sup>2</sup>, Thirunavukkarasu K<sup>3</sup>, Dr. Ajay S. Singh<sup>4</sup>**

<sup>1,2</sup> 2 Yr-B.Tech (CSE), Galgotias University, India

<sup>3</sup> Assistant Professor, SCSE, Galgotias University, India

<sup>4</sup> Professor, SCSE, Galgotias University, India

**Abstract**— Cloud Computing is emerging amongst all the ascending technologies as the most prodigious process in the chronicles of computing. As it still takes time to settle, a new challenge as felt whilst its implementation across has been a relatively more gratuitously new. Today as Cloud technology is increasingly expanding across the globe for its resulting benefits but still needs time to offer its full development. This technology provides a new way to trace banks information system. This paper in brief presents the characteristics and applications of the cloud computing. It also explores the innovative applications of cloud computing and elaborates the defects of the framework in banks.

**Keywords**—cloud computing; banking sector; services; technology

### **I. INTRODUCTION**

In 1994, a consortium of software engineers- the Pennine Group belonging to the university of Durham, University of Manchester Institute of Science and Technology and Keele University came up with a totally new idea of how the functionality of softwares is delivered to the users.[1]

The heart and spirit of SAAS is a demand-led market of softwares where software services are provided at the address of the requirement. SAAS emphasizes on the isolation of ownership and possession of software from its use. This concept of delivering the functionalities of softwares overcomes the limitations to the use of software, evolution and deployment that had arisen due to the conventional model of providing software services. This new model of software deployment opened the doors to new markets for all types of software service providers. Be it small scale specialist service providers or any large scale organization.

### **II. SAAS AND OTHER SERVICE FORMS**

Infrastructure Even though the time after 1960s there had been huge advancements in the development environment languages and programming language, there was hardly any change in the preexisting models of construction and maintenance of softwares. Developers still develop softwares through the use of different variations of edit-compile-link cycle in order to generate executable binary image from a source described using a procedural programming language. Even though the due to web the whole idea of what a software actually is, the process of development and implementation of the websites have a little difference if not much from the conventional and traditional processes of constructing software. The service based model of software usage has led to configuration disengaging and execution of services to meet a specific set of requirements[3]- a vision of instant service which is too consistent with the widely accepted definition: “an act or performance offered by one party to another. Although the process may be tied to a physical product, the performance is essentially intangible and does not normally result in the ownership of any of the factors of production.”[4]

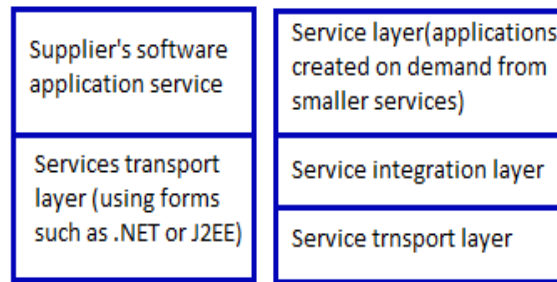


Figure 1. Service models- (a) The current supply led service model provides only a predetermined range of services from a remote server. (b) The proposed demand-led service model has a service integration layer inserted above the transport layer.

#### A. *Service integration layer*

The difference in the two models that have been represented in the figure 1 lies in the functionality that is the result of the combination of the facilities that the service integration layer provides with those from the component services that make up an application. In this layer the software technology is employed to support a set of concepts that have close relation to the supply and business models

#### B. *Service description*

Service description gives a way to for mapping between the provider's descriptions of the interfaces, functionality and the characteristics and constraints that are non-functional such as the cost and quality of service. It explains the parameters within which the client and the service providers negotiate with each other.

#### C. *Service discovery*

Service discovery is used by various users in locating services that may match their requirements and criteria of selection. With the use of these services a client is able to recognize the potential service providers whose offerings comply with its fundamental needs and those who are well prepared for negotiation within some acceptable bounds. Discovery can involve the recursive use of other services, including the brokers, and will eventually lead to a list of the candidate services and the providers.. Service negotiation includes the interaction of the client with one or more of the service providers that are identified either due to the discovery processes or are already known to the clients. Achieving agreement on the terms and conditions for supplying the services is one of the aims that is achieved through this negotiation.

#### D. *Service delivery*

This function consists of three steps - (1) Invocation, (2) Validation of the invocation (3) Suspension step. The first or invocation step is the calling-for step, a request is by client to the provider to supply the specified services that complies with the agreed terms and conditions. In the next step services agreed upon in the first step is supplied by the provider within the agreed amount of time period. The third step comes into play where the bounds of the provision has not been unspecified. It establishes the point where the provider no longer is required to supply the service to the client.

### III. CURRENT SERVICE RELATED PROOCOLS

The present research literature now and then uses the service-model concept to describe the Web service technologies such as Microsoft's .NET platform. Although the Web services paradigm is very

consistent with respect to our view of the SAAS, there is a further requirement of the developments in order to create a true service-oriented market. From the time when the Web services were first introduced, three protocols based on the XML emerged as the de facto standards.

- SOAP gives a format of messages in the communication with web services and invoking them.
- WSDL (web services description Language) describes how the web services should be accessed.
- UDDI (Universal description, discovery and integration) provides a registry that can be used by the clients to discover the available services.

The above three protocols were sufficient for simple services that required remote-procedure-call style communication. For the services that were more complex, other specifications based on XML provide functions at the intermediate and higher layers in stacks of protocols.

IBM produced one of the stack's original definition in its web services conceptual architecture document.[5] It consisted of three de-facto standards at description and discovery layers, service implementation layer and at the XML-based messaging layer with the service flow layer that incorporated Web Services Flow Language of the IBM.[6] However, IBM's Web Services Flow language has been combined with the microsoft's XLANG protocol in order to generate a totally new set of protocols: the Business process Execution language for web services. The few stacks that are available include details about the semantic web protocols or the more business-oriented Electronic Business Extensible Markup Language (ebXML). Due to these it is still quite unclear that which technologies are compatible. Due to this here is proposed an updated web services stack framework that places the currently available initiatives in context.

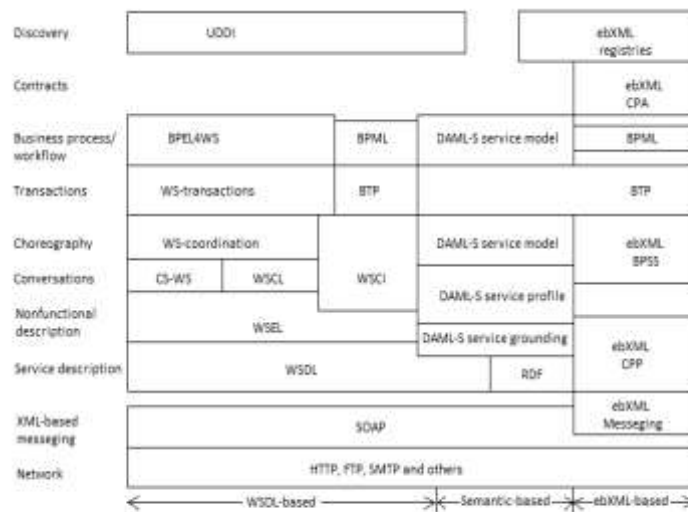


Figure 2. Proposed Web services stack framework. Separated into three vertical sections, the protocols use or extend WSDL, have roots

Figure 2 shows a stack framework that consists of many open-systems-interconnection-type layers in which at each of the levels use the services of the levels below it.

- NETWORK : this is the underlying transport protocol layer.
- XML-based messaging: XML is a format of the message meant for the communication of the communication and the procedural calls. For example this layer can make use of SOAP along with

any other underlying transport protocols in the network layer. Decoupling of the messaging from the physical transport protocol is done in order to let the messages concentrate on describing the service semantics.[7]

- Service description: This layer functionally describes the web services with respect to its implementation and inference. A major part of the description languages present belonging to this layer make use of the XML Schema languages for expressing the data type information.
- Nonfunctional description: at this layer protocols describes the services with respect to the less technical features for example number of retries, geographic location, quality of service, cost etc.[8]
- Conversations: In this regard a conversation implies the external view or the messages that a web service receives and sends.
- Choreography: Whereas all the previous layers are concerned mainly with the task of providing description of single web service, this layer coordinates several web services into a pattern in order to provide an overall result.[9]
- Transactions: The protocols that belong to this layer perform the task of facilitating and monitoring the transactions between web services. Numerous points of failures arise in case the services are themselves made up of services. This layer describes the way in which composition in atomic way is achieved, such that the whole process either gets completed successfully or gets rolled back.[10]
- Business processes and workflow: the protocols belonging to this layer provide a description of the way in which the composition of the higher-level services from other web services.[11]
- Contracts: This layer gives a detailed description of the format of the machine readable formats that are required for automating the service based electronic business.
- Discovery: this layer is utilized by the providers for publishing the details of their web services to facilitate their clients to meet their needs.

## REFERENCES

1. O.P. Brereton and D. Budgen, "Component-Based Systems: A Classification of Issues," *Computer*, Nov. 2000, pp. 54-62
2. K.H. Bennett et al., "An Architectural Model for Service-Based Software with Ultra-Rapid Evolution," *Proc. Int'l Conf. Software Maintenance (ICSM 2001)*, IEEE CS Press, 2001, pp. 292-300.
3. O.P. Brereton et al., "The Future of Software," *Comm. ACM*, Dec. 1999, pp. 78-84.
4. C. Lovelock, S. Vandermerwe, and B. Lewis, *Services Marketing*, Prentice-Hall, 1996.
5. H. Kreger, "Web Services Conceptual Architecture (WSCA 1.0)," 2001
6. F. Leymann, "Web Services Flow Language (WSFL) 1.0," 2001
7. D. Booth et al., "Web Services Architecture: W3C Working Draft," 14 May 2003
8. A. Ankolekar et al., "DAML-S: Web Services Description for the Semantic Web," *Proc. 1st Int'l Semantic Web Conf. (ISWC 2002)*, Springer-Verlag, 2002, pp. 348-363.
9. M. Paolucci et al., "Importing the Semantic Web in UDDI," *Proc. Web Services, E-Business, and Semantic Web Workshop (CAiSE 2002)*, Springer-Verlag, 2002, pp. 225-236.
10. ebXML Trading-Partners Team, "Collaboration-Protocol Profile and Agreement Specification," v. 1.0, 2001;
11. J.E. Hanson, P. Nandi, and S. Kumaran, "Conversation Support for Business Process Integration," *Proc. 6th IEEE Int'l Enterprise Distributed Object Computing Conf. (EDOC 2002)*, IEEE CS Press, 2002, pp. 65-75.