A Model for the Development of Universal Browser for Proper Utilization of Computer Resources Available in Service Cloud over Secured Environment

Bishnu Prasad Gautam, Dipesh Shrestha, Members IAENG[1]

***Abstract*—**Security is a major issue in cloud computing technology. In order to increase the service level of trust among the cloud service providers and its users, understanding of Cloud Computing Architecture requires an understanding of Cloud's principal characteristics such as how the services are delivered and deployed, how they are consumed and ultimately how they need to be partitioned, managed and safeguarded. Due to the uncertain and undefined security definitions in Cloud Computing Technology, business communities often raise concerns in this domain. Until date, most of the big giants of Cloud Computing Technology such as Google, Amazon, IBM and Yahoo have not proposed a trustable and reliable solution for storage of client's confidential data. In this proposal, we propose a different solution and security policy in order to promote a common level of understanding between the users, business communities and necessary security requirements. We are confident in achieving a high level of trust from the user of our proposed security model in which we facilitate the users to retain their data in their own environment so that they will have full control of their data while using the services from the service provider. Furthermore, we have developed an interface in which all kinds of applications can be executed and which is termed as the Universal Browser.

***Index Terms*—** Cloud Computing, Universal Browser, Single Point of Connection

I. Introduction

Cloud computing, a new prototype for solving complex and large-scale problems, is getting diverse attention from varied fields of science and technology recently. Though computational cloud alike services widely known as Grid services are already being used to solve large-scale problems in science and engineering, most of them are focused on defining low-level services. These services are less capable of implementing enterprises services which can be incorporated into a cloud computing framework.

[1] Bishnu Prasad Gautam is affiliated with Wakkanai Hokusei Gakuen University, Wakabadai 1 chome 2290-28, Wakkanai, Hokkaido, Japan. Currently doing research in cloud computing, networking and distributed computing and involved in teaching, management and research in the same University. (Contact phone: +81-162-32-7511, Fax: +81-162-32-7500, e-mail: gautam@wakhok.ac.jp)
Dipesh Shrestha was affiliated with Wakkanai Hokusei Gakuen University, Wakkanani, Hokkaido, Japan. His research interests are data mining, cloud computing and system engineering. He is now working as a System Engineer with DynaSystem Co., Ltd., 1-14, North 6, West 6, Kita-ku, Sapporo, Hokkaido, Japan (Contact phone: +81-11-708-6786, Fax: +81-11-708-6787, e-mail: d_shre@dynasystem.co.jp)

The vision of cloud computing and grid computing is the same as both of the technologies have a common goal to reduce the cost of computing, increase reliability, and increase flexibility and thus it is not a new idea; however the approach of solution and the requirement of analysis of new massive data have changed. As Foster states, "there is a common need to be able to manage large facilities; to define methods by which consumers discover, request, and use resources provided by the central facilities; and to implement the often highly parallel computations that execute on those resources. Details differ, but the two communities are struggling with many of the same issues" [1], [2]. It is obvious that the future of scientific computing relies not only on powerful processing power, huge data centers, fast and advanced networks, but also on the approach of the service oriented architecture. Furthermore, the technological impact in society and the market economy will be another deciding factor for the sustainability of cloud computing technology. To this effect, service oriented solution, though not a new approach would be a significant factor for enterprise consideration for which cloud computing technology should put its attention. A few years back, Grid Computing had started to architect the solution in Service Oriented Architecture to impress Web service technology and define the standard interface for business purposes [3], [4], [7]. However, the solution of grid computing alone could not meet the demand of e-business in the market and the demand of the end users.

Furthermore, the service level security in the cloud services is not carried out sufficiently. This certainly puts high risk for enterprises that do not want to sideline their clients with security concerns. The proposed research will aim to introduce trouble-free and comfortable ways of developing cloud services so as to initiate the remarkable participation on cloud based computing environments without having security concerns. The paper is organized as follows. Sec. I states the main issues of the research and Sec. II develops a baseline overview and main features of the application that we developed followed by the programming model in Sec III. Sec. IV, Sec. V and Sec. VI further elaborate about the overall scenario and the single point of connection of the system in relation to the RSI programming model. Sec. VII states about the legal and other security issues of the research. Finally, section VIII concludes the paper with highlighting the future works and our

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

self-evaluation of the research.

PROBLEM STATEMENT

We have found two major issues in Cloud Computing technology.

A. *Interface:*

One of which is that there is a lack of research in developing a universal interface on which all kinds of software components or applications can run. This issue would be the fundamental issue for the enhancement of the web in the coming days. We have observed that the web is becoming an essential information resource for all everyone. Yet millions of users are unable to access this important resource due to inaccessible web sites, documents and resources on the web. There are large numbers of services or resources in the web which if allowed using with simplicity or from the same browser or interface will add another level of accessibility to the end-users. While the benefits of cloud computing enabled services are well utilized by highly literate users and organizations, their potential to support knowledge exploring work which is required for general users have not been achieved as expected. A main reason for this situation is that cloud computing systems and knowledge exploring work have different characteristics and were developed in different infrastructural settings till date. Without coordinating these two different settings, it will be difficult to achieve the maximum utilization of resources. In order to properly coordinate knowledge exploration through cloud, shall we provide a tool or an interface from which general users can be connected with the pool of application through which they can be able to utilize the resources as per their needs?

B. *Security:*

The other issue we have observed is regarding security. The present solutions provided by cloud service providers do not ensure proper security. This problem persists when moving to cloud-based computing services as cloud users have to hand over control to the cloud provider on a number of issues, which may affect security. Therefore most of the companies are still reluctant to use this technology. For example, the providers may not assure the users of the encryption on their data and they also may not be tested with vulnerability assessment and penetration testing. Most of the general users and corporate users are reluctant to use this technology as the service providers are not giving assurances regarding the controls surrounding computing resources. In order to give an alternative and reliable assurance to the users, can we provide the option for cloud users to retain their data in their own system so that they have control over their own data?

II. FEATURE AND OVERVIEW OF THE RESEARCH

We have developed a prototype of the application during our past research [5] and our future research will be based on this in order to accomplish the maximum result. We have introduced a system termed as RSI system which stands for Remote Service Invocation, a modified model of

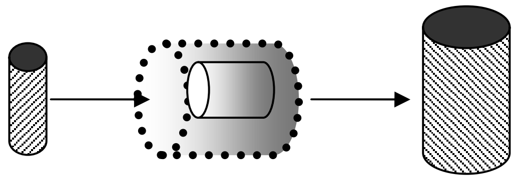

Figure 1：Service Wrapping Process

programming based on RMI architecture. The prominent difference between RMI and RSI system is the coupling of the services in the system. Services implemented in RSI system will be loosely coupled as shown in the figure 2.

Based on this core feature, a time based resource allocation module will also be developed which will play the key role for a service-on-demand system [5].

In a service side, a service wrapping tool as shown in figure 1 has been implemented. This tool will increase the productivity of the system and hence decrease the overall cost of the development of the services. Furthermore, figure 5 demonstrates the graphical overview of the system that will be provided as interface for the end-users. A similar kind of interface is also implemented for the service producer too.

III. PROGRAMMING MODEL -REMOTE SERVICE INVOCATION

The application built on Jyaguchi architecture [5] exploits the maximum usage of Jini programming model [14]. However, please note that Jyaguchi architecture does not advocate any particular programming model. It should be suggested that Jyaguchi applications were developed with full utilization of Jini in our research. As we know Java based distributed systems mostly rely on RMI for wire level communication, we have fully utilized the programming model adopted in RMI and this was further extended to RSI.

The test application which is based on this model also exploited most of the lower level features of RMI. But there are a few differences between the approach of RMI and RSI. Please note that RSI is not a different protocol; it does not replace RMI for low level communication. Instead, it leverages the RMI architecture with a new programming model and techniques while implementing distributed applications. It is a model or a new architectural pattern of the services that participate in the Jyaguchi cloud environments. You can see the difference between RMI and RSI in figure 2. Please note that at the client program, in the RMI system, interface is compulsory. It must be pre-wired in the client program. This made the RMI system tightly coupled, which is obviously not that favorable in distributed applications. However, in RSI system this requirement is avoided. Instead, the client program will require some information about the server program but this information is not necessarily pre-wired. The information about the server program will be transferred later on in a semi-permeable way. This trick made the RSI system to be more loosely coupled.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

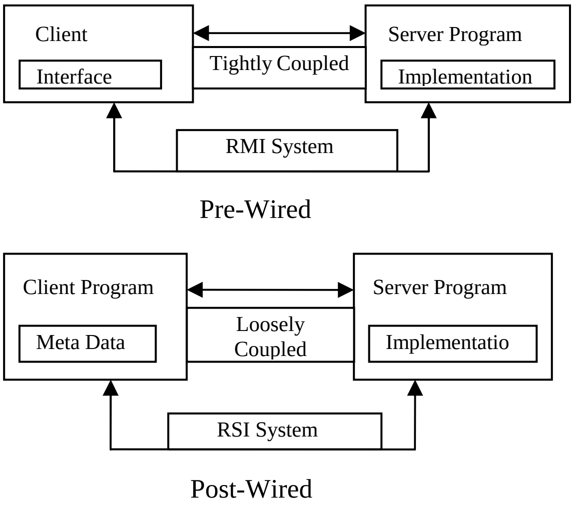

Figure 2 : Remote Service Invocation

During the research we maintain the programming model of calling a service by minimizing the methods in the interface. The calling of method returns the whole service at the client side. All services contain just a single method and will return a bundle of functionalities while the call is made from the client program. RSI programming model suggests the following procedure during the service development.

a) Write a single method in the interface.
b) Follow the facade pattern while defining the services.
c) Invoke the method of interface from client side in order to get the service.

Resolving a service request within a single method has a number of advantages in distributed applications. The most notable advantage is reduction of complexity of the service invocation. As we witnessed in SOAP where documents have their own protocol of procedure calls and which are layered over HTTP. This increases the complexity of SOAP procedure calls compared to simple document fetches using GET and POST [6], [8], [9]. In the case of web service too, interface can be written in WSDL document. However to generate simple interface, it requires complex definition of XML elements. In our case, we have achieved simplicity by reducing the number of methods in the interface.

IV. System Scenario

The overall flow of the service invocation and utilization process is depicted in figure 3 and figure 4. The minor details of sequences are intentionally omitted in order to simplify the overall scenario. In order to make Jyaguchi services act as utility services, the service provider needs to create an instance of the exportable service object, register this and keep the lease alive. To fulfill our architectural requirement [10], [11], [12], [13] and implement our services, we have utilized the API provided by Jini 2.1.

The sequence diagram figure 4 starts from the service registration process. All services implemented in our applications follow the same process. Service registration process follows two simple phases, first of which starts from finding the service registry. Once the service registry is found, the second phase of service registration starts for which it not only registers the service but also adds it to a resource manager, in order to keep the lease alive for a long time.

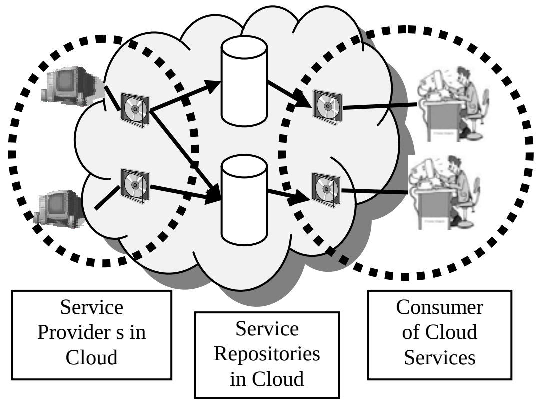

Figure 3: Simplified scenario diagram of Jyaguchi application

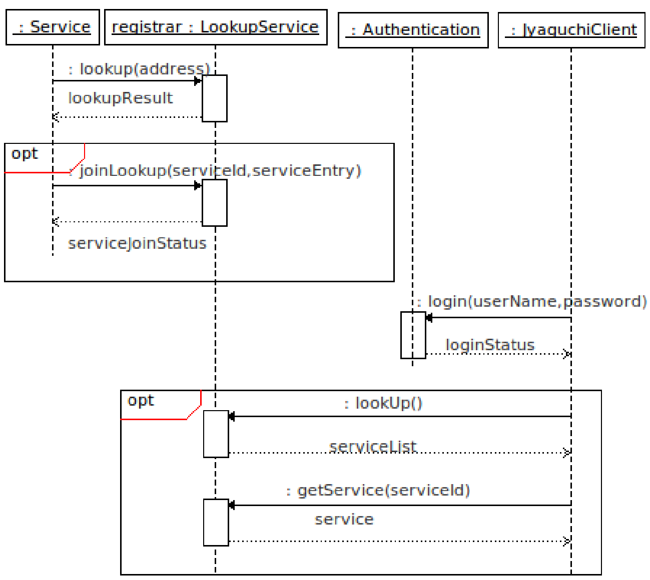

Figure 4: Simplified sequence diagram of Jyaguchi application

Resource manager is a LeaseRenewalManager class provided by Jini API that renews and monitors the entire leasing activity which runs its own threads to keep re-registering the leases. In case the service is terminated by service provider, the lease will fail to renew and the exported service will be discarded from the service registry [14], [15], [16].

On the other hand, in the client side, our system requires proper authentication of the user in order to properly consume the resource. Once the user enters his/her information and upon successful authentication, the client will be able to send a request to search the service registry in the network. To find the registry, we have tested both unicast and multicast discovery. In both processes, the client gets the proxy of a service registrar by which the client will be able to find the service that it wants to consume finally.

V. Experimented Services and Post Notification

The overall service usage scenario has been described in the earlier sections II, III and IV. In figure 5, the real scenario of the service usage has been depicted. As shown in the left most part of figure 5, we can see the service registry which stores the services registered in it. A list of service registry in the network can be depicted within this tool thereby creating a virtual network of services. In the very next part, the registered services are listed. These services are enabled with post notify capabilities which implies that new services can be seen in the browser and dynamically appended in the list.

Post notification and dynamic delivery of the services omit the requirement of complex procedure of software installation for the client. Post notification implies that services deployed in the registry are notified to the end-user without human interference. Thus, end-users do not have to worry about new installation and update of the software. Service providers can update the services in the server without disturbing the client so that client can utilize the updated service immediately.

VI. Universal Browser and Single Point of Connection

We believe that access to technology is crucial in changing human living standard and thus it has direct or indirect role to create the gap between rich and poor in the society. Therefore, we engineers, scientists, researchers are also responsible, to some extent in widening the gap in the society because our contribution led to limit the accessibility to the certain group in the society technology wise, affordability wise. In this research, we realize this point and propose a solution of developing a Universal Browser. In the case of Information Technology too there are millions of users who are unable to access this important resource due to inaccessible web sites, documents and the resources on the web. In order to maximize the accessibility, we realize that we must reduce the complexity of the user interface and provide a simple browser and this kind of browser should be enabled with the functionalities of executing all kinds of application within itself. Our concept of universal browser provides a single point of connectedness with the resource which is distributed across the web. Such functionality will certainly increase the accessibility of the users as it reduces the requirement of setting, installing and updating of the applications.

Single point of connectedness refers to the end-user perspective that end-users do not like to face the hurdles of software installation, update or removal. These sorts of actions require a bit of technical knowledge and relatively higher level of literacy. However, if the services are provided without the requirement of such technical knowledge, the usability of web resources increases more than ever. Universal browser shown in figure 6 provides the single point of connectedness to the end-users. We have tested Editor, Web Browser, Presentation and other few numbers of services which can be viewed in this browser and provides a single point of connectedness. End-users are not required to install these services as these services can be provided upon double click and all are connected in the same browser which we have termed '**Universal Browser**'.

VII. Security and Legal Assurances

Our proposed model is not the solution to the security issues of the present cloud applications; rather we have a different approach on how services can be offered to the users in a trustable environment. The approach of Jyaguchi to security issues can be described in terms of three major terms.

A. Location of Service

The location of services is the place from where service can be accessed. This lies in the Service Repository hosts. The Service Repository can host only the services from the providers that they have built trust with or they can explicitly categorize services accordingly to the level of trust that they define. This enables the clients to use their services upon their level of trust on the Service Repository. To add an extra level of trust between the client and the service providers, services published in the repository can be signed by the providers using public-private key encryptions.

Location aware services are important in order to minimize the legal issues that might arise due to differences in laws in different countries. It facilitates the user to decide the utilization of services on the basis of location, cost, and other factors. This solution enables accurate and quick determination of geographical coordinates of cloud provider.

a) In the location based services, the services are aware of the location where they are hosted.

b) When users want to use them, they are also aware of the location of the users that are requesting them.

c) When some legal issues arise, Location of Execution & Location of Service can be taken into consideration to sort out the issues.

B. Location of Data

The location of data defines where the client's valuable data gets stored. In our architecture, the services providers have the option to choose where the data resides, and the client makes the final decision in choosing which service provider they trust the most. Thus, clients wanting to keep their data at home can choose services that offer complete storage of data in the client's machine. If the client trusts the service provider, the service with storage in cloud can be consumed. The fear of losing control over their own data is why many companies are hesitating to adopt cloud computing. Companies offering services in cloud have been addressing security as a major issue but there is more to be done. Recent news regarding the Google Docs Team's claims that a bug enabled Google doc could be shared with other Google users without the consent of the doc owner [21] is an example of security risks over keeping data in the Internet.

C. Location of Execution

The location of execution defines where the service gets executed. It's the client's computer, and the client's hardware resources like CPU, memory, hard disk, etc. Since the services executes in their machine, clients have the option to define the level of access that they want to give to the services using the policy tools available in Java Development Kit.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

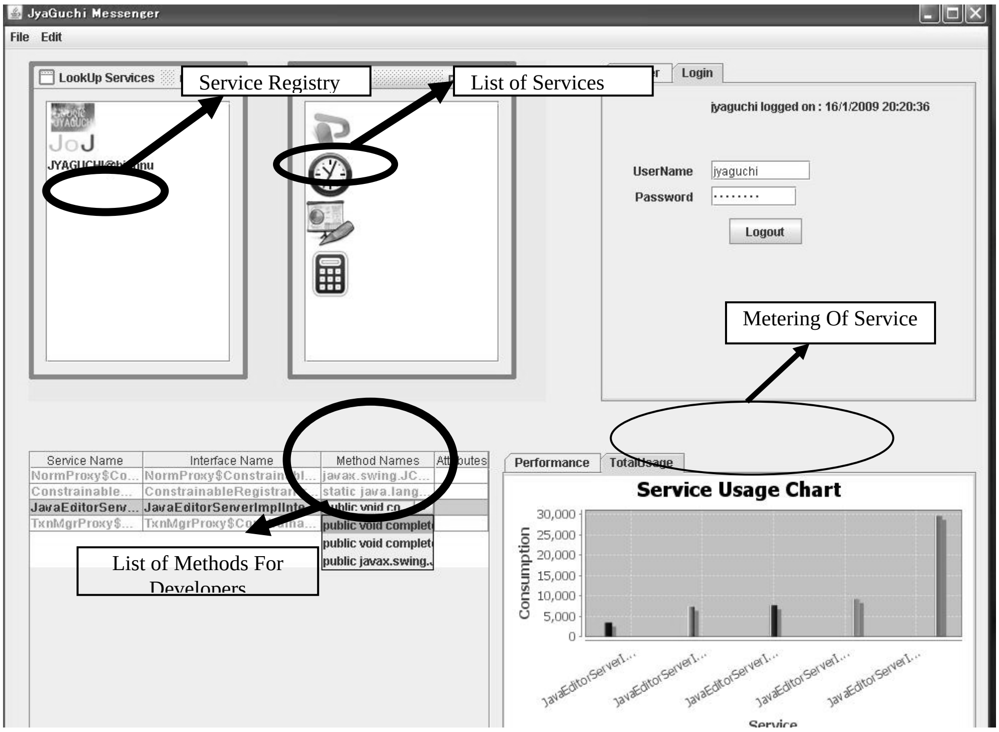

Figure 5: Jyaguchi Application Interface

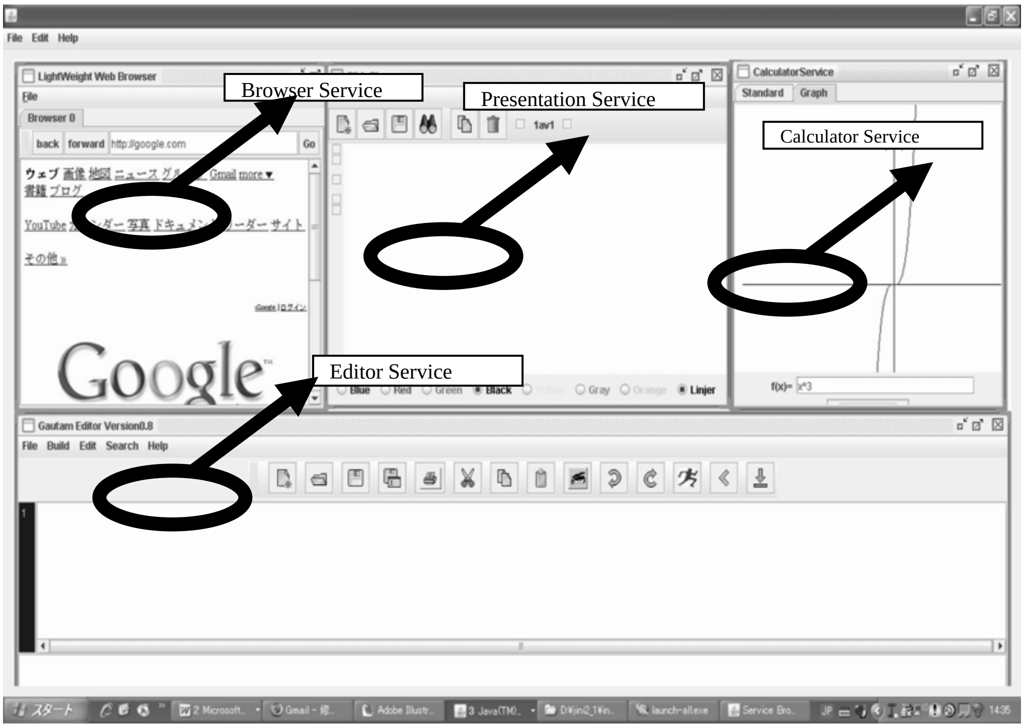

Figure 6: Prototype of Universal Browser

VIII. Evaluation, Conclusion and Future Works

In order to achieve success and improvements of distributed application, an understanding of the architectural elements and the key principles of the architecture is vital. Without identifying the architectural constraints, application developed in ad-hoc basis often violates the architectural principles which may deviate from the targeted goal in the long term. Our motivation behind designing the Jyaguchi application was to propose a model which shows how an improved architectural style can be derived and to apply that style to identify the new and broken features of the application [6], [7], [8].

We have developed the Jyaguchi application after analyzing the architectural constraints of distributed applications and have induced hybrid style [5], [6] into the architecture. We have used reverse engineering techniques by utilizing the reflection API of Java in order to extract architectural information from executing services in the network. It is quite possible to visualize this information at run time and regenerate the process view of the underlying system after implementation. This technique can give an

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

insight to generate the overall architectural representation and to evaluate the architectural representation of the system incurred at design phase. This in fact led to better understanding of the role of architectures in the application life-cycle process.

A. Conclusions

One of the prominent objectives of building services in Jyaguchi architecture is retaining the quality of SOA and to exploit benefits from the hybrid nature of architecture [6], [16] by utilizing the key architectural principles [10], [17] which finally may lead to improvements in building of distributed applications. The architecture of Jyaguchi virtualizes the services and renders the services to the end-users with minimal human intervention. This type of delivering of the services can offer dynamic resource allocation to the end-user thereby providing the services in pay per use basis.

IT enterprises will get no heeded attention shall the services are not modeled as per the business need of the market. In the past, we have witnessed new technologies surface and fade away without impacting society. We believe that Jyaguchi services can be extended to be the services of cloud computing technology. The current trend in IT giants such as Google, IBM, Amazon and others, in terms of cloud computing technology is providing hardware clouds and the services provided by them are server oriented which implies that services are run in the server side and most of the computational load persist in the server. However, this new trend of computing also cannot be taken as the universal way of computing if this technology does not meet the business goals of the enterprises. As SOA suggests, an enterprise business model is a prerequisite for successful technology and this will certainly apply in the case of Jyaguchi services for successful implementation in e-business communities.

The worldwide business demand requiring complex problem solving capabilities has driven IT enterprises the need for dynamic collaboration of many underlying legacy infrastructure or resources to be able to work together. The heterogeneous nature of resources and different security policy of the resource increases the complexities of the system. The resource management challenge in a cloud services infrastructure is a significant topic for the coming days for the researchers. This research addresses these complexities not only in service provider side but also to the general user side and propose the development of user friendly cloud application in order to enable the access of the source of knowledge underlying in the web by the general users via a single point of connection termed as ‘**Universal Browser’**.

B. Future Works

There are two main directions for future work in Jyaguchi application, one of which is at the application level and the other is at the architectural level. While evaluating our service prototype, we identified a number of issues for further research. For instance, we have to limit the size of service and identify what sort of service suit to utilize the services provided by this system. We also would like to develop workload partitioning packages for Jyaguchi services with more precise partitioning function. We also would like to work on the security layer of the application. The current implementation utilizes the security provided by Java security packages and also the policy based security feature by which we can restrict the unauthorized service to participate in the cloud. However, in order to incorporate the architecture into the business, substantial work to support the security of the system is vital. Further, in our future work, we would like to propose to develop the services which allow a user to access services based on their surroundings cloud providers; especially suitable to choose the services as per their trust on the cloud providers.

References

[1] I. Foster, A critique of Using Clouds to Provide Grids, http://ianfoster.typepad.com/blog/2008/01/theres-grid-in.html, September 11, 2008

[2] I. Foster, There's Grid in them thar Clouds, http://ianfoster.typepad.com/blog/2008/01/theres-grid-in.html

[3] R. High, Jr. *et al.,* IBM’s SOA Foundation , An Architectural Introduction and Overview, Version 1.0

[4] S. Carter, The New Language of Business, SOA and Web 2.0, 2006

[5] B. P. Gautam, Master Thesis, An Architectural Model for Time Based Resource Utilization and Optimized Resource Allocation in a Jini Based Service Cloud, Shinshu University, Nagano, Japan, 2009

[6] R. Thomas Fielding, Representational State Transfer, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[7] A. Dutta, The future of SOA -- A service-based delivery model with Web 2.0 capabilities, http://www.ibm.com/developerworks/rational/library/oct06/dutta/

[8] Jini™ Architecture Specification, http://www.cs.princeton.edu/courses/archive/fall99/cs597b/docs/jcpdoc1_0/specs/jini-spec/jini-spec.pdf

[9] J. Newmarch, A RESTful Approach: Clean UPnP without SOAP, http://jan.newmarch.name/publications/soap_rest.pdf

[10] D. Garlan and Mary Shaw, An Introduction to Software Architecture, 1994

[11] L. Bass, P. Clements, R. Kazman, Software Architecture in Practice, Second Edition, 2003.

[12] P. Kruchten, Architectural Blueprints—The “4+1” View Model of Software Architecture, Rational Software Corp. 1995.

[13] C. Gacek, Detecting Architectural Mismatches During Systems Composition, PHD Thesis, December 1998.

[14] W. Keith, Edwards, Core Jini, Java series, The Sun Microsystems Press, Second Edition, 2000.

[15] John *et al.* , Java Series, The Sun Microsystems Press , Java Spaces Example by Example, 2002

[16] J. Waldo, G. Wyant, A. Wollrath, and S. Kendall. *A note on distributed computing*. Technical Report SMLI TR-94-29, Sun Microsystems Laboratories, Inc., Nov. 1994.

[17] P. Eeles, Characteristics of a software architect,IBM, http://www.ibm.com/developerworks/rational/library/mar06/eeles/

[18] J. Kincaid, Google Privacy Blunder Shares Your Docs Without Permission, http://www.techcrunch.com/2009/03/07/huge-google-privacy-blunder-shares-your-docs-without-per mission/, March 7, 2009.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)