ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)

On 21st& 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Resolving Atomic Transaction Issues in Web Services-Business Activities

Rakesh Kumar .S, Ramyadevi .R, Dr.VijayaChamundeeswari

Student, Dept. of CSE, Velammal Engineering College, Anna University, Chennai, Tamil Nadu, India.

Assistant Professor , Dept. of CSE, Velammal Engineering College, Anna University, Chennai, Tamil Nadu, India.

Professor, Dept. of CSE, Velammal Engineering College, Anna University, Chennai, Tamil Nadu, India.

Abstract-In Browser - Server Architecture, Application server with its database and Web Services are the major components of Distributed System. Web Services are modular applications that are self-contained and self-describing. The functions specified in the Web services are performed from simple requests to complicated business processes. In Distributed Transaction, the services are carried out by including the WS-Atomic Transaction (WS-AT) and WS-Business Activity (WS-BA) specification that belongs to family of Web Service Transactions. WS-AT implements core services such as Activation service, Registration service and Coordination service. WS-BA specification does not specify a standard way for an initiator to communicate with the coordinator of Business Activity. To ensure interoperability issues among the initiators and coordinators, BFT algorithm is an optimal solution. To achieve fault tolerance in Web service transactions, Lightweight BFT algorithm is proposed which uses source ordering instead of total ordering of incoming guests.

Keywords— Web Services, Web Service-Atomic Transaction, Web Service-Business Activities, Distributed Transaction, Byzantine Fault Tolerance**.**

I. INTRODUCTION

Web Service is a method of communication between electronic devices over World Wide Web (WWW). Web Services (WSs) are application components that communicate through open protocols.

WSs can be used across Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Universal Description & Discovery Interface (UDDI) since they are modular, self-contained and self-describing. WSs, as a key technology enables interoperability and integration between heterogeneous systems and applications. WSs constitutes fundamental task in architectures for the discovery and selection of appropriate services for the given requests. For registering and locating, the web services aim at describing the service interface and of exchanged messages limiting the discovery process. Though interoperability at the syntactic level is necessary, the selection of appropriate services should be done in terms of semantics of the request. Semantic web enriches the service descriptions with semantic information by allowing software agents to reason the terms in these descriptions.

Finding a Web Service on the Web is a challenging issue in Service Oriented Computing. UDDI is a standard for publishing and discovery of web services and UDDI registries provides keyword searches for Web Services. With the development of Semantic Web technologies, more and more semantic data is generated which is used in web applications and enterprise information systems. Web Services should be semantically annotated to provide the best match to the service requestor as per the requirements.

Web Service provides various functions and business operations which can be deployed over the internet. The three major tasks of this model are Publishing, Binding and Discovering. To satisfy the specific requirements of the customer from the web services, Discovery task is used. Discovery is the most important task in Web Service

model; if Web Services cannot be discovered then it is useless. The interaction between a service requestor, service providers and a service discovery systems are as follows:

1. Web Services must be described in order to invoke the service providers. This facilitates discovery and composition. WSDL or service profile of semantic web service is used to carry out this function.
2. In order to locate the service, the service requester should describe the requirements. Service requesters usually contain a description of Web Service.
3. The service discovery also called as service registry is an intermediate search function. The service providers store their service information in the registry and when there is a request for the information it will be searched. UDDI is used as a registry for Web Service.

Fig 1: Web Service Model [2]

Publish: Through the discovery system, the service providers publish their service information for the requesters to discover.

Discovery: The service requesters retrieve service providers from the service registry. Based on service descriptions, the discovery system will output a list of Web service providers which satisfy the requirements.

Bind: The discovery system provides a number of Web service providers after discovering. The service provider is invoked by the service requesters. The binding takes place at runtime.

II. RELATED WORKS

A *Remote Procedure Call* (RPC) is an inter-process communication that allows a computer program to cause a subroutine or procedure to execute in another address space without the programmer explicitly coding the details for this remote interaction.
The Client will initiate a RPC, which executes a procedure with parameters by sending a request message to a known remote server. The client will receive a response from the remote server based on the request, and the application continues its process. The client will be blocked when the server is processing the request. The client has to wait until the server has finished processing before resuming execution.

Representation State Transfer (REST)is an architectural style that abstracts the elements within a distributed hypermedia systems. REST focuses on roles of components, constraints upon their interaction with other components and their interpretation of significant data elements.

Some of the key constraints in REST are:
1. Everything being resource
2. Resource identification through URI
3. Uniform interface
4. Manipulation of resources through representations
5. Self-descriptive messages
6. Stateless interactions
7. Hypermedia as the engine of application state

The constraints mentioned above represents some architectural properties for REST. Based on some perspectives some comparative analysis is done between REST and RPC.

a. Scalability

In RPC style architecture interface contract is required to interoperate the client and the web service. This does cause any trouble in relatively closed application but in distributed environment this approach causes tight coupling and interface complexity. This prevents scalability. REST works in a different way from RPC, that is it makes use of uniform interface. In REST request and response are performed through HTTP operations namely GET, POST, PUT and DELETE which helps in enhancing the scalability.

b. Coupling

In RPC for new definition of work, data types have to be recompiled. This splits the interface from implementation and makes the client and the server to have a tight coupling. But for web scale, coupling must be loose. In REST the client and the server can select appropriate data format such as images, text, spreadsheets, these types are specific forms of common representational which reduces coupling.

c. Security

The client and the server communicate through SOAP packages. SOAP packages are generally wrapped into HTTP envelopes and use HTTP POST operation to

transfer. This helps the SOAP messages to easily pass through the firewall. This tunneling protocol brings security risks. REST security model is simpler when compared to the RPC. Here every resource is identified through the unique URI. Also wrapping of SOAP message is not required instead four operations are used to make different security policies.

d. Performance

Performance is the one of the advantages of REST. Inherent simplicity makes the performance of REST better. REST also recommends cache responses for the clients to eliminate unnecessary interactions. These two helps REST to provide a better performance.

Web Service Atomic Transaction makes it possible for business to engage in standard distributed transaction processing. Web Service technology also allows business collaboration and integration of applications over the internet. Distributed transaction can be used for such applications. Trustworthy coordination of WS-AT is critical for such applications.

The Activation service, the Registration service, the Completion service and the Coordinator service are the services offered by the Coordinator to the Initiator and the Participants. For each new transaction, the Activation service creates a Coordinator object. A Participant is admitted into transaction by the Registration service. At the request of the Initiator, the Completion service initiates the distributed commit of the transaction. The Coordinator service coordinates the participants to commit or abort the transaction automatically.

The Web Services Atomic Transactions standard specifies two protocols (the 2PC protocol and the completion protocol) and a set of services. The automatic activation, registration, propagation and atomic termination can be ensured. The 2Pc protocol is a run between the Participants and the Coordinator, and the Completion protocol is a run between the Coordinator and the Initiator. To involve in the transaction, the Participants register with the Coordinator. Two phases of transaction is committed by the 2PC protocol. In the first phase (the Prepare phase), a request is disseminated by the Coordinator to all the Participants so that they can prepare to commit transaction. If the commit is success, the Participant prepares the transaction for commitment and responds with a prepared vote else responds with an abort vote. Participants enter the Ready state when it responds with a prepared vote. A Participant must be prepared to either commit or abort the transaction. The Coordinator starts with the second phase (the Commit/Abort phase) when the Coordinator receives the votes from every Participant.

The banking applications which represent distributed transaction phases using a conformant WS-AT are shown in the fig 2. The banking application illustrates that the bank provides the online banking service that a customer can access. The customer invokes a web service of the bank as a transaction, to transfer an amount of money from one account to another.

On the Coordinator-side, the services comprise:

Activation Service: Creates Coordinator object and a transaction context for each transaction.

Registration service: Allows the Participants and the Initiator to register their endpoint reference.

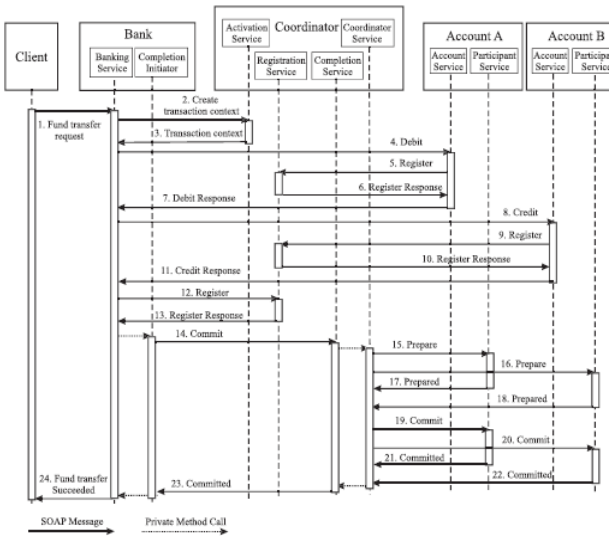

Fig 2: Sequence Diagram for Banking Application using WS-AT [4]

Completion service: Allows the Initiator to signal the start of the distributed commit.

Coordination service: Runs the 2Pc protocol, which ensures atomic commitment of the distributed transaction.

At the Initiator, the service comprises:

*CompletionInitiatorservice:*TheCoordinator informs the Initiator about the final outcome of the transaction using CompletionInitiator service.

At the Participant, the service comprises:

Participant service: The Participant service is used by the Coordinator to solicit votes from, and to send the transaction outcome to the Participant.

Web Service Business Activities (WS-BA) is a collection of tasks that are linked together so that they have agreed upon the outcome of the result. WS-BA allows coordinating the activities that are more loosely coupled than atomic transaction. WS-BA groups multiple units of works so that they can be completed in a common direction using compensational model also it coordinate pattern across different platforms, architecture and implementation. WS-BA also coordinates long running

business activities that are not appropriate by atomic transaction.

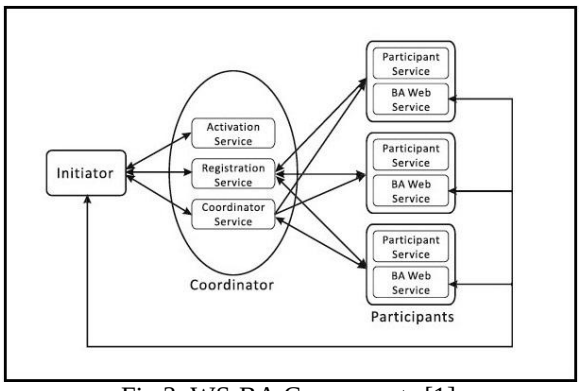

Fig 3: WS-BA Components [1]

A business activity consists of an initiator, a coordinator and one or more participants as shown in the fig.3. The initiator is responsible for the start and termination of the business activity. The initiator coordinates the business activity and also determines the outcome of the business logic.

WS-BA Specification defines the participant and coordinator side services. The coordinator side includes activation, registration and coordinator services and they run on the same address space of the coordinator. To generate the coordination context and coordinator object, the initiator requests the activation service. Business Agreement with Coordinator Completion (BAwCC) or Business Agreement with Participant Completion (BAwPC) are used the coordinator to interact with the participants.

The BAwCC protocol and BAwPC protocols are used by the participants to register with the coordinator of the business activity. For BAwPC protocol, the participant informs the coordinator by sending a completed message when it completes its work. The coordinator replies with either a close or a compensate message. When a business activity is successful, the participant receives a close message else compensate message will be received. For BAwCC protocol, the coordinator sends the notification to the participants. The complete message will sent informing that they will not receive any new requests within the current business activity. The participant replies with a completed message once it successfully finishes its work. Other interaction between the coordinator and the participants are of those BAwPC protocol.

III.PROPOSED WORK

One Time Password (OTP)

In order to verify that the user who has logged into the system was an authenticated one, a randomly generated OTP will be sent to the user's mobile. Once the user receives the OTP, the user has to enter the OTP for authentication. After the user has been authenticated further services will be provided.

A one-time password (OTP) is a password that is valid for only one login session or transaction. The shortcoming in the static password can be avoided by using OTPs. Static passwords are vulnerable to replay attacks while OTPs are not. Since the OTPs are valid for only one session, a potential intruder cannot make use of it by managing a record of an OTP that was already used to log into a service. To avoid prediction of future OTPs, OTP generation algorithms typically make use of pseudo randomness or randomness. Concrete OTP algorithms vary greatly in their details. Various approaches for the generation of OTPs are listed below:

- Time-Synchronization (OTPs are valid only for a short period of time)
- Using a mathematical **algorithm** and **based on the previous password** (OTPs are effectively a chain and must be used in a predefined order).
- Using a mathematical **algorithm** and **based on a challenge** (e.g., a random number chosen by the authentication server or transaction details) and/or a counter.

There are also different ways to make the user aware of the next OTP to use. Special electronic security tokens that the user carries can be used to generate OTPs and show them using a small display. User's mobile phoneand other systems can be used to run the software to generate OTPs. Finally, in some systems, OTPs are printed on paper that the user is required to carry. [9]

Three Phase Commit Protocol (3PC)

In order to overcome the blocking state caused by 2PC, 3PC has been proposed so that the service could take place without any failure between the transaction processes.

The three-phase commit protocol (3PC) is a distributed algorithm which lets all nodes in a distributed systemagrees to commit a transaction.3PC is non-blocking while the two-phase commit protocol (2PC) will lead to blocking state. Before a transaction either commits or aborts 3PC places an upper bound on the amount of time required. The resource locks will be released after the timeout if a given transaction is attempting to commit via 3PC.

A single coordinator site leading the transaction and a set of one or more cohorts being directed by the coordinator is used in describing this protocol. Fig.4 shows how 3PC protocol process takes place.

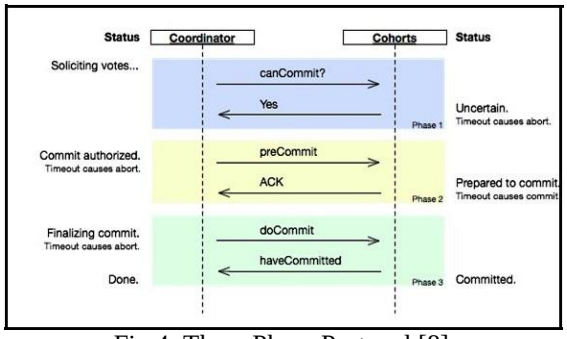

Fig 4: Three Phase Protocol [8]

Coordinator

1. If there is a transaction request, coordinator will send a**canCommit?** message to the cohorts and moves to the waiting state. If there is a failure at this point, the coordinator aborts the transaction.
2. In the waiting state, if the coordinator receives a **No** message or failure or timeout, the coordinator aborts the transaction and sends an **abort** message to all cohorts. Else the coordinator sends **preCommit**messagesto all cohorts and moves to the prepared state.
3. The coordinator will move the commit state if it succeeds in the prepared state. The coordinatorwill abort the transaction when there is a timeout while waiting for an acknowledgement from a cohort. When all the acknowledgements are received, the coordinator moves to the commit state as well.

Cohort

1. If the cohort agrees for the canCommit? message from the coordinator it sends a Yes message to the coordinator and moves to the prepared state. Otherwise it sends a No message and aborts.
2. The cohort aborts if it receives an abort message from the coordinator, fails, or times out waiting for a commit. If the cohort receives a preCommit message, it sends an ACK message back and awaits a final commit or abort.
3. If, after a cohort member receives a preCommit message, the coordinator fails or times out, the cohort member goes forward with the commit.[8]

Byzantine Agreement

In order to find the active service and also to reduce the searching time of the services, byzantine agreement has been introduced in the transaction processes.

Byzantine Agreementis a classical coordination problem in distributed computing.A basic problem in distributed computing is thatconcurrent process should be able to achieve coordination in spite of some of the faulty behavior.The most stringent requirement for a fault-tolerant protocol is to be resilient to so-called byzantine failures: a faulty process can behave in any arbitrary way, even conspire together with other faulty processes in an attempt to make the protocol work incorrectly. The identity of faulty processes is unknown, reflecting the fact that faults can (and do) happen unpredictably [10].

Besides the usual applications to killing and terrorizing people, protocols that withstand byzantine failures are useful in any situation where it is vital that a system performs correctly in spite of the malfunctioning of some its subparts, regardless of the type of malfunctioning. Examples include aircraft control systems and applications of computer technology to medicine. The agreement problem is a simplified version of a problem that originally arose in the development of on-board aircraft control systems. In this problem, a collection of processors, each with access to a separate altimeter, and some of which may be faulty, attempt to agree on the airplane's altitude. Byzantine agreement algorithms have also been incorporated into the hardware of fault-tolerant multiprocessor systems; there, they are used to help a small collection of processors to carry out identical computations, agreeing on the results at every step. This redundancy allows the processors to tolerate the (Byzantine) failure of one processor. Byzantine agreement algorithms are also useful in processor fault diagnosis, where they can permit a collection of processors to agree on which of their number have failed (and should therefore be replaced or ignored) [10].

IV. IMPLEMENTATION

To implement the proposed system, a ticket reservation system has been used and implemented. A web services business activity is an aggregation of multiple web services atomic transaction. So, here multiple web services have been developed and it has been integrated to a complete business activity. The multiple web services which are integrated into single system are shown in the fig.5

The various atomic activities developed are

1) Airline Ticket Reservation System
2) Bus Ticket Reservation System
3) Hotel Room Reservation System
4) Restaurant Table Reservation System
5) Car Booking System

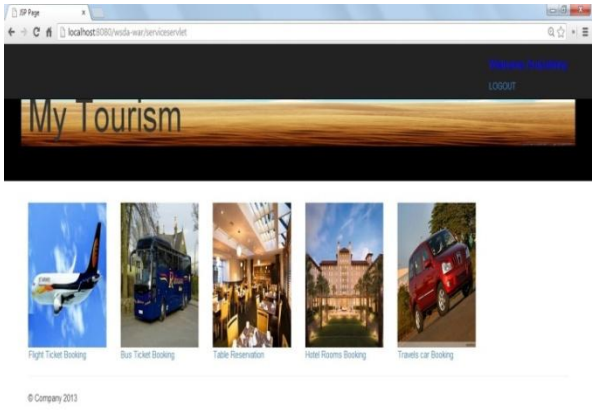

Fig 5: Integration of multiple atomic services

To make use of the web service business activity a customer has to create an account by entering their details and register. Once the customer registers, a key value will be generated and it will be sent to the registered mobile number of the customer. Then the customer has to download an android application and to register it by entering the received key value. Once the customer registers, a formula will be picked from the database and it will be assigned to the customer's id based on the entered key value. So, whenever the customer login to the web service a table of token values will be displayed to the customer. Then the customer has to open the android application, it will prompt to enter the values for the variables. The customer should enter the token values for the prompted parameters. Once he enters the values, the application will calculate the One Time Password (OTP) for the customer by using the assigned formula. Then the customer has to enter the generated OTP to make use of the web services. The fig.6a shows the android application page for the customer to register using the received key value and the fig.6b shows the OTP calculator page once the values has been given to the prompted variables, the OTP will be generated.

Once the customer has been authenticated, the customer will be allowed to make use of available web services to make a reservation. The customer has to reserve the tickets or room or restaurant table and has to make a confirmation on reserving it. After the data has been submitted, it will be sent to the server. Here the three phase commit protocol has been used to make sure that the required server is available by sending a precommit message. Once the data has been submitted the client system sends a commit message to the server first. If there is an acknowledgement for that commit, then client system again sends a precommit message. When there is an acknowledgement, the client system submits the data and request to process the service. If there is any failure in between the process, all the services will be aborted. It will avoid the blocking stage in the process.

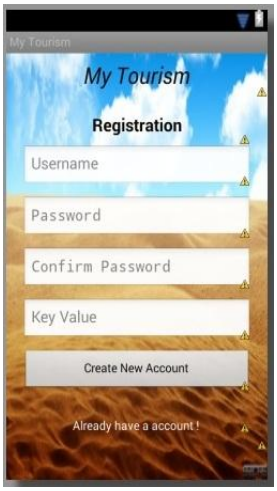

Fig 6a: Android Customer Registration Page

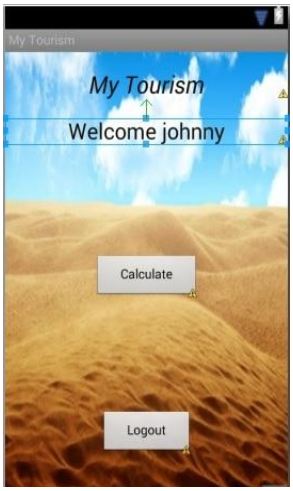

Fig 6b: Android OTP Calculator

When the server system receives the request, it makes use of byzantine agreement. For the interruption problems three replica Web services other than the main Web service has been used. Here, using Byzantine Agreement Algorithm it will check whether Web service is active or not. If the Web service is active, it will respond with acknowledgement and with that Web service the transaction will be done. And if inactive by using Byzantine Agreement Algorithm the coordinator will search for other replica services and which will respond as active the connection will switch to that service using 3PC protocol. The fig.7 shows the confirmation page once all the reservation has been done.

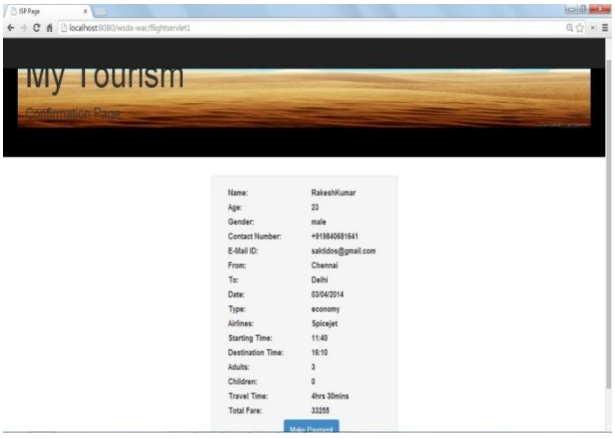

Fig 7: Confirmation Page of the Business Activity

The main idea that has been introduced here is by using the 3PC protocol it is able to reduce the site failures. In 2PC protocol with using AES algorithm if any error occurring then, it will go back to the first phase (Prepared) and will do the Web service searching from beginning. But in 3PC, even though any error occurring it won't go back. Instead, by using Byzantine Agreement Algorithm it will start searching for active Web service. So, it will reduce the searching time.

V. CONCLUSION AND FUTURE WORK

This work addresses the atomic transaction issues in web service business activities. The main idea that has been introduced here is by using the 3PC protocol; the system is able to reduce the failures. In 2PC protocol with using AES algorithm if any error occurring then, it will go back to the first phase (Prepared) and will do the Web service searching from beginning. But in 3PC, even though any error occurs it doesn't roll back to the initial phase. Instead, by using Byzantine Agreement Algorithm it will start searching for active Web service, so it will reduce the searching time of alternative services. This work also includes OTP for the authentication of the client to ensure only the authenticated user is allowed to access the system. The OTP will be generated using an android application during the authentication process to overcome network failure or loss of the registered mobile number. In the future work, theidea of business activity can be extended to an e-governance system.

REFERENCES

[1] Honglei Zhang, Hua Chai, and Wenbing Zhao, "Toward Trustworthy Coordination of Web Services Business Activities," IEEE Transactions on Service Computing, Vol. 6, No. 2, June 2013.

[2] Malaimalavathani. M, and Gowri. R, "A Survey on Semantic Web Service Discovery," 2013 International Conference on Information Communication and Embedded Systems (ICICES).

[3] XinyangFeng, JianjingShen, and Ying Fan, "REST: An Alternative to RPC for Web Services Architecture," ICFIN 2009 First International Conference on Future Information Networks, pp.7-10, Oct 2009.

[4] Honglei Zhang, Hua Chai, and Wenbing Zhao, "Trustworthy Coordination of Web Services Atomic Transactions," IEEE Transactions on Parallel and Distributed Systems, Vol. 23, No. 8, August 2012.

[5] Reihaneh Rabbany Khorasgani, Eleni Stroulia, and Osmar R. Zaiane, "Web Service Matching for RESTful Web Services." 13th IEEE International Symposium on Web Systems Evolution (WSE),2011 pp. 115-124, Sept 2011.

[6] Supriya Kurian and Ramya G. Franklin, "Trustworthy Coordination of Web Services Atomic Transaction for Net Banking," The SIJ Transactions on Computer Science Engineering & its Applications, Vol. 1, No. 1, April 2013.

[7] H. Erven, H. Hicker, C. Huemer, and M.Zapletal, "The Web Services-Business Activity-Initiator (WS-BA-I) Protocol: An Extension to the Web Services-Business Activity Specification," Proc. IEEE Int'l Conf. Web Services, pp. 216-224, July 2007.

[8] http://en.wikipedia.org/wiki/Three-phase_commit_protocol, referred on 5th September 2013.

[9] http://en.wikipedia.org/wiki/One-time_password, referred on 7th October 2013.

[10] http://en.wikipedia.org/wiki/Quantum_Byzantine_agreement, referred on 17th October 2013