



# Implementation of Real Time Messaging Server using Duplex Web Services

Ketan N. Kanere  
M.E Computer Engineering  
(Student)  
Department of Comp  
D. J. Sanghvi College of  
Engineering  
Vile Parle, Mumbai-400056

Hitesh K. Mhatre  
M.E Computer Engineering  
Head of Information  
Technology Department, Pravin  
Patil College of Diploma  
Engineering and  
Technology  
Bhayander(W).

Arjun Jaiswal  
M.E Computer Engineering  
Assistant Professor  
Department of IT  
D. J. Sanghvi College of  
Engineering  
Vile Parle, Mumbai-400056

## ABSTRACT

This architecture supports two-way client-server interactions. The Proxy Framework acts as a service broker between the clients and the CSTA Web services. It dynamically selects the Web service based on client profile, Web service registry and runtime service status. The Proxy framework separates service independent functions, such as session management, event subscriptions, and type conversion, from the service dependent logic, such as message composition and parsing, so that different services can be plugged into the framework easily. The Proxy Framework can be dynamically configured into a Client Proxy to facilitate development of CSTA applications in programming languages, including Visual C++, Visual Basic and Java. Due to the duplex web services interaction created, load will be reduced at the server end, at the same time push back service which is a part of the two-way interaction of the web service will pass the message to other client in the network. Emphasis will be on increasing the data transfer speed, at the same time use of a web service in a chat server will make it work better and faster in distributed environment.

## Keywords

Web service, Messaging

## 1. INTRODUCTION

A Web service is a type of communication between two electronic devices over World Wide Web. A web service is a software function used at runtime provided at a network address over the web or the cloud; it is a service that is "always on" and running as in the concept of utility computing.

W3C Definition of Web Services: - It is defined as a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-executable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

### Duplex Web Services

Two-way Web service is inherently asynchronous, since all messages, such as reply and event, initiated from the server

are asynchronously sent to the client. Two-way full duplex WS interaction involves both client initiated service request and proactive server push for event notification. A duplex contract allows clients and servers to communicate with each other independently so that either can initiate calls to the other. The duplex contract is one of three message patterns available to Windows Communication Foundation (WCF) services. The other two message patterns are one-way and request-reply. A duplex contract consists of two one-way contracts between the client and the server and does not require that the method calls be correlated.

Communication using web services Communication between applications is very important. Web services provide an efficient way of facilitating communication between applications. But there are limitations with web services too. The major limitation with web services is that the communication can happen over HTTP only. A second limitation with web services is that it provides simplex communication and there is no way to have half duplex or full duplex communication using web services. Windows Communication Foundation (WCF) comes to the rescue when we find ourselves not able to achieve what we want to achieve using web services, i.e., other protocols support and even duplex communication. With WCF, we can define our service once and then configure it in such a way that it can be used via HTTP, TCP, IPC, and even Message Queues. We can consume Web Services using server side scripts (ASP.NET), JavaScript Object Notations (JSON), and even REST (Representational State Transfer).

## 2. ONE WAY OPERATION PATTERNS

One-way and notification operations

This one-way operation pattern provides a high level of decoupling between the client and service provider by having two datagram's one for the request and one for the response.

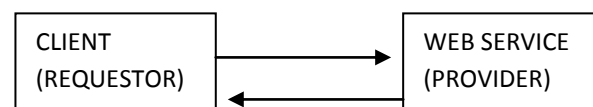


Figure 1: One-way and notification operations



Here, the client creates the correlation ID and sends it to the service via some specified mechanism. This mechanism is accepted by both the parties for communication between the client and the web service. Acceptable mechanisms can be HTTP, SOAP etc. The Request/Reply pattern has two messages request and response defined in a single request/reply operation. However, both messages are sent as two separate transmissions.

### 3. TWO-WAY WEB SERVICES INTERACTION

In a two-way Web services interaction, each Web services end point is both a client and a server; therefore, an interaction requires two appropriate WSDLs, one at each end point. These two WSDL interfaces must be correlated in a certain way to make two-way interaction possible. From the perspective of an end point, two-way full duplex Web services interaction involves both client-initiated service requests and proactive server pushes for event notification.

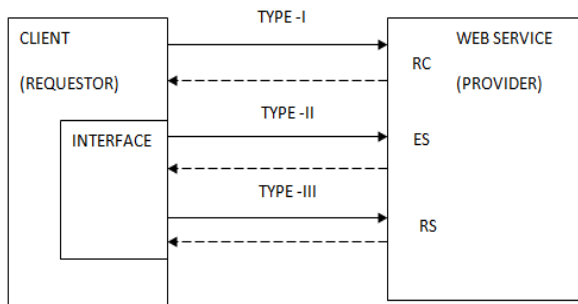


Figure 2: Two-way Interaction patterns between client and server

Type I (RC): Requests initiated by the client, with or without response.

Type II (ES): Event reports from the server, with acknowledgment (solicitation) or without (notification).

Type III (RS): Request initiated by the server, with or without response.

Type I is the conventional one-way WS interaction pattern, where the client makes a one-way WS request to the server.

Type II is the interaction pattern of asynchronous reply and event notification from the server to the client. Type III is the reversal of the Type I interaction pattern, where the server issues the service request to the client.

From service interaction point of view, three types of generic Web service interaction patterns are present. We use a generic client and server to separate two WS endpoints, although each of them is both a client and a server.

### 4. ARCHITECTURE OF REAL TIME MESSAGING SERVER

The architecture of real time messaging server will be consisting of five modules: The architecture of real time messaging server will be consisting of five modules: Text module (TM), White Board module (WBM), File Transfer Module (FTM), Audio Module (AM), Video Module (VM). Each of the module will be using duplex web services for the client-server communication.

Text Module(TM): Text module will be having a text message duplex web service which will pass text messages between the communicating clients. Messages can be transferred to selected users or multiple users based on the selection.

White Board Module (WBM): White board module will have a White Board Duplex Web Service which will allow drawing of images or diagrams such that the drawing or pattern made can be send to multiple users at the same time. This feature will enable users to share their idea regarding any concept or problem discussion.

File Transfer Module (FTM): File Transfer module will have a File Transfer Duplex Web Service which will allow uploading and downloading of files between multiple users. Sharing of files will become easier and faster using this feature.

Audio Module (AM): Audio Module will have an audio conferencing Duplex Web Service which will enable audio conferencing between different nodes connected in the network.

Video Module (VM): Video Module will have a video conferencing Duplex Web Service. This module will be used to do video conferencing between different computers connected in the network.

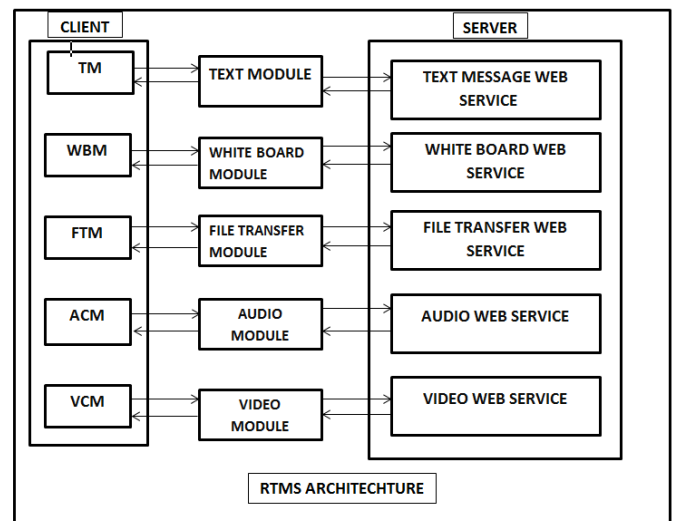


Figure 3: Architecture of Real Time Messaging Server

### 5. IMPLEMENTATION OF REAL TIME MESSAGING SERVER

The GUI is designed in visual studio 2010 and web services are also designed using the contract feature of WCF. WCF stands for Windows Communication Foundation. It consists of protocols and methodologies that are used to transfer message in the form of data communication between multiple machines. One module required for the transfer of the text messages is the text module. In this text module the messages can be sent to multiple users at the same time. At the same instance, you can send the message to the remote client. In the real time messaging server while sending data messages we have a feature of sending message to a single client. Messages can be sent also from one client to other selected clients based on the requirement i.e. multicasting. We can also send one data message from a single client to other multiple clients i.e. broadcasting a message.

The modules are implemented in the system according to the requirement of the communication between the client and the server. For each module, a separate web service will be designed. Each web service will accomplish the task given to



it according to the requirement of the application in the real time messaging server. As discussed in the above section, the architecture has different web service for different modules. The modules designed above are Text module (TM), White Board module (WBM), File Transfer Module (FTM), Audio Module (AM), Video Module (VM).

The Web Services associated with the different modules are as follows: WS-TM (Web Service for Text Module), WS-WBM (Web Service for White Board Module), WS-FTM (Web Service for File Transfer Module), WS-AM (Web Service for Audio Module), WS-VM (Web Service for Video Module).

**WS-TM (Web Service for Text Module):** In this Web Service, text messages are sent between the client and the server. This text messages can be sent from one single client to other remote client, or it can be sent from one single client to other multiple clients at the same time. Messages will be transferred in the form of block from one end to another using Web Services.

**WS-WBM (Web Service for White Board Module):** In this Web Service, the design of the White Board will be done in the XAML file. Different tools and components of drawing is made available in the window for creating a visual message which will be sent from one remote client to other multiple clients at the same time. Users can create a drawing or any figure which they want to share between multiple users. The Web Service will then carry the visual message from a client to other multiple clients. Similar to text message, the drawing can be sent to a single client or multiple clients depending on the requirement of the user.

**WS-FTM (Web Service for File Transfer Module):** In this Web Service, we can share files between multiple clients or single client as per the requirement. The file will be stored in a database maintained at the server. The Web Service will be responsible to fetch the file from the database and share it between multiple clients or single client as per the requirement, and also for storing the file coming from clients which will be stored in the database.

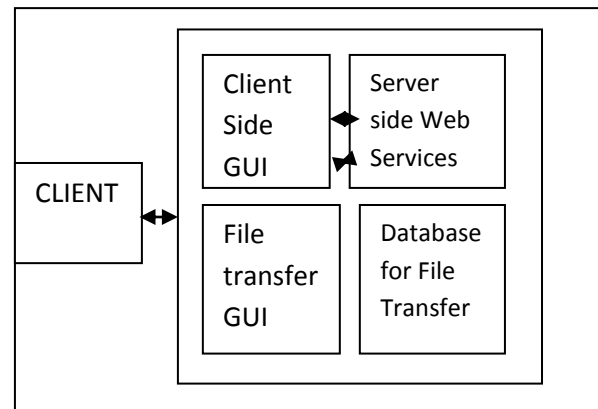
**WS-AM (Web Service for Audio Module):** In this Web Service, we will be able to transmit voice across multiple clients. This will make us possible to do audio conferencing with multiple clients. The voice modulated signals will be sent through the Web Service created.

**WS-VM (Web Service for Video Module):** In this Web service, we will be streaming live video through the webcam between multiple clients. This Web Service will be encoding video stream across clients.

In this implementation the web services for different operations will be active simultaneously. We cannot implement single services for all the users, so different operations are carried out by different web services. For e.g. a duplex web service is designed for white board operation which sends drawing across multiple clients at the same time.

The different Web Services used for the communication between multiple clients will finally give us the real time messaging server application.

The following diagram will give a concise idea about the working of the application:



**Figure 4: Design Layout of the implementation**

In the above diagram, connection or interfacing is shown between multiple clients. The client on the remote side will be having a machine on which the different modules which are implemented are just called from the server. On the server side, the following modules are implemented: Client side GUI, Server side web services, File transfer GUI, Database for file transfer.

**Client side GUI:** The user interface which is to be shown on client screen is designed in this module. The design will consist of controls which are to be displayed on the user screen like login and account window. The layout of the window is designed in xaml. This controls will enable client to communicate with other clients. Transfer of data, voice and video as well as file transfer is implemented using different modules. While coming up with a GUI design, the following factors are kept in mind.[1]The user interface should be simple and easy to understand.[2]Different type of media used for communication i.e. data, voice and video should be separately designed and associated with each other.[3]Video quality should be maintained between different users, so that it does not become lag between intervals in case of multiple users.[4]Size of the chat screen and color perception should be properly managed to keep it more descriptive.[5]Session should be present with a user name and password assigned to each and every user.[6]A list of users should be displayed where all the users are present on one screen.

**Server Side web services:** This module will consist of all the web services which are designed for different modules. These web services are interfaced with the GUI for different features available in the application.

**File Transfer GUI:** This GUI is created for the uploading and sharing of the files which are to be uploaded at the server and shared between multiple clients. The file which is uploaded by any client is stored in the database at the server and then passed to the Duplex web service which will send the file to multiple clients.

**Database for Multiple Clients:** The database is maintained at the server. This database is linked with the file transfer GUI and the web service associated with the file transfer. The file coming from the File transfer GUI is stored in the database at the server end.



## 6. CONCLUSION AND FUTURE SCOPE

The Real Time Messaging Server is implemented using Duplex Web Services. The messaging server created for communication will be taking the messages from the client which needs to be sent to other clients. Now this message is not sent by the user or the server to other machines. This message is sent via a web service from the client to the server. A pushback service will be present at the remote end which will send the message back to the client as reply or to other multiple clients as a reply. The proxy framework is present between the client and the server. The messages will be sent at a faster rate as compared to normal messaging applications because of the duplex web services used instead of normal data passing through socket,code etc.

The application can be implemented on mobile devices for the communication between mobile devices in future. The additional overhead can be reduced with the help of duplex web services in mobile devices. The application can also be used on a large scale by providing high end security and added features. We have take into consideration factors such as compatibility, interoperabilty and scalability if the applcation limit is expanded.

## 7. REFERENCES

- [1] OASIS. (2007, April.), “Web Services Business Activity(WSBusinessActivity) Version 1.1,” Available: <http://docs.oasisopen.org/ws-tx/wstx-wsba-1.1-spec-os.pdf> [Aug. 23, 2013].
- [2] N. Srirajun, P. Bhattarakosol, P. Tantasanawong, and S. Han, “Trustable Web Services with Dynamic Confidence Time Interval,” in Proc. of the 4th Int. Conf. on New Trends in Information Science and Service Science (NISS), 2010, pp.11-13.
- [3] WSDL 2.0: A Pragmatic Analysis and an Interoperation Framework, 2008 SYS-CONMedia,
- [4] W. Dong, "Multi-agent Test Environment for BPEL-based Web Service Composition", the IEEE Int. Conf. on Cybernetics and Intelligent Systems, 2008, pp.855-860.
- [5] C. -H. Liu, S. -L. Chen, and X. -Y. Li, "A WS-BPEL Based Structural Testing Approach for Web Service Compositions," the IEEE International Symposium on Service-Oriented System Engineering, 2008, pp.135-141.
- [6] “A Framework for Nested Web Services”, International Journal of Engineering Research & Technology, Volume/Issue: Vol. 3 - Issue 3 (March - 2014) e-ISSN: 2278-0181
- [7] N. Srirajun, P. Bhattarakosol, P. Tantasanawong, and S. Han, “A Trustable Software with A Dynamic Loop Control Mechanism,” in Proc. of the 5th Int. Conf. on Future Information Technology, 2010, pp. 20-24.
- [8] W3C. (2007, June), “Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language,” Available: <http://www.w3.org/TR/wsd120/> [Aug. 10, 2013].
- [9] N. Parimala, and A. Saini, “Web Service with Criteria: Extending WSDL,” in Proc. of the 6th Int. Conf. on Digital Information Management, 2011, pp. 205-210.
- [10] Real Time Messaging Server Using Duplex Web Services International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970) Volume-4 Number-3 Issue-16 September-2014