


WhatsApp-Engineering Inside-1

 **Suraj Kumar** Jan 20, 2019 · 5 min read

Real Time messaging are now an essential part of our day to day life, without them our day is incomplete. But, have you ever thought how ‘WhatsApp’ or other Real Time messaging app works?



In this article we are going to explore the high level engineering and system architecture behind whatsapp or any general real time messaging app.

Before diving deep let’s understand ‘How communication works’?

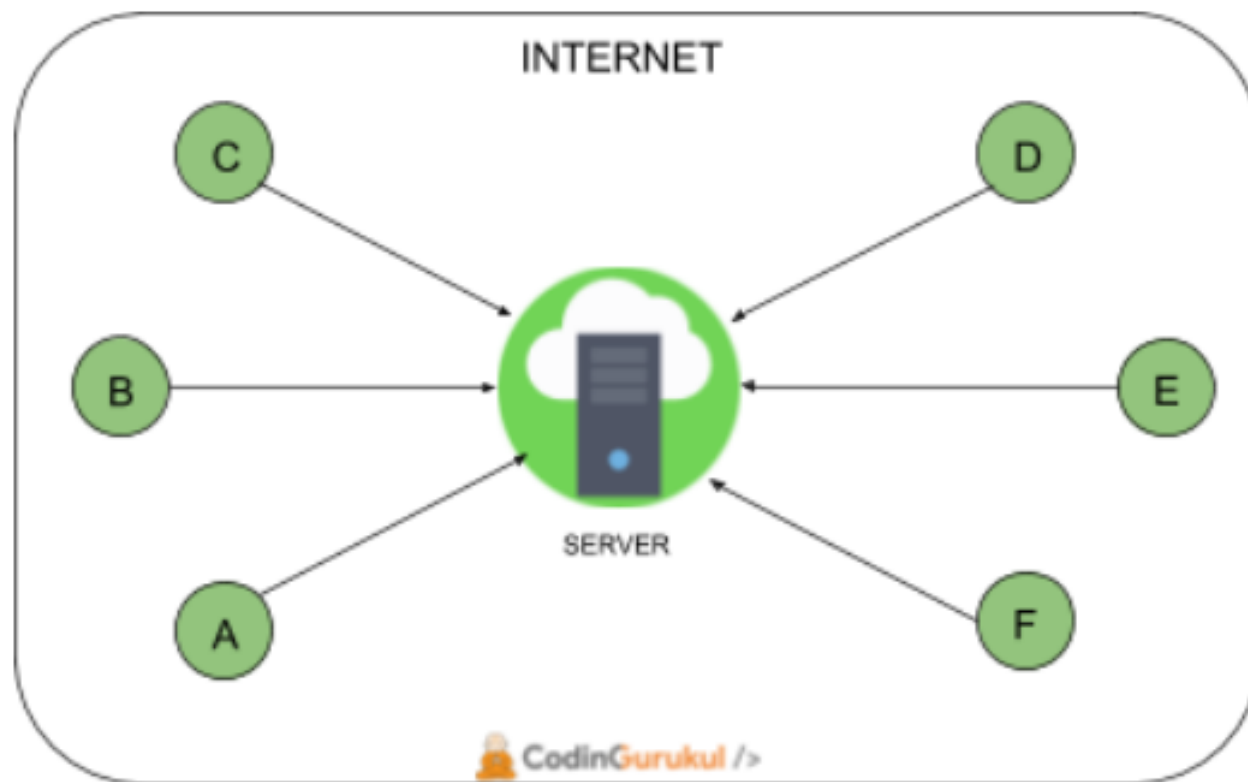


General Communication Architecture

identity) and they exchange messages with each other over a network, in this case it is INTERNET.

But, what if the network is very large and number of clients are in millions or billions?

In a very large network it is very difficult to know the address of each and every client, In this case to make this system more robust and highly available we need a component between the clients called “SERVER”. The task of this server is to coordinate between all the clients connected to it.



Client-Server based communication architecture.

After the introduction of server. All clients instead of making connection with each other they make connection with the server.

In this scenario when a client(A) want to send message to other client(D), it first sends the message to server and the server knows the address of other client(D), then it forwards the message to other client(D) and vice versa.

This is the overview of the communication architecture. Lets design Actual System design of a real time messaging system.

But before designing any product it is really important to understand the some requirements like:

1. **User base:** *It is really important to understand at what scale the application is to be used.*

2. **Features required**

So, let's list some of the features needed to be incorporated in whatsapp:

- 1). **Text messaging.**

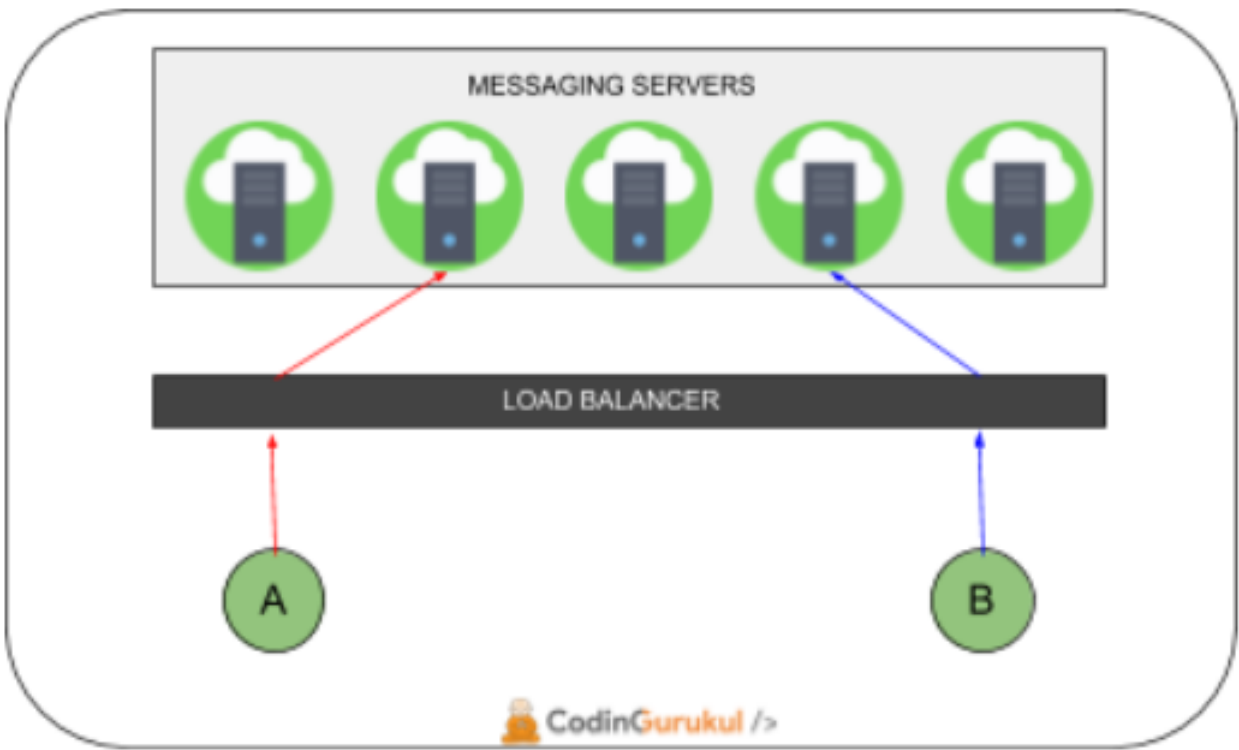
4). Message Encryption

5). Telephony services(Audio/Video)

Let’s start designing the system based on the application requirement.

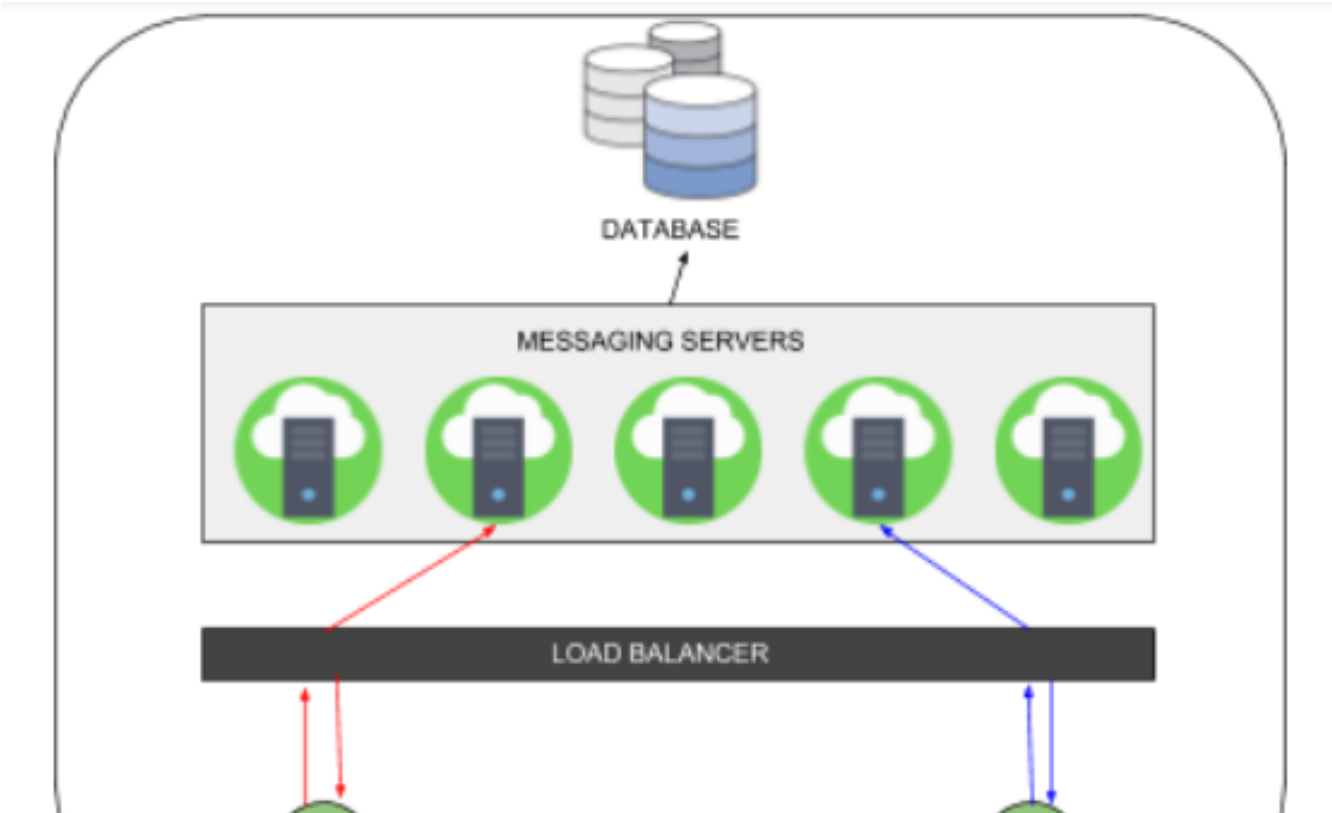
Based upon the user base we need multiple servers to handle this much traffic, so instead of one server we place multiple servers.

But the question is that, to which server the client will connect as there are multiple servers and the client cannot connect randomly to any server. To overcome this issue we introduced a load balancer between the client and the server.



Mutiple Servers and load balancer architecture

After implementing multiple servers and load balancer, our system architecture is capable of handling a large user base. Now when a client want to connect to the server, the connection request first hits the load balancer and then the load balancer redirects the connection to a server based on various parameters like load on individual servers etc.



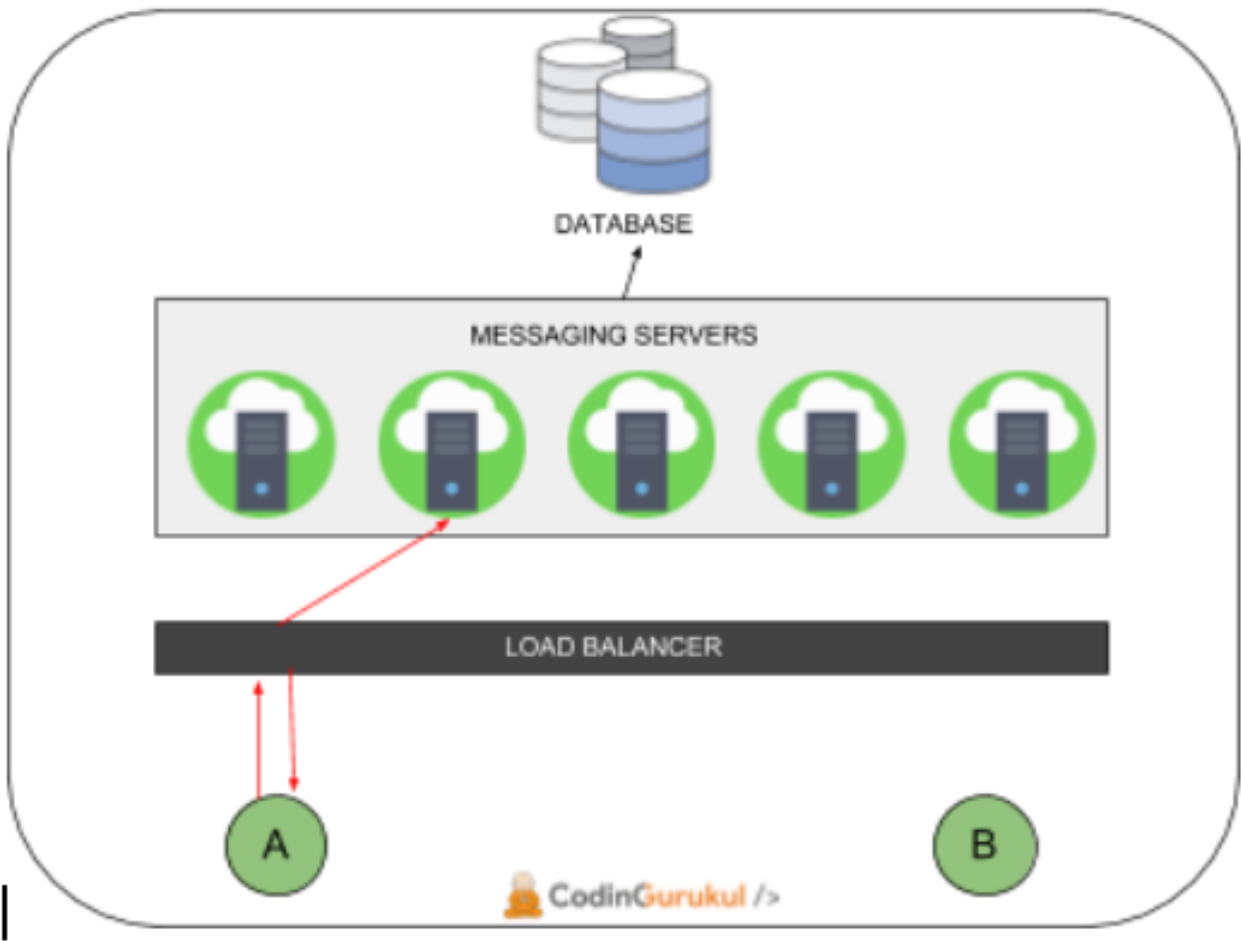
But our application also needs some storage mechanism to save some arbitrary state or data, to fulfill this requirement we also added database which is accessible to all the servers.

But, What kind of connection is used?

Generally, this kind of system uses a DUPLEX Connection or Bidirectional Connection. As the message can also be generated from the server, so bidirectional communication is required

Before moving ahead lets understand different connectivity scenarios and how the application works.

1. When Sender is connected to server but not the receiver.



Sender is connected to server but receiver is not.

In this case when the receiver is not connected to the server, the message is stored in the database and when the receiver connects to the server, the message is fetched from the database and forwarded to the receiver.

2 . When Sender is not Connected to the Server.

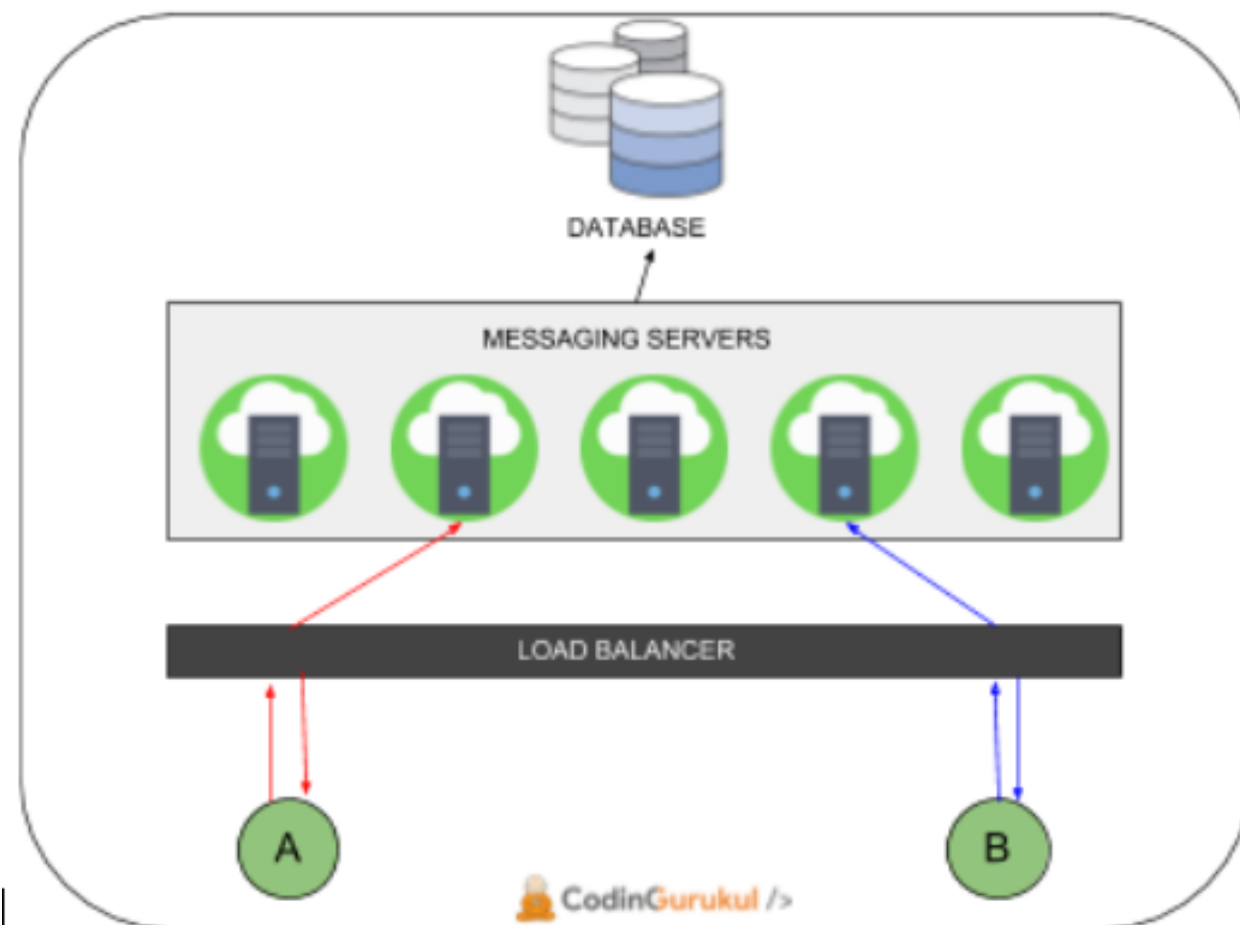




Client not connected and message is saved in device storage.

In this case when sender is not connected to the server, the message sent by the sender is saved in the device local storage (it may be SQLite or anything else based on platform). And when the sender goes online or connects to the server the message is fetched from the local storage and sent to the server.

3 . When both clients are connected to the server:



Both Clients are connected to the server.

In this case when both the clients are connected to the server, the sender sends the message and the server forwards that message to the receiver without storing the message to the database or device local storage.

This is thousand feet overview of whatsapp system architecture, In next article we will take a close and technical look inside the messaging server, and i assure it will be more interesting and knowledgeable. Make sure to follow, clap and share to get the notification of the next article.

Cheers !!! Keep Exploring, Keep Coding with Coding Gurukul.

[Click Here to read the Part 2 of this article.](#)



Get the Medium app

