Requirements of WhatsApp's Design

Learn about the functional and non-functional requirements for a chat application like WhatsApp.

We'll cover the following

Requirements

Our design of the WhatsApp messenger should meet the following requirements.

Functional requirements

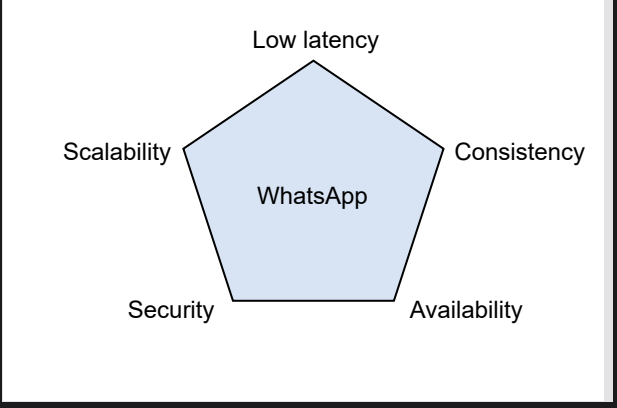
- **Conversation**: The system should support one-on-one and group conversations between users.
- **Acknowledgment**: The system should support message delivery acknowledgment, such as sent, delivered, and read.
- **Sharing**: The system should support sharing of media files, such as images, videos, and audio.

- **Chat storage**: The system must support the persistent storage of chat messages when a user is offline until the successful delivery of messages.
- **Push notifications**: The system should be able to notify offline users of new messages once their status becomes online.

> **Note:** The term “offline” typically refers to users not currently connected to the internet. When these users reconnect, the system will automatically send push notifications to alert them of any new messages. On the other hand, the term “online” indicates that the user is actively connected to the internet and can receive real-time messages.

Non-functional requirements

- **Low latency**: Users should be able to receive messages with low latency.
- **Consistency**: Messages should be delivered in the order they were sent. Moreover, users must see the same chat history on all of their devices.
- **Availability**: The system should be highly available. However, the availability can be compromised in the interest of consistency.
- **Security**: The system must be secure via end-to-end encryption. The end-to-end encryption ensures that only the two communicating parties can see the content of messages. Nobody in between, not even WhatsApp, should have access.
- **Scalability:** The system should be highly scalable to support an ever-increasing number of users and messages per day.



Resource estimation

WhatsApp is the most used messaging application across the globe. According to WhatsApp, it supports more than two billion users around the world who share more than 100 billion messages each day. We need to estimate the storage capacity, bandwidth, and number of servers to support such an enormous number of users and messages.

Storage estimation

As there are more than 100 billion messages shared per day over WhatsApp, let's estimate the storage capacity based on this figure. Assume that each message takes 100 Bytes on average. Moreover, the WhatsApp servers keep the messages only for 30 days. So, if the user doesn't get connected to the server within these days, the messages will be permanently deleted from the server.

$$100\ billion/day * 100\ Bytes = 10\ TB/day$$

For 30 days, the storage capacity would become the following:

$$30 * 10\ TB/day = 300\ TB/month$$

Besides chat messages, we also have media files, which take more than 100 Bytes per message. Moreover, we also have to store users' information and messages' metadata—for example, time stamp, ID, and so on. Along the way, we also need encryption and decryption for secure communication. Therefore, we would also need to store encryption keys and relevant metadata. So, to be precise, we need more than 300 TB per month, but for the sake of simplicity, let's stick to the number 300 TB per month.

The total storage required by WhatsApp in a month

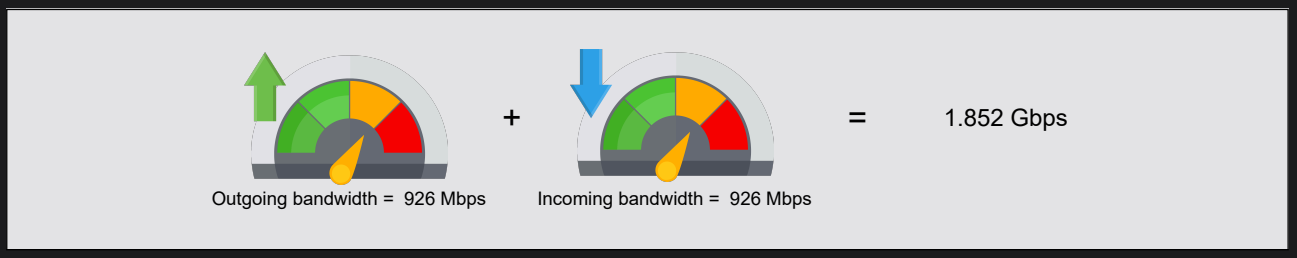
Bandwidth estimation

According to the storage capacity estimation, our service will get 10TB of data each day, giving us a bandwidth of 926 Mb/s.

$$10\ TB/86400sec \approx 926\ Mb/s$$

> **Note:** To keep our design simple, we've ignored the media content (images, videos, documents, and so on). So, the number 926 might seem low.

We also require an equal amount of outgoing bandwidth as the same message from the sender would need to be delivered to the receiver.



The total bandwidth required by WhatsApp

High-level Estimates

Type	Estimates
Total messages per day	100 billion
Storage required per day	10 TB

Storage for 30 days	300 TB
Incoming data per second	926 Mb/s
Outgoing data per second	926 Mb/s

Number of servers estimation

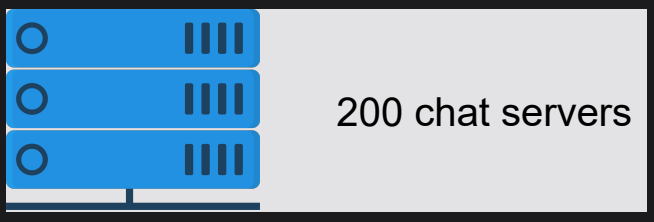
WhatsApp handles around 10 million connections on a single server, which seems quite high for a server. However, it's possible by extensive performance engineering. We'll need to know all the in-depth details of a system, such as a server's kernel, networking library, infrastructure configuration, and so on.

> **Note:** We can often optimize a general-purpose server for special tasks by careful performance engineering of the full software stack.

Let's move to the estimation of the number of servers:

$$No.\ of\ servers = \\ Total\ connections\ per\ day / No.\ of\ connections\ per\ server = \\ 2\ billion / 10\ million = 200\ servers$$

So, according to the above estimates, we require 200 chat servers.



Number of chat servers required for WhatsApp

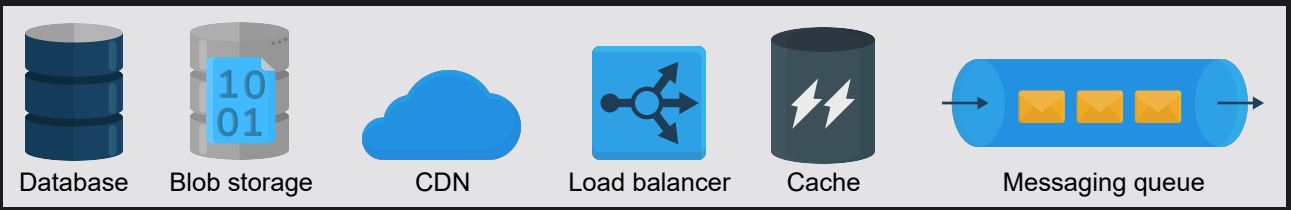
Try it out

Let's analyze how the number of messages per day affects the storage and bandwidth requirements. For this purpose, we can change values in the following table to compute the estimates:

Number of users per day (in billions)	2
Number of messages per day (in billions)	100
Size of each message (in bytes)	100
Number of connections a server can handle (in millions)	10
Storage estimation per day (in TB)	*f* 10
Incoming and Outgoing bandwidth (Mb/s)	*f* 926.4
Number of chat servers required	*f* 200

Building blocks we will use

The design of WhatsApp utilizes the following building blocks that have also been discussed in the initial chapters:



The building blocks required to design WhatsApp

- **Databases** are required to store users' and groups' metadata.
- **Blob storage** is used to store multimedia content shared in messages.
- A **CDN** is used to effectively deliver multimedia content that's frequently shared.
- A **load balancer** distributes incoming requests among the pool of available servers.

- **Caches** are required to keep frequently accessed data used by various services.
- A **messaging queue** is used to temporarily keep messages in a queue on a database when a user is offline.

Evaluate your abilities in designing WhatsApp using the AI widget provided at the end of this lesson. Beyond the mentioned building blocks, additional components contribute to creating a comprehensive design. These include:

- Message service
- Group message service
- Websocket manager
- Mnesia database
- Redis cluster

Provide your solution considering the following details:

- Which entity will process the messages of the two users intending to communicate with each other?
- How are the group messages handled?
- Where are the messages stored when a user is offline?
- Where will the media files be kept for frequent access?
- What protocol(s) is(are) required to transmit messages with low latency?
- How will different services be decoupled?

Want to know the correct answer?

Design WhatsApp system using the building blocks and components discussed above.

H1 H2 H3 | B I | | | | | |

Provide your answer here!