 master ▾


...

[system-design-interview](#) / [problems](#) / Build\_Messaging\_System.md



wuyichen24 Update Build\_Messaging\_System.md

 History

 1 contributor

# Build Messaging System

## Real-life examples

- Facebook Chat
- Whatapp
- Slack
- Discord

## Requirements clarification

- **Functional requirements**
  - Messaging: One user can send a message to another user or a group of users.
  - Status: Shows online/offline statuses of users.
  - Images and videos uploading: User can upload images and videos in addition to text messages.
  - Push notifications: Offline users can receive a push notification when there are new messages.
  - Read receipt: Senders can get a receipt when receivers read messages they sent.
- **Non-functional requirements**

- Low latency (real-time messaging).
- High consistency (users should be able to see the same chat history on all their devices).
- High availability is desirable (base on CAP theorem).

## Data model definition

---

- Schema

- Table 1: User
  - Description
    - Store user accounts
  - Columns

Column Name	Column Type	PK	Description
UserId	int	PK	The user ID.
Name	string		The name of the user.
LastActive	datetime		The timestamp of when the user is online (support online/offline status).

- Table 2: Message
  - Description
    - Store messages and their metadata.
  - Columns

Column Name	Column Type	PK	Description
MessageId	int	PK	The message ID.
SenderId	int		The user ID of the sender.
ConversationId	int		Identify the message belongs to which conversation.
Text	string		The text message of the message
MediaUrl	string		The url to access the media files attached to the message.

- Table 3: Conversation
  - Description

- Store conversation information
- Columns

Column Name	Column Type	PK	Description
ConversationId	int	PK	The conversation ID.
name	string		The name of the conversation (like channel name in Slack).

- Table 4: ConversationUsers

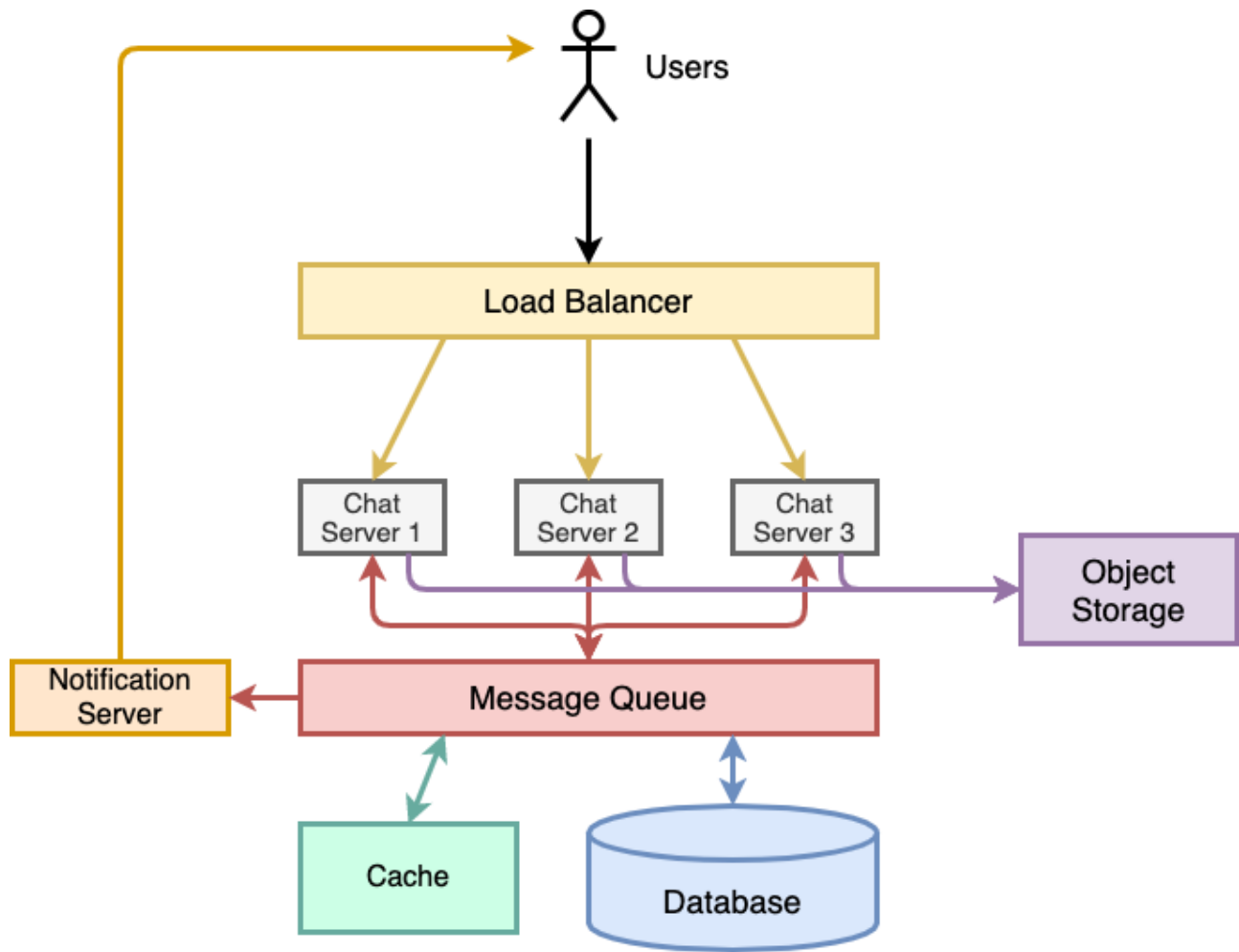
- Description
  - Store the relationship about which user is a part of the conversation.
- Columns

Column Name	Column Type	PK	Description
ConversationId	int		The conversation ID.
UserId	int		The user ID of each user belongs to this conversation.

## High-level design

☰ 103 lines (95 sloc) | 4.66 KB

...



- **Chat server**
  - Orchestrate all the communications between users (The direct connection between 2 users is not reliable).
- **Message Queue**
  - Handle the communication between chat servers.
  - Each chat server will have a channel for receiving messages for that chat.
- **Object Storage**
  - Store media files.

## Detailed design

- **Users**
  - Considerations
    - Consideration 1: Users receive new messages
      - Ideas
        - It cannot be server initiated, it must be client initiated.
      - Solutions for receiving messages
        - Solution 1: HTTP polling (poor)

- Description: Users can repeatedly ask the server if there are any new messages for them.
- Cons: Users will send lots of unnecessary to the server.
- Solution 2: HTTP long polling (acceptable)
  - Description
    - A user send one request to the server.
    - The server will hold the request, wait and response only if there is a new message for the user.
  - Cons: The server need to maintain lots of open connections.
- Solution 3: WebSocket (good)
  - Pros
    - Full duplex: Users and the server can send data at the same time.
    - Connections can keep open for the duration of the session.
- Solution 4: BOSH - Bidirectional-streams Over Synchronous HTTP

## Key points

---

- Use WebSocket for clients to get new messages.

## References

---

- <https://slack.engineering/how-slack-built-shared-channels/>
- <https://www.infoq.com/presentations/slack-scalability-2018/>
- [https://www.infoq.com/presentations/slack-infrastructure/?utm\\_source=youtube&utm\\_medium=link&utm\\_campaign=qcontalks](https://www.infoq.com/presentations/slack-infrastructure/?utm_source=youtube&utm_medium=link&utm_campaign=qcontalks)
- [https://www.youtube.com/watch?v=zKPNUMkwOJE&ab\\_channel=TusharRoy-CodingMadeSimple](https://www.youtube.com/watch?v=zKPNUMkwOJE&ab_channel=TusharRoy-CodingMadeSimple)
- [https://www.youtube.com/watch?v=uzeJb7ZjoQ4&ab\\_channel=Exponent](https://www.youtube.com/watch?v=uzeJb7ZjoQ4&ab_channel=Exponent)
- [https://www.youtube.com/watch?v=vvhC64hQZMk&ab\\_channel=GauravSen](https://www.youtube.com/watch?v=vvhC64hQZMk&ab_channel=GauravSen)
- <http://www.erlang-factory.com/upload/presentations/31/EugeneLetuchy-ErlangatFacebook.pdf>
- [https://www.facebook.com/note.php?note\\_id=14218138919&id=9445547199&index=0](https://www.facebook.com/note.php?note_id=14218138919&id=9445547199&index=0)