Social Messaging Service
Stefano Borghi | August 23, 2022

Client

Cache

User Service

Messaging Service

UI

Streaming Service

Web Socket

HTTP Streaming

Server

Load Balancer

User Service

Cache

Users DB

Load Balancer

Messages Service

Cache

Messages DB

CDN

Media Storage

Streaming Service
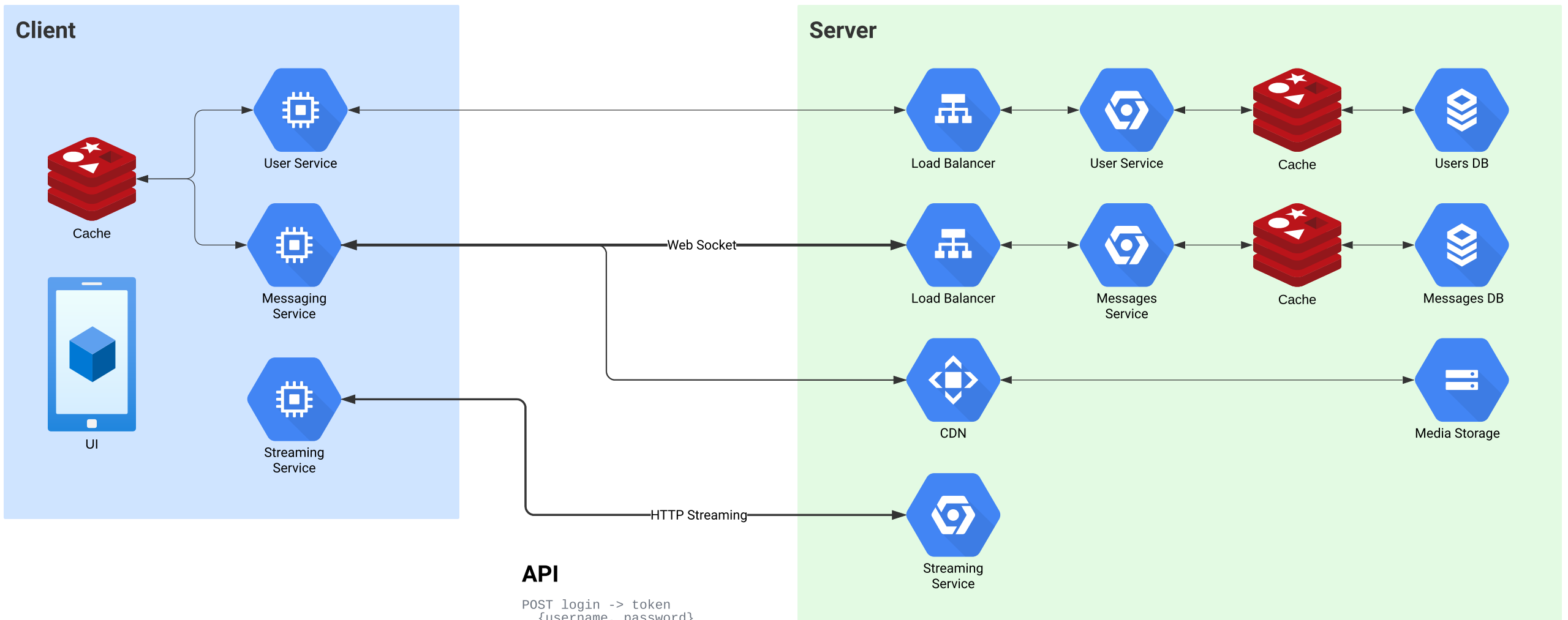
User Service: allows auth to the user and provides list of available users and their connection status

Message Service: establishes socket connection with the server to allow real-time async message exchange. It also allows to manage groups.

Streaming Service: allows to perform audio/video calls among clients

API

```
POST login -> token
  {username, password}

POST logout
  {token}

GET userStatus -> status
  {token, userId}

POST createGroup -> group
  {token, name, admin, partecipants}

DELETE deleteGroup
  {token, groupId}

PUT updateGroup
  {token, groupId, name, admin, partecipants}

POST sendMessage
  {token, text, recipient, attachments}

DELETE deleteMessage
  {token, messageId}

PUT updateMessage
  {token, messageId, text, recipient}

POST callGroup
  {token, groupId, withVideo}

POST endCall
  {token, groupId}
```

User
+ id
+ name
+ email
+ phone_number

Message
+ id
+ text
+ attachments
+ sender
+ recepient
+ group_id
+ sent_at

Group
+ id
+ name
+ admin
+ partecipants
+ created_at