Google Docs MLD / Collaborative editing
Google Docs Sheets
API Gateway in Go

Google Docs :
Q: What user can do ?
A : Single user doc editing
Q: Multiple same doc editing (Collaborative editing)

Q: What is DAU ?
A : 1 Million

Q : How many concurrent user can edit the same doc at a time ?
A : 100 users

Functional Req :
Single user editing : Single user can create/read/update /delete on a document
Collaborative editing : Multiple user can edit same document at a time.

Non-Functional Req :
1. Availability over Consistency
2. Eventual Consistency is accepted
3. Low Latency : Latency - User can see his/her own edits immediately.

Non-focused area
Authentication & Authorization
Access writes

Back of the envelope calculations :
Peak QPS = (1 Million / 60)
Peak QPS = QPS * 2

Read : write -> 10 : 1

QPS -> Read Query: Write Query

Storage unit : Each doc roughly consume 100KB
Estimate for 5 years.

API End points:

1. Create a new document:
POST /v0/documents/{id}
Request Body :
{
title : "First document"
}

Response :
{
documentid : "123",
title : "First document"
}

2. Read an existing doc:
GET /v0/documents/{documentid}

Response :
{
documentid : "",
content : "",
created_at : "",
updated_at : ""
}

3. Edit an existing document:
PUT /v0/documents/{documentid}
Request Body :
{
title : "",
content : ""
}

Response :
{
documentid : "",
title : "",
content : ""
}

4. Delete an existing doc:
DELETE /v0/documents/{documentid}
Response : 200 ok
{
documentid : ""
}

5. Editing within a document
POST /v0/documents/{documentid}/operation
Request Body :
{
operations : [
{
operational : "",
type : "insert",
position : 45,
text : "Hello"
}
]
}

Response : 200 ok
{
documentid : "",
status : "success"
}

6. Collaborative editing :

1. Using HTTP Polling.

Request-Response Arch -> Polling

Not a good approach.

Pros : Easy to implement

Cons :
1. Server will be bombarded with too much request.
2. latency will increase (slow)
3. Consume a lot of BW.

2. Long Polling :

pros :
1. Server load is slightly reduced.

3. Using Web Sockets.

REST APIs vs WEB Sockets (Real time)

HTTP : POST, GET, PUT, DELETE etc.

Why do we need Web Sockets in Google docs ?

6. Collaborative editing
WS /ws/document/{documentid}/collaborate
Response : 200
{
"message" : "Connection Established"
}

WS /ws/document/{documentid}
Request Body :
{
type : "operation",
operation : {
operational : "",
position : 45,
text : "Hello"
}
}

Broadcast response to every connected user
Response :
{
type : "operation",
userid : "user 1",
timestamp : "",
operation : {
type : "insert",
position : 45,
text : "Hello"
}
}

pros :
→ Latency will be low
→ Server is not bombarded with request.

Draw the High level architecture:

Document -> file (image, Video, document)

Amazon S3 Buckets

Link Sharing in Collaborative editing :
https://www.amazon.com/docs/text/documentid/...

Hierarchical file System :
Windows, MacOS

Home -> Coding -> HLD -> HLD_S_pdf
/home/coding/HLD/HLD_S_pdf

1. Hierarchical folder structure is not a good approach for file sharing

→ Too long URL
→ We dont want to expose entire folder hierarchy to all users

→ Flatten up the URL
→ generate a unique URL (unique ID) for each file /page

How to generate this unique ID ?
→ Hashing
→ In short as possible

→ Google Docs -> alphanumeric
→ Base62 or Base64 algorithm generate MD5 hash of document ID

URL shortener -> base 0 approach

How to handle conflict ??

s1 : insert "Hello" at pos 5
s2 : insert "Bye" at pos 5

Basic Approach :
Timestamp based approach :

Operational Transformation (OT)
-> use by google docs.

Simple version

WhatsAPP/Discord/Telegram

peer-to-peer communication

Client-Server Architecture using HTTP

Again, client can poll for MSG

Using Web Sockets

Table : Communicate
sender, Receiver, Msg, timestamp

Case 1 : Both clients are online

Case 2 : Client 2 is offline

Zookeeper

client -> server
client1 -> server1
client2 -> server3

Group chat

Zookeeper can hold one more mapping of G -> servers
g1 -> s1,s2,s3 -> s1, s2, s3

API GATEWAY

LB, DNS, CDNs, Web tier, DB tier, cache, MQ

Load Balancer vs API gateway

LB distributes traffic to multiple instances of same server.

Whereas API Gateway decide which Microservice to call.

What API gateway do ?

1. Transfer API calls to correct Microservice

2. It can be used for Authentication

3. We can implement Rate limiting logic to API Gateway.

How API gateway decide which MS to call ??

Service Discovery

www.google.com

Top level DNS