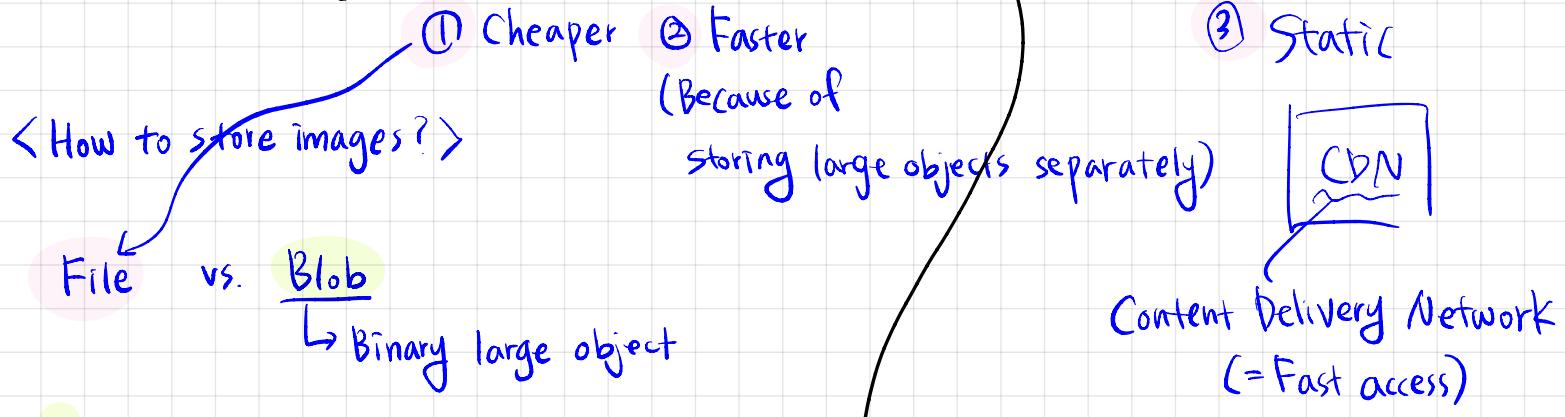


[System Design: TINDER as a microservice architecture]

TINDER ARCHITECTURE

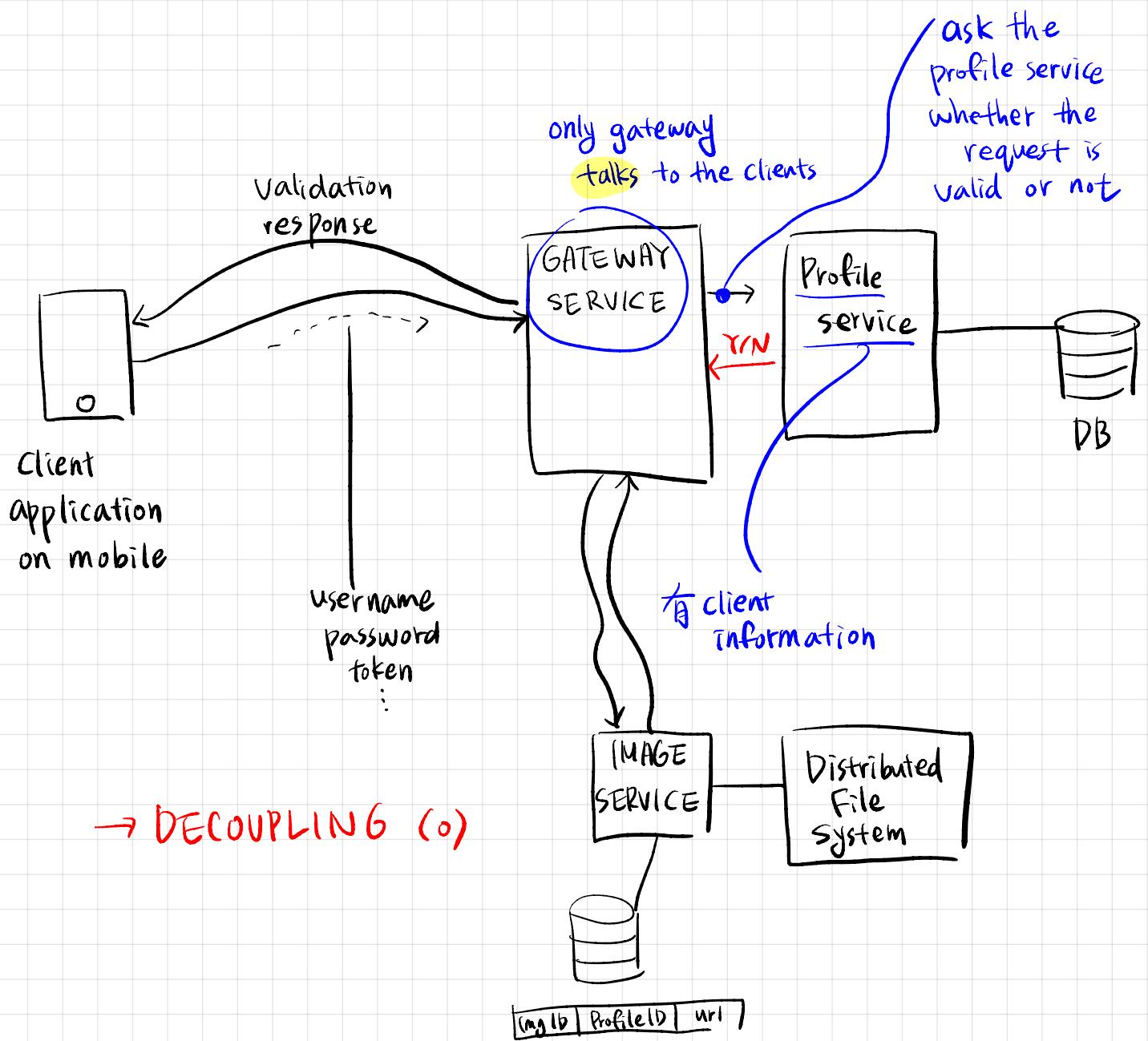
- ① Store Profiles (Images) - 5 images per user
- ② Recommend Matches (Number of active users)
- ③ Note matches $\times 10^3$
- ④ Direct Messaging



- ① Mutability - are you going to change an image? X (File replace O)
- ② Transaction - atomic operation on image X
- ③ Indexes (Search) - search on image X
- ④ Access control

ProfileId | ImageId | FileUrl

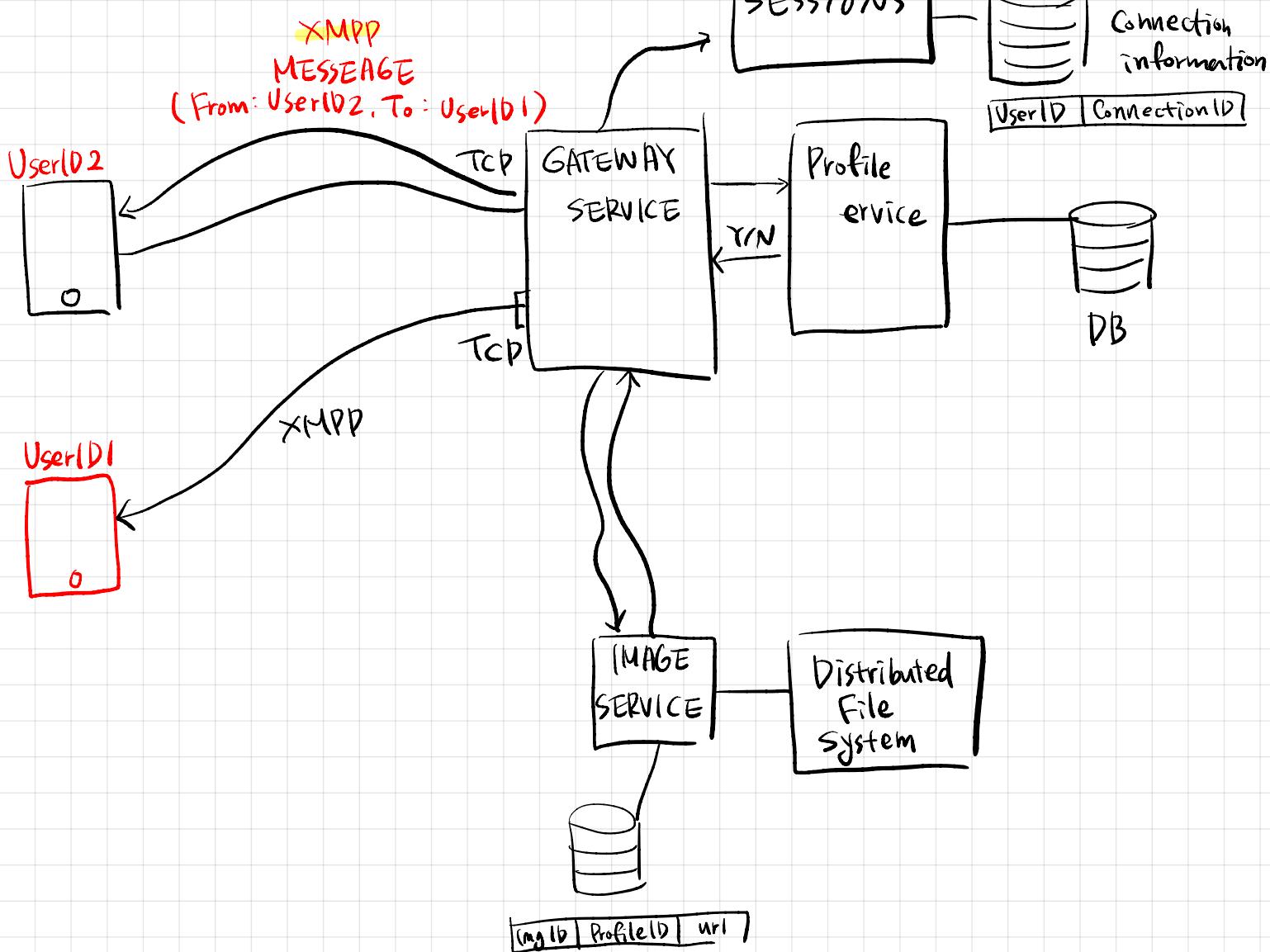
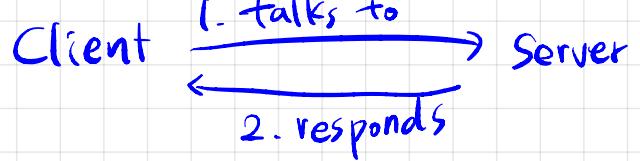
<STORE PROFILES (+USER AUTHENTICATION)>



< DIRECT MESSAGING >

* Decouple
as much as you can

< HTTP >



Client-Server Communication Protocol (
← has to check every 5 mins: "Is there a message received?")
vs.
→ need pull from server

Peer-Peer Protocol (Everyone is equal!)

vs.

XMP

vs.

HTTP

<RECOMMENDATION>

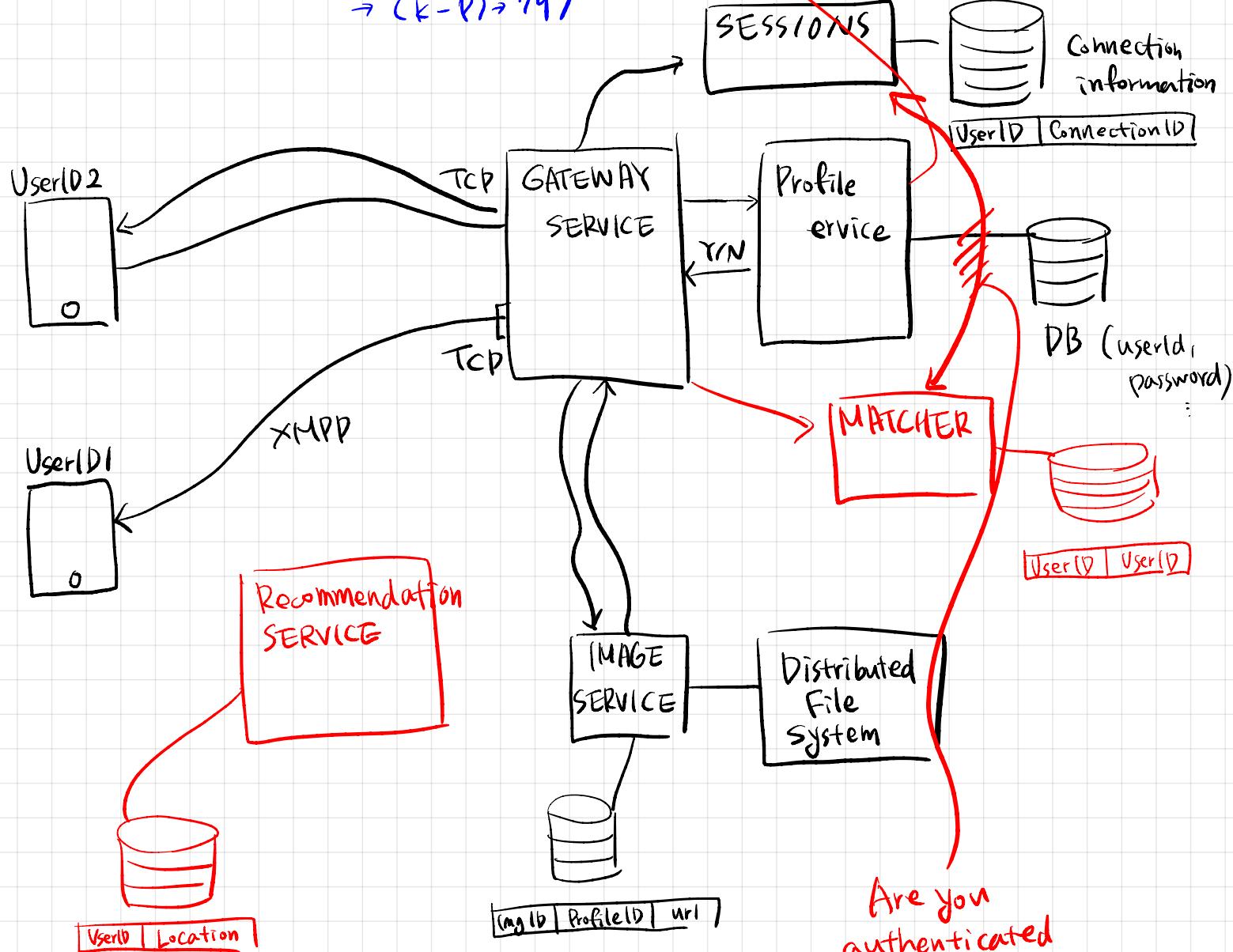
- options
- ① Distributed DB
(Cassandra, Amazon Dynamo)
 - ② Sharding (Horizontal Partitioning)

partition data according to a characteristic (ex. name)



age | gender | location

name → (A - J) → 36
→ (K - P) → 79



Are you authenticated to send a message to the following person?