Agenda.

→ Storage layer Concepts.

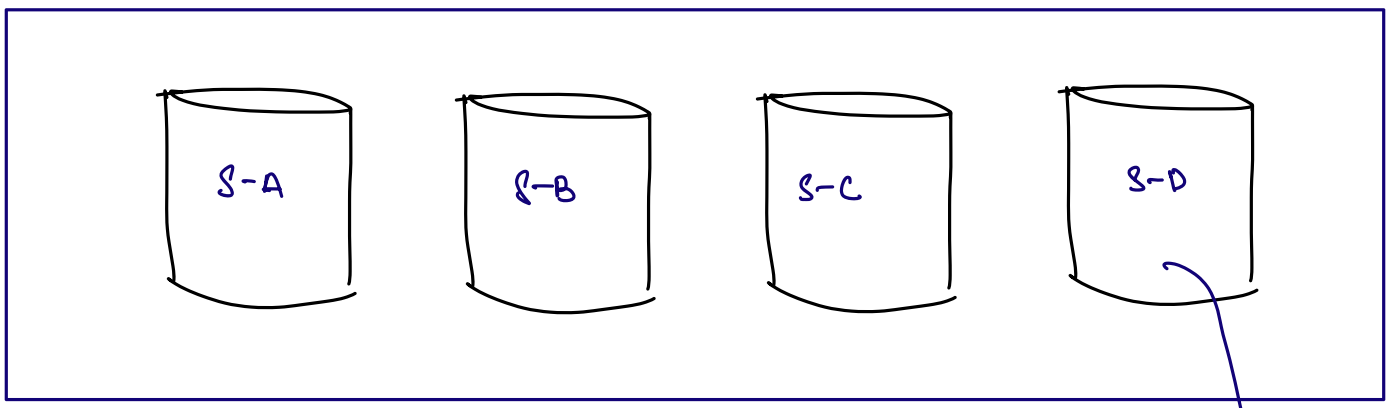
- → SHARDING
- → REPLICATION.

→ CAP Theorem

→ PACELC Theorem

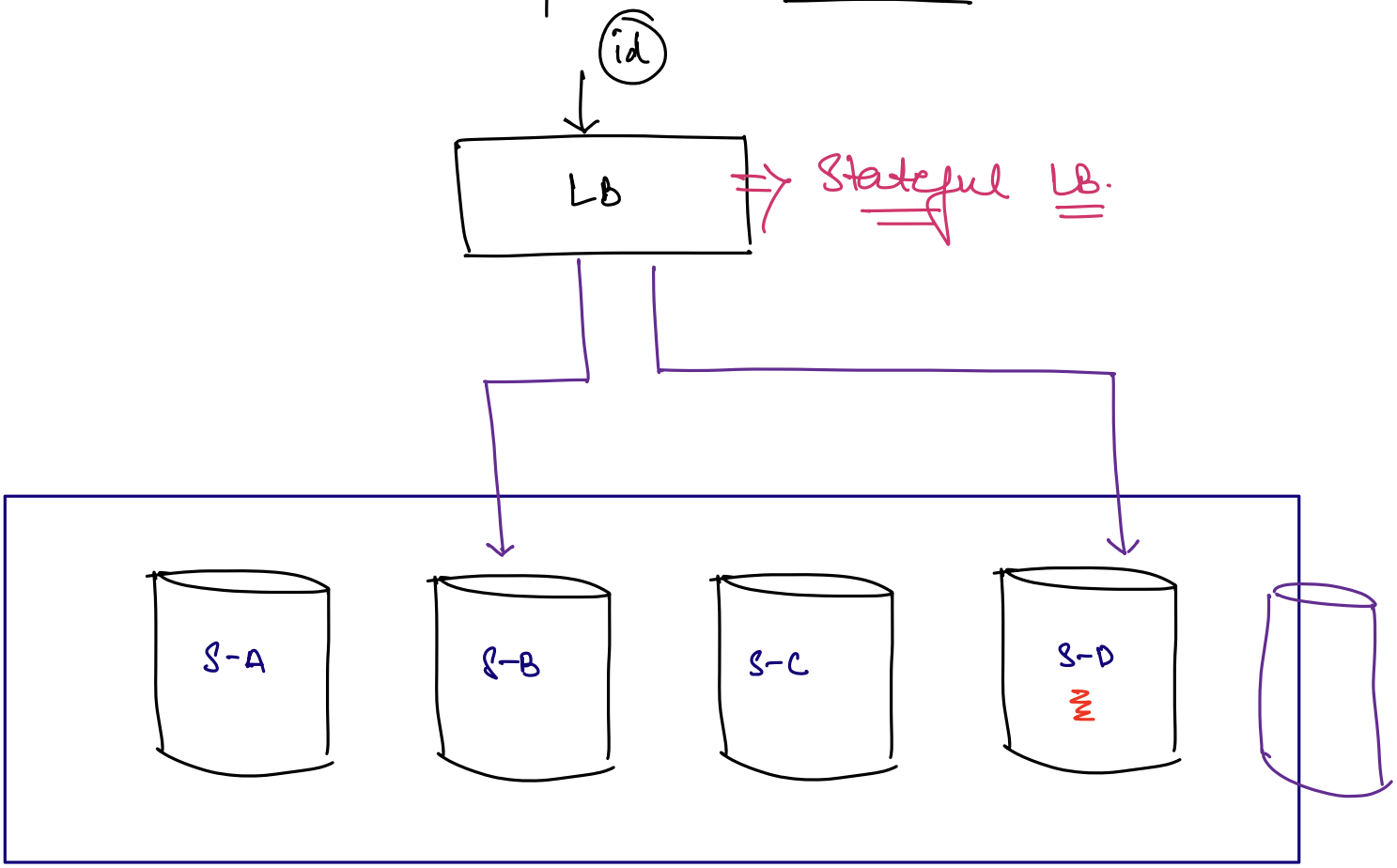
SHARDING.

⇒ Idea of distributing data across multiple machines so that we'll be store huge amount of data.

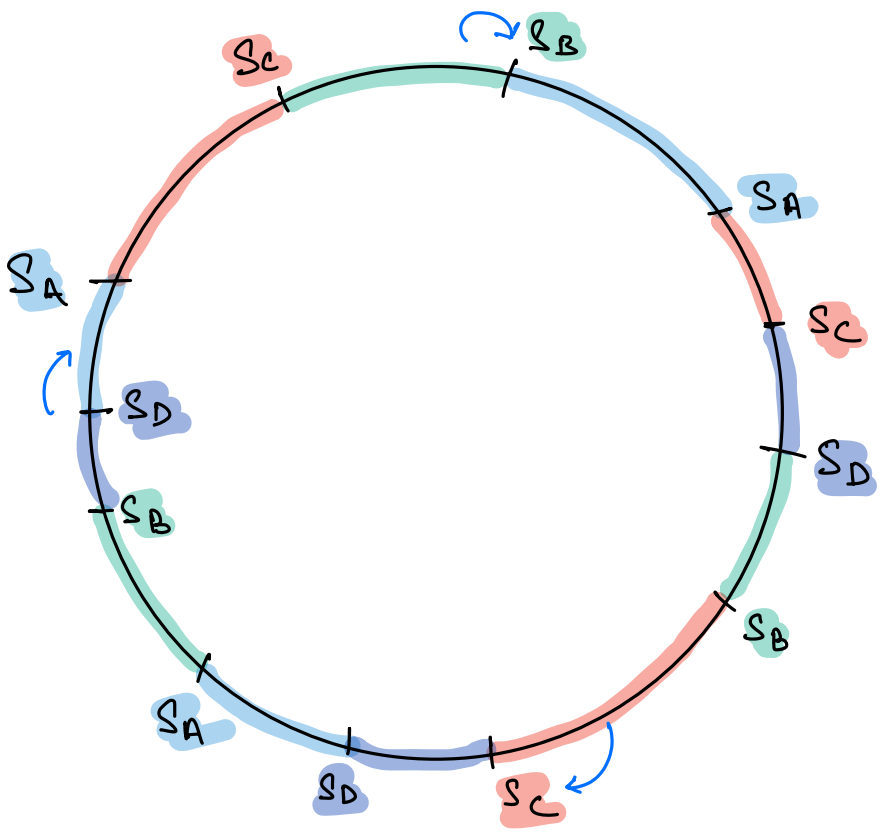


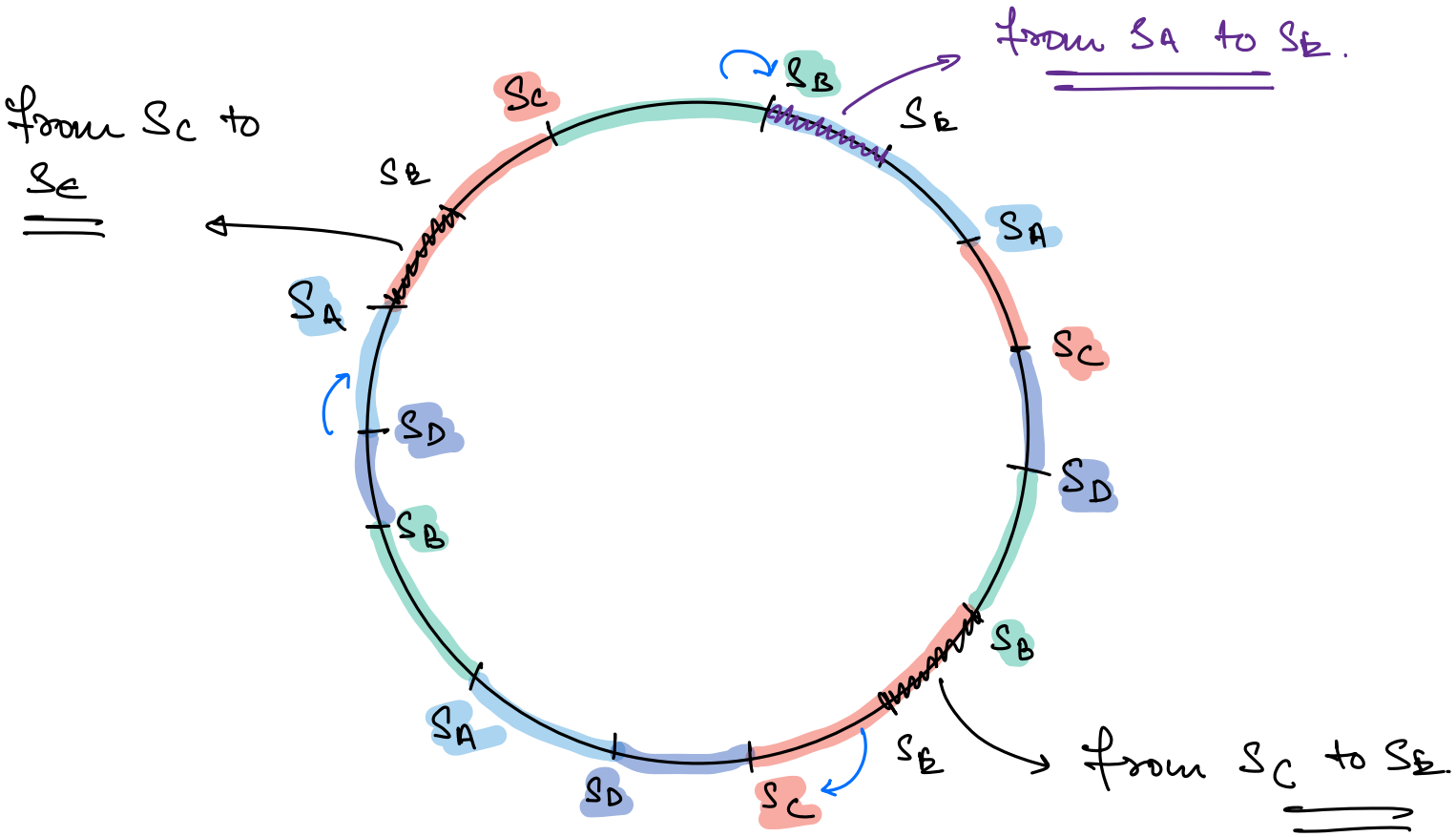
SHARDING ≡ Shards which contains mutually exclusive & collectively exhaustive data.

SHARDING ≠ REPLICATION.



⇒ Consistent Hashing Algorithm





Note: While doing data migration, our DB cluster might be unavailable for some time.

Objective

⇒ Most frequently accessed queries should be intra shard & NOT inter shard.

intra shard → Single m/c.

inter shard → multiple m/c.

⇒ Sharding key

Amazon's product DB.

↳ 1000 TB

Q:- Fetch all the products of a particular category.

I) product_id ⇒ Inter Shard Query.

II) category_id ⇒ Intra Shard Query.

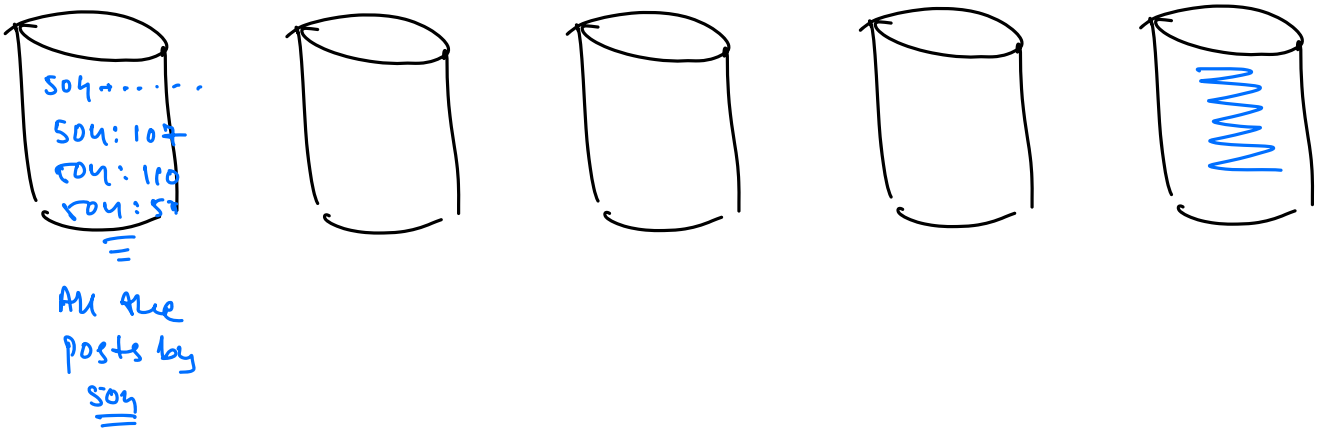
III) City_id

IV)

⇒ fb Newsfeed.

- users
- user_friends
- posts.

Sharding key : user_id.



getProfilePage (user_id) ⇒ intra

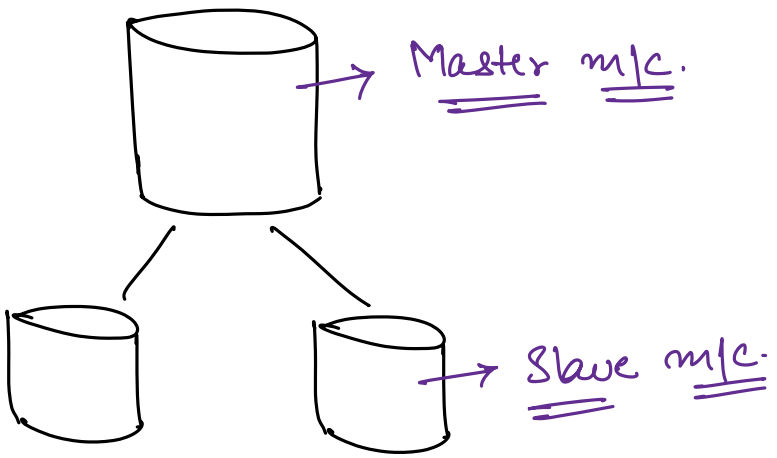
getNewsfeed (user_id) ⇒ inter

↳ To optimize such queries we store pre-processed information at cache layer for faster retrieval.

REPLICATION.

↳ Creating copies of data.

⇒ Master - Slave Architecture



→ Avoid SPOF.

→ Read Replicas.

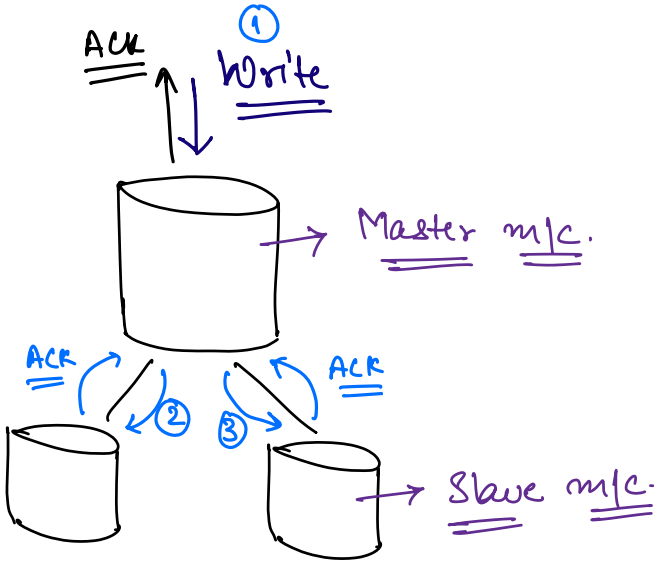
↳ Distribute on read traffic.

Write Operation

1) Sync.

→ Strong consistency

→ Write latency ↑

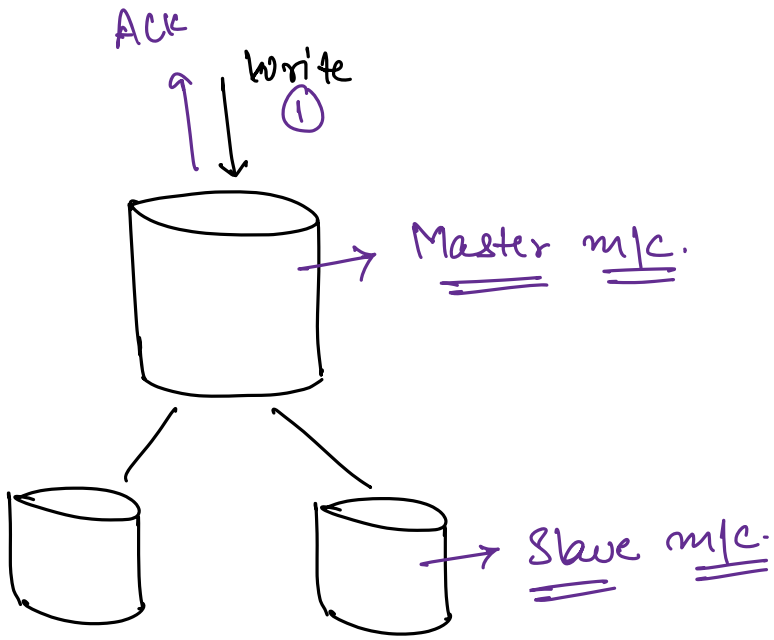


II) Async

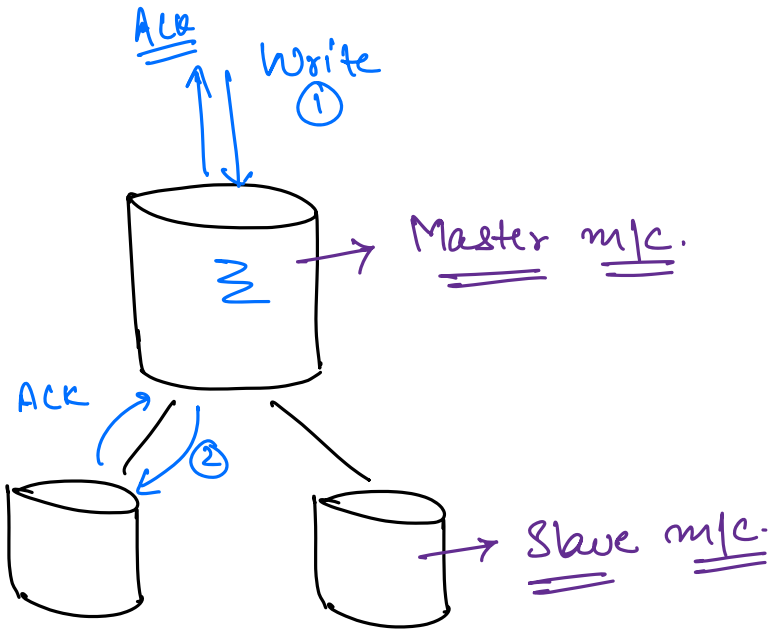
→ faster Write operations.

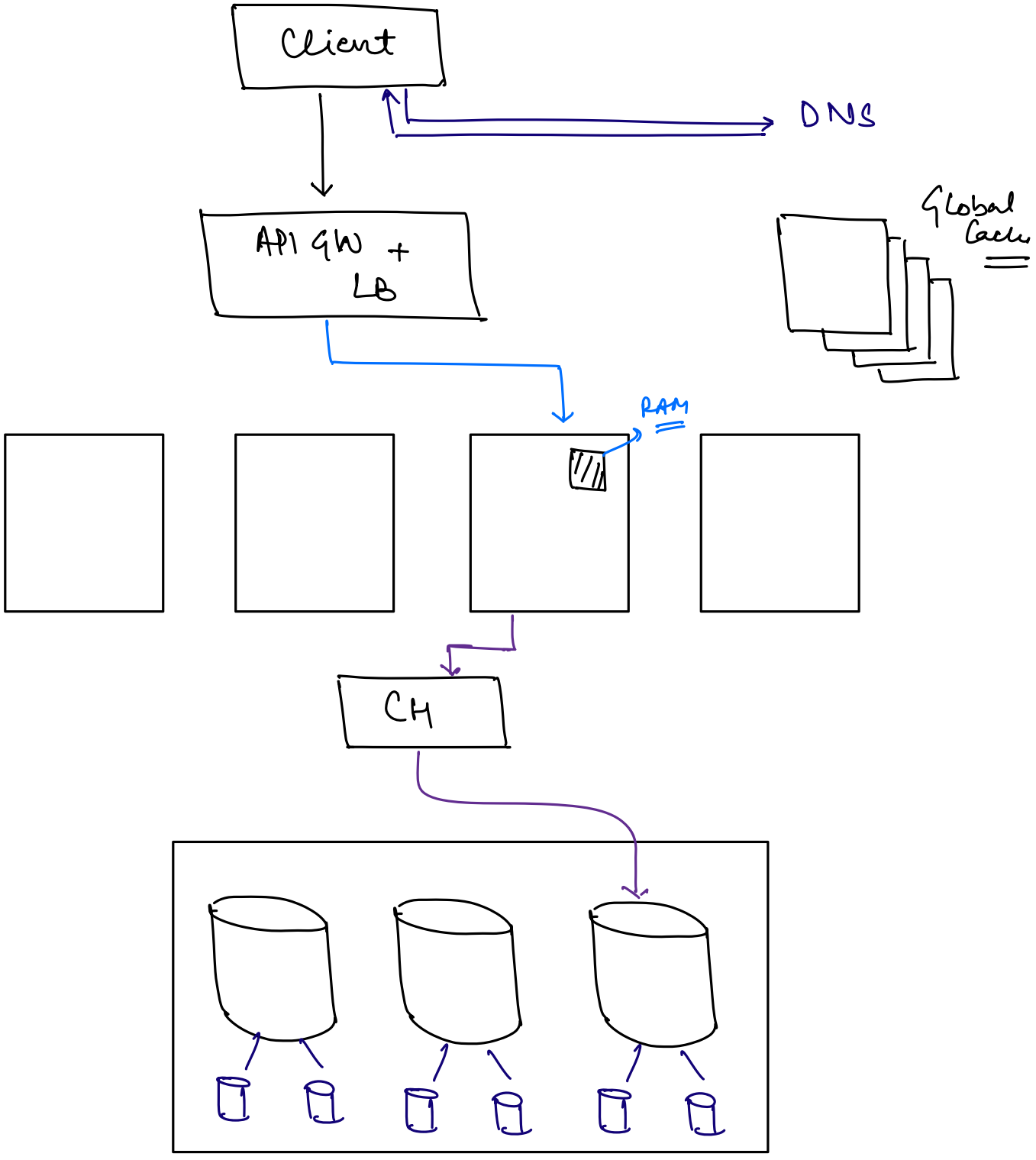
→ Eventual Consistency

→ Write operations might get lost if master m/c goes down before sync.



III) Quorum Based Approach.




Client
DNS
API GW + LB
Global Cache
RAM
CH

CAP Theorem.

Consistency

⇒ Everytime we read, we get the data of latest write.

⇒ All the machines / replicas contains the same data.

Availability

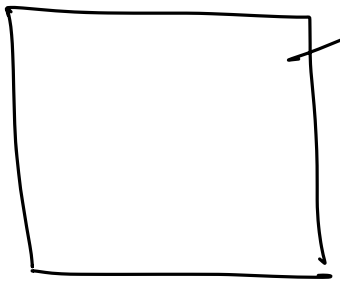
⇒ If we send a query then system must be available to give you a response.

Partition Tolerance

↳ In distributed System, we can't avoid partition 100%.

⇒ System should be able to handle network partitions.

In distributed system, we can only achieve (2) out of (3).



No partition
Consistency ✓
Availability ✓

⇒ for distributed System

Partition Tolerance is there.

Consistency (VS) Availability.