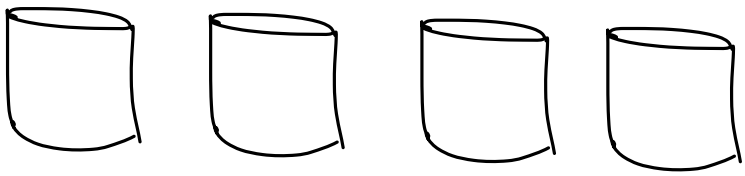
Agenda.

→ Sharding (vs) Replication.

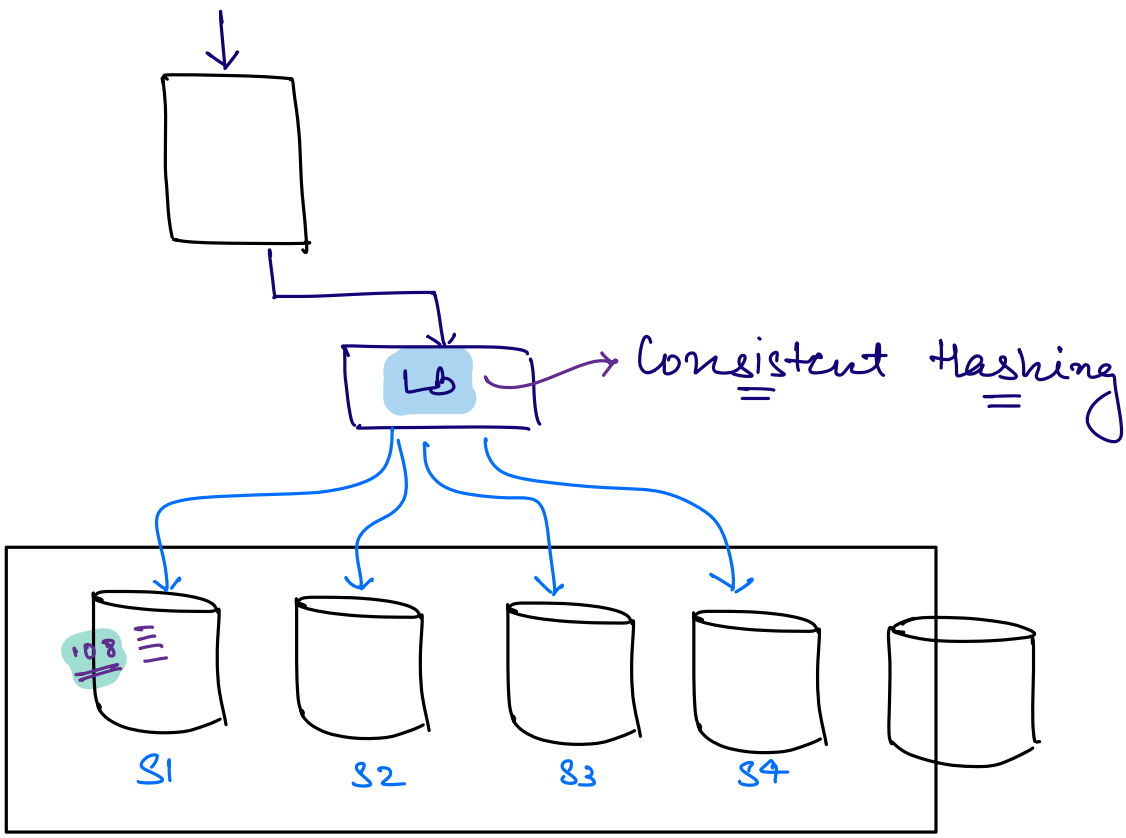
→ CAP Theorem.

→ PACELC Theorem.

SHARDING.



⇒ Distributing the data across multiple m/c when we huge amount of data to store.



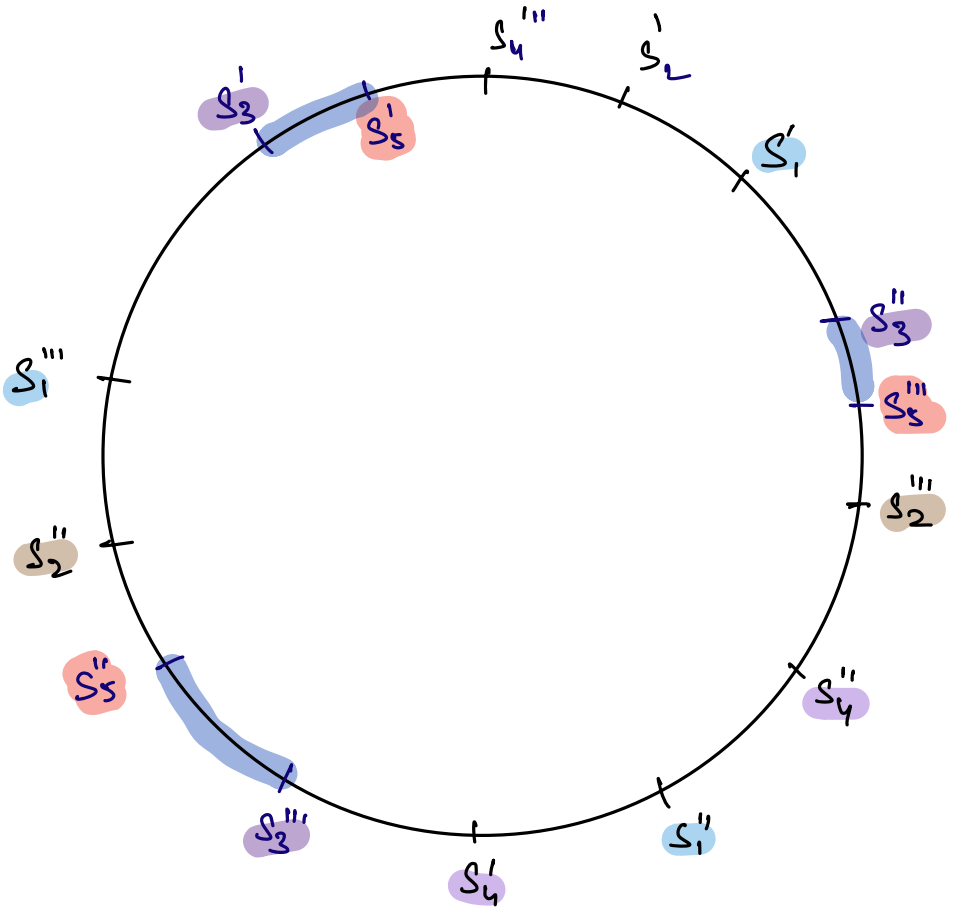
⇒ The key or id basis on which we distribute the data is called Sharding key.

Sharding = Shards which are

Mutually Exclusive

+

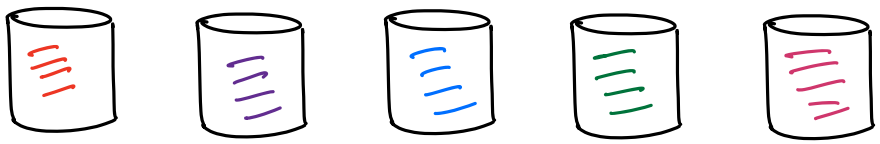
Collectively exhaustive.



S_5

Ex: Amazon's Product DB.

↓ product_id



Q: Get all the products of a given category.

→ productId → Inter Shard

→ CategoryId ⇒ Intra Shard.

→ SellerId

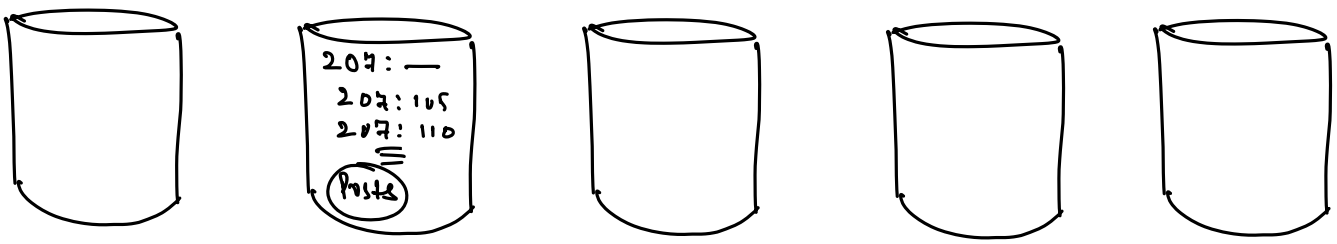
Ex fb Newsfeed.

users

friends

user_posts.

↓ userid ≡ Sharding Key.



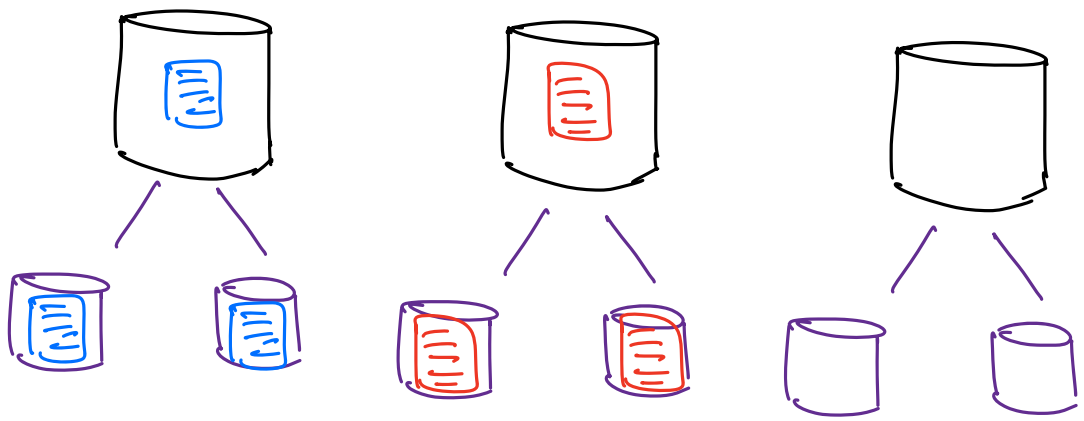
getProfilePage(user_id) ⇒ Intra Shard

getfriendList(user_id) ⇒ Intra Shard

getNewsfeed(user_id) ⇒ Inter Shard
(fan Out request)

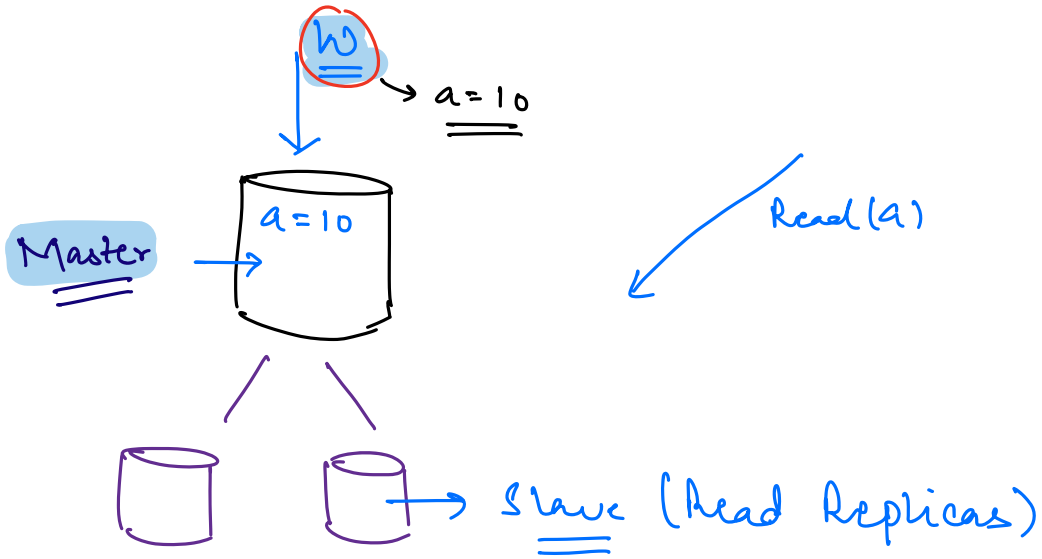
⇒ Caching

Replication.



⇒ Master Slave Architecture.

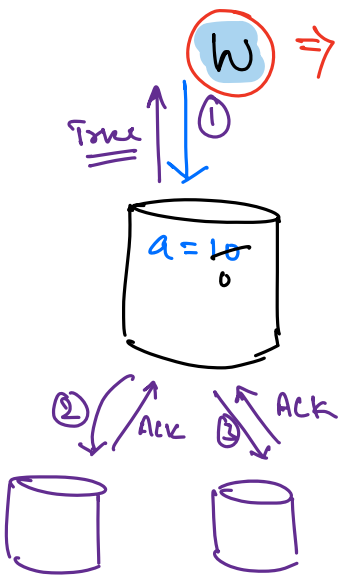
- Solves SPOF
- Read Replicas.



⇒ Consistency Issue.

[# of reads >>> # of writes]

1) Synchronous.

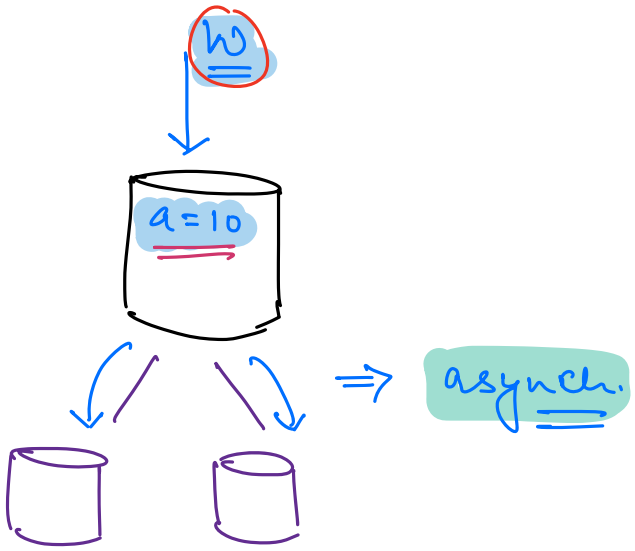


⇒ Atomic

⇒ Strongly Consistent System

⇒ Write latency ↑↑

2) ASynchronous.

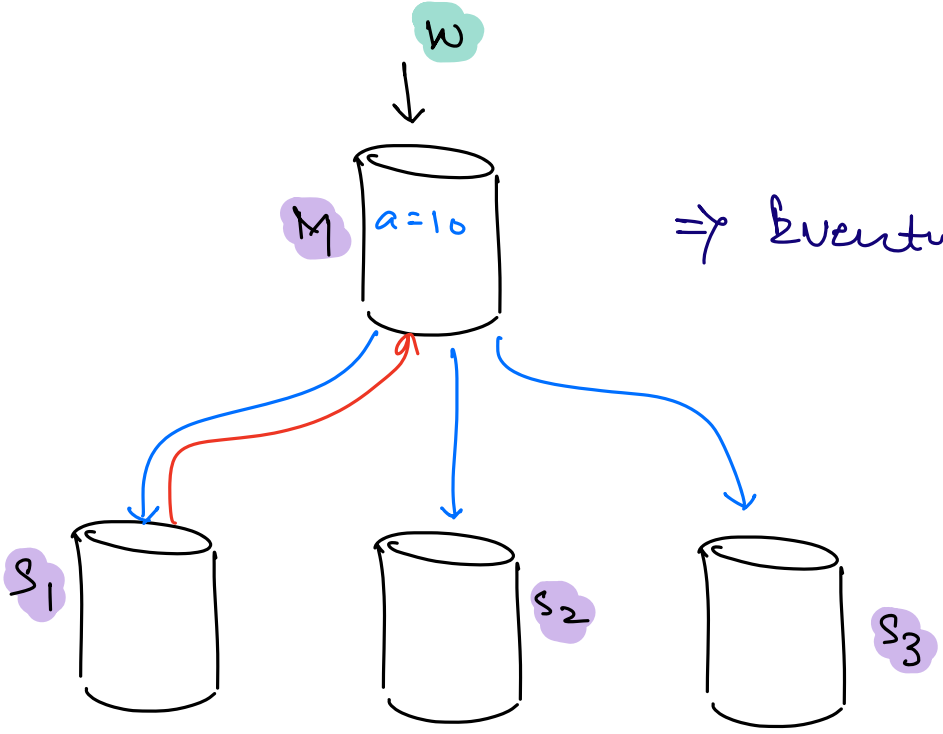


Every 10 mins

⇒ Eventual Consistency.

⇒ Write latency ↓↓

3) Quorum Based Approach.



⇒ Eventual Consistency.

Leader Election Algo.

CAP Theorem.

Consistency

→ No stale reads

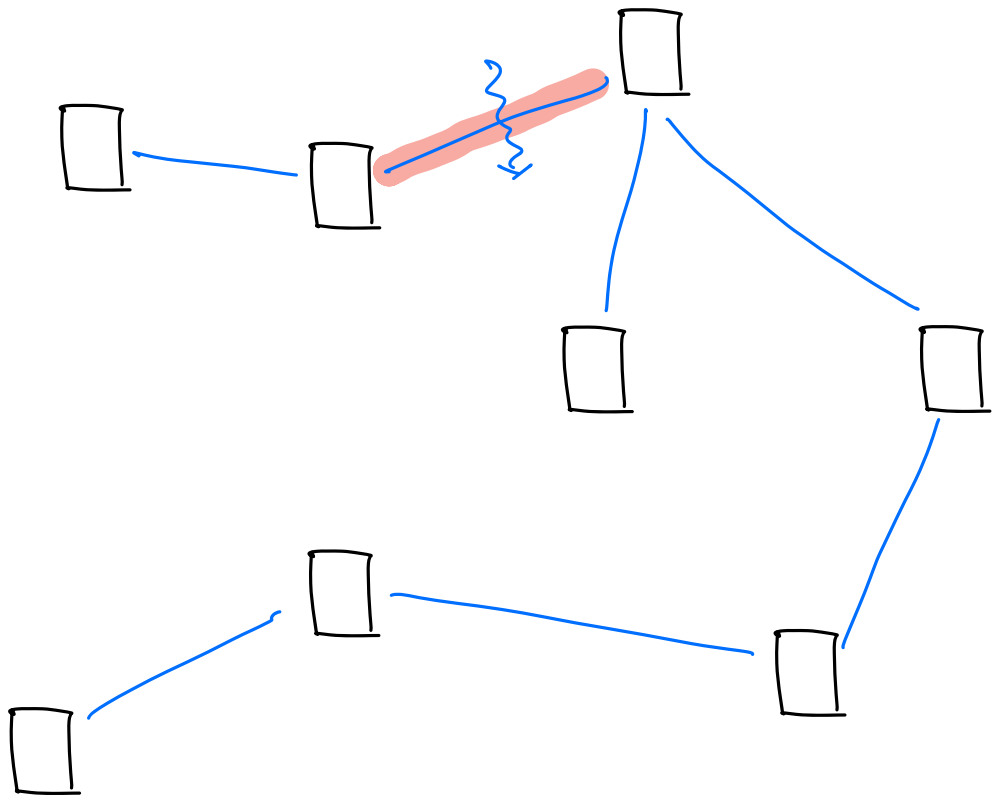
→ All machines should have latest information at any point of time.

Availability

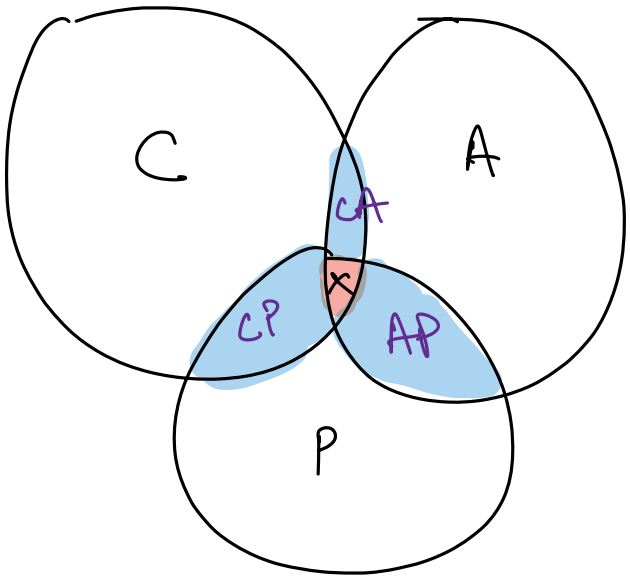
→ If we make a valid query and the system is available to send a response back.

Partition Tolerance

→ N/w partition.



A system which is able to handle the network partition.



In a distributed system, we can only get 2 of the 3 properties.

In a distributed system, PT is always Given.

A P

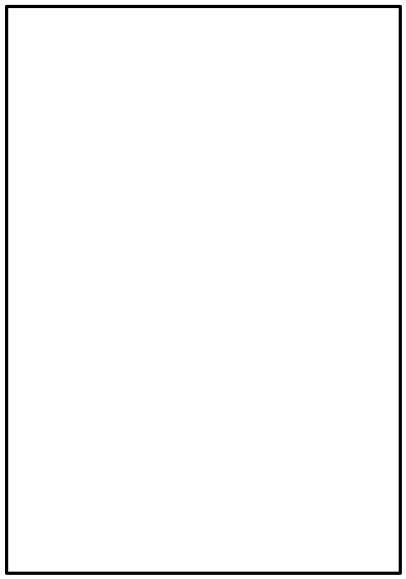
C P

Consistency (vs) Availability.

⇒ If there's No partition tolerance:

↳ Single Server.

↳ Consistent + Available.



HFT

↓

Qs

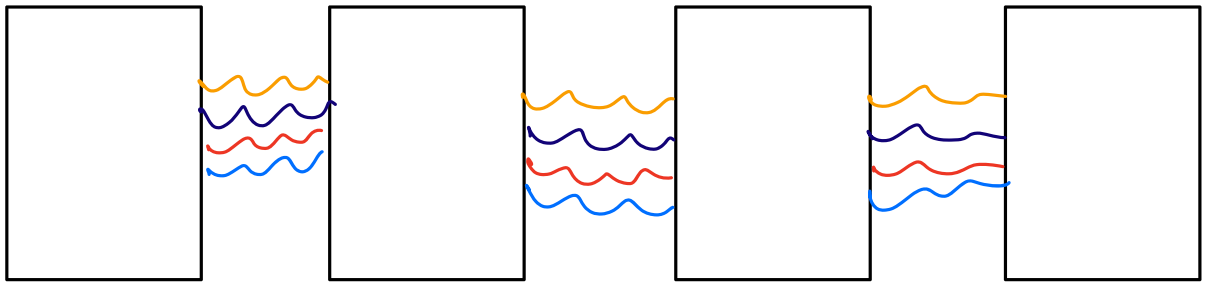
DE Shaw.

Stock Exchanges.

↳ BSE | NSE.

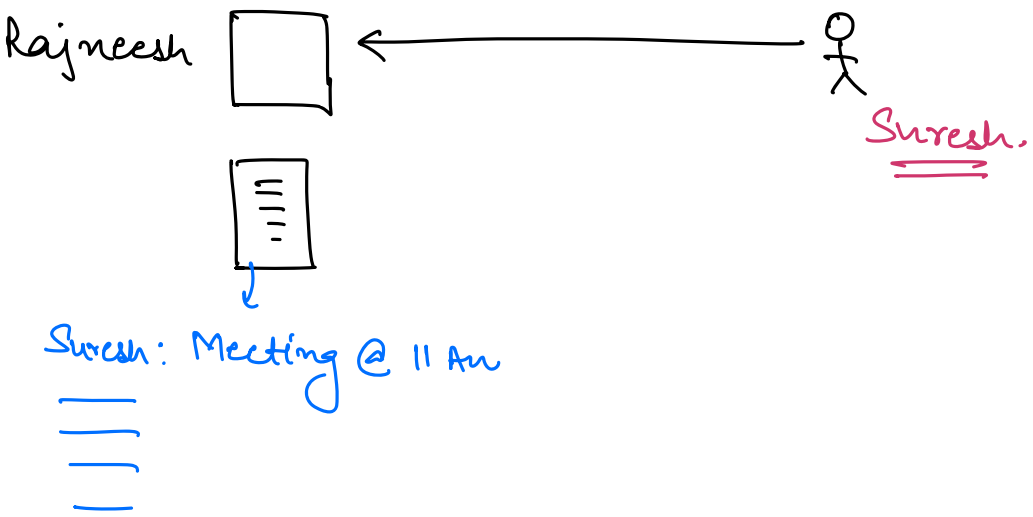
→ High Consistency

→ High Availability.

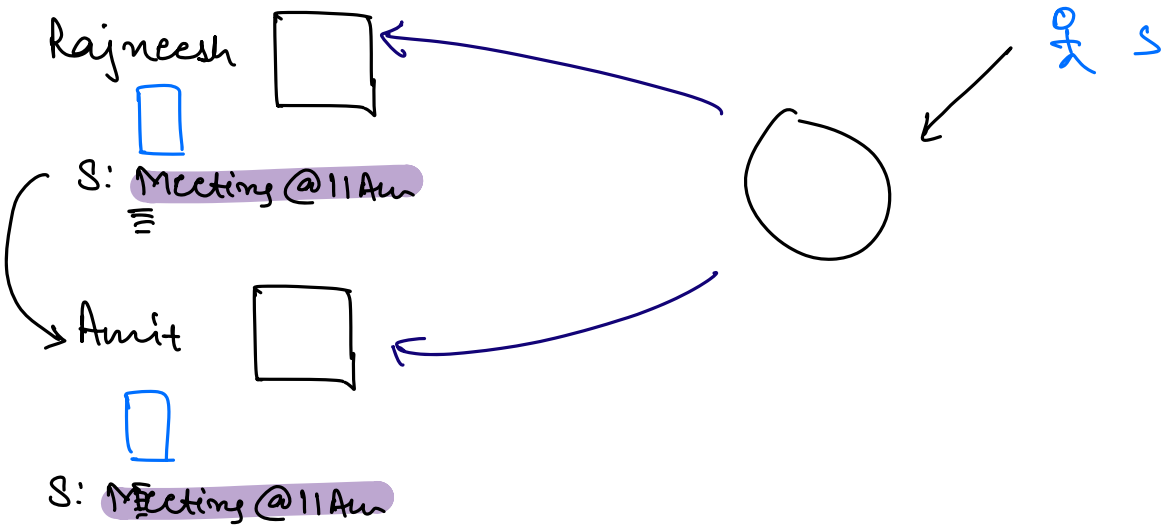


⇒ Multiple redundant connections.

#

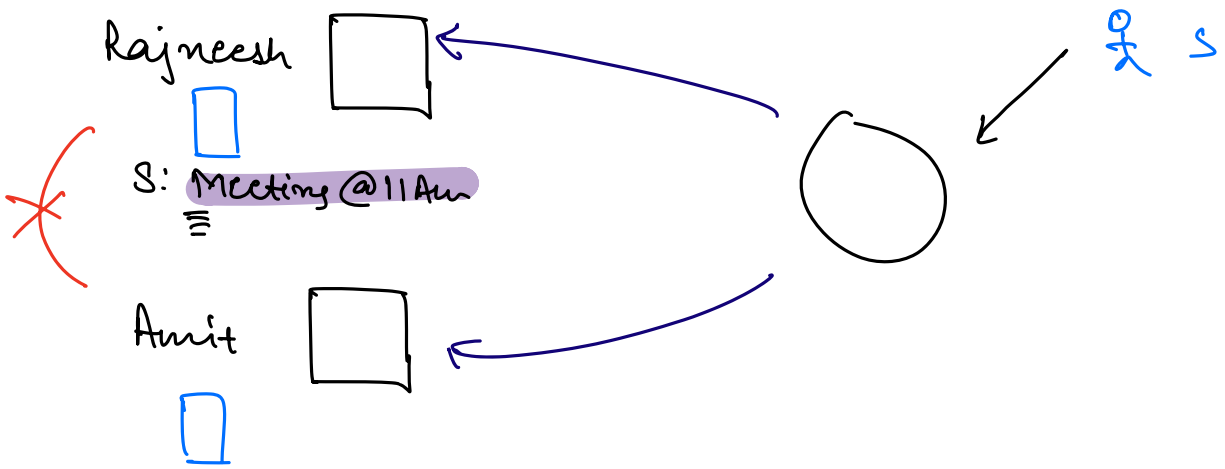


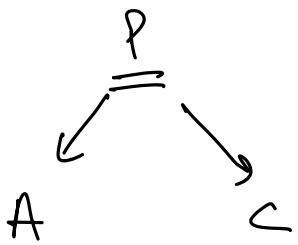
⇒ No Sync issue.



⇒ Sync Issue.

High consistency ⇒ high latency.





⇒ PACELC : Extension of CAP

Consistency
Availability
PT.

Even otherwise in distributed System

- Latency
- Consistency.

If we want a highly Consistent system, latency will get compromised.

—— * ——