

@thriver  
ashish

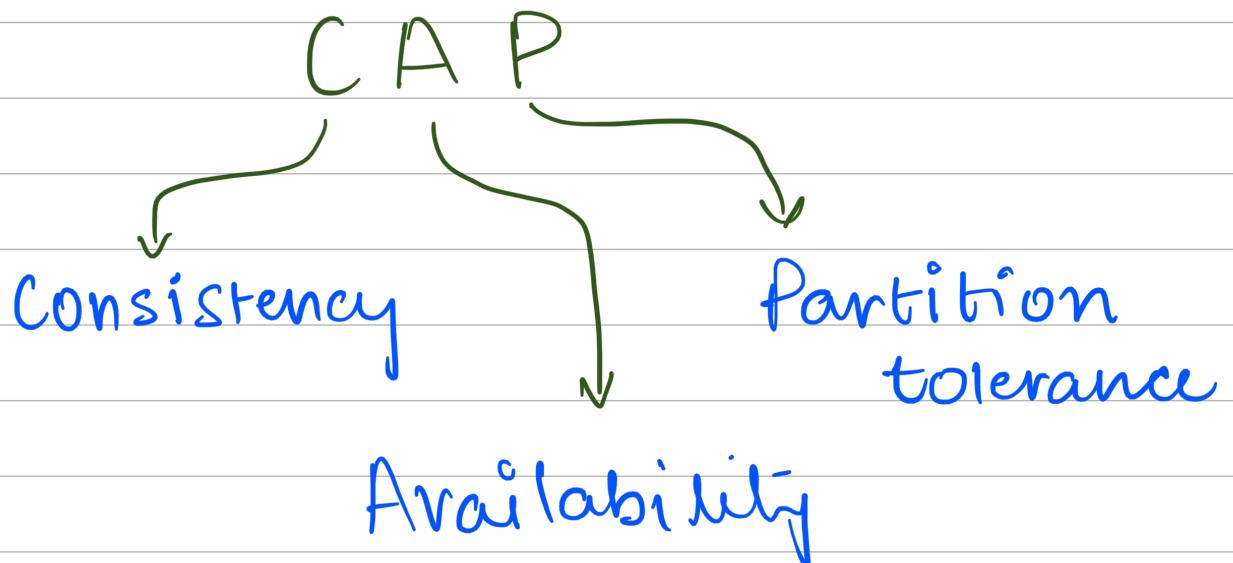
# CAP

## THEOREM

and

## PACELC

## THEOREM



CAP Theorem states that

Distributed Data Stores can provide  
at most two properties out of  
three .

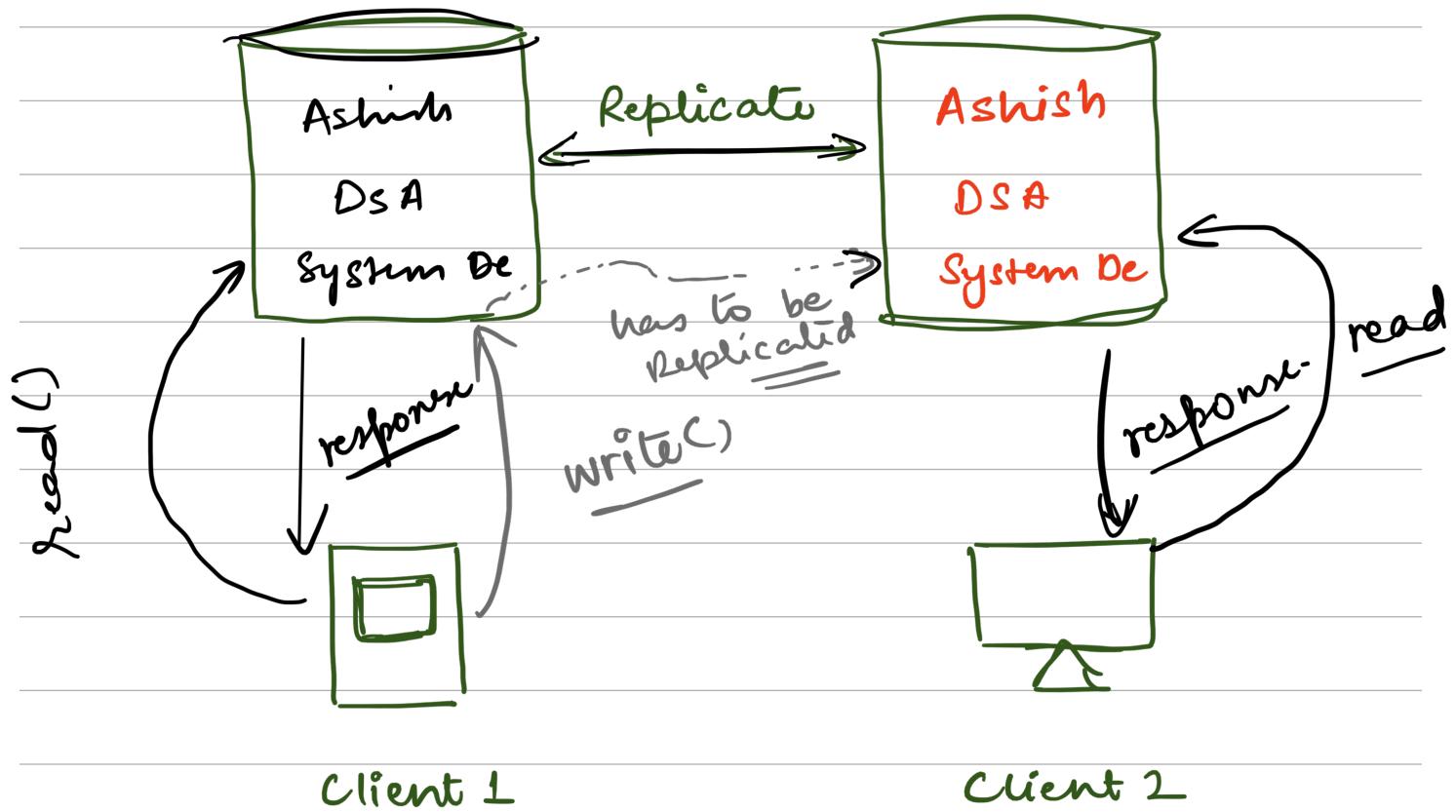
- \* Its not possible to provide all three simultaneously.

Remember

PARTITION here refers to Network  
partition.

DB 1

DB 2



So the two Data stores DB1 and DB2 have to communicate with each other over a network and they must be Partition Tolerant ,

even if the communication breaks in case of the network failure they must continue to function correctly .

So now the choice has to be made in between -

## Consistency OR Availability



Every Read Request Receives the most recent value or an error.



Every Request gets a valid response and no errors

\* So if you chose Availability then, in case of partition failure it won't be able to replicate the data to other Data Store Nodes and the data might be inconsistent.

Eg: Instagram is highly Available System.

\* When network failure resolves, then it will Replicate the Data.

If you chose Consistency, then in case of network failure, you must block the write request and send response that system is Down or is not Available.

Eg: Banking Systems

↳ highly consistent.

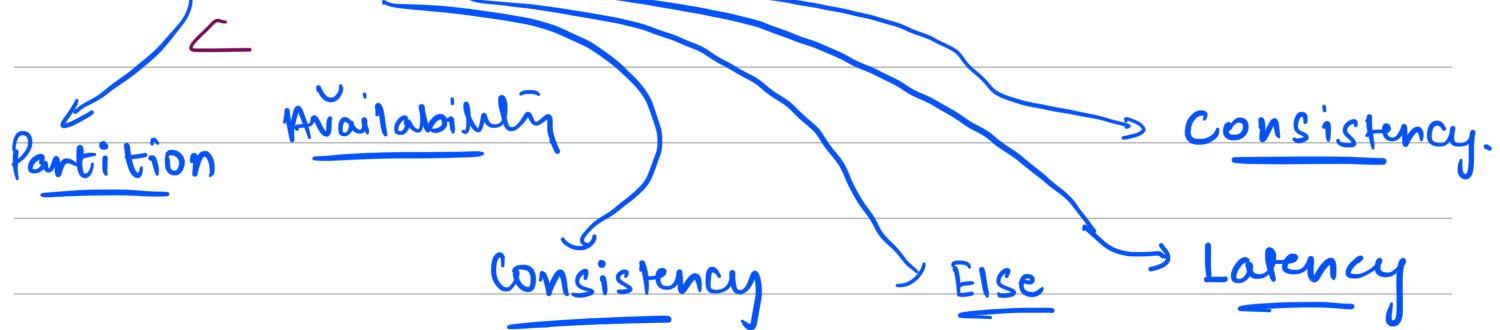
In Nutshell

\* You have to make choice in between Consistency and Availability while designing your system.

@thrivershish

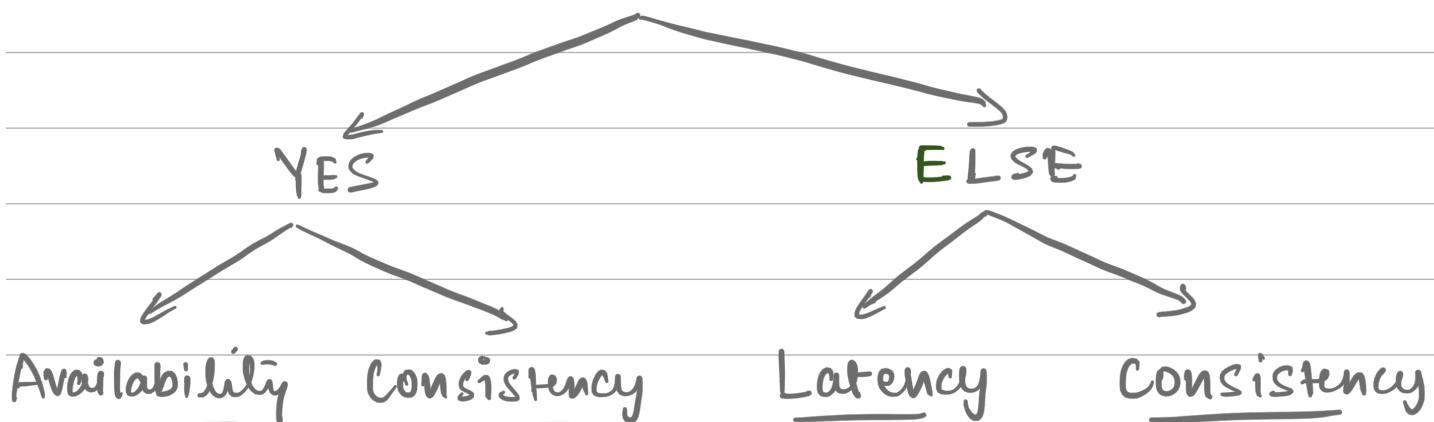
Swipe →

# PACELC Theorem



## Extension of CAP Theorem.

### PARTITION



\* [ Delay before a transfer of Data begins. ]

### Example

By default, Dynamo DB, Cassandra, Riak, Cosmos DB, are PA/EL systems.

If partition occurs, they give up Consistency for availability and under normal operation they give up Consistency for Lower Latency.

@thriverashish

insta twitter youtube