1 **What is consistency?**

consistency = every-one sees the same data at the same time.

example
- you like a post
- Refresh immediately
- like count increases

Thats strong consistency.

When consistency breaks
- you like a post
- Refresh → old count
- Refresh again → new count

This is eventual consistency.

2 **Why consistency is hard to at scale**

Because:
- Data is stored in multiple places
- Network is slow and unreliable
- machine fails

So update dont reach everywhere instantly

3 **CAP Theorem**

In a distributed system you can only choose 2 out of 3:

C - Consistency
A - Availability
P - Partition tolerant

4 **What is partition tolerance?**

Network failure between system components

example:
- App servers cannot reach DB
- Data centers can't talk
→ Partition will happen

So P is non-negotiable

5 **The Real choice you make**

Since P is mandatory you choose between:

CP (Consistency + Partition)
- Rejects request if data maybe wrong
- Safer, stricter

AP (Availability + Partition)
- Serve requests even if data is stale
- more user-friendly.

6 **Real-world example**
AP system
- instagram likes
- Twitter follower count
- Youtube views

The prefer:
Show something, even if slightly wrong.

CP system
- Banking transactions
- payments
- inventory system

They prefer
correctness over availability.

7 **Like Counter example revisited**
   During WB outage:
- AP choice → accept likes, sync later
- CP choice → block likes
most social apps choose AP

8 **Common beginner mistake**
   I'll make it consistent and available
   I'll ~~try~~ just retry.
   You can't beat physics.

9. **mini design exercise**
   Task 5
   For each system choose AP or CP and explain why:
   1. Like counter
   2. Bank balance
   3. Online shopping cart
   4. messaging app