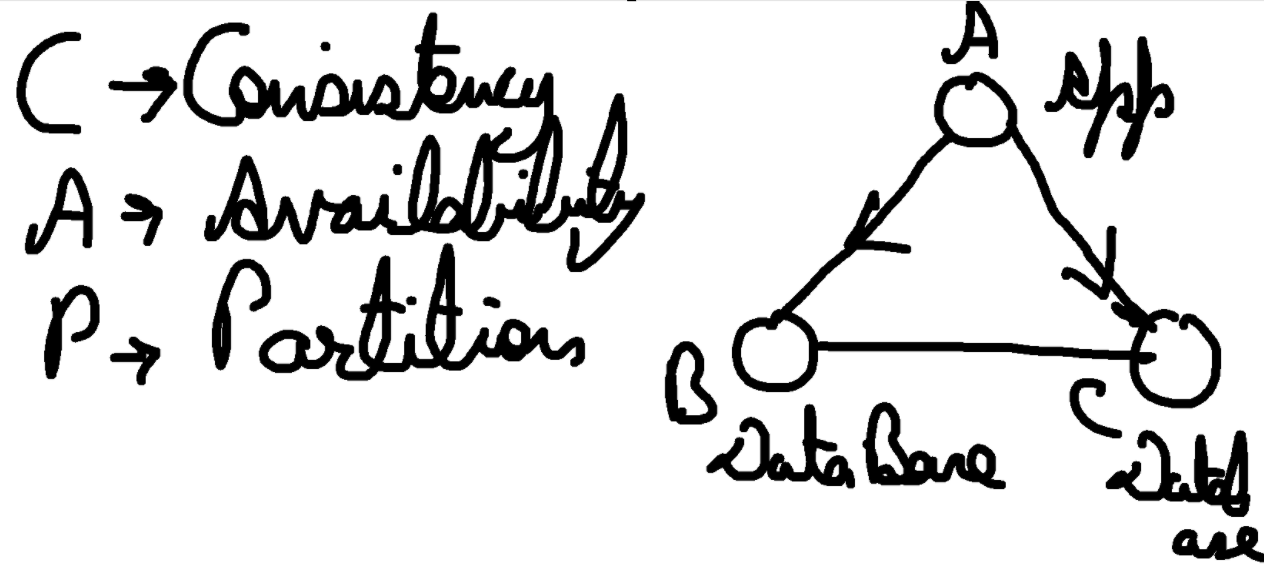
**High Level Design- CAP Theorem**

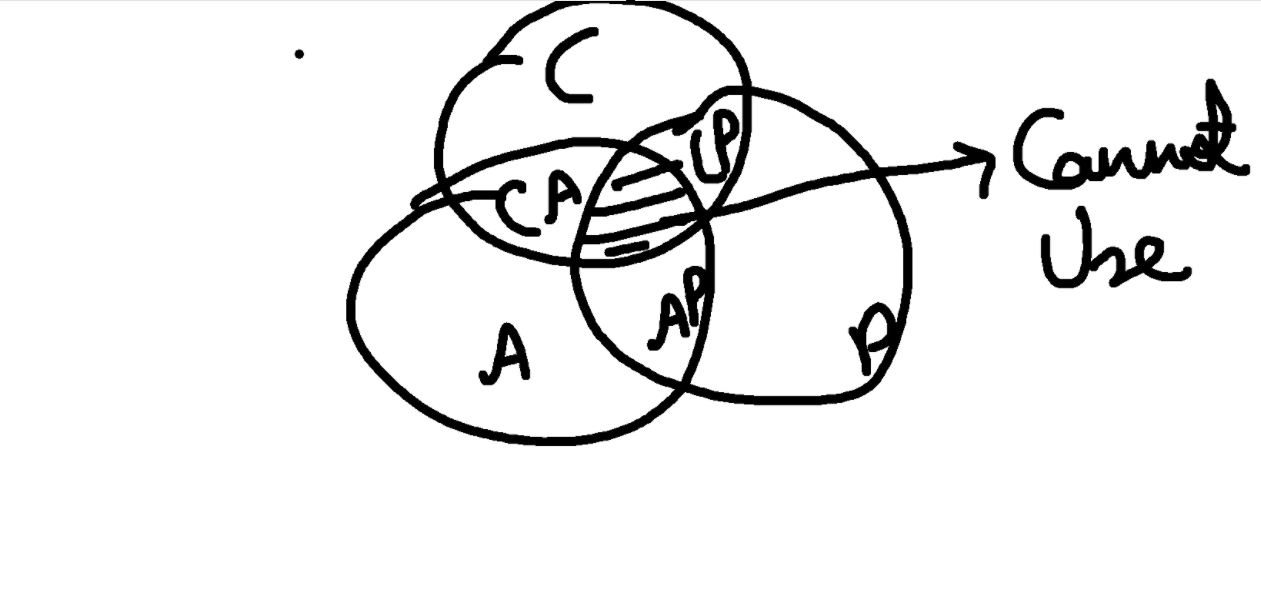
1. **Desirable property of distributed system with replicated data.**
2. **C-> Consistency, A-> Availability, P-> Partition**

**Example:-**

1. **A node is application and in distributed system**
2. **B, C nodes are database which is present in USA and India**
3. **A can get the data from both database**
4. **When A application sends the data then it will replicate the data in both the nodes B, C.**

****

1. **We cannot use the all 3**

****

**Consistency ->** Let us suppose B node has data as a=4 and C node also have data a=4, if A node will write data as a=5, then B node will become as a=5. Now B node wll replicate the data to C node with a=5.

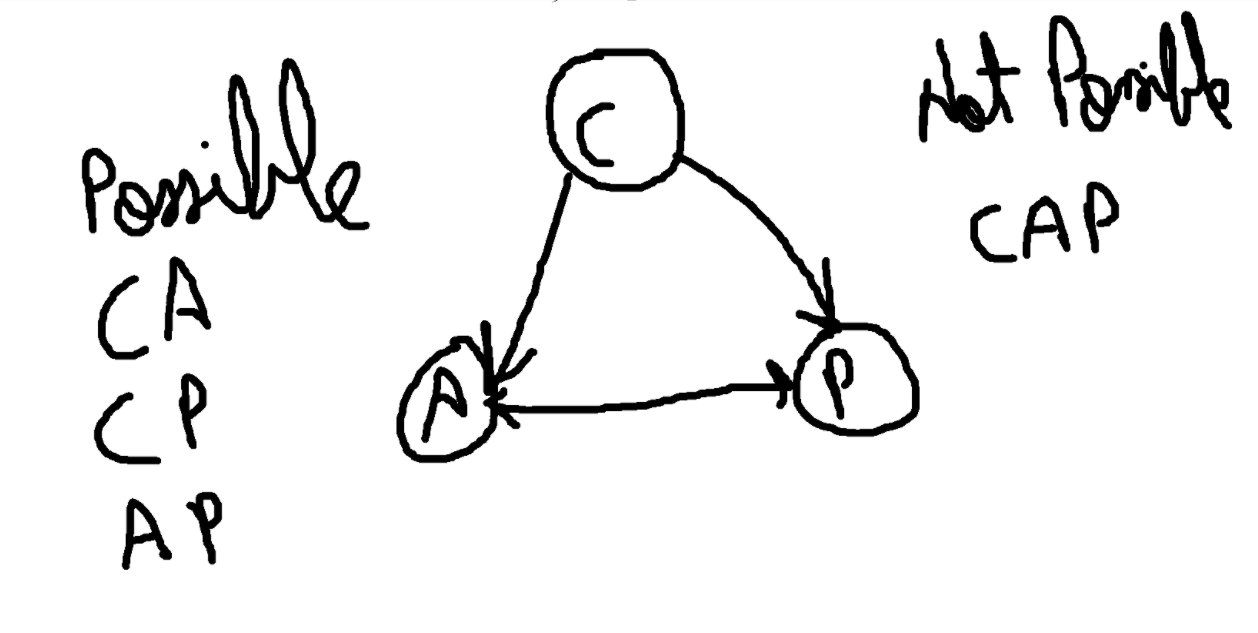
Now, if A node wants the data from C node then it should get as a=5.

**Availability ->** It means if we have 2 nodes as B and C, then all DB node should respond. It does not matter whether it sends correct data or not.

**Partition Tolerance ->** Let us suppose We have 2 Database node as B and C but due to some issue B is not able to replicate the data to C node.

Application node A is still able to send the data to and get the data. It means system is UP and User does not know about the Partition breakage. Then, it is called as Partition Tolerance.

**Why we cannot use all 3 together?**

****

**AP -> Let us suppose, there is partition breakage between nodes and b node has data 6 and C node has data 5, then there is no consistency. Because due to partition breakage, data 6 is not replicated to C node.**

**But data A can get the data from both the node.**

**CP -> Let us suppose, I cannot loose consistency. And if there is partition breakage, I will drop the node C and getting all the request from B node.**

**Since, I drop the node so I loose the availability.**

**CA - > Let us suppose, there is partition breakage issue and there is not partition tolerance, the system will not UP and A cannot query. So, I need to wait until partition will restore.**