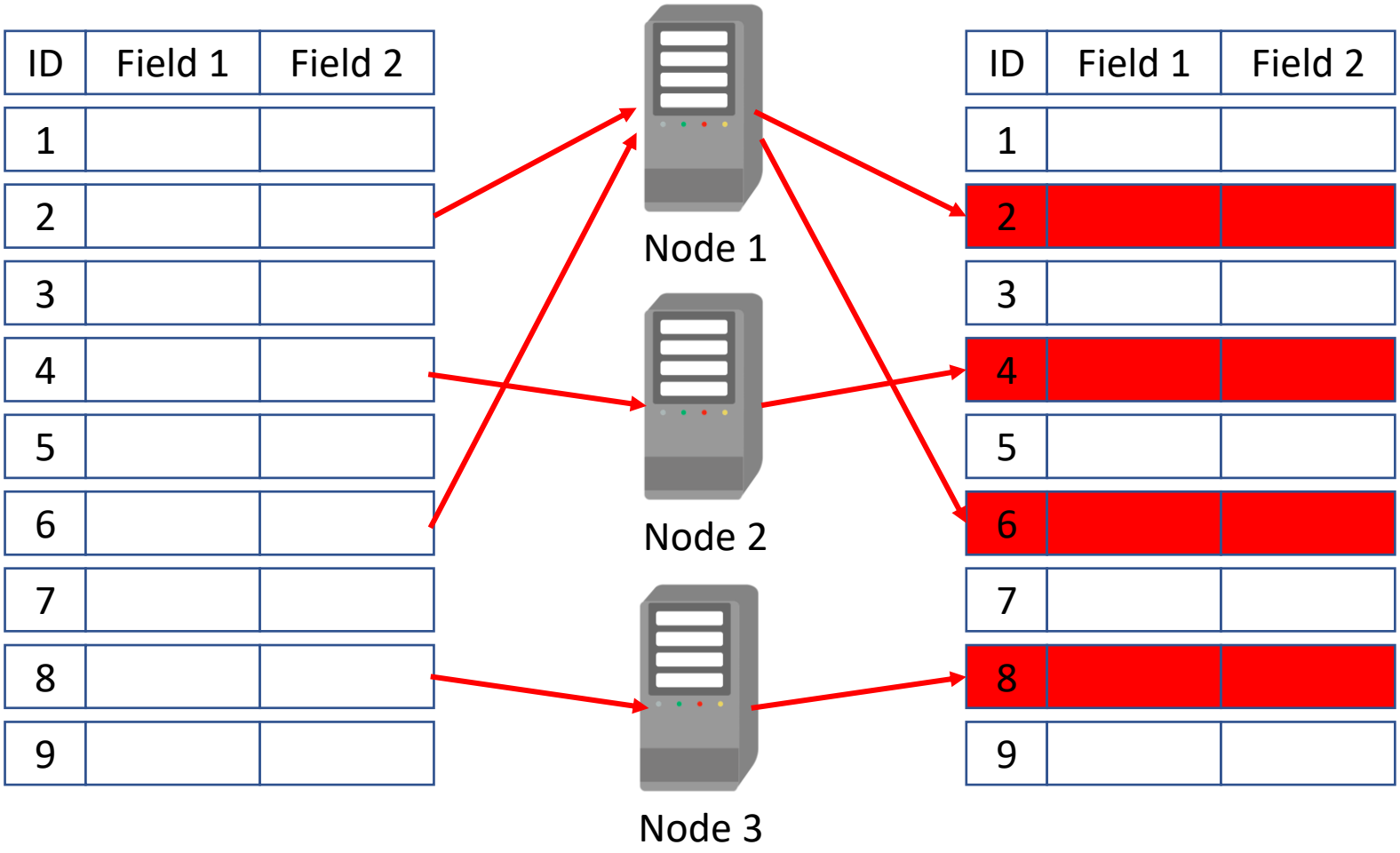
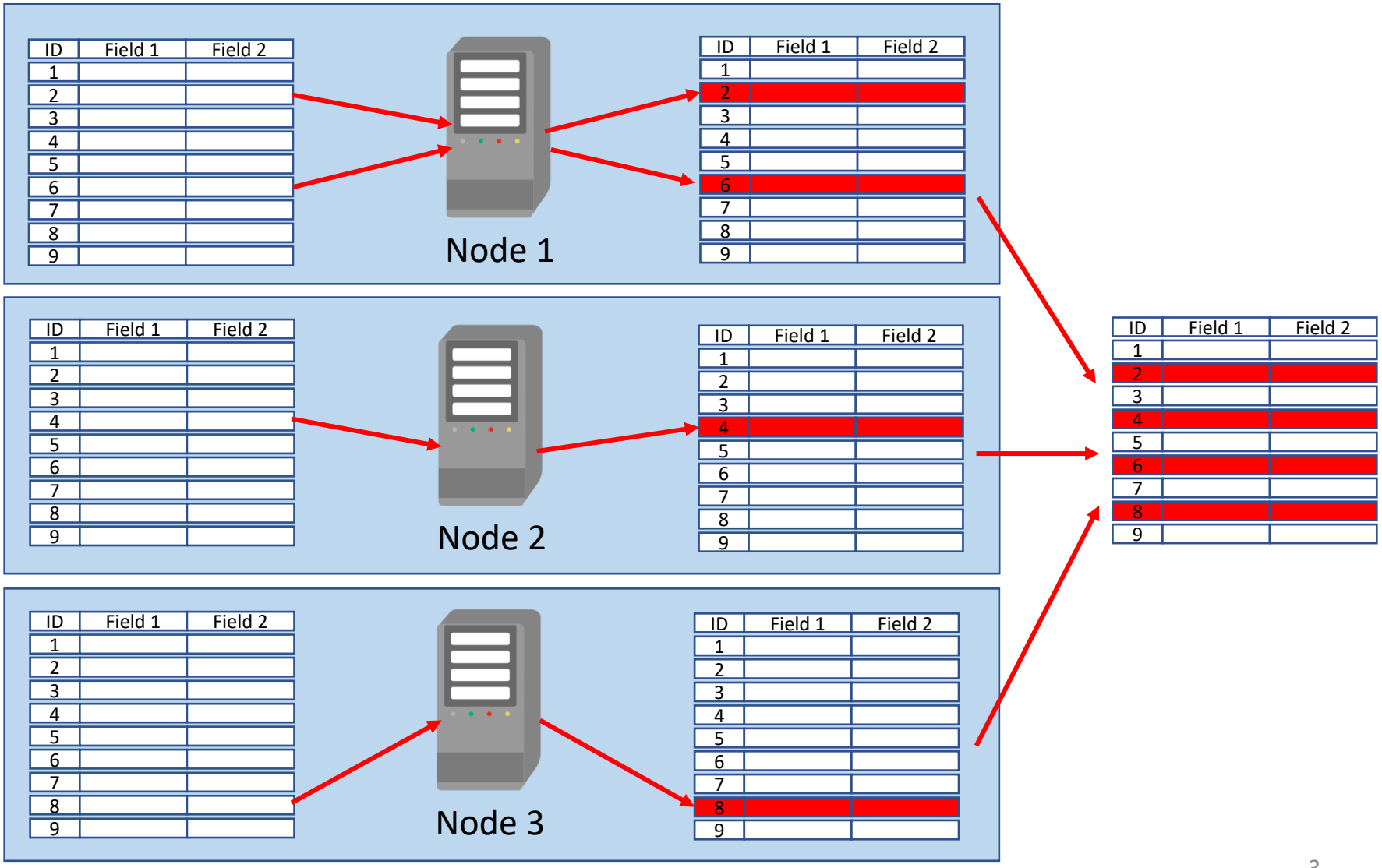
Distributed Storage

ENGR689 (Sprint)

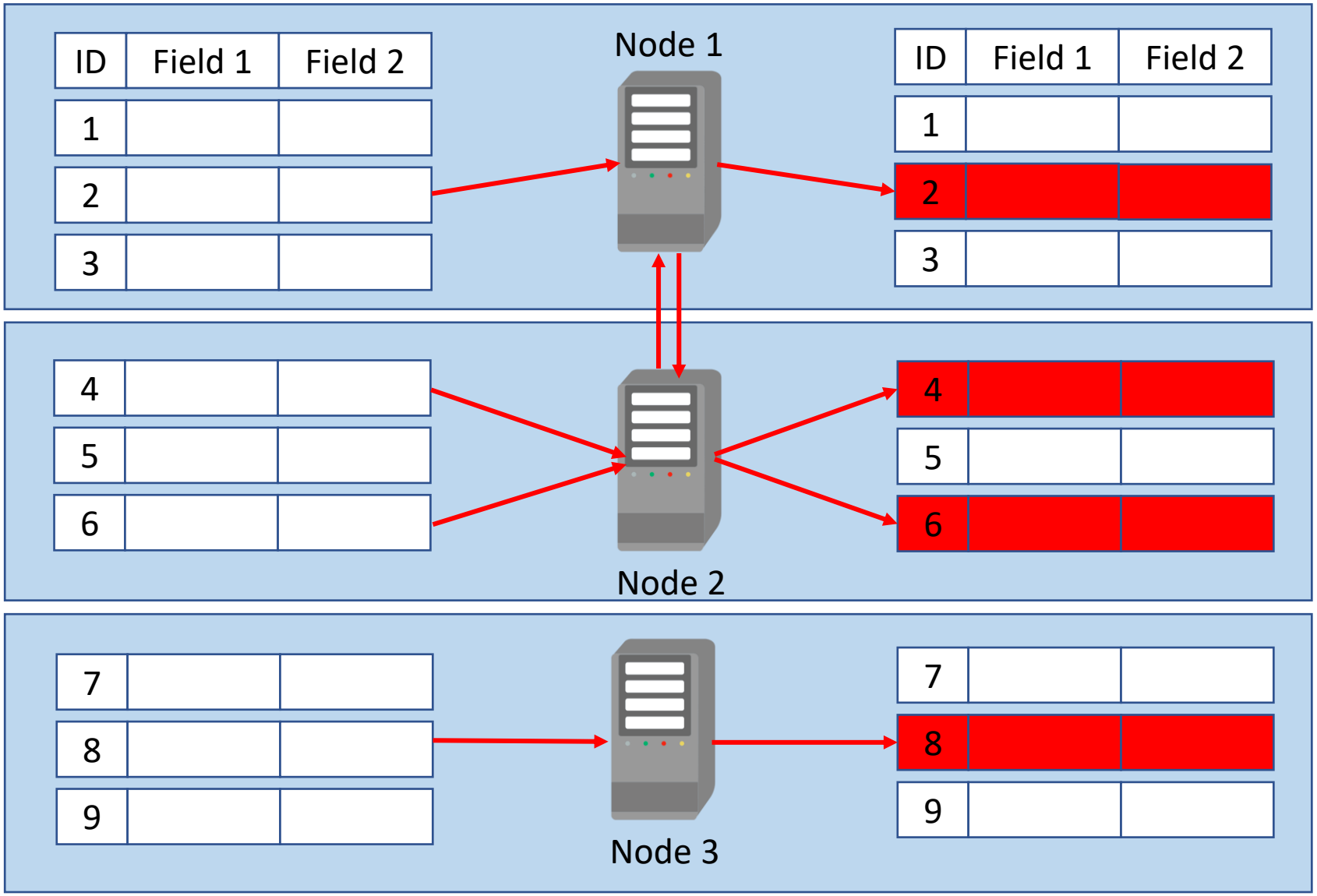
Parallel Access to Data



Replication



Partitioning

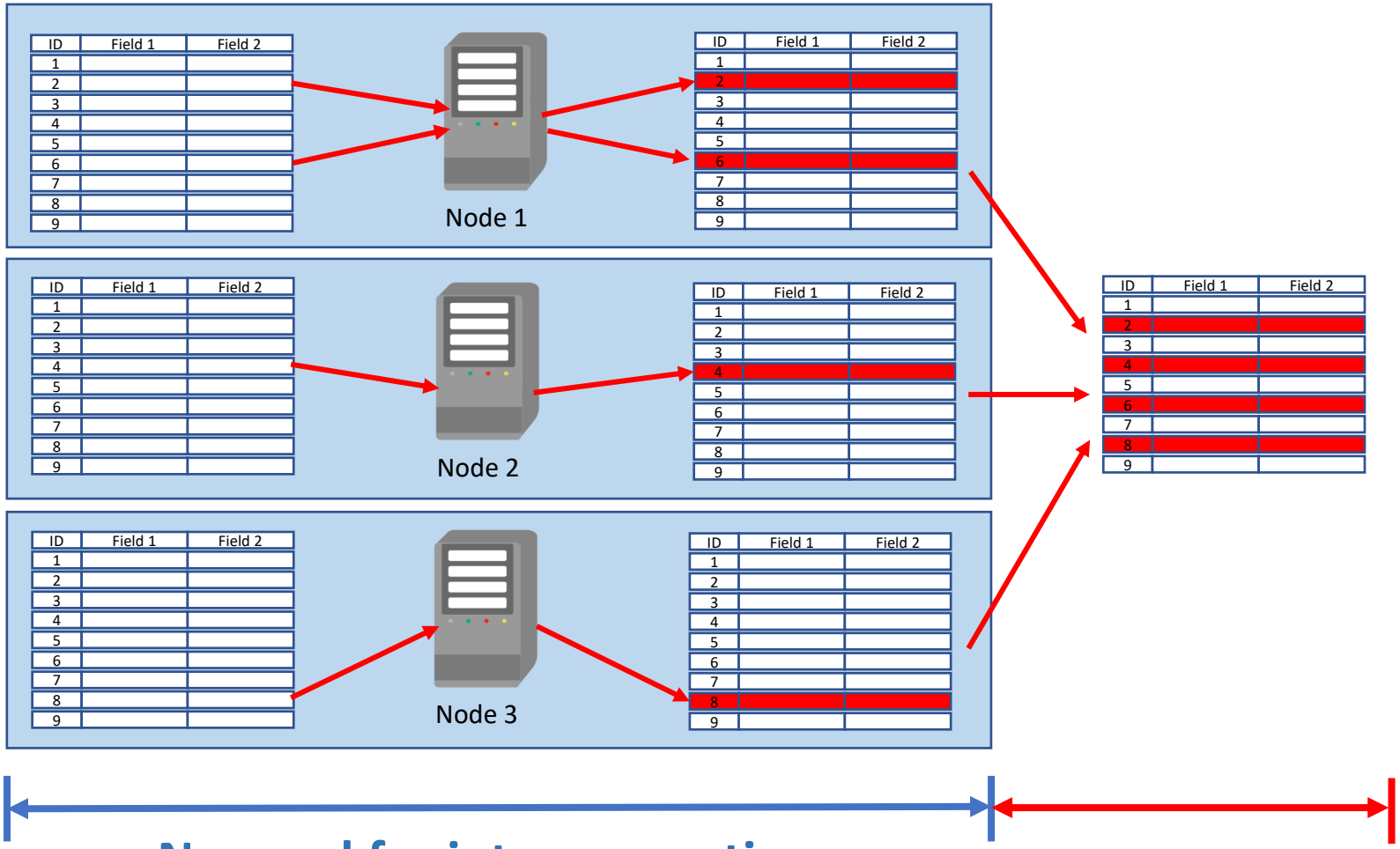


Network Issues

- Network latency (100ms+) > Local DRAM latency (10-30ns)
- Network will always fail:
  - 50% chance the interval between inter-data-center network failures is less than 5 minutes
  - 10% chance network repairment can take more than 1 hour or even 1 week

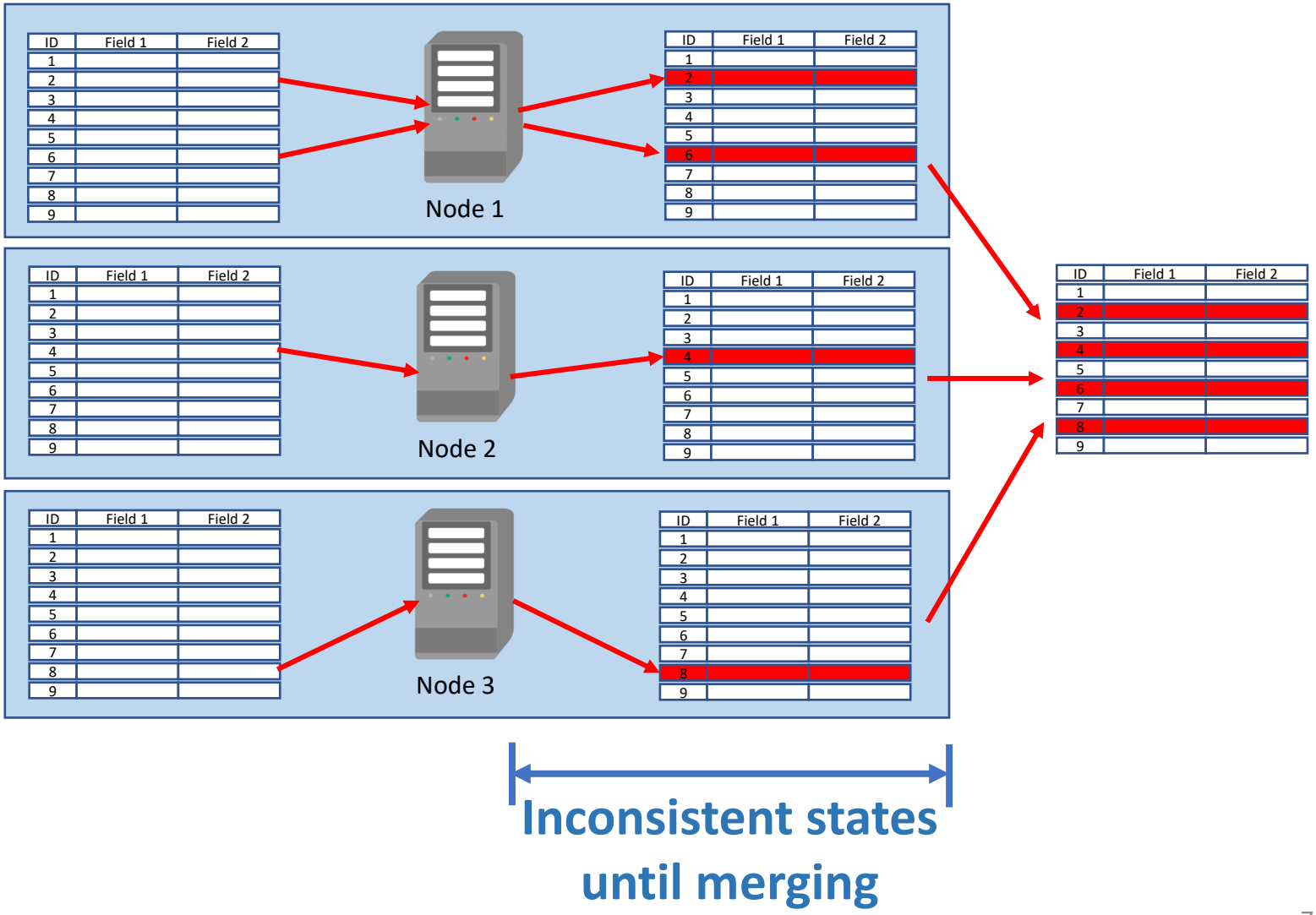
Source: Microsoft 2011 SIGCOMM paper

Partition Tolerance



(Can tolerate partitioning caused by network failure)

Consistency



CAP Theorem

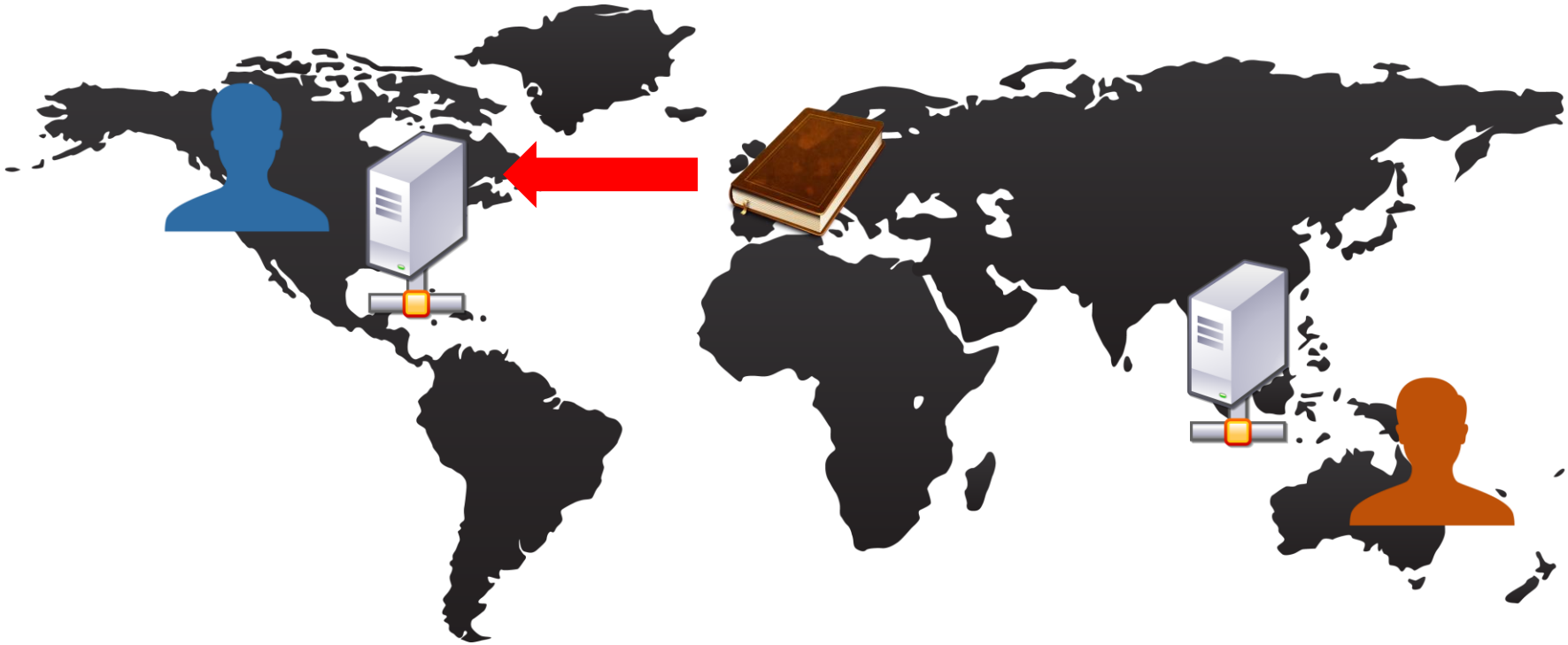
- **Trade-offs** of three properties in a distributed system
- **Consistency:**
  All nodes see the same states
- **Availability:**
  If one node fails, other nodes can still operate
- **Partition tolerance:**
  The system can be partitioned by network

By Eric Brewer in PODC 2000 Keynote

Think about Amazon

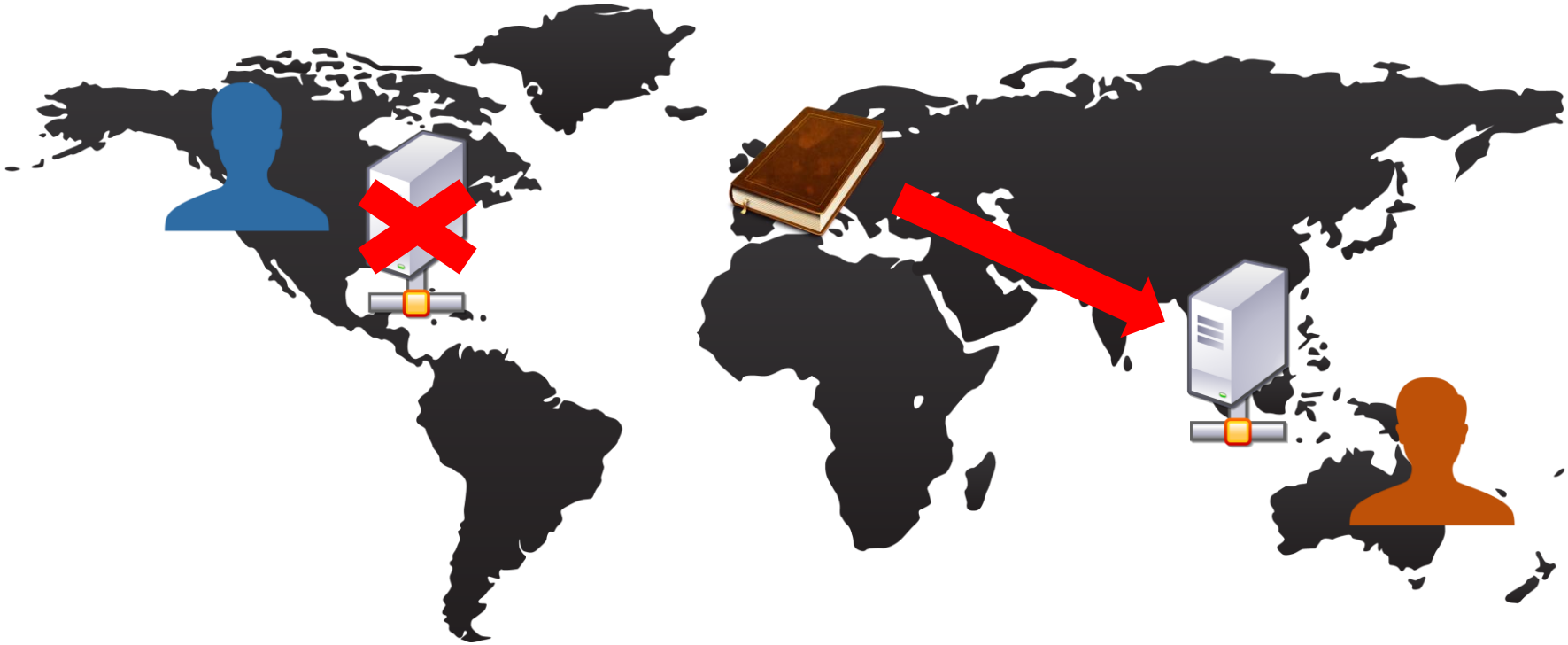
Consistency

- Each copy can only be sold once



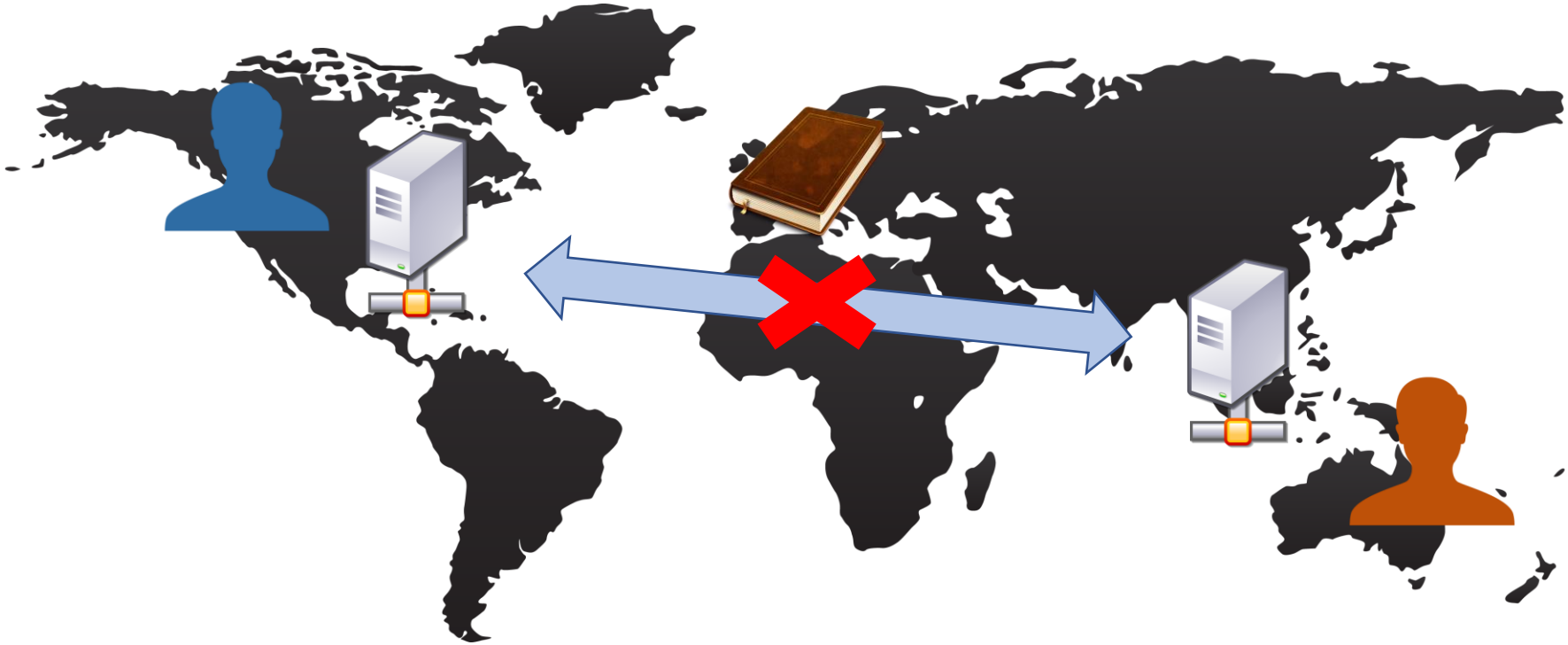
Availability

- If North America server failed, Asia server can still sell the book.



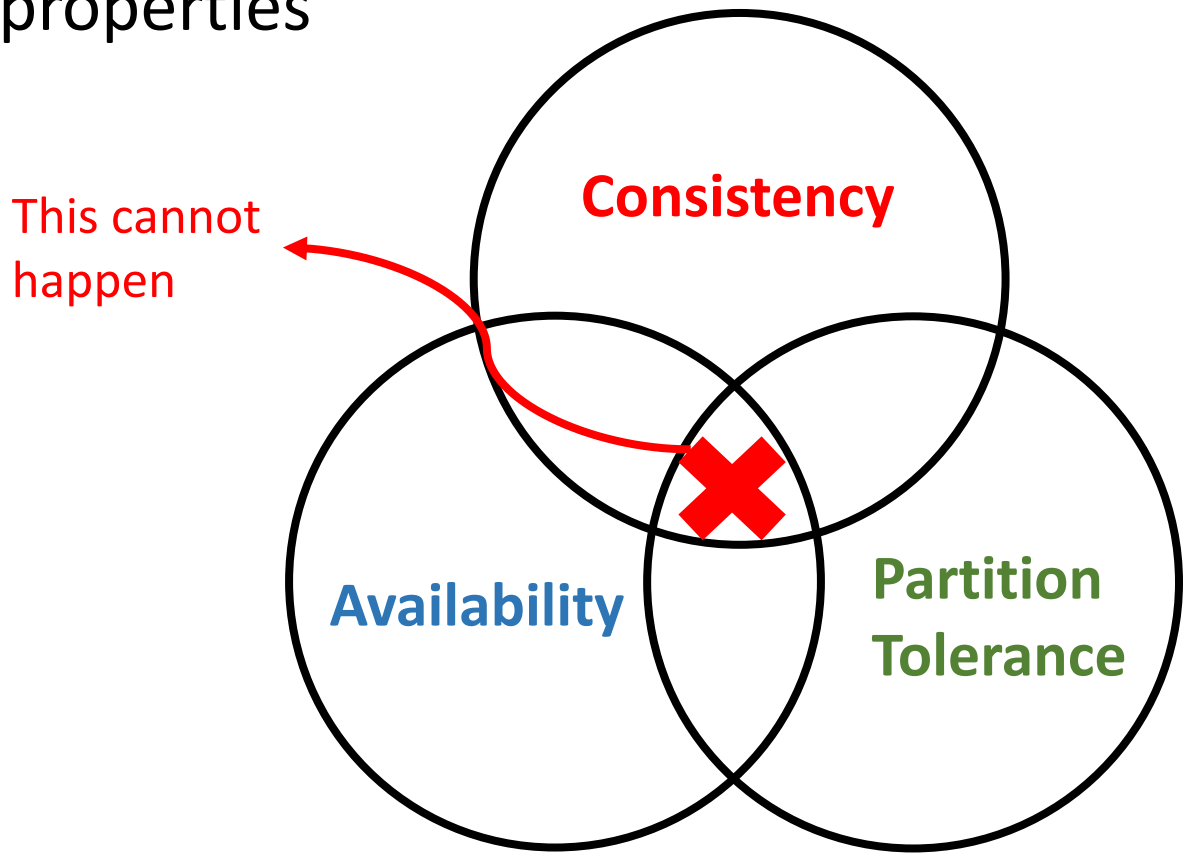
Partition Tolerance

- If interconnection temporarily fails, the system can still work (but with some trade-offs)



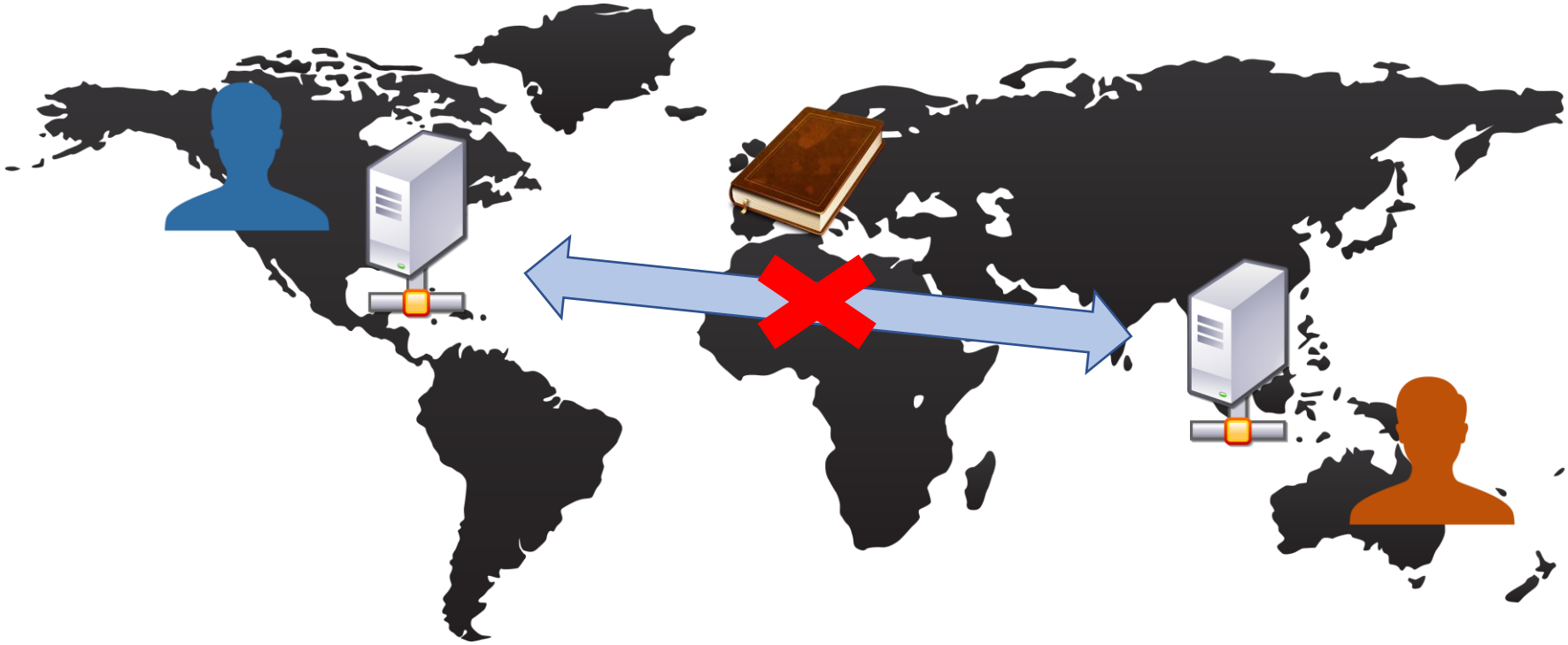
Impossibility in CAP Theorem

- Any distributed system cannot achieve all three properties



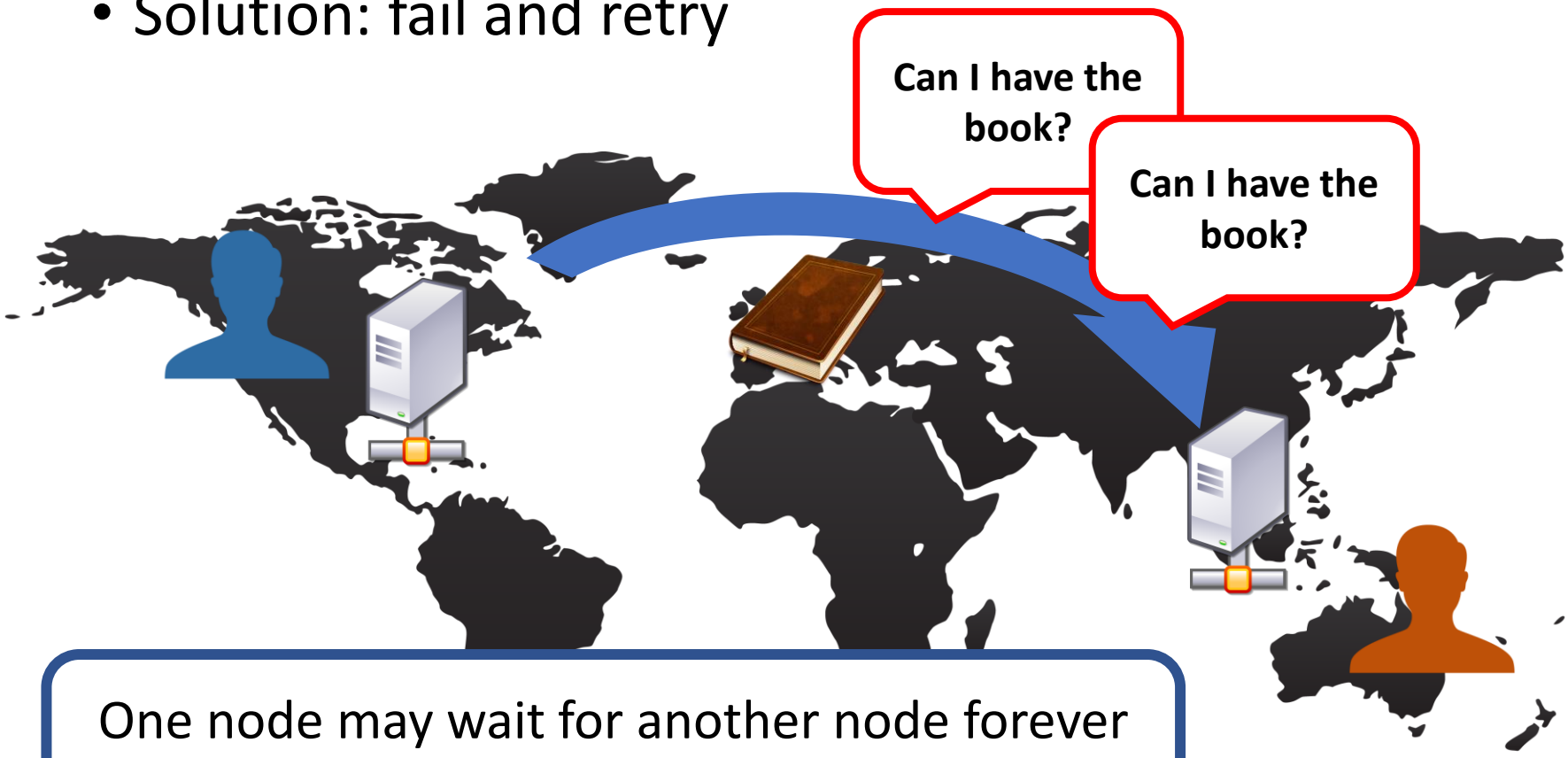
Consistency + Partition Tolerance

- Requirement: only one book to sell, but may lose interconnection



Consistency + Partition Tolerance

- Solution: fail and retry

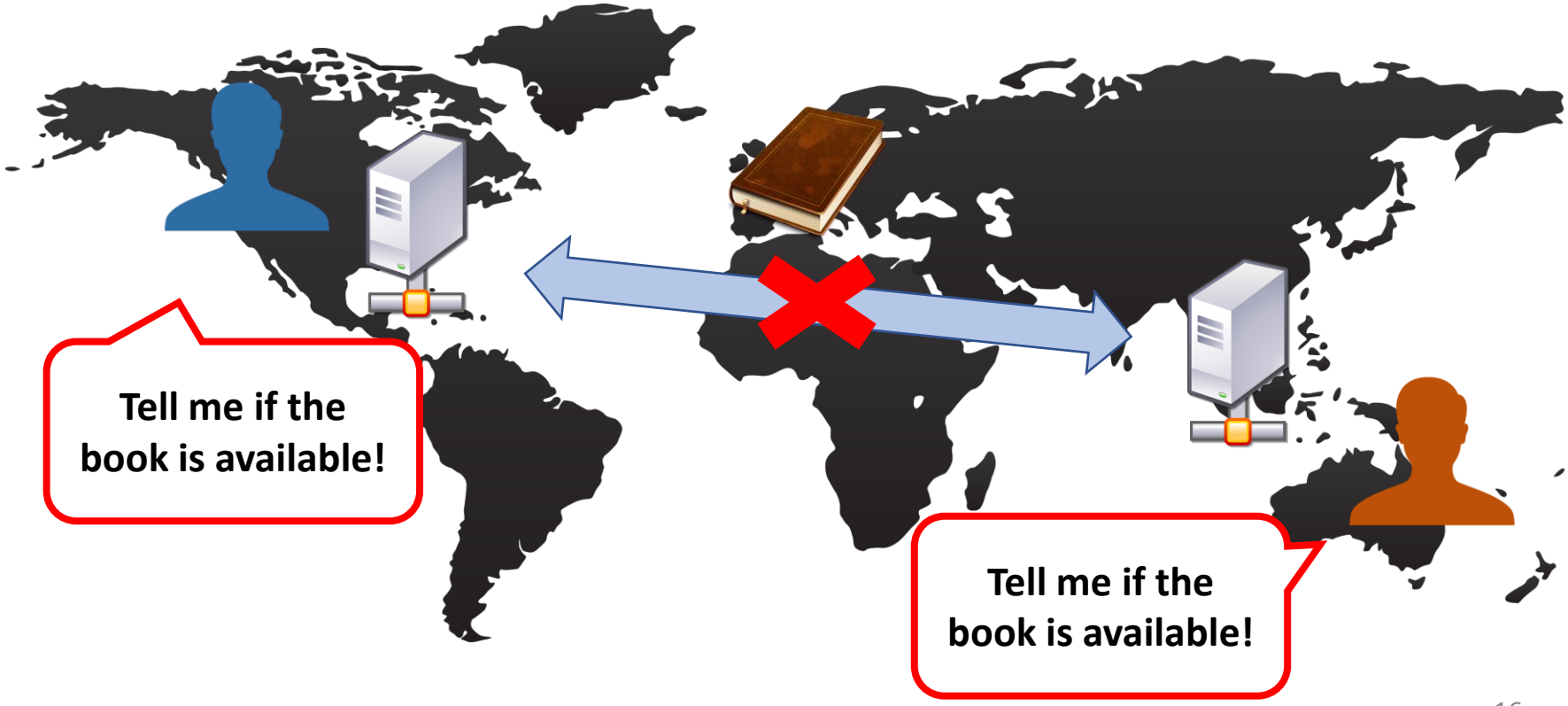


One node may wait for another node forever

➔ No availability

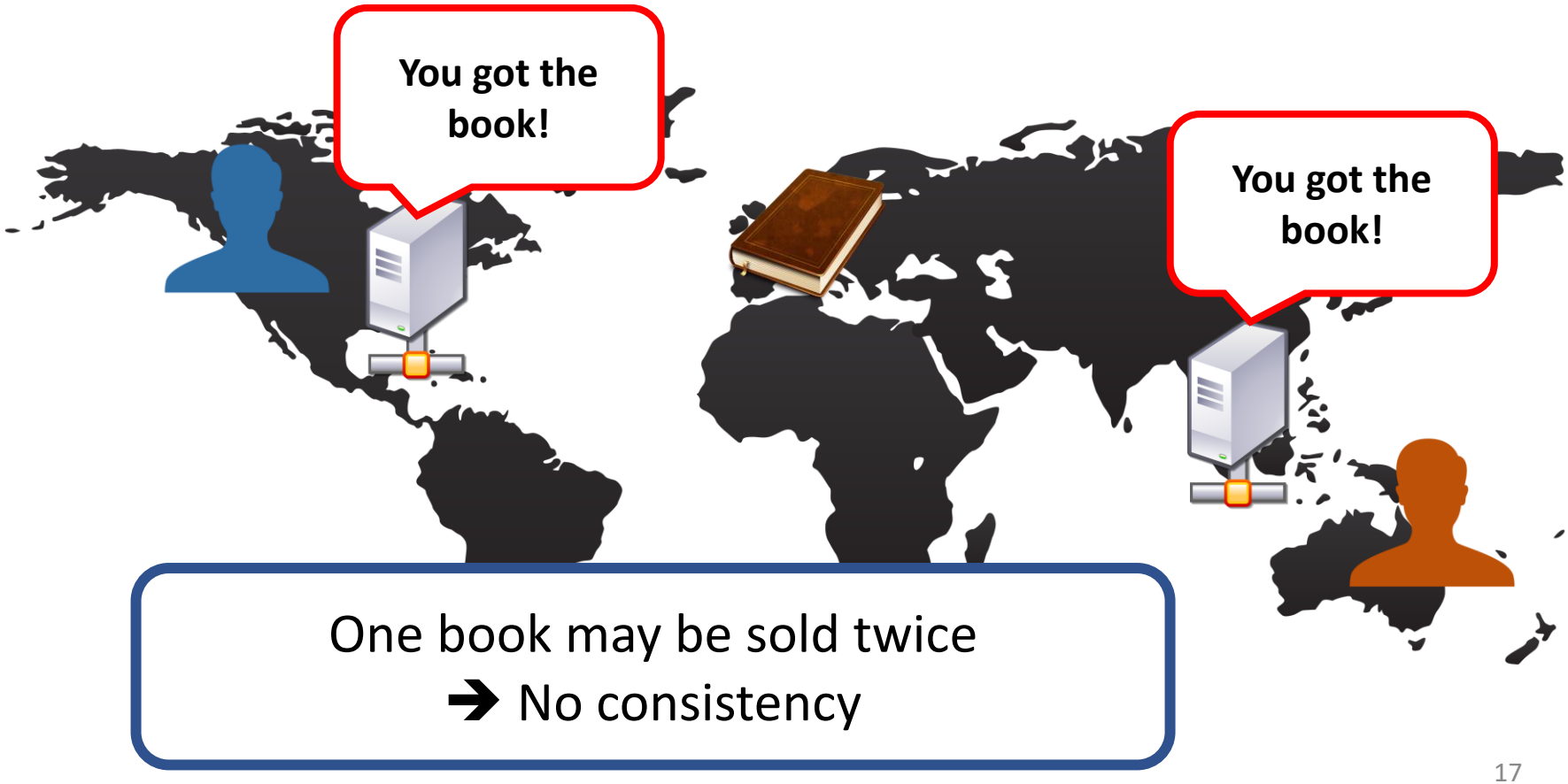
Availability + Partition Tolerance

- Requirement: everyone needs to get answered, even without interconnection.



Availability + Partition Tolerance

- Solution: distributed states

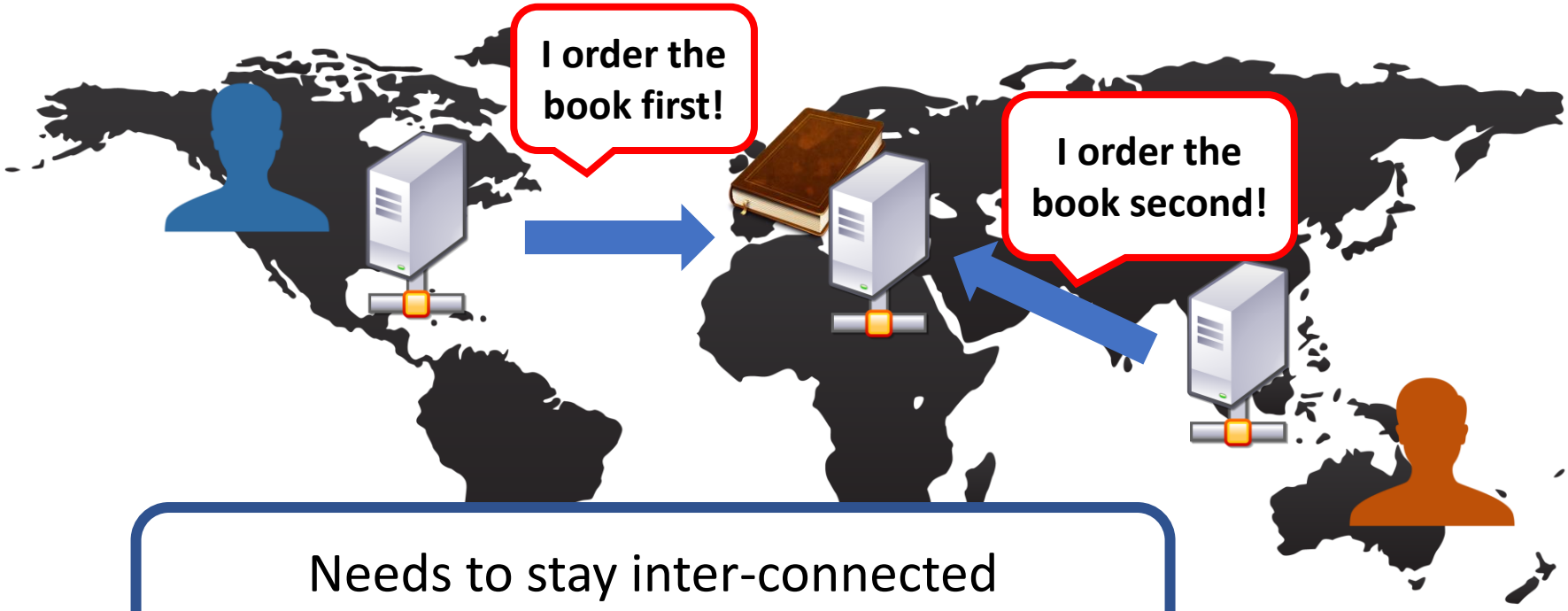


Consistency + Availability

- Requirement: only one book to sell, and everyone gets answered immediately

Consistency + Availability

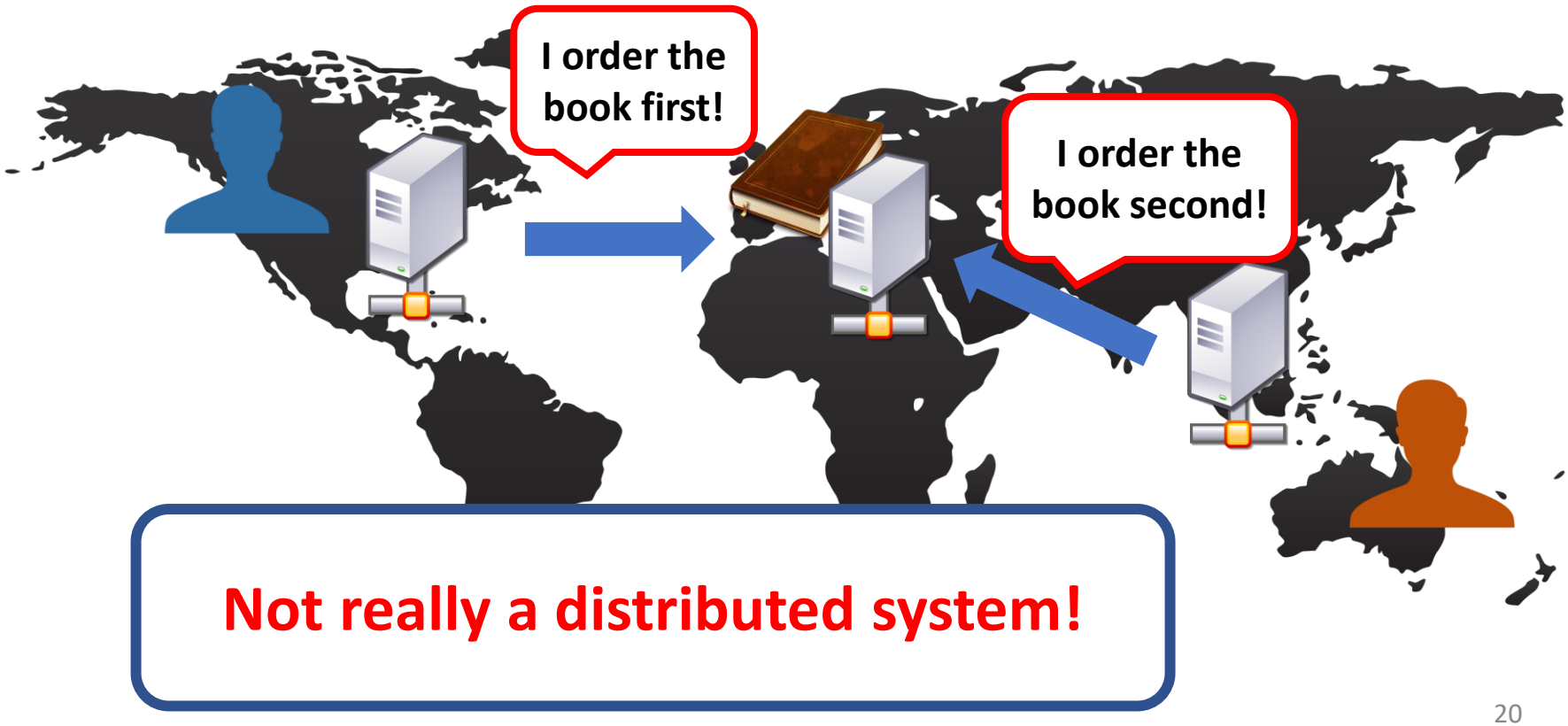
- Solution: centralized server



Needs to stay inter-connected
➔ no partition tolerance

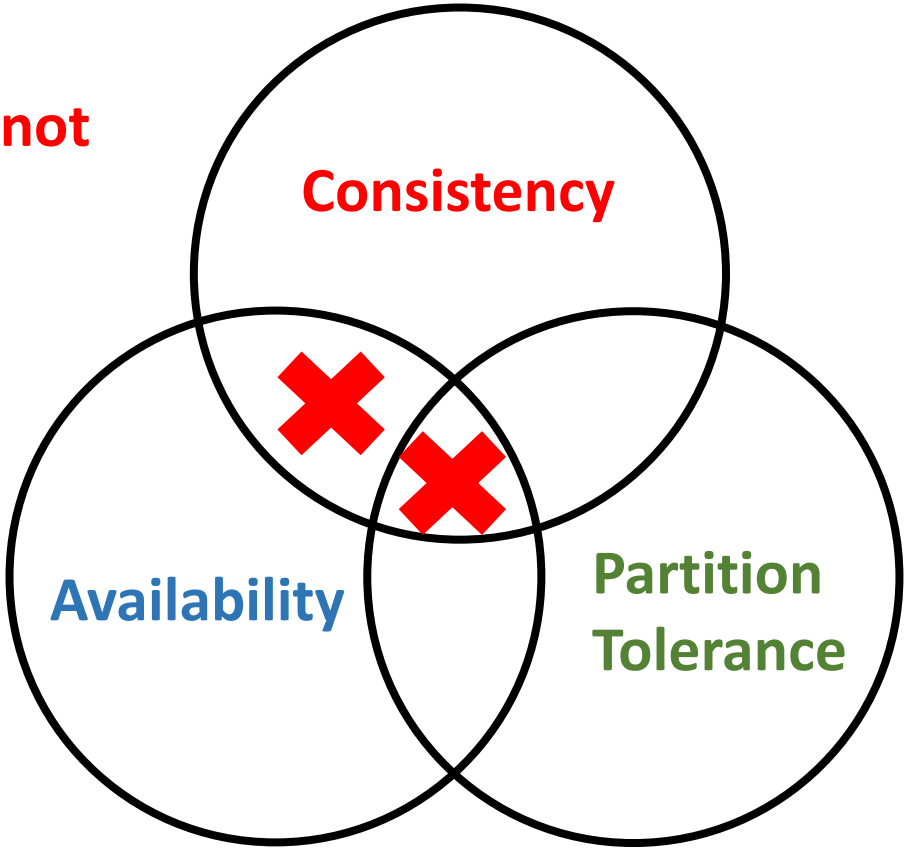
Consistency + Availability

- Solution: centralized server ✖



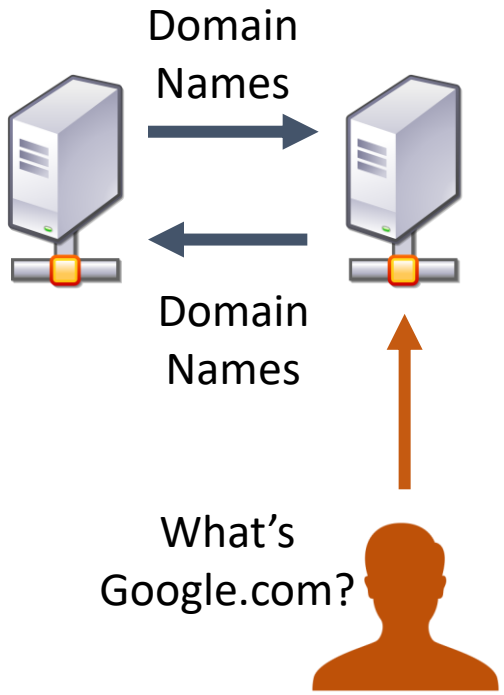
You Can't Have Consistency + Availability Only

Network failures cannot be prevented, so you must have P.



How to Trade Off CAP Theorem

- Example: DNS
  - **Highly partitioned**: Multiple servers, world-wide service
  - **High availability**: client needs to get response immediately
  - **Eventual consistency**: If a domain name is changed, takes hours to update

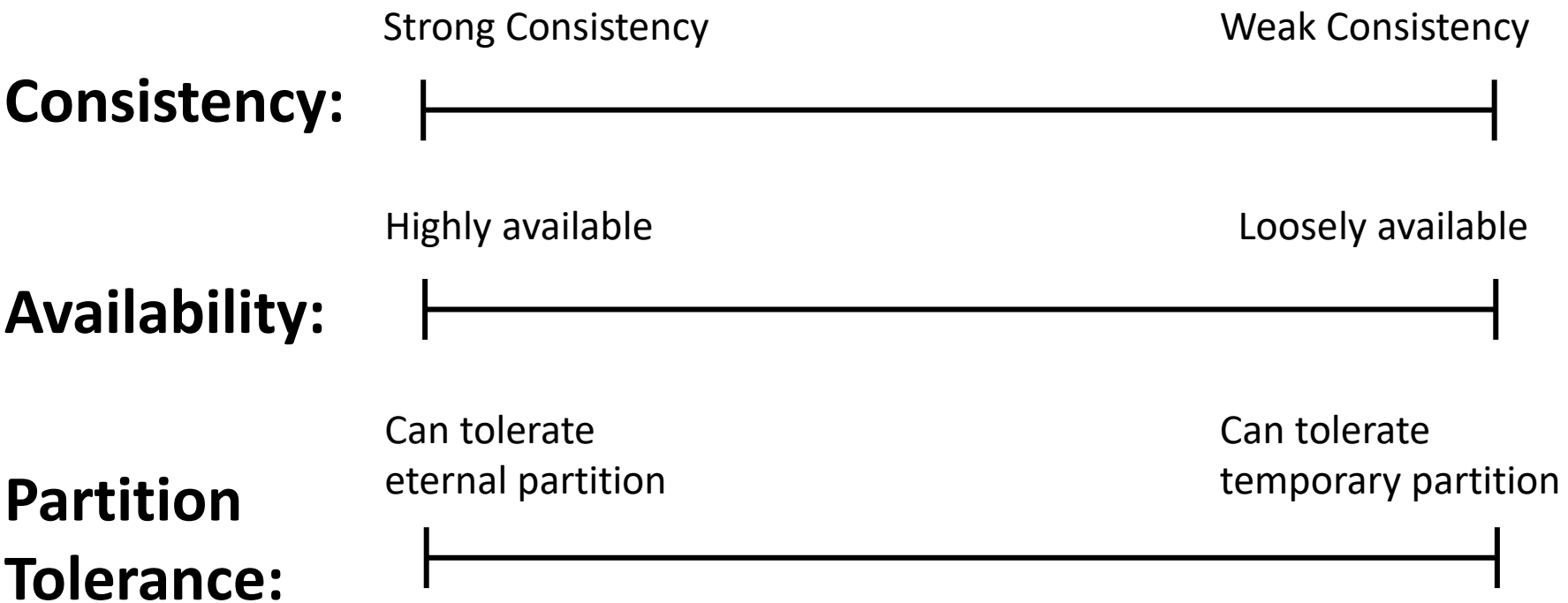


Criticism on CAP Theorem

- "CAP Twelve Years Later: How the "Rules" Have Changed" by Eric Brewer himself (2012)
- "A Critique of the CAP Theorem" by Martin Kleppmann (2015)

CAP Theorem is Misleading

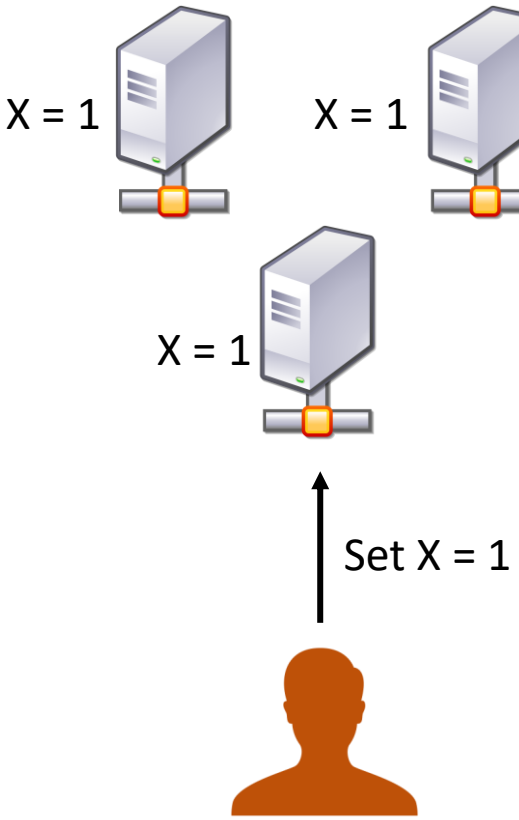
- C, A, and P should be **continuous** properties instead of **binary** properties



Strong & Weak Consistency

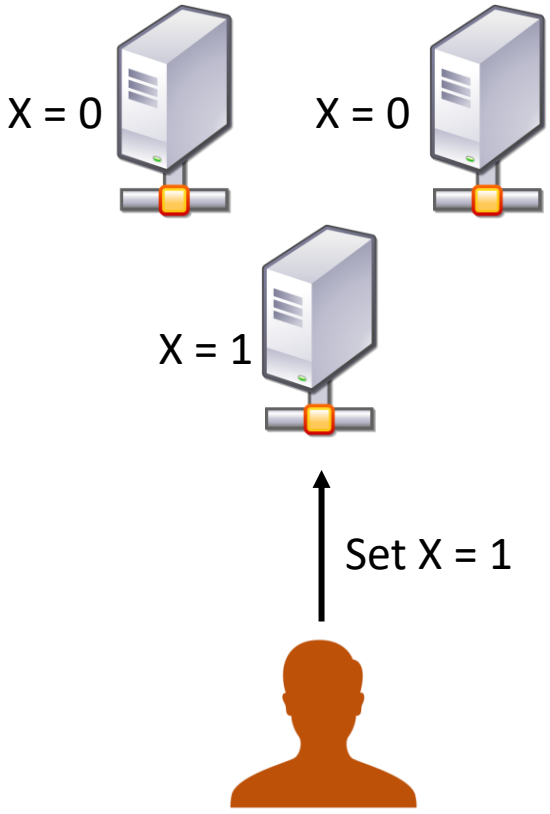
Strong consistency

(All nodes see the same states)



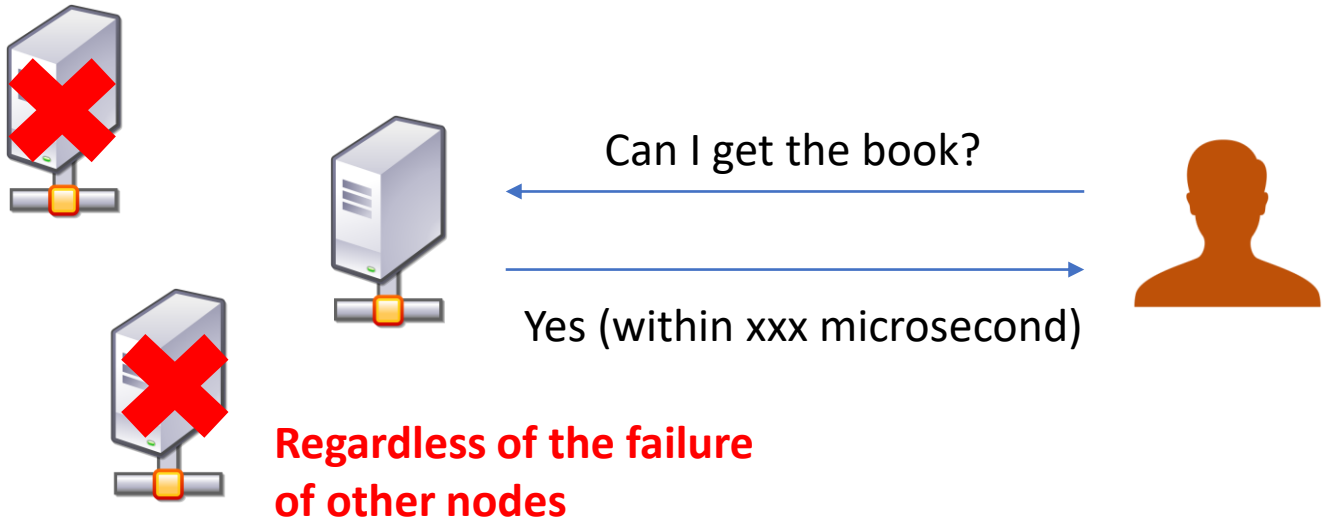
Weak consistency

(Some nodes may see different states temporarily)



Availability As Responsiveness

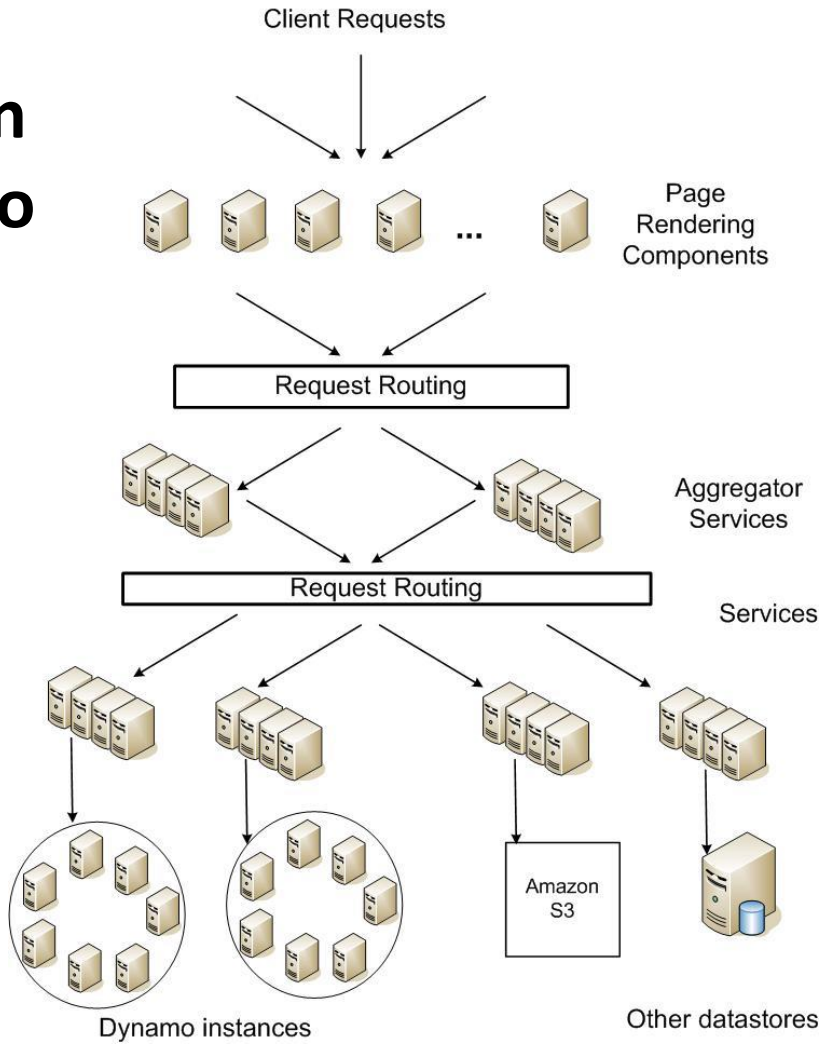
- Availability are often defined as “responsiveness” of the system



Availability As Responsiveness

Example:

Amazon Dynamo



From receiving a client request to generating a response:

99.9% of requests receive responses within 300ms

➔ **Service Level Agreement (SLA)**

Tolerance for Network Failure

- Define P as the tolerance for network failure
  - **Network latency:**
    How faster do you require the network packet to arrive? Seconds? Minutes? Hours?
  - **Packet lost:**
    How reliable do you expect the network to send your packets? 100%? 95%? 50%?
  - **Complete vs partial disconnection:**
    Do you expect a part to be complete cut off from the system? Or some node can still be connected?