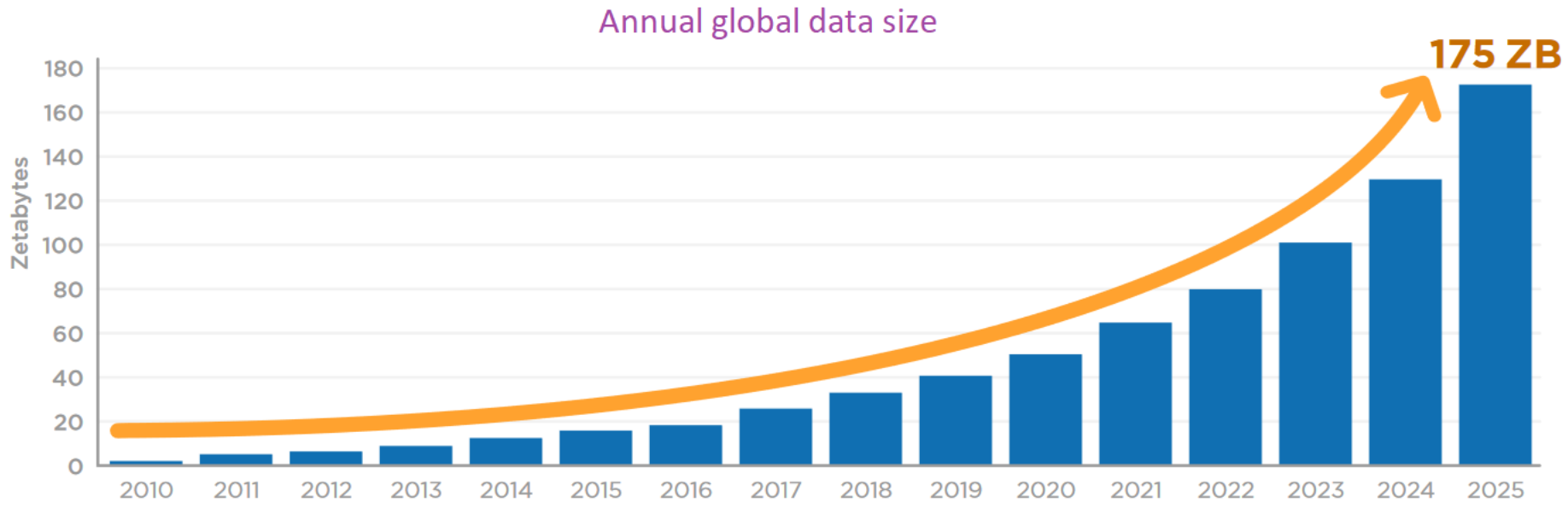

2CEIT702
BIG DATA
ANALYTICS

What is Data?
Difference Between Data and Information

DATA	**INFORMATION**
Data is defined as unstructured information such as text, observations, images, symbols, and descriptions. In other words, data provides no specific function and has no meaning on its own.	Information refers to processed, organized, and structured data. It gives context for the facts and facilitates decision making. In other words, information is processed data that makes sense to us.

Big Data?

Big data is a term used to **describe the massive amounts of data that is being generated every day.**



Big Data Analytics:-

The process of examining large and varied data sets—referred to as big data—to uncover hidden patterns, unknown correlations, market trends, customer preferences, and other useful information. This information can help organizations make more informed business decisions.

Tools and Technologies

Hadoop: Overview and components (HDFS, MapReduce)

Spark: In-memory processing

NoSQL Databases: Examples (MongoDB, Cassandra)

Data Visualization Tools: Examples (Tableau, Power BI)

Big Data Storage

Big data storage is a scalable architecture that allows businesses to collect, manage, and analyze immense sets of data in real-time. The design of big data storage solutions is specifically tailored to address the speed, volume, and complexity of the data sets.

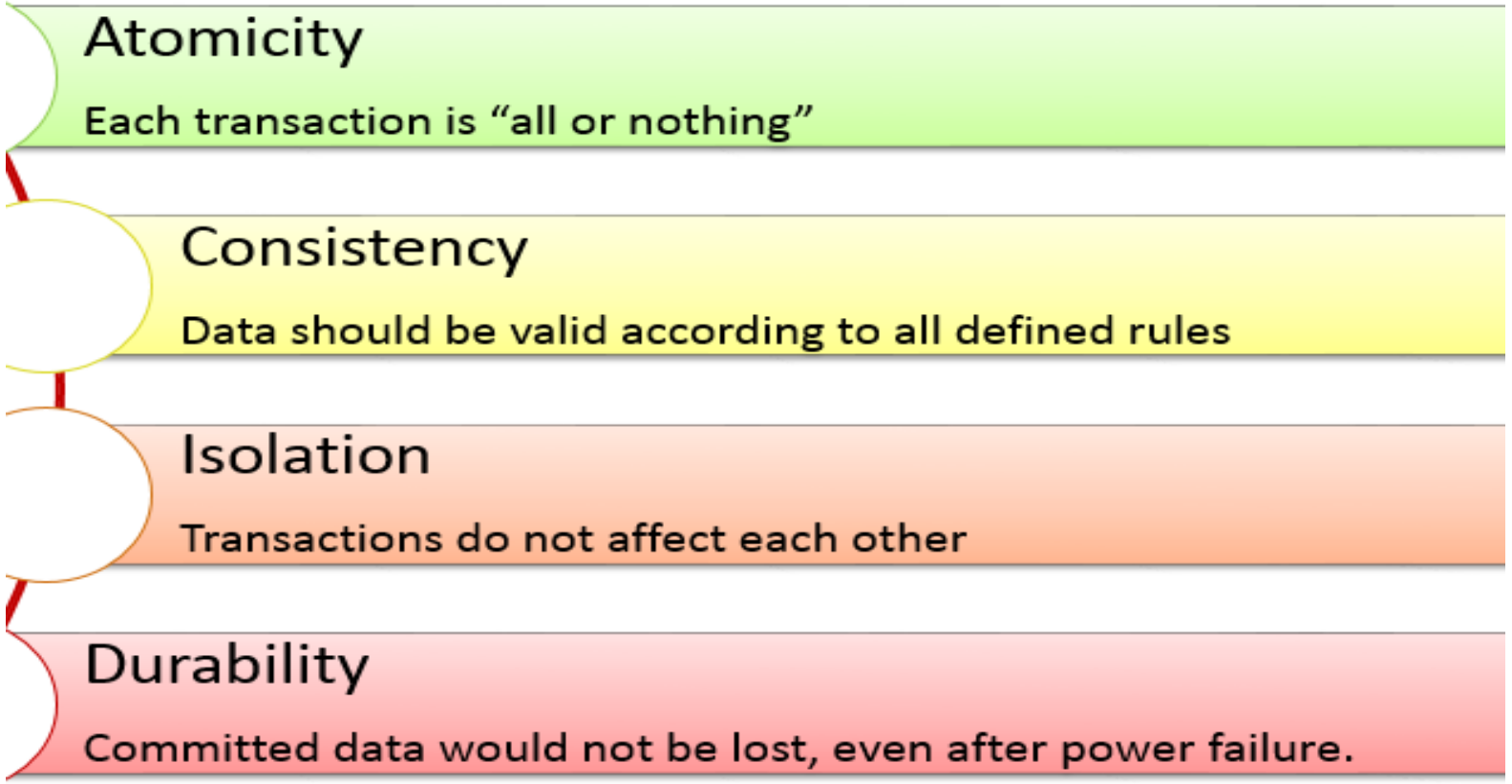
Importance of Big Data Storage

- Data Variety Management
- **Data Accessibility and Availability**
- **Analytics and Insights**
- **Scalability**
- **Disaster Recovery**

A.C.I.D. Properties

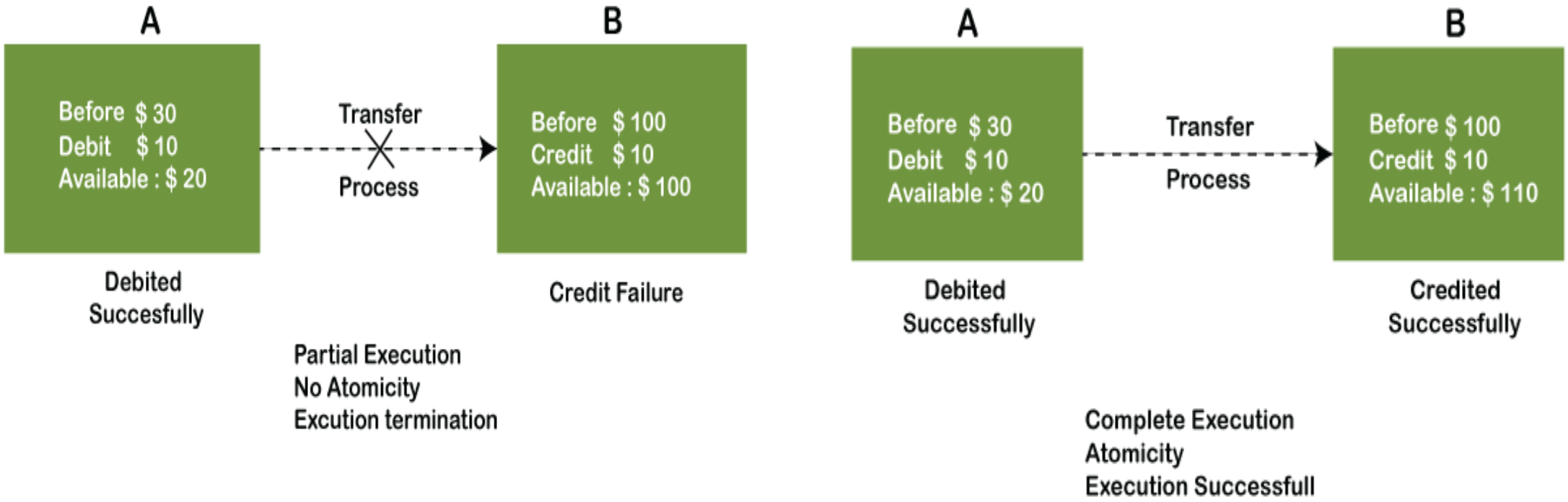
ACID properties refer to four key principles :-

- Atomicity,
- Consistency,
- Isolation, and
- Durability.



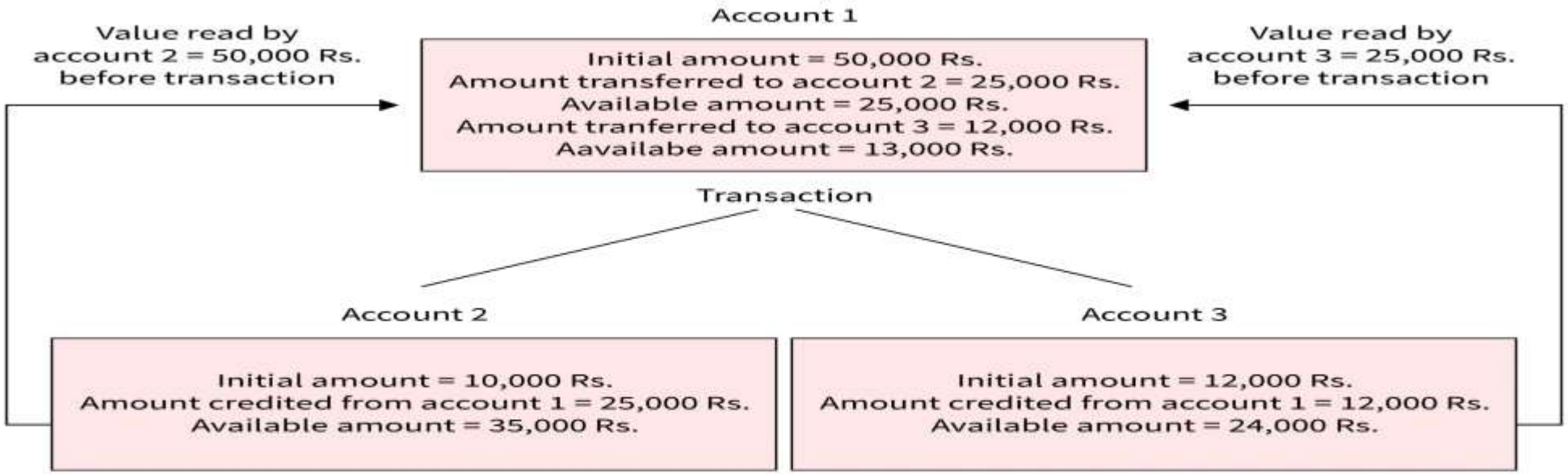
Atomicity:-

Atomicity ensures that a transaction is treated as a single indivisible unit, either executing all its operations or none at all.



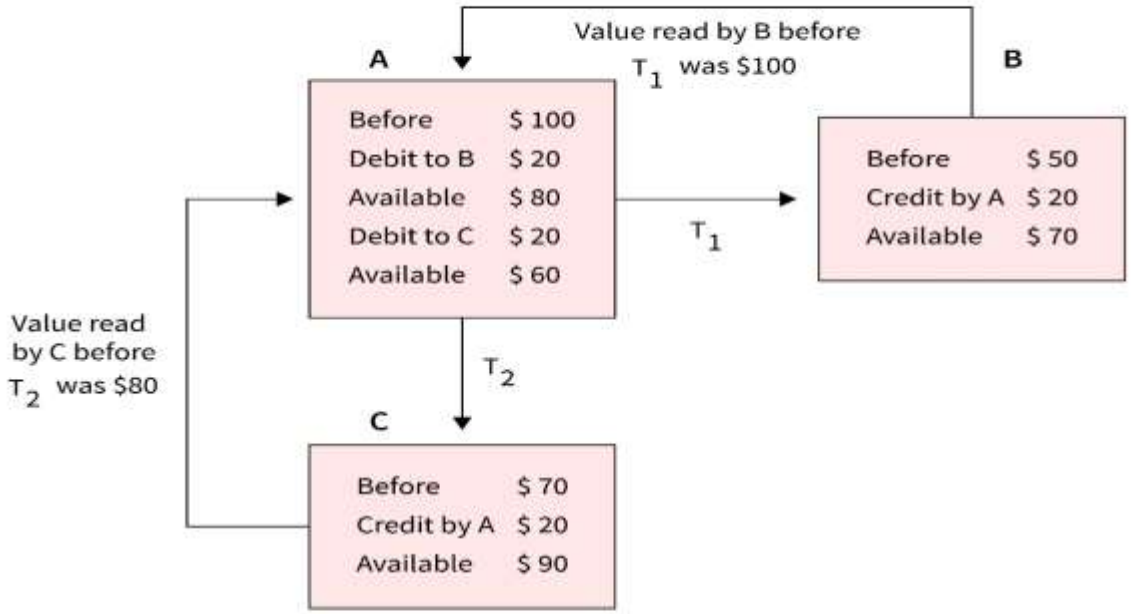
Consistency

Consistency ensures that the database remains in a valid state before and after a transaction.



Isolation:-

Isolation ensures the occurrence of multiple transactions concurrently without a database state leading to a state of inconsistency.



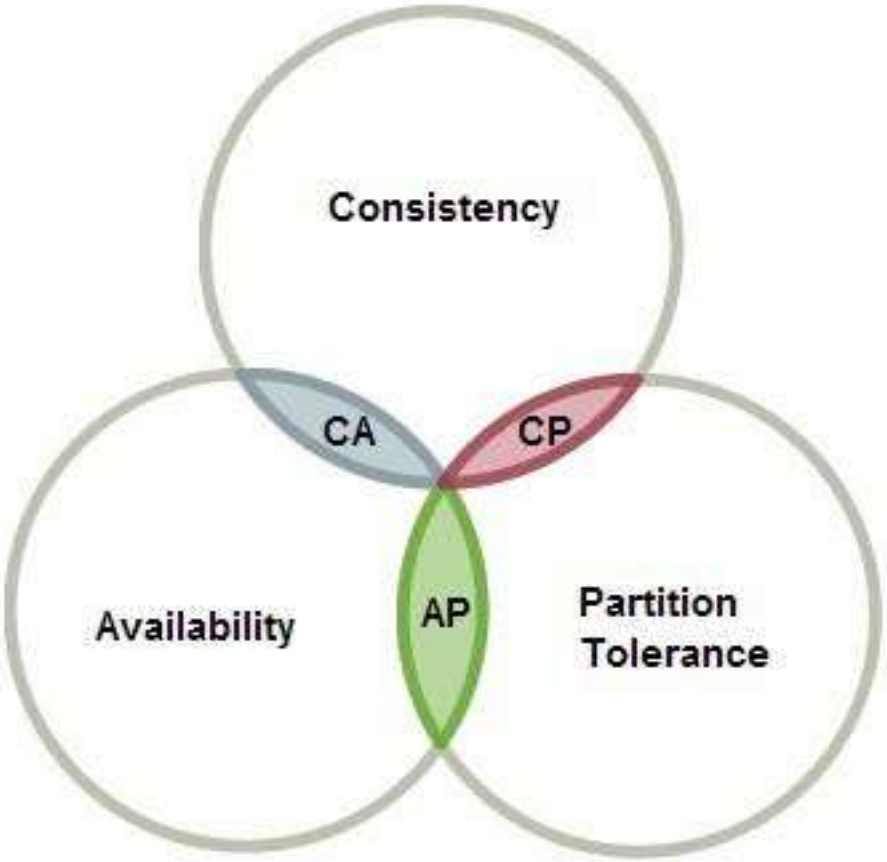
Durability:-

- The durability property states that once the execution of a transaction is completed, the modifications and updates on the database gets written on and stored in the disk.
- These persist even after the occurrence of a system failure. Such updates become permanent and get stored in non-volatile memory. Thus, the effects of this transaction are never lost.

CAP THEOREM (BREWER'S THEOREM)

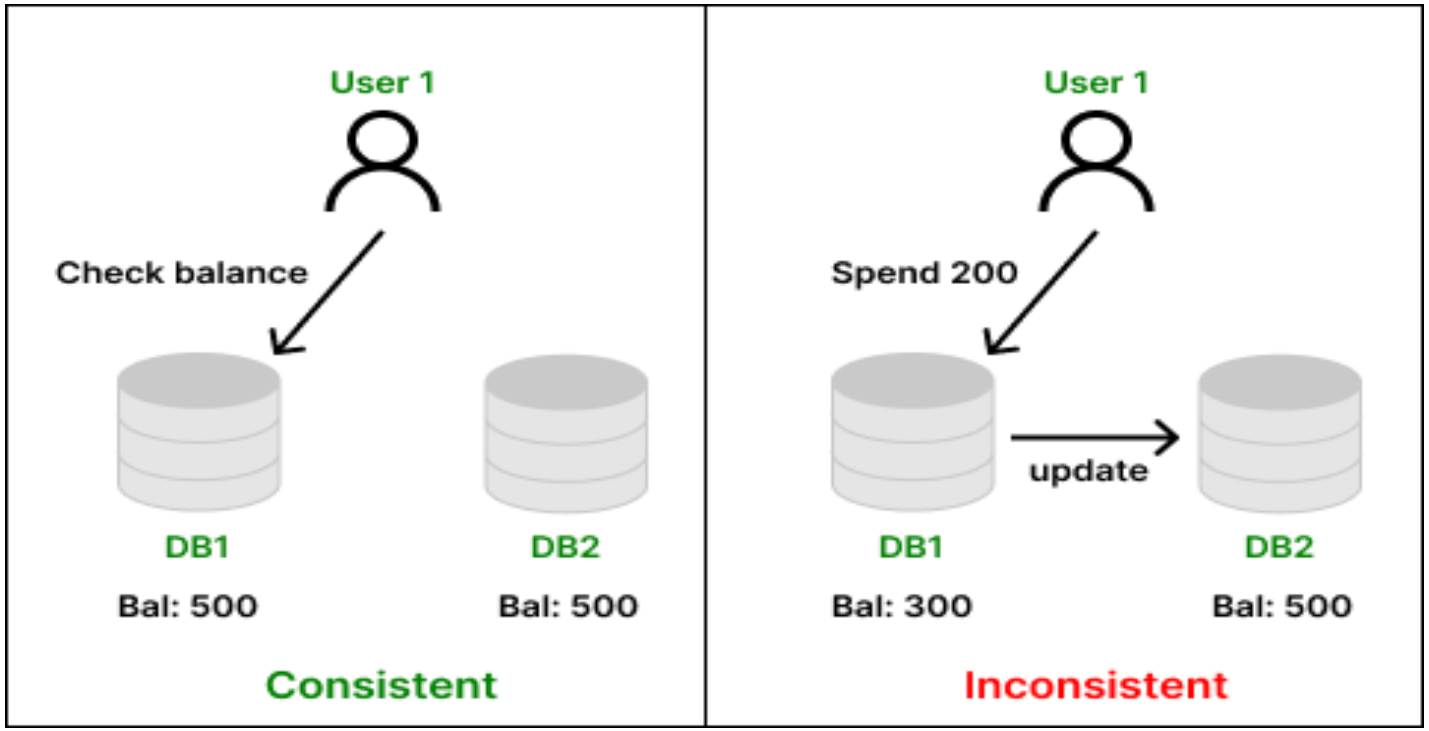
The **CAP theorem**, also named **Brewer's theorem** after computer scientist Eric Brewer, states that any distributed data store can provide only two of the following three guarantees:

- Consistency
- Availability
- Partition Tolerance



Consistency in CAP Theorem

- Consistency means that all the nodes (databases) inside a network will have the same copies of a replicated data item visible for various transactions.
- It guarantees that every node in a distributed cluster returns the same, most recent and a successful write.
- It refers to every client having the same view of the data.



Availability in CAP Theorem

- Availability means that each read or write request for a data item will either be processed successfully or will receive a message that the operation cannot be completed.
- Every non-failing node returns a response for all the read and write requests in a reasonable amount of time.

Partition Tolerance in CAP Theorem

- A Partition is a communications break within a distributed system—a lost or temporarily delayed connection between two nodes. Partition tolerance means that the cluster must continue to work despite any number of communication breakdowns between nodes in the system.

1. **Consistency and Partition Tolerance (CP):**

In this case, the system would prioritize consistency and partition tolerance.

Consistency: If a user posts a new status update, all nodes in the distributed database will reflect this update immediately. All users querying the system will see this update as soon as it is written.

Partition Tolerance: Even if there's a network partition or some nodes become unreachable, the system will continue to function and provide consistent data to the users. However, the system might not be able to serve all requests immediately if some nodes are down.

2. **Availability and Partition Tolerance (AP):**

Alternatively, the system might prioritize availability and partition tolerance. This is common in scenarios where the system needs to be always operational.

Availability: Even if a network partition occurs, the system will respond to requests. Some nodes might serve older data if they haven’t been updated yet, but the system as a whole remains operational.

Partition Tolerance: The system will still work even if some nodes cannot communicate with others due to network issues. Data consistency might be sacrificed temporarily to ensure that all users can access the system.

3. **Consistency and Availability (CA):**

Finally, consider a system that prioritizes consistency and availability but not partition tolerance. This setup is less common in practice because network partitions are almost inevitable in a distributed environment.

Consistency: All nodes have the most recent data, and any updates are immediately visible across the system.

Availability: Every request gets a response with the most recent data.

However, if a network partition occurs, the system might become unavailable because it can't ensure consistency across all nodes while handling the partition.