According to the CAP theorem, a distributed database system can only guarantee two of these three properties: consistency, availability, and partition tolerance. (the letters 'C,' 'A,' and 'P' in CAP). The letters CAP stand for three desired qualities of distributed systems with replicated data. Consistency: When all nodes view the same data at the same time, the system is said to be consistent. Simply put, if we conduct a read operation on a consistent system, the value of the most recent write operation should be returned. This indicates that the read should cause all nodes to provide the same data, namely the most recent write's value. Availability: In a distributed system, availability assures that the system is always operational.

Regardless of a node's particular status, every request receives a (non-error) response. Partition Tolerance: This condition specifies that the system will not fail if messages are lost or delayed across system nodes. In distributed systems, partition tolerance has become more of a must than a choice. It is made feasible by sufficiently replicating records across node and network combinations.

The CAP Theorem was initially proposed by Brewer in the context of a web service. A web service is deployed by a collection of servers, which may be scattered over a number of geographically dispersed data centers. Clients make service requests. When a server gets a service request, it provides a response. It's worth noting that such a broad definition of a web service may encompass a wide range of applications, including search engines, e-commerce, online music services, and cloud-based data storage.

However, another fascinating option that appears regularly in reality is segmenting a bigger system into multiple subsystems, each of which may select a different trade-off. From the standpoint of the overall system, the software architects may appear to have compromised both consistency and availability! Nonetheless, the resultant architecture frequently results in a system that answers effectively to most user requests, even in poor network conditions, and also delivers high levels of consistency when consistency is necessary. The presented CAP theorem is utilized to manage future scalable systems, tolerating assaults, and handling millions of sensor activities in smart city mobile systems. In a summary, this paper summarizes the history of the CAP theorem as well as its practical applications applied to the real world problems.