**Key Concepts**

**CAP theorem**

**CAP -** **PT is must or inevitable.** Then do trade-off between A & C. e.g. Consistency need in ticket, trading, airline seat booking and Availability is needed in social, streaming service.

**Strong consistency** can implement using - distributed transaction (cache & database), single node database and will have higher latency.

**Example tools – SQL, NO SQL with strong consistency mode (DynamoDB)**

**Availability** can implement using **more multiple replicas, CDC (Change Data Capture) and eventual consistency**

**Example tools – DynamoDB in multi-AZ mode, Casandra**

**Type of consistency** – Strong consistency, Casual consistency, Read-your-writes, eventual consistency

Partition tolerance means a distributed system continues to function correctly even if network communication breaks between some of its nodes. A network partition is a loss of communication, which can divide the system into isolated groups. Systems that are partition tolerant must make a trade-off between consistency and availability during a partition, as they cannot guarantee both when nodes cannot communicate with each other.

Different part of system can have different requirement.

**Ticketmaster**

* Event CRUD must satisfy availability property
* Event booking must satisfy consistency property

**Key aspects of partition tolerance**

* [**Network Partition**](https://www.google.com/search?q=Network+Partition&sca_esv=b46502feac7af8df&sxsrf=AE3TifNA3tw-QgLHEaTVEvcrIgDzK2EmxQ%3A1764666163044&ei=M6suaeOvAsyYjuMPx8KpmQU&oq=partion+tolerance+&gs_lp=-QEY9AMY9QMY9gPYAQGYAwC6BgYIARABGBOSBwQxNS4zoAfwrwGyBwQxNS4zuAfODMIHCDAuNS4xMi4xyAdX&sclient=gws-wiz-serp&mstk=AUtExfAZbQepNJfGbA8NzGjdc4s9Zq9vbyrGoVqkPovTwXGG18QATk_pyFCepNZRD2-Yo4HvCLg6DDnBCCUnsnG9Td01g0yeeOwEZ8WtRJJlHDnjLpwquEWKiNJG5q4Qcx7SNwsheam3DmIioGEjGjSvgKrjHjmP_2C96P2lQnU5XH3xdP8&csui=3&ved=2ahUKEwijjOu-k5-RAxWeV2wGHVMdL7EQgK4QegQIBBAB)**:** This is a communication failure where nodes cannot talk to each other. This can happen for many reasons, such as network congestion, router failures, or data center outages. It's important to note that the nodes themselves are still running, but they are isolated from one another.
* **Continued Operation:** The core of partition tolerance is that the system as a whole keeps working, at least to some extent, even when parts of it are disconnected.
* **The**[**CAP Theorem**](https://www.google.com/search?q=CAP+Theorem&sca_esv=b46502feac7af8df&sxsrf=AE3TifNA3tw-QgLHEaTVEvcrIgDzK2EmxQ%3A1764666163044&ei=M6suaeOvAsyYjuMPx8KpmQU&oq=partion+tolerance+&gs_lp=-QEY9AMY9QMY9gPYAQGYAwC6BgYIARABGBOSBwQxNS4zoAfwrwGyBwQxNS4zuAfODMIHCDAuNS4xMi4xyAdX&sclient=gws-wiz-serp&mstk=AUtExfAZbQepNJfGbA8NzGjdc4s9Zq9vbyrGoVqkPovTwXGG18QATk_pyFCepNZRD2-Yo4HvCLg6DDnBCCUnsnG9Td01g0yeeOwEZ8WtRJJlHDnjLpwquEWKiNJG5q4Qcx7SNwsheam3DmIioGEjGjSvgKrjHjmP_2C96P2lQnU5XH3xdP8&csui=3&ved=2ahUKEwijjOu-k5-RAxWeV2wGHVMdL7EQgK4QegQIBBAE)**Trade-off:** The CAP theorem states that a distributed system can only guarantee two of the following three properties at any given time: Consistency, Availability, and Partition Tolerance. Since network partitions are inevitable in distributed systems, they must be partition tolerant to some degree. This means during a partition, the system must choose between:
  + **Consistency:** Ensuring all nodes have the same data, which may require temporarily shutting down parts of the system (making it unavailable) to prevent conflicting updates.
  + **Availability:** Allowing the system to remain responsive, even if it means some nodes may return inconsistent or stale data until the network partition is resolved.