CAP theorem:

The **CAP theorem**, also known as Brewer's theorem, was introduced by Eric Brewer in 1998 as a conjecture. In 2002, it was proven as a theorem by Seth Gilbert and Nancy Lynch. The theorem states that any distributed system cannot have consistency, availability, and partition tolerance simultaneously:

- **Consistency** is a property that ensures that all nodes in a distributed system have a single, current, and identical copyof the data.

  Consistency is achieved using consensus algorithms in order to ensure that all nodes have the same copy of the data. This is also called **state machine replication**. The blockchain is a means for achieving state machine replication.

- **Availability** means that the nodes in the system are up, accessible for use, and are accepting incoming requests and responding with data without any failures as and when required. In other words, data is available at each node and the nodes are responding...

The CAP theorem states that **a distributed database system has to make a tradeoff between Consistency and Availability when a Partition occurs**. A distributed database system is bound to have partitions in a real-world systemdue to network failure or some other reason.

The CAP Theorem is comprised of three components (hence its name) as they relate to distributed data stores:

Consistency. All reads receive the most recent write or an error.

Availability. All reads contain data, but it might not be the most recent.

Partition tolerance.

The CAP Theorem is comprised ofthree components (hence its name) as theyrelate to distributed data stores:

- **Consistency.** Allreads receive the most recent write or an error.
- **Availability.** Allreads contain data, but it might not be the most recent.
- **Partition tolerance.** The system continues to operate despite network failures (ie; dropped partitions, slow network connections, or unavailable network connections between nodes.)

In normal operations, your data store provides all three functions. But the CAP theorem maintains that when a distributed database experiences a network failure, you can provide either consistency or availability.

It's a tradeoff. All other times, all three can be provided. But, in the event of a network failure, a choice must be made.In the theorem, partition tolerance is a must. The assumption is that the system operates on a distributed data store so the system, by nature, operates with network partitions. Network failures will happen, so to offer any kind ofreliable service, partition tolerance is necessary—the P of CAP.
That leaves a decision between the other two, C and A. When a network failure happens, one can choose to guarantee consistency or availability:

- High consistency comes at the cost of lower availability.
- High availability comes at the cost of lower consistency.

Benefits and limitations of blockchain:

Numerous benefits of blockchain technology are being discussed in the industry and proposed by thought leaders around the world in blockchain space. The top 10 benefits are listed and discussed as follows.

Decentralization :

This is a core concept and benefit of blockchain. There is no need for a trusted third partyor intermediary to validate transactions; instead a consensus mechanism is used to agree onthe validity of transactions.

Transparency and trust :

As blockchains are shared and everyone can see what is on the blockchain, this allows the system to be transparent and as a result trust is established. This is more relevant in scenarios such as the disbursement of funds or benefits where personal discretion should be restricted.

Immutability:

Once the data has been written to the blockchain, it is extremely difficult to change it back. It is not truly immutable but, due to the fact that changing data is extremely difficult and almost impossible, this is seen as a benefit to maintaining an immutable ledger of transactions.

High availability:

As the system is based on thousands of nodes in a peer-to-peer network, and the data is replicated and updated on each and every node, the system becomes highly available. Even if nodes leave the network or become inaccessible, the network as a whole continues to work, thus making it highly available.

Highly secure:

Alltransactions on a blockchain are cryptographically secured and provide integrity.

Simplification of current paradigms:

The current model in many industries such as finance or health is rather disorganized, wherein multiple entities maintain their own databases and data sharing can become very difficult due to the disparate nature of the systems. But as a blockchain can serve as a single shared ledger among interested parties, this can result in simplifying this model by reducing the complexity of managing the separate systems maintained by each entity.

Faster dealings:

In the financial industry, especially in post-trade settlement functions, blockchain can play a vital role by allowing the quicker settlement of trades as it does not require a lengthy process of verification, reconciliation, and clearance because a single version of agreed upon data is already available on a shared ledger between financial organizations.

Cost saving:

As no third party or clearing houses are required in the blockchain model, this can massively eliminate overhead costs in the form of fees that are paid to clearing houses or trusted third parties.

Decentralization:

Decentralization is a core benefit and service provided by blockchain technology. By design, blockchain is a perfect vehicle for providing a platform that does not need any intermediaries and that can function with many different leaders chosen via consensus mechanisms. This model allows anyone to compete to become the decision-making authority. A consensus mechanism governs this competition, and the most famous method is known as **Proof of Work** (**PoW**).

Decentralization is applied in varying degrees from a semi-decentralized model to a fully decentralized one depending on the requirements and circumstances. Decentralization can be viewed from a blockchain perspective as a mechanism that provides a way to remodel existing applications and paradigms, or to build new applications, to give full control to users.

Information and communication technology (**ICT**) has conventionally been based on a centralized paradigm whereby database or application servers are under the control of a central authority, such as a system administrator. With Bitcoin and the advent of blockchain technology, this model has changed, and now the technology exists to allow anyone to start a decentralized system and operate it with no single point of failure or single trusted authority. It can either be run autonomously or by requiring some human intervention, depending on the type and modelof governance used in the decentralized application running on the blockchain.

The following diagram shows the different types of systems that currently exist: central, distributed, and decentralized.

Different types of networks/systems

Centralized systems are conventional (client-server) IT systems in which there is a single authority that controls the system, and who is solely in charge of all operations on the system. All users of a centralized system are dependent on a single source of service. The majority of online service providers, including Google, Amazon, eBay, and Apple's App Store, use this conventional model to deliver services.

In a **distributed system**, data and computation are spread across multiple nodes in the network. Sometimes, this term is confused with *parallel computing*. While there is some overlap in the definition, the main difference between these systems is that in a parallel computing system, computation is performed by all nodes simultaneously in order to achieve the result; for example, parallel computing platforms are used in weather research and forecasting, simulation, and financial modeling. On the other hand, in a distributed system, computation may not happen in parallel and data is replicated across multiple nodes that users view as a single, coherent system. Variations of both of these models are used to achieve fault tolerance and speed. In the parallel system model, there is still a central authority that has control over all nodes and governs processing. This means that the system is still centralized in nature.

The critical difference between a decentralized system and distributed system is that in a distributed system, there is still a central authority that governs the entire system, whereas in a decentralized system, no such authority exists.

A **decentralized system** is a type of network where nodes are not dependent on a single master node; instead, control is distributed among many nodes. This is analogous to a model where each department in an organization is in charge of its own database server, thus taking away the power from the central server and distributing it to the sub-departments, who manage their own databases.

A significant innovation in the decentralized paradigm that has given rise to this new era of decentralization of applications is **decentralized consensus**. This mechanism came into play with Bitcoin, and it enables a user to agree on something via a consensus algorithm without the need for a central, trusted third party, intermediary, or service provider.

We can also now view the different types of networks shown earlier from a different perspective, where we highlight the controlling authority of these networks as a symbolic hand, as shown in the following diagram. This model provides a clearer understanding of the differences between these networks from a decentralization point of view,

Different types of networks/systems depicting decentralization from a modern perspective

In the middle we have distributed systems, where we still have a central controller but the system comprises many dispersed nodes. Onthe right-hand side, notice that there is no hand/controller controlling the networks.

This is the key difference between decentralized and distributed networks. A decentralized system may look like a distributed system from a topological point of view, but it doesn't have a central authority that controls the network.

A traditional distributed system comprises many servers performing different roles

The following diagram shows a decentralized system (based on blockchain) where an exact replica of the applications and data is maintained across the entire network on each participating node:

A comparison between centralized and decentralized systems (networks/applications) is shown in the following table:

Feature	Centralized	Decentralized
Ownership	Service provider	All users
Architecture	Client/server	Distributed, different topologies
Security	Basic	More secure
High availability	No	Yes
Fault tolerance	Basic, single point of failure	Highlytolerant, as service is replicated
Collusion resistance	Basic, because it's under the control of a group or even single individual	Highly resistant, as consensus algorithms ensure defense against adversaries
Application architecture	Single application	Application replicated across all nodes on the network
Trust	Consumers have to trust the service provider	No mutualtrust required
Cost for consumer	Higher	Lower

The comparison in the table only covers some main features and is not an exhaustive list ofall features. There may be other features of interest that can be compared too, but this list should provide a good level of comparison.