# Centralized and De-centralised Application

### Centralized Application:

The centralized and decentralized applications are characterized based on ownership of the application. In the case of centralized applications, the central authority/group/business controls the entire application. Therefore, the flow of information will always be through a single server or cluster of servers.
The dedicated server contains all the required logic to execute an application. Here flow data/information always moves from a dedicated single source (server), which could be a single machine or a single cluster of machines.

**Advantages** -

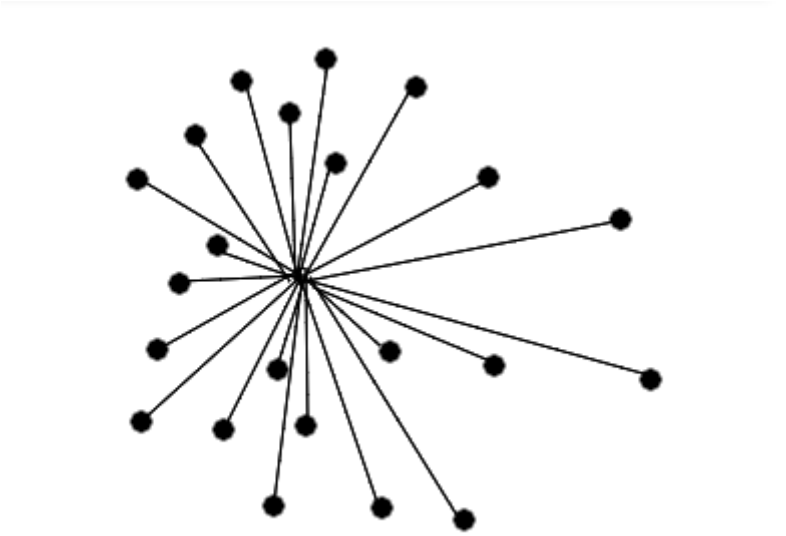
1. A clear chain of commands and control- ease of structure and processes.
2. Reduced cost - as there is the involvement of a single system or single cluster of machines.
3. Quick implementation or installation.
4. Easy bug detection, data sharing and maintenance.
5. Flexibility for quick updates.

**Disadvantages** -

1. A single point of failure may destroy the system.
2. System overload is a problem.
3. Difficult server maintenance.
4. It is prone to denial-of-service attacks, which gives rise to security concerns.
5. It may have high latency, which could result in long delays in work.
6. Low computing power or lesser performance.
7. Highly dependent on the network.
8. Less possibility of data backup - if the server node fails and there is no backup, you load the data instantaneously.

E.g., Facebook - all the information displayed, working and availability of the application are decided by the central authority, which is driven by Facebook.

**Fig: Centralized Applications**



## De-Centralized Application :

It doesn't have a single or central control but possesses distributed authority/control. We can say that they run on the Peer to Peer (P2P) or blockchain networks of computers rather than centralized servers. Therefore, more rich/effective or fast content will be delivered if there are more consumers.
Eg. - Blockchain or Bittorrect - No single authority as the application's consumers drive the application.  Therefore we can say that the more the consumer, the richer will be the content delivered.
Advantages -

1. Less downtime - due to availability of distributed control.
2. More secure - As these will be less prone to a denial-of-service attack.
3. Data integrity.
4. Redundancy
5. Availability and Reliability and Great control.
6. Low latency and high throughput.
7. High computing power

Disadvantages -

1. Difficult to produce global results.
2. Updated may take a lot of time.

There are a few challenges in centralized systems, in case of availability (chances of crashing increases), integrity, and security, which are overcome by decentralized systems.

**Note -**

- In the case of monolithic and distributed architectures, we are talking from the perspective of how the application was built, we are talking with the perspective of architecture, but in the case of centralized/decentralized, it's from the perspective of ownership or control.
- Monolithic architecture has tightly coupled Front End, Back End and Data storage layers. So, Given a monolithic architecture, it can operate on both centralized (single authority) and decentralized systems (distributed authority).
  Eg- Centralized Monolithic - Facebook and LinkedIn at an early time.
  Centralized Distributed - Facebook and LinkedIn at the current time.
- Similarly, in the case of distributed systems, if there is central control then it will be a centralized distributed system and if there is distributed control, then it will be called a decentralized distributed system.
  Eg - Decentralized Monolithic - BitTorrent
  Decentralized Distributed - Blockchain

**Fig: De-centralized Applications**