Summarize your understanding of what distributed systems are, why they're necessary and their characteristics which distinguish them from other systems.

Distributed systems are typically multiple CPUs connected over a network, where the program execution is performed by sharing resources over multiple systems. They are networked computers communicating and coordinating their actions only by passing messages.

They are necessary because the prime motivation for constructing and using distributed systems stems from a desire to share resources. The specific characteristics which make them necessary are:

- **Sharing of Computing Power**:
    - With distributed architecture, the systems can be run in parallel there by sharing the computing power in completing a task.
- **Resource sharing**:
    - Resources can be shared along with my computing power and this decentralized resource sharing capabilities allows the system to provide high availability resources.
- **Economical**:
    - It is economical to maintain cluster of servers to perform tasks rather than one huge server.
- **Reliability**:
    - A system in a distributed architecture provides continuous availability and data consistency.
- **Scalability**:
    - Using distributed systems, it is easier to take advantage of all the resources on all machines without the need to change the configuration.
- **Interoperability**:
    - Due to easier integration with other systems, distributed systems can have two different systems working together.

Some characteristics which also distinguish distributed system from others are:

- Complex design strategies for enabling communication between processes.
- Network availability is essential.
- Higher exposure- higher the security risk.

Apart from those characteristics, distributed systems have different design choices of development, and they are:

- No individual machine will have information about the entire state. This design helps it avoid bottleneck when compared to centralized systems.
- Makes decision based on local information
    - Checks if the system has all information to service a request.
    - Example of Load balancer: Partial load information or randomized algorithms help load balancer decide the server to which the request needs to be

forwarded. It doesn't need get complete information all the time to make a decision.

- No single point of failure.
- No single clock and assume always there is a clock skew.

Distributed System have evolved from centralized systems. The evolution can be tracked as follows:

- Each user had a local machine with intermittent connection to the network to fetch data.
- Later, instead of each user having a machine, group of users or organizations have a workstation and share resources requested by local machines.
- In the same line, Client-Server model evolved from workstation model dedicated to serve requests by workstations.
- Processor pool with powerful server and dumb client depending on server for every task.
- Cluster computing where a cluster is one rack of powerful servers.
- Grid computing- clusters are connected over a wide area network.
- Distributed Pervasive System:
    - Where each IoT device is a server/client within the distributed system.

Distributed systems also provide higher levels of transparency. They abstract the information to expose to user by design choice of what aspects are visible and what aspects are not visible. In other words, hide the data from user rather than expose as it makes the system easier to work together and develop, for the user and the designer. Also, distributed systems provide replication transparency among several. As an example, Search Engines hides the server which is going to serve the query and only provides the global interface to user rather than asking him to choose the server from the replicas. Also, distributed systems provide failure transparency. That is, when a server fails in the global system, the query is transmitted to another server due to the configuration. When this happens in the background, the user doesn't know about the failure.