1. **Distributed system** refers to the system that its **components** locate at **networked computers**, communicate and coordinate their actions by passing only **messages**
2. **The characteristics of distributed systems:**

Concurrency of components

Concurrency is a property of a system representing the fact that multiple activities are executed at the same time. The concurrent execution of activities takes place in different components running on multiple machines as part of a distributed system. In addition, these activities may perform some kind of interactions among them. Concurrency reduces the latency and increases the throughput of the distributed system.

Lack of a global clock: The **accuracy** that computers in a network can synchronize their **clocks** is very **limited** – there is **no** exact global notion of the **correct time**.

Independent failures of components

Independent failures: **The components of a distributed system running on different computers can continue to execute independent of each other**.

1. **The challenges of distributed systems include:**

* Heterogeneity of components (ability of the system to operate on a variety of different hardware and software components)

Heterogeneity refers to the differences that arise in networks, programming languages, hardware, operating systems and differences in software implementation.

* Openness (components to be added or replaced)

The openness of distributed systems refers to the system's extensability and ability to be reimplemented. More specifically, the openness of a distributed system can be measured by three characteristics: interoperability, portability, and extensability as we previously mentioned. Interoperability refers to the system's ability to effectively interchange information between computers by standardization, portability refers to the system's ability to properly function on different operating systems, and extensability allows developers to freely add new features or easily reimplement existing ones without impairing functionality

* Security

These concepts are related, availability regards the authenticated and authorized users, integrity protects through encryption and other methods, and confidentiality ensures that resources are not needlessly disclosed or made available.

Security is especially important in distributive systems due to their association with sensitive and private data. Take payment and transactions information

* Scalability (the ability to work well when the load increases)

A program is scalable if a program does not need to be redesigned to ensure stability and consistent performance as its workload increases.

* Failure handling

Failures, like in any program, are a major problem. However, in distributive systems, with so many processes and users, the consequences of failures are exacerbated. Additionally, many problems will arise due to the nature of distributive systems. Unexpected edge cases may present themselves in which the system is ill-equipped for, but developers must account for. Failures can occur in the software, hardware, and in the network; additionally the failure can be partial, causing some components to function and other to not. However, the most important part in failure handling is recognizing that not every failure can be accounted for.

* Concurrency of components

This discusses the shared access of resources which must be made available to the correct processes. Problems may arise when multiple processes attempt to access the same resources at the same time, thus steps need to be taken to ensure that any manipulation in the system remains in a stable state; however the illusion of simultaneous execution should be preserved. We refer to these preventative measure as concurrency control. Concurrency control should be implemented to ensure that processes are executed in a synchronous manner.

* Transparency

A problem with transparency may arise with distributed systems due to the nature of the system's complexity. In this context, transparency refers to the distributed system's ability to conceal its complexity and give off the apperance of a single system. And when we discuss transparency, we must also discuss to what extent.

* Quality of service (QoS)

1. **Disadvantages of Distributed Systems**

Some issues occur while using distributed Systems. Important ones are:

* **Security issue**

Security issues usually occur in many software and hardware devices. The same case is with Distributed Systems. Such security risks occur as a result of many nodes and connections in an open system setting that makes it difficult to ensure adequate security.

* **High set-up cost**

The initial cost of installation and set-up is high due to many hardware and software devices. There are other maintenance costs associated with the system which adds to the total cost, making it even more expensive.

* **Data loss**

There can be instances when the data sent from one node to another node can be lost midway in its journey from the source node to the destination node. This results in some important messages being lost, which is detrimental to a company.

* **Difficult to handle**

The hardware and software of a distributed system are quite complex. It’s complicated to maintain and operate the hardware components for companies. Also, software complexity makes it necessary for companies to pay special attention to the software components.

* **Overloading issue**

The Overloading issue can occur in the system if all the nodes of the distributed system try to send data at one particular instant of time.

1. **Transparency**

Transparency “is the concealment from the user of the separation of components of a distributed system so that the system is perceived as a whole”. Transparency in distributed systems is applied at several aspects such as :

**Access Transparency** – Local and Remote access to the resources should be done with same efforts and operations. It enables local and remote objects to be accessed using identical operations.

**Location transparency** – User should not be aware of the location of resources. Wherever is the location of resource it should be made available to him as and when required.

**Migration transparency** – It is the ability to move resources without changing their names.

**Replication Transparency** – In distributed systems to achieve fault tolerance, replicas of resources are maintained. The Replication transparency ensures that users cannot tell how many copies exist.

**Concurrency Transparency** – As in distributed system multiple users work concurrently, the resource sharing should happen automatically without the awareness of concurrent execution by multiple users.

**Failure Transparency** – Users should be concealed from partial failures. The system should cope up with partial failures without the users awareness.

**Parallelism transparency** - Activities can happen in parallel without users knowing

**Performance Transparency:** Performance Transparency enables system reconfiguration to increase or enhance performance.

1. 1.ACPUresource can be shared in a distributed system. There areapplications that can run on multiple computers at once by sharing theCPU resources. For example, when a heavy running application needsto process, in distributed computing all the computers get theapplication from a central server and share the processing time amongthem and merge the results once all the clients are done processing.

2.Memorycan be useful to store or cache recently accessed pages inweb which helps in accessing those pages quickly when the user isrevisiting them.

3.Diskspace can also be shared among system which can be used as afile server where files can be stored. In distributing system there isconcept called Shared disk architecture in which all disks are availablefor access from all the available clusters.

4. Multiple users can access a single file that they have access. Fore xample, I can use Sharepoint to host a file so that my employees canaccess the document.

5. A web page can be shared or accessed by multiple clients. For example, anyone with internet access can access google website Types of data or software resources that can be shared are:

1. A single point of failure (SPOF) is **a system component which, upon failure, renders an entire system unavailable or unreliable**. Single points of failure should be removed to lessen the risk of service outages. **Deployment of redundant hardware and network resources in the right places can help lower and even eliminate outages**.
2. we need distributed systems when **different users or computers want to share information and** when **different users or computers want to** split up the work, coordinating their efforts to complete the job more efficiently than if a single device had been responsible for the task and also when the jobs are so massive and complex that it would be impossible for a single computer to handle them alone.