**Distrusted System:** A distributed system is a collection of independent computers that appear to the users of the system as a single computer.

Examples:

1. **Networks:** If the system as a whole looked and acted like a classical single-processor timesharing system, it would qualify as a distributed system
2. **Banks**
3. **Cloud:** collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on SLA.
4. **Cluster:** A type of parallel or distributed processing system, which consists of a collection of interconnected stand-alone computers cooperatively working together as a single, integrated computing resource.

**Networks Vs Distributed System**

* **Networks:** A media for interconnecting local and wide area computers and exchange messages based on protocols (TCP/UDP), focus on packets, routing, etc.
* **Distributed System:** existence of multiple autonomous computers is transparent, focus on application.

**Characteristic of DS:**

* **Parallel activities**
* Autonomous components executing concurrent tasks
* **Communication via message passing**
* No shared memory
* **Resource sharing**
* Printer, database, other services
* **No global state**
* No single process can have knowledge of the current global state of the system
* **No global clock**
* Only limited precision for processes to synchronize their clocks

**Global state** VS **global clock**

* Global State: No single process can have knowledge of the current global state of the system
* Clock: There is no global clock that can be used to synchronize events across all nodes in the system.

**Goals of Distributed Systems**

Connecting Users and Resources

* **Transparency:** Hide that processes & resources are physically distributed across multiple computers.
* **Openness:** ability of integration to other systems without special modification.
* Scalability:
* Enhanced Availability

**Challenges:**

1. Heterogeneity
2. Concurrency
3. **Transparency:** Distribution should be hidden from the user as much as possible
4. **Fault tolerance:** Failure of a component (partial failure) should not result in failure of the whole system
5. Security
6. **Scalability:** easy to expand and manage.
7. Reuse and Openness

**Heterogeneity at many levels:**

* **Network:** different kinds of software and hardware.
* **Operating system:** different APIs to internet.
* **Programming languages:** many different Programming Languages.
* **Data:** different representations of data (Big Indian, Small Indian).
* **Hardware:** Different clock cycles and memory capacity.
* **Data Structures:** Implementations by different developer.

**Scaling Up Vs Scaling Out:**

* Scaling up involves adding more resources to an existing server or machine to increase its capacity (e.g., CPU, memory, storage)
* Scaling out, involves adding more machines to a system to increase its capacity.

**The Characteristics of Inter-process Communication**

1. **Message Communication:**
2. **Synchronous and asynchronous communication:**
3. **Reliability:**
4. **Ordering**:

**Networking Basics**

1. Physical/Link Layer
   * Functionalities for transmission of signals representing a stream of data from one computer to another
2. Internet/Network Layer
   * Internet Protocols (IP): a packet of data to be addressed to a remote computer and delivered.
3. Transport Layer
   * Functionalities for delivering data packets to a specific process on a remote computer
   * TCP (Transmission Control Protocol)
   * UDP (User Datagram Protocol)
   * Programming Interface: Sockets
4. Applications Layer:
   * Message exchange between standard or user applications:
   * HTTP, FTP, Telnet, Skype.

**TCP Vs UDP:**

* **Transmission Control Protocol – TCP:** is a connection-oriented communication protocol that provides a reliable flow of data between two computers. (Skype)
* **User Datagram Protocol - UDP:** is a connectionless communication protocol that sends independent packets of data, from one computer to another with no guarantees about arrival or order of arrival. (Live Streaming).

**IP Vs Port:**

* **IP:** addresses are unique numerical labels used to identify and locate devices on a network.
* **Port:**  port numbers are used to identify specific communication endpoints within a device associated with a particular network service or application

**Socket**: Sockets provide an interface for programming networks at the transport layer.

**Threads VS Process:**

* **Thread:** is a lightweight unit of execution within a process that shares the same memory space and resources as the parent process.
* **Processes:** A process is an independent instance of a program that has its own memory space, resources, and state

processes and threads are units of concurrent execution within a program, where processes are isolated instances of a program with their own memory space and resources, and threads are lightweight units of execution that share the same memory space and resources as the parent process