Nigerian Army University Biu.

Faculty:

Computing

Department of:

Computer Science

Course:

Netcentric Computing (CSC-425)

Course Lecturer

Mr. AM Gimba

- **What is Net-Centric Computing?**

Net-Centric Computing (NCC) is a distributed environment where applications and data are downloaded from servers and exchanged with peers across a network. Net-centric Computing focuses on large-scale distributed computing systems and applications that communicate through open, wide-area networks like the Internet.

General examples of large-scale network-centric systems are the World-Wide Web and Computational Grids.

For several years, major changes are being brought to the world by universal networking capabilities, such as the Internet. Today's technology solutions represent the convergence of computing power, networking capability and the information, data or knowledge that forms the content of these solutions. At the center these solutions net-centric computing lies.

Net-centric computing refers to an emerging technology architecture and an evolutionary stage of client/server computing. It is a common architecture built on open standards that supports in different ways for different people to collaborate and to reach different information sources.

Net-centric computing is an interesting and remarkable discipline that moves intelligence to the edge of the network. A wide range of careers related to net-centric computing include but are not limited to,

- **Network Administrator**

Network administrators are responsible for keeping computer network up-to-date and operating as intended. It is network administrator who implements, manages and troubleshoots a networks.

- **Network Engineer**

Network Engineer is responsible for the planning, designing and implementing computer networks. Although the job titles “network engineer” and “network administrator” looks similar, a network engineer usually has more responsibilities than a network administrator.

- **Web Applications Developer**

Web Applications Developers are responsible for building, and making improvements to websites and web applications. Generally, a typical web application developer is responsible for both back-end development and front-end development. Web application developers are a type of software developer, specialize in utilizing web technologies, use to create websites and web applications.

- **What is distributed computing?**

Distributed computing is a model in which components of a software system are shared among multiple computers or nodes. Even though the software components may be spread out across multiple computers in multiple locations, they're run as one system. This is done to improve efficiency and performance. The systems on different networked computers communicate and coordinate by sending messages back and forth to achieve a defined task.

Distributed computing can increase performance, resilience and scalability, making it a common computing model in database and application design.

A distributed computer system consists of multiple software components that are on multiple computers, but run as a single system. The computers that are in a distributed system can be physically close together and connected by a local network, or they can be geographically distant and connected by a wide area network. A distributed system can consist of any number of possible configurations, such as mainframes, personal computers, workstations, minicomputers, and so on. The goal of distributed computing is to make such a network work as a single computer.

Examples of Distributed Computing

- The internet (World Wide Web) itself.
- Telecommunication networks with multiple antennas, amplifiers, and other networking devices appear as a single system to end-users.

How distributed computing works

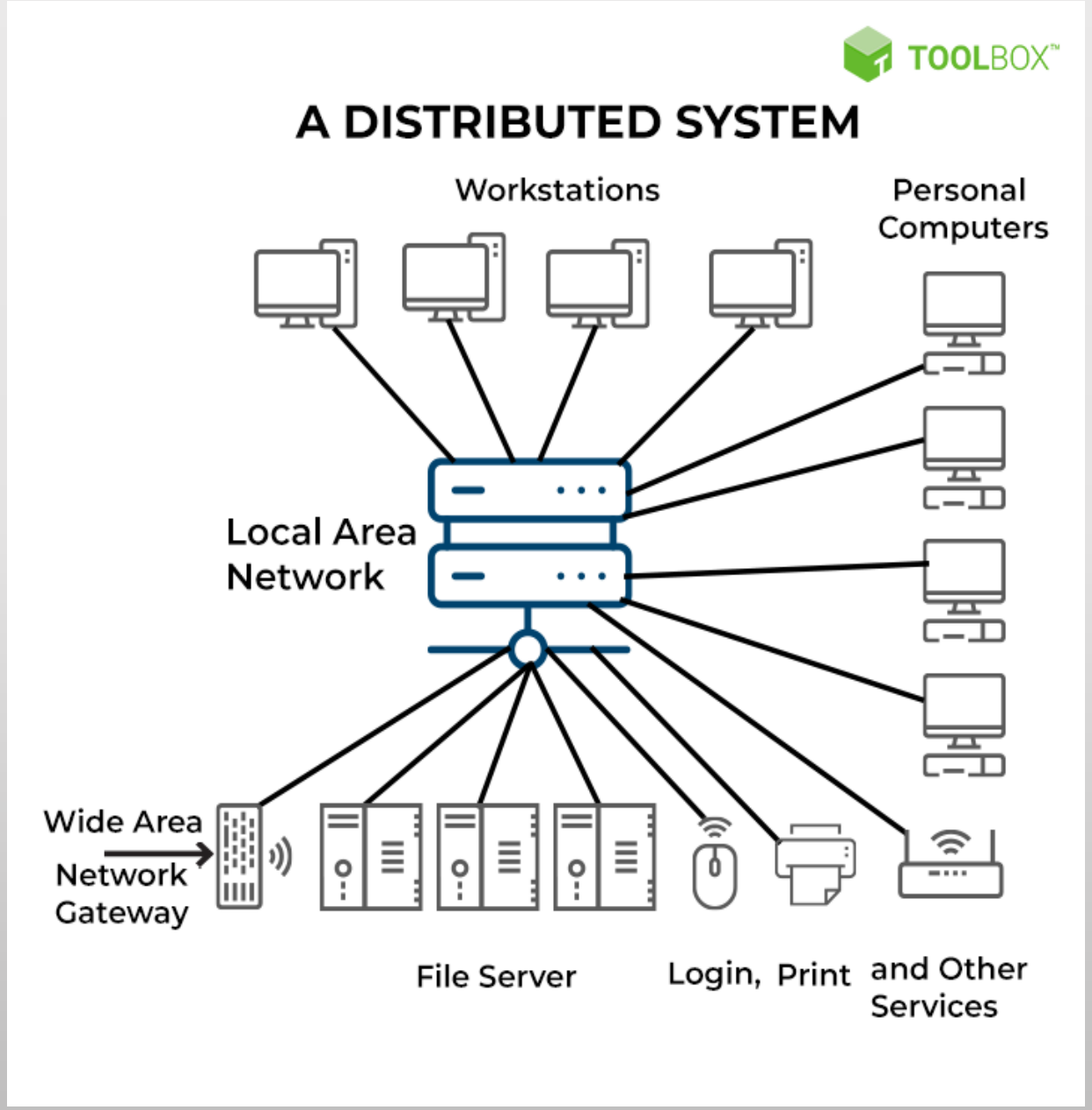
Distributed computing networks can be connected as local networks or through a wide area network if the machines are in a different geographic location. Processors in distributed computing systems typically run in parallel.

Distributed computing connects hardware and software resources to do many things, including:

- Work in collaboration to achieve a single goal through optional **resource sharing**;
- Manage **access rights** per the authority level of users;
- Keep resources, e.g., distributed computing software, **open** for further development;

- Achieve **concurrency** that lets multiple machines work on the same process;
- Ensure all computing resources are **scalable** and operate faster when multiple machines work together;
- Detect and handle errors in connected components of the distributed network so that the network doesn’t fail and stays **fault-tolerant**.

Distributed computing diagram



Benefits of distributed computing

Distributed computing includes the following benefits:

- **Performance.** Distributed computing can help improve performance by having each computer in a cluster handle different parts of a task simultaneously.
- **Scalability.** Distributed computing clusters are scalable by adding new hardware when needed.
- **Resilience and redundancy.** Multiple computers can provide the same services. This way, if one machine isn't available, others can fill in for the service. Likewise, if two machines that perform the same service are in different data centers and one data center goes down, an organization can still operate.

- **Cost-effectiveness.** Distributed computing can use low-cost, off-the-shelf hardware.
- **Efficiency.** Complex requests can be broken down into smaller pieces and distributed among different systems. This way, the request is simplified and worked on as a form of parallel computing, reducing the time needed to compute requests.
- **Distributed applications**. Unlike traditional applications that run on a single system, distributed applications run on multiple systems simultaneously.

Four Types of Distributed Systems

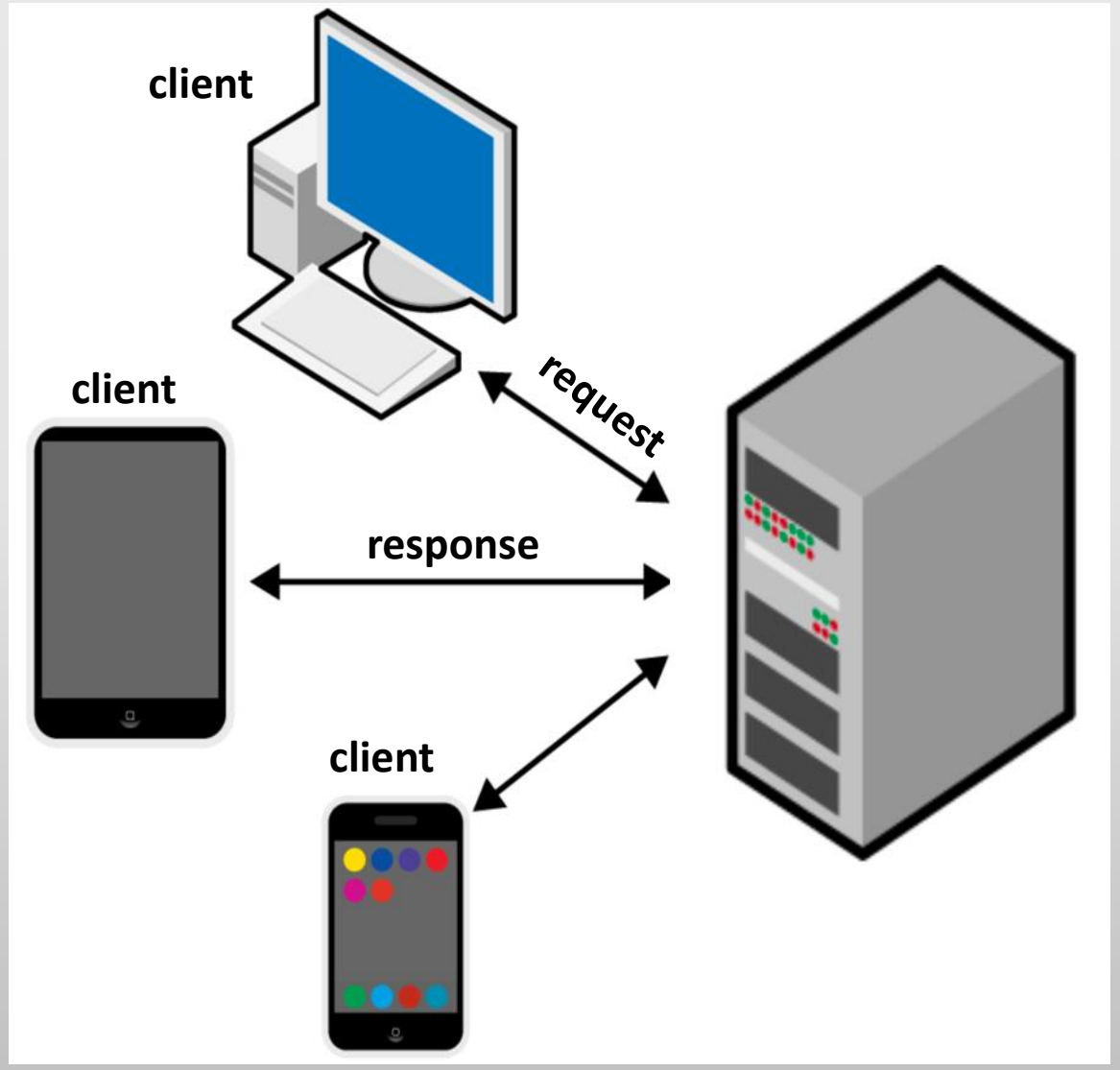
Under the umbrella of distributed systems, there are a few different architectures. Broadly, we can divide distributed cloud systems into four models:

- **Client-Server Model**

In this model, the client fetches data from the server directly then formats the data and renders it for the end-user. To modify this data, end-users can directly submit their edits back to the server.

For example, companies like Amazon that store customer information. When a customer updates their address or phone number, the client sends this to the server, where the server updates the information in the database.

Client-Server Model



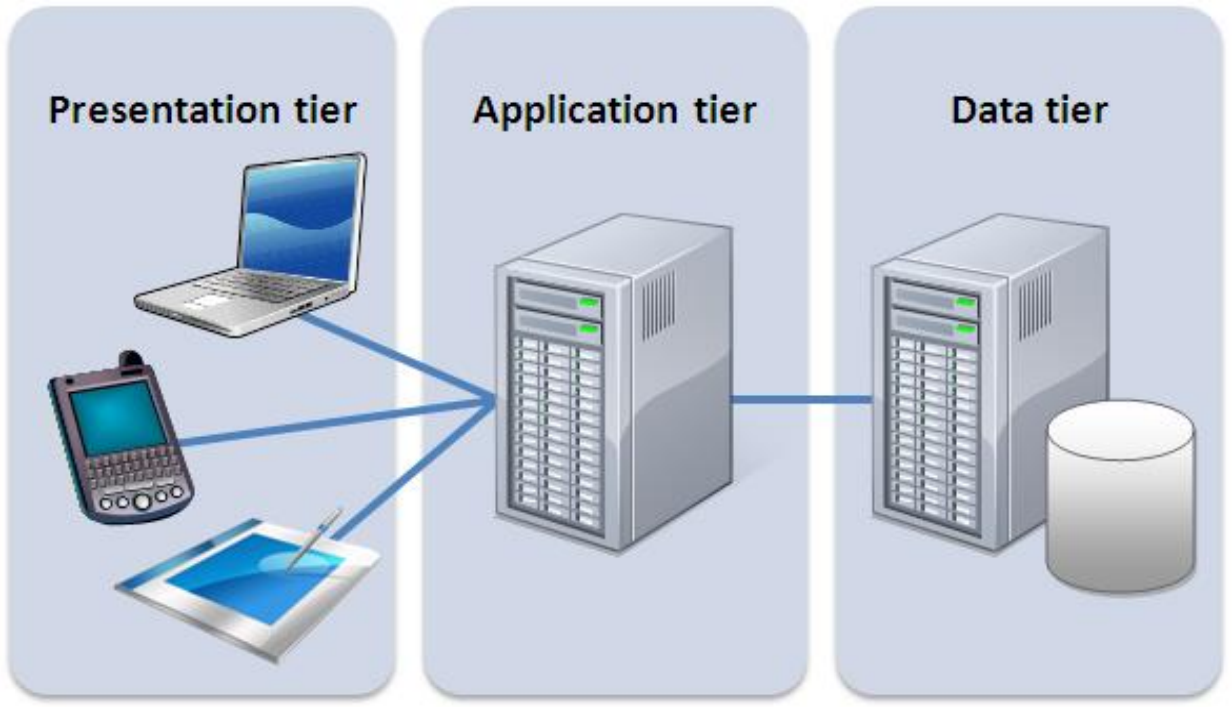
- **Three-Tier Model**

The three-tier model introduces an additional tier between client and server — the agent tier.

This middle tier holds the client data, releasing the client from the burden of managing its own information. The client can access its data through a web application, typically. Through this, the client application's and the user's work is reduced and automated easily.

For example, a cloud storage space with the ability to store your files and a document editor. Such a storage solution can make your file available anywhere for you through the internet, saving you from managing data on your local machine.

Three-Tier Model



The **Presentation tier** - the part of the application which is visible to the user; it enables the input of requirements and the presentation of results. It is dependent on the platform (e.g. web applications, Windows applications, Android applications, etc.). It may therefore be different for different devices or platforms.

The **Application tier** (also functional) - the middle layer of the model (middleware), it assures the calculations and operations performed between input-output requirements and data. Also known as the application server.

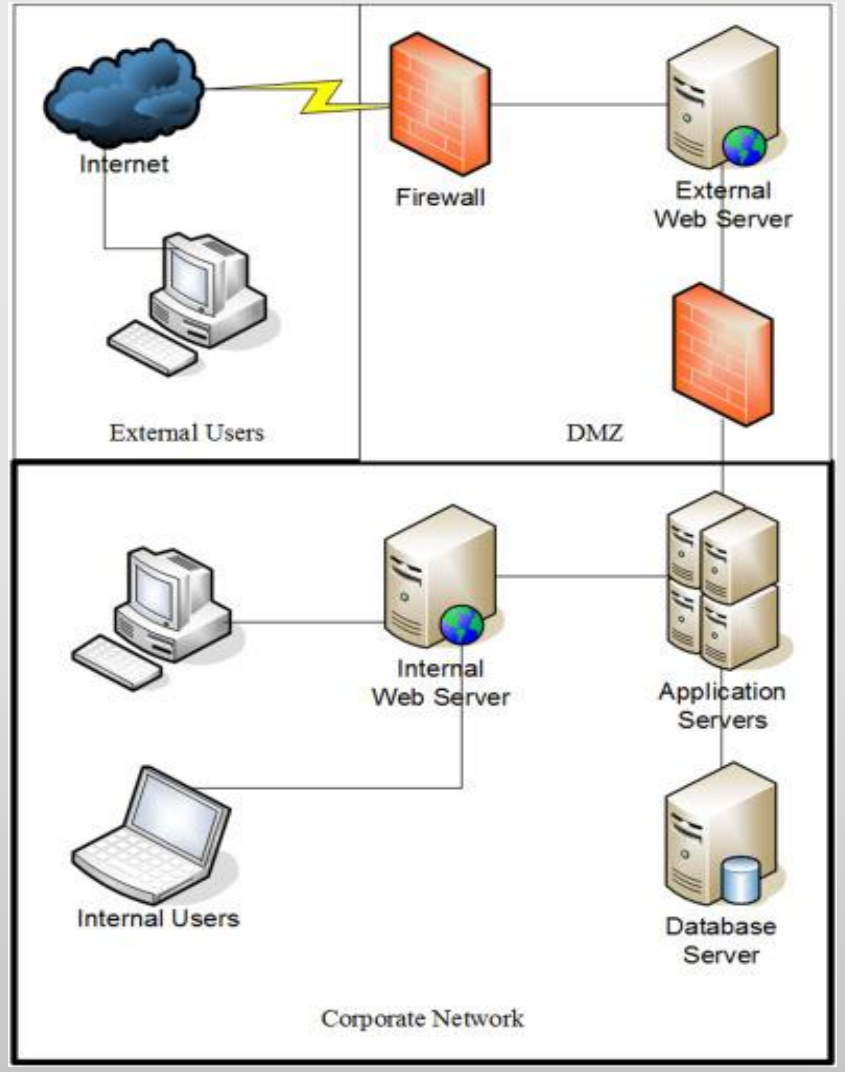
The **Data tier** (also database) - the lowest layer of the model, it ensures all operations with data, i.e. database management system and basic data-base operations for functional storage, selection, aggregation, processing, integrity, and data audit.

- **Multi-Tier Model**

Enterprises need business logic to interact with various backend data tiers and frontend presentation tiers. This logic sends requests to multiple enterprise network services easily. That’s why large organizations prefer the n-tier or multi-tier distributed computing model.

For example, an enterprise network with n-tiers that collaborate when a user publishes a social media post to multiple platforms. The post itself goes from data tier to presentation tier.

Multi-Tier Model

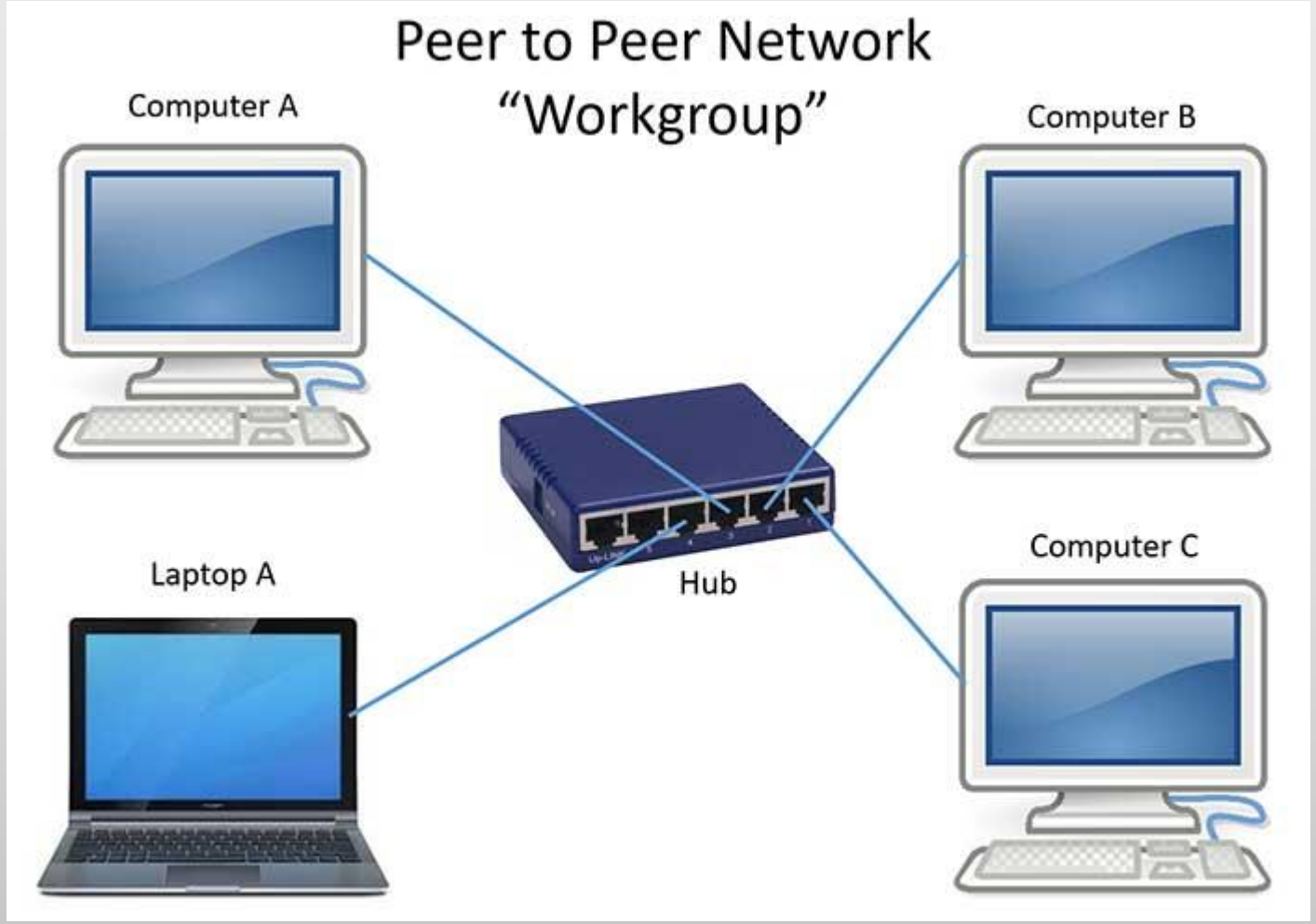


- **Peer-to-Peer Model**

Unlike the hierarchical client and server model, this model comprises peers. Each peer can act as a client or server, depending upon the request it is processing. These peers share their computing power, decision-making power, and capabilities to work better in collaboration. Peer-to-peer (P2P) is a decentralized communications model in which each party has the same capabilities and either party can initiate a communication session.

For example, blockchain nodes collaboratively work to make decisions regarding adding, deleting, and updating data in the network.

Peer-to-Peer Model



End