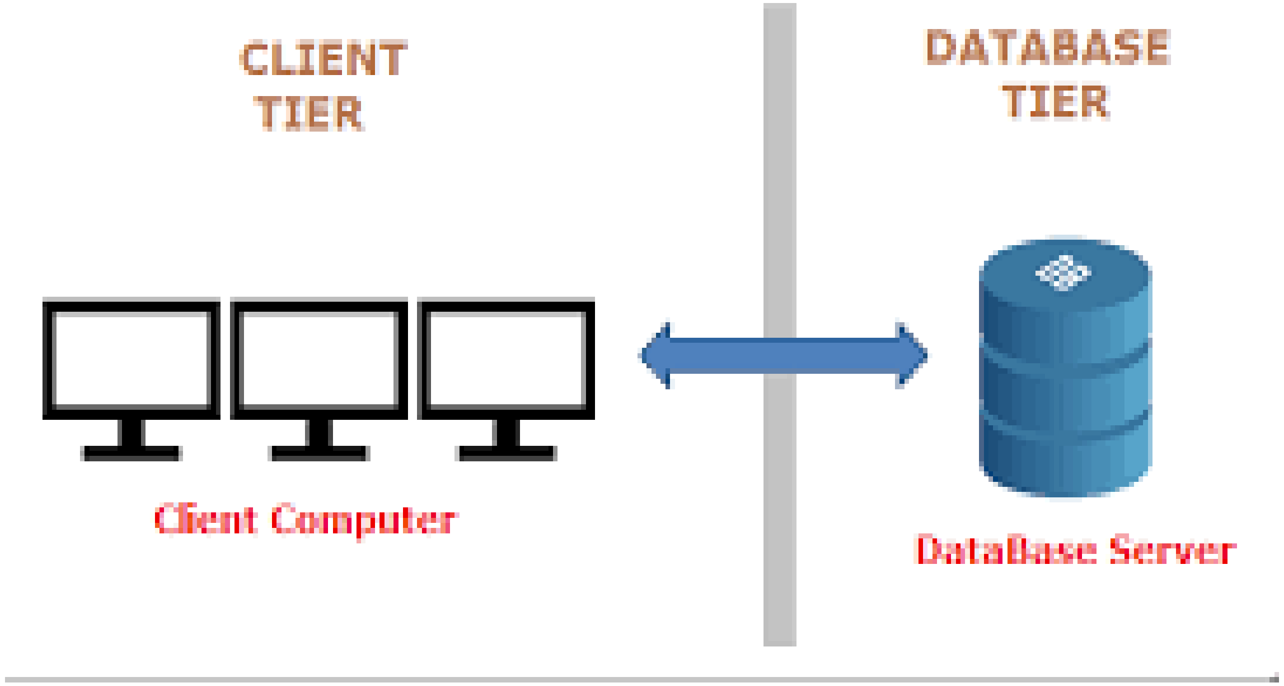
Data Base system Architecture

Ram Datta Bhatta

Client Server Architecture

- ✓is a common approach for organizing the components and interactions between clients (applications) and database servers.
- ✓The workload and responsibilities are distributed between the client application (front-end) and the database server (back-end).
- ✓ It allows multiple clients to access and interact with a centralized database or a set of distributed databases efficiently.
- ✓ Clients send requests to the server, and the server responds with the requested data or results.
- ✓Client/server architecture provides a high degree of independence between the client and server components.

Client Server Architecture



Client

- ✓The client is the front-end component of the architecture, representing the user interface or application that interacts with the end-users.
- ✓Clients can be desktop applications, web applications, mobile apps, or any software that requires data from the database.
- ✓They are responsible for presenting data to users, capturing user inputs, and sending requests to the database server for data manipulation or retrieval.

Server in database

- ✓The server is the back-end component that manages the database and handles the data processing and storage operations.
- ✓The database server can be a dedicated machine or a cluster of servers, depending on the scale and complexity of the database system.
- ✓It is responsible for executing queries, managing data integrity, enforcing security, and controlling access to the database

Tier Architecture

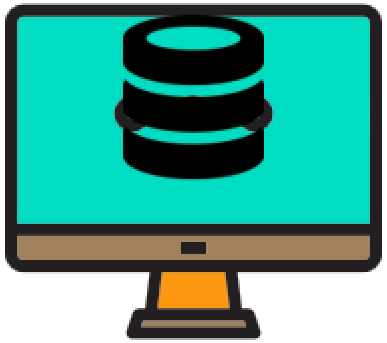
- ✓The tier architecture refers to the organization of a database system into distinct tiers or layers.
- ✓Each tier has specific responsibilities and interacts with other tiers in a well-defined manner.
- ✓There are mainly three types of DBMS architecture:

1. One Tier Architecture (Single Tier Architecture)
2. Two Tier Architecture
3. Three Tier Architecture

https://www.guru99.com/dbms-architecture.html

One Tier

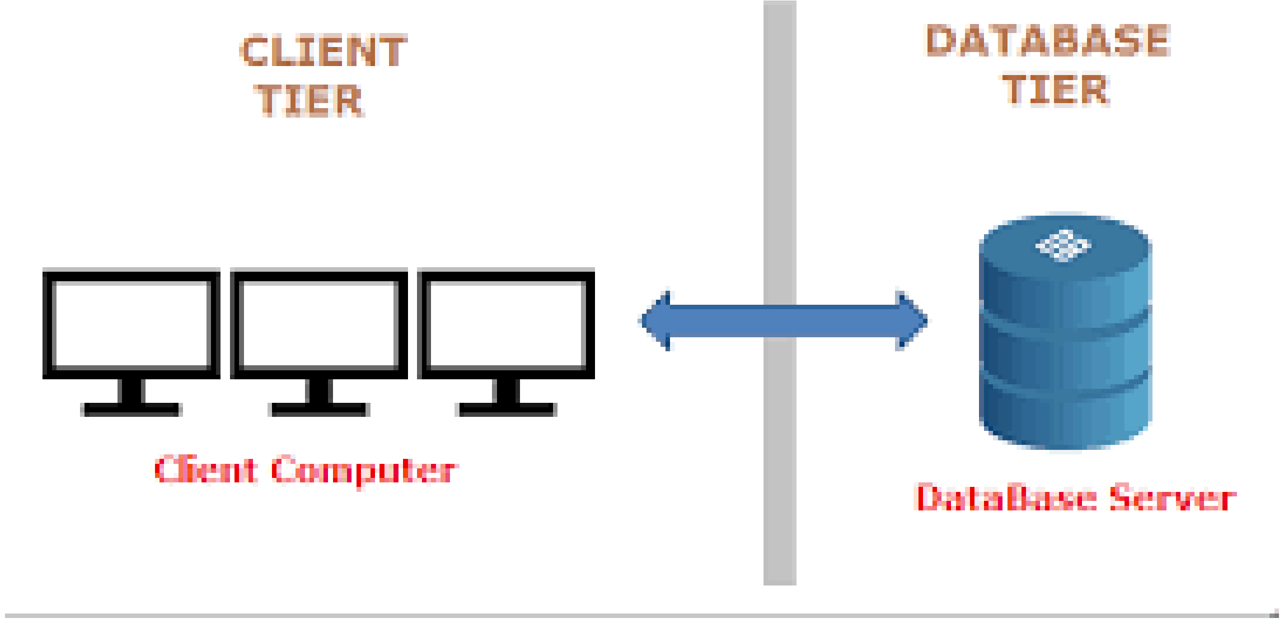
One Tier Architecture in DBMS is the simplest architecture of Database in which the client, server, and Database all reside on the same machine.



Single Tier Architecture

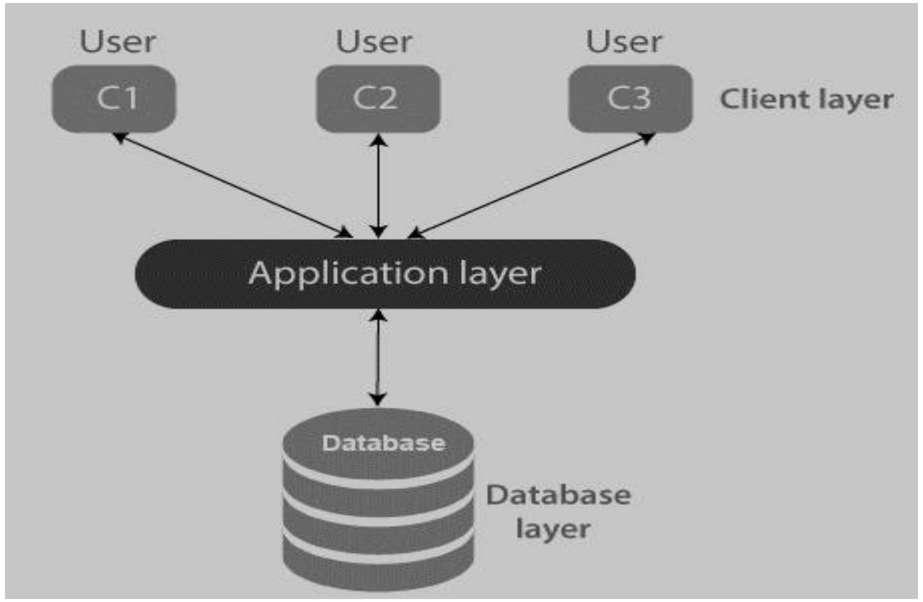
Two Tier

- ✓A **2 Tier Architecture** in DBMS is a Database architecture where the presentation layer runs on a client (PC, Mobile, Tablet, etc.), and data is stored on a server called the second tier.
- ✓ Two tier architecture provides added security to the DBMS as it is not exposed to the end-user directly



3 Tier

- ✓A **3 Tier Architecture** in DBMS is the most popular client server architecture in DBMS in which the development and maintenance of functional processes, logic, data access, data storage, and user interface is done independently as separate modules.
- ✓ Three Tier architecture contains a presentation layer, an application layer, and a database server



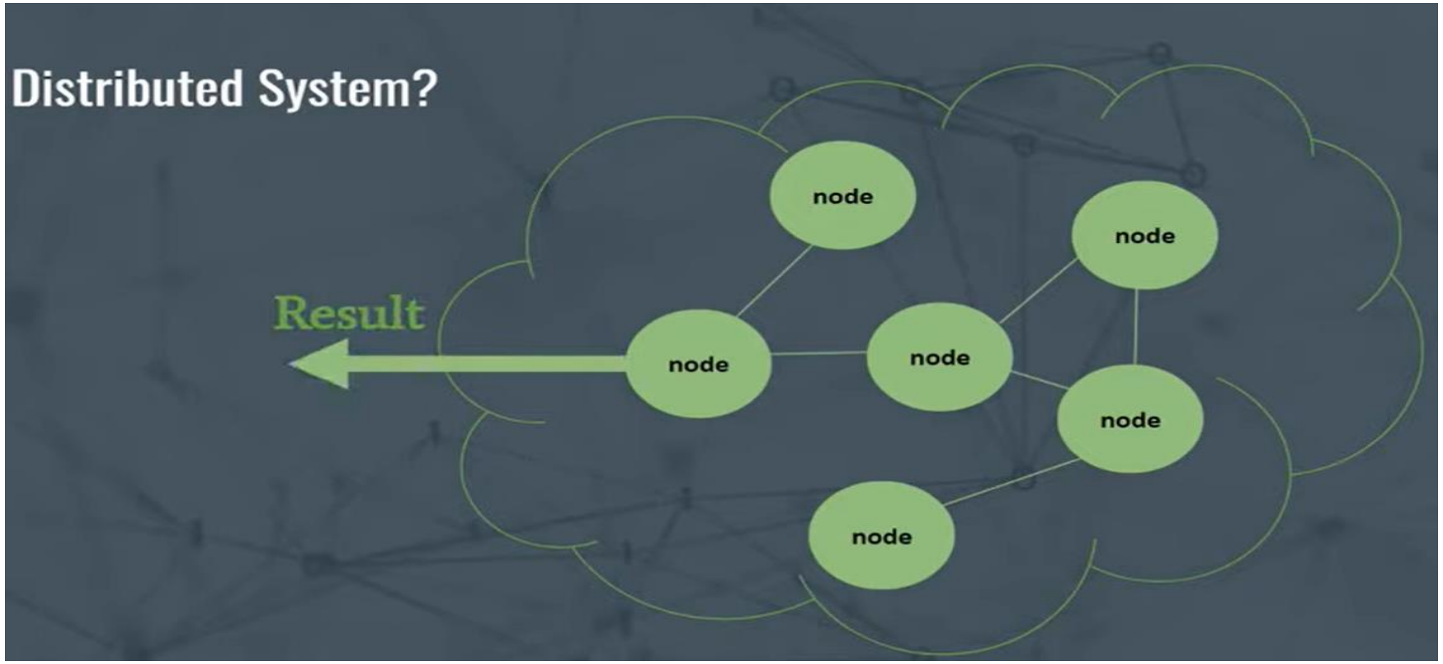
Distributed System

A distributed system is a **collection of autonomous computing elements** that appears to its users as a **single coherent system**.

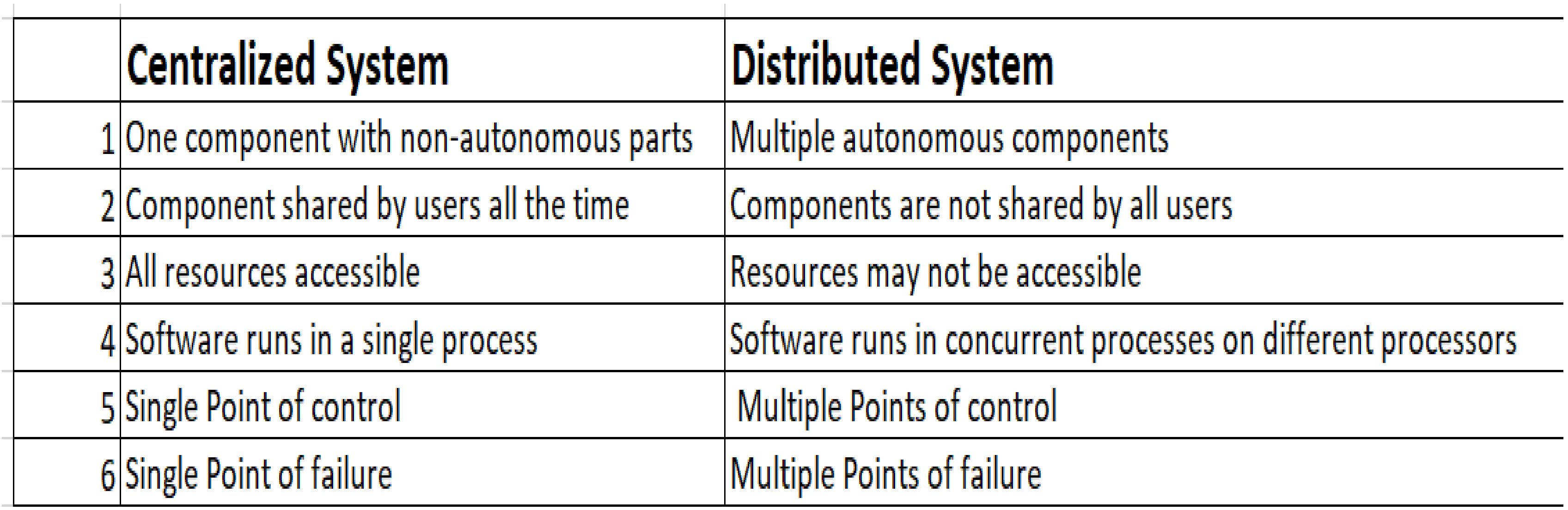
- 1: **Collection of autonomous computing elements:** A computing element, which we will generally refer to as a node, can be either a hardware device or a software process. They are linked by networks and equipped with distributed system s/w.
- 2: **Single coherent system** (the complexities of the system are hidden to the end user, making the whole system appear as a single computer to its users)

Example: Internet, Server-client, cloud computing, internet, Google search, DNS

Distributed System is a collection of separate and independent software/hardware components called nodes, that are networked and work together coherent by coordinating and communicating through message passing or events, to fulfill one end goal.



Centralized Vs Distributed System



	Centralized System	Distributed System
1	One component with non-autonomous parts	Multiple autonomous components
2	Component shared by users all the time	Components are not shared by all users
3	All resources accessible	Resources may not be accessible
4	Software runs in a single process	Software runs in concurrent processes on different processors
5	Single Point of control	Multiple Points of control
6	Single Point of failure	Multiple Points of failure

No Shared Clock

No Shared Memory

Concurrency

Characteristics of distributed system

1. Resource Sharing : Ability to use any hardware, software or data anywhere in the system.
2. Openness : Detailed interfaces of components need to be published
3. Concurrency : multiple operations and tasks performed in parallel
4. Scalability
5. Fault Tolerance : nodes fail independently
6. No shared clocks and no shared memory : don’t share a global clock (i.e. independent clocks)
7. Heterogeneity

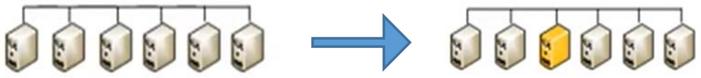
Why distributed system? Why not a single supercomputer rather than distributed system?

Characteristics of Distributed Computing

1

RELIABILITY

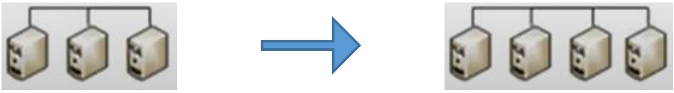
Multiple computers provide redundancy at multiple levels



2

INCREMENTAL GROWTH

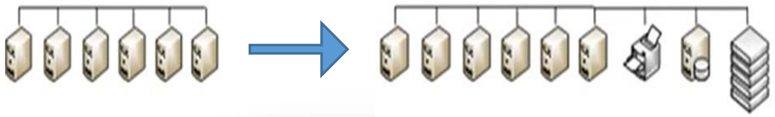
A well designed distributed system can grow as needed without affecting the way it works



3

RESOURCE SHARING

Existing resources in a distributed system can be accessed or remotely accessed across multiple computers in a system



4

OPEN AND HETEROGENOUS

Multiple computers of different types, OSes and manufacturers can interact together in a simple system

Goals of Distributed Computing

The distributed computing allows scalability, sharing resources and helps to perform computation tasks efficiently

Thank you.