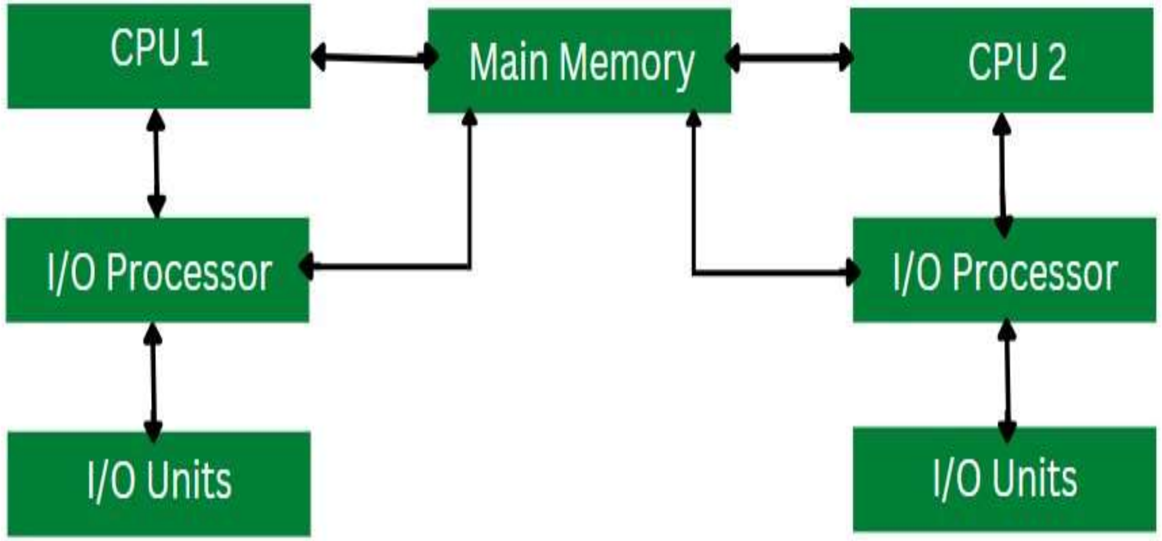
Distributed Operating Systems

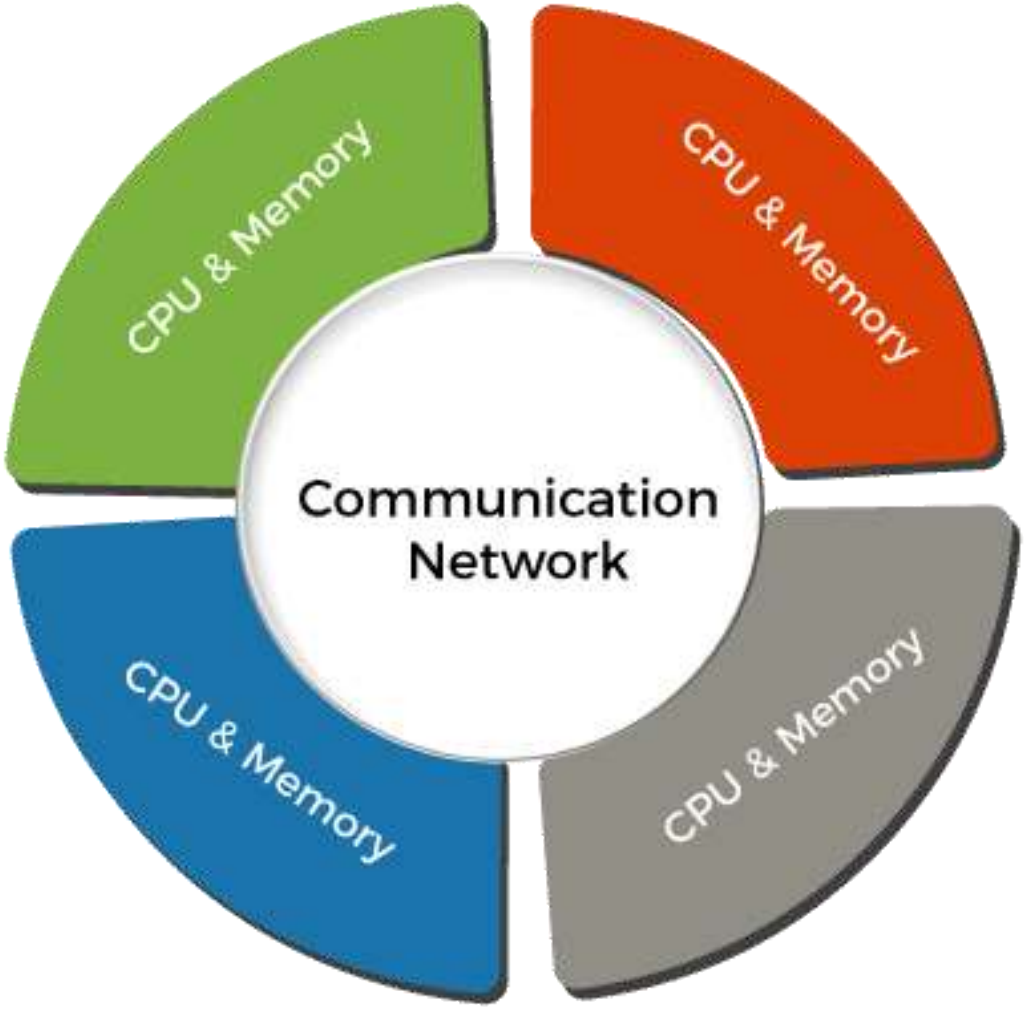
MULTI-PROCESSOR SYSTEMS



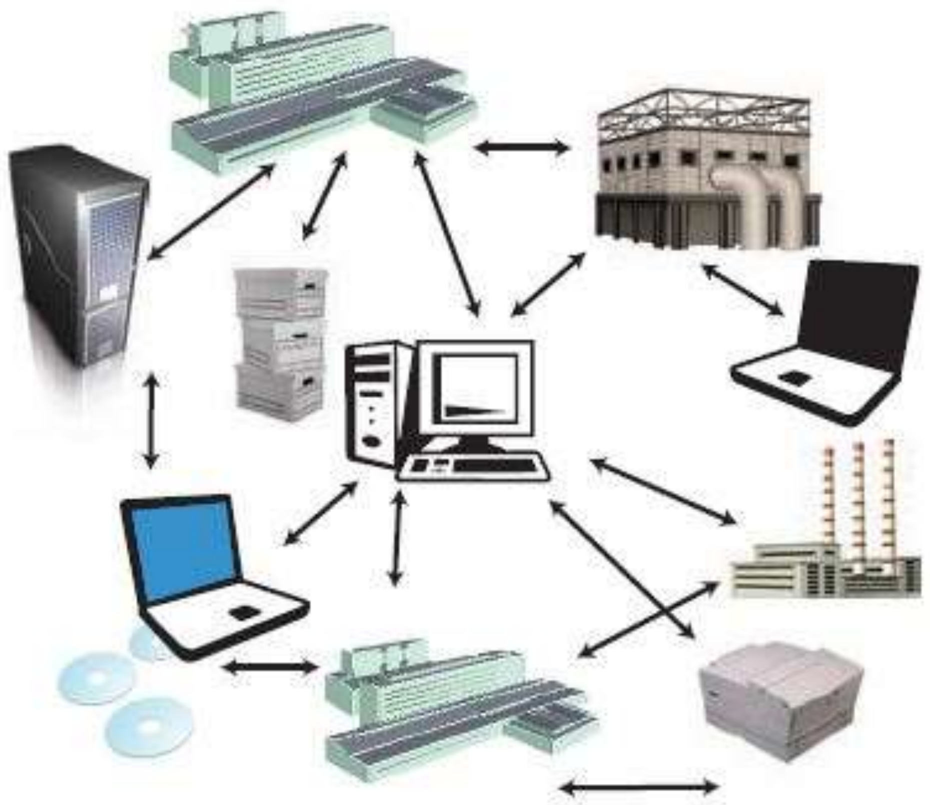
- ✓ A system that contains more than one central processing unit (CPU)

- ✓ Allow multiple processors to work together concurrently, sharing the workload and thereby increasing overall system performance and throughput

- Distributed systems use many central processors to serve multiple real-time applications and users.
- As a result, data processing jobs are distributed between the processors.



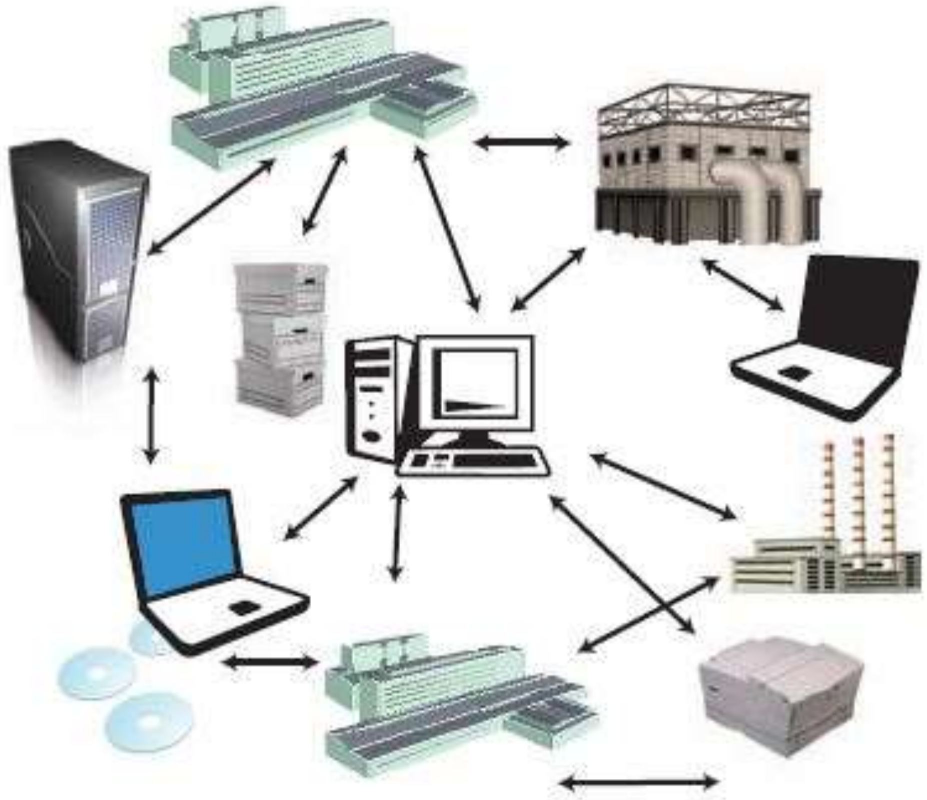
DISTRIBUTED SYSTEMS



- A distributed system is a collection of autonomous computers linked by a computer network and equipped with distributed system software.

DOS manage resources across a network of interconnected computers.

DISTRIBUTED SYSTEMS

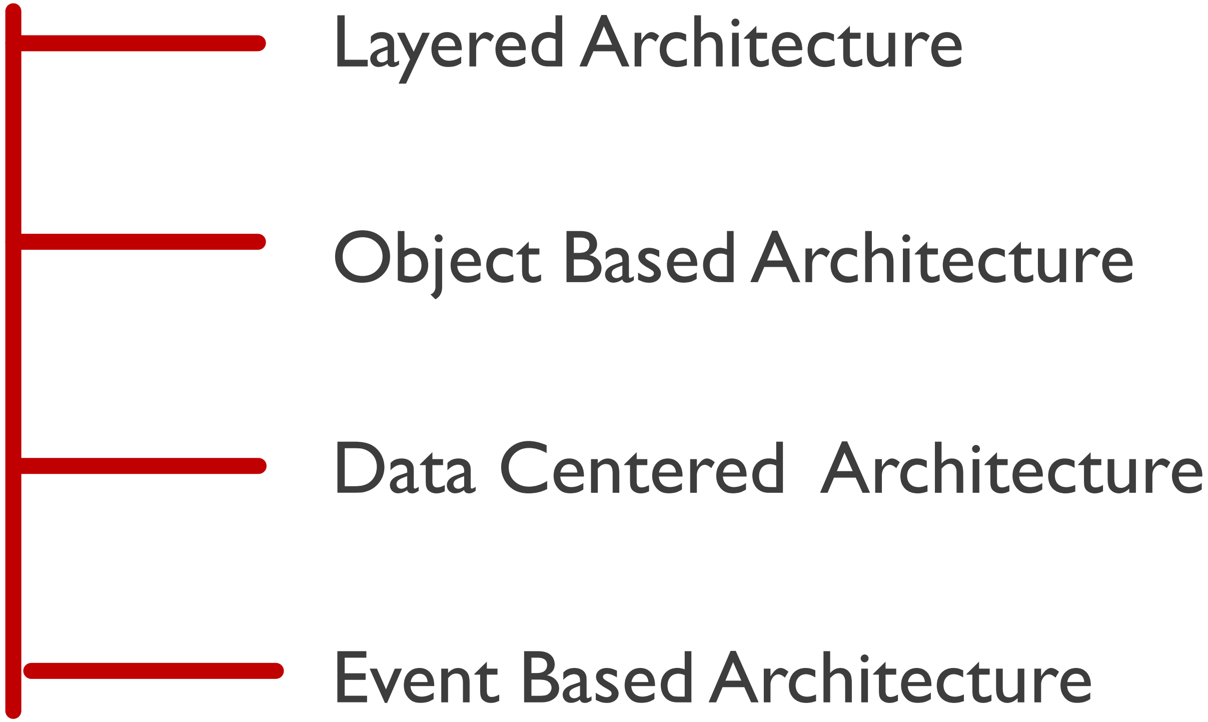


- Multiple Nodes
- Physically Separate

DOS manage resources across a network of interconnected computers.

DISTRIBUTED ARCHITECTURES

- Logical organization of software components and their interaction with other structures.



Layered Architecture

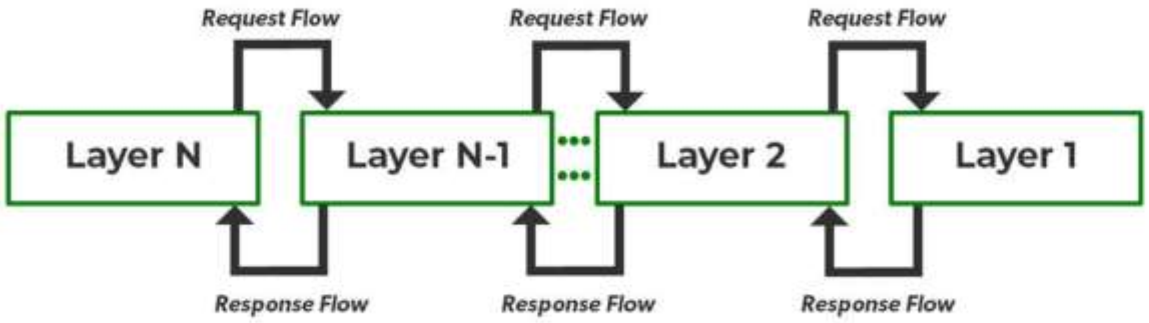
Object Based Architecture

Data Centered Architecture

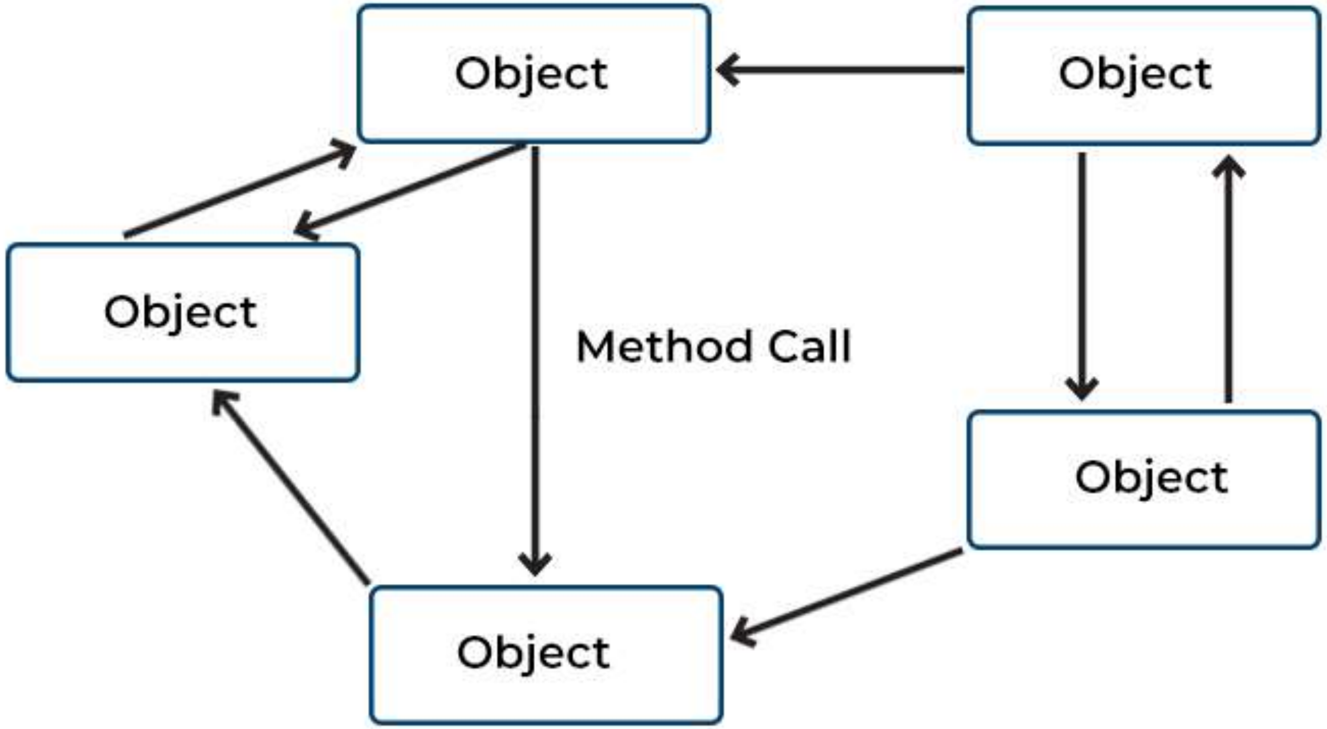
Event Based Architecture

LAYERED ARCHITECTURE

- It organizes the system into hierarchical layers, each with specific functions and responsibilities.
- Interact only with adjacent layers.
- Manage complexity and promotes separation of concerns.
- This separation helps in managing and scaling the system more effectively.



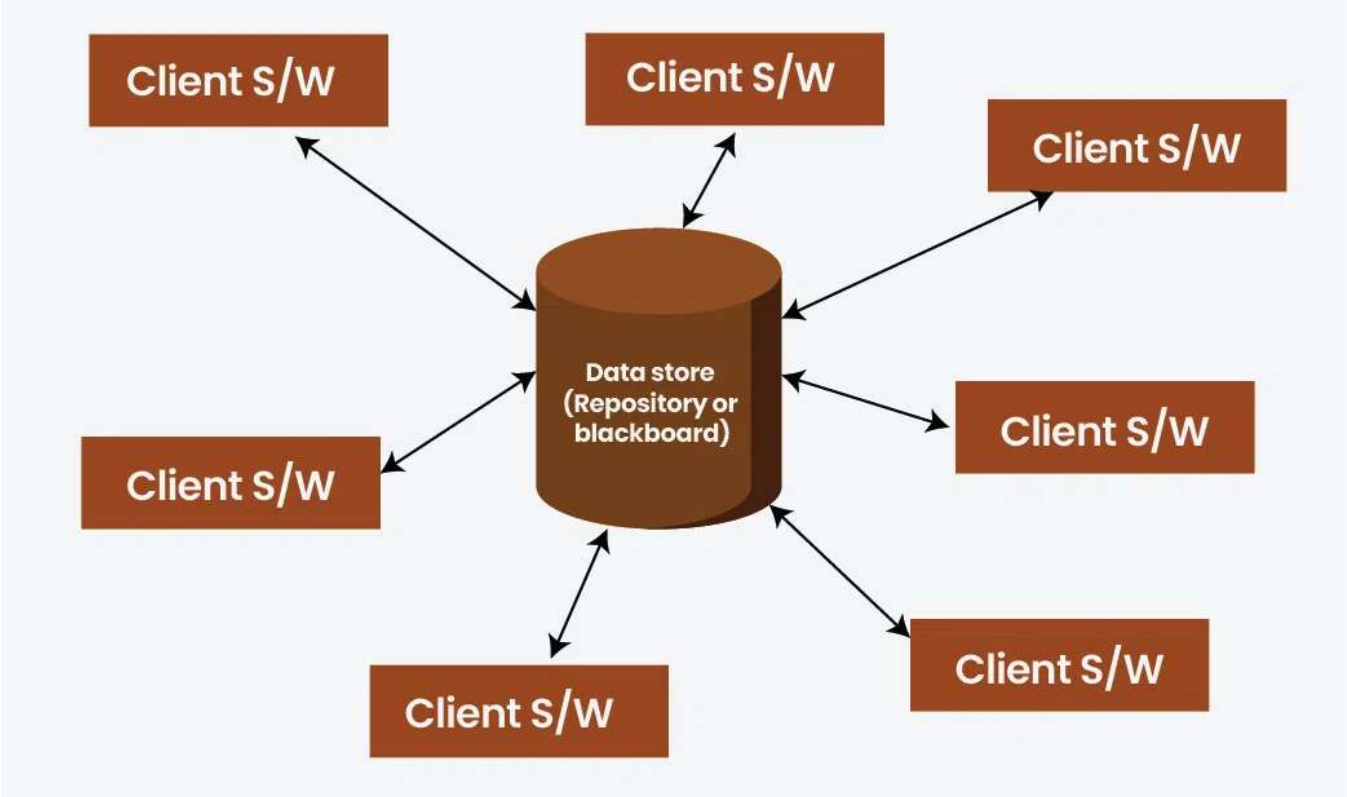
OBJECT BASED ARCHITECTURE



OBJECT BASED ARCHITECTURE

- Components are treated as objects which convey information to each other.
- Object-Oriented Architecture contains an arrangement of loosely coupled objects.
- Objects can interact with each other through method calls.
- Objects are connected to each other through the Remote Procedure Call (RPC) mechanism or Remote Method Invocation (RMI) mechanism.
- Web Services and REST API are examples of object-oriented architecture.
- Invocations of methods are how objects communicate with one another. Typically, these are referred to as Remote Procedure Calls (RPC)

DATA CENTRED ARCHITECTURE

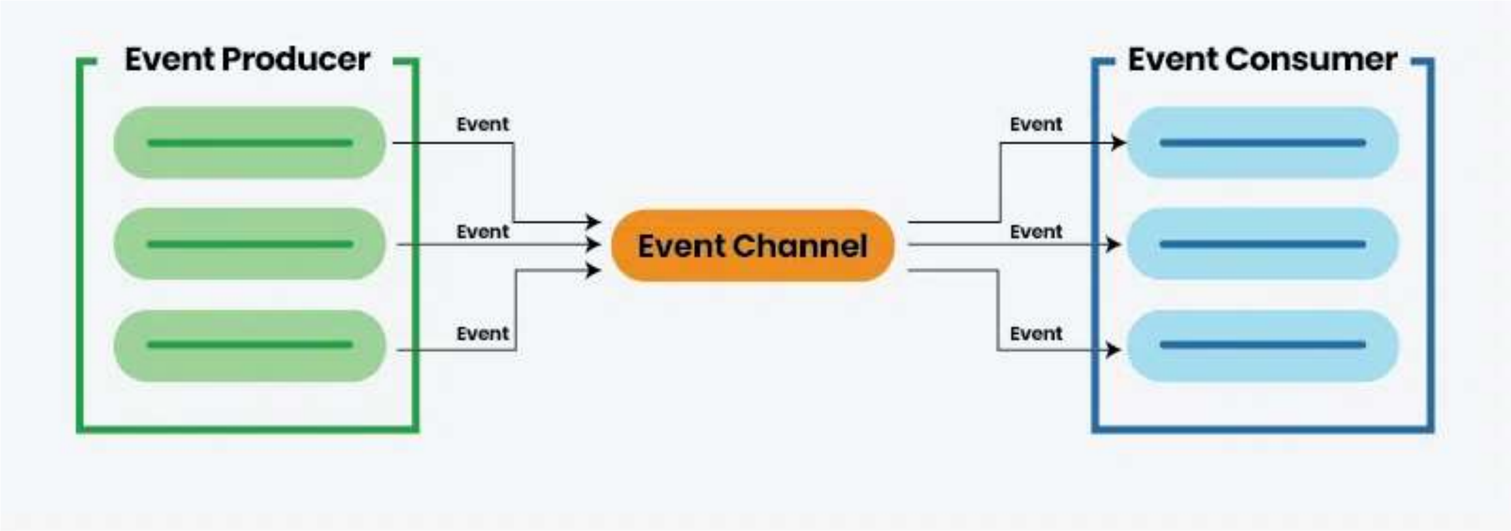


OBJECT BASED ARCHITECTURE

- Focuses on the central management and utilization of data.
- It has a central data repository at the centre. Required data is then delivered to the components.
- Data is treated as a critical asset, and the system is designed around data management, storage, and retrieval processes rather than just the application logic or user interfaces.

EVENT BASED ARCHITECTURE

- Flow of data and control in a system is driven by events.
- When an event occurs, the system, as well as the receiver, get notified.



EVENT BASED ARCHITECTURE

- Event Producers: Components or services that generate events to signal state changes or actions.
- Event Consumers: Components or services that listen for and react to events, processing them as needed.
- Event Channels: Mechanisms for transmitting events between producers and consumers, such as message queues or event streams.
- Loose Coupling: Producers and consumers are decoupled, interacting through events rather than direct calls, allowing for more flexible system interactions.