Module 1
Introduction to Distributed Systems

Ms. Rashmi Bhat

Characterization of Distributed Systems

- Introduction
- Goals and Issues of Distributed Systems
- Types of Distributed Systems
- Hardware and Software Concepts
- Software Concepts
- Models of Middleware
- Services offered by Middleware

Introduction

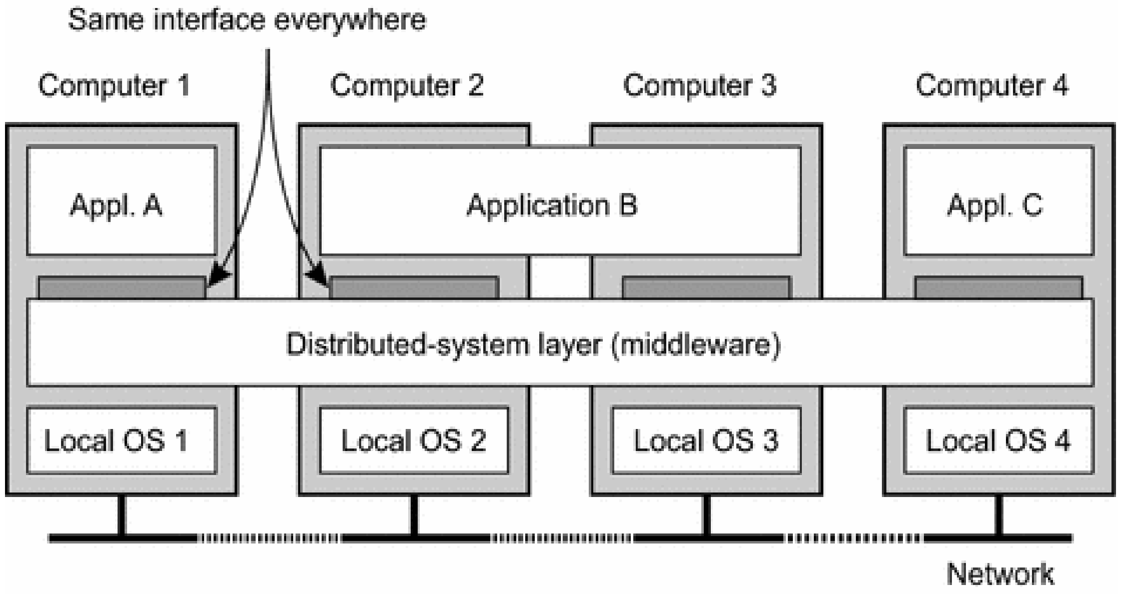
- Definition

  “A distributed system is a collection of independent computers that appears to its users as a single coherent system.”

  - Distributed system is a collection of autonomous hosts that are connected through a computer network.
  - Each host executes components and operates a distribution middleware.
  - Middleware enables the components to coordinate their activities.
  - User perceives the system as a single, integrated computing facility.

Introduction

- Distributed systems are often organized by means of a layer of software and a layer underneath consisting of operating systems and basic communication facilities.
- A distributed system is sometimes called middleware.



Introduction

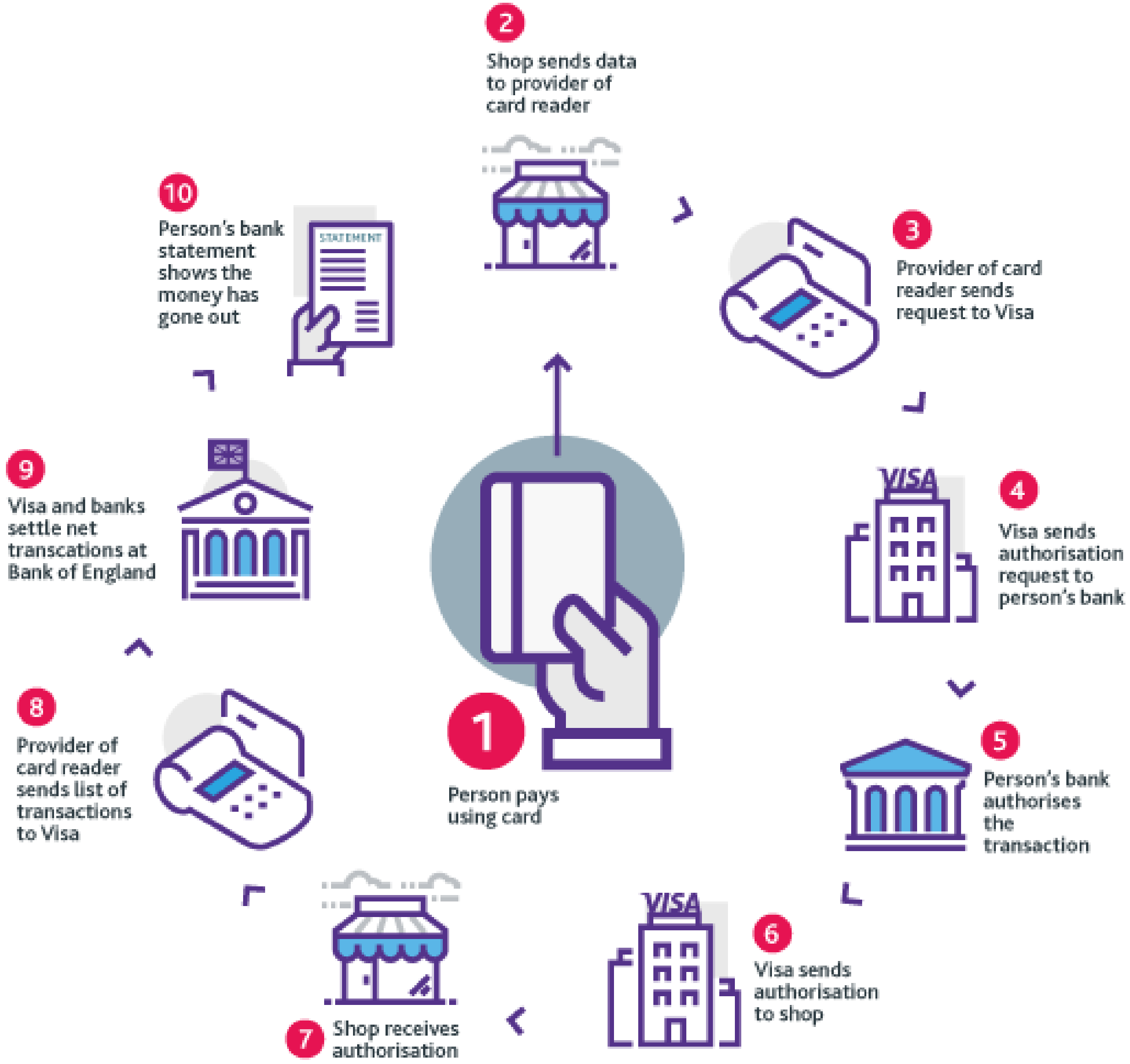
- Characteristics of Distributed Systems
  - Differences and way of communication between computers is hidden
  - Hidden internal organization
  - Users and applications can interact in a consistent and uniform way
  - Continuously available
  - Allows resource sharing
  - Global clock is not required
  - Autonomous components
  - Greater fault tolerance

Introduction

- Examples of Distributed Systems
  - Finance and commerce
  - The information Society
  - Entertainment
  - Healthcare
  - Transport and logistics
  - Environmental management

Introduction

- Example:
  - E-commerce



1 Person pays using card

2 Shop sends data to provider of card reader

3 Provider of card reader sends request to Visa

4 Visa sends authorisation request to person's bank

5 Person's bank authorises the transaction

6 Visa sends authorisation to shop

7 Shop receives authorisation

8 Provider of card reader sends list of transactions to Visa

9 Visa and banks settle net transcations at Bank of England

10 Person's bank statement shows the money has gone out

Goals of Distributed Systems

- Four goals must be met to build a distributed system
- It should
  - Make resources available
  - Hide the fact that resources are distributed across a network
  - Be open
  - Be scalable.

Goals of Distributed Systems

1. Making resource available
   - Users and applications should be able to access the remote resources easily
   - Resource sharing
   - Should ensure defense mechanism against eavesdropping or intrusion on communication

2. Distribution transparency
   - A distributed system that is able to present itself to users and applications as if it were only a single computer is called to be ***transparent***.
     - **Access** transparency
     - **Location** transparency
     - **Relocation** transparency
     - **Migration** transparency
     - **Concurrency** transparency
     - **Replication** transparency
     - **Failure** transparency

Goals of Distributed Systems

3. Openness
   - An open distribution system is a system that offers services according to the standard rules that describe the syntax and semantics of those services.
   - An open distributed system should be
     - standard
     - Extensible
     - Interoperable
     - Portable
     - Flexible
   - E.g. WWW

Goals of Distributed Systems

4. Scalability
   - Scalability of a system is measured along three dimensions
     - Size scalable

       One can easily add more users and resources to the system
     - Geographical scalable

       The users and resources may lie far apart.
     - Administrative scalable

       it can still be easy to manage even if it spans many independent administrative organizations.

Issues in Distributed Systems

- Heterogeneity
  - Heterogeneity applies to hardware devices, OS, network, programming languages
  - Common standards need to be agreed and adopted.
- Transparency
  - User should be unaware of where the services are located and transferring from local machine to a remote one should be transparent.
- Openness
  - Determines whether the system can be extended and re-implemented in various ways or not.

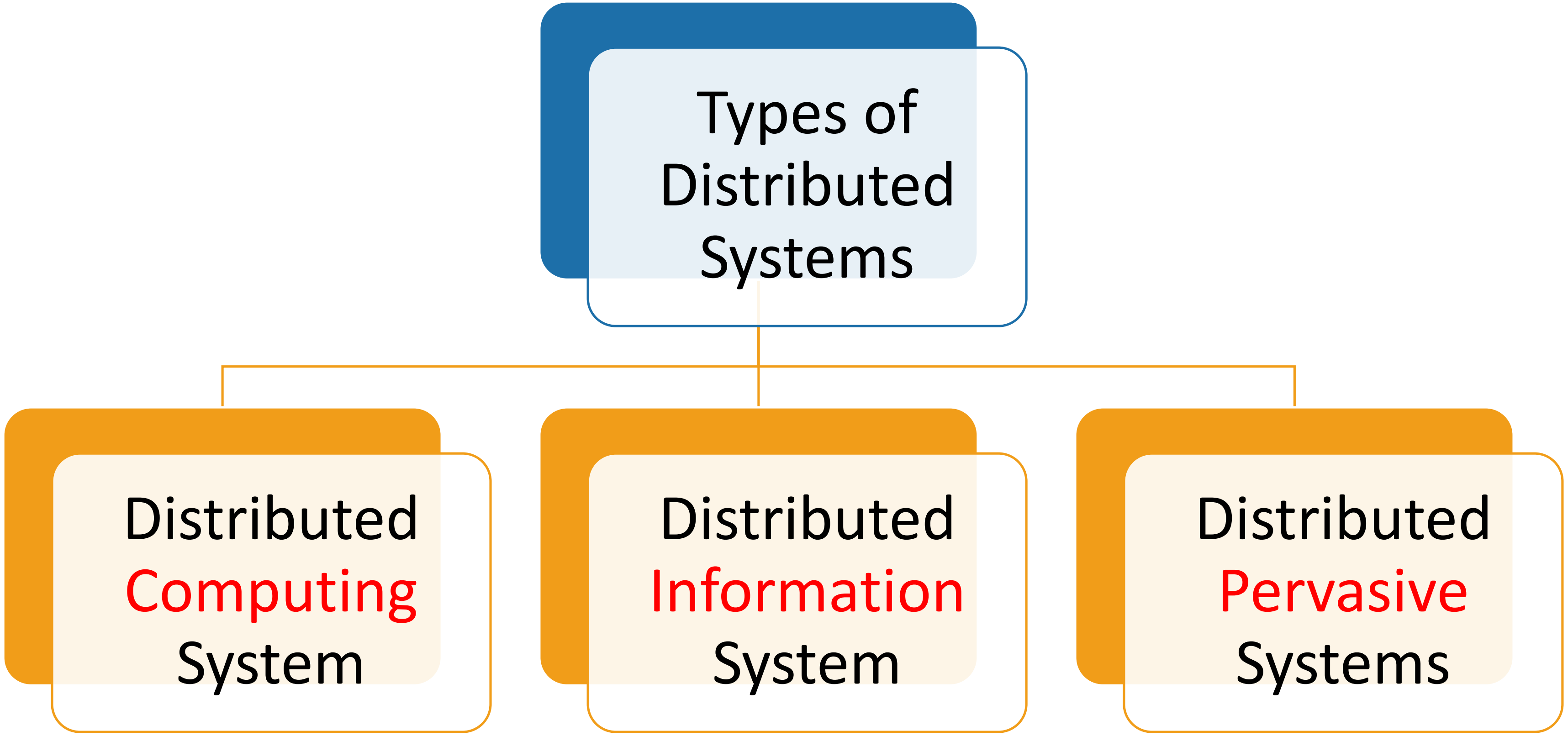
Issues in Distributed Systems

- Concurrency
  - Multiple users make requests on the same resources
  - Resources utilization must be synchronized in order to maintain consistency
- Security
  - Confidentiality
  - Integrity
  - Availability
- Scalability
  - Has 3 dimensions
    - Size
    - Geography
    - administration

Issues in Distributed Systems

- Failure handling
  - Failures in distributed systems are partial; hence handling failure is difficult
- Quality of Service
  - QoS is measured in terms of
    - Reliability
    - Security
    - Performance

Types of Distributed Systems



Types of Distributed Systems

Distributed Computing System

Distributed Information System

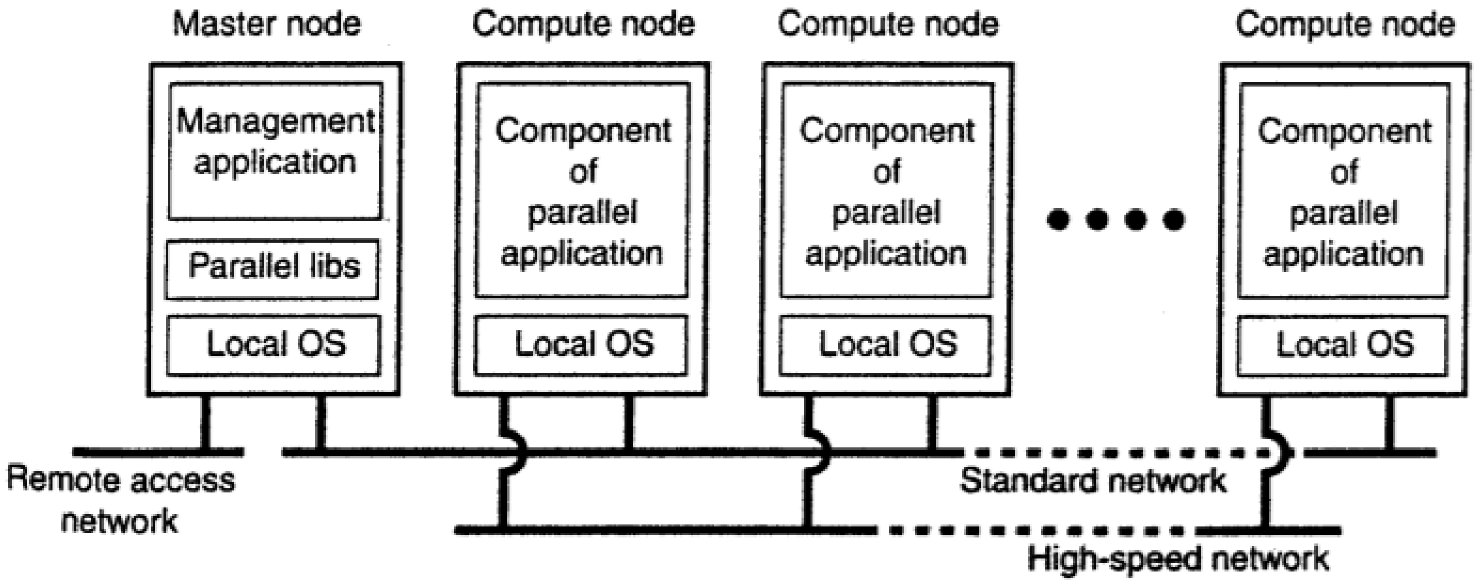
Distributed Pervasive Systems

Distributed Computing Systems

- Used for high performance computing tasks
- Cluster Computing Systems
  - A collection of similar computers, closely connected by means of high speed local area network
  - Each computer runs same OS.
  - Cluster computing is used for parallel programming in which a single (compute intensive) program is run in parallel on multiple machines.

Distributed Computing Systems

- Cluster Computing Systems



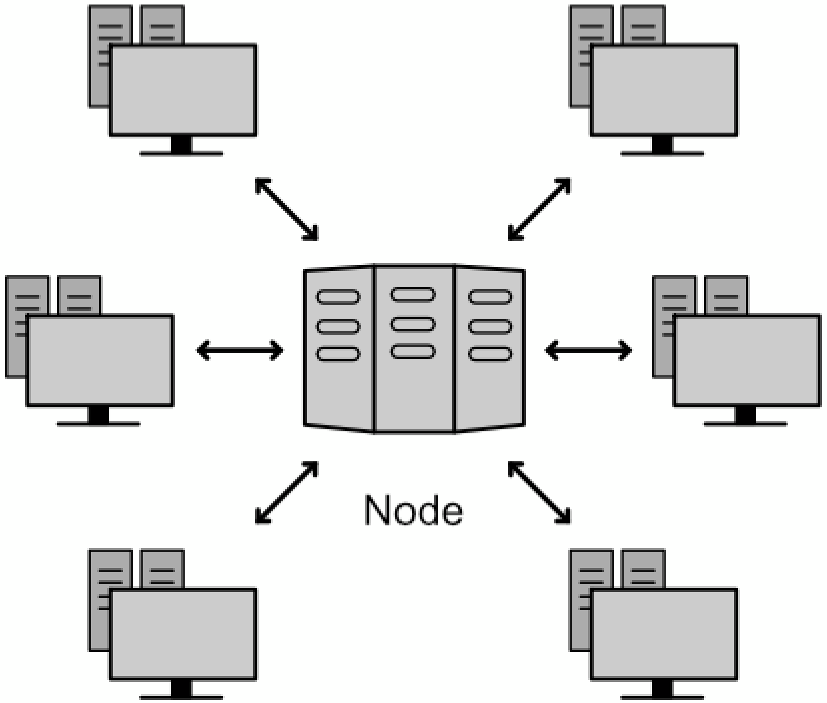
  - The master actually runs the middleware needed for the execution of programs and management of the cluster, while the compute nodes often need nothing else but a standard operating system.

Distributed Computing Systems

- Cluster Computing Systems
  - Features:
    - Consists of many homogeneous machines
    - Tightly coupled using dedicated network
    - Share resources
  - Types
    - Load balancing clusters
    - High availability (HA) clusters
    - High Performance (HP) clusters
  - Advantages:
    - Cost efficiency
    - Processing speed
    - Expandability
    - High availability of resources

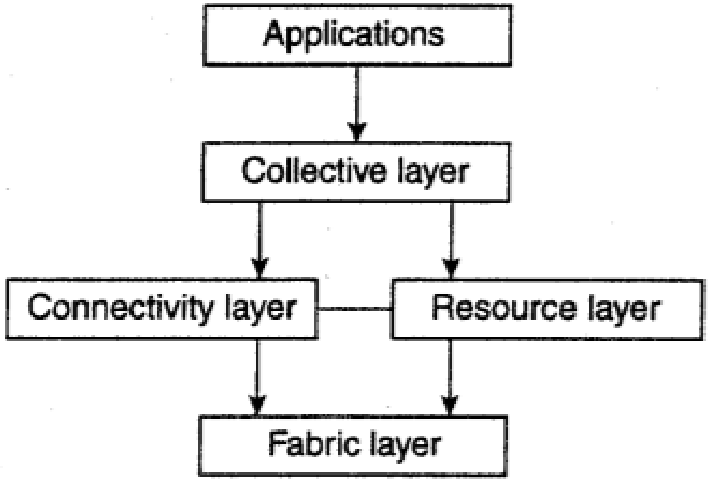
Distributed Computing Systems

- Grid Computing Systems
  - Grid computing systems have a high degree of heterogeneity.
  - Resources from different organizations are brought together to allow the collaboration of a group of people or institutions.
  - A virtual organization



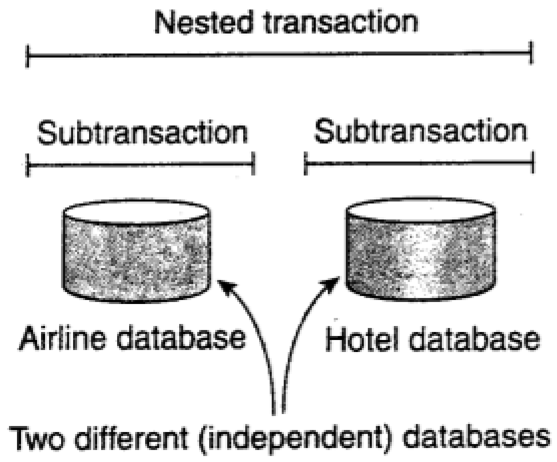
Distributed Computing Systems

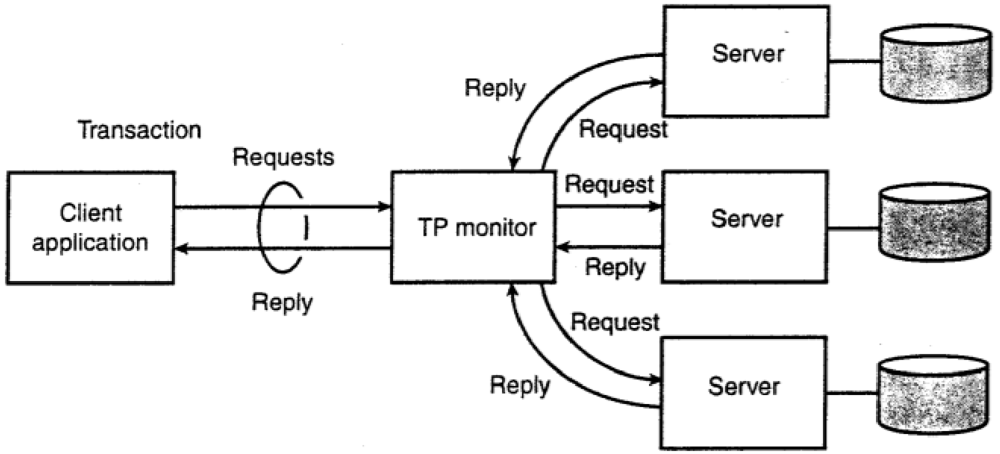
- Grid Computing Systems
  - Layered Architecture for grid computing systems:
    - **Fabric layer** provides interfaces to local resources at a specific site.
    - **Connectivity layer** consists of communication protocols for supporting grid transactions.
    - **Resource layer** is responsible for managing a single resource.
    - **Collective layer** deals with handling access to multiple resources
    - **The application layer** consists of the applications that operate within a virtual organization



Distributed Computing Systems

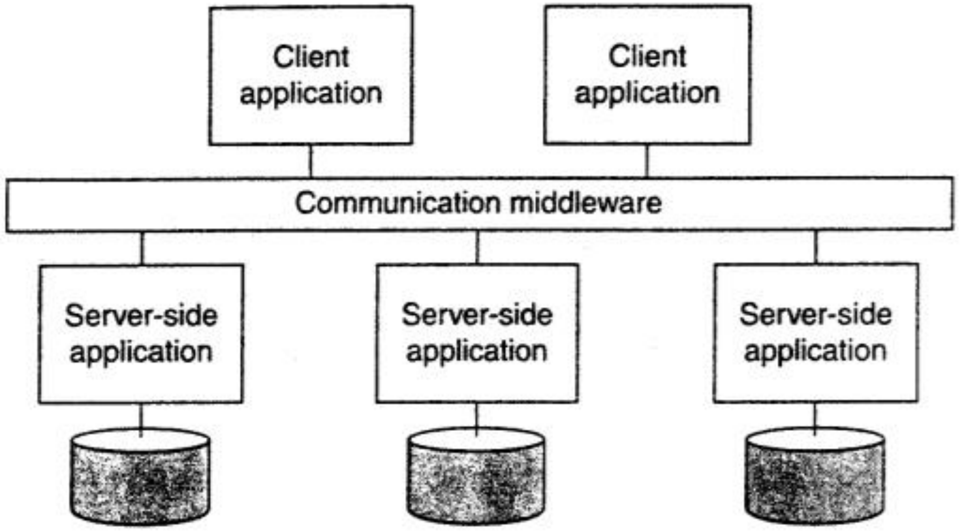
- Distributed Information Systems
  - Networked applications consists of a server running application and making it available to remote programs, called clients.
  - Transaction Processing Systems
    - Transaction primitives
    - Transactions are Atomic, Consistent, Isolated and Durable (ACID).
    - Nested transactions follow a logical division of the work of the original transaction





Distributed Computing Systems

- Distributed Information Systems
  - Enterprise Application Integration
    - Application components should be able to communicate directly with each other and not merely by means of the request/reply behaviour that was supported by transaction processing systems.
    - Existing applications could directly exchange information
    - Remote Procedure Call (RPC)
    - Remote Method Invocation (RMI)



Distributed Computing Systems

- Distributed Pervasive Systems
  - Characterized by being small, battery-powered, mobile, and having only a wireless connection.
  - Unstable behaviour of these distributed systems
  - Lack of human administrative control.
  - Requirements for pervasive applications are
    - Embrace contextual changes.
    - Encourage ad hoc composition.
    - Recognize sharing as the default.
  - Distribution of data, processes, and control is inherent

Distributed Computing Systems

- Distributed Pervasive Systems
  - Home systems
    - Integrates typical consumer electronics such as TVs, audio and video equipment, gaming devices, (smart) phones, PDAs, and other personal wearables into a single system.
    - a system should be completely self-configuring and self-managing.
    - Personal space should be managed
  - Electronic Health care systems
    - Systems are equipped with various sensors organized in a (preferably wireless) body-area network (BAN).

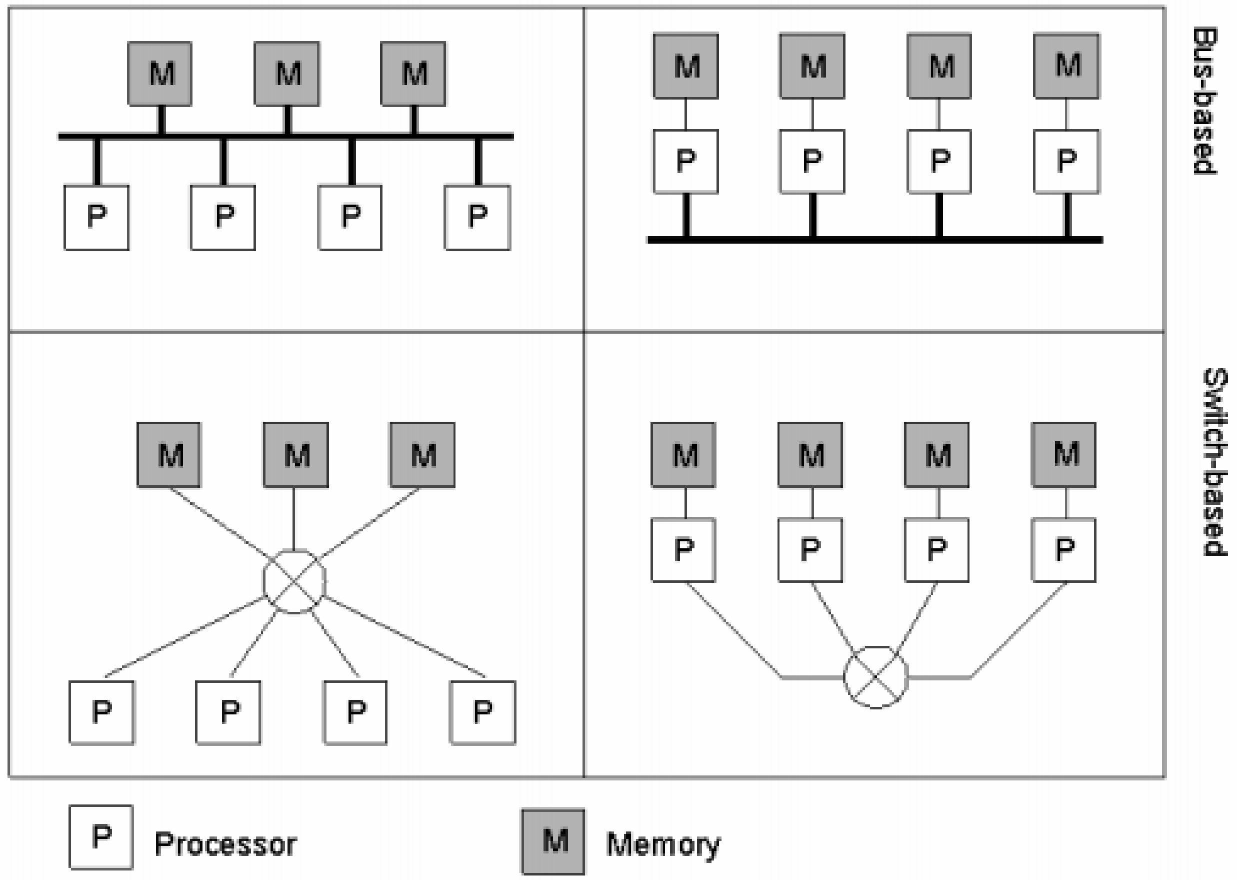
Hardware Concept

- Explains the organization of hardware, their interconnection and the manner in which they communicate with each other.
- The hardware of distributed system has two types
  - Multiprocessor system
  - Multicomputer system
- Both the types are divided into two categories
  - Bus-based architecture
  - Switch-based architecture

Hardware Concept

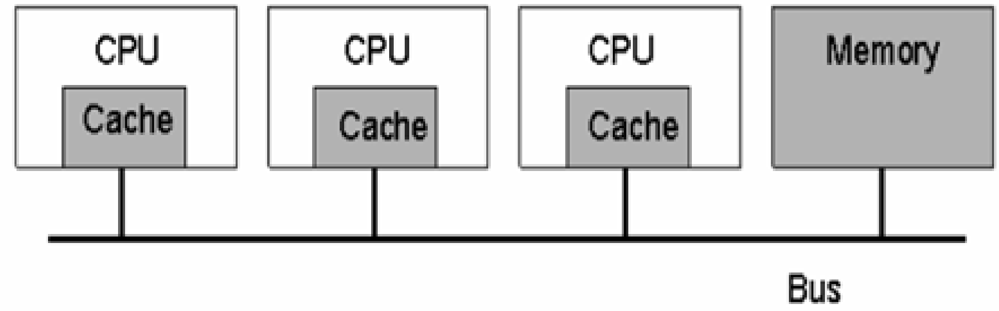
Multiprocessor (Shared Memory)

Multicomputer (private Memory)



Hardware Concept

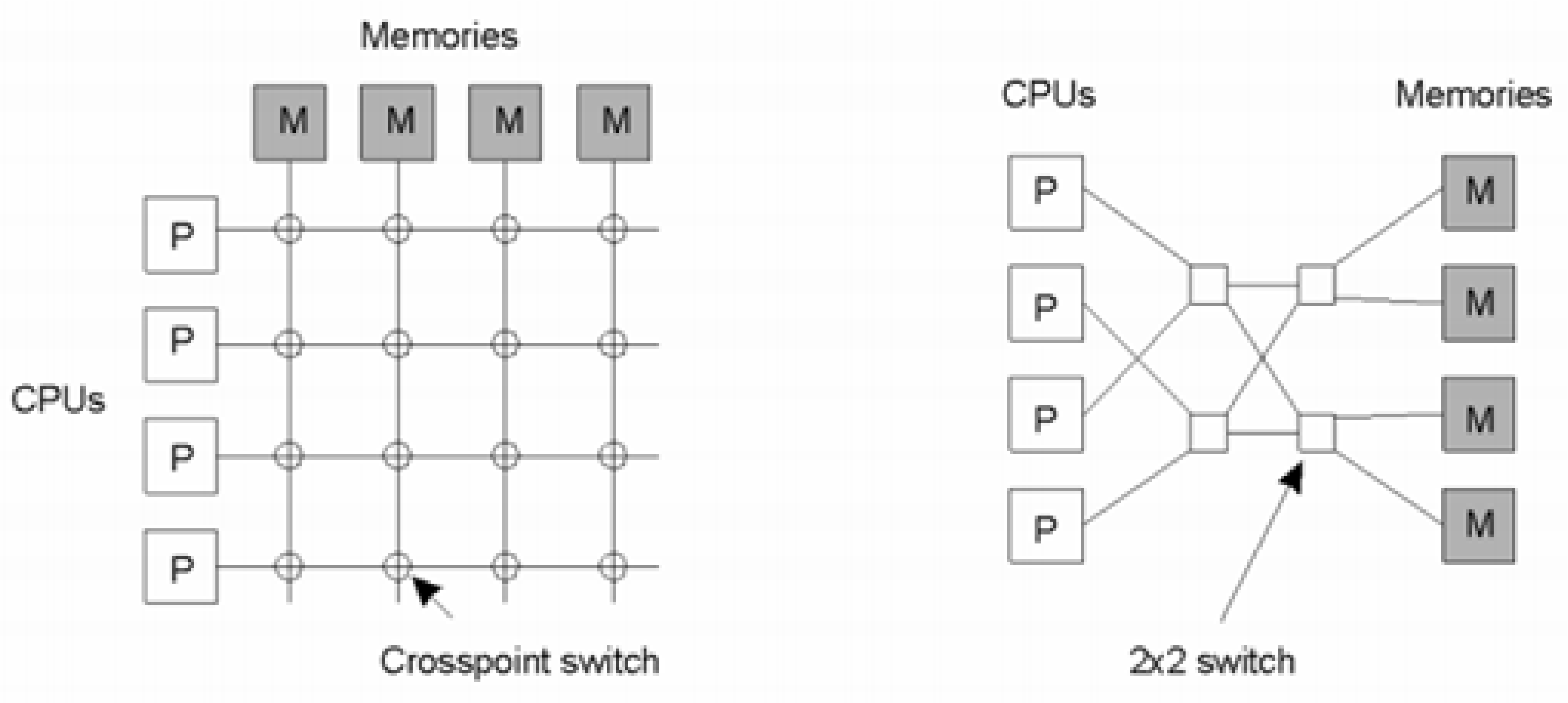
- Multiprocessor system
  - Multiple CPUs and memory modules are connected through bus.



  - Only few numbers of processors are connected in bus.
  - Memory can be incoherent if updated word in one cache is not propagated to other copies of word present in different caches.
  - To allow for more processors, memory and processor modules are connected by crossbar-switch.

Hardware Concept

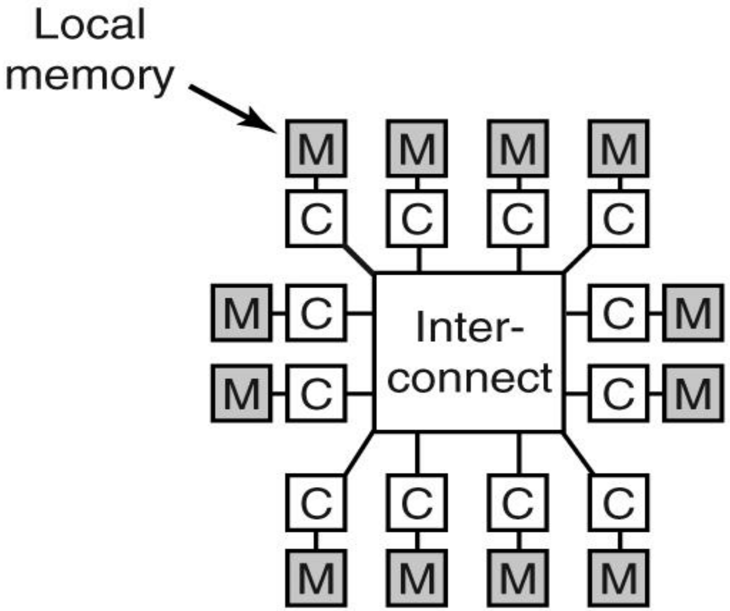
- Multiprocessor system
  - Omega network contains 2×2 switches, each switch having 2 inputs and 2 outputs



Hardware Concept

- Multicomputer system
  - Multiple computers with their independent processors and memories are connected by a computer network.
  - Various network technologies such as Ethernet, FDDI and ATM are used.
  - Message passing is done over a communication network
  - Can be
    - Homogeneous

      Or
    - Heterogeneous



Hardware Concept

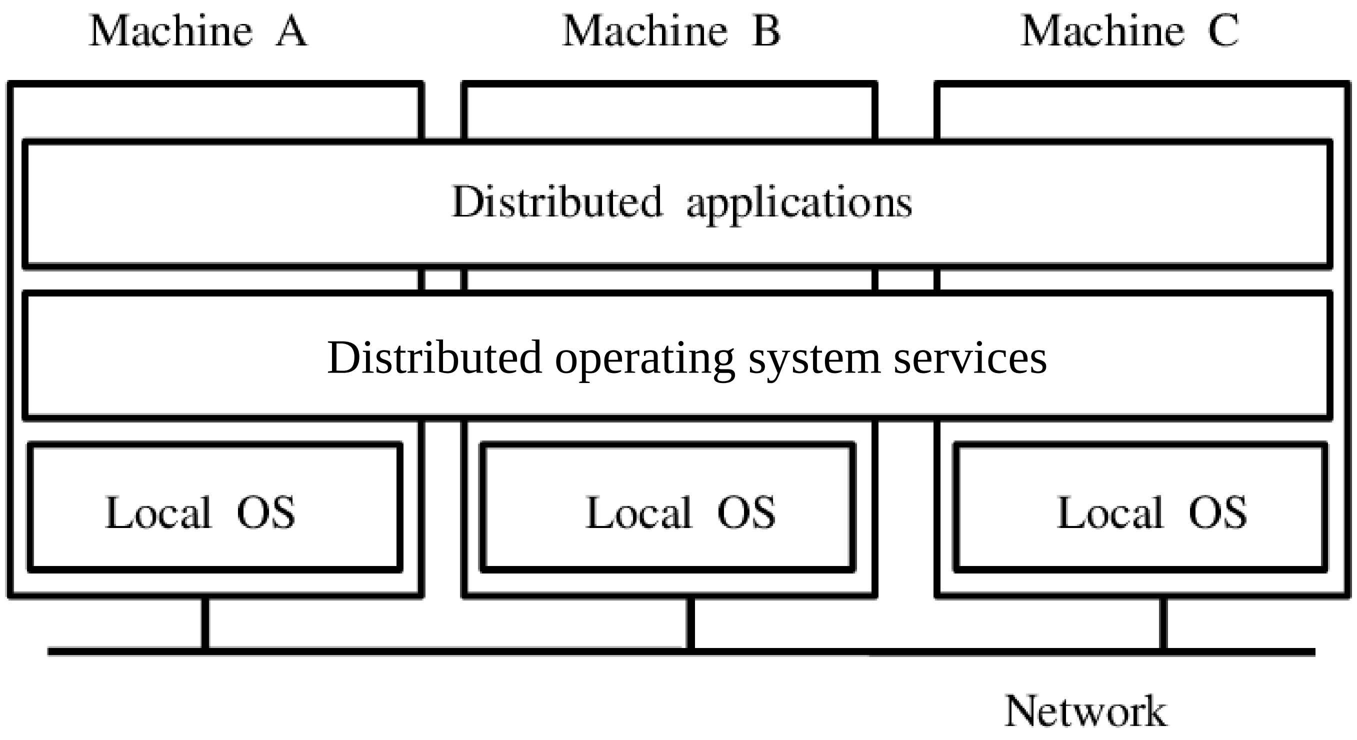
- Multicomputer system
  - Homogeneous Multicomputer Systems
    - Same interconnection network and hardware platform is used
    - Nodes are connected through a high-speed interconnection network.
    - Routing of messages can be done through interconnection network.
  - Heterogeneous Multicomputer Systems
    - Multicomputers have different processors, memory sizes i.e different hardware platforms
    - Interconnection networks have different technology.

Software Concept

- Software of a distributed system is a suitable operating system used to facilitate interaction between a user and the hardware.
- Three types of software is used in distributed systems
  - Distributed Operating System (DOS)
  - Network Operating System (NOS)
  - Middleware

Software Concept

- Distributed Operating System (DOS)
  - A common set of services is shared amongst multiple processors
  - Provides services to separate independent computers connected in the network

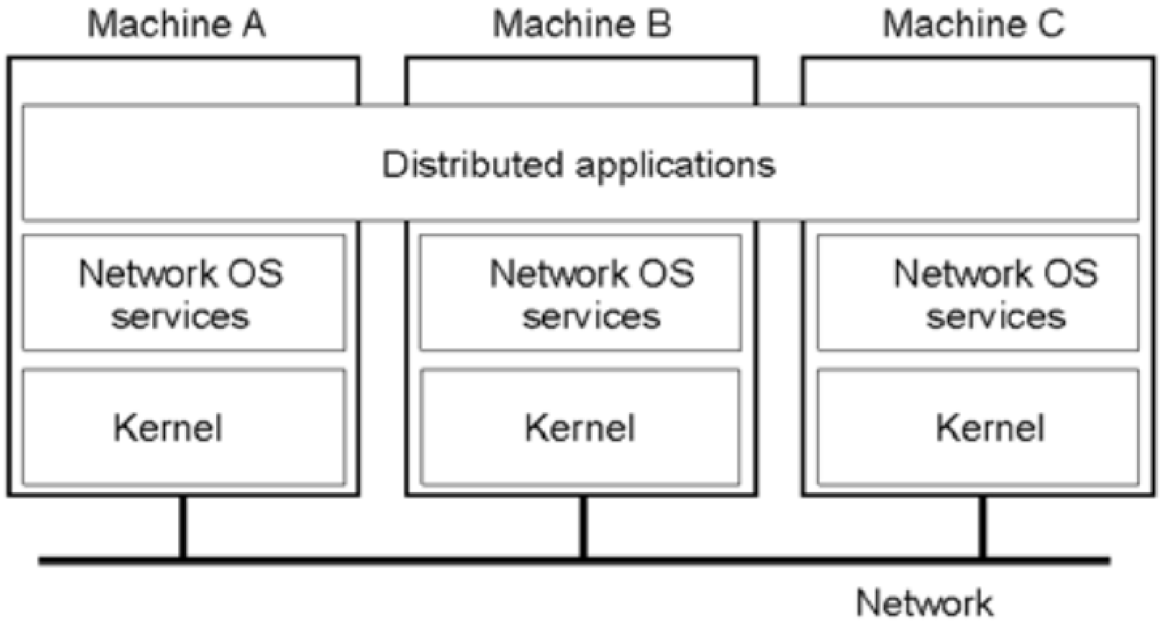


Software Concept

- Distributed Operating System (DOS)
  - Communication with all computers in multicomputer system is carried out using message passing interface (MPI).
  - Possesses tightly coupled architecture
  - E.g. Railway reservation systems, automated banking system etc.
  - Lack of scalability

Software Concept

- Network Operating System (NOS)
  - Specifically designed for heterogeneous multicomputer systems
  - Provides separate set of system services independently to each computer
  - It has loosely coupled architecture.



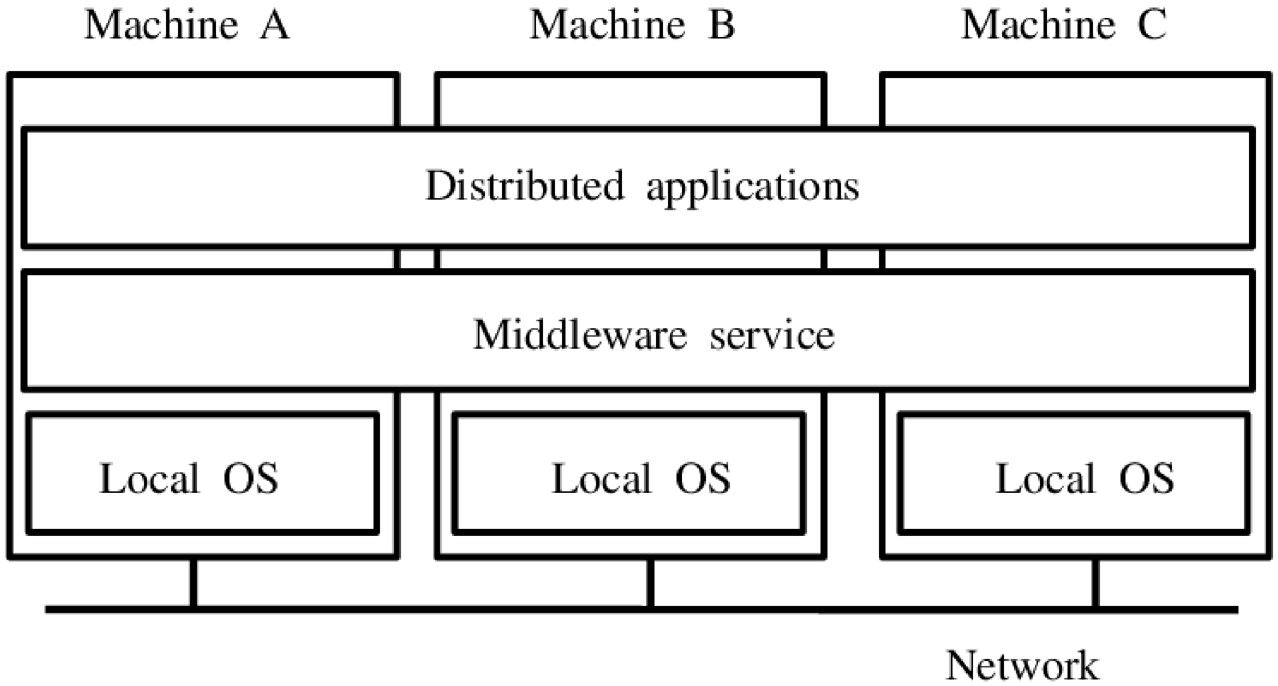
Software Concept

- Network Operating System (NOS)
  - E.g. remote login, centralized file storage system
  - NOS is scalable, where a large no. of resources and users are supported.
  - It fails to provide a single coherent view.

- Middleware
  - To overcome disadvantages of DOS and NOS, new layer is formed called as middleware
  - Uses combination of DOS and NOS services

Middleware

- It uses a combination of DOS and NOS services.
- Common set of services is provided for local applications
- An independent set of services is provided for remote applications
- Supports heterogeneity



Middleware

- Functions of Middleware
  - Hide the distributed nature of the application
  - Hide the heterogeneity of the enterprise
  - Provide uniform, standard, high-level interfaces to the application developers and integrators
  - Supply a set of common services to perform various general purpose functions
- Middleware makes application development easier
  - by providing common programming abstractions,
  - by masking application heterogeneity and the distribution of the underlying hardware and operating systems, and
  - by hiding low-level programming details.

Models of Middleware

- Remote Procedure Call (RPC)
  - Used to call a procedure residing on a remote machine using a local call
  - Server receives a remote procedure call from client
  - Server locates method definition and passes parameter to calculate results
  - Results are returned to the client
  - This is synchronous communication

- Message Oriented Middleware (MOM)
  - Used to send and receive communication messages between multiple clients and servers.
  - Uses queue data structure to store and retrieve messages
  - Asynchronous communication

Models of Middleware

- Remote Method Invocation (RMI)
  - Invokes the method of a remote object.
  - Client can access remote objects using interfaces
  - Objects are distributed and located using RMI registry
  - Does not support heterogeneity

- Common Object Request Broker Architecture (CORBA)
  - Objects can be accessible from remote location through ORB interface
  - Supports heterogeneity
  - A server and client communicate using object request broker bus, where interoperable protocols are used to facilitate the communication and locate objects.

Services Offered by Middleware

- Naming
  - Allows entities to be shared and looked up
- Persistence
  - Databases or facilities to connect to databases
- Messaging
  - Send and receive messages
- Querying
  - Query on distributed objects
- Concurrency
  - Allows multiple read and write operations to occur simultaneously
- Security
  - Security to shared resources

Key Points to remember

- ✓ Distributed system definition
- ✓ Main goal is to make it easy for users to access remote resources and share them in controlled way.
- ✓ Types of transparencies
- ✓ Types of distributed systems
- ✓ Middleware models