

CS-251: Parallel and Distributed Computing

Lecture 06 – Networks of Workstations-Distributed Memory

Dr. Zeeshan Rafi

PhD MIS, MPhil IT, BS Software Engineering
Former Software Engineer, Database Administrator
System Analyst, Big Data Analyst
Member Turkish Scientific & Technological Research Council

Department of Computing and Information Technology

Istanbul University, TR



KHAS University, TR



University of Gujrat, PK

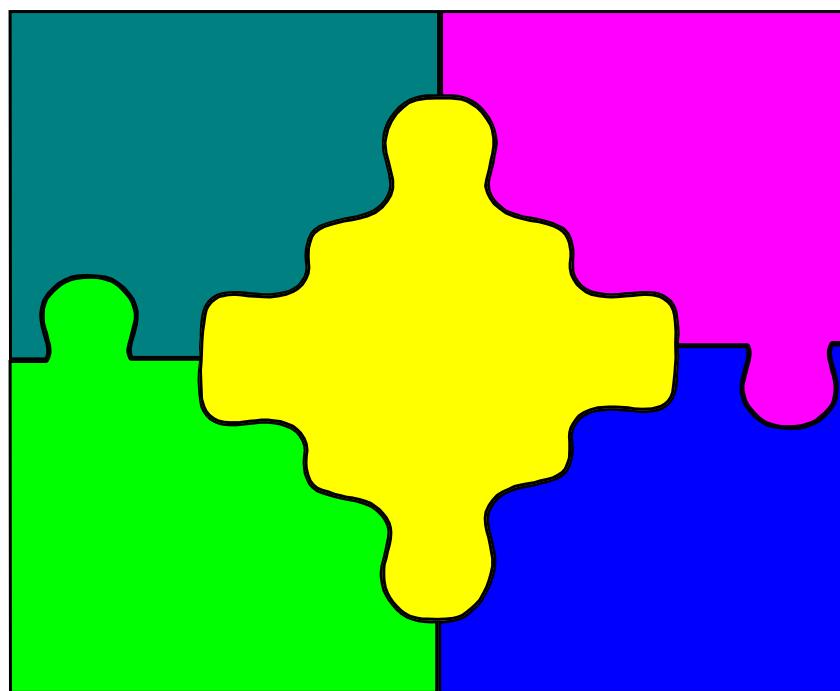


GCF University, Pk



Networks Of Workstations- Distributed Memory

Distributed Processing and Client/Server



Distributed systems

- Remote computers cooperate via a network to appear as a local machine
- Users are given the impression that they are interacting with just one machine
- Spread computation and storage throughout a network of computers
- Applications are able to execute code on local machines and remote machines and to share data, files and other resources among these machines
- Attributes of distributed systems: Performance, Scalability, Connectivity, Security, Reliability, Fault tolerance

Reliability and Fault Tolerance

- Fault tolerance
 - Implemented by providing replication of resources across the system
- Replication
 - Offers users increased reliability and availability over single-machine implementations
 - Designers must provide mechanisms to ensure consistency among the state information at different machines

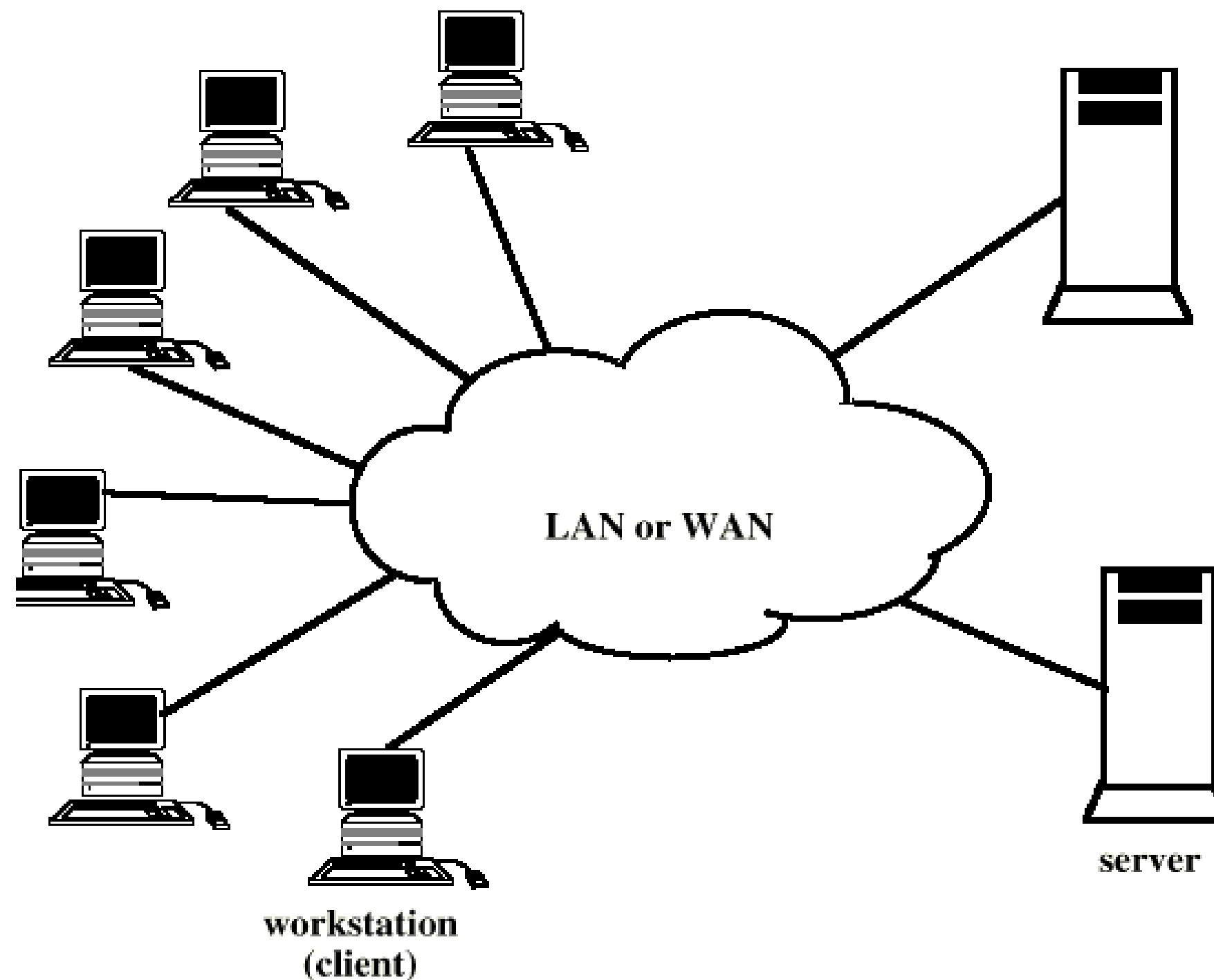
Client/Server Computing

- Client machines are generally single-user PCs or workstations that provide a highly user-friendly interface to the end user
- Each server provides a set of shared user services to the clients
- The server enables many clients to share access to the same database and enables the use of a high-performance computer system to manage the database

Client/Server Computing

- Involves splitting an application into tasks and putting each task on the platform where it can be handled most efficiently
- Processing for the presentation on the user's machine
- Data management and storage on a server
- Involves a network

Generic Client/Server Environment



Client/Server Applications

- Platforms and the operating systems of client and server may differ
- These lower-level differences are irrelevant as long as a client and server share the same communications protocols and support the same applications

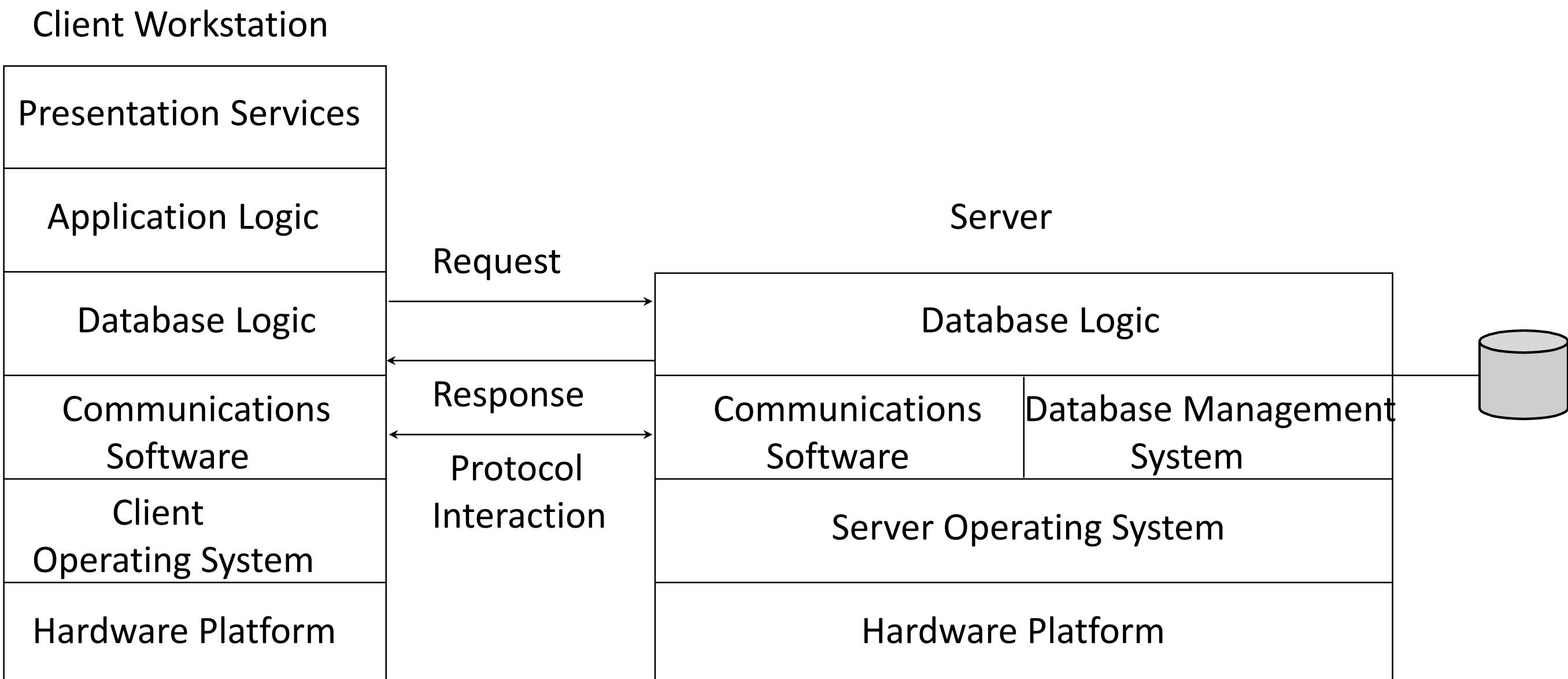
Client/Server Applications

- Actual functions performed by the application can be split up between client and server
- Optimize platform and network resources
- Optimize the ability of users to perform various tasks
- Optimize the ability to cooperate with one another using shared resources

Database Applications

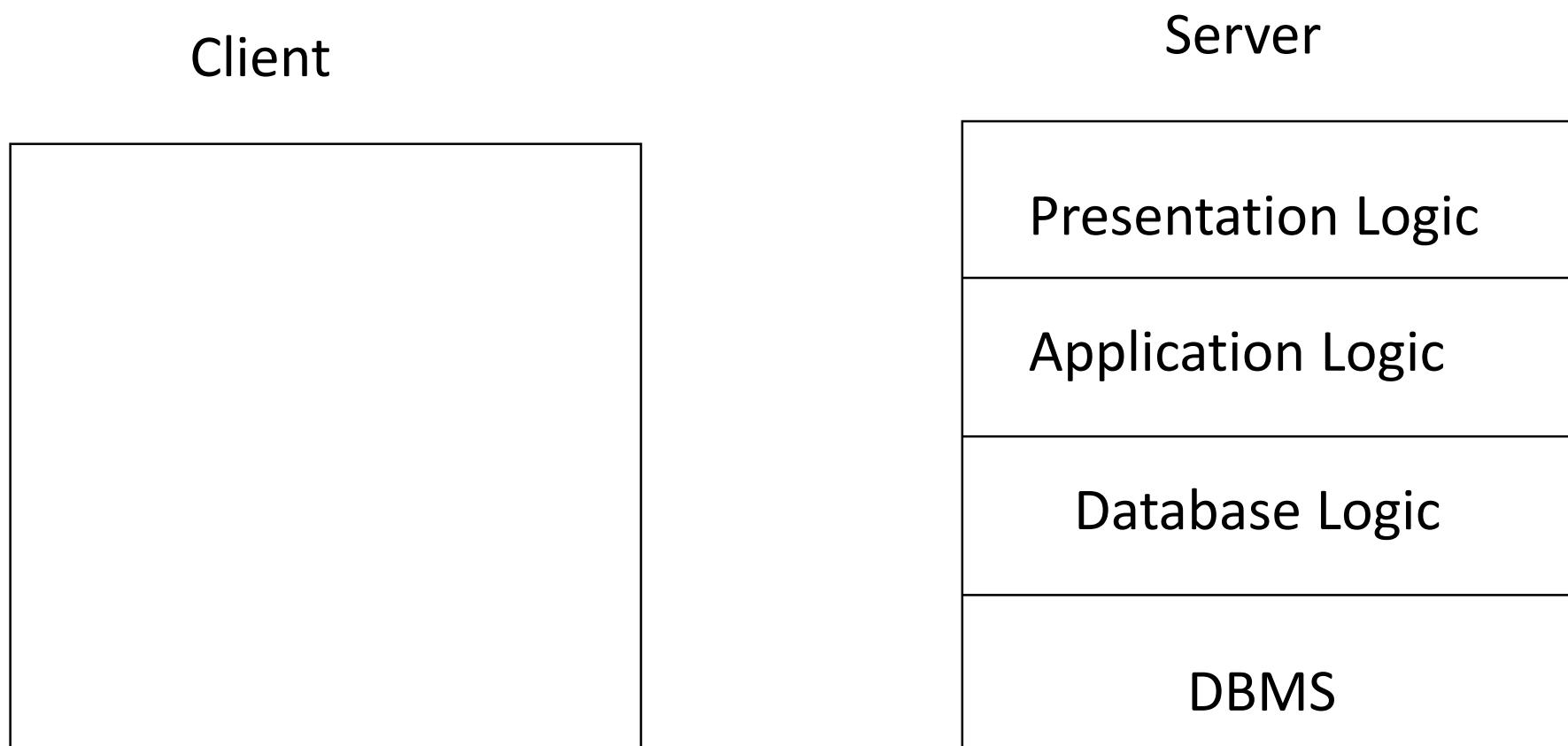
- The server is a database server
- Interaction between client and server is in the form of transactions
 - the client makes a database request and receives a database response
- Server is responsible for maintaining the database

Client/Server Architecture for Database Applications



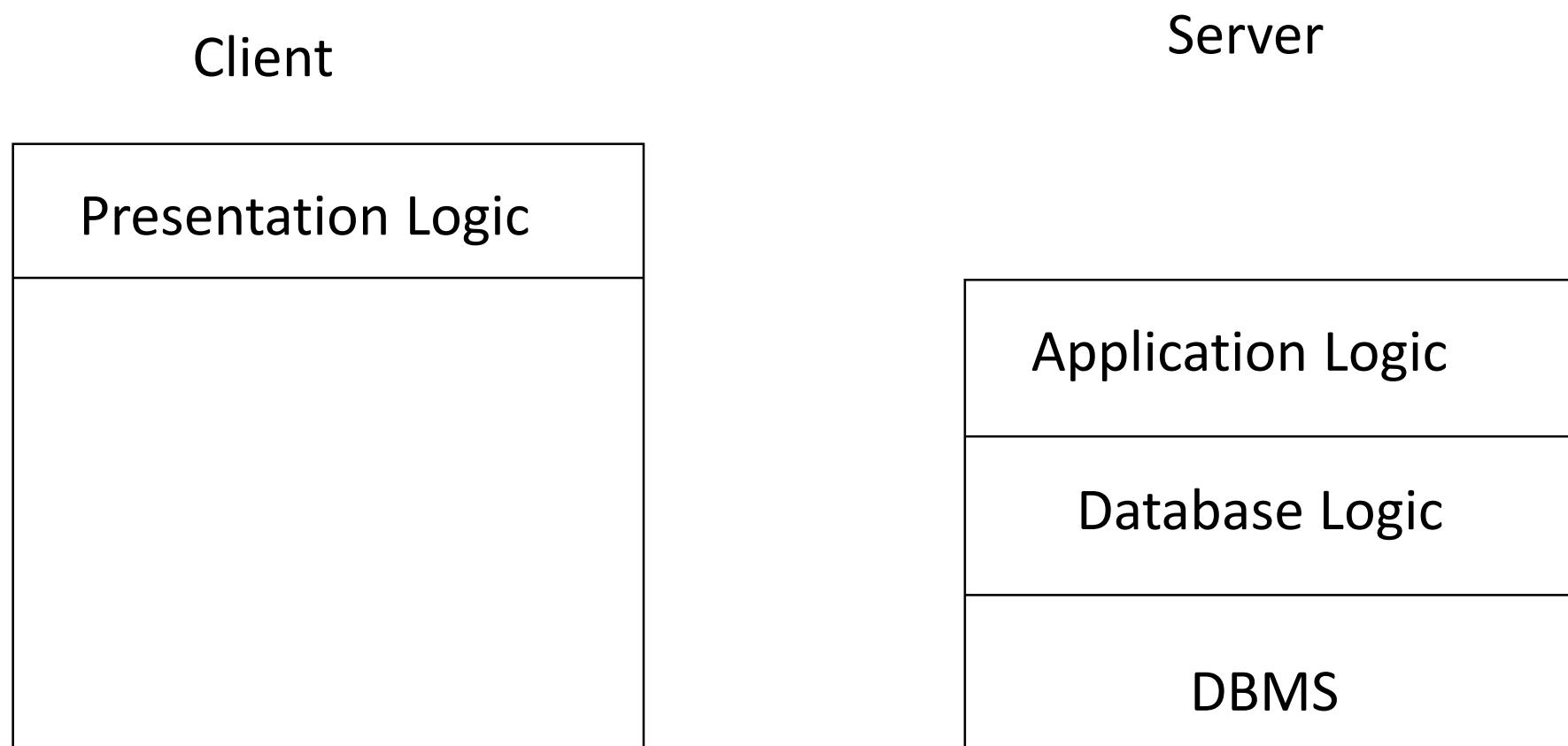
Classes of Client/Server Applications

- Host-based processing
 - not true client/server computing
 - traditional mainframe environment



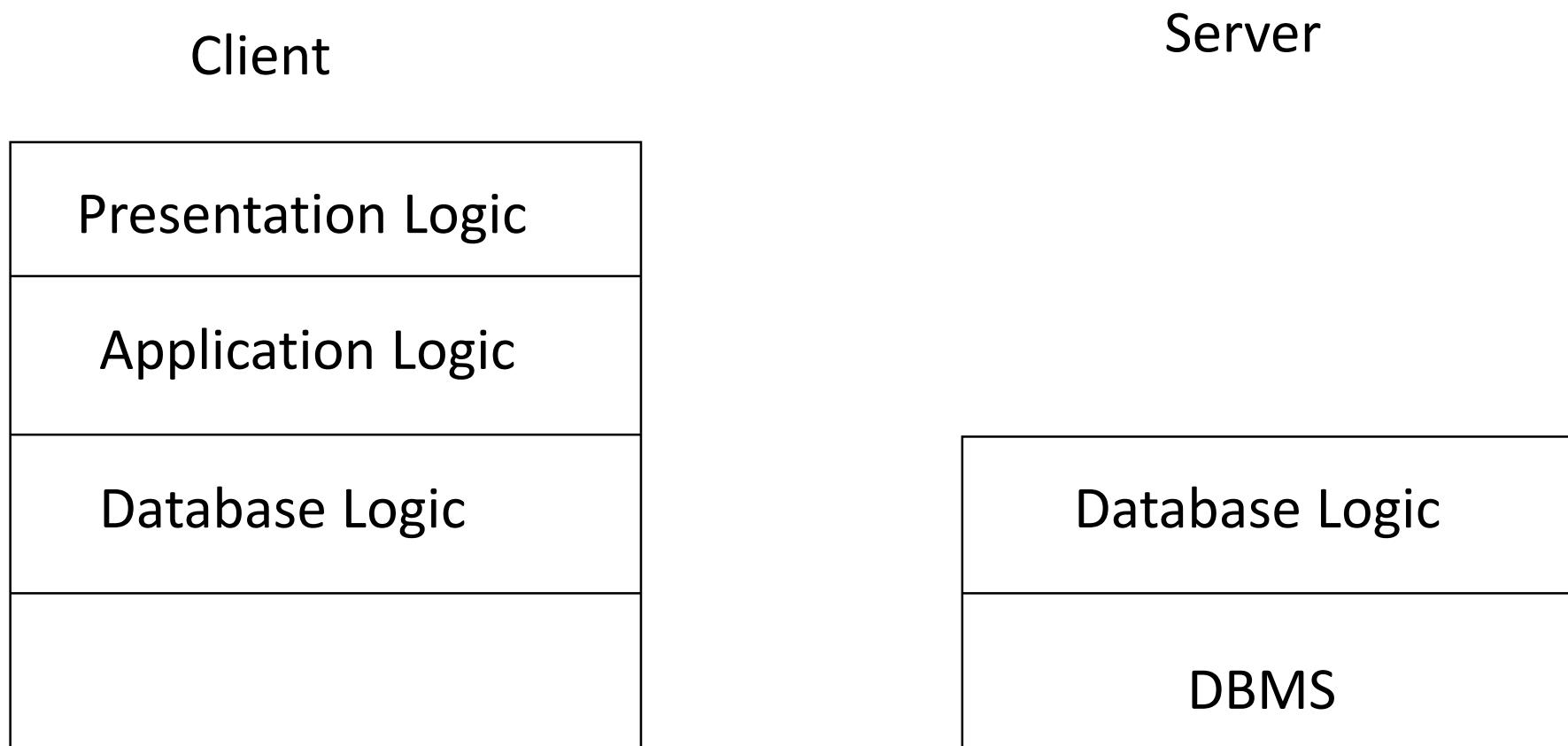
Classes of Client/Server Applications

- Server-based processing
 - server does all the processing
 - user workstation provides a user-friendly interface



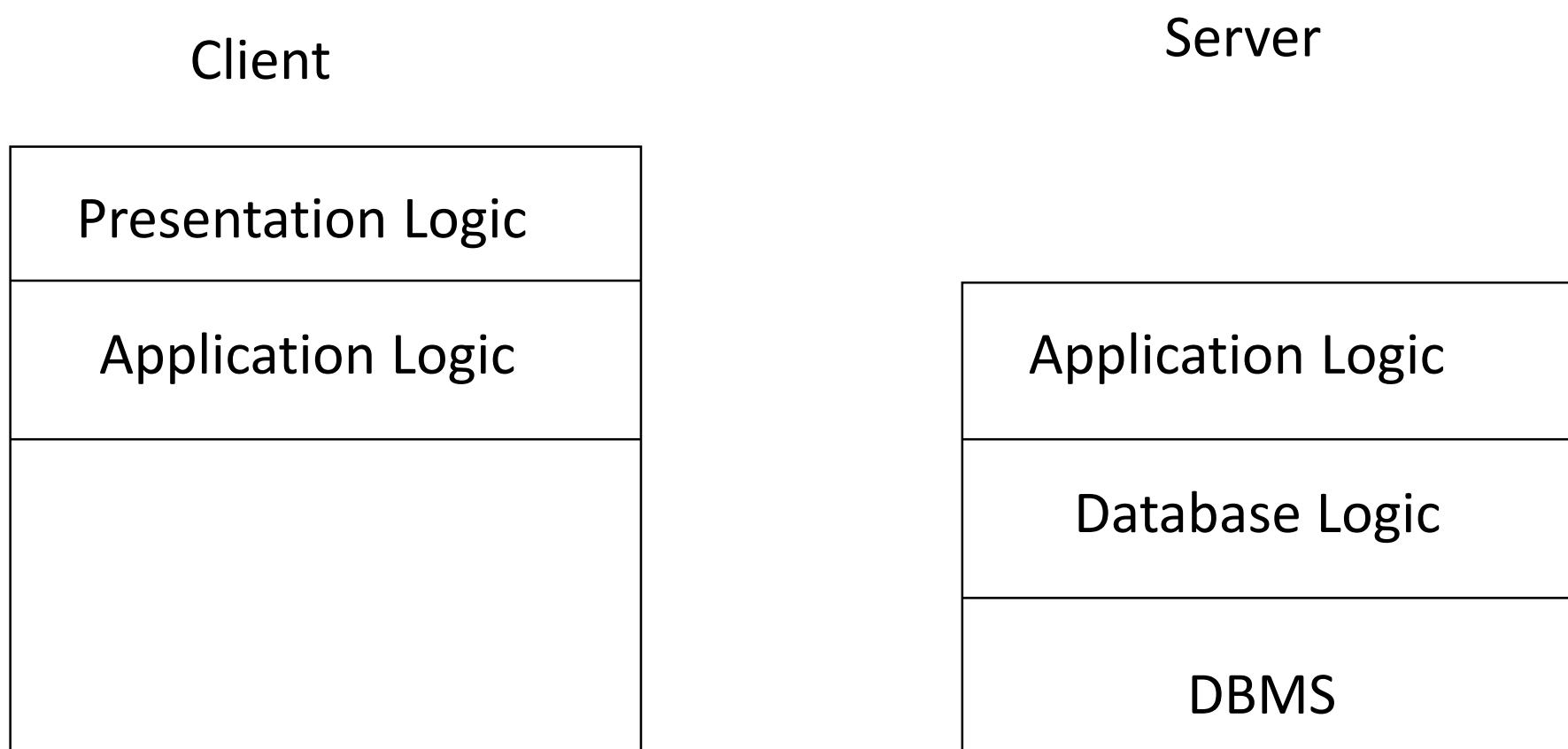
Classes of Client/Server Applications

- Client-based processing
 - all application processing may be done at the client
 - data validation routines and other database logic function are done at the server



Classes of Client/Server Applications

- Cooperative processing
 - application processing is performed in an optimized fashion
 - complex to set up and maintain



File Cache Consistency

- File caches hold recently accessed file records
- Caches are consistent when they contain exact copies for remote data
- File-locking prevents simultaneous access to a file
 - writing causes the server cached to be updated

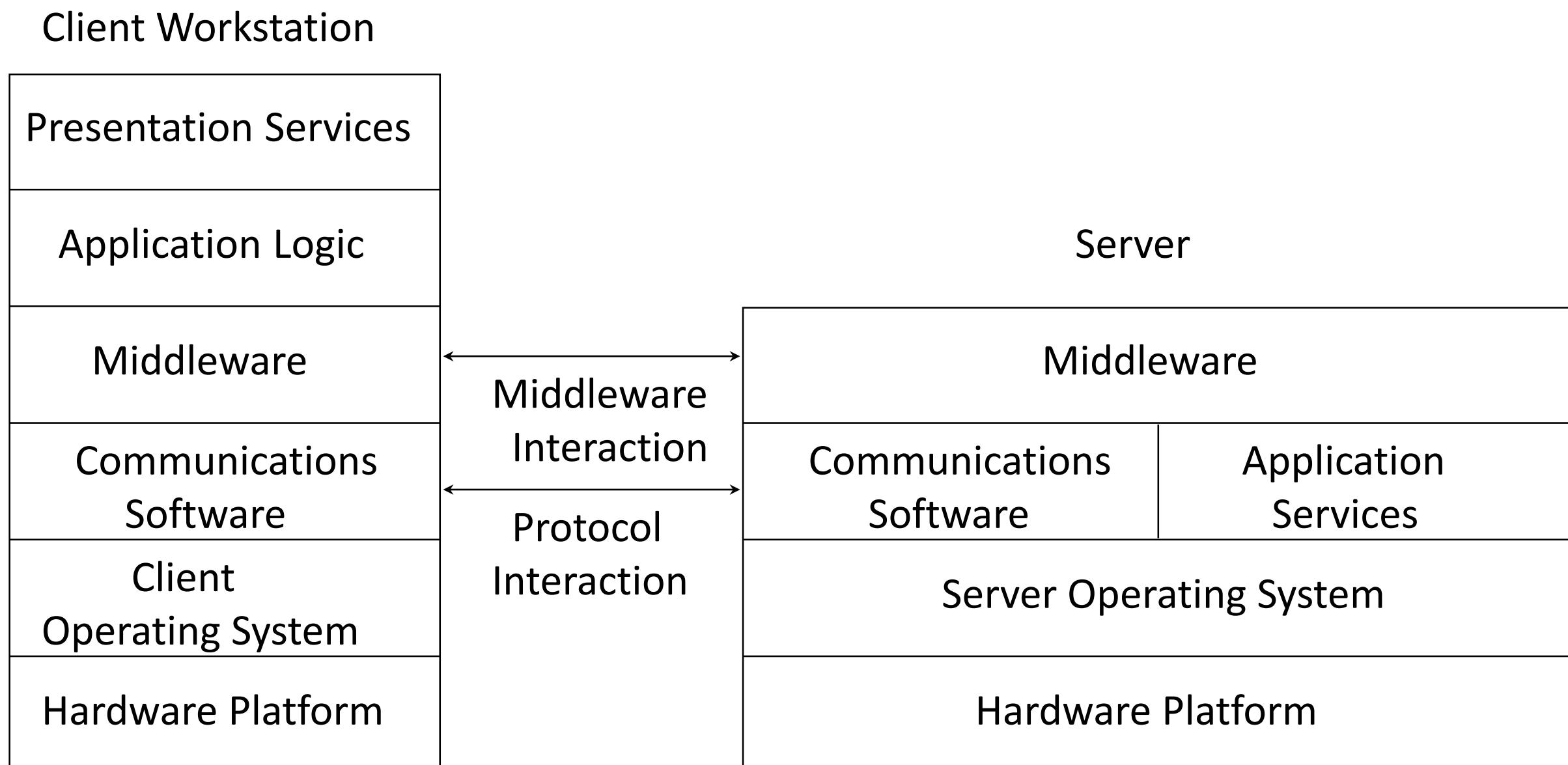
Middleware

- Set of tools that provide a uniform means and style of access to system resources across all platforms
- Enable programmers to build applications that look and feel the same
- Enable programmers to use the same method to access data

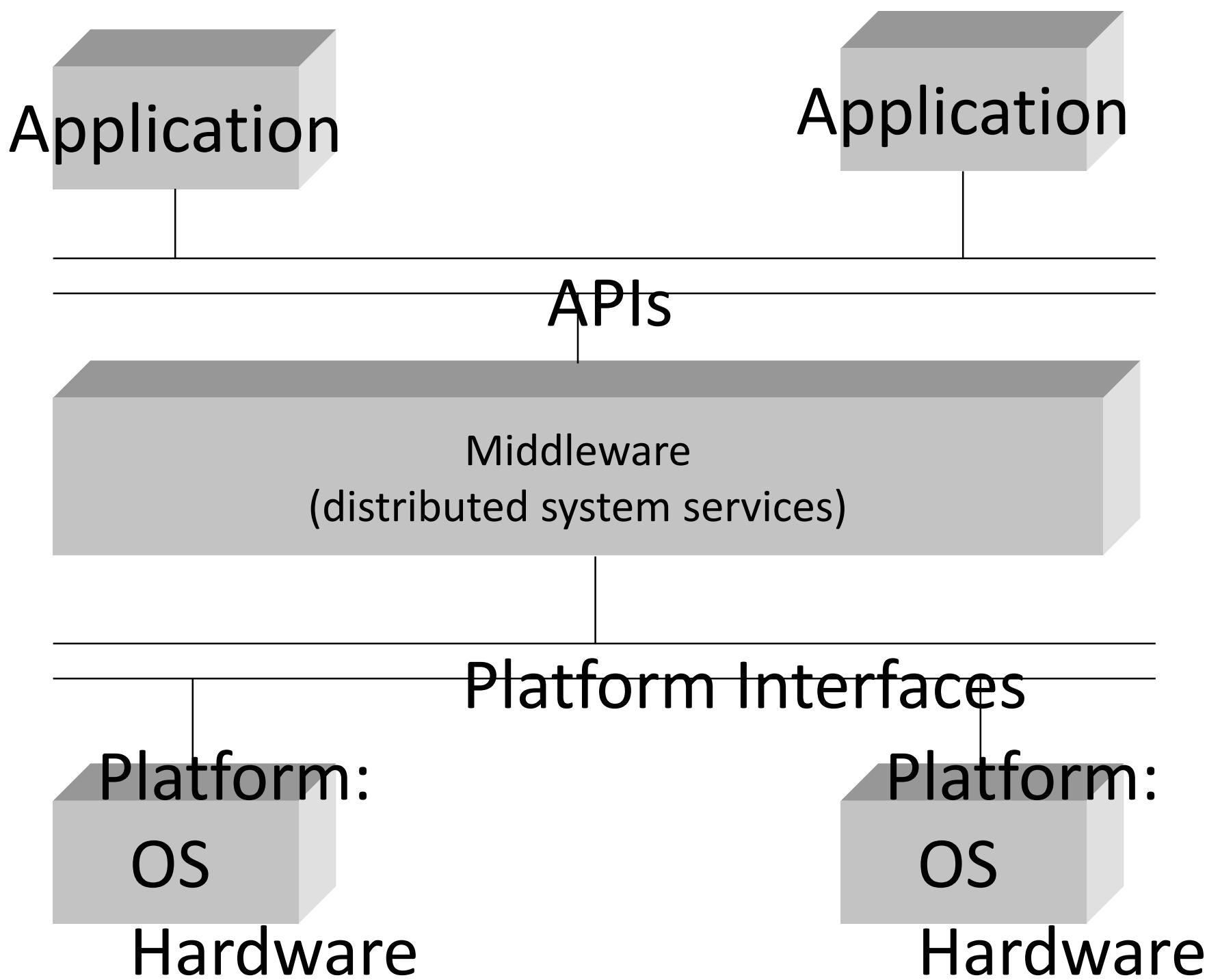
Middleware

- Use of a standard programming interface and protocols
- There is both a client and server component to middleware
- Provides uniform access to different systems

The Role of Middleware in Client/Server Architecture



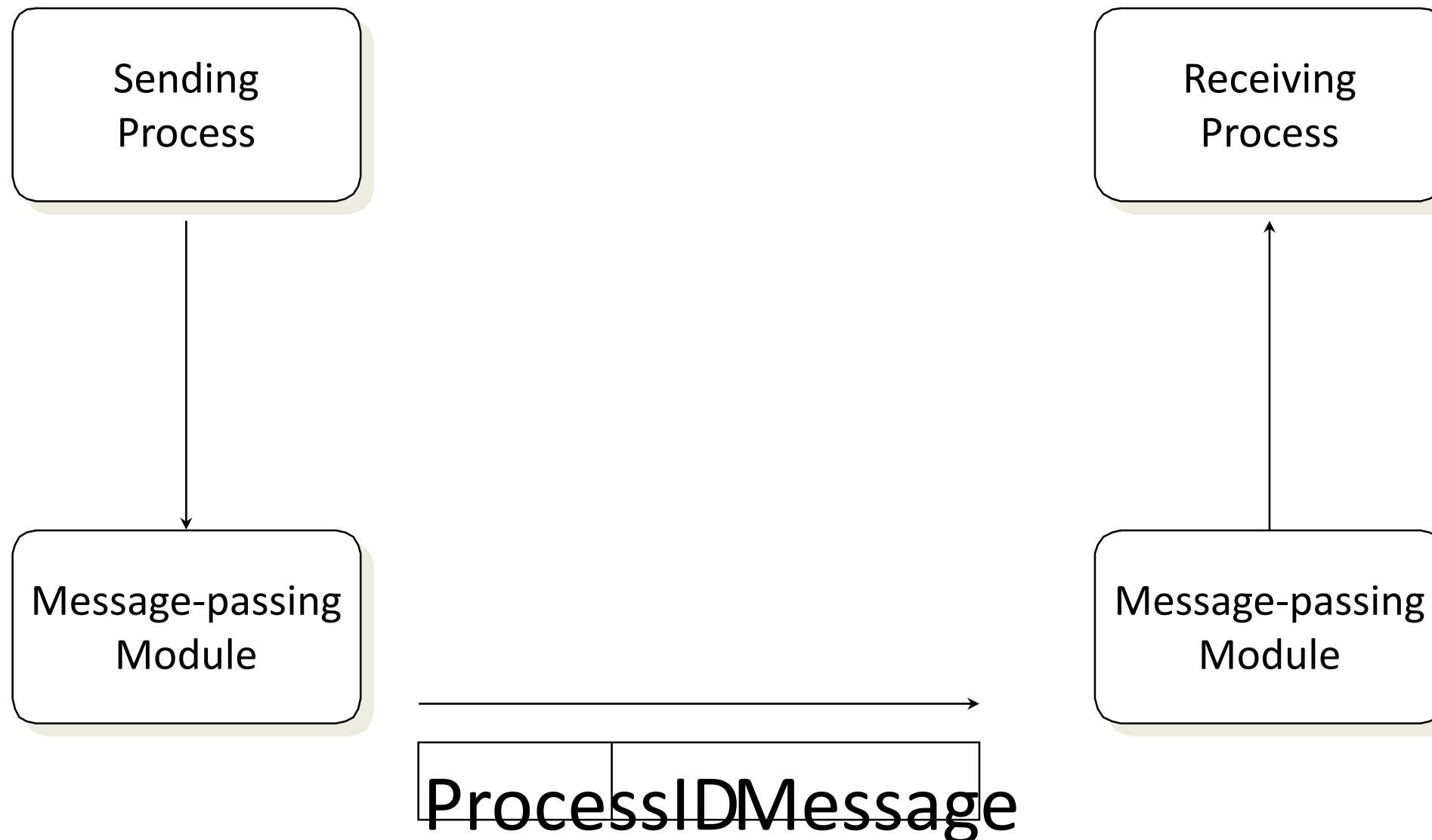
Logical View of Middleware



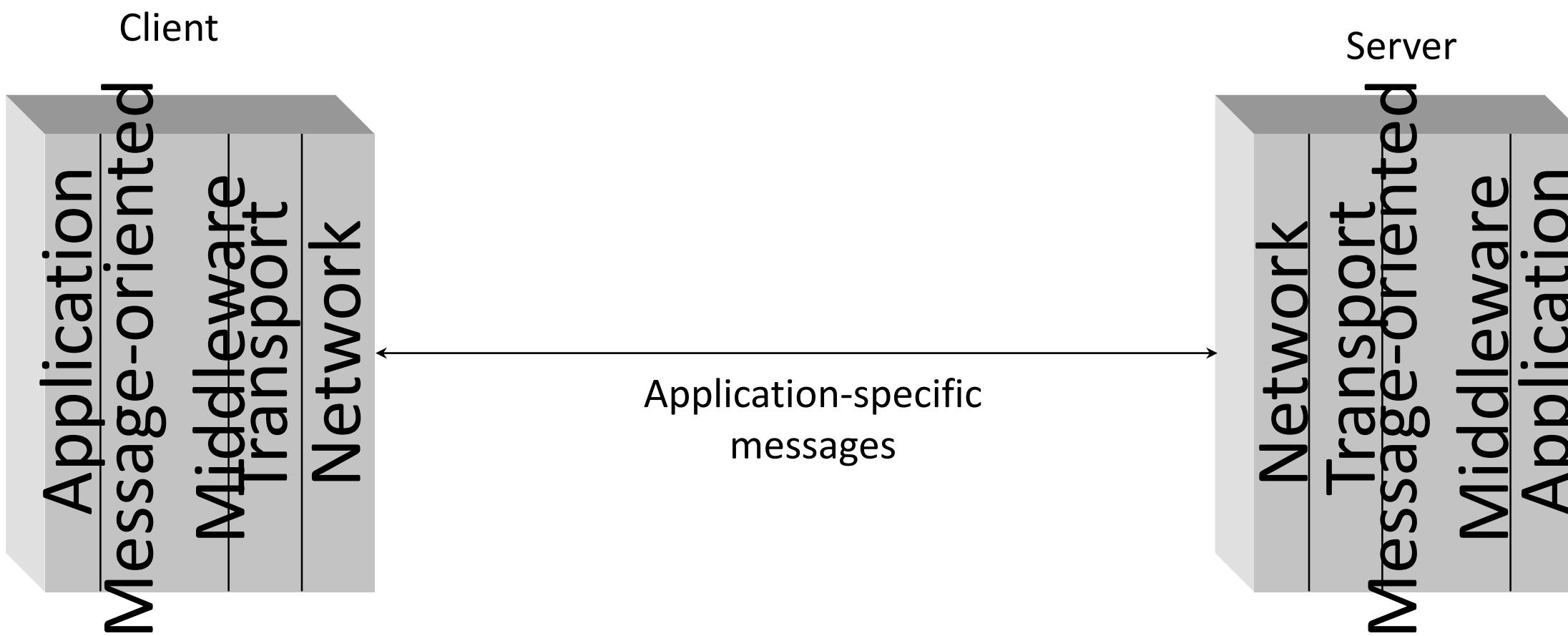
Distributed Message Passing

- Message passed used to communicate among processes
- Send and receive messages as used in a single system
OR
- Remote procedure calls

Basic Message-Passing Primitives



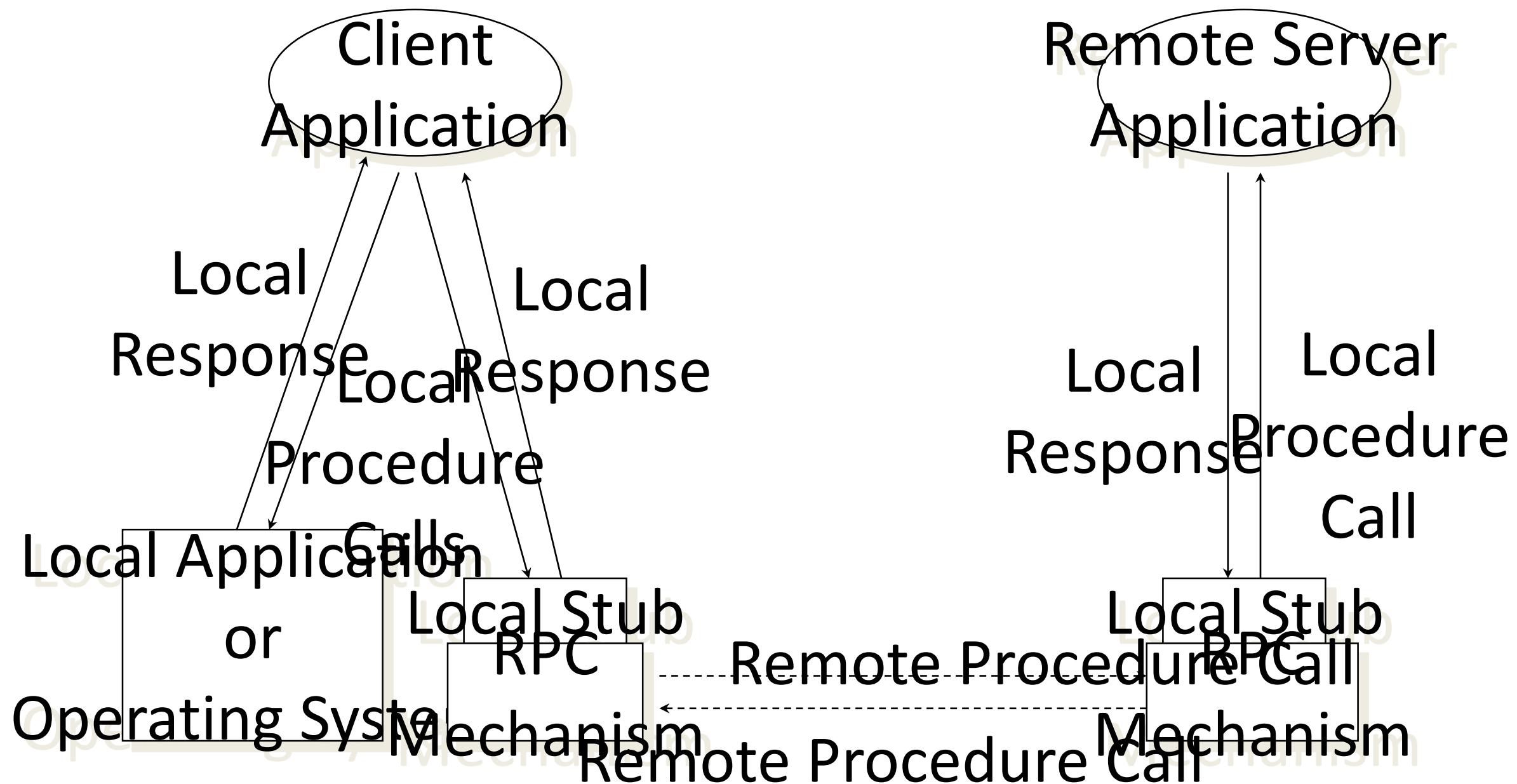
Message-oriented Middleware



Remote Procedure Calls

- Allow programs on different machines to interact using simple procedure call/return semantics
- Widely accepted
- Standardized
 - client and server modules can be moved among computers and operating systems easily

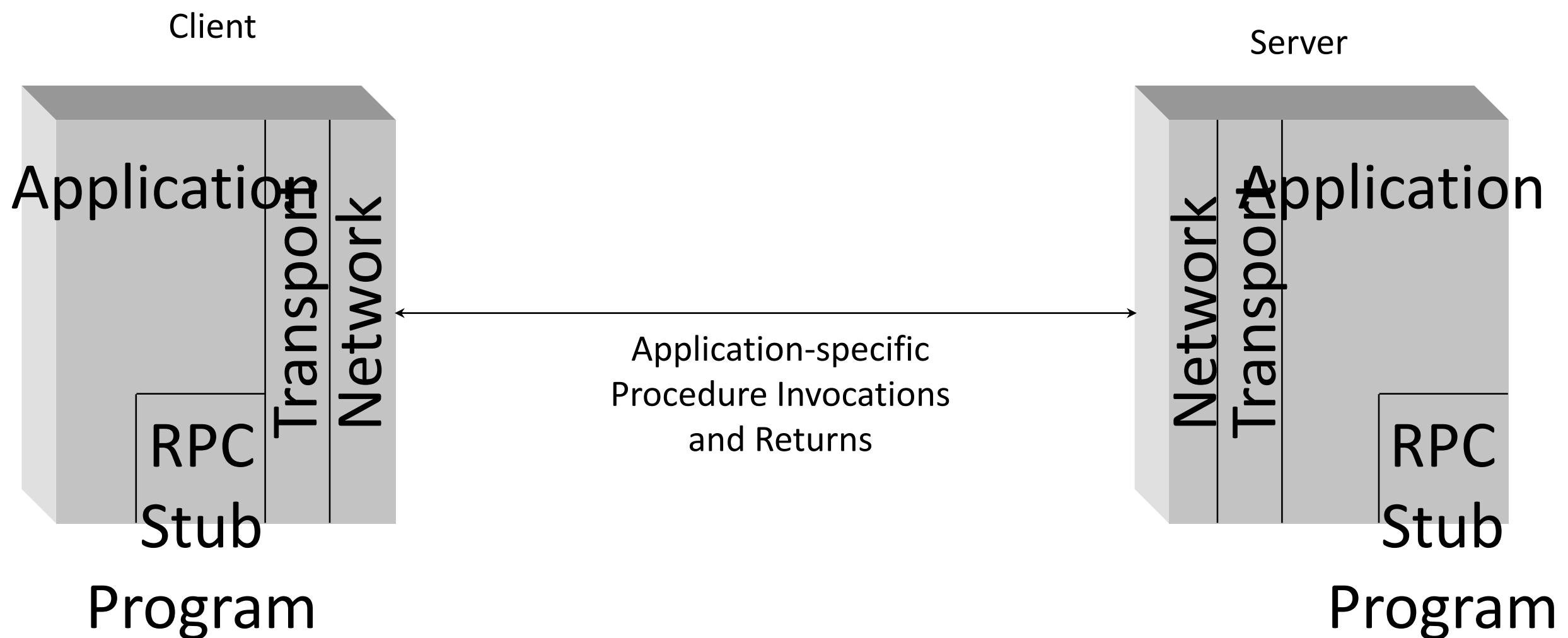
Remote Procedure Call Mechanism



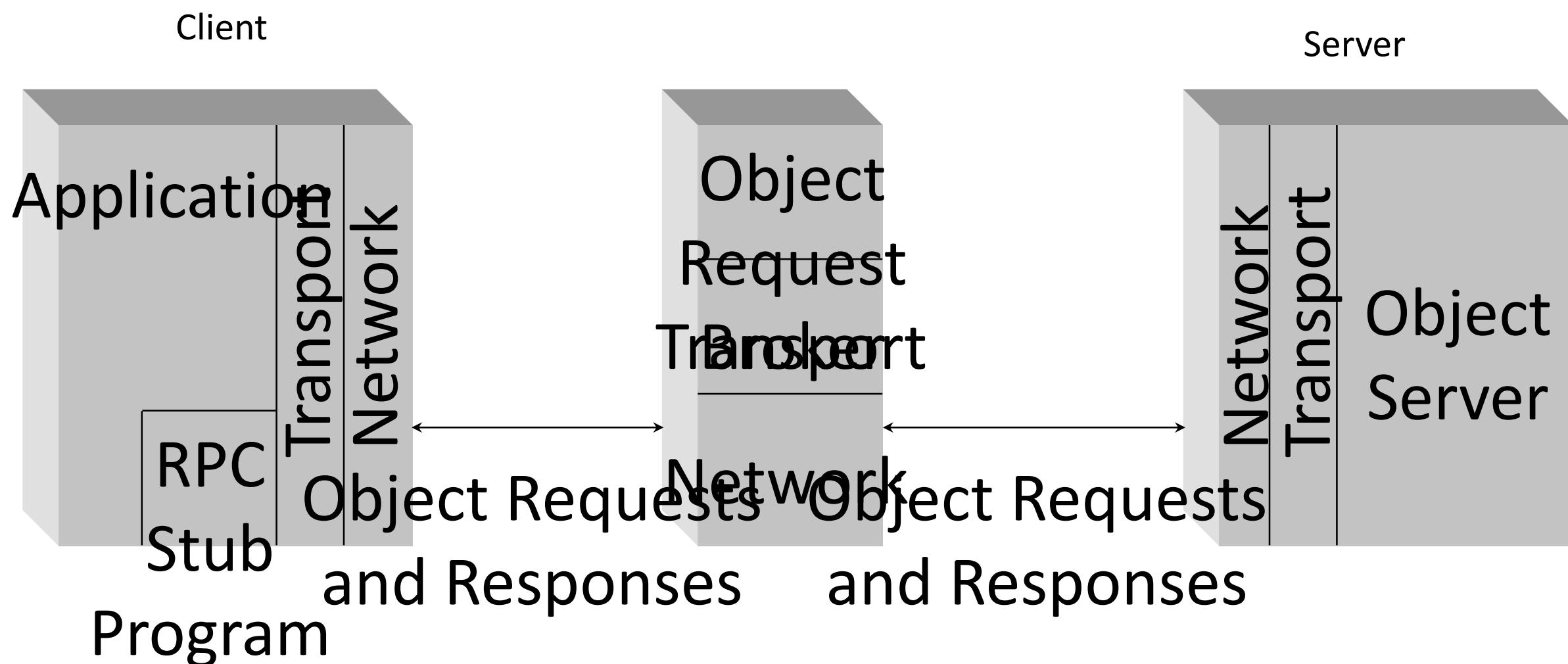
Client/Server Binding

- Binding specifies the relationship between remote procedure and calling program
- Nonpersistent binding
 - logical connection established during remote procedure call
- Persistent binding
 - connection is sustained after the procedure returns

Remote Procedure Calls



Object Request Broker



CORBA (Common Object Request Broker Architecture)

- CORBA
 - Open standard designed to enable interoperation among programs in heterogeneous as well as homogeneous systems
 - Supports objects as parameters or return values in remote procedures during interprocess communication
- CORBA implementation
 - The process on the client passes the procedure call along with the required arguments to the client stub
 - The client stub marshals the parameters and sends the procedure call through its Object Request Broker (ORB), which communicates with the ORB on the server
 - CORBA provides programmers language independence with the Interface Definition Language (IDL), which allows them to strictly define the procedures that can be called on the object

DCOM (Distributed Component Object Model)

- DCOM
 - Designed to allow software components residing on remote computers to interact with one another
 - As in CORBA, objects in DCOM are accessed via interfaces
 - Unlike CORBA, however, DCOM objects may have multiple interfaces
 - When a client requests a DCOM object from a server, the client must also request a specific interface of the object

Workstation

- Workstation, a high-performance computer system than mainstream personal computers, that is basically designed for a single user and has *advanced graphics capabilities, large storage capacity, and a powerful central processing unit.*
- A workstation is *more capable than a personal computer* (PC) but is *less advanced than a server* (which can manage a large network of peripheral PCs or workstations and handle immense data-processing and reporting tasks).

Workstation cont...

- Workstations were the first segment of the computer market to present advanced accessories and collaboration tools.
- The increasing capabilities of mainstream PCs in the late 1990s have blurred the lines between PCs and technical/scientific workstations.

Network of Workstations (NOW)

- High-speed networks and rapidly improving microprocessor performance make *networks of workstations* an increasingly *appealing vehicle for parallel computing*.
- By relying solely on commodity hardware and software, networks of workstations offer parallel processing at a relatively low cost.

Network of Workstations (NOW) cont... commodity hardware and software

- **Commodity hardware** (unlike ***purpose-built hardware***), sometimes known as ***off-the-shelf hardware***, is a computer device or IT component that is relatively ***inexpensive, widely available*** and basically ***interchangeable with other hardware*** of its type.

Network of Workstations (NOW) cont...

- A network-of-workstations multiprocessor may be realized as a
- processor bank, a number of processors dedicated for the purpose of
 - providing computing cycles.
- Alternatively, it may consist of a dynamically varying set of machines
 - on which idle cycles are used to perform long-running computations.

Network of Workstations (NOW) cont...

- In the latter case, the (hardware) *cost is essentially zero*, since many organizations already have *extensive workstation networks* in place.
- In terms of *performance*, *improvements in processor speed* and *network bandwidth* and *latency* allow *networked workstations* to *deliver performance approaching* or *exceeding supercomputer* performance for an increasing class of applications.

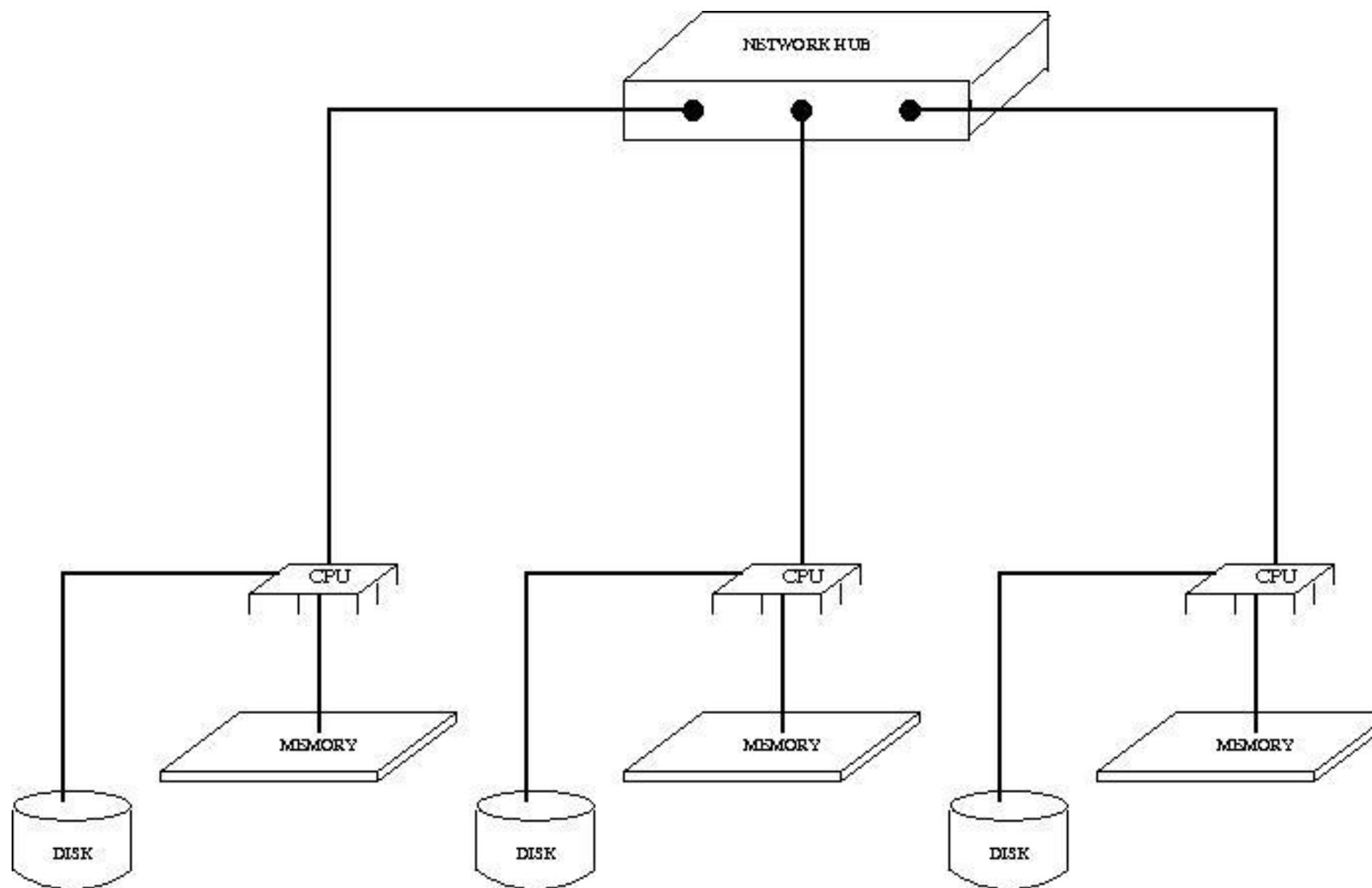
Network of Workstations (NOW) cont...

Distributed Memory

- Distributed memory refers to a multiprocessor computer system in which *each processor has its own private memory*.
- Computational tasks can only operate on *local data*, and if *remote data* are required, the computational task must *communicate with one or more remote processors*.

Network of Workstations (NOW) cont...

Distributed Memory



Network of Workstations (NOW) cont...

Distributed Memory

- In a distributed memory system there is typically a ***processor***, a ***memory***, and some ***form of interconnection*** that allows programs on each processor to interact with each other.

Network of Workstations (NOW) cont...

Distributed Memory (programming issues)

- The key issue in **programming** distributed memory systems is **how to distribute the data over the memories.**
- Depending on the problem solved, the data can be **distributed statically**, or it can **be moved through the nodes.**
- Data can be **moved on demand**, or data can be **pushed to the new nodes in advance.**

Network of Workstations (NOW) cont... Distributed Memory (programming issues)

- Data can be kept statically in nodes if most computations happen locally, and only changes on edges have to be reported to other nodes.
- An example of this is simulation where data is modeled using a grid (non-interactive workloads), and each node simulates a small part of the larger grid.

Network of Workstations (NOW) cont... Distributed Memory (programming issues)

- On every iteration, nodes inform all neighboring nodes of the new edge data.

Shared vs. Distributed Memory cont...

- There are two kinds of multiple-processor systems exist:
- **Multi-Processors** (Shared/ Distributed)
- **Multi-Computers** (Distributed)

Shared vs. Distributed Memory cont...

- In a **multi-processor** two or more CPUs **share a common main memory**.
- Any ***process*** on any ***processor***, can read or write any word in shared memory, simply by moving data to or from the desired location.
- In a **multi-computer**, in contrast, **each CPU has its own private memory**. Nothing is shared.

Shared vs. Distributed Memory cont...

- To make an **agriculture analogy**, a multiprocessor is a system with a **herd of sheep (processes)** eating from a **single feeding through (shared memory)**.



Shared vs. Distributed Memory cont...

- A **multicomputer**, on the other hand, is a design in which each sheep has its own **private feeding through (*distributed memory*)**.

Shared vs. Distributed Memory cont...

