Middleware Technologies

MC9251

Unit-I Introduction

Middleware

- Middleware is a general term for any program that serves to "glue together" or mediate between two separate programs.

- A common application of middleware is to allow programs written for access to a particular database to access other databases

Middleware

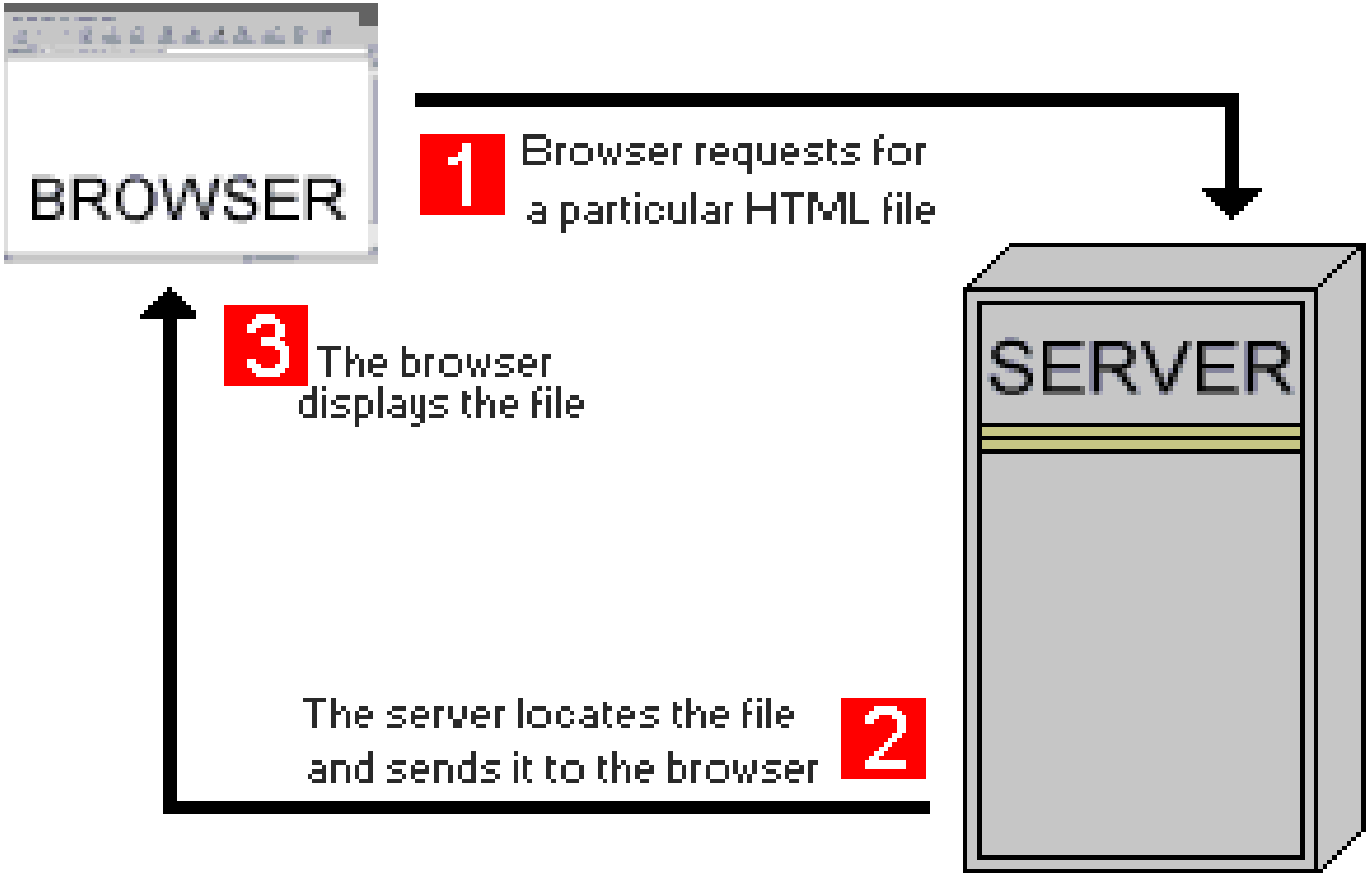
- Enterprise Application Integration-EAI
- To exploit the Internet, E-commerce, Extranet, and other new technologies
- Middleware categories
  - TP monitors
  - RPC systems
  - Object Request Brokers (ORBs)
  - Database access systems
  - Message Passing

Client-Server Architecture

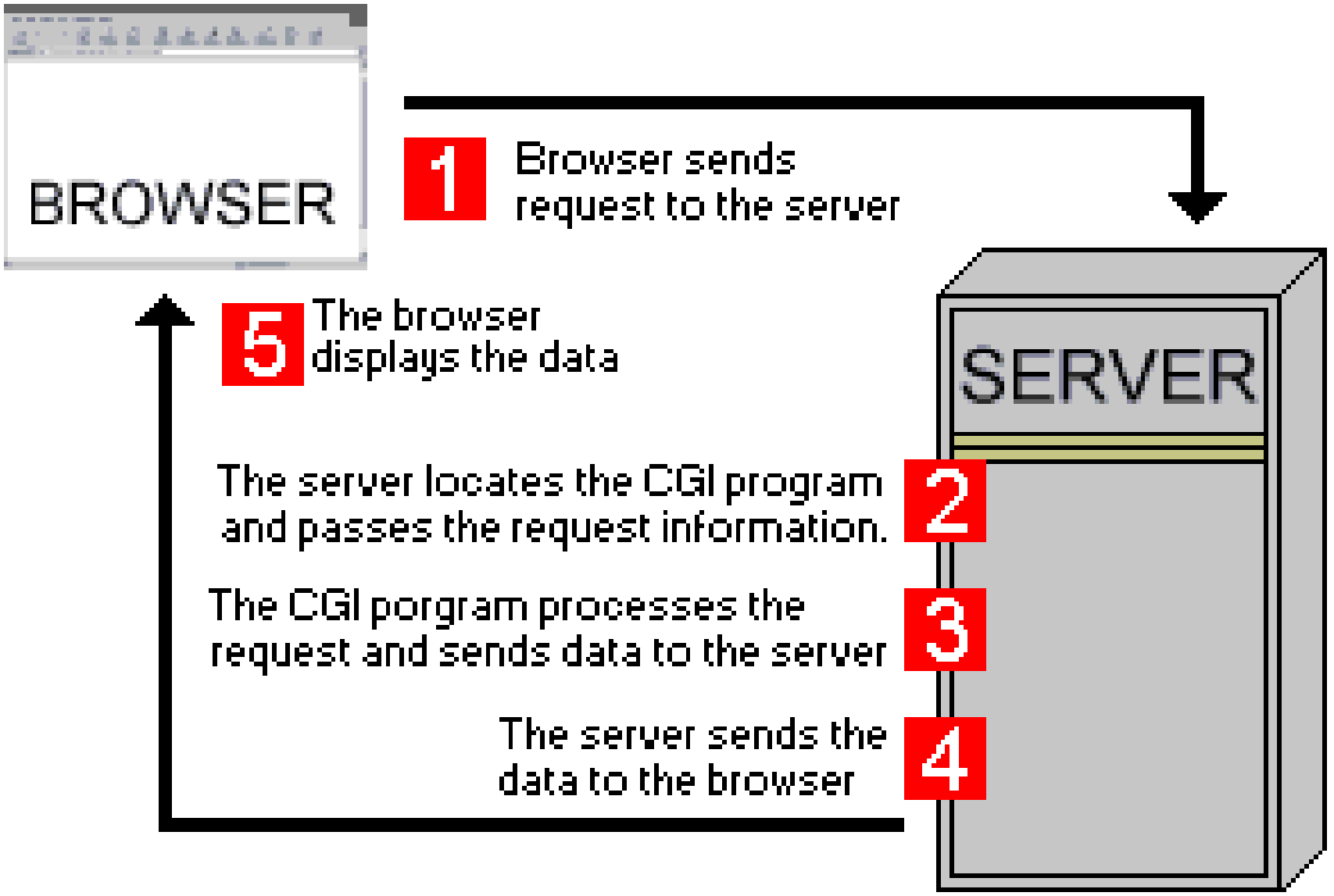
"Relationship between two computer programs, the client makes a service request from another program, the server".

- **Can be used in a single computer**
- **More important in Computer Networks**
- **Example: FTP, Internet, Internet banking**

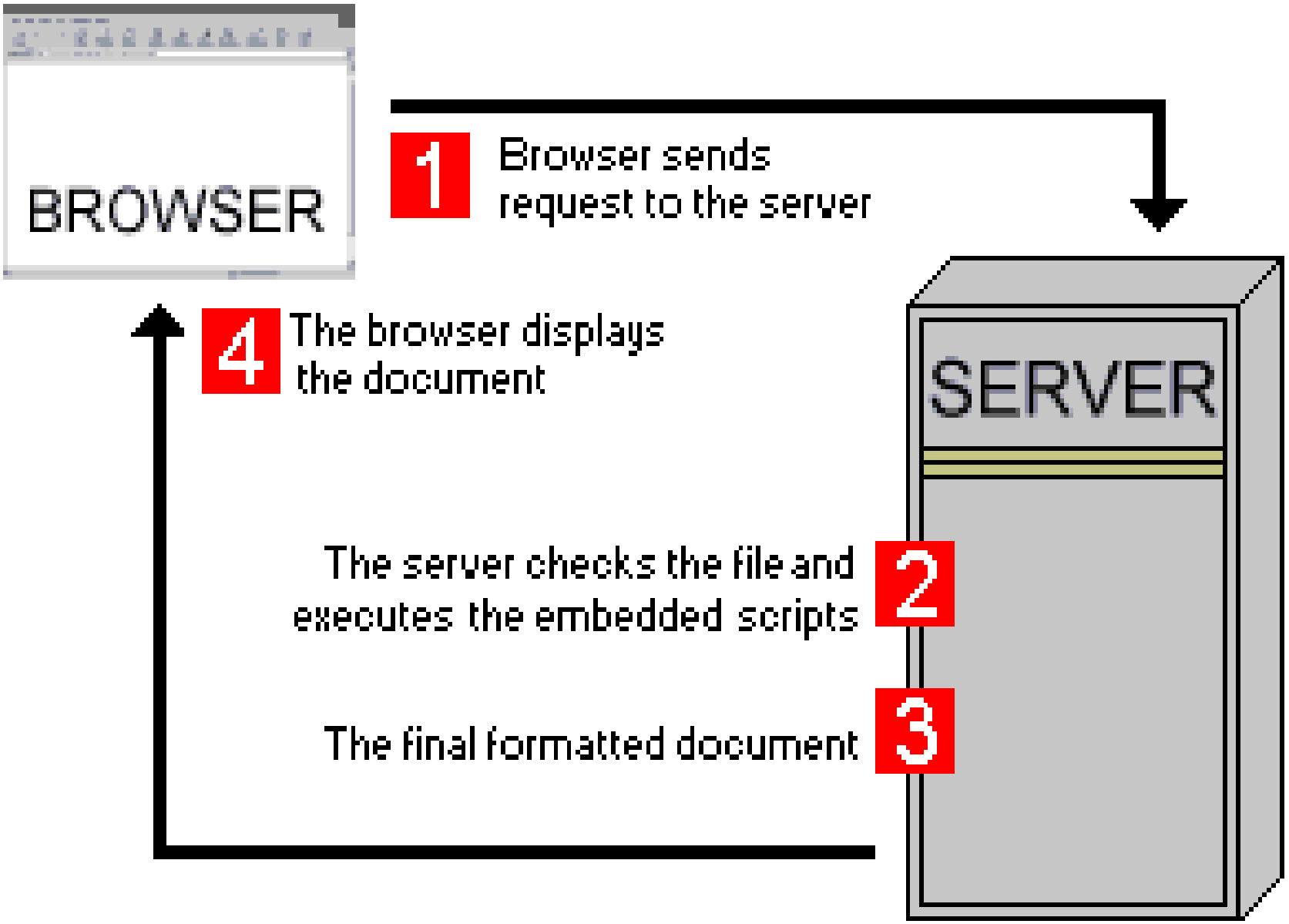
Client-server architecture - Static HTML pages



Client-server - CGI Scripts



Server side scripting technologies



Client-Server Architecture

- Two tier architectures
- Three tier architectures
- Three tier architecture with transaction processing monitor technology
- Three tier with an ORB architecture
- Distributed/collaborative enterprise architecture

Transaction Processing monitor technology

- The ability to update multiple different DBMSs in a single transaction
- Connectivity to a variety of data sources including flat files, non-relational DBMS, and the mainframe
- The ability to attach priorities to transactions
- robust security

Client-Server Architecture Characteristics

- Service
- Shared Resources
- A symmetrical protocols
- Transparency of location
- Mix and match
- Message based exchanges
- Encapsulation of services
- Scalability
- Integrity

Types of Servers

- File server
- Database server
- Transaction server
- Group server
- Object server
- Web server

File Server

- It is a computer responsible for the central storage and management of data files
- Allows users to share information
- File server – a normal PC – Dedicated network attached storage
- System security to limit access to files to specific users or groups
  - Novell's eDirectory, MS's Active directory

Database Server

- SQL requests and Data
- Server uses processing power to find the requested data
- DBMS provides server functionality
- Database Master servers and Slave servers
- Client application written by the user

Transaction Server

- Transaction- a group of SQL statements
- Client invokes Remote procedures – Servers execute transactions
- Both client and server component coded by the user
- Online transaction Processing (OLTP)

Groupware Server

- Addresses the management of semi-structured information
- Applications are created using a scripting language and form based interfaces

Object Server

- Client objects communicates with server objects using an ORB
- ORB locates an instance of object server class, invokes requested method
- Server objects provide support for concurrency and sharing
- Various ORB's
  - CORBA - Object management Group
  - Dcom - Microsoft
  - SOM - IBM
  - NEO - SUN

Web Servers

- HTTP requests - HTTP responses along with optional data contents
- Error response
- Supposed to serve requests quickly from more than one TCP/IP connection at a time.

Client/Server building blocks

Architecture Analogy

- We buy houses and not plans
- Thus, computer users buy business solutions and not client/server architectures
- How is the application split between the client and the server?
- What functions go into the client and server?
- Can the client/server model accommodate businesses of all sizes?
- Can a single client/server model accommodate all these type of users?

The Basic Building Blocks

- **Components:**
  - **Client**
  - **Server**
  - **Middleware and is catered for**

The Four situations

- **Client/server** for tiny **shops**
  - Client/server software and most of the business services runs on the same machine – one person shop
- **Client/server** for **small shops** and departments
  - Classic Ethernet **client**/single-**server** building block implementation
- **Client/server** for intergalactic enterprise
  - Multi-**server** building block implementation of **client/server**
- **Client/server** for the post-scarcity world
  - Every machine is transformed in the world for both **client** and a **server**. Personal agents on every machine and do the negotiations

Client Components

- **Operating system with:**
  - **a Graphical User Interface (GUI)**
  - **the ability to find and access distributed services**
- **Web browser to:**
  - **provide the user interface**
  - **download the necessary components from the server on demand**
- **Middleware components handle the non-local services.**
- **Clients may also run a component of a Distributed System Management (DSM) system**

Server Components

- **Server Operating System**
- **A server software package of some kind:**
  - **SQL Database server**
  - **Transaction Processing (TP) monitor**
  - **Groupware server**
  - **Object server**
  - **Web server**
- **Middleware components handle the reception of requests for services**
- **A server may also run a DSM component**

Middleware Components

- **These run on both the client and the server sides of a client/server application**
  - **Transport Stacks**
  - **Network Operating Systems (NOSs)**
  - **Service-specific middleware**
- **May also have a DSM component**

General middleware – provide substate for most client/server

- **Communication stacks**
- **Distributed directories**
- **Authentication services**
- **Network time services**
- **Remote Procedure Calls**
- **Queuing services**
- **NOS extensions:**
  - **Distributed file and print services**

Service-Specific Middleware

- **Database:**
  - **ODBC, JDBC, SQLJ, DRDA, OLE DB, etc.**
- **OLTP:**
  - **A variety of proprietary products**
- **Groupware:**
  - **MAPI, VIM, JavaMail, SMTP, POP3, IMAP, etc.**
- **Object:**
  - **CORBA, Microsoft's COM+**
- **Internet:**
  - **HTTP, CGI, XML, SET**

Server-to-Server Middleware

- Middleware software may also be used to coordinate inter-server interactions
- Servers are often clients to other servers, and vice-versa
- Some server-to-server interactions require special middleware:
  - Mail servers may do store-and-forward type messaging
  - Databases and groupware use daemons to automatically replicate data