Internet Applications

Web-Based Information System Implementation
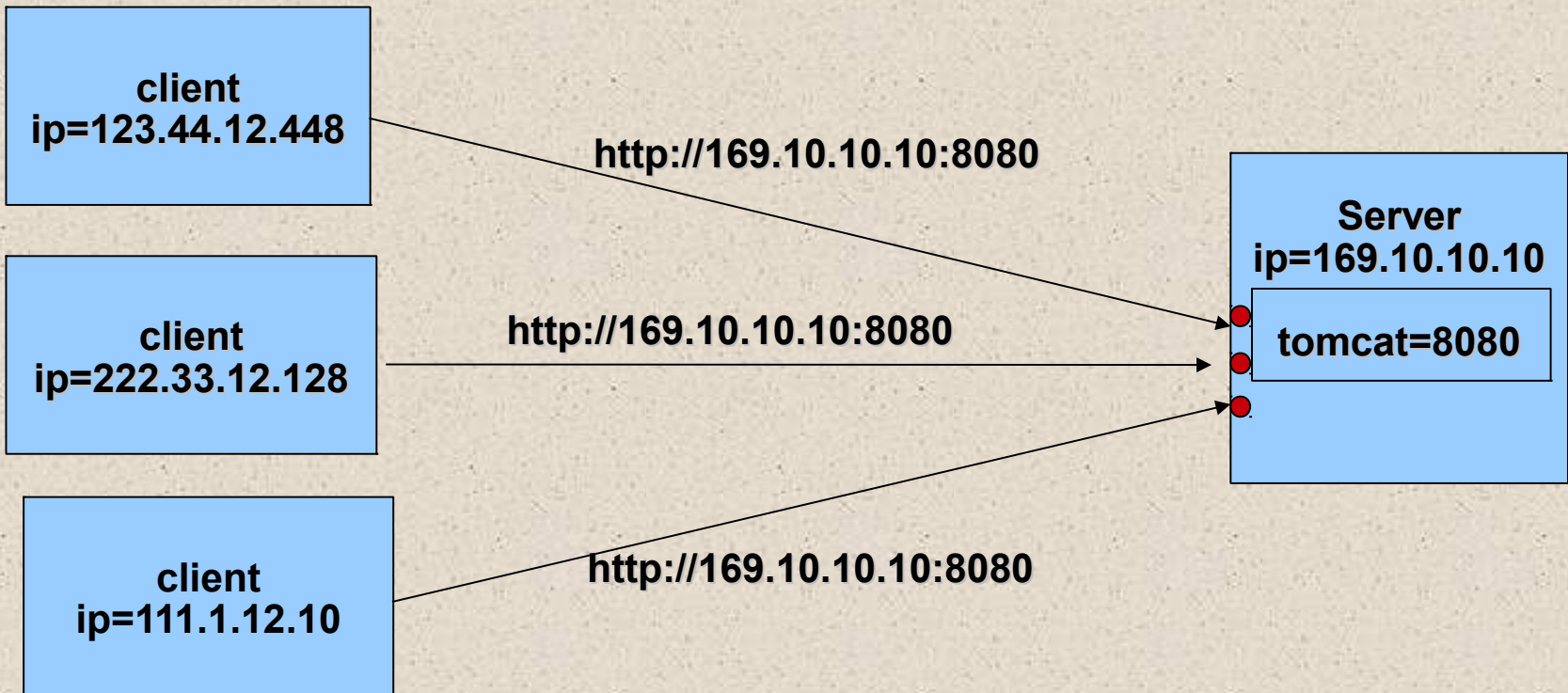
Identifying The Server Machine
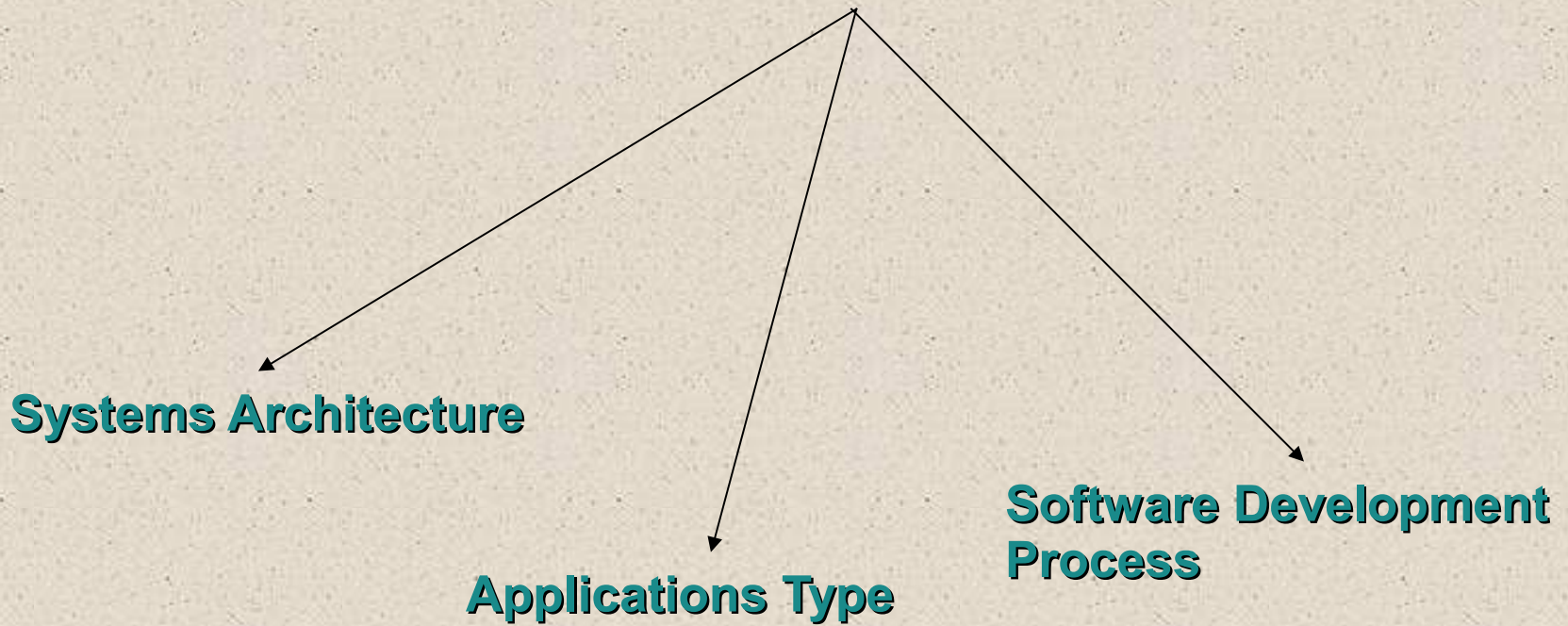
DNS
ip=10.10.10.10

http://169.10.10.10

http://CNN.com

PC
ip=222.33.12.128

http://169.10.10.10

Server
ip=169.10.10.10
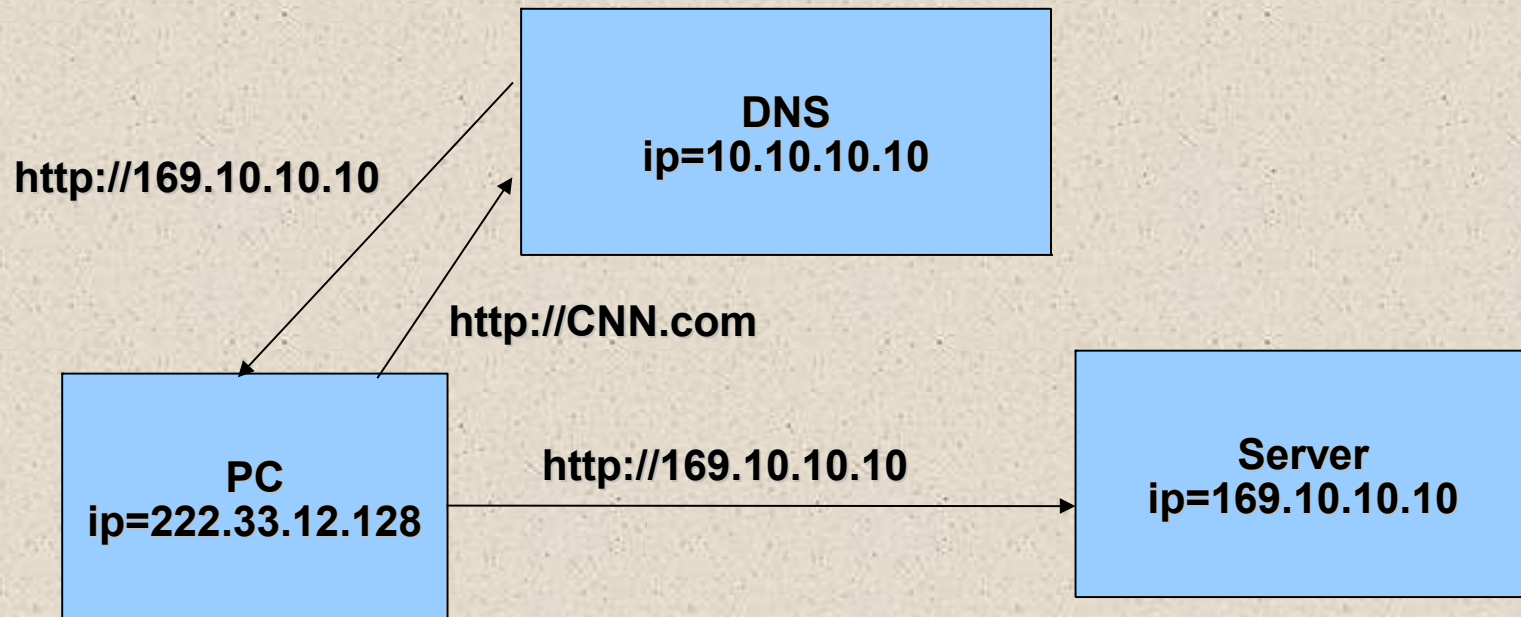
Identifying the Server Application

client
IP =
222.33.12.128

http://169.10.10.10:8080

Server

IP = 169.10.10.10

Mysql => 3306
tomcat => 8080
Apache => 80

Server Identifies Clients

client
ip=123.44.12.448

client
ip=222.33.12.128

client
ip=111.1.12.10

http://169.10.10.10:8080

http://169.10.10.10:8080

http://169.10.10.10:8080

Server
ip=169.10.10.10

tomcat=8080

The Three Dimensions of Web-Based Information Systems Implementation

Systems Architecture

Applications Type

Software Development Process

Components of the an information system architecture

Some of the most important components of an information System architecture are

- User Interface
- Data management
- Computation

User Interface

Sometimes it called *front end*.

It consist of a keyboard, a video display and a mouse.

There are three main user interface technologies are used today:

- Dumb terminal
- X – terminal
- PCs

Dumb terminal

The early dumb terminals keyboard and monochrom text – only display.

Keyboard imput/Text – based output.

Dumb terminal was user – unfrendly.

X- Terminals

Such as in Unix systems.

Keyboard and mouse input/Text or graphical output.

Traditional Architectures

There are two traditional Architectures:

- ❑ *Mainframe Architecture (1960)*
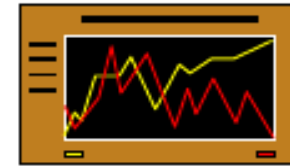- ❑ *File – Server Architecture (1980)*

Mainframe Era

Mainframe Architecture

Mainframe Server

Dumb Terminal
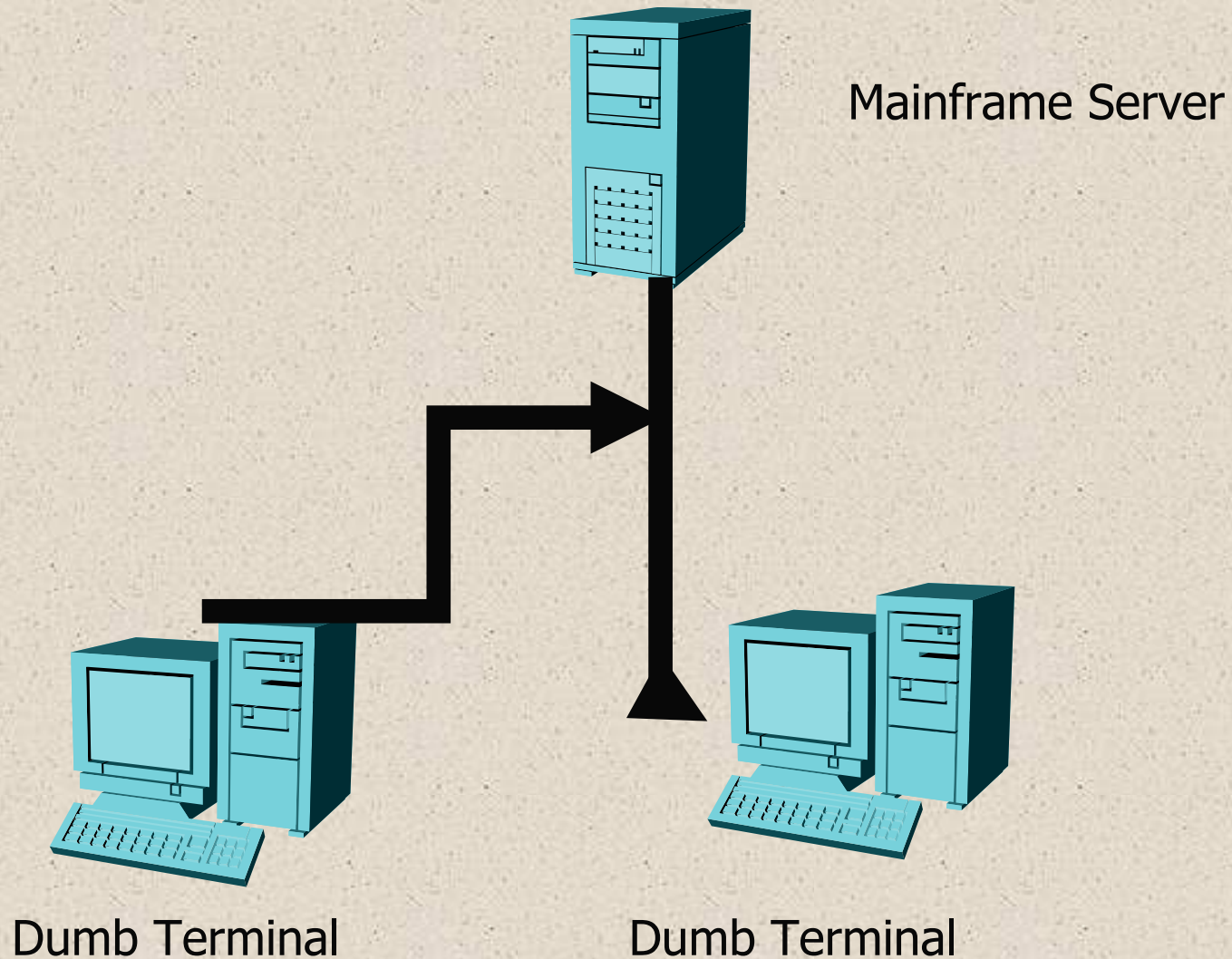
Dumb Terminal

Characteristics of Mainframe Architecture

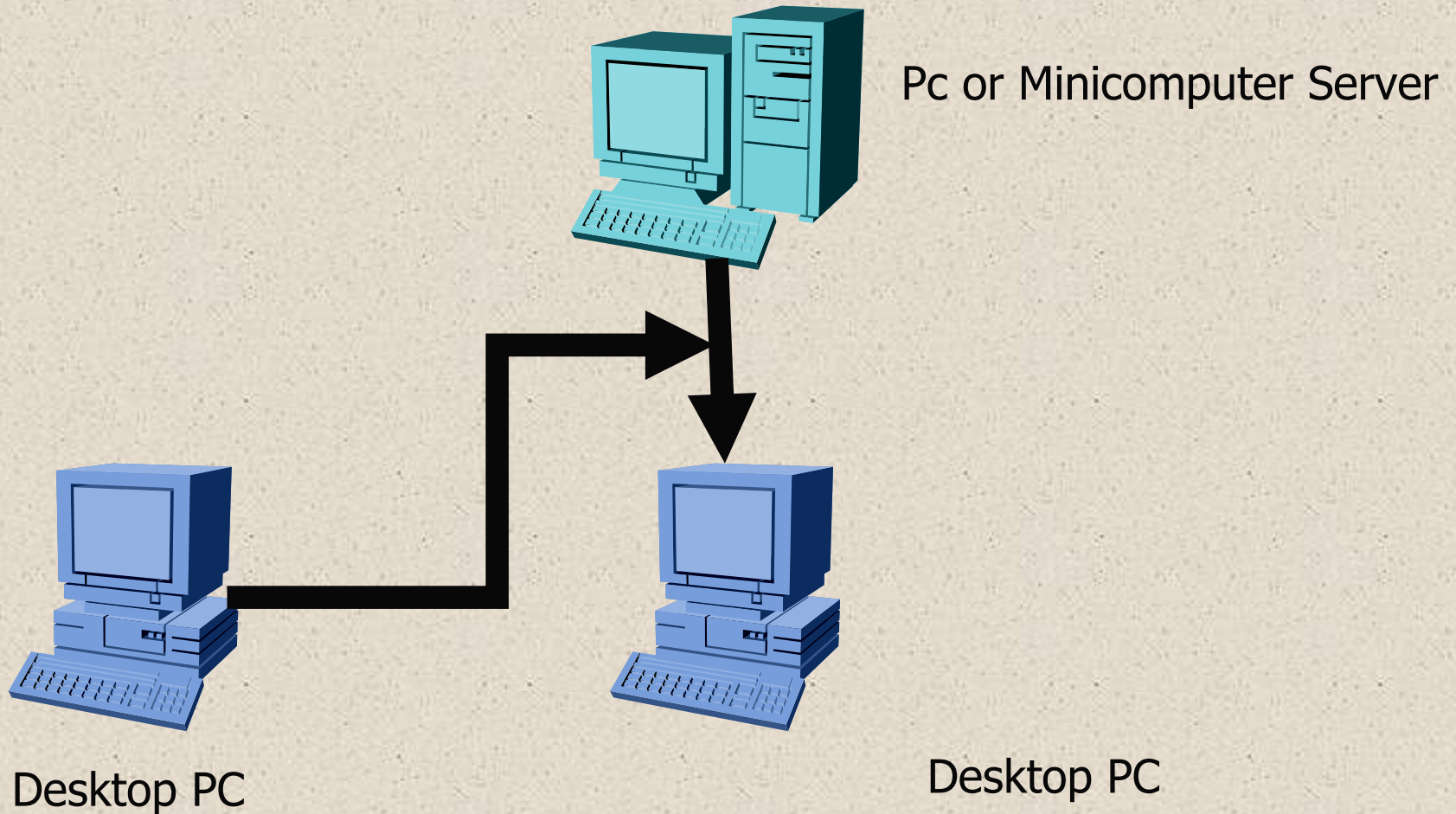| Component	Characteristic
Server hardware	Mainframe computer or minicomputer
Client hardware	Dumb terminals
User interface	Keyboard input, text output
Data management	Flat files
Computation management	COBOL programs (non-portable) executed on server
Cost	Medium or high
Reliability	High
Security	High
Scalability	High
Flexibility	Low

PC Era

Data Management
Business Logic
Presentation

Data Management
Business Logic
Presentation

Data Management
Business Logic
Presentation

Focus: User Control

File-Server Architecture

Pc or Minicomputer Server

Desktop PC

Desktop PC

Characteristics of File-Server Architecture

Component   Characteristic

- Server hardware   PC
- Client hardware   PCs
- User interface   Keyboard input, text output
- Data management   Flat files
- Computation management   Programs Written in various   languages (BASIC or   dBase) executed on client
- Cost Low
- Reliability   Low
- Security   Low
- Scalability   Low
- Flexibility   Low

Client/Server Architecture

Pc Minicomputer or Mainframe Server

Desktop PC

Desktop PC
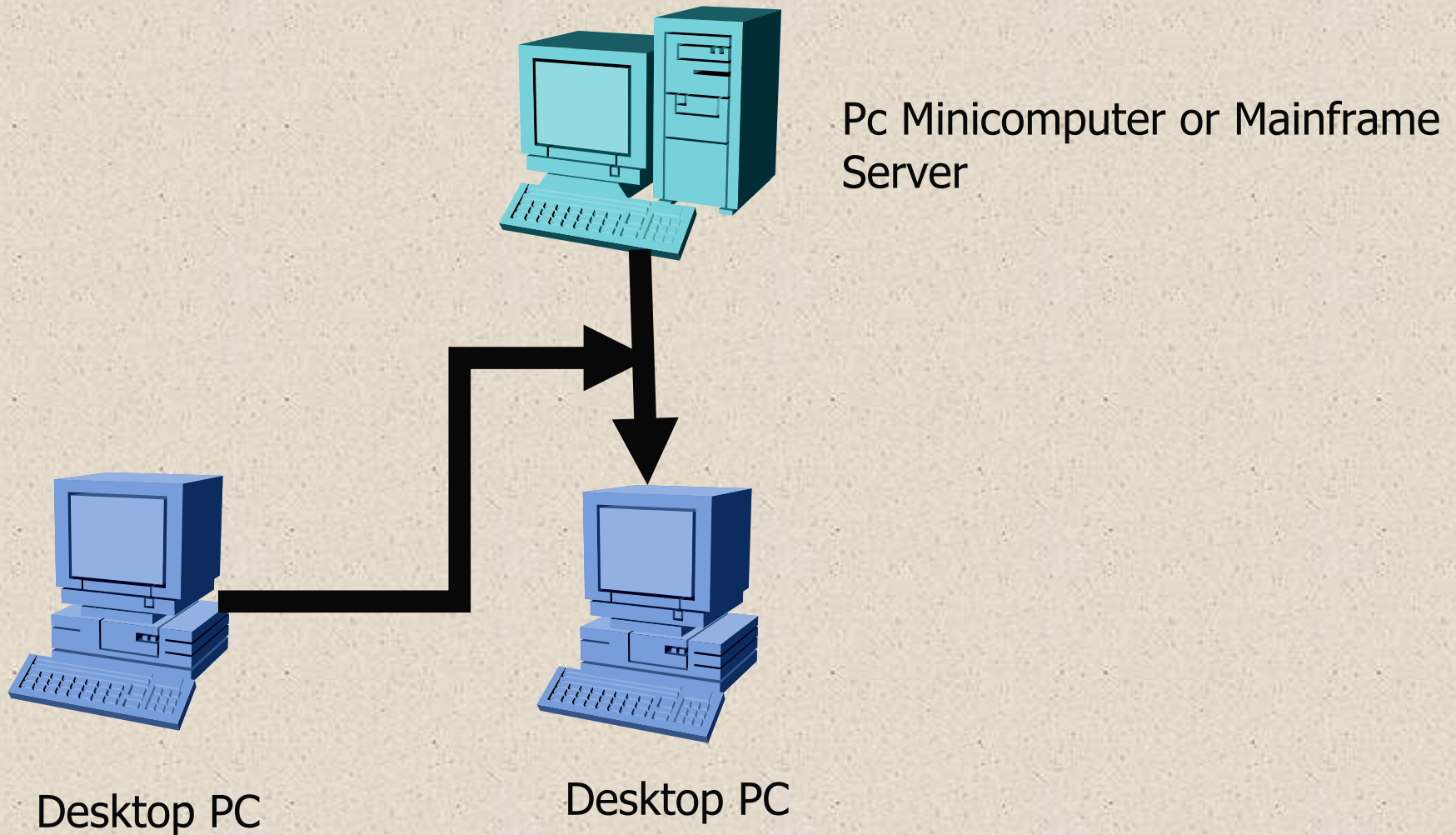
Characteristics of Client/Server Architecture

Component   Characteristic

- Server hardware   PC, minicomputer, or mainframe
- Client hardware   PC
- User interface   Graphical
- Data management   Relational database
- Computation management   Programs Written in various   languages executed on   server or client
- Cost Low to medium
- Reliability   High
- Security   High
- Scalability   High
- Flexibility   High

Web-Server-Based Architecture

Web Server

Web Browser
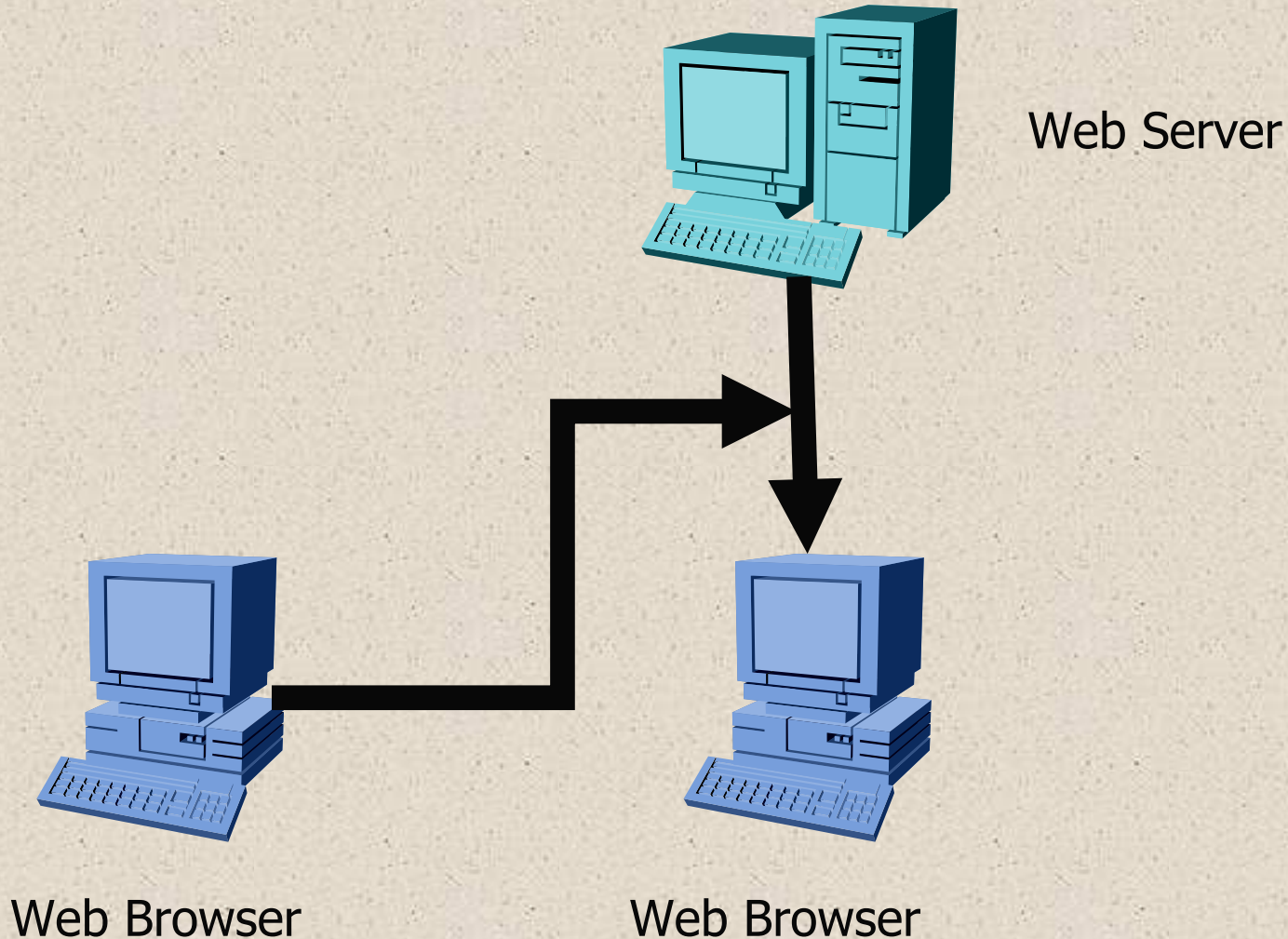
Web Browser

Characteristics of Web Server Architecture

This is a particular form of Client-Server Architecture.

- ❑ Web server – act as Server.
- ❑ Web browser – act as Client.

Three-Tier Architecture

Server

Server

Middleware Server

Desktop PC

Desktop PC
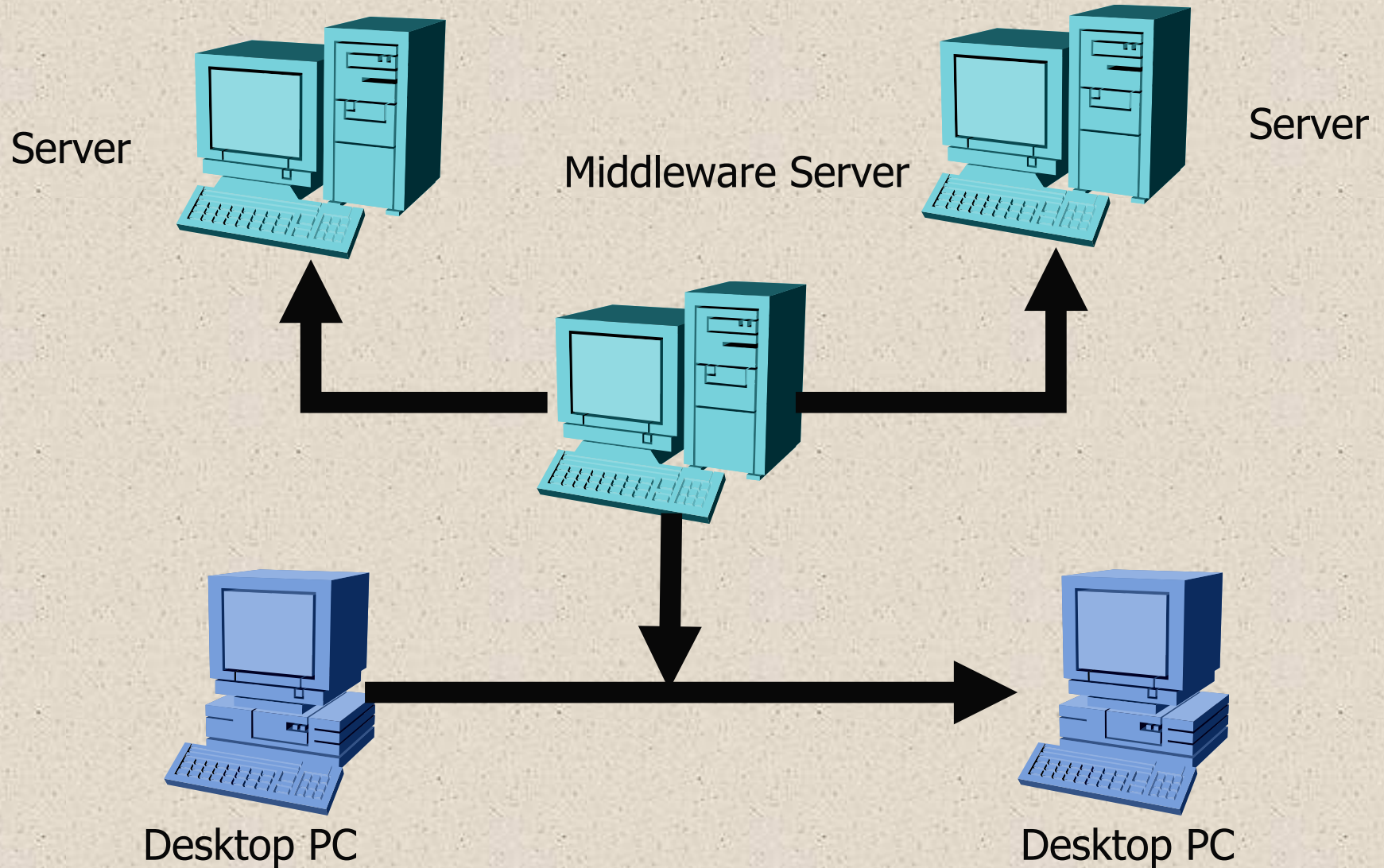
Characteristics of Three-Tier Architecture

Client can access to the Middleware server through a thin driver.

Network Architecture

Desktop PC

Desktop PC

Network

Desktop PC

Desktop PC
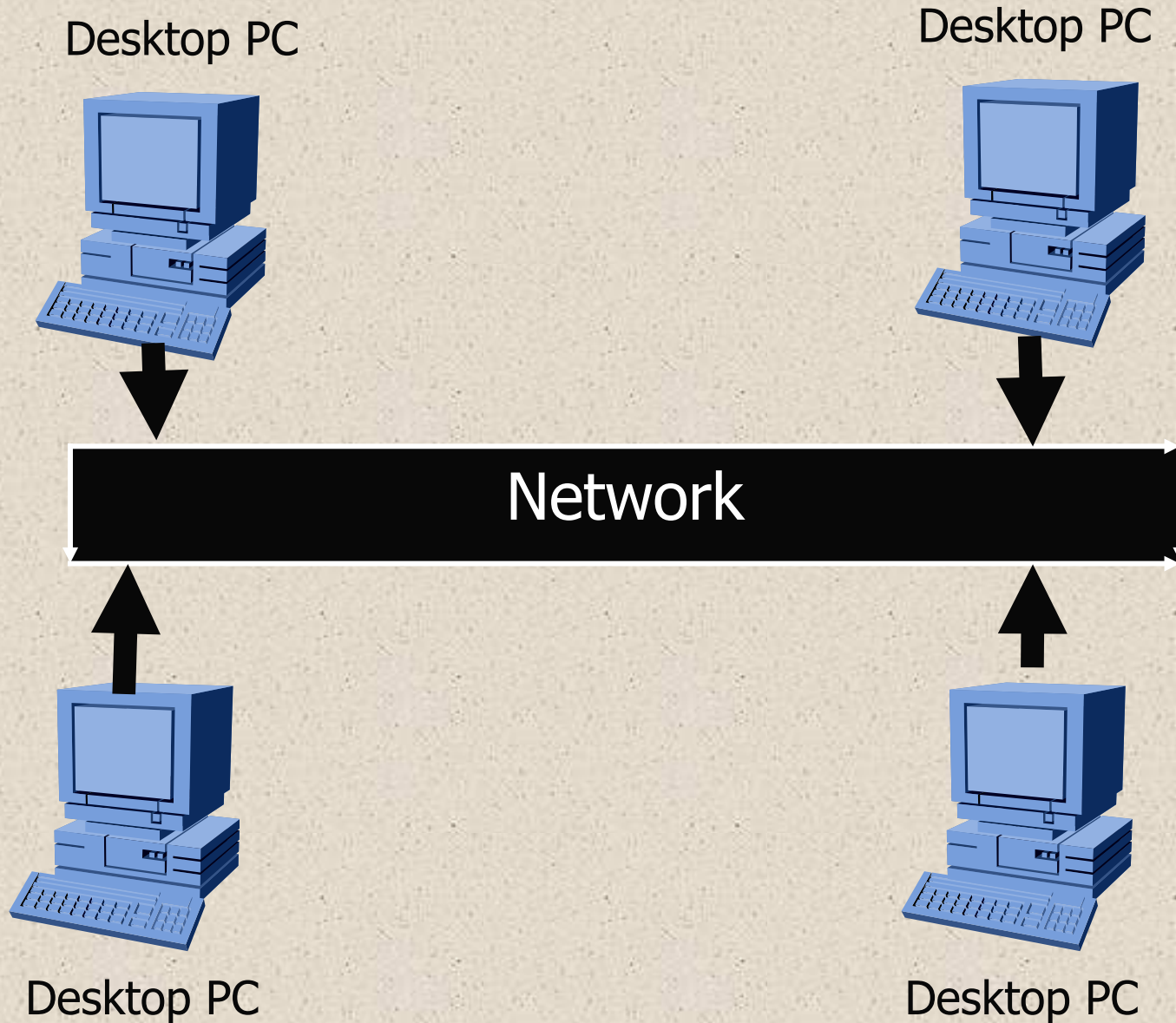
Characteristics of network Architecture

- Distributed architecture includes multiple servers.
- Every host potentially acts as both client and server. (Point to point architecture)

Characteristics of Distributed Architecture

Component   Characteristic

- Server hardware   PC, minicomputer, or mainframe
- Client hardware   PC
- User interface   Graphical
- Data management   Relational database
- Computation management   Programs Written in various languages executed on   server or client
- Cost Medium
- Reliability   High
- Security   High
- Scalability   High
- Flexibility   High

System Architecture

Distributed versus Centralized Systems

A **distributed system** is one in which the DATA, PROCESS, and INTERFACE components of an information system are distributed to multiple locations in a computer network. Accordingly, the processing workload is distributed across the network.

In **centralized systems**, a central, multi-user computer hosts all the DATA, PROCESS, and INTERFACE components of an information system. Users interact with the system via terminals (or terminal emulators).

Why Distributed Systems?

- Modern <u>business</u> systems are already decentralized and distributed.

- Distributed computing moves information and services closer to the customers and users who need them.

- Distributed computing consolidates the power of personal computers across the enterprise.

- Distributed computing solutions are more user-friendly because they utilize the PC as the end user interface.

- Personal computers and network servers are cheaper than centralized mainframe computers.

Distributed Computing Layers

- Presentation layer—the user interface

- Presentation layer logic—such as input editing

- Application logic layer—the business rules, policies, and procedures

- Data manipulation layer—to store and retrieve data to and from the database

- Data layer—the actual business data

Flavors of Distributed Computing

	FILE SERVER SOLUTION	CLIENT / SERVER SOLUTIONS			NETWORK COMPUTING SOLUTION
		DISTRIBUTED PRESENTATION (2 TIER)	DISTRIBUTED DATA (2 TIER)	DISTRIBUTED DATA & APPLICATION (N TIER)	
DATA LAYER	Stored on the File Server	Stored on the Database Server	Stored on the Database Server	Stored on the Database Server	Stored on the Database Server
DATA MANIPULATION LAYER	Executed on the Client	Executed on the Database Server	Executed on the Database Server	Executed on the Database Server	Executed on the Database Server
APPLICATION LOGIC LAYER	Executed on the Client	Executed on the Server	Executed on the Client	Executed on the Application Server	Executed on the Application Server
PRESENTATION LOGIC LAYER	Executed on the Client	Executed on the Client	Executed on the Client	Executed on the Client	Distributed from the Web Server
PRESENTATION LAYER	Displayed on the Client	Displayed on the Client	Displayed on the Client	Displayed on the Client	Displayed on the Client

Executing Programs on the Web

- **CGI programs are the most common way for Web servers to interact dynamically with users. Many HTML pages that contain forms, for example, use a CGI program to process the form's data once it's submitted. The use of CGI is a <u>server-side solution</u> because the processing occurs on the Web server.**

- **Another increasingly common way to provide dynamic feedback for Web users is to include scripts or programs that run on the user's machine rather than the Web server. These programs can be Java applets, Java scripts, or ActiveX controls. These technologies are known collectively as <u>client-side solutions</u>.**

CGI

- **Abbreviation of Common Gateway Interface, a specification for transferring information between a World Wide Web server and a CGI program. A CGI program is any program designed to accept and return data that conforms to the CGI specification. The program could be written in any programming language, including C, Perl, Java, or Visual Basic.**

- **One problem with CGI is that each time a CGI script is executed, a new process is started. For busy Web sites, this can slow down the server noticeably.**

Applets

- A Java applet is an applet delivered to users in the form of Java bytecode. Java applets can run in a Web browser using a Java Virtual Machine (JVM), or in Sun's AppletViewer, a stand-alone tool for testing applets. Java applets were introduced in the first version of the Java language in 1995, and are written in programming languages that compile to Java bytecode, usually in Java, but also in other languages such as Jython,[8] JRuby,[9] or Eiffel (via SmartEiffel)]

- Java applets run at speeds comparable to, but generally slower than, other compiled languages such as C++, but until approximately 2011 many times faster than JavaScript.[11] In addition they can use 3D hardware acceleration that is available from Java. This makes applets well suited for non trivial, computation intensive visualizations. When browsers have gained support for native hardware accelerated graphics in the form of Canvas and WebGL, as well as Just in Time compiled JavaScript, the speed difference has become less noticeable.

- Since Java's bytecode is cross-platform or platform independent, Java applets can be executed by browsers for many platforms, including Microsoft Windows, Unix, Mac OS and Linux. It is also trivial to run a Java applet as an application with very little extra code. This has the advantage of running a Java applet in offline mode without the need for any Internet browser software and also directly from the integrated development environment (IDE)

- The applet can be displayed on the web page by making use of the deprecated applet HTML element,[20] or the recommended object element. Embed element can be used with Mozilla family browsers
- This specifies the applet's source and location.
- Object and embed tags can also download and install Java virtual machine (if required) or at least lead to the plugin page.
- Applet and object tags also support loading of the serialized applets that start in some particular (rather than initial) state. Tags also specify the message that shows up in place of the applet if the browser cannot run it due any reason.

•Advantages

- * Applets are supported by most web browsers.

- * The same applet can work on "all" installed versions of Java at the same time, rather than just the latest plug-in version only.

- * Most web browsers cache applets, so will be quick to load when returning to a web page.

- * It can move the work from the server to the client, making a web solution more scalable with the number of users/clients.

Disadvantages

* It requires the Java plug-in.
* Some browsers, notably mobile browsers running Apple iOS or Android do not run Java applets at all.
* Some organizations only allow software installed by the administrators. As a result, some users can only view applets that are important enough to justify contacting the administrator to request installation of the Java plug-in.
 * As with any client-side scripting, security restrictions may make it difficult or even impossible for an untrusted applet to achieve the desired goals.

Disadvantages (cont.)

* Some applets require a specific JRE. This is discouraged.

* If an applet requires a newer JRE than available on the system, or a specific JRE, the user running it the first time will need to wait for the large JRE download to complete.

* Java automatic installation or update may fail if a proxy server is used to access the web. This makes applets with specific requirements impossible to run unless Java is manually updated. The Java automatic updater that is part of a Java installation also may be complex to configure if it must work through a proxy.

* Unlike the older applet tag, the object tag needs workarounds to write a cross-browser HTML document.

Javascript

- **JavaScript is a compact, object-based scripting language for developing client Internet applications. JavaScript statements can be embedded directly in an HTML page. These statements can recognize and respond to user events such as mouse clicks, form input, and page navigation.**
- **For example, you can write a JavaScript function to verify that users enter valid information into a form. Without any network transmission, an HTML page with embedded JavaScript can interpret the entered text and alert the user with a message dialog if the input is invalid. Or you can use JavaScript to perform an action (such as play an audio file, execute an applet, or communicate with a plug-in) in response to the user opening or exiting a page.**

- Why it is expensive to allow each mainframe user to have graphical user interfaces?

- Why file server architecture is not practical as an Information System for any Enterprise?

- Compare Client-server Architecture Distributed presentation(2-tier) with Distributed Data(2-tier). Show the difficulties of each approach as an implementation architecture for a large Enterprise information system.

- What did the Client-Server architecture (n-tier) improve over the (2-tier) systems with regard to Information systems implementation.

- What is the difference between (n-tier) systems and Web-architecture

- Why Client-Server architecture (n-tier) is not suitable for building Internet Information system, how did the Web architecture solved that.

- What is the main characteristics of implementing Information Systems. Compare it with the implementation of Operating Systems and Telecommunication Systems.

- Why did we need the CGI?
- Why did we need Javascript's?
- Why did we need applets?
- Why may we need Javascript and applet in the same time?
- Distinguish between Applet, Javascript's, Servlets, and CGI from (compilation/interpretation time) and their execution.