Introduction to Web Technologies IFT 203

Lecturer: Dr. O. A. AYILARA-ADEWALE

Topic: Web Architecture: Browsers, Servers, and Protocols

Lecture Objectives

i. By the end of this lecture, students should be able to:

ii. Explain the roles and operations of web browsers

iii. Describe the roles and operations of web servers

iv. Understand the client–server architecture

v. Explain how HTTP enables communication on the web

vi. Illustrate web communication using real-world examples

Web Architecture

Web architecture refers to the **structural design of the World Wide Web**, including:

Clients (browsers)

Servers

Communication protocols

It defines **how information is requested, transmitted, processed, and displayed**.

Example:
When you open www.google.com, your browser communicates with Google’s servers using HTTP.

Components of Web Architecture

Main components: **Web Browser (Client)**, **Web Server, Network & Protocols**

These components work together using a **client–server model**

Web Browsers

A **web browser** is a client-side software application used to: Request web resources, Interpret and display web content

Examples:

Google Chrome

Mozilla Firefox

Microsoft Edge

Safari

Roles of Web Browsers

Web browsers perform the following roles:

i. Send requests to web servers

ii. Interpret HTML, CSS, and JavaScript

iii. Render web pages for users

iv. Manage cookies, cache, and sessions

v. Provide security features (HTTPS, sandboxing)

Operations of Web Browsers (Step-by-Step)

i. User enters a URL

ii. Browser resolves the domain name via DNS

iii. Browser sends an HTTP request to the server

iv. Server sends an HTTP response

v. Browser renders the content on screen

Example:
Typing **www.uniosun.edu.ng** in a browser.

Web Servers

A **web server** is a system that: Stores web resources, processes client requests, sends responses back to clients.

Examples: Apache HTTP Server, Nginx and Microsoft IIS

Roles of Web Servers

Web servers:

i. Receive HTTP requests from clients

ii. Process requests (static or dynamic)

iii. Retrieve files or execute server-side code

iv. Send HTTP responses

v. Enforce security and access control

Operations of Web Servers

i. Listens for incoming requests

ii. Validates the request

iii. Locates requested resource

iv. Executes server-side logic (if required)

v. Sends response to client

Example:
A server processing a login request on a banking website.

Client-Server Architecture

The **client-server architecture** divides tasks between:

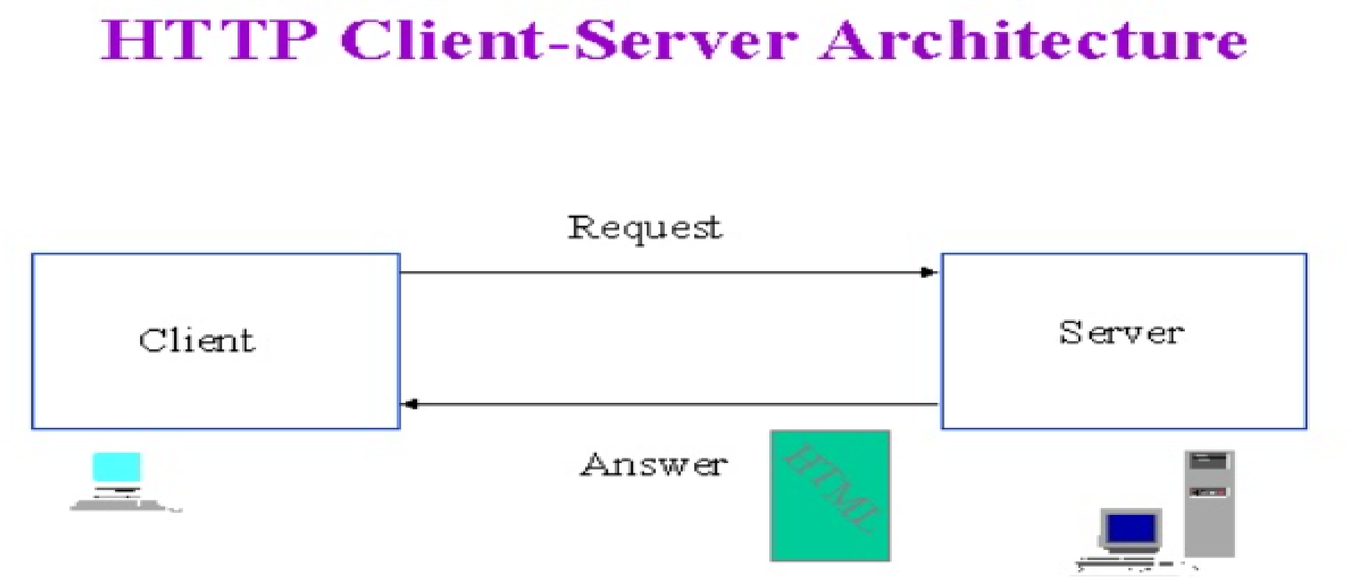
Clients: Request services

Servers: Provide services

Clients and servers communicate over a network.

Client–Server Architecture Diagram

- Client (Browser)
  ↓ HTTP Request
  Server (Web Server + Database)
  ↑ HTTP Response
- This separation improves scalability and security.



Advantages of Client–Server Architecture

i. Centralized data management

ii. Improved security

iii. Easier maintenance and updates

iv. Supports multiple users simultaneously

Example:
Online learning platforms like Moodle or Google Classroom.

Introduction to HTTP Protocol

HTTP (HyperText Transfer Protocol) is the foundation of data communication on the web. It defines:

- How requests are sent
- How responses are formatted

Characteristics of HTTP

i. Stateless protocol

ii. Request–response based

iii. Text-based

iv. Runs over TCP/IP

Modern variant: HTTPS (secure HTTP)

HTTP Request Structure

An HTTP request contains:

- Request method (GET, POST, PUT, DELETE)
- URL
- Headers
- Optional body

Example:

```
GET /index.html HTTP/1.1
Host: www.example.com
```

HTTP Response Structure

An HTTP response contains:

Status code (200, 404, 500)

Headers

Body (HTML, JSON, images)

Example:

```
HTTP/1.1 200 OK
Content-Type: text/html
```

Common HTTP Methods

GET – Retrieve data

POST – Send data

PUT – Update data

DELETE – Remove data

Example: Submitting a login form uses POST.

Example: Full Web Communication Flow

Scenario: Student logs into a university portal

Browser sends login request (POST)

Server validates credentials

Server queries database

Server responds with dashboard page

Browser displays dashboard

Conclusion

i. Browsers act as clients that request and display content

ii. Servers store, process, and respond to requests

iii. Client–server architecture structures web communication

iv. HTTP enables standardized web data exchange

THANK YOU !!!