CSE 3002
INTERNET AND WEB PROGRAMMING

Module:1

Web Browsers and Web Servers / Hosting - Security and Vulnerability Client side vs Serverside scripting

Topics

- Web Browsers
  - Introduction and stats
  - High Level Structure
- Web Servers
  - Introduction and type
  - Models
  - Hosting types
- Security and Vulnerability
- Client-side and server-side scripting


WEB BROWSERS

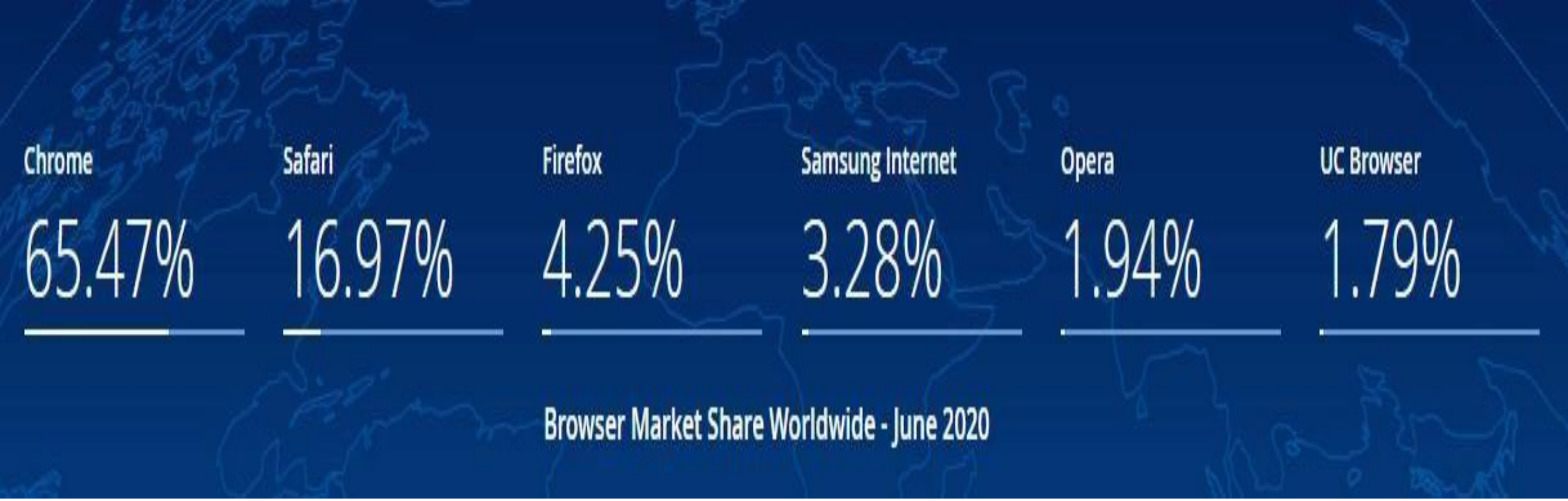
Web Browsers

- A software application that retrieves and displays information from a server including web pages, text, images, videos, and other contents.

Why do different browsers respond differently to websites, and why is there more than one to begin with?

How do browsers work and where did the need for *cross-browser testing* come from?

- By understanding the history and backend of some major browsers

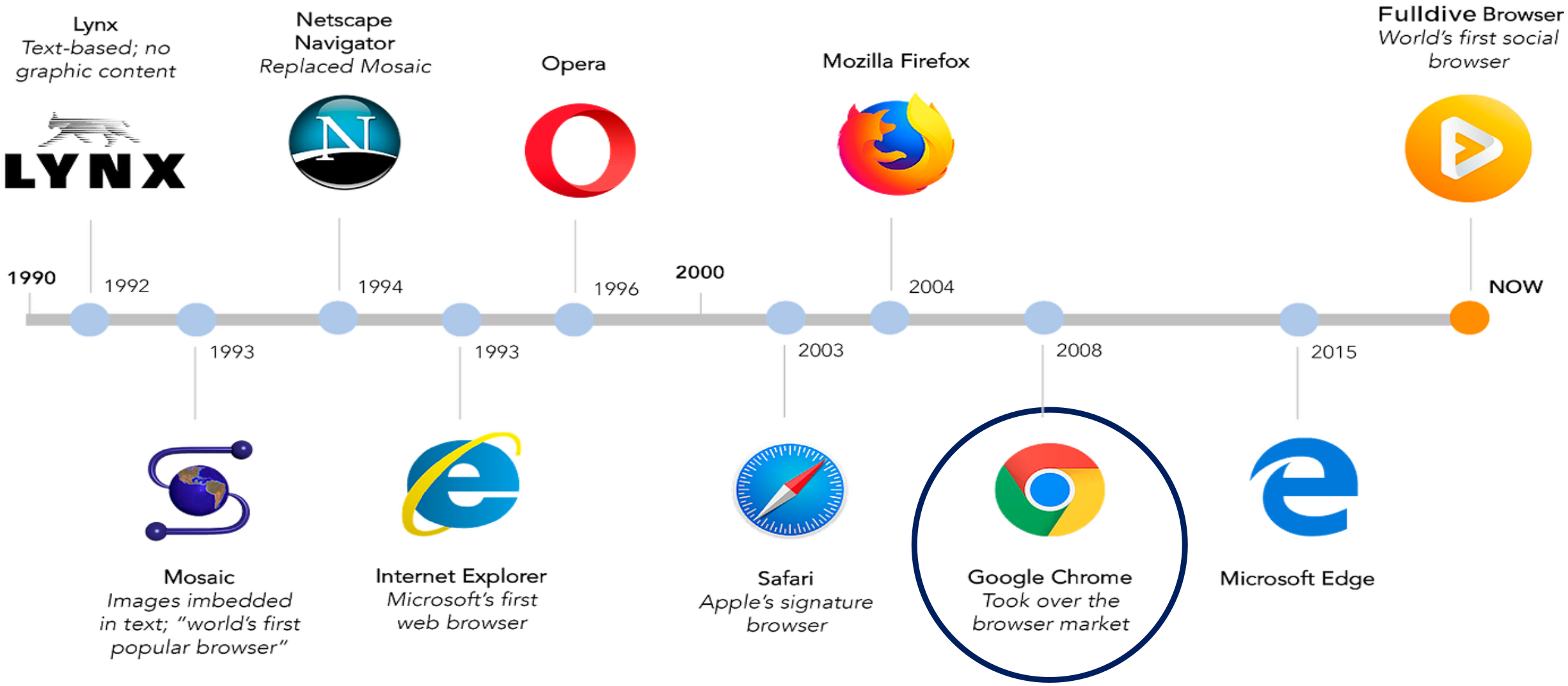


The Fastest Browser? Surfing Speedily in 2020

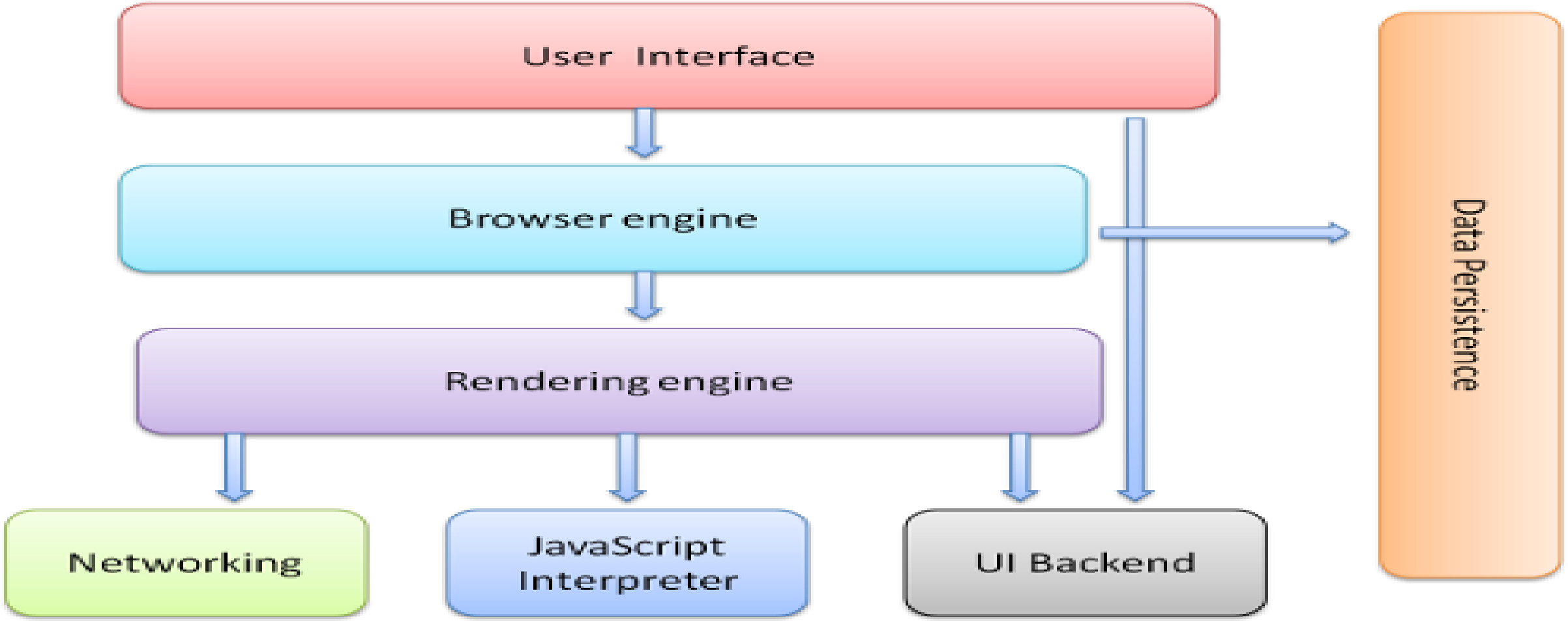
1. Vivaldi
2. Opera
3. Brave
4. Mozilla Firefox
5. Google Chrome & Chromium

Source : Cloudwards.net/browser

Timeline of Web Browsers



Browser High Level Structure



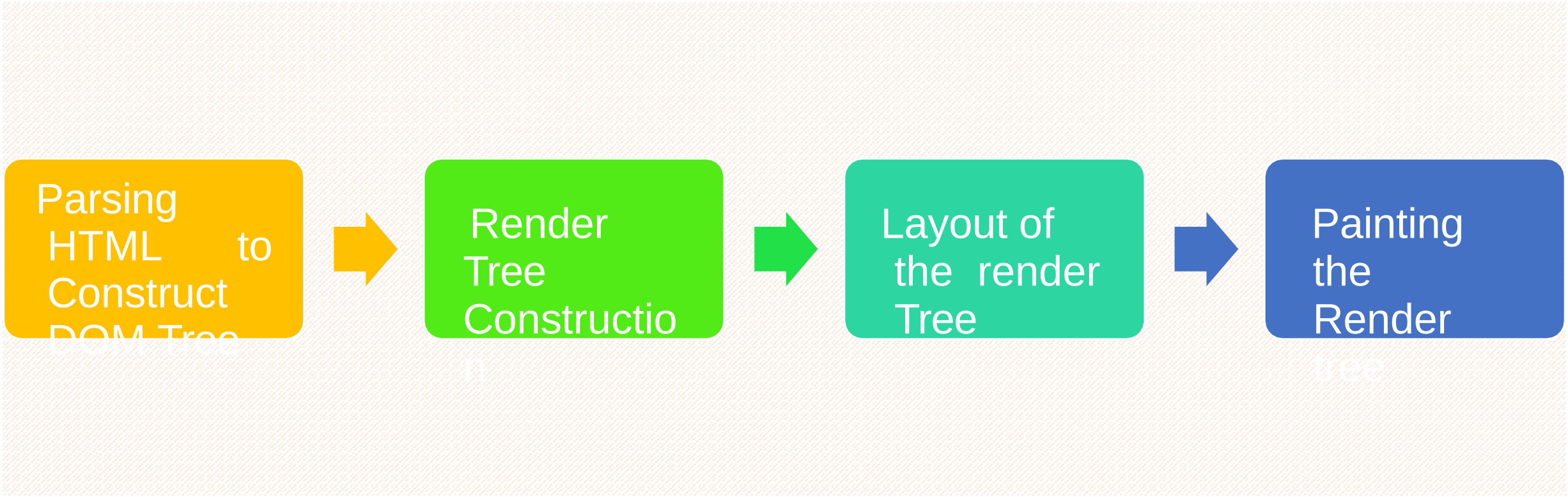
Browser High Level Structure

- **The user interface**: Includes the address bar, back/forward button, bookmarking menu,
  etc.
- **The browser engine**: Marshals actions between the UI and the rendering engine.
- **The rendering engine** : Responsible for displaying requested content. For example if the
  requested content is HTML, the rendering engine parses HTML and CSS, and displays the parsed content on the screen.
- **Networking**: for network calls such as HTTP requests, using different implementations for different platform behind a platform-independent interface.
- **UI backend**: used for drawing basic widgets like combo boxes and windows. This backend
  exposes a generic interface that is not platform specific. Underneath it uses operating
  system user interface methods.
- **JavaScript interpreter**. Used to parse and execute JavaScript code.
- **Data storage**. This is a persistence layer. The browser may need to save all sorts of data

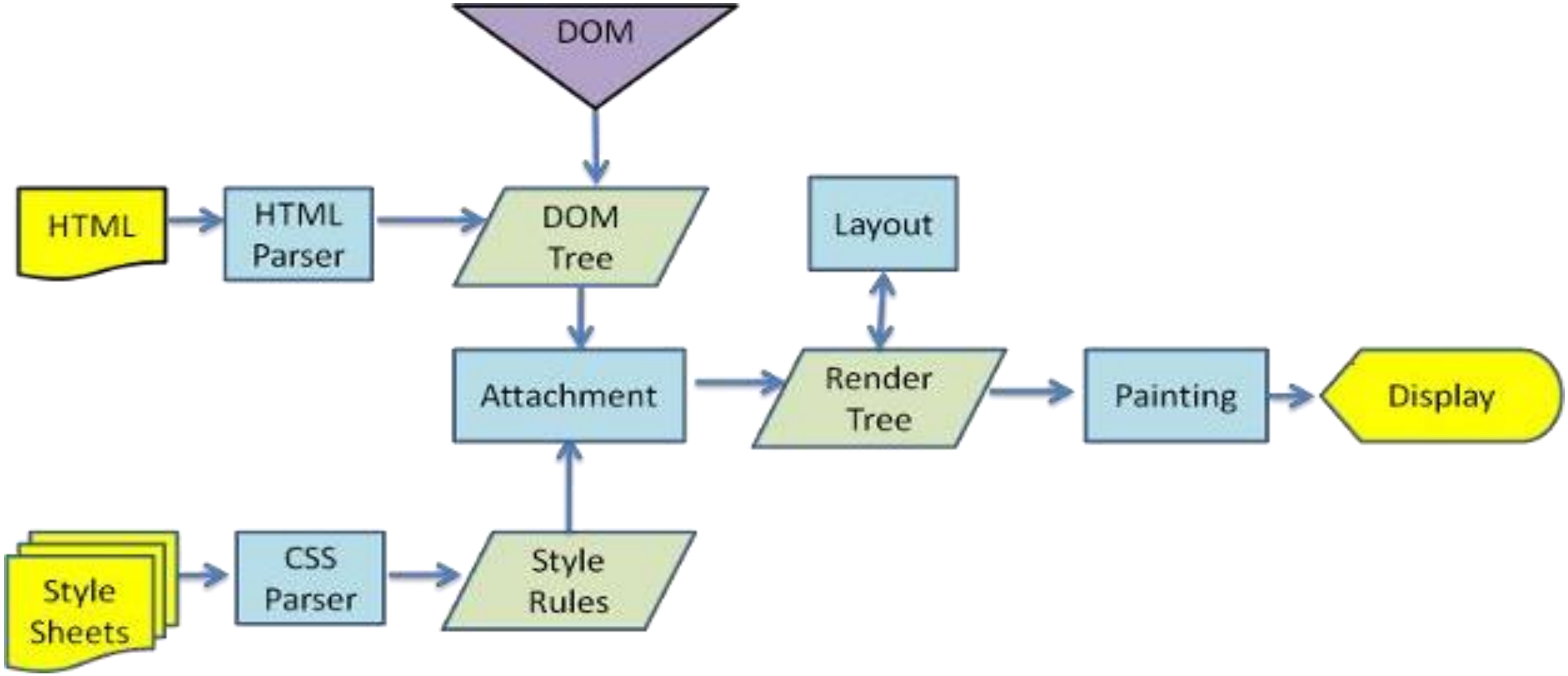
Rendering engines

- Different browsers use different rendering engines:
  - Internet Explorer - **Trident**
  - Firefox - **Gecko**
  - Safari -**WebKit**
  - Chrome and Opera – Blink **(a fork of WebKit)**
- WebKit is an open source rendering engine which started as an engine for the Linux platform and was modified by Apple to support Mac and Windows.

Rendering Engine Main Flow



Parsing main flow

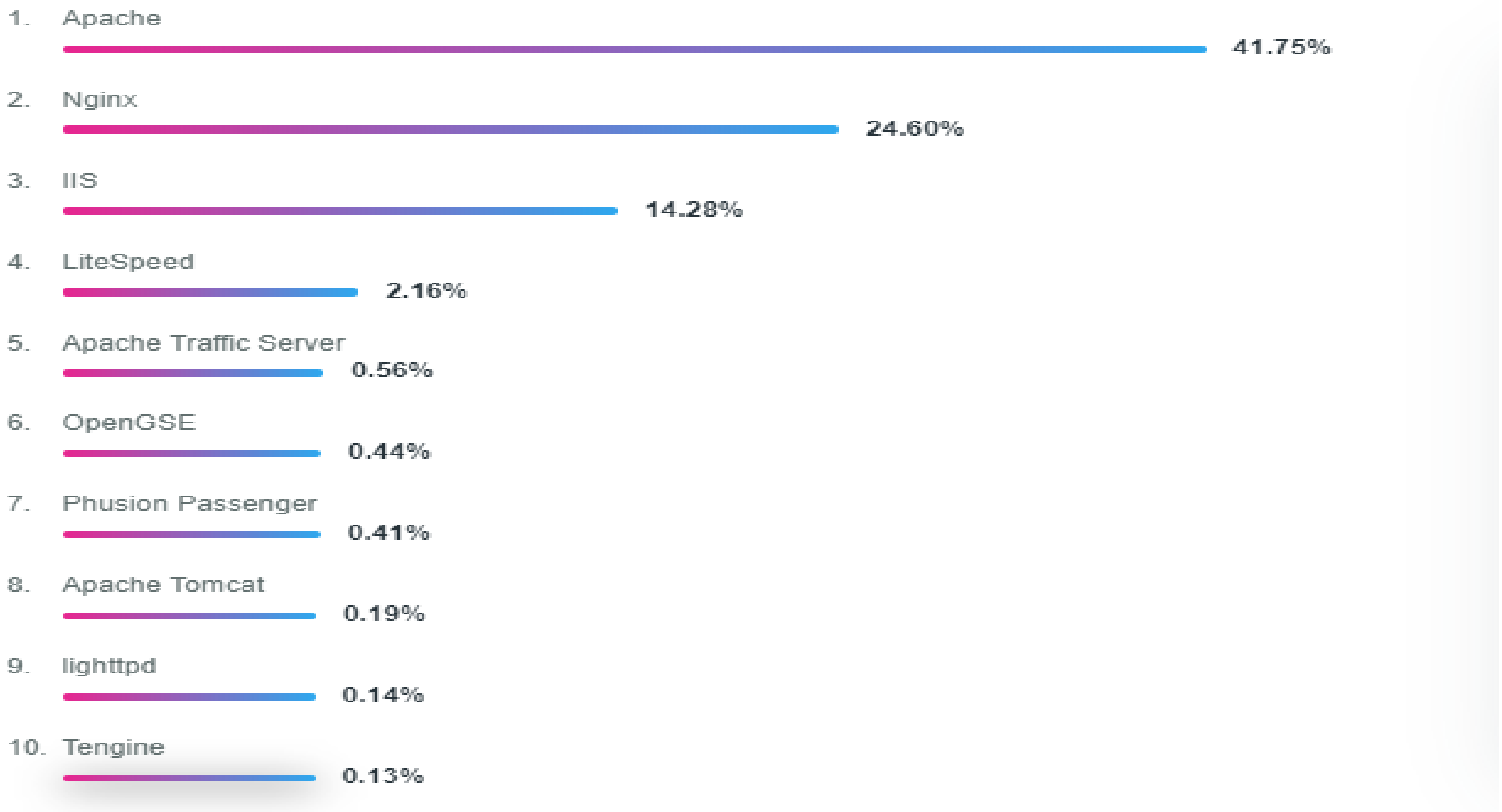


So, what lessons can we take away from the web browsers?

As software professionals, the only way to approach this diverse user behavior is to develop web applications to be cross-compatible and test them across different browsers to make sure they're visually and functionally acceptable

WEB SERVER

Global Web Server Market Share July 2020



Web Server Definition (1)

- A Web server is a program that generates and transmits responses to client requests for Web resources.

- Handling a client request consists of several key steps:
  - *Parsing the request message*
  - *Checking that the request is authorized*
  - *Associating the URL in the request with a file name*
  - *Constructing the response message*
  - *Transmitting the response message to the requesting client*

Web Server Definition (1)

- The server can generate the response message in a variety of ways:

1. The server simply retrieves the file associated with the URL and returns the contents to the client.
2. The server may invoke a script that communicates with other servers or a back-end database to construct the response message.

Client/Server Architecture

Two-Tier

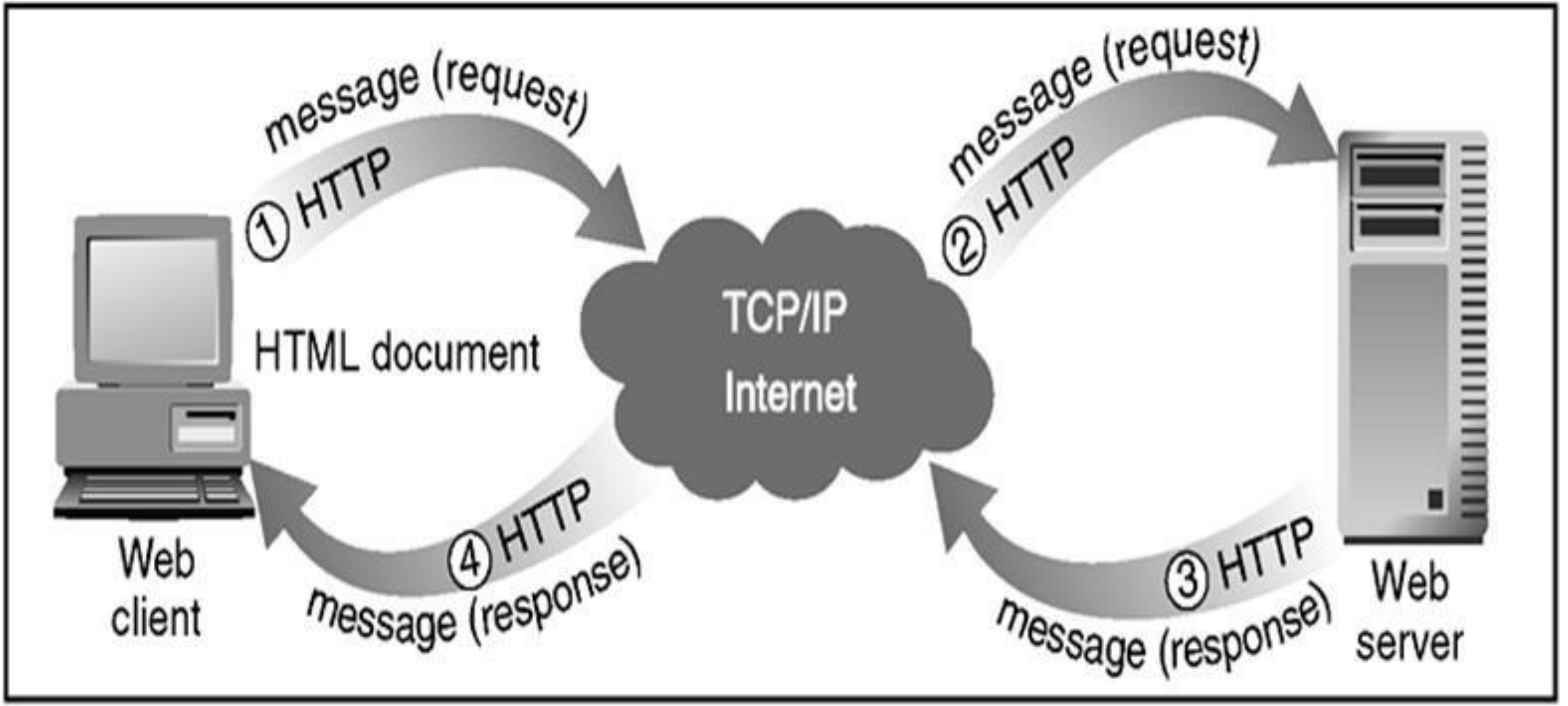
- Has only one client and one server
- Request message
  - Message that a Web client sends to request a file or files from a Web server
- Typical request message
  - Request line
  - Optional request headers
  - Optional entity body

Client/Server Architecture

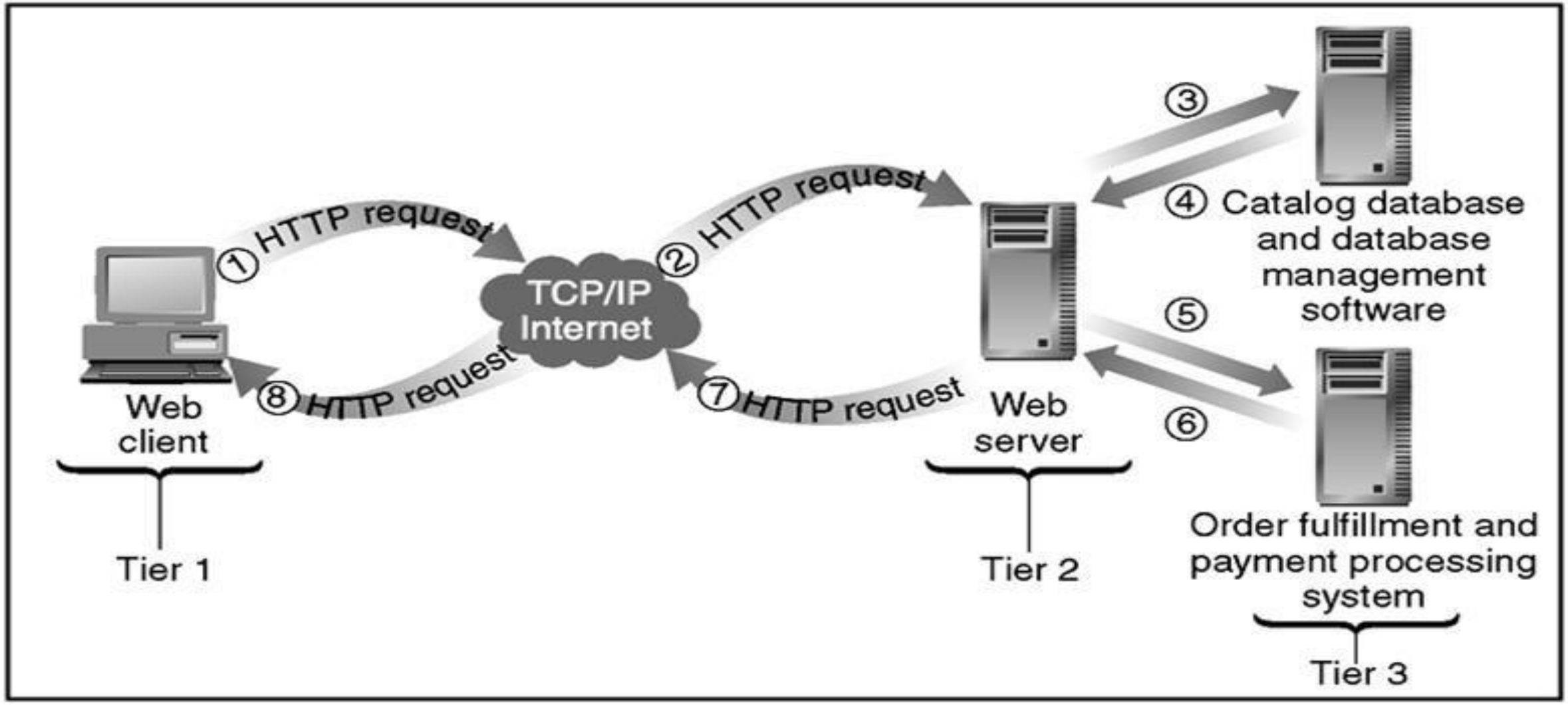
Multi Tier

- Three-tier architecture
  - Extends two-tier architecture to allow additional processing
- N-tier architectures
  - Higher-order architectures
  - Third tier includes software applications that supply information to Web server

Message Flows in a Two-tier



Message Flows in a Three-tier



Web Server Access Control

- A Web server may limit which users can access certain resources. Access control requires a combination of ***authentication*** and ***authorization***.

  - Authentication identifies the user who originated the request.

  - Authorization determines which users have access to a particular resource.

Dynamically Generated Responses

- This feature differentiates the Web from earlier file transfer services on the Internet.
- Dynamically generated responses are created in a variety of ways:
  - Server-side include(SSI)
  - Server script

Dynamically Generated Responses

Server-side include(SSI)

- A ***server-side include*** instructs the Web server to customize a static resource based on directives in an HTML-like file.

Server script

- A ***server script*** is a separate program that generates the request resource.
- The program may run as
  - Part of the server
  - A separate process

Server Architecture

- Some techniques for allocating system resources among competing client requests are :
  - Event-driven server architecture
  - Process-driven server architecture
  - Hybrid server architecture

Event-Driven Server Architecture

- An ***event-driven*** server
  - Has a single process that alternates between servicing different requests
  - Allows the server to serialize operations that modify the same data
  - Performs non-blocking system calls
  - Not used in Most high-end Web servers

Process-Driven Server Architecture

- A ***process-driven*** server
  - Allocates each request to a separate process
    - One master process listens for new connection
    - The master process creates, or forks, a separate process for each new connection
  - Terminates the process after parsing the client request and transmitting the response
    - To prevent memory leak
  - Introduces overhead for switching from one process to another

Hybrid Server Architecture

- In ***Hybrid*** server architectures
  - The strengths of the event-driven and process-driven models are combined
  - Each process would become an event-driven server that alternates between a small collection of requests
  - A single process has multiple independent threads
  - Main process instructs a separate helper process to perform time-consuming operations

Web Server Hardware Architectures : More

- Server farms
  - Large collections of servers
- Centralized architecture
  - Uses a few very large and fast computers
- Distributed/decentralized architecture
  - Uses large number of less powerful computers
  - Divides the workload among them

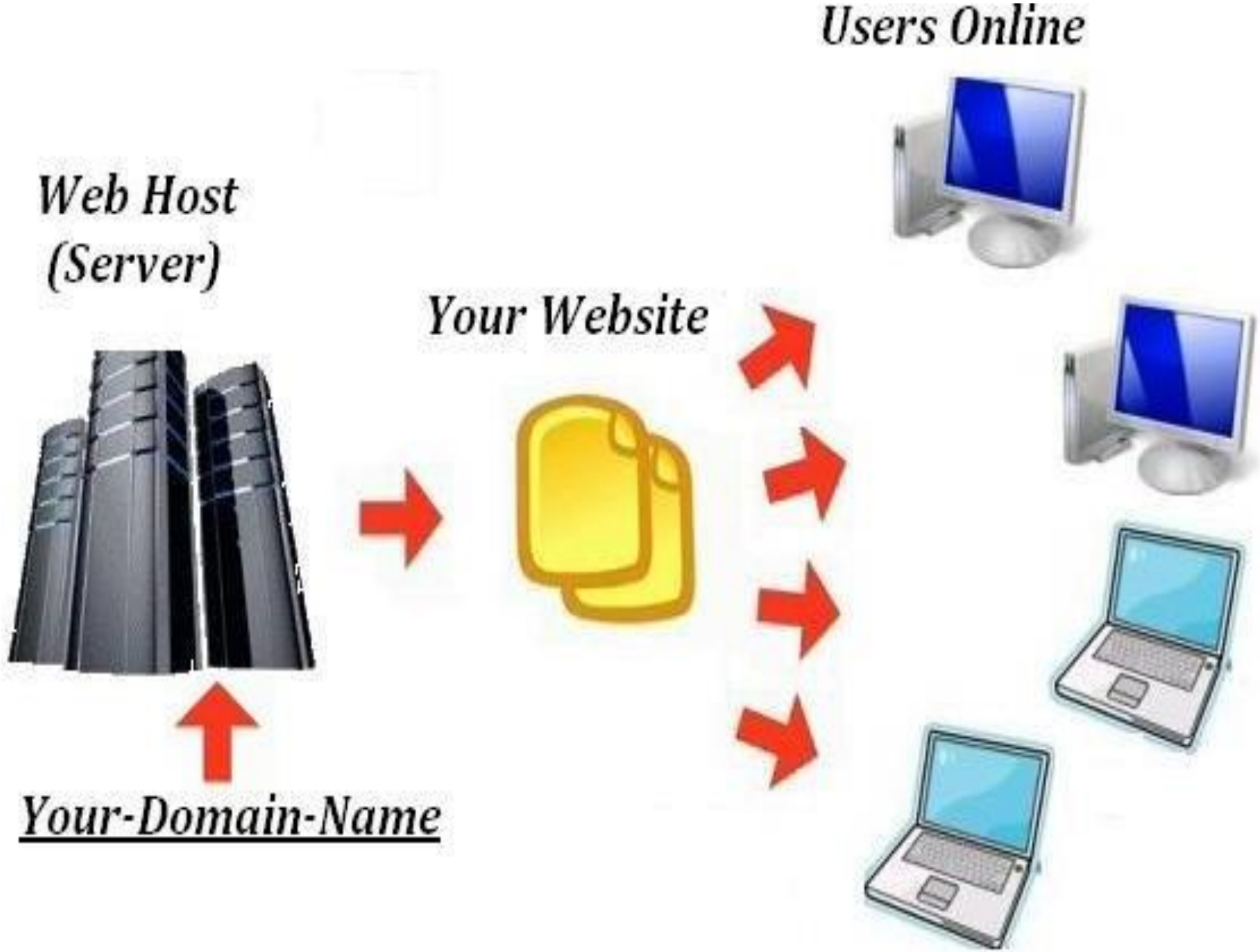
Web Server Hardware

- Web server computers
  - More memory, larger hard disk drives, and faster processors
- Blade servers
  - Placing small server computers on a single computer board, then installing boards into a rack-mounted frame
- Virtual server (virtual host)
  - Maintains more than one server on one machine

Web Server Performance Evaluation

- Benchmarking
  - Testing used to compare the performance of hardware and software
- Throughput
  - Number of HTTP requests that hardware and software combination can process in a unit of time
- Response time
  - Time required by server to process one request

WEB HOSTING


Users Online
Web Host
(Server)
Your Website
Your-Domain-Name

What is Web Hosting?

- A web hosting service is a type of Internet hosting service that allows individuals and organizations to make their website accessible via the World Wide Web.
- To make your Web site visible to the world, it has to be hosted on a Web server.

Things to Consider with an ISP for Web Hosting

- 24-hour support
- Daily Backup
- Traffic Volume
- E-mail Capabilities
- Database Access

Web Hosting Domain Names

- Web Hosting Domain Names
  - ✓ A domain name is a unique name for your web site. Choosing a hosting solution should include domain name registration. Your domain name should be easy to remember and easy to type.

✓ What is a Domain Name? (Next Class)

✓ Registering a Domain

✓ Choosing a Domain Name

✓ Sub Domains

Types of hosting services

- Self-hosting
- Shared hosting
- Dedicated hosting
- Collocated hosting
- Mailing Lists

WEB SECURITY & VULNERABILITIES

Top Vulnerabilities in 2020 :

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access control
- Security misconfigurations
- Cross Site Scripting (XSS)
- Insecure Deserialization

SQL Injection

- Attacker sends invalid data to the web application with the intention to make it do something that the application was not designed/programmed to do.

- **SQL query consuming untrusted data**.
- Examples

String query = “SELECT * FROM accounts WHERE custID = ‘” +
request.getParameter(“id”) + “‘”;

- This query can be exploited by calling up the web page executing it with the following URL: *http://example.com/app/accountView?id=’ or ‘1’=’1* causing the return of all the rows stored on the database table.
- The core of a code injection vulnerability is the lack of validation and sanitization of the data used by the web application, which means that this vulnerability can be present on almost any type of technology.

INJECTION PREVENTION

- **Preventing SQL injections requires keeping data separate from commands and queries**
  - Separation of data from the web application logic.
  - Implement settings and/or restrictions to limit data exposure in case of successful injection attacks.

Broken Authentication

- **Broken authentication** is process to steal a user's login data, or forge **session** data, such as cookies, to gain unauthorized access to websites.
- How do you prevent broken authentication vulnerabilities?
  - Implement multi-factor authentication to prevent automated, credential stuffing, brute force, and stolen credential reuse attacks.
  - Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login
  - Align password length, complexity and rotation policies with **NIST**

Sensitive Data Exposure

Two types of data:

- Stored data – data at rest (Credentials, Credit card numbers, Social Security Numbers, Medical information etc)
- Transmitted data – data that is transmitted internally between servers, or to web browsers.
- SSL is the acronym for **Secure Sockets Layer**. The standard security technology for establishing an encrypted link between a web server and a browser.

XML External Entities (XXE)

- XML External Entity attack is a type of attack against an application that parses XML input. This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser.
- Vulnerable XML processors, Vulnerable code, Vulnerable dependencies, Vulnerable integrations
- **How to prevent XML external entity attacks?**
  - Use less complex data formats ,such as JSON, and avoid serialization of sensitive data.
  - Virtual patching
  - API security gateways
  - **Web Application Firewalls (WAFs)** to detect, monitor, and block XXE attacks

Broken Access Control

- Access unauthorized functionality and/or data
- View sensitive files
- Change access rights

Broken Access Control Prevention

- To avoid broken access control is to develop and configure software with a security-first philosophy.
- Unique application business limit requirements should be enforced by domain models

Security Misconfigurations

- At its core, brute force is the act of trying many possible combinations, but there are many variants of this attack to increase its success rate. Here are the most common:
  - Unpatched flaws
  - Default configurations
  - Unused pages
  - Unprotected files and directories
  - Unnecessary services

The most recent examples of application misconfigurations is the **memcached servers** used to **DDoS** huge services in the tech industry.

How to Have Secure Installation Systems

A task to review and update the configurations appropriate to all security notes, updates,

Sending security directives to clients, e.g. Security Headers.

Cross Site Scripting (XSS)

- XSS attacks consist of injecting malicious client-side scripts into a website and using the website as a propagation method.
- XSS is present in about two-thirds of all applications.

How to Prevent XSS Vulnerabilities

- Using frameworks that automatically escape XSS by design, such as Ruby on Rails, React JS
- Enabling a **content security policy (CSP)** is a defense-in-depth mitigating control against XSS.

Insecure Deserialization

One of the attack vectors presented regarding this security risk was a **super cookie** containing serialized information about the logged-in user.

The roles of the user was specified in this cookie.

How to Prevent Insecure Deserializations

- The best way to protect the web application from this type of risk is not to accept serialized objects from untrusted sources.
- Implementing integrity checks such as digital signatures on any serialized objects to prevent hostile object creation or data tampering

CLIENT SIDE vs SERVER SIDE

	Client-side Scripting	Server-side scripting
Facing	Frontend – Runs on the user's computer.	Backend – Runs on the server.
Purpose	Collection of user input, interfacing with the server.	Processes the user input, do transactions.
Processes	Mostly deals with visual and user input.	Mostly deals with transactions and complex computations.
Code Transparency	Scripts are downloaded onto the client computer, which can be accessed by the users. Processes can be easily tampered with.	Scripts are not open to users. Processes are transparent or totally invisible to the users.
Security	Restricted to a sandbox, but generally less secure as users can see and mess with the scripts.	A lot more secure as users cannot see the source code, and they usually cannot interrupt the process.

Client-side Scripting : More

Client-side Uses

- Makes interactive web pages
- Make stuff work dynamically
- Interact with temporary storage
- Works as an interface between user and server
- Sends requests to the server
- Retrieval of data from Server
- Interact with local storage
- Provides remote access for client-server program

Client-side Languages

- JavaScript
- VBScript
- HTML (Structure)
- CSS (Designing)
- AJAX
- jQuery

Server-side Programming : More

Server-side Uses

- It processes the user input
- Displays the requested pages
- Structure of web applications
- Interaction with servers/storages
- Interaction with databases
- Querying the database
- Encoding of data into HTML
- Operations over databases like delete, update.

Server-side Languages

- PHP
- ASP.NET (C# OR Visual Basic)
- C++
- Java and JSP
- Python
- Ruby on Rails and so on.