

# Full Stack Application Development

Understanding the Basic Web application

Akshaya Ganesan





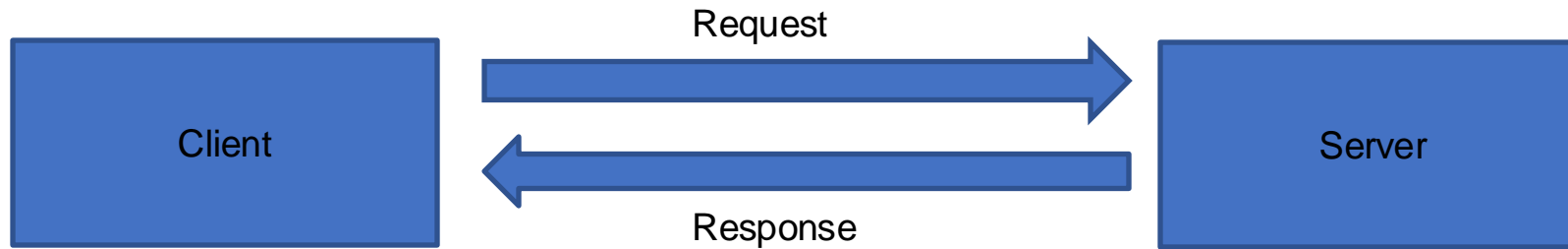
# Client Server Pattern

- **Context:** shared resources and services accessed by distributed clients
- **Problem:** By managing a set of shared resources and services, promote modifiability and reuse
- **Solution:** Clients interact by requesting services of servers, which provide a set of services. Some components may act as both clients and servers.



# Client Server Architecture

- Request – Response Model
- Providers of a resource or service, Server
- Service requester/Consumer - Client





# Client Server Architecture

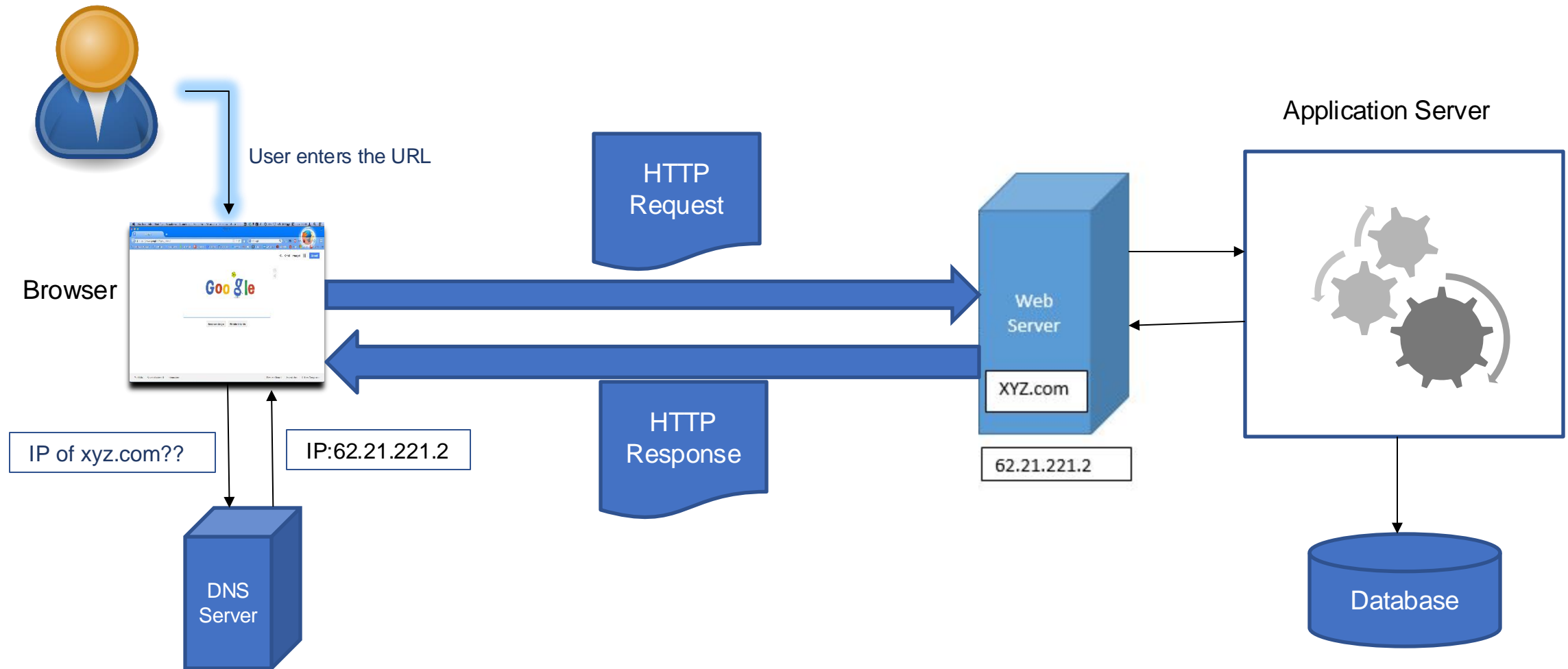
- **Benefits**
  - ✓ **Higher security**
  - ✓ **Centralized data access.**
  - ✓ **Ease of maintenance.**
- **The client-server architectural style has evolved into the more general 3-tier (or N-tier) architectural style for the web**



# Variants of Client Server Pattern:

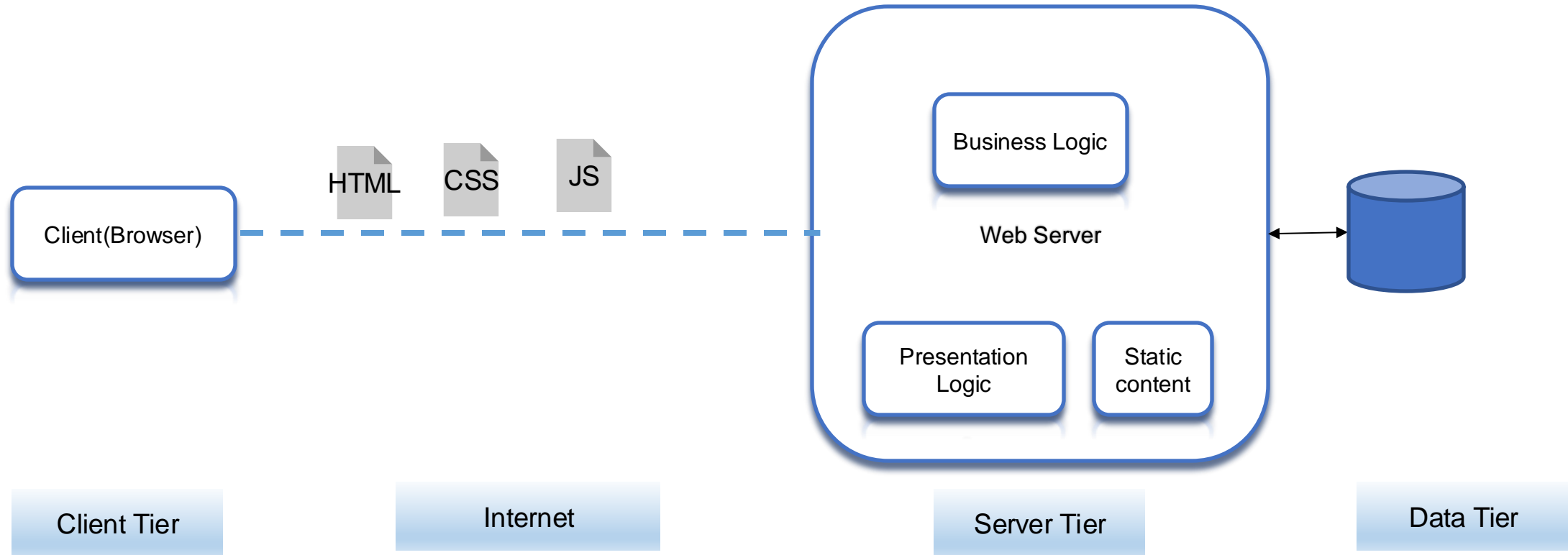
- Peer-to-Peer (P2P) applications
- Application servers
- Variations
  - ✓ Web browsers don't block until the data request is served up - Asynchronous
  - ✓ In some client-server patterns, servers can initiate certain actions on their clients.
  - ✓ Service calls over a request/reply connector are bracketed by a “session”

# Working of the Web



# Traditional Web Application

- Three Tier

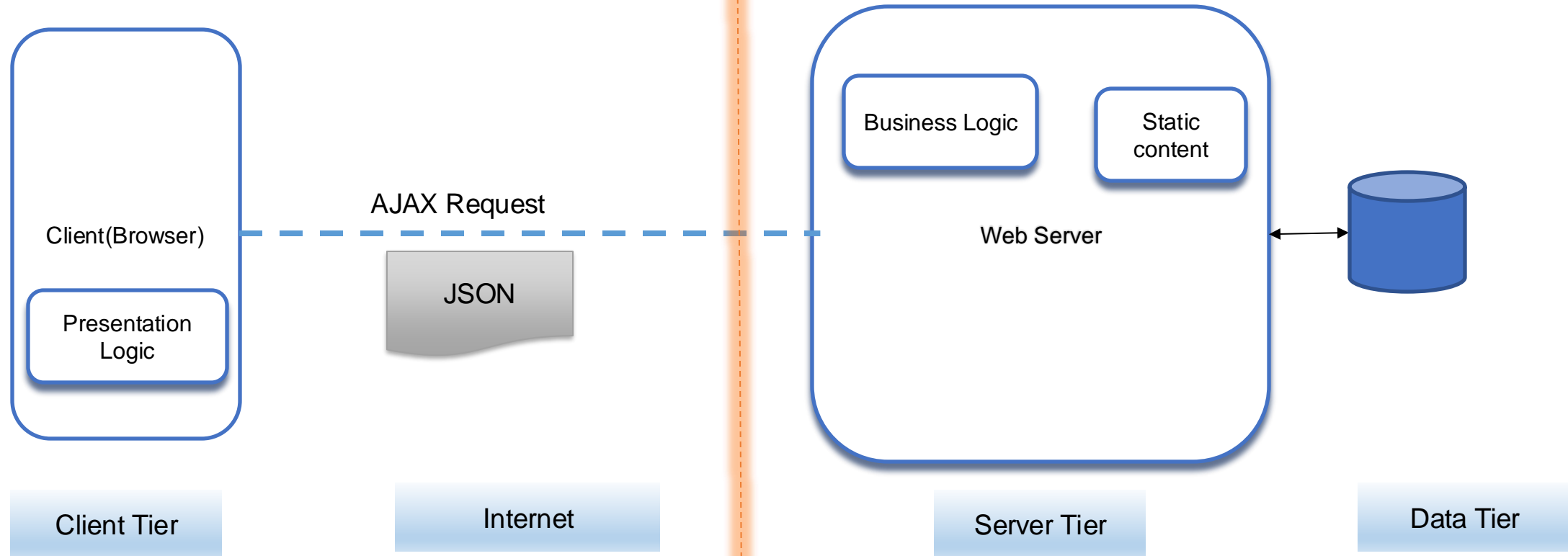




# Modern Web Application

## Frontend

## Backend





# AJAX



- Asynchronous JavaScript and XML (Ajax) refer to a concept that is used to develop web applications in a better way.
- Ajax defines a method of initiating client-to-server communication without page reloads.
- It provides a way to enable partial page updates.
- In an Ajax web application, the user is not interrupted in interactions with the web application.

# Model of a Web application

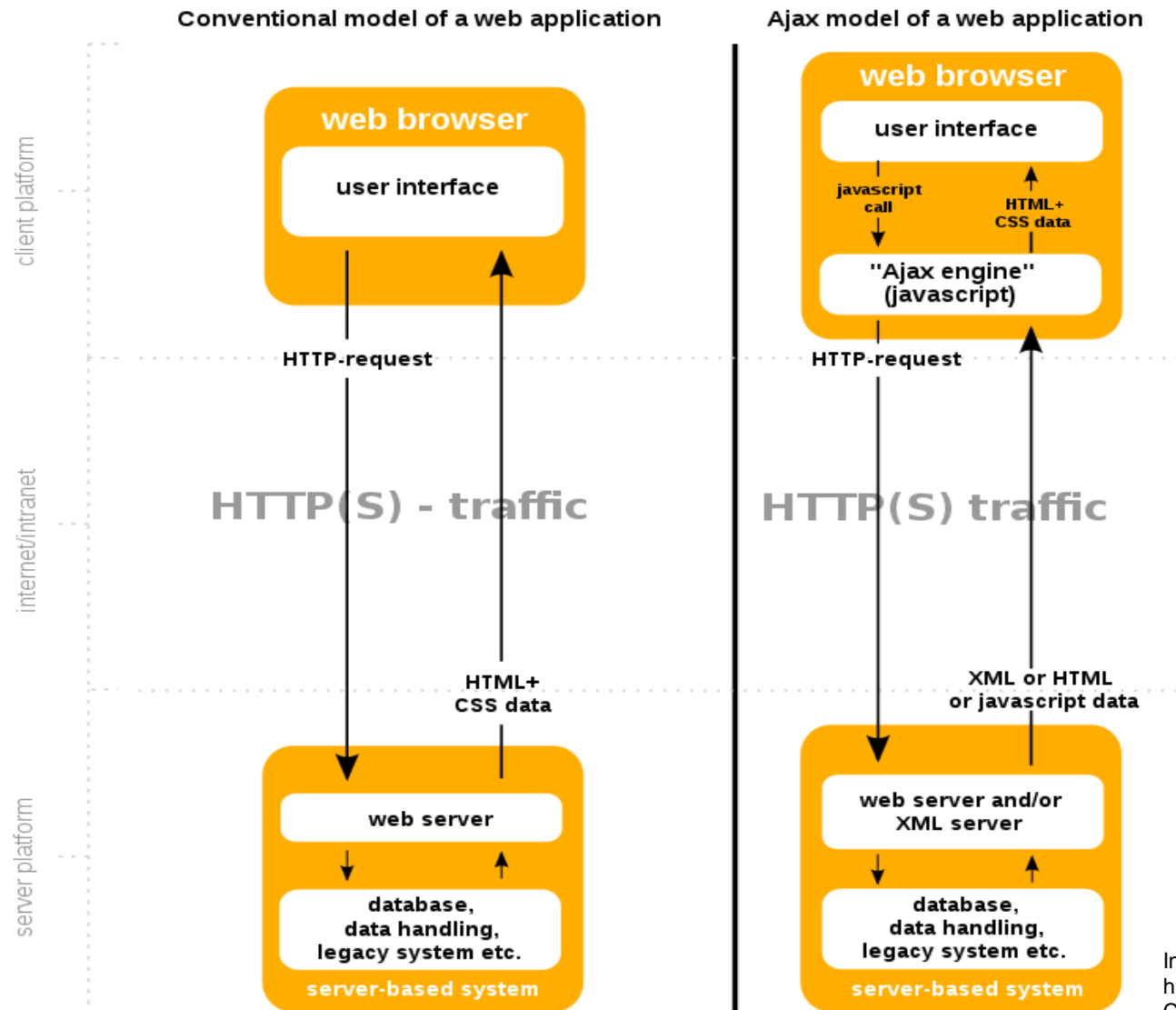


Image source: By DanielSHaischt, via Wikimedia Commons - <https://commons.wikimedia.org/wiki/File%3AAjax-vergleich.svg>, CC BY-SA 3.0,



# Front-end Responsibilities

- User Interface elements
- Mark-up and web languages such as HTML, CSS, JavaScript and supporting libraries
- Asynchronous request handling and AJAX
- Single-page applications (with frameworks like React, AngularJS or Vue.js)
- Web performance
- Responsive web design
- Cross-browser compatibility issues and workarounds
- End-to-end testing with a headless browser
- Build automation to transform and bundle JavaScript files, reduce image size
- Search engine optimization
- Accessibility concerns



# Backend Responsibilities

- Software Architecture
- Application Business Logic
- Application Data Access
- Database management
- Scripting languages like JavaScript, Node.js, PHP, Python, Ruby, Java etc.
- Automated testing frameworks for the language being used
- Scalability
- High availability
- Security concerns, authentication and authorization



# Parts of an URL

- <http://abc.company.com:80/a/b/c.html?user=John&year=2020#p2>
- The scheme identifies the protocol used to fetch the content.
- Host name name of a machine to connect to.
- The server's port number allows multiple servers to run on the same machine.
- The hierarchical portion is used by the server to find content.
- The Query parameters provide additional parameters
- Fragment : Have the browser scroll the page to a specific part of the webpage fragment



# Different types of links

- Full URL: `<a href="http://www.xyz.com/news/a.html"> News</a>`
- Absolute URL: `<a href="/stock/quote.html">`
  - same as `http://www.xyz.com/stock/quote.html`
- Relative URL (intra-site links): `<a href="b/March.html">`
  - same as `http://www.xyz.com/news/2008/March.html`
- Define an anchor point (a position that can be referenced with # notation):
  - `<a id="sec3">`
- Go to a different place in the same page: `<a href="#sec3">`



# URL Encoding

- How are special characters sent in the URL?
  - `http://www.xyz.com/companyInfo?name=A&B CO`
  - Any character in a URL other than A-Z, a-z, 0-9, or any of -\_~ must be represented as %xx, where xx is the hexadecimal value of the character:
  - `http://www.xyz.com/companyInfo?name=A%26B%20CO`
- Escaping is a commonly used technique.



# Domain Name System

- The Domain Name System (DNS) is the phonebook of the Internet.
- Humans access information online through domain names such as google.com or facebook.com.
- The network of devices interacts through Internet Protocol (IP) addresses.
- DNS translates domain names to IP addresses so browsers can load Internet resources.
- Each device connected to the Internet has a unique IP address, which other machines use to find the device.
- DNS servers eliminate the need for humans to memorize IP addresses





# Domain Name System

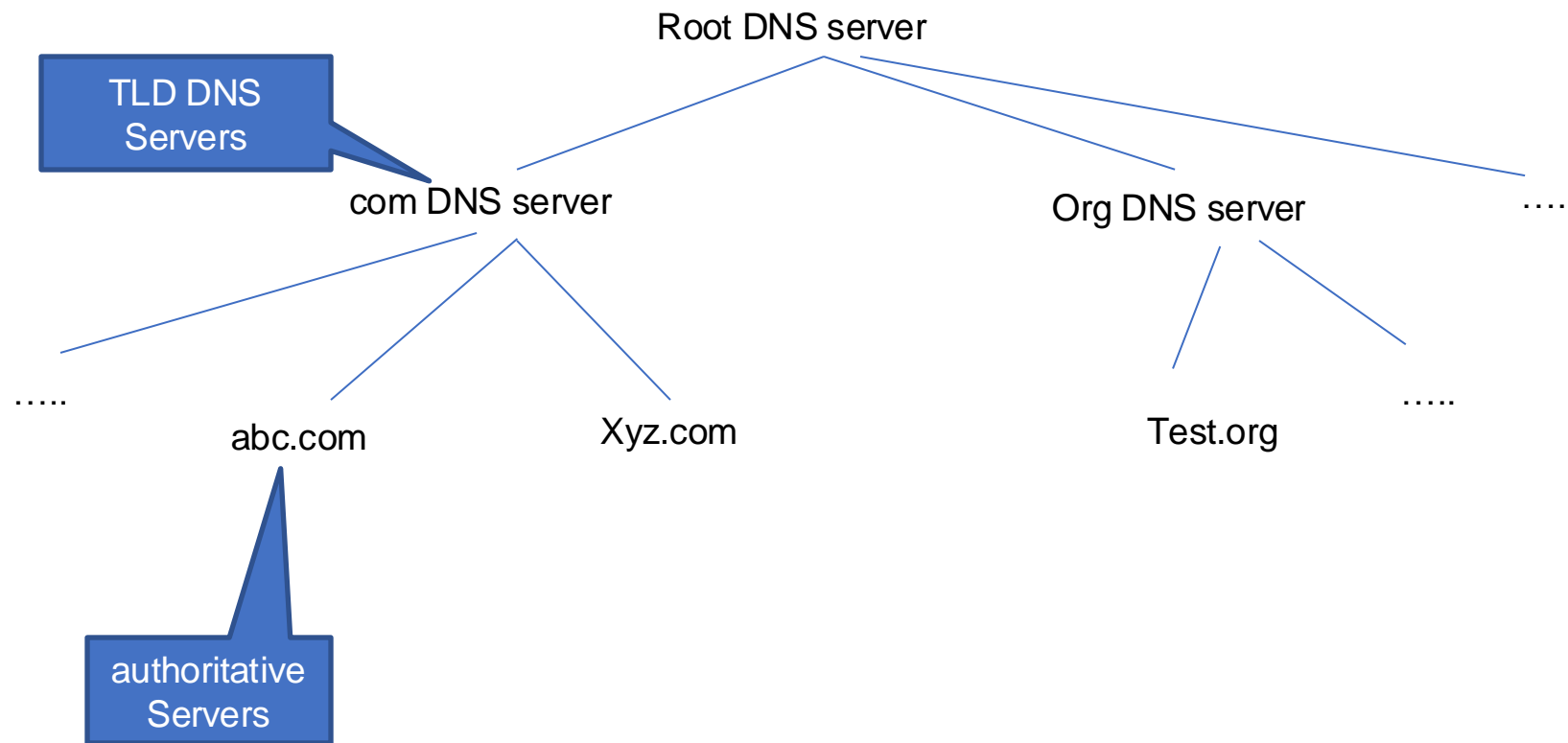
- The Domain Name System resolves the names of internet sites with their underlying IP addresses.
- A DNS server is a computer server that contains a database of public IP addresses and their associated hostnames.
- DNS is a distributed database implemented in a hierarchy of name servers.
- It is an application layer protocol for message exchange between clients and servers.



# Domain Name System

- The **DNS recursor** is a server designed to receive queries from client machines through applications such as web browsers.
- The **root server** is the first step in translating (resolving) human-readable host names into IP addresses.
- TLD nameserver - The top level domain server (TLD)
- The authoritative nameserver is the last stop in the nameserver query

# Domain Name System Lookup





# Content Delivery Network

- A content delivery network (CDN) is a geographically distributed group of servers that caches content close to end users.
- A CDN allows for the quick transfer of assets needed for loading Internet content, including HTML pages, JavaScript files, stylesheets, images, and videos.
- **Is a CDN the same as a web host?**
- **CDN can't replace the need for proper web hosting, It improve Web performance!!**

# CDN



- A CDN improves website load times.
- The globally distributed nature of a CDN reduces the distance between users and website resources.
- A CDN caches content (such as images, videos, or webpages) in proxy servers that are located closer to end users than origin servers.
- CDNs also reduce the amount of data transferred by reducing file sizes using minification and file compression tactics.
- Load balancing distributes network traffic evenly across several servers, making it easier to scale rapid boosts in traffic.



# Benefits of using a CDN

- Improving website load times
- Reducing bandwidth costs
- Increasing content availability and redundancy
- Improving website security

# Web hosting

[https://w3techs.com/technologies/history\\_overview/web\\_hosting](https://w3techs.com/technologies/history_overview/web_hosting)



	2023 1 Aug	2023 1 Sep	2023 1 Oct	2023 1 Nov	2023 1 Dec	2024 1 Jan	2024 1 Feb	2024 1 Mar	2024 1 Apr	2024 1 May	2024 1 Jun	2024 1 Jul	2024 1 Aug	2024 16 Aug
Amazon	6.0%	6.0%	6.0%	6.0%	6.0%	5.9%	5.9%	5.8%	5.8%	5.7%	5.6%	5.6%	5.5%	5.5%
Shopify	4.0%	4.0%	4.0%	4.1%	4.1%	4.2%	4.3%	4.3%	4.3%	4.4%	4.4%	4.5%	4.5%	4.5%
Newfold Digital Group	4.1%	4.1%	4.1%	4.0%	3.9%	3.9%	3.8%	3.7%	3.7%	3.6%	3.6%	3.5%	3.5%	3.4%
OVH	3.3%	3.3%	3.3%	3.3%	3.3%	3.2%	3.2%	3.2%	3.2%	3.2%	3.2%	3.2%	3.1%	3.1%
Hostinger	1.6%	1.7%	1.8%	1.9%	2.0%	2.0%	2.1%	2.3%	2.6%	2.7%	2.8%	2.9%	3.0%	3.1%
Wix	2.5%	2.5%	2.5%	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%	2.7%	2.7%	2.8%	2.8%	2.9%
GoDaddy Group	3.3%	3.1%	3.0%	2.9%	2.9%	2.9%	2.8%	2.8%	2.8%	2.7%	2.7%	2.7%	2.7%	2.6%
Hetzner	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%	2.6%
United Internet	2.1%	2.1%	2.1%	2.1%	2.1%	2.2%	2.2%	2.3%	2.3%	2.3%	2.4%	2.4%	2.4%	2.4%
Squarespace	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.2%
SiteGround	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%
team.blue	1.8%	1.8%	1.8%	1.8%	1.8%	1.9%	1.9%	1.9%	1.9%	1.9%	1.9%	1.9%	1.9%	1.9%

# Rendering Patterns



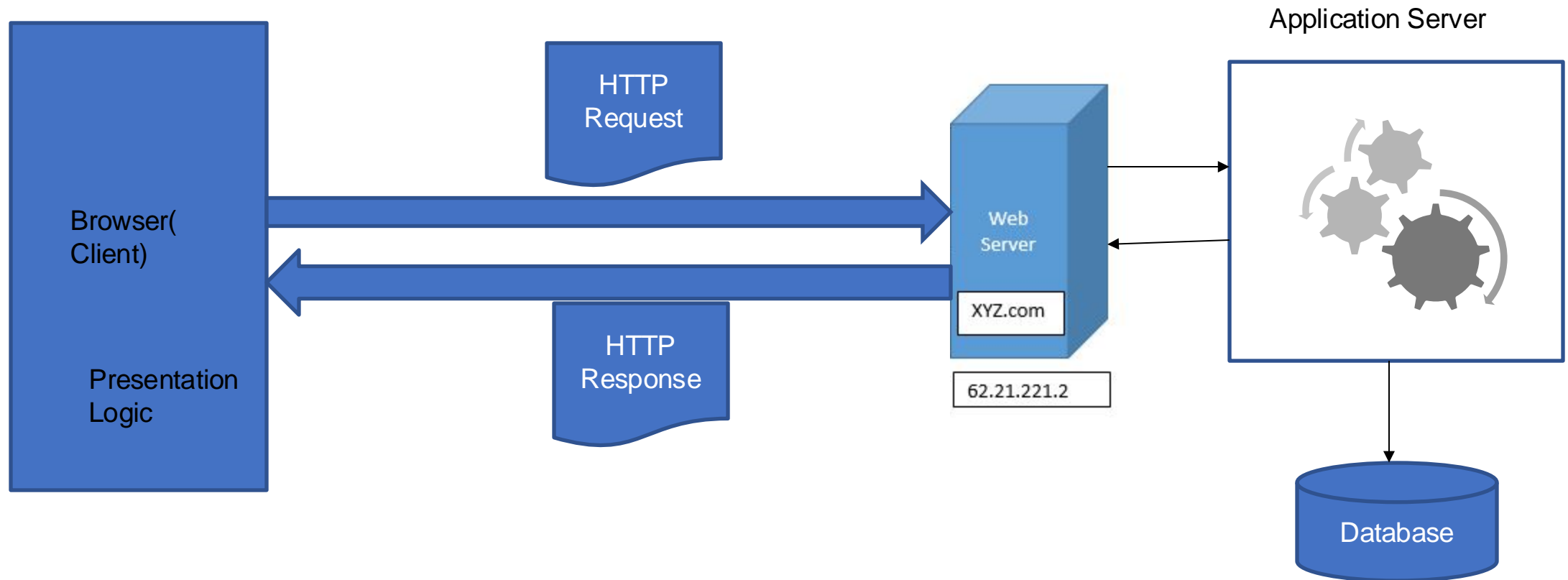




# Client-side rendering (CSR)

- **Client-side rendering (CSR) is a technique for rendering web content on the client-side**, i.e., in the user's browser.
- With CSR, the client requests a minimal HTML file from the server containing the necessary JavaScript and CSS files.
- When the client loads the JavaScript files, the JavaScript code is executed, which renders the content in the browser.

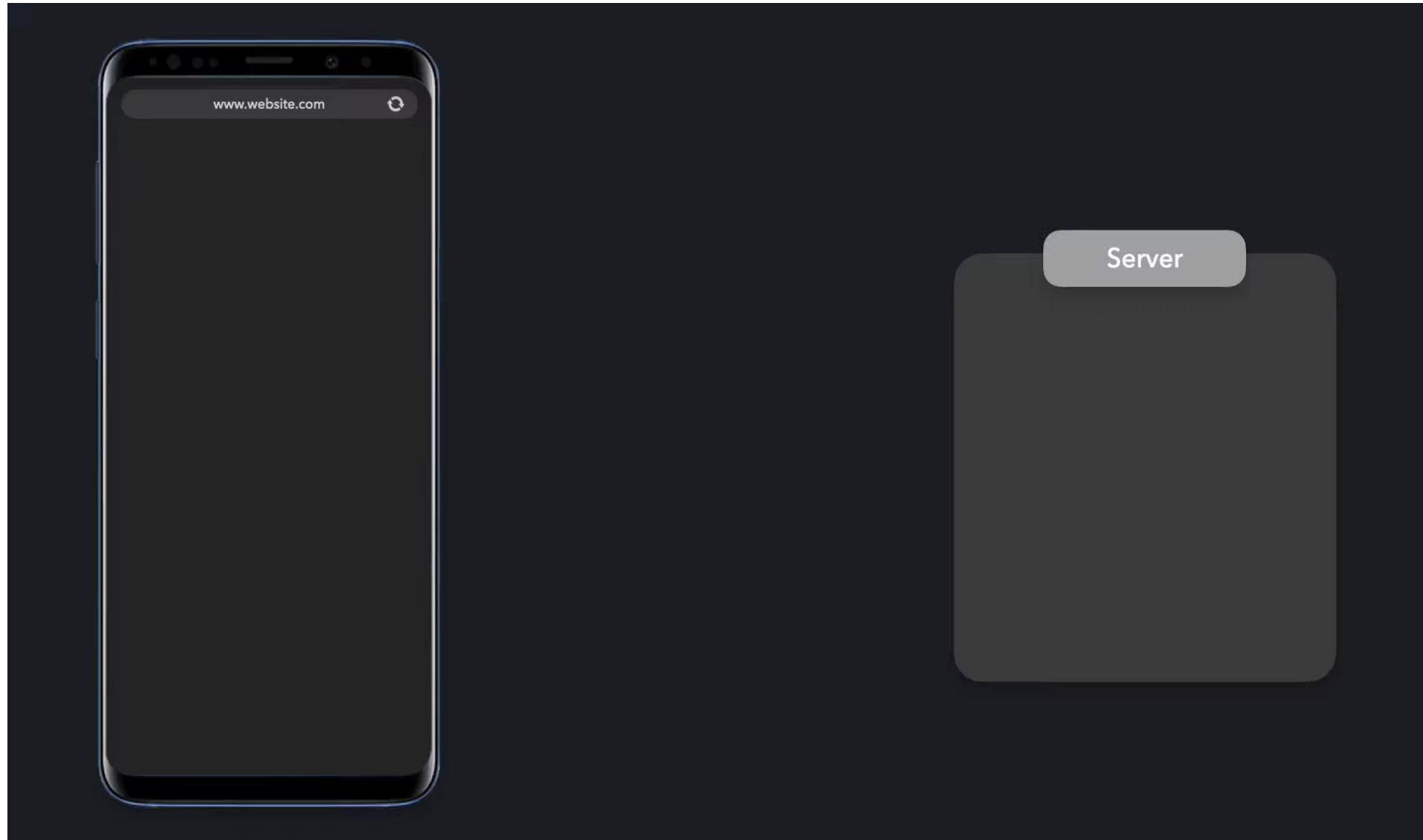
# Client Side rendering





# Server-Side Rendering

- Server-side rendering (SSR) is a technique for rendering web content on the server-side, i.e., before the page is sent to the client.
- Server-Side Rendering is also named Pre-Rendering because the fetching of external data and transformation of components, content, and data into HTML happens before the result is sent to the client.
- There are several benefits of using server-side rendering:
  - Better SEO
  - Faster Initial Page Load
  - Improved Accessibility
  - Better Performance on Low-Powered Devices





# Static Site Generation

- Static site generation (SSG) is a popular approach to website development that involves generating a website's content ahead of time and delivering it as static HTML files to end-users.
- In SSG, the HTML is generated once, at build time.
- The HTML is stored in a CDN or elsewhere and re-used for each request.
- A static site generator combines the content and templates into a collection of static HTML files.
- This process may also include optimizing images, minifying code, and generating metadata.



# Static Site Generation

- Benefits:
  - Performance
  - SEO
  - Cost
- Limitations:
  - Not suitable for all types
  - No Real-time data
- Some popular static site generators include Jekyll, Hugo, and Gatsby.



# Client-Side Programming

- Involves everything users see on their screens.
- Major frontend technology stack components:
  - **HTML, CSS and JS**
- Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS)
  - HTML tells a browser how to display the content of web pages
  - CSS styles that content
  - Bootstrap is a helpful framework for managing HTML and CSS
- JavaScript (JS)
  - Makes web pages interactive
  - Many JavaScript libraries (such as jQuery, React.js)
  - frameworks (such as Angular, Vue, Backbone, and Ember)



# Server-Side Programming

- Major server-side technology stack components:
  - Programming language, Framework, web server and databases
- Server-side programming languages used to create the logic of applications
- Frameworks offer lots of tools for simpler and faster development of applications
- Options
  - Ruby (Ruby on Rails)
  - Python (Django, Flask, Pylons)
  - PHP (Laravel)
  - Java (Spring)
  - Scala (Play)
  - Javascript (Express Node.js)





# Server-Side Programming

- Storage
  - Apps needs a place to store its data
  - Two types of databases:
    - relational and non-relational
  - Most common databases for web development:
    - MySQL (relational)
    - PostgreSQL (relational)
    - MongoDB (non-relational, document)



# Server-Side Programming

- Caching system
  - Used to reduce the load on the database and to handle large amounts of traffic
  - Memcached and Redis are the most widespread.
- Web servers/Load balancers/Proxy servers
  - Needs a server to handle requests from clients' computers
  - Two major players: Apache, Nginx
  - Cloud Based Servers (EC2, Serverless, ELB)



# Example TechStack

- MEAN / MERN / MEVN Stack
  - MongoDB: A NoSQL database that stores data in a flexible, JSON-like format.
  - Express.js: A minimal and flexible Node.js web application framework that provides robust features for web and mobile applications.
  - Frontend Framework
    - Vue.js: A JavaScript framework for building user interfaces.
    - Angular: A TypeScript-based open-source front-end web application framework maintained by Google.
    - React: A JavaScript library for building user interfaces, developed by Facebook.
  - Node.js: A JavaScript runtime built on Chrome's V8 JavaScript engine. It allows developers to use JavaScript for server-side scripting.
- Python Stack:
  - Flask or Django: Flask is a lightweight Python web framework, and Django is a high-level Python web framework.
  - SQLAlchemy: An SQL toolkit and Object-Relational Mapping (ORM) library.
  - Django REST framework: If using Django for building APIs.
  - Database: Various options, including SQLite, PostgreSQL, or MySQL.
  - HTML/CSS/JavaScript or React/Angular for frontend



# References

- <https://developer.chrome.com/docs/devtools/>
- [https://developer.mozilla.org/en-US/docs/Learn/Tools\\_and\\_testing/Client-side\\_JavaScript\\_frameworks/Introduction](https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Introduction)
- Chapter 1: Full Stack Web Development: The Comprehensive Guide by Philip Ackermann Shroff/Rheinwerk Computing; First Edition (2 August 2023)

# Thank You!

