Full Stack Application Development

BITS Pilani

innovate achieve lead

BITS Pilani
Pilani Campus

Module 1:
Modern Applications Landscape

What to you expect from a Modern Application?

innovate achieve lead

What to expect from Modern Applications?

innovate achieve lead

A Modern App is a resilient, multi-cloud *supportive* software service *comprised of* orchestrated releases *of* virtual machines, containers, and serverless functions.

https://octo.vmware.com/defining-modern-application/

Today's most relevant technologies

innovate achieve lead

ChatGPT

Bard

Duet AI for Google Workspace

Duet AI brings the power of generative AI to Google Workspace. Integrated across all Google Workspace apps, Duet AI enhances business workflows and productivity by enabling text generation and summarization, image generation from prompts, data organization, and much more.

Llama

Llama is Meta's answer to the growing demand for LLMs. Unlike its well-known technological relative, ChatGPT, Llama can run in full-on under-specced machines such as a MacBook Pro.

GitHub Copilot

Using GitHub Copilot can feel like magic. The tool automatically fills out entire blocks of code—but leveraging it requires true skill.

WebAssembly

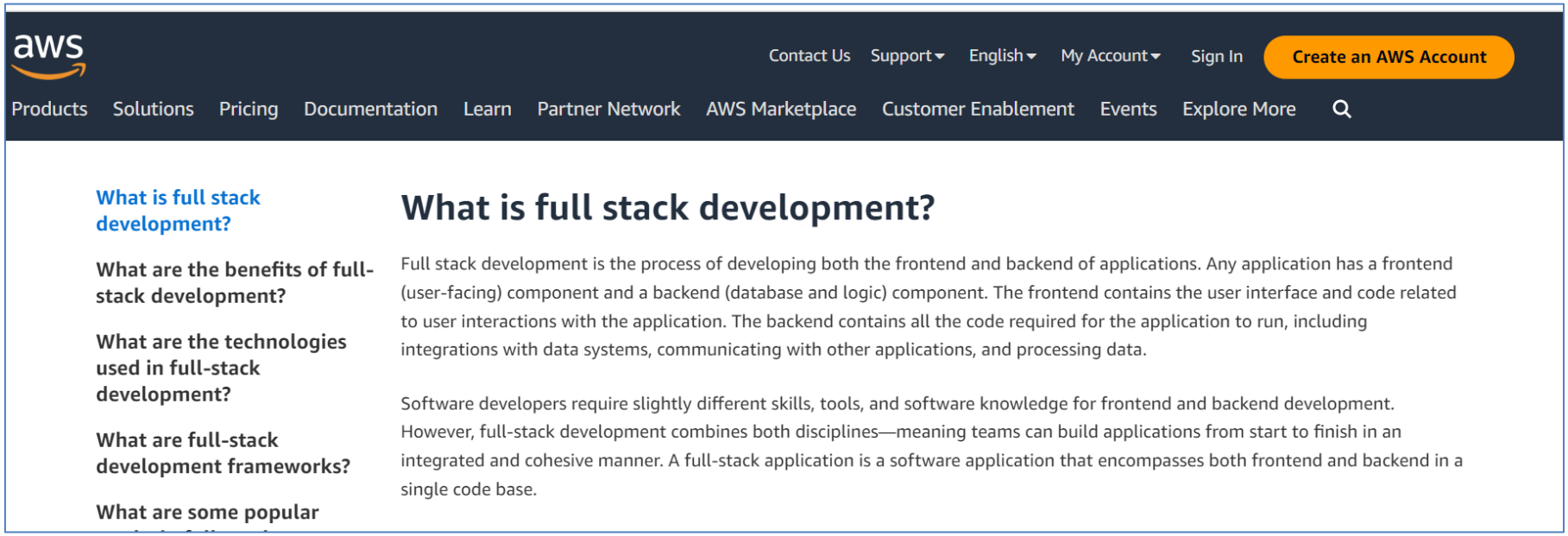
WebAssembly (Wasm) is a collection of standards for facilitating safe, fast, portable code, deployable into almost any modern runtime environment—browser, server, cloud, edge, and embedded.

You learn all technologies ?

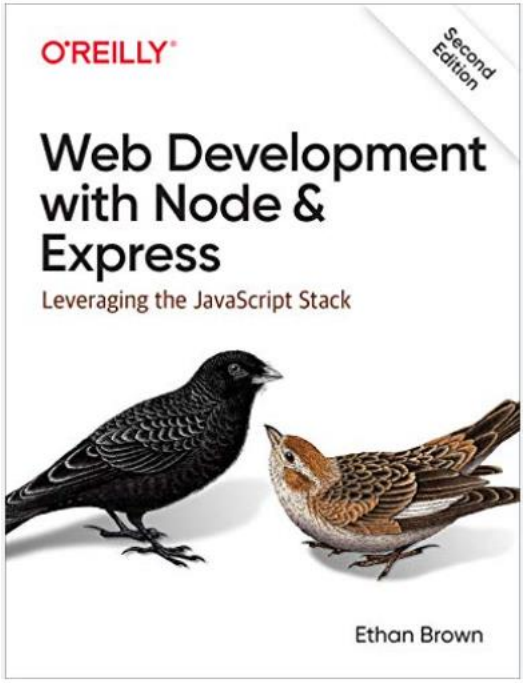
Full Stack Development

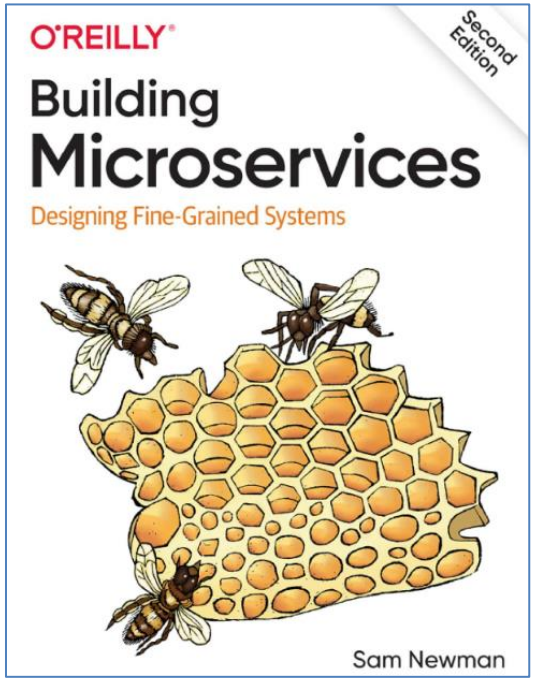
innovate achieve lead

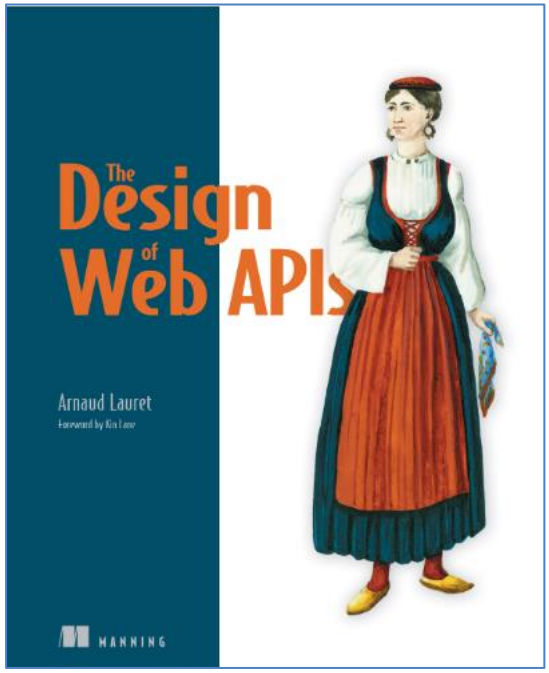
https://aws.amazon.com/what-is/full-stack-development/



Text & Reference books







innovate achieve lead

BITS Pilani

Pilani Campus

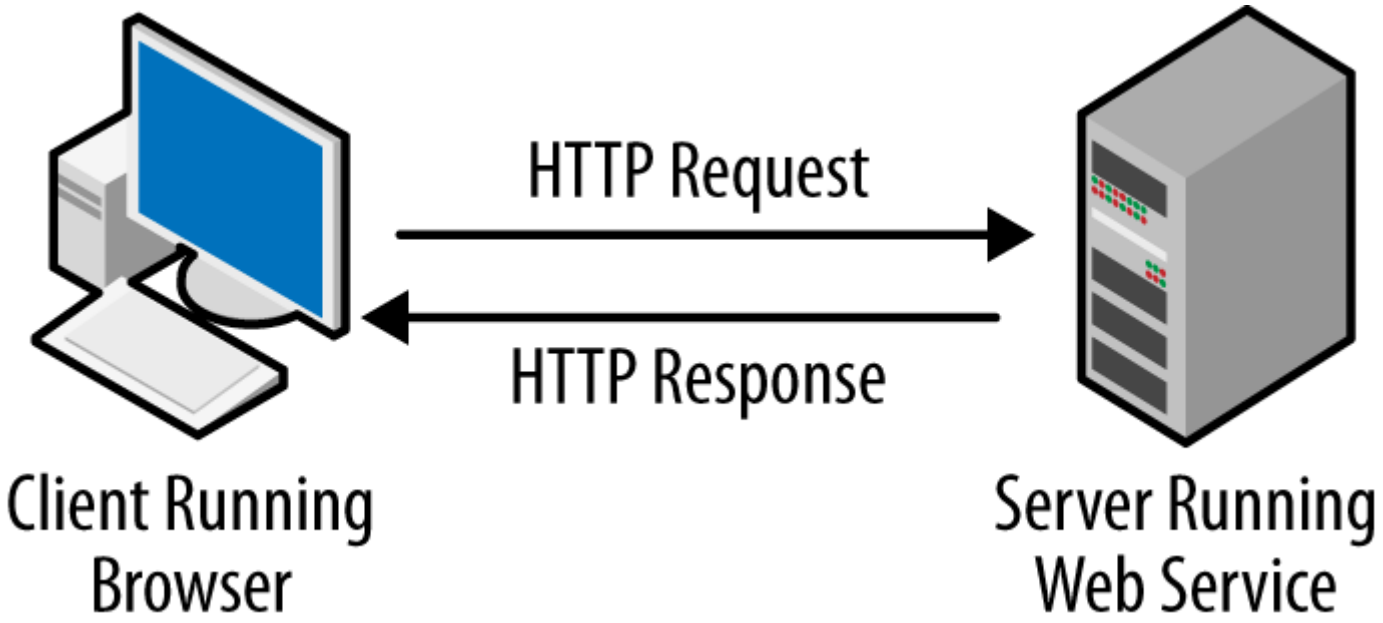
Topic 1.1: Web Apps

Client - Server Application

Useful when several clients want to access the services provided by the server

Examples

- Database client & DB server (using ODBC)
- Mail client & Mail server (MS Outlook mail client)
- Web client and Web server (HTTP request – reply)



Client – Server

Variants of client – server

- Earlier client server communication was synchronous

Now we have

- Asynchronous communication (using AJAX)
- Clients providing call back functions (ex. Javascript)
- Sessions containing series of requests and responses (Request – reply bracketed sessions)

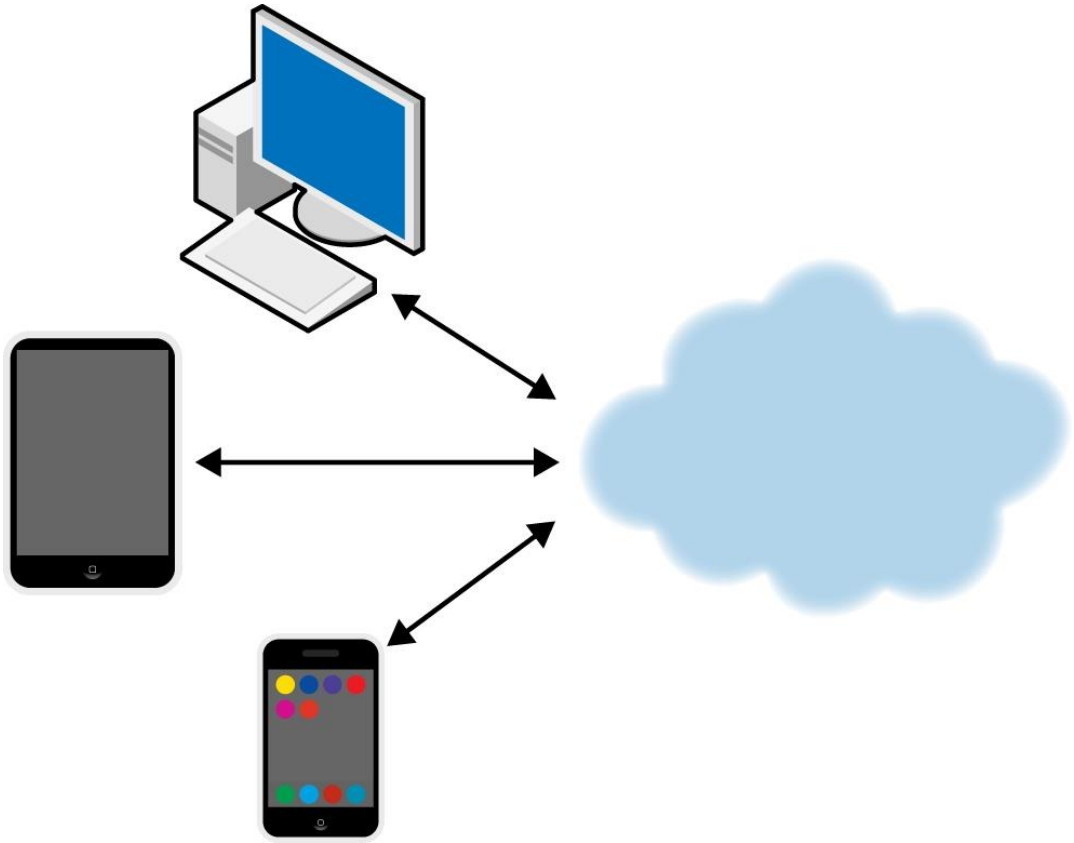
Demerits

- **Server can be a single point of failure**
- Server can be a bottleneck

Web Application

What is a Web Application?

Examples?



Web Applications

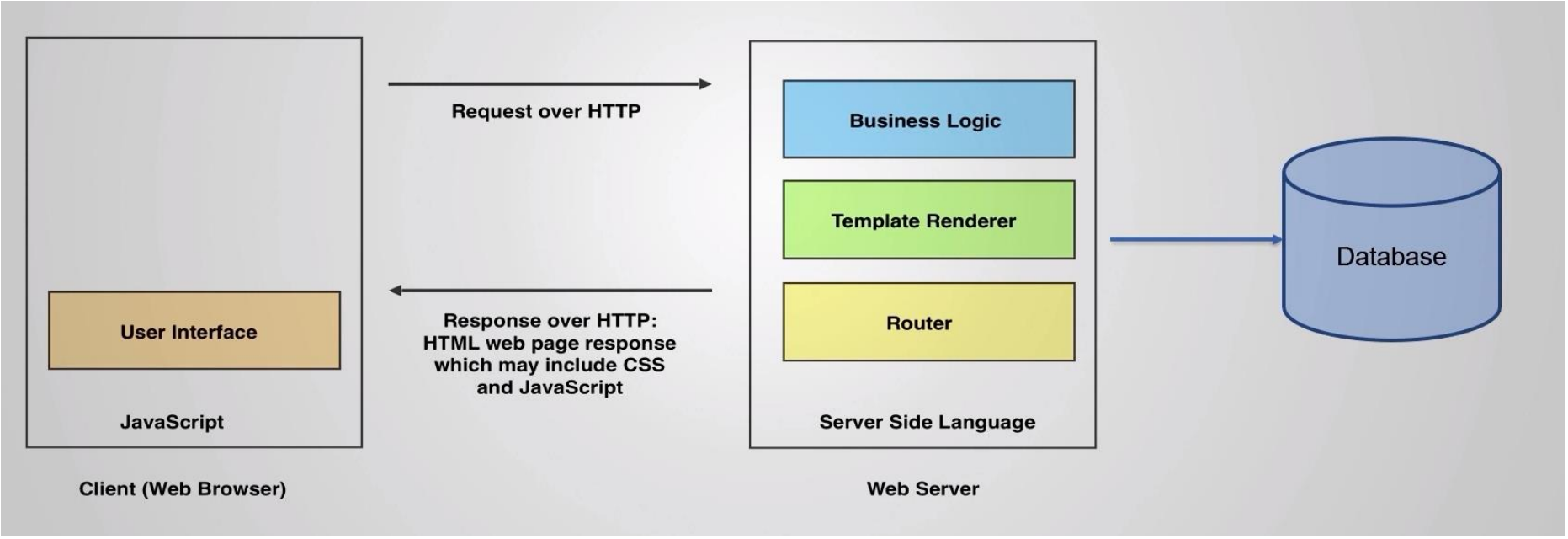
- **Application software** that runs on a (usually) remote computer
  - **Hosted** on a web server
  - **Accessible** over the web with an internet connection through browsers
  - These websites **use programming languages** like PHP, Python, or Ruby, to implement business logic.
  - They use databases to store and retrieve information.

Web Applications

A Web application (Web app) is **an application program that is stored on a remote server and delivered over the Internet through a browser interface**. Web services are Web apps by definition and many, although not all, websites contain Web apps.

Web applications include **online forms, shopping carts, word processors, spreadsheets, video and photo editing, file conversion, file scanning, and email programs such as Gmail, Yahoo and AOL**. Popular applications include Google Apps and Microsoft 365.

Web Application Architecture
Three Layered



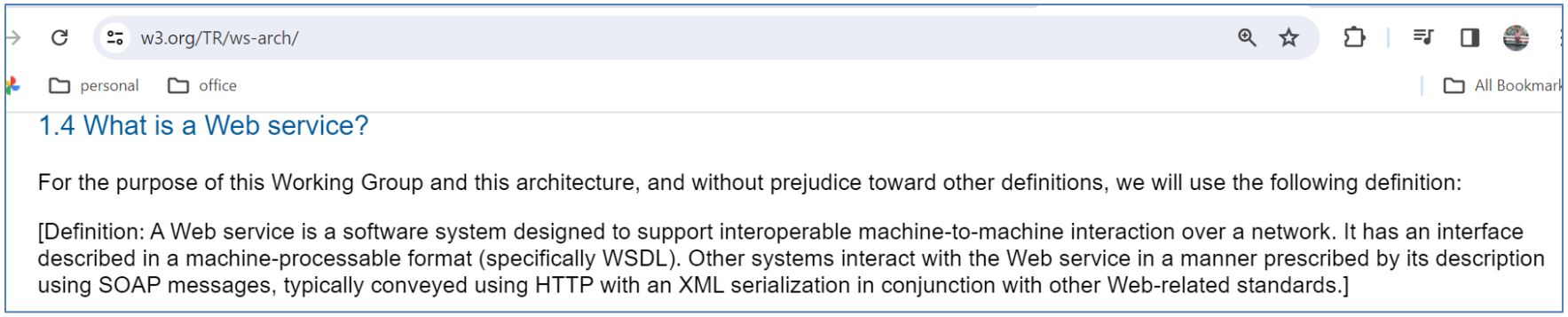
Question

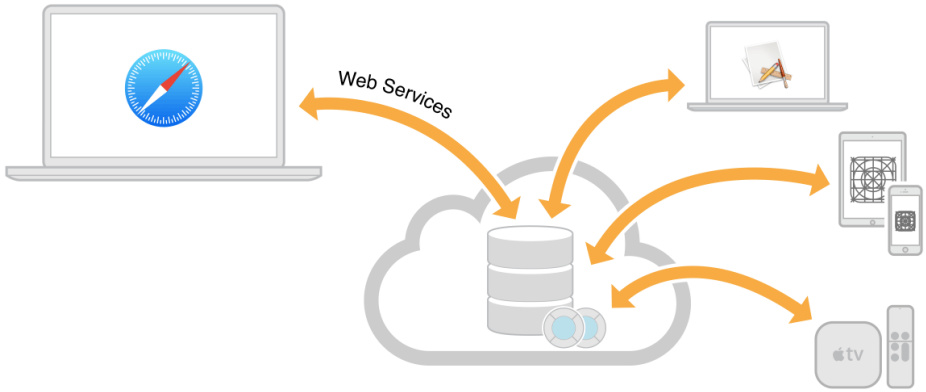
So if I deliver a service over the internet, over HTTP, does it really become a web service?

Facebook, Twitter, your shopping applications, wide variety of applications, are all of them web services?

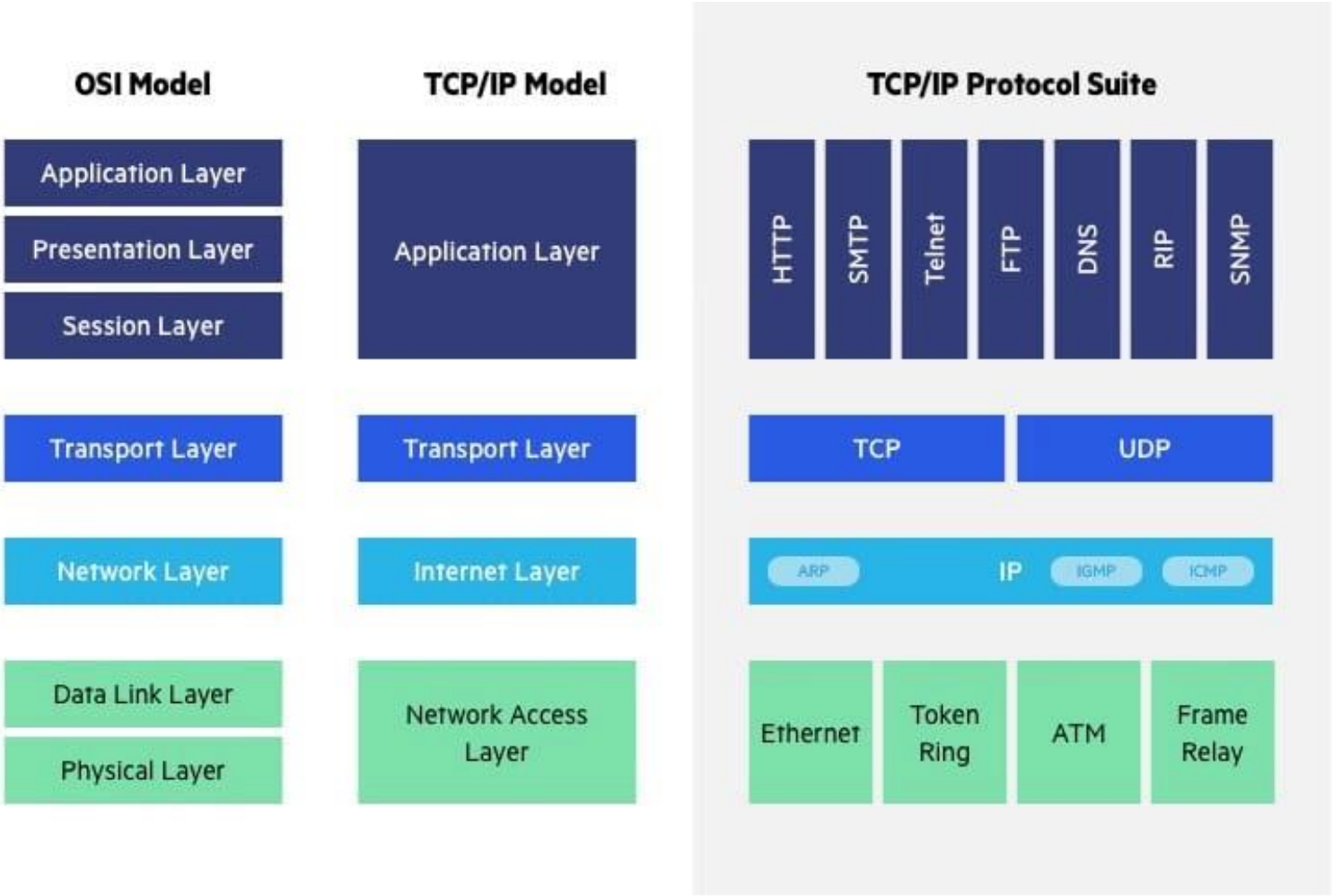
Web Service W3C Definition

https://www.w3.org/TR/ws-arch/





OSI Model


OSI Model
TCP/IP Model
TCP/IP Protocol Suite
Application Layer
Presentation Layer
Session Layer
Application Layer
HTTP
SMTP
Telnet
FTP
DNS
RIP
SNMP
Transport Layer
Transport Layer
TCP
UDP
Network Layer
Internet Layer
ARP
IP
IGMP
ICMP
Data Link Layer
Physical Layer
Network Access Layer
Ethernet
Token Ring
ATM
Frame Relay

Quiz

Which layer of the OSI model primarily handles the interactions between web browsers and web servers in a web application?

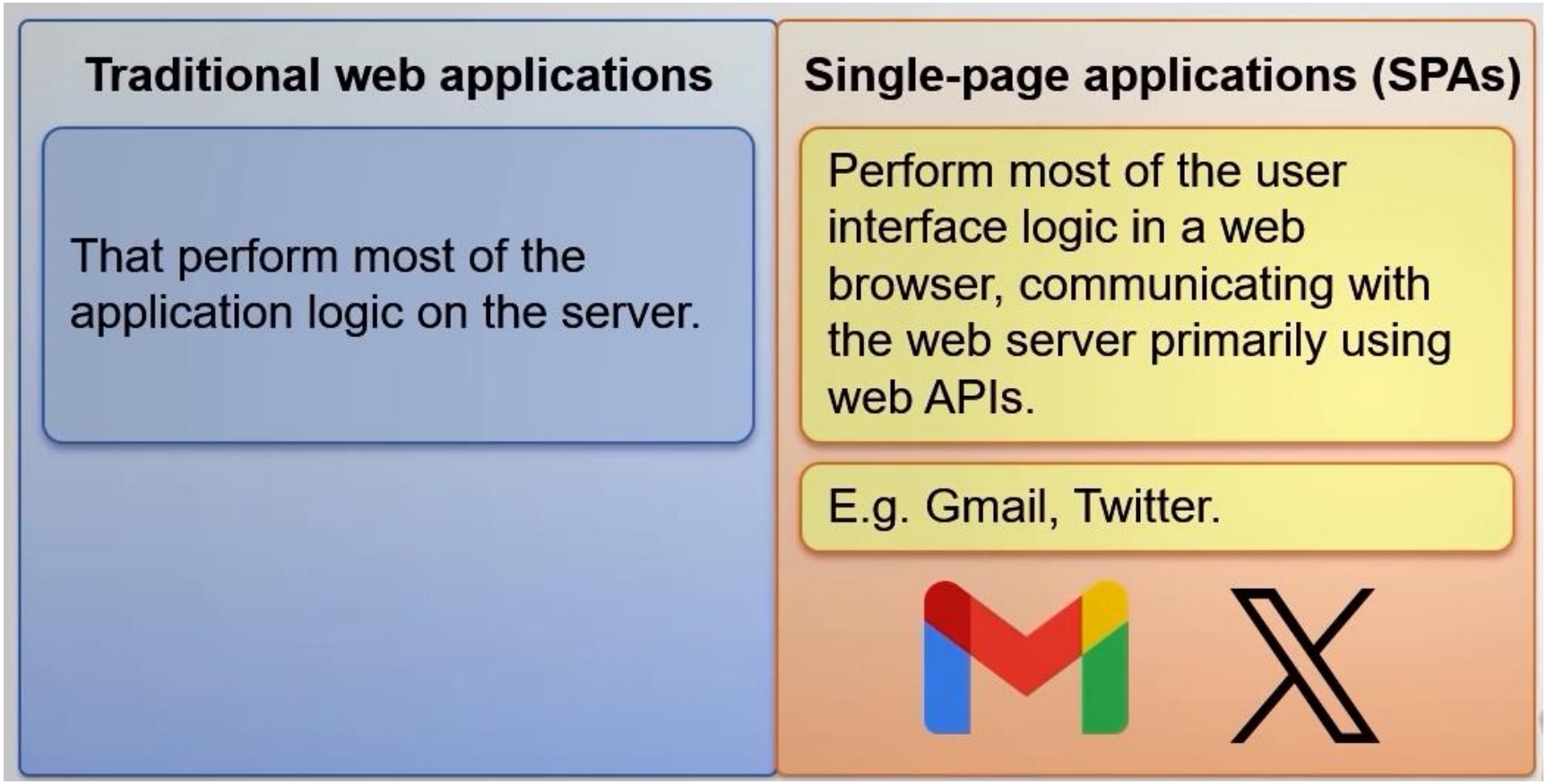
a) Application Layer
b) Transport Layer
c) Data Link Layer
d) Physical Layer

Quiz - 2

In the OSI model, which layer is responsible for ensuring the reliability of data transmission in a web application?

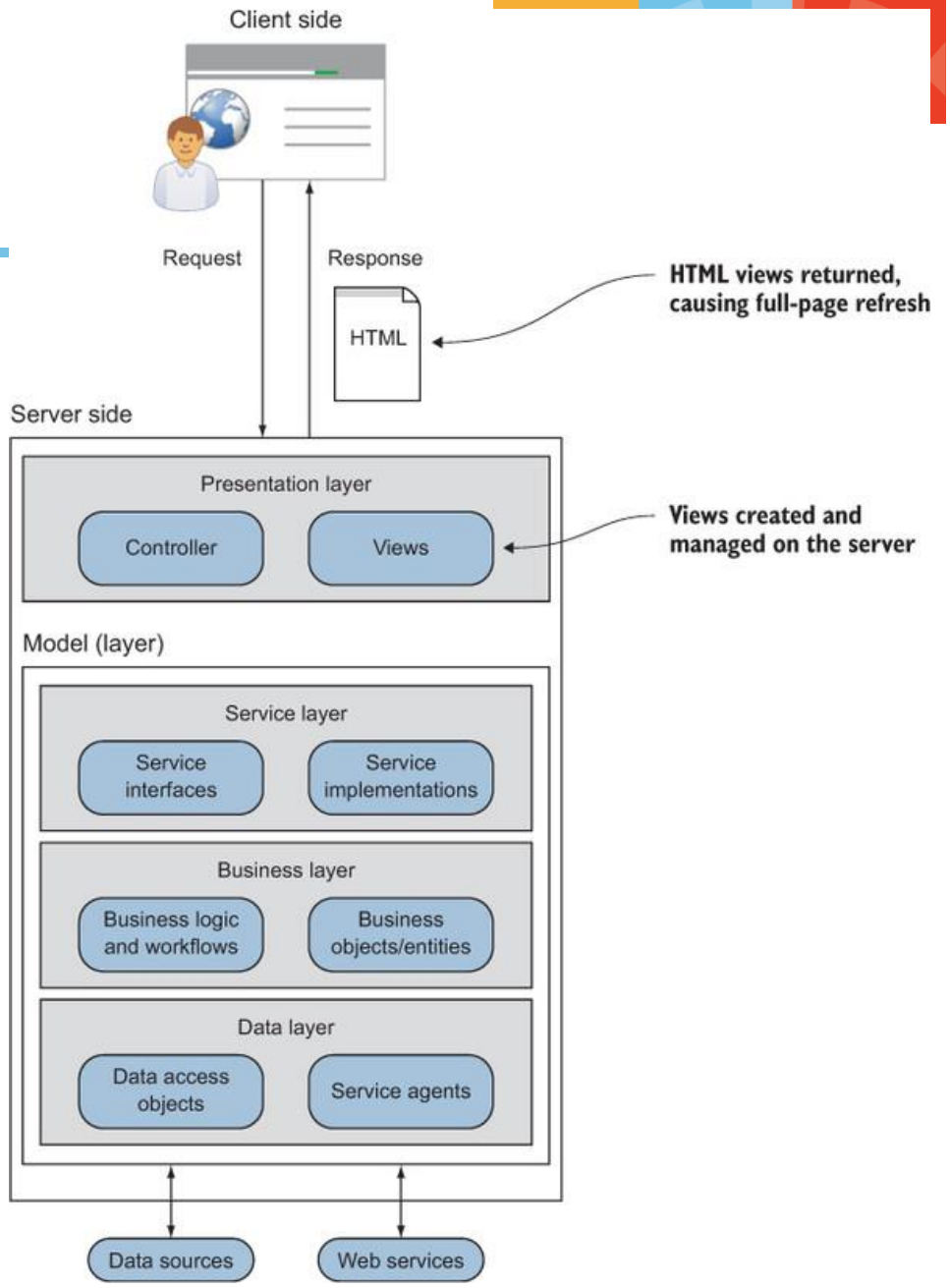
a) Application Layer
b) Session Layer
c) Transport Layer
d) Network Layer

Web Applications Development



Traditional Web Applications

With this design, each request for a new view (HTML page) results in a round-trip to the server. When fresh data is needed on the client side, the request is sent to the server side. On the server side, the request is intercepted by a controller object inside the presentation layer. The controller then interacts with the model layer via the service layer, which determines the components required to complete the model layer's task. After the data is fetched, either by a data access object (DAO) or by a service agent, any necessary changes to the data are then made by the business logic in the business layer.



Traditional Web Applications

Control is passed back to the presentation layer, where the appropriate view is chosen. Presentation logic dictates how the freshly obtained data is represented in the selected view. Often the resulting view starts off as a source file with placeholders, where data is to be inserted (and possibly other rendering instructions). This file acts as a kind of template for how the view gets stamped whenever the controller routes a request to it.

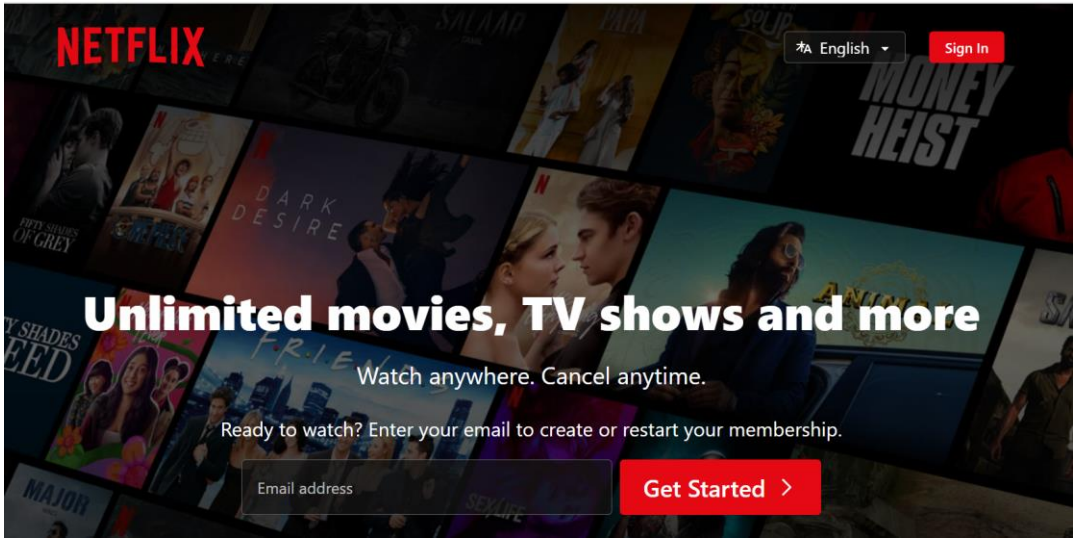
After the data and view are merged, the view is returned to the browser. The browser then receives the new HTML page and, via a UI refresh, the user sees the new view containing the requested data.

Single Page Applications

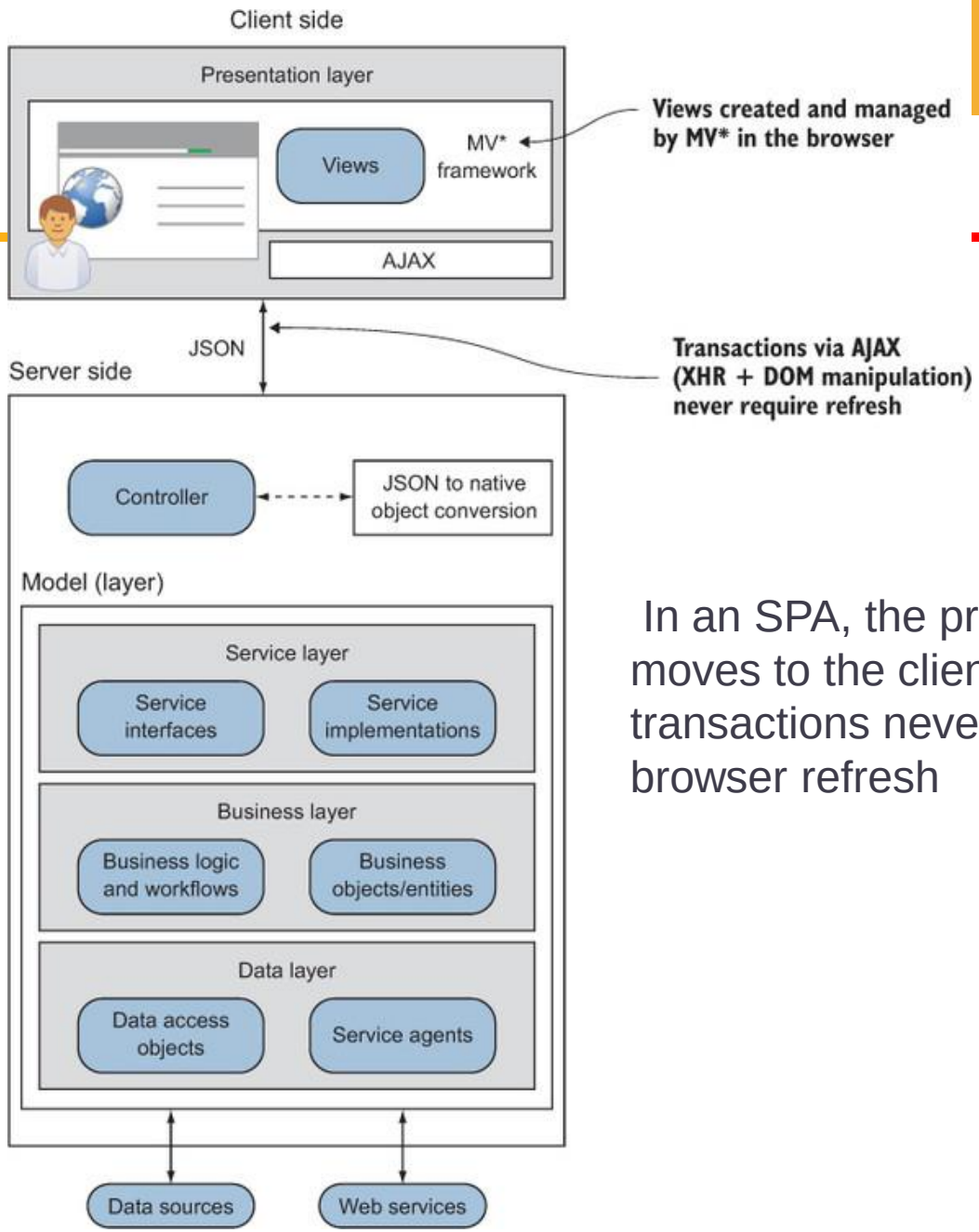
Single Page Applications (SPA)

In an SPA, the entire application runs as a single web page. In this approach, the presentation layer for the entire application has been factored out of the server and is managed from within the browser.

SPA Applications



SPA



In an SPA, the presentation layer moves to the client-side code, and transactions never require a browser refresh

SPA

Moving the process for creating and managing views into the UI decouples it from the server. From an architectural standpoint, this gives the SPA an interesting advantage. Unless you're doing partial rendering on the server, the server is no longer required to be involved in how the data is presented.

The overall SPA design is nearly the same as the traditional design. The key changes are as follows: no full browser refreshes, the presentation logic resides in the client, and server transactions can be data-only, depending on your preference for data rendering.

SPA Advantages

innovate achieve lead

Renders like a desktop application, but runs in a browser

SPA has the ability to redraw portions of the screen dynamically, and the user sees the update instantly. Because the SPA downloads the web-page structure in advance, there's no need for the disruptive request to get a new page from the server. This is similar to the experience a user would get from a native desktop application; therefore, it "feels" more natural. An advantage over even the desktop application, the SPA runs in the browser, making its native-like, browser-based environment the best of both worlds.

SPA Advantages

innovate achieve lead

Decoupled presentation layer

As mentioned previously, the code that governs how the UI appears and how it behaves is kept on the client side instead of the server. This leaves both server and client as decoupled as possible. The benefit here is that each can be maintained and updated separately.

SPA Advantages

innovate achieve lead

Faster, lightweight transaction payloads

Transactions with the server are lighter and faster, because after initial delivery, only data is sent and received from the server. Traditional applications have the overhead of having to respond with the next page's content. Because the entire page is re-rendered, the content returned in traditional applications also includes HTML markup. Asynchronous, data-only transactions make the operational aspect of this architecture extremely fast.

SPA Advantages

innovate achieve lead

Less user wait time

In today's web-centric world, the less time a user has to wait for the page to load, the more likely the person is to stay on the site and return in the future. Because the SPA loads with a shell and a small number of supporting files and then builds as the user navigates, application startup is perceived as being quick. As the previous points state, screens render quickly and smoothly, and transactions are lightweight and fast. These characteristics all lead to less user wait time. Performance isn't just a nice-to-have. It equates to real dollars when online commerce is involved.

A study by Walmart that was published in Web Performance Today indicated that for every 100 ms of performance improvement, incremental revenue grew by up to 1%.

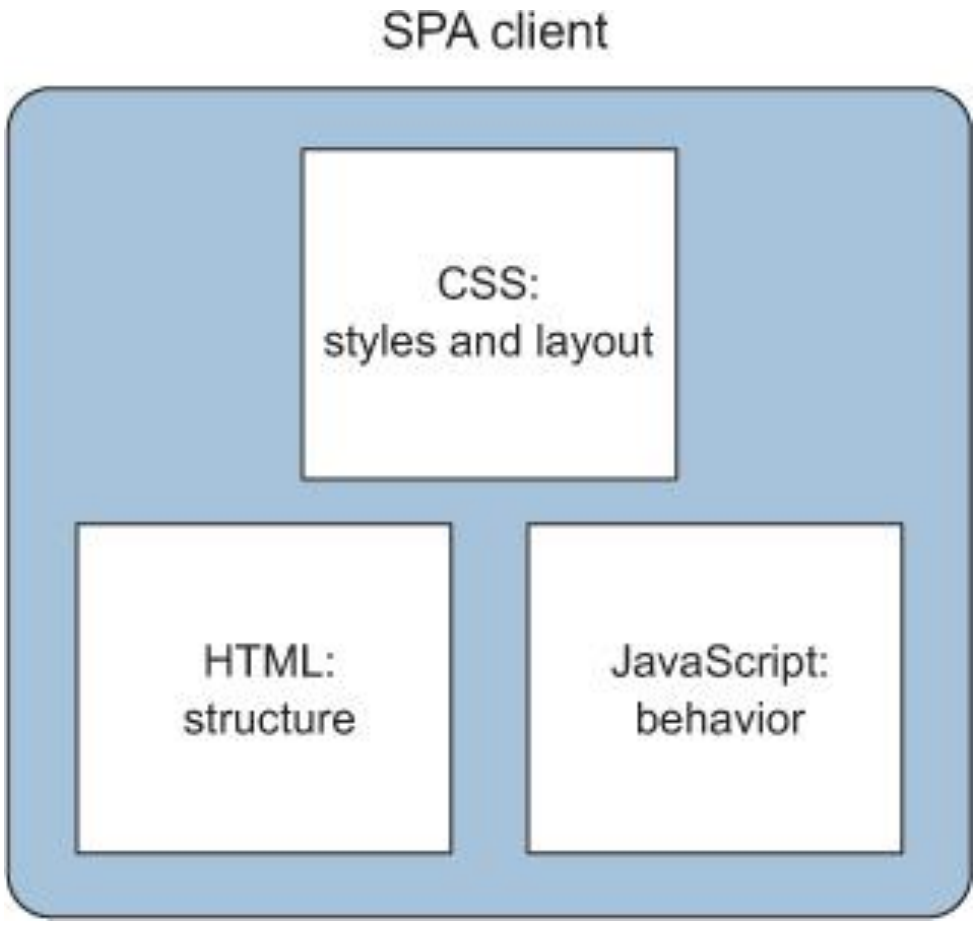
SPA Advantages

innovate achieve lead

Easier code maintenance

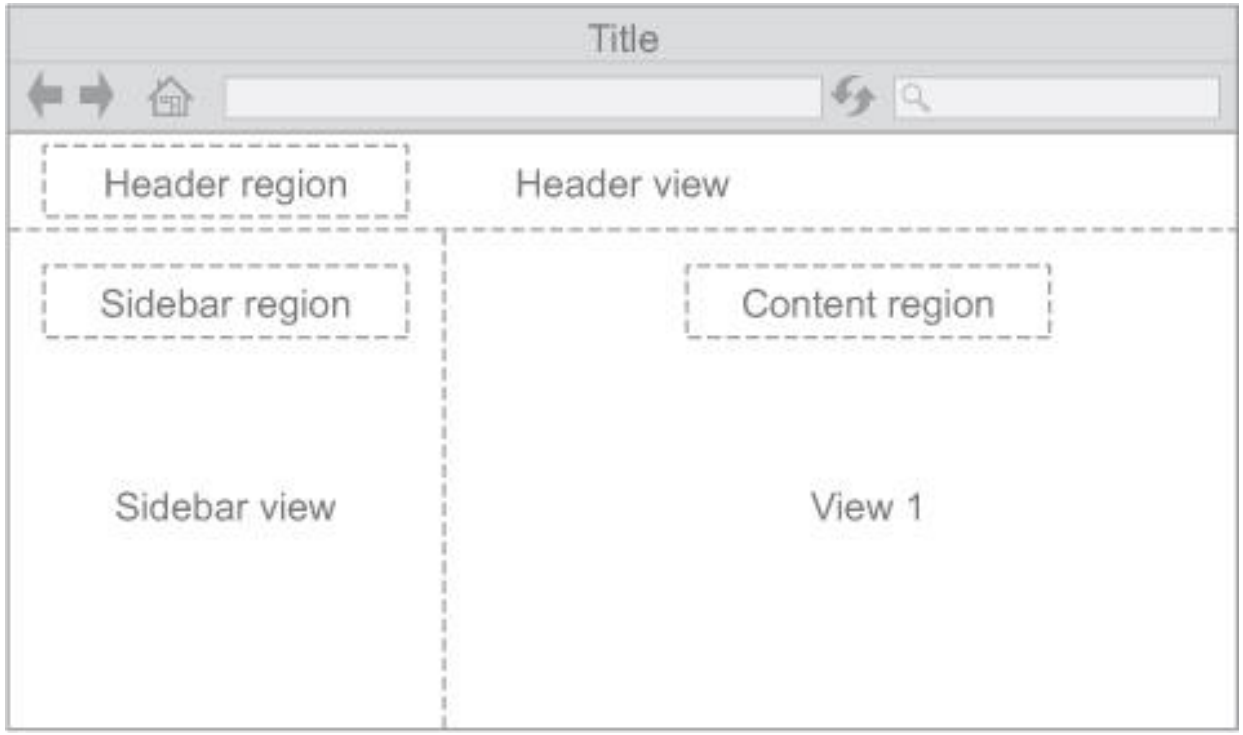
Software developers are always looking for better ways to develop and maintain their code base. Traditionally, web applications are a bit of a Wild West kind of environment, where HTML, JavaScript, and CSS can be intertwined into a maintenance nightmare. Add in the ability to combine server-side code with the HTML source (think Active Server Pages or JavaServer Pages scriptlets) and you've got a giant, steaming pile of goo. As you'll see in upcoming chapters, MV* frameworks like the ones covered in this book help us separate our code into different areas of concern. JavaScript code is kept where it needs to be—out of the HTML and in distinct units. With the help of third-party libraries and frameworks (for example, Knockout, Backbone.js, and AngularJS), the HTML structure for an area of the screen and its data can be maintained separately. The amount of coupling between the client and the server is dramatically reduced as well.

SPA Structure



CSS, HTML, and JavaScript are the building blocks for the single-page application. There's no special language to learn and no browser plugins required.

SPA Views



Subsections of the shell are called regions. A region's content is provided by a view.

SPA Views

Using regions, an SPA's views can be placed so that it looks exactly like a traditional web page.

Question

What types of applications SPA is not suited for?

Answer

Content-heavy Websites

- SPAs may not be well-suited for content-heavy websites such as blogs, news portals, or documentation sites where the primary focus is on displaying large amounts of static content. SPAs typically require the initial loading of JavaScript bundles, which can delay the rendering of content, especially on slower connections.

SEO Requirements

- SPAs can present challenges for search engine optimization (SEO) due to their reliance on JavaScript for content rendering. While search engines have improved their ability to index JavaScript-generated content, traditional server-rendered websites often have an advantage in terms of SEO. For applications where SEO is a critical factor, server-side rendering (SSR) or hybrid approaches may be more appropriate.

Answer

innovate achieve lead

Simple Websites

- For simple websites with minimal interactivity and content, the overhead of building a SPA may outweigh the benefits. In such cases, traditional server-rendered websites or static site generators may offer simpler and more efficient solutions.

E commerce Platforms

Authentication and Security

Complex Data Processing on the Server