CS2104 Problem Solving in Computer Science

Margaret Ellis,
Naren Ramakrishnan,
Sehrish Basir Nizamani

VIRGINIA TECH™

Basics of the Internet

VIRGINIA TECH

Overview

What is Internet?

What are the components of the internet?

Data Centers

What is a protocol?

TCP/IP model and protocol layers

What is the Internet? How does it Work?

Interconnected computer networks that use the Internet Protocol (IP) to communicate

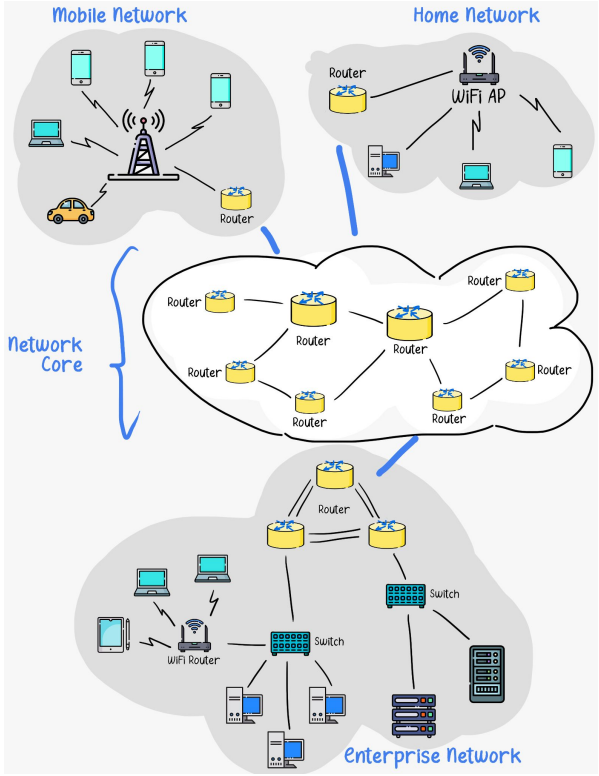


Image source: https://blog.bytebytego.com/p/a-crash-course-in-networking

Internet Connected Edge Devices at Home

Amazon Echo

Internet refrigerator

IP picture frame

Pacemaker & Monitor

Tweet-a-watt: monitor energy use

bikes

cars

scooters

Security Camera

Slingbox: remote control cable TV

Web-enabled toaster + weather forecaster

AR devices

Internet phones

Gaming devices

sensorized, bed mattress

Fitbit

Others?

Data Centers

Amazon Web Services in Ashburn, VA

https://www.vpm.org/2023-10-19/virginia-is-the-worlds-data-center-hub-but-whats-the-cost

Data Centers are used for

Internet traffic

- CDN - Content Delivery Network or Content Distribution Network
- Facilitate high speed data transfer
- Locations around the world
- For example Cloudflare, Akami, Fastly

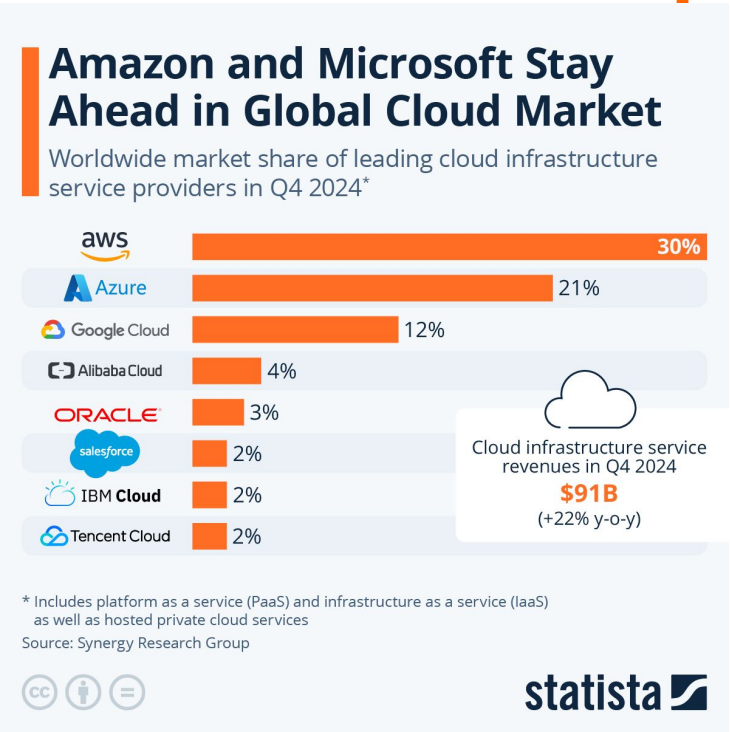
Cloudflare global network

https://www.cloudflare.com/network/

Data Centers

Cloud services

- Hosting Websites
- Storing large data sets
- Heavy Computation
  - Data Analysis
  - Cryptocurrency
  - Large Language Models
- Can be integrated with CDN, like Cloud services from Microsoft, Azure, Google, AWS



Examples of Cloud Services

Here are examples of cloud service models:

1. **Infrastructure as a Service (IaaS):**
   - Users can rent virtualized computing resources(servers, storage, networking) over the internet.
1. **Platform as a Service (PaaS):**
   - Users can develop and deploy applications without worrying about the underlying infrastructure. Various programming languages, databases, and monitoring are provided services.
1. **Software as a Service (SaaS):**
   - Users can access these applications from any device with an internet connection.

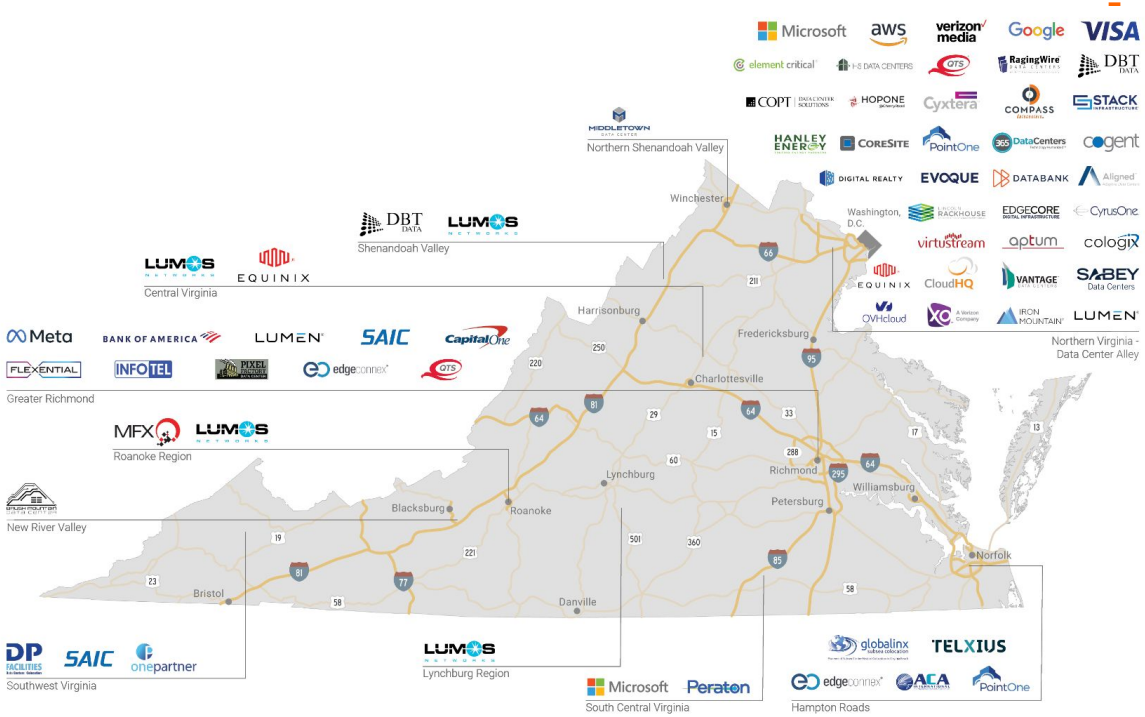
Powering Data Centers

AI is data hungry so need new data centers to support the AI demands:

- Not enough energy to support (talk of opening nuclear reactors to support)
- DeepSeek and other software innovations could improve this
- Concern in VA regarding how to power data centers

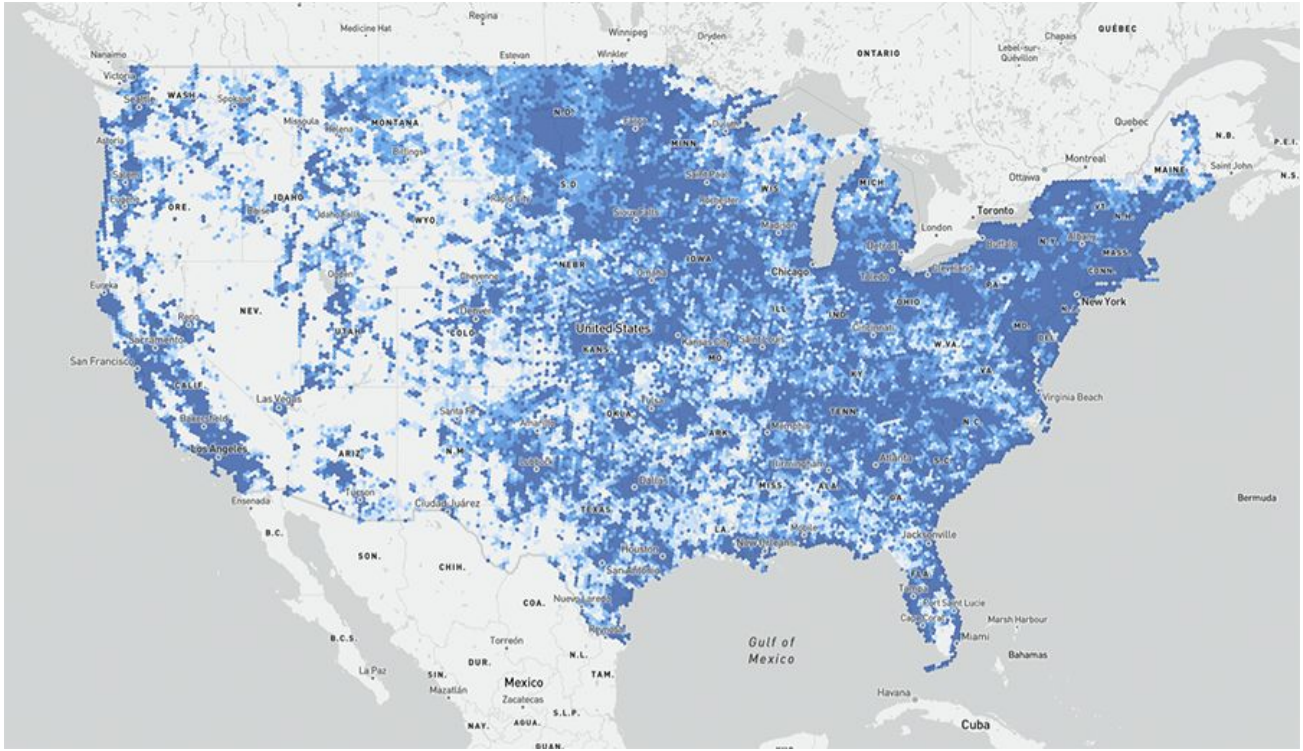
A Map of Data Centers in Virginia
(courtesy of Virginia Economic Development Partnership)
May 2024

Virginia is the data center capital of world (70% of internet traffic) “the cloud is in virginia”



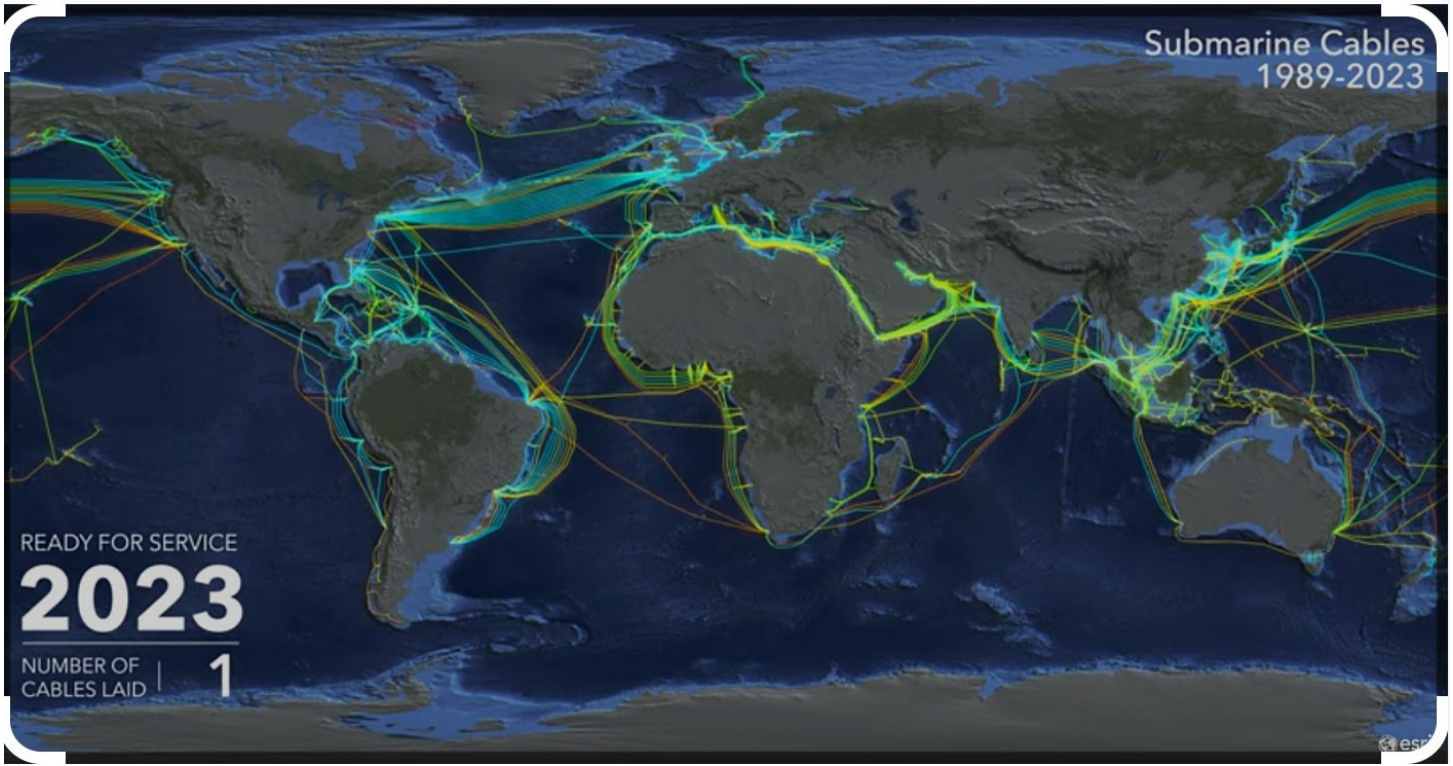
https://virginiamercury.com/2024/05/28/virginia-explained-data-center-expansion-with-all-its-challenges-and-benefits/

National Broadband Map (FCC 2022)



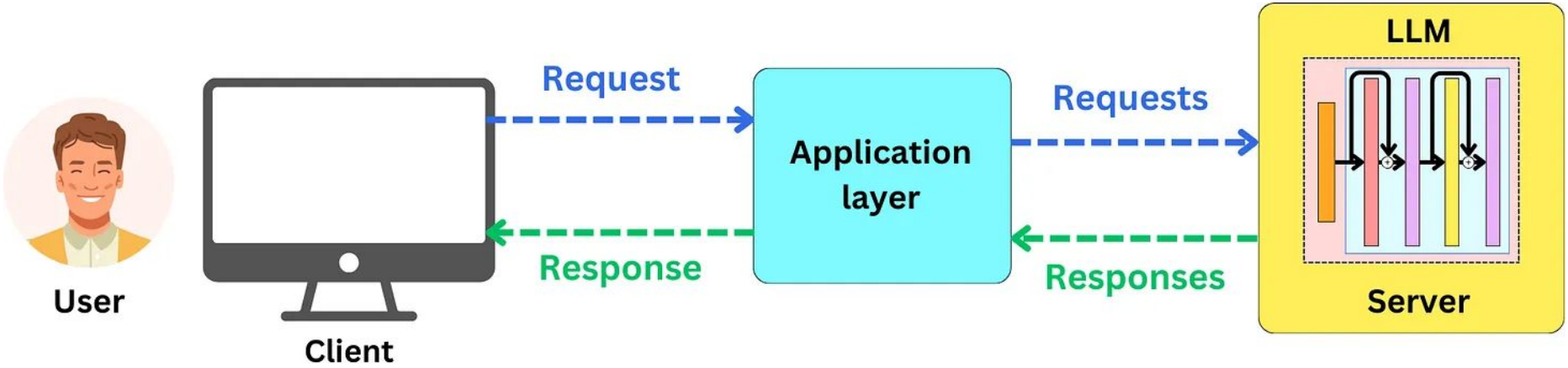
https://broadband.colorado.gov/news-article/fcc-releases-new-national-broadband-map

A Wired World



https://www.visualcapitalist.com/wired-world-35-years-of-submarine-cables-in-one-map/

Networking and LLMs



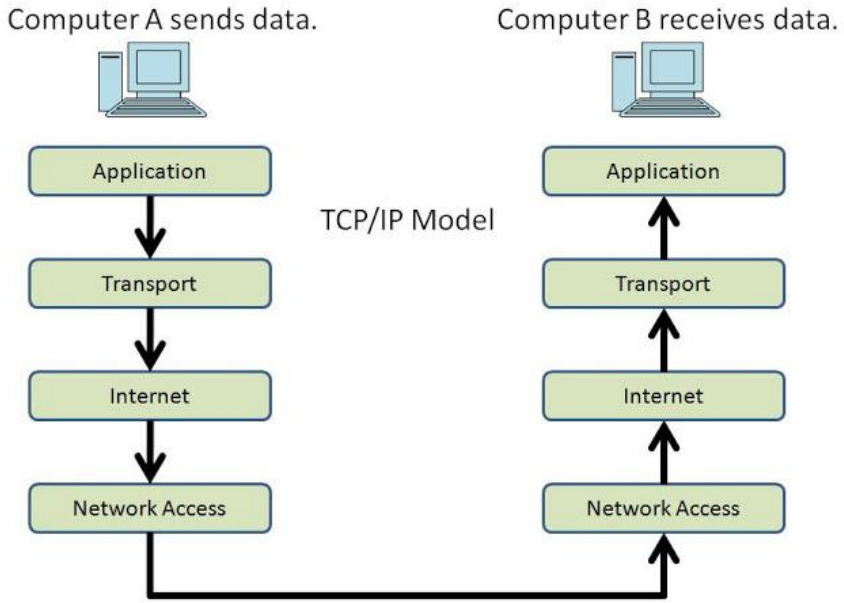
https://newsletter.theaiedge.io/p/the-application-layer-for-llm-applications

Networking and LLMs

- Most LLMs are currently hosted on a server with powerful GPUs and plenty of space. Client machines make a request to these LLMs across a network.
- Active area of research to run LLMs on edge devices such as your phone
  - Reduces carbon footprint
  - Increases privacy
- LLM architecture and operations is an area of interest for many business.

Fundamentals of Networking

- Networking TCP/IP Layers
- (Transmission Control Protocol/Internet Protocol Model)

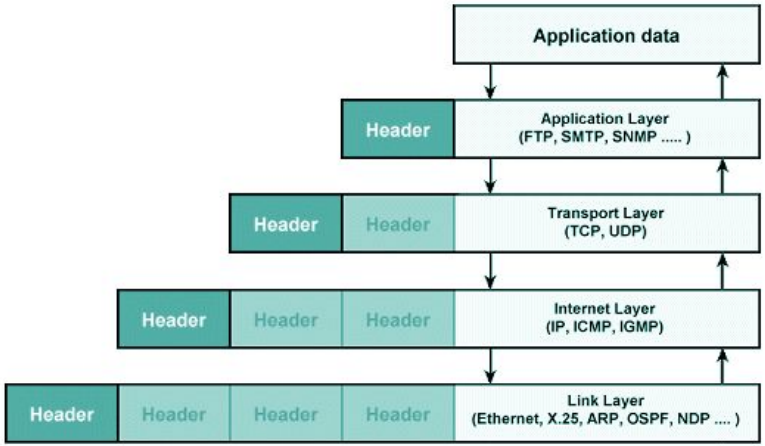


Network Layers and Purpose

Layer	Purpose	Example Protocols	Type of Addressing
Application Layer	Provides network services directly to applications	HTTP, FTP, SMTP, DNS	socket (port + IP address)
Transport Layer	Ensure reliable data transfer, error recovery, and flow control.	TCP(Transmission Control Protocol), UDP (User Datagram Protocol)	Port (identifies specific process or services on a device)
Internet Layer	Determines the best path to move data packets from source to destination.	IP (Internet Protocol)	IP address (unique identifier for routing data packets across networks)
Network Access Layer	Handles the physical transmission of data over network hardware	Ethernet, Wi-Fi, ARP(Address Resolution Protocol)	MAC address (unique identifier for physical network interfaces on device)

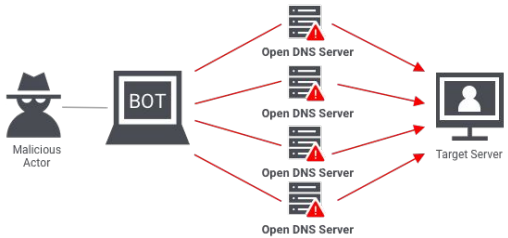
Encapsulation – headers at each layer

- At each layer data is encapsulated with header data on the way down.
- At the far end, the data is decapsulated on the way back up the protocol stack.
- Packets should be encrypted



https://www.technologyuk.net/telecommunications/internet/tcp-ip-stack.shtml

Networking Security Threats



1. **Packet Sniffing:** Attackers intercept data packets to capture sensitive information. Encryption can mitigate this risk.
2. **Man-in-the-Middle Attacks:** Attackers intercept and alter communication between two parties. Secure protocols and encryption help prevent this.
3. **Denial of Service (DoS) Attacks:** Overwhelming a network with traffic to disrupt services. Firewalls and IDS can help detect and mitigate these attacks

Generated with assistance of co-pilot Dec 2024 and resources:
https://www.vpnunlimited.com/help/cybersecurity/tcp-ip

Image sources:
https://www.avast.com/c-packet-sniffing
/https://aboutssl.org/man-in-the-middle-attacks-prevention/
https://www.onelogin.com/learn/ddos-attack

Security Measures

Encryption: Encrypting data packets ensures that even if intercepted, the data remains unreadable without the decryption key

Secure Protocols: Using secure versions of protocols, such as HTTPS instead of HTTP, ensures data integrity and confidentiality.

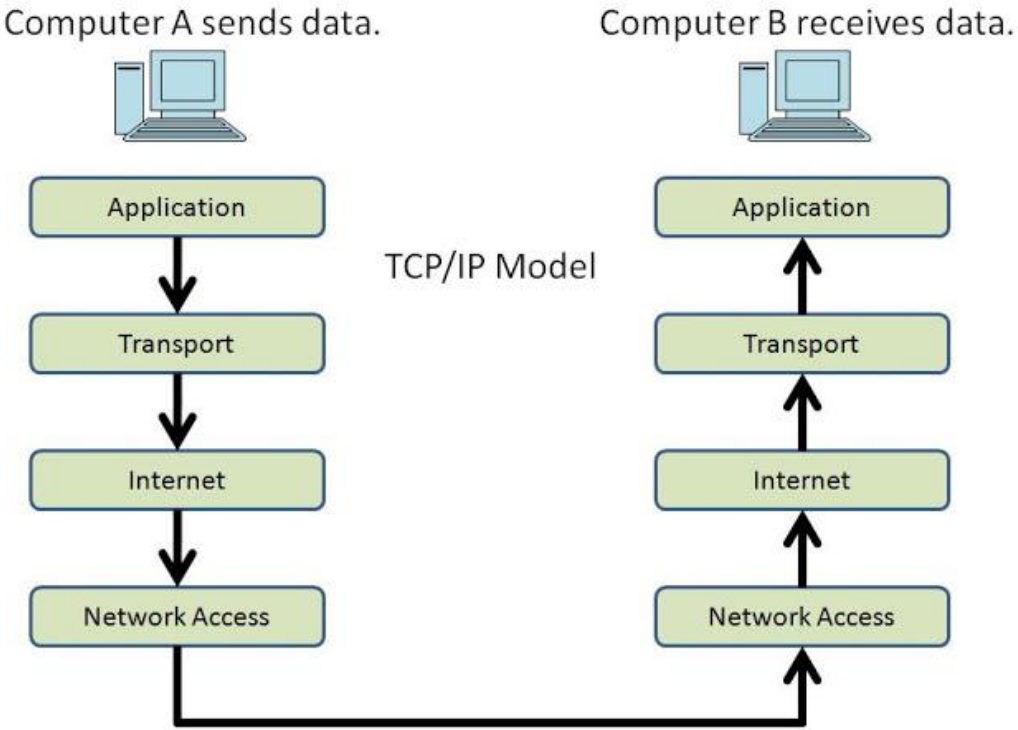
Authentication: Ensuring that only authorized users and devices can access the network helps prevent unauthorized access.

Firewalls: Firewalls monitor and control incoming and outgoing network traffic based on predefined security rules, acting as a barrier between trusted and untrusted networks.

Intrusion Detection Systems (IDS): IDS monitor network traffic for suspicious activities and potential threats.

Generated with assistance of co-pilot Dec 2024 and resources:
https://www.vpnunlimited.com/help/cybersecurity/tcp-ip

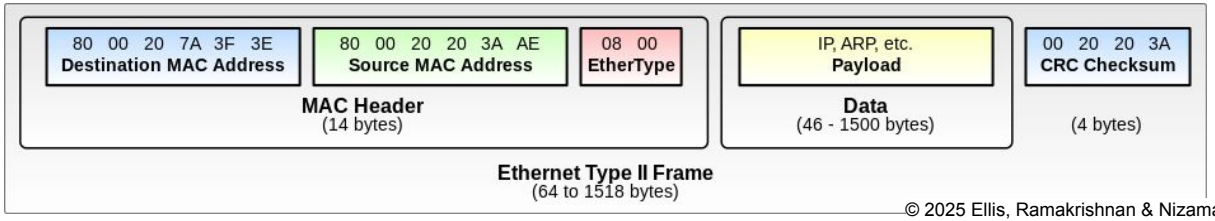
Let's look at the Layers Bottom to Top…



Network Access Layer

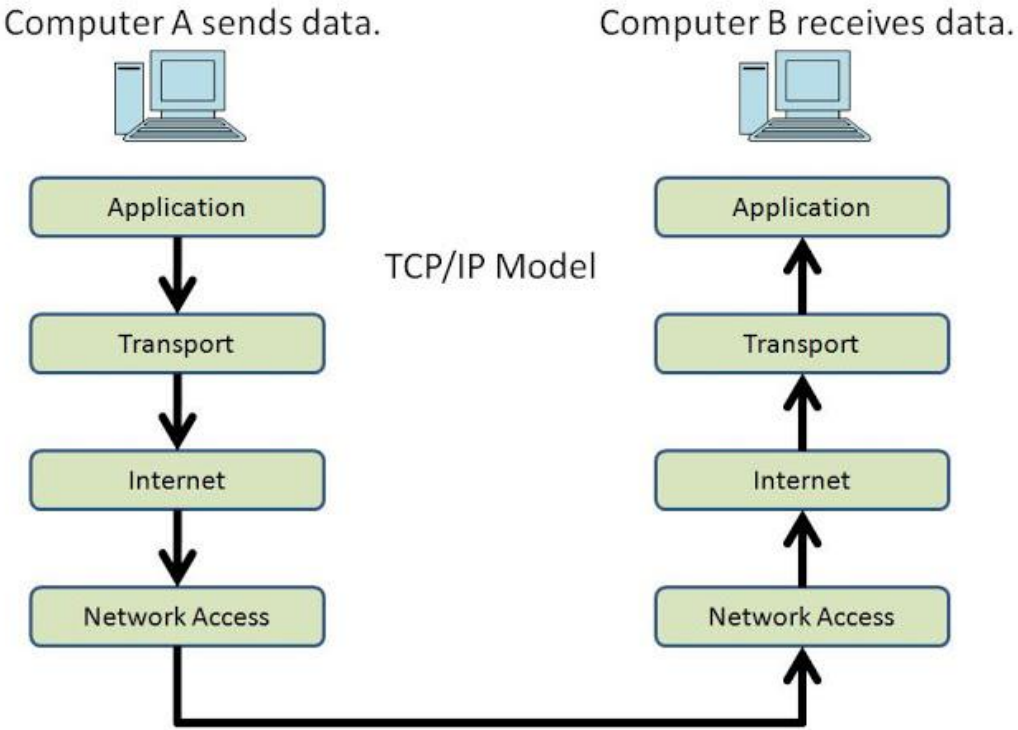
Delivers data between hosts

- send on the path to the right machine as electrical pulses
- Media Access Control(MAC) Address is the identification number that uniquely identifies each device on a network used by WiFi and Ethernet
- to send a packet to the internet the MAC destination will be the MAC address of your router



https://en.wikipedia.org/wiki/Ethernet_frame

Next up is the **Internet Layer…**



Internet Layer – IP Addresses

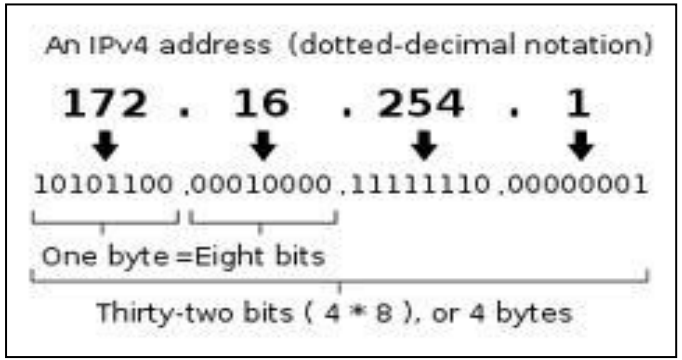
Manage the IP addresses, keeps track of the origin and destination IP addresses

- Select the best path through the network
- IP – Internet Protocol
- ICMP – Internet Control Message Protocol (reverse IP(find host names))
- ARP – Address Resolution Protocol

  (IP -> MAC Address)

https://www.slideshare.net/MIT2012001/tcp-ip-49226613

Addresses on the Internet



- Each node that is (directly) connected to the internet has a unique address, typically (for now with IPv4) represented as a 4-octet number like 198.82.184.178.
- These numbers get more specific, just like mailing addresses.
- Humans occasionally need to know addresses and can't generally remember numbers, which is why we need DNS.

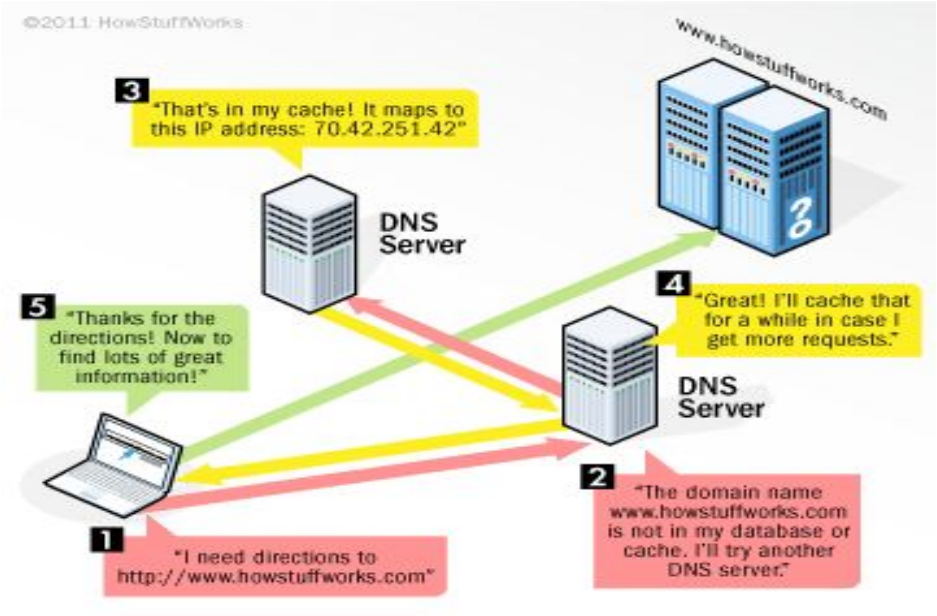
Analogous Problem

Top Down

IP addresses and DNS

Domain Name System – DNS

- DNS Lookup
  - IP(Internet Protocol) address matches domain name.
- Connected in a distributed hierarchy and divided into zones
- DNS spoofing is when DNS server gets hacked and domain name gets matched with wrong IP address



Adapted from Khan Academy:
https://www.khanacademy.org/computing/computer-science

Ping a host

(screenshot from rlogin, windows command prompt, git bash, powershell)

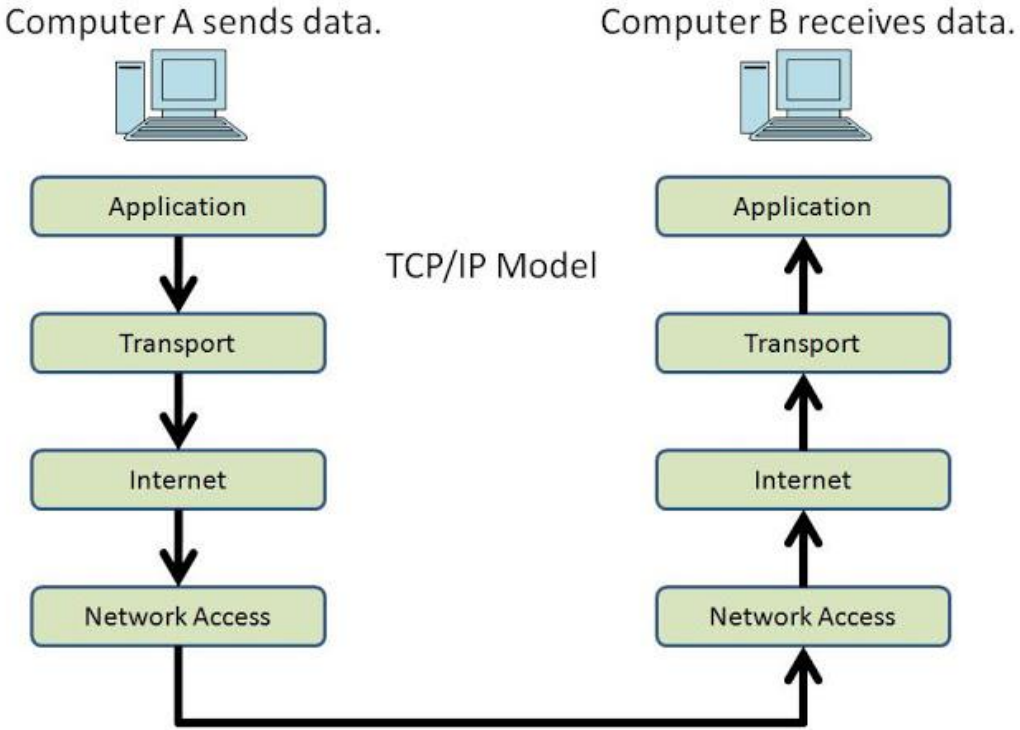
```
[csiwei@holly ~]$ ping -c 4 amazon.com
PING amazon.com (54.239.28.85) 56(84) bytes of data.
64 bytes from 54.239.28.85 (54.239.28.85): icmp_seq=1 ttl=234 time=6.83 ms
64 bytes from 54.239.28.85 (54.239.28.85): icmp_seq=2 ttl=234 time=6.91 ms
64 bytes from 54.239.28.85 (54.239.28.85): icmp_seq=3 ttl=234 time=6.94 ms
64 bytes from 54.239.28.85 (54.239.28.85): icmp_seq=4 ttl=234 time=6.91 ms

--- amazon.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 6.829/6.896/6.940/0.040 ms
```

```
C:\Users\swee>ping amazon.com

Pinging amazon.com [205.251.242.103] with 32 bytes of data:
Reply from 205.251.242.103: bytes=32 time=16ms TTL=235
Reply from 205.251.242.103: bytes=32 time=17ms TTL=235
Reply from 205.251.242.103: bytes=32 time=40ms TTL=235
Reply from 205.251.242.103: bytes=32 time=16ms TTL=235

Ping statistics for 205.251.242.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 16ms, Maximum = 40ms, Average = 22ms
```

```
swee@DESKTOP-CBR9HA5 MINGW64 ~
$ ping amazon.com

Pinging amazon.com [205.251.242.103] with 32 bytes of data:
Reply from 205.251.242.103: bytes=32 time=19ms TTL=235
Reply from 205.251.242.103: bytes=32 time=17ms TTL=235
Reply from 205.251.242.103: bytes=32 time=17ms TTL=235
Reply from 205.251.242.103: bytes=32 time=22ms TTL=235

Ping statistics for 205.251.242.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 17ms, Maximum = 22ms, Average = 18ms
```

```
PS C:\Users\swee> ping amazon.com

Pinging amazon.com [205.251.242.103] with 32 bytes of data:
Reply from 205.251.242.103: bytes=32 time=18ms TTL=235
Reply from 205.251.242.103: bytes=32 time=18ms TTL=235
Reply from 205.251.242.103: bytes=32 time=24ms TTL=235
Reply from 205.251.242.103: bytes=32 time=17ms TTL=235

Ping statistics for 205.251.242.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 17ms, Maximum = 24ms, Average = 19ms
```

Now the **Transport Layer…**



Transport Layer

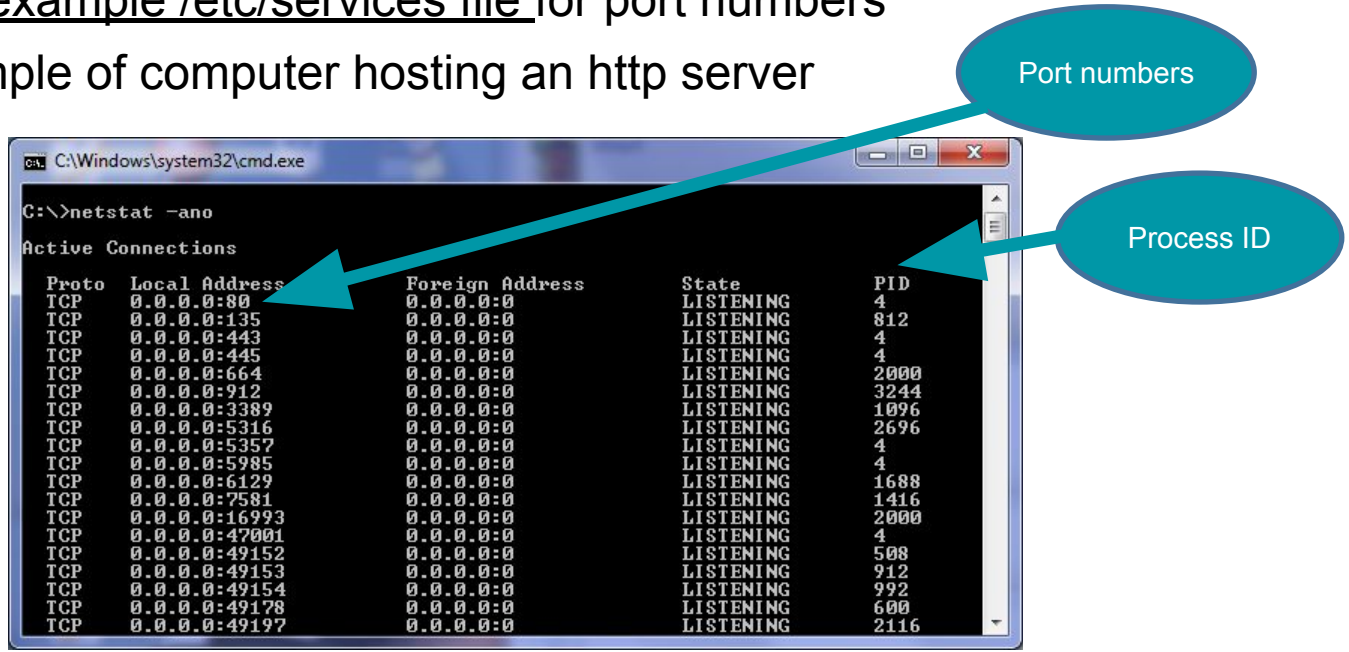
Transport Layer delivers from process to process

there may be many processes running on each machine

- Most common transport protocols are UDP and TCP
  - UDP is faster but not reliable
  - In TCP the transport layer also breaks the data up into packets
    - Header contains how to put it back together
    - Put together in order and do error checking
- Transport layer uses port numbers to identify application processes
  - The Internet Layer handles the I.P. address <-> Street Address
  - The Transport Layer handles the port number <-> Apartment number
  - The Transport Layer is like the mail room in an office building

Port Numbers in Processes

- See example /etc/services file for port numbers
- Example of computer hosting an http server

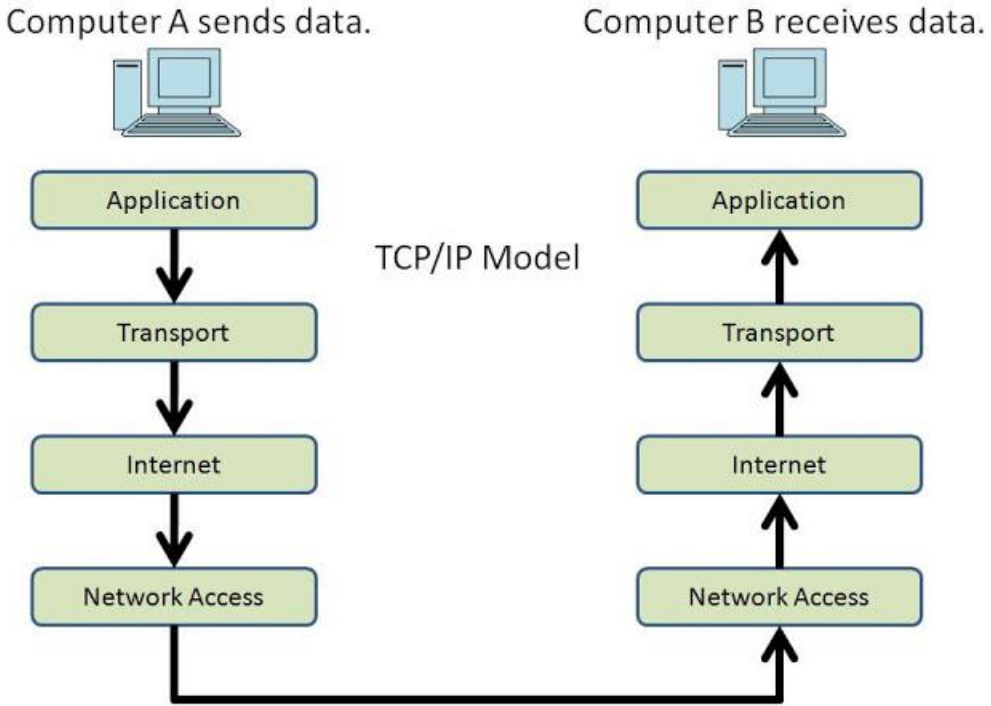


Packets, Routers and Reliability

- Data travels the Internet in packets
- In TCP each packet takes a cheapest route from sender to receiver and gets recompiled at the receiver
  - time, politics, and business relationships
  - once all packets arrive TCP signs and acknowledges receipt
- Routers act as traffic managers
- Multiple paths provide redundancy and therefore reliability

Adapted from Khan Academy:
https://www.khanacademy.org/computing/computer-science

Finally the **Application Layer…**



Application Layer

Application Layer delivers from Application to Application

- Domain Name System(DNS)
  - Resolves domain names to IP addresses
  - Essential to making the other application protocols work
- Hypertext Transfer Protocol (HTTP)
- Simple Mail Transfer Protocol (SMTP)
- Post Office Protocol (POP3) and Internet Message Access Protocol (IMAP4)
- Secure Shell (SSH) [don’t use Telnet…]
- File Transfer Protocol (FTP)
- MySQL client/server protocol

Summary: *What are the basics of the Internet?*

Co-pilot, Dec 2024

1.Network of Networks: The Internet is essentially a global system of interconnected computer networks that use the Internet Protocol (IP) to communicate.

2. **IP Addresses**: Every device connected to the Internet has a unique IP address, which acts like a postal address, allowing data to be sent and received accurately.

3. Domain Names: Instead of using IP addresses, we use domain names (like www.example.com) which are easier to remember. The Domain Name System (DNS) translates these names into IP addresses.

4. Protocols: The Internet relies on a set of rules called protocols to ensure data is transmitted correctly. The most important ones are the Transmission Control Protocol (TCP) and Internet Protocol (IP), often referred to together as TCP/IP.

5. World Wide Web (WWW): Often confused with the Internet, the WWW is a collection of information accessed via the Internet. It uses Hypertext Transfer Protocol (HTTP) to transfer web pages.

6. **Internet Service Providers (ISPs):** These are companies that provide access to the Internet. They connect users to the broader Internet infrastructure.

7. **Routers and Switches:** These devices direct data traffic on the Internet, ensuring it reaches its destination efficiently.

8. **Data Packets:** Information sent over the Internet is broken down into smaller pieces called packets. These packets travel independently and are reassembled at the destination.

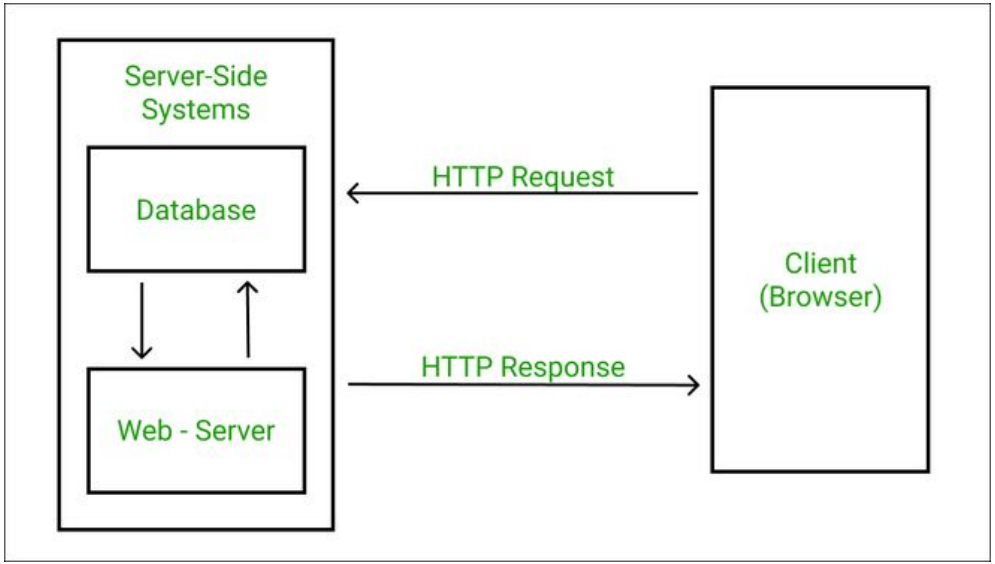
9. **Security:** Various measures like firewalls, encryption, and secure protocols (like HTTPS) are used to protect data and ensure privacy.

10. **Applications:** The Internet supports a wide range of applications, including email, social media, online gaming, streaming services, and

How Websites Work

Web and HTTP

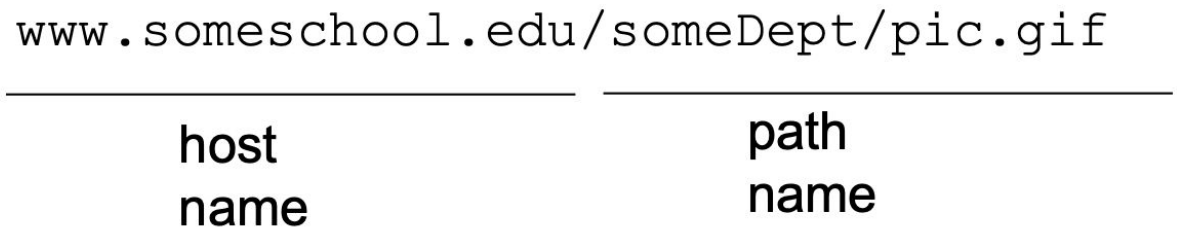
- Web applications involve a client making a request and server replying with a response
  - HTTP is how the client and server communicate
- Simplest case is for a static file



https://www.geeksforgeeks.org/state-the-core-components-of-an-http-response/

Webpages and HTTP

- web page consists of *objects,* each of which can be stored on different Web servers
- object can be HTML file, JPEG image, Java applet, audio file,…
- web page consists of *base HTML-file* which includes *several referenced objects, each* addressable by a *URL,* e.g.,

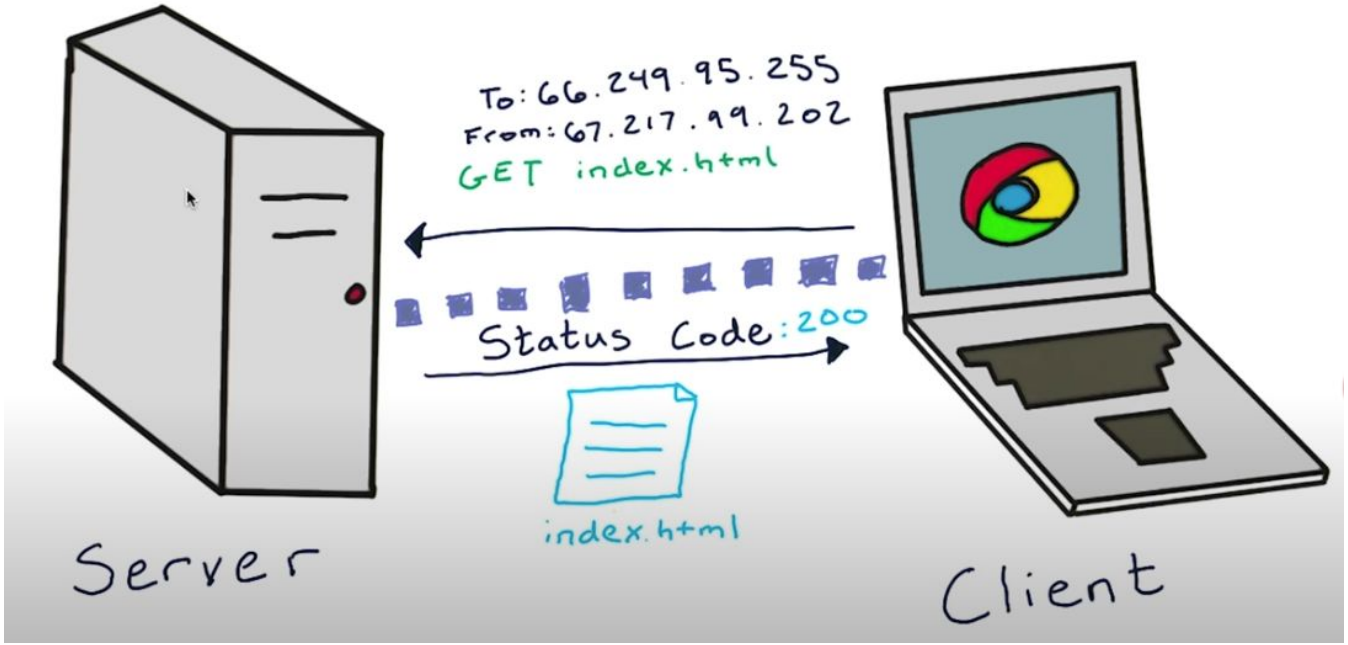


HTTP Methods

Resources can be html code, javascript, a document, a photo, a video, a json file, etc. **Resources** are typically identified by a URL

- GET
  - Requests specified resource
- POST
  - Sends data to the server (often causing a change in state)
- PUT
  - Sends data to the server (idempotent: calling it once or several times has the same affect)
- DELETE
  - Deletes specified resource

Get contents through GUI web browsers



Udacity Full Stack Foundations: HTTP and Response Codes:https://www.youtube.com/watch?v=K7h1PzqNgOs

Bird’s Eye View of Web App

Server-Side

Key Components

- **Model**
  - Handles data management.
- **View**
  - Manages client-side display.
- **Controller**
  - Uses frameworks, libraries, templating engines, server-side scripts, and configuration files to control application logic.

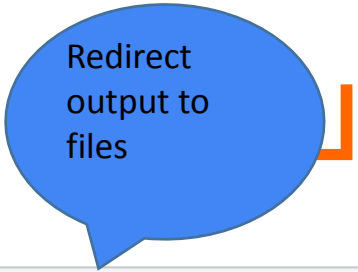
Client-Side

Key Components

- **Content**
  - Formats include HTML, XHTML, XML, JSON.
- **Presentation**
  - Managed using CSS.
- **Behavior**
  - Implemented with JavaScript frameworks such as React.js, Angular.js, and jQuery.

Get CONTENTS via Command Line

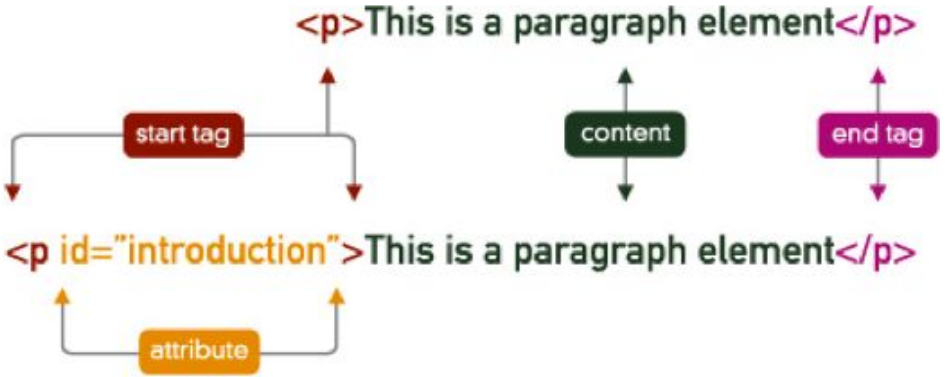
Remember this from setting up vscode?



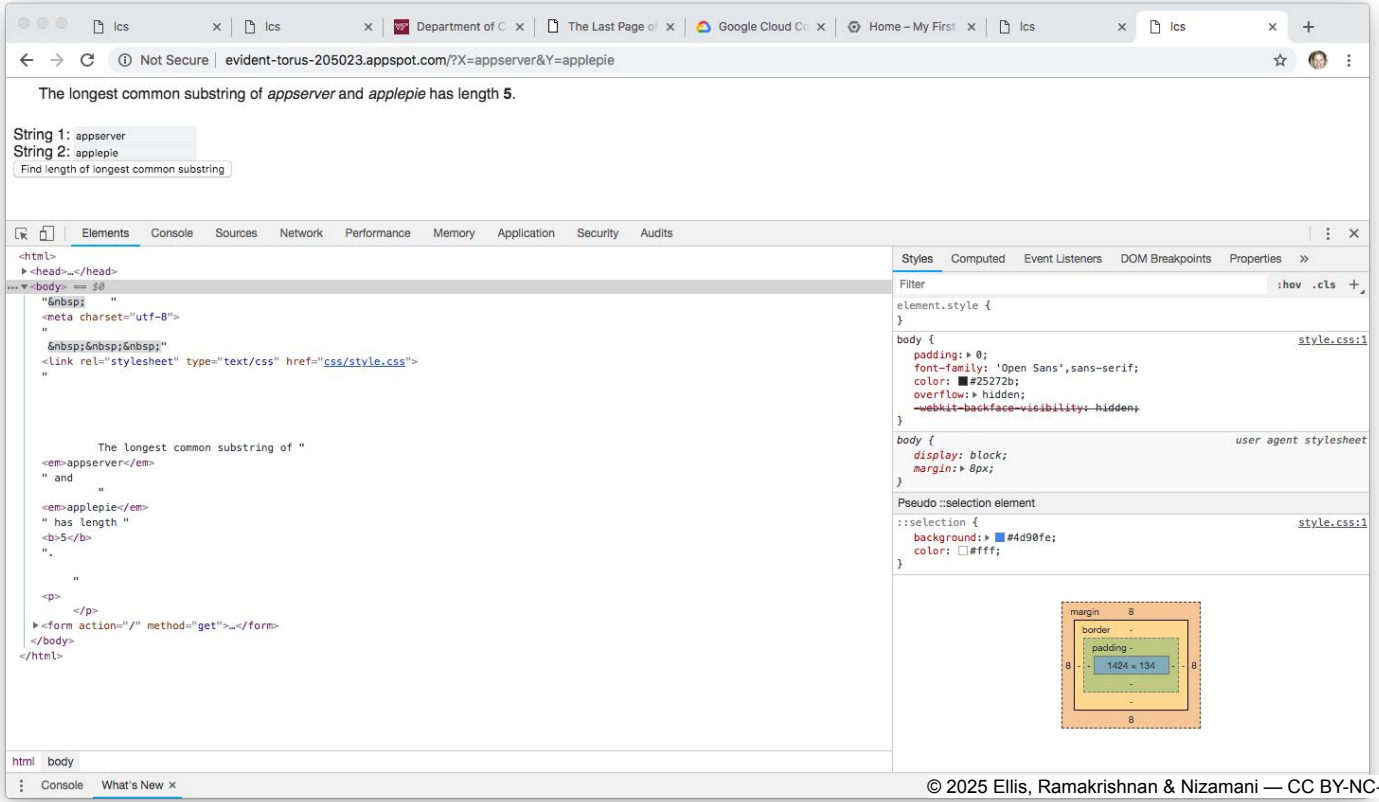

```
$ curl https://git.cs.vt.edu/cs2104-staff/requirements-file/-/raw/main/requirements.txt > requirements.txt
$ pip install -r requirements.txt
```

HTML – rendered client side

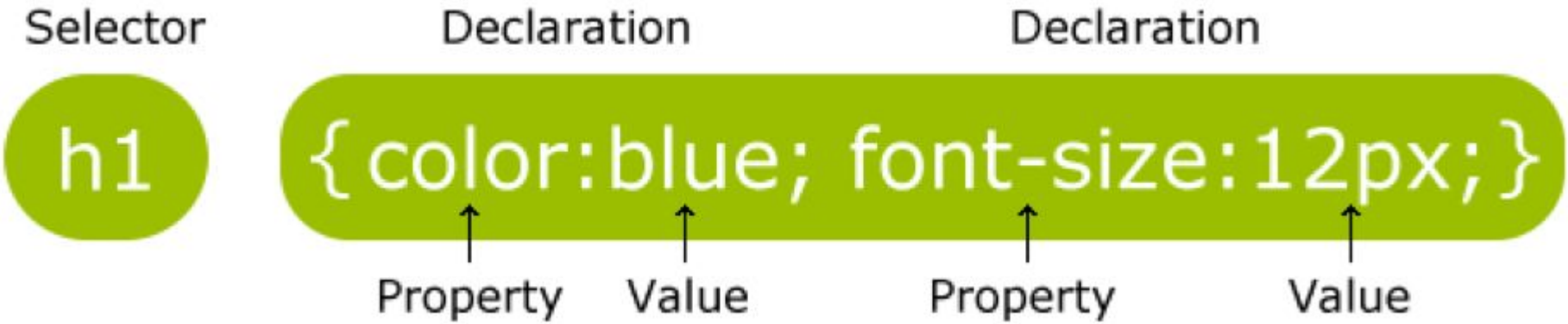
Common
– Paragraph (<p>)
– Headings (<h1>, <h2>, <h3>)
– List (<ul> and <ol>, <li>)
– Anchor <a>



Client Side HTML

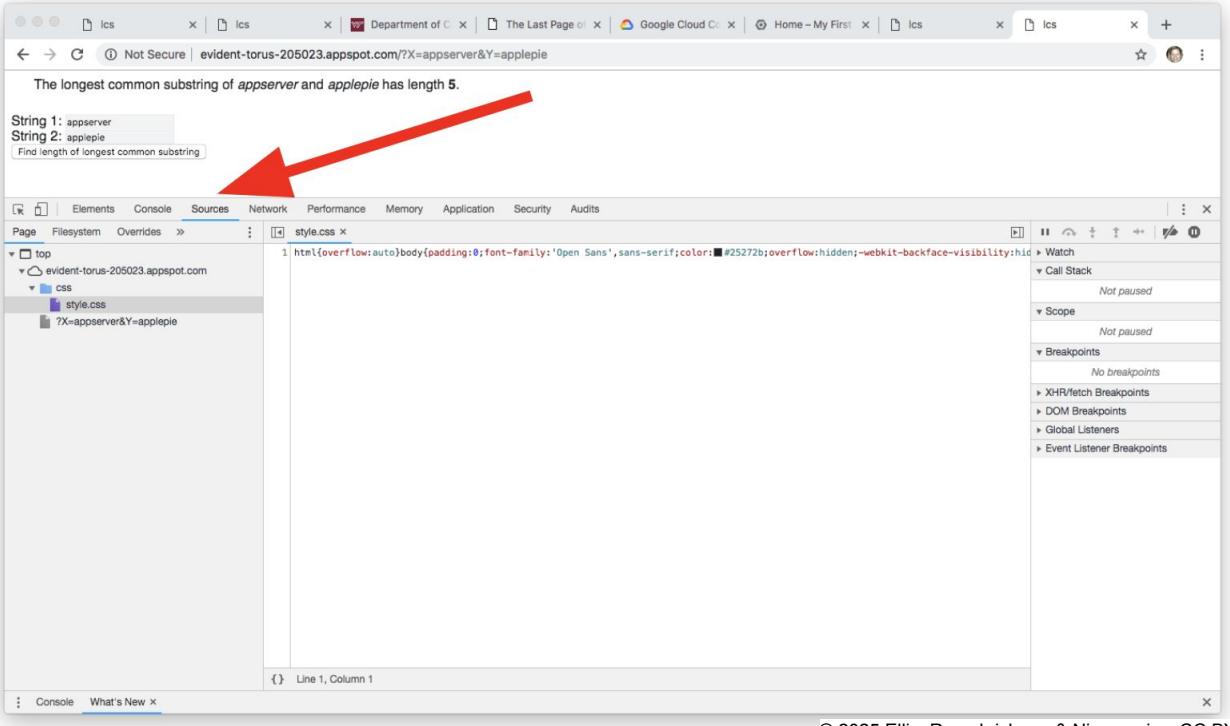


CSS – provides style client side



- Simple: Eric Meyer's CSS reset
  - http://meyerweb.com/eric/tools/css/reset/
- Fancy: normalize.css
  - http://necolas.github.io/normalize.css/
- Plenty of others

Client-side CSS file



Dynamic Content

- Web browsers
  - HTML5, CSS and Javascript allow web pages to be responsive to user actions
  - Frameworks on top of those make this easier (e.g. angular, react)
  - The data sent to the client (browser) may still be retrieved from static files on the server's file system
- Server-side is programmable
  - Resource may be dynamically computed or pulled from a database
  - Many language and frameworks (e.g. python, node.js)
  - Some frameworks (next.js) aim to provide dynamic server-side and client-side functionality

Sample Web App Architecture

Users

Web Browser (or curl, or some other client)

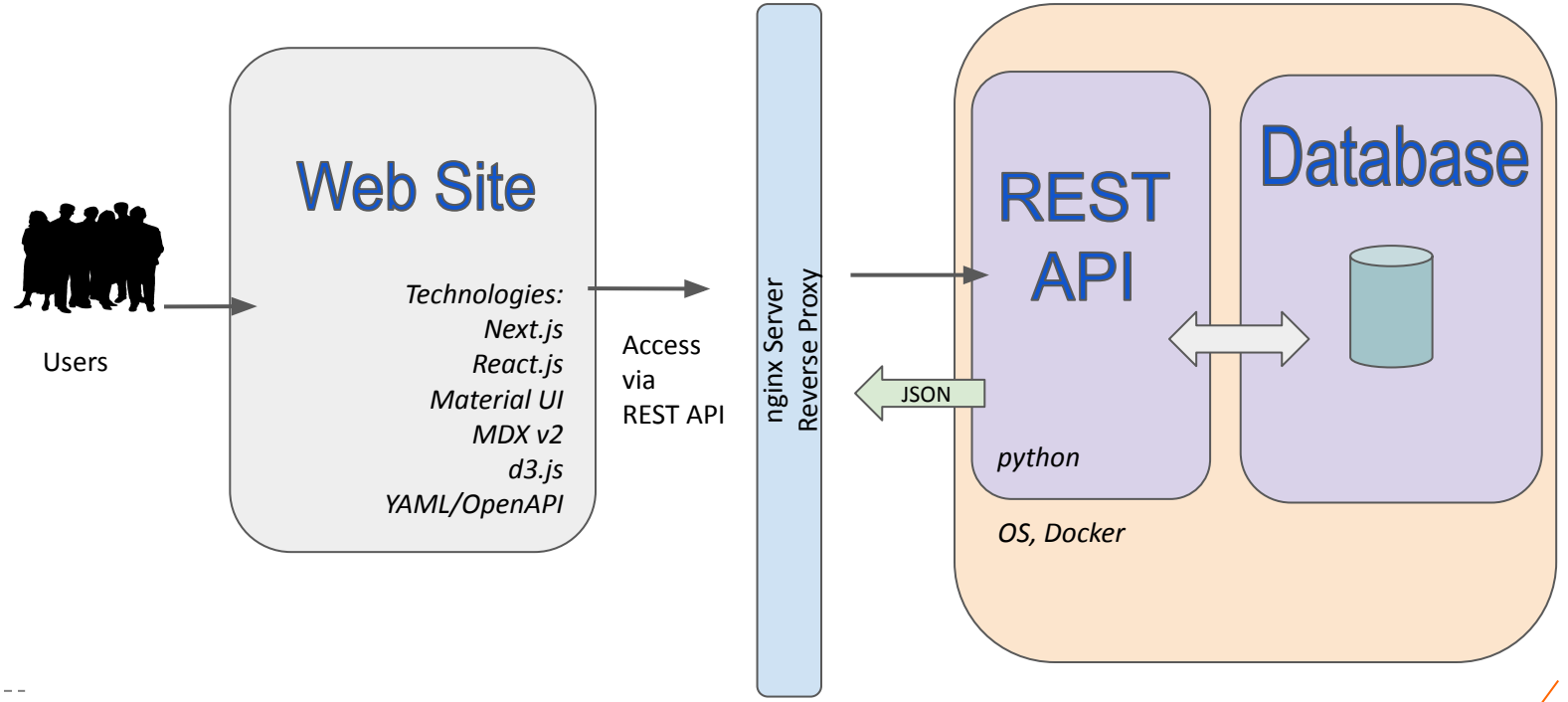
Access via HTTP GET

response

Web Server

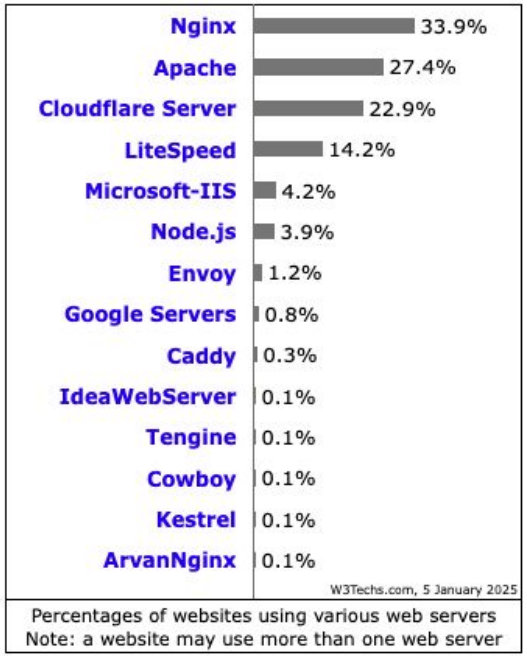
html

Sample Web App Architecture



Overarching Technologies

Server-side, web servers



Percentages of websites using various web servers
Note: a website may use more than one web server

Client-side, web browsers

Browser	StatCounter
Chrome	62.38%
Safari	17.9%
Edge	4.92%
Firefox	2.46%
Samsung Internet	2.23%
Opera	2.01%
UC Browser	0.92%
Android	0.51%

https://w3techs.com/technologies/overview/web_server/all

http://gs.statcounter.com/browser-market-share

Cloud Services and Web Servers

Web Servers(Apache, nginx, IIS...)

- Accept and parse HTTP requests
- Invoke backend application logic to perform computations and retrieve data
- Send results back to the client

Cloud services (GCP, AWS, Azure…)

provide a way to deploy applications at scale, on demand, and without your having to manage a web server yourself

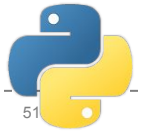
Common Server-Side technologies

Web Frameworks: provide a skeleton for server-side development, providing libraries and tools:

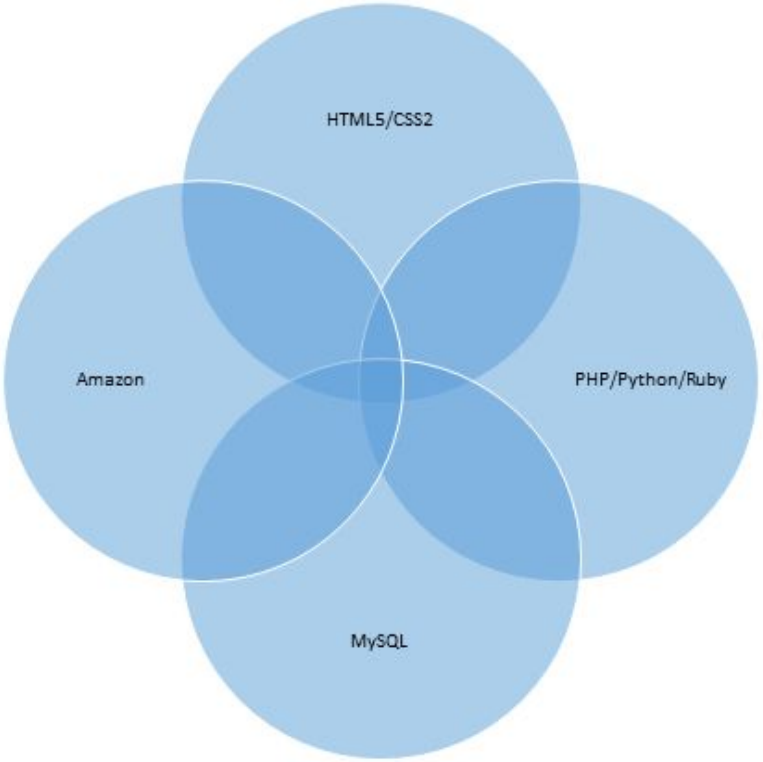
- Express(Node.js) and Next.js (React + Node.js) are full stack
- Django(Python), Rails(Ruby), Node.js are backend

Programming Languages: PHP, Ruby, Python, Java, Node.js

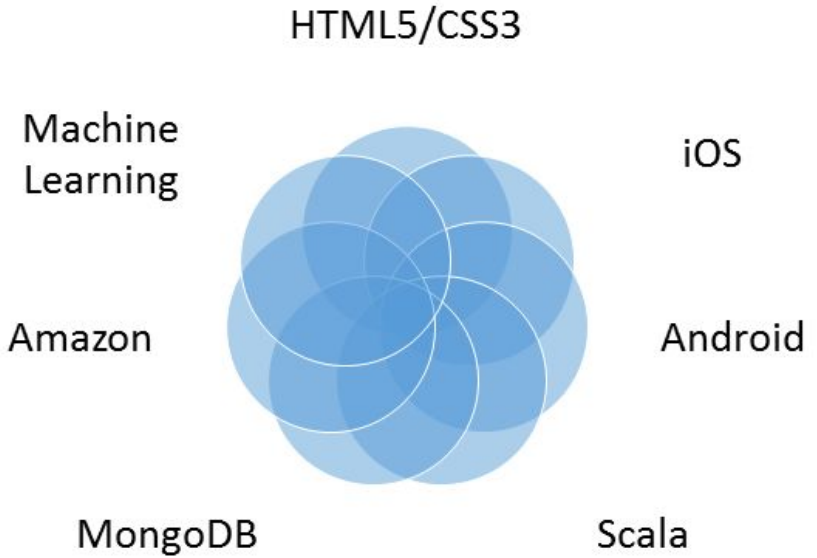
Data Formats: jinja2, YAML, XML , JSON, MySQL



Becoming a full stack web developer
Source: TechCrunch (Peter Yared)



Just one example of possible technology combinations in web development.



HTTP Methods revisit

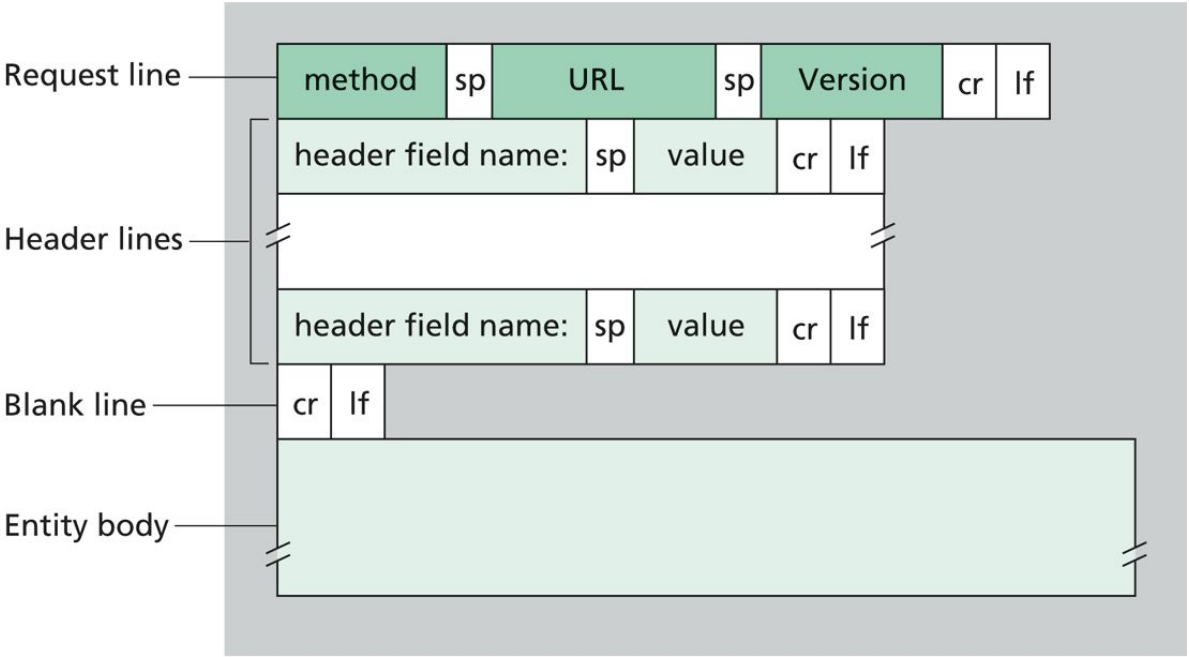
Resources can be html code, javascript, a document, a photo, a video, a json file, etc. **Resources** are typically identified by a URL

- GET
  - Requests specified resource
- POST
  - Sends data to the server (often causing a change in state)
- PUT
  - Sends data to the server (idempotent: calling it once or several times has the same affect)
- DELETE
  - Deletes specified resource

HTTP response status codes

- status code appears in 1st line in server-to-client response message.
- some sample codes: with some fun: https://http.cat/
  - 200 OK
    - request succeeded, requested object later in this message
  - 301 Moved Permanently
    - requested object moved, new location specified later in this message (in Location: field)
  - 400 Bad Request
    - request msg not understood by server
  - 404 Not Found
    - requested document not found on this server
  - 505 HTTP Version Not Supported

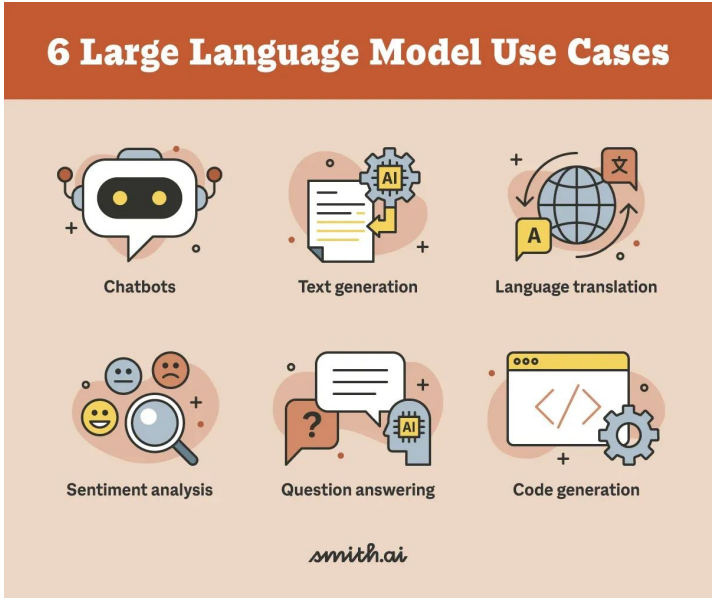
HTTP Request Message General Format



https://mscancer22.tripod.com/applicationlayer/id19.html

Examples of LLM usage by Websites

- Chatbots and Virtual Assistants
- Assist with Software Development
- Assist with Content Generation (including for Search Engine Optimization SEO)
- Language Translation
- Accessibility (e.g. text-to-speech, speech-to-text)



https://altois.com/blog/understanding-large-language-models-llms-2025-a-comprehensive-guide/
https://botpenguin.com/blogs/llm-use-cases

Assignments

- **Next class: class to create a webpage that you will deploy next week**
- **More on networking and web development next week**
- **HTML-related HW 9 for next week….**

Hypertext Markup Language (HTML)

- Lingua franca (common language) of websites
- HTML + CSS helps factorize
  - Content (Textual + Graphic) of a Website
  - Styling of a Website
- HTML Syntax consists of tags (<tag>, </tag>)
  - <head>, <title>, <body>, <h1>, <p>

Basic HTML Elements

- Bold Text Elements
  a. `<b>` creates bold text without semantic meaning[1]
  b. `<strong>` creates bold text with emphasis on importance[1]
- Basic Text Structure
  a. `<p>` defines a paragraph block
  b. `<br>` creates a line break within text
  c. `<i>` creates italic text for visual distinction
- Container Elements
  a. `<div>` acts as a container to group content and create distinct sections
  b. `<span>` allows inline styling of specific text portions
- List Structures
  a. `<ul>` creates an unordered list with bullets
  b. `<ol>` creates a numbered list
  c. `<li>` defines individual list items within either list type

Consider Tables in HTML

- <table> defines the table
- <th> </th> defines the table headers (column headers)
- <tr> </tr> defines each rows
- <td> </td> defines individual cells

```
<table>
    <thead>
        <tr>
            <th>Header 1</th>
            <th>Header 2</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Data 1</td>
            <td>Data 2</td>
        </tr>
    </tbody>
</table>
```

Introduction to CSS

- CSS separates content from presentation, allowing developers to control the appearance of HTML elements including layout, colors, fonts, and spacing.
  - Think of HTML as the skeleton of a webpage, while CSS acts as its clothing and appearance .
- Simple CSS example

```
body {
    background-color: lightblue;
}
p {
    color: blue;
    font-size: 20px;
}
```

Linking stylesheets to HTML

Connecting CSS to HTML

```
<head>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
```

You still have to link up the entries in the stylesheet to elements in the HTML!

- Three ways to do so!
  - Standard element selector
  - Class selector
  - ID selector

Standard element selector

- Just use the standard HTML tags and specify how they should be styled

```
p {
    color: blue;
    font-size: 16px;
}
```

- The above CSS spec will apply the style to all paragraph (p) tags

Class selector

What if you want the same element to be styled differently in different places!

-
```
.main-paragraph {
    color: red;
    margin: 20px;
}
```

- Then use this class in your HTML tag!

```
<p class="main-paragraph">This paragraph uses the class styles</p>
```

ID selector

- This provides even more fine-grained control

```
#unique-element {
    background-color: yellow;
    padding: 10px;
}
```

- Then mention this ID in your HTML tag!

```
<div id="unique-element">This uses ID-specific styles</div>
```

Basic HTML table

Prompt: “Can you write a html code for me that displays grey color for **even rows and white for odd rows**?”

#	Name	Age
1	Alice	25
2	Bob	30
3	Charlie	35
4	Diana	28

Basic HTML table - adding up constraint

Prompt: “can you do **every third row** to be a different color?”

```
    th {
        background-color: #f4f4f4;
    }
    tr:nth-child(3n) {
        background-color: lightblue; /* Every third row */
    }
    tr:nth-child(odd):not(:nth-child(3n)) {
        background-color: white; /* Odd rows excluding every third */
    }
    tr:nth-child(even):not(:nth-child(3n)) {
        background-color: grey; /* Even rows excluding every third */
    }
</style>
```

#	Name	Age
1	Alice	25
2	Bob	30
3	Charlie	35
4	Diana	28
5	Edward	40
6	F	

Your HW9 for next week will be

- Style your table to the following specification, you are allowed to use and LLM

"Create a table with the following features: every third row has a unique background color, other rows alternate between white and light gray, the header row is sticky, and on hover, the entire row smoothly changes its background color to light blue? Additionally, apply a bold font style to the third column of each row."

First some HTML practice

Today in class you will create a webpage using HTML.

You can create this on rlogin and use an extension to view it (demo).