

THE INTERNET EXPLAINED



WHAT WE'LL LEARN

HISTORY AND CURRENT STRUCTURE OF INTERNET

CLIENT-SERVER ARCHITECTURE

HTTP PROTOCOL

REST APIS



BRIEF HISTORY OF INTERNET

The history of the technology goes back to the first part of the 20th century. Nikola Tesla prototyped an early wireless communication system and although its success was limited it provided inspiration and direction in the field of connected communications. Later on scientist developed packet switching. Packet switching is a protocol for communication between multiple computers across a single network.

The first real prototype of the Internet was developed in the late 1960s by the US Department of Defense, with the creation of ARPAnet. In 1969, on October 29, ARPAnet delivered its first message between two computers.

In the 1970s scientists Robert Kahn and Vinton Cerf developed Transmission Control Protocol and Internet Protocol, or TCP/IP, a communications model that set the standards for data transmission across multiple networks. At the beginning of 1983, ARPAnet adopted TCP/IP and researchers began to assemble the “network of networks” that became the modern Internet. In 1990 computer scientist Tim Berners-Lee, whilst working at the Cern Laboratories in Switzerland, created the www (World Wide Web). While it’s often confused with the internet itself, the web is actually just the most common means of accessing data online in the form of websites and hyperlinks.



BRIEF HISTORY OF INTERNET: EVOLUTION OF THE WWW

Initially, the WWW (World Wide Web) was based around simple messaging communications. It was Tim Berners-Lee who had developed a prototype Hypertext system that gave the idea of a markup language to link various documents across computers.

Bill Atkinson, a programmer working for Apple Computer Inc, created an application called Hypercard for the Mac. This along with Tim Berners-Lee's work and various other contributors eventually lead to the first version of HTML (Hypertext Markup Language).

HTML is what enables Web Designers to create the graphical interfaces we see on web sites. HTML has developed into a powerful markup language and is now on version 5, progressing from a few simple markup tags to a world of interactive markup and interaction with programming languages (i.e. Javascript, developed by Netscape in the mid 1990's) and CSS (Cascading Style Sheets). Both of which have also developed over a number of iterations.

Javascript is the most commonly used programming language for Client side web development (The graphical user interface that we interact with). These days Javascript is used both for client side development and server side via node.js.



BRIEF HISTORY OF INTERNET: EVOLUTION OF THE WWW

CSS enables Web Designers to stylise in terms of position, composition the visual effects of HTML. The first draft proposal of CSS was published by Håkon Wium Lie in 1994, a few days before Netscape had released Netscape Navigator a successor to Mosaic, pretty much the first Internet browser of any note.

Today, almost every new application we use is either web based (accessible over the internet) or on a mobile device. Both use HTML, Javascript and various frameworks on top of Javascript and of course CSS.



WHAT DOES IT MEAN FOR YOU AND I ?

Think of the Internet as a big spiders web, and each connection of the web (Connected Thread) there is a computer. That computer may belong to you or I or it may belong to an organisation that provides information or a service, such as music or video streaming. Think Google, Spotify and Netflix. These are prime examples of information/service providers. Requests and Responses (more about those later) are what compose the interaction between computers on the web.

There is rarely a direct path between your computer and a service, i.e. the starting point and the destination, and always more than one route to get to that service. Along the way the request and later the response will make several hops over various computers to end up at the correct destination.



IN SIMPLE TERMS

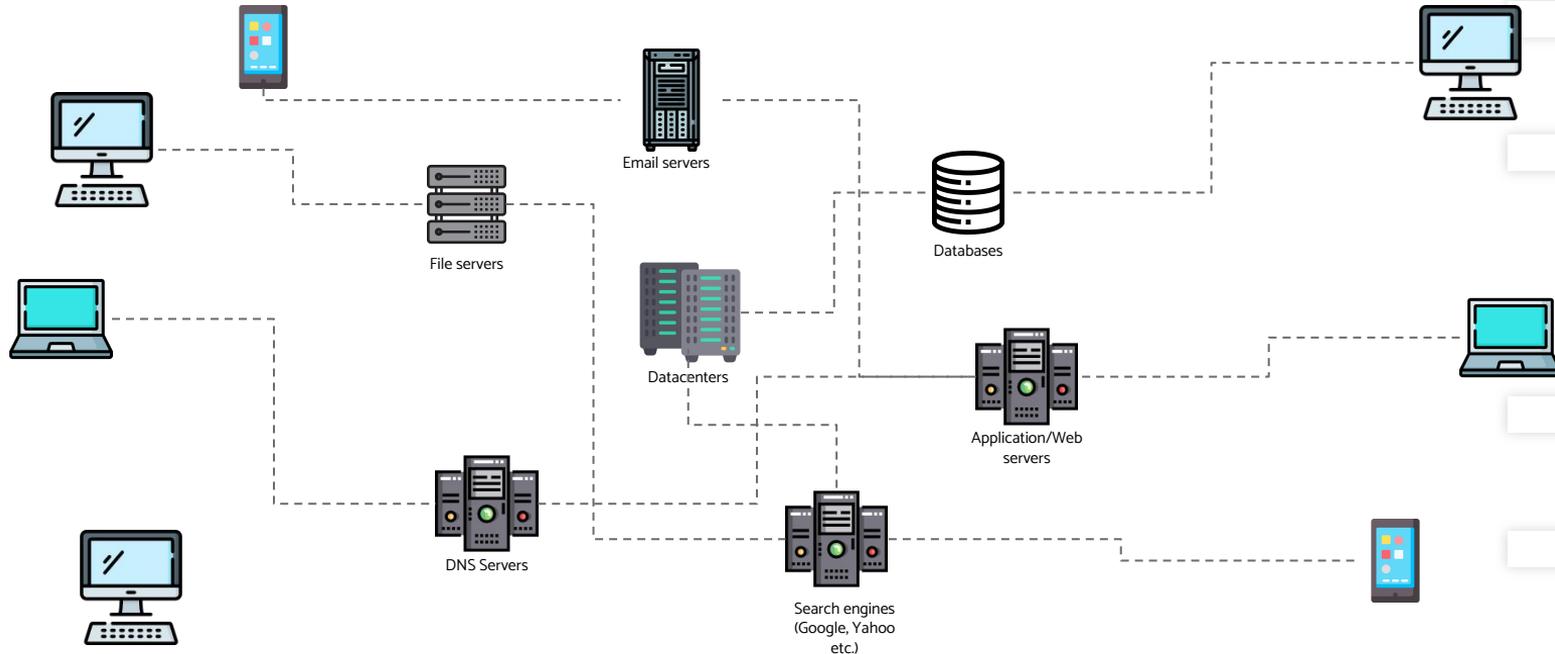
When you or I want to access information or a service we do so via our web browser or smart device. When we load a webpage we go through a number of simple steps.

1. We point our browser at the webpage we require by putting in the address, i.e. fathat.org...
2. Our browsers take care of sending out the address to an intermediary computer on another part of the web, commonly known as a DNS (Domain Name Server). This server translates the address into an IP address of the computer holding the information or providing the service we require. It then redirects the request to that computer along with your computer's IP address. It does this so that the computer receiving the request service knows where to send the response.
3. On receipt of your request that computer verifies that your request is valid and if so starts sending information back to your computer across various links in the web. If the request is not valid (perhaps, you haven't signed in yet), it will also send back details to let you know it is not valid.
4. Once the requesting computer receives the information it sends back an acknowledgment to let the computer know that it has received the information. This process may continue back and forth until all the information you have requested is received.



STRUCTURE OF TODAY'S INTERNET: ALL COMPUTERS VIRTUALLY CONNECTED

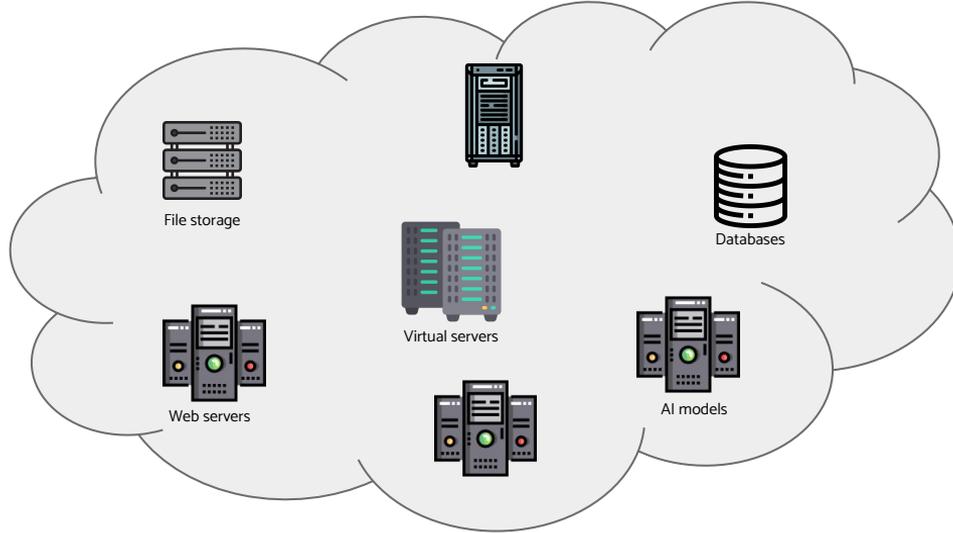
The Internet today comprises almost all the computing devices (desktops, laptops, mobile phones, servers, data centers) across the world, virtually connected through ISPs, routers, gateways.



CLOUD: SOFTWARE AND SERVICES THAT RUN ON THE INTERNET



As developer, you write your code and deploy it in the cloud. The cloud takes care of serving your code to users.



Cloud providers: Google, Amazon, Microsoft, Oracle, Superonline, BlueOcean, ...



User access services developed through the cloud

Cloud allow you to distribute the services at large scale to millions of users.

CLIENT-SERVER: COMMUNICATION EXAMPLES



Customer
(Client)

Request: Do you have vegetarian food?



Response: Yes, we have spinach, ...



Waiter
(Server)



Customer
(Client)

Request: Can you bring the menu please.



Response: Yes, here you are.



Waiter
(Server)



CLIENT-SERVER: COMMUNICATION EXAMPLES

- 1) Waiter serves to customer.
- 2) Waiter asks cook to prepare food.



Request: I want to order a pizza.



Response: Here is your pizza.



Customer
(Client)



Request: Please prepare a pizza.



Response: Here is the pizza.

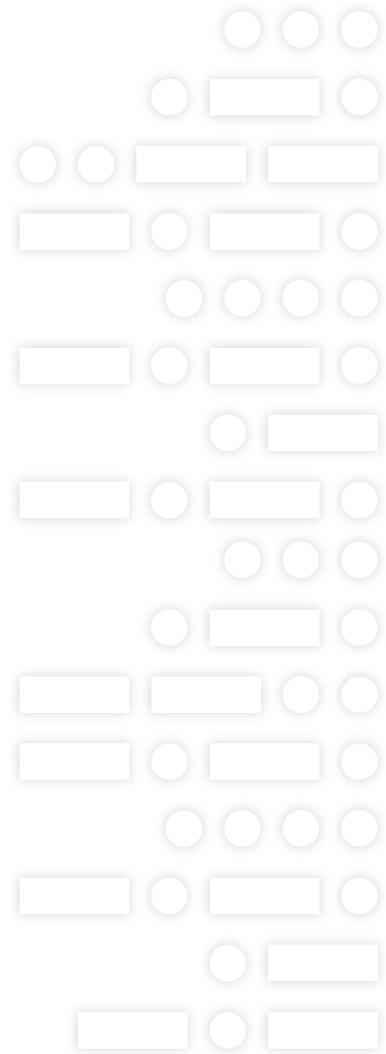
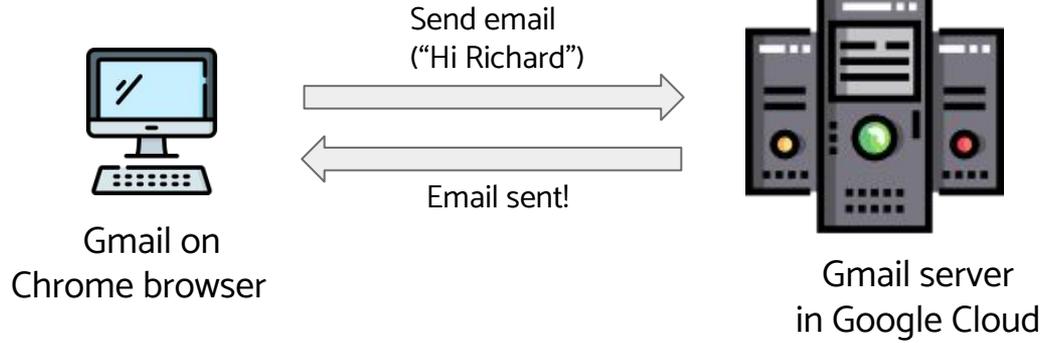


Waiter
(Client and server)

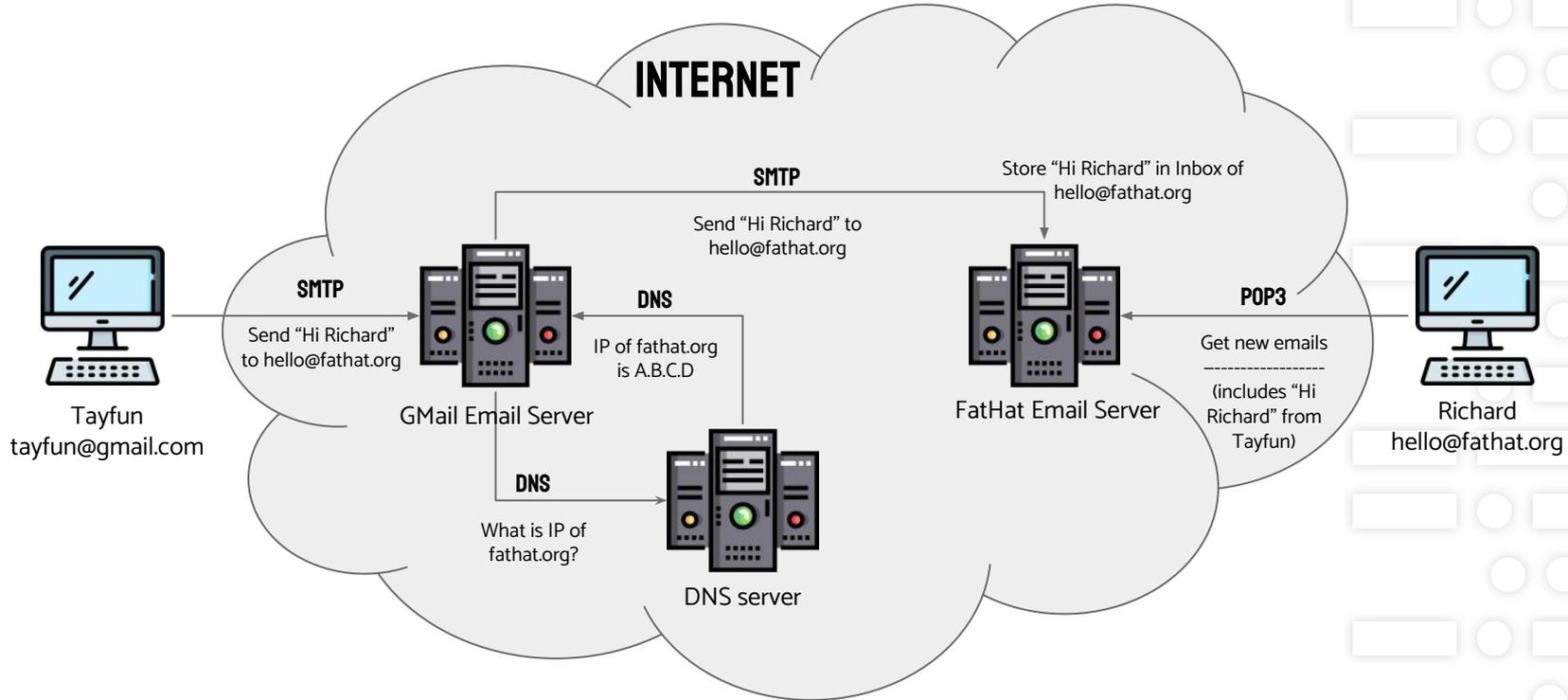


Cook
(Server)

CLIENT-SERVER: ARCHITECTURE



EXAMPLE: EMAIL



EVOLUTION OF WEB: FROM STATIC WEB PAGES TO DYNAMIC APPLICATIONS

WEB 1.0

Static web pages.

Server serves static files (HTML, images) and client renders them on screen.

WEB 2.0

Complex applications with many distributed components.

Server serves dynamically created pages and manipulates data by APIs.

WEB 3.0

Evolution of searching, better search facilities, more refined and relative search results...

WEB 4.0

Intelligent systems - evolution into a complete Operating System rather than distributed components. Less fragmented, always on, the Internet of Things (IOT), continuously connected devices...

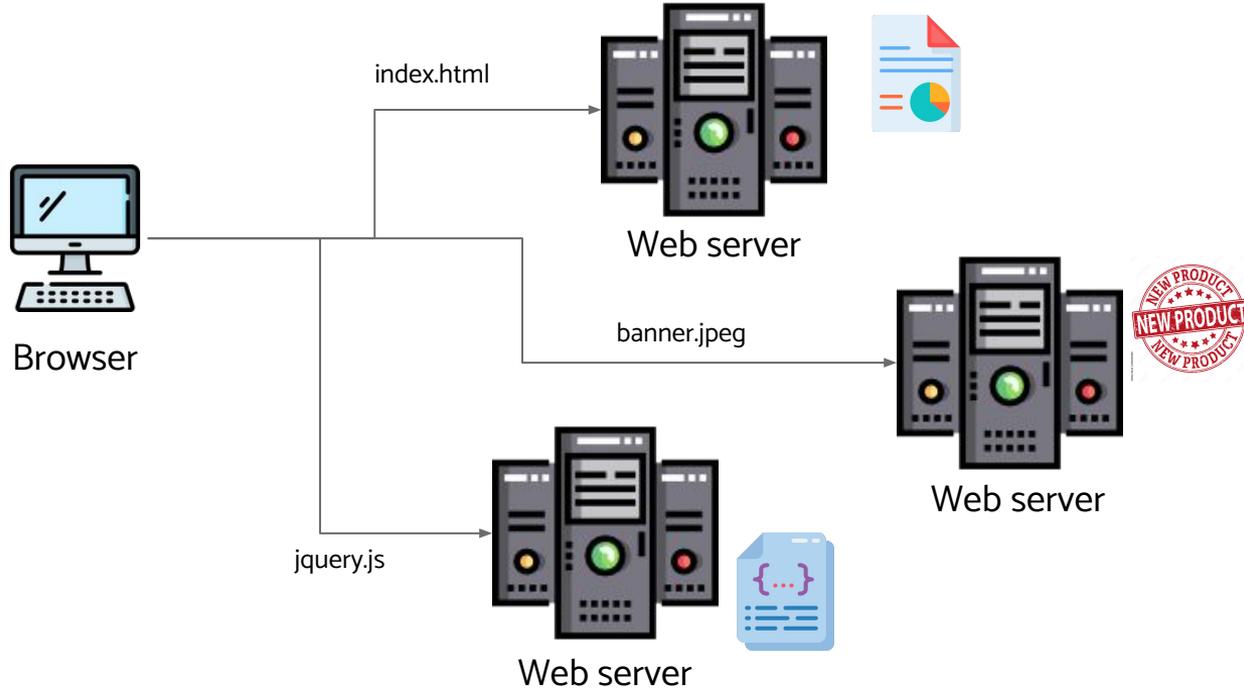
WEB 5.0 - THE FUTURE

Interaction between machines and humans, More AI (Artificial Intelligence), lots of scary stuff.



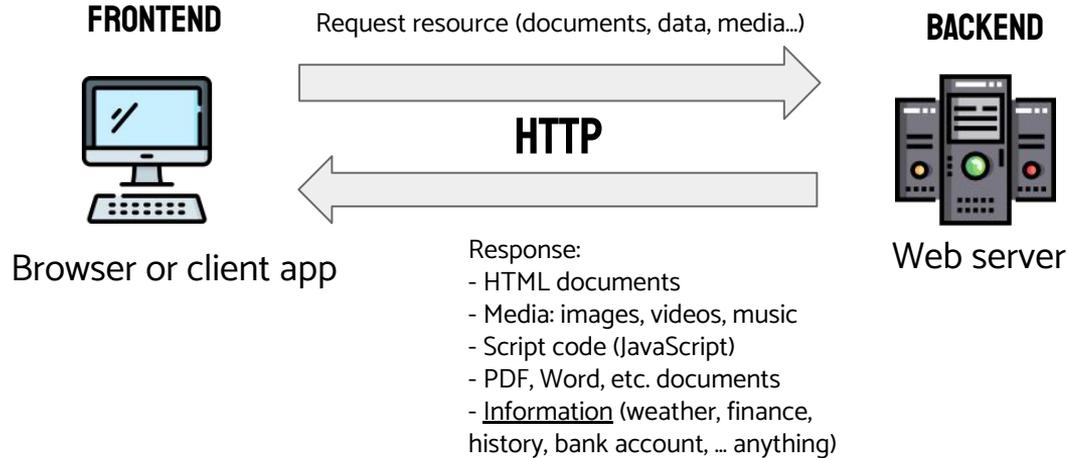
WEB I.O: SERVING STATIC FILES

Browser requests HTML document and its dependencies (eg. images, CSS files, scripts) from one or more web servers and renders a page by combining these as described in the HTML document.



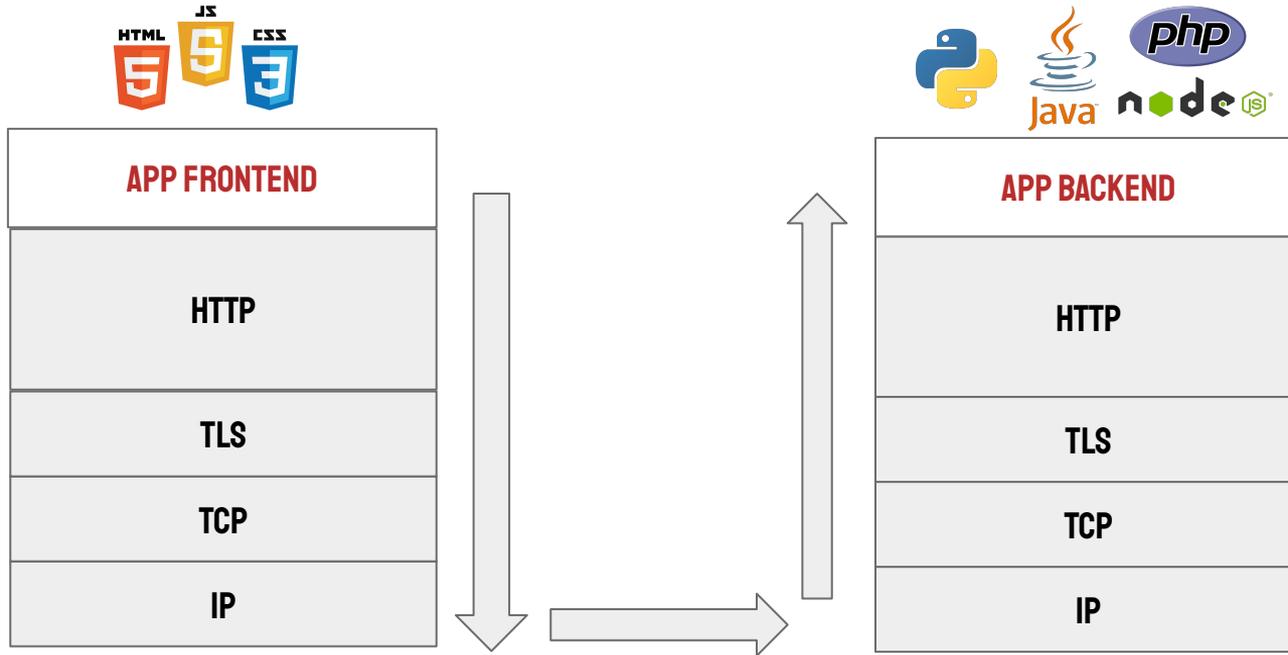
HTTP: HYPERTEXT TRANSFER PROTOCOL

HTTP is a protocol for handling requests for resources across the Internet. Like a language, it has a set of rules that must be applied during those communications.



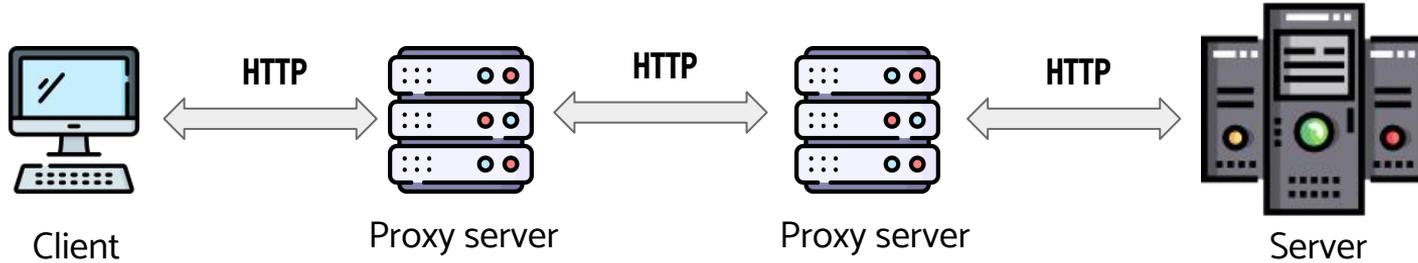
HTTP: AND OTHER PROTOCOLS

HTTP is an application layer protocol and uses TCP/IP protocols to send data between devices in a predictable sequence.



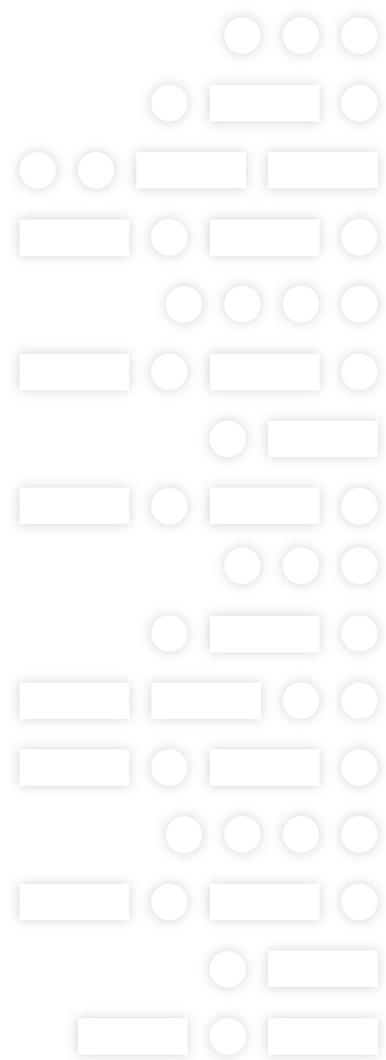
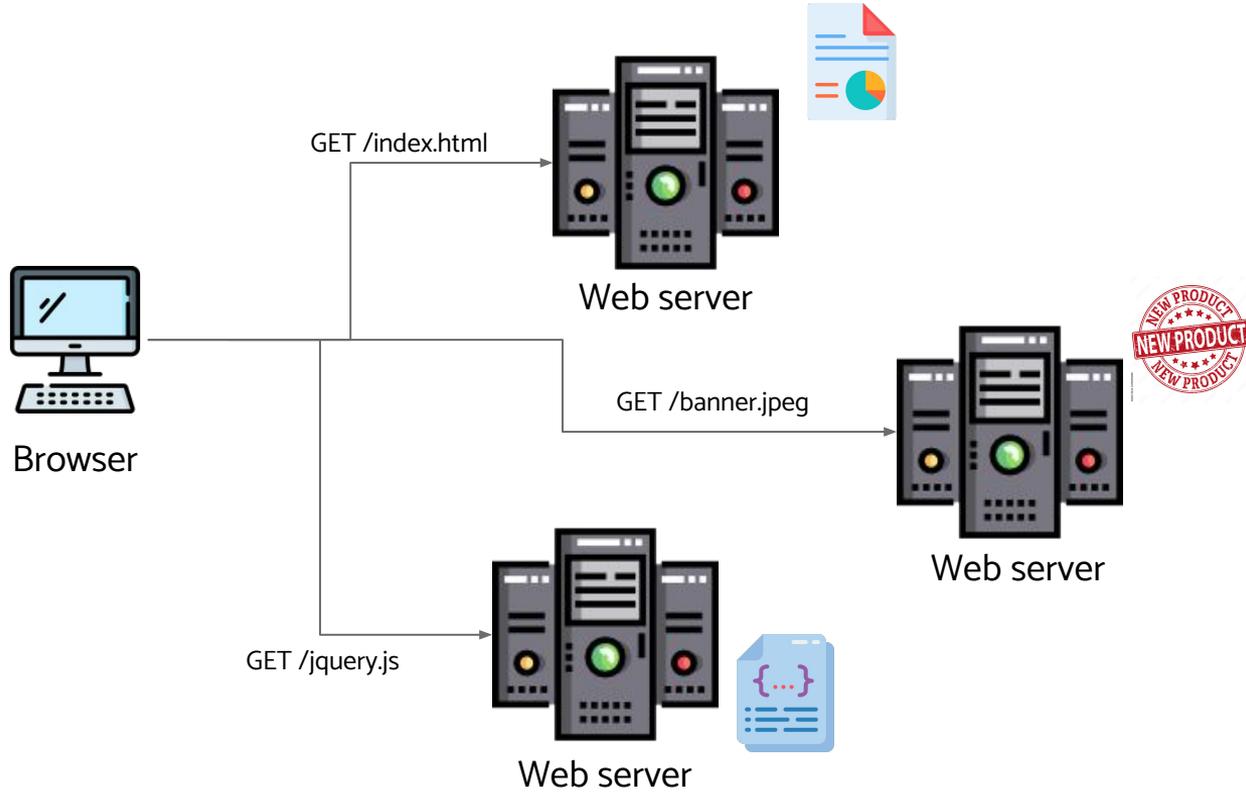
HTTP: COMPONENTS OF HTTP-BASED SYSTEMS

Requests and responses go through zero or more proxy servers before they reach their destination. Communication along the way is performed using HTTP.

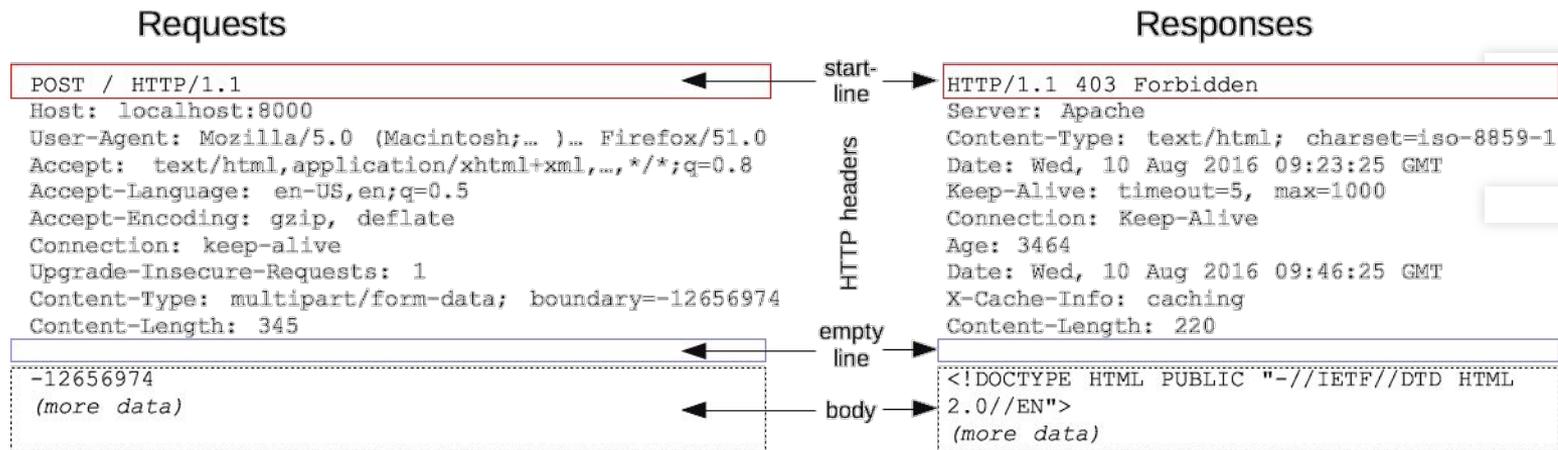


- Caching (to speed up)
- Filtering (antivirus, parental control)
- Load balancing (distribute load to multiple servers)
- Authentication/access-control (who is allowed to access specific resources)
- Logging (store requests/responses)

HTTP: FOR WEB PAGES



HTTP:REQUEST AND RESPONSE



HTTP: WHAT CAN IT CONTROL

Caching: How documents are cached by a browser.

Origin constraint: Whether a domain can request resources from another - CORS (Cross Origin resource sharing).

Authentication: How authentication is performed.

Proxy/tunneling: How requests can be proxied.

Sessions: How user state is preserved across requests.



HTTP: RESPONSE CODES

Http always returns a response with a code. The code dictating whether the request was successful or not and if not why not. A service communicating using the HTTP protocol should follow these standard error codes.

1XX
Informational

101 - Switching protocols

2XX
Success

200 - Success (OK)
204 - No content

3XX
Redirection

301 - Moved permanently
302 - Moved temporarily
303 - See other

4XX
Client error

400 - Bad request 405 - Method not allowed
401 - Unauthorized 407 - Proxy auth required
403 - Forbidden
404 - Not found

5XX
Server error

500 - Internal server error
501 - Not implemented
502 - Bad gateway
503 - Service unavailable

