Client Server Architecture

Client Server Architecture

- The **clients requests** a resource and the **server respond** with that resource.
- Client-server architecture (client/server) is a **network architecture** in which each computer or process on the network is either a **client** or a **server**.

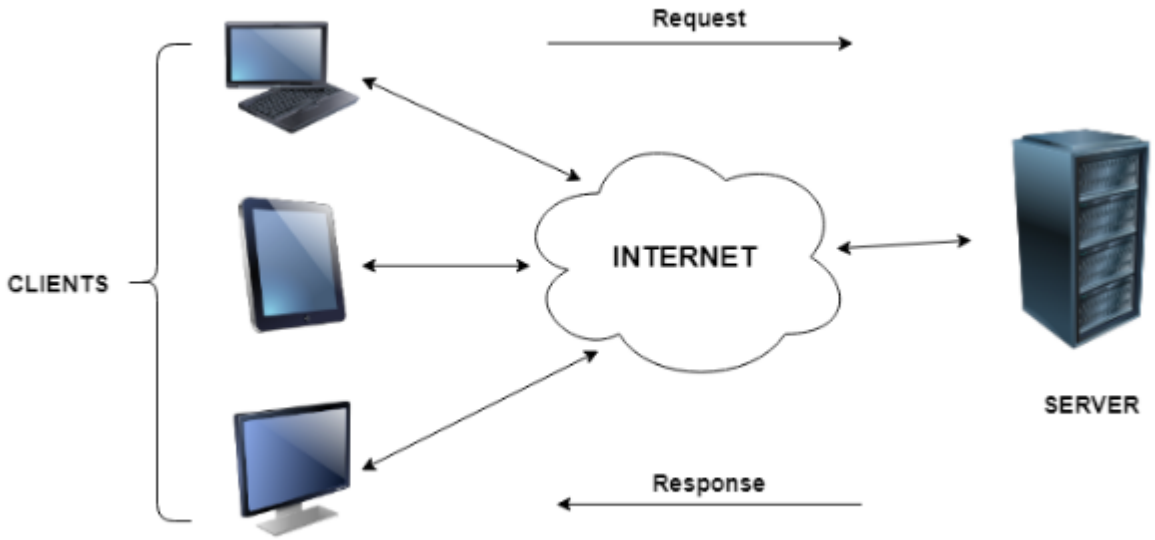
Types of Client Server Architecture

- **There are three different structures :**
  - Two – Tier Client/Server Structure
  - Three - Tier Client/Server Structure

Two – Tier Architecture

- The two tier architecture primarily has two parts, **a client tier** and a **server tier.**
- The client tier sends a **request** to the server tier and the server tier **responds** with the desired information.
- An example of a two tier client/server structure is a **web server**. It returns the required web pages to the clients that requested them.

2 – Tier Architecture



Advantages of Two – Tier Architecture

Some of the advantages of the two-tier client/server structure are:

- This structure is quite easy to maintain and modify.
- The communication between the client and server in the form of request response messages is quite fast.

Disadvantages of Two - Tier Client/Server Structure

- If the client nodes are increased beyond capacity in the structure, then the server is not able to handle the request overflow and performance of the system degrades.

Advantages of Two – Tier Architecture

Some of the advantages of the two-tier client/server structure are:

- This structure is quite easy to maintain and modify.
- The communication between the client and server in the form of request response messages is quite fast.

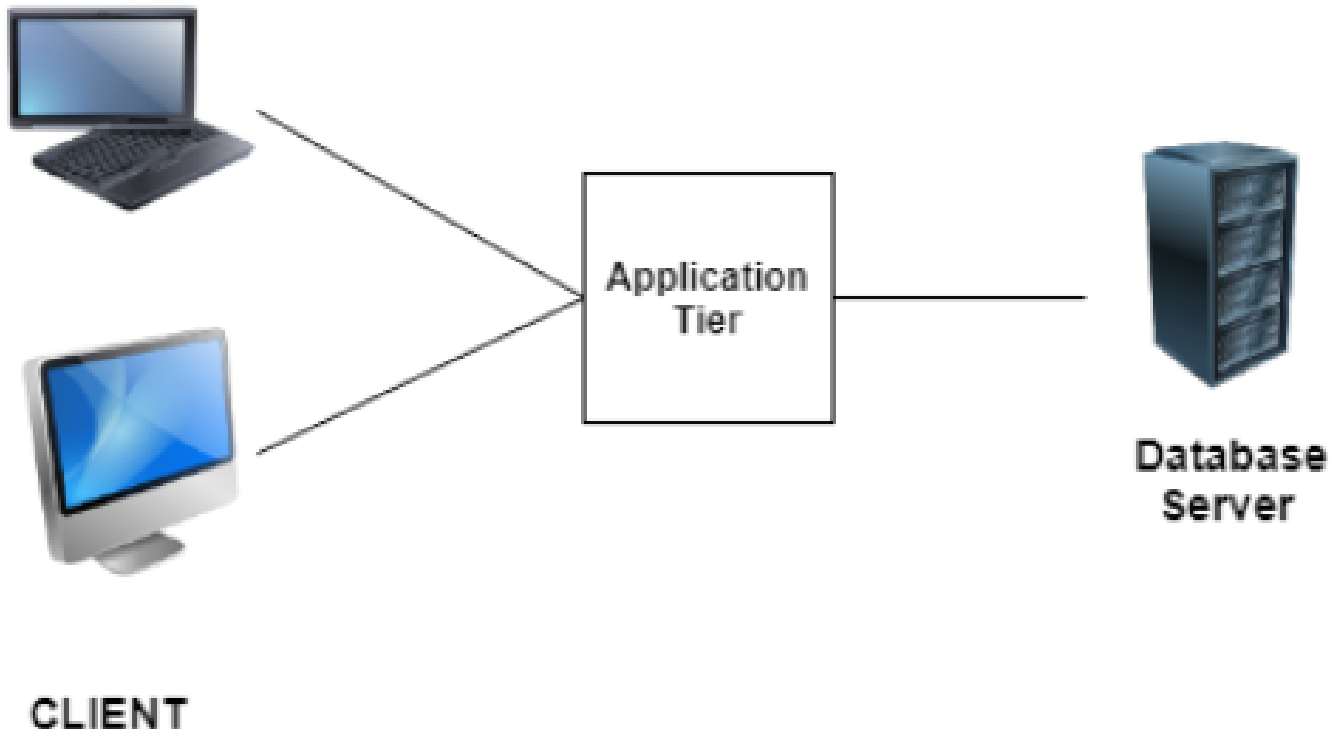
Disadvantages of Two - Tier Client/Server Structure

- If the client nodes are increased beyond capacity in the structure, then the server is not able to handle the request overflow and performance of the system degrades.

Three – Tier Architecture

- The three tier architecture has three layers namely **client**, **application** and **data layer**.
- **The client layer** is the one that requests the information. In this case it could be the GUI, web interface etc.
- **The application layer** acts as an interface between the client and data layer. It helps in communication and also provides security.
- **The data layer** is the one that actually contains the required data.

3 – Tier Architecture



Advantages of Two – Tier Architecture

Some of the advantages of the three-tier client/server structure are:

- The three tier structure provides much better service and fast performance.
- The structure can be scaled according to requirements without any problem.
- Data security is much improved in the three tier structure.

Disadvantages of Three - Tier Client/Server Structure

- Three - tier client/server structure is quite complex due to advanced features.

C-S Architecture

Simple
Isn't it?

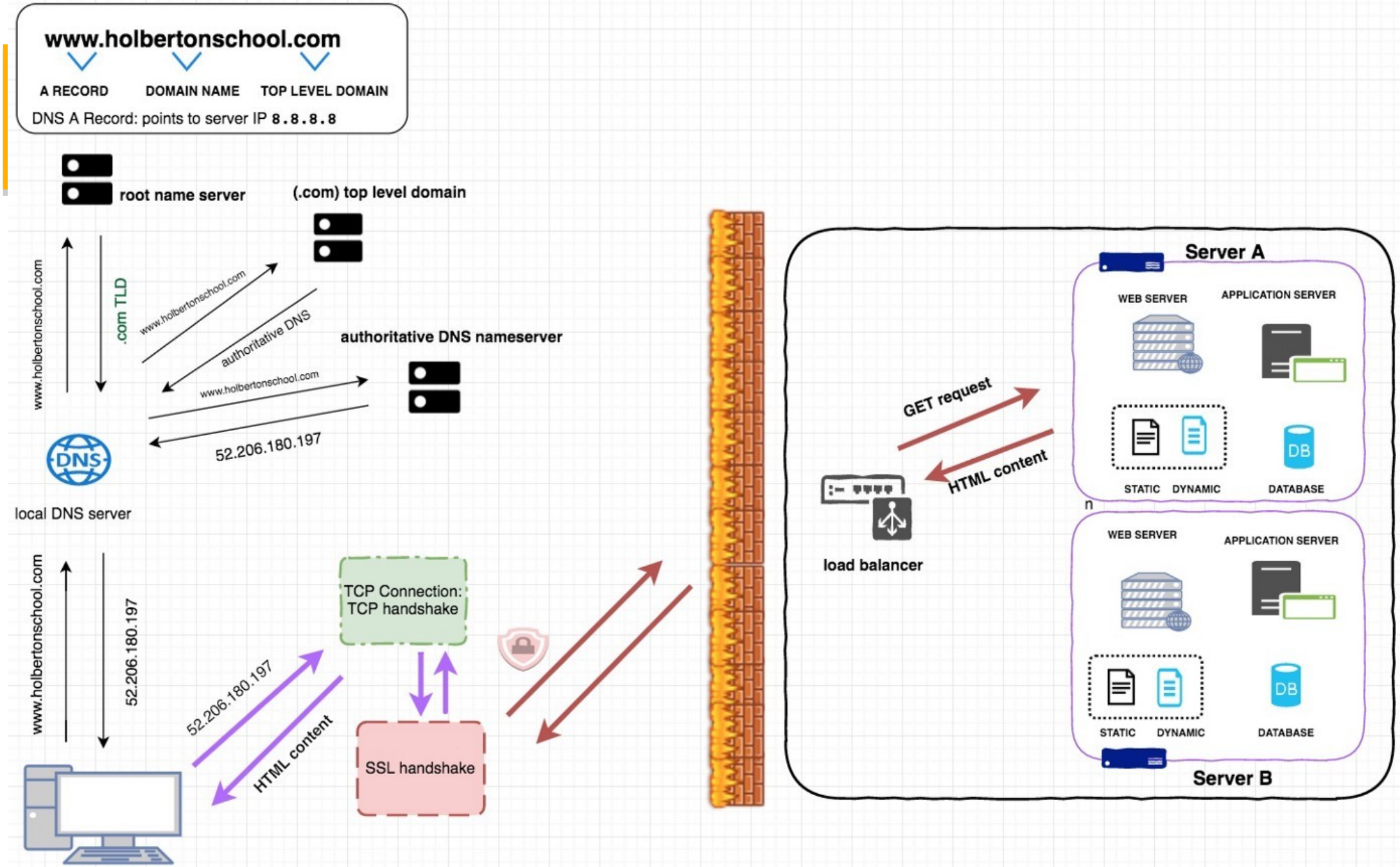
Now i know

What happens when you type a URL in a browser and hit enter?

Topics Covered

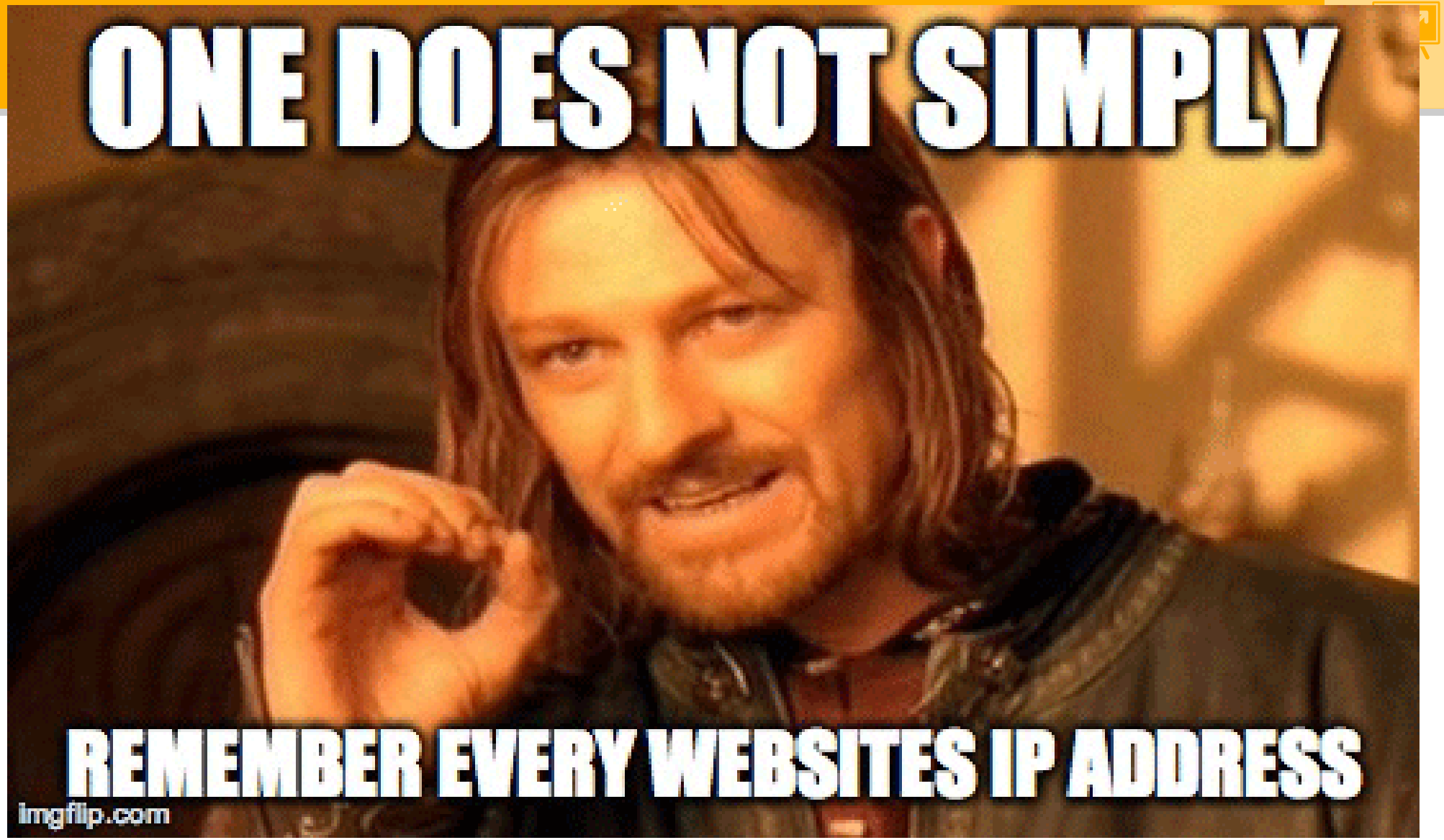
Topics covered:

- DNS Lookup
- TCP/IP
- HTTPS/SSL/TLS
- **Server**: Firewall, Load Balancer, Web Server, Application Server, Database


www.holbertonschool.com
A RECORD
DOMAIN NAME
TOP LEVEL DOMAIN
DNS A Record: points to server IP 8.8.8.8
root name server
(.com) top level domain
www.holbertonschool.com
.com TLD
www.holbertonschool.com
authoritative DNS
authoritative DNS nameserver
www.holbertonschool.com
52.206.180.197
DNS
local DNS server
www.holbertonschool.com
52.206.180.197
TCP Connection:
TCP handshake
52.206.180.197
HTML content
SSL handshake
GET request
HTML content
load balancer
Server A
WEB SERVER
APPLICATION SERVER
STATIC
DYNAMIC
DATABASE
n
WEB SERVER
APPLICATION SERVER
STATIC
DYNAMIC
DATABASE
Server B

- The **internet** is made up of **a network of computers** connected to each other
- In order to connect and **communicate between computers**, they must follow a set of rules, **Internet Protocol(IP),** that govern how data is transmitted over a network.
- Every machine on the web has a unique identifier to distinguish from one another. It’s similar to having a telephone number or a physical address.

- A typical IP address(IPv4) follows the format of 4 sets of numbers between 0–255. xxx.xxx.xxx.xxx.
- There are a total of 4,294,967,296 **IPv4** addresses available and 340,282,366,920,938,463,463,374,607,431,768,211,456 possible **IPV6** addresses.
- Having to look for the IP address when you want to visit a webpage will take a lifetime.
- So we just type in the domain name we’re familiar with and let the magical system that is the DNS to take care of the rest :)


ONE DOES NOT SIMPLY
REMEMBER EVERY WEBSITES IP ADDRESS
imgflip.com

DNS(Domain Name System)

- The Domain Name System is created to **keep track of IP addresses** for us so we can enter human-readable addresses in our browser’s URL bar instead.
- To translate from **domain name** to **IP address**.
- When you type www.google.com in your web browser and hit enter, the request will be forwarded to a DNS server. DNS server will then perform a ***DNS lookup*** to locate the corresponding IP address
- Nslookup

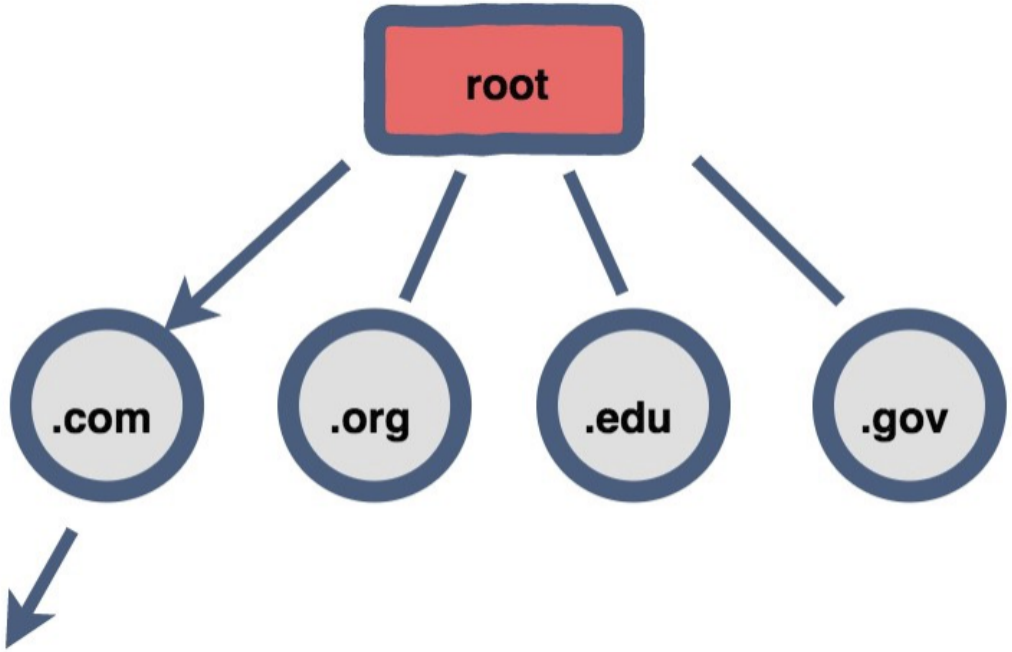
DNS

- DNS uses a **client/server architecture** and the **DNS servers are organized in a hierarchical and distributed fashion**.

- In order to connect and communicate between computers, they must follow a set of rules, Internet Protocol(IP), that govern how data is transmitted over a network.

Advantages of Two – Tier Architecture

- And the route in which we take will look similar to a **upside down tree-like structure**



www.google.com

Steps to resolving a domain name

1. Web browser and OS will first **check whether the domain is in their cache. If yes, done**
2. The web browser then will send a request to a **DNS resolver**. A DNS resolver is a local server with a central database of DNS nameservers. This DNS resolver will be hosted with your ISP
3. The resolver will first check its cache. If the IP address for google.com isn't in its cache it will forward the query **recursively up the to the root servers**,
4. Down to the Top Level Domain (TLD) of google.com(.com would be the TLD in this case), and then down to the authoritative name servers responsible for www.google.com.

DNS system is organized in an upside down tree-like structure right?

We will go to the top and search downward. First stop is the root servers.

Root servers respond with address to the **.com** Top-Level Domain(TLD). Top-level domain just refers to the last chunk of a domain name after the dot symbol. Here we go to the .com TLD.

The resolver then queries .com servers for the **authoritative name servers** of our domain, google.com

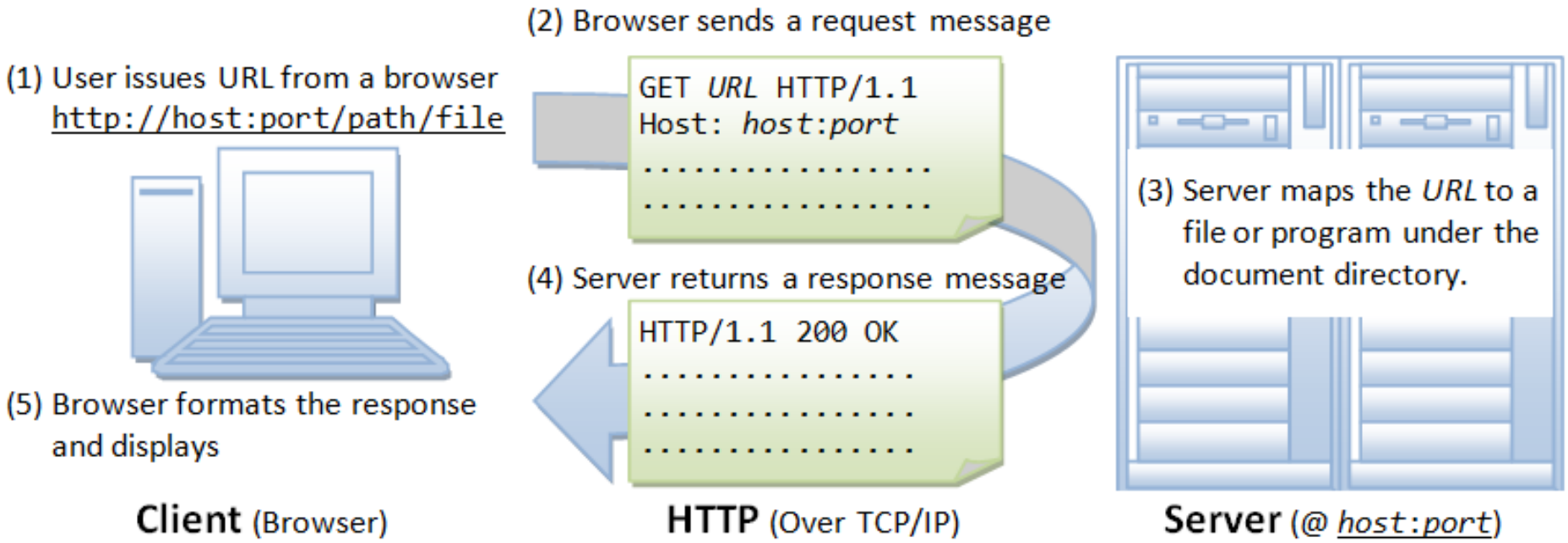
- Authoritative name servers will respond with the corresponding IP address of www.google.com

- But first, it will save this IP to its cache. **Caching every step of the way**!

- The user's operating system will also cache this IP address for reference in the future—in case you want to visit this website again 🏝

DNS DONE !! NOW HTTP

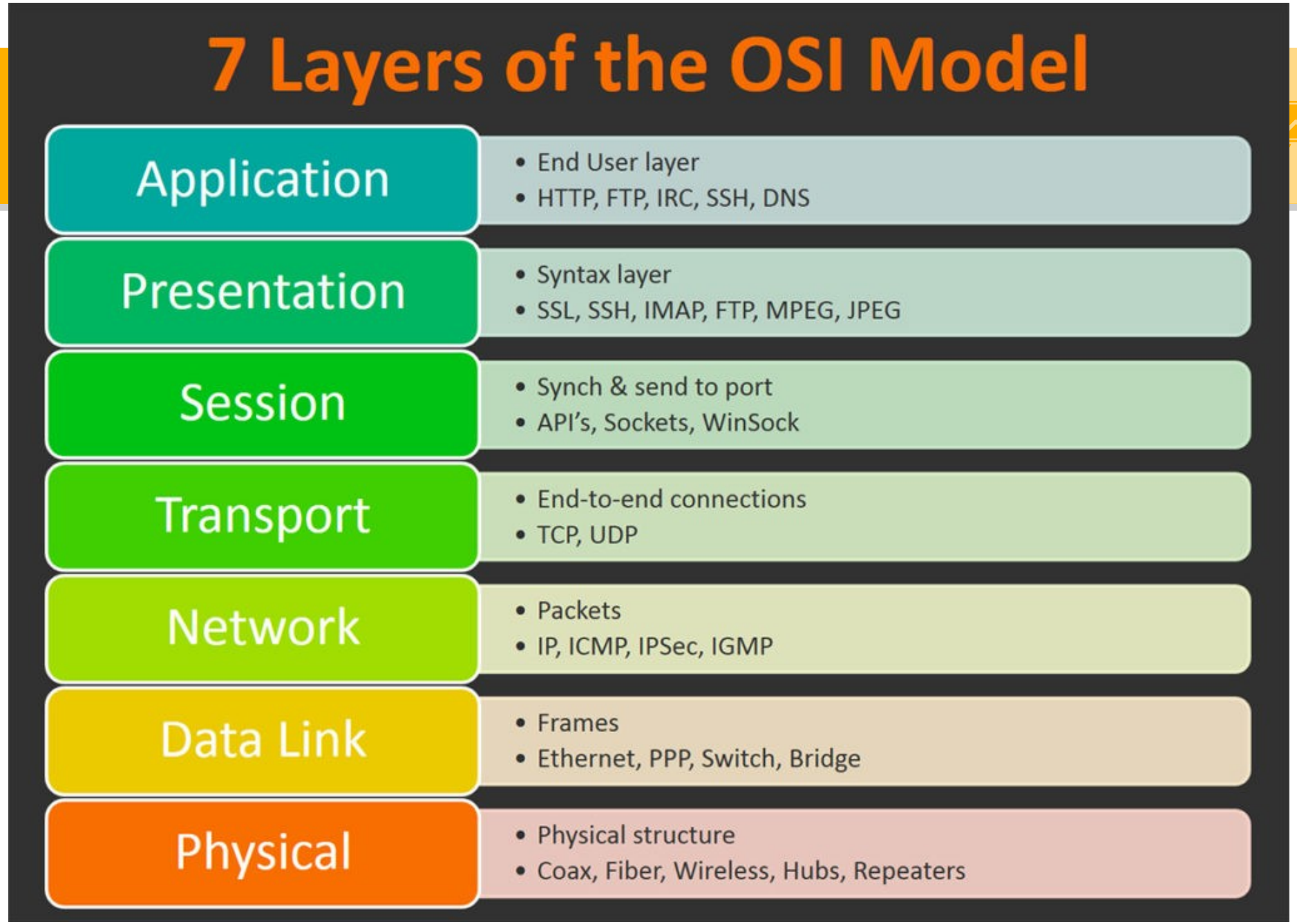
- The user's web browser can now follow **HTTP(HyperText Transport Protocol)** and send a **GET request** to the server at google.com's



HTTP

- HTTP is the protocol used to transfer data to and from the website.
- WWW is the identifier that indicates that it is a web site and it uses the HTTP protocol.
- HTTP://anything.com, WWW.anything.com, HTTP://WWW.anything.com leads to the same site.

7 Layers of the OSI Model

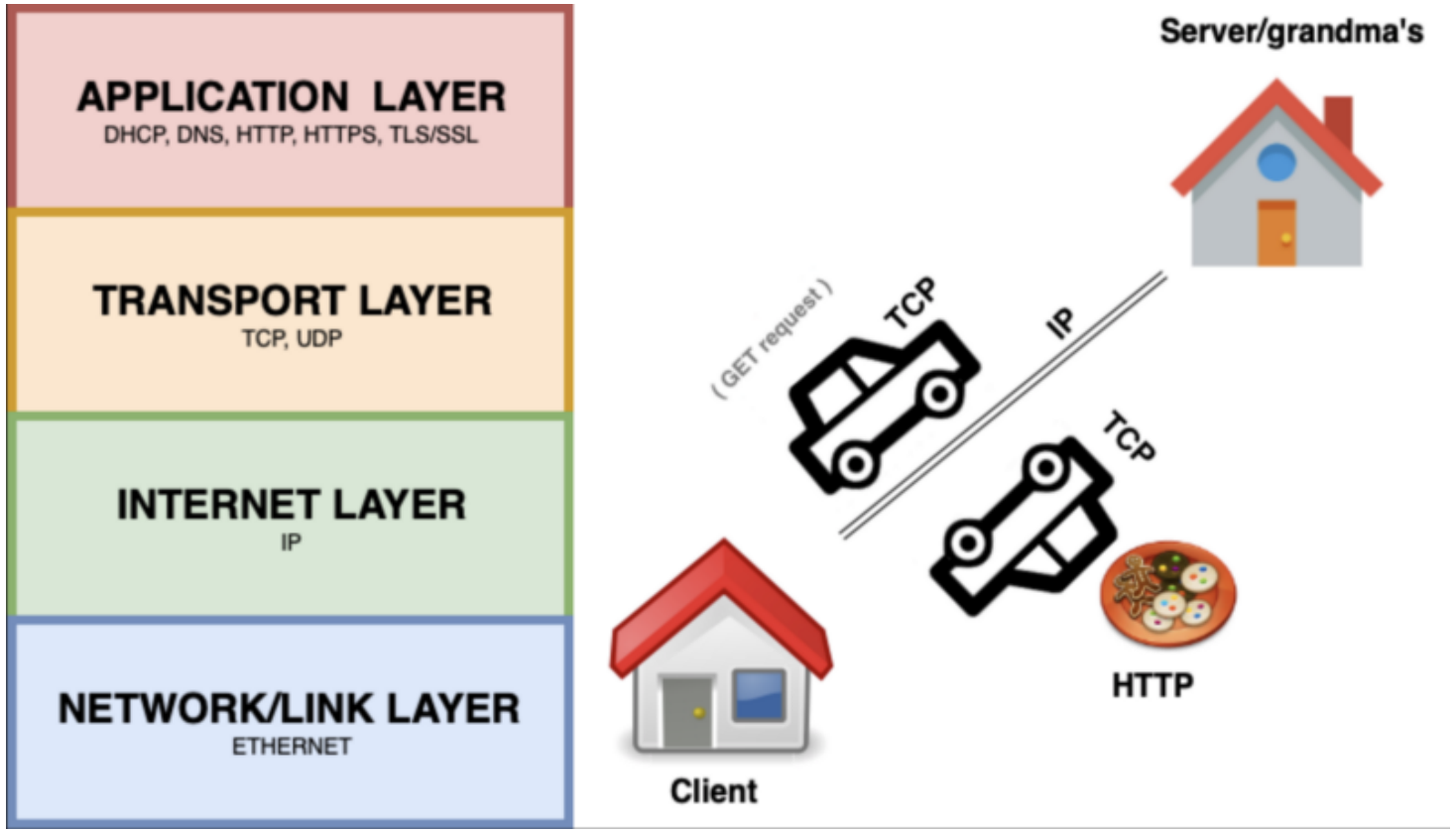


HTTP & TCP

1. **Hypertext Transport Protocol (HTTP)** is an application layer protocol used for transmitting files/data across the web through **TCP/IP sockets**
2. **TCP(Transmission Control Protocol)** resides in the transport layer and is responsible for creating a reliable end-to-end connection between two hosts. It’s similar to a messenger.
3. Allows data transfers of other protocols(like HTTP). TCP will break the data down into smaller packets and then reassemble them at the other end.

Analogy of picking up cookies from grandma's house

1. **IP** would be the **road on which we drive**
2. **TCP** would be the **car**
3. And **HTTP** would be the box of **cookies** moving from one location to another



TCP/IP

1. **HTTP** relies on **TCP** to establish a reliable connection between client and server. Four pieces of information are needed to establish a **TCP connection**:

Client IP address
Client Port number
Source IP address
Source Port number

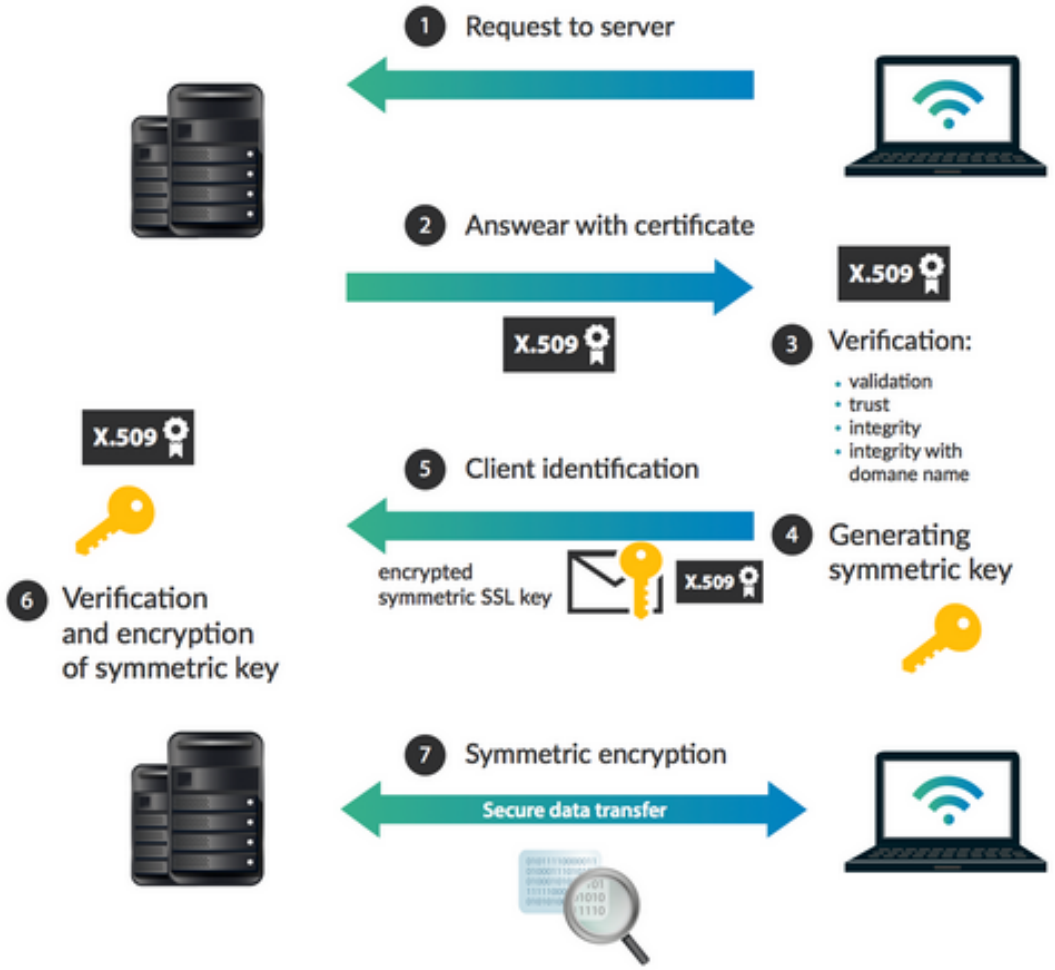
An IP address will identify the device, but a port number is also needed to identify the specific application/service.

It’s similar to having an address to an apartment and a specific unit number.
IP + port number = socket

HTTP protocol uses port 80 as default

SSL (HTTPS)

- Client say Hello
- Server says Hello(do you have cer
- Client generated symmetric key
- Server verify the key
- Once verification is successful
- ☾ Secure data transfer



Firewall

1. TCP network breaks data into chunks (packets). Along with data, a packet will have a header including control information such as **source address, destination address, connection state**, etc.
2. Protect (a network or system) from unauthorized access with a firewall

Load Balancer, WebServer, Application Server and Database

1. **Load Balancer**: The job of a load balancer is to…. balance loads.
2. **WebServers**: Web servers supply the web content for web browsers; what the browser requests, the web server delivers through Internet network connections.
3. **Application servers** host and execute applications and can be used to communicate and extract data from a database
4. Da[redacted]