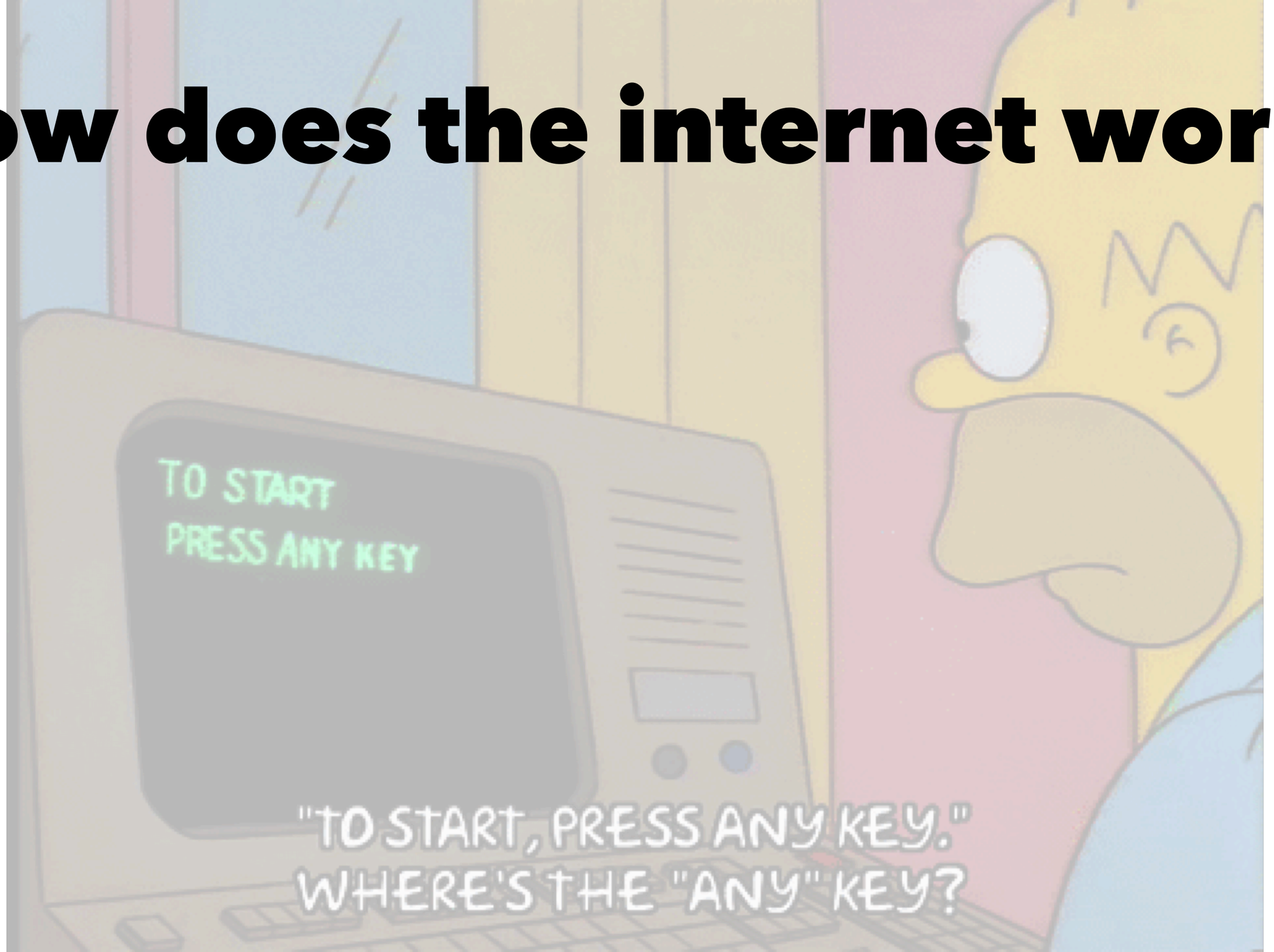
How does the internet work?



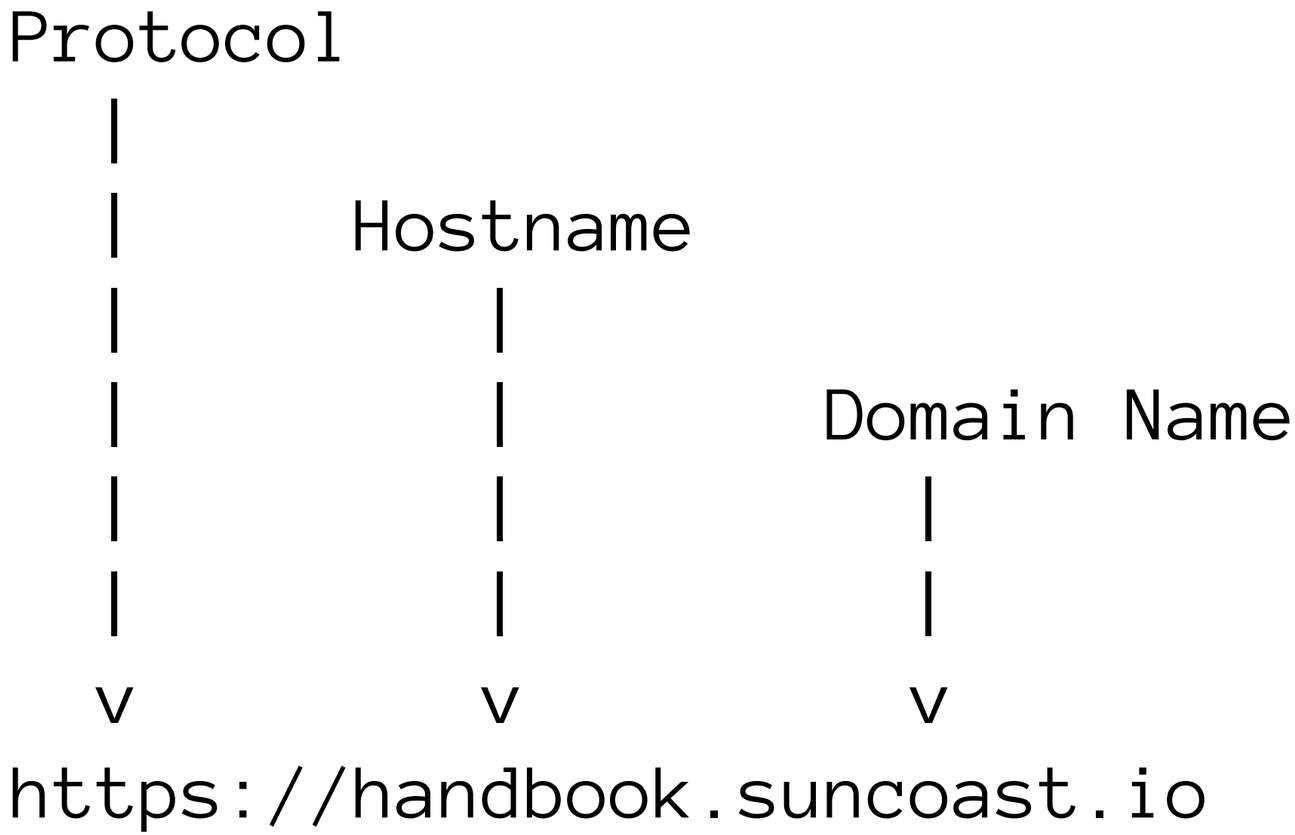
Start at the beginning

If we visit `handbook.suncoast.io` how does our browser know where to go?

What are the technologies and processes involved in making this work?

Domains and Hosts

URL



But where is it?

IP Address

- Once the browser realizes you wish to visit `handbook.suncoast.io` it must figure out how to address the server.
- The internet itself does not deal in terms of names like `handbook.suncoast.io` but in terms of a numerical address in the form `1.2.3.4` or `192.168.145.241`.

DNS

So how does the browser know to find the `IP Address` of `handbook.suncoast.io`?

It uses a service known as `DNS` (`Domain Name Services`).

Every client has a preset list of `IP Addresses` that are equipped to perform a translation of a domain name like `handbook.suncoast.io` into its IP address `104.248.50.87`.

The `DNS` process allows your computer to quickly translate the address.

- Checks defined DNS server.
- Likely your local router.
- Sees if it is recently resolved, given "Time To Live" (TTL).
- If not, check's its DNS server. Typically your ISP.

This is a nice visualization of the DNS process.

Let's lookup a few addresses

NOTE: Use `nslookup` *if* `dig` *isn't available.*

```
dig handbook.suncoast.io
dig amazon.com
dig www.yahoo.com
```

We will see that DNS can return multiple values.

We will see examples of CNAMEs.

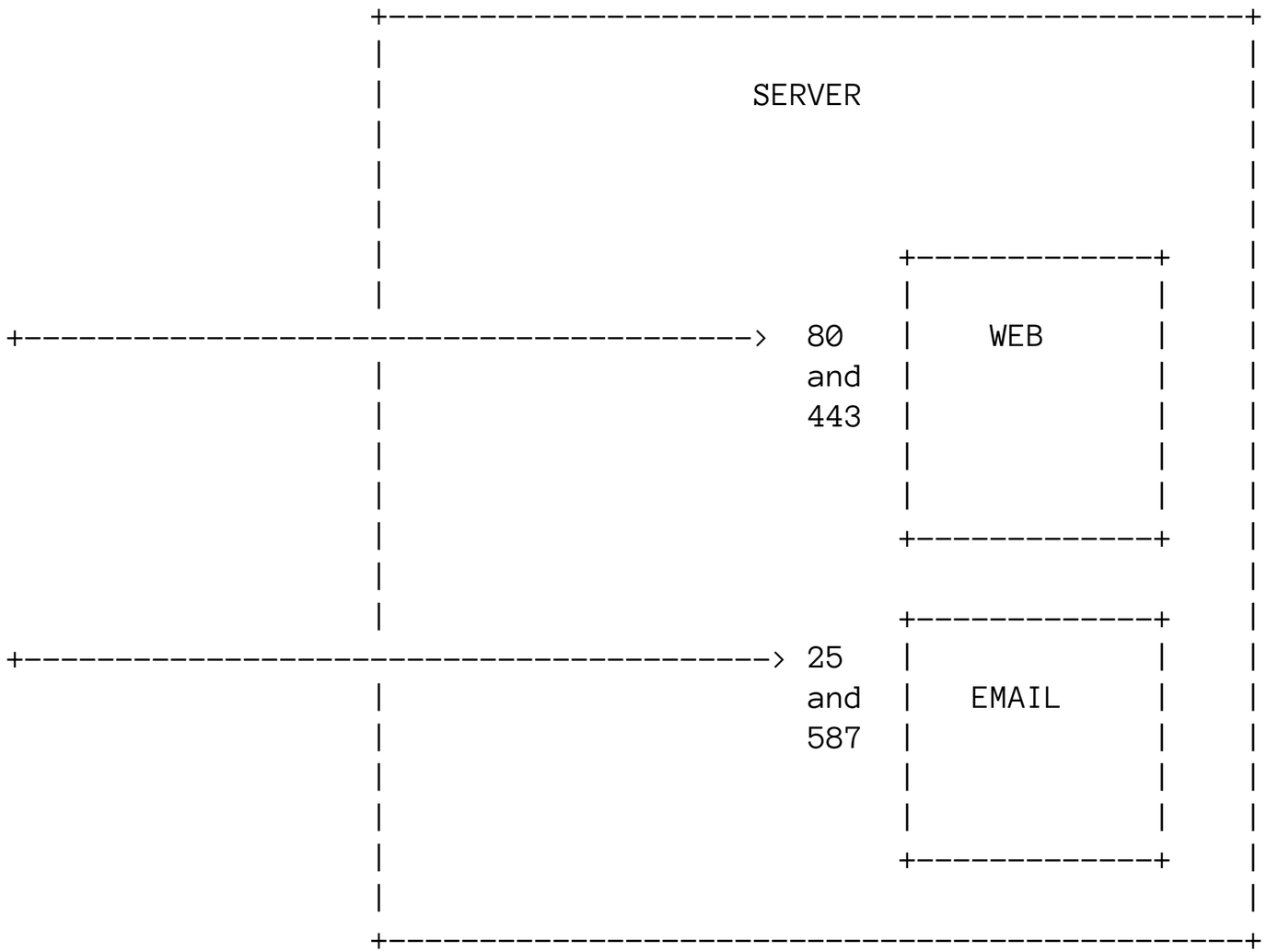
Making Connections

What happens after our computer resolves the address of a server?

- The next thing we need to do is connect to a particular *service* on that machine.
- Remember when we mentioned that part of the URL is a `protocol`. This also indicates which `service` we want to speak to.

Port Numbers

- Each of these `services` will be *listening* for a connection from a client on a particular `port number`.
- If you think of the `IP Address` as a street address of an office building, you can think of the `port number` as which room in the building the service is in.



^ Ports 1 - 65535
^ Ports 1 - 1024 popular, require operating system privileges

Sockets

To connect to a service on a port the computer creates something called a socket.

A socket is a virtual connection between your computer and a port on a remote computer.

Think of it like a pipe that information can flow through in both directions.

Once this socket is established we can *send* information and we can *receive* information in return.

How do we connect to one of these ports?

Let's try connecting like the browser does.

For this, we will be using a tool named `netcat`.

- On Mac OS install it with: `brew install nc`
- On Windows install it with: `scoop install netcat`
- On Linux it is likely `sudo apt install netcat`

To connect: `nc handbook.suncoast.io 80`

HTTP

Now that we are connected, how do we talk? We use the http protocol that is well documented.

GET a page

```
GET / HTTP/1.1
Host: handbook.suncoast.io
```

(important blank line after Host:)

Response (Headers)

- Status codes
- See this funny list of codes

Range	Guide
200–299	Everything ok
300–399	Go elsewhere
400–499	Client mistake
500–599	Server mistake

What are other HTTP headers?

- Common headers:

Header	Meaning
`Date`	*Timestamp on Server*
`Content-Type`	*How should this content be interpreted*
`Content-Length`	*How long is this content in bytes*
`Last-Modified`	*When was this content last modified*

Other tools

- `curl`
- `http` (httpie)
- `Insomnia`