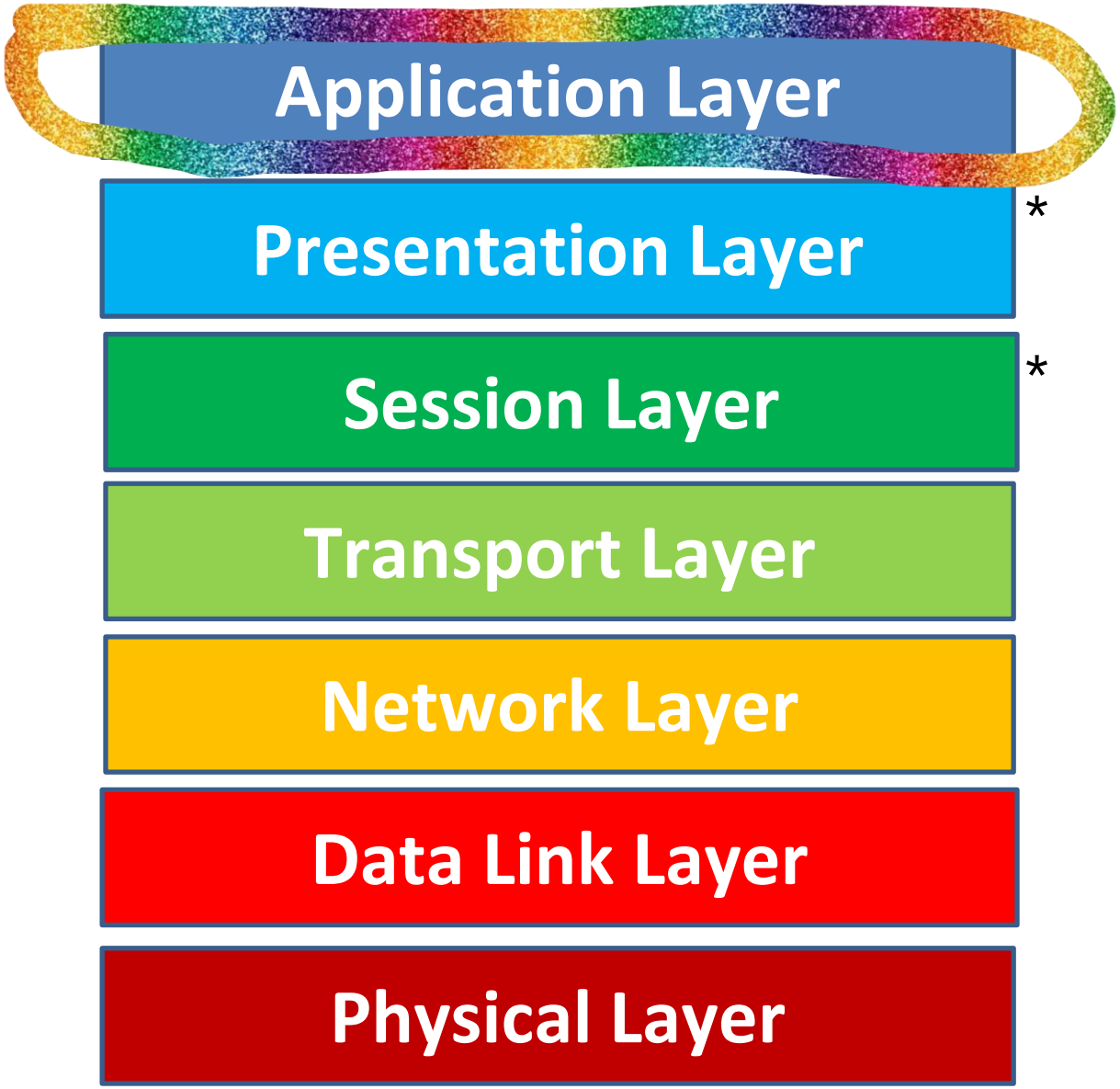
CSCI 466: Networks

Lecture 4: Application Layer

Reese Pearsall

Fall 2022

OSI Model



Application Layer

Messages from Network Applications

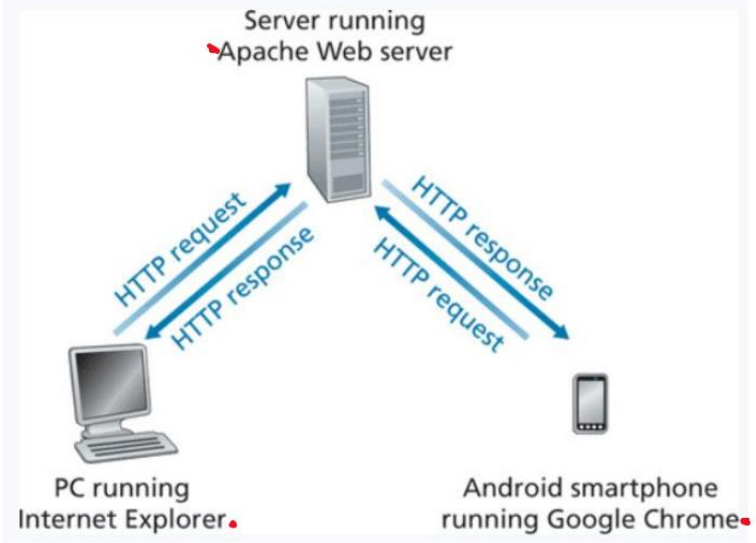
Physical Layer

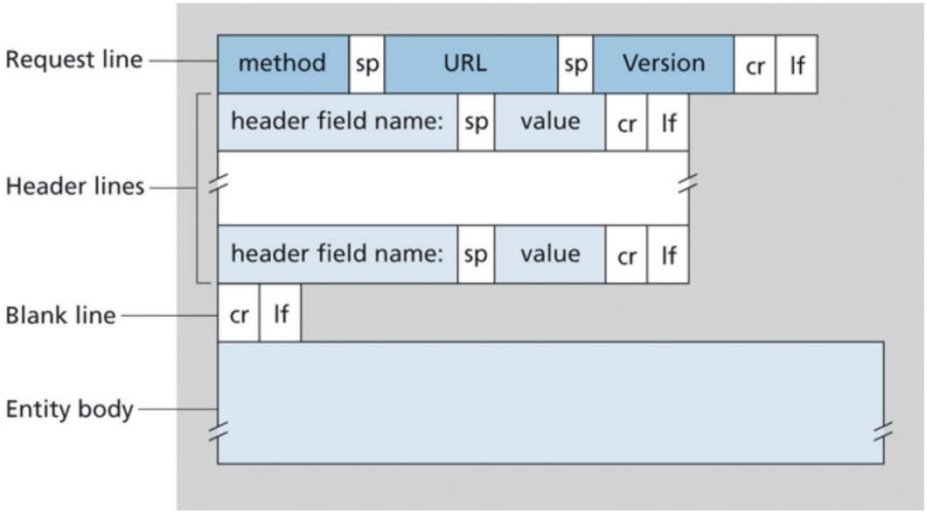
Bits being transmitted over a copper wire

**In the textbook, they condense it to a 5-layer model, but 7 layers is what is most used*

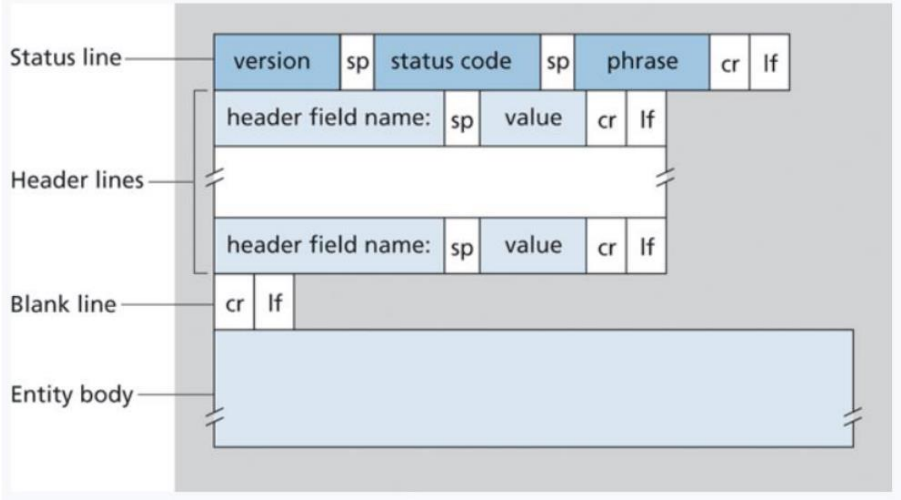
HyperText Transfer Protocol (HTTP)- protocol that dictates the transmitting of hypermedia documents such as HTML and other webpage objects

All web navigation is done through HTTP





REQUEST



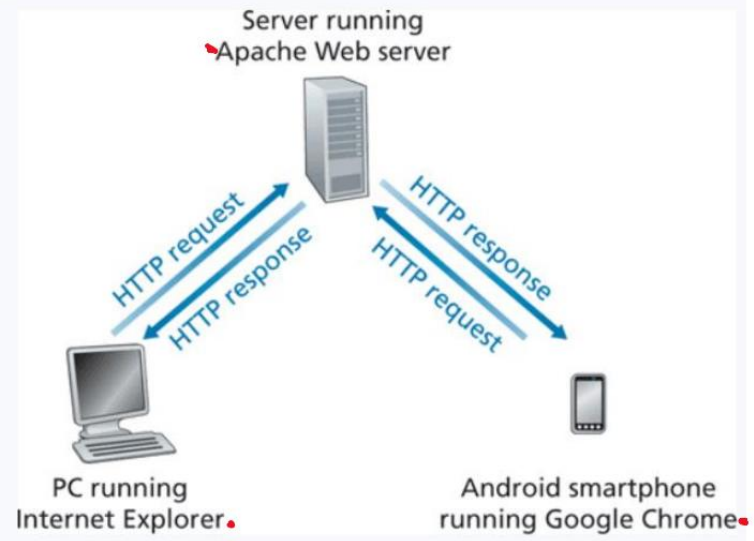
RESPONSE

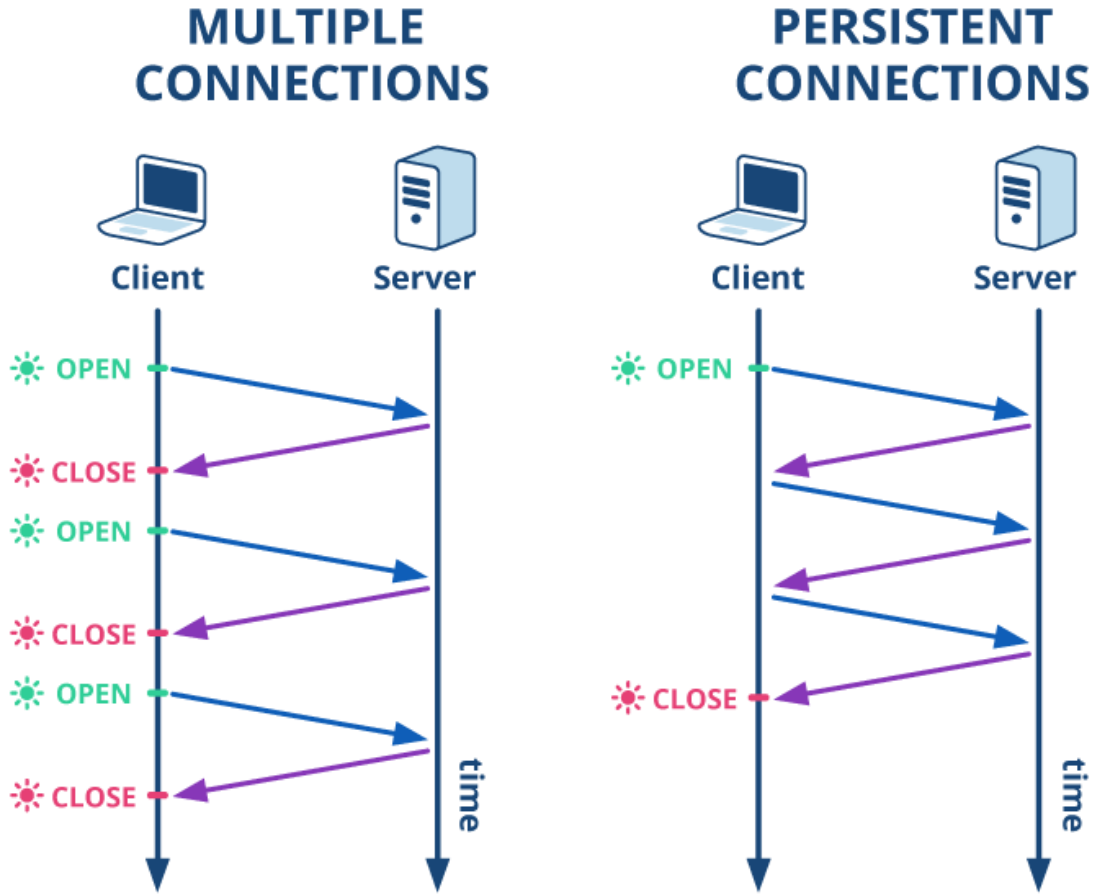
HyperText Transfer Protocol (HTTP)- protocol that dictates the transmitting of hypermedia documents such as HTML and other webpage objects

HTTPS- is the secure implementation of HTTP

HyperText Transfer Protocol (HTTP)- protocol that dictates the transmitting of hypermedia documents such as HTML and other webpage objects

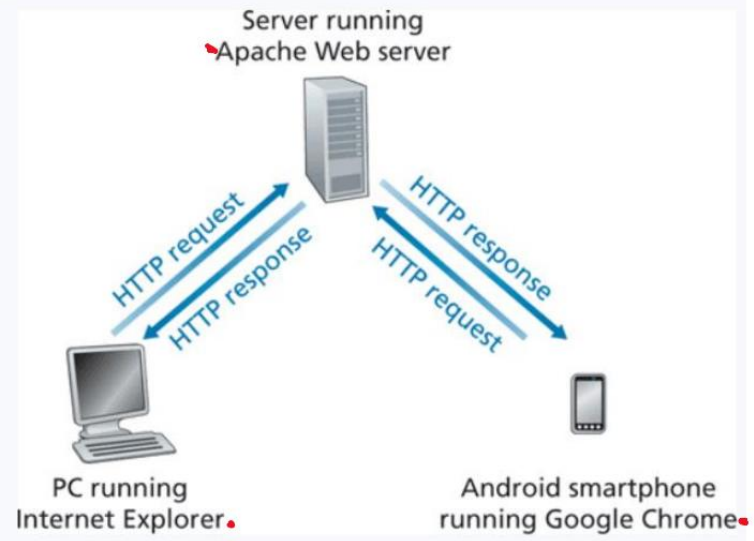
All web navigation is done through HTTP



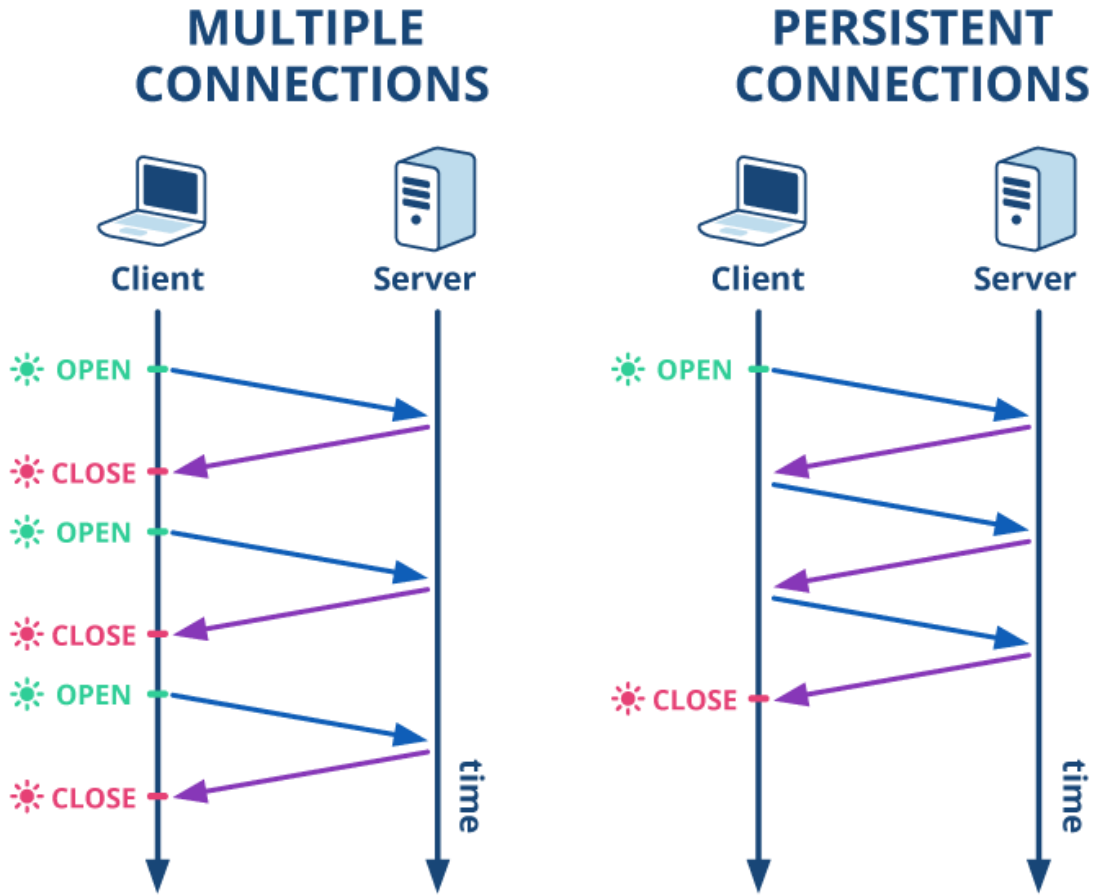


HyperText Transfer Protocol (HTTP)- protocol that dictates the transmitting of hypermedia documents such as HTML and other webpage objects

All web navigation is done through HTTP



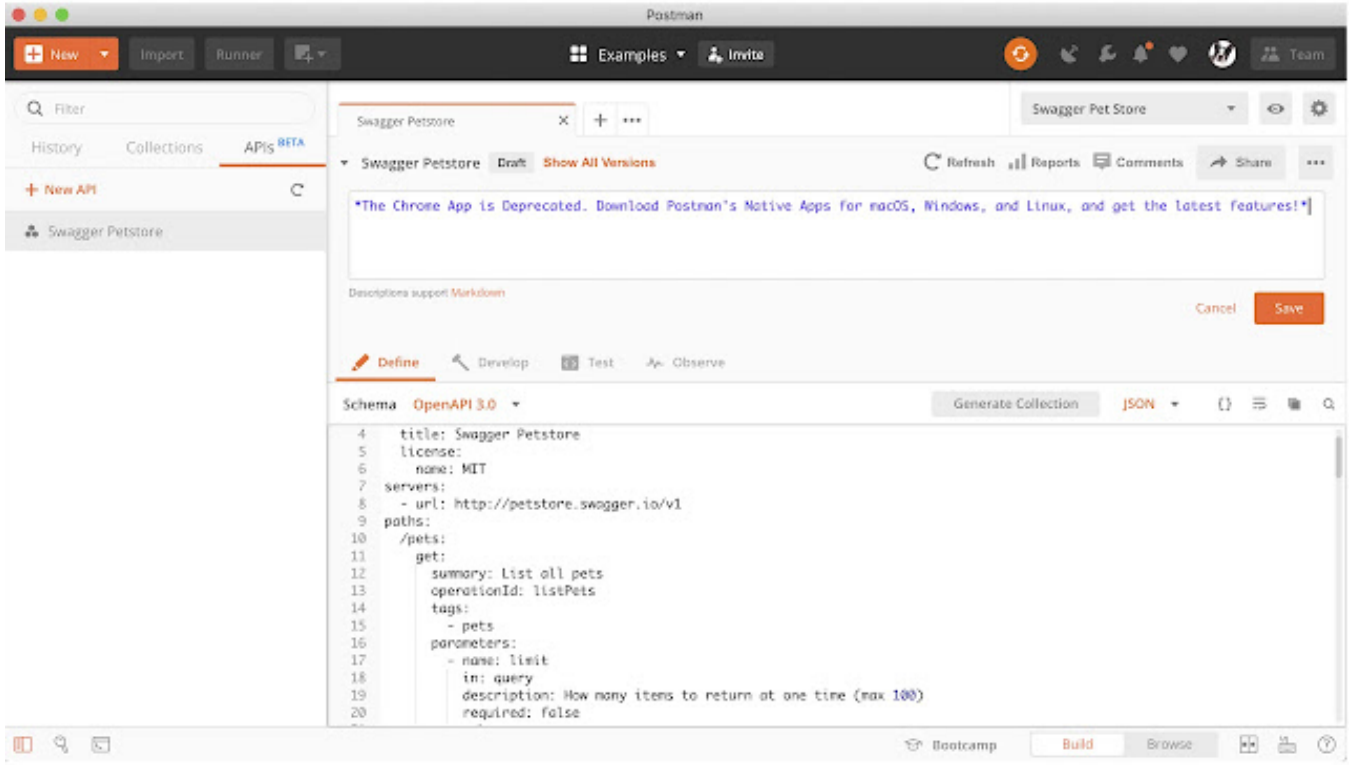
The process of one computer establishing a connection with another computer or device is called a **handshake**



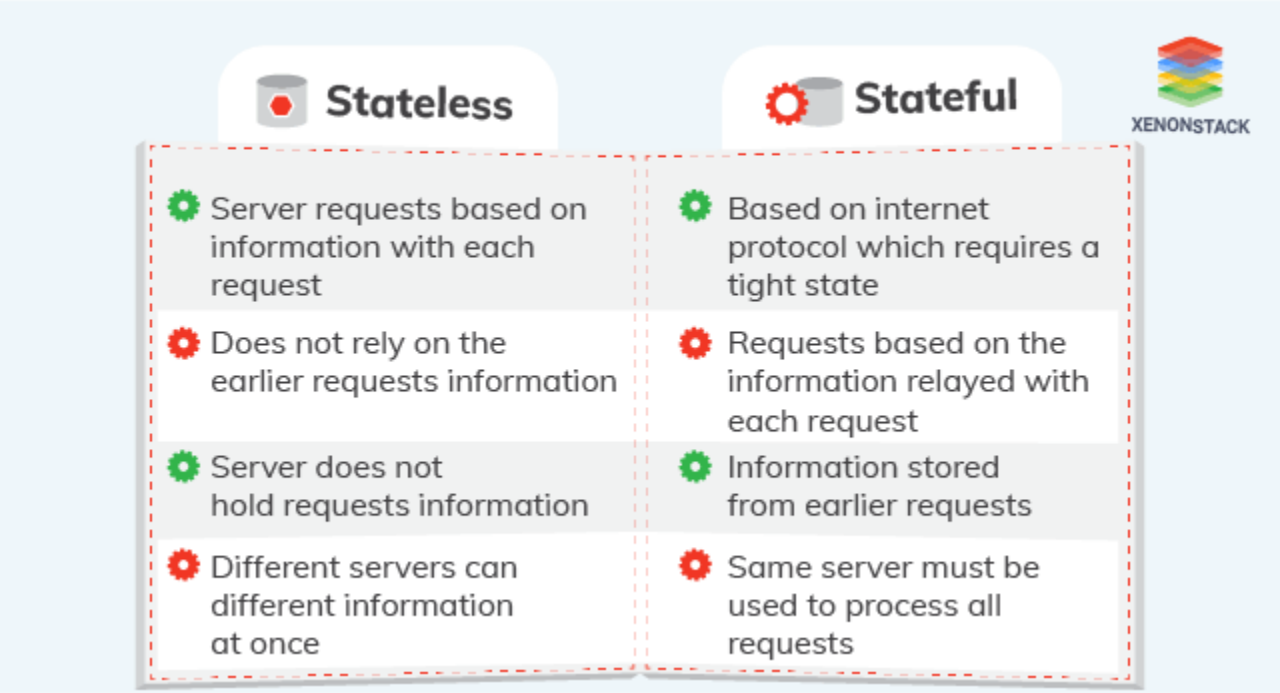
cURL requests

```
C:\Users\Reese Pearsall>curl --location --request GET https://v2.jokeapi.dev/joke/Any?safe-mode
{
    "error": false,
    "category": "Spooky",
    "type": "twopart",
    "setup": "Why do ghosts go on diets?",
    "delivery": "So they can keep their ghoulish figures.",
    "flags": {
        "nsfw": false,
        "religious": false,
        "political": false,
        "racist": false,
        "sexist": false,
        "explicit": false
    },
    "safe": true,
    "id": 295,
    "lang": "en"
}
C:\Users\Reese Pearsall>
```

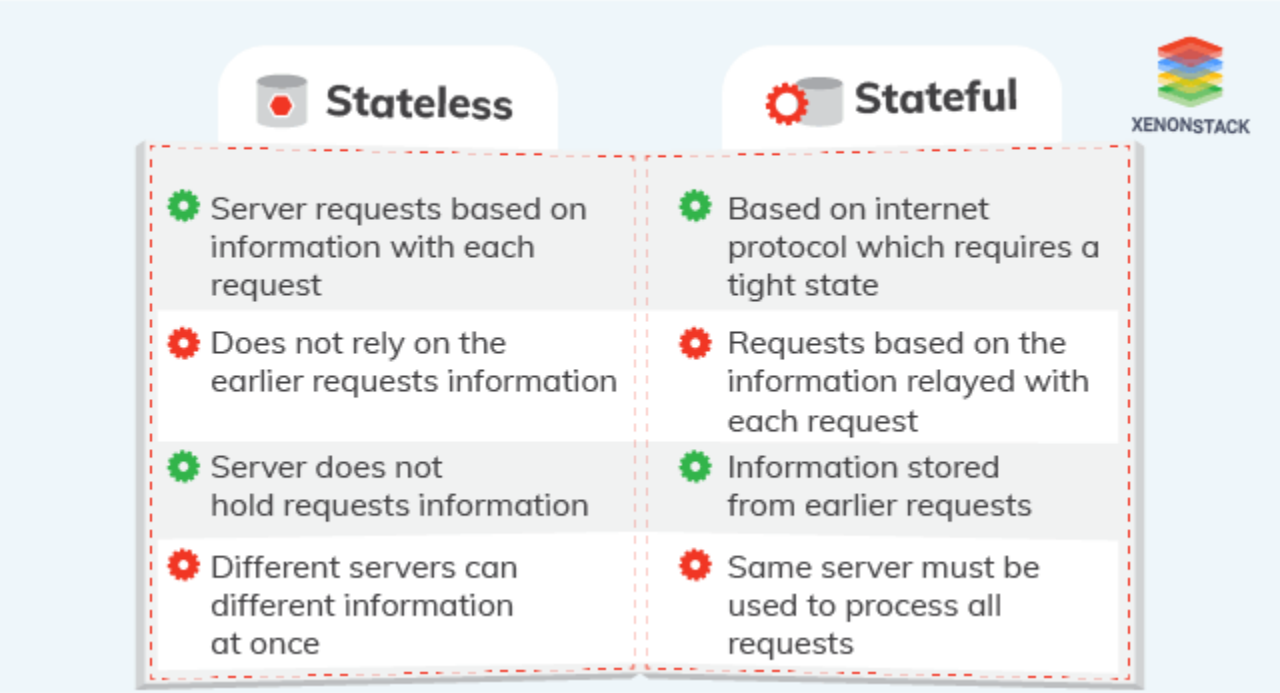
Postman


Postman
New
Import
Runner
Examples
Invite
Team
Filter
History
Collections
APIs BETA
New API
Swagger Petstore
Swagger Pet Store
Swagger Petstore
Draft
Show All Versions
Refresh
Reports
Comments
Share
The Chrome App is Deprecated. Download Postman's Native Apps for macOS, Windows, and Linux, and get the latest features!
Descriptions support Markdown
Cancel
Save
Define
Develop
Test
Observe
Schema
OpenAPI 3.0
Generate Collection
JSON
4 title: Swagger Petstore
5 license:
6 name: MIT
7 servers:
8 - url: http://petstore.swagger.io/v1
9 paths:
10 /pets:
11 get:
12 summary: List all pets
13 operationId: listPets
14 tags:
15 - pets
16 parameters:
17 - name: limit
18 in: query
19 description: How many items to return at one time (max 100)
20 required: false
Bootcamp
Build
Browse

HTTP is a **stateless** protocol; it does not save information about the sender-receiver session and treats each user as a brand-new connection



HTTP is a **stateless** protocol; it does not save information about the sender-receiver session and treats each user as a brand-new connection



often it can be useful to identify a user.

- User access and permissions
- Dynamic continent

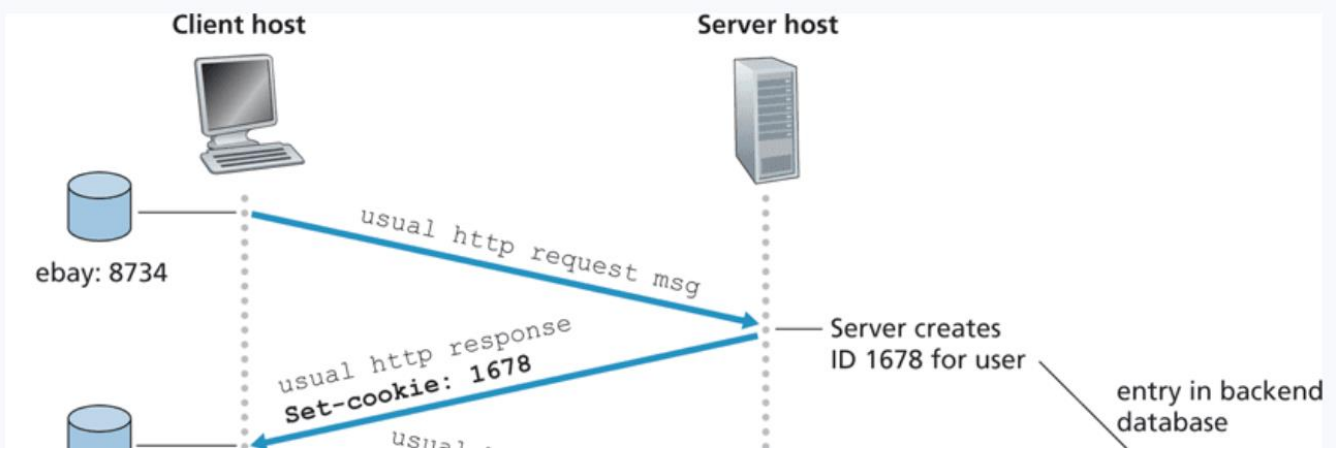
Cookies are pieces of information that are exchanged between browsers and web servers to identify users in active connections

Cookies are pieces of information that are exchanged between browsers and web servers to identify users in active connections

- Authentication
- Tracking & Advertisement
- Session Management

Cookies in HTTP

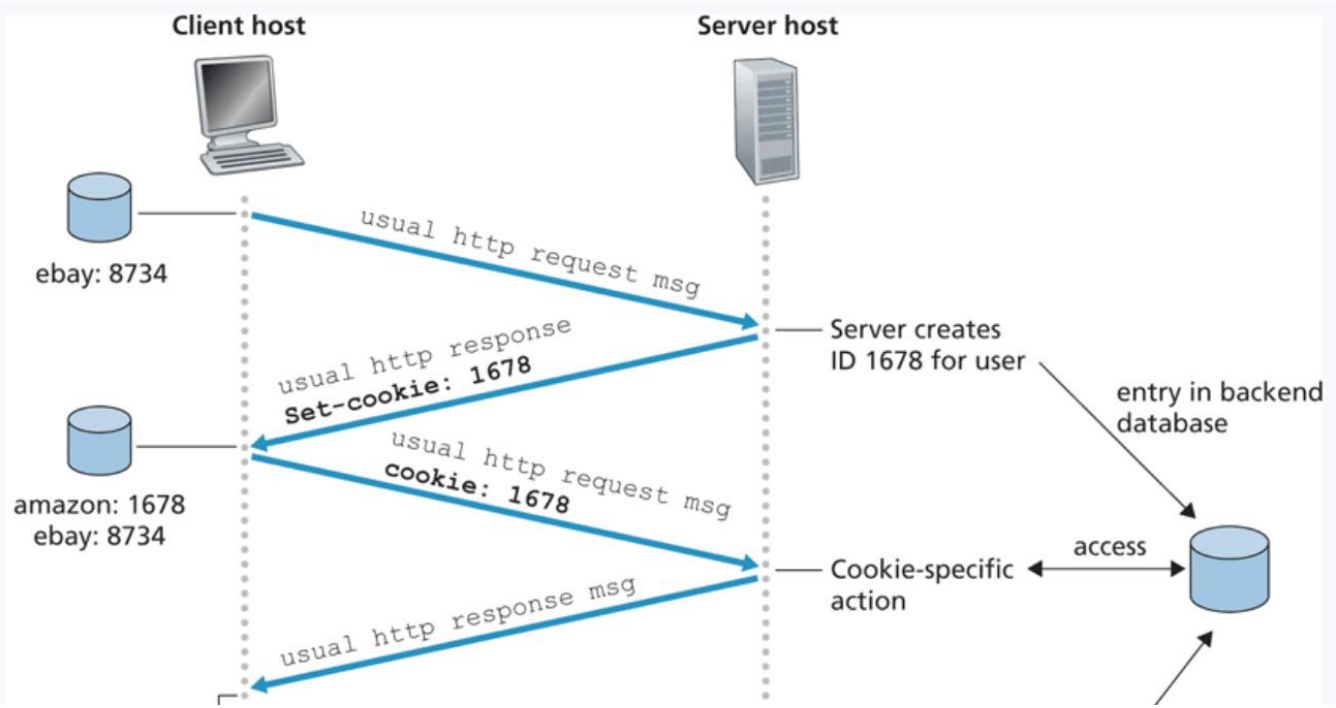
After sending a request to a webserver for the first time, the user is assigned a unique cookie value



Cookies in HTTP

After sending a request to a webserver for the first time, the user is assigned a unique cookie value

Cookie 1678 is created and stored in the user's browser as well as some database backend on the server side

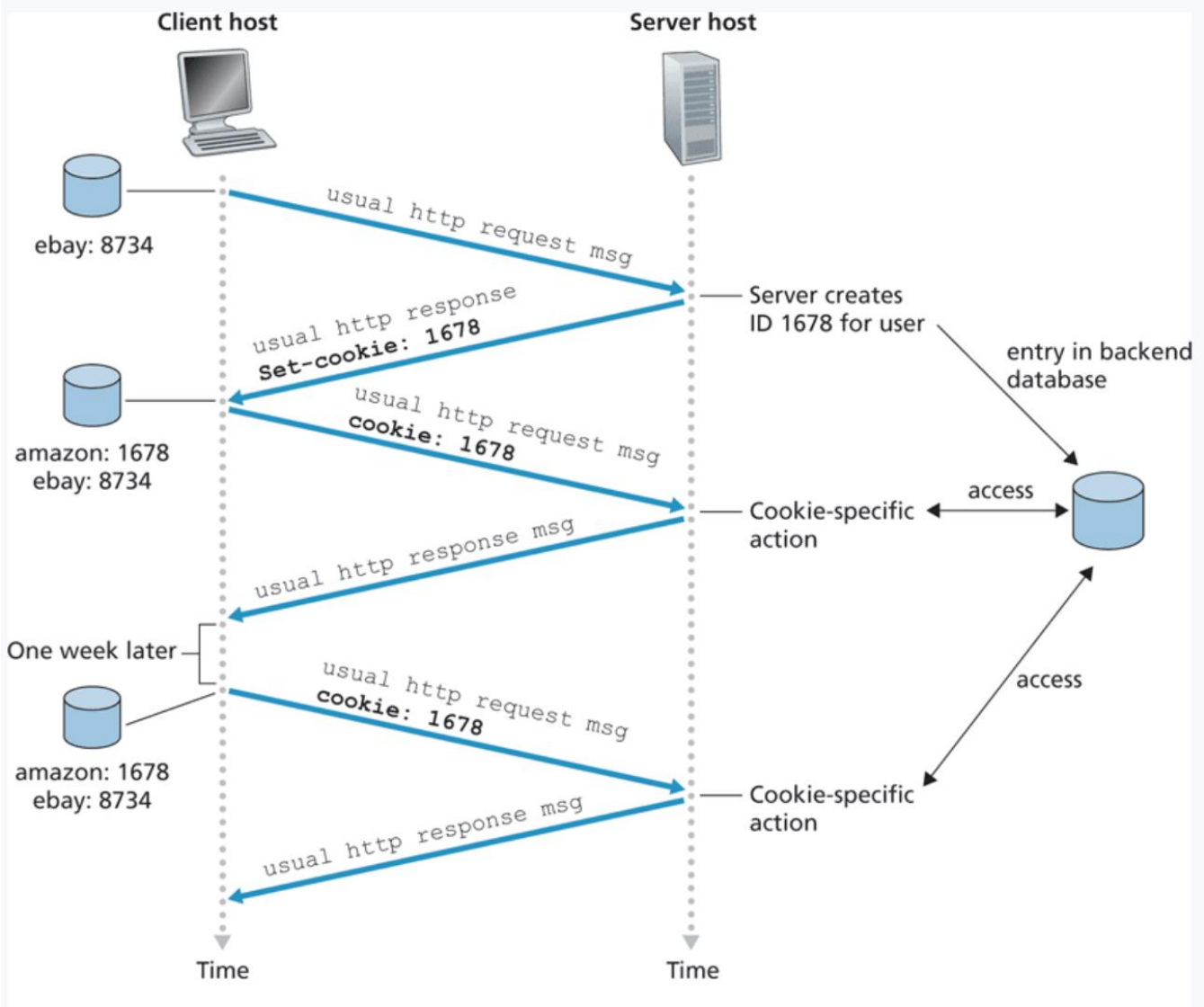


Cookies in HTTP

After sending a request to a webserver for the first time, the user is assigned a unique cookie value

Cookie 1678 is created and stored in the user's browser as well as some database backend on the server side

When the user goes back to visit the website, the cookie will be exchanged between to client and host so the website can execute cookie-specific actions

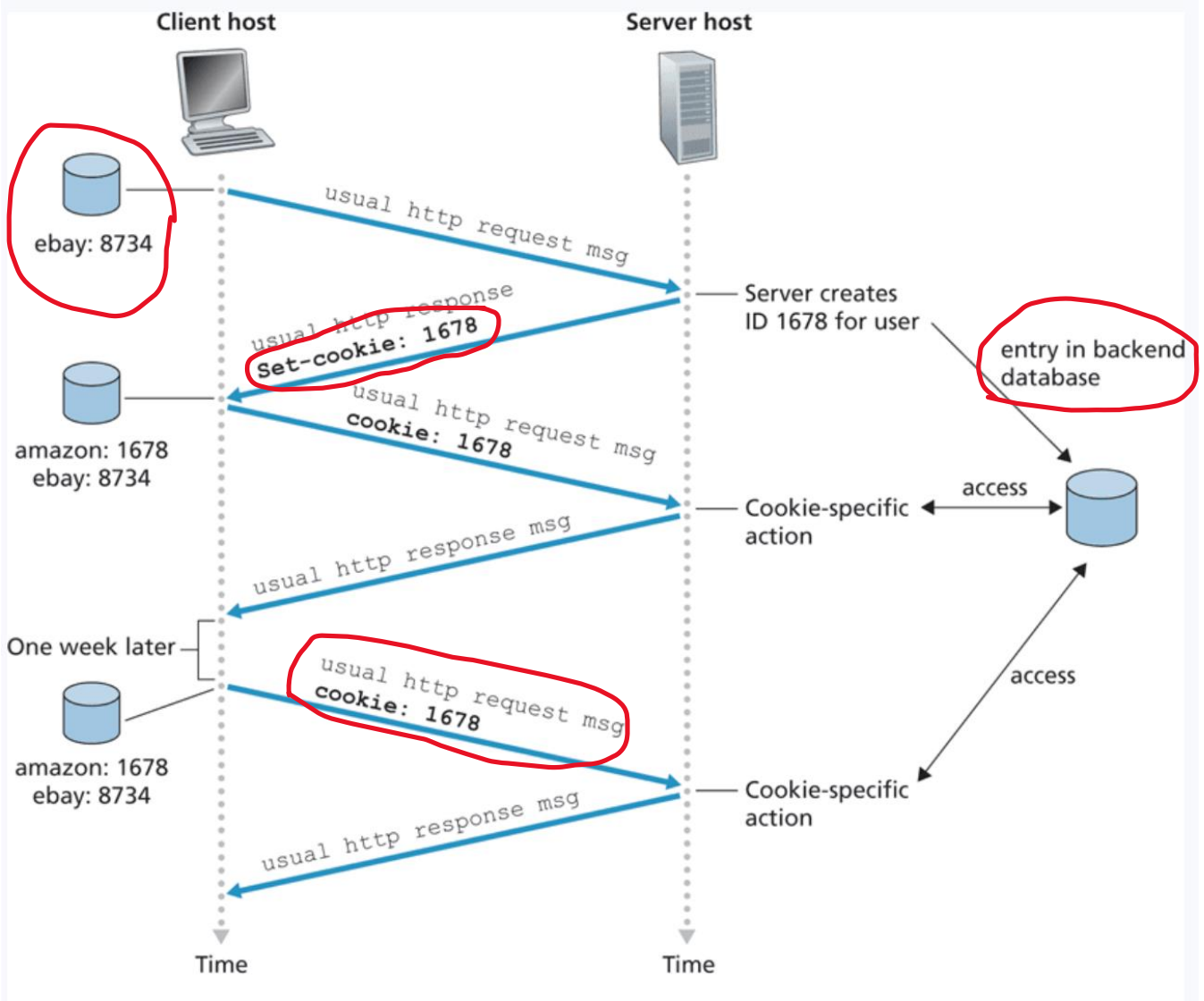


Cookies in HTTP

After sending a request to a webserver for the first time, the user is assigned a unique cookie value

Cookie 1678 is created and stored in the user's browser as well as some database backend on the server side

When the user goes back to visit the website, the cookie will be exchanged between to client and host so the website can execute cookie-specific actions



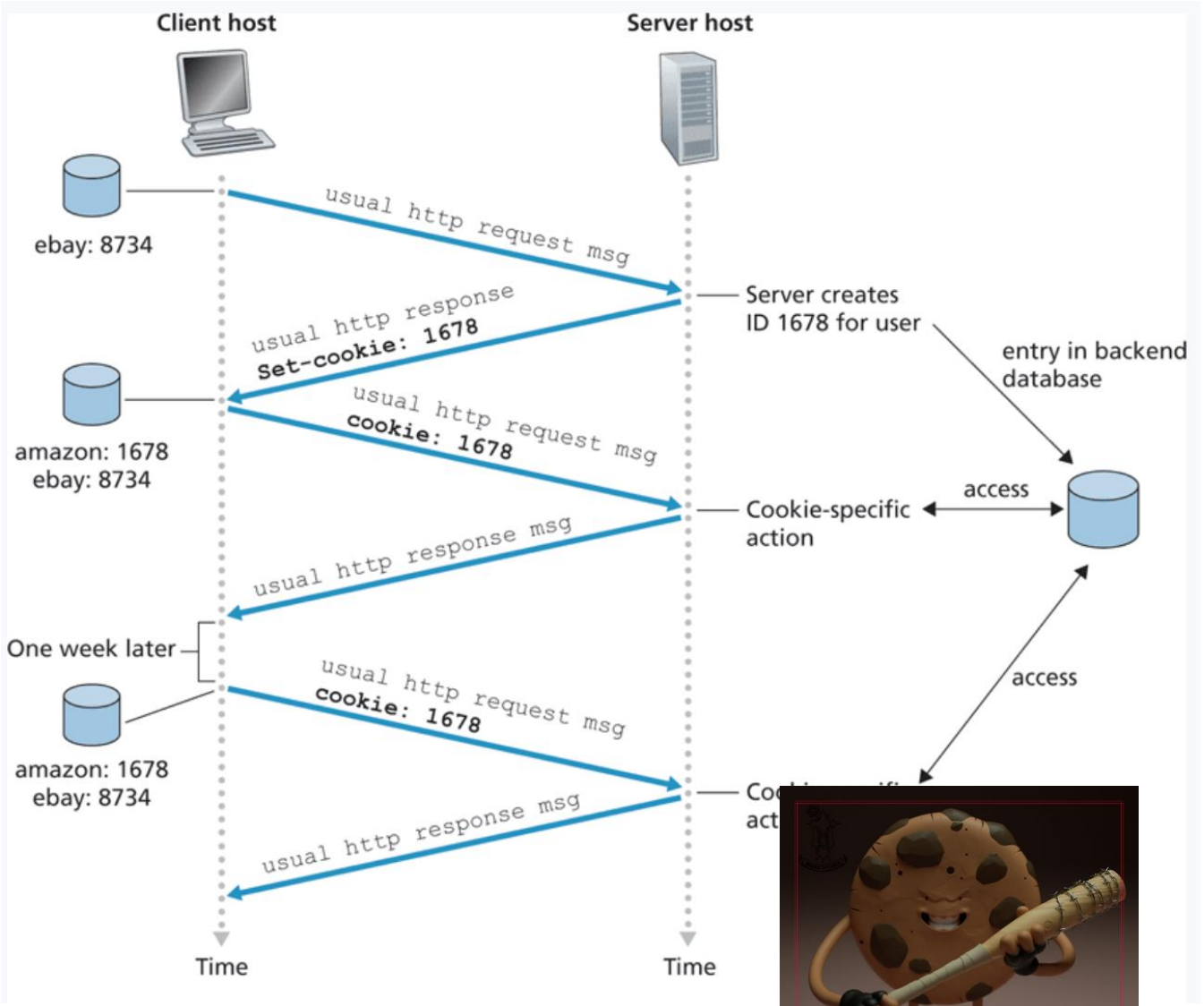
Cookie technology consists of four main components

Cookies in HTTP

After sending a request to a webserver for the first time, the user is assigned a unique cookie value

Cookie 1678 is created and stored in the user's browser as well as some database backend on the server side

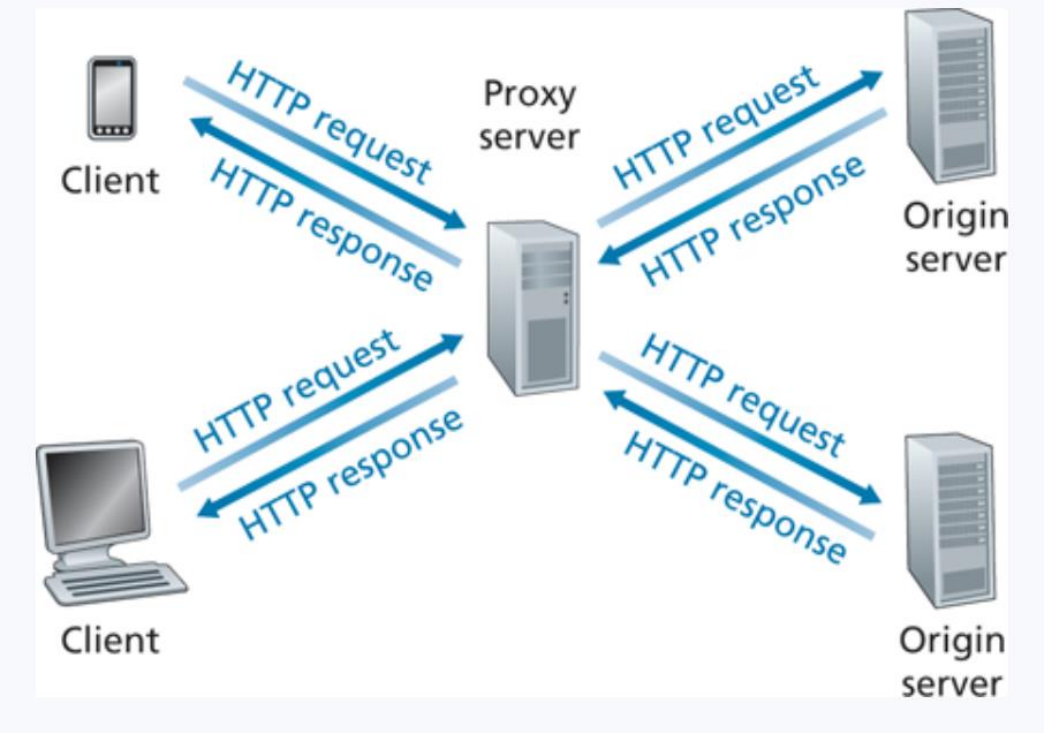
When the user goes back to visit the website, the cookie will be exchanged between to client and host so the website can execute cookie-specific actions



There is lots of controversy regarding the privacy and morality of tracking user information with cookies

Web Caching

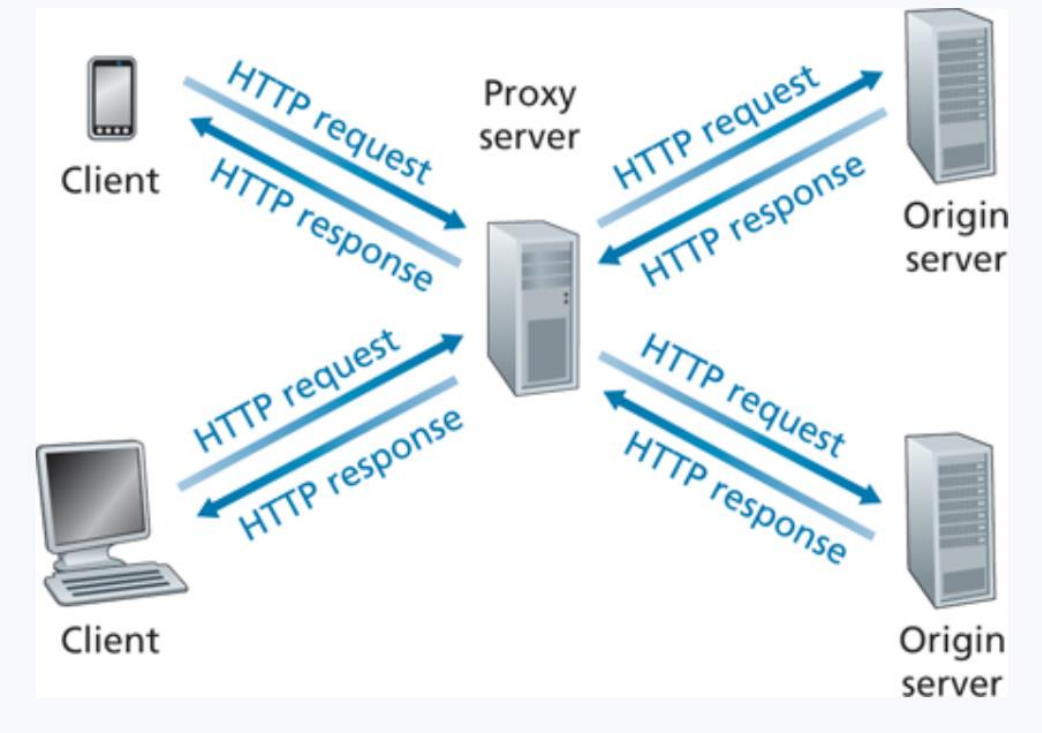
A **web cache**– also called a **proxy server**– is a network entity that satisfies HTTP requests on the behalf of an origin Web server



Web Caching

A **web cache**– also called a **proxy server**– is a network entity that satisfies HTTP requests on the behalf of an origin Web server

1. Browser/Client establishes a TCP connection to the Web cache and sends an HTTP request

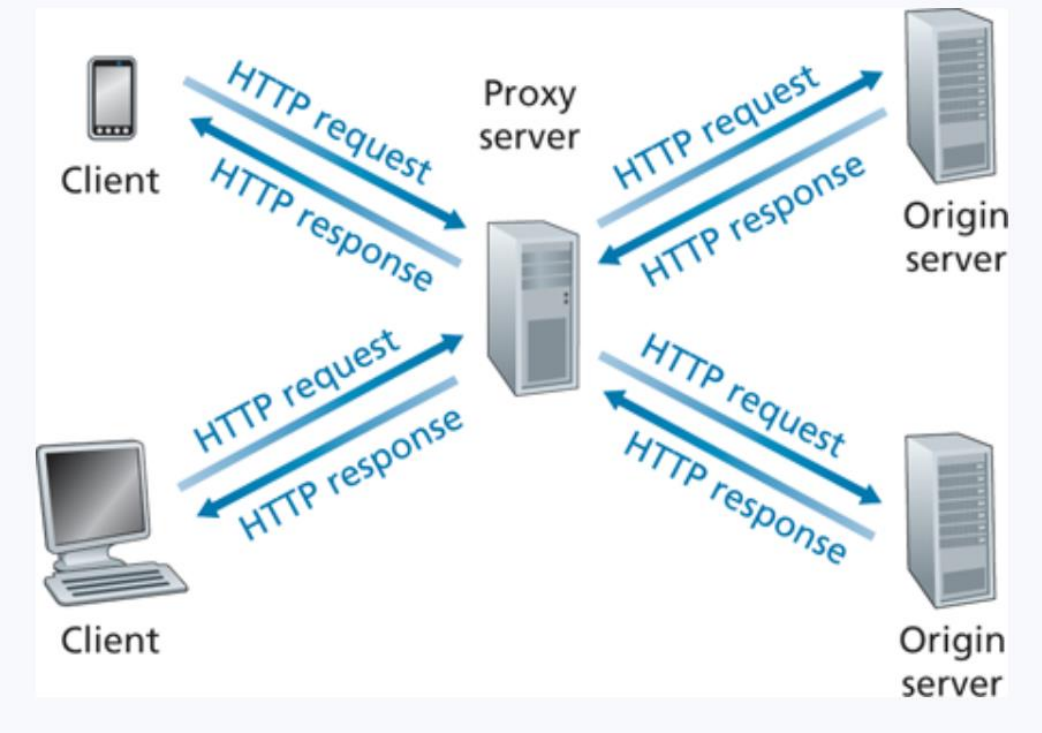


Web Caching

A **web cache**– also called a **proxy server**– is a network entity that satisfies HTTP requests on the behalf of an origin Web server

1. Browser/Client establishes a TCP connection to the Web cache and sends an HTTP request

2. Web cache checks its local storage for the requested object



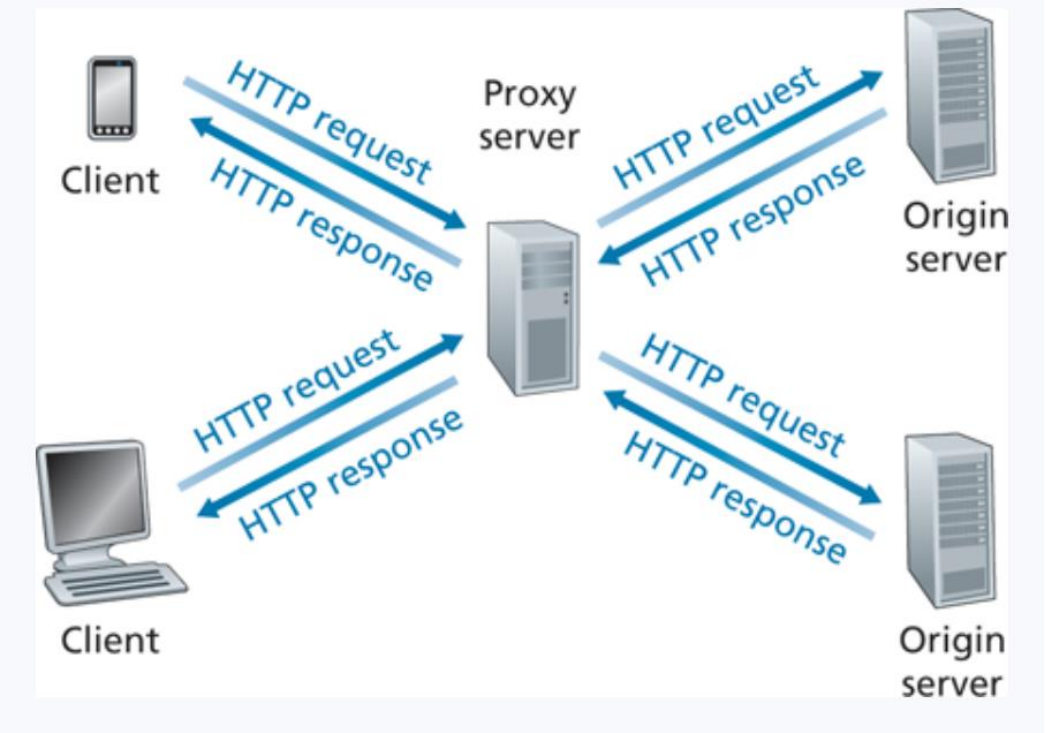
Web Caching

A **web cache**– also called a **proxy server**– is a network entity that satisfies HTTP requests on the behalf of an origin Web server

1. Browser/Client establishes a TCP connection to the Web cache and sends an HTTP request

2. Web cache checks its local storage for the requested object

3. If the web cache does not have the object, establish TCP with an origin server and issue an HTTP request



Web Caching

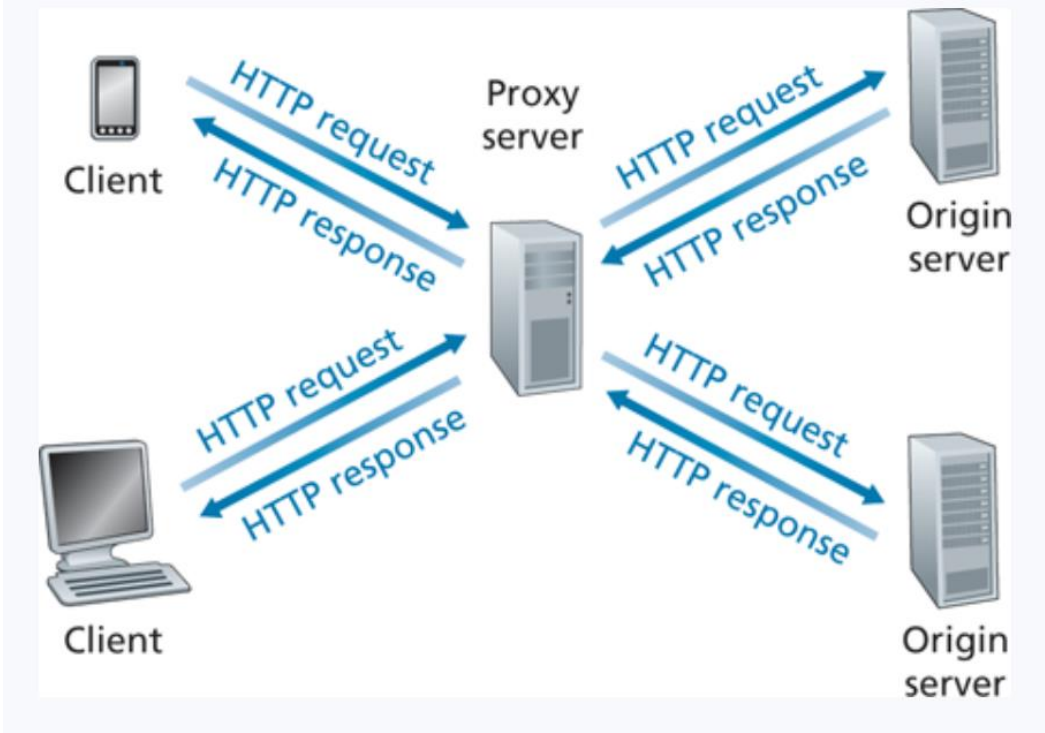
A **web cache**– also called a **proxy server**– is a network entity that satisfies HTTP requests on the behalf of an origin Web server

1. Browser/Client establishes a TCP connection to the Web cache and sends an HTTP request

2. Web cache checks its local storage for the requested object

3. If the web cache does not have the object, establish TCP with an origin server and issue an HTTP request

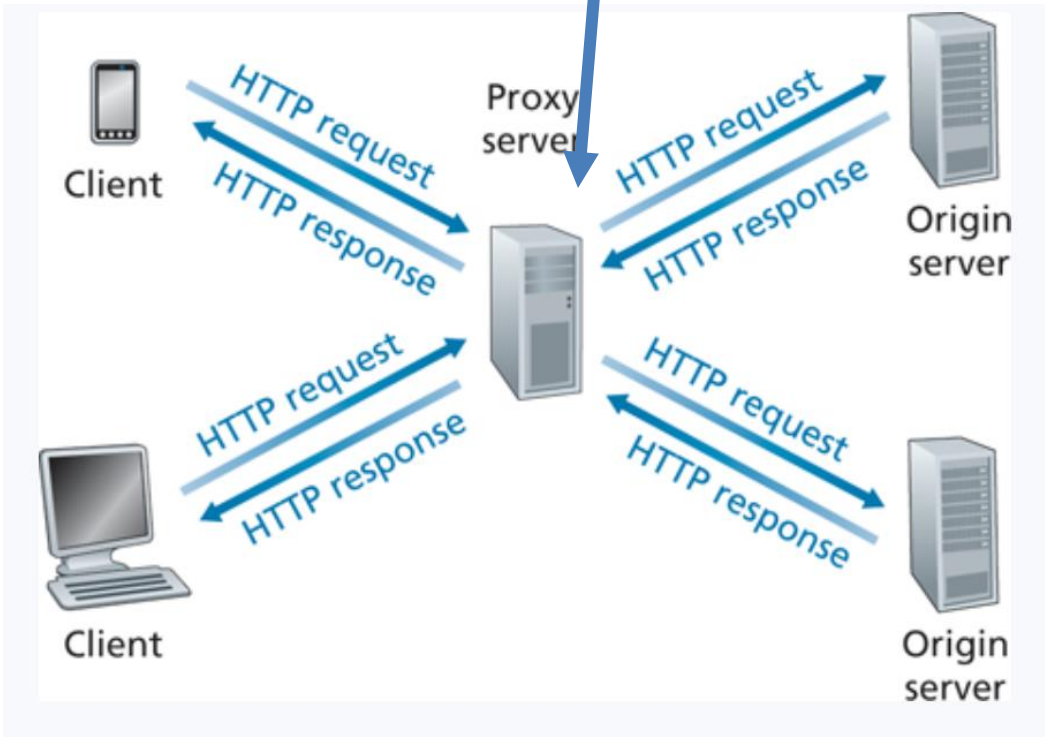
4. Web cache stores a local copy of the object, then issues an HTTP response with the object



Web Caching

A **web cache**– also called a **proxy server**– is a network entity that satisfies HTTP requests on the behalf of an origin Web server

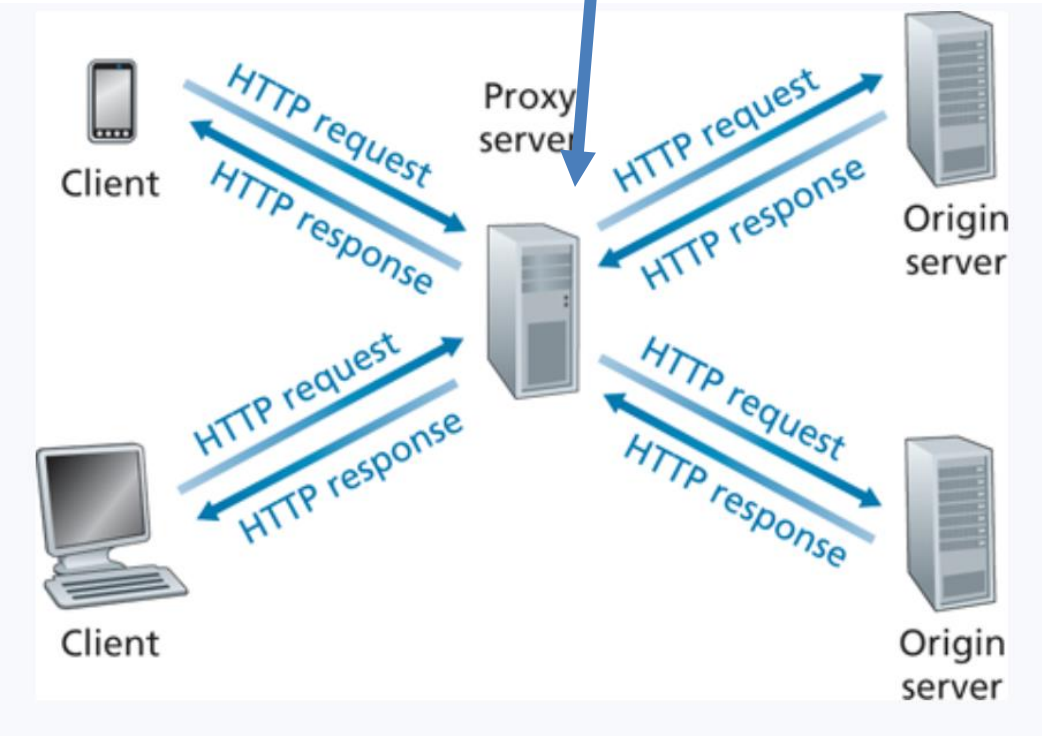
Typically installed and maintained by an ISP



Web Caching

A **web cache**– also called a **proxy server**– is a network entity that satisfies HTTP requests on the behalf of an origin Web server

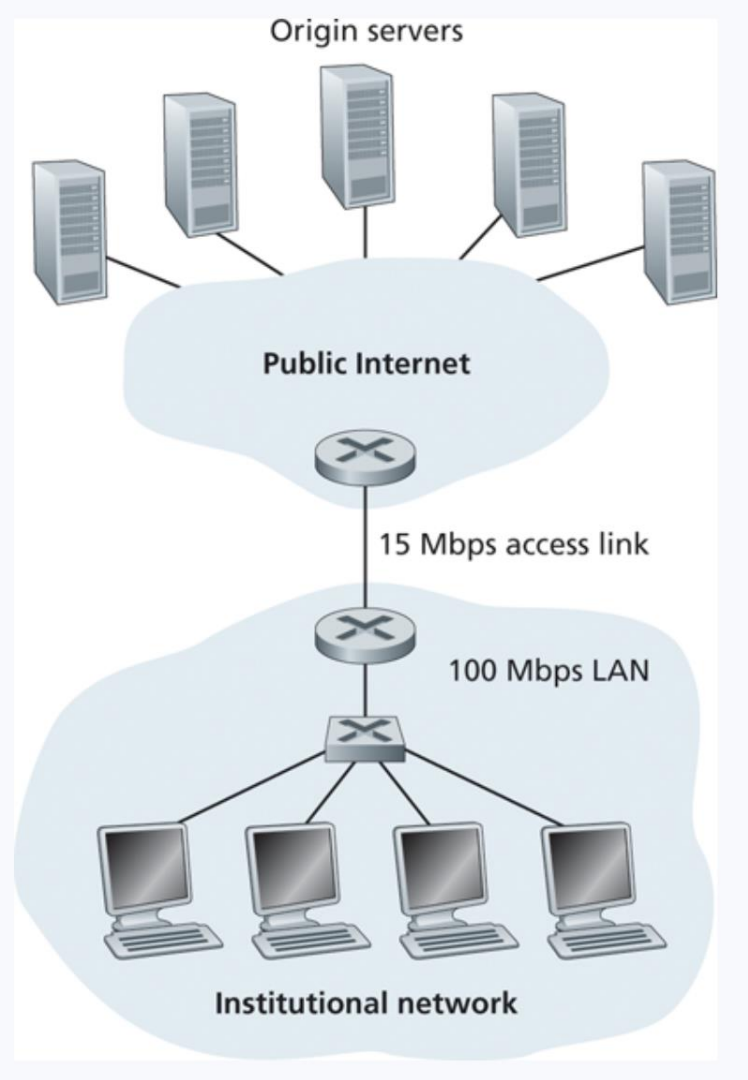
Typically installed and maintained by an ISP

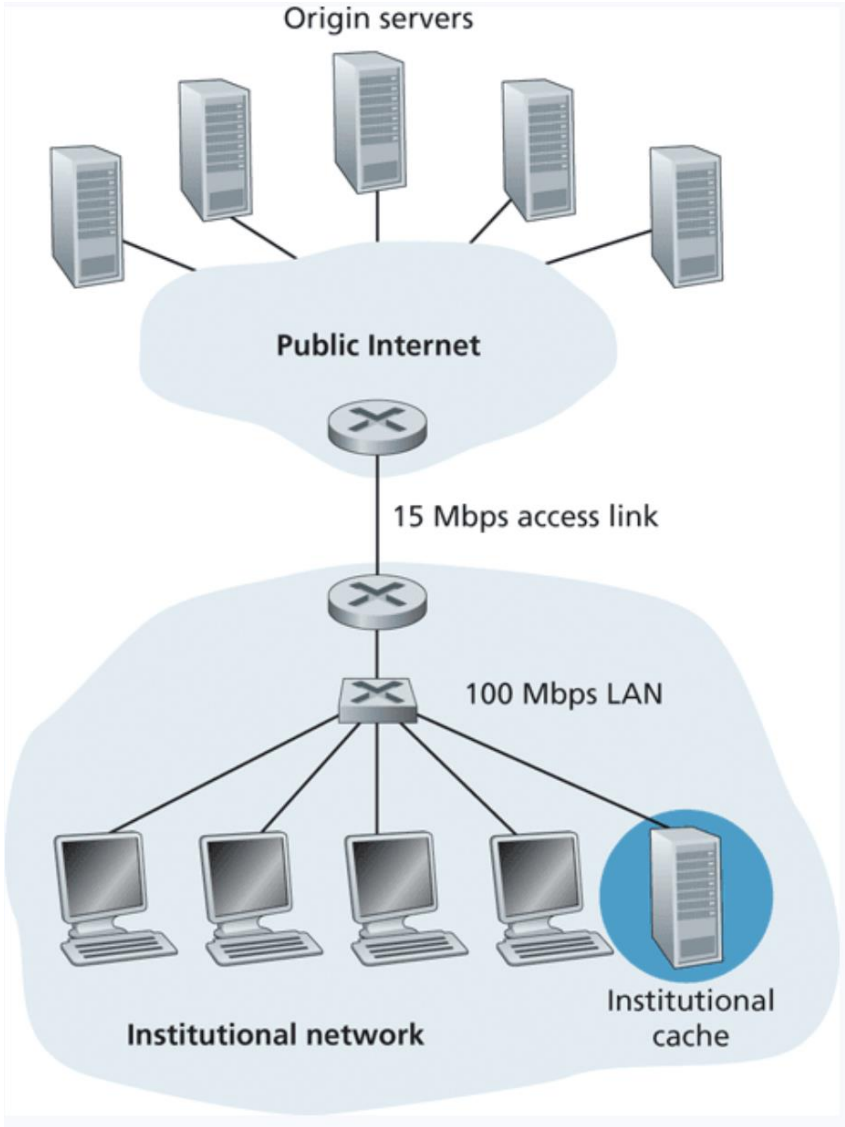


* Improves response time (especially if the the cache has the object that is requested)

* The connection from the client to the cache is typically much faster than the connection from client to host server

Web Caching

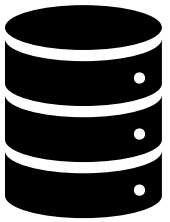

Origin servers
Public Internet
15 Mbps access link
100 Mbps LAN
Institutional network


Origin servers
Public Internet
15 Mbps access link
100 Mbps LAN
Institutional network
Institutional cache

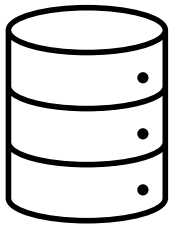
Web Caching

The cache might not always have the most up to date version in its local storage. “Stale” objects

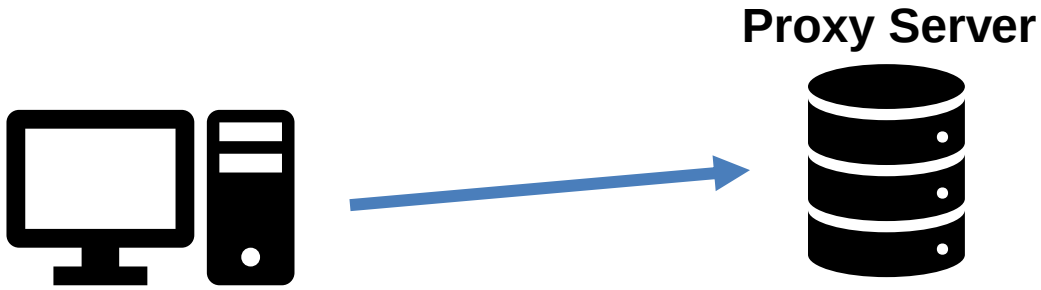
Web Caching



The cache might not always have the most up to date version in its local storage. “Stale” objects

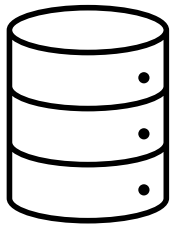


Web Caching

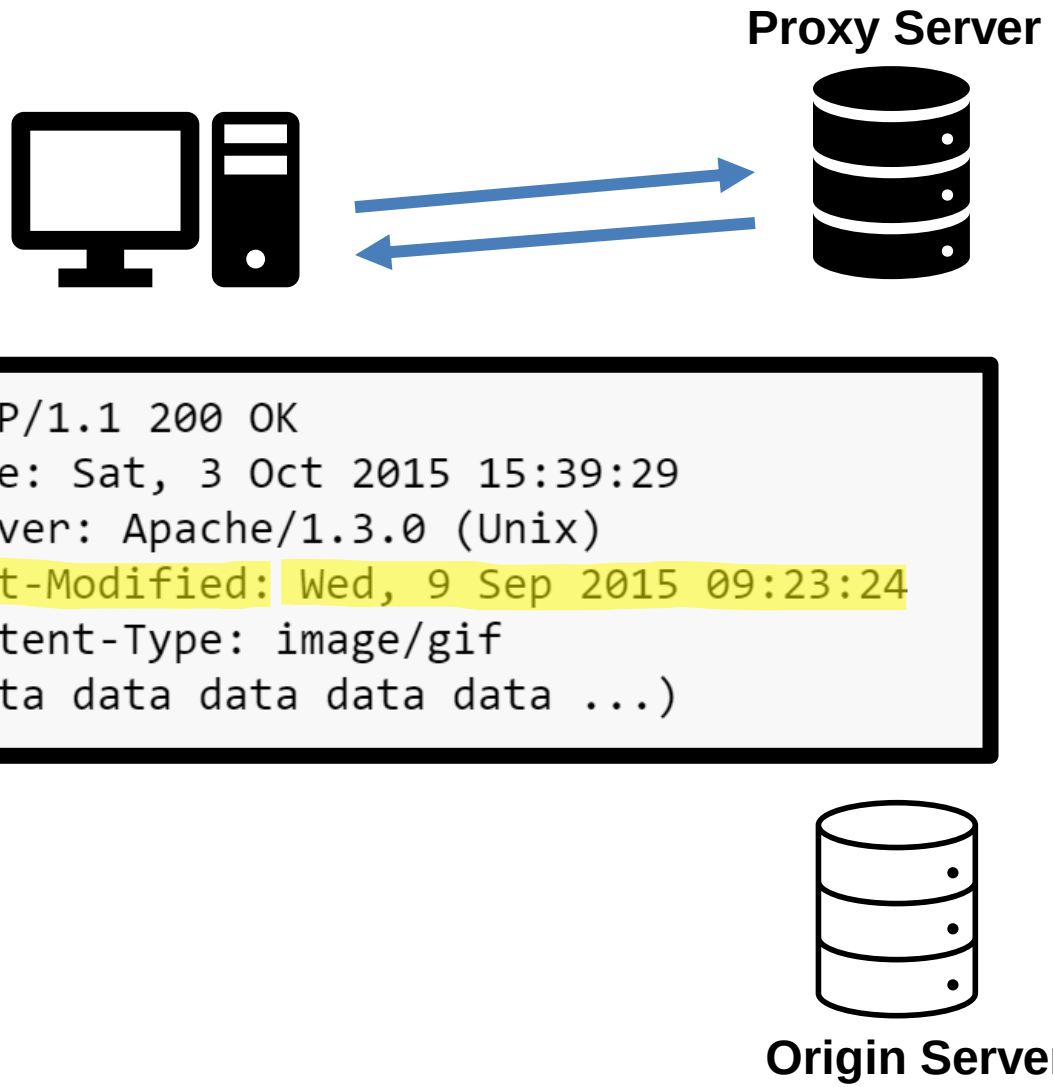


The cache might not always have the most up to date version in its local storage. “Stale” objects

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
```

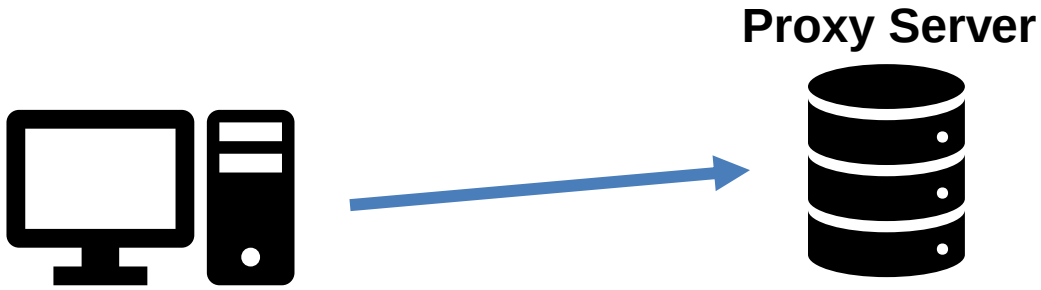


Web Caching



The cache might not always have the most up to date version in its local storage. “Stale” objects

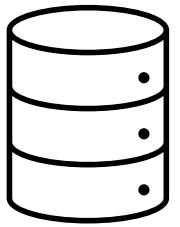
Web Caching



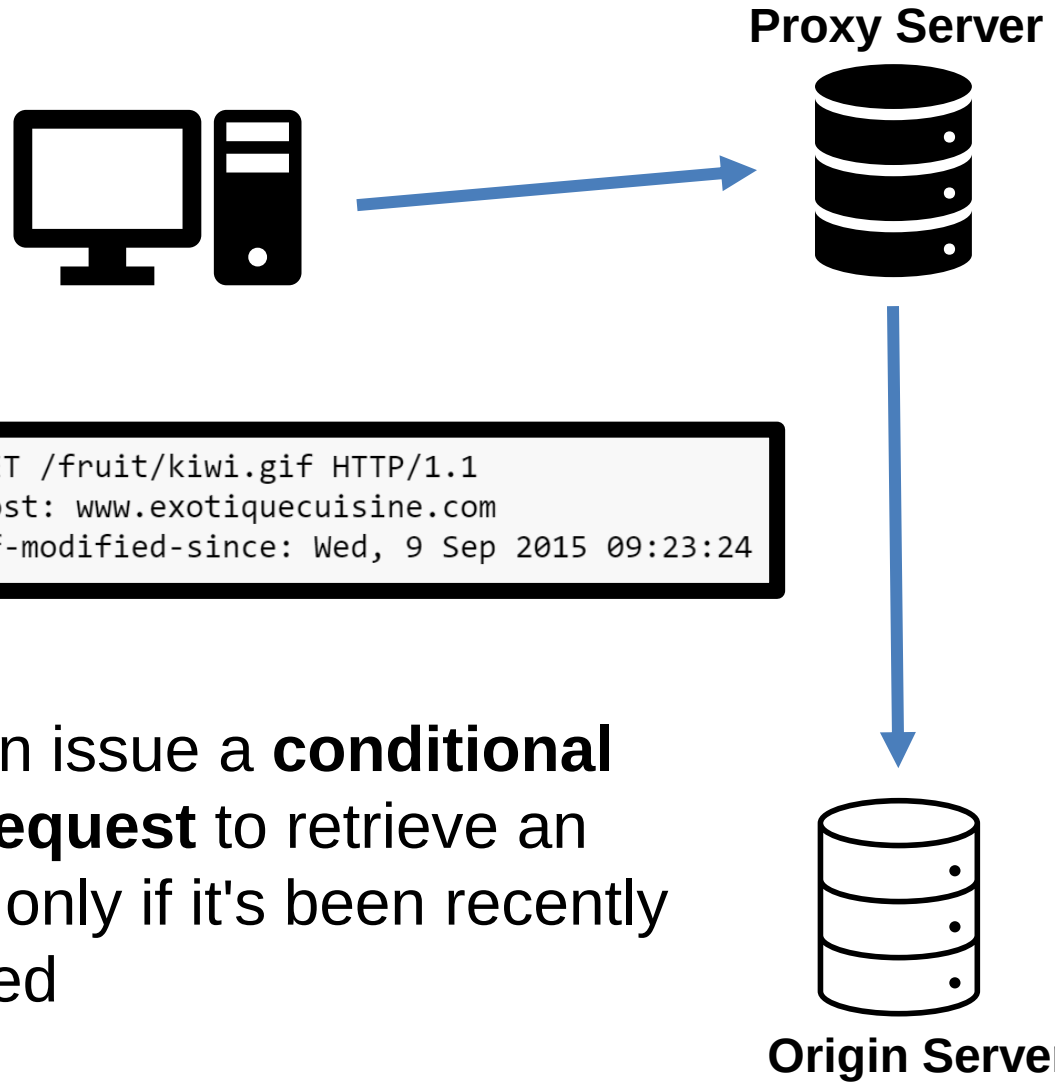
The cache might not always have the most up to date version in its local storage. “Stale” objects

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
If-modified-since: Wed, 9 Sep 2015 09:23:24
```

We can issue a **conditional GET request** to retrieve an object only if it's been recently modified



Web Caching

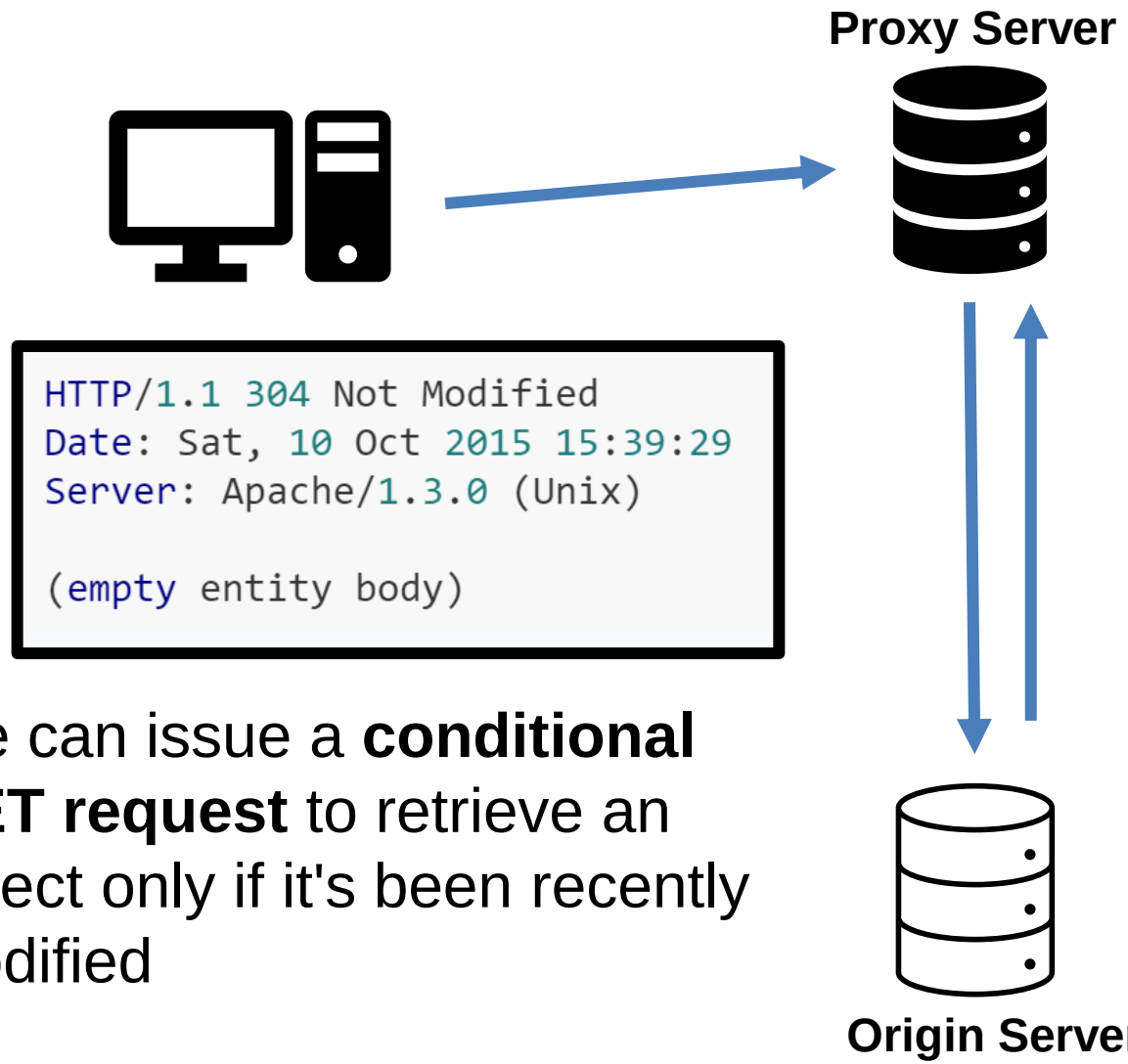


The cache might not always have the most up to date version in its local storage. “Stale” objects

We can issue a **conditional GET request** to retrieve an object only if it's been recently modified

Web Caching

The cache might not always have the most up to date version in its local storage. “Stale” objects



We can issue a **conditional GET request** to retrieve an object only if it's been recently modified

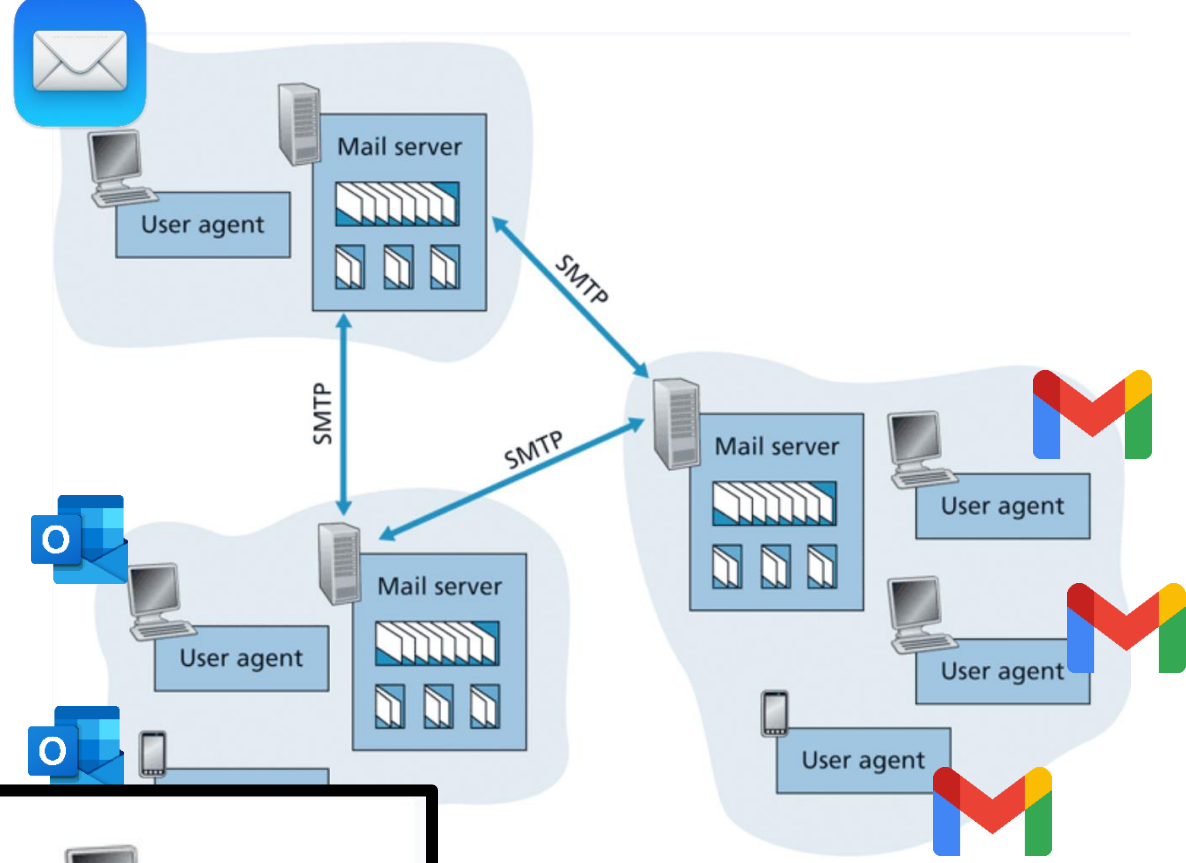
SMTP

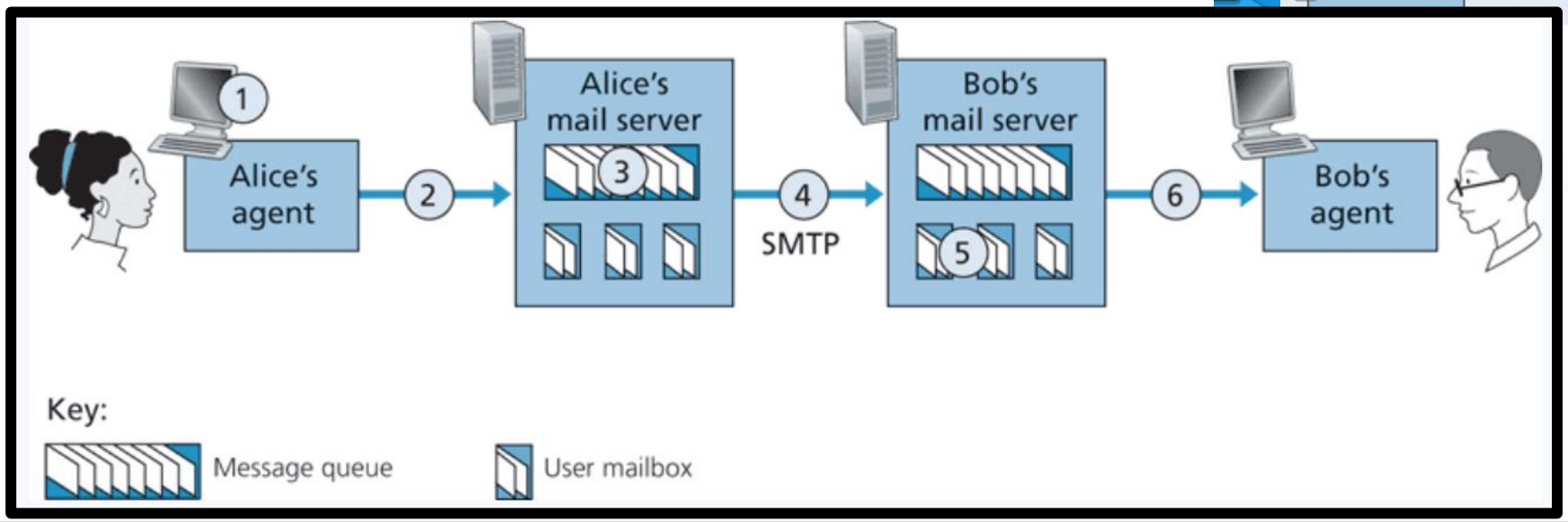
Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another

SMTP

Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another

This is not a protocol for *retrieving* emails

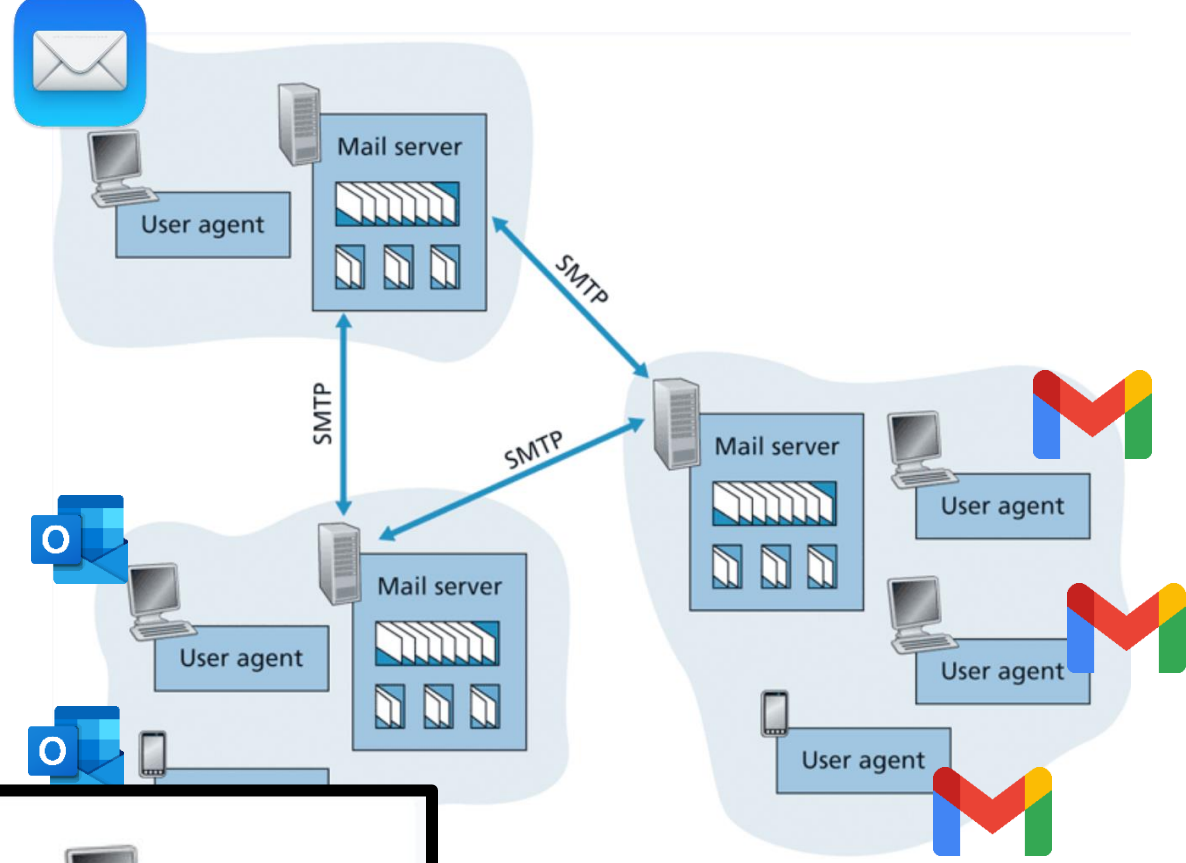


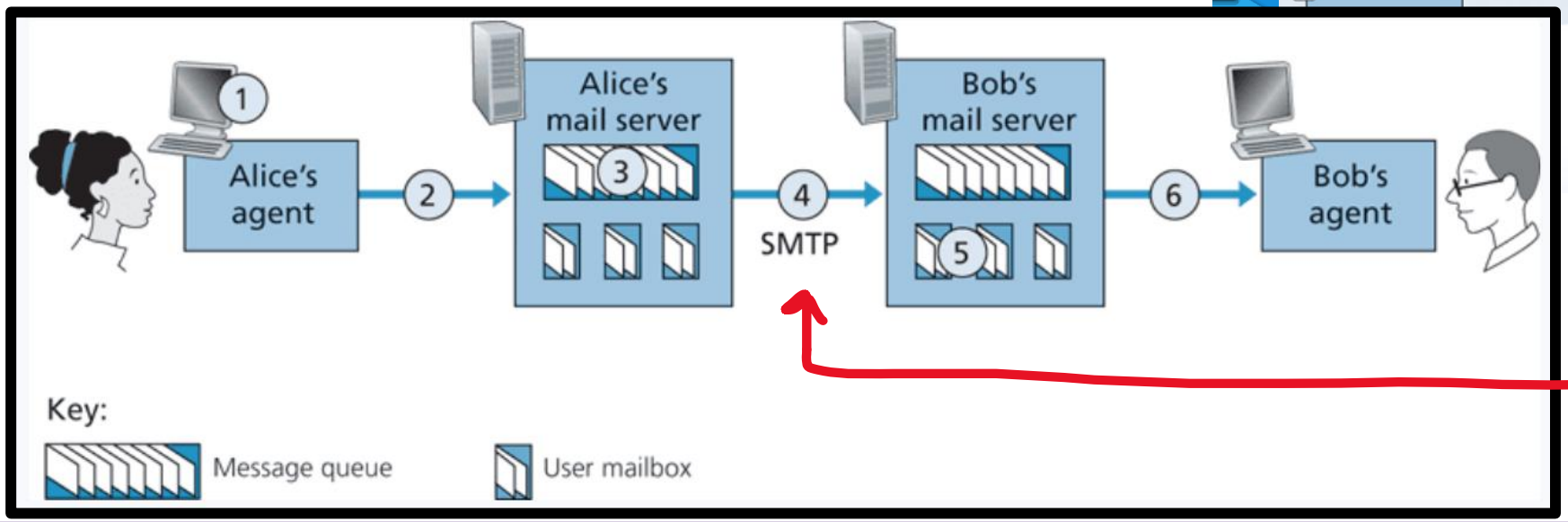


SMTP

Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another

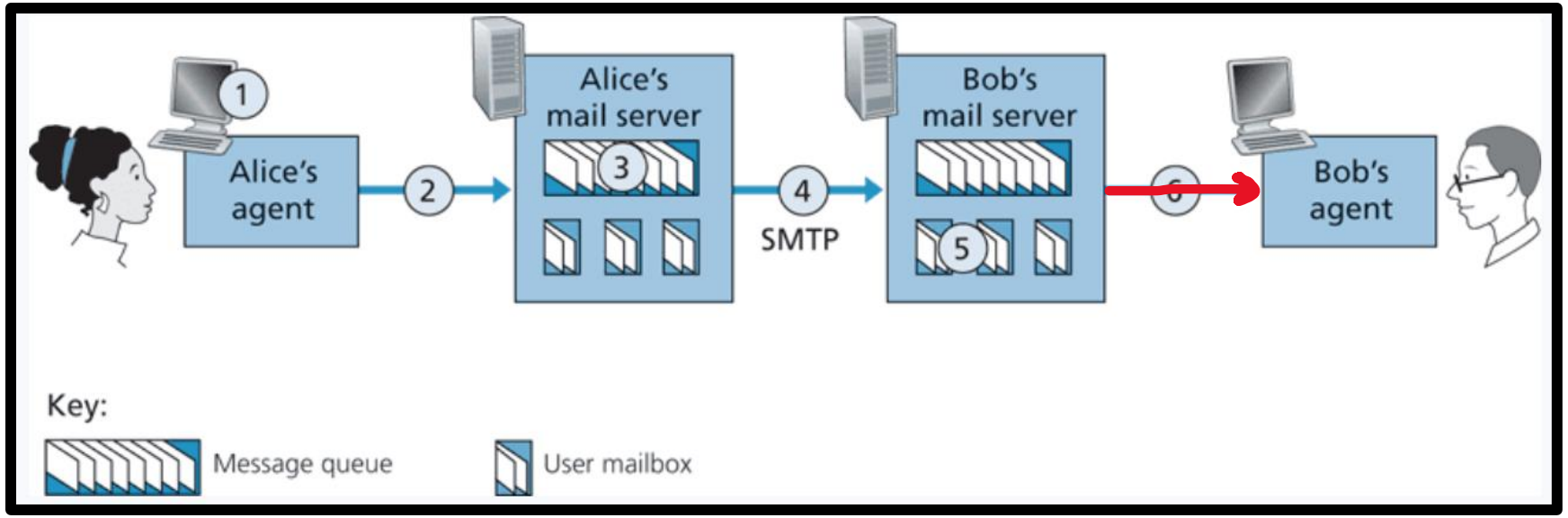
This is not a protocol for *retrieving* emails





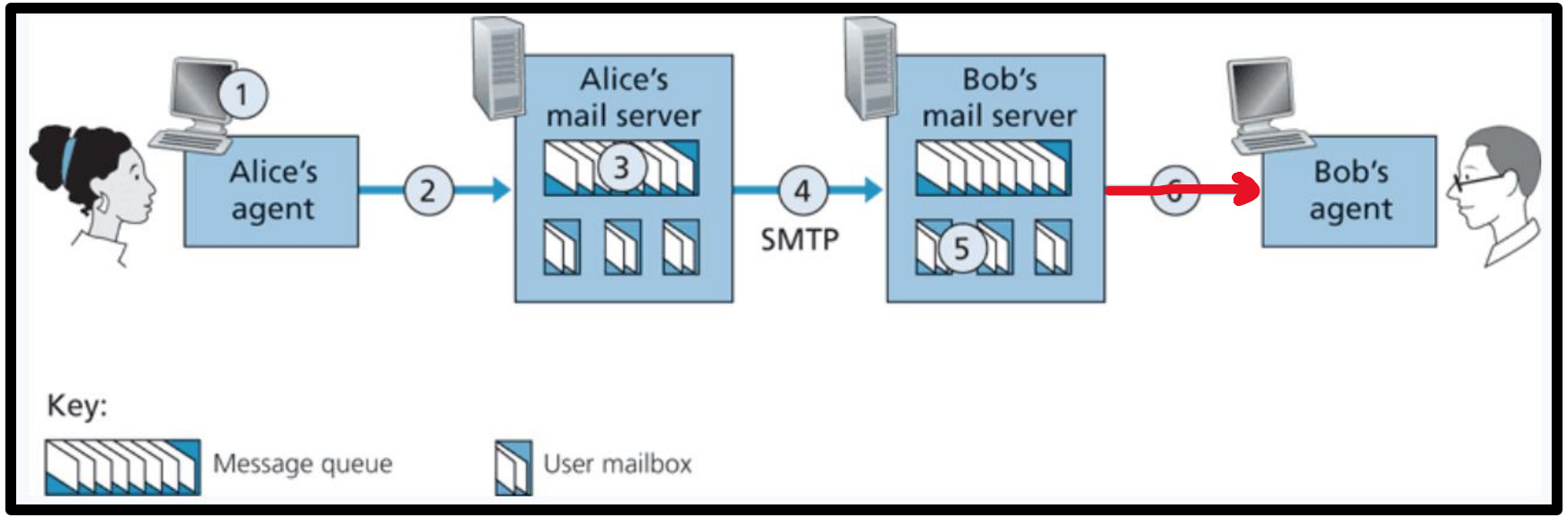
STMP uses TCP for the end-to-end delivery

SMTP



POP3 (post office protocol) or **IMAP** (internet message access protocol) are used to retrieve emails from mail servers.

SMTP



POP3 (post office protocol) or **IMAP** (internet message access protocol) are used to retrieve emails from mail servers.

POP3 deletes the email of the web server, IMAP maintains a copy to synchronize across multiple devices

SMTP

(Very verbose)

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr ... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with ”.” on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

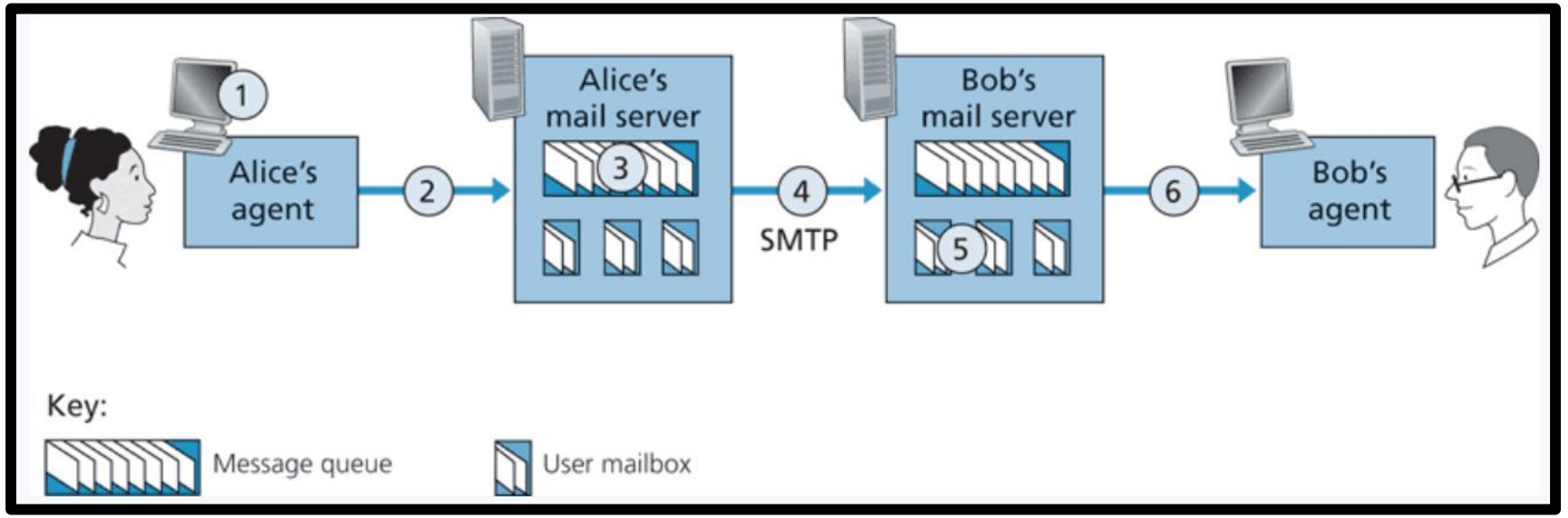
Announcements

Homework 1 due TONIGHT @ 11:59 PM

SMTP

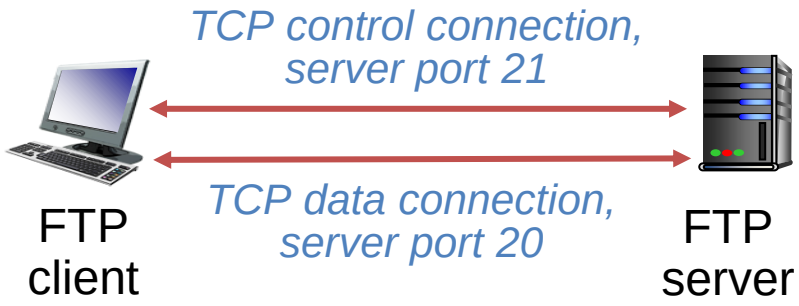
Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another *asynchronously*

Ports 25 is reserved for SMTP traffic (and also port 587 & 465)



FTP

File Transfer Protocol (FTP)- protocol used for transferring files from server to client



- FTP communicates over two connections
  - Port 21 for control information
  - Port 20 for data
- Differences from HTTP
  - Control communication “out-of-band”
  - Server maintains per client state: authentication, current directory

- **FTP procedure:**
  1. FTP client contacts FTP server at port 21, using TCP
  2. Client authorized over control connection
  3. Client browses remote directory, sends commands over control connection
  4. When server receives file transfer command, server opens 2nd TCP data connection (for file) to client
  5. After transferring one file, server closes data connection

Why use a separate control connection?

DNS

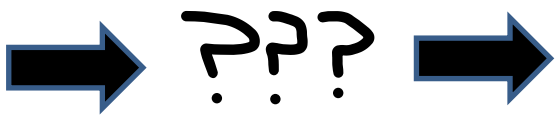
Humans browse the web using hostnames
- (They need English)

Computers understand numbers
- (They need IP addresses)

DNS

Humans browse the web using hostnames
- (They need English)

Computers understand numbers
- (They need IP addresses)



153.90.127.197

DNS

Humans browse the web using hostnames
- (They need English)

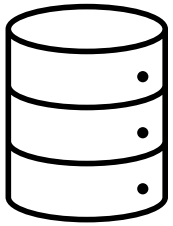
Computers understand numbers
- (They need IP addresses)

DNS

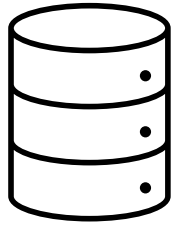
153.90.127.197

Domain Name System (DNS) is a database of mappings between hostnames and IP addresses

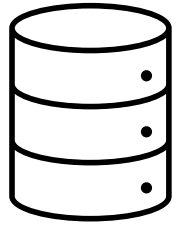
DNS Architecture



montana.edu



youtube.com



irs.gov

Clients will first need to resolve hostnames with DNS

Clients

HTTP
153.90.3.95

3.

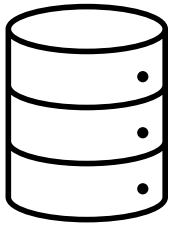
1.

montana.edu ?

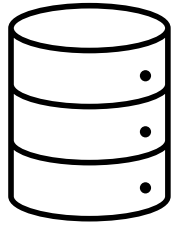
153.90.3.95

2.

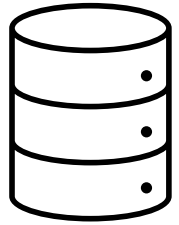
DNS Architecture



montana.edu



youtube.com



irs.gov

Clients will first need to resolve hostnames with DNS

HTTP
153.90.3.95

3.

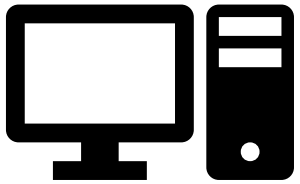
1.

montana.edu ?

153.90.3.95

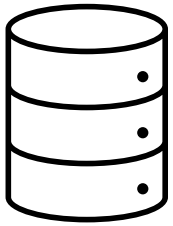
2.

Is this a good architecture?

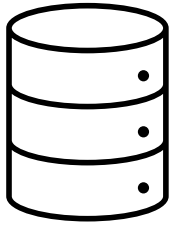


Clients

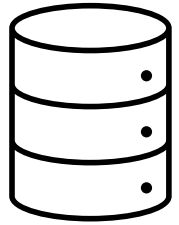
DNS Architecture



montana.edu



youtube.com



irs.gov

Clients will first need to resolve hostnames with DNS

HTTP
153.90.3.95

3.

1.

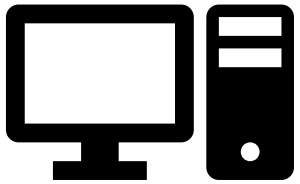
montana.edu ?

153.90.3.95

2.

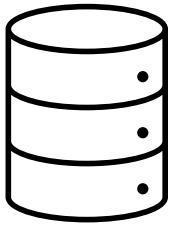
Single point of failure…

Is this a good architecture?

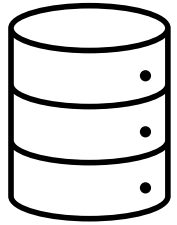


Clients

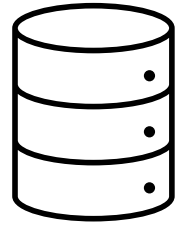
DNS Architecture



montana.edu



youtube.com



irs.gov

.com

.gov

.edu

Clients

DNS Architecture

(how big would that map be?)

- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

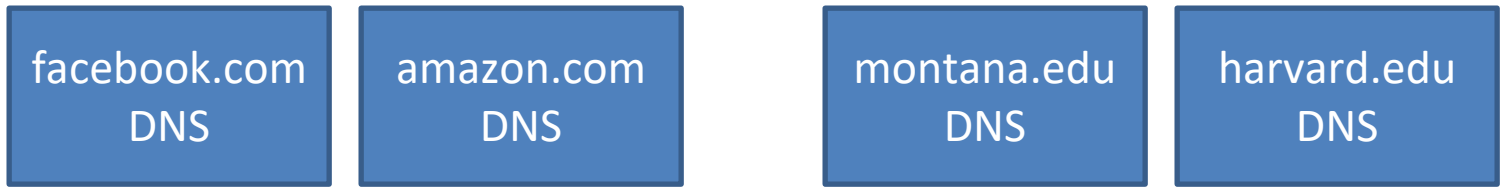
Hierarchy consists of different types of DNS servers:

DNS Architecture

- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

Hierarchy consists of different types of DNS servers:

Authoritative DNS servers- Organization's own DNS with up-to-date records



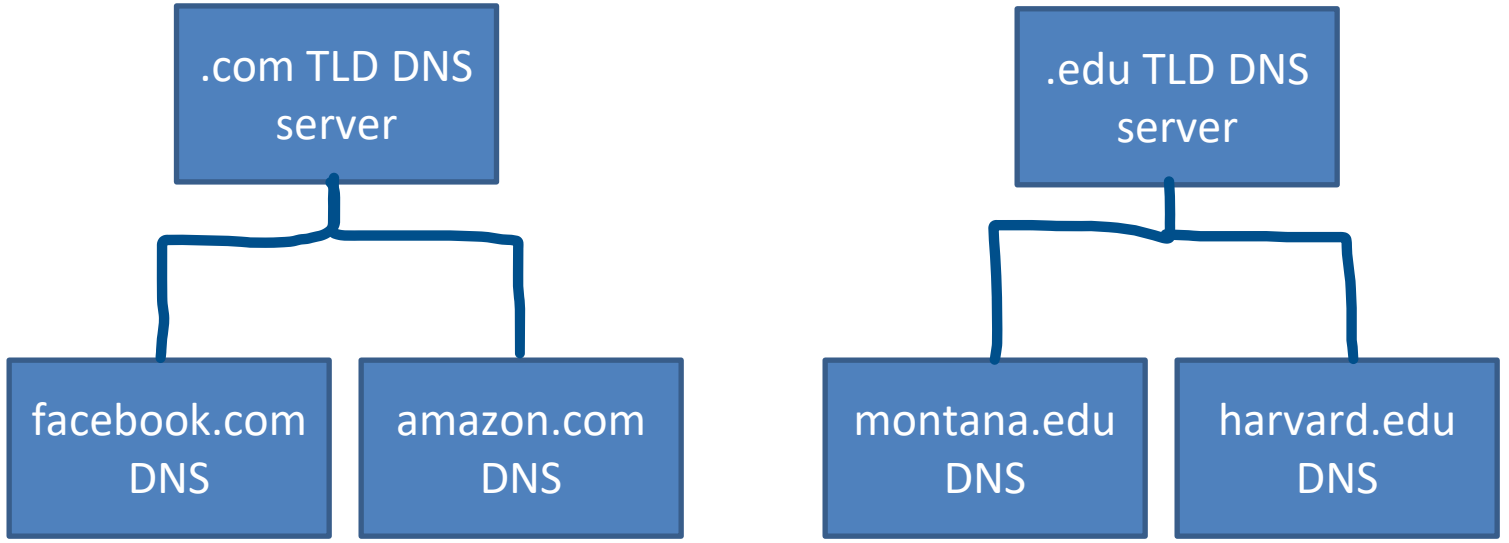
DNS Architecture

- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

Hierarchy consists of different types of DNS servers:

Authoritative DNS servers- Organization's own DNS with up-to-date records

Top-level domain (TLD) servers- responsible for keeping IP addresses for authoritative DNS servers for each top-level domain (.com, .edu, .jp, etc)



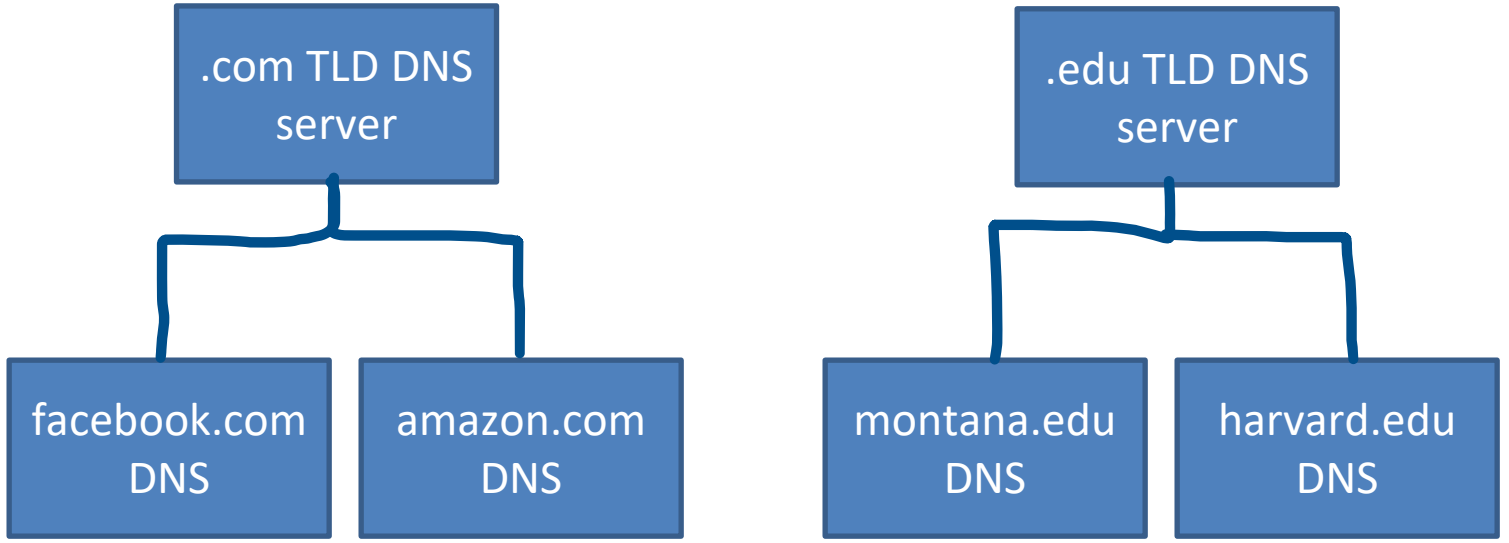
DNS Architecture

- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

Hierarchy consists of different types of DNS servers:

Authoritative DNS servers- Organization's own DNS with up-to-date records

Top-level domain (TLD) servers- responsible for keeping IP addresses for authoritative DNS servers for each top-level domain (.com, .edu, .jp, etc)



DNS Architecture

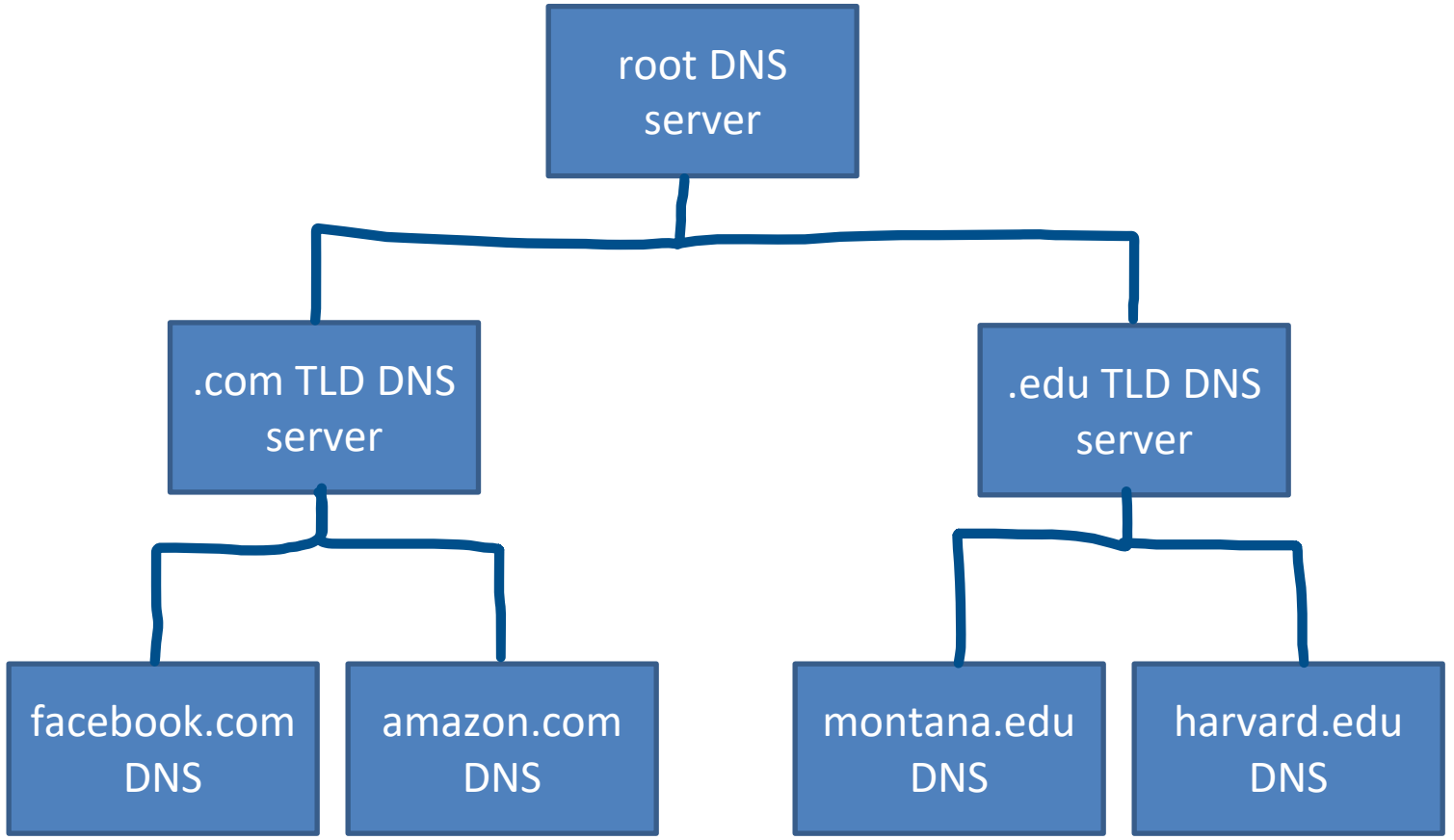
- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

Hierarchy consists of different types of DNS servers:

Authoritative DNS servers- Organization's own DNS with up-to-date records

Top-level domain (TLD) servers- responsible for keeping IP addresses for authoritative DNS servers for each top-level domain (.com, .edu, .jp, etc)

Root DNS servers- responsible for maintaining IP addresses for TLD servers



DNS Architecture

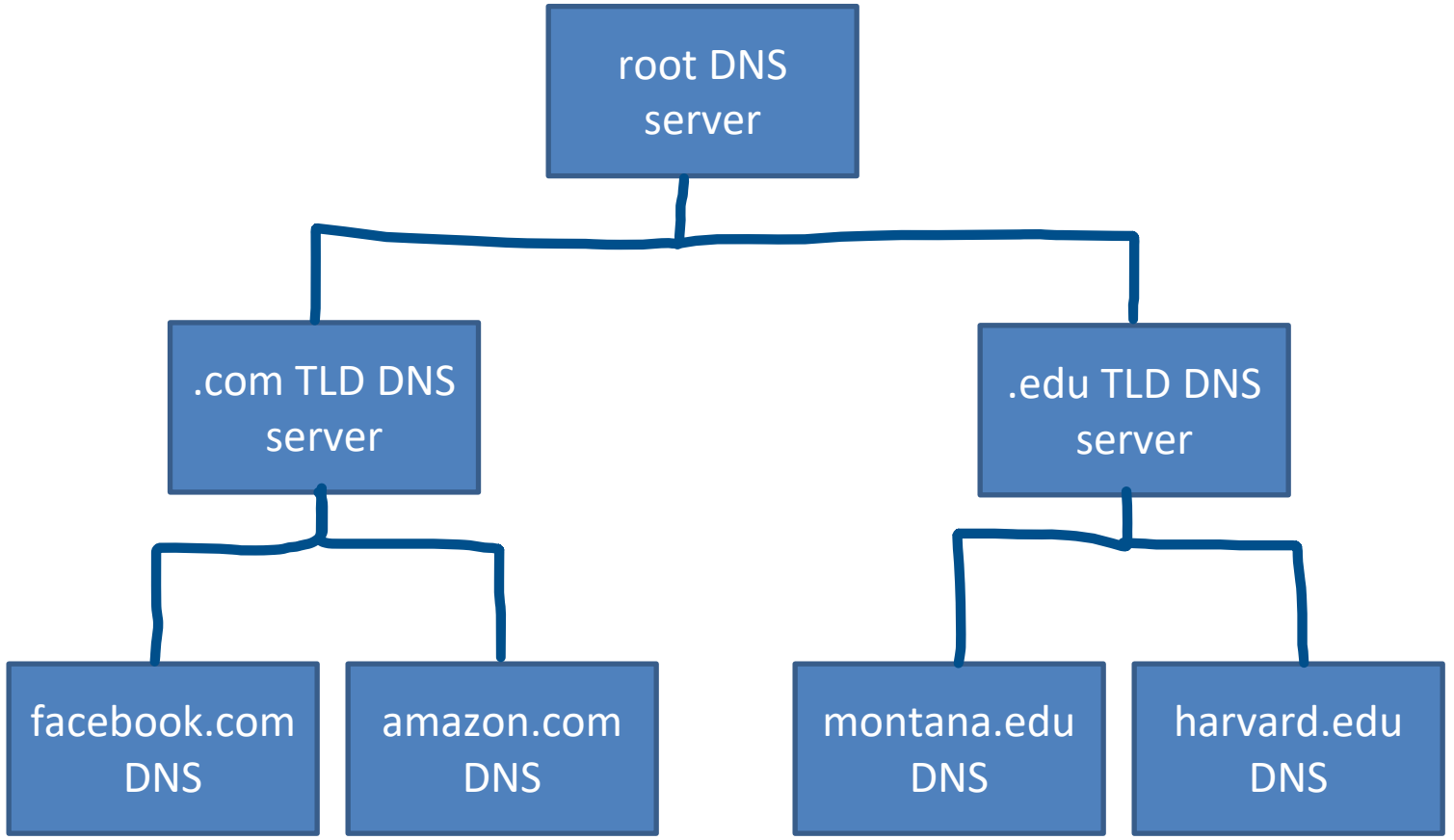
- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

Hierarchy consists of different types of DNS servers:

Authoritative DNS servers- Organization's own DNS with up-to-date records

Top-level domain (TLD) servers- responsible for keeping IP addresses for authoritative DNS servers for each top-level domain (.com, .edu, .jp, etc)

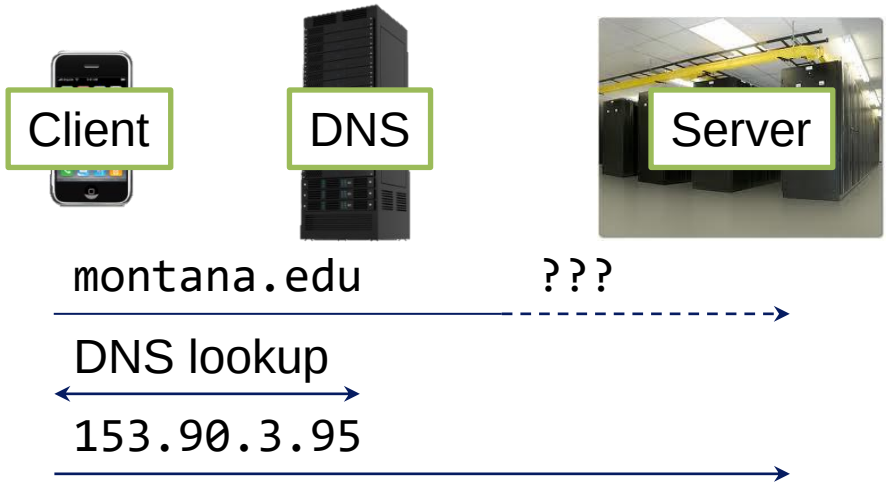
Root DNS servers- responsible for maintaining IP addresses for TLD servers



DNS Root server locations

https://root-servers.org/

DNS



Application layer protocol

- Lookups over UDP on port 53

(handshake not needed)
(DNS requests are small)
(reliability can be added in the application layer)

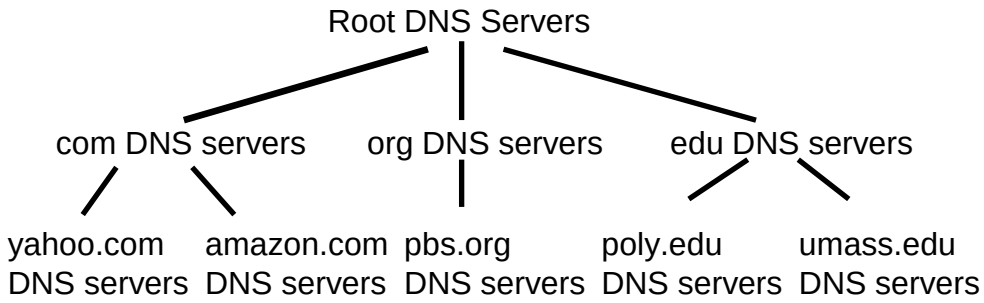
Local DNS servers are also used

- Acts as a proxy
- Maintained by ISP
- Caches records

```
C:\Users\Reese Pearsall>ipconfig/displaydns

Windows IP Configuration

    www.gstatic.com
    ----------------------------------------
    Record Name . . . . . : www.gstatic.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 18
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 142.251.211.227
```

Some DNS records are also stored and maintained in your computer

- Any issues?? 👀



What if an IP address gets changed?

DNS Commands

```
[09/09/22]seed@VM:~$ host montana.edu
montana.edu has address 153.90.3.95
montana.edu has address 153.90.2.191
montana.edu mail is handled by 50 montana-edu.mail.protection.outlook.com.
[09/09/22]seed@VM:~$
```

- DNS services
  - Hostname to IP address translation

    `host montana.edu`
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    `host -t CNAME img.huffingtonpost.com`
  - Mail server aliasing

    `host -t MX montana.edu`
  - Load distribution

    `host huffpost.com | grep "address" | sed -n -e 's/^.*address //p'`
  - Redirection
    - Look up same host from servers in different regions

    `host google.com 8.8.8.8`

153.90.3.95

(*nslookup* also works)

DNS Commands

- DNS services
  - Hostname to IP address translation

    `host montana.edu`
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    `host -t CNAME img.huffingtonpost.com`
  - Mail server aliasing

    `host -t MX montana.edu`
  - Load distribution

    `host huffpost.com | grep "address" | sed -n -e 's/^.*address //p'`
  - Redirection
    - Look up same host from servers in different regions

    `host google.com 8.8.8.8`

```
[09/09/22]seed@VM:~$ host -t AAAA montana.edu
montana.edu has no AAAA record
[09/09/22]seed@VM:~$
```

DNS Commands

- DNS services
  - Hostname to IP address translation

    ```
    host montana.edu
    ```
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    ```
    host -t CNAME img.huffingtonpost.com
    ```
  - Mail server aliasing

    ```
    host -t MX montana.edu
    ```
  - Load distribution

    ```
    host huffpost.com | grep "address" | sed -n -e
    's/^.*address //p'
    ```
  - Redirection
    - Look up same host from servers in different regions

    ```
    host google.com 8.8.8.8
    ```

```
[09/09/22]seed@VM:~$ host -t CNAME img.huffingtonpost.com
img.huffingtonpost.com is an alias for buzzfeed2.map.fastly.net.
[09/09/22]seed@VM:~$
```

DNS Commands

- DNS services
  - Hostname to IP address translation

    ```
    host montana.edu
    ```
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    ```
    host -t CNAME img.huffingtonpost.com
    ```
  - Mail server aliasing

    ```
    host -t MX montana.edu
    ```

    ```
    [09/09/22]seed@VM:~$ host -t MX montana.edu
    montana.edu mail is handled by 50 montana-edu.mail.protection.outlook.com.
    ```
  - Load distribution

    ```
    host huffpost.com | grep "address" | sed -n -e
    's/^.*address //p'
    ```
  - Redirection
    - Look up same host from servers in different regions

    ```
    host google.com 8.8.8.8
    ```

DNS Commands

- DNS services
  - Hostname to IP address translation

    `host montana.edu`
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    `host -t CNAME img.huffingtonpost.com`
  - Mail server aliasing

    `host -t MX montana.edu`
  - Load distribution

    `host huffpost.com | grep "address" | sed -n -e 's/^.*address //p'`
  - Redirection
    - Look up same host from servers in different regions

    `host google.com 8.8.8.8`

```
[09/09/22]seed@VM:~$ host huffpost.com | grep "address" | sed -n -e 's/^.*addres
s //p'
108.138.94.40
108.138.94.73
108.138.94.78
108.138.94.30
[09/09/22]seed@VM:~$ host huffpost.com | grep "address" | sed -n -e 's/^.*addres
s //p'
108.138.94.30
108.138.94.78
108.138.94.73
108.138.94.40
```

Rotation!

DNS Commands

- DNS services
  - Hostname to IP address translation

    `host montana.edu`
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    `host -t CNAME img.huffingtonpost.com`
  - Mail server aliasing

    `host -t MX montana.edu`
  - Load distribution

    `host huffpost.com | grep "address" | sed -n -e 's/^.*address //p'`
  - Redirection
    - Look up same host from servers in different regions

    `host google.com 8.8.8.8`

```
[09/09/22]seed@VM:~$ host google.com 8.8.8.8
Using domain server:
Name: 8.8.8.8
Address: 8.8.8.8#53
Aliases:

google.com has address 172.217.14.206
google.com has IPv6 address 2607:f8b0:400a:80a::200e
google.com mail is handled by 10 smtp.google.com.
[09/09/22]seed@VM:~$ host google.com
google.com has address 142.251.211.238
google.com has IPv6 address 2607:f8b0:400a:804::200e
google.com mail is handled by 10 smtp.google.com.
```

DNS Commands

- DNS services
  - Hostname to IP address translation

    `host montana.edu`
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    `host -t CNAME img.huffingtonpost.com`
  - Mail server aliasing

    `host -t MX montana.edu`
  - Load distribution

    `host huffpost.com | grep "address" | sed -n -e 's/^.*address //p'`
  - Redirection
    - Look up same host from servers in different regions

    `host google.com 8.8.8.8`

See cached DNS entries on computer

- `ipconfig/displaydns`

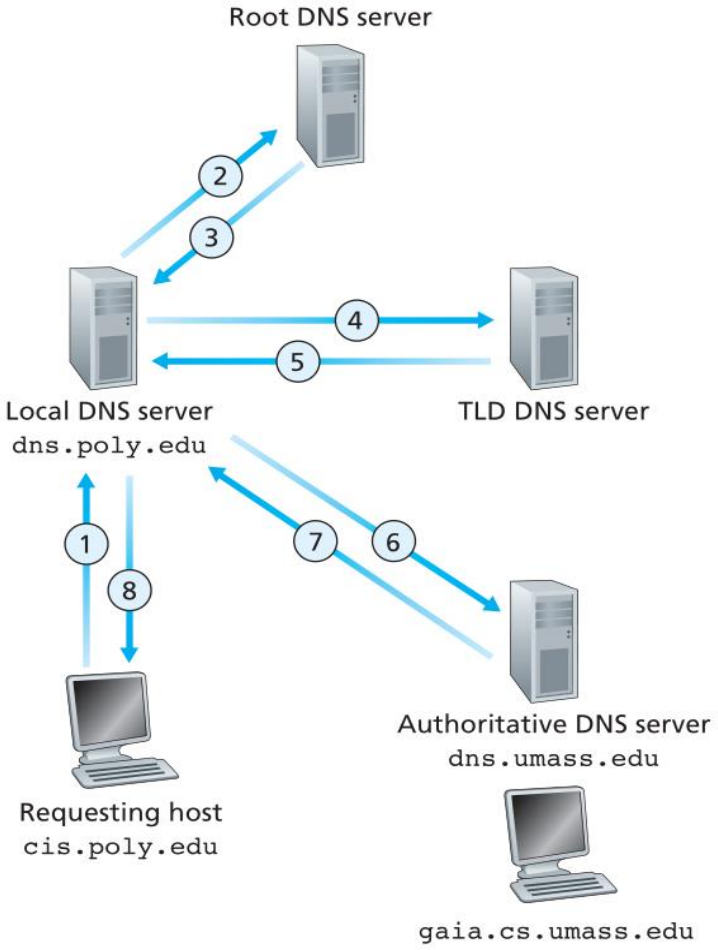
```
C:\Users\Reese Pearsall>ipconfig/displaydns

Windows IP Configuration

    safebrowsing.googleapis.com
    ----------------------------------------
    Record Name . . . . . : safebrowsing.googleapis.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 34
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 142.250.69.202
```

```
www.cs.montana.edu
----------------------------------------
Record Name . . . . . : www.cs.montana.edu
Record Type . . . . . : 5
Time To Live  . . . . : 3002
Data Length . . . . . : 8
Section . . . . . . . : Answer
CNAME Record  . . . . : web1.cs.montana.edu


Record Name . . . . . : web1.cs.montana.edu
Record Type . . . . . : 1
Time To Live  . . . . : 3002
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 153.90.127.197
```

```
www.tcpipguide.com
----------------------------------------
Record Name . . . . . : www.tcpipguide.com
Record Type . . . . . : 5
Time To Live  . . . . : 1543
Data Length . . . . . : 8
Section . . . . . . . : Answer
CNAME Record  . . . . : tcpipguide.com


Record Name . . . . . : tcpipguide.com
Record Type . . . . . : 1
Time To Live  . . . . : 1543
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 216.92.67.219
```

```
calendar.google.com
----------------------------------------
Record Name . . . . . : calendar.google.com
Record Type . . . . . : 1
Time To Live  . . . . : 144
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 142.251.211.238
```

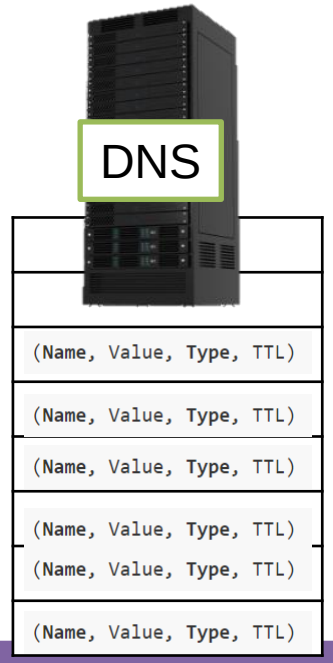
DNS Requests



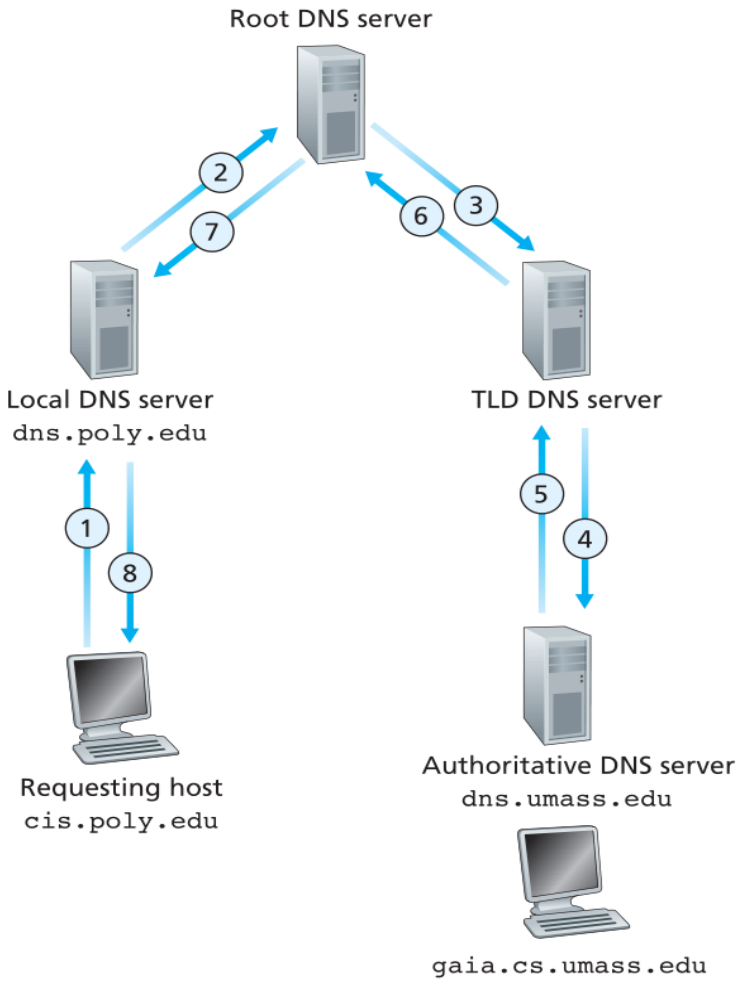
Iterative Lookup

Identification	Flags	12 bytes
Number of questions	Number of answer RRs	
Number of authority RRs	Number of additional RRs	
Questions (variable number of questions)		Name, type fields for a query
Answers (variable number of resource records)		RRs in response to query
Authority (variable number of resource records)		Records for authoritative servers
Additional information (variable number of resource records)		Additional "helpful" info that may be used

Is there a better option?



DNS servers store **resource records (RRs)**



Recursive Lookup