How the Web Works

Chapter 1: Introduction to Web Development

Chapter 2: How the Web Works

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development

Adapted to IT329 by Afnan Alsubaihin

What You'll Learn

1. Definitions and History
2. Internet Protocols
3. World Wide Web
4. Client-Server Model
5. Hypertext Transfer Protocol (HTTP)
6. Uniform Resource Locators (URL)
7. Server Types
8. Where is the Internet?

Section 1 of 8

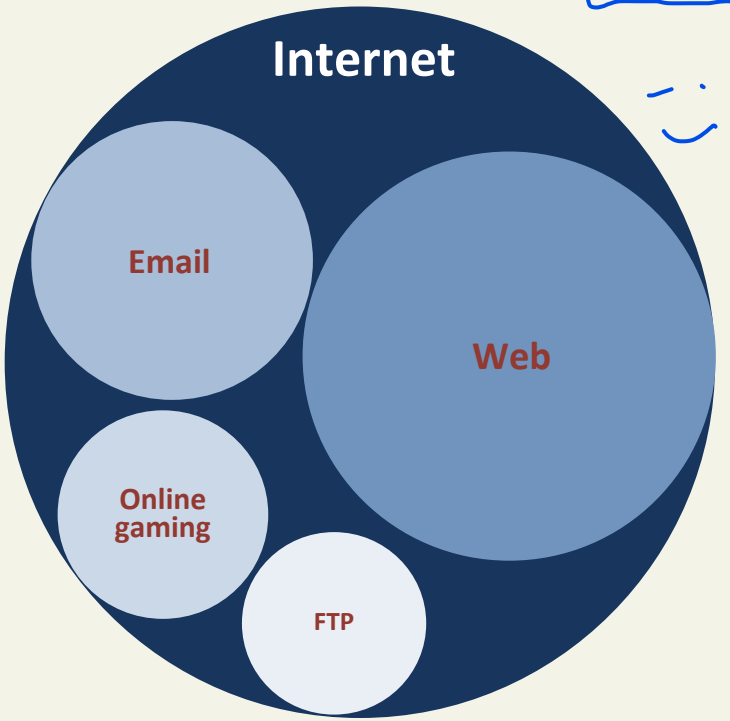
DEFINITIONS AND HISTORY

Internet = Web?

The answer is no

The World-Wide Web (WWW or simply the Web) is certainly what most people think of when they see the word “internet.”

But the WWW is only a subset of the Internet.



مبني على الانترنت

Short History of the Internet

The beginnings of the Internet

The research network ARPANET was created in the 1960s

- ARPANET used **packet switching** instead of circuit switching

A packet-switched network does not require a continuous connection:

- more robust (it is not reliant on a single pathway that may fail) and
- a more efficient use of network resources (since a circuit can communicate multiple connections).
- ARPANET started small with just a handful of connected campuses in 1969 and grew to a few hundred by the early 1980s.
- Then more networks came along!

TCP/IP

Helps computers talk to each other!

To promote the growth and unification of the disparate networks a suite of protocols was invented to **unify the networks together**.

By 1981, new networks built in the US began to adopt the **TCP/IP** (**Transmission Control Protocol / Internet Protocol**) communication model (discussed in the next section), while older networks were transitioned over to it.

Section 2 of 8

INTERNET PROTOCOLS

What's a Protocol?

The internet exists today because of a suite of interrelated communications protocols.

A **protocol** is a set of rules that partners in communication use when they communicate.

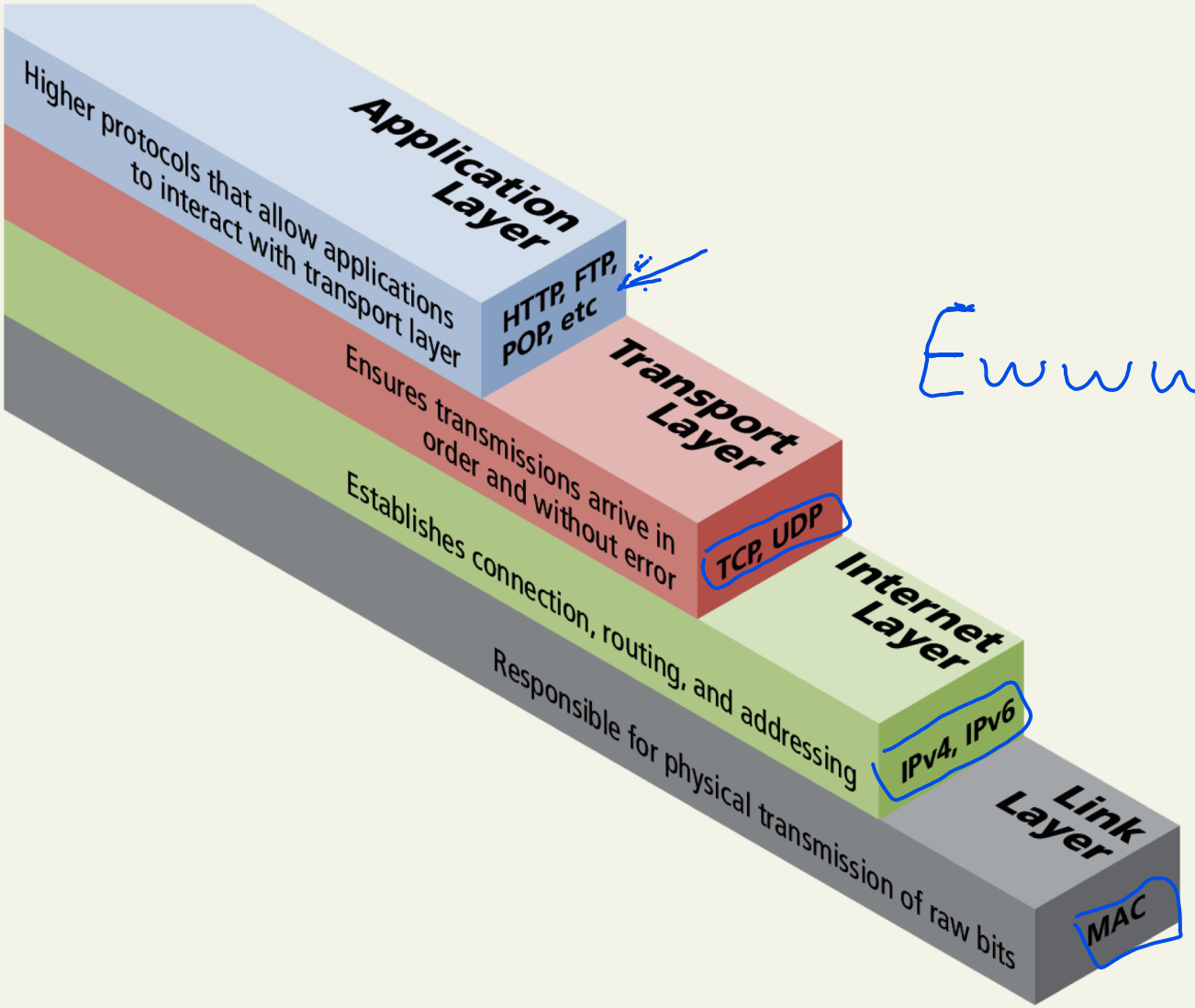
A Layered Architecture

The TCP/IP Internet protocols were originally abstracted as a four-layer stack.

Later abstractions subdivide it further into five or seven layers.

Since we are focused on the top layer anyhow, we will use the earliest and simplest **four-layer network model**.

Four Layer Network Model



Ewww..

Link Layer

Save this for your networking course

The **link layer** is the lowest layer, responsible for both the physical transmission across media (wires, wireless) and establishing logical links.

It handles issues like packet creation, transmission, reception and error detection, collisions, line sharing and more.

Internet Layer

The **internet layer** (sometimes also called the IP Layer) routes packets between communication partners across networks.

Internet Protocol (IP)

The Internet uses the **Internet Protocol** (**IP**) addresses to identify destinations on the Internet.

Every device connected to the Internet has an **IP address**, which is a numeric code that is meant to uniquely identify it.

Transport Layer

The **transport layer** ensures transmissions arrive, in order, and without error.

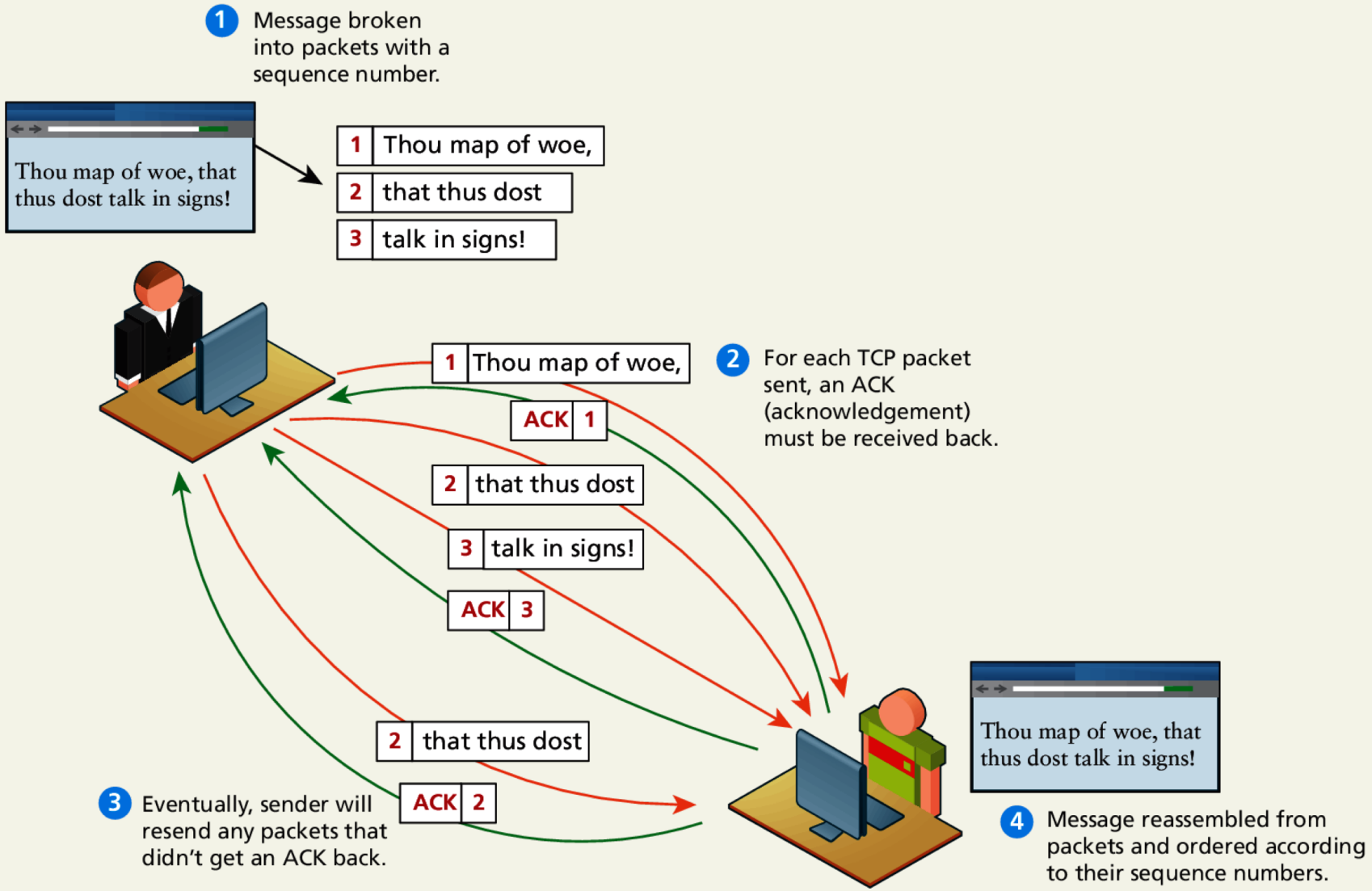
This is accomplished through a few mechanisms.

First, the data is broken into packets formated according to the **Transmission Control Protocol** (**TCP**).

Secondly, each packet is acknowledged back to the sender so in the event of a lost packet, the transmitter will realize a packet has been lost since no ACK arrived for that packet.

That packet is retransmitted, and although out of order, is reordered at the destination.

TCP Packets


1 Message broken into packets with a sequence number.
Thou map of woe, that thus dost talk in signs!
1 Thou map of woe,
2 that thus dost
3 talk in signs!
1 Thou map of woe,
ACK 1
2 For each TCP packet sent, an ACK (acknowledgement) must be received back.
2 that thus dost
3 talk in signs!
ACK 3
2 that thus dost
ACK 2
3 Eventually, sender will resend any packets that didn't get an ACK back.
Thou map of woe, that thus dost talk in signs!
4 Message reassembled from packets and ordered according to their sequence numbers.

Application Layer

With the **application layer**, we are the level of protocols familiar to most web developers.

Application layer protocols implement process-to-process communication and are at a higher level of abstraction in comparison to the low-level packet and IP addresses protocols in the layers below it.

Examples: HTTP, SSH, FTP, DNS, POP, SMTP.

Section 3 of 8

WORLD WIDE WEB

World Wide Web

Did you mean 'Sir Tim Berners-Lee'?

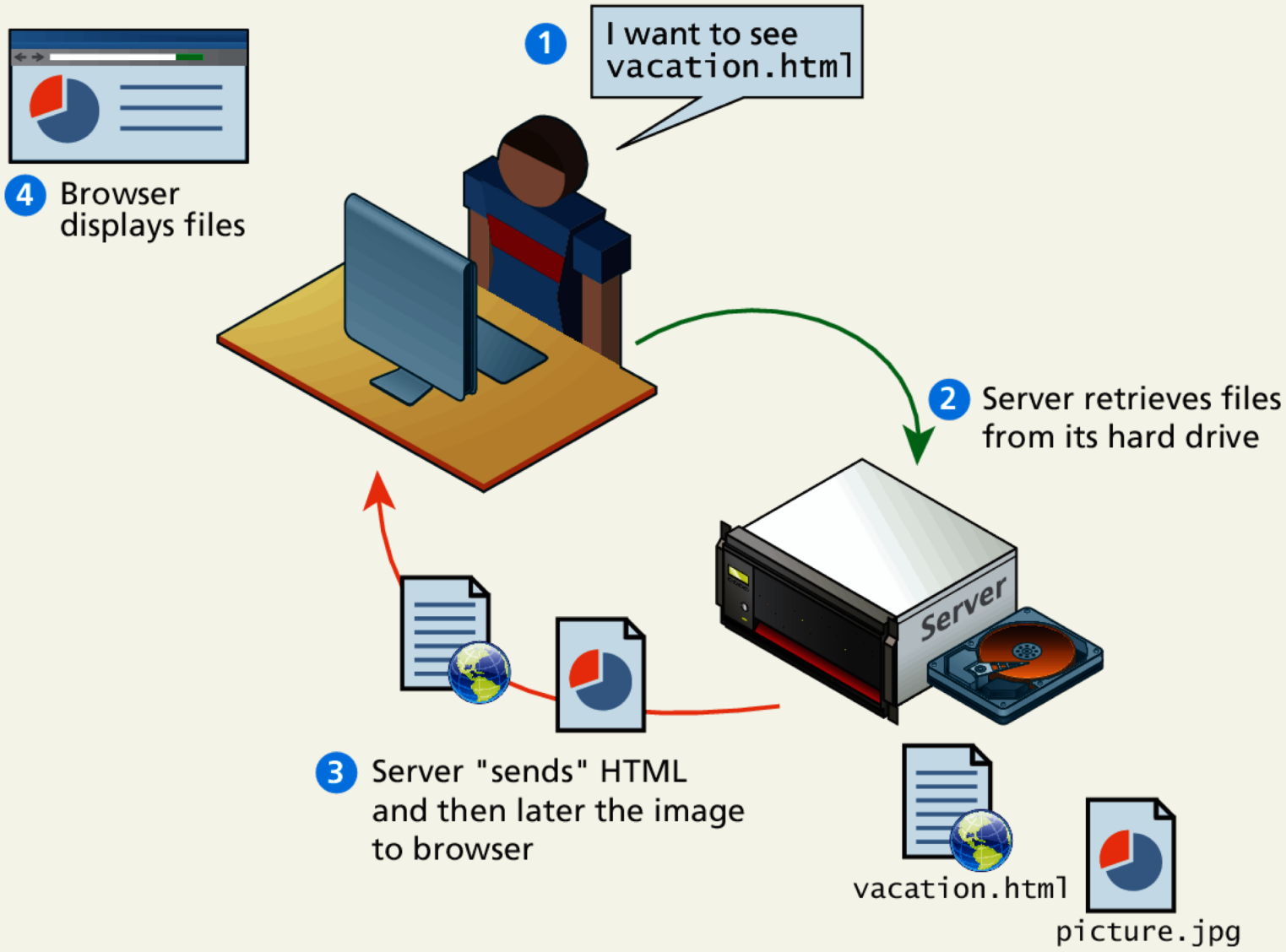
The WWW was born in Switzerland at CERN:

- Tim Berners Lee, along with the Belgian Robert Cailliau, published a proposal in 1990 for a hypertext system.

In those early days, the skills needed to create a web site were pretty basic: one needed knowledge of the HTML markup language and perhaps familiarity with editing and creating images.

This type of web site is commonly referred to as a **static web site**, in that it consists only of HTML pages that look identical for all users at all times.

Static Web Sites

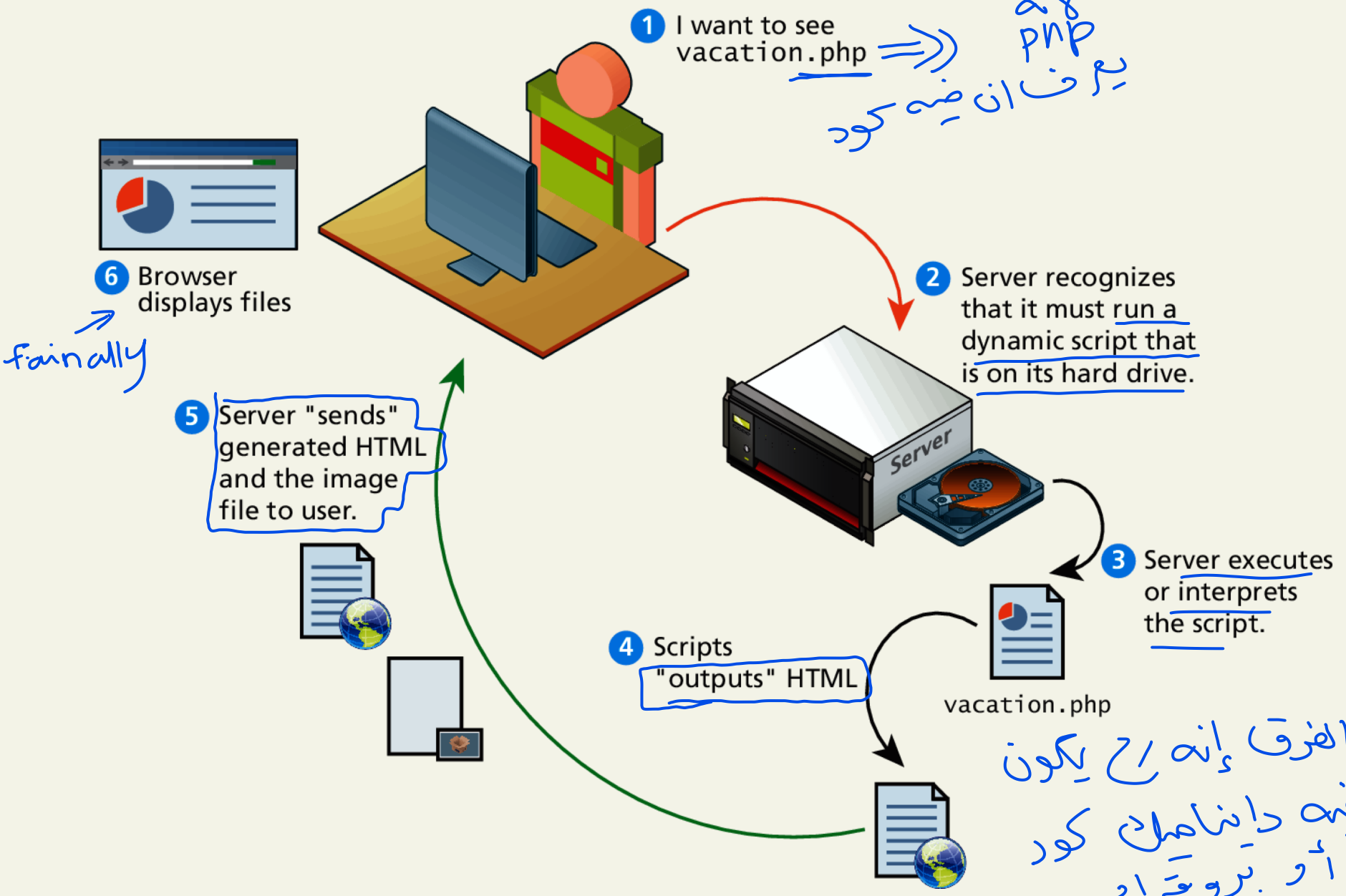


Dynamic Web Sites

Within a few years of the invention of the web, sites began to get more complicated.

- 'Web Pages' began to use programs running on web servers to generate their content dynamically.
- Content of the same web page may change dynamically depending on a server-side code.
- A dynamic web page can:
  - Read content from databases.
  - Interface with existing enterprise computer systems.
- And then output HTML that would be sent back to the users' browsers.
- This gave the rise to **Web Applications**

Dynamic Web Sites



Web Apps Compared to Native Apps

First the advantages of web apps

Some of the **advantages** of web applications include:

- Accessible from any internet-enabled device.
- Usable with different operating systems and browser platforms.
- Easier to roll out program updates since only need to update software on server and not on every desktop in organization.
- Centralized storage on the server means fewer concerns about local storage (which is important for sensitive information such as health care data).

١) تخزين عن السيرفر
فنقلل التخزين عن الكلاينت
٢) ممكن تكون داتا حساسة تحتاج privacy

Web Apps Compared to Native Apps

Now the disadvantages of web apps

Some of the **disadvantages** of web applications include:

- Requirement to have an active internet connection (the internet is not always available everywhere at all times).
- Security concerns about sensitive private data being transmitted over the internet.
- Concerns over the storage, licensing and use of uploaded data.
- Problems with certain websites on certain browsers not looking quite right.
- Limited access to the operating system can prevent software and hardware from being installed or accessed (like using the device's camera or location).

Section 4 of 8

CLIENT-SERVER MODEL

Client-Server Model

What is it?

The web is sometimes referred to as a client-server model of communications.

In the **client-server model**, there are two types of actors: clients and servers.

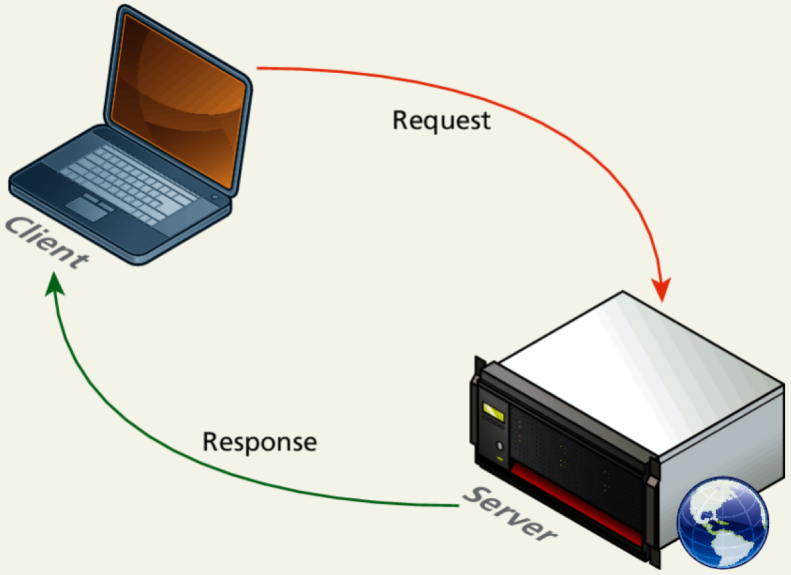
The **server** is a computer agent that is normally active 24 hours a day, 7 days a week, listening for queries from any client who make a request.

A **client** is a computer agent that makes requests and receives responses from the server, in the form of response codes, images, text files, and other data.

Request-Response Loop

Within the client-server model, the **request-response loop** is the most basic mechanism on the server for receiving requests and transmitting data in response.

The client initiates a **request** to a server and gets a **response** that could include some resource like an HTML file, an image or some other data.



Section 5 of 8

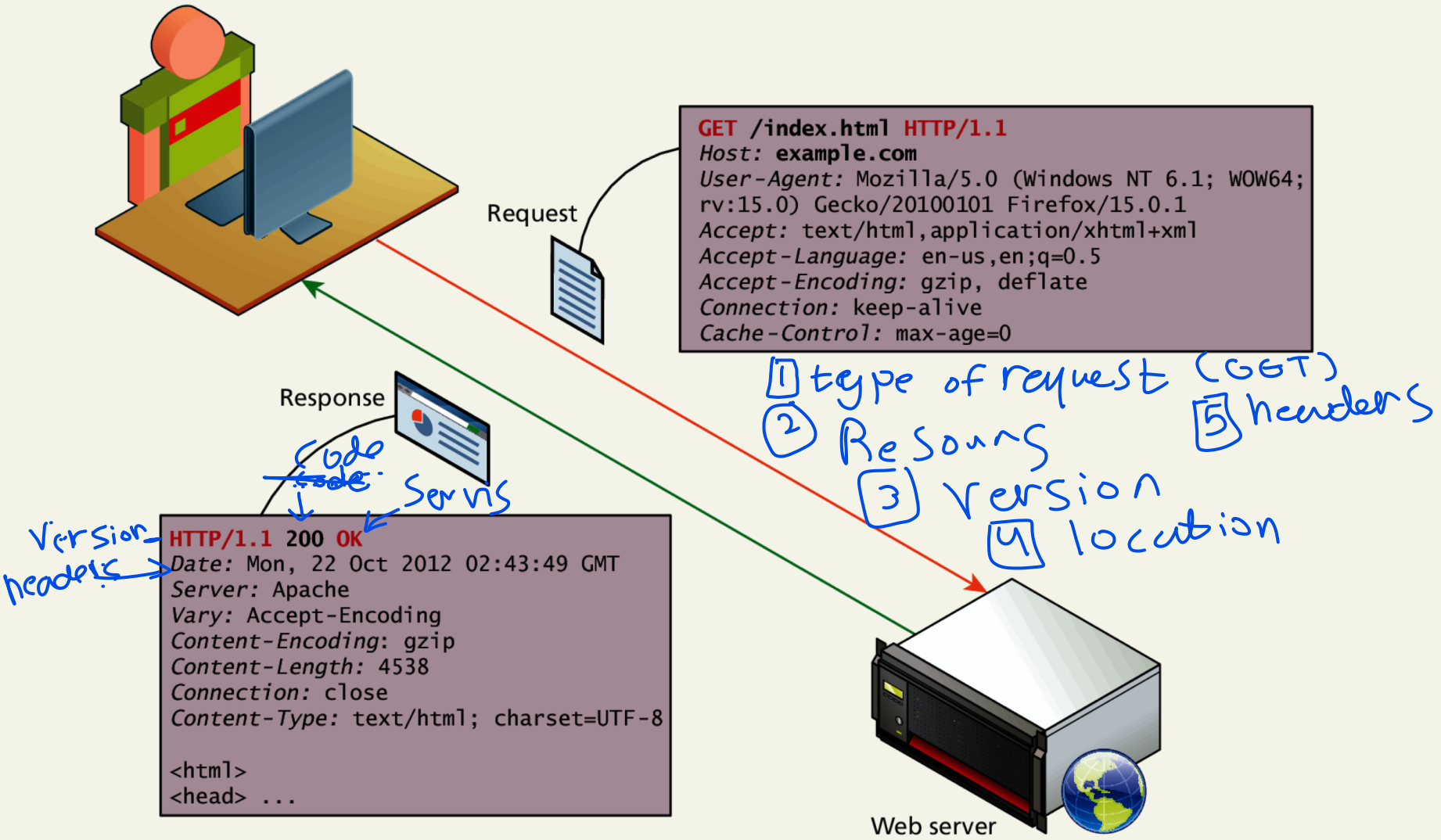
HYPERTEXT TRANSFER PROTOCOL (HTTP)

HTTP

The HTTP protocol establishes a TCP connection on port 80 (by default).

The server waits for the request, and then responds with a response code, headers and an optional message (which can include files).

HTTP


Request
GET /index.html HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20100101 Firefox/15.0.1
Accept: text/html,application/xhtml+xml
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cache-Control: max-age=0
1 type of request (GET)
2 Resource
3 version
4 location
5 headers
Response
code
Server
Version
headers
HTTP/1.1 200 OK
Date: Mon, 22 Oct 2012 02:43:49 GMT
Server: Apache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 4538
Connection: close
Content-Type: text/html; charset=UTF-8
<html>
<head> ...
Web server

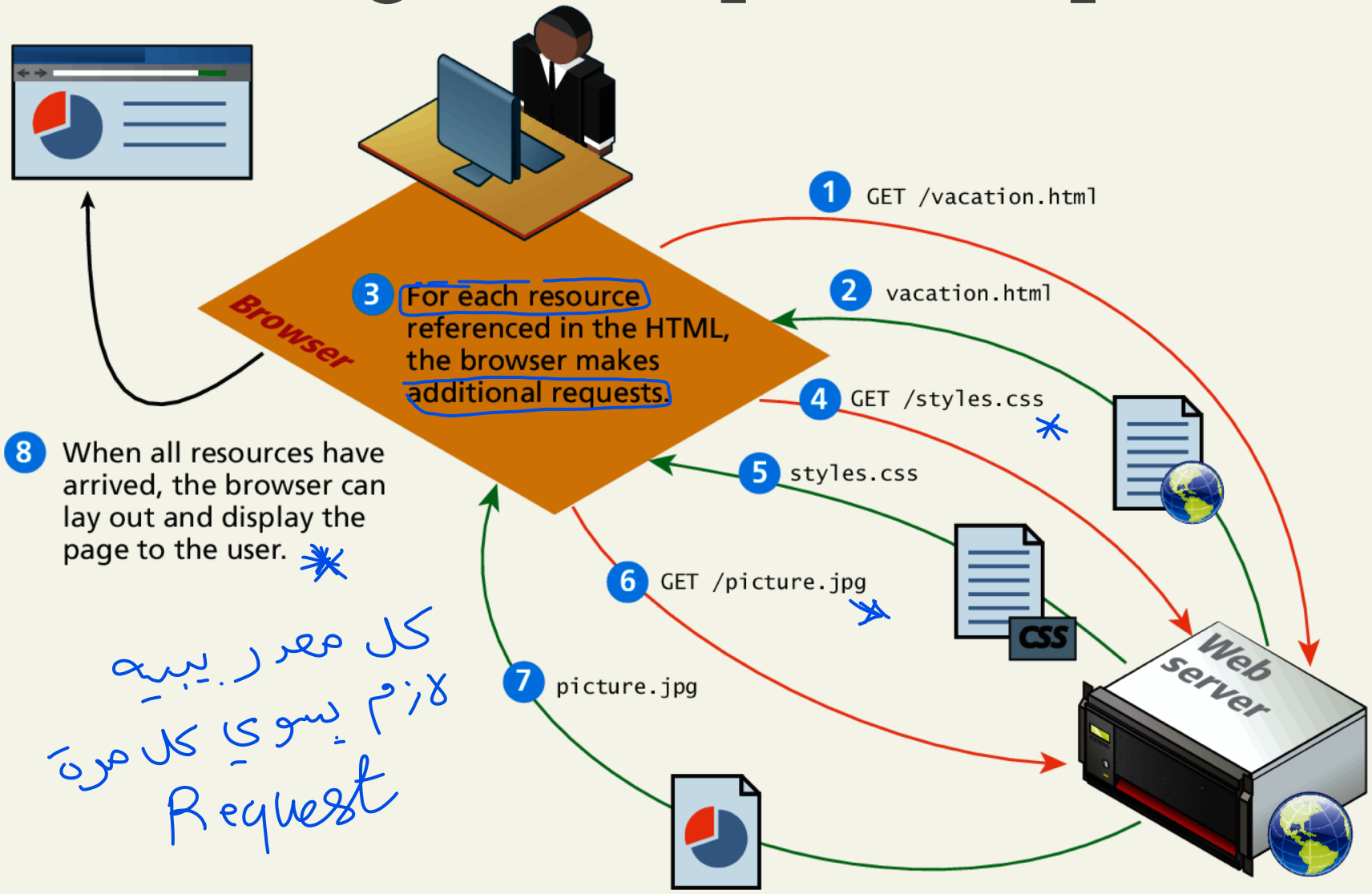
Web Requests

While we as web users might be tempted to think of an entire page being returned in a single HTTP response, this is not in fact what happens.

In reality the experience of seeing a single web page is facilitated by the client's browser which requests the initial HTML page, then parses the returned HTML to find all the resources referenced from within it, like images, style sheets and scripts.

Only when all the files have been retrieved is the page fully loaded for the user

Browser parsing HTML and making subsequent requests

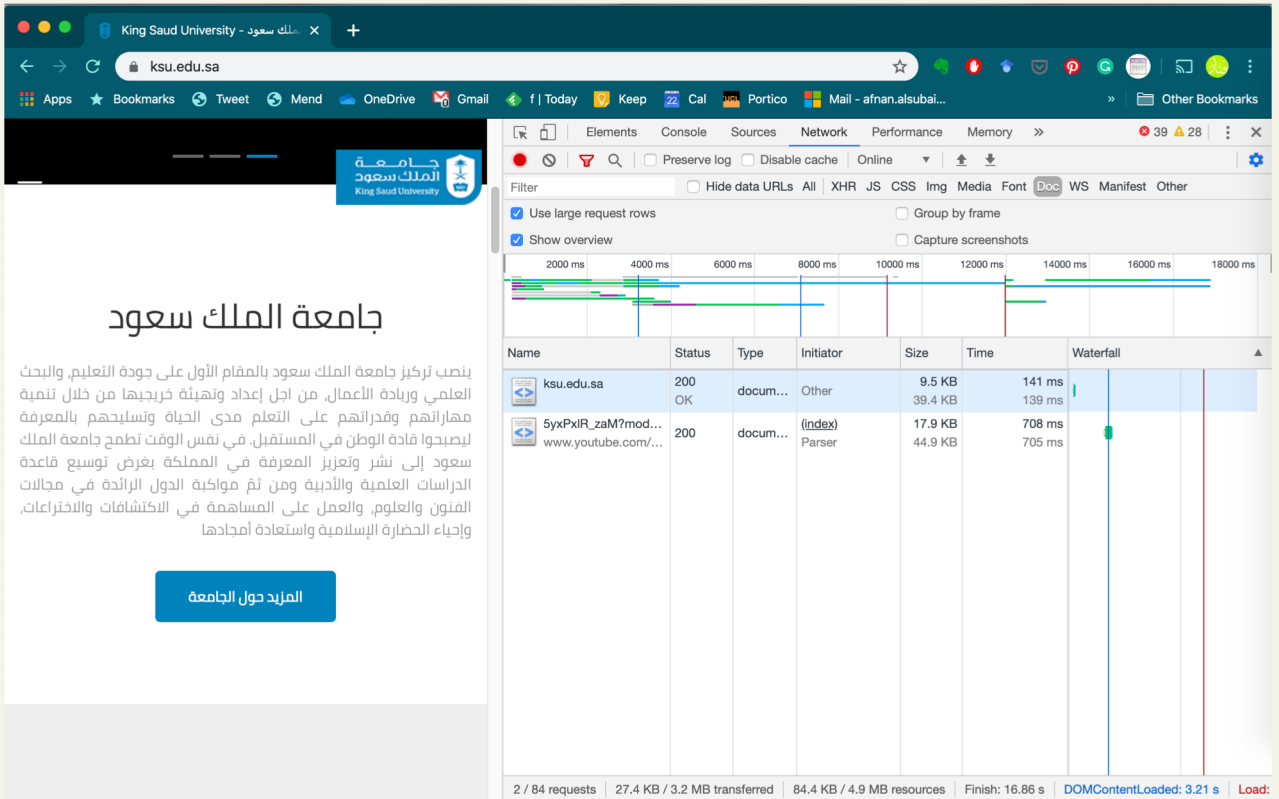


كل معدر بيبه
لازم يسوي كل مرة
Request

Browser Tools for HTTP

Modern browsers provide the developer with tools that can help us understand the HTTP traffic for a given page.

عن طريق Developer tools نعرف الـ Resources



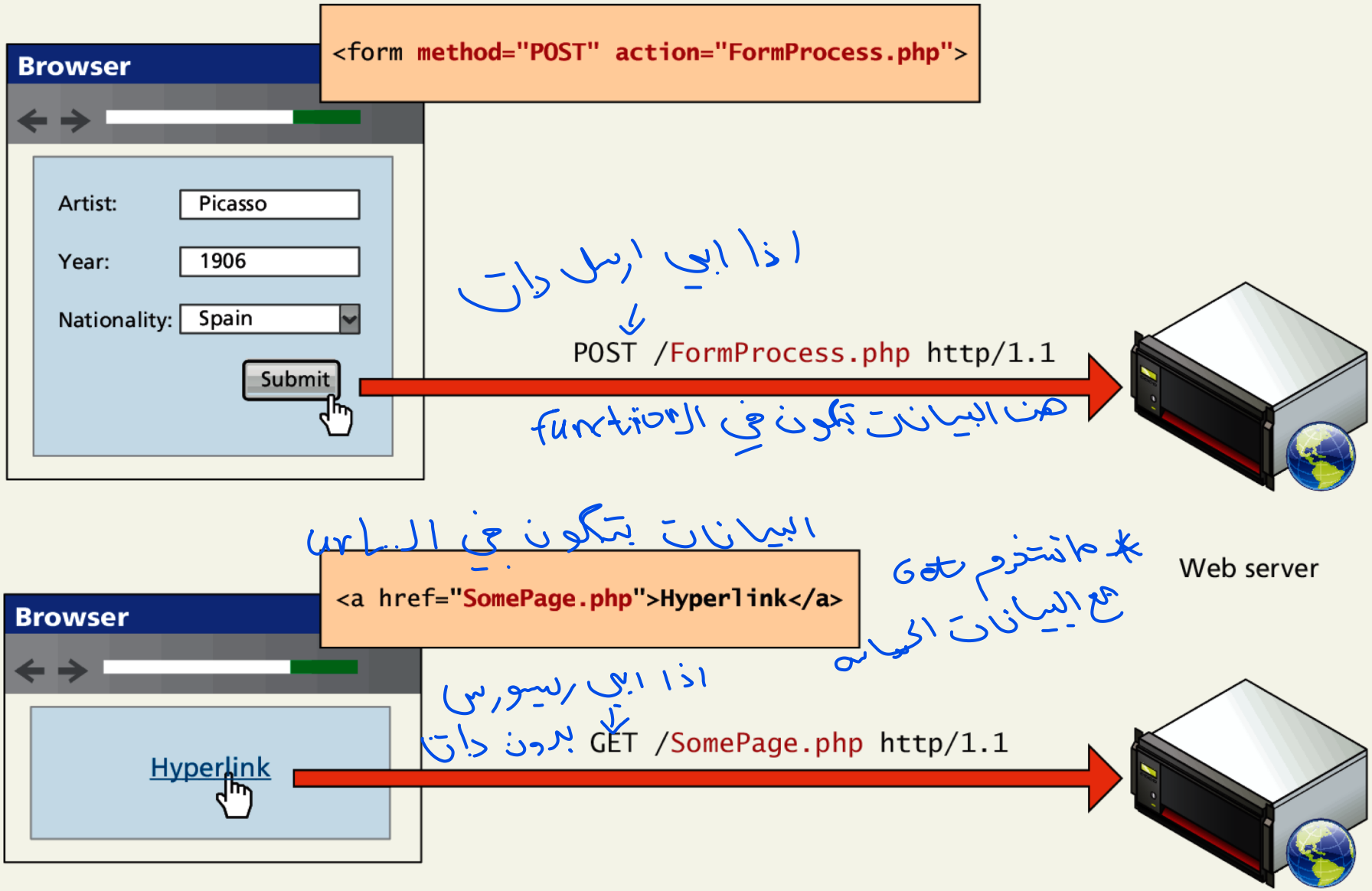
HTTP Request Methods

The HTTP protocol defines several different types of requests, each with a different intent and characteristics.

The most common requests are the GET and POST request, along with the HEAD request.

Other requests, such as PUT, DELETE, CONNECT, TRACE and OPTIONS are seldom used, and are not covered here.

GET versus POST requests



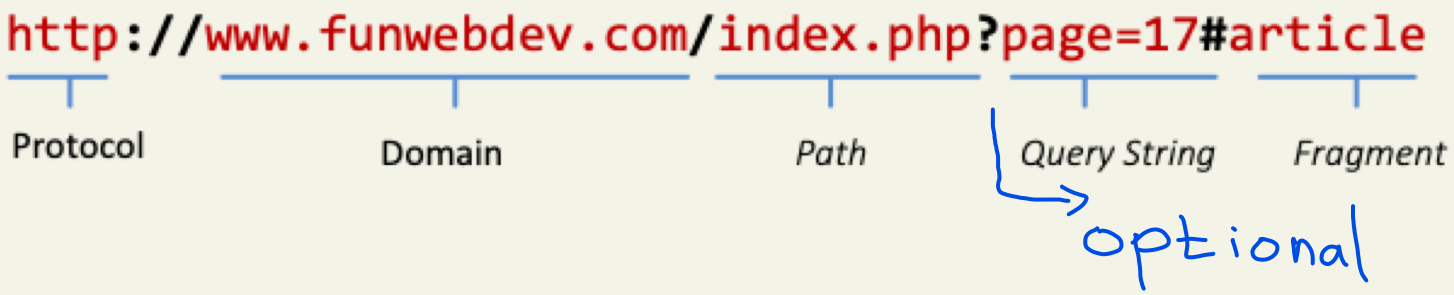
Section 6 of 8

UNIFORM RESOURCE LOCATORS (URL)

URL Components

In order to allow clients to request particular resources from the server, a naming mechanism is required so that the client knows how to ask the server for the file.

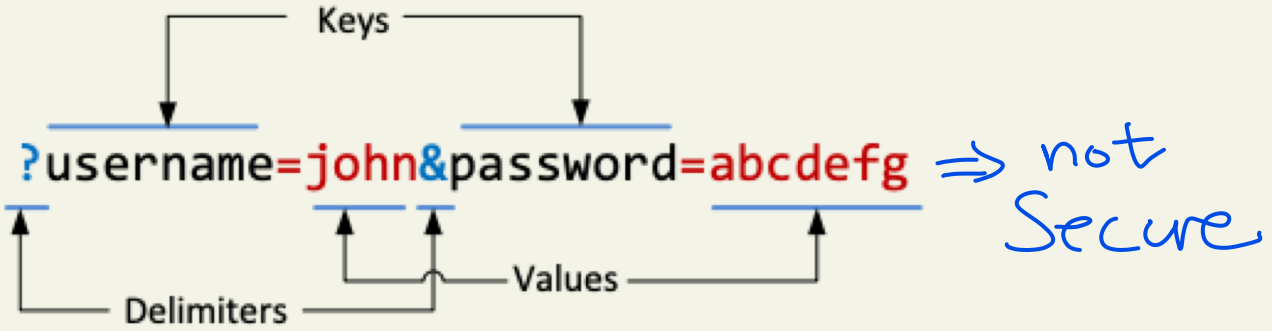
For the web that naming mechanism is the **Uniform Resource Locator** (**URL**). uniform resours Locator



Query String

Query strings will be covered in depth when we learn more about HTML forms and server-side programming.

They are the way of passing information such as user form input from the client to the server. In URL's they are encoded as key-value pairs delimited by “&” symbols and preceded by the “?” symbol.



⇒ not Secure

Section 7 of 8

SERVER TYPES

Server Types

A server is rarely just a single computer

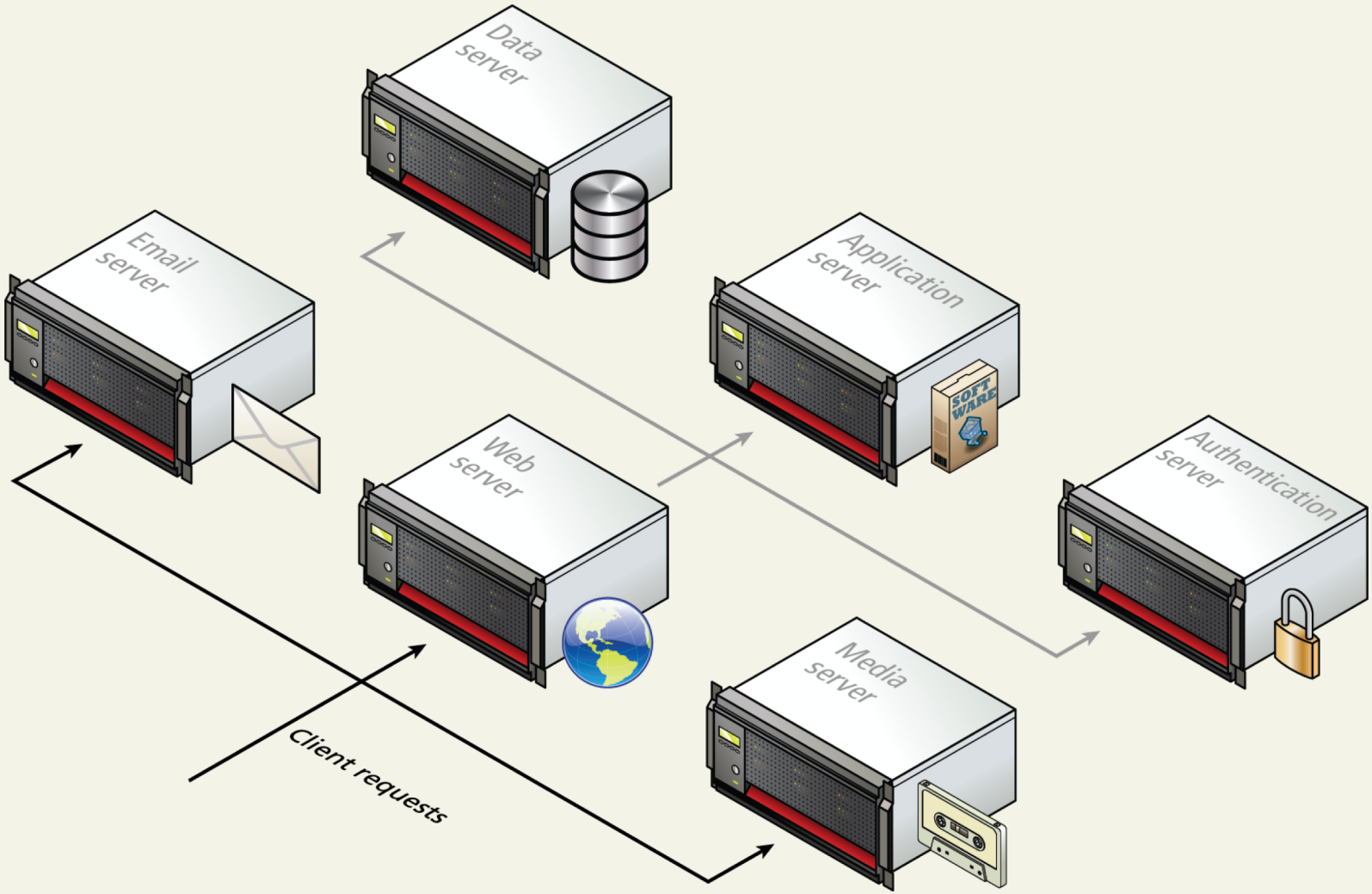
Earlier, the server was shown as a single machine, which is fine from a conceptual standpoint.

Clients make requests for resources from a URL; to the client, the server *is* a single machine.

However, most real-world web sites are typically not served from a single server machine, but by many servers.

It is common to split the functionality of a web site between several different types of server.

Server Types



Web Servers

A **web server** is, at a fundamental level, nothing more than a computer that responds to HTTP requests.

Real-World Server Installations

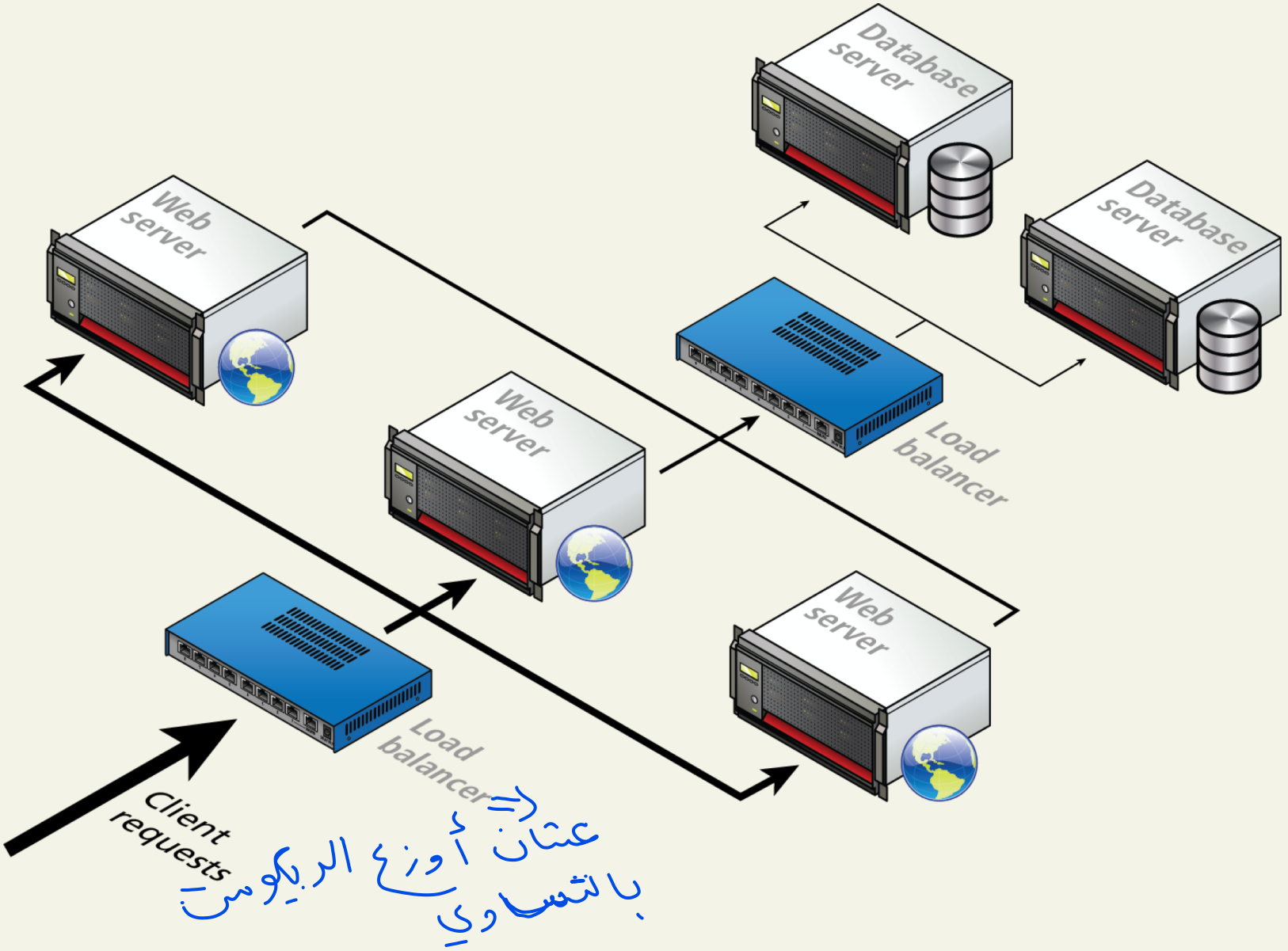
Not only are there different types of servers, there is often **replication** of each of the different server types.

A busy site can receive thousands or even tens of thousands of requests a second; globally popular sites such as Facebook receive millions of requests a second.

many requests = many servers

A single web server that is also acting as an application or database server cannot handle more than a few hundred requests a second, so the usual strategy for busier sites is to use a **server farm**.

Server Farm



عشان أوزع الريكوس
بالتساوي

Server Farms

The goal behind server farms is to distribute incoming requests between clusters of machines so that any given web or data server is not excessively overloaded.

Special routers called **load balancers** distribute incoming requests to available machines.

Even if a site can handle its load via a single server, it is common to still use a server farm because it provides **failover redundancy**.

That is, if the hardware fails in a single server, one of the replicated servers in the farm will maintain the site's availability.

Server Racks

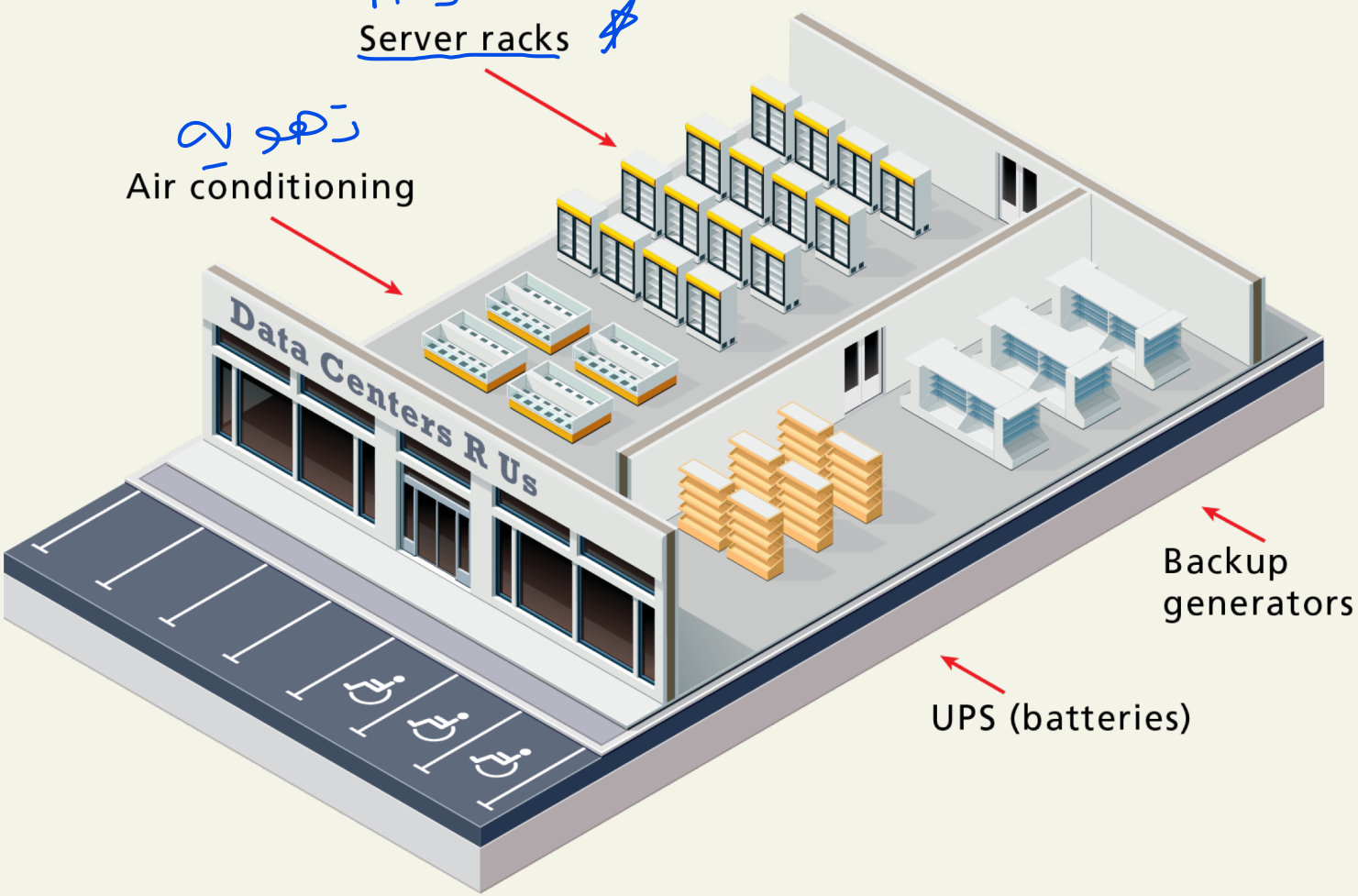
In a server farm, the computers do not look like the ones in your house.

A farm will have its servers and hard drives stacked on top of each other in **server racks**.

A typical server farm will consist of many server racks, each containing many servers.

Server farms are typically housed in special facilities called **data centers**.

Hypothetical Data Center



Data Centers

Where are they?

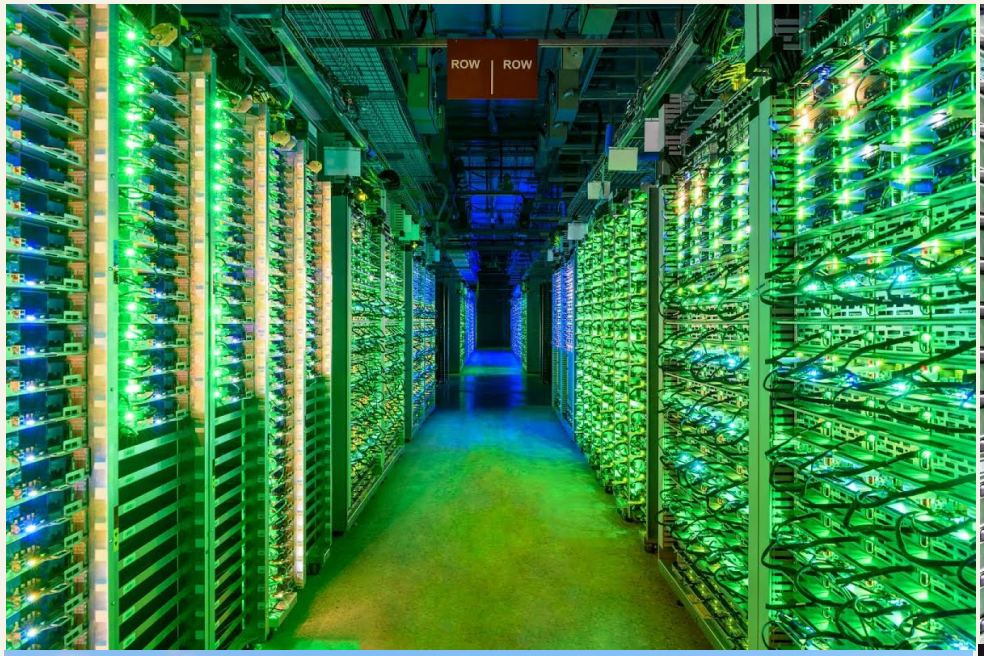
To prevent the potential for site down times, most large web sites will exist in **mirrored data centers** in different parts of the country, or even world.

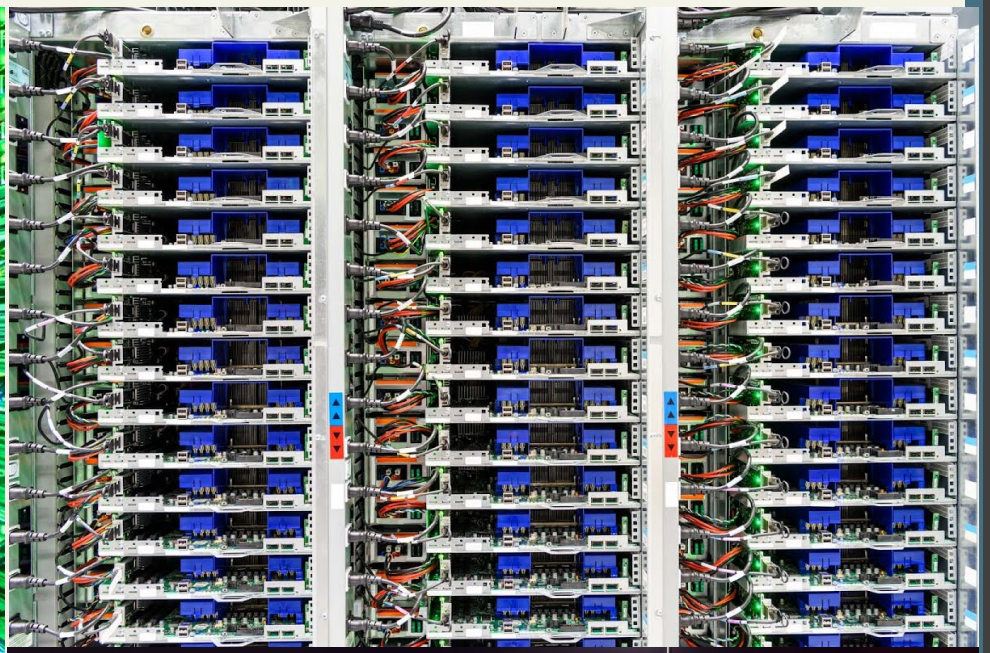
As a consequence, the costs for multiple redundant data centers are quite high, and only larger web companies can afford to create and manage their own.

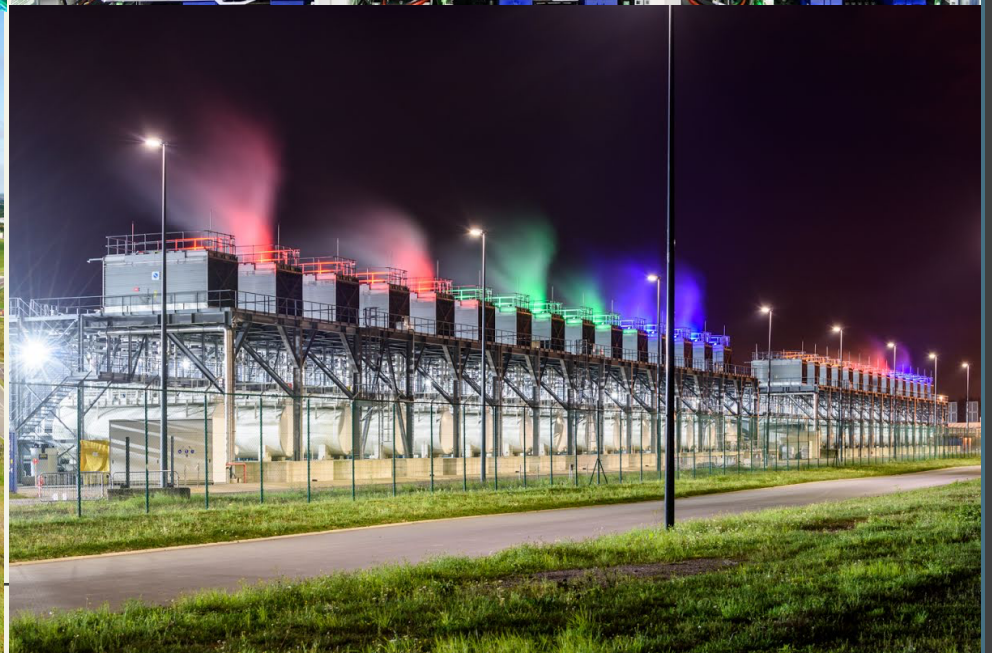
Most web companies will instead lease space from a **third-party data center.**

Data Centers

Google Data Centers:







Section 8 of 8

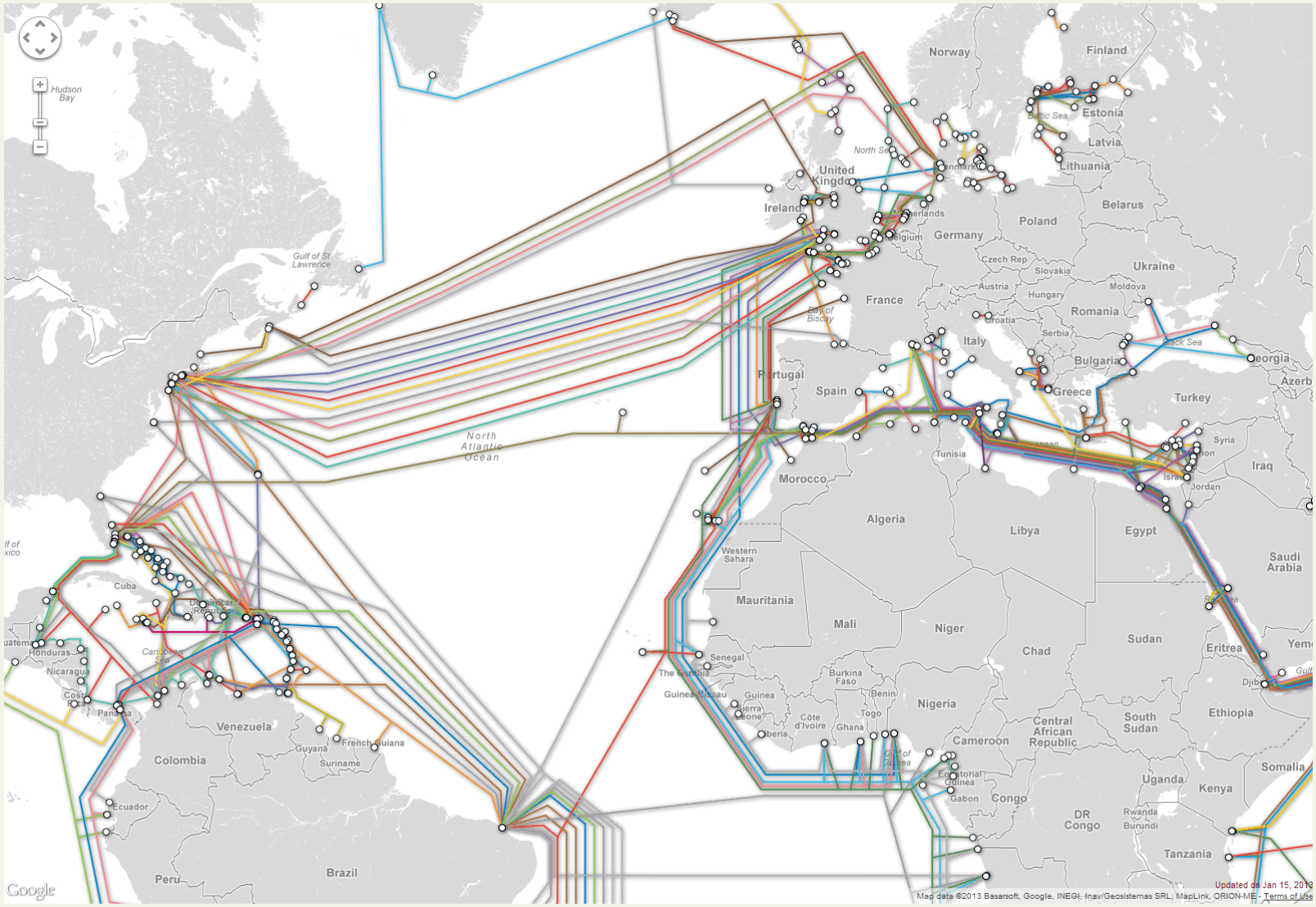
WHERE IS THE INTERNET?

Is the Internet a Cloud?

No

- The Internet is often visually represented as a **cloud**,
- Our global network of networks has nothing to do with actual clouds, but is implemented via
  - millions of kilometers of copper wires and fiber optic cables, as well as via
  - hundreds of thousands of server computers
  - and probably an equal number of routers, switches, and other networked devices,
  - along with many thousands of air conditioning units and specially-constructed server rooms and buildings.

Undersea fiber optic lines (courtesy TeleGeography)



What You've Learned

1. Definitions and History
2. Internet Protocols
3. World Wide Web
4. Client-Server Model
5. Hypertext Transfer Protocol (HTTP)
6. Uniform Resource Locators (URL)
7. Server Types
8. Where is the Internet?