

CSI 3140 - LECTURE 01

INTERNET AND THE WORLD WIDE WEB

OBJECTIVES

- ▶ Chapter 1 of textbook
- ▶ Evolution of the Internet
- ▶ HTML, CSS, and JavaScript
- ▶ Many examples of fun (and boring) things you can build
- ▶ Code available at deitel.com/books/iw3htp5 and
pearsonhighered.com/deitel

MOTIVATION FOR THE COURSE

- ▶ The most accessible platform in the world
- ▶ 3+ billion people on the Internet
- ▶ 20+ billion IoT devices
- ▶ Client-Side Programming
 - ▶ Runs on the user's device (or browser)
- ▶ Server-Side programming
 - ▶ Responds to customer requests
 - ▶ Buy a toy, book a train ride, check your inventory, order concert tickets

MOORE'S LAW

- ▶ In 1965, Intel co-founder Gordon Moore noted that the number of components in integrated circuits doubled every 12 months or so.
- ▶ 1970 - 2000 increased processor speed
 - ▶ 1970: 740 KHz to 8MHz
 - ▶ 2000: 1.3 GHz to 2.8 GHz
 - ▶ 2016: 2.1 GHz to 3.5 GHz

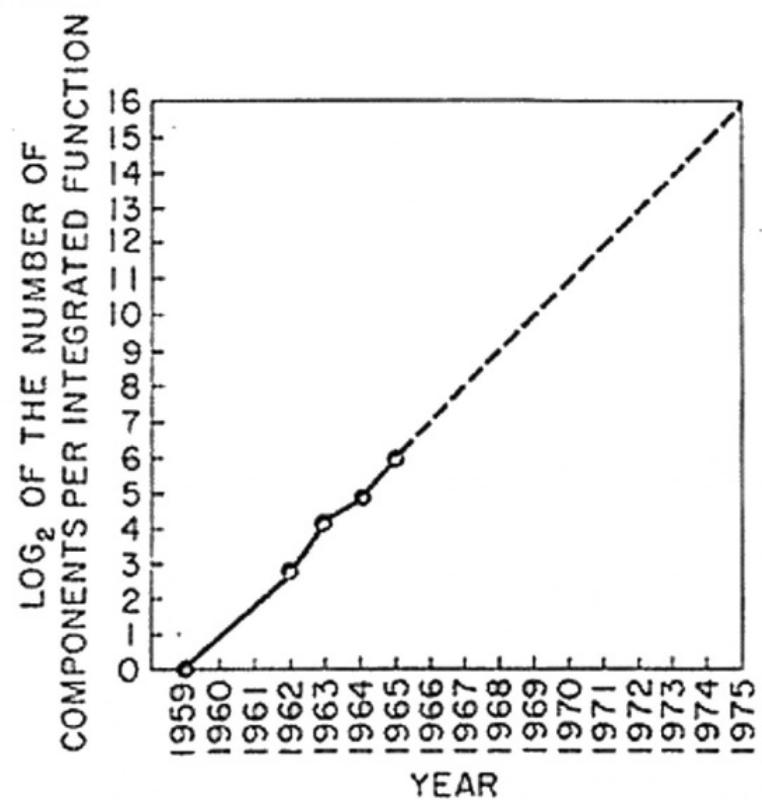


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

Is MOORE's LAW STILL A THING?

YES - EXPLAIN

No - WHEN/WHY STOP?

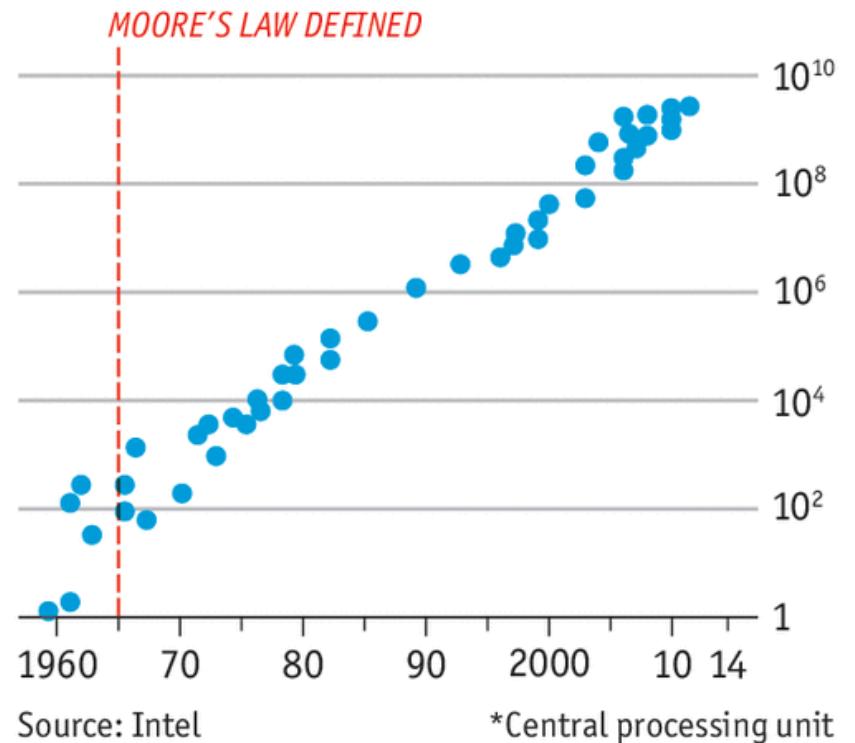
TRANSISTORS NOT THE SPEED THAT WAS DOUBLING

- ▶ In 2000, approx. 40 million transistors per CPU
 - ▶ in 2009 900 million
 - ▶ in 2016, 7.2 billion
- ▶ “Clock speed” is finished
 - ▶ Because of the speed of light and
 - ▶ heat generated
- ▶ But multi-core continues
 - ▶ More and more concurrent / parallel languages are needed to use the “new” speed

A persevering prediction

Number of transistors in CPU*

Log scale



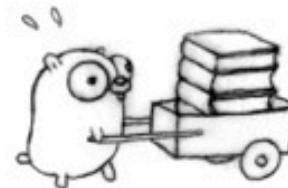
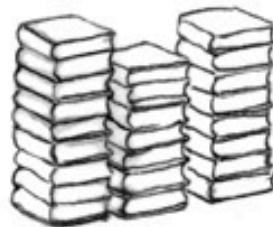
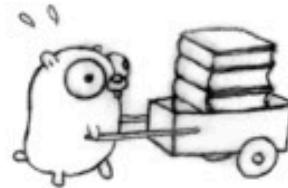
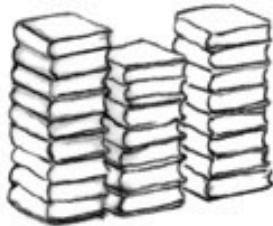
CONCURRENT VS. PARALLEL?

- ▶ Concurrent is the more generic term for overlapping task executions
 - ▶ Not necessarily at the same time
- ▶ Parallel is when several tasks (concurrently) execute at the same time

(CONT) CONCURRENT VS. PARALLEL?

- ▶ Concurrency is about **managing** with many things at once
- ▶ Parallelism is about **doing** a lot of things at once.
 - ▶ Not the same, but related.
- ▶ Concurrency is structure,
 - ▶ parallelism is execution.
- ▶ Concurrency provides a way to structure a solution to solve a problem that can (but not necessarily) be parallelizable.

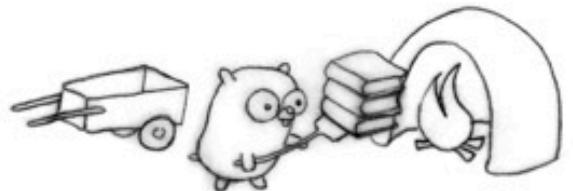
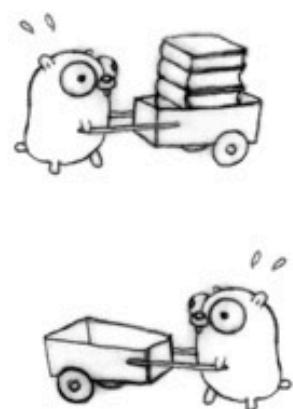
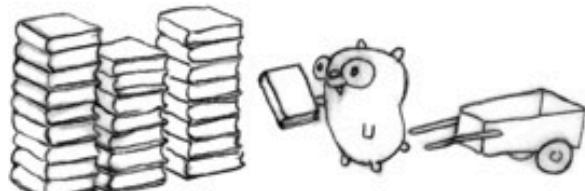
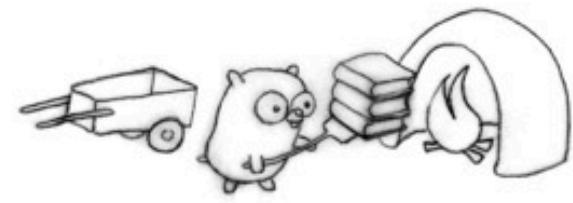
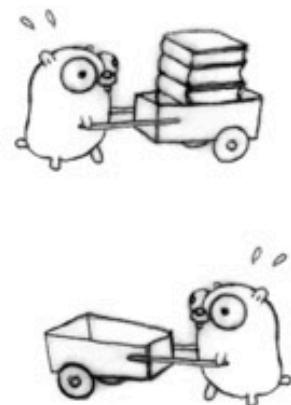
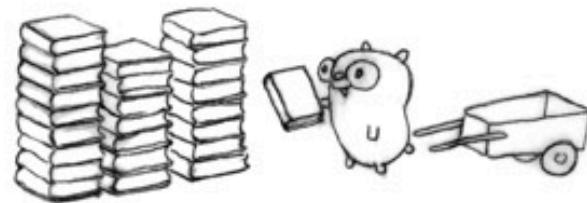
CONCURRENT, AN EXAMPLE



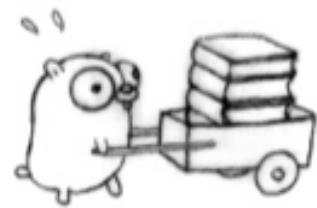
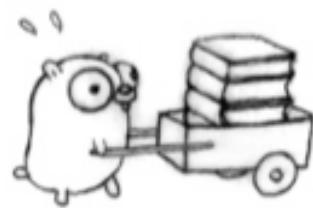
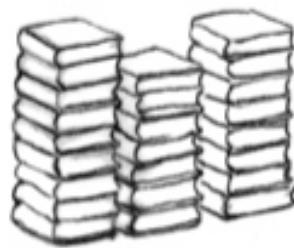
CONCURRENCY ANALYSIS

- ▶ Four separate gopher procedures:
 - ▶ Put books on the basket
 - ▶ Move the basket to the incinerator
 - ▶ Unload the basket
 - ▶ Return empty basket
- ▶ Different concurrency models allow different ways to parallelize.

MULTIPLE DIMENSIONS OF CONCURRENCY



MANY WAYS TO ACHIEVE CONCURRENCY

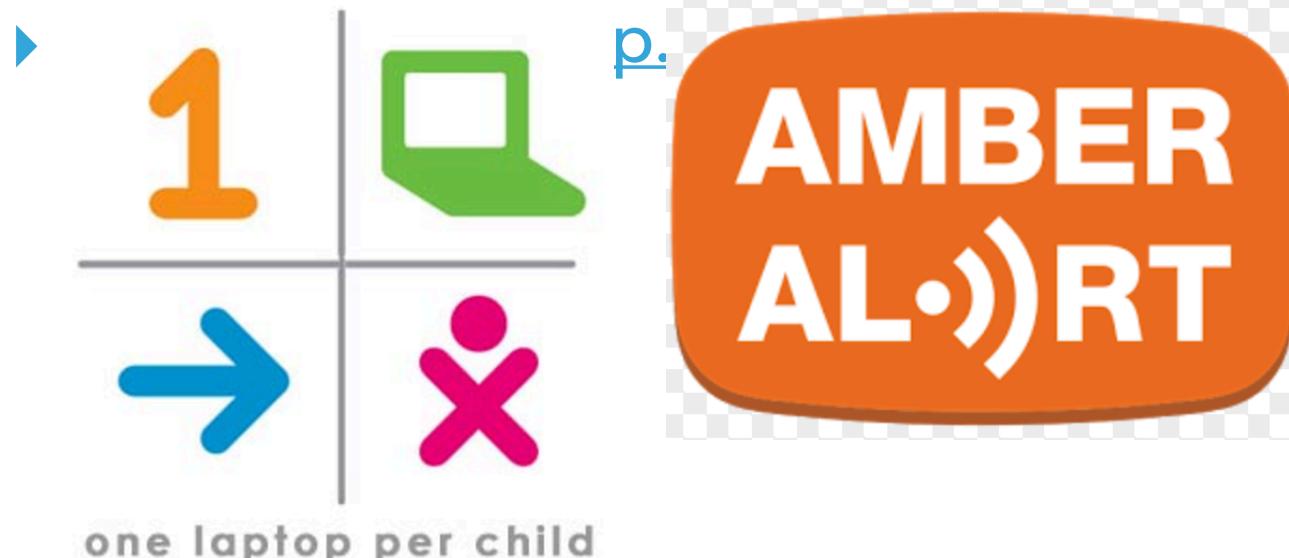


MOORE'S LAW AND OTHER CONCEPTS

- ▶ Applies to memory
- ▶ Secondary storage (aka disk)
- ▶ Similar increases in communications
- ▶ Prices fall as demand for "bandwidth" increases

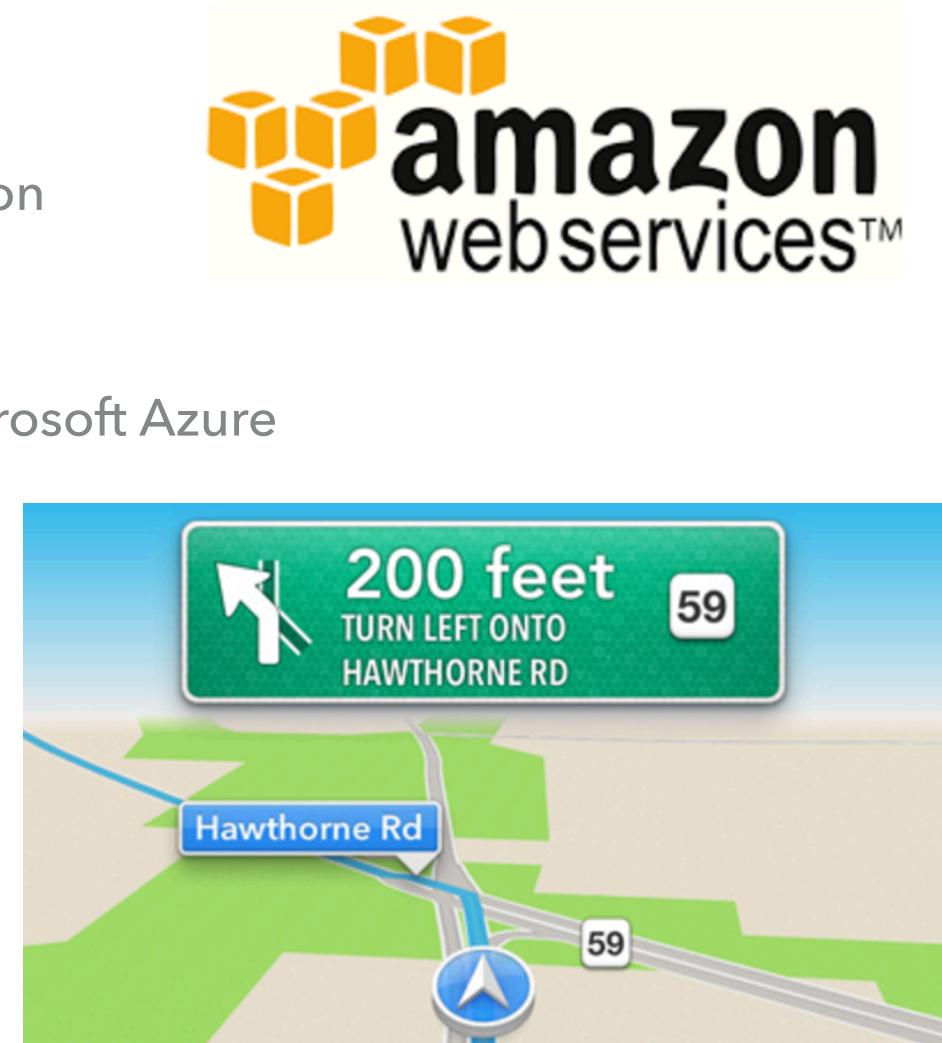
INDUSTRY AND RESEARCH

- ▶ Health care (human genome project)
- ▶ AMBER (tm) Alert (to locate abducted children)
- ▶ OLPC (one.laptop.org)



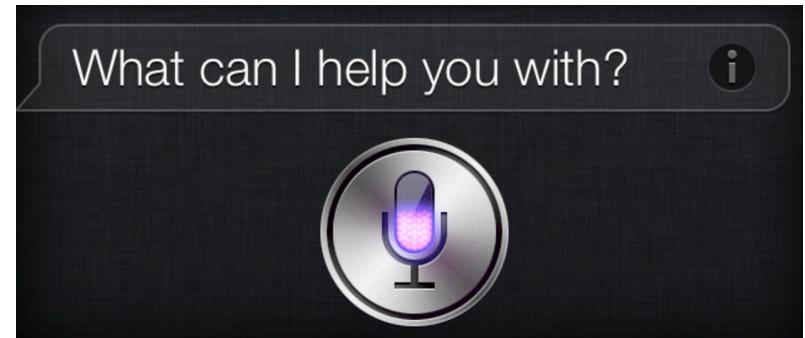
INFRASTRUCTURE FOR ...

- ▶ Communication, navigation, collaboration
- ▶ "Cloud computing"
 - ▶ Amazon S3, EC2, Google Cloud, Microsoft Azure
 - ▶ Digital Ocean, Linode
 - ▶ GitHub, Bitbucket, CircleCI
- ▶ GPS
 - ▶ Step by step instructions
 - ▶ Traffic patterns
 - ▶ "Find My Phone"



(CONT) INFRASTRUCTURE FOR ...

- ▶ "Robots"
 - ▶ Roomba vacuum cleaner
 - ▶ Exploration (deep sea, space)
 - ▶ Personal assistants
 - ▶ Siri, Facebook M, Alexa
- ▶ messaging
 - ▶ E-mail
 - ▶ IM (Slack, Hangouts, Skype)
 - ▶ Phone (WhatsApp)
 - ▶ Images (SnapChat, Instagram, Pinterest)



ENTERTAINMENT

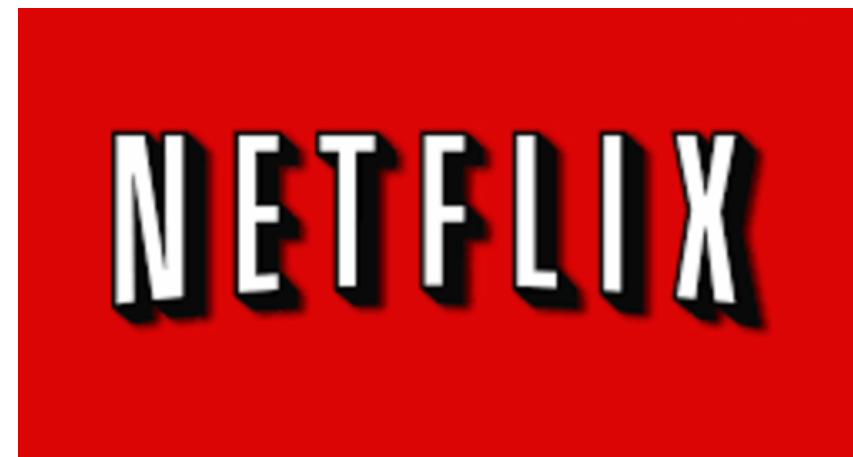
- ▶ Online Stores
 - ▶ iTunes
 - ▶ Google Play
- ▶ Internet TV
 - ▶ Netflix
 - ▶ Hulu
- ▶ App Stores
 - ▶ Mario Run
 - ▶ Questrade



App Store



Google play



HTML

- ▶ **HyperText Markup Language**
 - ▶ *markup language*
 - ▶ *Content and information structure*
 - ▶ *Originally designed as “**formatted**” documents*
 - ▶ *Now an extremely portable data format for sharing information (not just “documents”)*
- ▶ *Allows you to create systems adapted to a multiple device connected to the Internet*
 - ▶ *smartphones,*
 - ▶ *the tablets,*
 - ▶ *laptops / desktops,*
 - ▶ *giant screens (e.g. in sports stadiums)*

HTML VERSION HISTORY

- ▶ HTML 1 - Tim Berners-Lee 1990/1991
- ▶ HTML 2 - November 24, 1995
- ▶ HTML 3.2 - January 14, 1997
- ▶ HTML 4.0 - December 18, 1997
- ▶ HTML 5 - October 28, 2014
 - ▶ HTML 5.1 - November 1, 2016
 - ▶ HTML 5.2 - December 14, 2017

**WHAT'S THE CURRENT
VERSION OF HTML?**

XHTML

- ▶ Extensible HTML
- ▶ XML-based
- ▶ More Strict
- ▶ But, the trend is HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<br>  vs <br />
```

```
 vs 
```

```
<li>Bonjour, Monde! vs <li>Bonjour, Monde!</li>
```



<https://morioh.com/p/6d422fc49bd2?f=5c21fb01c16e2556b555ab32>

HTML6?

- ▶ No, 5.3 is the latest “nightly” version
- ▶ Worldwide Web Consortium (W3C) continues to use version numbers
- ▶ Web Hypertext Application Technology Working Group (WHATWG) have stopped
- ▶ “just” HTML - now

W3C

- ▶ Devoted to the development of inter-professional and non-proprietary technologies for the World Wide Web
- ▶ One of the main goals of W3C is to make websites universally accessible, regardless of disability, language or culture.
- ▶ A standardization body

WHATWG

- ▶ WHATWG was founded by individuals from Apple, the Mozilla Foundation and Opera Software in 2004, after a W3C workshop
- ▶ They were increasingly concerned about the direction of the W3C with XHTML, the lack of interest in HTML and the apparent disregard for the needs of real-world users.
- ▶ Mainly focuses on web standards, in particular:
 - ▶ HTML
 - ▶ fetch
 - ▶ DOM (Document Object Model)
 - ▶ URLs



<http://meanderingmouse.blogspot.com/2011/12/whats-true-meaning-of-christmas-its.html>

W3C AND THE
WHATWG
SIGNED AN
AGREEMENT TO
COLLABORATE
ON A SINGLE
VERSION OF
HTML AND DOM

May 2019

<https://www.w3.org/blog/news/archives/7753>

<https://html.spec.whatwg.org/review-drafts/2020-01/#history-2>

CSS

- ▶ Cascading Style Sheets
- ▶ Separate the style of the structure
 - ▶ Fonts, borders, colors, size, positioning
 - ▶ Please use a table for tabular data, not “divs”

THE PROMISE OF CSS

- ▶ (somewhat) easily change the look and feel without changing the content
- ▶ Far from perfect

The screenshot shows the Dividends.io homepage. At the top, it says "Dividends.io" and has a "Login" button. Below that, the tagline "Track your **payouts** like it's **payday**" is displayed. A sub-tagline reads: "Do you love all things dividends? Find it hard to track down how much passive income you are actually making? If so, then check us out. Our approach is simple, input your portfolio holdings and we will track your dividends like a paycheck." There is a large input field for "Provide your email to get started" and a blue button labeled "Email Me An Access Code". Below the input field, a small note says: "By clicking this button, you agree to our [Terms of Service](#)". At the bottom, there are two sections: "Fine print?" and "Do I have to worry about you selling my email or data?", each with a detailed explanation.

The screenshot shows a simplified version of the Dividends.io homepage. It features a "Dividends.io" logo and a "Login" link. Below it, the tagline "Track your payouts like it's payday" is present. A sub-tagline reads: "Do you love all things dividends? Find it hard to track down how much passive income you are actually making? If so, then check us out. Our approach is simple, input your portfolio holdings and we will track your dividends like a paycheck." There is a "Provide your email to get" input field and an "Email Me An Access Code" button. Below the input field, a note says: "By clicking this button, you agree to our [Terms of Service](#)". Further down, there are sections for "Fine print?", "Pricing?", and "Are there any limits?", each with a brief explanation. The overall layout is cleaner than the original screenshot.

CSS VERSION HISTORY

- ▶ CSS 1 - December 17, 1997
- ▶ CSS 2 - May 1998
- ▶ CSS 2.1 - February 25, 2004 (official June 7, 2011)
- ▶ CSS 3 - divided into 4 modules
 - ▶ Media - June 9, 2012
 - ▶ Namespaces - September 29, 2011
 - ▶ Selectors - September 29, 2011
 - ▶ Color - June 7, 2011

JAVASCRIPT

- ▶ Scripting language available in the browser
 - ▶ Not the only one
 - ▶ Dart by Google
 - ▶ VBScript and Internet Explorer
 - ▶ But yes it is the only one
- ▶ Created by Netscape
- ▶ Allows interactivity in the browser
- ▶ Use events
 - ▶ Keyboard and mouse actions

STANDARDIZATION OF JAVASCRIPT

- ▶ NetScape and Microsoft
- ▶ Developed by ECMA International
 - ▶ Formerly known as the European Computer Manufacturers Association
 - ▶ Now called ECMAScript

JAVASCRIPT VERSION HISTORY

- ▶ ECMAScript 1 - June 1997
- ▶ ECMAScript 2 - June 1998
- ▶ ECMAScript 3 - December 1999
- ▶ ECMAScript 5 - December 2009
 - ▶ 4th edition has been discontinued, due to differences in linguistic complexity
- ▶ ECMAScript 2015 (version 6)
- ▶ ECMAScript 2016 (version 7)

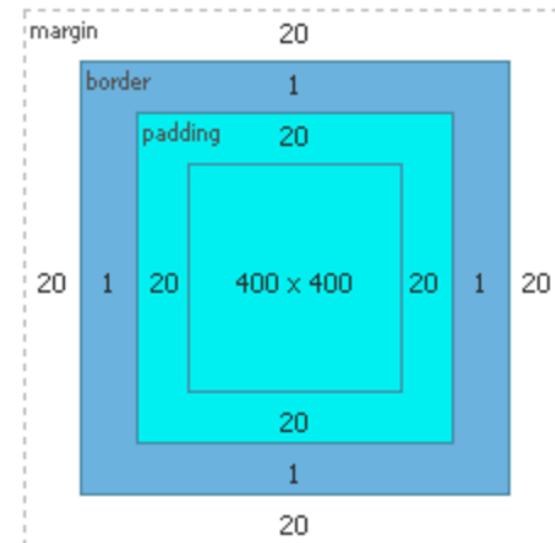
... IT KEEPS GOING

- ▶ ECMAScript 2017 (version 8)
- ▶ ECMAScript 2018 (version 9)
- ▶ ECMAScript 2019 (version 10)
- ▶ ES.Next ...

WHAT'S NEXT FOR JAVASCRIPT?

INTERNET BROWSERS

- ▶ A consistent “look and feel” is difficult
- ▶ Browsers are just software, no standard that must be followed



SO MANY OPTIONS

- ▶ Browsers share common functionality (and code), but differences will appear
- ▶ Different versions of the same browser will behave differently
- ▶ Different operating systems Windows, OS X, Linux
- ▶ Several approaches:
 - ▶ Most recent,
 - ▶ Most preferred,
 - ▶ The most accessible,
 - ▶ Or, the most common

HTML5 TEST (.com)

▶ html5test.com

YOUR BROWSER SCORES **499** OUT OF 555 POINTS

You are using Chrome Dev 54.0.2840.98 on macOS Sierra 10.12

Correct? ✓ ✗

[Save results](#) [Compare to...](#) [Share](#) [Donate](#)

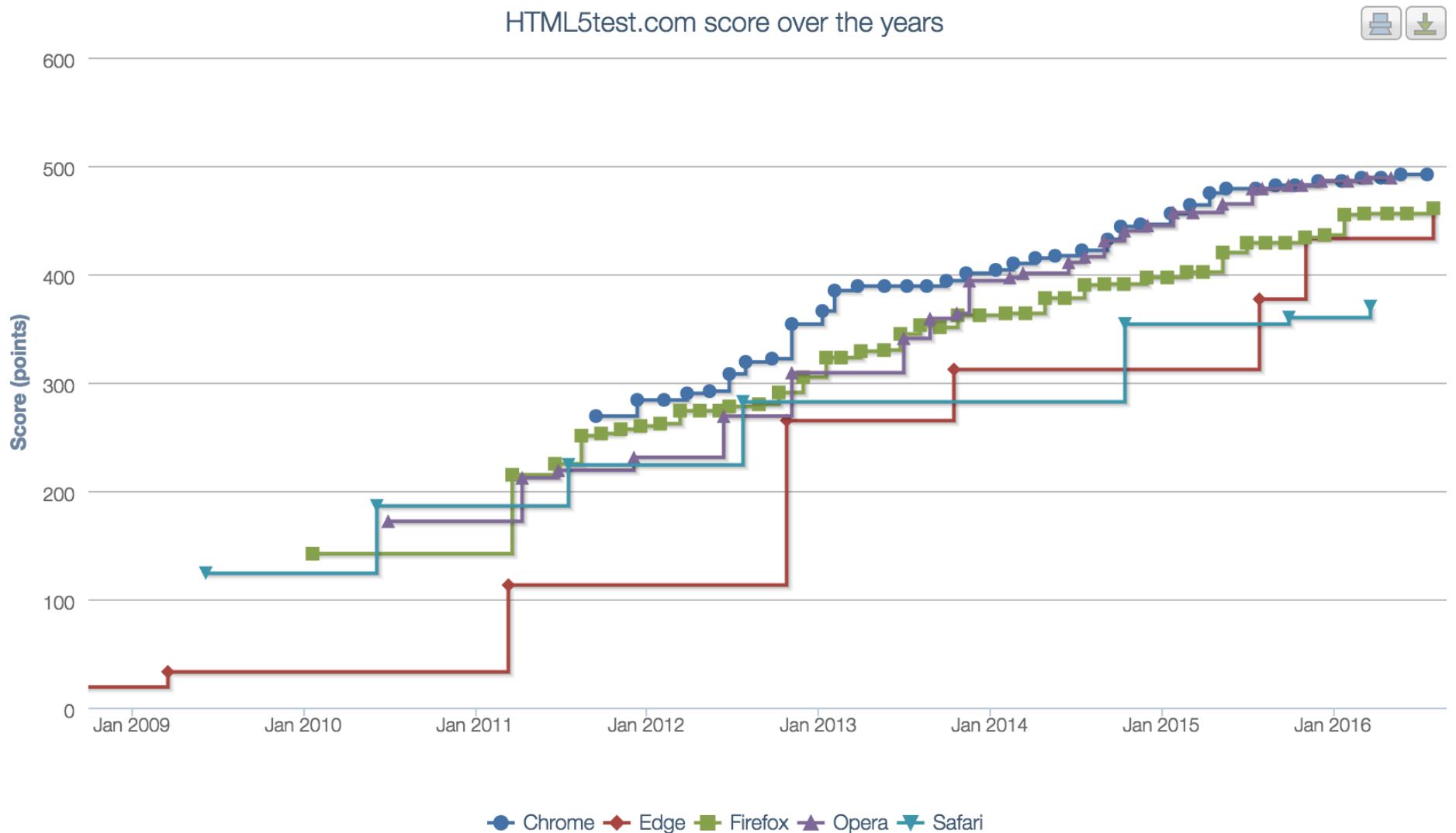
JavaScript Diagrams

Click here for GoJS

OTHER BROWSERS

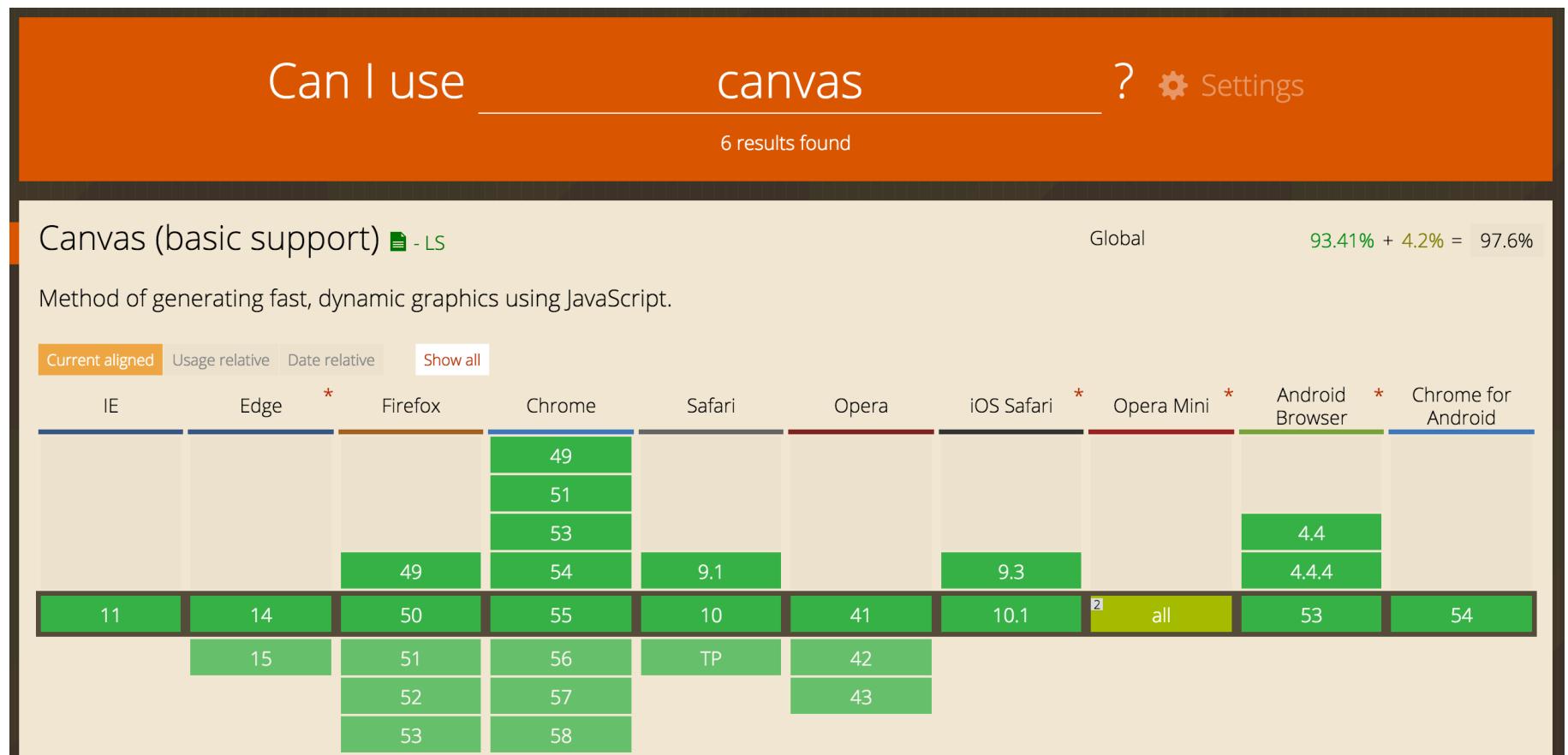
	Chrome	Opera	Firefox	Edge	Safari
Upcoming					
Current	52 ➔ 492		48 ➔ 461	14 ➔ 460	9.1 ➔ 370
Older	51 ➔ 492	37 ➔ 489	47 ➔ 456	13 ➔ 433	9.0 ➔ 360
	50 ➔ 489	30 ➔ 479	46 ➔ 456	12 ➔ 377	8.0 ➔ 354
	45 ➔ 482	12.10 ➔ 309	45 ➔ 456	Internet Explorer	
	40 ➔ 456		40 ➔ 429	11 ➔ 312	
	30 ➔ 394		35 ➔ 397	10 ➔ 265	
			30 ➔ 378	9 ➔ 113	

CHRONOLOGY



CAN I USE?

► caniuse.com



UNIVERSAL “CLIENT-SIDE” PLATFORM

- ▶ <https://www.w3.org/2006/webapi/admin/charter>
- ▶ Unlikely to be fully universal
- ▶ Innovation, by definition, is outside the norm (aka standards)

JQUERY

- ▶ very popular JS library
- ▶ Simplifies JS development in two ways
 - ▶ Easier to manipulate page elements
 - ▶ Portable API between browsers
- ▶ Provides additional UI / UX controls
- ▶ jqueryui.com

FULL CIRCLE JS RAW

- ▶ <http://youmightnotneedjquery.com/>

jQuery

IE9+

```
$(el).after(htmlString);
```

```
el.insertAdjacentHTML('afterend', htmlString);
```

```
$(parent).append(el);
```

```
parent.appendChild(el);
```

```
$(el).children();
```

```
el.children
```

```
$(el).empty();
```

```
el.innerHTML = "";
```

CODE VALIDATION

Technologie

HTML

CSS

JavaScript

Util

validator.w3.org

html5.validator.nu

jigsaw.w3.org/css-validator

javascriptlint.com

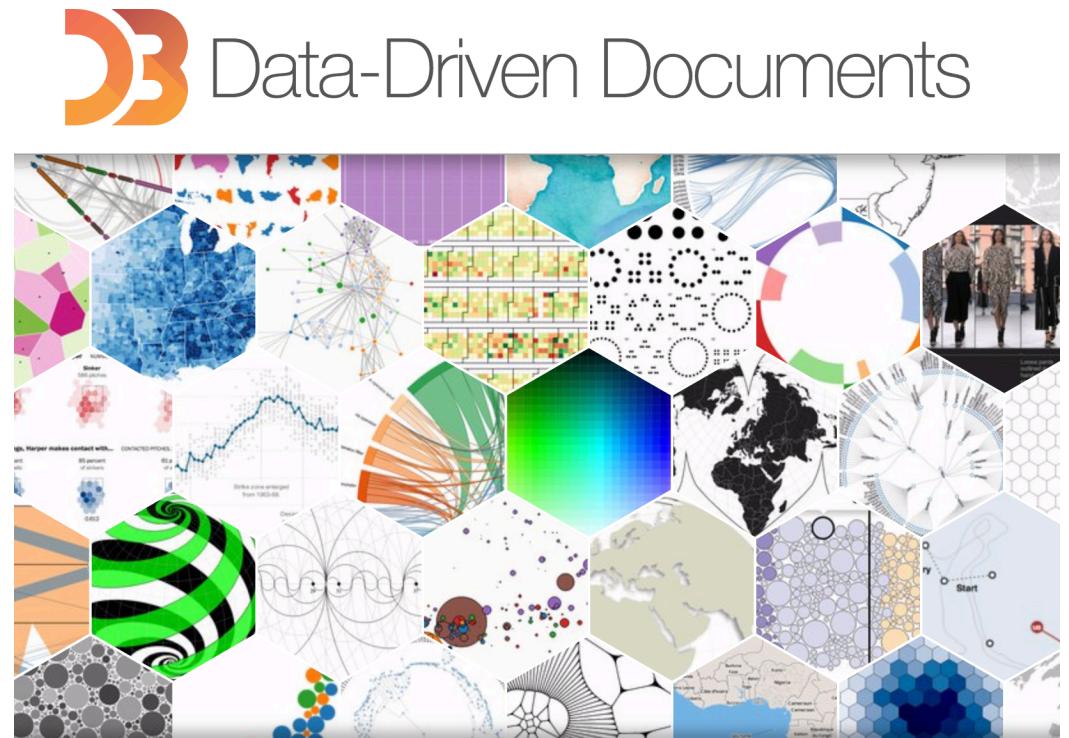
jslint.com

DEMOS

- ▶ [https://developer.mozilla.org/en-US/docs/Web/
Demos_of_open_web_technologies](https://developer.mozilla.org/en-US/docs/Web/Demos_of_open_web_technologies)
- ▶ <http://9elements.com/io/projects/html5/canvas/>
- ▶ [http://www.zachstronaut.com/lab/text-shadow-box/text-
shadow-box.html](http://www.zachstronaut.com/lab/text-shadow-box/text-shadow-box.html)
- ▶ <http://clublime.com/lab/html5/sphere/>
- ▶ <http://spielzeugz.de/html5/liquid-particles/>

D3.js

- ▶ <https://d3js.org/>
 - ▶ Brings data to life in
HTML, SVG and CSS



EVOLUTION OF THE INTERNET

- ▶ Convergence of IT and communications
- ▶ In the late 1960s, the ARPA (Advanced Research Projects Agency), funded by the United States Department of Defense, put in place plans to network the main computer systems of a dozen universities
- ▶ Communication by e-mail, instead of sharing computers with one another, becomes one of the first key advantages of ARPANET.

CHRONOLOGY 1965 – 1983

- 1965** Two computers communicate using "packet switching" technology at MIT Lincoln Lab.
- 1968** Beranek and Newman (BBN) won the ARPANET (Advanced Research Projects Agency) contract for the IMP (Interface Message Processor) specifications.
- 1969** On October 29, three universities install nodes and the first message is "LO", which was an attempt by student Charles Kline at "LOGIN". The message could not be completed because the system crashed.
- 1972** Ray Tomlinson of BBN presents the email.
- 1973** The term Internet was born
- 1974** The first Internet service provider (ISP) was born with the introduction of a commercial version of ARPANET, known as Telenet.
- 1974** Vinton Cerf and Bob Kahn (the fathers of the Internet) publish "a protocol for packet network interconnection", which details the design of TCP.
- 1979** USENET is created with news and discussion groups.
- 1982** TCP / IP is emerging as the protocol for ARPANET and remains the standard protocol for the Internet.
- 1983** The Domain Name System (DNS) establishes the .edu, .gov, .com, .mil, .org, .net and .int systems for naming websites.

CHRONOLOGY 1986 - 2009

1986 The NSFNET goes online with supercomputers connected to 56Kbps. The NSFNET was essentially a network of networks which linked university users with ARPANET.

1987 Cisco sells its first router. There are 20,000 Internet hosts.

1990 Tim Berners-Lee, a CERN scientist, develops HTML

1991 CERN introduces the World Wide Web to the public.

1994 Tim Berners-Lee creates World Wide Web Consortium (W3C). Netscape Communications was born.

1998 Google was born

2001 Federal judge arrests Napster, saying he must find a way to prevent users from sharing copyrighted material online.

2004 Facebook is born, and Firefox is created (replacing Netscape)

2005 YouTube.com launched

2009 The Internet marks its 40th anniversary.

GOOGLE

- ▶ In 1996, at Stanford Ph.D. in computer science, the candidates Larry Page and Sergey Brin started to collaborate on a new search engine.
- ▶ In 1997, they chose the name Google - a play on the mathematical word googol, a number which is equal to 1 followed by 100 zeros (or 10^{100}), an enormously large number.
- ▶ Google's ability to return extremely accurate search results quickly has helped it become the most widely used search engine and one of the most popular websites in the world.

A BIT

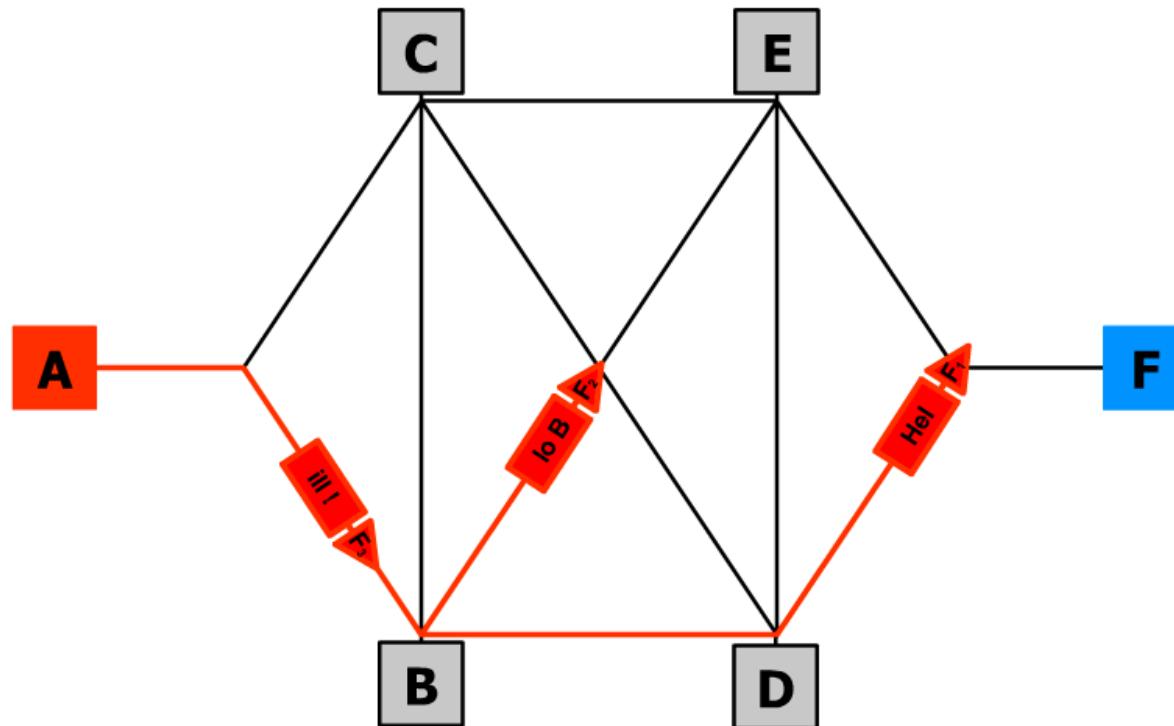
- ▶ A bit (abbreviation of "binary digit") is the smallest data on a computer
- ▶ It can take the value 0 or 1.
- ▶ A byte ("byte") is 8 bits
- ▶ One Megabyte is 1,048,576 bytes or 1,024 Kilobytes
- ▶ Watch out for rounding errors like 1000 bytes != 1 kilobyte

PACKET SWITCHING

- ▶ One of the main objectives of ARPANET was to allow several users to send and receive information simultaneously on the same communication paths (for example, telephone lines).
- ▶ The network worked by sending data in small pieces called packets, containing
 - ▶ Address,
 - ▶ Error checking
 - ▶ Sequencing information

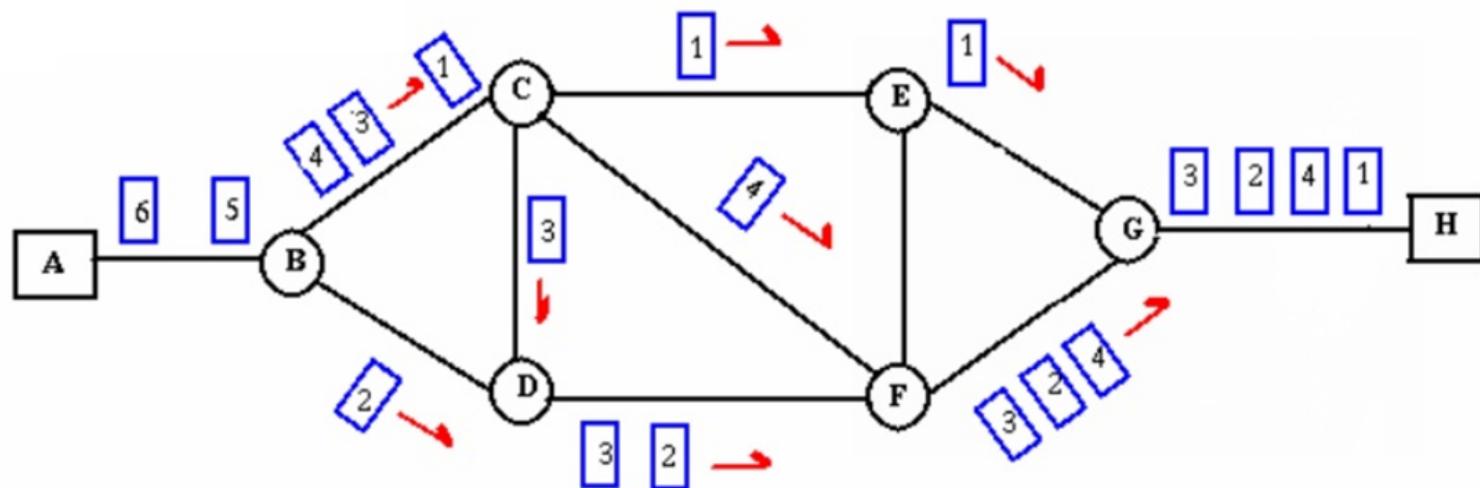
PACKET ADDRESSES

- ▶ The address information allowed routing of packets to their destinations.



SEQUENCING INFORMATION

- ▶ Sequencing information helps reassemble packets into their original order for presentation to the recipient
- ▶ Due to complex routing mechanisms, packets could actually arrive out of order



DECENTRALIZED BY DESIGN

- ▶ Packets of different messages were mixed on the same lines to efficiently use the available bandwidth.
- ▶ The network was designed to operate without centralized control.
- ▶ If part of the network fails, the working portions can continue to pass message packets to receivers on alternative routes.

TCP – TRANSPORT CONTROL PROTOCOL

- ▶ The communication protocol on ARPANET
- ▶ TCP ensures messages are received intact
 - ▶ Correct order
 - ▶ Complete reception
- ▶ One of the challenges was to communicate between the "intra" and "inter" networks.

IP - INTERNET PROTOCOL

- ▶ Developed by ARPA, the IP created the protocol basis for a "network of networks"
- ▶ Each connected device has a unique IP address

Nom	Exemple	Détails
IPv4	192.168.0.1	Established in 1981 (1984), based on 32 bits (therefore 4 billion unique addresses)
IPv6	2001:0db8:85a3:0000:0000:08a2:e037:7334	Introduced in 1998 (last change in 2006) and based on 128 bits (3.4×10^{38} unique addresses or 6.7×10^{23} per square meter on earth)

WHY NOT SWITCH TO IPv6?

- ▶ Different networks requiring software and hardware upgrades
- ▶ No tangible reasons to improve, only the costs
 - ▶ All the useful features of IPv6 have been added to IPv4
 - ▶ Internal networks are generally unaffected
 - ▶ Final IPs submitted in 2011
 - ▶ Scarcity finally becomes a reality

INTRANET

- ▶ A network of computers within an organization
- ▶ LAN (Local Area Network)
- ▶ RFC 1918
 - ▶ Class A 10.0.0.0/8 (10.0.0.1 to 10.255.255.254)
 - ▶ Class B 172.16.0.0/12 (172.16.0.1 to 172.31.255.254)
 - ▶ Class C 192.168.0.0/16 (192.168.0.1 to 192.168.255.254)

IP ADDRESS CLASSES

Class	IP Address	Network IP	Host IP
A	a.b.c.d	a	b.c.d
B	a.b.c.d	a.b	c.d
C	a.b.c.d	a.b.c	d

WHAT IS A COMPUTER PROTOCOL?

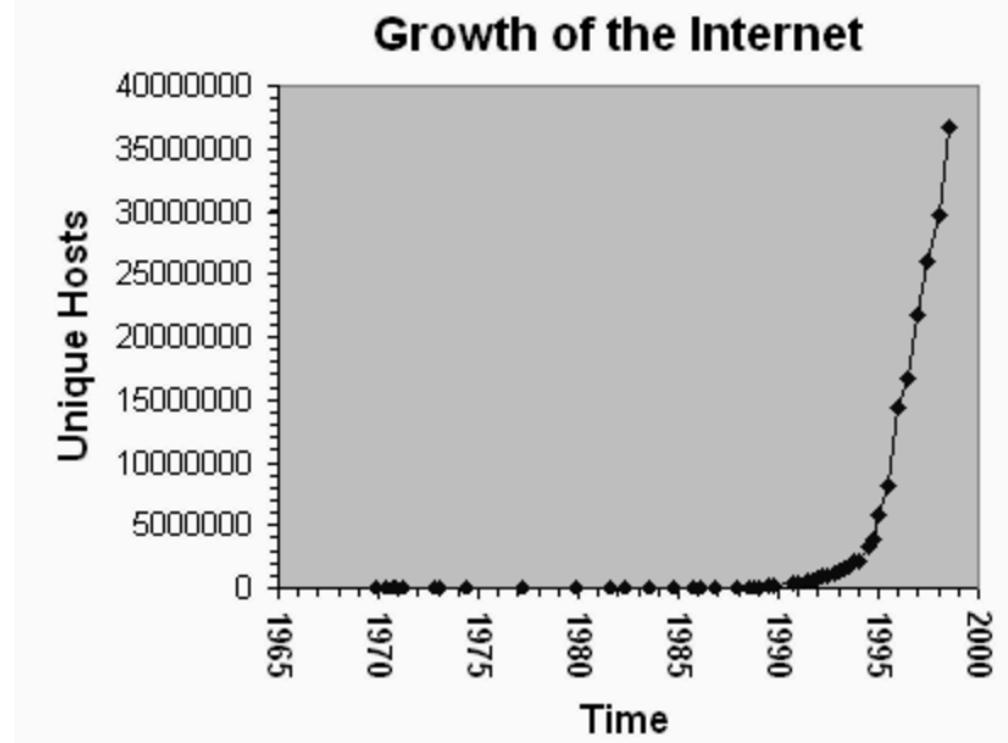
- ▶ A set of rules in which computers communicate with each other
- ▶ The protocol indicates which part of the conversation is happening at this time
- ▶ It also shows how to end the communication

PROTOCOL EXAMPLES

Abbr	Name	Details
TCP	Transmission Control Protocol	A main protocol in the Internet
UDP	User Datagram Protocol	Used to establish low latency and loss tolerant connections between applications
IP	Internet Protocol	Data is transmitted from one computer to another over the Internet
FTP	File Transfer Protocol	Used to transmit files between devices
HTTP	Hypertext Transfer Protocol	A standard method and format for transferring data over the Internet
HTTPS	HTTP over SSL ou TLS	Encrypted data sent over HTTP
SMTP	Simple Mail Transfer Protocol	Sending and receiving emails

EXPLOSIVE DEVELOPMENT

- ▶ Initially limited to universities and research institutes
- ▶ Then the military started using it extensively
- ▶ Ultimately, the government decided to allow Internet access for commercial purposes.



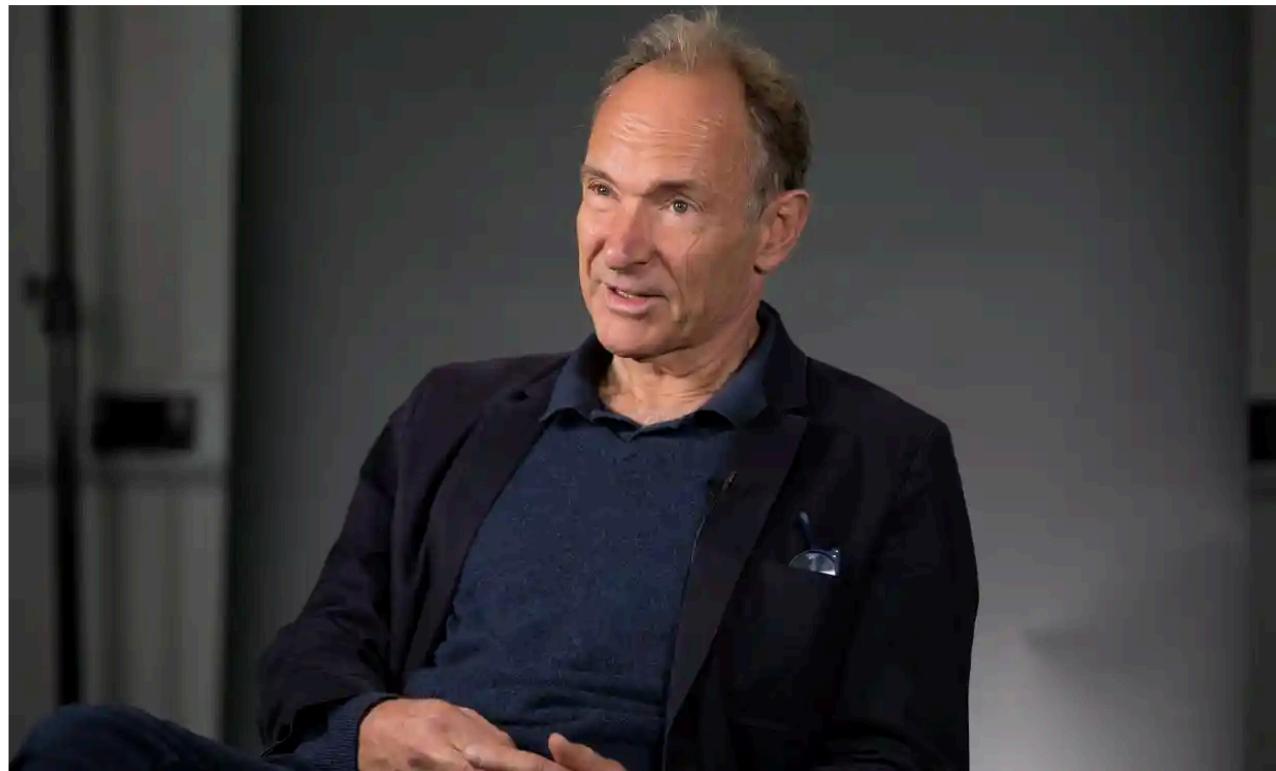
TIM BERNERS-LEE

- ▶ Invented HTML in 1989
- ▶ Wrote the HTTP protocol for WWW
- ▶ Defined URL (Uniform Resource Locator)
 - ▶ Locate a document anywhere on the Internet
 - ▶ <https://www.uottawa.ca/graduate-studies/programs-admission/programs/computer-science>

WHAT IS TIM DOING THESE DAYS?

Tim Berners-Lee launches campaign to save the web from abuse

A 'Magna Carta for the web' will protect people's rights online from threats such as fake news, prejudice and hate, says founder of the world wide web



▲ Berners-Lee says urgent action is needed to protect users' privacy so they can be online 'freely, safely and without fear'. Photograph: Simon Dawson/Reuters

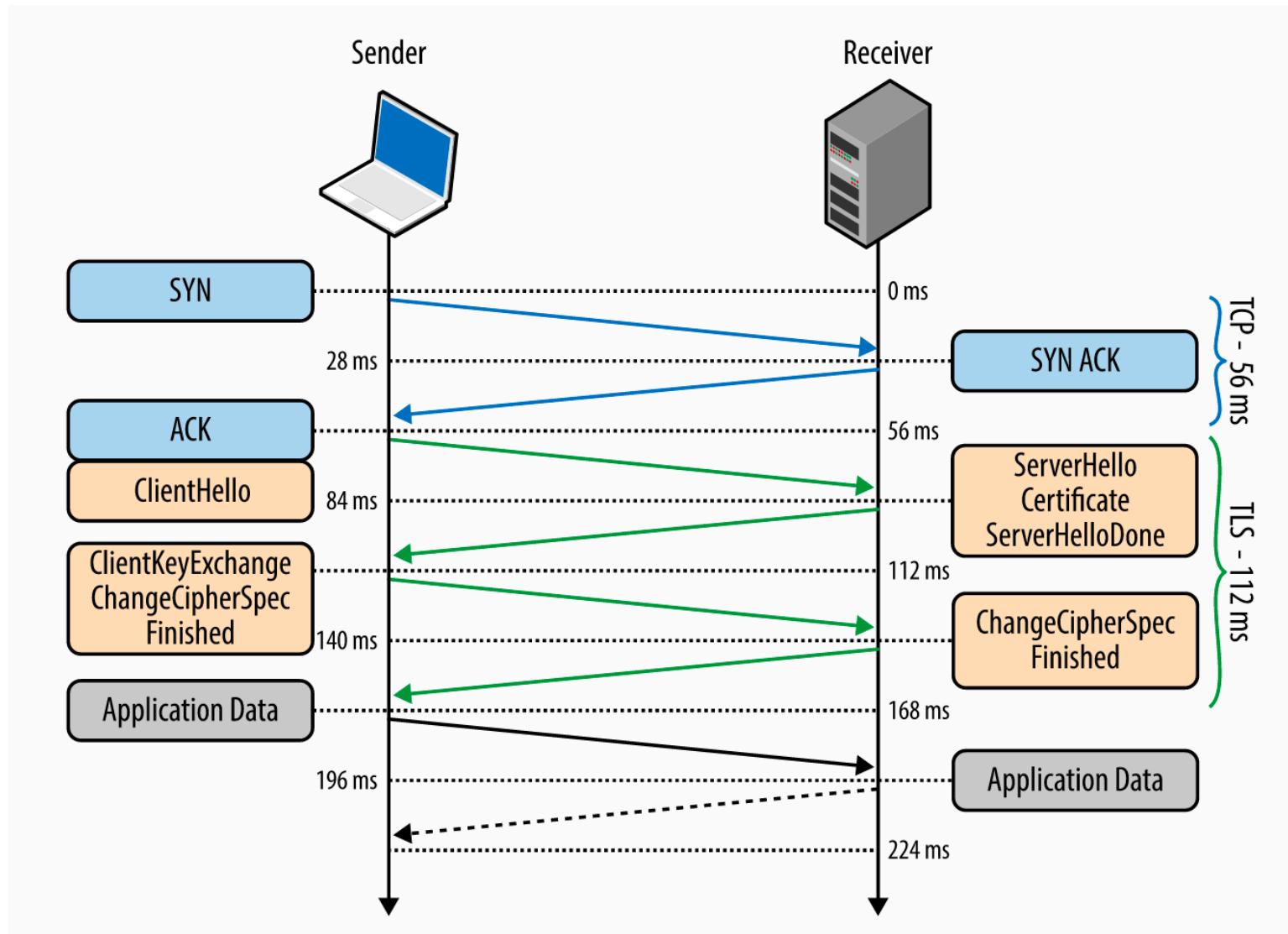
<https://www.theguardian.com/technology/2018/nov/05/tim-berners-lee-launches-campaign-to-save-the-web-from-abuse>

Sir Tim Berners-Lee, the creator of the World Wide Web, has confessed that the // in a web address were actually "unnecessary".

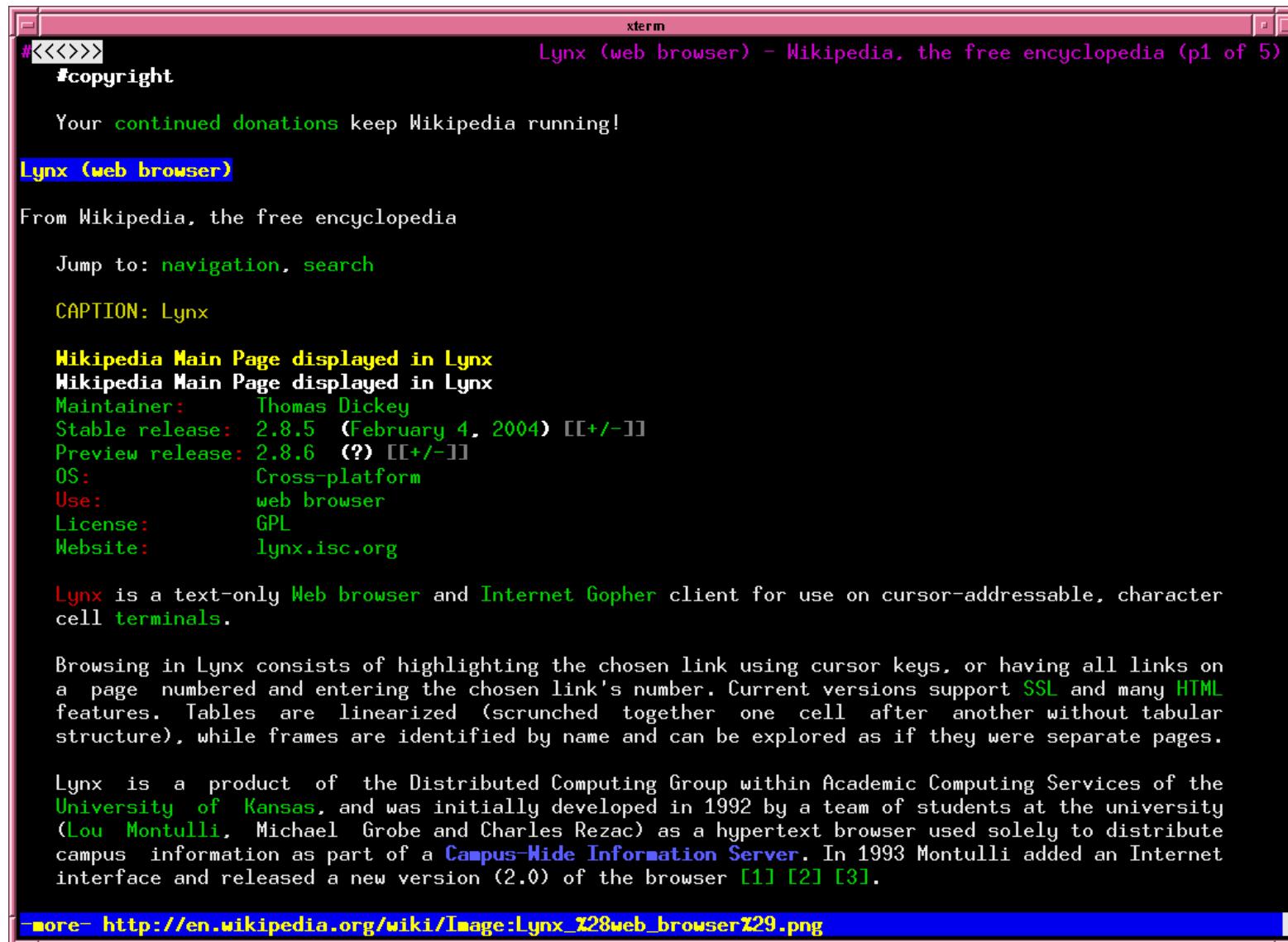
HTTPS

- ▶ HTTP +
 - ▶ SSL (Secure Socket Layer) or
 - ▶ TLS (Transport Layer Security)
- ▶ Must be used for all traffic, not just sensitive traffic
 - ▶ All traffic is sensitive?

TLS HANDSHAKE

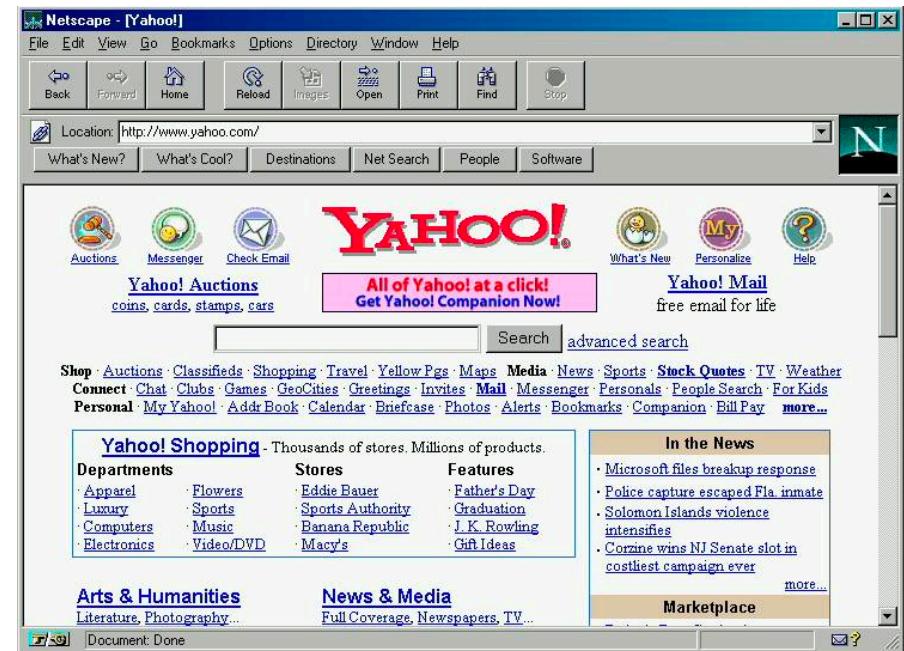


WEB 1.0



WEB 2.0

- ▶ The use of the Web exploded with the availability in 1993 of the Mosaic browser, which presented a user-friendly graphical interface.
- ▶ Developed by Marc Andreessen at NCSA who then founded Netscape
- ▶ Dotcom bust in 2000 put a temporary decline in the commercial use of the internet
- ▶ The 2003 resurgence of the Internet was * Web *, as named by Dale Dougherty (and Tim O'Reilly)
- ▶ Collaborative and community websites (social media, wikis, blogs)



WEB 3.0

- ▶ Nova Spivack defines it as the "Semantic Web", intelligently connecting data, concepts, applications and people
- ▶ The first rule of Web 3.0 is to stop calling it that
- ▶ No need for a version number, rather talk about the intention
 - ▶ Mobile First?
 - ▶ IoT (Internet of Things)?
 - ▶ Semantic web

WAIT, SO WHAT'S WEB3?

My first impressions of web3

Jan 07, 2022

Despite considering myself a cryptographer, I have not found myself particularly drawn to “crypto.” I don’t think I’ve ever actually said the words “get off my lawn,” but I’m much more likely to click on *Pepperidge Farm Remembers* flavored memes about how “crypto” used to mean “cryptography” than I am the latest NFT drop.

Also – cards on the table here – I don’t share the same generational excitement for moving all aspects of life into an instrumented economy.

Even strictly on the technological level, though, I haven’t yet managed to become a believer. So given all of the recent attention into what is now being called web3, I decided to explore some of what has been happening in that space more thoroughly to see what I may be missing.

<https://moxie.org/2022/01/07/web3-first-impressions.html>

WEB X.0 COMPARISONS

- ▶ Web 1.0 is like a conference, a small number of teachers inform a large audience of students.
- ▶ Web 2.0 is a conversation, with everyone able to speak and share views.
- ▶ Web 3.0 is like “notebooks” for conversations as there are now too many people talking (IoT, AI, semantic web)

DATA DRIVEN ARCHITECTURE

- ▶ Using collective intelligence, the concept that a large and diverse group of people will create intelligent ideas
- ▶ Using machine learning and Big Data for news retrieval, language translators and personal assistants

BASIC CONCEPT ON THE WEB

index.htm

```
1 <html>
2   <head>
3     <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
4     <link rel="stylesheet" href="scripts/base.css" id="theme">
5     <title>CSI 3540 - Structures, techniques et normes du web</title>
6   </head>
7   <body class="syllabus">
8     <h1>CSI 3540</h1>
9     <h2>Structures, techniques et normes du web</h2>
10    <hr/>
11    <h3>Description</h3>
12    <p>
13      Infrastructure de base du Web. Serveurs et navigateurs. Exemples de protocoles.
14      Internet et virus. Architecture de moteur de recherche. Contenu et présentation Web.
15      Pages Web, leur structure et leur interprétation. HTML, XML et leurs dérivés.
16      Interfaces Web vers les logiciels et bases de données. Témoins et droit à la vie privée
17      Web sémantique et ontologies. Services Web.
18    </p>
19
20    <h3>
21      Préalables :
22    </h3>
23    <p>
24      CSI2510,
25      CSI2532
26    </p>
27
28    <h3>Professeur</h3>
29    <table>
30      <tr>
31        <th>Enseignante</th>
32        <td>Andrew Forward</td>
33      </tr>
34      <tr>
35        <th>Courriel</th>
36        <td>
37          <a href="mailto:aforward@uottawa.ca">aforward@uottawa.ca</a><br />
38          <a href="mailto:aforward@gmail.com">aforward@gmail.com</a></td>
39        </td>
40      </tr>
41      <tr>
42        <th>Site web</th>
43        <td>
44          <a href="https://uottawa.blackboard.com" target="_blank" rel="noopener">Collapse 2</a>
45        </td>
46      </tr>
47    </table>
48  </body>
49</html>
```

HTML Document

Hyperlinks

Internet

Web server

Website 1

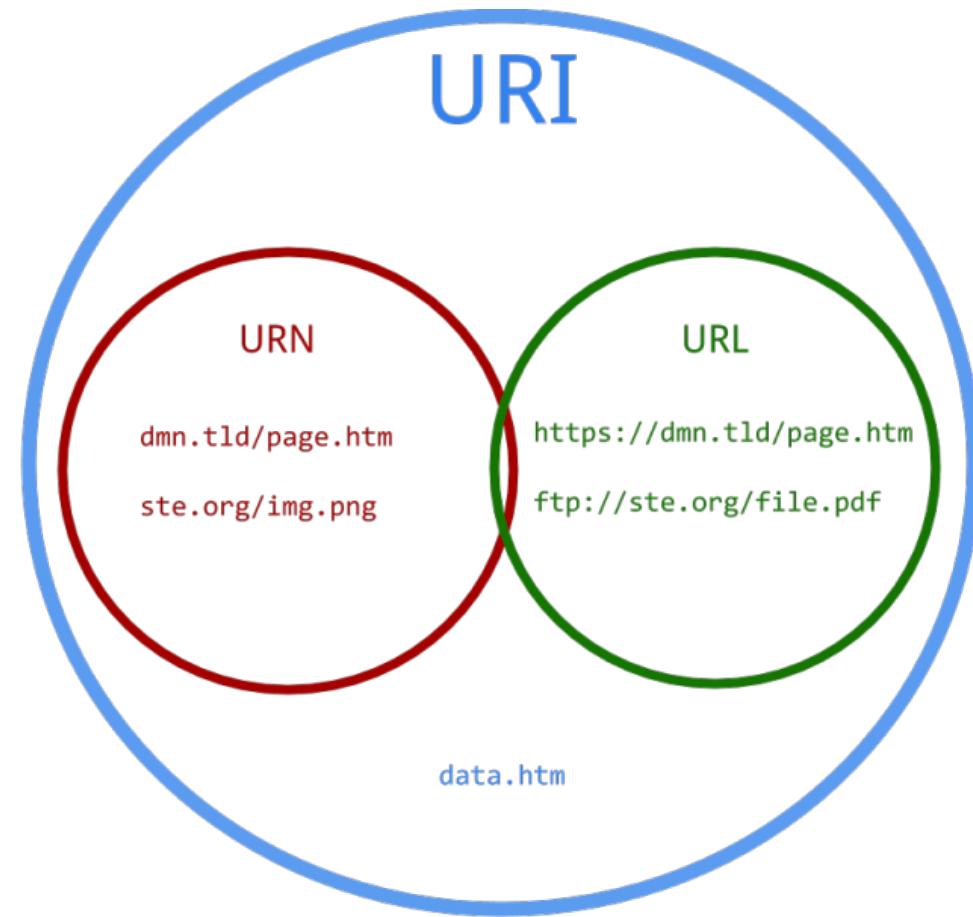
Website 1

HYPERLINKS

- ▶ <mailto:aforward@gmail.com>
- ▶ <http://a4word.com>
- ▶ //scripts/base.css
- ▶ Prefix “//” to keep the same protocol (for example http versus https)

URI = URN + URL + ...

- ▶ Uniform Resource Identifier
- ▶ Uniform Resource Name
 - ▶ Identity of the object
- ▶ Uniform Resource Locator
 - ▶ Name + Access method
- ▶ All healthy butterflies can fly,
but not everything that can fly
is a healthy butterfly

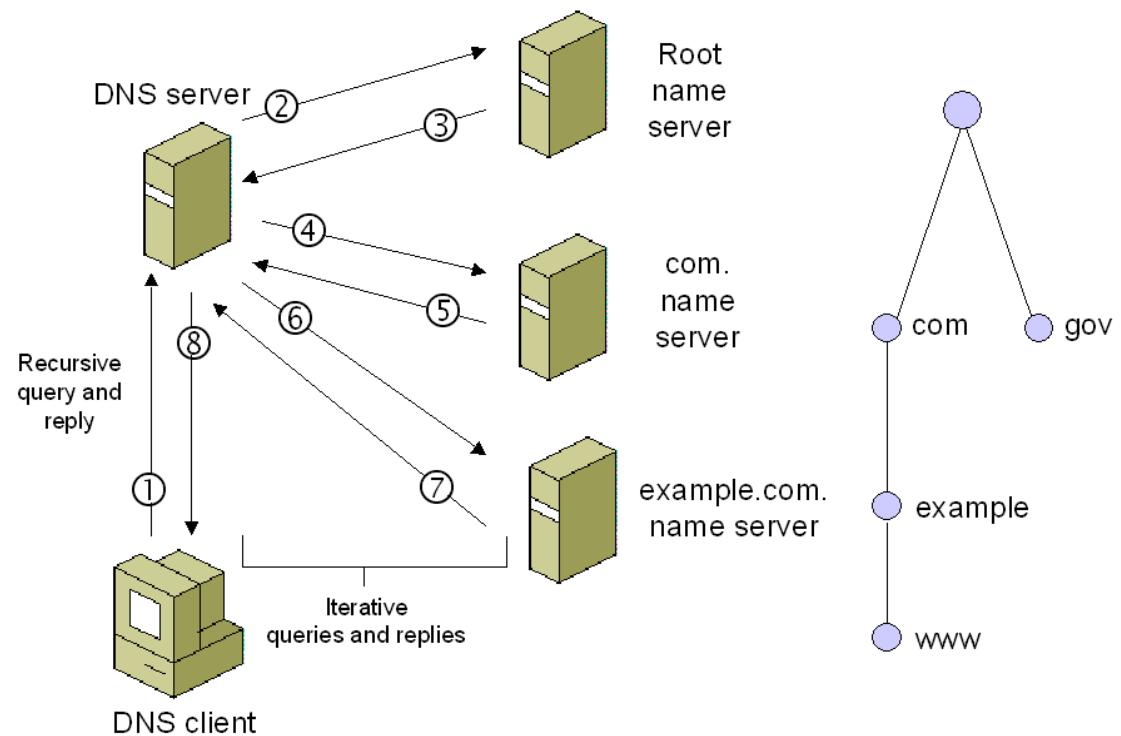


MIME

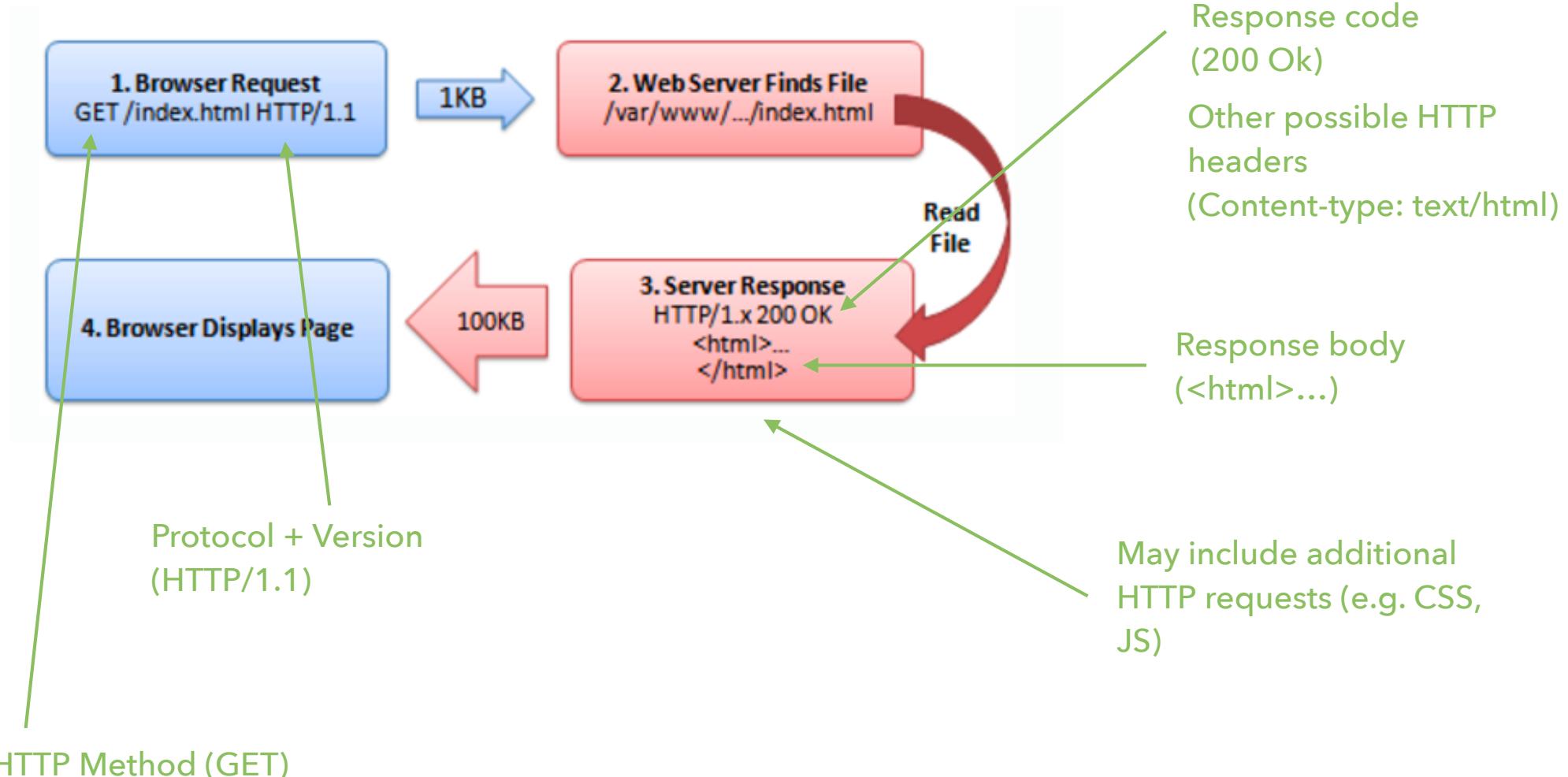
- ▶ Multipurpose Internet Mail Extensions
- ▶ The type of body that the server transmits to the browser
- ▶ Specifies data formats, which programs can be used to correctly interpret the data
 - ▶ text / plain
 - ▶ image / jpeg

DNS (DOMAIN NAME SERVER)

- ▶ A database of host names and their corresponding IP addresses that can automatically perform URL (to IP) translations.



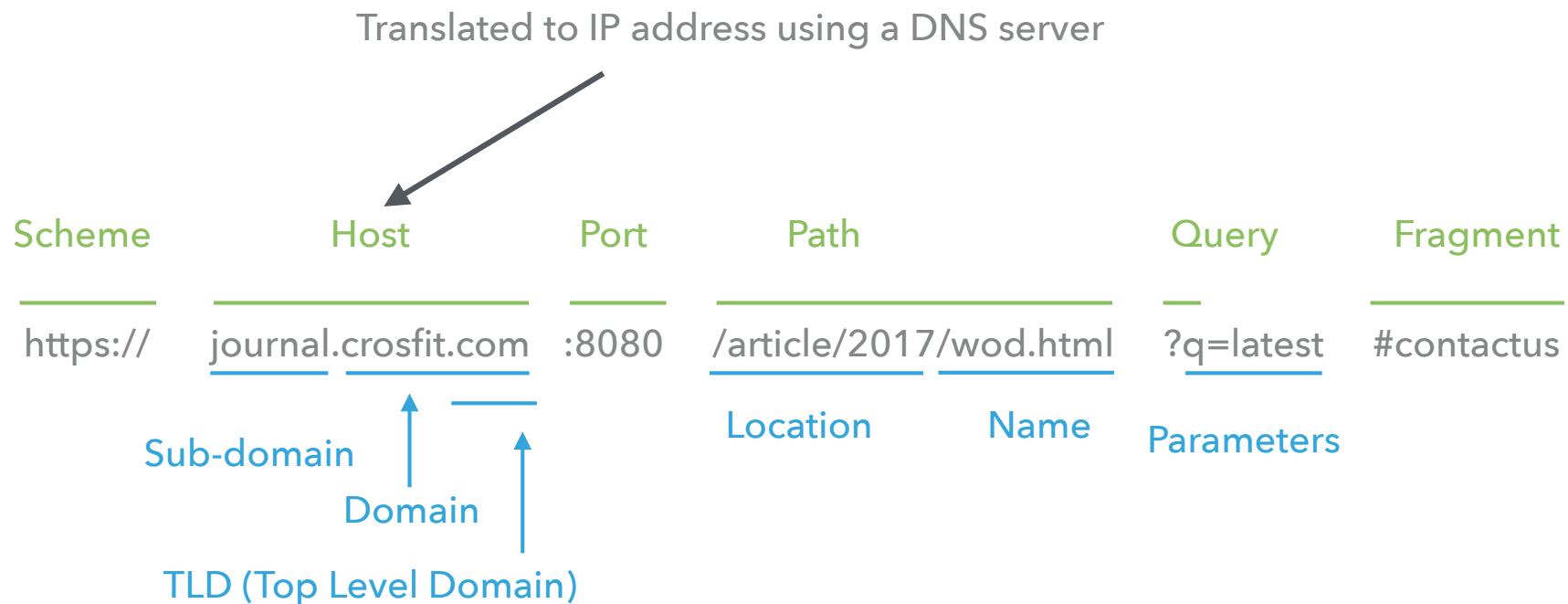
REQUEST A DOCUMENT



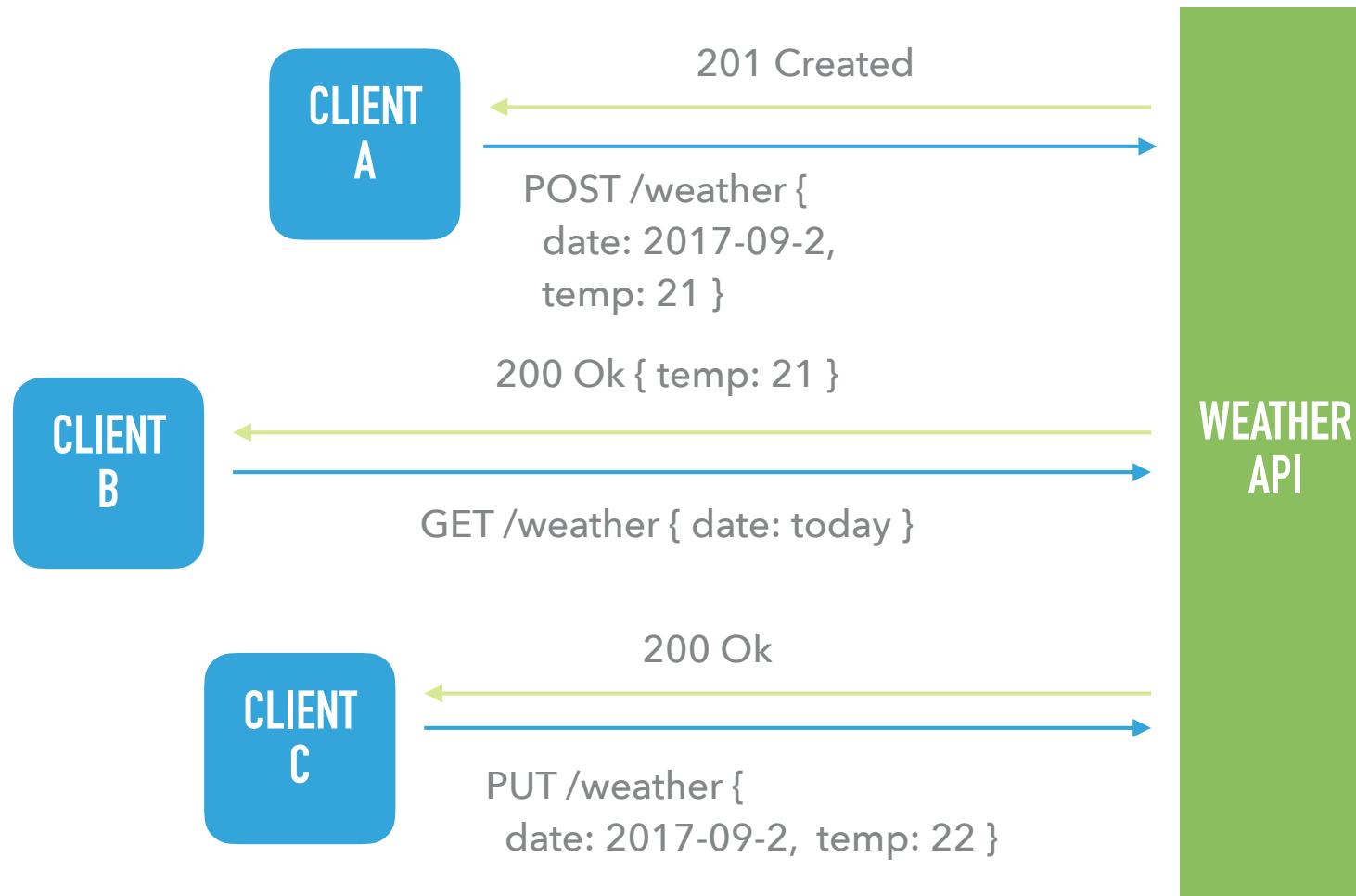
COMMON MIME TYPES

Ext	type MIME	Ext	type MIME	Ext	type MIME
.avi	video/msvideo	.jar	application/java-archive	.tar.gz	application/x-tar
.bmp	images/bmp	.jpeg	image/jpeg	.txt	text/plain
.dtd	application/xml-dtd	.js	application/x-javascript	.xls	application/vnd.ms-excel
.doc	application/msword	.mp3	audio/mpeg	.xml	application/xml
.exe	application/octet-stream	.mpeg	video/mpeg	.zip	application/zip
.gif	image/gif	.pdf	application/pdf		
.gz	application/x-gzip	.png	image/png		
.html	text/html	.ppt	application/vnd.ms-powerpoint		

PARTIES D'UNE URL



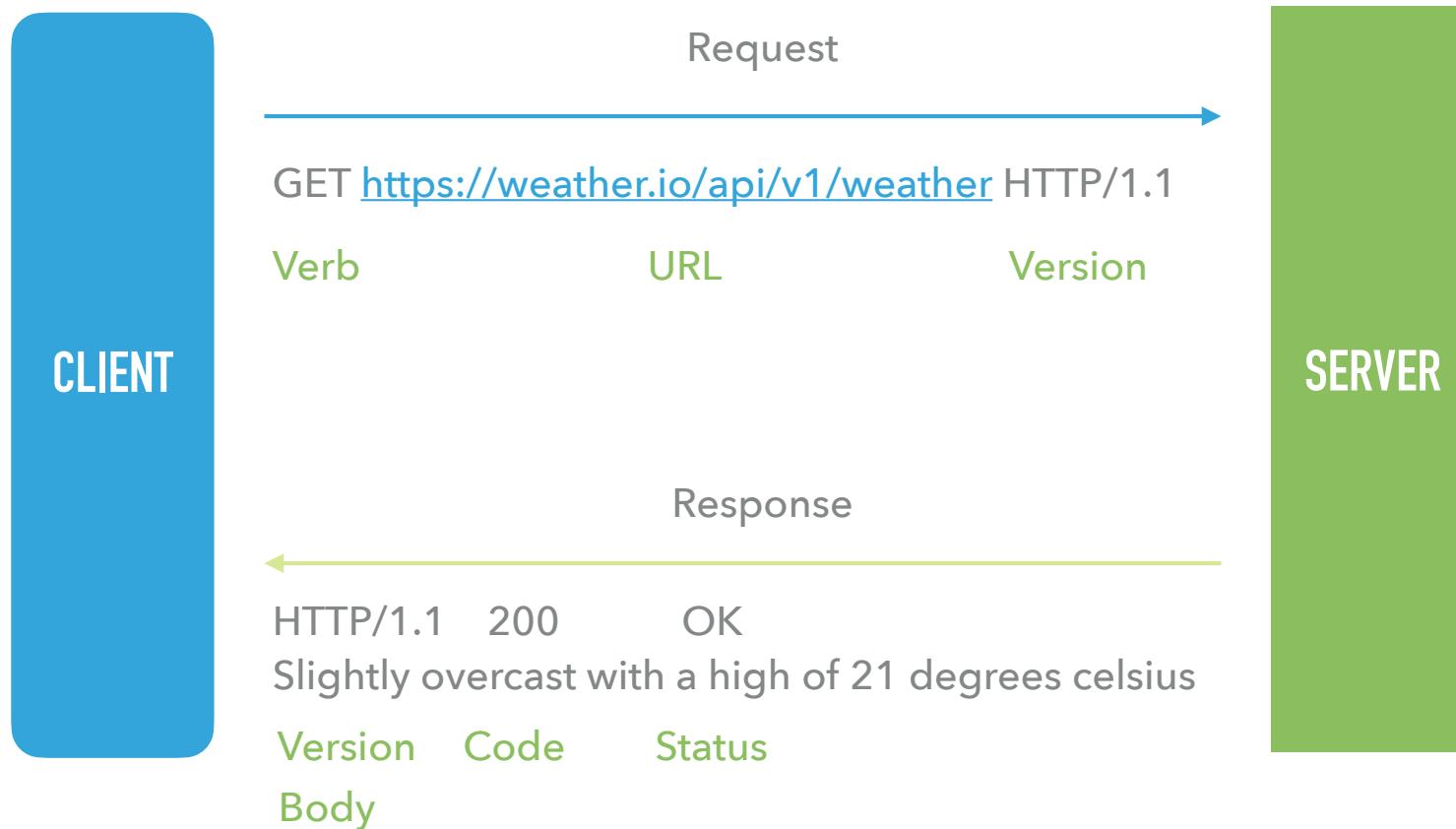
HTTP API REQUEST



REST API

- ▶ Representational State Transfer (REST) was defined by Roy Thomas Fielding in his 2000 doctoral thesis entitled "Architectural Styles and the Design of Network-based Software Architectures" at UC Irvine.
- ▶ Not all REST APIs are the same, which leads to a maturity model to help describe the level of RESTfulness
- ▶ JSON the unofficial standard for the request / response body (aka your data)

LEVEL 0: HTTP TRANSPORT



LEVEL 1: RESOURCES

- ▶ We divide the “end-points” according to the resources.

Resource	End-point (aka URL)
Account	https://api.digitalocean.com/v2/account
Droplets	https://api.digitalocean.com/v2/droplets
Images	https://api.digitalocean.com/v2/images
Snapshots	https://api.digitalocean.com/v2/snapshots
Regions	https://api.digitalocean.com/v2/regions

LEVEL 2: VERBS AND STATUS CODES

- ▶ Consistent use of HTTP verbs (aka methods) and status codes.

Verb	Status	Details
GET	200 Ok	Get data only, shouldn't change anything
POST	201 Created	Create a new record for the resource provided
PUT	200 Ok	Replace an existing resource (aka update)
PATCH	200 Ok	Edit an existing resource (aka update)
DELETE	204 No Content	Delete a resource

LEVEL 2: MORE STATUS CODES

1xx Informational

100 Continue

101 Switching Protocols

102 Processing (WebDAV)

2xx Success

★ 200 OK

203 Non-Authoritative Information

206 Partial Content

226 IM Used

★ 201 Created

★ 204 No Content

207 Multi-Status (WebDAV)

202 Accepted

205 Reset Content

208 Already Reported (WebDAV)

3xx Redirection

300 Multiple Choices

303 See Other

306 (Unused)

301 Moved Permanently

★ 304 Not Modified

307 Temporary Redirect

302 Found

305 Use Proxy

308 Permanent Redirect (experimental)

4xx Client Error

★ 400 Bad Request

★ 403 Forbidden

406 Not Acceptable

★ 409 Conflict

412 Precondition Failed

415 Unsupported Media Type

418 I'm a teapot (RFC 2324)

423 Locked (WebDAV)

426 Upgrade Required

431 Request Header Fields Too Large

450 Blocked by Windows Parental Controls (Microsoft)

★ 401 Unauthorized

★ 404 Not Found

407 Proxy Authentication Required

410 Gone

413 Request Entity Too Large

416 Requested Range Not Satisfiable

420 Enhance Your Calm (Twitter)

424 Failed Dependency (WebDAV)

428 Precondition Required

444 No Response (Nginx)

451Unavailable For Legal Reasons

402 Payment Required

405 Method Not Allowed

408 Request Timeout

411 Length Required

414 Request-URI Too Long

417 Expectation Failed

422 Unprocessable Entity (WebDAV)

425 Reserved for WebDAV

429 Too Many Requests

449 Retry With (Microsoft)

499 Client Closed Request (Nginx)

5xx Server Error

★ 500 Internal Server Error

503 Service Unavailable

506 Variant Also Negotiates (Experimental)

509 Bandwidth Limit Exceeded (Apache)

598 Network read timeout error

501 Not Implemented

504 Gateway Timeout

507 Insufficient Storage (WebDAV)

510 Not Extended

599 Network connect timeout error

502 Bad Gateway

505 HTTP Version Not Supported

508 Loop Detected (WebDAV)

511 Network Authentication Required

LEVEL 3: HYPERMEDIA

- ▶ HATEOAS (Hypertext As The Engine Of Application State)
- ▶ Use of links in the response

```
"links": {  
    "actions": [  
        {  
            "id": 36805096,  
            "rel": "create_multiple",  
            "href": "https://api.digitalocean.com/v2/actions/36805096"  
        }  
    ]  
}
```

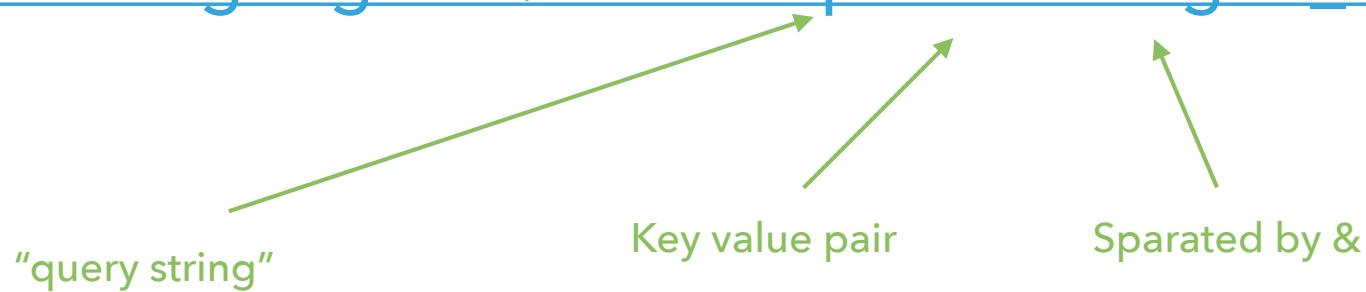


Additional link data telling the user where they can go next

Using href data instead of hardcoding the resource directly in the client user interface

GET METHOD

▶ https://www.google.ca/search?q=a4word&gws_rd=cr,ssl



POST METHOD

The screenshot shows the Postman application interface. At the top, there is a header bar with the method **POST** and the URL **www.example.com/users**. Below the header, there are tabs for **Authorization**, **Headers (1)**, **Body** (which is selected and highlighted in blue), **Pre-request Script**, and **Tests**. Under the **Body** tab, there are five options: **form-data**, **x-www-form-urlencoded**, **raw** (which is selected and highlighted in orange), **binary**, and **JSON (application/json)**. The **JSON (application/json)** option has a dropdown arrow next to it. Below these settings, the request body is displayed as a JSON object:

```
1 {  
2   "name": "Andrew",  
3   "email": "aforward@gmail.com"  
4 }
```

Two green arrows point from the text "The data in the body of the request" to the JSON object and from the text "The data format depends on the API, usually JSON" to the word "JSON" in the "JSON (application/json)" dropdown.

The data in the body of the request

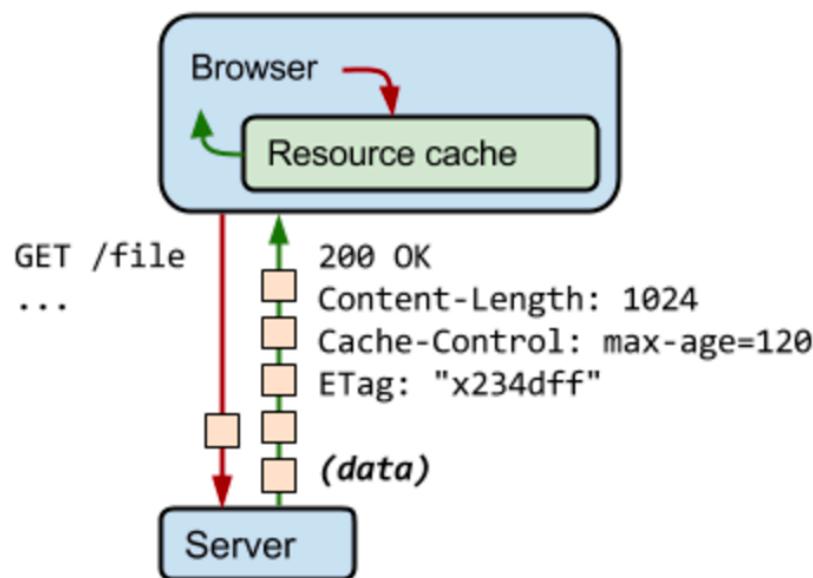
The data format depends on the API, usually JSON

MASHUPS

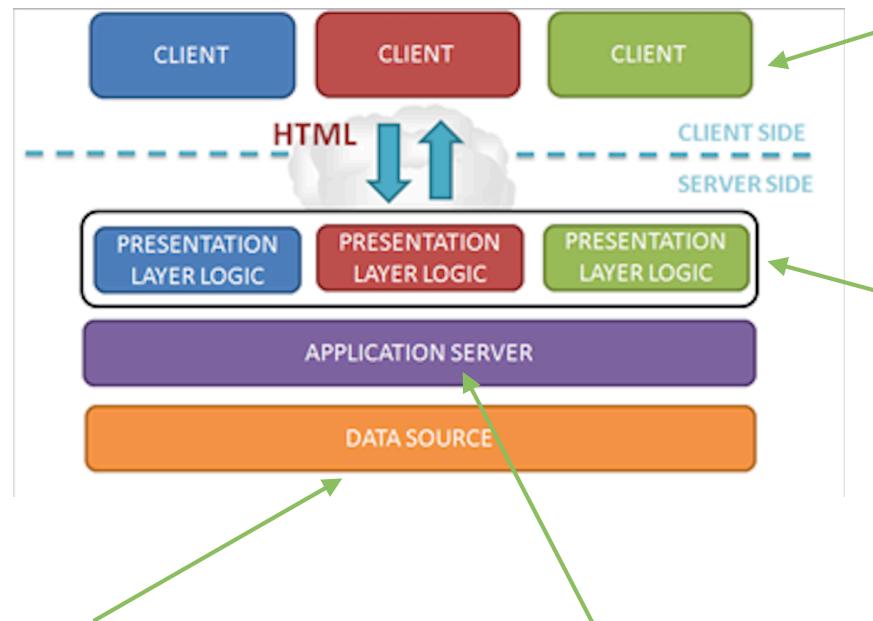
Web Service	URL
Google Maps	https://developers.google.com/maps/documentation/javascript/tutorial
GitHub	http://developer.github.com/
Foursquare	https://developer.foursquare.com/
Gravatar	http://en.gravatar.com/site/implement/
Instagram	http://instagram.com/developer/
Meetup	http://www.meetup.com/meetup_api/
Wolfram Alpha	http://products.wolframalpha.com/api/
YouTube	https://developers.google.com/youtube/v3/
Etsy	http://www.etsy.com/developers/documentation/getting_started/jsonp
Netflix	https://twitter.com/netflixapi?lang=en
Wikipedia	https://en.wikipedia.org/w/api.php
LinkedIn	https://developer.linkedin.com/docs/rest-api

CLIENT SIDE CACHE

- ▶ Browsers cache recently viewed documents (web page) for fast reloading



N-TIER APPLICATION ARCHITECTURE



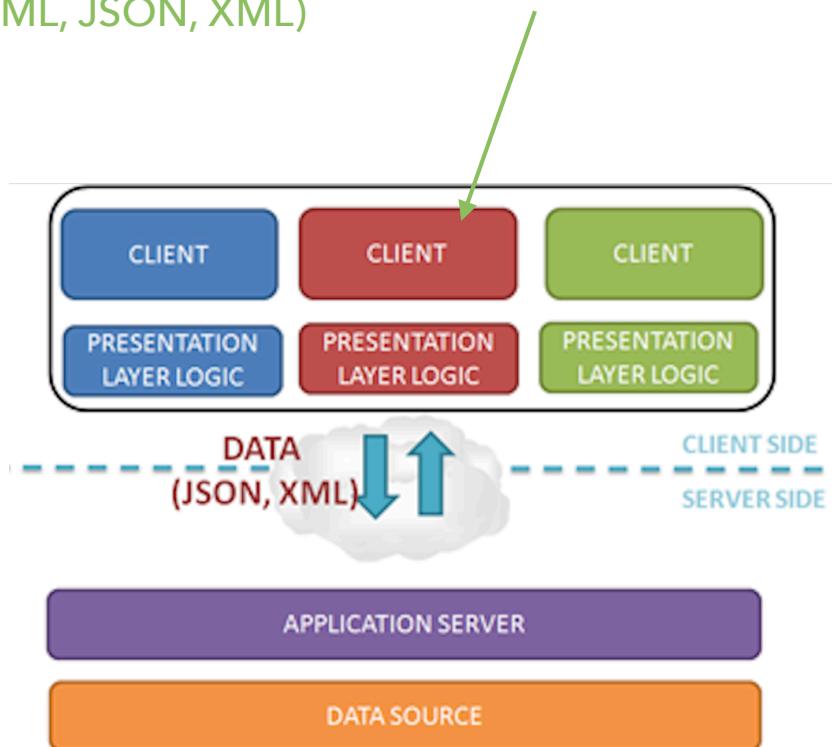
Database
(SQL, NoSQL, NewSQL)

Company and
controller logic

User interface /
browser

Presentation logic
(HTML, JSON, XML)

More recently, frameworks are
pushing user interface
rendering to the client



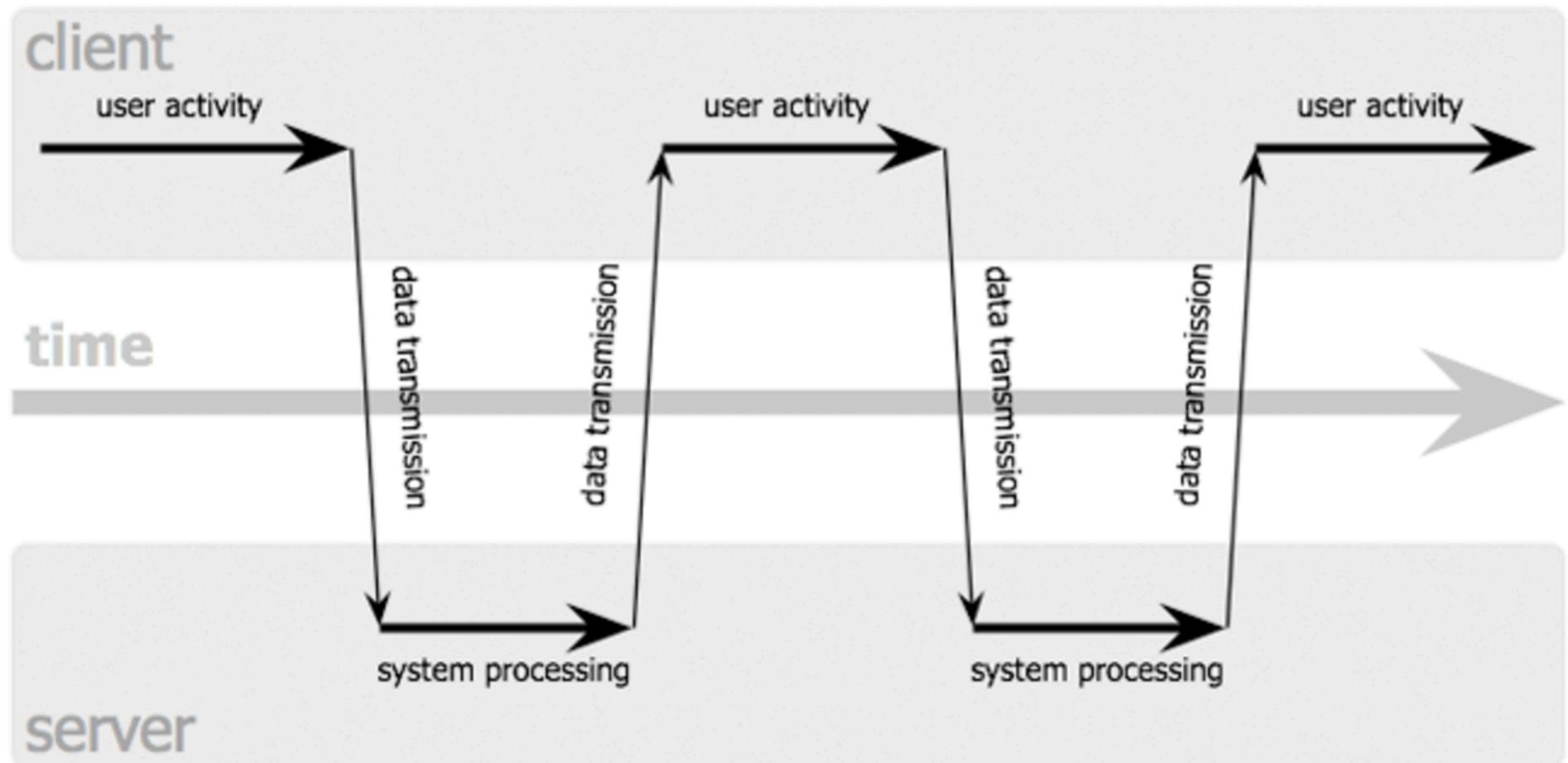
CLIENT-SIDE SCRIPTING

- ▶ JavaScript can be used to validate entries, draw the interface, and send additional requests (AJAX / Web sockets)
- ▶ But browser dependency and availability of required functionality are issues
- ▶ Code is also openly visible to the customer, and can be overwritten by the customer
- ▶ Validations must therefore be processed in the same way on the server

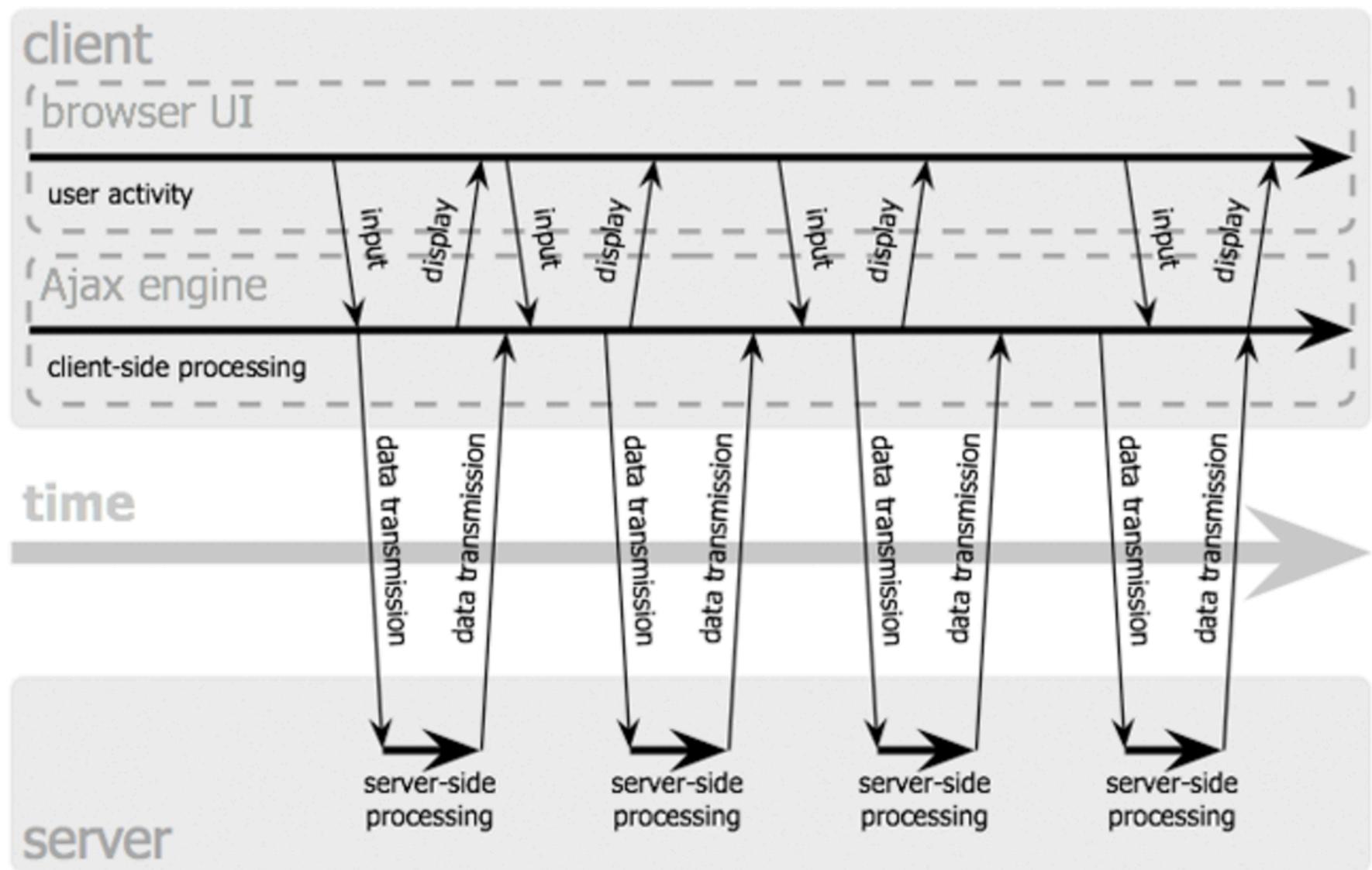
AJAX

- ▶ Asynchronous JavaScript and XML
 - ▶ Not an informative name
 - ▶ It can be synchronous
 - ▶ Not only XML (e.g. JSON)
- ▶ Ability for the browser to communicate directly with a server without a full reload of the page

TYPICAL CLIENT / SERVER COMMUNICATION



AJAX STYLE COMMUNICATION



DATA HIERARCHY

- ▶ bits
- ▶ Characters / Strings
- ▶ fields,
- ▶ Records
- ▶ files,
- ▶ Databases

OPERATING SYSTEMS

- ▶ Window's, OS X (Mac), Linux for server / desktop
- ▶ Apple iOS and Google Android for mobile
- ▶ Java may be open source, but it belongs to Oracle
 - ▶ Same for MySQL

TYPES OF PROGRAMMING LANGUAGES

- ▶ Machine language
- ▶ Assembler
- ▶ High level languages
 - ▶ Imperative vs declarative
 - ▶ Object oriented vs functional
 - ▶ AI / Machine Learning

OBJECT TECHNOLOGY

- ▶ Objects (an instance of a class) mix attributes and behaviors (methods)
- ▶ Functional languages are more concerned with data transformation and the passing of messages.
- ▶ Objects hide information (encapsulation), which is not necessarily a good thing
- ▶ Inheritance vs. composition are two design approaches in OO systems

REFERENCES

- ▶ <http://www.moorelaw.org/>
- ▶ https://en.wikipedia.org/wiki/Transistor_count
- ▶ <http://www.economist.com/blogs/economist-explains/2015/04/economist-explains-17>
- ▶ <http://arstechnica.com/information-technology/2016/02/moores-law-really-is-dead-this-time/>
- ▶ <https://vimeo.com/49718712>
- ▶ <https://blog.whatwg.org/html-is-the-new-html5>
- ▶ <http://w3c.github.io/html/>

MORE REFERENCES

- ▶ [http://www.virtualmv.com/wiki/index.php?
title=Internet%3AHistory_and_future](http://www.virtualmv.com/wiki/index.php?title=Internet%3AHistory_and_future)
- ▶ <https://danielmiessler.com/study/url-uri/>
- ▶ <https://hpbn.co/transport-layer-security-tls/>
- ▶ [http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-
web-20.html?page=1](http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1)
- ▶ <http://radar.oreilly.com/2007/10/todays-web-30-nonsense-blogsto.html>
- ▶ <https://callnerds.com/parts-of-a-link-url-explained/>
- ▶ <https://doepud.co.uk/blog/anatomy-of-a-url>