

CSI 3540 - LECTURE 02

INTERNET ET LE WORLD WIDE WEB

OBJECTIFS

- ▶ Chapitre 1 du manuel
- ▶ Évolution de l'Internet
- ▶ HTML, CSS, et JavaScript
- ▶ De nombreux exemples de choses amusantes (et ennuyeuses) que vous pouvez construire
- ▶ Code disponible à deitel.com/books/iw3htp5 et pearsonhighered.com/deitel

MOTIVATION POUR LE COURS

- ▶ La plate-forme la plus accessible au monde
- ▶ 3+ milliards de personnes sur l'Internet
- ▶ 20+ milliards appareil IoT
- ▶ Programmation "Client-Side"
 - ▶ Exécute sur l'appareil (ou navigateur) de l'utilisateur
- ▶ Programmation "Server-Side"
 - ▶ Répond à les requête des client
 - ▶ Achetez un jouet, réservez un trajet en train, vérifiez vos stocks, commandez des billets de concert

MOORE'S LAW

- ▶ En 1965, le cofondateur d'Intel, Gordon Moore, a fait observer que le nombre de composants dans les circuits intégrés doublait tous les 12 mois environ
- ▶ 1970 - 2000 augmentation de la vitesse du processeur
 - ▶ 1970 : 740 KHz à 8MHz
 - ▶ 2000 : 1.3 GHz à 2.8 GHz
 - ▶ 2016 : 2.1 GHz à 3.5 GHz

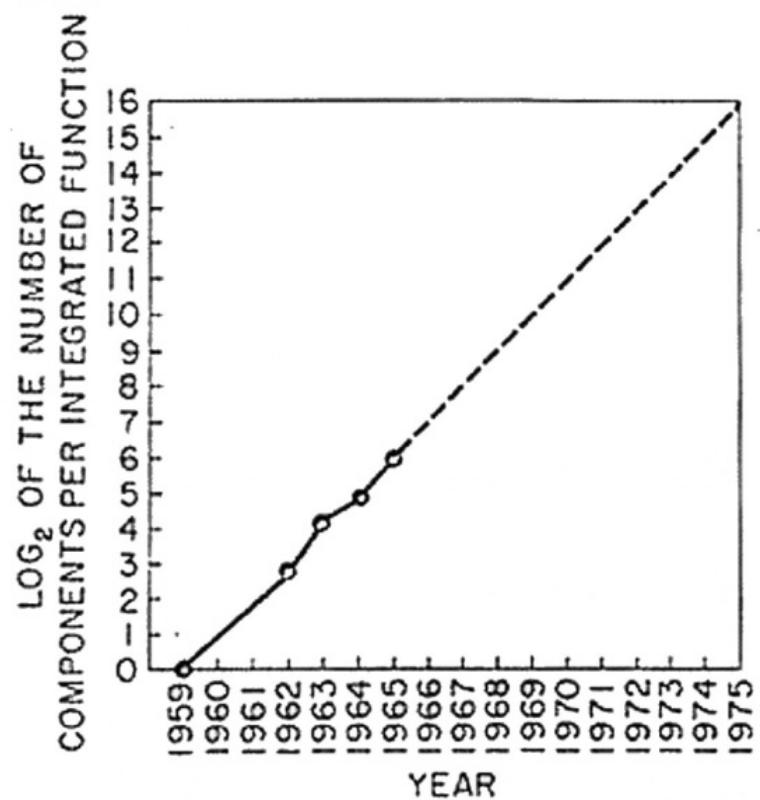


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

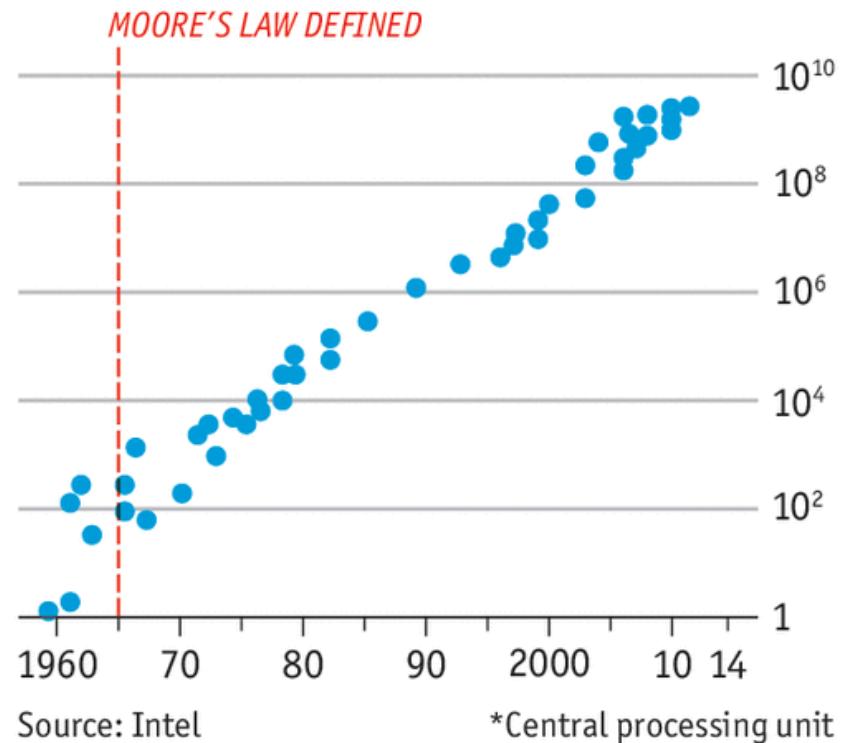
TRANSISTORS PAS LE VITESSE QUI DOUBLE

- ▶ En 2000, approx. 40 million transistors par CPU
 - ▶ en 2009, 900 million
 - ▶ en 2016, 7.2 milliards
- ▶ "Clock speed" est fini
 - ▶ En raison de la vitesse de la lumière et
 - ▶ de la chaleur générée
- ▶ Mais multi-core continue
 - ▶ Il faut de plus en plus de langues concurrentes / parallèles pour utiliser la "nouvelle" vitesse

A persevering prediction

Number of transistors in CPU*

Log scale



CONCURRENT VS. PARALLEL?

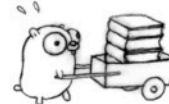
- ▶ Concurrent est le terme plus générique pour les exécutions de tâches qui chevauche ("overlap")
 - ▶ Pas nécessairement à la même moment
- ▶ Parallel est quand plusieurs tâches (concurrent) exécutent en même temps

(CONT) CONCURRENTE VS. PARALLEL?

- ▶ La concurrence est de traiter beaucoup de choses à la fois.
- ▶ Le parallélisme consiste à faire beaucoup de choses à la fois.
 - ▶ Pas la même, mais liée.
- ▶ Concurrence, c'est la structure,
 - ▶ le parallélisme, c'est l'exécution.
- ▶ Concurrence fournit un moyen de structurer une solution pour résoudre un problème qui peut (mais pas nécessairement) être parallélisable.

CONCURRENT, UN EXEMPLE

- ▶ Disons que nous avons dû déplacer des blocs
 - ▶ Oui concurrent, pas nécessairement parallel.
 - ▶ Peut-être qu'un seul gopher peut se déplacer à la fois

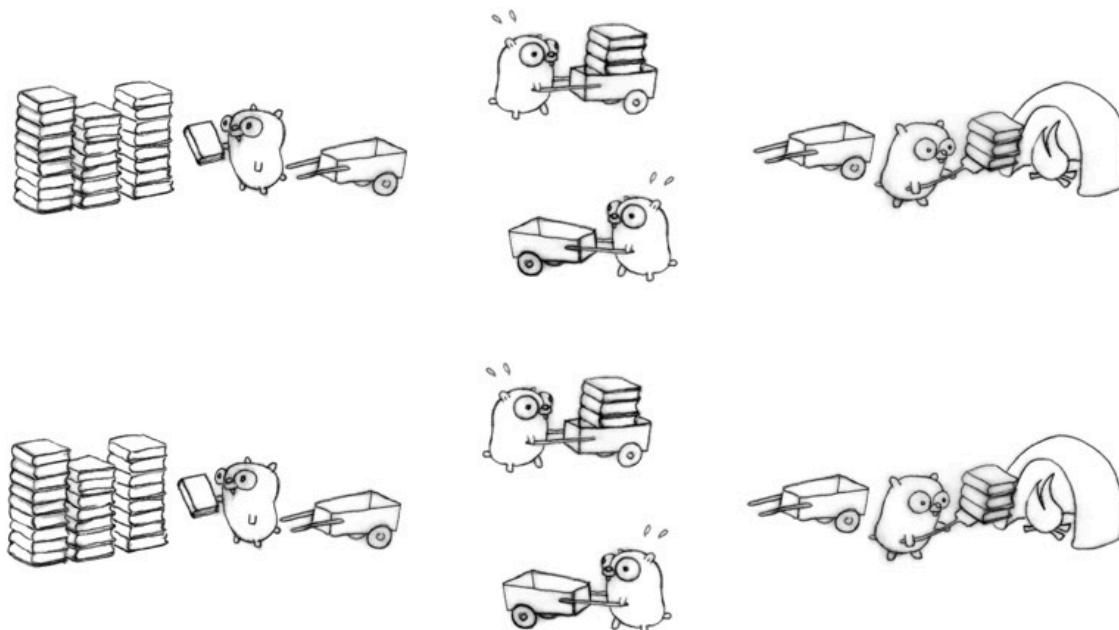


CONCURRENCE, D'ABORD

- ▶ Quatre procédures distinctes gopher:
 - ▶ Mettre des livres sur le panier
 - ▶ Déplacer le panier vers l'incinérateur
 - ▶ Débarquer le panier
 - ▶ Retour le panier vide
- ▶ Différents modèles concurrent permettent différentes façons de paralléliser.

MULTIPLE DIMENSIONS DE LA CONCURRENCE

- ▶ Encore une fois, si un seul gopher peut se déplacer à la fois, il est toujours concurrent (juste pas parallèle)



NOMBREUSES FAÇONS D'ÊTRE CONCURRENT

- ▶ Nous pouvons alors mélanger et assortir les différents autres modèles de concurrence
- ▶ Pensez d'abord à la façon de diviser le problème en composants indépendants puis sur la façon de les distribuer avec sûreté



MOORE'S LAW ET D'AUTRES CONCEPTS INFORMATIQUES

- ▶ S'applique à la mémoire
- ▶ Stockage secondaire (aka disque)
- ▶ Une augmentation similaire dans le domaine des communications
 - ▶ Les prix baissent à mesure que la demande de "bandwidth" augmente

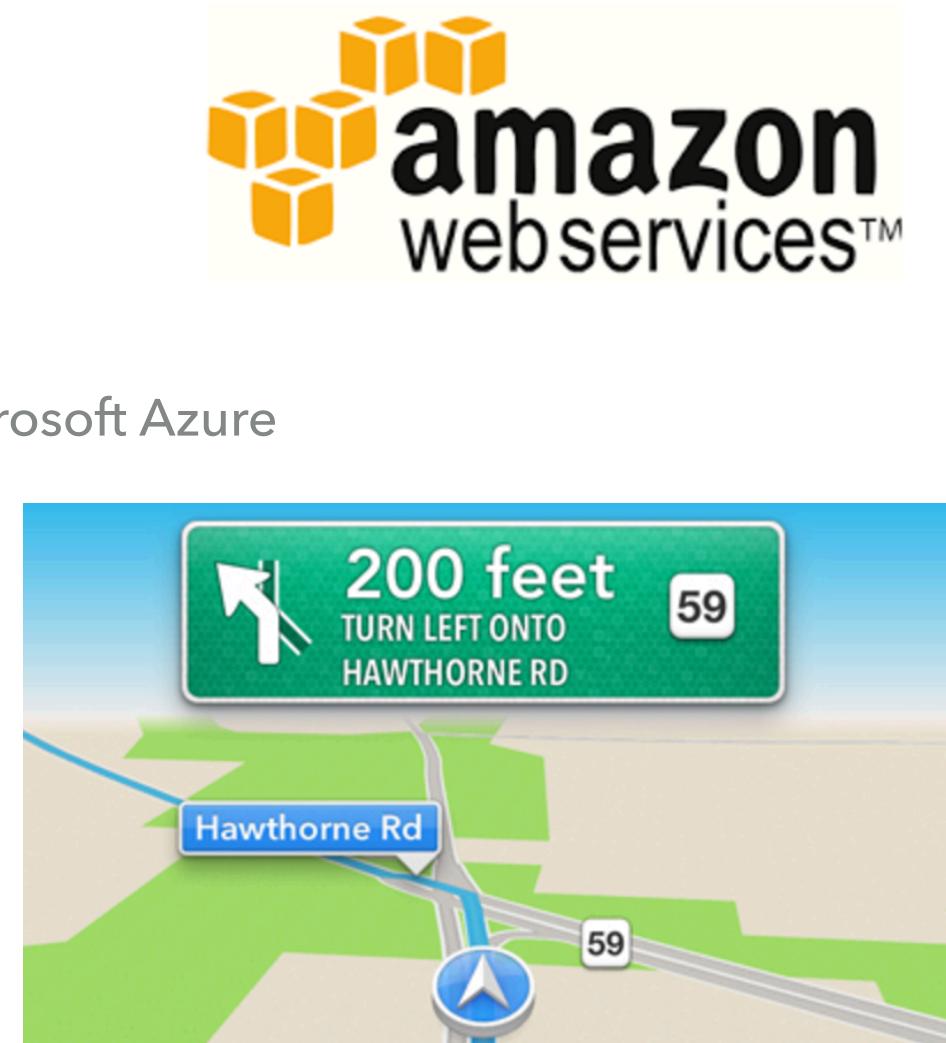
INDUSTRIE ET RECHERCHE (FIG 1.1 - 1.2)

- ▶ Soins de santé (projet du génome humain)
- ▶ AMBER(tm) Alert (pour localiser les enfants enlevés)
- ▶ OLPC (one.laptop.org)



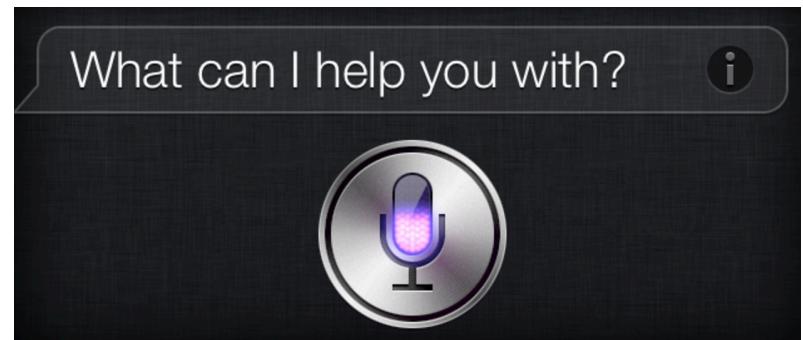
INFRASTRUCTURE POUR ... (FIG 1.3)

- ▶ Communiquer, naviguer, collaborer
- ▶ “Cloud Computing”
 - ▶ Amazon S3, EC2, Google Cloud, Microsoft Azure
 - ▶ Digital Ocean, Linode
 - ▶ GitHub, Bitbucket, CircleCI
- ▶ GPS
 - ▶ Instructions étape par étape
 - ▶ Modèles de trafic
 - ▶ “Find My Phone”



(CONT) INFRASTRUCTURE POUR ... (FIG 1.3)

- ▶ "Robots"
 - ▶ aspirateur Roomba
 - ▶ Exploration (mer profonde, l'espace)
 - ▶ Assistants personnels
 - ▶ Siri, Facebook M, Alexa
- ▶ Messagerie
 - ▶ E-mail
 - ▶ IM (Slack, Hangouts, Skype)
 - ▶ Téléphone (WhatsApp)
 - ▶ Images (SnapChat, Instagram, Pinterest)



DIVERTISSEMENT

- ▶ Magasins en ligne
 - ▶ iTunes
 - ▶ Google Play
- ▶ Télé sur l'Internet
 - ▶ Netflix
 - ▶ Hulu
- ▶ App Stores
 - ▶ Mario Run
 - ▶ Questrade



App Store



Google play



HTML

- ▶ HyperText Markup Language
 - ▶ *langage de "markup"*
 - ▶ *Contenu et structure d'information*
 - ▶ *Initialement conçus comme des documents "formatés"*
 - ▶ *Maintenant, un format de données extrêmement portable pour le partage d'informations (pas seulement "documents")*
- ▶ *Vous permet de créer des systèmes adaptés à une plusieurs appareil connectés à Internet*
 - ▶ *les smartphones,*
 - ▶ *les tablettes,*
 - ▶ *les ordinateurs portables / bureau,*
 - ▶ *les écrans géants (e.g. dans les stades sportifs)*

HISTORIES DES VERSIONS HTML

- ▶ HTML 1 - Tim Berners-Lee 1990/1991
- ▶ HTML 2 - le 24 novembre 1995
- ▶ HTML 3.2 - le 14 janvier 1997
- ▶ HTML 4.0 - le 18 décembre 1997
- ▶ HTML 5 - le 28 octobre 2014

HTML6?

- ▶ Non, 5.2 est la dernière version “nightly”
- ▶ Worldwide Web Consortium (W3C) continue à utiliser les numéros de version
- ▶ Web Hypertext Application Technology Working Group (WHATWG) avons arrêté
- ▶ “juste” HTML - des que maintenant

W3C

- ▶ Consacré au développement de technologies interprofessionnelles et non propriétaires pour le World Wide Web
- ▶ L'un des principaux objectifs du W3C est de rendre le site Web universellement accessible, sans tenir compte du handicap, de la langue ou de la culture.
- ▶ Un organisme de normalisation

WHATWG

- ▶ Le WHATWG a été fondé par des individus d'Apple, la Fondation Mozilla et Opera Software en 2004, après un atelier du W3C
- ▶ Ils étaient de plus en plus préoccupés par la direction du W3C avec XHTML, le manque d'intérêt pour le HTML et le mépris apparent pour les besoins des auteurs du monde réel
- ▶ Se concentre principalement sur les normes Web, en particulier:
 - ▶ HTML
 - ▶ Fetch
 - ▶ DOM (Document Object Model)
 - ▶ URLs

XHTML

- ▶ Extensible HTML
- ▶ Basé sur XML
- ▶ Plus stricte
- ▶ Mais, le tendance est 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>
```

CSS

- ▶ Cascading Style Sheets
- ▶ Sépare le style de la structure
- ▶ Fontes, bordures, couleurs, dimension, positionnement
- ▶ Veuillez utiliser une table pour les données tabulaires, et non pas les "divs"

LA PROMESSE DE CSS

- ▶ Facilement (un peu) changer l'aspect et la sensation sans changer le contenu
- ▶ Loin d'être parfait

Dividends.io

[Login](#)

Track your *payouts* like it's *payday*

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.

Provide your email to get started

Email Me An Access Code

By clicking this button, you agree to our [Terms of Service](#).

Fine print?
Nope. All it takes to start your account is your email address. We don't ask for your credit card until you are ready to upgrade to our premium services.

Are there any limits?
With our Free Plan you can track upto 10 stocks in one portfolio. If you want to to track multiple portfolios, or more stocks, then you

Do I have to worry about you selling my email or data?
No. We will not sell your contact information or private data to others. We will email you about product and company updates, but you can opt out at any time. Learn more about our [Privacy Policy](#).

Want more features?
We are still in beta, more features added each month. If you notice anything weird, or want to suggest a feature then [let us know](#).

Dividends.io

• [Login](#)

Track your payouts like it's payday

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.

Provide your email to get [Email Me An Access Code](#)

By clicking this button, you agree to our [Terms of Service](#).

Fine print?
Nope. All it takes to start your account is your email address. We don't ask for your credit card until you are ready to upgrade to our premium services.

Are there any limits?
With our Free Plan you can track upto 10 stocks in one portfolio. If you want to to track multiple portfolios, or more stocks, then you

Pricing?
We are still in beta, so try so much as you like free. Once out of beta, you can opt to stay in the Free Plan, or upgrade to Premium.

Do I have to worry about you selling my email or data?
No. We will not sell your contact information or private data to others. We will email you about product and company updates, bu

HISTORIES DES VERSIONS CSS

- ▶ CSS 1 - le 17 décembre 1997
- ▶ CSS 2 - mai 1998
- ▶ CSS 2.1 - le 25 février 2004 (officiel le 7 juin 2011)
- ▶ CSS 3 - divisé en 4 modules
 - ▶ Media - le 9 juin 2012
 - ▶ Namespaces - le 29 septembre 2011
 - ▶ Selectors - le 29 septembre 2011
 - ▶ Color - le 7 juin 2011

JAVASCRIPT

- ▶ Langage de script disponible dans le navigateur
 - ▶ Pas le seul
 - ▶ Dart par Google
 - ▶ VBScript et Internet Explorer
 - ▶ Mais oui c'est le seul
- ▶ Créé par Netscape
- ▶ Permet l'interactivité dans le navigateur
- ▶ Utilise les événements
 - ▶ Les actions du clavier et souris

STANDARDIZATION DE JAVASCRIPT

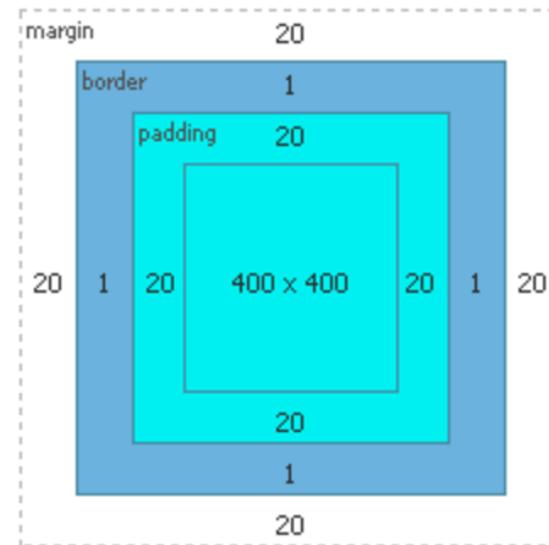
- ▶ NetScape et Microsoft
- ▶ Développé par ECMA International
 - ▶ Anciennement connue sous le nom European Computer Manufacturers Association
 - ▶ Maintenant appelé ECMAScript

HISTORIES DES VERSIONS JAVASCRIPT

- ▶ ECMAScript 1 - juin 1997
- ▶ ECMAScript 2 - juin 1998
- ▶ ECMAScript 3 - décembre 1999
- ▶ ECMAScript 5 - décembre 2009
 - ▶ 4ième édition a été abandonnée, en raison de différences concernant la complexité linguistique
- ▶ ECMAScript 2015 (version 6)
- ▶ ECMAScript 2016 (version 7)

NAVIGATEURS INTERNET

- ▶ Un "look and feel" cohérents est difficile
- ▶ Les navigateurs sont simplement des logiciels, pas de standard à respecter



TANT D'OPTIONS

- ▶ Les navigateurs partagent des fonctionnalités communes (et code), mais des différences apparaîtront
- ▶ Différentes versions du même navigateur se comporteront différemment
- ▶ Différents systèmes d'opérateur Windows, OS X, Linux
- ▶ Plusieurs approches:
 - ▶ Le plus récent,
 - ▶ Le plus préféré,
 - ▶ Le plus accessible,
 - ▶ Ou, le plus commun

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](#)

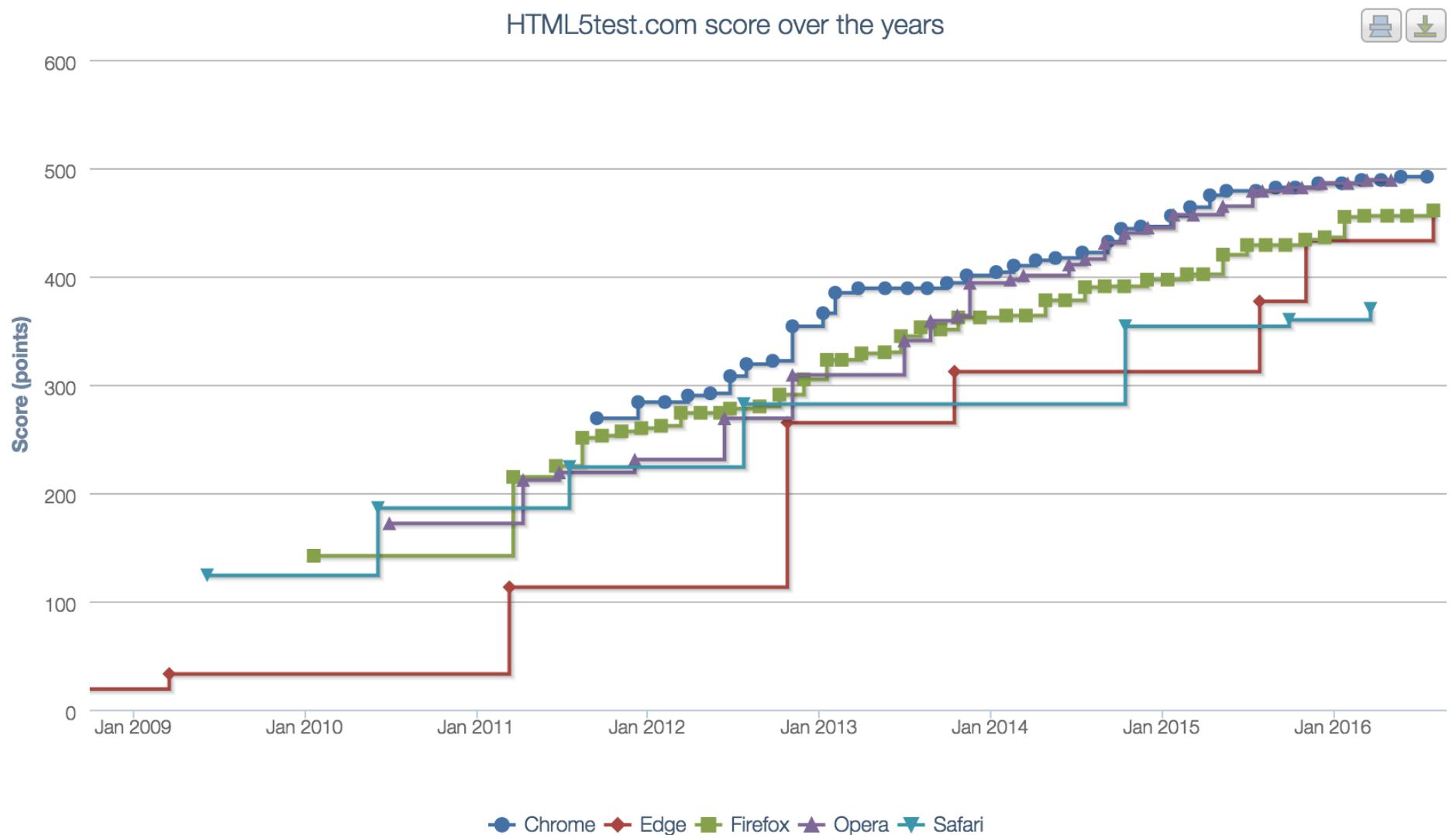
JavaScript Diagrams

Click here for GoJS

D'AUTRES NAVIGATEURS

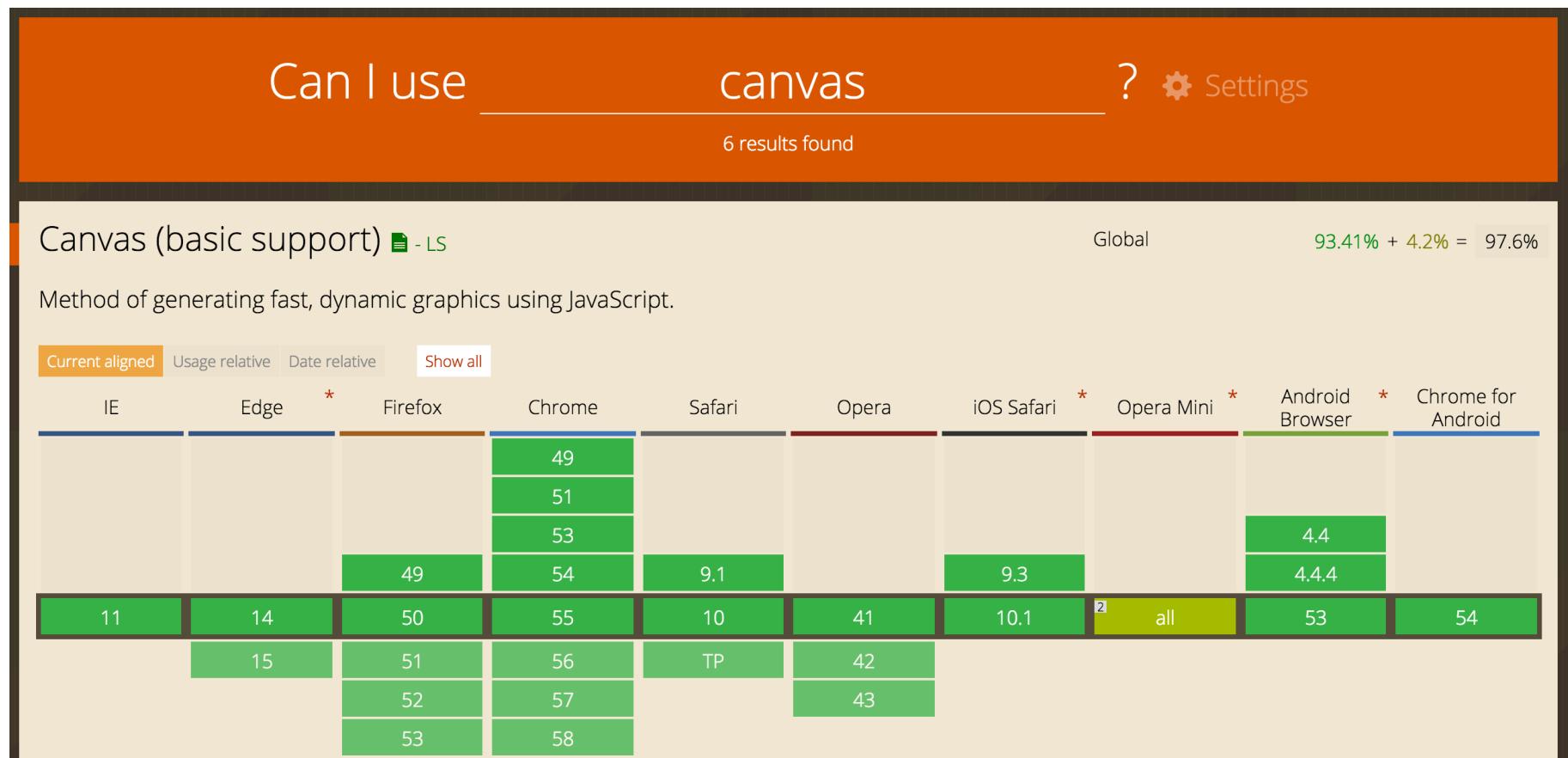
	Chrome	Opera	Firefox	Edge	Safari
Upcoming					
Current	52 ➔ 492		48 ➔ 461	14 ➔ 460	9.1 ➔ 370
Older	51 ➔ 492	37 ➔ 489	47 ➔ 456	13 ➔ 433	9.0 ➔ 360
	50 ➔ 489	30 ➔ 479	46 ➔ 456	12 ➔ 377	8.0 ➔ 354
	45 ➔ 482	12.10 ➔ 309	45 ➔ 456	Internet Explorer	
	40 ➔ 456		40 ➔ 429	11 ➔ 312	
	30 ➔ 394		35 ➔ 397	10 ➔ 265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

CHRONOLOGIE



PUIS-JE UTILISER?

► caniuse.com



PLATEFORME UNIVERSELLE “CLIENT-SIDE”

- ▶ <https://www.w3.org/2006/webapi/admin/charter>
- ▶ Improbable d'être entièrement universel
- ▶ l'innovation, par définition, est en dehors de la norme (aka les standards)

JQUERY

- ▶ librairie JS très populaire
- ▶ Simplifie le développement des JS de deux façons
 - ▶ Plus facile à manipuler des éléments de page
 - ▶ API portable entre les navigateurs
- ▶ Fournit des contrôles UI / UX supplémentaires
- ▶ jqueryui.com

CIRCLE COMPLET JS BRUT

- ▶ <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 = "";
```

VALIDATION DU CODE (FIG 1.6)

Technologie

HTML

CSS

JavaScript

Outil

validator.w3.org

html5.validator.nu

jigsaw.w3.org/css-validator

javascriptlint.com

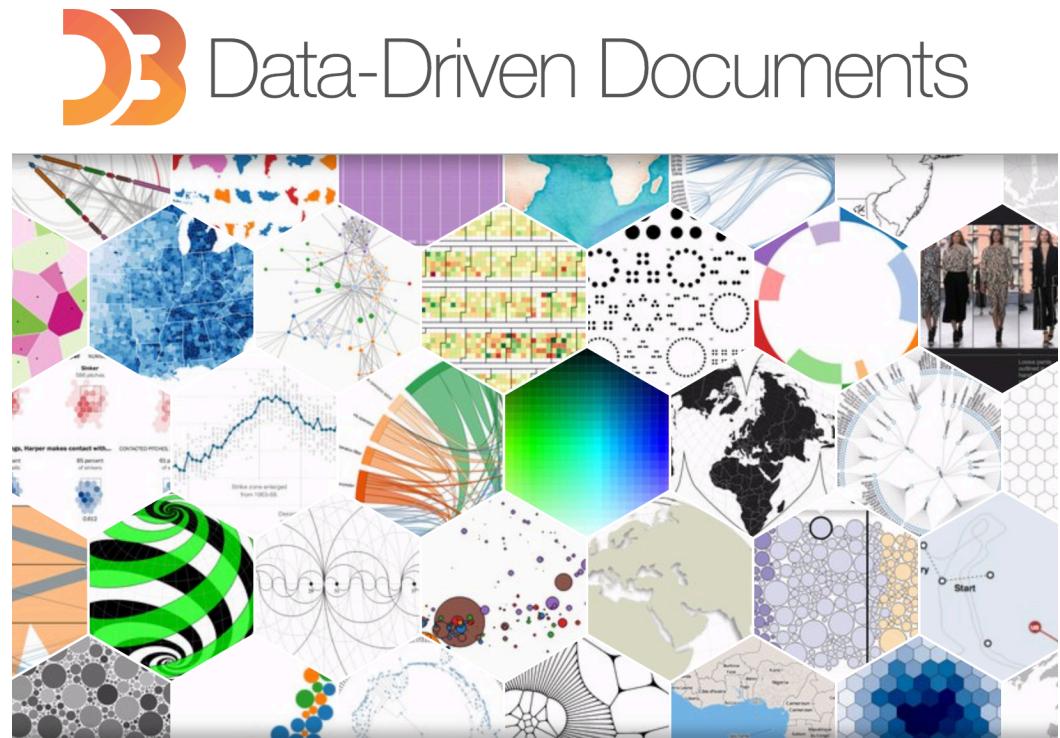
jslint.com

DEMOS (FIG 1.7)

- ▶ [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/>
 - ▶ Apporte des données à la vie en HTML, SVG et CSS



EVOLUTION DE L'INTERNET

- ▶ Convergence de l'informatique et des communications
- ▶ À la fin des années 1960, l'ARPA (Advanced Research Projects Agency), financé par le ministère américain de la Défense, a mis en place des plans pour mettre en réseau les principaux systèmes informatiques d'une douzaine d'universités
- ▶ La communication par courrier électronique, au lieu de partager les ordinateurs uns avec les autres, devient un des premiers avantages clés de l'ARPANET.

CHRONOLOGIE 1965 - 1983

- 1965** Deux ordinateurs communiquent en utilisant de la technologie "packet switching" à MIT Lincoln Lab.
- 1968** Beranek et Newman (BBN) remporte le contrat ARPANET (Advanced Research Projects Agency) pour les spécifications IMP (Interface Message Processor).
- 1969** Le 29 octobre, trois universités installent des nœuds et le premier message est "LO", qui a été une essay par l'étudiant Charles Kline à "LOGIN". Le message n'a pas pu être complété parce que le système SRI s'est écrasé.
- 1972** Ray Tomlinson de BBN présente le courrier électronique.
- 1973** Terme Internet est né
- 1974** Le premier fournisseur de services Internet (ISP) est né avec l'introduction d'une version commerciale d'ARPANET, connue sous le nom de Telenet.
- 1974** Vinton Cerf et Bob Kahn (les pères de l'Internet) publient «un protocole pour l'interconnexion de réseau de paquet», qui détaille la conception de TCP.
- 1979** USENET est créée avec les nouvelles et des groupes de discussion.
- 1982** TCP / IP émergent comme le protocole pour ARPANET et reste le protocole standard pour Internet.
- 1983** Le système de noms de domaine (DNS) établit les systèmes .edu, .gov, .com, .mil, .org, .net et .int pour nommer des sites Web.

CHRONOLOGIE 1986 - 2009

1986 Le NSFNET va en ligne aux avec des super-ordinateurs connectés à 56Kbps. La NSFNET était essentiellement un réseau de réseaux qui reliaient les utilisateurs universitaires avec l'ARPANET.

1987 Cisco vend son premier routeur. Il y a 20,000 hôtes d'Internet.

1990 Tim Berners-Lee, un scientifique au CERN, développe HTML

1991 CERN introduit le World Wide Web au public.

1994 Tim Berners-Lee crée World Wide Web Consortium (W3C). Netscape Communications est née.

1998 Google est né

2001 Un juge fédéral arrête Napster, jugeant qu'il doit trouver un moyen d'empêcher les utilisateurs de partager du matériel protégé en ligne.

2004 Facebook est né, et Firefox est créé (en remplacement Netscape)

2005 Lancement de YouTube.com

2009 L'Internet marque son 40e anniversaire.

GOOGLE

- ▶ En 1996, le Ph.D. de Stanford informatique. Les candidats Larry Page et Sergey Brin ont commencé à collaborer sur un nouveau moteur de recherche.
- ▶ En 1997, ils ont choisi le nom Google - un jeu sur le mot mathématique googol, qui est un nombre qui est égal à 1 suivi par 100 zéros (ou 10^{100}), un nombre énormément grand.
- ▶ La capacité de Google à retourner des résultats de recherche extrêmement précis rapidement l'a aidé à devenir le moteur de recherche le plus utilisé et l'un des sites Web les plus populaires dans le monde.

UN BIT

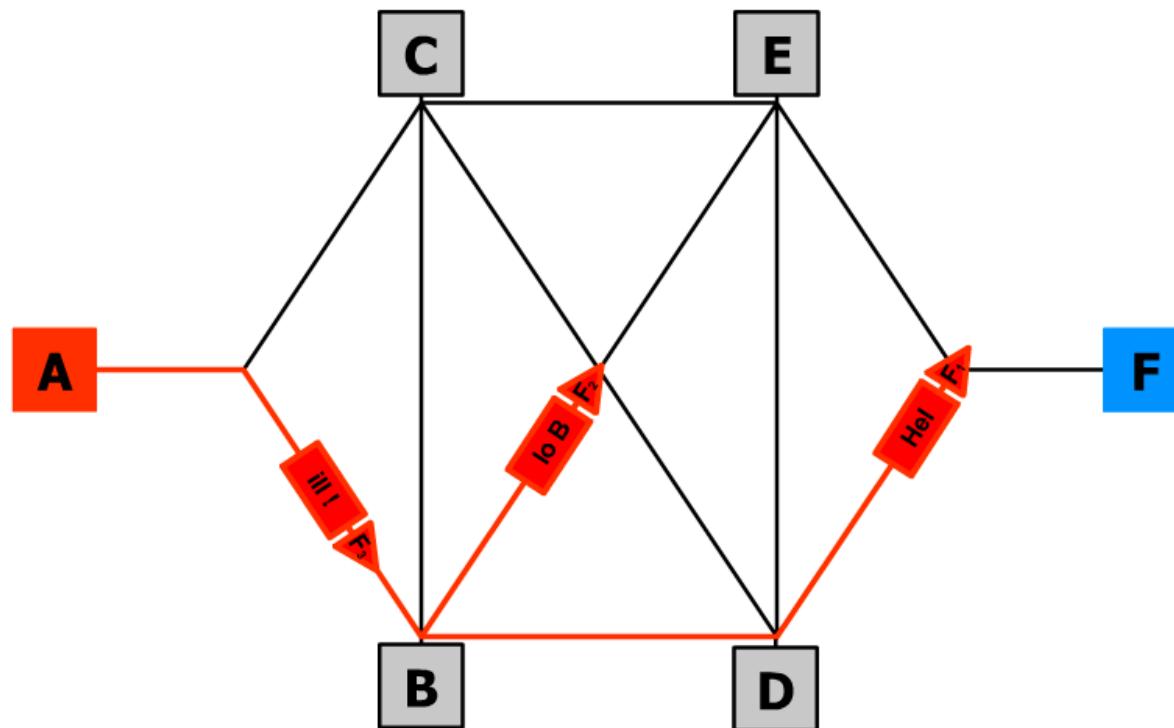
- ▶ Un bit (abréviation de «chiffre binaire») est la plus petite donnée d'un ordinateur
- ▶ Il peut prendre la valeur 0 ou 1.
- ▶ Un octet ("byte") est 8 bits
- ▶ Un Megabyte est 1,048,576 bytes or 1,024 Kilobytes
- ▶ Attention aux erreurs d'arrondi comme 1000 bytes != 1 kilobyte

COMMUNICATION DE PAQUETS (“PACKET SWITCHING”)

- ▶ L'un des principaux objectifs de ARPANET était de permettre à plusieurs utilisateurs d'envoyer et de recevoir des informations simultanément sur les mêmes chemins de communication (par exemple, lignes téléphoniques).
- ▶ Le réseau travaillé par envoyant des données en petits morceau appelés paquets, contenant
 - ▶ Adresse,
 - ▶ Contrôle d'erreur
 - ▶ Information de séquençage

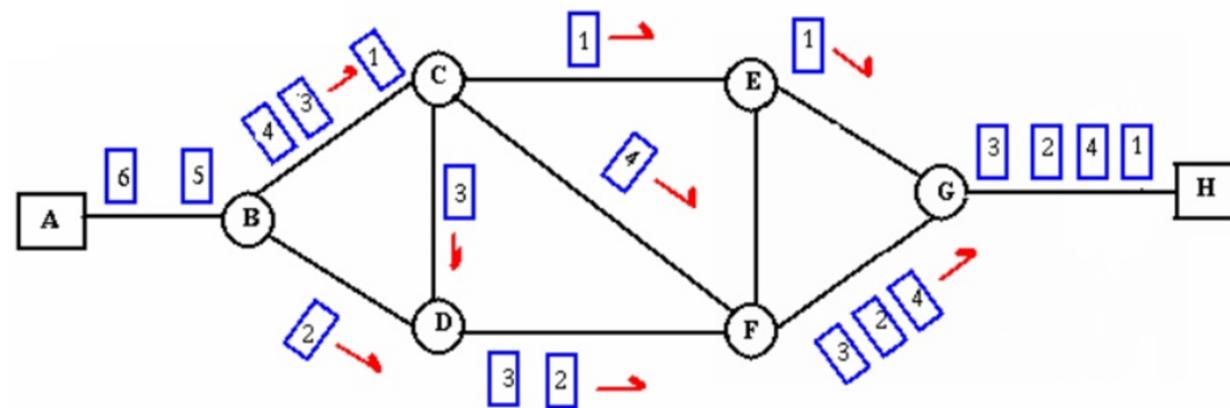
L'ADRESS DES PAQUETS

- ▶ Les informations d'adresse autorisaient le routage des paquets vers leurs destinations.



INFORMATION DE SÉQUENÇAGE

- ▶ L'information de séquencement aide à rassembler les paquets dans leur ordre d'origine pour la présentation au destinataire
- ▶ En raison de mécanismes de routage complexes, pourraient réellement arriver hors de l'ordre



DÉCENTRALISÉ PAR CONCEPTION

- ▶ Les paquets de différents messages ont été mélangés sur les mêmes lignes pour utiliser efficacement la bande passante disponible.
- ▶ Le réseau a été conçu pour fonctionner sans contrôle centralisé.
- ▶ Si une partie du réseau échouait, les portions qui fonctionnait peuvent continuer à passer les paquets des messages vers les récepteurs sur des chemins alternatifs.

TCP – TRANSPORT CONTROL PROTOCOL

- ▶ Le protocole de communication sur l'ARPANET
- ▶ TCP assure que les messages sont reçus intacts
 - ▶ Ordre correct
 - ▶ Réception complète
- ▶ Un des défis consistait à faire communiquer entre les réseaux "intra" et "inter".

IP - INTERNET PROTOCOL

- ▶ Développé par ARPA, l'IP a créé la base protocol pour un "réseau de réseaux"
- ▶ Chaque périphérique connecté possède une adresse IP unique

Nom	Exemple	Détails
IPv4	192.168.0.1	Établi en 1981 (1984), basé sur 32 bits (donc 4 milliards d'adresses uniques)
IPv6	2001:0db8:85a3:0000:0000:08a2e:0370:7334	Introduit en 1998 (dernière changement en 2006) et basé sur 128 bits (3.4×10^{38} adresses uniques ou 6.7×10^{23} par mètre carré sur terre)

POURQUOI NE PAS CHANGER À IPv6?

- ▶ Différents réseaux nécessitant des mises à niveau de logiciels et de hardware
- ▶ Pas de raisons tangibles de améliorer, seuls les coûts
 - ▶ Toutes les fonctionnalités utiles d'IPv6 ont été ajoutées à IPv4
- ▶ Les réseaux internes ne sont généralement pas affectés
- ▶ Les IP finales remises en 2011
 - ▶ La rareté enfin devient une réalité

INTRANET

- ▶ Un réseau d'ordinateurs au sein d'une organisation
- ▶ LAN (Local Area Network)
- ▶ RFC 1918
 - ▶ Class A 10.0.0.0/8
(10.0.0.1 à 10.255.255.254)
 - ▶ Class B 172.16.0.0/12
(172.16.0.1 à 172.31.255.254)
 - ▶ Class C 192.168.0.0/16
(192.168.0.1 à 192.168.255.254)

CLASSES D'ADRESSES IP

Classe	Address IP	IP du réseau	IP de l'hôte
--------	------------	--------------	--------------

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
---	---------	-------	---

QU'EST-CE QU'UN PROTOCOLE INFORMATIQUE?

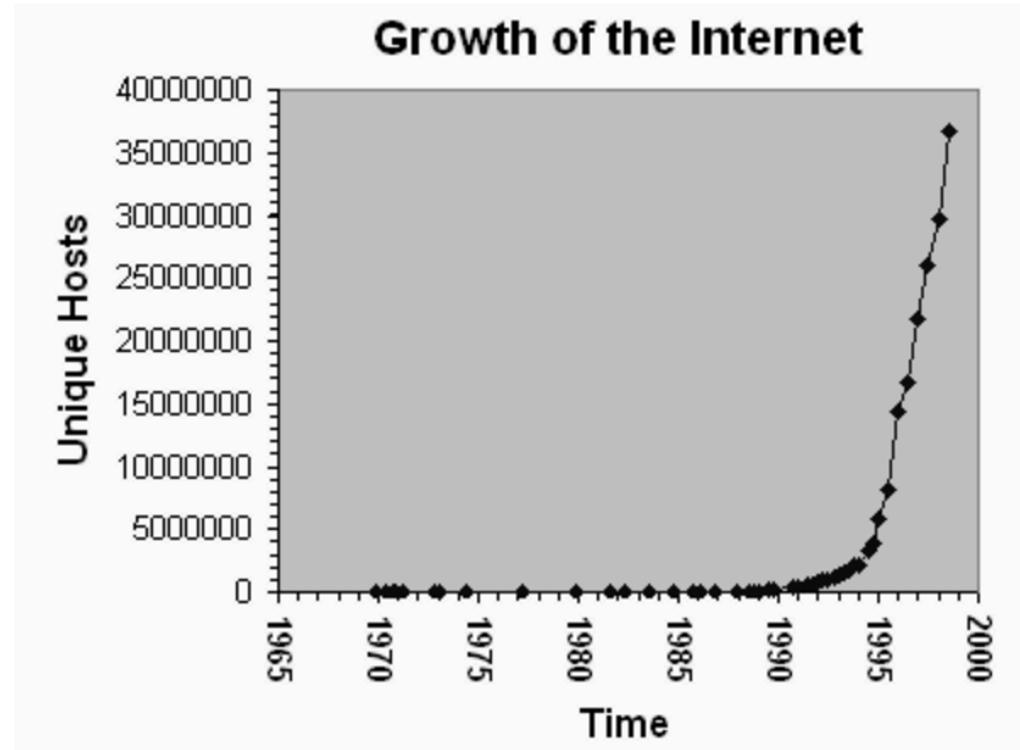
- ▶ Un ensemble de règles dans lesquelles les ordinateurs communiquent entre eux
- ▶ Le protocole indique quelle partie de la conversation arrive à ce moment
- ▶ Il indique également comment mettre fin à la communication

EXEMPLES DE PROTOCOLES

Abbr	Nom	Détails
TCP	Transmission Control Protocol	Un protocole principal dans la suite Internet
UDP	User Datagram Protocol	Utilisé pour établir des connexions à faible latence et à tolérance de perte entre des applications
IP	Internet Protocol	Les données sont transmises d'un ordinateur à un autre sur Internet
FTP	File Transfer Protocol	Utilisé pour montrer les fichiers à copier entre les périphériques
HTTP	Hypertext Transfer Protocol	Une méthode et un format standard pour le transfert de données sur Internet
HTTPS	HTTP over SSL ou TLS	Données cryptées envoyées via HTTP
SMTP	Simple Mail Transfer Protocol	L'envoi et la réception de courriels

DÉVELOPPEMENT EXPLOSIVE

- ▶ Initialement, limitée aux universités et aux instituts de recherche
 - ▶ Puis les militaires ont commencé à l'utiliser intensivement
 - ▶ Finalement, le gouvernement a décidé de permettre l'accès à Internet à des fins commerciales.



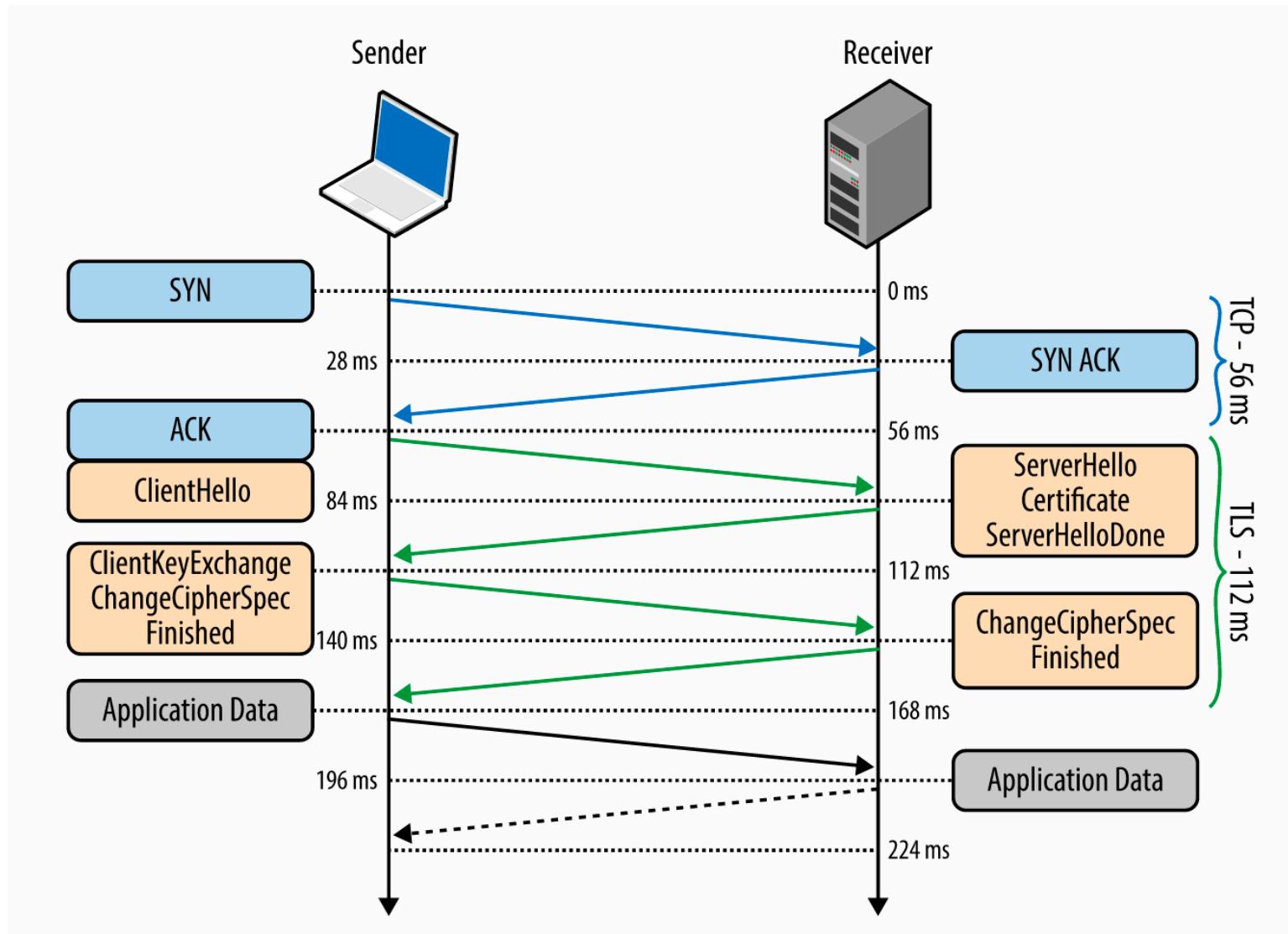
TIM BERNERS-LEE

- ▶ HTML inventé en 1989
- ▶ Écrit le protocole HTTP pour la WWW
- ▶ Défini l'URL (Uniform Resource Locator)
 - ▶ Localiser un document n'importe où sur Internet
 - ▶ <https://www.uottawa.ca/graduate-studies/programs-admission/programs/computer-science>

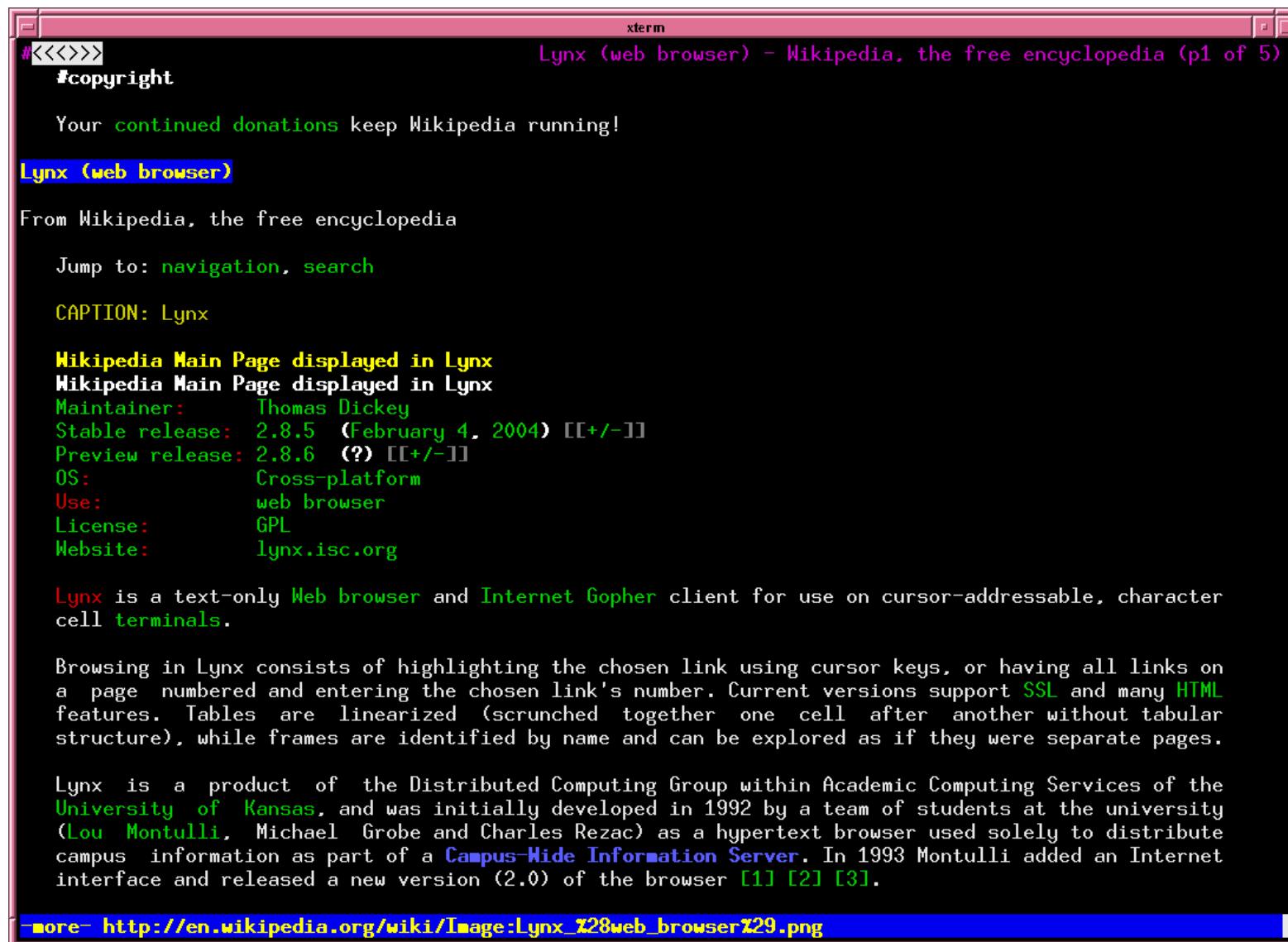
HTTPS

- ▶ HTTP +
 - ▶ SSL (Secure Socket Layer) ou
 - ▶ TLS (Transport Layer Security)
- ▶ Doit être utilisé pour tout le trafic, pas seulement pour le trafic privé
- ▶ En tant que privé?

TLS HANDSHAKE

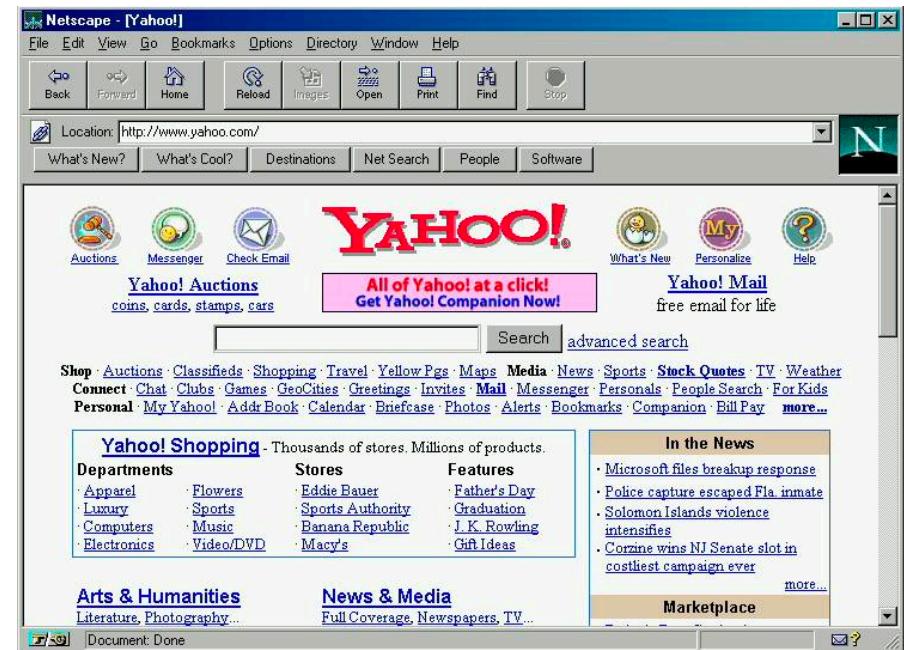


WEB 1.0



WEB 2.0

- ▶ L'utilisation du Web a explosé avec la disponibilité en 1993 du navigateur Mosaic, qui présentait une interface graphique conviviale.
- ▶ Développé par Marc Andreessen chez NCSA qui a ensuite fondé Netscape
- ▶ Dotcom buste en 2000 mis un déclin temporaire dans l'utilisation commerciale de l'Internet
- ▶ La résurgence 2003 de l'Internet a été * le * Web 2.0, comme nommé par Dale Dougherty (et Tim O'Reilly)
- ▶ Des sites Web collaboratifs et communautaires (média sociaux, wikis, blogs)



WEB 3.0

- ▶ Nova Spivack le définit comme le «Web sémantique», en reliant intelligemment les données, les concepts, les applications et les gens
- ▶ La première règle du Web 3.0 est de cesser de l'appeler que
- ▶ Pas besoin d'un numéro de version, plutôt parler de l'intention
 - ▶ Mobile First?
 - ▶ IoT (Internet of Things)?
 - ▶ Web sémantique

COMPARAISON DE WEB X.0

- ▶ Web 1.0 est comme une conférence, un petit nombre de professeurs d'informer un large public d'étudiants.
- ▶ Web 2.0 est une conversation, chacun ayant la possibilité de parler et de partager des vues.
- ▶ Web 3.0 est comme des “coles notes” pour les conversations comme il ya maintenant trop de gens parlant (IoT, AI, semantic web)

ARCHITECTURE PILOTÉE PAR DONNÉES

- ▶ En utilisant l'intelligence collective, le concept qu'un grand groupe diversifié de personnes créera des idées intelligentes
- ▶ Utilisation de l'apprentissage machine et de Big Data pour la récupération de nouvelles, les traducteurs de langues et les assistants personnels

NOTION DE BASE SUR LE WEB

```
index.htm
```

The diagram illustrates the structure of a Document HTML. A green line connects the title bar of the code editor to the text "Document HTML". Another green line points from the word "Hyperlinks" to the [tag in the code. A third green line points from the "Courriel" column header to the email links \(\[aforward@uottawa.ca\]\(mailto:aforward@uottawa.ca\) and \[aforward@gmail.com\]\(mailto:aforward@gmail.com\)\). A fourth green line points from the "Site web" column header to the link \(<https://uottawa.blackboard.com>\) in the code.](#)

```
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <link rel="stylesheet" href="scripts/base.css" id="theme">
    <title>CSI 3540 - Structures, techniques et normes du web</title>
</head>
<body class="syllabus">
    <h1>CSI 3540</h1>
    <h2>Structures, techniques et normes du web</h2>
    <hr/>
    <h3>Description</h3>
    <p>
        Infrastructure de base du Web. Serveurs et navigateurs. Exemples de protocoles.
        Internet et virus. Architecture de moteur de recherche. Contenu et présentation Web.
        Pages Web, leur structure et leur interprétation. HTML, XML et leurs dérivés.
        Interfaces Web vers les logiciels et bases de données. Témoins et droit à la vie privée
        Web sémantique et ontologies. Services Web.
    </p>
    <h3>Préalables :</h3>
    </h3>
    <p>
        CSI2510,
        CSI2532
    </p>
    <h3>Professeur</h3>
    <table>
        <tr>
            <th>Enseignante</th>
            <td>Andrew Forward</td>
        </tr>
        <tr>
            <th>Courriel</th>
            <td>
                <a href="mailto:aforward@uottawa.ca">aforward@uottawa.ca</a><br />
                <a href="mailto:aforward@gmail.com">aforward@gmail.com</a>
            </td>
        </tr>
        <tr>
            <th>Site web</th>
            <td>
                <a href="https://uottawa.blackboard.com" target="_blank" rel="noopener">Collapse 2</a>
            </td>
        </tr>
    </table>
</body>

```

Document HTML

Hyperlinks

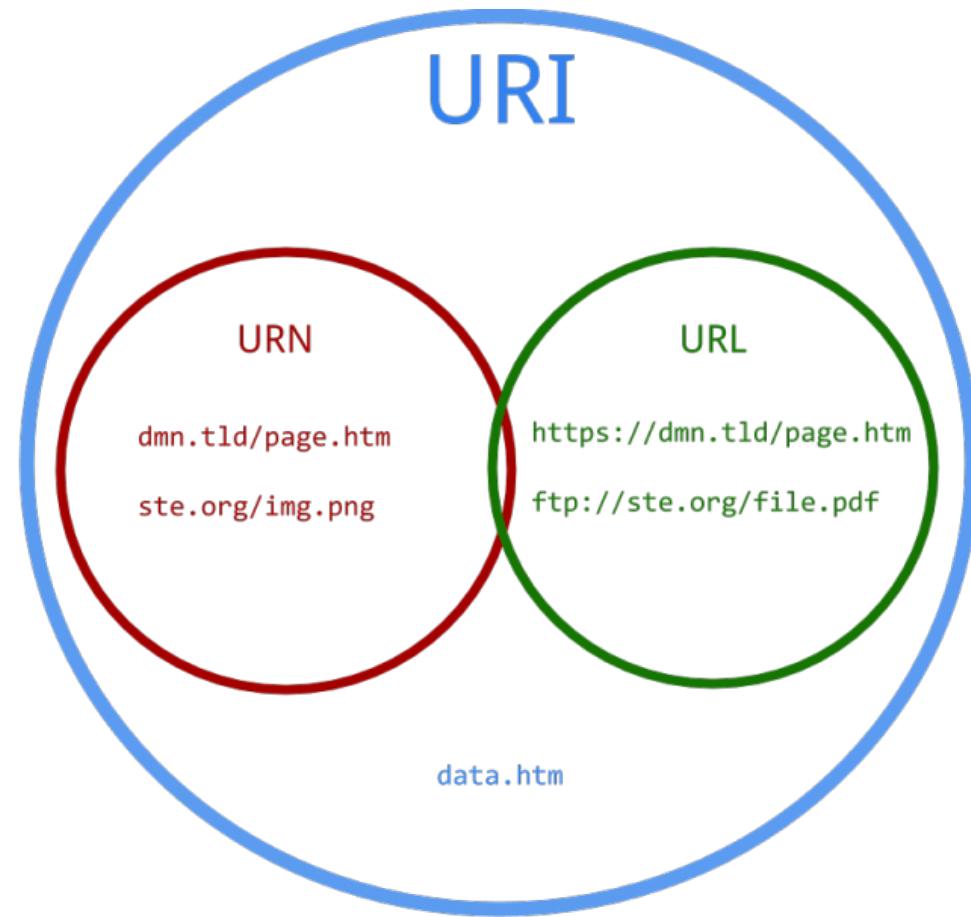


HYPERLINKS

- ▶ <mailto:aforward@gmail.com>
- ▶ <http://a4word.com>
- ▶ //scripts/base.css
 - ▶ Prefix “//” pour conserver le même protocole (par exemple http contre https)

URI = URN + URL + ...

- ▶ Uniform Resource Identifier
- ▶ Uniform Resource Name
 - ▶ Identité de l'objet
- ▶ Uniform Resource Locator
 - ▶ Nom + Méthode d'accès
- ▶ Tous les papillons volent, mais pas tout ce qui vole est un papillon

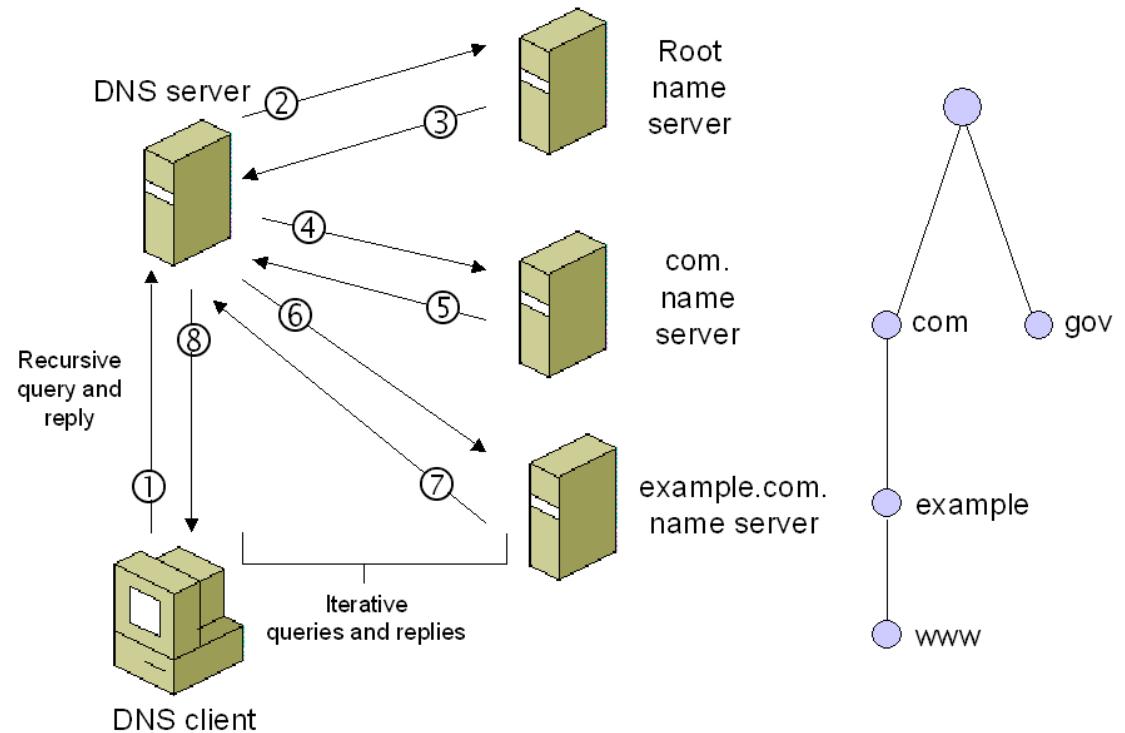


MIME

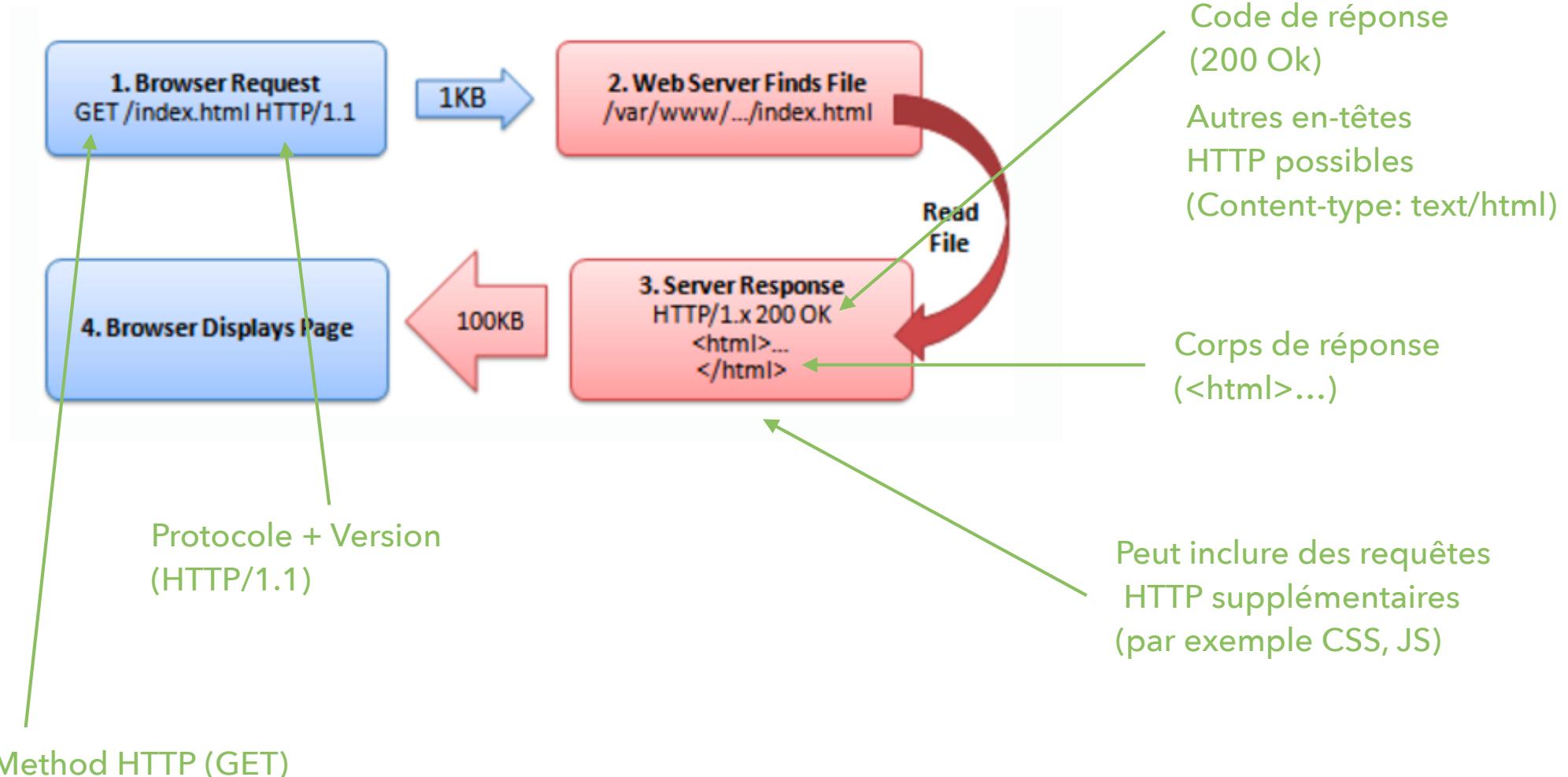
- ▶ Multipurpose Internet Mail Extensions
- ▶ Le type de corps que le serveur transmet au navigateur
- ▶ Spécifie les formats de données, quels programmes peuvent être utilisés pour interpréter correctement les données
 - ▶ text/plain
 - ▶ image/jpeg

DNS (DOMAIN NAME SERVER)

- ▶ Une base de données de noms d'hôtes et leurs adresses IP correspondantes et effectue les traductions automatiquement.



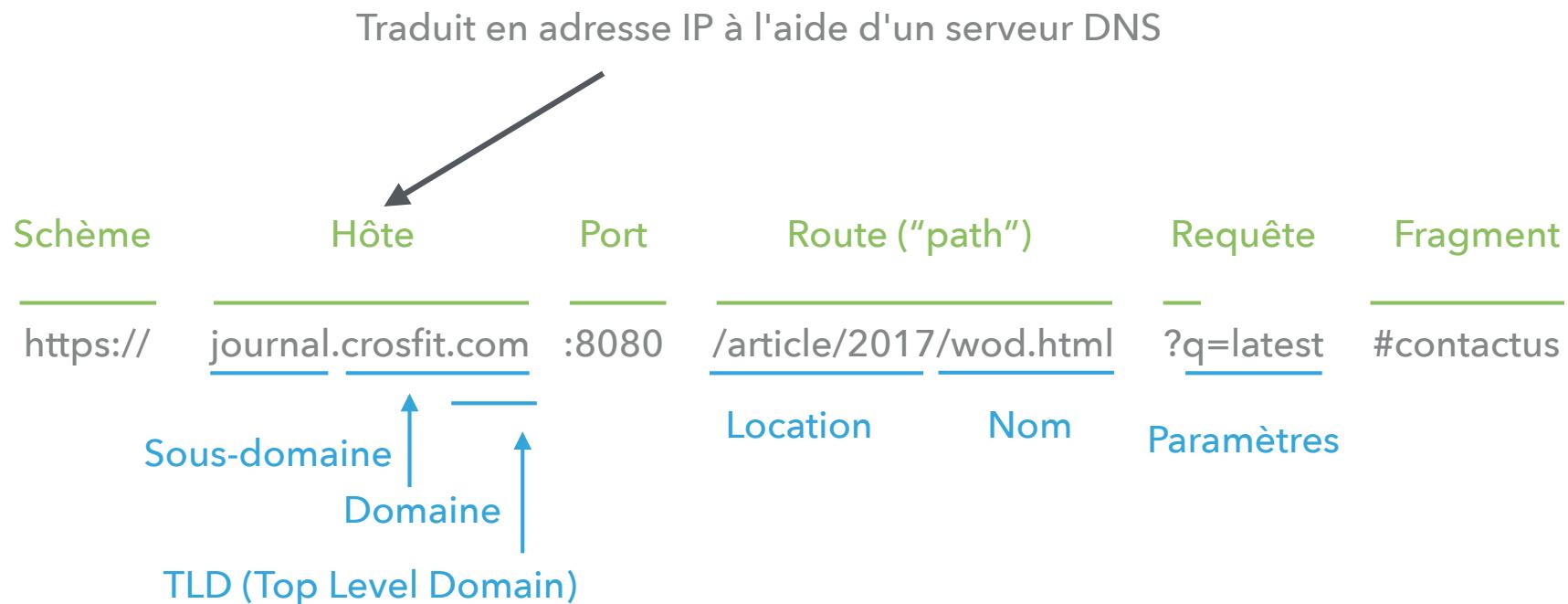
DEMANDER UN DOCUMENT



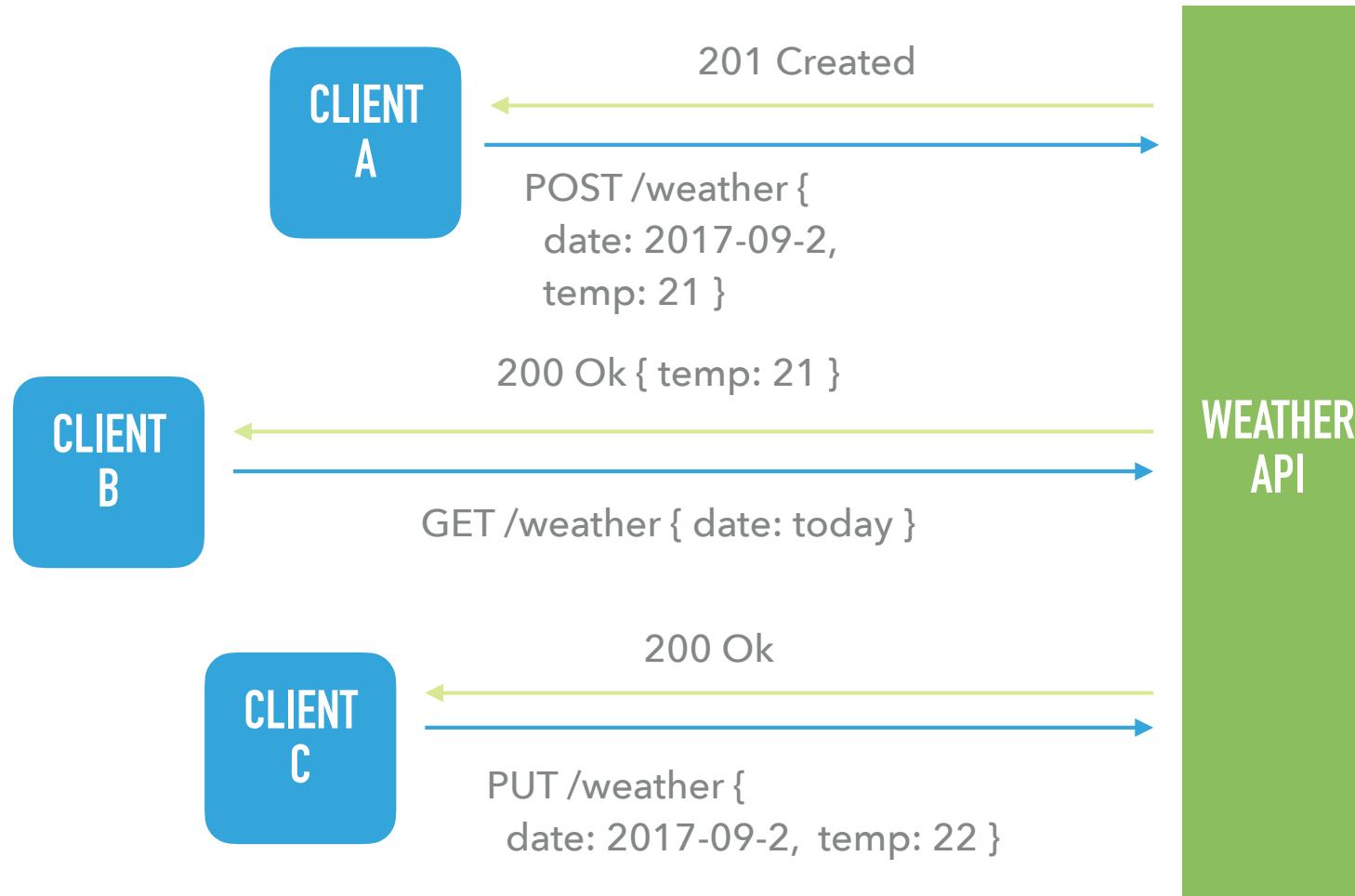
TYPES MIME COMMUNS

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



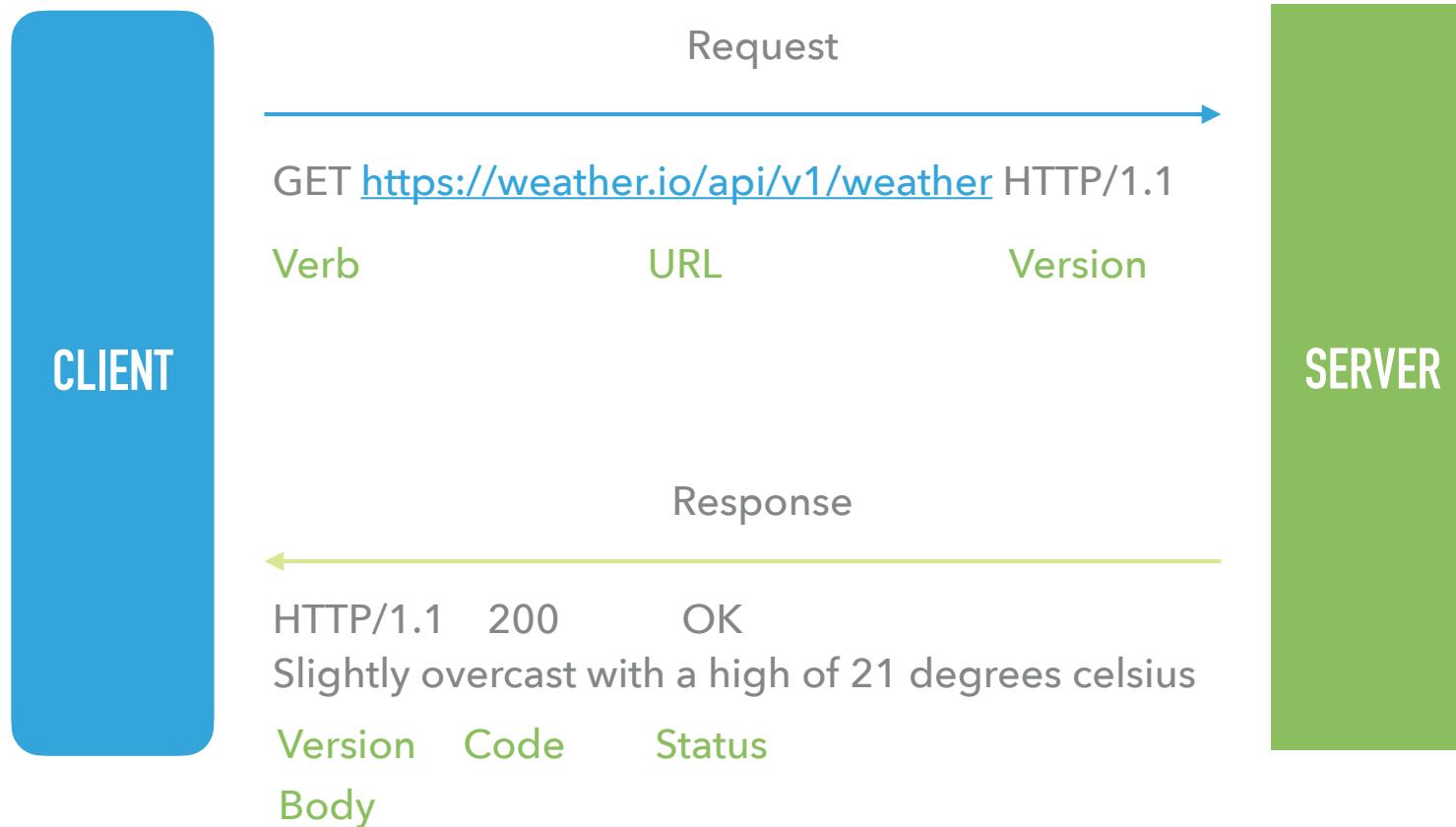
DEMANDE D'API HTTP



API REST

- ▶ Le Representational State Transfer (REST) a été défini par Roy Thomas Fielding dans sa thèse de doctorat de 2000 intitulée «Architectural Styles and the Design of Network-based Software Architectures» à UC Irvine.
- ▶ Toutes les API REST ne sont pas identiques, ce qui conduit à un modèle de maturité pour aider à décrire le niveau de RESTfulness
- ▶ JSON la norme non officielle pour le corps de demande / réponse (aka vos données)

NIVEAU 0: TRANSPORT HTTP



NIVEAU 1: RESSOURCES

- ▶ Nous divisons les “end-points” en fonction des ressources.

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

NIVEAU 2: VERBES ET CODES D'ÉTAT

- ▶ Utilisation des verbes HTTP (aka méthodes) et des codes d'état de manière cohérente.

Verbe	État	Détails
GET	200 Ok	Obtenir des données seulement, ne doit rien changer
POST	201 Created	Créer un nouvel enregistrement pour la ressource fournie
PUT	200 Ok	Remplacer une ressource existante (aka update)
PATCH	200 Ok	Modifier une ressource existante (aka update)
DELETE	204 No Content	Supprimer une ressource

NIVEAU 2: PLUS DE CODES D'ÉTAT

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

NIVEAU 3: HYPERMEDIA

- ▶ HATEOAS (Hypertext As The Engine Of Application State)
- ▶ Utilisation de liens dans la réponse

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

Données de lien supplémentaires
indiquant au client où elles peuvent
aller après

Utilisation des données href au lieu de
hardcoding le resource directly dans
l'interface utilisateur client

METHOD GET

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

“query string”

Paire nom/valeur

Séparé par &

METHOD POST

POST ▼ www.example.com/users

Authorization Headers (1) **Body** ● Pre-request Script Tests

form-data x-www-form-urlencoded raw binary **JSON (application/json)** ▾

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

Les données dans le corps de la requête

Le format des données dépend de l'API, généralement JSON

The screenshot shows a Postman interface for a POST request to 'www.example.com/users'. The 'Body' tab is selected, showing a JSON payload:

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

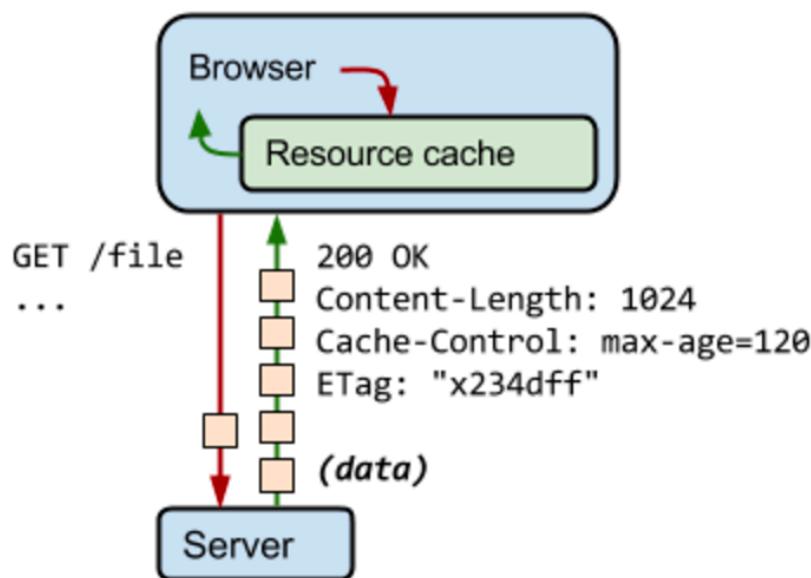
Annotations explain the JSON structure: one arrow points to the JSON object, and another points to the 'email' key.

MASHUPS

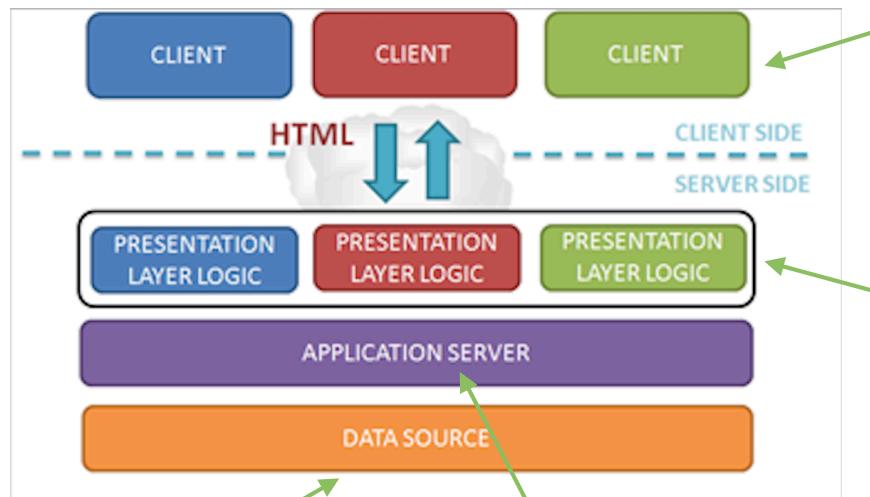
Service Web	URL
Google Maps	https://developers.google.com/maps/documentation/javascript/tutorial
GitHub	http://developer.github.com/
Foursquare	https://developer.foursquare.com/
Gravatar	http://en.gravatar.com/site/implement/
Instagram	http://instagram.com/developer/
Meetup	http://www.meetup.com/meetup_api/
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

CACHE CÔTÉ-CLIENT

- ▶ Les navigateurs cachent les documents (page web) récemment consultés pour recharger rapidement



APPLICATION N-TIER



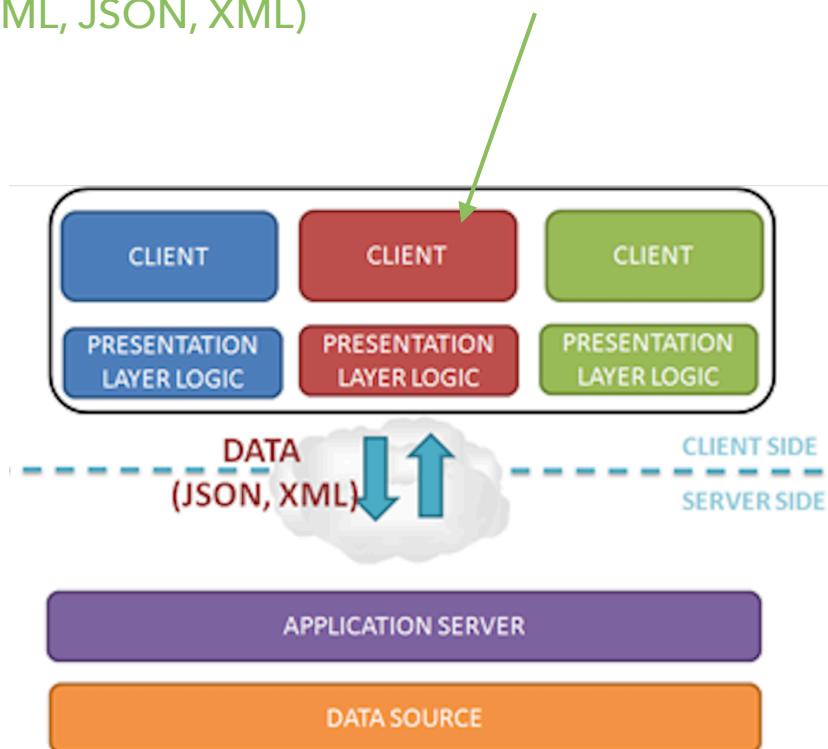
Base de données
(SQL, NoSQL, NewSQL)

Logique d'entreprise
et de contrôleur

Interface utilisateur
/ navigateur

Logique de présentation
(HTML, JSON, XML)

Plus récemment, plus
de framework poussent
le rendu d'interface
utilisateur vers le client



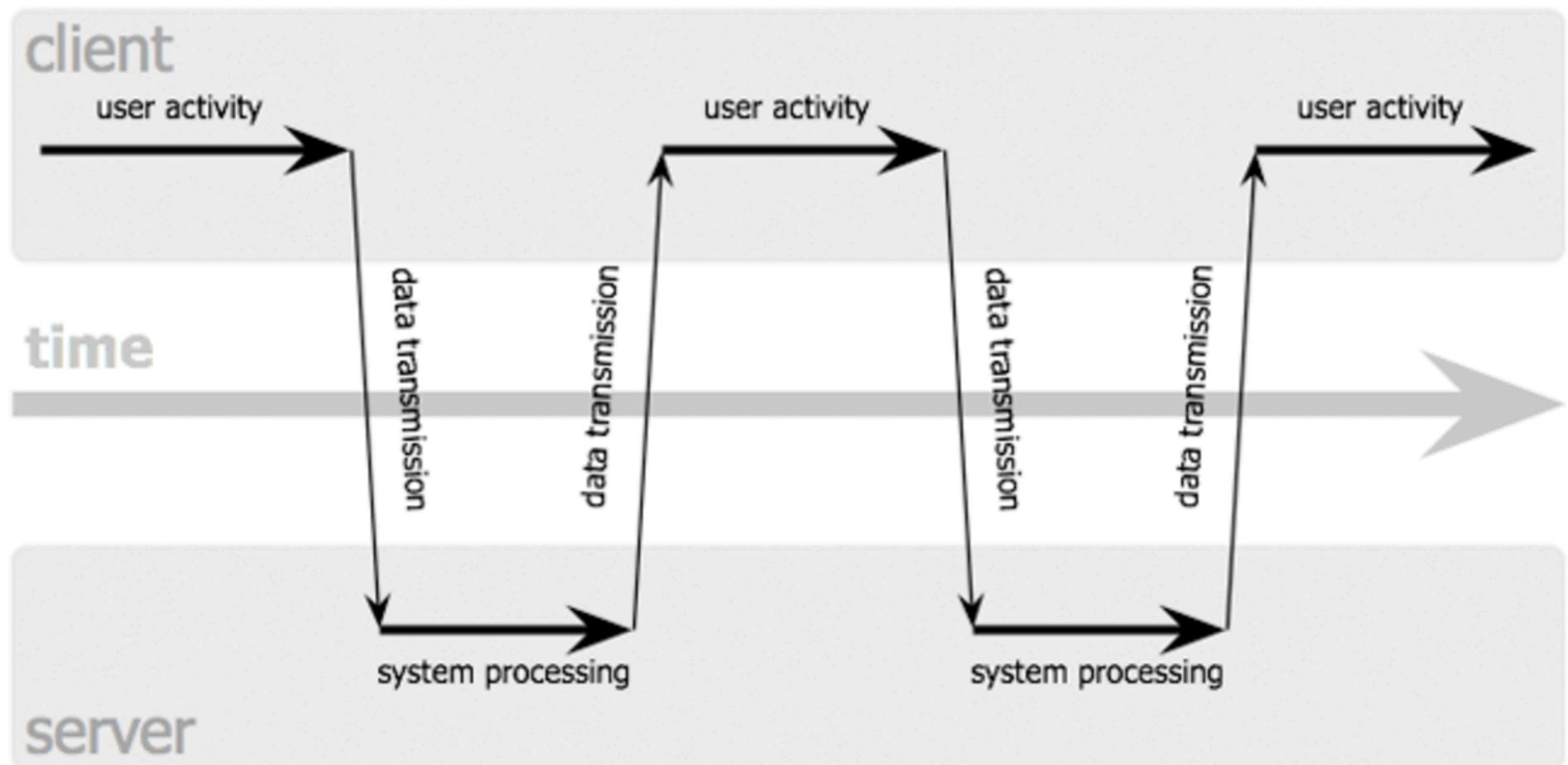
SCRIPTING CÔTÉ CLIENT

- ▶ JavaScript peut être utilisé pour valider les entrées, dessiner l'interface, et envoyer les demandes supplémentaire (AJAX / Web sockets)
- ▶ Mais la dépendance du navigateur et la disponibilité de la fonctionnalité requise sont des problèmes
- ▶ Code est également ouvertement visibles par le client, et peut être écrasé par le client
- ▶ Les validations doivent donc être traitées de la même manière sur le serveur

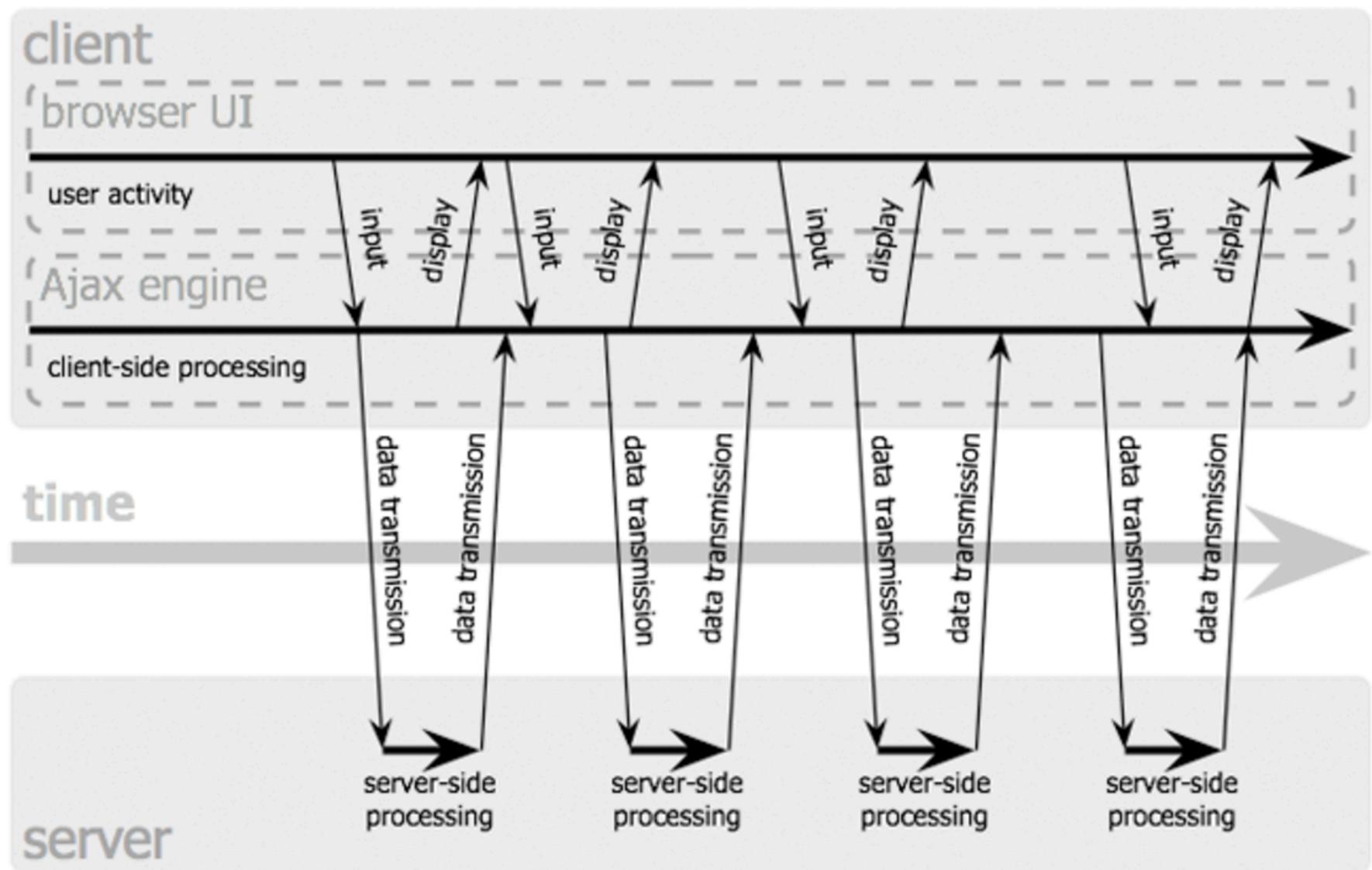
AJAX

- ▶ Asynchronous JavaScript and XML
 - ▶ Pas un nom informatif
 - ▶ Il peut être synchrone
 - ▶ Non seulement XML (par exemple JSON)
- ▶ Possibilité pour le navigateur de communiquer directement avec un serveur sans une recharge complète de la page

COMMUNICATION TYPIQUE DU CLIENT / SERVER



COMMUNICATION DU STYLE AJAX



HIÉRARCHIE DES DONNÉES

- ▶ Bits,
- ▶ Caractères / Chaines
- ▶ Champs,
- ▶ Records
- ▶ Fichiers,
- ▶ Bases de données

SYSTÈMES D'EXPLOITATION

- ▶ Window's, OS X (Mac), Linux pour server / desktop
- ▶ Apple iOS et Google Android pour mobile
- ▶ Java peut être open source, mais il appartient à Oracle
 - ▶ Idem pour MySQL

TYPES DE LANGAGES DE PROGRAMMATION

- ▶ Langue machine
- ▶ Assembleur
- ▶ Langues de haut niveau
 - ▶ Imperatif vs déclaratif
 - ▶ Orienté objet vs fonctionnel
 - ▶ Apprentissage automatique (Machine Learning) \

TECHNOLOGY D'OBJET

- ▶ Les objets (une instance d'une classe) mélangeant des attributs et des comportements (methods)
- ▶ Les langages fonctionnels concernent davantage la transformation des données et le passage des messages.
- ▶ Les objets cachent des informations (encapsulation), ce qui n'est pas nécessairement une bonne chose
- ▶ L'héritage vs la composition sont deux approches de conception dans les systèmes OO

RÉFÉRENCES

- ▶ <http://www.mooreslaw.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/>

D'AUTRE RÉFÉRENCE

- ▶ [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>