Internet and the Web

Overview of this lesson

- Internet, the Web, protocols, IP, domain names and DNS
- client-server architecture, servers, search engines
- web content, languages and formats, creation tools, W3C
- web browser, history, architecture
- HTTP protocol, URLs, download/upload, cookies, HTTP2
- crawling

Link to a PDF of these slides : pdf

Internet vs. The Web

- Internet
  - physical network of computers (or hosts)
  - communicating with a set of protocols
- World Wide Web, Web, WWW :
  - logical collection of hyperlinked documents
    - static and dynamic
    - accessible from the Internet
  - each document (or Web page, or resource) identified by a URL
- Intranet
  - Network of computers not accessible from the general internet
  - Private Web Pages, not accessible from the Internet
- Deep Web
  - World Wide Web not indexed by search engines
- Dark Web
  - World Wide Web content that exists on networks which use the public Internet but which require specific software,

The Internet

Protocol Stack

Layers	Protocols
Applications	HTTP, FTP, SMTP, DNS
Transport	TCP, UDP, ICMP
Network	IP (v4, v6)
Link	Ethernet, 802.11 (ARP)
Physical	Ethernet, 802.11 (physical)

IP : Internet Protocol

- Defined by IETF in 1981
- **Addressing** machines and **routing** over the Internet
- Two versions of the IP protocol on the Internet :
  - IPv4 (very well spread)
  - IPv6 (not that well-spread yet)
- IPv4
  - 4-byte addresses assigned to each computer, e.g., 137.194.2.24.
  - Institutions are given ranges of such addresses, to assign as they will.
  - Problem :
    - only 232 possible addresses
    - a large number of them cannot be assigned to new hosts.
    - many hosts connected to the Internet do not have an IPv4 address (see IPv4 Address Exhaustion)
    - some network address translation (NAT) occurs.
- IPv6 :
  - 16-byte addresses ;
  - much larger address space.Addresses look like

TCP : Transmission Control Protocol

- Defined also by IETF in 1981
- One of the two main transport protocols used on IP, with UDP (User Datagram Protocol)
- Contrarily to UDP, provides reliable transmission of data (acknowledgments)
- Data is divided into small datagrams (<= MTU) that are sent over the network, and possibly reordered at the end point
- Port Number
  - Like UDP, each TCP transmission indicates a source and a destination port number (between 0 and 65535) to distinguish it from other traffic
  - A client usually select a random port number for establishing a connection to a fixed port number on a server
  - The port number on a server conventionally identifies an application protocol on top of TCP/IP : 22 for SSH, 25 for SMTP, 110 for POP3. . .

DNS : Domain Name System

- Defined and modified by IETF
- IPv4 addresses are hard to memorize, and a given service (e.g., a Web site) may change IP addresses (e.g., new Internet service provider)
  - Even more so for IPv6 addresses !
- DNS : a UDP/IP-based protocol for associating human-friendly names (e.g., `www.google.com`, `weather.yahoo.com`) to IP addresses
- Hierarchical domain names :
  - `com` is a top-level domain (TLD), `yahoo.com` is a subdomain thereof, etc.
- Hierarchical domain name resolution :
  - root servers with fixed IPs know who is in charge of TLDs, servers in charge of a domain know who is in charge of a subdomain, etc.
- Nothing magic with `www` in `www.google.com` : just a subdomain of `google.com`.

Pause / Wake up

What do you need to create a web site ?

Pause / Solution

- an IP number
  - is your home IP constant or does it change every time you reboot the box ?
- a domain name
- a configuration of the domain name to point to the IP
- a port redirection on your box for port 80
- a web server on the target machine

The Web

a Client/Server architecture

- A variety of clients are used :
  - graphical browsers
  - textual browsers : w3m, lynx . . .
    - used by visually-impaired people when sites are accessible
  - browsers with speech-synthesis engines
  - crawlers, spiders, robots . . .
- Servers deliver content to the clients :
  - static content (pages, images, . . .)
  - dynamically generated content (php, js, asp, . . .)
- Architectural choice : light-client/heavy-client

Web Servers

Many large software companies have either their own Web server or their own modified version of Apache (notably, GWS for Google).

nginx and lighthttpd are lighter (i.e., less feature-rich, but faster in some contexts) than Apache.

The versions of Microsoft IIS released with consumer versions of Windows are very limited.

Web search engines

- A large number of different search engines, with market shares varying a lot from country to country.
- At the world level :
  - Google vastly dominating (around 80% of the market ; more than 90% market share in France !)
  - Yahoo !+Bing still resists to its main competitor (around 10% of the market)
- In some countries, local search engines dominate the market
  - Baidu with 75% in China,
  - Naver in Korea,
  - Yahoo ! Japan in Japan,
  - Yandex in Russia
- Other search engines mostly either use one of these as backend (e.g., Google for AOL) or combine the results of existing search engines
- Many others : DuckDuckGo, Exalead, . . .

What is Web Content ?

- Textual, visual or aural content experienced when using a browser
  - «Web Page»
  - «Web Site»
  - «Web Application»
- A mix of multiple languages and file formats
  - Used by the client (don't mix with server-side languages)
  - Each with its own usefulness (HTML, CSS, JS . . .)
  - Hierarchically nested : e.g. CSS content in HTML content
  - Referencing each other : hyperlinks, e.g. JS content referenced from HTML content

Statistics

Languages of the Web

- HTML / XHTML
  - Content structuration
  - Basic rendering
- CSS
  - Presentation instructions to render the HTML content
  - Layout, animations, . . .
- SVG
  - Presentation instructions to render rich graphical content
- JS
  - Programmatic behavior to be added to HTML or SVG content

Additional Languages of the Web

- XML
  - Data exchange, validation, . . .
- JSON
  - Data exchange
- MathML
  - Mathematical formulas
- And many others (standards or not)

Example of mixed languages

```
<!DOCTYPE html>
<html>
 <head>
  <title>Hello</title>
  <script>
   window.onload=function(e) { alert("Page loaded!"); };
  </script>
  <style type="text/css">
   body { width: 30%; margin: auto; }
   p {
    font-size: 30px;
    font-family: sans-serif;
   }
  </style>
 </head>
 <body>
  <p>
   <img src="image01.png" style="float: left; margin-right: 5p
```

Tools to create web content

- ***After 40 years, not one good universal HTML editor yet !***
- General purpose tools
  - Text Editor (Atom, Sublime . . .)
  - Integrated Development Environment (Visual Studio, Eclipse . . .)
- Specific Tools
  - DreamWeaver
  - Brackets
  - Aptana
  - . . .
- Code playgrounds
  - JS Fiddle
  - Code Pen
  - CSS Deck
  - JS Bin

Tools to Debug

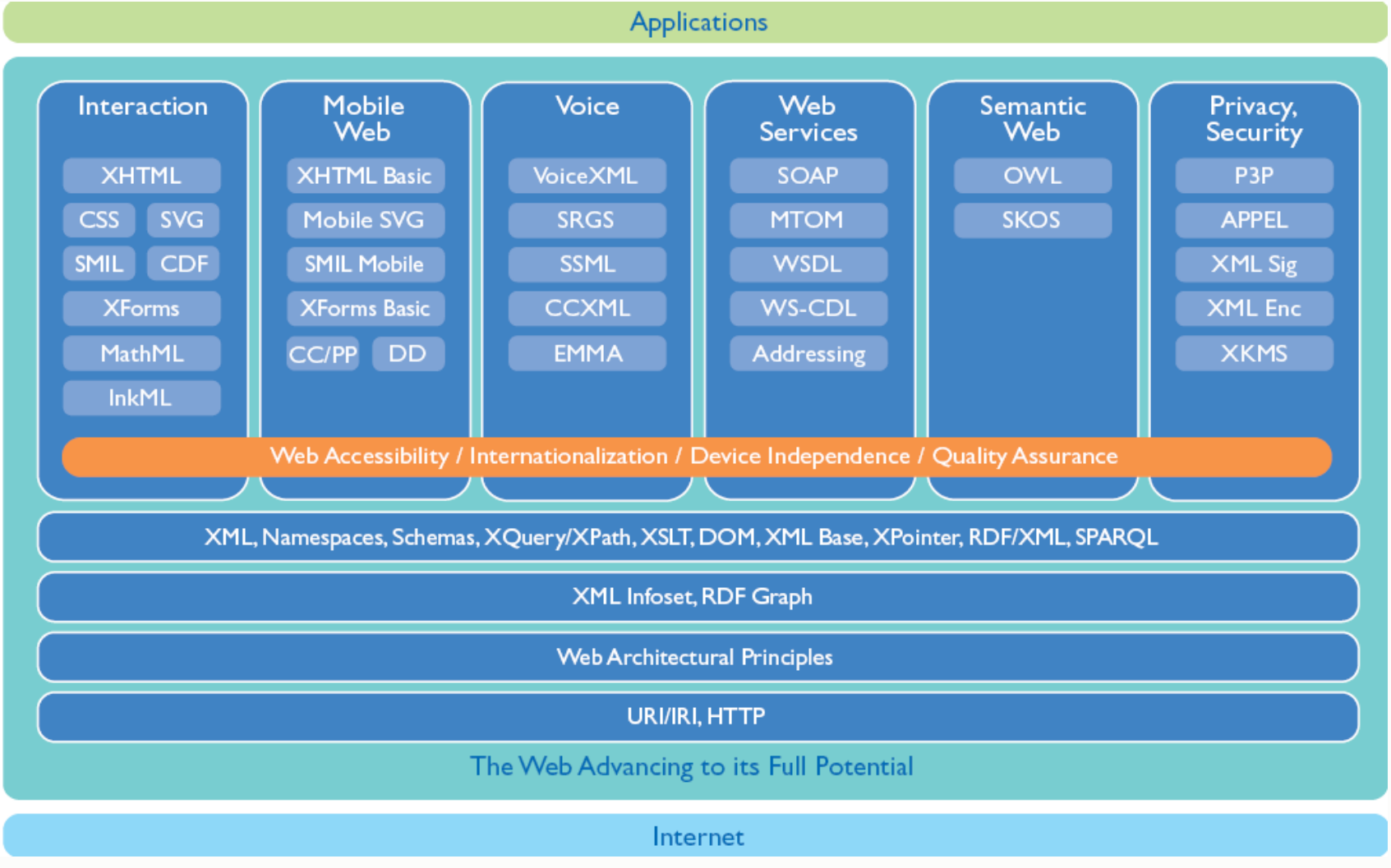
- What kind of debugging ?
  - Source Code Inspector : HTML, CSS, . . .
  - Advanced Inspectors : DOM Tree, CSS Rules
  - JavaScript debug : step-by-step, breakpoints, . . .
  - Network monitoring : requests, timing, . . .
  - Performances : frame rate, CPU, smoothness, memory . . .
- Browser-integrated debugger (F12 / Cmd + Opt + I on Mac)
  - Chrome Dev Tools
  - Mozilla Firebug
  - Microsoft Developer Tools
  - Safari Developer tools

The World Wide Web Consortium (W3C)

- International consortium

Web Content

A Lot of W3C recommendations



Documentation

- W3C Web Platform Docs
- Mozilla Developper Netwok (MDN)
- Microsoft MSDN
- W3C Website
- W3C Document Validator
- WHATWG
- and many more

Pause

What do you need to create web content ?

Pause / Answer

- a web server (see previous pause)
- some content (text with style, images, videos, sounds, etc)
- a way to create an HTML page
  - an editor with export to HTML
  - or knowledge of HTML and a simple text editor
- possibly a validator to check your content is following standards, that it is accessible, that the media are not too big for download or too poor quality. . .
- some way to upload your production onto the web server

Web Browsers

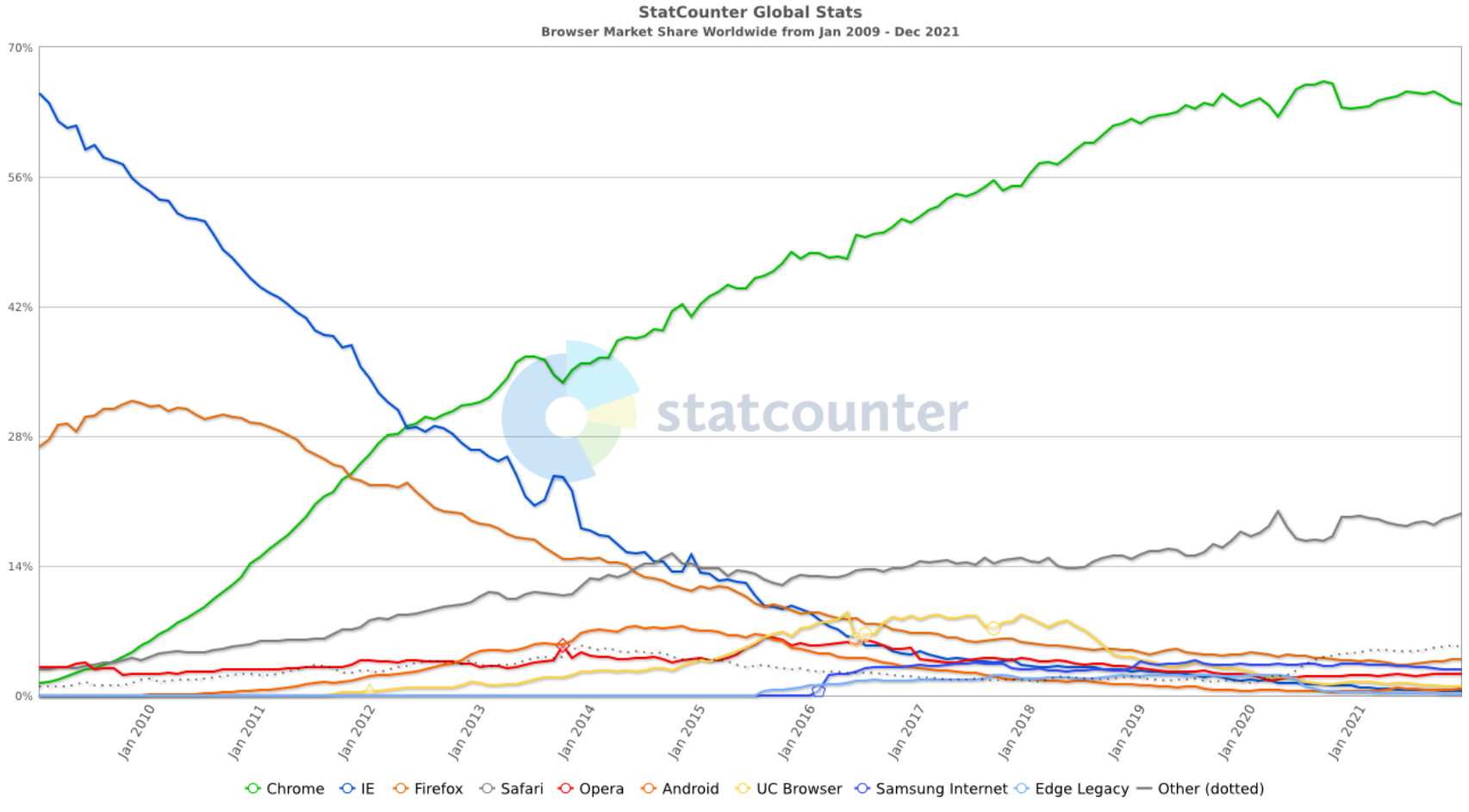
What is a browser ?

- Processing of Web Resources
- **Downloading** of HTML/JS/CSS/Images/Videos . . . using Internet Protocols
  - Sequential/Syncronous vs. Parallel/Asynchronous
- **Rendering** (aural and visual)
- Handling **dynamicity**
- Reacting to **user interactions**
  - Navigation, Click, . . .
- Reacting to **network conditions**
  - TCP Congestion, Streaming, . . . -Processing **animations**

Browsers categories

- Desktop

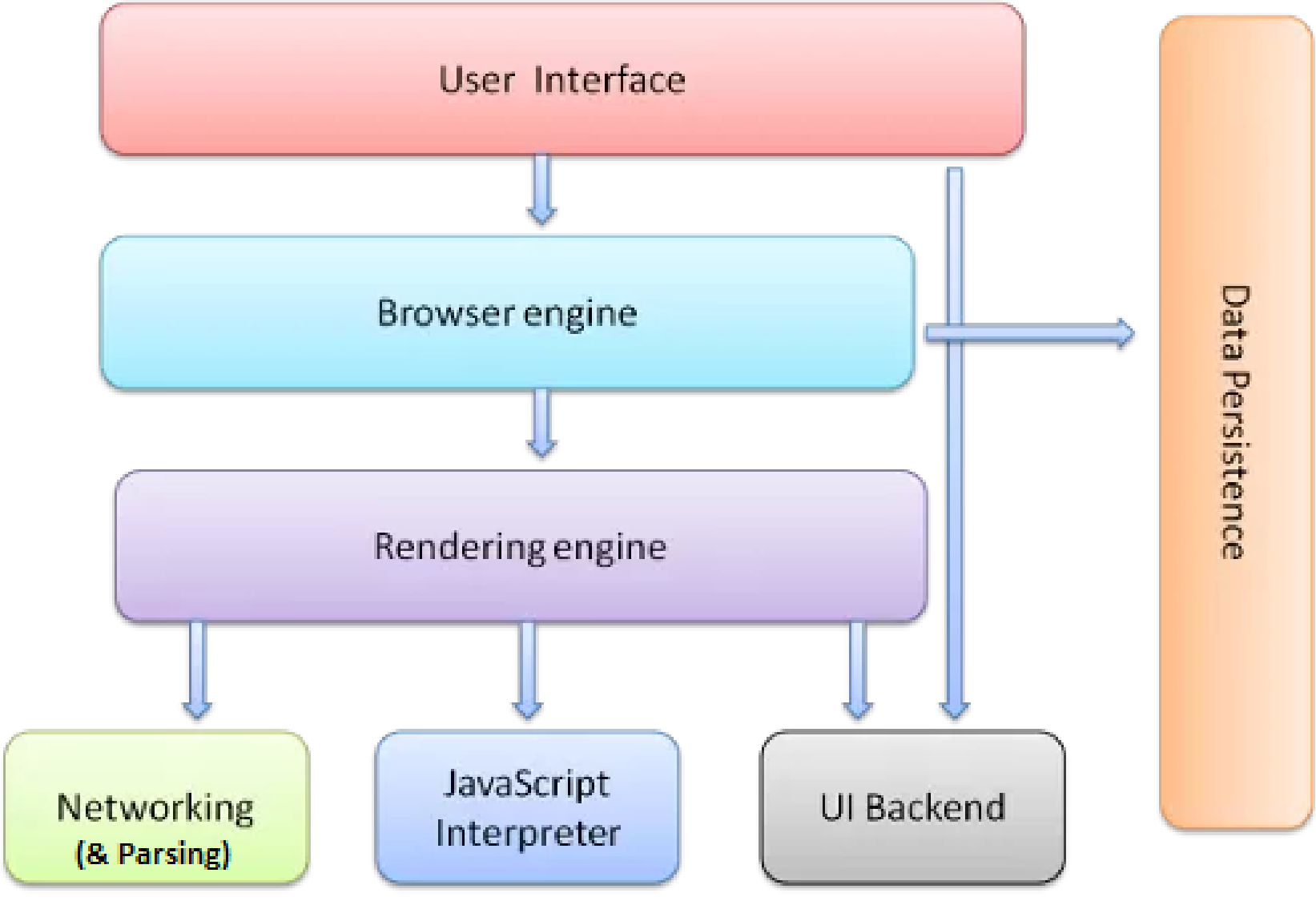
Browser Wars



Browser History

- Long history of browsers
- Rapid evolution recently
- Next versions of major browsers very often
  - Ex : Chrome release a new version every 6 weeks
  - Ex : Firefox 5 (June 2011), Firefox 25 (Oct. 2013)
- Browsers are converging in standards support
  - But still have differences (see sites like CanIUse.com or QuirksMode)

Browsers Simplified Architecture

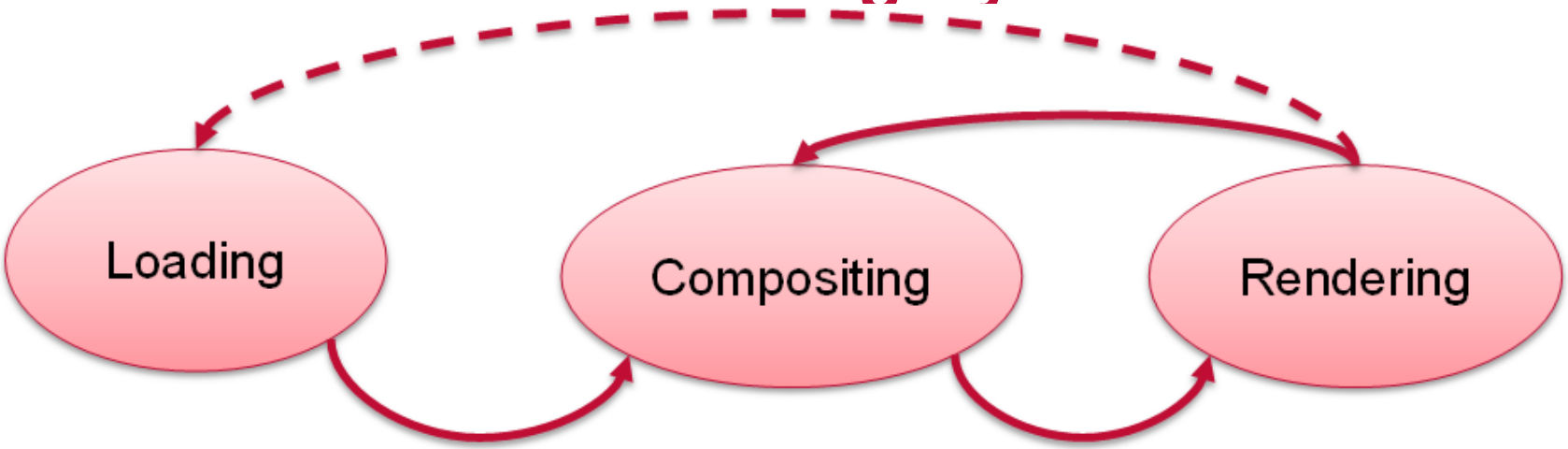


HTML5 Rocks

Browsers components

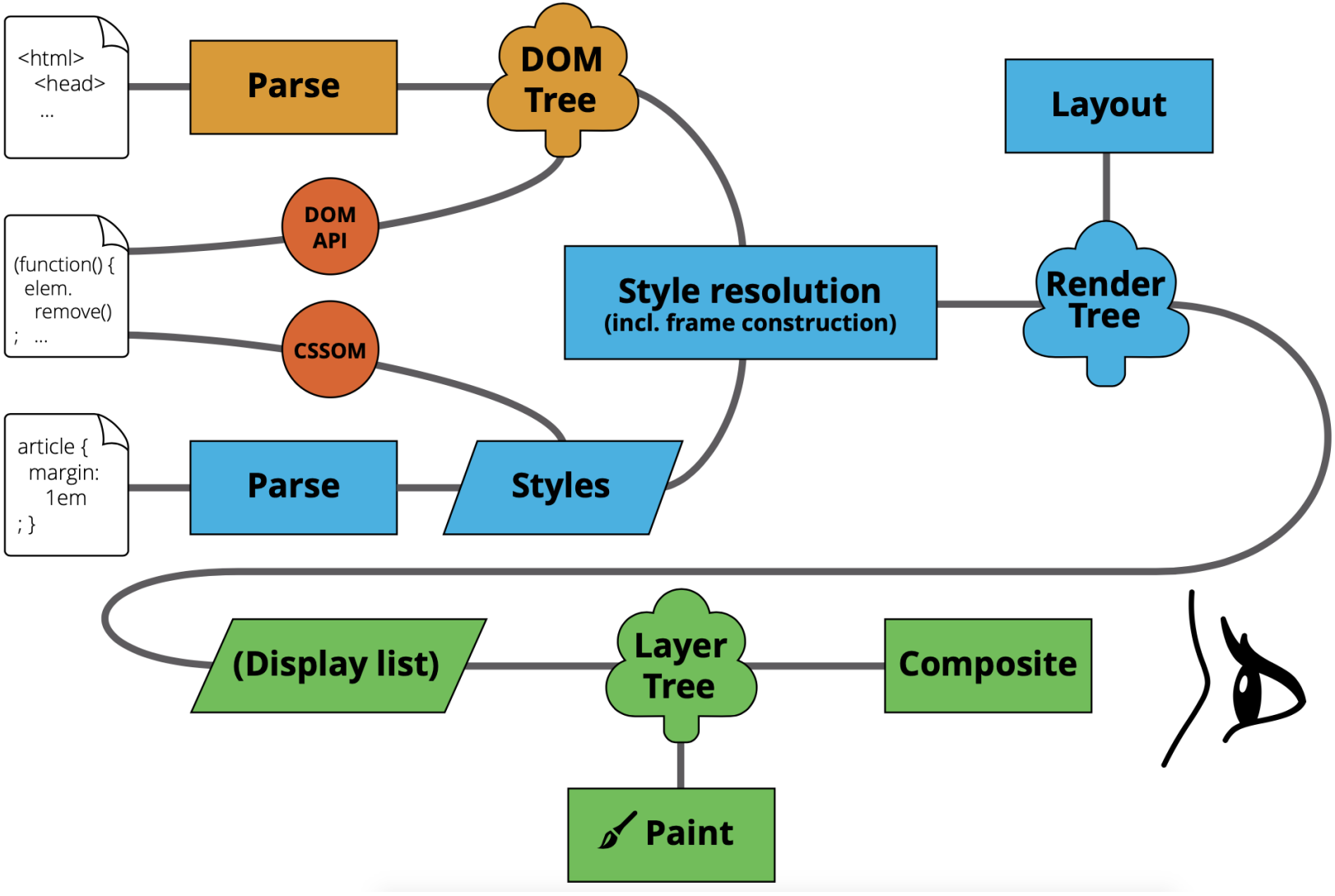
Browser	Rendering Engine
Edge	EdgeHTML
Internet Explorer	Trident
Firefox and alike (IceWeasel, Seamonkey. . .)	Gecko
Safari	WebKit
Chrome	Blink (previously Webk
Opera	Blink (previously Prest

Browser Processing Cycle



- Loading :
  - creating a memory representation from input
- Compositing :
  - applying styles, interactivity (scripting), animations, synchronizing media elements . . . to produce static data to be rendered
- Rendering of static data
  - Converting text to pixels, vector graphics to pixels, applying graphical effects
  - Management of graphics card, video card, sound card
- Performance
  - High quality rendering, frame rate, reactivity, memory

Browser Processing



See Mozilla's presentation

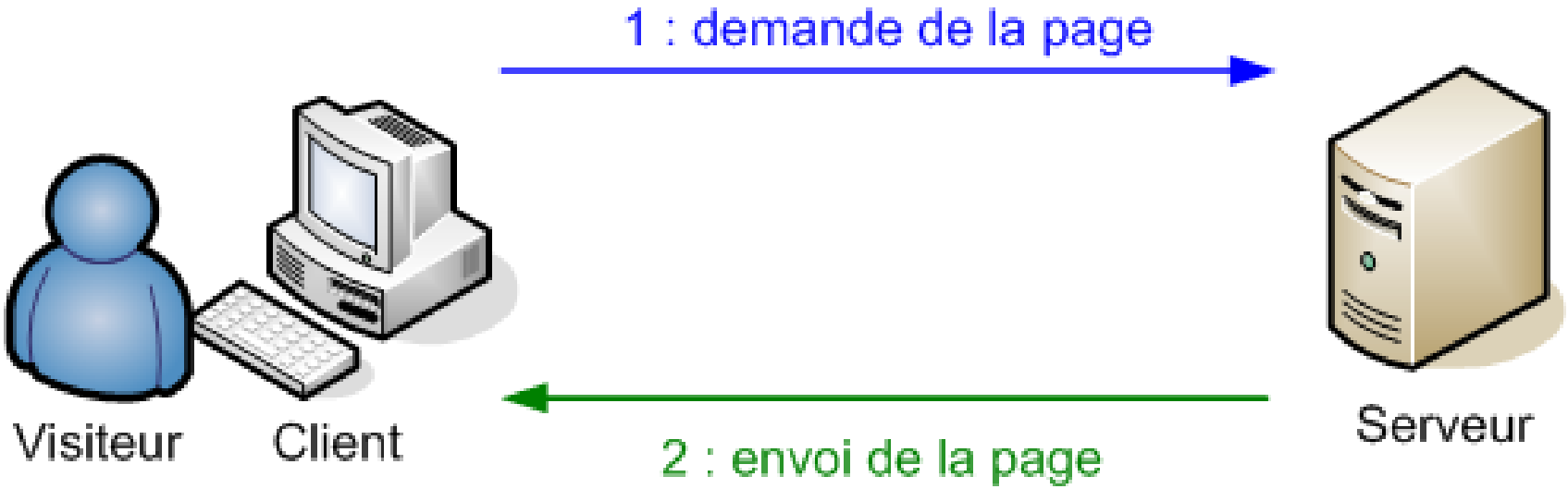
Pause

How should you choose your browser ?

Pause / Answer

- standard : you should see/experience no difference
- fast : obvious
- not a memory hog : so that you can use it at the same time as all your other applications (the web browser is now the biggest application on my machine)
- has all the extensions you need : e.g. password manager, ad block, scripting, your video streaming provider. . .
- helps you reduce tracking
- does not lock you into a silo (e.g. Google, Apple, Microsoft, Amazon. . .)

HTTP



- Hyper Text Transfer Protocol, standardized by IETF
- Application protocol at the basis of the World Wide Web
- history & versions :
  - HTTP (1991, proposed by Tim Berners-Lee),
  - HTTP/1.0 (1996, initial version, RFC 1945),
  - HTTP/1.1 (1997, current deployments, RFC 2068 and 2616),
  - HTTP/2.0 (2015, latest version, in deployment, RFC 7540)
- Client/server protocol
  - The client is a “User-Agent” (Firefox, wget, curl . . .)
  - HTTP servers : Apache, Microsoft IIS, node.js
- Protocol used to download resources

URL : Uniform Resource Locator

Initially standardized by IETF (new versions in development by IETF/W3C/WHATWG)

https

://

www.example.com

:443

/path/to/

?name=foo&town=bar

#para

scheme

hostname

port

Relative URLs

- With respect to a context (e.g., the URL of the parent document, the **base URL**)
- If context is : `https://www.example.com/toto/index.html`

relative URL	Absolute URL
/titi	https ://www.example.com/titi
tata	https ://www.example.com/toto/tata
#tutu	https ://www.example.com/index.html#tutu

Identifying Web Resources

- File/URL extension
- Resources may not have one, or it may be wrong
  - Ex : http ://www.example.org/
  - Ex : http ://www.example.org/generate.cgi ?user=12
- Not reliable !
- Sniffed type
- E.g. use of ‘magic number’ (registered in MIME type)
  - Ex : “47 49 46 38 37 61” GIF89a
- E.g Detection of file header (XML)
- May be abused
- MIME type or Internet Media Type
  - Used in HTTP Content-Type header
  - ‘/’ (‘ ;’ parameters)*
  - 5 major types : audio, video, image, text, application
  - Subtypes specific to a payload (‘x-. . .’ are proprietary)
  - Should be trusted

HTTP Messages

- Message = Header + Body
  - Textual header (not necessarily for the resources)
- Message type = Requests or responses
- Request=Method+URL+ProtocolVersion+Header(+data)
- Method
  - GET
  - POST
  - HEAD
  - OPTIONS
  - PUT
  - DELETE
  - TRACE
  - CONNECT
  - PATCH
- Response=ProtocolVersion+Response Code+Header+Resource

GET

- Simplest type of request.
- Possible parameter are sent at the end of a URL, after a ‘?’
  - Not applicable when there are too many parameters, or when their values are too long (total length < 2000 chars).
- Example :
  - URL in the browser

    `http://www.google.com/search?q=hello`
  - Corresponding HTTP Request

```
GET /search?q=hello HTTP/1.1
Host: www.google.com
```

POST

- Method only used for submitting forms.
- Example :

```
POST /php/test.php HTTP/1.1
Host: www.w3.org
Content-Type: application/x-www-form-urlencoded
Content-Length: 100
type=search&title=The+Dictator&format=long&country=US
```

Parameter encoding

- By default, parameters are sent (with GET or POST) in the form : `name1=value1&name2=value2`
  - special characters (accented characters, spaces. . .) are replaced by codes such as +, %20
  - This way of sending parameters is called `application/x-www-form-urlencoded`.
- For the POST method, another heavier encoding can be used (several lines per parameter)
  - similar to the way emails are built : mostly useful for sending large quantity of information.
  - Encoding named `multipart/form-data`.

Response Codes

- Success (2xx)
  - OK (200)
  - ...
- Redirections (3xx)
  - Permanent redirection (301)
  - Temporary redirection (302)
  - No modification (304)
  - ...
- Request Errors (4xx)
  - Bad request (400)
  - ...
  - Forbidden(403)
  - Not found (404)
- Server Errors (5xx)
  - Internal Error (500)
  - ...

Identifying clients/servers

- « User Agent »
- Identifier string exchanged in HTTP
  - Browser name, rendering engine,
- Used to
  - Work around known bugs
  - Serve tailored content (e.g. smartphone version)
- User Agent detection vs. Feature detection

```
Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20120427 Fire
Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, l
Mozilla/5.0 (MSIE 9.0; Windows NT 6.1; Trident/5.0)
Opera/9.80 (Macintosh; Intel Mac OS X; U; en) Presto/2.2.15 Vers
Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_3_2 like Mac OS X; en-us)
```

```
Server: Apache/2.0.59 (Unix) mod_ssl/2.0.59 OpenSSL/0.9.8e PHP/
```

Authentication

- HTTP allows for protecting access to a Web site by an identifier and a password
  - Warning : (most of the time) the password goes through the network uncrypted (but for instance, just encoded in Base64, revertible encoding)

```
GET ... HTTP/1.1
Authorization: Basic dG90bzp0aXRp
```

- HTTPS (variant of HTTP that includes encryption, cryptographic authentication, session tracking, etc.) can be used instead to transmit sensitive data

Content Negotiation

- A Web client can specify to the Web server :
  - the content type it can process (text, images, multimedia content), with preference indicators
  - the languages prefered by the user
- The Web server can thus propose different file formats, in different languages.
- In practice
  - content negociation on the language works, and is used,

```
Accept: text/html,application/xhtml+xml,application/xml; q=
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
```

  - content negociation on file types does not work well because of bad default configuration of some browsers.

Cookies

- Information, as key/value pairs, that a Web server asks a Web client to keep and retransmit with each HTTP request (for a given domain name).
- Can be used to keep information on a user as she is visiting a Web site, between visits, etc. : electronic cart, identifier, and so on.
- Practically speaking, most often only stores a session identifier, connected, on the server side, to all session information (connected or not, user name, data. . .)
- Simulates the notion of session, absent from HTTP itself
- Limited in size

```
Set-Cookie: session-token=RJYBsG//azkfZrRazQ3SPQhlo1FpkQka2; p

Cookie: session-token=RJYBsG//azkfZrRazQ3SPQhlo1FpkQka2
```

Byte-ranges

- The client can ask for only a portion of the file
- This is useful if the download is interrupted

```
Range: bytes=0-42
```

Conditional downloads

- A client can ask for downloading a page only if it has been modified since some given date.
- Most often not applicable, the server giving rarely a reliable last modification date (difficult to obtain for dynamically generated content !).

```
If-Modified-Since: Wed, 15 Oct 2008 19:40:06 GMT

304 Not Modified

Last-Modified: Wed, 15 Oct 2008 19:20:00 GMT
```

Originating URL

- When a Web browser follows a link or submits a form, it transmits the originating URL to the destination Web server.
- Even if it is not on the same server !

```
Referer: http://www.google.fr/
```

Persistent connection / Keep-alive

- Ability to reuse the same TCP Connection for multiple HTTP requests
- But not full duplex : exchange 1 (GET file1) has to be finished to start exchange 2 (GET file2)

Pipelining

- Make multiple HTTP requests in parallel, receive in sequence
- Head of line blocking problem
- Parallel TCP connections : often rejected by servers

HTTP/2

- Initially developped/deployed by Google as the SPDY protocol
  - 2009 - SPDY 1
  - 2010 - Google Chrome
  - 2011 - Twitter.com, Google.com
  - 2012 - Apache, Nginx, Facebook, F5, Wordpress
  - 25/09/2012 - HTTPBis > SPDY > HTTP2
  - 2015 - HTTP2 Draft 17
  - 2015 - HTTP2 Approved
- Developped to reduce latency and adapt to Web pages requiring many documents/resources
  - Backward compatibility with HTTP/1.1, but with
    - Upgrade mechanism
    - Frames are binary
    - Header compression
  - Advanced keep-alive/pipelining with data multiplexing
  - full-duplex HTTP requests/responses (no head-of-line blocking)
  - push content

WebSocket

- New protocol, built on top of TCP (RFC 6455) :
  - ws ://example.com/path/to/ws
  - wss ://example.com/path/to/ws
- Initiated by the client, but full-duplex message-based communication
- Communication starts with HTTP GET with Upgrade on port 80/443
- Used for server-sent events (no more long-polling)
- Can exchange text or binary messages
- Can replace AJAX
- ***Allows the server to send messages at any time after the initial client connection***

WebSocket in practice

- WebSocket protocol is an extension of the HTTP protocol
- client-side :
  - `const connection = new WebSocket("ws://"+location.host);`
  - `connection.onmessage = function(msg) {...};`
  - `connection.send(...);`
- server-side :
  - WebSocket server is created from an HTTP server
  - `const wsServer = new WebSocketServer({httpServer: server});`
  - `wsServer.on('request', processRequest);`

```
function processRequest(request) {
    const connection = request.accept(null, request.origin);
    connection.on('message', (o) => ...);
    connection.on('close', () => ...);
    connection.on('error', (o) => ...);
    connection.send(...);
```

Pause

What happens between client and server on the web ?

Pause / Anwser

- download of text and resources
- upload of text and resources
- exchange of admin info on client and server in **text**, some of it inside the URLs
- the server stores context on the client (cookie)
- all communication in clear unless encrypted with HTTPS
- optimisations are possible : caching, better protocol

Crawling

- Crawlers, (Web) spiders, (Web) robots :
  - autonomous user agents that retrieve pages from the Web
  - ex : Scrapy
- Basics of crawling :
  - 1. Start from a given URL or set of URLs
  - 2. Retrieve and process the corresponding page
  - 3. Discover new URLs (cf. next slide)
  - 4. Repeat on each found URL
- Termination condition
  - No real condition : need to refresh content
  - Limit the number of pages ?
  - Size of the Web : at least 4.62 billion pages (source : Indexed Web)
- Graph-browsing problem
  - **deep-first** : not very adapted, possibility of being lost in robot traps
  - **breadth-first**, with priority for popular sites
  - combination of both : breadth-first with limited-depth

Sources of new URLs

- From HTML pages :
  - hyperlinks <a href="...">
  - media links <img src="..."> <embed src="..."> <object data="..."> <video src="..."> ...
  - nested documents <frame src="..."> <iframe src="...">
  - JavaScript links window.open("...")
  - etc.
- From other hyperlinked content (e.g. PDF, ...)
- From non-hyperlinked URLs (in text files, in HTML text content, ...)
- From Sitemaps

Scope of a crawler

- Goals :
  - Limit the size of the crawled content, to important pages
  - Avoid robot traps
- Filter by **DNS domains** : easy filtering of URLs
- Filter by a given topic : **focused crawling** techniques [Chakrabarti et al., 1999, Diligenti et al., 2000] based on classifiers of Web page content and predictors of the interest of a link.

Identifying duplicates

- Problem :
  - Identifying duplicates or near-duplicates on the Web to prevent multiple indexing
- trivial duplicates : same resource at the same canonized URL :

```
http://example.com:80/toto
http://example.com/titi/../toto
```

- exact duplicates
  - identification by hashing
- near-duplicates :
  - more complex !
  - timestamps, tip of the day, etc.

Identifying duplicates : Hashing

- Definition :
  - A **hash function** is a deterministic mathematical function transforming objects (numbers, character strings, binary. . .) into fixed-size, seemingly random, numbers. The more random the transformation is, the better.

Identifying duplicates : near-duplicates

- Edit distance.
  - Count the minimum number of basic modifications (additions or deletions of characters or words, etc.) to obtain a document from another one.
  - Good measure of similarity, and can be computed in O(mn) where m and n are the size of the documents.
  - Does not scale to a large collection of documents (unreasonable to compute the edit distance for every pair !).
- Shingles.
  - Idea : two documents similar if they mostly share the same succession of k-grams (succession of tokens of length k).
  - Example :

```
I like to watch the sun set with my friend.
My friend and I like to watch the sun set.
S ={i like, like to, my friend, set with, sun set, the sun, to
T ={and i, friend and, i like, like to, my friend, sun set, the
```

Crawling architecture : Ethics

- Per-server exclusion : robots.txt at the root of a Web server.

```
User-agent: *
Allow: /searchhistory/
Disallow: /search
```

- Per-page exclusion (de facto standard). <meta name="ROBOTS" content="NOINDEX,NOFOLLOW">
- Per-link exclusion (de facto standard). <a href="toto.html" rel="nofollow">Toto
- Avoid **Denial Of Service (DOS)**
  - wait 100ms/1s between two repeated requests to the same Web server

Crawling architecture : Parallel Processing

- Network delays, waits between requests :
  - **Per-server queue** of URLs
  - Parallel processing of requests to different hosts :
    - **multi-threaded** programming
    - **asynchronous** inputs and outputs (select, classes from java.util.concurrent) : less overhead
  - Use of **keep-alive** to reduce connexion overheads

Crawling architecture : Refreshing URLs

- Content on the Web changes
- Different change rates :
  - online newspaper main page : every hour or so
  - published article : virtually no change
- Continuous crawling, and identification of change rates for adaptive crawling : how to know the time of last modification of a Web page ?

Summary of this lesson

- Internet, the Web, protocols, IP, domain names and DNS
- client-server architecture, servers, search engines
- web content, languages and formats, creation tools, W3C
- web browser, history, architecture
- HTTP protocol, URLs, download/upload, cookies, HTTP2
- crawling