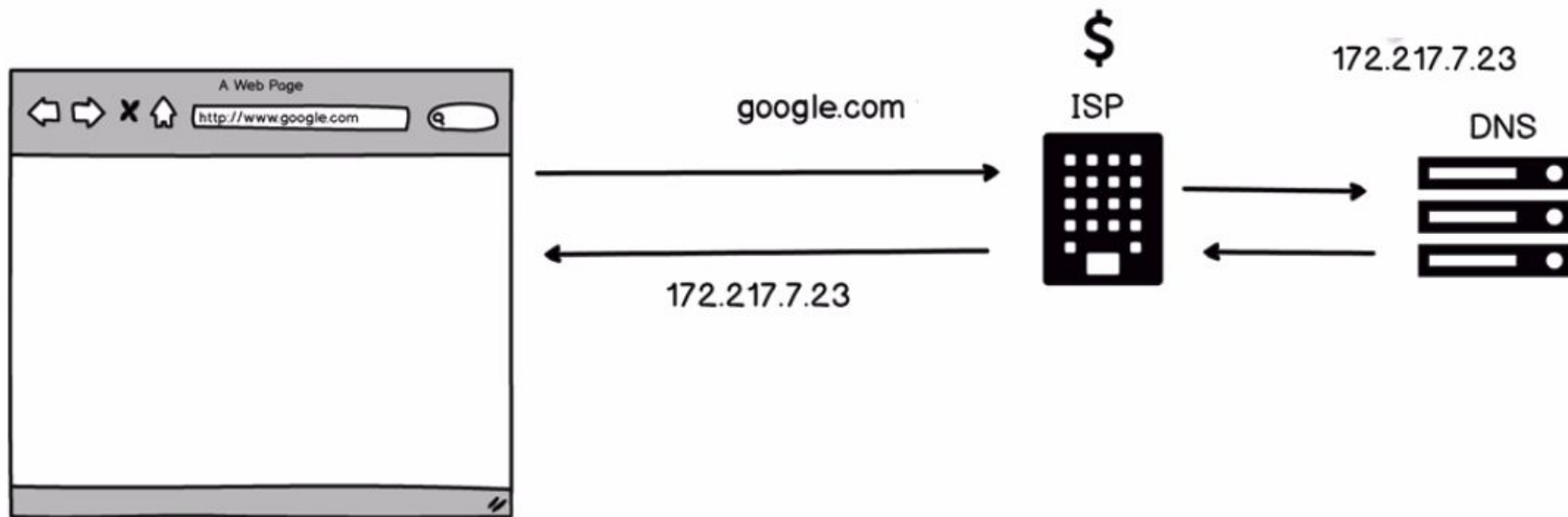
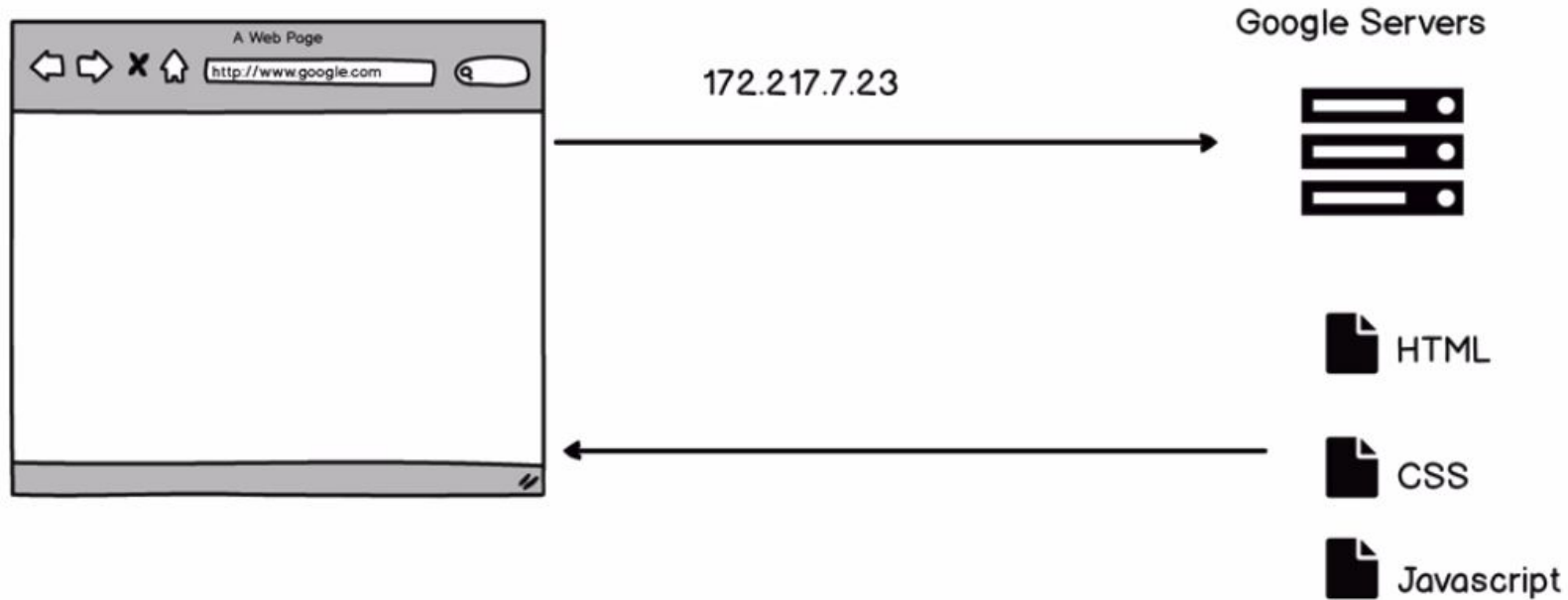

Développement Web

— Gounane Said —
gounane.said@gmail.com

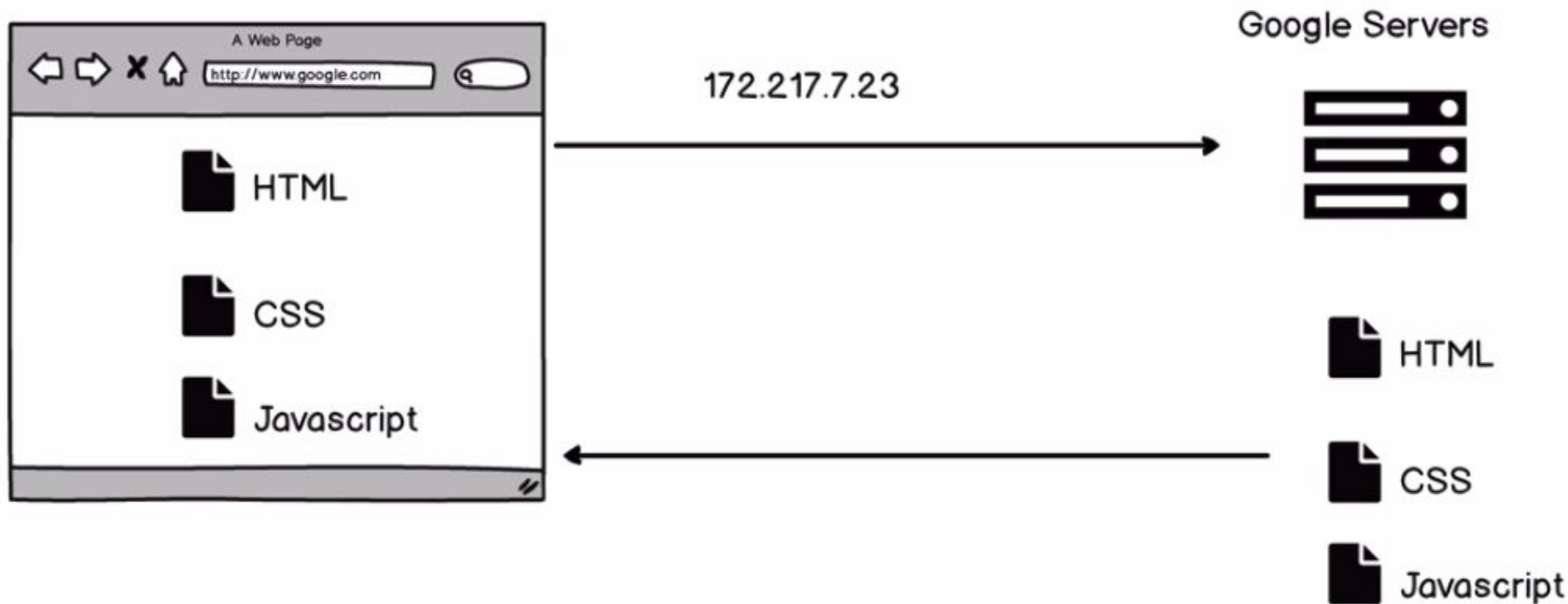
Introduction



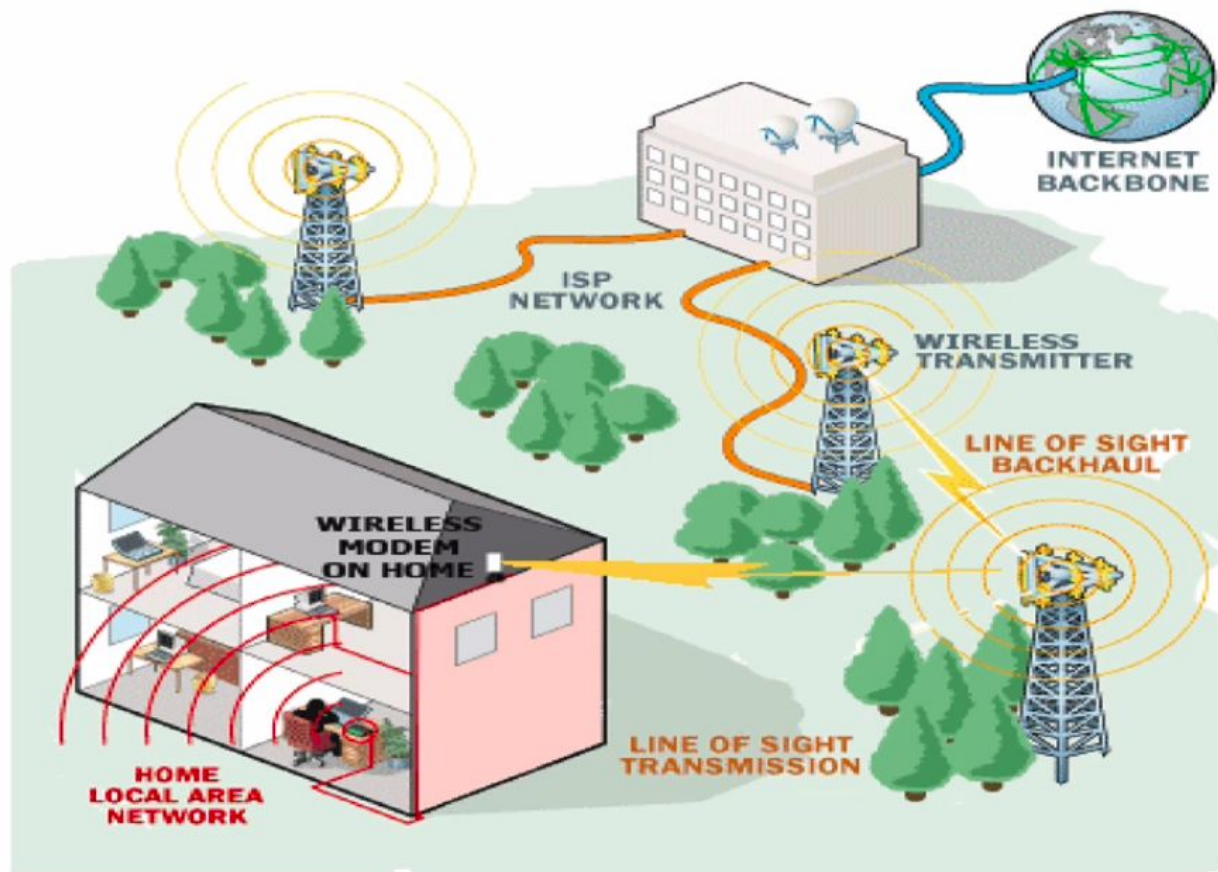
Introduction



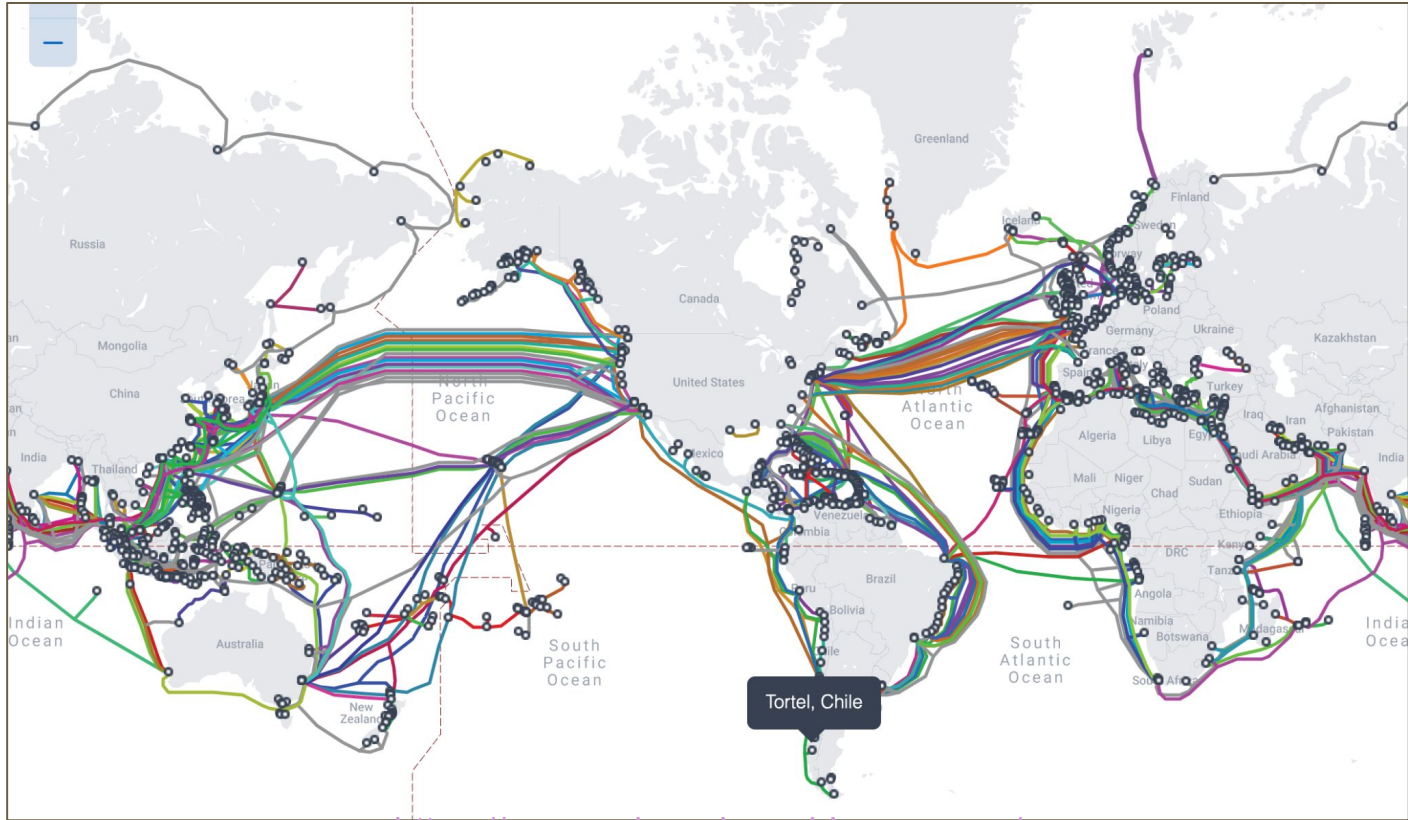
Introduction



Internet



Internet



<https://www.submarinecablemap.com/>

Traceroute

Dans un terminal tapez: \$traceroute google.com

```
~/Desktop traceroute google.com
```

```
traceroute to google.com (172.217.1.14), 64 hops max, 52 byte packets
```

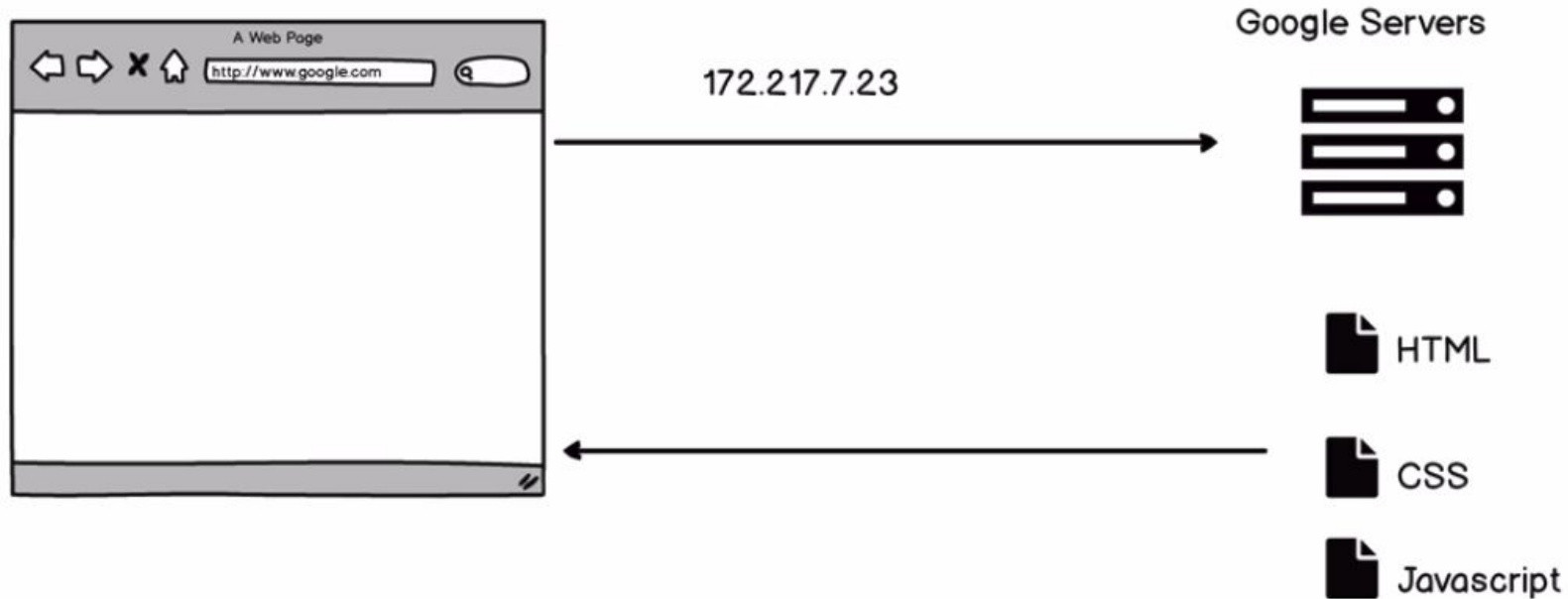
```
1  router.asus.com (192.168.1.1)  1.282 ms  1.282 ms  1.282 ms
2  10.68.128.1 (10.68.128.1)  8.355 ms  8.345 ms  8.009 ms
3  10.0.66.69 (10.0.66.69)  8.970 ms  8.446 ms  9.744 ms
4  10.0.18.33 (10.0.18.33)  10.317 ms  11.940 ms
   10.0.18.25 (10.0.18.25)  11.968 ms
5  17-6-226-24.rev.cgocable.net (24.226.6.17)  12.075 ms  12.430 ms  12.095 ms
6  209.85.173.40 (209.85.173.40)  11.064 ms  11.185 ms  10.714 ms
7  108.170.250.225 (108.170.250.225)  11.732 ms  11.092 ms
   108.170.250.241 (108.170.250.241)  10.771 ms
8  216.239.35.233 (216.239.35.233)  10.259 ms  11.475 ms
   216.239.35.235 (216.239.35.235)  10.905 ms
9  yyz10s14-in-f14.1e100.net (172.217.1.14)  10.000 ms  10.000 ms  10.000 ms
```

```
~/Desktop
```

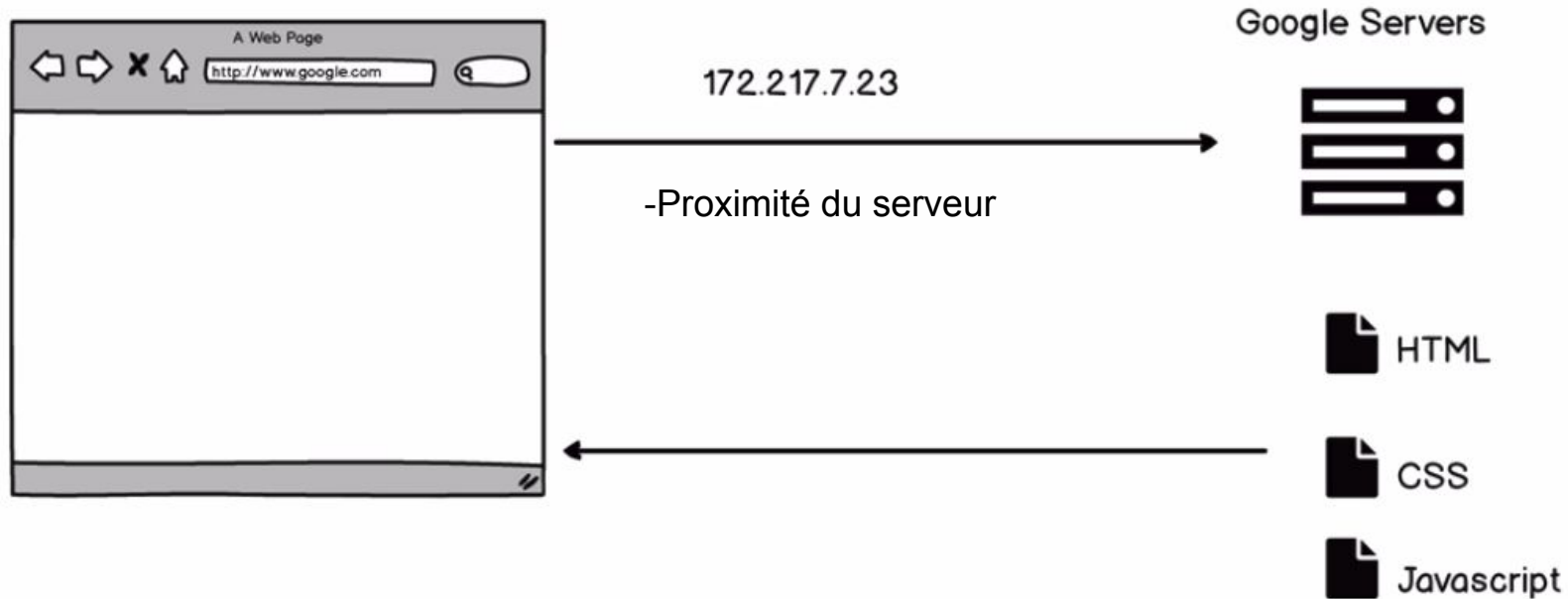
Home Router

Google Server

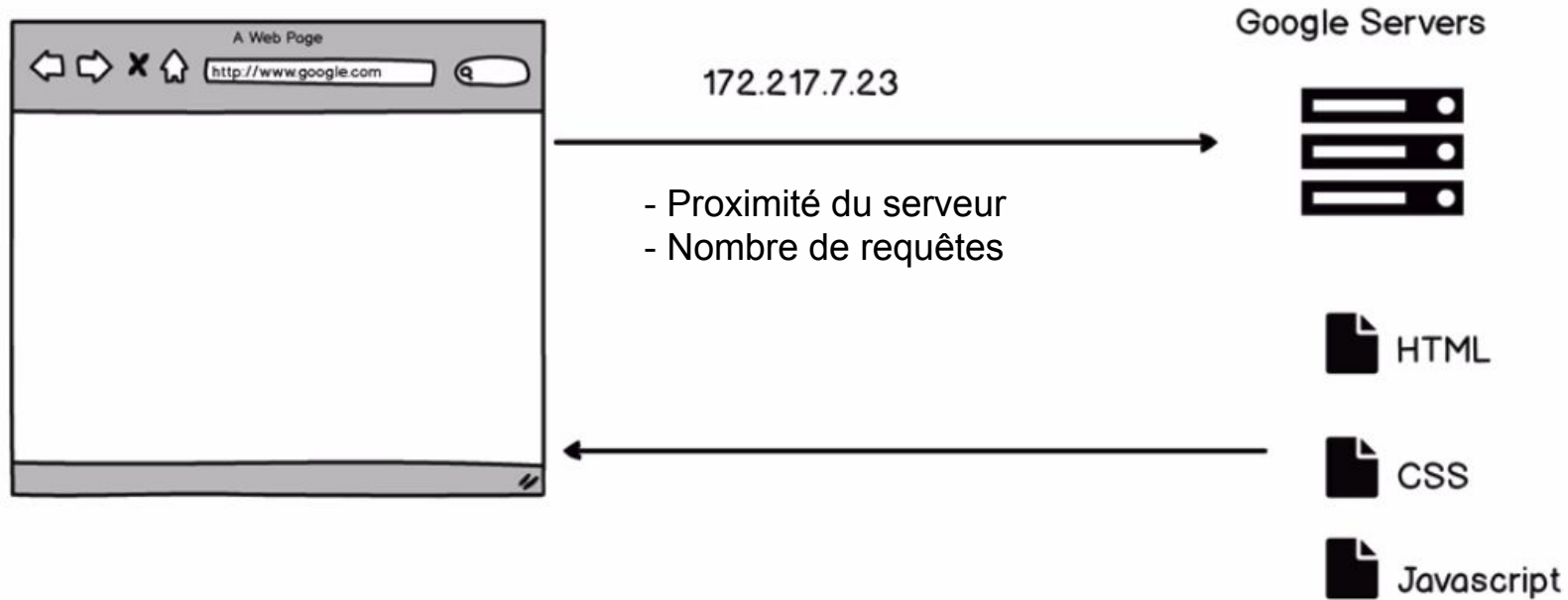
Pour un site réactif



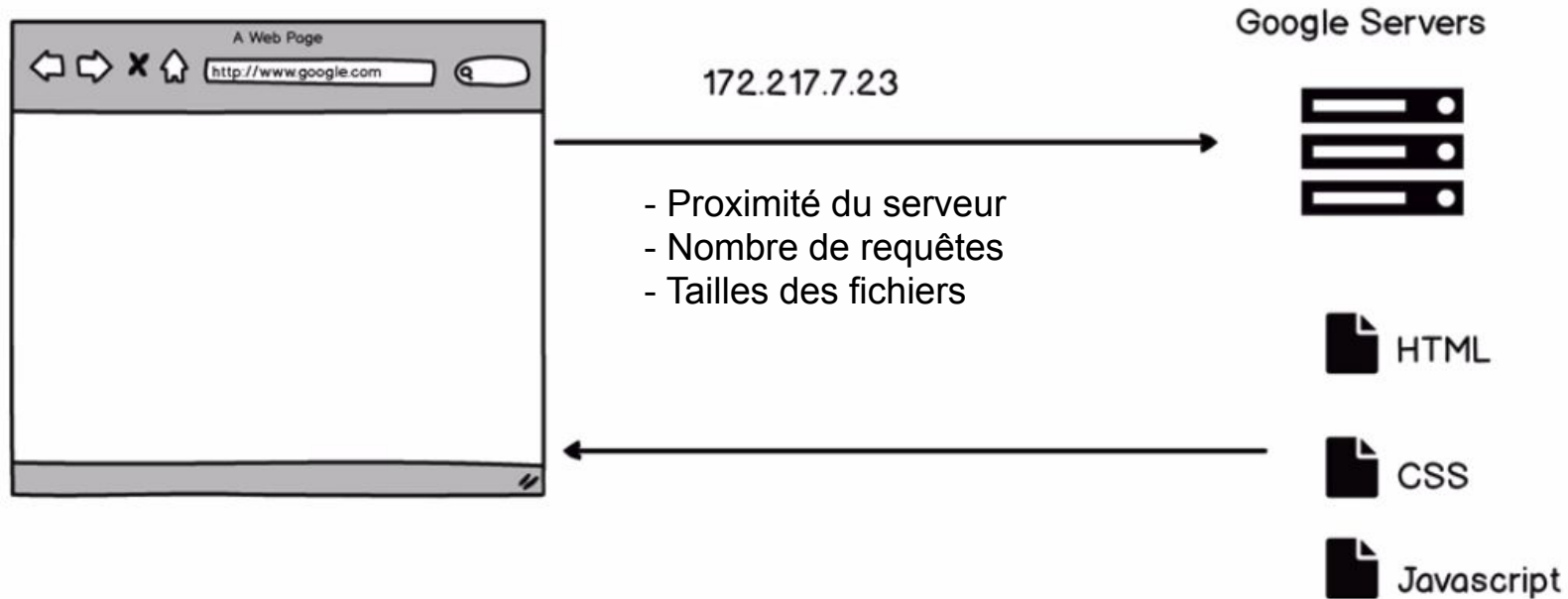
Pour un site réactif



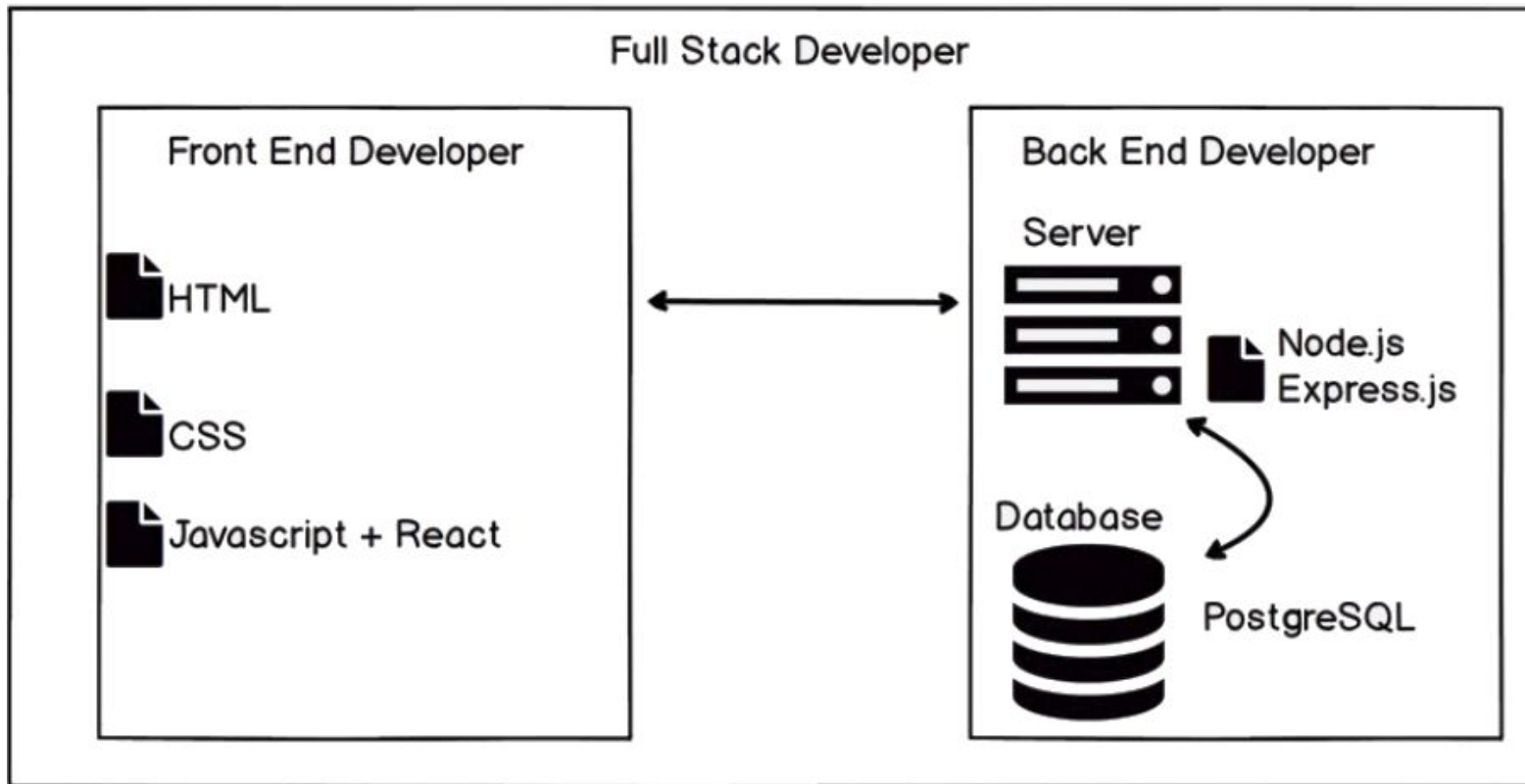
Pour un site réactif



Pour un site réactif



Que fait un développeur web?



Histoire du web

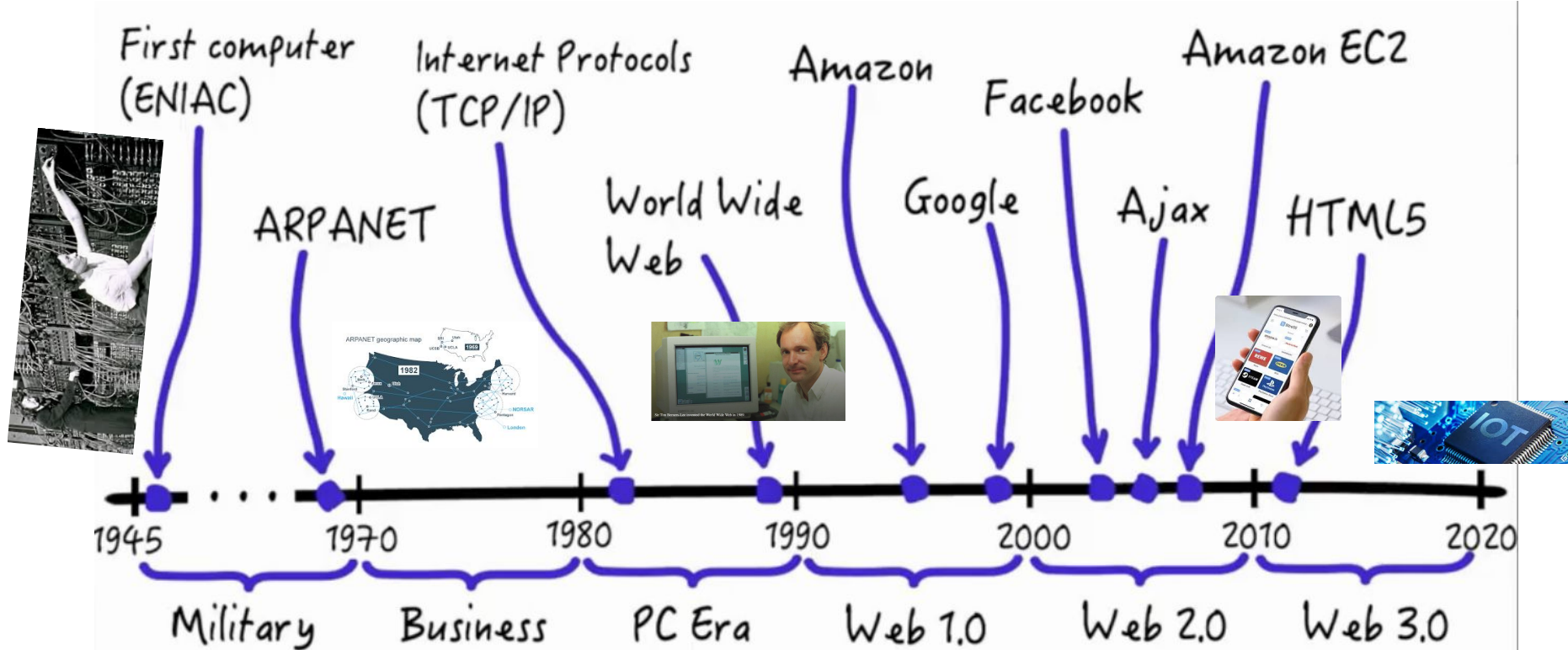


Sir Tim Berners-Lee invented the World Wide Web in 1989.

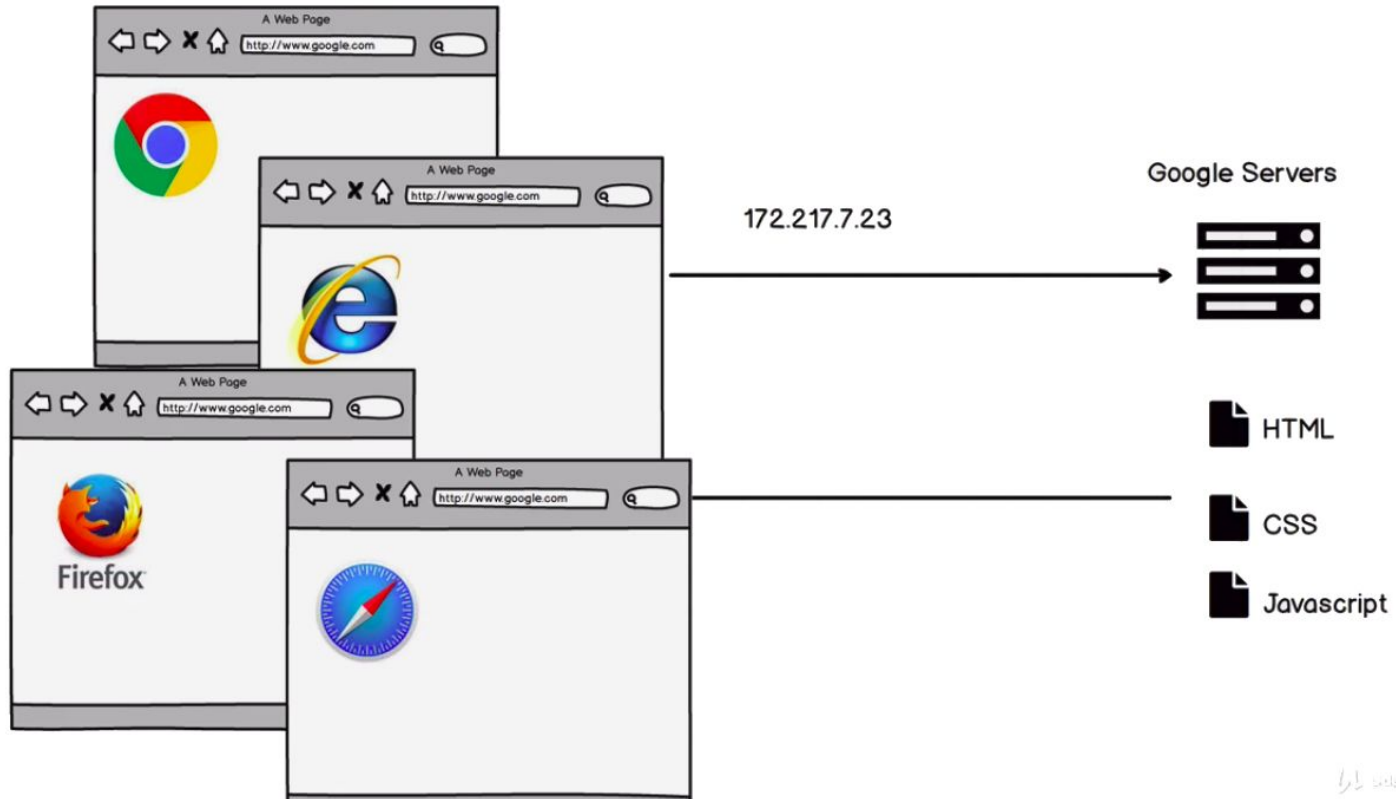
Histoire du web (internet vs web)

1. 40 maps that explain the internet : <https://www.vox.com/a/internet-maps>
2. Premier site web: <http://info.cern.ch/hypertext/WWW/TheProject.html>

Histoire du web (internet vs web)



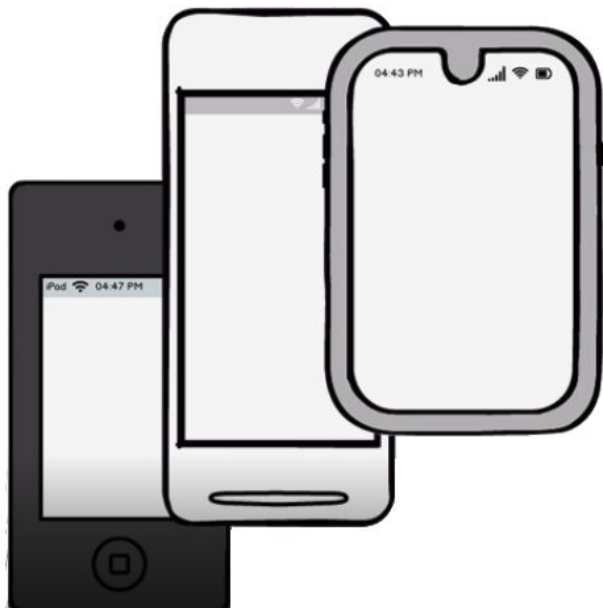
Browser war



Devices



Firefox



172.217.7.23



Google Servers



HTML



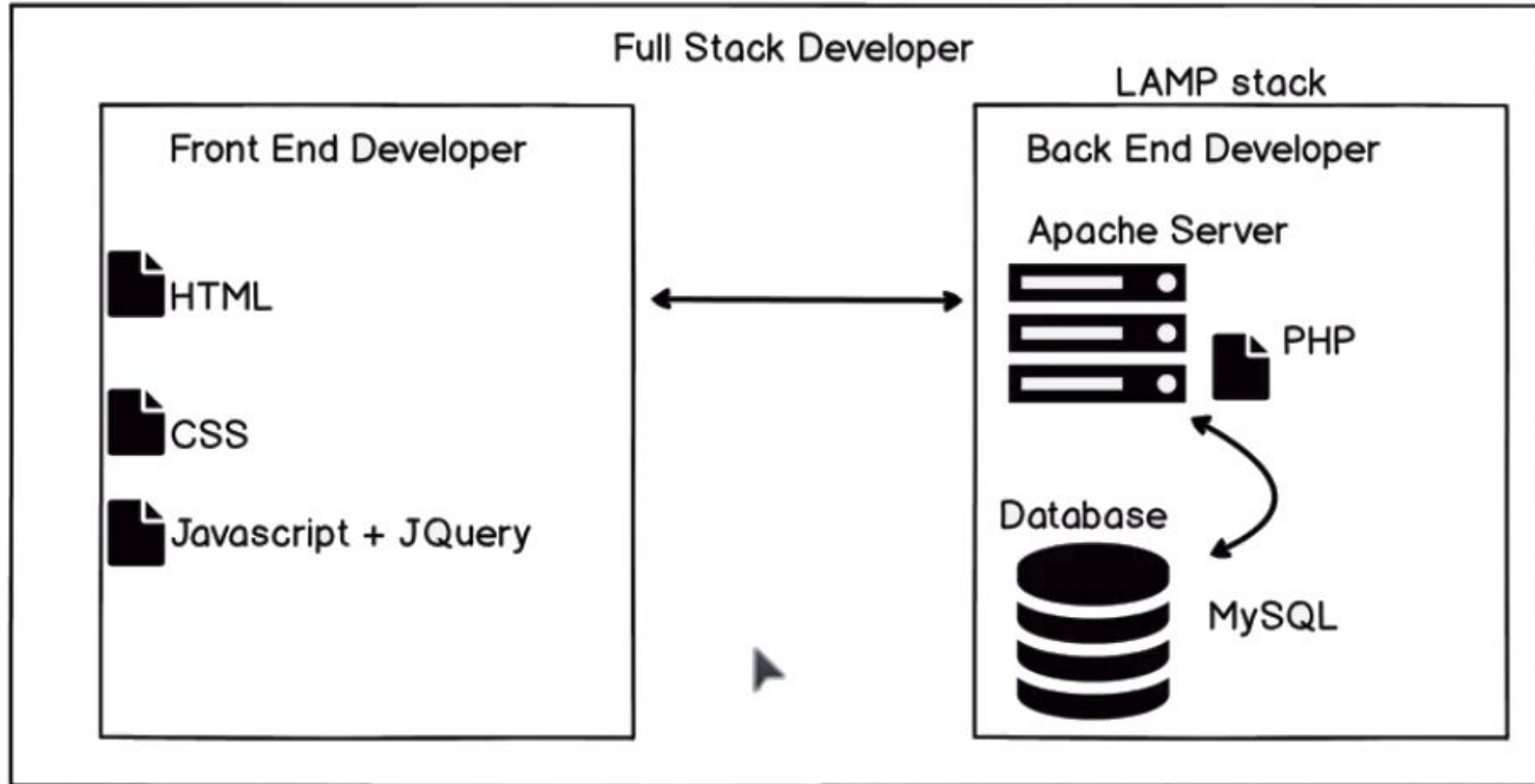
CSS



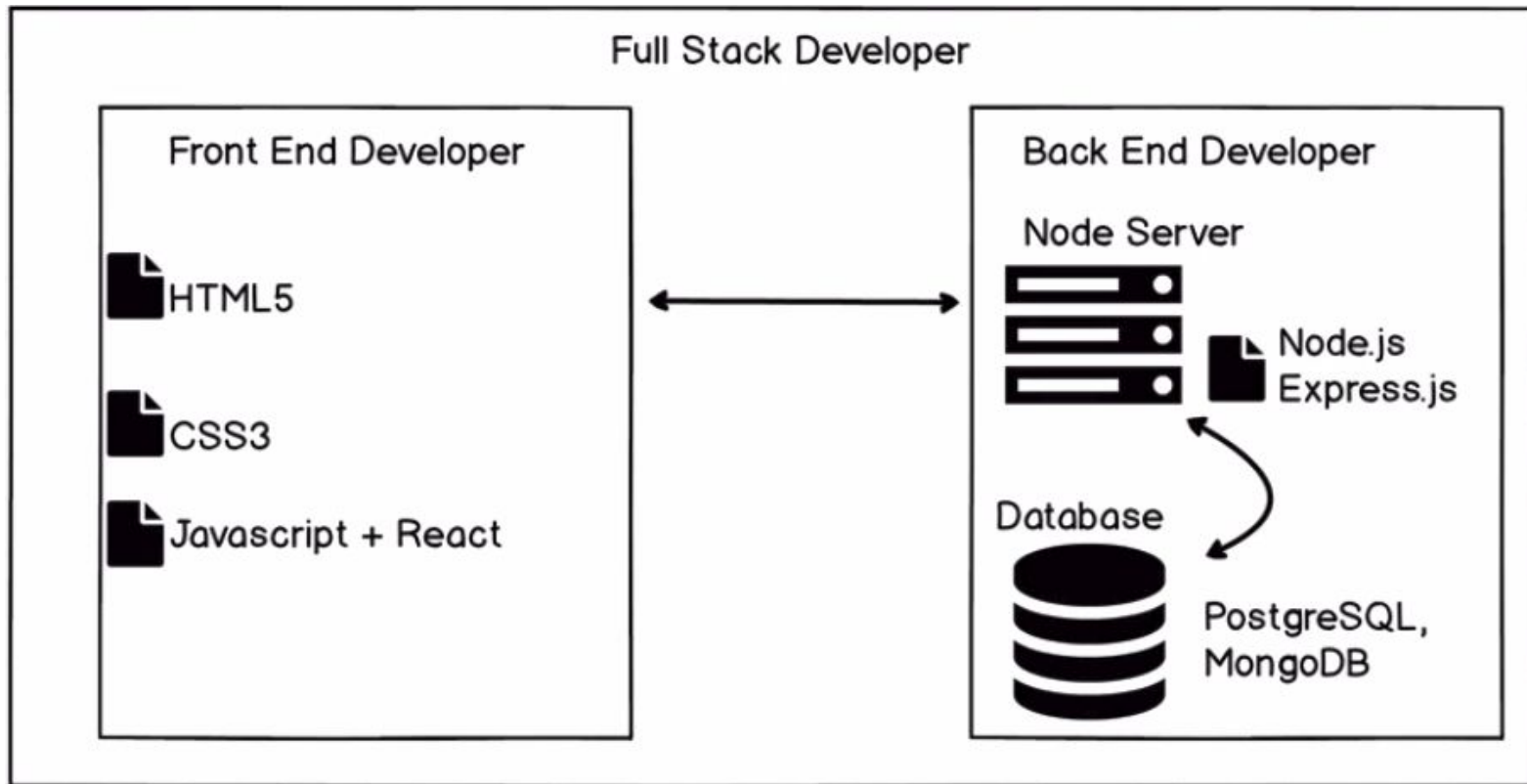
Javascript



Développement web



Développement web



Le Protocole HTTP

C'est quoi HTTP

- **H**yper **T**ext **T**ransfer **P**rotocol
- Assure la Communication entre un client et un serveur
- Essentiellement sous forme de Requête/Réponse
- Chaque requête est indépendante des précédentes (Stateless)
- Charger des pages, envoyer des formulaires ...

C'est quoi HTTPS

- **H**yper **T**ext **T**ransfer **P**rotocol **S**ecure
- Echange des données cryptées
- SSL (Secure Sockets Layer) / TLS (Transfer Layer Security)
- Il faut installer un certificat sur le serveur

HTTP: Méthodes

- GET: récupérer des données du serveur
- POST: envoyer des données au serveur
- PUT: mettre à jour des données sur le serveur
- DELETE: supprimer des données du serveur

HTTP : Header

► GET https://www.google.com/

Status **200 OK** (?)
Version HTTP/2
Transferred 43.05 KB (131.40 KB size)
Request Priority Highest

▼ Request Headers (804 B)

Raw ☐

(?) **Accept:** text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
(?) **Accept-Encoding:** gzip, deflate, br
(?) **Accept-Language:** en-US,en;q=0.5
(?) **Connection:** keep-alive
(?) **Cookie:** 1P_JAR=2022-10-20-12; AEC=AakniGMjy6zoHKAjDRmDfX5Hd3gctDWSL_1f43WA1t2m6n77swJzmmLFLA; NID=511=E7loqCuG-9dh6cq8dl8sLJ6JEIL3YCDBHlzfif_YDnKW84irDX9WZ6NFgwiiGWSqaMgzQR-wW2rEOICib3e13RaC3fN9HCmnE8Ru9j4c6A-YIZ9wl-Yxmbiu9M56hmTzdAvvrz2Y7a9aXfW18QXx6ds0iHgYK3xmcgtFjlE9on0; ANID=AHWqTUmGdJzSLn8w8l5l8929NGrdJf1ovJdpK7tnf9hDLUiiQJtkZ7B9FVT3vrjt
(?) **DNT:** 1
(?) **Host:** www.google.com
(?) **Sec-Fetch-Dest:** document
(?) **Sec-Fetch-Mode:** navigate
(?) **Sec-Fetch-Site:** none
(?) **Sec-Fetch-User:** ?1
(?) **TE:** trailers
(?) **Upgrade-Insecure-Requests:** 1
(?) **User-Agent:** Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:105.0) Gecko/20100101 Firefox/105.0

HTTP : Header

▼ Response Headers (612 B)		Raw <input type="checkbox"/>
	<code>alt-svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"</code>	
?	<code>cache-control: private, max-age=0</code>	
?	<code>content-encoding: br</code>	
?	<code>content-length: 40941</code>	
?	<code>content-type: text/html; charset=UTF-8</code>	
?	<code>date: Thu, 20 Oct 2022 12:37:55 GMT</code>	
?	<code>expires: -1</code>	
?	<code>server: gws</code>	
?	<code>set-cookie: 1P_JAR=2022-10-20-12; expires=Sat, 19-Nov-2022 12:37:55 GMT; path=/; domain=.google.com; Secure; SameSite=none</code>	
?	<code>strict-transport-security: max-age=31536000</code>	
	<code>X-Firefox-Spdy: h2</code>	
?	<code>x-frame-options: SAMEORIGIN</code>	
?	<code>x-xss-protection: 0</code>	

HTTP: Status Code

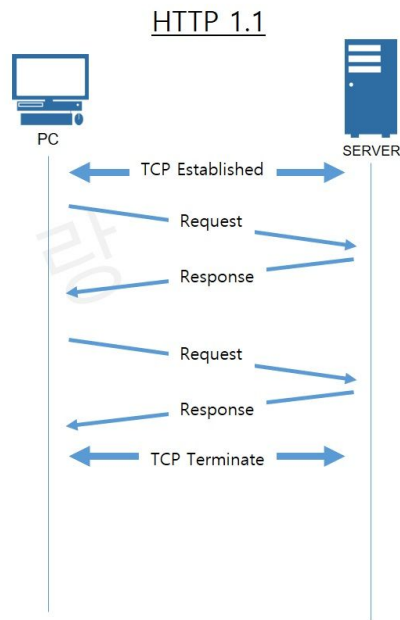
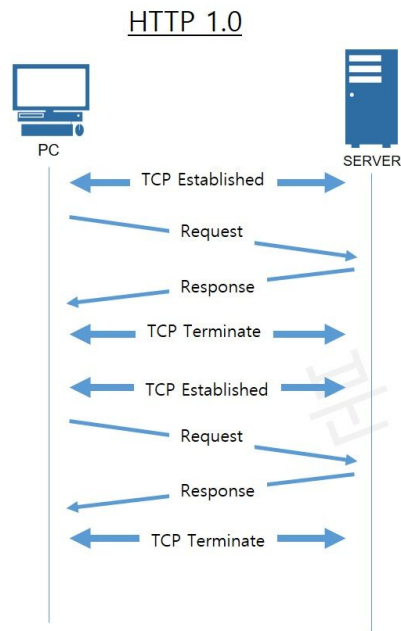
- 100 - 199 : Les réponses informatives,
- 200 - 299 : Les réponses de succès ,
- 300 - 399 : Les messages de redirection,
- 400 - 499 : Les erreurs du client,
- 500 - 599 : Les erreurs du serveur.

<https://developer.mozilla.org/fr/docs/Web/HTTP/Status>

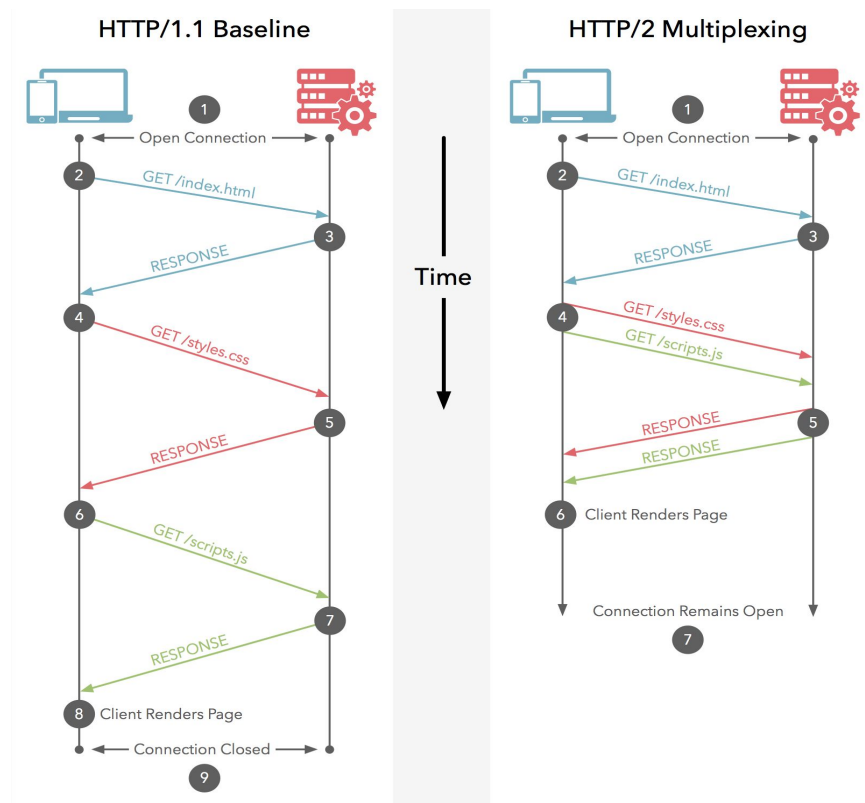
HTTP: Status Code

- 200 : OK
- 201 : OK Created
- 301 : Moved Permanently
- 304 : Not Modified
- 400 : Bad Request,
- 401 : Unauthorized
- 403 : Forbidden
- 404 : Not Found
- 405 : Method Not Allowed
- 500 : Internal Server Error
- 502 : Bad Gateway

HTTP 1.0 Vs HTTP 1.1



HTTP 1.1 Vs HTTP 2



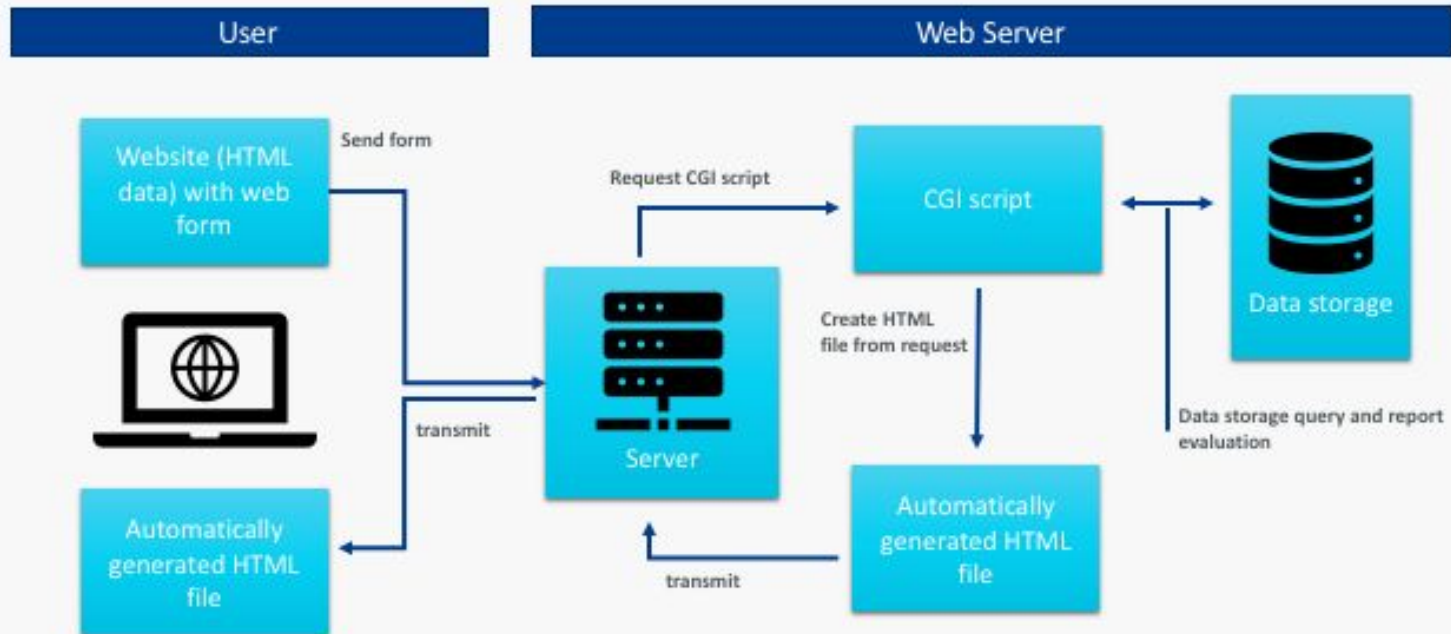
CGI

CGI, qu'est-ce que c'est ?

- CGI signifie Common Gateway Interface.
- Permet à un serveur web d'exécuter des script et envoyer la sortie aux clients.
- Ces script peuvent être écrits dans n'importe quel langage supporté par le serveur: BASH, Korn shell, Perl, Python, java, C !

CGI, qu'est-ce que c'est ?

Mode of operation of Common Gateway Interfaces



Premier script CGI

- Pour les exemples présentés dans le reste de cette partie il faut :
 - Installer un serveur web : apache (xamp)
 - Shell Bash sous une distro Linux et mac ou Linux Bash Shell sous windows
 - Éditeur de texte (vi, nano, vs code ..)
- Dans le dossier cgi-bin créer un fichier *hello* avec le contenu ci contre.
- `$ chmod +x hello`
- Dans le navigateur visitez : `http://localhost/cgi-bin/hello`

cgi-bin/hello

```
#!/bin/bash

echo "Content-type: text/html "
echo ""
echo '<html>'
echo '<head>'
echo '<meta'
http-equiv="Content-Type"
content="text/html; charset=UTF-8">'
echo '<title>Hello World</title>'
echo '</head>'
echo '<body>'
echo 'Hello World'
echo '<br>'
date
echo '</body>'
echo '</html>'

exit 0
```

Si le serveur ne reconnaît pas le chemin du script, ajouter la ligne: **`ScriptAlias "/cgi-bin/" "/chemin/vers/ledossier/cgi-bin/"`** au fichier httpd.conf

Les variables d'environnement

cgi-bin/echoenv

```
#!/bin/bash

echo "Content-type: text/html"
echo ""
cat << EOT
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
  EOT
env
echo "</body></html>"
exit 0
```

```
SERVER_SIGNATURE=
HTTP_SEC_FETCH_DEST=document
HTTP_SEC_FETCH_USER=?1
UNIQUE_ID=Y1HTTghoXpsLtp_kBhFQ0AAAAAQ
HTTP_USER_AGENT=Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:105.0) Gecko/
SERVER_PORT=80
HTTP_HOST=localhost
DOCUMENT_ROOT=/Applications/XAMPP/xamppfiles/htdocs
SCRIPT_FILENAME=/Applications/XAMPP/xamppfiles/cgi-bin/echoenv
REQUEST_URI=/cgi-bin/echoenv
SCRIPT_NAME=/cgi-bin/echoenv
HTTP_CONNECTION=keep-alive
REMOTE_PORT=56070
PATH=/usr/bin:/bin:/usr/sbin:/sbin
CONTEXT_PREFIX=/cgi-bin/
PWD=/Applications/XAMPP/xamppfiles/cgi-bin
SERVER_ADMIN=you@example.com
REQUEST_SCHEME=http
HTTP_ACCEPT_LANGUAGE=en-US,en;q=0.5
HTTP_ACCEPT=text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
REMOTE_ADDR=127.0.0.1
SHLVL=1
SERVER_NAME=localhost
SERVER_SOFTWARE=Apache/2.4.53 (Unix) OpenSSL/1.1.1o PHP/8.1.6 mod_perl/2.0.12
HTTP_SEC_FETCH_MODE=navigate
QUERY_STRING=
SERVER_ADDR=127.0.0.1
GATEWAY_INTERFACE=CGI/1.1
HTTP_UPGRADE_INSECURE_REQUESTS=1
SERVER_PROTOCOL=HTTP/1.1
HTTP_ACCEPT_ENCODING=gzip, deflate, br
HTTP_SEC_FETCH_SITE=none
REQUEST_METHOD=GET
HTTP_COOKIE=username=localhost-8888="2|1:0|10:1665008079|23:username=localhost
CONTEXT_DOCUMENT_ROOT=/Applications/XAMPP/xamppfiles/cgi-bin/
```

La Query String

```
#!/bin/bash
```

```
echo "Content-type: text/html "
```

```
echo ""
```

```
cat << EOT
```

```
<html>
```

```
  <head>
```

```
    <title>Hello World</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Hello World</h1>
```

```
EOT
```

```
oldIFS=$IFS
```

cgi-bin/qs

```
IFS='=&'
```

```
param=$QUERY_STRING
```

```
#echo "$param"
```

```
for i in $param
```

```
do
```

```
  echo "$i"
```

```
  echo "<br>"
```

```
done
```

```
IFS=$oldIFS
```

```
echo "</body>"
```

```
echo "</html>"
```

```
exit 0
```