

# Internet et Réseaux

## Le WEB

Département BAT  
Jean-Yves Tigli – [tigli@polytech.unice.fr](mailto:tigli@polytech.unice.fr)  
<http://rainbow.polytech.unice.fr/~tigli>

BAT 4 ième année



# Quelques Définitions

# Rappels sur le Jargon du Web

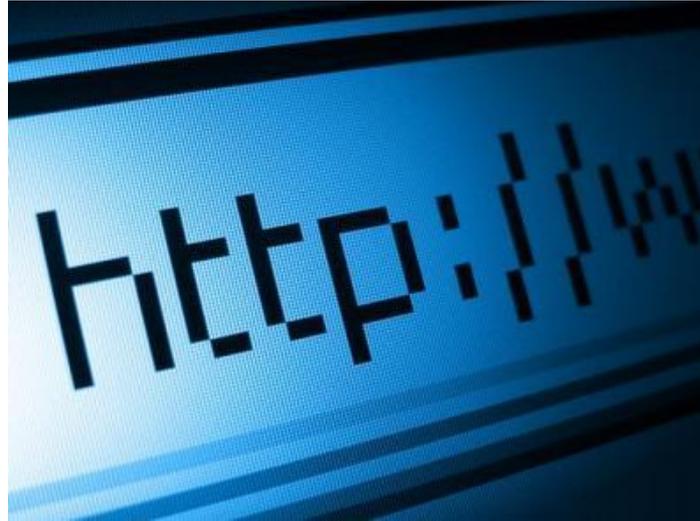
- ✓ **Page Web:**
  - Pointés par une URL
  - La plupart des pages WEB se composent de:
    - Une page HTML de base,
    - Différentes références à des « objets »
  
- ✓ **L'agent utilisateur pour le Web s'appelle un "browser" (butineur en français)**
  - Microsoft Internet Explorer, Mozilla FireFox, Opera, Safari, Google Chrome, ...
  
- ✓ **Un serveur pour le Web s'appelle un serveur Web :**
  - Apache, Microsoft Internet Information Server (IIS), ...

# URL: Uniform Resource Locator

- ✓ Une URL (Uniform Resource Locator) a au moins deux champs (protocole, adresse de la ressource)
  - Le protocole: schéma de représentation
  - L'Adresse : localisation complète de la ressource
  - Ex:
    - mailto:Quidam.no-spam@example.com
    - news:fr.comp.infosystemes.www.auteurs
- ✓ Une URL HTTP a au moins trois champs (protocole, adresse, emplacement) :
  - Le protocole: *http* suivi de :
  - L'adresse: le nom complet de la ressource // *login : password @ nom domaine : port*
  - Emplacement: Emplacement de la ressource à l'adresse
  - Données supplémentaires optionnelles transmises

Exemple :

<http://Jojo:IApIn@www.example.com:8888/chemin/d/acc%C3%A8s.php?q=req&q2=req2#signet>



# Le Protocole HTTP

HyperText Transfert Protocol

HTTP 1.0 : RFC 1945

HTTP 1.1 : RFC 2616

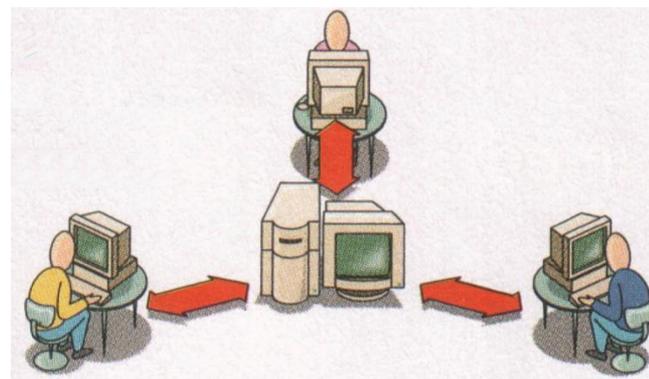
# HyperText Transfer Protocol

- ✓ ***HTTP : HyperText Transfert Protocole***
  - Un des protocoles les plus courants sur Internet
    - Un protocole omniprésent: de IT à Embedded
  - Il est utilisé pour la navigation sur les sites Web
    - protocole de rapatriement des documents
    - protocole de soumission de formulaires
  
- ✓ **Il en existe trois versions :**
  - 0.9 (1991) : complètement obsolète
  - 1.0 (février 1997), de nos jours très rarement utilisée
  - 1.1 (octobre 2000). Les principaux changements entre les v1.0 et v1.1 sont l'ajout de 2 types de requêtes ainsi que la possibilité d'héberger plusieurs sites Web sur un même serveur dans la version 1.1.

# Un Modèle Client - Serveur

## ✓ Modèle Client / Serveur

- client: « browser » qui demande, reçoit et affiche des documents Web.
- server: serveur Web qui envoie des documents en réponse aux requêtes des clients.



## ✓ L'échange entre le client et le serveur se fait en mode texte.

- Le charset généralement utilisé est l'US-ASCII sur 8 bits.
- Il est cependant possible que cet encodage soit modifié selon le client ou le serveur.

# Principe de Fonctionnement de HTTP

- ✓ **HTTP : TCP transport service**
  - Le client initialise une connexion TCP/IP (voir sockets) sur le serveur et le port 80.
  - Le serveur accepte la connexion du client et fournit un port de communication (utilisateur).
  - Les messages http (messages au protocole de l'application) sont échangés entre le client http et le serveur http.
  - Enfin, la connexion TCP/IP est fermée.
  
- ✓ **HTTP est "stateless"**
  - En principe, le serveur ne maintient pas d'information sur les requêtes passées du client.
  - En pratique, certaines techniques le permette
  - HTTP : Transport ou Session ?

# Zoom sur un Exemple de Communication

1. Le client http initialise une connexion TCP sur le serveur http *www.unice.fr*. (sur le port 80)
2. Le serveur http *www.unice.fr* en l'attente de connexions sur le port 80, accepte la demande de connexion du client
3. Le client http envoie un message de requête (contenant l'URL) au travers le socket de communication TCP.
4. Le serveur http reçoit le message de requête, compose le message de réponse contenant les objets demandés et renvoie le message au travers le socket de communication.
5. Le client http reçoit le message de réponse contenant le fichier HTML et l'affiche.
6. Le serveur http ferme la connexion.
7. En « parsant » le fichier HTML, le client http trouve 10 références à des objets jpeg. Les étapes 1 à 6 sont répétées pour chaque référence aux 10 objets jpeg.

# Illustration sous Google Chrome

- ✓ Menu : Paramètres / Outils / Outils de développement
- ✓ Permet de tracer l'exécution des requêtes pour la récupération d'une page Web.
  
- ✓ De nombreuses manipulations sont alors possibles
  - Avec ou sans prétraitement
- ✓ Exemple :
  - Parsing de page et cascade de requêtes
  - Avec ou sans cache et temps de réponse
  - Quelques champs intéressants :
    - Pour [www.polytech.unice.fr](http://www.polytech.unice.fr)
    - Pour [www.google.fr](http://www.google.fr)
  - Les cookies ? Où sont les cookies ?

# Tests avec Telnet

- ✓ **Telnet : un client TCP/IP Générique**
- ✓ **Permet d'établir une connexion TCP/IP avec n'importe quel serveur en mode caractère :**
- ✓ **telnet <serveur> <port>**
- ✓ **Exemple :**
  - telnet [www.polytech.unice.fr](http://www.polytech.unice.fr) 80
  - Saisi de GET /
  - Retour ...

# Dialogue HTTP

- ✓ **Dialogue**
  - en mode caractères ASCII (7 bits)
    - telnet www.sun.com 80
  
- ✓ **Types de dialogue**
  - Récupération d'un document
    - méthode GET
  - Soumission d'un formulaire
    - méthodes GET ou POST
  - Envoi de Document et Gestion de Site
    - méthodes PUT, DELETE, LINK, UNLINK
  - Gestion de proxy/cache
    - méthode HEAD (récupération des informations sur le document)

# Structure

- ✓ **Les requêtes et les réponses sont bâties sur le même modèle**

```
{Ligne d'introduction}{SEP}  
{En-têtes séparées par des {SEP}}  
{SEP}{SEP}  
{Corps}
```

- ✓ **Le seul élément capable de différencier une requête d'une réponse, c'est la *Ligne d'introduction*.**

# Format de la Requête

<Méthode> <URI> HTTP/<Version>

[<Champ d'entête>: <Valeur>]

[<tab><Suite Valeur si >1024>]

*ligne blanche*

[corps de la requête pour la méthode POST]

**GET /docu2.html HTTP/1.0**

Accept: www/source

Accept: text/html

Accept: image/gif

User-Agent: Lynx/2.2 libwww/2.14

From: alice@pays.merveilles.net

*\* une ligne blanche \**

**POST /script HTTP/1.0**

Accept: www/source

Accept: text/html

Accept: image/gif

User-Agent: Lynx/2.2 libwww/2.14

From: alice@pays.merveilles.net

Content-Length: 24

*\* une ligne blanche \**

name1=value1&

name2=value2

Source: Didier Donsez

# Méthodes de la Requête

- ✓ **GET**
  - demande pour obtenir des informations et une zone de données concernant l'URI
- ✓ **HEAD**
  - demande pour seulement obtenir des informations concernant l'URI
- ✓ **POST**
  - envoie de données (contenu du formulaire vers le serveur, requête SOAP ...). Ces données sont situées après l'entête et un saut de ligne
- ✓ **PUT**
  - enregistrement du corps de la requête à l'URI indiqué
- ✓ **DELETE**
  - suppression des données désignées par l'URI

# Méthodes de la Requête

- ✓ **OPTIONS**
  - demande des options de communication disponibles
- ✓ **TRACE**
  - retourne le corps de la requête intacte (débugage)
- ✓ **LINK / UNLINK**
  - association (et désassociations) des informations de l'entête au document sur le serveur
- ✓ **Nouvelles extensions de WebDAV**
  - PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK
- ✓ **Nouvelles extensions HTTP/U HTTP/MU**
  - NOTIFY, ... (UPnP)

# Champs d'Entête

- ✓ Ils permettent la transmission d'informations complémentaires sur la requête, et le client lui-même.
- ✓ Ces champs agissent comme "modificateurs" de la requête, utilisant une sémantique identique à celle des paramètres passés par un appel d'une méthode de langage de programmation de haut niveau.

# Format de la Réponse

HTTP/<Version> <Status> <Commentaire Status>

Content-Type: <Type MIME du contenu>

[< Champ d 'entête >: <Valeur>]

[<tab><Suite Valeur si >1024>]

*Ligne blanche*

Document

HTTP/1.0 200 OK

Date: Wed, 02Feb97 23:04:12 GMT

Server: NCSA/1.1

MIME-version: 1.0

Last-modified: Mon,15Nov96 23:33:16 GMT

Content-type: text/html

Content-length: 2345

**\* une ligne blanche \***

<HTML><HEAD><TITLE> ...

</BODY></HTML>

Source: Didier Donsez

# Statuts des Réponses HTTP (RFC2068)

- ✓ **1xx Information**
  - 100 : Continue (le client peut envoyer la suite de la requête), ...
- ✓ **2xx Succès de la requête client**
  - 200: OK, 201: Created, 204 : No Content, ...
- ✓ **3xx Redirection de la Requête client**
  - 301: Redirection, 302: Found, 304: Not Modified, 305 : Use Proxy,
- ✓ **4xx Requête client incomplète**
  - 400: Bad Request, 401: Unauthorized, 403: Forbidden, 404: Not Found
- ✓ **5xx Erreur Serveur**
  - 500: Server Error, 501: Not Implemented,
  - 502: Bad Gateway, 503: Out Of Resources (Service Unavailable)

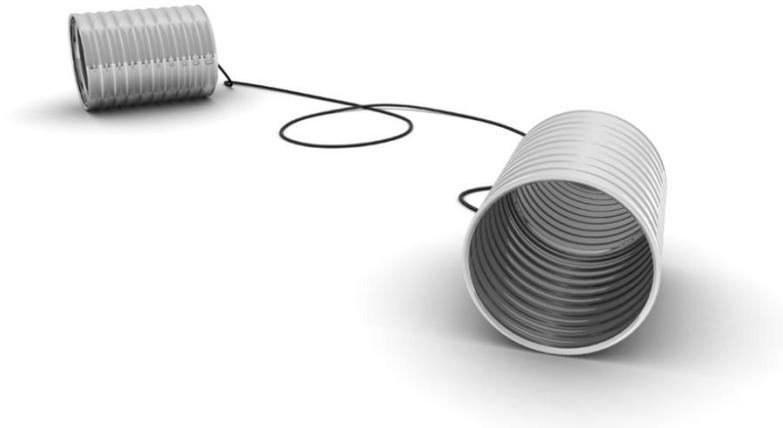
# Internationalisation

## ✓ Langage Accepté

- fr, de, it, en, sq (albanais), ru, (russe), ja (japonais), zh (chinois), el (grec), he (hébreu), ca (catalan) ...

## ✓ Charset (table de caractère)

- par défaut ISO-8859-1 (Latin-1)
  - ISO-8859-2 (hongrois, albanais, ... )
  - ISO-8859- 4
  - ISO-8859-5, KOI8-R (russe, bulgare, polonais)
  - ISO-8859-7 (grec)
  - ISO-8859-8 (hébreu)
  - ISO-8859-9 (turc)
  - Shift\_JIS, ISO-2022-JP, EUC-JP (japonais)
  - Big5 (chinois simplifié)
  - GB2312(chinois traditionnel - Taiwan)



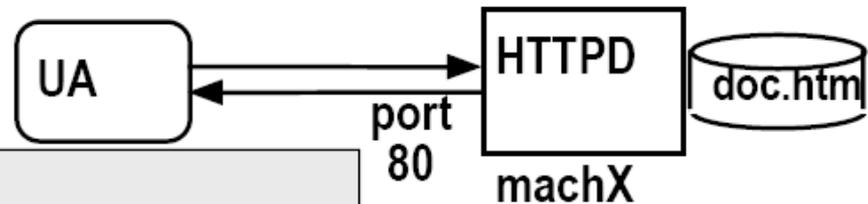
# Echange de Documents

Réception et Envoi de Données

# Récupération d'un Document Méthode GET

## ✓ GET /fichier

GET /doc.htm



le Client envoie

```
GET /doc.htm HTTP/1.0      méthode, chemin, version
Accept: www/source        documents acceptés
Accept: text/html
Accept: image/gif
User-Agent: Lynx/2.2 libwww/2.14
From: alice@pays.merveilles.net
* une ligne blanche *
```

le Serveur répond

```
HTTP/1.0 200 OK          ligne de status
Date: Wed, 02Feb97 23:04:12 GMT
Server: NCSA/1.1
MIME-version: 1.0
Last-modified: Mon, 15Nov96 23:33:16 GMT
Content-type: text/html   type du document retourné
Content-length: 2345      sa taille
* une ligne blanche *
<HTML><HEAD><TITLE> ...
```

# Récupération

## Méthode GET conditionnelle

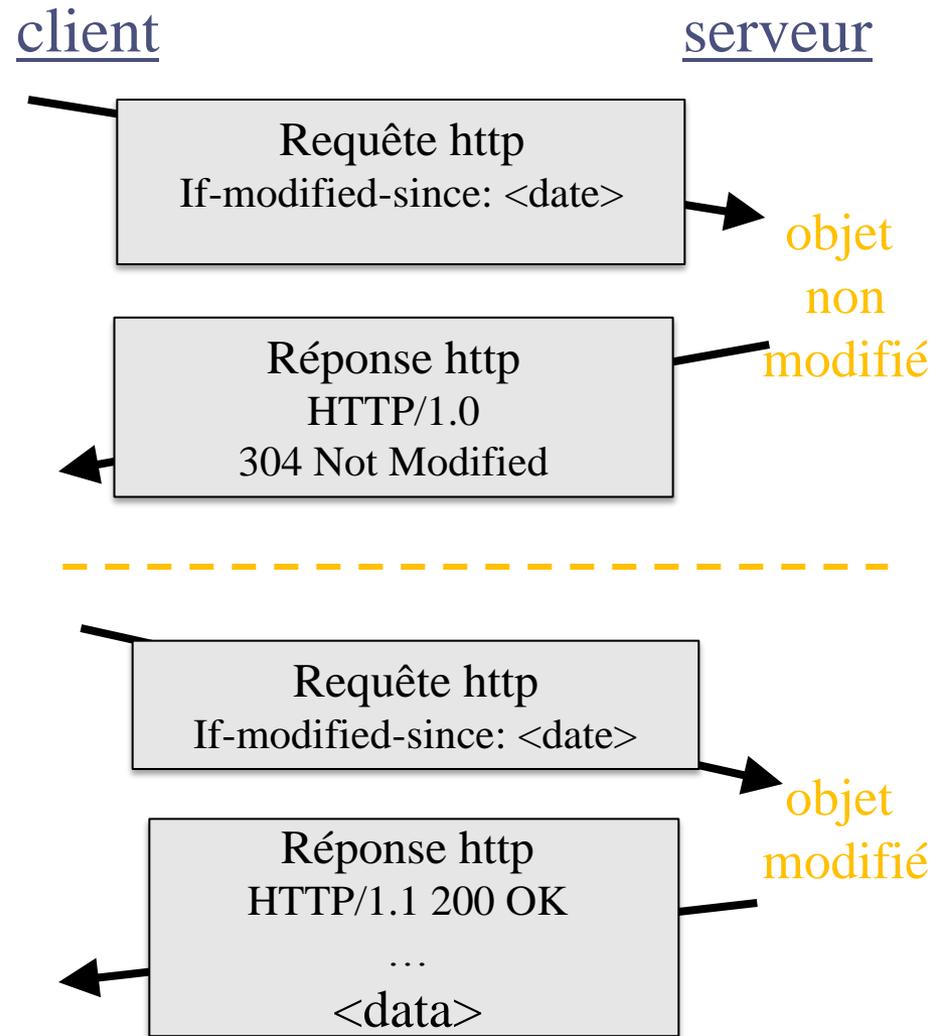
✓ **Objectif** : ne pas envoyer d'objet si le client a une version chargée à jour (en cache).

✓ **Client**: spécifie la date de la copie en cache dans la requête :

`If-modified-since: <date>`

✓ **Serveur**: la réponse ne contient pas de données si l'objet est à jour :

`HTTP/1.0 304 Not Modified`

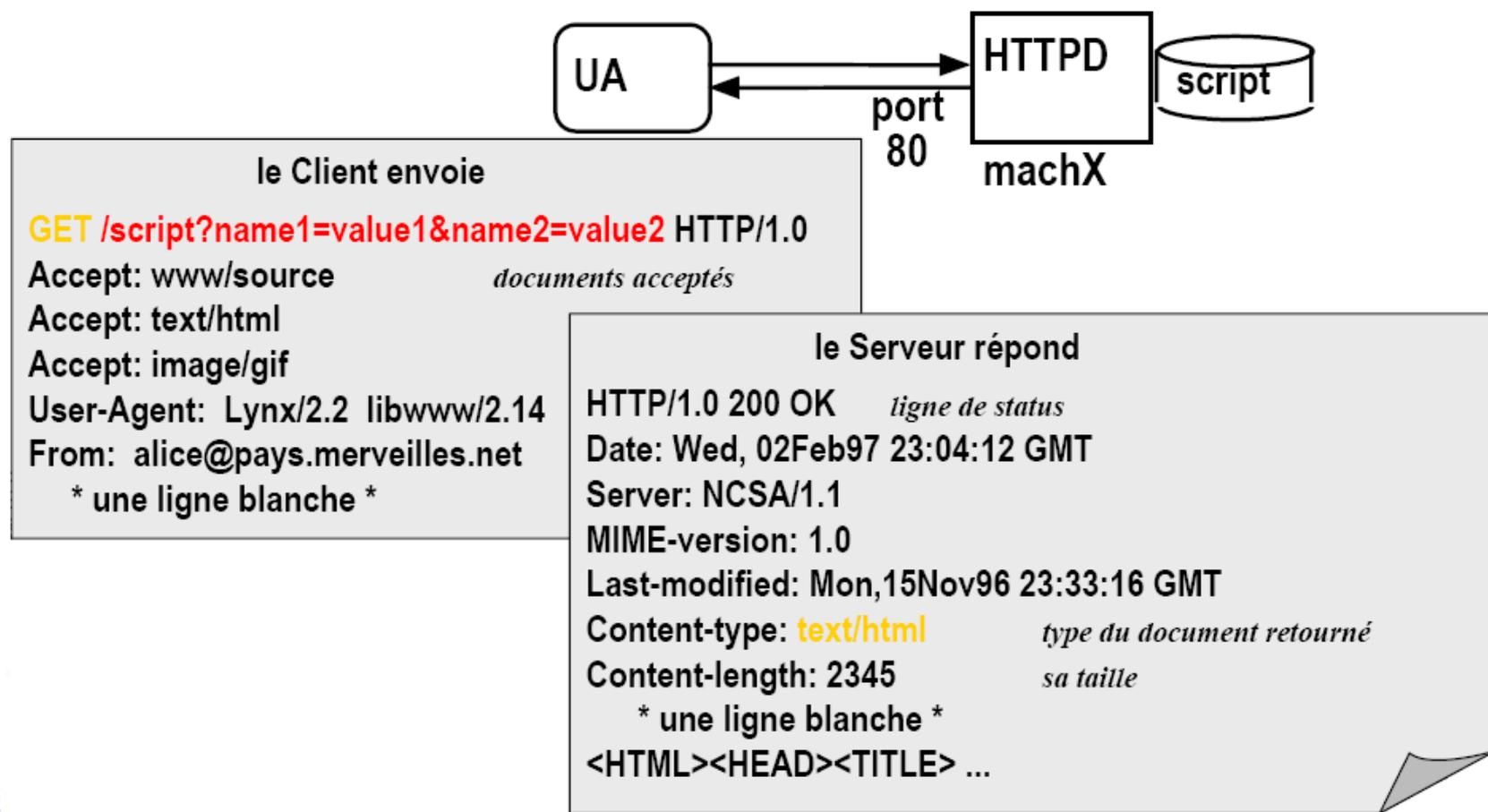


# Soumission d'un Formulaire

## Méthode GET

✓ GET/script?name1=value1&name2=value2

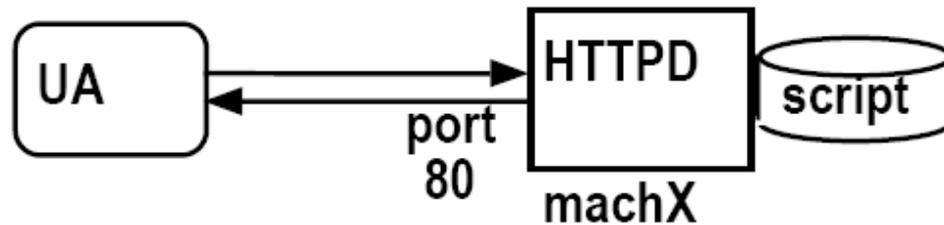
GET /script?name1=value1&name2=value2



# Soumission d'un Formulaire méthode POST

✓ POST /script

POST /script



le Client envoie

```
POST /script HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/gif
User-Agent: Lynx/2.2 libwww/2.14
From: alice@pays.merveilles.net
  * une ligne blanche *
name1=value1&
name2=value2
```

le Serveur répond

```
HTTP/1.0 200 OK
...
Content-length: 2345
  * une ligne blanche *
<HTML><HEAD><TITLE> ...
```

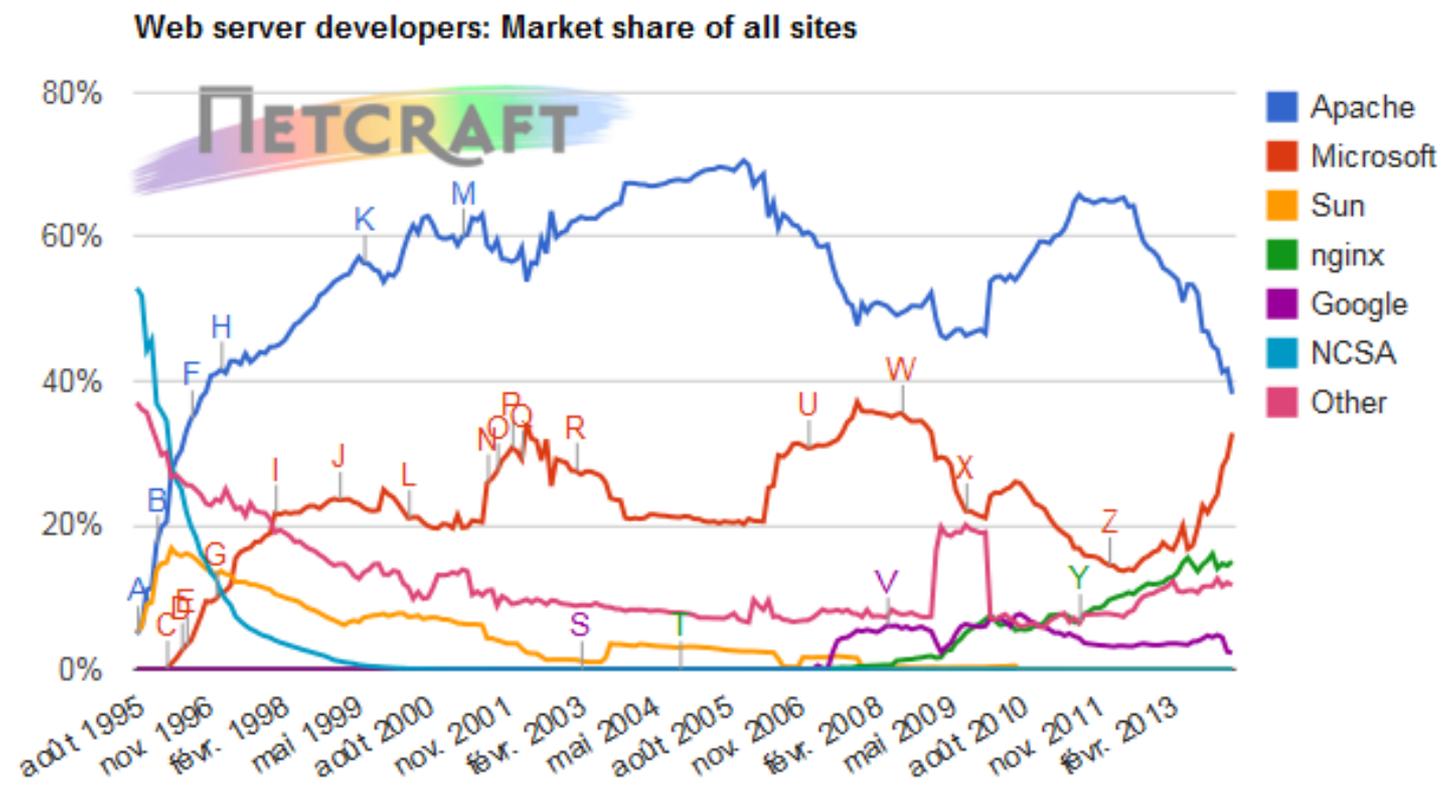


# Infrastructure Web

Des serveurs Web qui ne sont rien d'autres que des serveur socket TCP/IP échangeant des données suivant le protocole HTTP

# Répartition part de Marché Serveurs Web

✓ Apache ou Microsoft ?



Source: <http://news.netcraft.com/>

# Les Serveurs du Marché



[Apache Web Server](#)

[Apache Tomcat](#)



[Microsoft IIS Windows Web Server](#)

[Nginx web server](#)



[lighttpd web server](#)

[The Jigsaw web server software from W3C](#)



[Klone web server](#)



[Abyss web server](#)

[Oracle Web Tier](#)



[X5 \(formerly Xitami\) web server](#)

[Zeus Technology Ltd. - Zeus web server](#)



## ✓ A patch of NCSA HTTPD

- serveur le plus répandu (« toujours » la ,plus grosse part de marché)
- gratuit, issu du serveur NCSA HTTPD
- très nombreuses plates-formes Unix et Windows NT
- extensible par des modules tiers

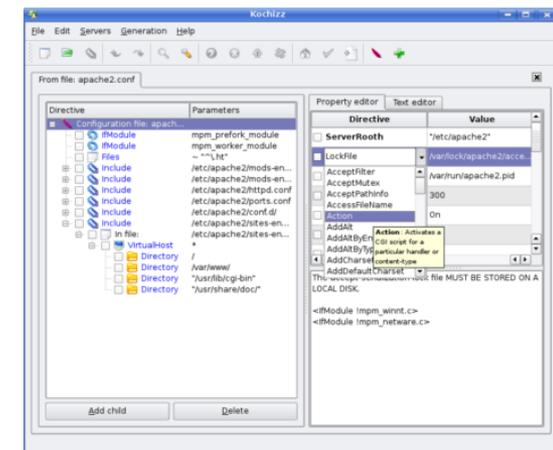
## ✓ Nombreux Modules Tiers

- possibilité d'étendre Apache avec des modules externe  
[http://www.zyzzzyva.com/server/module\\_registry](http://www.zyzzzyva.com/server/module_registry)
  - mod\_auth\_cookies\_file, mod\_auth\_cookies\_mysql, mod\_cgi\_sugid, mod\_perl, mod\_perl\_fast, mod\_auth\_kerb, mod\_auth\_dbi, mod\_rewrite, mod\_jserv(servlet), mod\_java (CGI écrit en Java), php3
- nombreux sous-projets autour de Java (Jakarta) et XML (Xerces, Xalan, XSP, Cocoon, ...)

# Configuration Apache

- ✓ Fichiers de configuration
  - httpd.conf
    - comportement de base port TCP/IP, journaux, keepalive, UID, virtualhost, proxy, ...
  - Les autres fichiers sont rajouté à l'aide de l'instruction *Include*
    - Exemples :
    - `Include /usr/local/apache2/conf/ssl.conf`
    - `Include /usr/local/apache2/conf/vhosts/*.conf`

Outil GUI : Kochizz, éditeur de configuration Apache.



...

## 2. Quelques Manipulations ....

### ✓ 1. Utilisation de Telnet pour contacter un serveur Web :

```
telnet www.unice.fr 80
```

Ouvre une connexion sur le port 80 (port par défaut) de www.unice.fr

Tout ce qui est tapé est maintenant transmis au serveur sur le port 80

### 2. Envoi d'une requête GET

```
GET /index.html HTTP/1.0
```

En tapant ceci, vous envoyez cette requête GET, minimale mais complète au serveur http (suivi de 2 « retour chariot »).

### 3. Récupération de la réponse du serveur Web