

<http://membres-liglab.imag.fr/donsez/cours>

# Le protocole HTTP

---

**Didier DONSEZ**

Université Joseph Fourier (Grenoble 1)

PolyTech'Grenoble – LIG/ADELE

`Didier.Donsez@imag.fr`

`Didier.Donsez@ieee.org`

# Au sommaire

---

- Historique
- Le protocole HTTP
- Les méthodes GET et POST
- Le suivi de session
- Clients et Serveurs HTTP
- Autour d 'HTTP



# Le Protocole HTTP

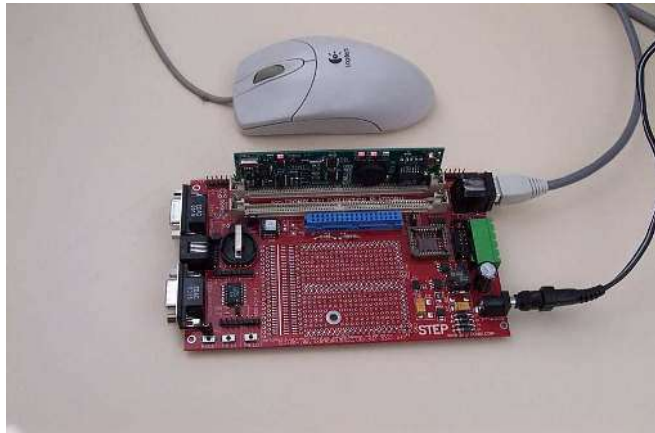
---

- HTTP : HyperText Tranfert Protocol (*RFC 1945 et 2068*)
  - protocole de rapatriement des documents
  - protocole de soumission de formulaires
- Fonctionnement (*très simple en HTTP/1.0*)
  - connexion
  - demande (GET) d'un document
  - renvoi du document (status=200) ou d'une erreur
  - déconnexion
- *Cependant*
  - *dialogue plus complexe en cas d'identification*
  - *optimisation :*
    - une série de plusieurs requêtes sur une connexion*
    - Connexion « KeepAlive » de HTTP/1.1 (*RFC 2068*)

# Un protocole omniscient

- IT Servers
- Embedded Servers

Didier Donsez, 1995-2010, HTTP



# Dialogue HTTP

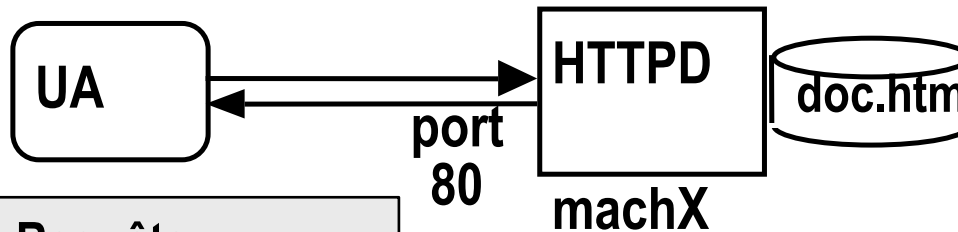
---

- Protocole en mode de lignes de caractère
  - le plus basic de butineur (User Agent) HTTP :
    - 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)*

# Dialogue HTTP

## exemple de requête HTTP (méthode POST)

**POST /script**



le Client envoie la Requête

**POST /script HTTP/1.0** *méthode, chemin, version*  
**Accept: www/source** *documents accetés*  
**Accept: text/html**  
**Accept: image/gif**  
**Accept: image/jpeg**  
**User-Agent: Lynx/2.2 libwww/2.14**

\* une ligne blanche \*

**name1=value1&  
name2=value2**

Corps de la requête

le Serveur retourne la Réponse

**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> ...**

Entête de la réponse

Corps de la réponse

# Format de la requête

- Envoyé par le client au serveur

<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

# Méthodes de la requête (i)

---

- 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 ...) situées dans le corps.
- PUT
  - enregistrement du corps de la requête à l'URI indiqué
- DELETE
  - suppression des données désignées par l'URI



## Méthodes de la requête (ii)

- OPTIONS
  - demande des options de communication disponibles
- TRACE
  - retourne le corps de la requête intacte (débugage)
- CONNECT
- + **Nouvelles extensions de WebDAV**
  - PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK
- + **Nouvelles extensions de HTTP/U et HTTP/MU**
  - NOTIFY, ... (UPnP)



# Format de la réponse

- Réponse envoyé par le serveur au client

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>
```

# Status des réponses HTTP (RFC2068)

- Code de réponse donné par le serveur au client
  - 100-199      Informationnel
    - 100 : Continue (le client peut envoyer la suite de la requête), ...
  - 200-299      Succès de la requête client
    - 200: OK, 201: Created, 204 : No Content, ...
  - 300-399      Redirection de la Requête client
    - 301: Redirection, 302: Found, 304: Not Modified, 305 : Use Proxy, ...
  - 400-499      Requête client incomplète
    - 400: Bad Request, 401: Unauthorized, 403: Forbidden, 404: Not Found
  - 500-599      Erreur Serveur
    - 500: Server Error, 501: Not Implemented,
    - 502: Bad Gateway, 503: Out Of Resources (Service Unavailable)

# Entêtes HTTP

---

- 4 types de champs d'entête
  - Général
    - commun au serveur, au client ou à HTTP
  - Requête du client
    - formats de documents et paramètres pour le serveur
  - Réponse du serveur
    - information concernant le serveur
  - Entité
    - informations concernant les données échangés

# Entêtes Généraux HTTP

---

- **Cache-Control** = contrôle du caching.
- Connection = listes d'option
  - close pour terminer une connexion.
- Date = date actuelle (format RFC1123 mais aussi RFC850).
- MIME-Version = version MIME utilisé.
- Pragma = instruction pour le proxy.
- Transfer-Encoding = type de la transformation appliquée au corps du message.
- Upgrade = indique le protocole souhaité.
- Via = utilisé par les proxys pour indiquer les machines et protocoles intermédiaires.

# Entêtes de requêtes client HTTP

- **Accept** = type MIME visualisable par l'agent
- **Accept-Language** : liste de langues (fr, en, ... )
- **Authorization** = type d'autorisation
  - BASIC nom:mot de passe (en base64) donc transmis en clair !!!
  - NB : Préalablement le serveur a répondu un WWW-Authenticate
- **Cookie** = cookie retourné
- Host = spécifie la machine et le port du serveur
  - un serveur peut héberger plusieurs serveurs
- Referer = URL d'origine
  - page à contenant l'ancre à partir de laquelle le visualisateur a trouvé l'URL.
- **User-Agent** = modèle du visualisateur

# Entêtes de requêtes client HTTP

- From = adresse email de l'utilisateur
  - rarement envoyé pour conserver l'anonymat de l'utilisateur
- If-Modified-Since = condition de retrait
  - la page n'est transférée que si elle a été modifiée depuis la date précisée. Utilisé par les caches
  - indique si le document demandé peut être caché ou pas.
- If-Unmodified-Since = condition de retrait
- If-Match = condition de retrait
- If-None-Match = condition de retrait
- If-Range = condition de retrait

# Entêtes de requêtes client HTTP

---

- Max-Forwards = nombre max de proxy
- Proxy-Authorization = identification
- **Range** = zone du document à renvoyer
  - bytes=x-y (x=0 correspond au premier octet, y peut être omis pour spécifier jusqu'à la fin)



# Entêtes des réponses serveur HTTP

- Accept-Range = accepte ou refus d'une requête par intervalle
- Age = ancienneté du document en secondes
- Proxy-Authenticate = système d'authentification du proxy
- Public = liste de méthodes non standards gérées par le serveur
- Retry-After = date ou nombre de secondes pour un ressay en cas de code 503 (service unavailable)
- Server = modèle de HTTPD
  - utilisé par Satan !!!!
- **Set-Cookie** = créer ou modifie un cookie sur le client
- Vary = l'entité possède plusieurs sources
- Warning = informations supplémentaire du code d'état
  - 14 Transformation applied : le proxy a changé de Content-Type ou le Content-Encoding
- **WWW-Authenticate** = système d'authentification pour l'URI

# Entêtes d'entité HTTP

---

- Allow = méthodes autorisées pour l'URI
- Content-Base = URI de base
  - pour la résolution des URL
- **Last-Modified** = date de dernière modification du doc.
  - Utilisé par les caches
- **Content-Length** = taille du document en octet
  - utilisé par le client pour gauger la progression des chargements
- Content-Encoding = type encodage du document renvoyé
  - compress, x-gzip, x-zip
- Content-Language : le langage du document retourné
  - fr, en ...
- Content-Location : URI de l'entité
  - quand l'URI est à plusieurs endroits

# Entêtes d'entité HTTP

- Content-MD5 : résumé MD5 de l'entité
- Content-Range : position du corps partiel dans l'entité
  - bytes x-y/taille
- Content-Transfer-Encoding : transformation appliqué du corps de l'entité
  - 7bit, binary, base64, quoted-printable
- **Content-Type** = type MIME du document renvoyé
  - utilisé par le client pour sélectionner le visualisateur (plugin)
  - RFC2045
- ETag : transformation appliqué du corps de l'entité
  - 7bit, binary, base64, quoted-printable
- **Expires** : date de péremption de l'entité
- **Refresh** : 10
- **Refresh**: 10; Location=/newloc.htm

# Internationalisation

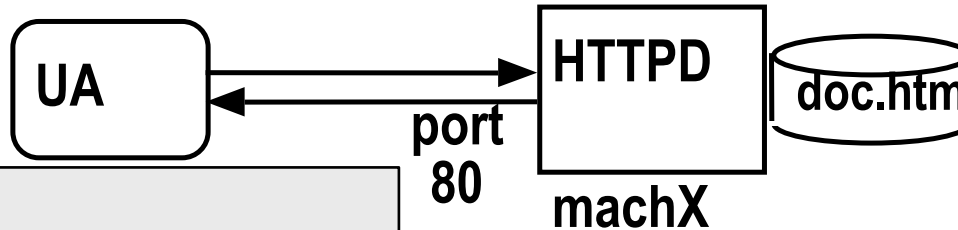
---

- Language Accept
  - 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)

# Récupération d'un Document

## Méthode GET

**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 *
```

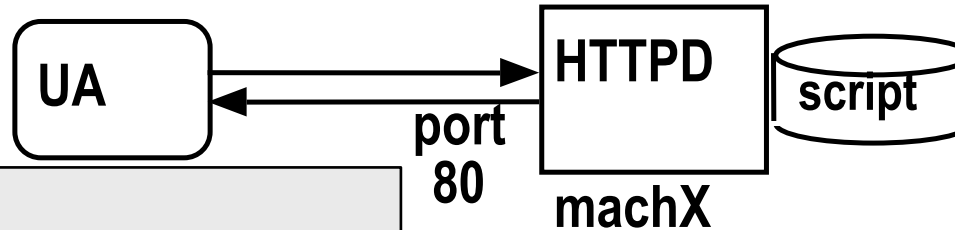
le Serveur répond

```
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> ...
```

# Soumission d'un Formulaire

## Méthode GET

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



le Client envoie

**GET** /script?name1=value1&name2=value2 HTTP/1.0

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*

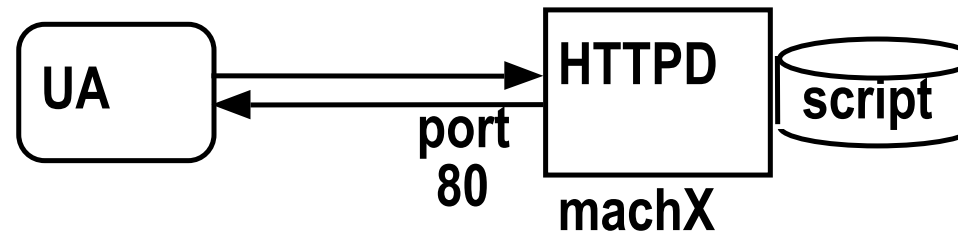
\* une ligne blanche \*

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

# Soumission d'un Formulaire

## Méthode POST

**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> ...
```

# Codage des « paramètres »

- Les valeurs passées (URL et contenu des entrées des formulaires) doivent être sur 7 bits et sans caractères spéciaux
- Format d'encodage : **x-www-form-urlencoded**
  - Espace → +
  - Tous les caractères spéciaux et accentués → %code  
**ascii**
    - @ %40
    - é %e9
  - Les entrées des formulaires sont encodés dans une chaîne composée de paires (nom de l'entrée)=(valeur de l'entrée) séparé par de &
    - **nom=Dupont+Jean&adresse=3+rue+de+la+Gait%e9%0a75014+Paris**
  - il existe de nombreuses bibliothèques d'encodage/décodage
    - dans le JDK :
      - static String java.net.URLDecoder.decode(String urlencoded, String encodingscheme)
      - static String java.net.URLEncoder.encode(String str , String encodingscheme)



# Comportement du User Agent face au type du document retourné

- A partir du type MIME de Content-Type
  - Visualisation native
    - la fonction de visualisation est dans le noyau (core) du client
      - `text/html, image/jpeg`
  - Visualisation par plugin
    - la fonction est présente dans un DLL, SO, ou un JAR
    - elle est liée dynamiquement pour réaliser la visualisation
      - `world/vrml, text/tex`
  - Visualisation externe
    - la fonction n'est pas présente dans le client
    - le client rapporte le document et le sauvegarde dans un fichier temporaire
      - `video/mpeg, application/postscript`

# Requête Multi-parties (multipart)

## ■ Motivation

- Requête multi-document [RFC1867]
  - formulaire HTML contenant des Upload de fichiers
    - `<FORM ACTION="/servlet/UploadTest" ENCTYPE="multipart/form-data" METHOD=POST>`
    - `Your name? <INPUT TYPE=TEXT NAME=submitter> <BR>`
    - `Your first file to upload? <INPUT TYPE=FILE NAME=file1> <BR>`
    - `Your second file to upload? <INPUT TYPE=FILE NAME=file2> <BR>`
    - `<INPUT TYPE=SUBMIT>`
    - `</FORM>`
  - Remarque : Mail multi-documents
    - (fichiers attachés, mail enrichi d'images, audio-mail ...)

# Requête Multi-parties (multipart)

## *Codage de la requête*

Une chaîne tirée aléatoirement sert de séparateur

```
Content-Type : multipart/form;boundary=End9989822
```

```
--End9989822
```

```
Content-Disposition; form-data; name="file1"; filename="test.htm"
```

```
Content-Type : text/html
```

```
<HTML><BODY> Ceci est un fichier de test !</BODY></HTML>
```

```
--End9989822
```

```
Content-Disposition; form-data; name="file2"; filename="test2.txt"
```

```
Content-Type : text/plain
```

```
Ceci est un deuxieme fichier de test !
```

```
--End9989822
```

- Classes de manipulation (non standard)
  - voir [Hunter Ex4-18 p 107]

# Réponse Multi-parties

---

## ■ Codage

- Content-Type : multipart/x-mixed-replace;
- Frontière entre les parties
  - Déclaration : `boundary=chaîne_aléatoire`
  - Séparateur : `--chaîne_aléatoire`

## ■ Comportement

- le navigateur affiche le sous-document suivant dès qu'il commence à le recevoir après avoir effacer la fenêtre.
  - Voir [Hunter ex6-12 p193]

# Réponse Multi-parties

## Exemple

Content-Type : multipart/x-mixed-replace;boundary=End65577565679001838

*ligne blanche*

--End65577565679001838

Content-Type : text/html

*ligne blanche*

<HTML><BODY><H1>Trois ... </H1><BODY></HTML>

--End65577565679001838

Content-Type : text/html

*ligne blanche*

<HTML><BODY><H1>Deux ... </H1><BODY></HTML>

--End65577565679001838

Content-Type : text/html

*ligne blanche*

<HTML><BODY><H1>Un ... </H1><BODY></HTML>

--End65577565679001838

Content-Type : text/html

*ligne blanche*

<HTML><BODY><H1>Partez ! </H1><BODY></HTML>

--End65577565679001838

Le serveur définit une chaîne séparateur des documents

Le serveur attend 1 seconde avant de renvoyer la suite : le client affiche « Trois... »

Le serveur attend 1 seconde avant de renvoyer la suite : le client affiche « Deux... »

Le serveur attend 1 seconde avant de renvoyer la suite : le client affiche « Un... »

Le serveur clôt la connexion TCP/IP avant de renvoyer la suite : le client affiche finalement « Partez ! »

# Suivi de Sessions avec HTTP

## *(Session Tracking)*

---

- Motivations :
  - La notion de session est importante dans une application conversationnelle
    - commerce électronique
      - « j 'ajoute ce produit à mon panier (existant)»
  - Cependant HTTP est un protocole « stateless »
    - le serveur ne maintient pas d 'informations liées aux requêtes précédentes d 'un même client.
    - HTTP est donc « sessionless »
  - Comment implanter la notion de session sur plusieurs requêtes HTTP
    - documents, CGI, SSS, Servlet/JSP, ASP, PHP, ...

# Suivi de Sessions avec HTTP

## *(Session Tracking)*

---

### ■ Méthodes

- Le serveur génère un identificateur de session et associe un état (et une date limite de validité) à une session
- Le client renvoie l'identificateur de session à chaque requête HTTP vers le serveur

### ■ Echange et Stockage de l'identificateur de session

- Input HIDDEN dans les formulaires
- Réécriture des URLs (EXTRA\_PATH)
- Cookies (déactivable)
- Identificateur de session SSL (Secure Socket Layer)

# Suivi de session avec HTTP

---

- Une session s'étend sur plusieurs requêtes
  - documents, CGI, SSS, Servlet, ASP
  - le serveur maintient un contexte de session et y associe un identifiant de session
- 3 solutions de suivi
  - input HIDDEN
    - contient l'identifiant de la session
  - la Ré-écriture d'URL
    - l'identifiant dans chaque URL (dans les documents)
  - les Cookies
    - information positionnée par le serveur sur le client
      - la durée de vie du cookie dépasse la session
    - puis envoyé par le client à chaque requête
- Implantation
  - objet Session (ASP), classe HttpSession (JSP/Servlet)



# Suivi de Session

## une entrée HIDDEN par formulaire (i)

- Chaque réponse retournée par le serveur est un formulaire qui contient un identifiant caché dans une entrée HIDDEN
- Exemple

### • *page de proposition*

```
<FORM METHOD="POST" ACTION="/cgi-bin/command">
  <INPUT TYPE="checkbox" NAME="art12387">
  Chaussures
  .
  .
  .
</FORM>
```

### • *réponse de /cgi-bin/command*

```
<FORM METHOD="POST" ACTION="/cgi-bin/envoi">
  <INPUT TYPE="hidden" NAME="TransID"
  VALUE="54109848932">
  Nom: <INPUT TYPE="text"
  NAME="nom">
  Adresse: <INPUT
  TYPE="text" NAME="adresse">
  N° de Carte de Credit: <INPUT
  TYPE="text" NAME="numcarte">
  .
  .
  .
  <INPUT TYPE="hidden" NAME="Language"
  VALUE="French">
</FORM>
```

# Suivi de Session

## *une entrée HIDDEN par formulaire (ii)*

- Inconvénients
  - dialogue uniquement par formulaire
    - car pas de persistance de l'identifiant côté client
  - Ambiguïté dans le cas des retours-arrière de l'utilisateur
    - annulation d'une série d'actions
    - ou série d'action supplémentaire

# Suivi de Session

## *la ré-écriture des URLs*

- L 'identifiant de session est encodé dans les URLs des documents HTML retourné par le serveur.

`http://localhost:8080/servlet/cart?PROD_ID=383`

devient

`http://localhost:8080/servlet/cart;jsessionid=To1128mC33718557521577075At?PROD_ID=383`

- Codage différent d'un serveur à l'autre.

- `field_sessionID = "JServSessionIdroot";` //Apache/Jserv
- `field_sessionID = ";jsessionid=";` //Jakarta-Tomcat
- `field_sessionID = "JIGSAW-SESSION-ID";` //Jigsaw web servers
- `field_sessionID = ";$sessionid$";` // WebSphere

- Limites

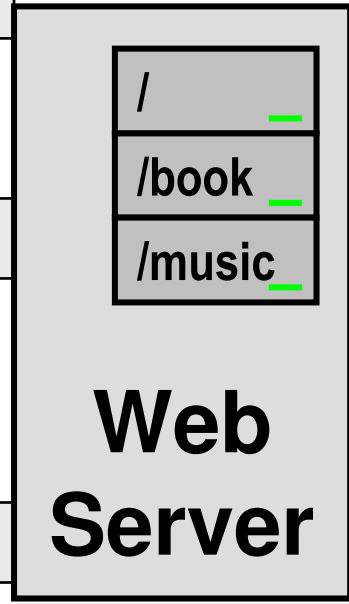
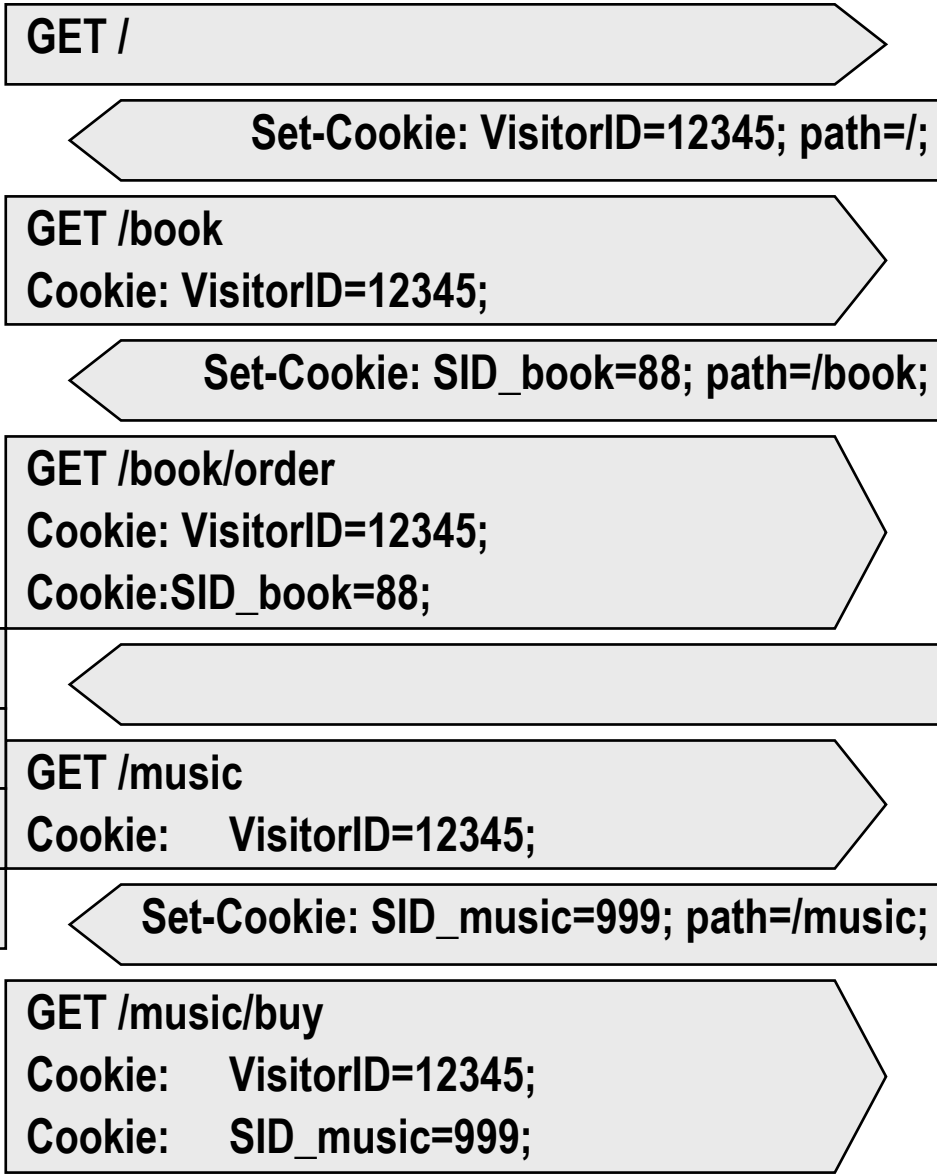
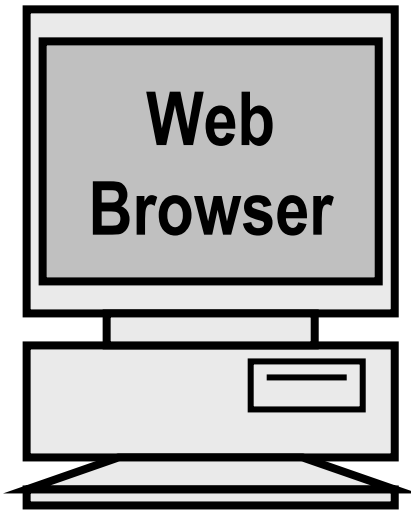
- URL générée par un script (=>programmation)
- ou parsing des documents HTML retournés
  - mais disfonctionnement en présence de scripts JavaScript ou VBScript générant eux aussi des URL !

# Suivi de Session

## *les Cookies [Netscape puis RFC2109]*

- chaîne décrivant l'état d'une session
  - NAME=VALUE;
  - expires=DATE;
  - path=PATH\_HEAD;                    / << /foo << /foobar ou /foo/bar.html
  - domain=DOMAIN\_TAIL;   fr << mycomp.fr << sales.mycomp.fr
- stocké sur le client
  - Limite
    - 300 cookies simultanées par client, 20 cookies par serveur ou domaine, 4Ko par cookie (limite la taille des VALUES)
- communiqué dans les entêtes de requêtes (Cookie:) et dans les entêtes des réponses HTTP (Set-Cookie:)
- accessible par les scripts JavaScript dans une page HTML

# Positionnement des Cookies (i)



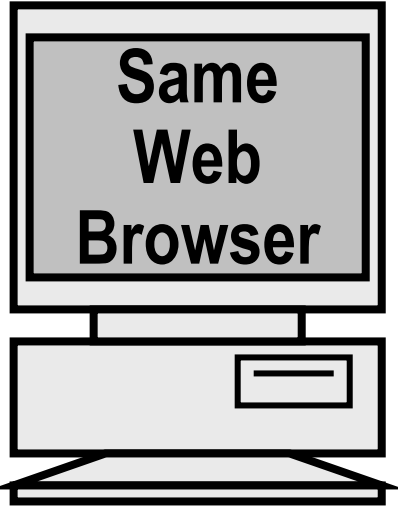
*Web browser' cookies table*

VisitorID=12345; path=/;
SID_book=88; path=/book;
SID_music=999; path=/music;

Didier Donsez, 1995-2010, HTTP

# Positionnement des Cookies (ii)

*Un autre jour !*



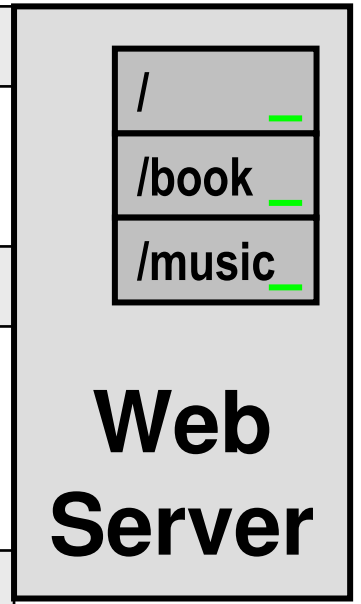
GET /  
Cookie: VisitorID=12345;

GET /book  
Cookie: VisitorID=12345;

Set-Cookie: SID\_book=101; path=/book;

GET /book/logout  
Cookie: VisitorID=12345;  
Cookie:SID\_book=101;

Set-Cookie: SID\_book=0; path=/book; \ expires=0



*Web browser' cookies table*

VisitorID=12345; path=/;

GET /  
Cookie: VisitorID=12345;

# CC/PP *(Composite Capabilities/Preference Profiles)*

- Motivation
  - Capacité du UA mobile et préférence usager envoyé au serveur en vue d'adapter la réponse
- Descriptions RDF
  - Terminal
    - Display size
    - CPU
    - Operating system
    - Supported HTML Version
  - Usager
- Compatible WAP UAPROF
- Remarque:
  - Devenu un nouveau WG « Device Independence »

# HTTPS (*S comme Secure*)

- Motivation
  - Sécuriser (authentification, confidentialité) l'accès à un service Web
- SSL/TLS
  - Phase 1 : Authentification du serveur et/ou du client par PKI
  - Phase 2: Chiffrage avec une clé (secrète) symétrique de session
  - Phase 2bis : Reprise après déconnexion
- HTTP over SSL (TLS)
  - URL : `https://host/document` (Port TCP par défaut : 443)
- HTTPS dans le JDK
  - JSSE (`javax.net.ssl.*`) inclus dans J2SE 1.4
  - classe `javax.net.HttpsURLConnection`
  - Voir cours JCE (<http://www-adele.imag.fr/users/Didier.Donsez/cours/jce.pdf>)
- Serveurs
  - Apache/SSL, iPlanet, MS IIS, OracleAS, IBM WebSphere ...
  - Carte hardware accélératrice (*nFast de nCipher*)



# HTTP-R (*R* comme « *Reliable* » *Fiable*)

Under Construction  
En Construction

- Proposé par IBM au W3C, mais toujours en cours , (
- Rappel
  - HTTP est une couche de transport pour les messages SOAP dans la mise en œuvre les Web Services (B2B, EAI)
- Motivation
  - Fiabiliser l'échange de messages entre le client et le service Web
    - Échanger une et une seul fois même en cas d'interruption (déconnexion, ...)
  - Batch de plusieurs messages (requêtes et réponses) par commande
- Principe d'HTTPR
  - Journaliser les messages (requêtes et réponses) côté client et côté serveur
  - Notion de transaction
  - Valider/Acquiter les échanges avec un protocole de validation
  - Ajout de nouvelles commandes

# Nouvelles « commandes » de HTTP-R

Ces commandes sont portées par la commande HTTP PC

**Under Construction**  
**En Construction**

- **PUSH**
  - Le client envoie un ou plusieurs messages qui seront journalisées par le serveur
  - La séquence de messages est associée à un identifiant de transaction
  - Peut contenir des acquittements à des commandes précédentes
- **PULL**
  - Demande à recevoir des messages en attente coté serveur
  - Peut contenir des acquittements à des commandes précédentes
- **EXCHANGE**
  - Combine PUSH et PULL
- **RESOLVE**
  - Permet de déterminer les messages recus et journalisés par le serveur
- **REPORT**
  - Permet de rapporter au serveur les messages recus et journalisés par le client
- **GET RESPONDER INFO**
  - Information pour le serveur sur le niveau de protocole supporté par le client

# HTTPTU & HTTPMU

---



- HTTPTU (HTTP Unicast over UDP)
- HTTPMU (HTTP Multicast over UDP)
  
- Utilisé par UPnP
  - <http://www.upnp.org/about/default.asp>
  - pour le protocole de découverte de service (SSDS (Simple Service Discovery Protocol))

# XML-RPC vs SOAP vs REST

---

- Motivations
  - Program-to-Program
  - B2B, AJAX, ...
- XML-RPC
  - HTTP POST
  - Simple encoding schema
- SOAP
  - operation-centric
  - WSDL
- REST
  - more URL-centric

# REST (Representational State Transfer)

---

- *“Amazon has both SOAP and REST interfaces to their web services, and 85% of their usage is of the REST interface.”*
- *Motivation*
  - *L’encodage d’une requete SOAP est complexe pour le néophyte*
  - *Utilisation du passage de paramètre de HTTP GET*
  
- Roy Thomas Fielding. Architectural styles and the design of network-based software architectures. PhD thesis, University of California, Irvine, 2000.
  - <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

# Les clients (UA) et les serveurs HTTP

---

Didier DONSEZ

Université Joseph Fourier (Grenoble 1)

PolyTech'Grenoble – LIG/ADELE

`Didier.Donsez@imag.fr` `Didier.Donsez@ieee.org`

# Les Client (*User-Agents*) HTTP

- Butineur ou Arpenteur ou Brouteur Web (*browser Web*)
  - Récupère un document puis « visualise » ce document
- Sur station
  - *le tout premier : NCSA Mosaic*
  - *Netscape Communicator/Mozilla/Firefox*
  - *MS Internet Explorer*
  - *Opera*
  - *Konqueror*
  - *Javasoft HotJava*
  - ...
  - *Lynx (pour VT, Minitel 24x80)*
  - *AvantGo, MS Pocket Explorer, ... (pour PDA)*



# Autres Client (*User-Agents*) HTTP

- HTTPClient <http://hc.apache.org/>
  - components for client-side authentication, HTTP state management, connection management, and an HTTP/1.1. compliant HTTP agent implementation based on those components
- Curl <http://curl.haxx.se/docs/manpage.html>
  - command line tool for transferring files with URL syntax, supporting FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS and FILE. curl supports SSL certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form based upload, proxies, cookies, user+password authentication (Basic, Digest, NTLM, Negotiate, kerberos...), file transfer resume, proxy tunneling and a busload of other [useful tricks](#).



# Palmarès d'utilisation des navigateurs

## *MS IE est en tête*

---

	<i>12/2002</i>	<i>07/2003</i>
■ Microsoft IE 6.0	57.6%	66.3%
■ Microsoft IE 5.5	18.3%	14.5%
■ Microsoft IE 5.0	16.9%	12.7%
■ Netscape Navigator 4.0	1.1%	0.6%
■ Mozilla 1	1.1%	1.6%
■ Microsoft IE 4.0	0.9%	0.8%
■ Opera 6.0	0.8%	0.6%

*The winner is ...*

■ <b>Microsoft IE</b>	<b>95.0%</b>	<b>95.4%</b>
■ Netscape Navigator	3.0%	2.5%
■ Mozilla	1.1%	1.6%

# Les *User-Agents* HTTP

- Mais aussi
- sur Console de Jeux, PDA, iTV, Téléphone Cellulaire
  - WinCE Pocket Explorer, AvantGo ...
- Navigation Online / Offline
  - Ladtop, PDA, ...
  - Récupération en push dans un cache
  - Puis consultation des documents dans le cache

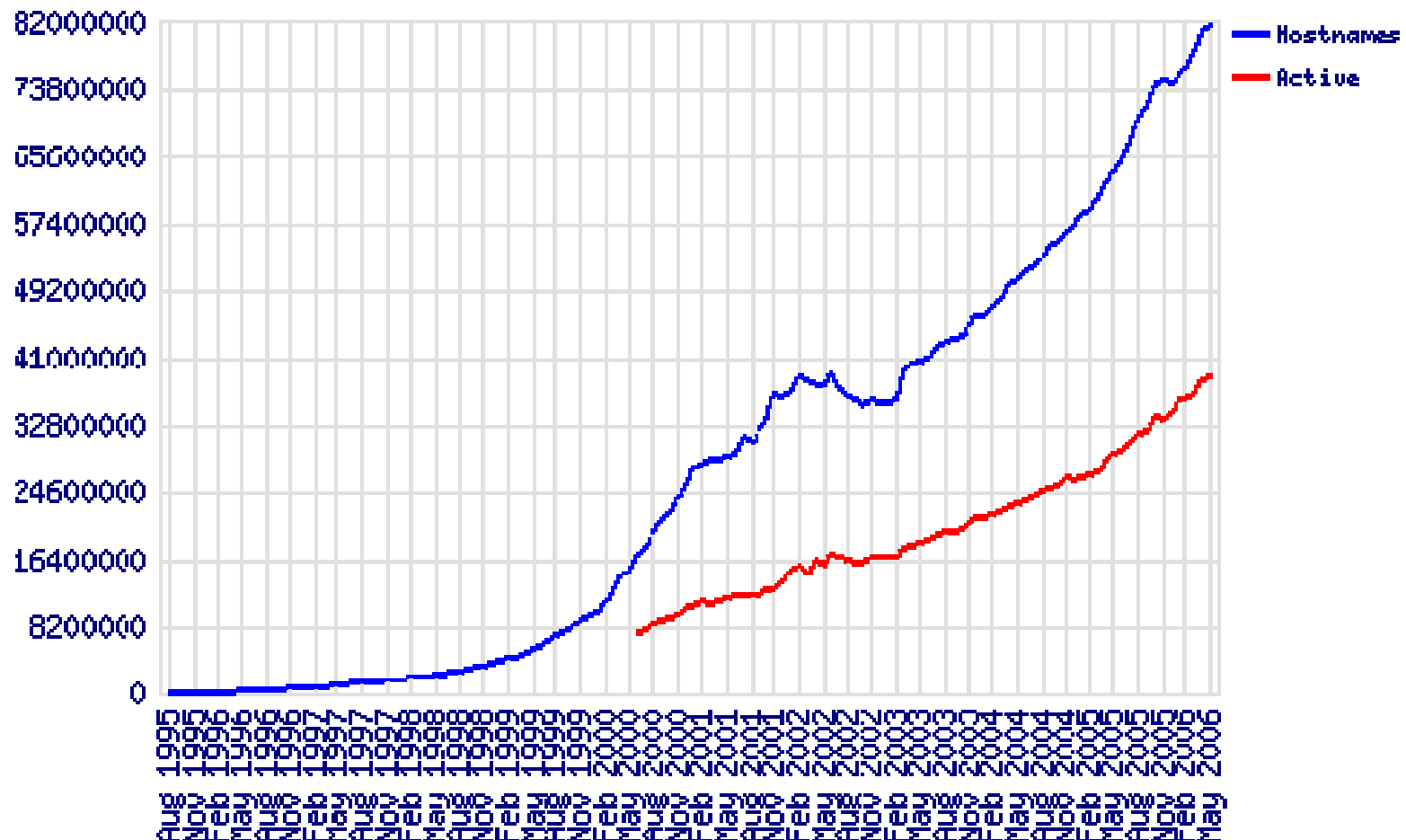
# Les Serveurs du Marché

---

- Offre logiciel très large
  - *NSCA HTTPD, l'ancêtre*
  - Apache & Jakarta/Tomcat (19 millions de serveurs (09/2001))
  - iPlanet et Netscape Entreprise Server
  - Microsoft Internet Information Server (8.9 millions de serveurs (09/2001))
  - W3C Jigsaw
  - Sun JavaServer
  - Oracle Web Server, *abandonné par Oracle au profit d'Apache*
  - IBM Web Sphere, *intègre Apache*
  - Resin, Jetty, Zeus...
- Offre matériel
  - Server Appliances : Qube, ...
  - Composants pour des serveurs embarqués
- Fonctionnalités supplémentaires
  - gestion des sessions, des transactions, ...
  - accès aux serveurs d'applications, ...
- A regarder régulièrement
  - [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html)

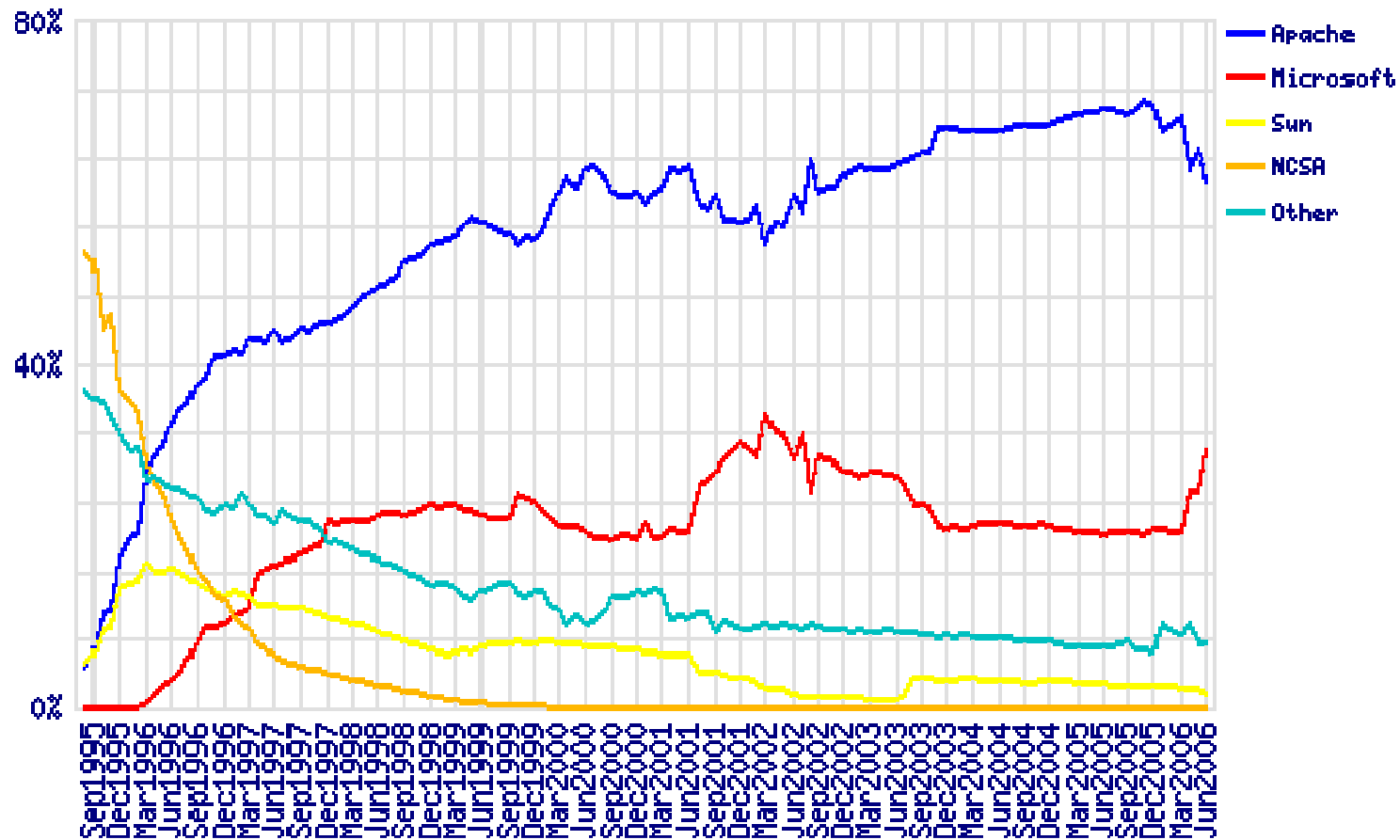
# Les Serveurs du Marché

- Nombre de site
  - [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html)



# Les Serveurs du Marché

- Market share
  - [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html)



# Rôle d'un serveur HTTP

---

- Transformation de l'URL en fichier ou en script
- Vérification d'identité
  - Le client est-il qui il prétend être ?
- Vérification d'accès
  - Le client est-il autorisé à effectuer cette requête ?
    - ACL, ...
- Constitution de l'entête de la réponse
  - Type MIME des données
    - mime.types fichier de correspondance extension vers type MIME
  - Taille des données, Langage, ...
- Envoi de la réponse au client
  - éventuellement transformé à la volée
- Mise à jour des journaux d'audit (log)
  - access\_log, error\_log, ...

# L'authentification dans HTTP

- Indiqué dans les ACL
- Mode d'authentification
  - BASIC (RFC 2617)
    - nom d'utilisateur et mot de passe échangé en clair (base64) !
    - base des mots de passe dans un fichier htpasswd
      - utilitaires de gestion du fichier
  - DIGEST (RFC 2617)
    - sécurisation de BASIC
    - hachage sécurisé MD5 du (nom,password,URI, méthode,nombre aléatoire fourni par le serveur)
  - SSL/TLS
    - Secure Socket Layer (TLS : Transport Layer Security)
    - authentification avec CA du serveur (2.0) et du client (3.0)
    - confidentialité avec DES
    - puis dialogue HTTP sur la connexion SSL
- Remarque : utilisation des cartes PKI pour l'identification du client

# L'authentification applicative

- Motivations
  - interface de login
  - authentification plus forte (SSL)
  - Base d'identité
    - Fichier htpasswd, BD, Annuaire LDAP, ... (voir mod\_auth\_xxx de Apache)
- L'application gère l'authentification de l'utilisateur
  - formulaire d'accueil HTML (nom, password)
    - attention le mot de passe circule en « clair »
      - Utilisez une fonction JavaScript de hachage (MD5,SHA1) du mot de passe avec un nombre aléatoire pour éviter le « replay »
  - gestion des tables d'utilisateur
  - une session est ensuite ouverte associée à un utilisateur authentifié (ou non : par exemple rejet à bout de 3 tentatives)
- Remarque
  - TomCat peut préciser les documents de login et d'erreur pour une arborescence de documents/servlets/JSP



# Contrôle d'Accès dans HTTP

---

- ACL (Access Control List)
  - spécifie les autorisations (ALLOW) ou les interdictions (DENY) d'accès à une arborescence virtuelle du serveur
  - en fonction
    - de l'authentification
    - de la localisation du client
      - sous domaine DNS
      - réseau ou adresse IP
- ACF (Access Control File)
  - fichier regroupant les ACL
    - global : access.conf dans Apache
    - par arborescence : .htaccess
  - combinaison des ALLOW et des DENY

# Audit des Requêtes

---

- Journaux des requêtes
  - les accès (access.log, refferee.log), et les erreurs (error.log), ... sont journalisés
- Exploitation des Journaux
  - erreur dans les liens, ...
  - clientèle, analyse d 'activité, ...
- Reporting (Présentation Synthétique)
  - Pour Apache
    - AccessWatch, Wusage, Analog, wwwstat
  - IIS, NS
    - intégré et visualisé par un script
  - Généraux
    - Net Analysis (Net Genesis), Enterprise Suite (Web Trends)

# Apache

[www.apache.org](http://www.apache.org)

---

- A patch of NCSA HTTPD
  - serveur le plus répandu (62% des serveurs au 04/2001, 18 Millions de serveur)
  - 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 externes ([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, ...)

# Jakarta Tomcat

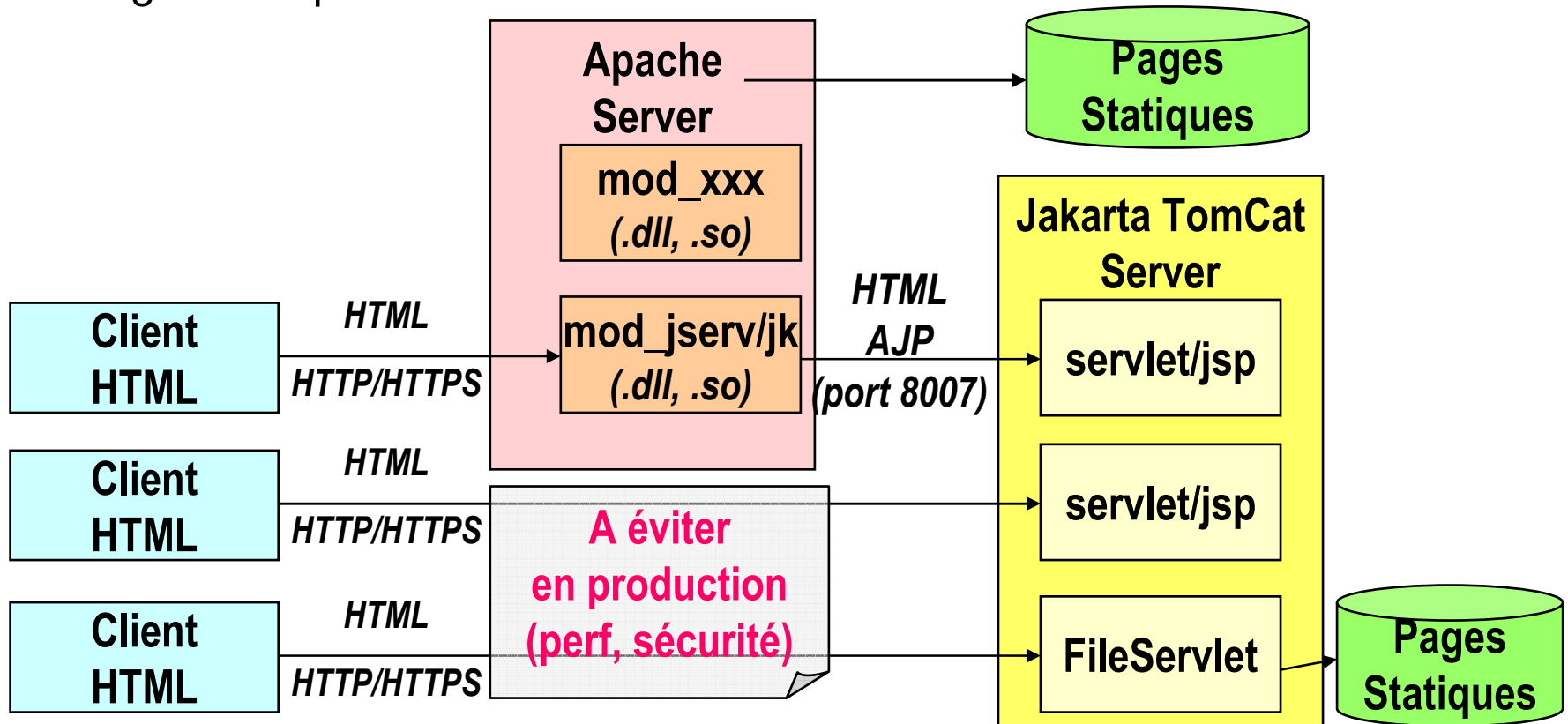
*([jakarta.apache.org](http://jakarta.apache.org))*

---

- Moteur (engine) de Servlet/JSP
  - Catalina : Container JavaEE de Servlet
  - Jasper : Container JavaEE de JSP
- Connecteurs
  - Admin (8007)
    - Démarrer/relancer/arrêter Tomcat
  - HTTP 1.1
    - peut servir directement des clients HTTP
  - HTTPS
    - peut servir directement des clients HTTP sur SSL/TLS
      - Il est préférable d'utiliser Apache/SSL + AJP12, AJP13, WARP
  - AJP12, AJP13, WARP
    - Avec des serveurs HTTP supportant WARP
- Fonctionnement possible en « embedded »
  - cad dans une machine virtuelle avec d'autres services (containers EJB, ...)

# Apache/JServ et Jakarta Tomcat *(java.apache.org)*

- mod\_Jserv / mod\_jk : module d'extension Apache
  - Démarre une JVM et invoque des servlets
- TomCat : serveur écrit en Java
  - A utiliser comme Servlet et JSP engine
- Configuration possible



# JavaServer

---

- Serveur HTTPD de SUN
  - anciennement « Jeeves », écrit en Java
  - Abandonné au profit de Tomcat ([jakarta.apache.com](http://jakarta.apache.com))
    - qui est devenu l'implémentation de références pour les Servlets, JSP, JSTL
- Servlets
  - Scripts serveur écrit en Java
  - Servlets de Base : FileServlet, CGIServlet, ...
- Remarque
  - le bytecode d'une servlet peut être téléchargé et exécuté par la JVM du Serveur (dans un espace protégé ou non)
  - Jigsaw du W3C fonctionne suivant le même principe

# MS Internet Information Server

---

- Plate-forme Systèmes Serveur de Windows
  - 8.9 millions de serveurs (09/2001)
- Scripts
  - ASP, ASP .NET
  - Interface native ISAPI (C, C++, ...)
- Remarque
  - PWS : Personal Web Server (pour les systèmes WorkStation)
  - Maintenant intégré à MS .NET (Web Services, ...)
  - Cassini (ASP .NET Web Server pour le développement)

# Autour d 'HTTP

---

- Proxy
  - seul point de passage entre le réseau d 'entreprise et l extérieur
    - sécurité, firewall
  - accès à des protocoles non implémentés par les clients Web : WAIS, ...
- Cache
  - soulager les accès externes (moins de bande passante)
- Miroir
  - réplication d 'une base documentaire
  - améliorer le temps de réponse, soulager le réseau
- Robot
  - récupération online/offline d 'une arborescence de documents
  - constitution d 'un miroir local
- Mise à jour de sites
  - copie publique / copies de travail
- Portail
  - Point d'accès centralisant à un ensemble de sources Web
- Embedded Web Server
  - Serveur HTTP pour informatique enfouie et embarquée

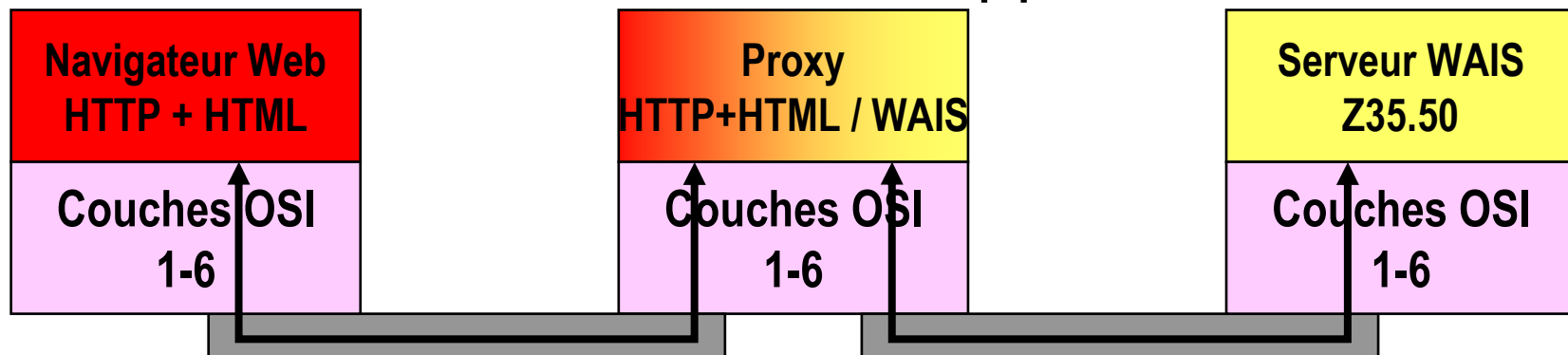


# Proxy

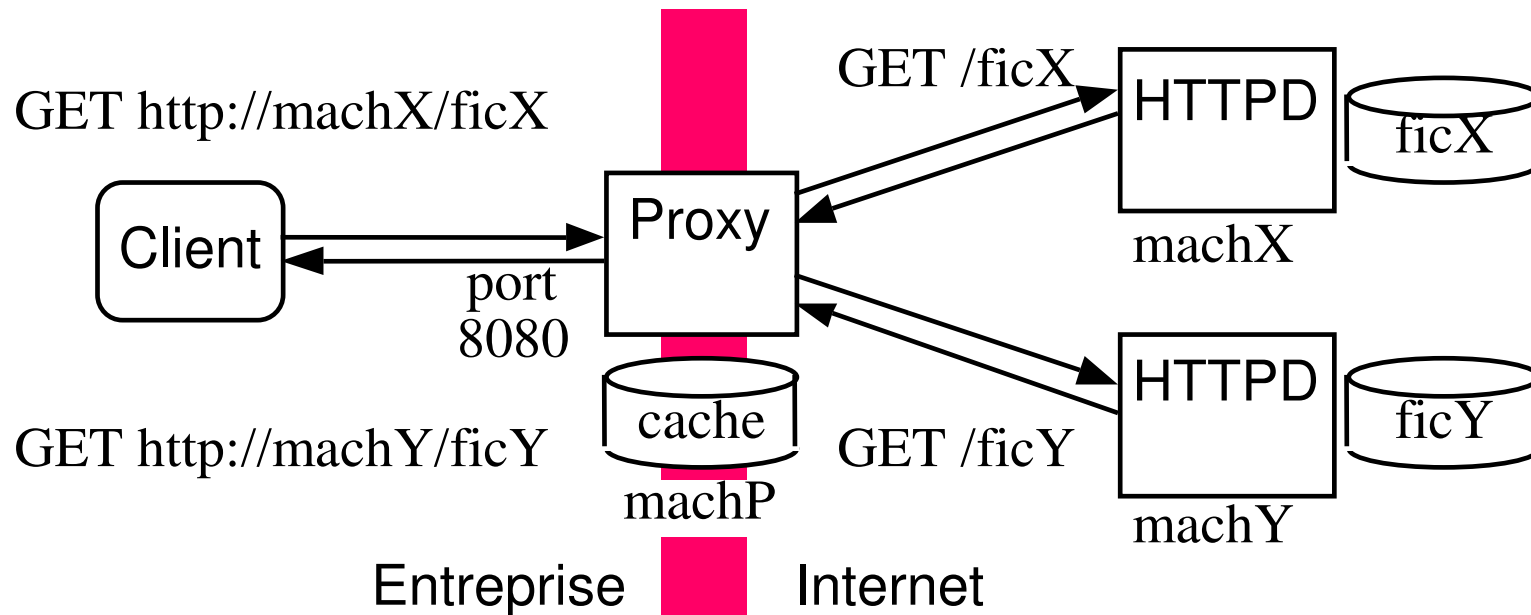
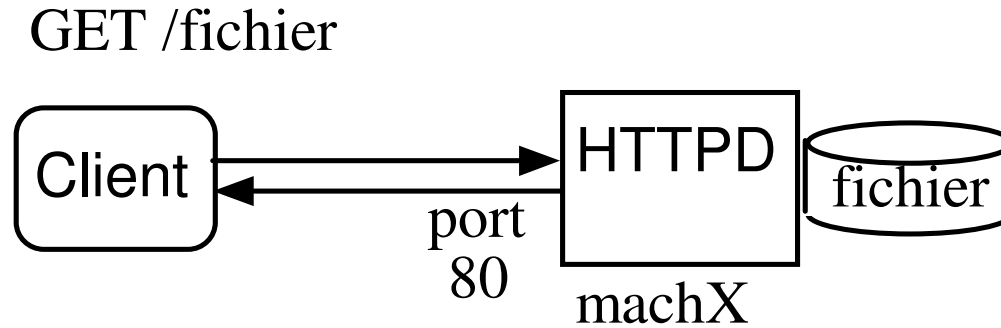
## ■ Fonctions

- seul point de passage entre le réseau d'entreprise et l'extérieur
  - firewall, contrôler le profil d'utilisation, portail vers des ressources externes contrôlées (banques de données, ...)
- accès à des protocoles non implémentés par les clients Web
  - WAIS, LDAP...

## ■ Passerelle réseau de niveau applicatif



# Fonctionnement du Proxy/Cache



# Cache Web

---

## ■ But d'un cache Réseau

### ■ Usager

- Améliorer les performances du browser en utilisant les documents présents dans le cache local

### ■ ISP

- Soulager le réseau fédérateur en cachant les documents demandés par les usagers du sous-réseau

### ■ Provider Web

- Soulager les serveurs applicatifs

## ■ Documents

### ■ Textes (HTML, XML, ...) et Images fixes

### ■ Flux Audio et Video

- têtes de réseau cable pour la Buffered-VOD

## ■ Serveurs

### ■ Il fonctionne en mode Proxy

- Squid, Harvest, Apache, MicroSoft, Sun, iPlanet, OracleAS WebCache, WebSphere

...

# Cache Web ii

---

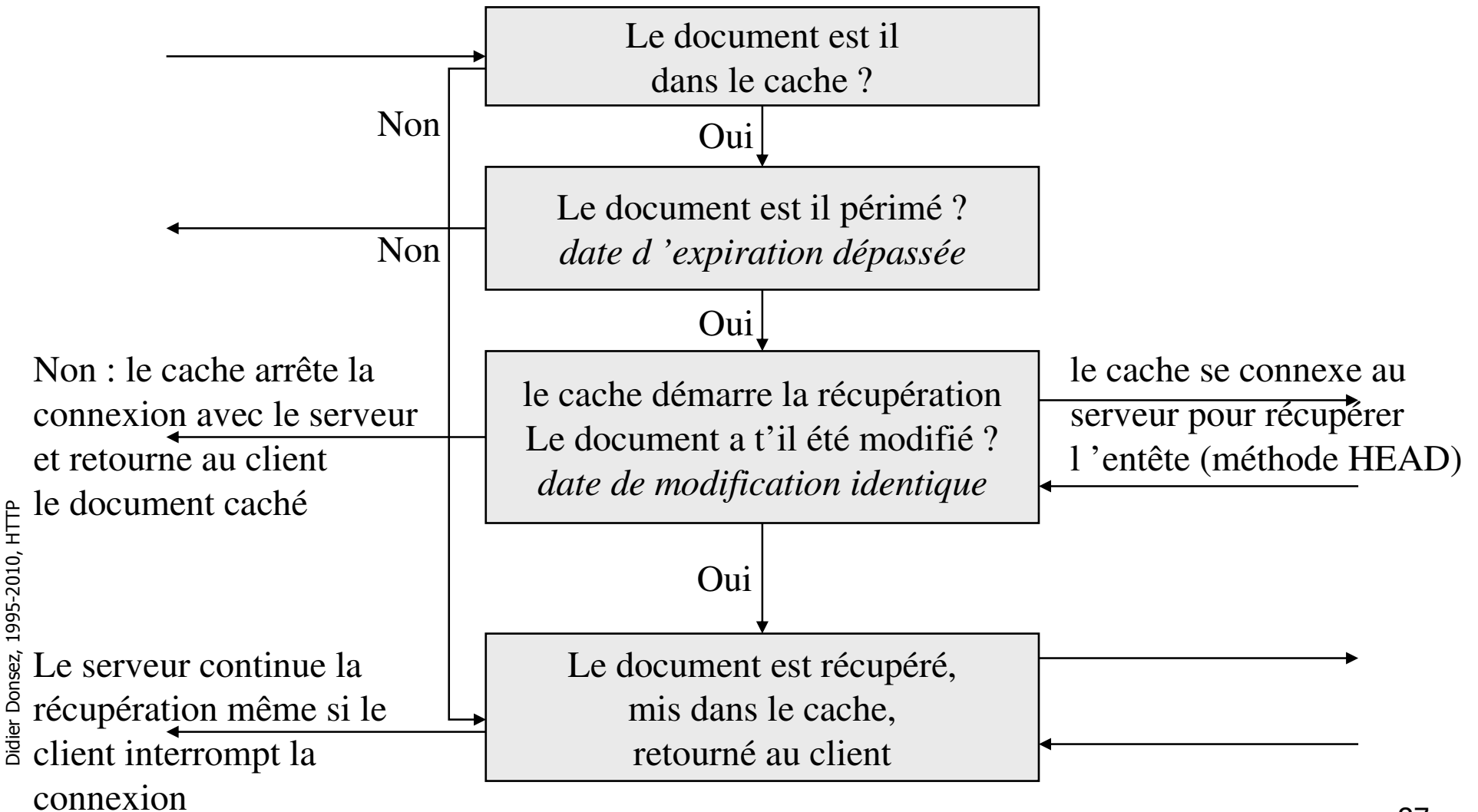
- Politique de remplacement de cache Web
  - Least Recently Used (LRU) *très populaire*
    - conserve la date du dernier accès
  - Least Frequently Used (LFU)
    - comptage du nombre de références
    - éviction des documents les moins référencés
  - LFU-Aging
    - LFU + prise en compte de la durée de la présence dans le cache
  - Greedy Dual-Size (GDS)
    - prise en compte la taille
  - GDS-Frequency (GDSF)
    - GDS + prise en compte du nombre de référence

# Cache Web iii

---

- Cache de Caches
  - Hiérarchie multi-niveaux de caches
    - exemple : RENATER
    - *cache Client -> cache Sous Réseau -> cache Backbone -> ...*
  - équilibrage dynamique
    - entre plusieurs caches de même niveau
- ICP : Inter Cache Protocol
  - mise à jour, invalidation, redirections de requête

# Gestion du Cache

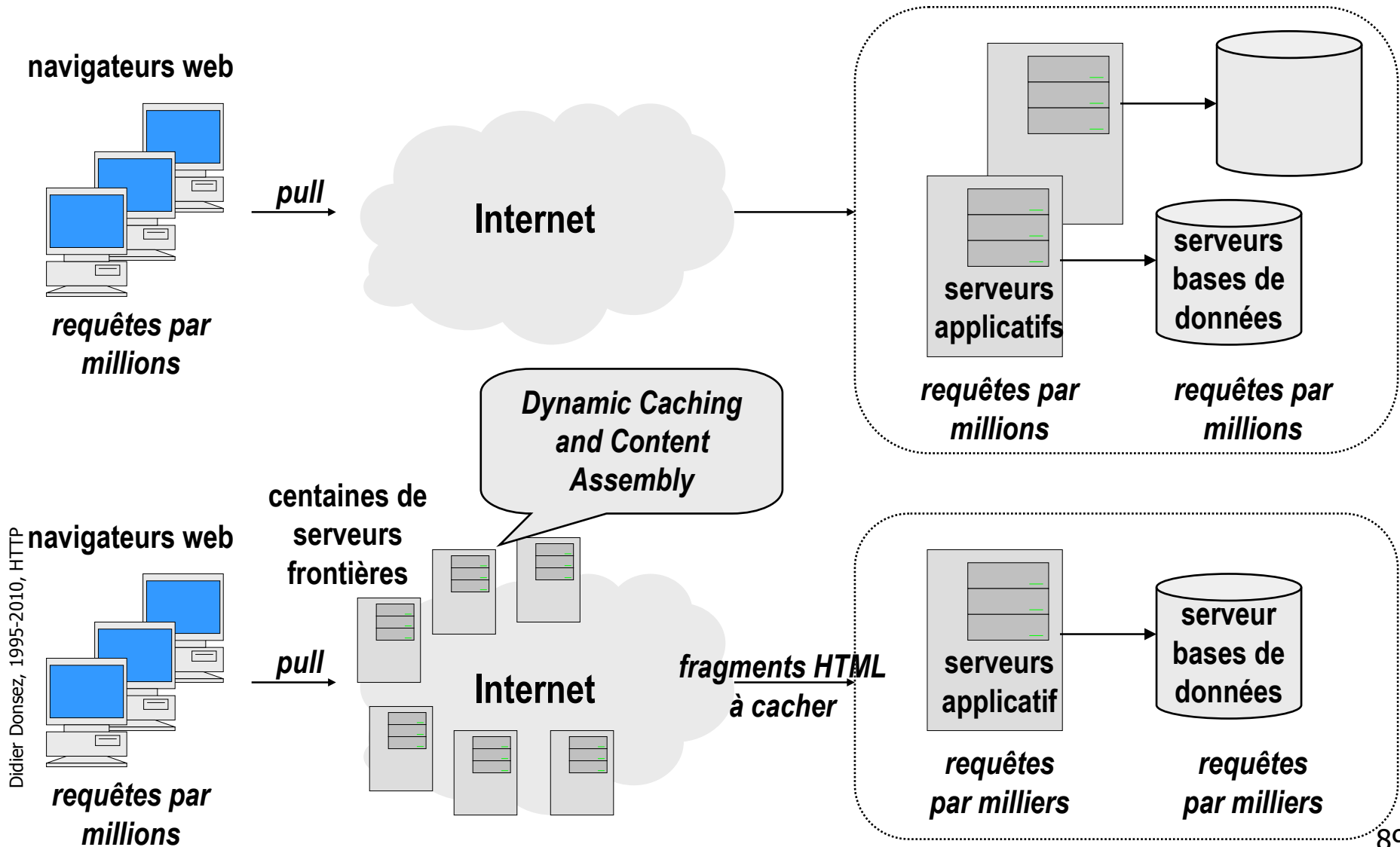


Didier Donsez, 1995-2010, HTTP

# Content Delivery Network (CDN)

- Web cache aux points de présence du(des) FAI entre le fournisseur et ses clients
- Documents cachés
  - Statique, Stream, Fragments de compositions dynamiques (ESI)
- Avantages
  - Latence réduite, décharge du serveur fournisseur (allocation de serveurs à la demande, ...), décharge du réseau central FAIs
- Inconvénients
  - Usage de DNS dynamique dissociée pour cachable et non cachable
  - Réécriture dynamique d'URL
- Acteurs
  - Akamaï (+Digital Island) ~80% du marché
    - plus de 20000 (*surogate*) serveurs low-cost sur 63 pays
  - Adero, Mirror Image, Inktomi, ...

# Content Delivery Network (CDN) *le Edge Computing*





# Miroir

---

- But
  - Créer une copie miroir d'un site (d'une partie de site)
    - accélère l'accès au document
    - diminue la contention d'accès au site
    - diminue le trafic global sur Internet
- Mise à jour du miroir
  - déclenchement automatique, manuel, push
  - incrémentale / non

# iCAP (i)

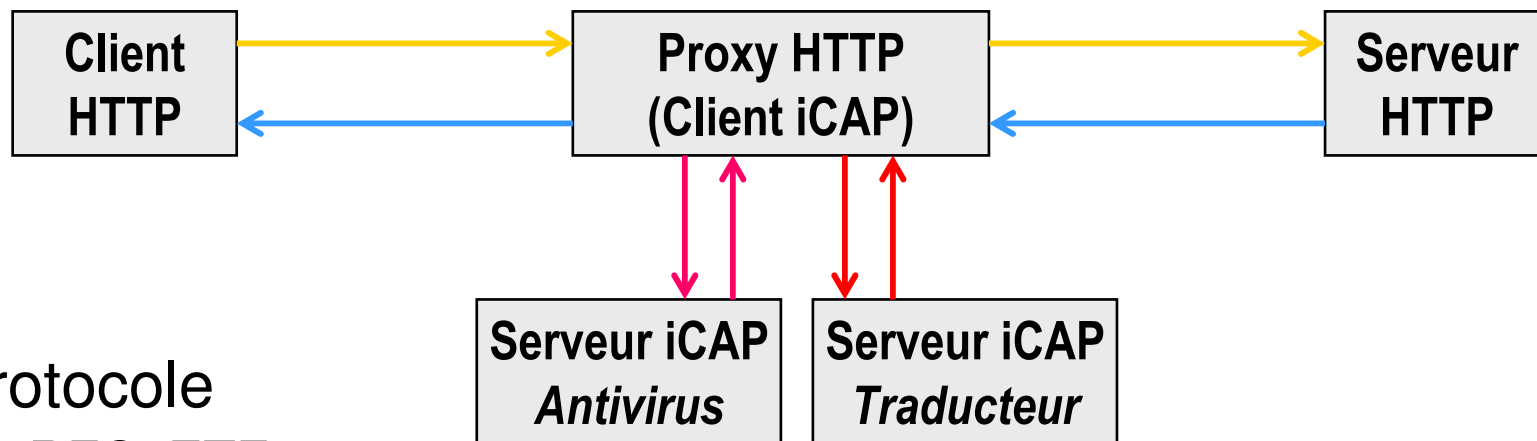
## *Internet Content Adaptation Protocol*

- Motivations
  - permettre à un proxy (HTTP, FTP, NNTP, SMTP) d'intercepter les messages et de les faire traiter/modifier « à la volée » par d'autres services
- Exemple d'utilisation :
  - Détection de virus, Insertion de bannière publicitaire, Reroutage, Détection de langage, Traduction multilingue, Conversion de format (image, ...), Contrôle parental ...

# iCAP (ii)

## *Internet Content Adaptation Protocol*

### ■ Architecture



### ■ Protocole

- RFC IETF
- Couche Transport : HTTP
- Méthodes : RESPMOD, REQMOD, OPTIONS
  - Encapsulation du message (HTTP, ...)
- URL : icap://host:port/service (port par défaut : 1344)
- Implémentation : iPlanet, ...

# iCAP(iii)

Service	Architecture			
	Request Modification	Request Satisfaction	Response Modification	Result Modification
Content Filtering	Yes	Yes	Yes	Yes
Gateway Translation	Yes		Yes	
Language Translation	Yes	Yes	Yes	
Virus Scanning		Yes		
Ad Insertion	Yes	Yes	Yes	Yes
Data Compression		Yes	Yes	

# Robot

## (Aspirateur de Site, Glaneur, ...)

---

- Récupération d'une arborescence de documents à partir d'une URL « racine »
  - Règles de récupération
    - par rapport aux extensions (que les textes)
    - profondeur du suivi de URL et limite de la récupération
  - Déclenchement programmé
    - heures creuses (débit et \$)
- Produits
  - nombreux freewares/sharewares
    - webcopy, wget, w3mir, Jobo ...
  - Fonction miroir intégrée dans les logiciels auteurs
    - MS FrontPage, Goto MemoWeb, ...
- Remarque
  - permet de créer facilement un miroir
  - utile à la navigation offline

# Robot

---

- Directive pour les robots
  - Faciliter le travail des robots des moteurs d'indexation
- META HTML
  - <META NAME='revisit-after' CONTENT='15 days'>
  - <META NAME='robots' CONTENT='index,follow'>
  - <META NAME="keywords" CONTENT="Cours HTTP, Robot, WebDAV ">
- Fichier /robots.txt (<http://www.robotstxt.org/wc/norobots.html>)
  - Interdit aux robots certains répertoires ou documents
  - Exemple

```
# interdit cyberworld/map/ a tout robot
User-agent: *
Disallow: /cyberworld/map/
# sauf à Cybermapper qui peut tout visiter
User-agent: cybermapper
Disallow:
```

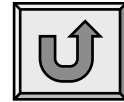
# Mise à jour des sites

---

- Actuellement, chaque auteur travaille sur une copie locale et remplace la copie publiée sur le Web par celle-ci régulièrement
- Remplacement des pages
  - par FTP
    - problème de gestion des versions multiples lors d'un travail de groupe
  - par un script HTTP serveur propriétaire
    - et information propriétaire associée
    - utilisé par MS FrontPage, NetObjects Team Fusion, Macromedia DreamWeaver, ...
    - mais pas d'interopérabilité
  - WebDAV [RFC 2518]  
*Web Distributed Authoring and Versioning Protocol*
    - le dernier effort de standardisation

# WebDAV (1/3)

## *Web Distributed Authoring and Versioning Protocol*



- Extension de HTTP pour la mise à jour de site (RFC 2518)
  - [www.webdav.org](http://www.webdav.org), [www.ics.uci.edu/pub/ietf/webdav](http://www.ics.uci.edu/pub/ietf/webdav)

### ■ Notions

- Propertie (propriété)
  - décrit un document (auteur, taille, date de dernière modification, ...) au format XML/RDF
- Collection
- Locking (verrouillage)
  - verrou partagé (shared) / verrou exclusif (exclusive)
- Namespace (espace de nommage)
  - groupement logique de ressource pour la gestion (verrouillage, contrôle d'accès, ...)



# WebDAV (2/3)

## *Web Distributed Authoring and Versioning Protocol*

- Ajout de nouvelles commandes HTTP
  - PROPFIND
    - retourne les propriétés
  - PROPPATCH
    - modifie les propriétés
  - MKCOL
    - crée une nouvelle collection
  - COPY & MOVE
    - copie ou déplace une ressource au sein d'un espace de nommage
  - LOCK & UNLOCK
    - verrouille et déverrouille un ressource

# WebDAV (3/3)

## *Web Distributed Authoring and Versioning Protocol*

- des Serveurs WebDAV
  - MS IIS5
  - CyberTeams ' Web Site Director
  - mod\_dav pour Apache ([www.webdav.org/mod\\_dav](http://www.webdav.org/mod_dav))
  - Support dans Tomcat
    - *org.apache.catalina.servlets.WebdavServlet*
- des Clients WebDAV
  - MS Office 2000 et Explorer 5
  - Adobe GoLive)
  - Cadaver 0.15
  - DAV Explorer 0.60 and 0.70
  - Jakarta Slide 1.0 WebDAV client library
  - SkunkDAV 1.0
- A voir
  - <http://www.webdav.org>

## WebDAV/DeltaV (RFC 3253)

- Extension de WebDAV pour le support des révisions
  
- Auto-versionning
  - Possibilité de révision avec des clients WebDAV non DeltaV
  
- Utilisé par SubVersion mod\_dav\_svn (http/https)

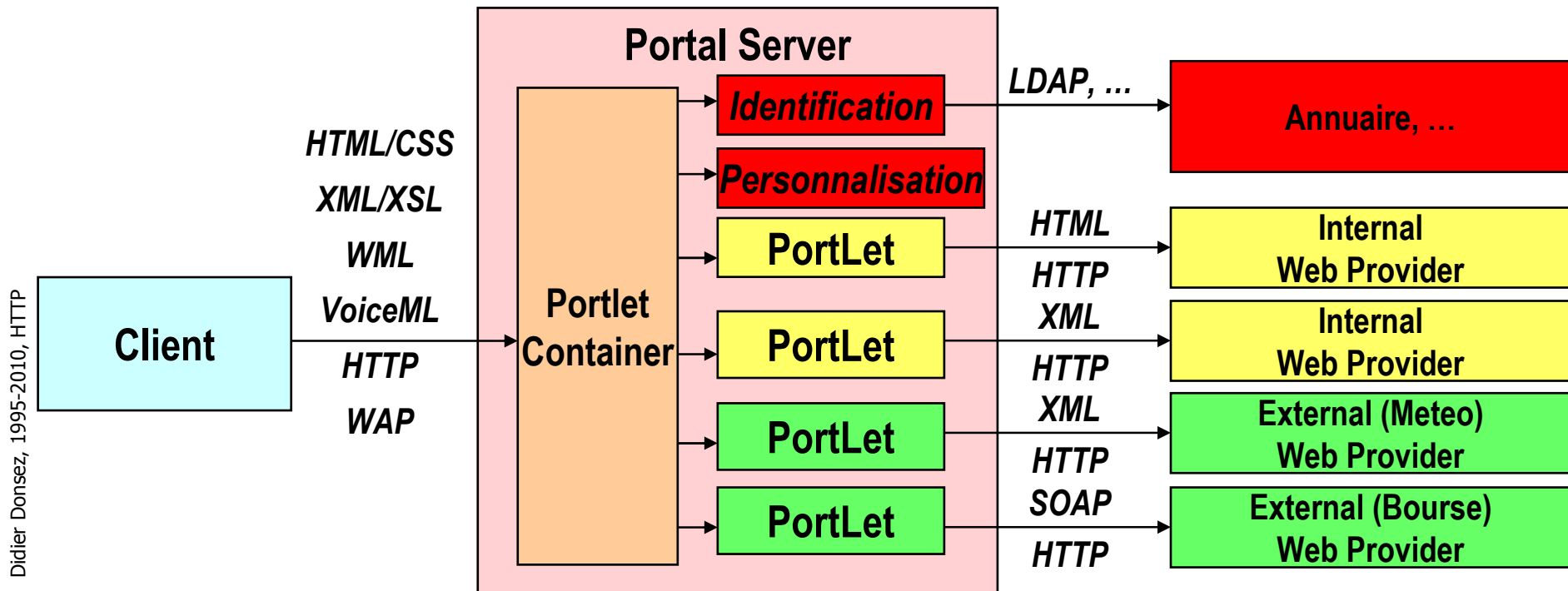
# Portail (i)

---

- Objectif
  - Point d'accès centralisant un ensemble de sources d'information Web
    - Intra-entreprise (Enterprise Information Portal ou Web Based Groupware)
      - Contenu : Favorite links, Agenda, Mail, Incident, Challenge Force de Vente ...
    - Grand public
      - Contenu : Info Bourse, Enchère, Météo, Programme TV et Ciné, News Headline, Courrier (HTML), SMS ...
    - B2B pour des agrégateurs de Web Services
- Fonctionnalités
  - Usagers Anonymes/Authentifiés
    - Sécurisation (SSL, X509, cookie ...)
  - Personnalisé
    - en fonction de l'utilisateur, de son rôle, de son emplacement géographique, ...
  - Personnalisable(Self-Service)
    - Composition dynamique (utilisation d'assistants, ...)
  - Multi-accès (PC, Téléphone Cellulaire, PDA, ...)
- Outils
  - JSR 168, Oracle Portal, NetPortal, WebPortal, iPlanet Portal, Apache Portals, ...

# Portail (ii)

- Présentation (GUI)
  - 1 page d'accueil composite -> N onglets -> M zones (*div, layer, frame, card, ...*)
    - iconifiable, agrandissement/réduction, ajout/retrait, ...
- Développement de Portlet
  - Notion de PortLet : représente une source de données à placer dans la page



- Nouveau métier : provider de PortLet

# Portail (iii) Exemple de GUI HTML (*My Yahoo*)

The screenshot shows the My Yahoo! homepage with several portlets highlighted by red dashed boxes and labeled with red text:

- Identification:** A box at the top right containing the text "Bienvenue, Didier" and navigation links like "Infos compte", "Aide", and "Fermer session".
- Portlet Mail:** A box containing the "Yahoo! Courrier" section, which includes a login prompt "Pour lire votre courrier, saisissez votre mot de passe" and an "Envoyer" button.
- Portlet Portfolio:** A box containing the "Portefeuilles" section, which lists various stock indices and their values, such as "350000.PA" and "FTSEL".
- Portlet News:** A box containing the "Nouvelle Page Actualités" section, which lists news items under categories like "France: Culture", "France: Monde", and "France: Économie".
- Portlet Weather:** A box containing the "Météo" section, which shows weather information for "Chamonix-Mont Blanc", "Chamrousse", and "Grenoble".

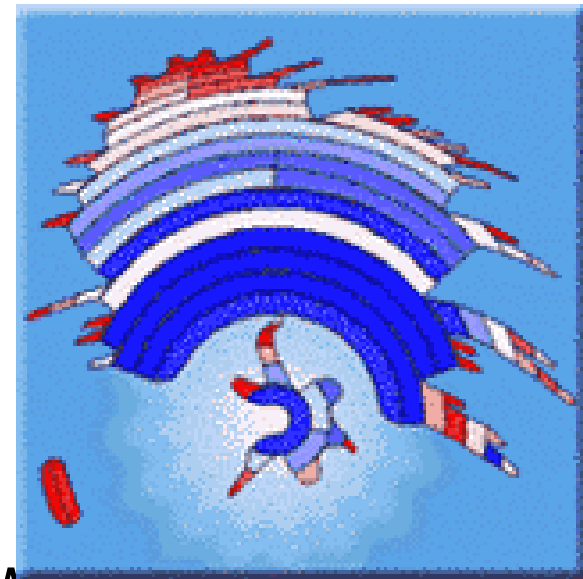
Didier Donsez, 1995-2010, HTTP

# Portails Apache <http://portals.apache.org/>

- JetSpeed 1 et 2
  - Servlet de portail Web et WAP
  - Description
    - PSML (Portal Structure Markup Language) par utilisateur
      - Description des portlets et de leur contenu, de leur mise en page
- Pluto
  - Reference Implementation of the **Java Portlet Specification (JSR 168)**
  - Pluto is a portlet container.
- WSRP4J
  - Reference Implementation of the Oasis (**Web Services for Remote Portlets**) Specification. WSRP standard simplifies integration of remote applications/content into portals so that portal administrators can pick from a rich choice of services and integrate it in their portal without programming effort.

# Cartographie du Web

- Grouper les sites en « régions »
  - en fonction de leur relation sémantique, des références d'URL
  
- Représentation de la carte
  - hyperbolique, cible 2D, ...
  
- Outils et sites
  - [www.semio.com](http://www.semio.com), [www.acetic.fr](http://www.acetic.fr), [www.unimap.com](http://www.unimap.com), ...





# Mesures et Rapports d'audience (MédiaMétrie)

---

- Pourquoi faire : suivi et analyse de l'activité du site
  - webmaster : dimensionner le système (période de charge)
  - commercial : pister le client dans sa navigation
  - investisseur : le rassurer avant l'introduction en Bourse ;-)
  - publicitaire : fixer le prix des bandeaux publicitaires
- 2 fonctions
  - Récupération des clics
    - à partir des journaux du serveur HTTP (access.log)
      - inconvénients : les journaux grossissent très vite
    - scripts JavaScript/applet embarqués dans les pages
    - Images masquées
  - Analyse
    - Datawarehouse, DataMining
    - Rapport, OLAP, Segmentation online (Profilage online)

# Mesures et Rapports d'audience (MédiaMétrie)

---

## ■ Acteurs

- Insight (Accrue), WebHound (SAS Institute)
- <http://awstats.sourceforge.net/>

## ■ A lire

- DEVANSHU DHYANI, WEE KEONG NG, SOURAV S. BHOWMICK, « A Survey of Web Metrics », ACM Computing Surveys, Vol. 34, No. 4, December 2002.

# Mesures et Rapports d'audience (MédiaMétrie)

---

- Externalisation des rapports d'audience
  - Pourquoi faire : rapport de l'activité du site
    - investisseur : le rassurer avant l'introduction en Bourse ;-)
    - publicitaire : fixer le prix des bandeaux publicitaires
- Méthodes
  - société tierce « indépendante » proposant des rapports « normalisés » (Exemple : [www.estat.com](http://www.estat.com), ...).
  - chaque page à tracer contient une image transparente de 1x1 pixel qui est chargée depuis la société tierce. La société tierce comptabilise les chargements.
    - `<IMG SRC="http://www.tierce.com?idcust=190874+doc=/product/index.htm">`

# Rapports d'audience

## Exemple

- Rapport d'audience



### STATISTICS

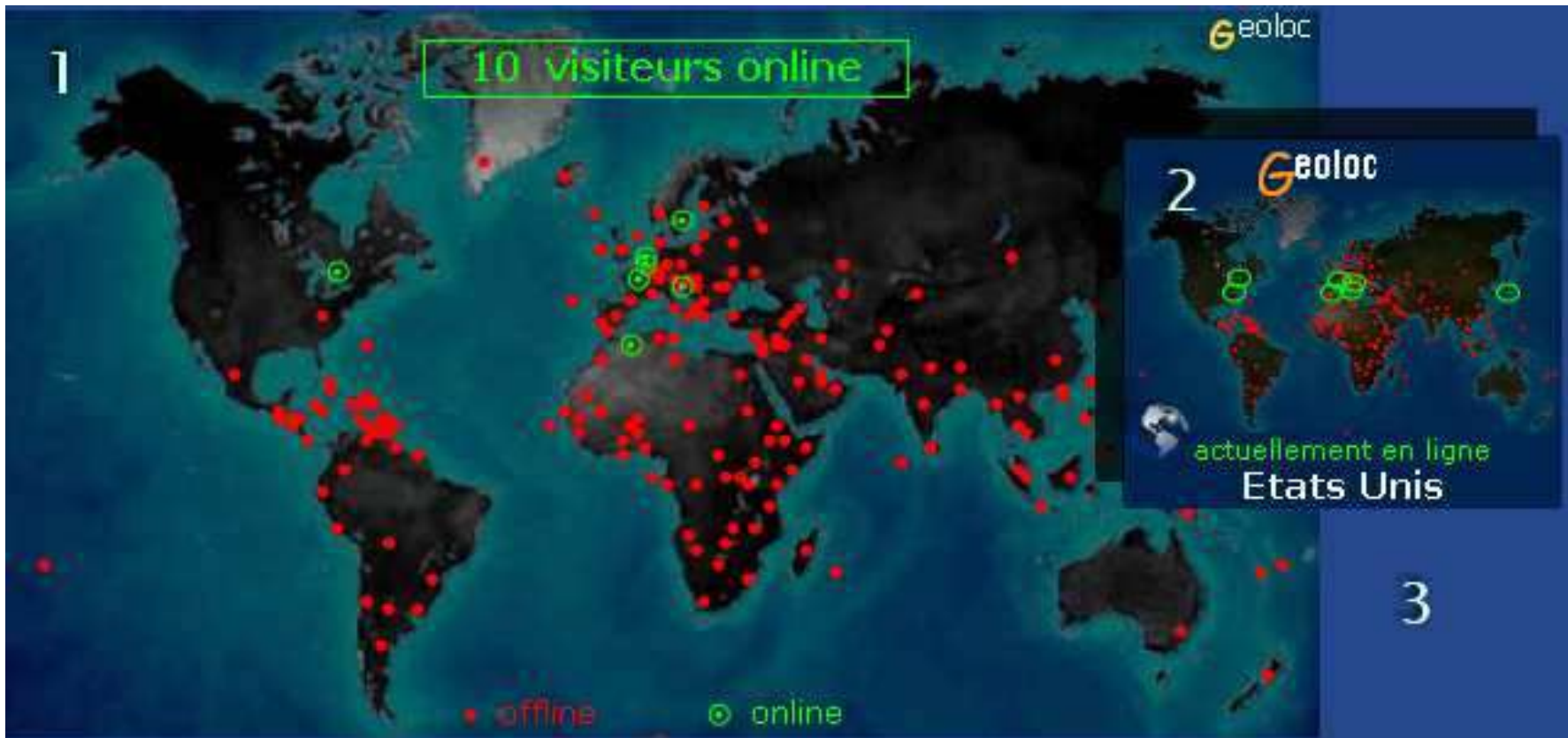


windows, mac, unknown, linux

# Rapports d'audience

## Exemple

- Rapport d'audience Temps réel de [www.geoloc.com](http://www.geoloc.com)



# Service pour le e-Commerce

- Géo-localisation IP
  - Pays d'origine d'une adresse/domaine IP
    - Ex: <http://www.maxmind.com/>
  
- Validation des cartes de crédit/bancaires
  - Pays d'origine fortement sujet à la fraude
    - Ukraine, Indonesia, Yugoslavia, Lithuania, Egypt, Romania, Bulgaria Turkey, Russia, Pakistan, Malaysia, Israel, ...
  
  - Services externes via des Web-Services

# La Personnalisation

---

## ■ Motivations

- Les usagers d'un site ont des goûts et des besoins différents
- La personnalisation tente d'offrir à l'utilisateur une interface correspondant le plus possible à ses goûts et ses besoins

## ■ Solutions

### ■ Analyse (*E-Analytics*)

- Analyse des navigations précédentes de l'utilisateur
  - Net Perceptions, Accrue Software, NetGenesis Corp

### ■ Personnalisation sur règles (*Rules-based Personalization*)

- Création des pages personnalisées à la volée en fonction de règles
  - BroadVision, Vignette Corp.

### ■ Filtrage collaboratif (*Collaborative Filtering*)

- similarité entre utilisateurs : comportement de groupes
  - Net Perceptions, Macromedia, Be Free

# Qualité

---

- Motivation
  - Vérification de la Conformité (Correctness) aux spécifications (donné par le client)
  - Non régression
- Outils
  - Test unitaire et rapport : JUnit+Cactus

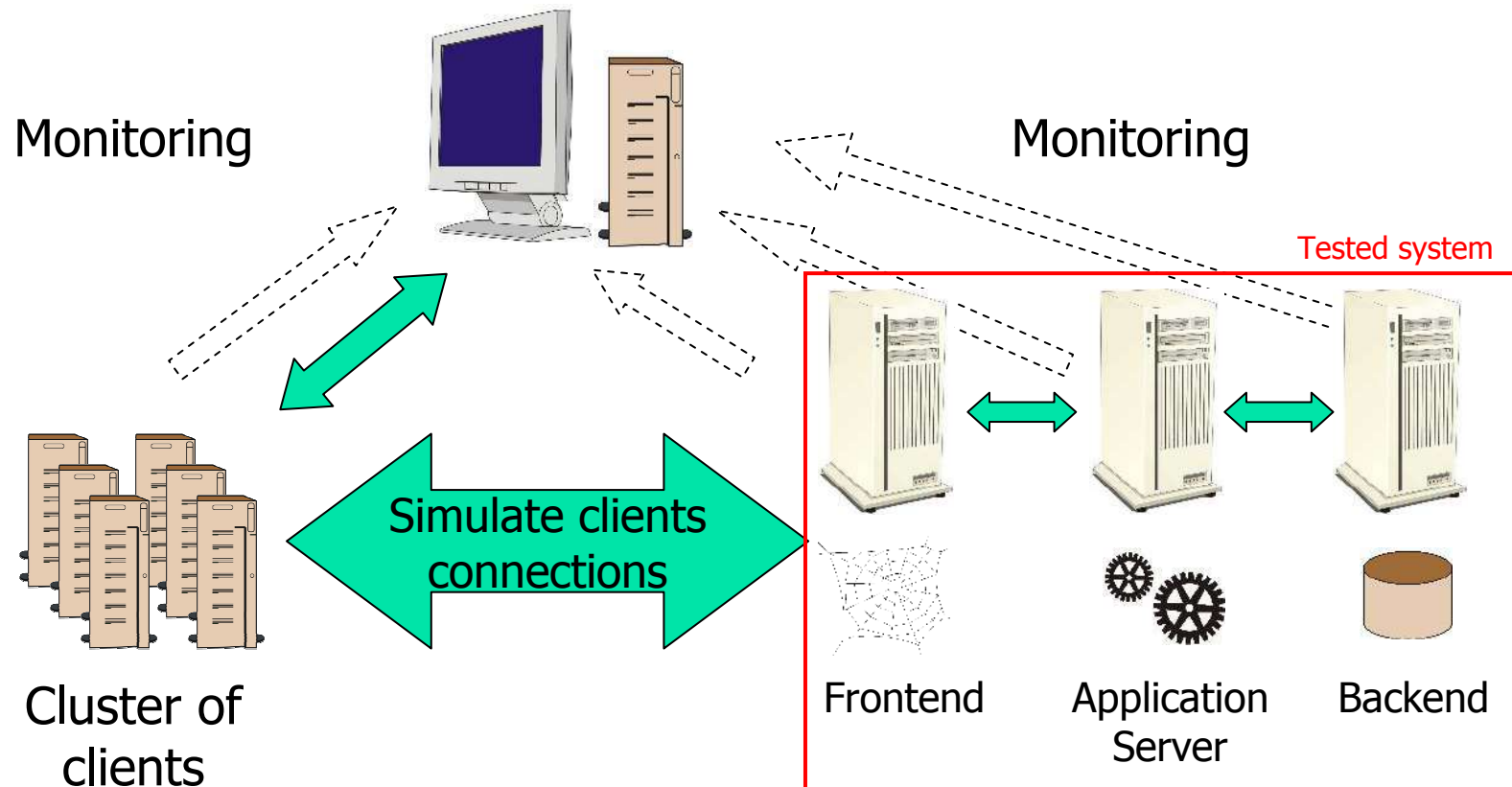


# Test de performance / charge

- Motivation
  - Vérification du niveau de charge supporté avant la mise en production
- Benchmark
  - **TPC-W** (Bookstore online) <http://www.tpc.org> voir *Cours Benchmarks BD*
  - WebMark : orienté requête HTTP/GET sur des documents statiques
  - SPECWeb : <http://www.spec.org/osg/web99/>
  - Rubis (eBay auction) <http://rubis.objectweb.org>
  - Rubos (PHP News SlashDot)
  - CSIRO (Stock online)
    - *Mauro Andreolini, Valeria Cardellini, Michele Colajanni: Benchmarking Models and Tools for Distributed Web-Server Systems. Performance 2002: 208-235*
- Analyse et diagnostique de la charge
  - Détecter quel composant (HTTPD, EJB, MT, SGBD, Mailhost, MQ ...) en créant l'embouteillage (*bottleneck*)

# Test de performance / charge (ii)

- Injecteur de charge
  - 1 à N clients en parallèle qui émulent M WebSurfers sur le site
    - Script définissant la navigation (automate, temporisation, ...)



## Test de performance / charge (iii)

### ■ Outils

- Apache Benchmark Tool (ab), Siege, Apache JMeter, ObjectWeb CLIF, IBM/Rational SiteLoad, LoadRunner, TestMaker, Grinder (<http://grinder.sourceforge.net/>), ...

### ■ Remarque

- Mêmes outils pour les attaques DoS (Deny-Of-Service) et Distributed DoS

## Test de performance / charge (iv)

- Exemple: Apache JMeter
  - Screenshot to add

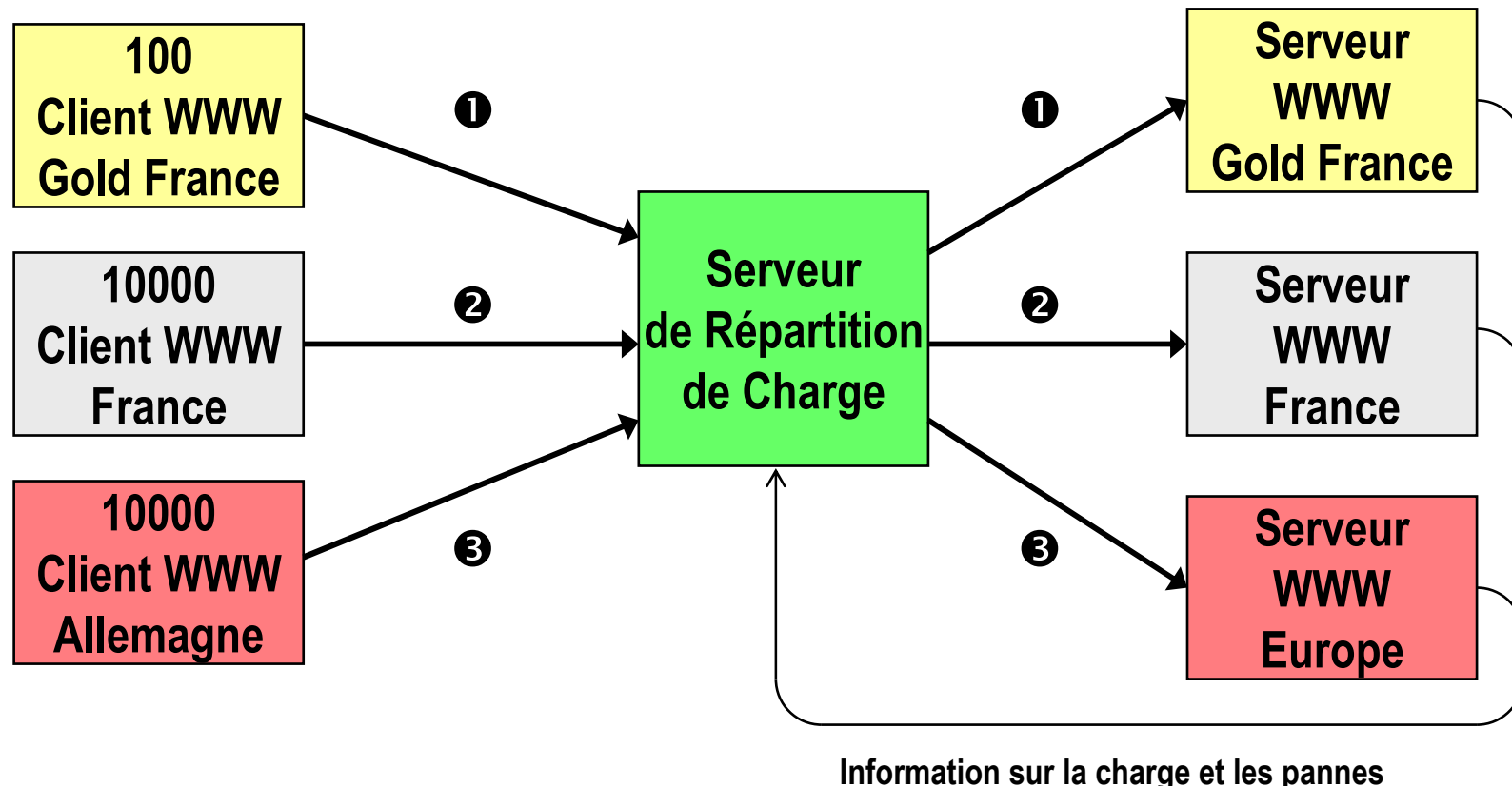
# *Balancing)*

## Disponibilité (*Avaibility*)

- Motivations
  - Performance
    - Améliorer/garantir les temps de réponse
      - Exigé par des portails (vers votre service) comme Yahoo
    - QoS pondérée pour différentes classes d'utilisateurs (gold, silver, ...)
    - Évolution incrémentale de la ferme (parc) de serveurs
  - Disponibilité
    - Tolérance aux pannes
      - Différents niveaux de disponibilité
    - fonctionnement dégradé en cas de panne
      - Le serveur Europe reçoit les requêtes des clients français
    - Failover : Interruption de service
      - Basculement d'une session vers un autre serveur sans interruption de service
- Voir:
  - V. Cardellini, M. Colajanni, P.S.Yu, Dynamic Load Balancing on Web-Server Systems, IEEE Internet Computing, May-June 1999, pp28-39
  - Swaminathan Sivasubramanian, Michal Szymaniak, Guillaume Pierre, Maarten van Steen, Replication for web hosting systems. ACM Computing Surveys, Vol 36, N°3, 2004 ,pp 291-334

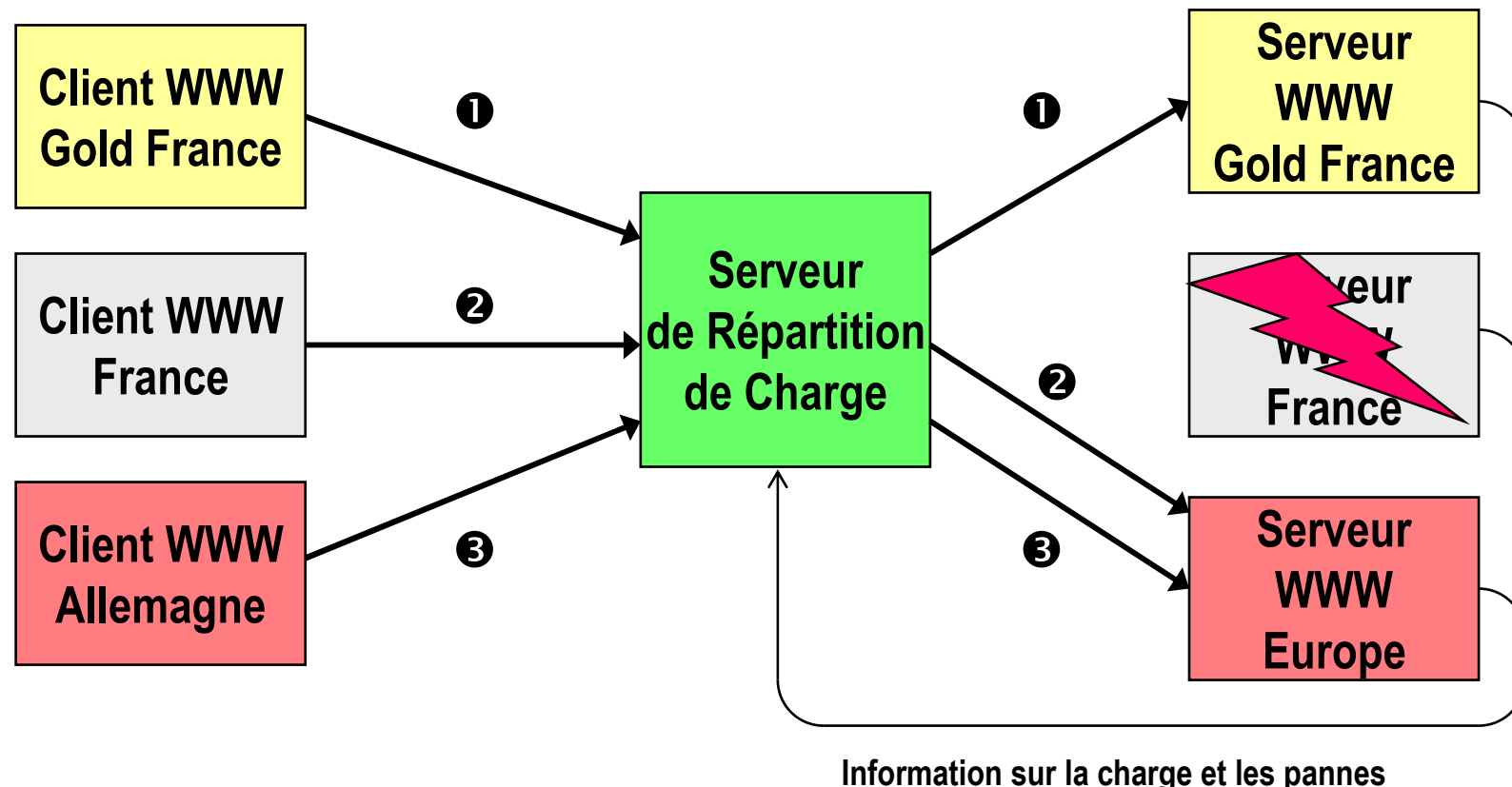
# Répartition de Charge (*Load Balancing*)

- Architecture
  - Un serveur de répartition de charge
  - Une ferme de serveurs Web



# Disponibilité (*Availability*)

- Cas d'une panne
  - Les clients français sont servis par le serveur Europe
    - La panne peut être détectée par des pings réguliers !



# FailOver

---

- Ininterruption de service
  - Reprise de la session par un autre nœud du cluster
- Solutions
  - Réplication de données de HttpSession entre les nœuds
- Technologies
  - Données de session persistantes (BD, Fichiers)
    - ☹ IO traffic bottleneck
  - In memory Replication (Communication de Groupes)
    - Voir article Clustering, theServerSide, April 2002
      - JavaGroup (reliable and synchronous multicast)



# Serveur de Répartition de Charge

- Fonction
  - répartir les requêtes entre les serveurs Web (entre les serveurs applicatifs)
- Politiques de répartition
  - donneur de cartes (*[weighted Round Robin]*)
  - aléatoire
  - vers les serveurs les moins chargés
  - topologique
    - Optimiser la bande passante de l'infrastructure réseau
      - L'entreprise dispose de serveurs répartis sur plusieurs backbones du Web
  - en fonction de la classe d'utilisateurs
    - Offrir différentes QoS (temps de réponse, ...)
  - en fonction des sessions en cours
    - Une session démarrée sur un serveur doit continuer sur ce serveur
      - SSL, Cookies de session pour Http, Stream du RealServer, ...
- Fournisseurs

# Niveaux de la Répartition de Charge par le Serveur de Répartition (i)

- Niveau 3 – IP
  - Répartition par le DNS
    - La résolution DNS n'est pas une constante (Round-Robin DNS, GeoDNS)
      - une adresse DNS correspond une liste d'adresses IP
      - à tester nslookup plusieurs fois de suite sur www.sun.com
    - Future RFP VRRP (Virtual Router Redondancy Protocol)
      - Grappe de routeurs derrière un routeur virtuel
      - Adresse MAC sur LAN
      - Adresse IP sur WAN
- Niveau 4 (TCP-Router)
  - Session SSL

# Niveaux de la Répartition de Charge par le Serveur de Répartition (ii)

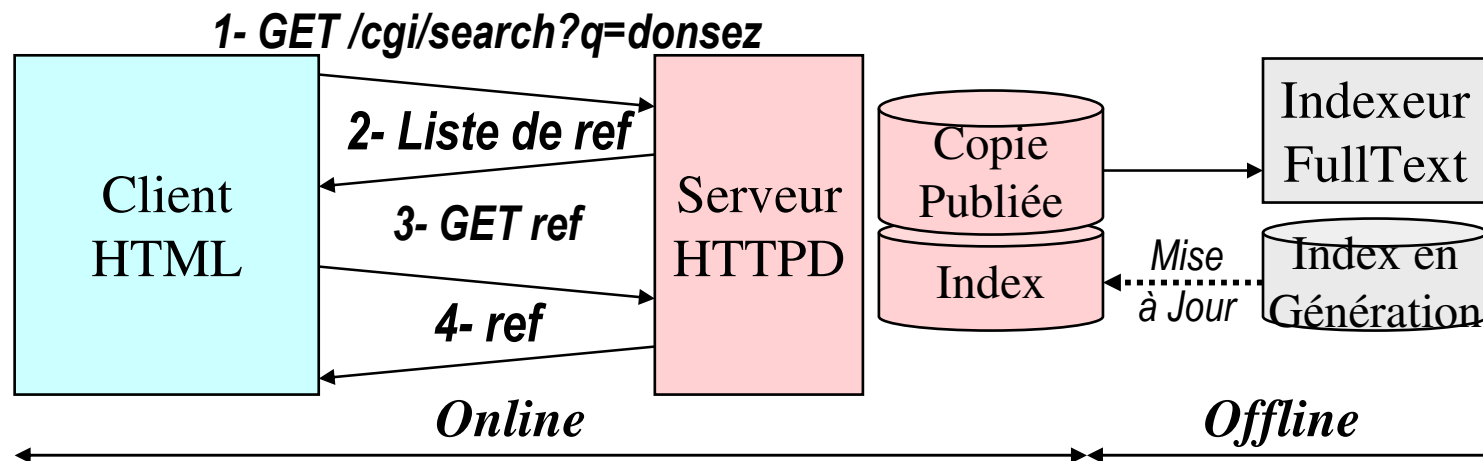
- Niveau 7 – Applicatif
  - Protocole HTTP
    - Détection du Cookie de la classe d'utilisateur dans une requête HTTP
    - Détection du Cookie de l'identifiant de session dans une requête HTTP
  - Protocoles FTP, SMTP, ...
  - Flux Audio/Vidéo, ...
- Rappel : Redirection (301) de la requête HTTP
  - Aisement intégrable à une webapp JavaEE grâce au `javax.servlet.Filter`
- En savoir plus
  - *Valeria Cardellini, Michele Colajanni, Philip S. Yu: Dynamic Load Balancing on Web-Server Systems. IEEE Internet Computing 3 (3): 28-39 (1999)*
  - *Valeria Cardellini, Emiliano Casalicchio, Michele Colajanni, Philip S. Yu: The state of the art in locally distributed Web-server systems. ACM Comput. Surv. 34 (2): 263-311 (2002)*

# Performance grâce aux caches

- 2 types de page
  - Dynamique
    - Le contenu varie à chaque requête
    - Le contenu peut rester constante pour un client au cours de sa session
  - Statique
    - Le contenu ne varie pas jusqu'à la prochaine modification
      - Exemple : Mise à jour des nouvelles du jour tous les matins à 7H00 GMT
- Les caches
  - dans les proxys et dans les navigateurs
  - utilise la date d'expiration du document
- Conseil
  - Pour éviter que le navigateur recharge le document inutile (ce qui charge le serveur), configurez les dates d'expiration des documents sur le serveur Web et dans les entêtes des réponses des scripts.

# Les Moteurs d'Indexation (1/2)

- L'utilisateur utilise deux modes de parcours d'un site
  - Le suivi de liens
  - La recherche Full-Text par combinaison de mots-clé et de concepts
    - (antoine NEAR(15) cléopatre) AND NOT césar
- Recherche Plein Texte (Full-Text)
  - pré-indexation du corpus documentaire
    - *Remarque : le corpus peut provenir du Web (i.e. Portails)*
    - *Problème : les documents générés à la volée*
  - le script /cgi-bin/search interroge l'index généré en Offline

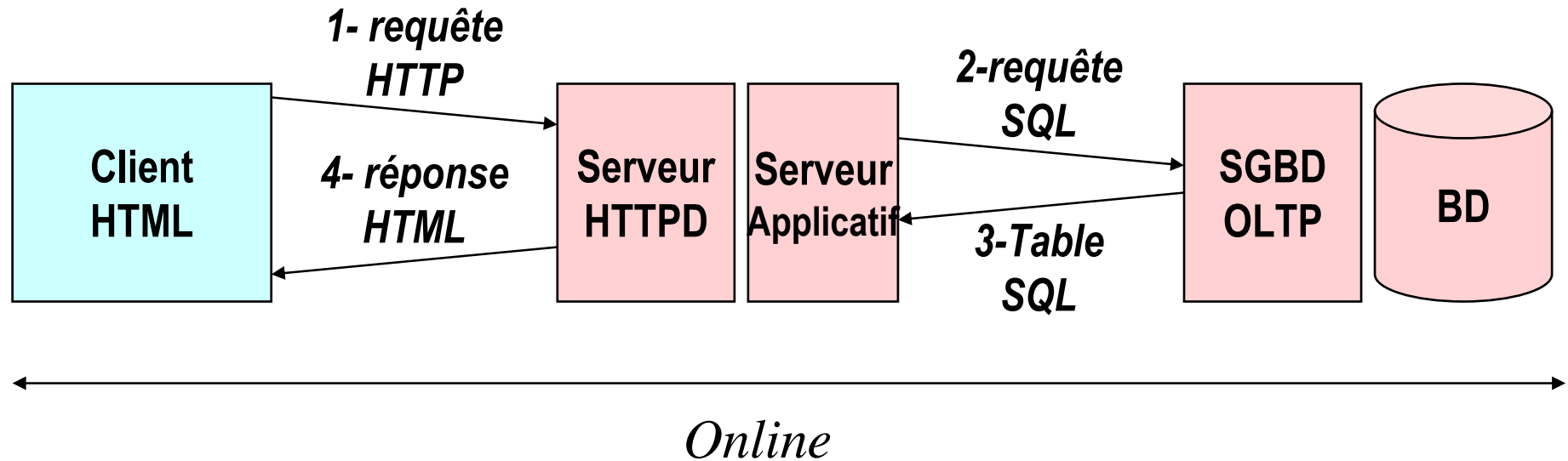


# Les Moteurs d'Indexation (2/2)

- Indexeurs (gratuit ou payant)
  - FreeWAIS, Glimpse, ht://Dig, Harvest, AltaVista, Verity Search, AIRS, Basis, Oracle Context, Doris-Floras, RetrievalWare, Virage, QBIC, Google, Jakarta lucene, ...
- Caractéristiques
  - Indexation incrémentale vs totale
    - l'ajout d'un document au corpus ne nécessite pas de reconstruire l'index
  - Types de document indexés
    - Texte, Texte structuré
    - Multimédia : Images, Sons, Vidéo (MPEG7), Monde 3D ...
  - Formats de document analysés
    - HTML, XML, PDF, Word/RTF, GIF, JPEG, MPEG, ...
- Bibliographie
  - C. Leloup, "Moteurs d'Indexation et de Recherche", Ed. Eyrolles, 1998, ISBN 2-212-08976-7
- Remarque: Google Desktop Search
  - indexation des volumes locaux, plugins tiers additionnables

# Les Antéserveurs (i)

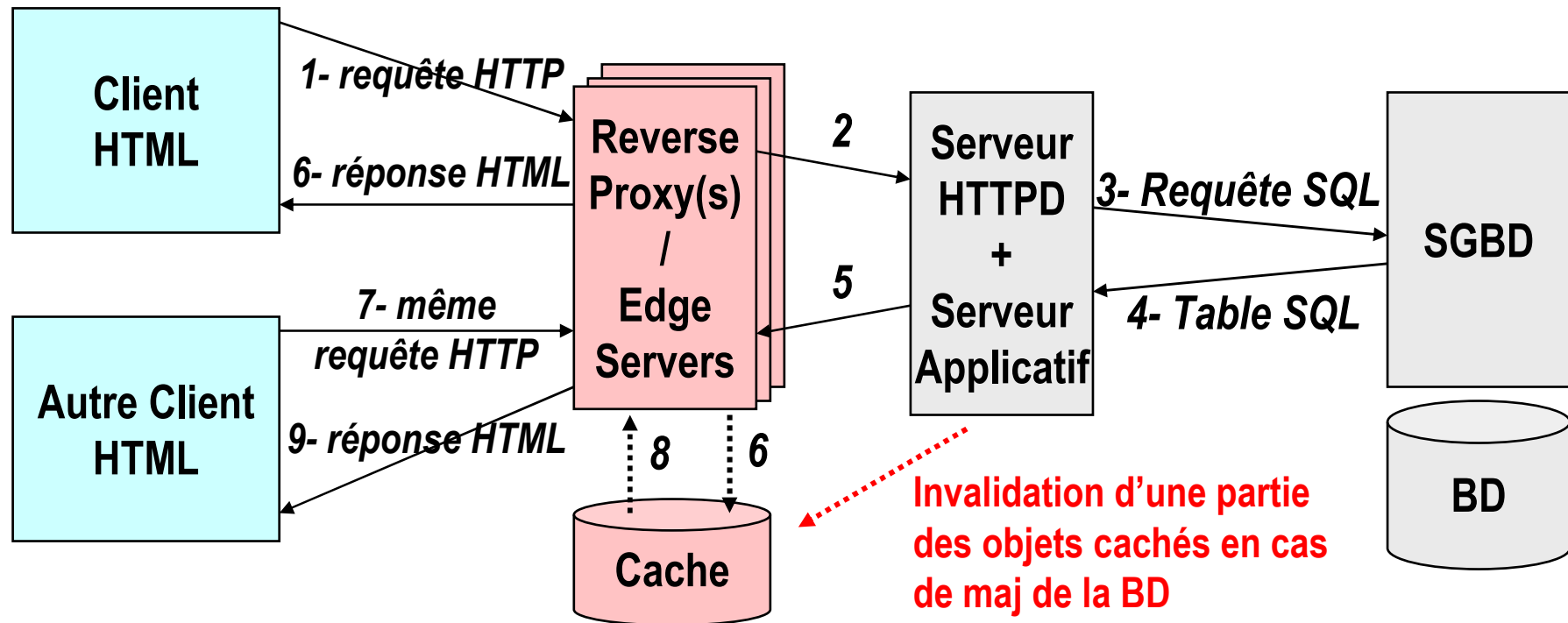
- Documents générés à la volée
  - nécessite de bonne performance du serveur HTTP et du serveur BD
  - Coût des requêtes SQL auprès du serveur BD



# Les Antéserveurs (ii)

- Cache des résultats (page HTML) des requêtes SQL fréquemment posées
  - Remarque : importance des informations de cache dans les entêtes de réponse HTTP

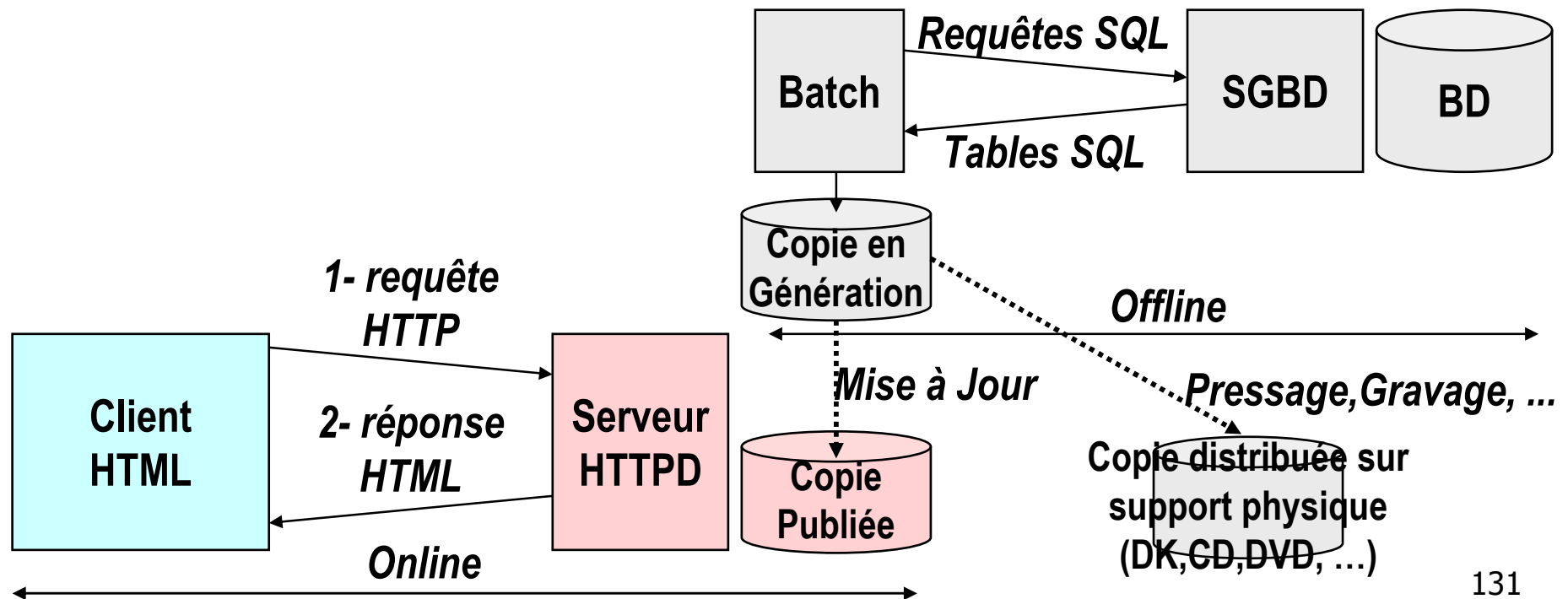
Terme:  
Edge Computing





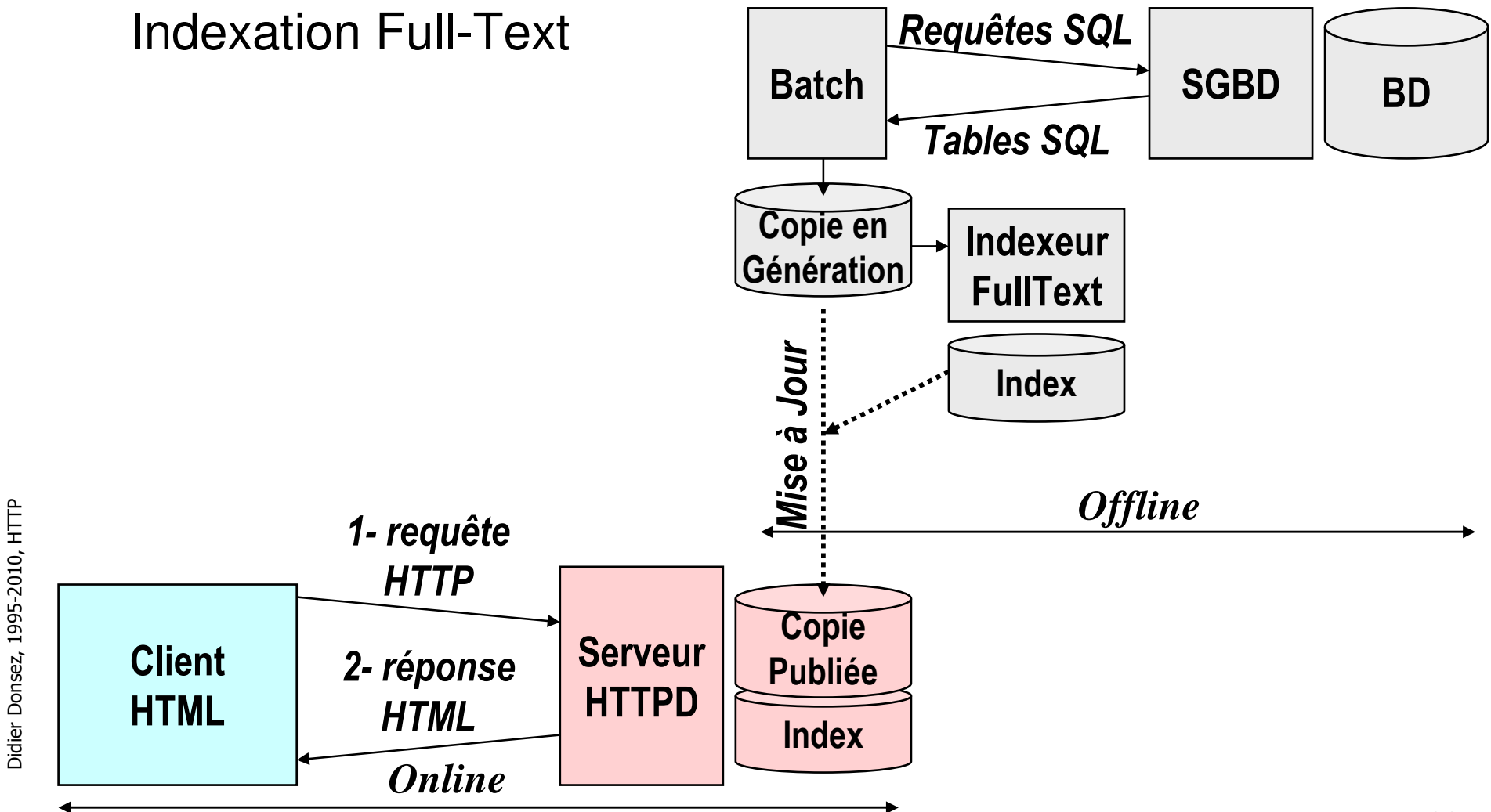
## Les Antéserveurs (iii)

- Génération à intervalle régulier d'une base documentaire à partir d'une base de données
  - Cas de données à faible variation et à caractère non transactionnelle
  - Exemple : Catalogue de Produits remis à jour une fois par jour



# Les Antéserveurs (iv)

- Génération et Indexation Full-Text



# Embedded Web Server (i)

- *Rappel : HTTP est un protocole très répandu et très outillé*

## ■ Motivation

- Dialoguer (piloter/administrer) avec des ordinateurs enfouis ou embarqués via HTTP
  - Réseaux
    - Routeurs, Coupleur réseaux domestiques (X10, Echelon, ...) ou industriels (ModBus, ...) ..
  - Electronic consuming
    - Téléphonie cellulaire, Répondeur domestique, ...
  - Domotique
    - Proxy Domotique pilotant l'alarme, le chauffage, Fri ...
  - Entertainment Network (UPnP) ...
    - TV, HIFI, Digital Camera, Consoles de Jeux, WiFi Toys ...

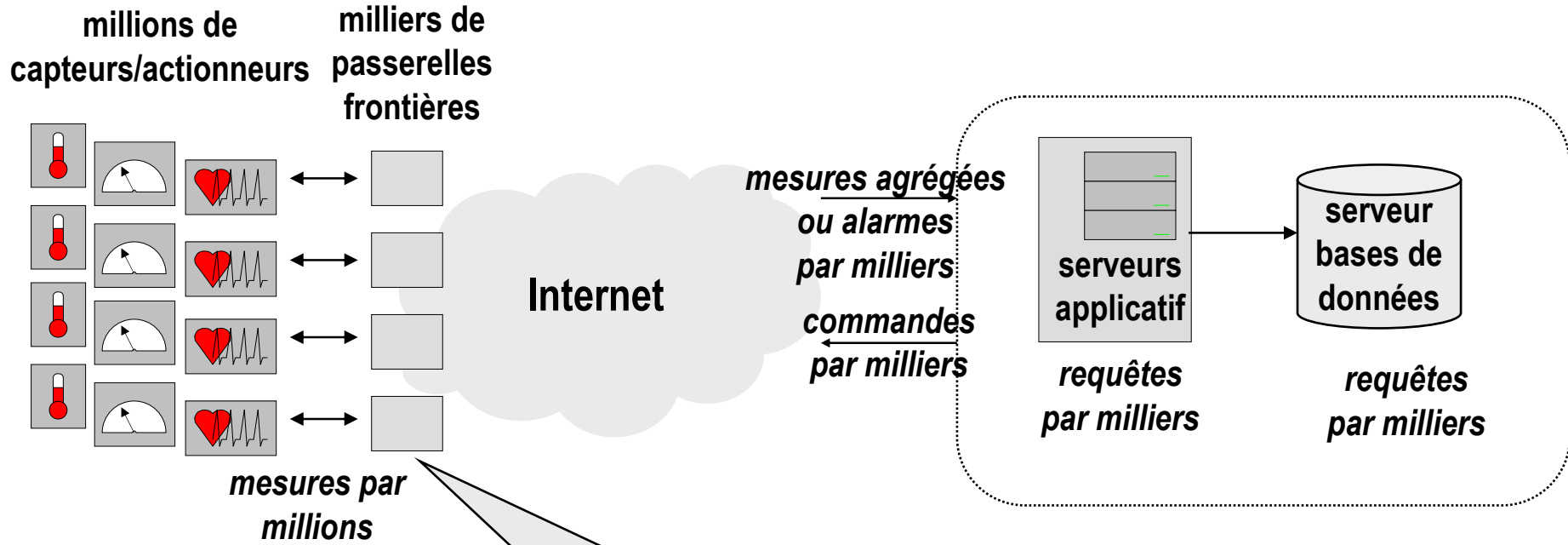


## Embedded Web Server (ii)

---

- Serveur HTTP
  - pour ordinateurs enfouis ou embarqués
    - Empreinte mémoire réduite, CPU
    - Passerelle OSGi ([www.osgi.org](http://www.osgi.org)) *voir cours sur OSGi*
      - Sun' JES (Java Embedded Server), OSCAR, ...
    - Brazil Project, Apache PicoServer, ACME micro\_httpd, TINIWebServer ...
  
- HTTP sur « Silicium »
  - Couches IP/TPC/HTTP directement incluses dans des DSP, ...

# Embedded Web Server (iii)



**Serveurs HTTP embarqués**

- passerelles domotiques,
- passerelles immotiques,
- passerelles véhiculaires,
- passerelles industriels, ...

# Serveurs Dédiés (ou *Server Appliances*)

---

## ■ Motivations

- S'oppose aux serveurs généralistes
- Configurations matérielles et logicielles dédiées aux services Web
  - OS: généralement un Linux « tuné », ...
  - HTTPD : Apache, ...
  - configuration Plug-and-Play et administration simple (console HTML, SMTP)
  - Enpilable, Rackable, ...
- Cibles : ISP et PME/PMI
  - Préconfiguration, configuration aisée, ...

## ■ Fonctionnalités

- HTTPD, Proxy, Cache, Firewall, Répartiteur de Charge, AntiVirus, Messagerie, ...
- Mais aussi SGBD (Oracle), MediaCaster, Indexeur (Google Appliances) ...

## ■ Acteurs

- Cobalt (racheté par Sun), RightVision, Akamai et HP, IBM, ...

# Clients HTTP dans le J2SE et J2ME

- Implémente directement un client HTTP
  - J2SE et J2ME
    - *java.net.URL*, *java.net.URLConnection*, *java.net.HttpURLConnection*
    - *javax.microedition.io.Connection*, *javax.microedition.io.HttpConnexion*
  - Jakarta HttpClient
  - Tache Jakarta Ant/Nant <get> (<post> <http://ant-contrib.sourceforge.net>)
- Fonctionnalités
  - Suivi de session (indépendance de *cookies* serveurs), multipart request, WebDav (PUT,DELETE, LOCK, ...), HTTP/SSL ...
- Quelques Usages
  - Robot de récupération, Injection de charge ...
  - Déploiement automatique (Ant,Maven,WebDav)
  - Tunneling HTTP, RPC sur HTTP (XMLRPC, SOAP/HTTP, JMX Adaptor)
- Voir le cours « Programmation Réseau en Java »

# HTTP dans HTML (i)

- Eléments META dans l'élément HEAD d'un document HTML 3.2

- `<META HTTP-EQUIV=name CONTENT=value>`

- équivalent à ajouter des Header 's
    - pour documents statiques

`<META HTTP-EQUIV="Expires" CONTENT=" Tue, 20 Aug 1996 14:25:27 GMT">`

`<META HTTP-EQUIV="Refresh" CONTENT="3; URL= http://newsite.mycomp.com ">`

`<META HTTP-EQUIV="Cache-Control" CONTENT="no-cache">`

`<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-1">`

`<META HTTP-EQUIV="Set-Cookie" content="cookie=set; path=/">`

- Exemple

`<HTML><HEAD>`

`<META HTTP-EQUIV="Refresh" CONTENT="3; URL= http://newsite.mycomp.com ">`

`</HEAD><BODY>`

`<H1> This site moves to <A HREF=" http://newsite.mycomp.com "> http://newsite.mycomp.com`

`</A></H1>`

`</BODY></HTML>`



## HTTP dans HTML (ii)

- Éléments META dans l'élément HEAD d'un document HTML 3.2

- `<META HTTP-EQUIV=name CONTENT=value>`

- équivalent à ajouter des Header 's
    - pour documents statiques

`<META HTTP-EQUIV="Expires" CONTENT=" Tue, 20 Aug 1996 14:25:27 GMT">`

`<META HTTP-EQUIV="Refresh" CONTENT="3; URL= http://newsite.mycomp.com ">`

`<META HTTP-EQUIV="Cache-Control" CONTENT="no-cache">`

`<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-1">`

`<META HTTP-EQUIV="Set-Cookie" content="cookie=set; path=/">`

- Exemple

`<HTML><HEAD>`

`<META HTTP-EQUIV="Refresh" CONTENT="3; URL= http://newsite.mycomp.com ">`

`</HEAD><BODY>`

`<H1> This site moves to <A HREF=" http://newsite.mycomp.com "> http://newsite.mycomp.com`

`</A></H1>`

`</BODY></HTML>`

## HTTP dans HTML (iii)

---

- **AJAX** (*Asynchronous JavaScript Technology and XML*)
  - **Objet Javascript XMLHttpRequest**
    - permet d'invoquer des requêtes GET ou POST
      - l'arbre DOM du résultat (XML ou HTML) peut être parcouru par le script
- **Exemple**
  - **TODO**

# WAP et WML

---

## ■ Motivation

- Adapter HTTP et HTML aux « handsets » nomades
  - Affichage limité, contrôle limité (i.e. clavier, pointage, ...)
  - Débit limité et coût de communication

## ■ WAP *Wireless Application Protocol*

- protocole HTTP « like » et « light »
  - Utilise les couches réseaux des réseaux cellulaires (à la place d'IP)
  - Utilise la sécurité des réseaux cellulaires

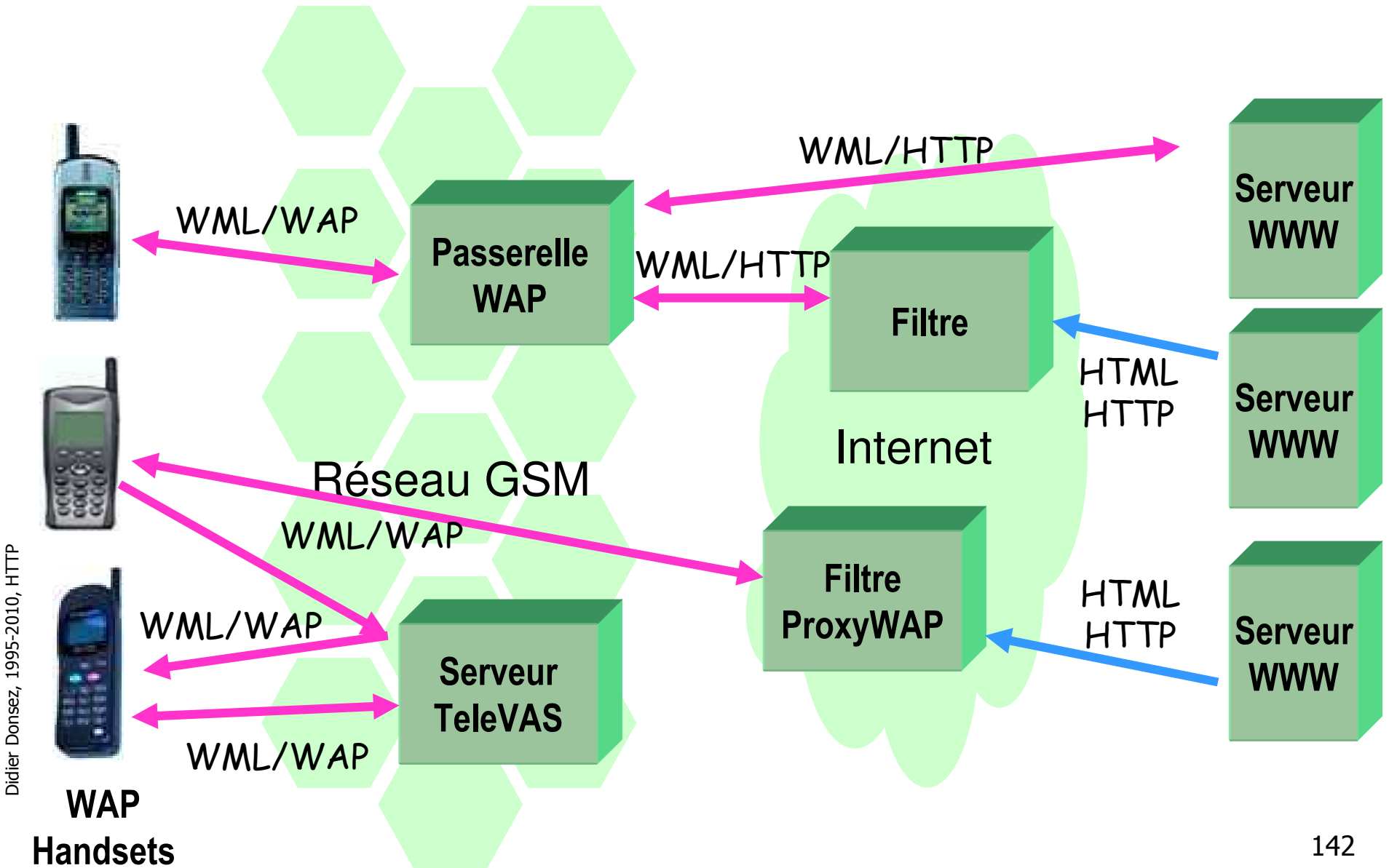
## ■ WML *Wireless Markup Language*

- langage hypertextuel adapté aux handsets nomades
  - basé sur XML/DTD : il existe un format plus compact (bytecodé)

## ■ voir *www.wapforum.org* et le cours sur WAP et WML

# WAP

## Architecture Réseau



Didier Donsez, 1995-2010, HTTP

# Faire du PUSH avec HTTP

- Motivation : le Push

- Une donnée sur le serveur change de valeur
- Le navigateur doit charger un document (généré) présentant la nouvelle valeur

- Problème

- Le serveur ne peut notifier le navigateur du changement

- Solution : le **Polling**

1. Le navigateur recharge le document régulièrement

- Inconvénient: le réaffichage du document perturbe l'utilisateur même si rien de nouveau arrive

- `<META HTTP-EQUIV="Refresh" CONTENT="5">`

2. Une applet nichée scrute régulièrement auprès du serveur si la valeur à changer et provoque le rechargement de la page en cas de changement

Remarque: la valeur du **Refresh** peut être adaptée au cours du temps

# Constrained Application Protocol (CoAP)

- Motivation : HTTP/REST-like protocol for constrained sensors
  - 128KB FlashRAM and 4KB RAM
  - Battery consumption (sleep and periodical wakeup)
- Interaction
  - Request-response
  - Subscribe-Notify
- REST (READ, WRITE, CREATE, DESTROY)
- Resource discovery
  - /profile URI multicast + DISCOVER method
- Protocol binding
  - UDP and UDP Multicast (16-bit sequence number for reliability)
  - Optionally TCP without “stop and wait”
- Caching
  - Important since sleeping mode
  - CoAP proxy (for subscription ...)
- See <http://www.ietf.org/id/draft-shelby-6lowapp-coap-00.txt>

# Web 2.0 (TODO)

---

- Motivations
  - Composer l'interface de plusieurs sources fournisseurs
  - Rendre les interfaces riches et fluides

# Technologies RIA (Rich Internet Application)

---

## ■ Motivations

- Constat : taux élevé des abandons lors d'une navigation sur les sites de e-Commerce
- Raisons (d'après une étude de Global Millennia Marketing)
  - High shipping cost (69%)
  - Change of mind (61%)
  - High total cost (49%)
  - **Long checkout process (41%)**
  - **Poor site navigation and long download times (31%)**
  - **Confusing checkout process (27%)**

## ■ Solutions

- DHTML/AJAX, Flash, OpenLaszlo, ... + RESTFul, JSON, ...



# The Bayeux protocol

---

- Motivation
  - « bi-directional » HTTP-based messaging for RIA
    - Operation invocation RIA → server
      - data source, RESTful, SOAP, ...
    - Notification Server → RIA
      - state change, alert, completed Future object ...
- Principe
  - Pull
  - Polling
- Implementations
  - JavaScript : Dojo'cometd
  - ...

# Bibliographie

---

- Beaucoup de Guides, Tutoriels, Manuels
  - <http://ds.internic.net/rfc/rfc2068.txt>
  - <http://www.w3.org>
  - [http://fr.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](http://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol)
- Des livres
  - *Attention, ca change très vite !*
  - *La traduction en français a au moins 1 an de retard sur la version anglaise*

## Bibliographie - HTTP

---

- Stephen Spainhour & Robert Eckstein, « Webmaster in a Nutshell », 2nd Edition, June 1999 (est.), ISBN 1-56592-325-1, Ed : O'Reilly
  - très complet
- Clinton Wong, « Programmation de clients Web avec Perl », Ed Oreilly, 1997, ISBN 2-84177-050-8
  - *l'automatisation de requêtes sur HTTP (en Perl)* Revues
- Duane Wessels , Web Caching, June 2001, Oreilly, 1-56592-536-X
- Patrick Killelea , Web Performance Tuning, 1st Edition October 1998, ISBN 1-56592-379-0
- Revue IEEE Internet Computing <http://www.computer.org/internet>
- Revue IEEE IT Professional <http://www.computer.org/itpro>

# Bibliographie - Serveurs

- Jason Brittain, Ian F. Darwin, "Tomcat: The Definitive Guide", Ed O'Reilly, June 2003, ISBN: 0-596-00318-8

## Bibliographie - Autres

---

- Robert Orfali, Dan Harkey, Jeri Edwards, « Client/Server Survival Guide », 3rd edition, February 1999, Ed John Wiley & Sons; ISBN: 0471316156
  - information générale sur les composants d'un SI
- Louis Rosenfeld, Peter Morville , "Information Architecture for the World Wide Web - Designing Large-scale Web Sites", 1st Edition February 1998, ISBN 1-56592-282-4, O'Reilly, 226 pages, \$24.95
- C. Leloup, "Moteurs d'Indexation et de Recherche", Ed. Eyrolles, 1998, ISBN 2-212-08976-7

# Bonus Track

---