

IUP Miage 3ème année AR - Apache

Pascal AUBRY

IFSIC - Université de Rennes 1

<http://perso.ifsic.univ-rennes1.fr/aubry>

Le World Wide Web

- Système d'information hyper-média réparti
 - texte, images, images animées, son, vidéo ...
- Informations stockées sur les serveurs
 - www.jres.org, www.ifsic.univ-rennes1.fr, www.ulaval.ca
- Interrogation des serveurs
 - tels que Apache, IIS, Netscape, Roxen, AOL, iPlanet, Lotus Domino, NCSA httpd, goAhead, Tomcat
 - par clients tels que Netscape, Mosaic, Internet Explorer, Lynx, Opera, Amaya, Chrome, Firefox

Les principes du Web

- Modèle client-serveur
- Le client envoie des requêtes au serveur
 - transfert de fichiers
 - exécution de programmes sur le serveur
 - mise à jour de fichiers
 - ...
- Objets manipulés repérés par leur URL
- Utilisation du protocole HTTP

Le protocole HTTP

- Définit le langage utilisé pour les échanges entre client et serveur Web
 - version 0.9
 - simple protocole de transfert de données (GET et réponse)
 - version 1.0
 - restée un Internet Draft (RFC 1945)
 - actuellement version 1.1
 - RFC 2616 (juin 1999)
- Pas de session permanente entre client/serveur

Déroulement d'une requête HTTP



- Demande de connexion
- Attente de la réponse du serveur
- Établissement de la connexion
- Envoi d'une requête (URL)
- Réponse du serveur
- Affichage de la réponse
- Fermeture de la connexion

Déroulement d'une requête HTTP



- Demande de connexion
- Attente de la réponse du serveur
- Établissement de la connexion
- Envoi d'une requête (URL)
- Réponse du serveur
- Affichage de la réponse
- Fermeture de la connexion

Exemple de transaction HTTP

```
% telnet www.ifsic.univ-rennes1.fr 80 .....connexion au serveur web
```

Exemple de transaction HTTP

```
% telnet www.ifsic.univ-rennes1.fr 80 .....connexion au serveur web  
Trying 148.60.4.30...  
Connected to apollon.univ-rennes1.fr.  
Escape character is '^['.
```

Exemple de transaction HTTP

```
% telnet www.ifsic.univ-rennes1.fr 80 .....connexion au serveur web
Trying 148.60.4.30...
Connected to apollon.univ-rennes1.fr.
Escape character is '^]'.
GET /index.html HTTP/1.1 ..... demande de transfert
Host: apollon.univ-rennes1.fr ..... nom du serveur
From: pa@ifsic.univ-rennes1.fr ..... adresse demandeur (optionnelle)
(ligne blanche = fin de l'entête HTTP de la requête)
```

Exemple de transaction HTTP

```
% telnet www.ifsic.univ-rennes1.fr 80 .....connexion au serveur web
Trying 148.60.4.30...
Connected to apollon.univ-rennes1.fr.
Escape character is '^]'.
GET /index.html HTTP/1.1 ..... demande de transfert
Host: apollon.univ-rennes1.fr ..... nom du serveur
From: pa@ifsic.univ-rennes1.fr ..... adresse demandeur (optionnelle)
(ligne blanche = fin de l'entête HTTP de la requête)

HTTP/1.1 200 OK ..... réponse du serveur
Date: Tue, 02 Jun 2001 14:11:17 GMT
Server: Apache/1.3b6
Last-Modified: Mon, 07 Apr 2001 10:39:08 GMT ..... informations sur la ressource
ETag: "b3dd-524-33b78ccc"
Content-Length: 1316 ..... taille de la ressource
Accept-Ranges: bytes
Content-Type: text/html ..... type MIME
(ligne blanche = fin de l'entête HTTP de la réponse)

<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> (contenu)
<HTML>
...
</HTML>
```

Exemple de transaction HTTP

```
% telnet www.ifsic.univ-rennes1.fr 80 .....connexion au serveur web
Trying 148.60.4.30...
Connected to apollon.univ-rennes1.fr.
Escape character is '^]'.
GET /index.html HTTP/1.1 .....demande de transfert
Host: apollon.univ-rennes1.fr .....nom du serveur
From: pa@ifsic.univ-rennes1.fr .....adresse demandeur (optionnelle)
(ligne blanche = fin de l'entête HTTP de la requête)

HTTP/1.1 200 OK .....réponse du serveur
Date: Tue, 02 Jun 2001 14:11:17 GMT
Server: Apache/1.3b6
Last-Modified: Mon, 07 Apr 2001 10:39:08 GMT .....informations sur la ressource
ETag: "b3dd-524-33b78ccc"
Content-Length: 1316 .....taille de la ressource
Accept-Ranges: bytes
Content-Type: text/html .....type MIME
(ligne blanche = fin de l'entête HTTP de la réponse)
<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> (contenu)
<HTML>
...
</HTML>
Connection closed by foreign host. ....fermeture de la connexion
```

Les nouveautés de HTTP 1.1

- Identification obligatoire du *hostname* par le client
 - « *virtual hosting* »
- Négociation du contenu
 - documents français/anglais, HTML/PDF
- Connexions persistantes
 - amélioration temps de chargement de pages avec images
- Prise en charge des mandataires (*proxies*)
- Nouveaux codes réponse
- Nouvelles méthodes

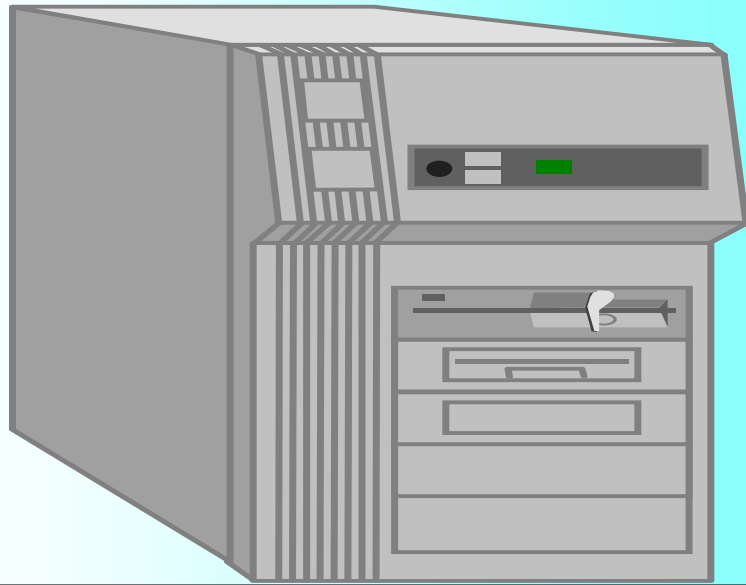
Les méthodes HTTP

- Demander une ressource
 - **GET** : rapatrier une ressource
 - **POST** : envoyer des données et rapatrier une ressource
- Avoir de l'information sur une ressource
 - **HEAD** : connaître ses caractéristiques
 - **OPTIONS** : connaître les options qui lui sont applicables
- Mettre à jour une ressource à distance
 - **PUT** : la créer ou remplacer son contenu
 - **DELETE** : la détruire
- Déboguer
 - **TRACE** : tracer les mandataires (*proxies*)

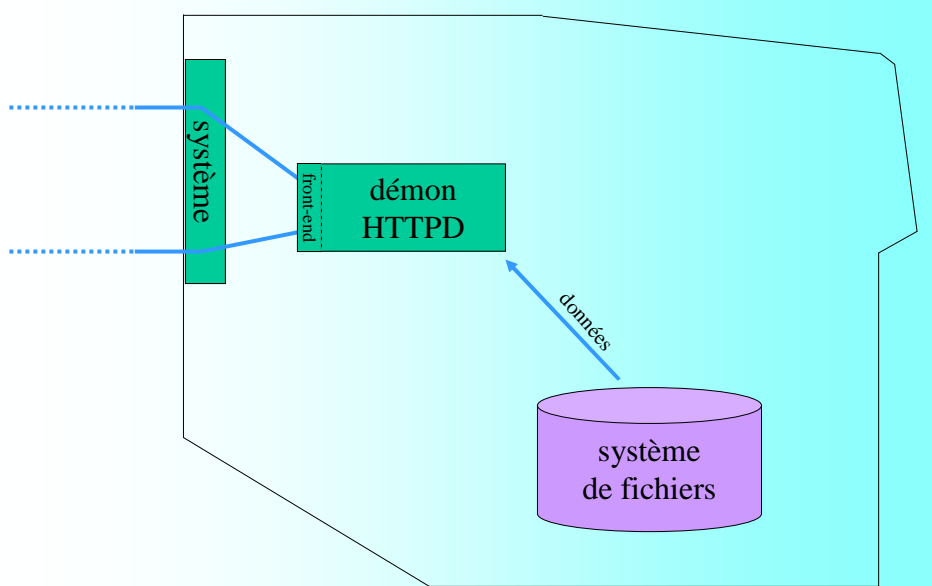
Les codes de réponse sous HTTP

- | | |
|---------------------------|---------------------------|
| • 10x : information | |
| • 20x : succès | 200 OK |
| • 30x : redirection HTTP | 301 Moved permanently |
| • 40x : erreur du client | 401 Unauthorized |
| | 402 Payment required |
| | 403 Forbidden |
| | 404 Not found |
| • 50x : erreur du serveur | 500 Internal server error |

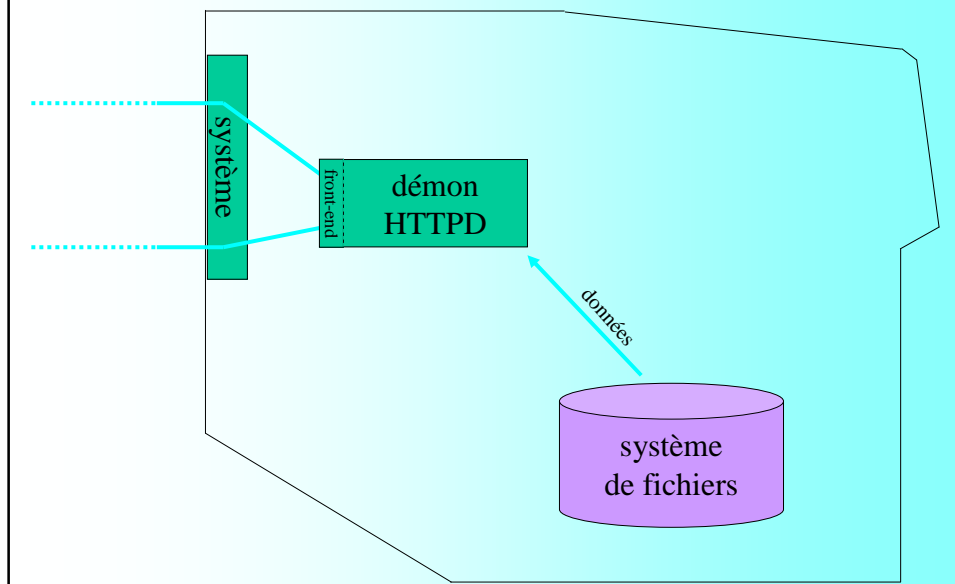
Délivrer un document statique



Délivrer un document statique



Délivrer un document statique



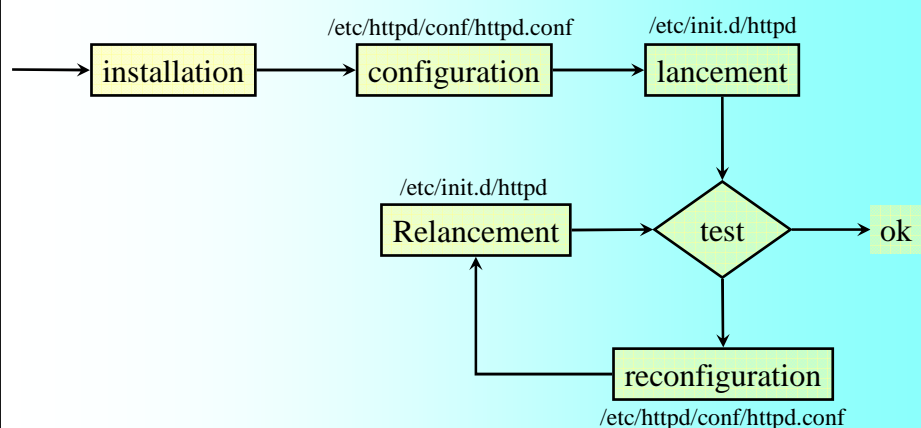
Le serveur Apache

- Le serveur HTTP le plus utilisé
- Une approche modulaire
- Portabilité
- Performance
- Un outil libre

Les fichiers du paquetage Apache

- `/usr/bin/httpd` : le démon (binaire)
 - ne pas manipuler directement
- `/etc/init.d/httpd` : le script de gestion du démon
 - start, stop, restart, reload, status
- `/etc/httpd/conf/httpd.conf` : le fichier de configuration
- `/var/www/*` : les fichiers servis par le serveur

La mise en œuvre d'un service (HTTP en particulier)



Apache : mise à disposition de documents

- Le serveur web fait la correspondance entre :
 - une URL (`http://serveur/rep/fichier.html`)
 - un fichier du serveur (`/var/www/html/rep/fichier.html`)
- DocumentRoot
 - définit la racine du serveur
- Alias
 - définissent des chemins supplémentaires pour les URLs
- DirectoryIndex
 - définissent les fichiers qui seront lus par défaut lorsque l'URL correspond à un répertoire (et non un fichier)

Types de documents

- Le serveur envoie le type (MIME) du document au serveur selon :
 - l'extension du fichier demandé (par défaut)
 - le contenu du fichier (moins performant)
- Le client décide, selon le type MIME renvoyé par le client, ce qu'il fait de la réponse :
 - affichage dans la fenêtre du navigateur (avec ou sans interprétation des tags HTML)
 - sauvegarde
 - lancement d'une application externe ou d'un plug-in

Apache : protection des données

- Par fichier, avec les droits du système de fichier
 - l'utilisateur qui exécute le démon httpd doit être capable de lire le contenu des fichiers
- Par répertoire, en configurant Apache
 - dans tous les cas
 - selon le client
 - selon l'utilisateur (authentification HTTP)
 - en mixant les contrôles
- Les protections s'additionnent

Pourquoi développer sur le web ?

- Universalité du protocole HTTP
 - indépendance vis-à-vis des clients
 - pérennité des applications
- Simplicité de l'interface
- Parce que ça fait moderne ;-)

Web dynamique : qui fait quoi ?

- Le serveur exécute, le client reçoit
 - SSI, XSSI, CGI, FastCGI, PHP, ASP.Net, JSP(Servlet),JSF
 - indépendance vis-à-vis du client (navigateur)
 - interactivité limitée
- Le serveur envoie, le client exécute
 - JavaScript embarqué (DHTML), Applet Java
 - dépendance vis-à-vis du client
 - plus d'interactivité

Pages dynamiques : solutions côté serveur

Les premiers pas
Le protocole CGI
PHP et les autres
(FastCGI, Perl, PHP, Servlets, JSP, ASP)

SSI, XSSI :

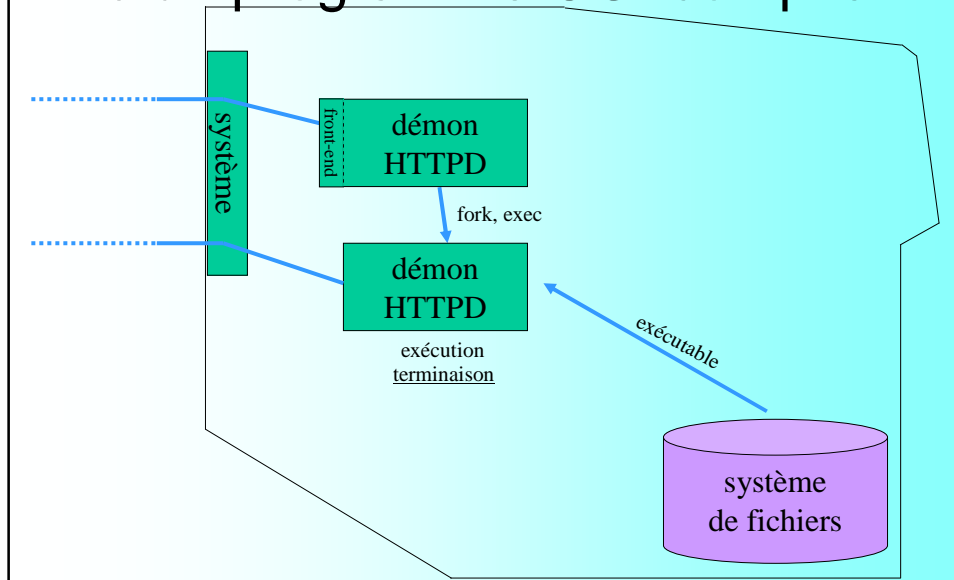
le premier pas vers la dynamique

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<head><title>Exemple XSSI</title></head>
<body bgcolor="#FFFFFF">
  <!--#config errmsg="erreur de syntaxe dans un (x)SSI" -->
  <!--#set var="x" value="test"--><!--#echo var="x"--><br>
  Fichier modifié le : <!--#echo var="LAST_MODIFIED" --> <br>
  Nom du serveur : <!--#echo var="SERVER_NAME" --> <br>
  <!--#if expr="$HTTP_USER_AGENT = 'Mozilla/2.0 (compatible;
  MSIE 3.01; Windows 95)'" -->
    Il est temps de se mettre à jour !
  <!--#endif -->
</body>
</html>
```

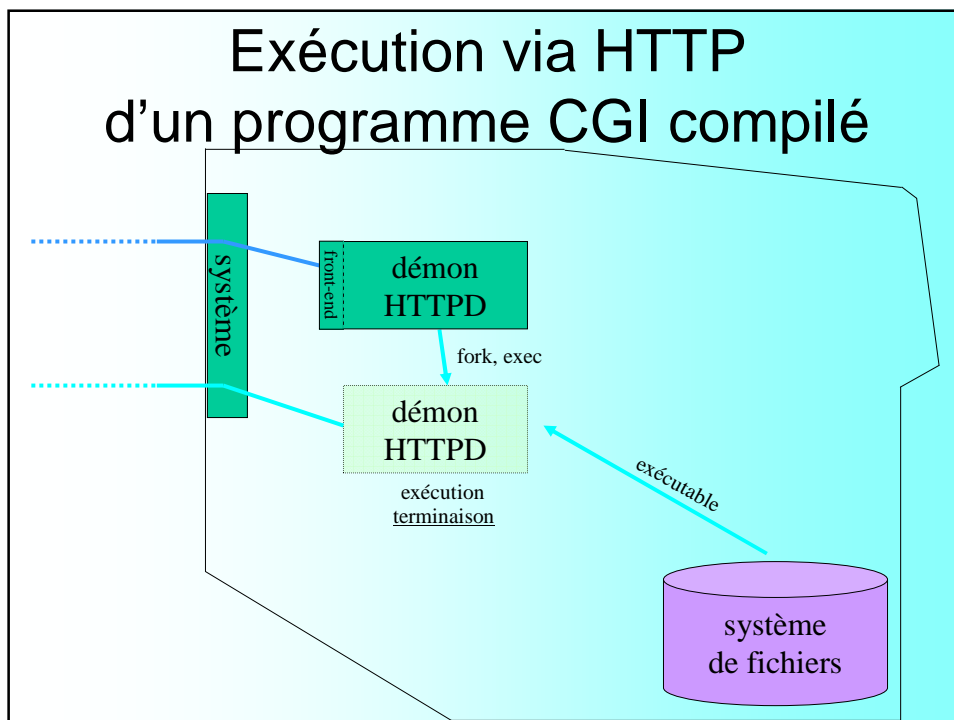
CGI : Common Gateway Interface

- Un standard pour l'interface entre applications et serveurs d'informations
- Permet de passer des paramètres aux requêtes
 - dans l'URL avec la méthode GET
<http://serv.dom.org/cgi-bin/script?arg1=val1&arg2=val2>
 - Variable CGI correspondante : QUERY_STRING
 - Caché avec les données avec la méthode POST
- Exécution d'un programme sur le serveur
 - Les informations renvoyées au client sont statiques
 - Des requêtes successives permettent le dynamisme

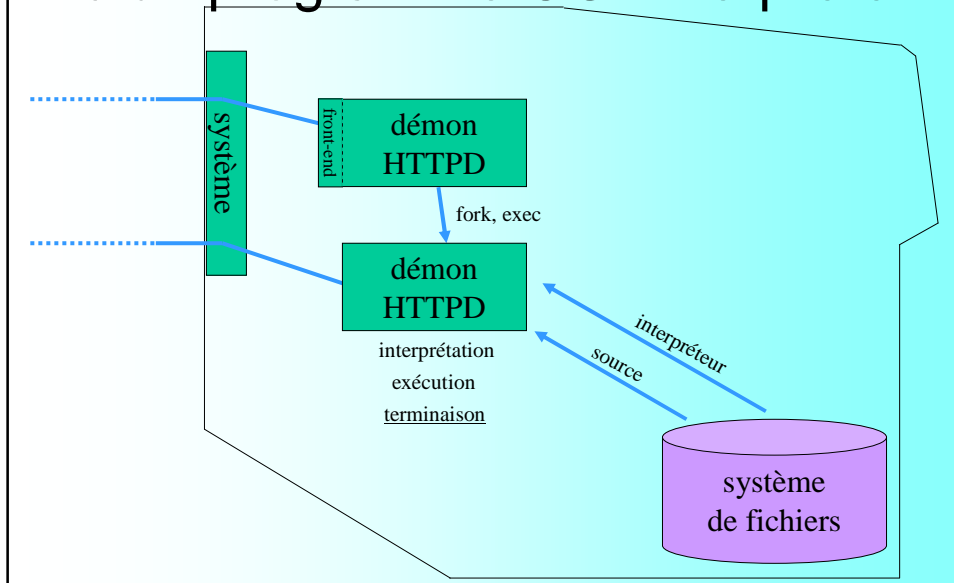
Exécution via HTTP d'un programme CGI compilé



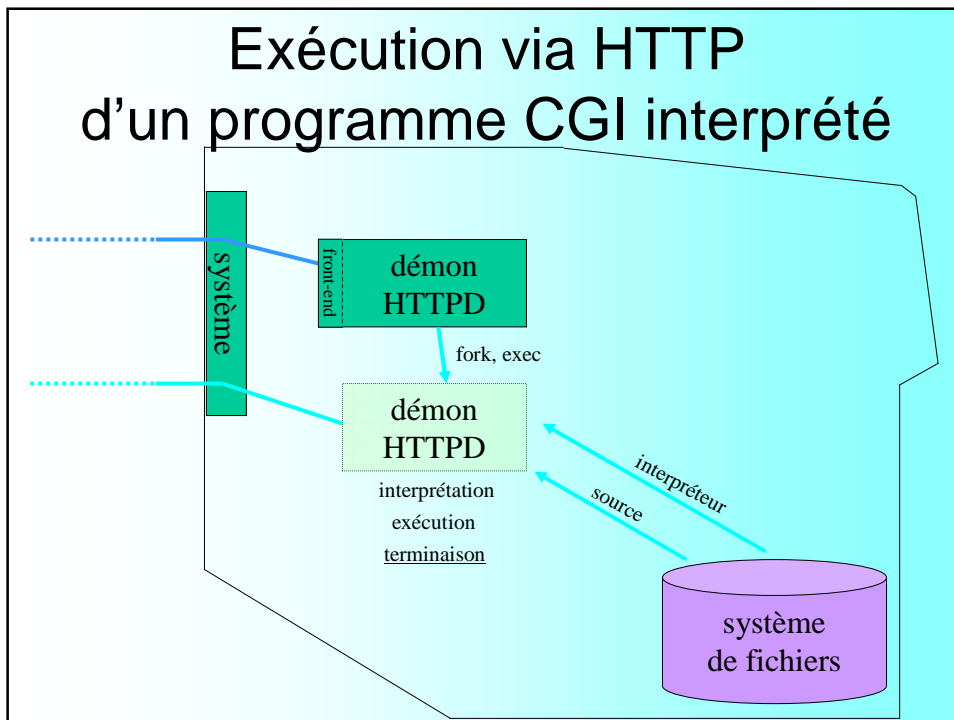
Exécution via HTTP d'un programme CGI compilé



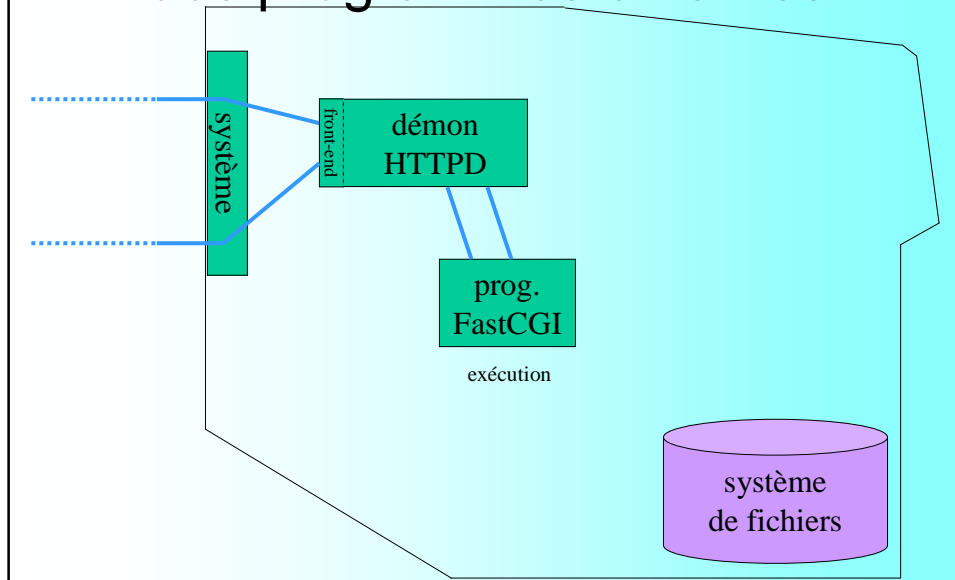
Exécution via HTTP d'un programme CGI interprété



Exécution via HTTP d'un programme CGI interprété



FastCGI : une solution au coût des programmes externes



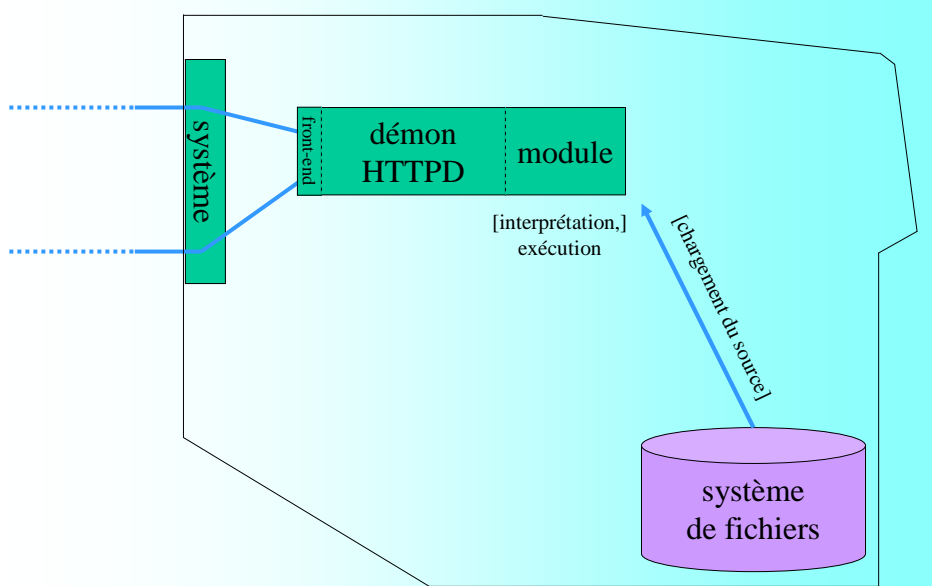
FastCGI

- Performance, simplicité (comme CGI)
- Migration facile (de l'existant CGI)
- Isolation du démon httpd
 - sécurité en cas de crash
 - Indépendance de l'architecture du serveur
- Distribution possible de la charge

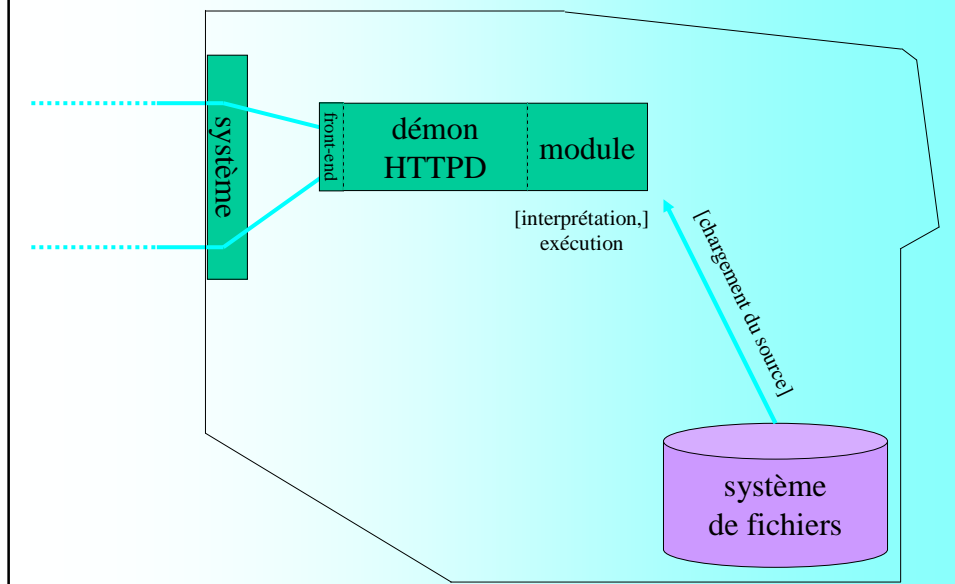
Scripts côté serveur : les quatre leaders du marché

- Perl
- PHP (Php Hypertext Preprocessor)
- JSP (Java Server Pages)
- ASP (Active Server Pages)

L'approche modulaire (PHP, Perl)



L'approche modulaire (PHP, Perl)



Perl

- Practical Extraction and Report Language
- Ressources illimitées
- Module d'Apache (mod_perl) ou stand-alone
- Multi-plateformes
- On aime ou on n'aime pas ;-)

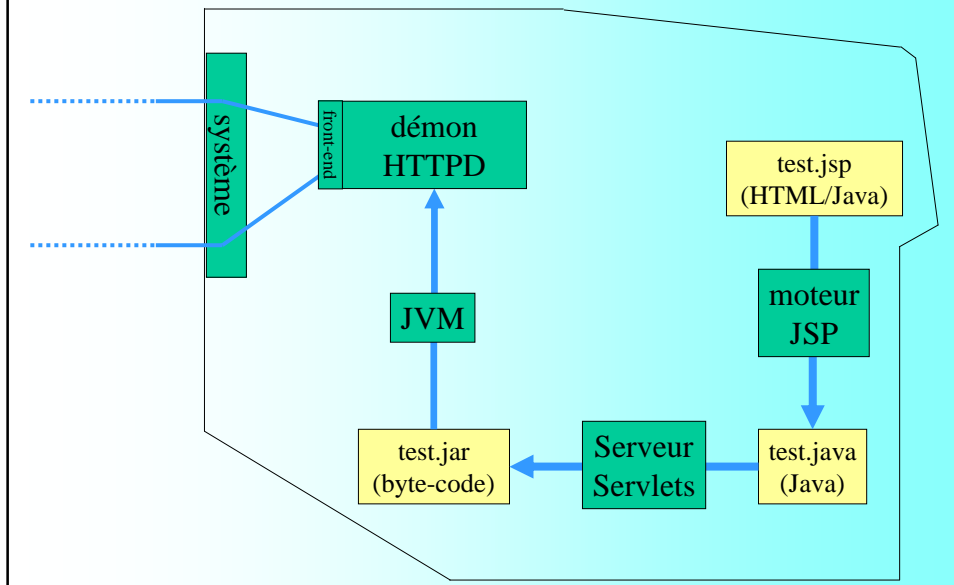
PHP

- Le plus simple
- Multi-plateformes
- Un langage non généraliste (comme Java et Perl) mais néanmoins très riche (bibliothèques)
- Un produit libre

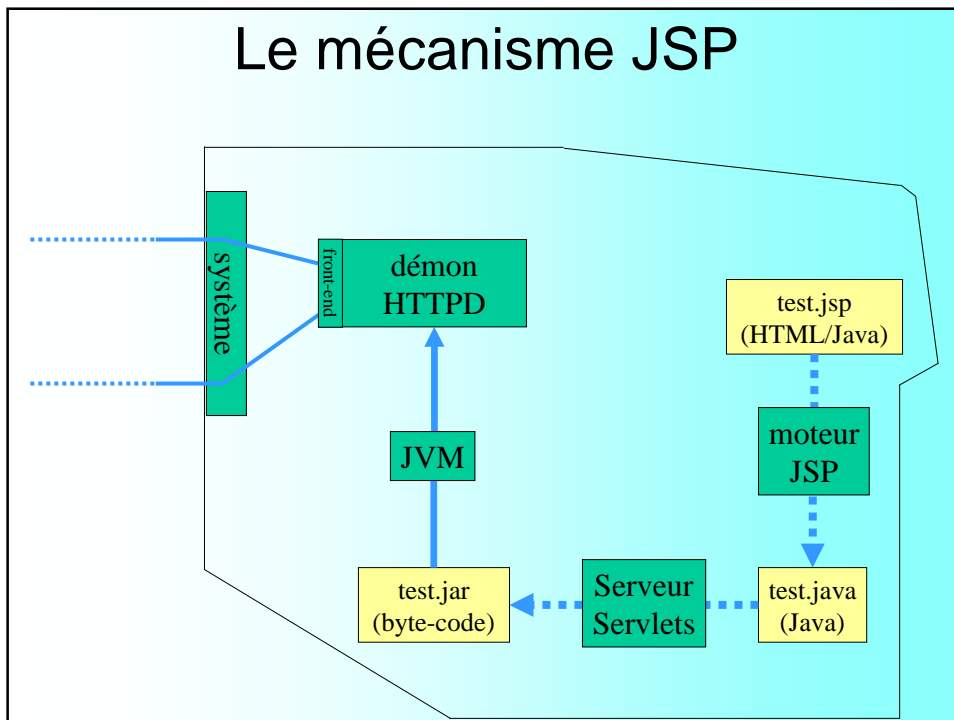
Java

- Les servlets
 - Programme Java exécuté côté serveur
 - Transformation en byte-code avant exécution
 - Servies par un serveur dédié (ex : Tomcat)
- JSP (Java Server Pages)
 - Pages HTML avec Java embarqué
 - Transformation en Servlets

Le mécanisme JSP



Le mécanisme JSP



ASP

- Tourne nativement sur IIS (MicroSoft)
- JavaScript ou VBScript
- ASP+ :
 - support langages compilés (VB, C++, C#)
 - compilation intermédiaire JS et VBS
 - mécanisme de cache des objets compilés
 - configuration format XML
 - contrôle accru des formulaires
- portabilité limitée (www.chilisoft.com)

Les langages de CGI

- Les shells
 - sh, tcsh, bash
- Les langages compilés
 - C, C++, Pascal, ...
- Perl
- Java (JSP, Servlets)
- PHP
- Python
- VBScript, JavaScript (ASP)

Apache et les programmes CGI

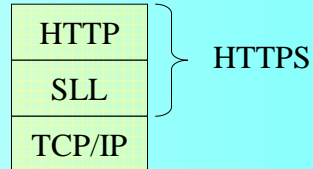
- À l'aide du module mod_cgi
- Directive ScriptAlias
- Problèmes de sécurité
 - groupement de tous les programmes CGI
 - Pour limiter la casse : suExec, CgiWrap

Hôtes virtuels (« virtual hosting »)

- Possibilité pour un serveur de se faire passer pour plusieurs machines (virtuelles)
- La(es) adresse(s) IP des machines virtuelles doivent être connues des clients
 - résolution de noms par DNS, NIS, WINS, ...
- Configuration de Apache :
 - NameVirtualHost, <VirtualHost>
- depuis HTTP 1.1

Le cryptage de HTTP (HTTPS)

- SSL : Secured Socket Layout
- HTTPS : HTTP sur SSL
- Objectifs de HTTPS :
 - crypter les communications
 - identifier le serveur
 - le serveur auquel je suis connecté est-il le bon ?
 - Identifier le client
 - la personne qui se connecte au serveur est-elle la bonne ?
- Certificats (X509)



Un site web : des responsabilités, des métiers

- Technique (ingénieur)
 - fonctionnement du service
 - mise à disposition d'outils
 - développement
- Éditoriale (rédacteur)
 - mise à jour du contenu
- Graphique (infographiste)
 - présentation du site
- Pénale (juriste)

Pourquoi un environnement de développement ?

- Pour gérer l'accès concurrent
 - les droits sur les systèmes de fichiers sont insuffisants
 - on utilise souvent une base de données et/ou une interface web
- Pour séparer les tâches
 - graphisme, interfaçage Homme/Machine
 - développement
 - mise à jour du contenu
- Pour intégrer plusieurs technologies
 - COM, CORBA, EJB, C/C++, Java, XML

Les environnements de développement

- BEA WebLogic (Java)
- Lutris Enhydra (Java, open source)
- Netscape Application Server
- Jbuilder (Java)
- Zope (Python)
- Midgard (PHP, open source)
- Allaire/Macromedia ColdFusion 5 (CFML)
- Vignette Content Suite V6
- ...

Une comparaison rapide...

	[Java VB]Script (+ASP)	Java (+JSP)	Perl (+mod_perl)	PHP	Python (+Zope)
Installation	★★★★★	★★	★★★★★	★★★★★	★★★★★
Apprentissage	★★★	★★	★	★★★★★	★★★★★
Puissance	★★★	★★★★★	★★★	★★★	★★★
Portabilité	★	★★★★★	★★★★★	★★★	★★★
Outils	★★★★★	★★★★★	★★	★★★	★★★
Ressources	★★★	★★★★★	★★★★★	★★★★★	★★★

...et subjective !

Comment choisir ?

- 1. Évaluer le ou les projets à mettre en œuvre
- 2. Évaluer ses moyens
 - humains
 - financiers
- 3. Tenir compte de l'existant
 - les habitudes
 - les compétences
 - les préférences
- 4. Se faire soi-même une idée des produits