

<http://www-adele.imag.fr/users/Didier.Donsez/cours>

Le Scripting Serveur

Didier Donsez

Université Joseph Fourier - Grenoble 1

PolyTech'Grenoble - LIG/ADELE

`Didier.Donsez@imag.fr`

`Didier.Donsez@ieee.fr`

Au sommaire

- Répartition Client-Serveur
- Architecture 3 Tiers
- CGI - Common Gateway Interface
- ISAPI et NSAPI
- Servlets
- SSI - Server Side Include
- Server Side Script (ASP, JSP, ...)

Répartition Client-Serveur

- Traitement coté client
 - extériorisation de l'information
 - formats visualisables par le client
 - interactivité non lié au réseau
- Traitement coté serveur
 - serveur applicatif
 - conserve les traitements en interne (business logic)
 - adapté le document retourné
au capacité du visualisateur
 - prise en compte des champs accept:
communiqués par le client
 - charge le serveur

Le Scripting Serveur

■ Utilisation

- Traitement des formulaires par le serveur
- Génération de pages HTML en réponse
 - requête sur des bases de données, EJB, ...

■ Les Techniques

- Script générant du HTML
 - CGI : *Common Gateway Interface*, Fast-CGI
 - NSAPI, ISAPI, **Servlets**
- Script embarqué dans des pages HTML
 - SSI : *Server Side Include*
 - SSS : *Server Side Script*
 - Active Server Pages (MicroSoft), **JSP** (Sun), PHP ...
 - ESI : *Edge Side Include*

Les Scripts CGI

(Common Gateway Interface)

- Exécutable du serveur invoqué sur réception les méthodes GET, POST, ...
 - en général pour le traitement de contenu d'un formulaire
 - retourne une réponse
 - au format HTML en général
 - défini par `Content-Type` : *code-mime*
- Paramètres de la requête
 - Variables d'environnement et STDIN
 - le serveur configure des variables d'environnement avant le lancer le script
 - Contenu du formulaire : les Paires Name-Value
 - dans `$QUERY_STRING` `si $METHOD==GET`
 - sur l'entrée standard `STDIN` `si $METHOD==POST`

Encodage des paires Name-Value

- Chaque entrée du formulaire soumis est encodé sous la forme d'une paire
nom de l'entrée (INPUT) = valeur de l'entrée
- Chaque paire est séparé par un &

nom=Dupont+Jean&adresse=3+rue+de+la+Gait%e9%0a75014+Paris

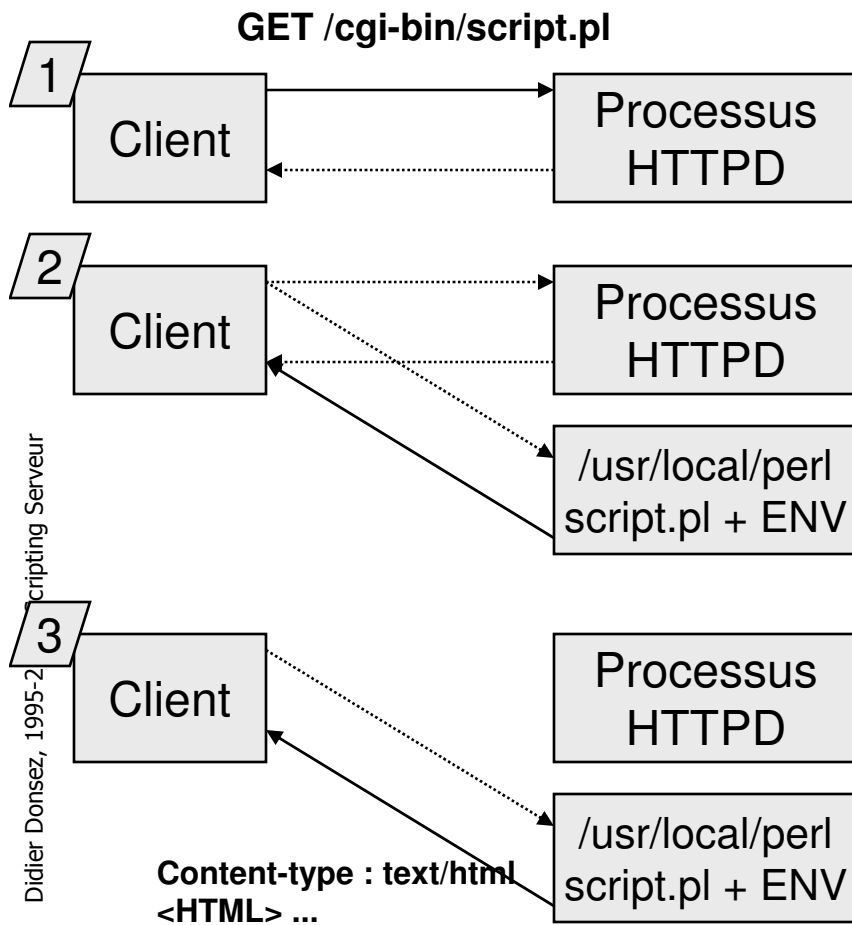
- **Format d'encodage : x-www-form-urlencoded**
 - Les valeurs passées (URL et contenu des entrées des formulaires) doivent être sur 7 bits et sans caractères spéciaux
 - Espace -> « + »
 - Tous les caractères spéciaux (+,&,...) et accentués -> %code ascii
 - @ %40
 - é %e9

Décodage des paires name-value

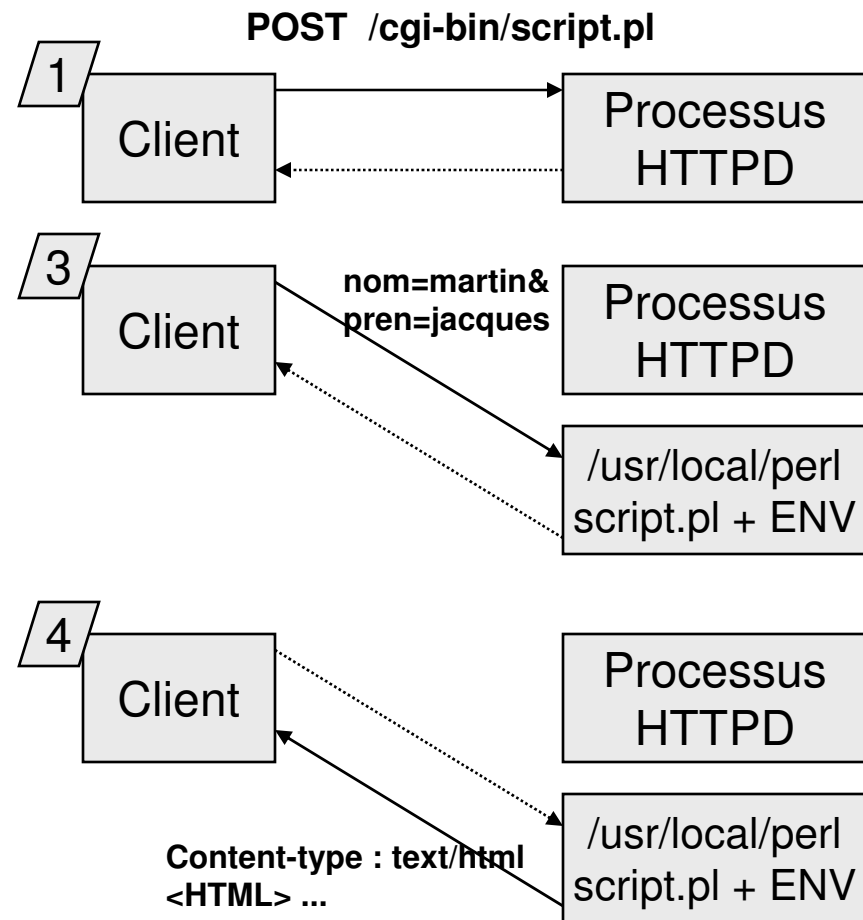
- Nombreux outils pour le décodage de la chaîne
- Shell Unix
 - cgiparse, ...
 - [/ftp.oleane.net:/pub/mirrors/www/cern/bin/osf1/cern_httpd_utils_3.0.tar.gz](ftp://ftp.oleane.net:/pub/mirrors/www/cern/bin/osf1/cern_httpd_utils_3.0.tar.gz)
- Perl
 - lib-cgi.pl, CGI.pm
 - <http://www-genome.wi.mit.edu/ftp/pub/software/WWW/>
- C
 - cgic, CGIlibrary, ..
 - <http://sunsite.unc.edu/boutell/cgic/cgic.html>
 - <http://wsk.eit.com/wsk/dist/doc/libcgi/libcgi.html>
- TCL
 - <http://www.lbl.gov/~clarsen/projects/htcl/http-proc-args.html>

Etapes de création d'un processus CGI

■ Méthode GET



■ Méthode POST



CGI avec "Non Parsed Headers"

- Le serveur interprète l'entête retourné par le CGI
 - notamment pour calculer la longueur Content-Length
- CGI avec "Non Parsed Headers »
 - le serveur n'interprète pas l'entête et retourne les informations à la volée
 - le CGI génère l'entête en entier
 - convention de nommage (fichier préfixé par nph-)

```
#!/bin/sh
```

```
# le header complet sans Content-length
```

```
echo HTTP/1.0 200 OK
```

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

```
echo Server: $SERVER_SOFTWARE
```

```
echo; echo "<HTML><BODY>Liste des Nombres Pairs<HR>"; pair=0;
```

```
while true
```

```
do
```

```
    echo "$pair"; pair = `expr $pair * 2`
```

```
done
```

Les langages pour les CGI

■ Universel

■ compilé

- C, C++, Pascal, ADA, ...

■ interprété ou bytecode

- SH, Perl, Python, Tcl, OCAML, Java, ...

■ Attention au trou de sécurité

- le contenu des formulaires ou des variables doit être contrôlé (particulièrement avec SH, Perl, Python, appel C à system())

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

```
echo
```

```
echo "<HTML><BODY>"
```

```
/usr/bin/finger "$QUERY_STRING" # secure
```

```
echo "<HR>"
```

```
/usr/bin/finger $QUERY_STRING # unsecure
```

```
echo "</BODY></HTML>"
```

```
http://www.mycomp.com/cgi-bin/finger.sh?root+;+/bin/rm+-r+/
```

L 'interface FastCGI (www.fastcgi.com)

■ Principe

- 1 processus instancié l 'est pour plusieurs requêtes
- un programme FastCGI comporte une boucle
 - attente requête, lecture paramètre, traitement, ...
- le dialogue se fait via STDIN, STDOUT et STDERR

■ Caractéristiques

- indépendance du langage
 - C, C++, Perl, Tcl, Python
- sûreté de fonctionnement
 - contexte d 'exécution différent du serveur
- distribution de la charge sur plusieurs machines
 - les processus FastCGI peuvent être sur des machines distantes

Exemple de FastCGI

```
#include "fcgi_stdio.h"
#include <stdio.h>
int main (void) {
    int timesVisited = 0;
    while(FCGI_Accept() >=0) {
        printf ("Content-type: text/html\r\n\r\n");
        printf("<HTML>");
        printf("<HEAD><TITLE>Hello World!</TITLE></HEAD>");
        printf("<BODY><H1>Hello, this is a FastCGI program!</H1>");
        printf("<BIG><P>I can tell that you are visiting from %s.</P>");
        printf("<p>This page has been accessed: %d times</P></BIG>",
                getenv("REMOTE_HOST"), ++timesVisited);
        printf("</BODY></HTML>");
    }
}
```

Les interfaces natives (i)

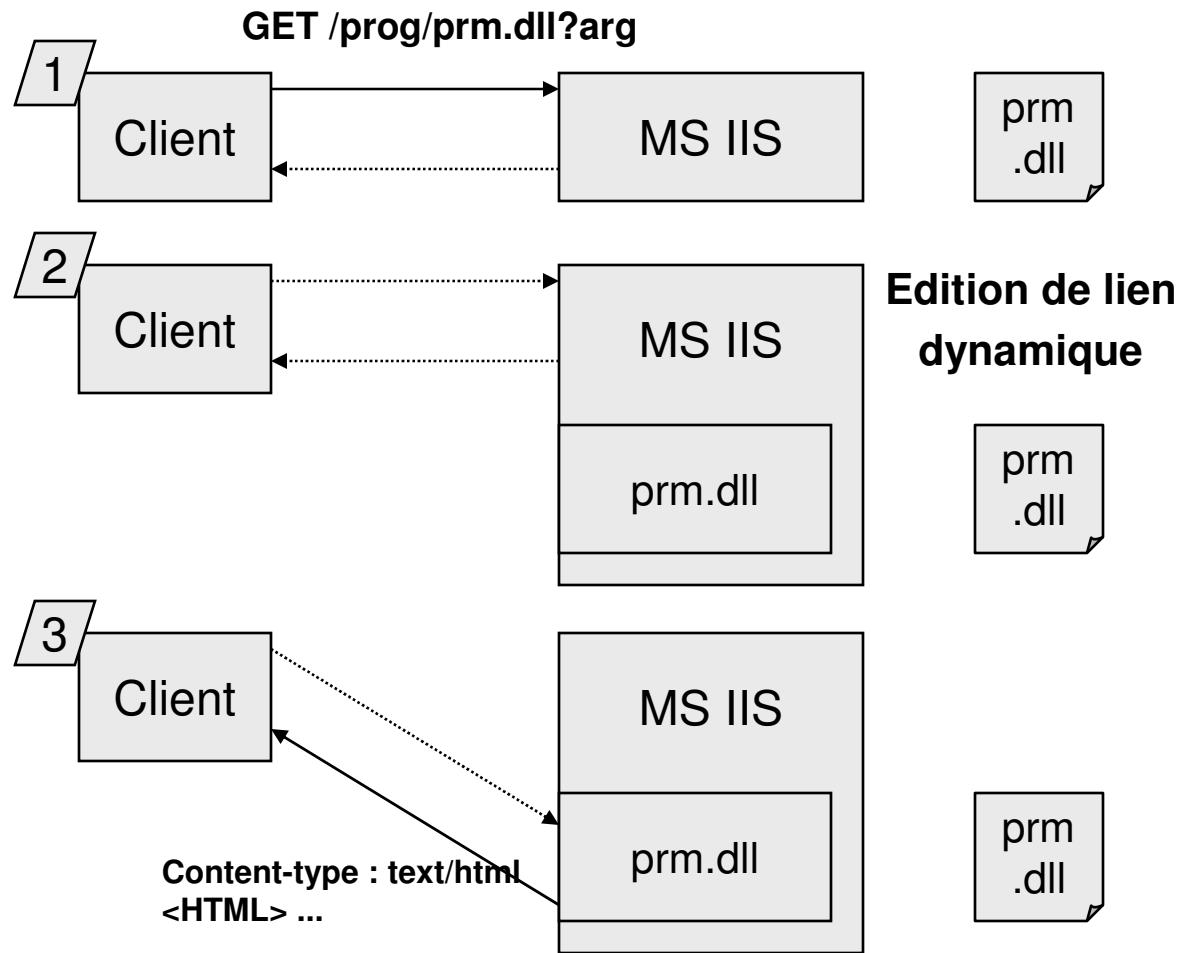
- Principe
 - le code (DLL) des scripts (écrits en C, C++) est compilé
 - puis lié au noyau (core) du serveur HTTPD
 - une requête est prise en charge de bout en bout par une thread du processus serveur

- Chargement des modules contenant les scripts
 - Windows: Dynamic Link Libraries (DLLs)
 - UNIX: Shareable Objects (SOs)

- Standards en compétition
 - Netscape's Netscape API (NSAPI)
 - Microsoft's Internet Server API (ISAPI)

Les interfaces natives (ii)

Exemple avec MS IIS et ISAPI sur WinNT



Les interfaces natives (iii)

Avantages et Inconvénients

■ Avantages

- rapidité dans la prise en charge d'une requête
 - multithreading (plutôt que multi-processus)
 - faible coût du passage des paramètres

■ Inconvénients

- risque de « plantage » total du serveur provoqué par un script non fiable
 - donc interruption de service
- Arrêt du serveur pour incorporer de nouveaux scripts
- Portage à effectuer
 - APIs différentes d'un serveur à l'autre (NSAPI, MS ISAPI)

Développement Web en J2EE

- Développement en Java
- Partie « présentation » d'une application J2EE
 - Servlet
 - Filter
 - JSP (Java Server Page)
 - JSTL (Java Servlet Tag Library)
 - JSF (Java Server Face)
- Voir cours « Développement Web en J2EE »
 - <http://www-adele.imag.fr/~donsez/cours/servletjsp.pdf>

Remarque sur le chargement des classes dans TomCat

- Commun à toutes les WebApps
 - \$CATALINA_HOME/shared/classes/**/*.*
 - \$CATALINA_HOME/shared/lib/*.jar
- Pour la Web App myapp
 - webapps/myapp/WEB-INF/classes/**/*.*
 - webapps/myapp/WEB-INF/lib/*.jar

Brazil

- Alternative aux Web App J2EE

- Boite à outils pour la construction modulaires de serveurs HTTP et des services Web
 - poids léger (<100 Ko)
 - Adapté aux environnements embarqués (TINI, ...)
jusqu'au aux meta-serveurs (agrégation de contenu, portails)

- Voir
 - http://lmi92.cnam.fr:8080/tp_cdi/uhler/java-one.pdf
 - http://lmi92.cnam.fr:8080/tp_cdi/douin/
 - <http://www.ExperimentalStuff.com>
 - <http://www.brazilhandlers.com:9090>

Handlers de Brazil

- Application Brazil
 - = Ensemble de chaînes de Handlers
- Handler
 - Similaire au Filter+Servlet allégée
 - Manipule la requête et la réponse
 - Exemple
 - FilterHandler
 - Réécrit la sortie d'un handler (Adaptation de contenu par exemple)
 - TemplateHandler
 - Réécrit la sortie d'un handler à partir de l'interprétation d'étiquettes , <sql ...>, <ldap ...>, <debug ...>

Handlers de Brazil : la configuration

- Fichier de configuration de la chaîne

```
>java -cp brazil-1_1.jar sunlabs.brazil.server.Main -c config.properties
```

```
## config.properties
```

```
handler=main
```

```
port=9090
```

```
log=5
```

```
root=.
```

```
#
```

```
main.class=sunlabs.brazil.server.ChainHandler
```

```
main.handlers=file dir fileNotFound
```

```
#
```

```
file.class=sunlabs.brazil.server.FileHandler
```

```
file.default=index.html
```

```
#
```

```
dir.class=sunlabs.brazil.handler.DirectoryHandler
```

```
#
```

```
fileNotFound.class=sunlabs.brazil.handler.NotFoundHandler
```

```
fileNotFound.fileName=notFound.html
```

Handlers de Brazil : la configuration

■ Fichier de propriétés

```
>java -cp brazil-1_1.jar sunlabs.brazil.server.Main -c config.properties
```

```
## config.properties
```

```
handler=main
```

```
port=9090
```

```
log=5
```

```
root=.
```

```
#
```

```
main.class=sunlabs.brazil.server.ChainHandler
```

```
main.handlers=clock file fileNotFound
```

```
#
```

```
clock.class= sample.tini.ClockHandler
```

```
clock.name=/clock/
```

```
#
```

```
file.class=sunlabs.brazil.server.FileHandler
```

```
file.default=index.html
```

```
#
```

```
fileNotFound.class=sunlabs.brazil.handler.NotFoundHandler
```

```
fileNotFound.fileName=notFound.html
```

Brazil : Exemple de Handler

// Exemple : ClockHandler sur la carte TINI

```
public class ClockHandler implements Handler {
    private String propsPrefix; private Server server;
    private String name = "/clock/";
    // init() est invoqué une seule fois à la première requête HTTP
    public boolean init(Server server, String prefix) {
        this.server = server;
        this.propsPrefix = prefix; // dans la configuration ="clock."
        try { // lecture du parametre prefix.name
            String n = server.props.getProperty(propsPrefix + "name");
            if (n != null) name = n;
        } catch (Exception e) {}
        return true;
    } ...
}
```

```
clock.class=sample.tini.ClockHandler
clock.name=/clock
```

Brazil : Exemple de Handler

... *// respond() est invoqué à chaque requête HTTP*

```
public boolean respond(Request request) throws IOException {
    String urlPrefix = request.props.getProperty(propsPrefix, name);
    if (!request.url.startsWith(urlPrefix)) {
        server.log(Server.LOG_INFORMATIONAL, null,
            "Not " + name + " prefix: " + request.url);
        return false;
    }
    // la réponse à la requête HTTP
    Clock C = new Clock(); C.getRTC();
    String date = C.getDate() + "/" + C.getMonth() + "/" +
        (C.getYear()+2000) + " " + C.getHour() + ":" +
        C.getMinute() + ":" + C.getSecond() + "." + C.getHundredth();
    String result = "<b>" + date + "</b>";
    request.sendResponse(result);
    return true; }}
```

Brazil : Exemple de Filter

```

public class PlainFilter implements Filter {
    String htmlprefix = "<title>text document</title><body bgcolor=white><pre>";
    String htmlsuffix = "</pre></body>";
    public boolean init(Server server, String prefix) { return true; }
    public boolean respond(Request request) { return false; }
    public boolean shouldFilter(Request r, MimeHeaders headers) {
        String type = headers.get("content-type");
        return (type != null && type.startsWith("text/plain"));
    }
    public byte[] filter(Request request, MimeHeaders headers, byte[] content) {
        String result = htmlprefix
            + new String(content)
            + htmlsuffix;
        headers.put("content-type",
            "text/html");
        return result.getBytes(); }
}

```

```

main.class=sunlabs.brazil.server.ChainHandler
main.handlers=clock tini reflect filter file
filter.class=sunlabs.brazil.filter.FilterHandler
filter.handler=file
filter.filters=plain
plain.class=mybrazil.PlainFilter
plain.header=<b>document de type mime text/plain</b><br>
plain.footer=<hr>

```


Brazil : Exemple de Template SqlTemplate+BSLTemplate

```

<sql prefix=inscrit> SELECT * FROM PEOPLE</sql>
<TABLE border=2>
<foreach name=index property=inscrit.rows >
  <tr>
    <foreach name=field list="FIRSTNAME LASTNAME PHONE EMAIL">
      <td>
        <property name=inscrit.PEOPLE.${field}.${index}>
      </td>
    </foreach>
  </tr>
</foreach>
</TABLE>

```

```

sql.class=sunlabs.brazil.template.TemplateHandler
sql.templates=sunlabs.brazil.template.PropsTemplate \
  cnam.handler.SqlTemplate \
  sunlabs.brazil.template.BSLTemplate
sql.driver=org.hsqldb.jdbcDriver
sql.url=jdbc:hsqldb:http://dbms.mycomp.com:8181
sql.sqlPrefix=param
param.user=adimin
param.password=toto

```

Server Side Include (i)

■ Motivation

- aussi appelé SPML (Server Parsed HTML)
- inclusion de fichiers et de sorties (STDOUT) de commandes dans le document HTML demandé (de type.shtml)

Format `<!--#command option=value option=value ...-->`

`<!--#include file="footer.html" -->`

`<!--#exec cmd="cgi/maximedujour.cgi"-->`

`<!--#printenv-->`

`<!--#echo var="LAST_MODIFIED" -->`

...

■ Configuration d'Apache http://www.apache.org/docs/mod/mod_include.html

- fichier conf/srm.conf :
 - AddHandler server-parsed .shtml
 - AddType text/html shmtl
- un module XSSI pour Apache définit un sur-ensemble de SSI

Server Side Include (ii)

exemple.shtml

```
<HTML><HEAD> <TITLE>La réponse à une CGI</TITLE> </HEAD>
  <!--#if expr="\$QUERY_STRING" = "mode=blue" -->
    <BODY BGCOLOR="#8888FF" TEXT="#0000FF">
  <!--#else -->
    <BODY BGCOLOR="#FFFFFF" TEXT="#000000">
  <!--#endif -->
  <!--#include file="bannierecommune.html" -->
  <!--#goto ="suite" -->
  Jamais retourné
  <!--#label ="suite" -->
  <CENTER>Voila le résultat</CENTER>
  <FORM ACTION="exemple.shtml" METHODE="get">
    <INPUT TYPE="submit" VALUE="Cette page en bleu ...">
    <INPUT TYPE="hidden" NAME="mode" VALUE="blue">
  </FORM> <HR> Dernière Mofication : <!--#echo var="LAST_MODIFIED" -->
</BODY></HTML>
```

Server Side Script (i)

■ Fonctionnement

- le script serveur est imbriqué dans une page HTML
- le serveur lit le fichier HTML avant de l'envoyer au navigateur et exécute les scripts imbriqués qui modifient le source HTML envoyé.

■ Intérêt

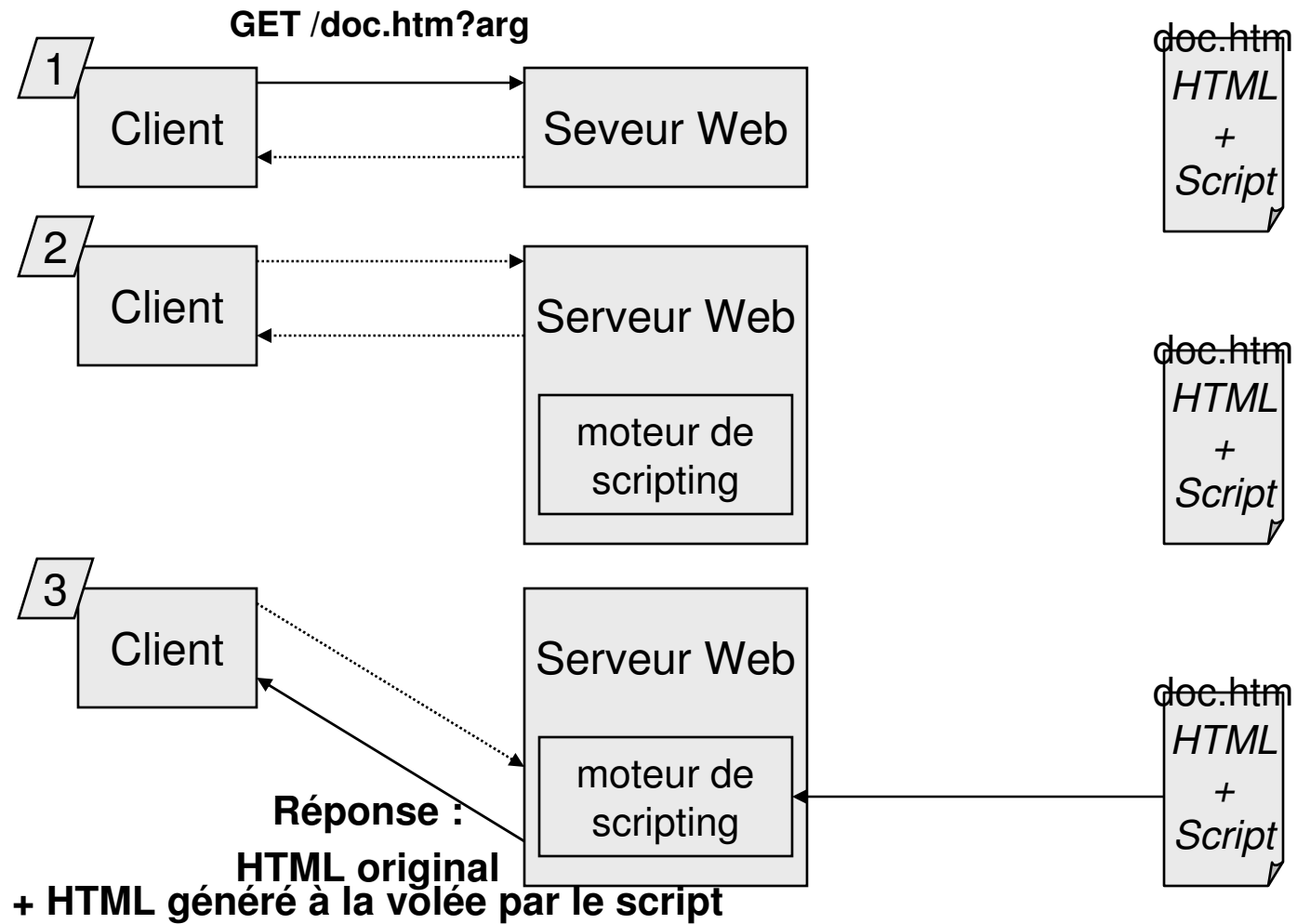
- Développement de pages avec des outils auteur
- Insertion de scripts pour les parties « contextuelles »

■ Nombreuses solutions

- ASP (Active Server Page) de MicroSoft Jscript, VBScript
- LiveWire Server Side JavaScript de Netscape JavaScript
- JSP (Java Server Page) de Sun Java
- PHP (Personal Home Page) PHP
- ...

Server Side Script (ii)

Fonctionnement



Java Server Page (Sun)

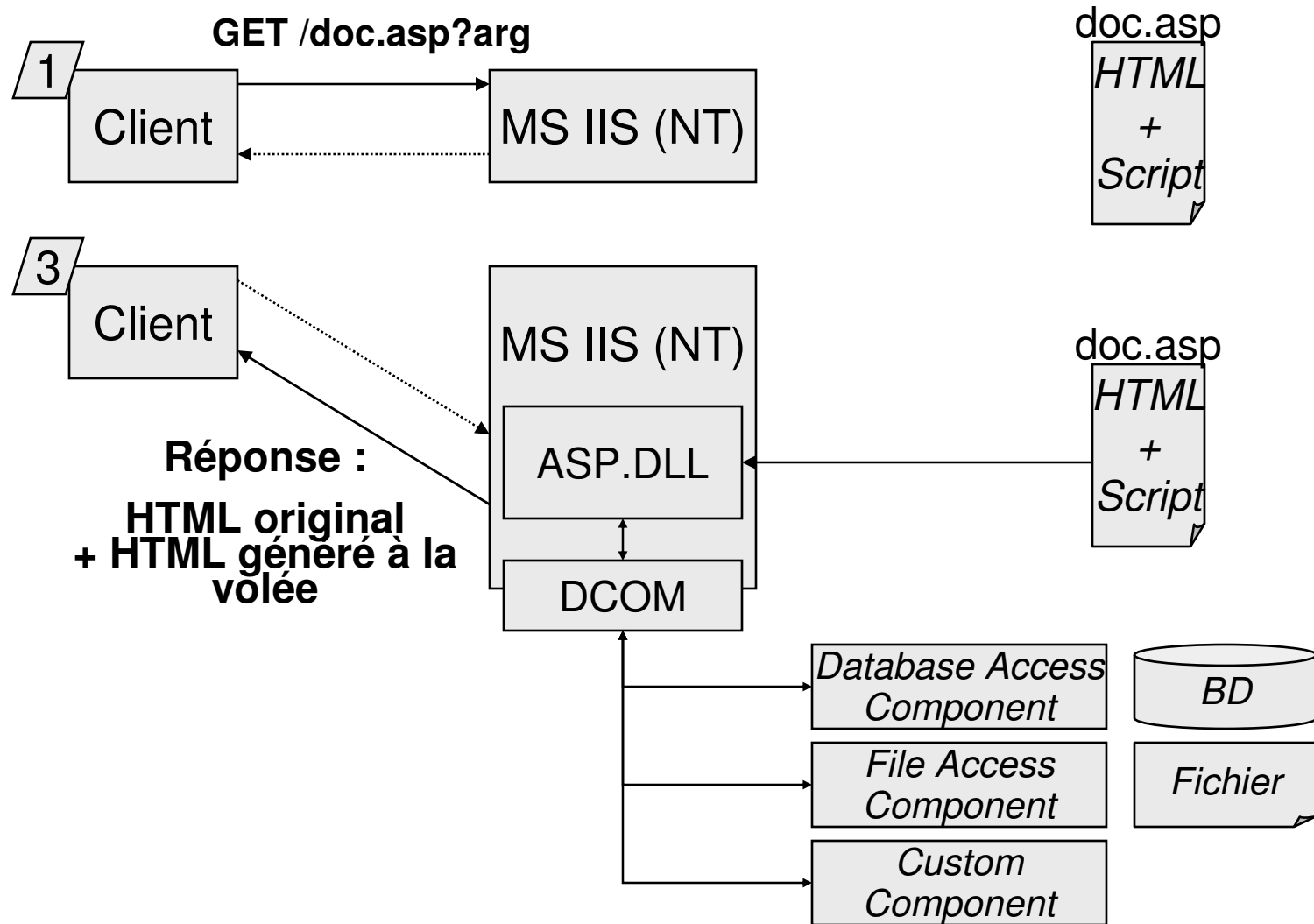
- Principe identique au ASP et SSJ
 - Insertion de SSS (*syntaxe Java*) dans les pages HTML
- Mais le moteur est une servlet (`JspServlet`)

- Voir cours « Développement Web en J2EE »
 - <http://www-adele.imag.fr/~donsez/cours/servletjsp.pdf>

Active Server Page (*MicroSoft*)

- Insertion de SSS dans les pages HTML
 - `<% script %>`
 - langages : JScript et VBScript
- Plateformes et HTTPD
 - ASP.DLL pour MS IIS et MS IPS (WinNT)
 - Chili!Soft ASP sur Unix (<http://www.chilisoft.com/>) mais aussi MainWin, InstantASP, ActiveScripting, Apache::ASP, ASP2PHP ...
- ASP.NET
 - Version .NET des ASP (C#, VB, J#, ...)
 - Services B2C et B2B (SOAP)
 - Serveurs : MS IIS et MS Cassini (gratuit)

ASP Fonctionnement



ASP

Interfaces

Objets

- Application
- ObjectContext
- Request
- Response
- Server
- Session

Autres

- GLOBAL.ASA
- Directives du préprocesseur
- Server Side Include
- ActiveX Data Object (ADO)
 - accès à des sources de données (BD, Fichiers, ...)

ASP

Cycle de vie

- Application
 - ensemble des ASP placées dans un même répertoire
- Initialisation
 - la première requête sur une de ces ASP initialise des informations globales à ces ASP et quelque soit les utilisateurs
 - l'initialisation se fait au moyen des scripts
 - du fichier GLOBAL.ASA placé dans le répertoire
 - au moyen de l'objet Application modifié dans l'ASP
- Terminaison
 - la terminaison est déclenchée par le serveur
 - le script OnEnd défini dans GLOBAL.ASA est exécuté

ASP

l'objet Application

- 2 collections
 - Contents
 - collection de valeurs ou d'objets
 - StaticObjets
 - collections des <OBJECT RUNAT=SERVER> définis dans GLOBAL.ASA
- 2 méthodes : lock/unlock
 - préviens les accès concurrents de plusieurs ASP simultanées.

ASP

L'objet Application

```
<%  
Application.Lock  
intNumVisits = Application.Contents("AppUserCount")  
intNumVisits = intNumVisits + 1  
Application.Contents("AppUserCount") = intNumVisits  
Application.Unlock  
%>  
<HTML>  
<HEAD><TITLE>Home Page</TITLE></HEAD>  
<BODY BGCOLOR = #ffffcc>  
Welcome to our homepage. You are client number  
<%= Application.Contents("AppUserCount")%> to our site.  
</BODY>  
</HTML>
```


ASP

l'objet Request

- donne accès à l'entête et au corps de la requête HTTP
- propriété
 - TotalBytes
- collections
 - ClientCertificate, Cookies, Form, QueryString, ServerVariable
- méthode
 - BinaryRead
 - permet de lire directement le corps d'une requête POST

ASP

I 'objet Request

```
<%  
If Not IsEmpty(Request.Form("txtFirstName")) And _  
    Not IsEmpty(Request.Form("txtLastName")) Then  
%>  
  <HTML><HEAD><TITLE>Thank You</TITLE></HEAD><BODY>  
  Thank you, <%= Request.Form("txtFirstName")%>&nbsp;nbsp;  
  <%= Request.Form("txtLastName")%> for your information. Have a nice day.  
  </BODY></HTML>  
<% Else %>  
  <HTML><HEAD><TITLE>Thank You</TITLE></HEAD><BODY>  
  <FORM NAME = "frmInfo" ACTION="UserInfo2.ASP" METHOD = "POST">  
    First Name:      <INPUT TYPE="text" NAME = "txtFirstName"><BR>  
    Last Name:       <INPUT TYPE="text" NAME = "txtLastName"><BR>  
  <INPUT TYPE = "Submit" VALUE = "Submit User Info">  
  </FORM></BODY></HTML>  
<%End If %>
```

ASP

L'objet Response

- donne accès à la réponse de la requête
- propriétés
 - Buffer, CacheControl, Charset, ContentType, Expires, ExpiresAbsolute, IsClientConnected, PICS, Status
- collections
 - Cookies
- méthode
 - AddHeader, AppendToLog, BinaryWrite, Clear, End, Flush, Redirect, Write

ASP

les objets Request et Response

```
<% Response.Buffer = True %>
<HTML><HEAD><TITLE>Thank You</TITLE></HEAD><BODY>
  <FORM NAME = "frmInfo" ACTION="UserInfo2.ASP" METHOD = "POST">
    First Name:      <INPUT TYPE="text" NAME = "txtFirstName"><BR>
    Last Name:       <INPUT TYPE="text" NAME = "txtLastName"><BR>
    <INPUT TYPE = "Submit" VALUE = "Submit User Info">
  </FORM>
<% If Not IsEmpty(Request.Form("txtFirstName")) And _
      Not IsEmpty(Request.Form("txtLastName")) Then
      Response.Clear
%>
  <HTML><HEAD><TITLE>Thank You</TITLE></HEAD><BODY>
  Thank you, <%= Request.Form("txtFirstName")%>&nbsp;
  <%= Request.Form("txtLastName")%> for your information. Have a nice day.
<%End If %>
</BODY></HTML>
```

ASP

les objets Request et Response

```
<% Response.Buffer = True %>
<HTML><HEAD><TITLE>Thank You</TITLE></HEAD><BODY>
  <FORM NAME = "frmInfo" ACTION="UserInfo2.ASP" METHOD = "POST">
    First Name:      <INPUT TYPE="text" NAME = "txtFirstName"><BR>
    Last Name:       <INPUT TYPE="text" NAME = "txtLastName"><BR>
    <INPUT TYPE = "Submit" VALUE = "Submit User Info">
  </FORM>
<% If Not IsEmpty(Request.Form("txtFirstName")) And _
      Not IsEmpty(Request.Form("txtLastName")) Then
  Response.Clear
  Dim strUserId
  strUserId = Request.Form("txtFirstName") + " " + Request.Form("txtLastName")
  Response.Cookies("UserId") = strUserId %>
<HTML><HEAD><TITLE>Thank You</TITLE></HEAD><BODY>
  Thank you, <%= strUserId %> and have a nice day.
<%End If %></BODY></HTML>
```

ASP

l'objetObjectContext

- Permet de définir une ASP transactionnel
 - la portée est limitée à la requête
 - le contrôle XA est réalisé par MTS
- Directive
 - `<%@ TRANSACTION=Required%>`
 - `Required, Required_New, Supported, Not_Supported`
- Méthodes
 - `SetComplete, SetAbort`
 - force la terminaison de la transaction
- Événements
 - `OnTransactionCommit, OnTransactionAbort`
 - capture l'événement pour valider ou abandonner des modifications opérées sur des sources non transactionnelles (ex: SGF, ...)

ASP

l'objet Session

- Permet de traquer une session utilisateur
 - la session est commune à tous les ASP du même répertoire ou sous-répertoire
 - durée par défaut : 20 minutes
 - SessionID implantée avec un cookie
- Propriétés
 - CodePage, LCID, SessionID, TimeOut
- Méthodes
 - Abandon
 - la session est abandonnée

ASP

I 'objet Session (ii)

■ Collections

■ Contents

- ensemble des informations définies au niveau de la session
 - `Session.Content.Item("Nom")=Request.Form("name")`

■ StaticObjects

- ensemble des objets définis au niveau de la session
 - `<OBJECT RUNAT=SERVER ID=`

■ Événements

- `Session_OnStart, Session_OnEnd`

■ Directive

- `<%@ ENABLESESSION=False%>`
 - True (par défaut), False (évite le suivi inutile)

ASP

I 'objet Server

- Propriétés
 - ScriptTimeout
 - limite la durée d 'exécution d 'un ASP (90 sec par défaut)
- Méthodes
 - CreateObject
 - instancie un objet (DLL) sur le serveur
 - MapPath
 - donne le chemin physique d 'un chemin virtuel
 - HTML Encode
 - encodage de code neutre au HTML
 - HTML Encode("<TABLE>") donne "<TABLE>"
 - URLEncode
 - encode de code neutre par rapport au URL
 - URLEncode("http://yahoo") donne "http%3A%2A%2Ayahoo"

ASP

Exemple 1 : Formulaire

```
<% @Language=JScript    %>
<HTML><HEAD> <TITLE>Form Posting</TITLE></HEAD> <BODY>
<H1>Form Posting</H1><HR>
    <FORM Name=Form1 METHOD=POST ACTION="Form_JScript.asp">
        Votre Nom: <Input Type=Text Name=nom><p>
        Votre Prénom: <Input Type=Text Name=prenom><p>
        <INPUT Type=Submit Value="Submit">
    </FORM>
    <HR>
    Bonjour
    <% Response.Write(Request.Form("nom"))%>
    <% Response.Write(Request.Form("prenom"))%>
    <HR>
<H1>Affiche toutes les variables du serveur</H1><HR>
    <%= Request.ServerVariables("ALL_HTTP")%>
</BODY></HTML>
```

ASP

Exemple 2 : Accès à une BD

```
<% @Language=JScript    %>
<HTML><HEAD> <TITLE>Accès à un base</TITLE></HEAD> <BODY>
<h1>Contenu de la Bibliotheque<h1>
<%
var curDir = Server.MapPath("\\iissamples\\sdk\\asp\\database\\biblio.mdb");
var oConn = Server.CreateObject("ADODB.Connection");
oConn.Open("DBQ="+curDir+";Driver={Microsoft Access Driver (*.mdb)};" +
           "DriverId=25;FIL=MS Access;");
var oRs = oConn.Execute("Selet * From livres");
%>
<TABLE border = 1> <% while (!oRs.eof) { %>
    <TR>
        <% for(Index=0; Index < (oRs.fields.count); Index++) { %>
            <TD VAlign=top><% = oRs(Index)%></TD>
        <% } %>
    </TR>
        <% oRs.MoveNext(); } %>
</TABLE> <%  oRs.close(); oConn.close(); %> </BODY></HTML>
```




ASP .NET

- Version des ASP sur le plate-forme .NET
 - .aspx

- Gère également les Web Services
 - SOAP et WSDL



PHP (www.php.net)

- Insertion de SSS dans les pages HTML
 - `<? script ?>`
 - Langage : PHP (Personal Home Page)
 - syntaxe semblable au C et typage à la Perl

- Plateformes et HTTPD
 - Unix, WinNT et FastCGI, ISAPI, NSAPI
 - module pour Apache
 - Sa force : de nombreuses fonctions natives
 - IMAP, SNMP, ODBC, Accès natifs SGBD, LDAP, Génération d'images à partir de tables BD, XMLDOM/XSLT, ...
 - débogueur rudimentaire

Exemple PHP

Création d'une image GIF à la volée

```
<?php
  Header("Content-type: image/gif");
  $string=implode($argv," ");
  $im = imagecreatefromgif("images/button1.gif");
  $orange = ImageColorAllocate($im, 220, 210, 60);
  $px = (imagesx($im)-7.5*strlen($string))/2;
  ImageString($im,3,$px,9,$string,$orange);
  ImageGif($im);
  ImageDestroy($im);
?>
```

- Appel de script dans une page HTML

```

```

Exemple PHP

Accès à une BD (Interface Native Informix)

- **Example 1. Show all rows of the "orders" table as a html table**

```
ifx_textasvarchar(1);    // use "text mode" for blobs
$res_id = ifx_query("select * from orders", $conn_id);
if (! $res_id) {
    printf("Can't select orders : %s\n<br>%s<br>\n",
        ifx_error();
        ifx_errormsg());
    die;
}
ifx_htmltbl_result($res_id, "border=\"1\"");
ifx_free_result($res_id);
```

Exemple PHP

Accès à une BD (Interface Native Informix)

Example 2. Insert some values into the "catalog" table

```
// create blob id's for a byte and text column
$textid = ifx_create_blob(0, 0, "Text column in memory");
$byteid = ifx_create_blob(1, 0, "Byte column in memory");
// store blob id's in a blobid array
$blobidarray[] = $textid;
$blobidarray[] = $byteid;
// launch query
$query = "insert into catalog (stock_num, manu_code, " .
        "cat_descr,cat_picture) values(1,'HRO',?,?)";
$res_id = ifx_query($query, $conn_id, $blobidarray);
if (! $res_id) {
    ... error ...
}
// free result id
ifx_free_result($res_id);
```

Exemple PHP

Accès à une BD (Interface Native MySQL)

```
<?
    if($vote && !$already_voted)
        SetCookie("already_voted","1");
?>
<HTML><HEAD><TITLE>Name the Baby</TITLE>
</HEAD><H3>Name the Baby</H3>
<FORM ACTION="baby.php3" METHOD="POST">
Suggestion: <INPUT TYPE=text NAME=new_name><P>
<INPUT TYPE=submit
    VALUE="Submit idea and/or vote">
<?
mysql_pconnect("localhost","","");
$db = "test";
$table = "baby_names";

if($new_name) {
if(!mysql_db_query($db,
"insert into $table values
('$new_name',0)") {
echo mysql_errno().": ";
echo mysql_error()."<BR>";
```

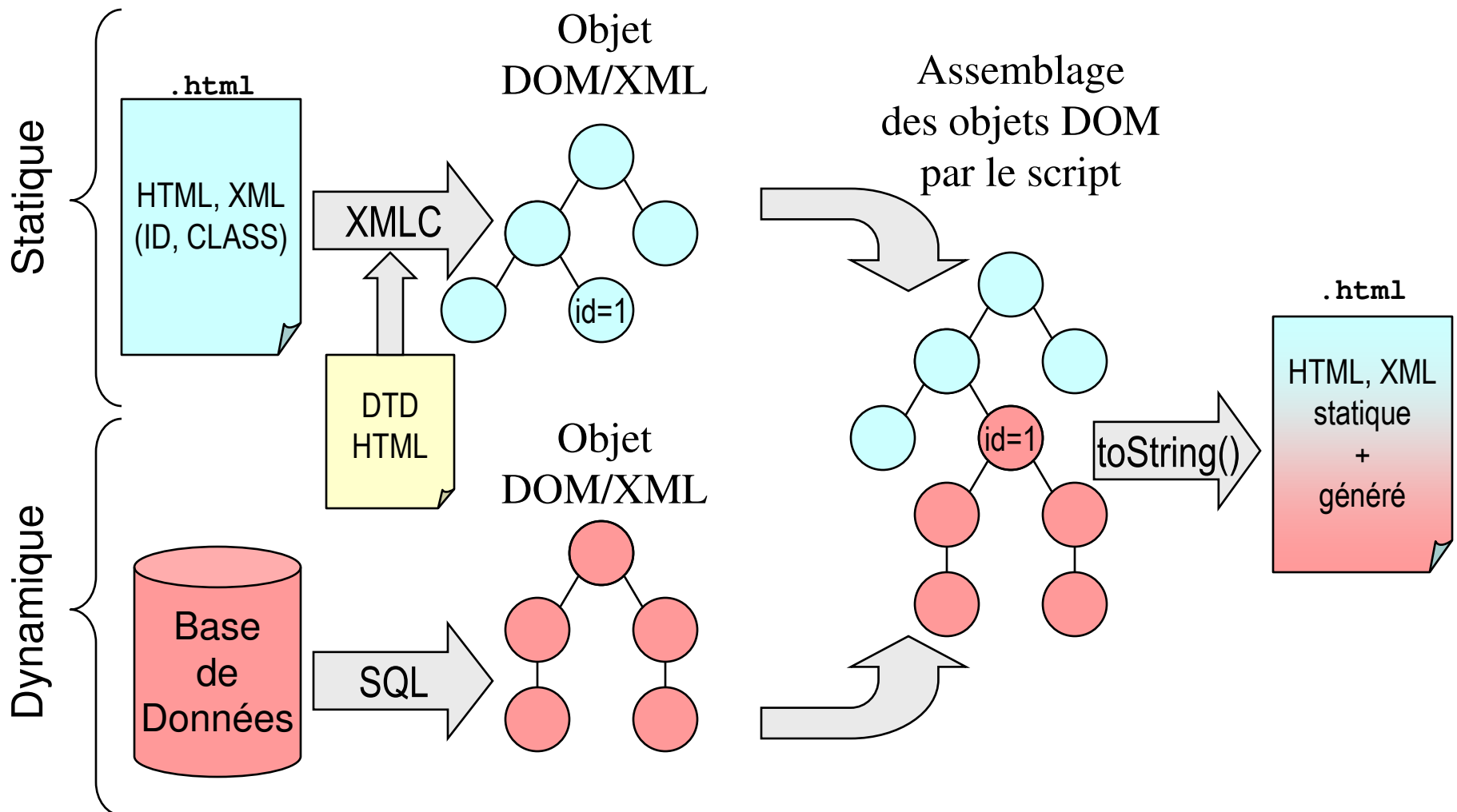
Scripting Serveur et XML/DOM

- Inconvénients des solutions précédentes
 - Script générant du HTML
 - génération par programmation (« à la main »)
 - Script niché dans un document HTML
 - Utilisation d 'outils auteur pour créer les pages, OUI !
 - MAIS l 'insertion de scripts empêche de retravailler les pages
- Alternative
 - Utilisation du modèle DOM dans les scripts
 - www.enhydra.org

Scripting Serveur et XML/DOM

Principe (i)

- Principe : le script ne manipule que des objets DOM



Scripting Serveur et XML/DOM

Cycle de Développement (i)

- 1) Conception de document HTML avec des outils auteur
 - L'infographiste conçoit le document avec une touche graphique et des animations standards (bannière, boutons animés, menu, ...)
 - Certaines zones du document seront générés à la volée par un script. Ces zones sont repérées par des attributs ID ou CLASS.
 - Le document pourra être modifié ultérieurement (après 4)
- 2) Compilation des pages HTML en objet Java DOM
 - le document HTML est compilé en une classe Java instanciant des objets DOM/XML
 - nécessite parfois une mise en conformité (DTD/HTML4) du source HTML produit par les outils auteur
 - packages org.w3c.dom et org.w3c.dom.html

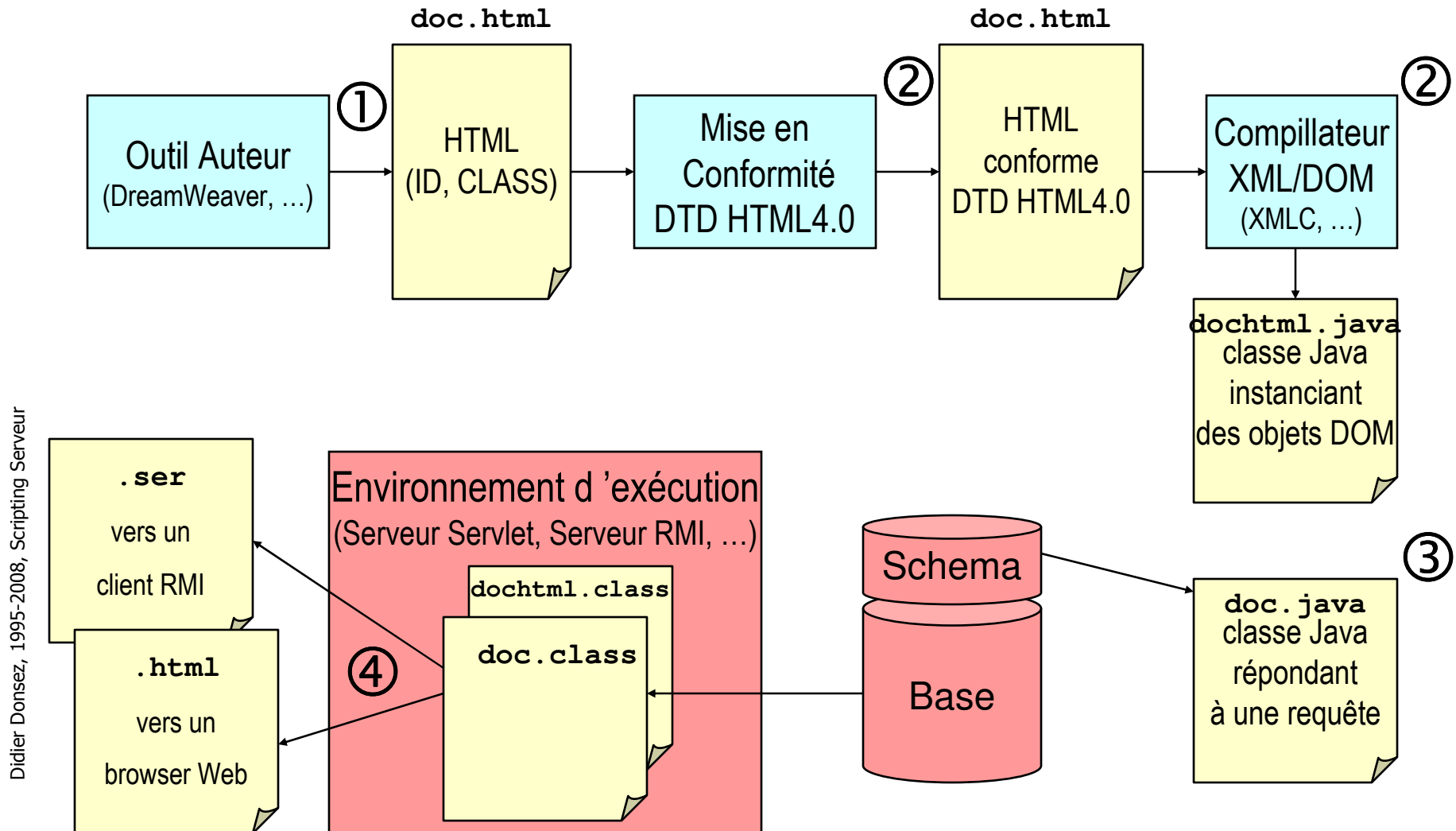
Scripting Serveur et XML/DOM

Cycle de Développement (ii)

- 3) Développement du script
 - Le script (servlet) instance un objet DOM à partir de la classe produit par le compilateur DOM. Cet objet est un arbre
 - Les sous-arbres identifiés (ID ou CLASS) par l'infographiste correspondent aux zones qui vont être générés à la volée.
 - Le script modifie ou remplace ces sous-arbres à partir de de sous-arbres générés à partir de données retirées du Système d'Information (à l'exécution).
- 4) Requête
 - Le script assemble les sous-arbre DOM générés avec l'arbre DOM statique.
 - Le script répond à la requête du client en retournant
 - l'arbre DOM sérialisé dans le cas d'une réponse RMI
 - méthode writeObject()
 - une génération texte de l'arbre DOM dans le cas d'une réponse HTTP
 - méthode toString()

Scripting Serveur et XML/DOM

Cycle de Développement (iii)



Scripting Serveur et XML/DOM

Source HTML Hello.html

```
<HTML> <HEAD> <TITLE>Hello World</TITLE>
<BODY>
    <H1>Hello World</H1>
    <P>Hello, you are <SPAN ID=name >Didier</SPAN>.
    <P>Today's date is <SPAN ID="date" >Oct 6, 2001
</HTML>
```

- La mise en conformité selon la DTD HTML 4.0 donne

```
<HTML>
<HEAD><TITLE>Hello World</TITLE></HEAD>
<BODY>
    <H1>Hello World</H1>
    <P>Hello, you are <SPAN ID="name" >Didier</SPAN>.</P>
    <P>Today's date is <SPAN ID="date" >Oct 6, 2001</SPAN></P>
</BODY>
</HTML>
```

Scripting Serveur et XML/DOM

Objet DOM instance de la classe HelloHTML.class

- La compilation XML/DOM produit la classe HelloHTML instanciant des objets (arbre) DOM suivants :

HTMLDocument

HTMLHtmlElement: html

HTMLHeadElement: head

HTMLTitleElement: title

Text: text="Hello World"

HTMLBodyElement: body

HTMLHeadingElement: h1

Text: text="Hello World"

HTMLParagraphElement: p

Text: text="Hello, you are"

HTMLSpanElement: span: id='name'

Text: text="Didier"

Text: text="."

HTMLParagraphElement: p

Text: text="Today's date is "

HTMLSpanElement: span: id='date'

Text: text="Oct 6, 2001"

Scripting Serveur et XML/DOM

Objet DOM instance de la classe HelloHTML.class

- Les méthodes getElementXX() retrouvent les sous-arbres identifiés par l'infographiste :

```
// HelloHTML.java
class HelloHTML extends org.enhydra.xmlc.HtmlDocument
implements org.w3c.dom.html.HtmlDocument {

    HelloHTML() { ... } // constructeur qui initialise un arbre DOM
                        // représentant le document Hello.html

    ...
    ... getElementName() { ... } // retourne le sous-arbre DOM
                                // identifié par ID="name"
    ... getElementDate() { ... } // retourne le sous-arbre DOM
                                // identifié par ID="date"

    ..
}
```

Scripting Serveur et XML/DOM

Script modifiant une instance de HelloHTML.class

```
// Hello.java
public class Hello extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        HelloHTML htmlObj = new HelloHTML();

        String firstname = req.getParameter("firstname");
        if(firstname==null) firstname="unidentified";

        // assemblage des sous-arbres DOMs générés avec l'arbre DOM « statique »

        XMLCUtil.getFirstText(htmlObj.getElementName()).setData(firstname);
        XMLCUtil.getFirstText(htmlObj.getElementDate()).setData(getDate());

        PrintWriter out = res.getWriter();
        out.println(htmlObj.toString());
    }
}
```

Scripting Serveur et XML/DOM

Script modifiant une instance de HelloHTML.class

```
public class Hello extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        HelloHTML htmlObj = new HelloHTML();

        XMLCUtil.getFirstText(htmlObj.getElementName()).setData(getRandomName());
        XMLCUtil.getFirstText(htmlObj.getElementDate()).setData(getDate());

        PrintWriter out = res.getWriter();
        out.println(htmlObj.toString());
    }
}
```


Scripting Serveur et XML/DOM

Script modifiant une instance de HelloHTML.class

```
// inport org.enhydra.*
public class Hello implements HttpPresentation {
    public void run(HttpPresentationComms comms) {
        HelloHTML htmlObj = new HelloHTML();

        XMLCUtil.getFirstText(htmlObj.getElementName()).setData(getRandomName());
        XMLCUtil.getFirstText(htmlObj.getElementDate()).setData(getDate());

        comms.response.writeHTML(htmlObj.toString());
    }
}
```

XSP (eXtensible Server Page)

- équivalent aux JSP pour XML
 - voir Cocoon <http://xml.apache.org>

- Exemple

```

<?xml version="1.0"?>
<?cocoon-process type="xsp"?><?cocoon-process type="xslt"?>
<?xml-stylesheet href="myStylesheet.xsl" type="text/xsl"?>
<xsp:page language="java"
  xmlns:xsp="http://www.apache.org/1999/XSP/Core">
  <xsp:logic>
    private static int numHits = 0;
    private synchronized int getNumHits( ) { return ++numHits; }
  </xsp:logic>
  <page><title>Hit Counter</title>
  <p>I've been requested<xsp:expr>getNumHits( )</xsp:expr> times.</p>
</page>
</xsp:page>

```

Velocity

<http://jakarta.apache.org>



- Moteur de transformation à partir de Template
 - Les templates contiennent des « balises » pouvant être remplacé par des propriétés
 - Peut être utilisé au sien du servlet (JetSpeed) pour de la génération de pages ou statiquement (Jonas pour la génération du code Java des containers EJB)
- Exemple

Struts

<http://jakarta.apache.org/struts>

Under Construction
En Construction

- Modèle MVC pour concevoir des applications Web
 - Basé sur les TagLibs pour JSP

- Voir

<http://jakarta.apache.org/struts/userGuide/index.html>

JSF - Java Server Faces (JSR127).



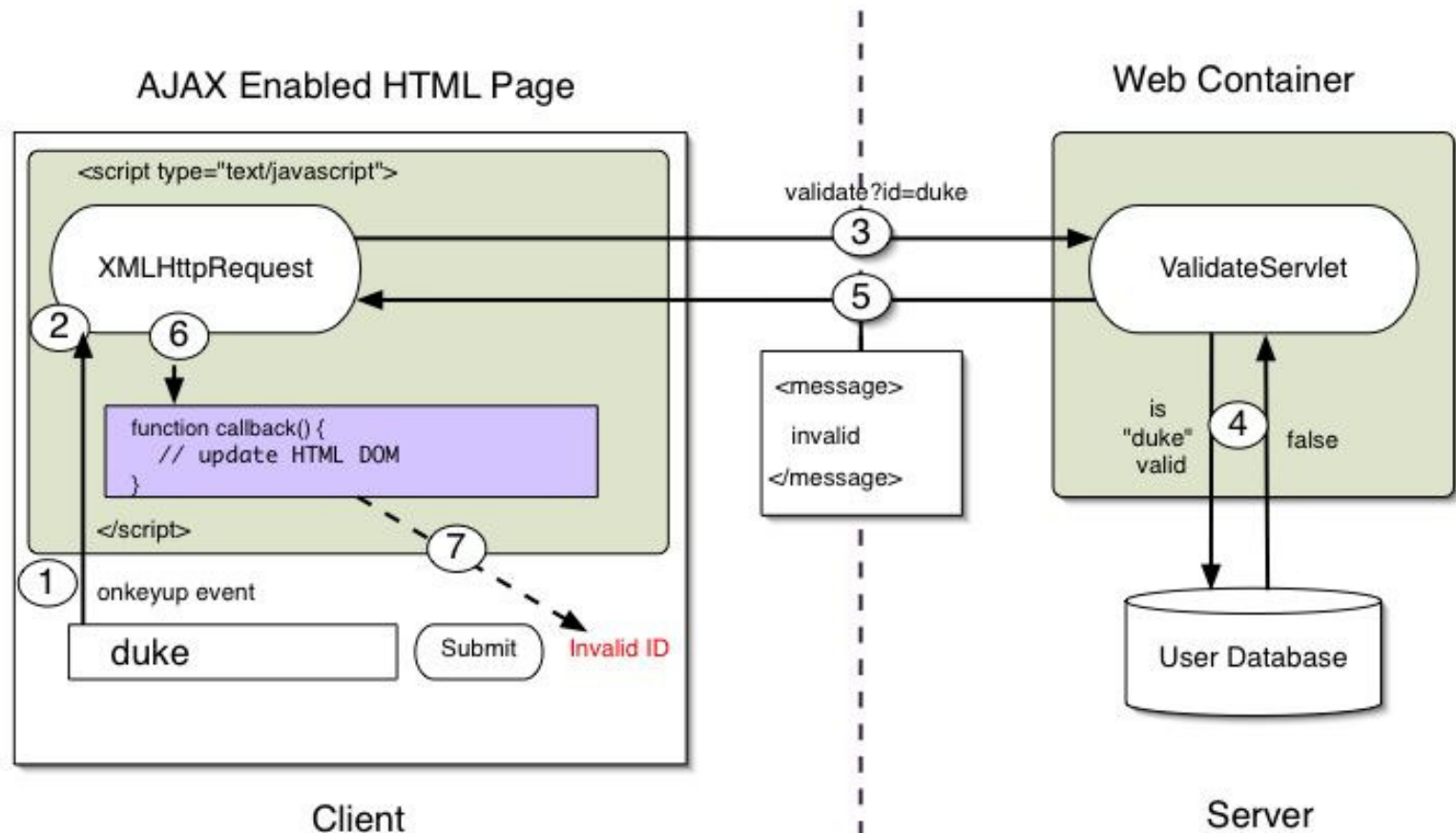
BSL



AJAX

Asynchronous JavaScript Technology and XML

- Permet la création de Rich Client en JavaScript



- Lire

- <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/>

HTTP et Flash

- TODO

ZOPE (Z Object Publishing Environment)

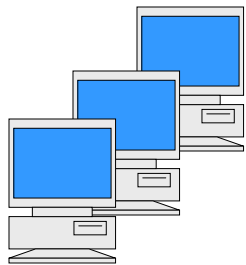
- Environnement de développement de services Web
 - Administration via des objets ZOPE (prédéfinis)
 - Gestion intégrée des utilisateurs, ...
 - HTTPD standalone ou avec Apache comme Frontal
- Développement
 - Orienté objet
 - Persistance des objets
 - Héritage (propriétés et méthodes)
 - DTML (Document Template Markup Language)
 - Langage de balise insérant dans HTML les propriétés des objets
 - `<dtml-var >` `<dtml-if >` `<dtml-in >` ...
 - Manipulation des objets en Python

ZOPE

- Composition de services
 - ZClasses : logique, structure et présentation
 - ZCatalog : indexation et recherche sur les propriétés des objets
 - ZSQL : interfacage à des bases SQL
- Web
 - www.zope.org
- Livres
 - PJ Grizel, « ZOPE », Ed Eyrolles, 2001, ISBN 2-212-09281-4

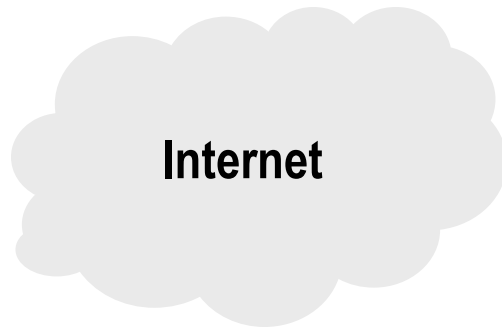
Edge Computing

navigateurs web

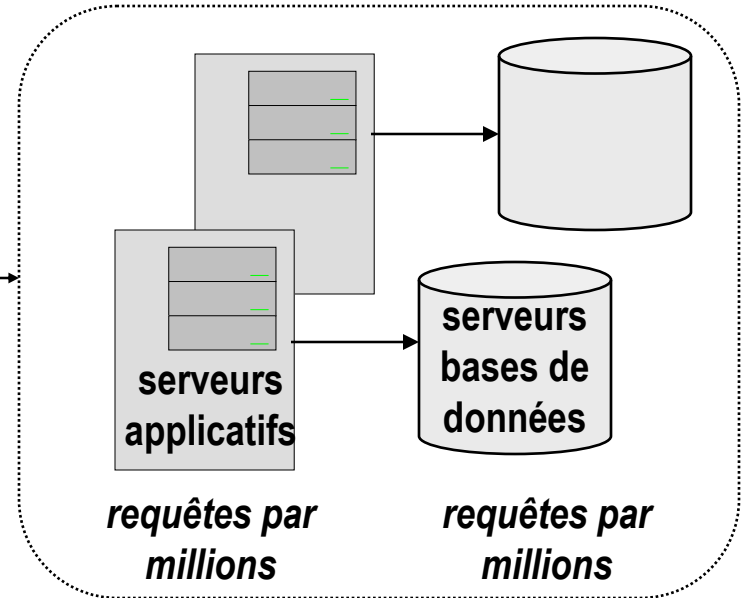


requêtes par millions

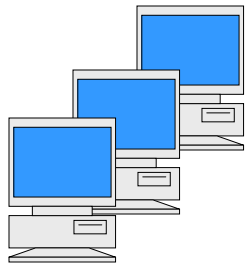
pull



Internet



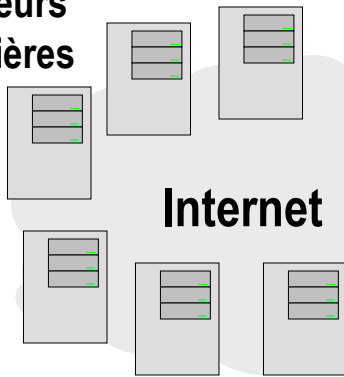
navigateurs web



requêtes par millions

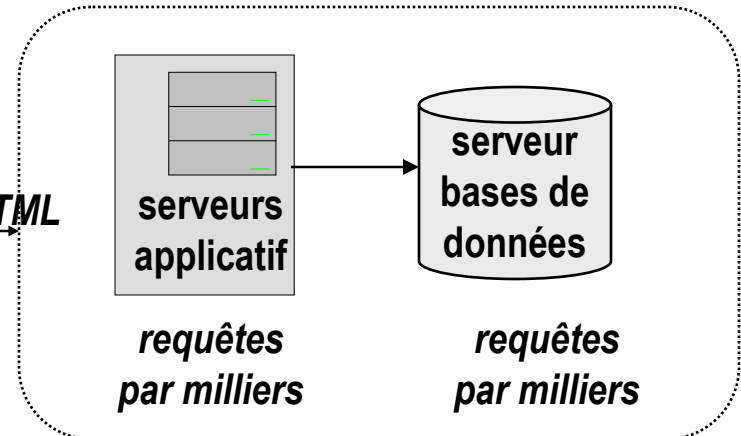
pull

centaines de serveurs frontières

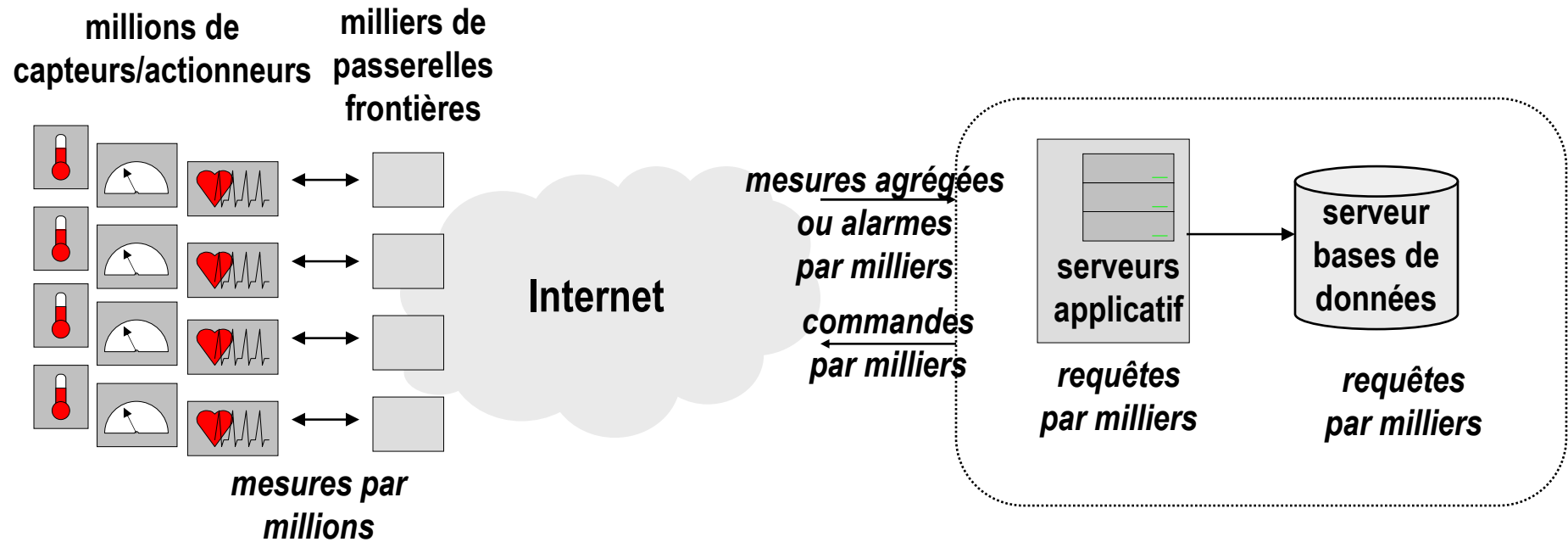


Internet

fragments HTML à cacher



Edge Computing



Edge Side Includes (ESI)

- Conception du contenu
 - Les documents sont divisés en plusieurs fragments
 - cachés sur les « Edges Servers » et assemblés par ceux-ci
- Pratique
 - L'assemblage s'exprime au moyen de balises exprimant inclusion, condition, variables d'environnement (cookie), ...
- Expiration/Invalidation
 - « Time-To-Live » des fragments
 - Invalidation des fragments cachés par les serveurs d'origine au moyen d'un protocole d'invalidation

Edge Side Includes (ESI)

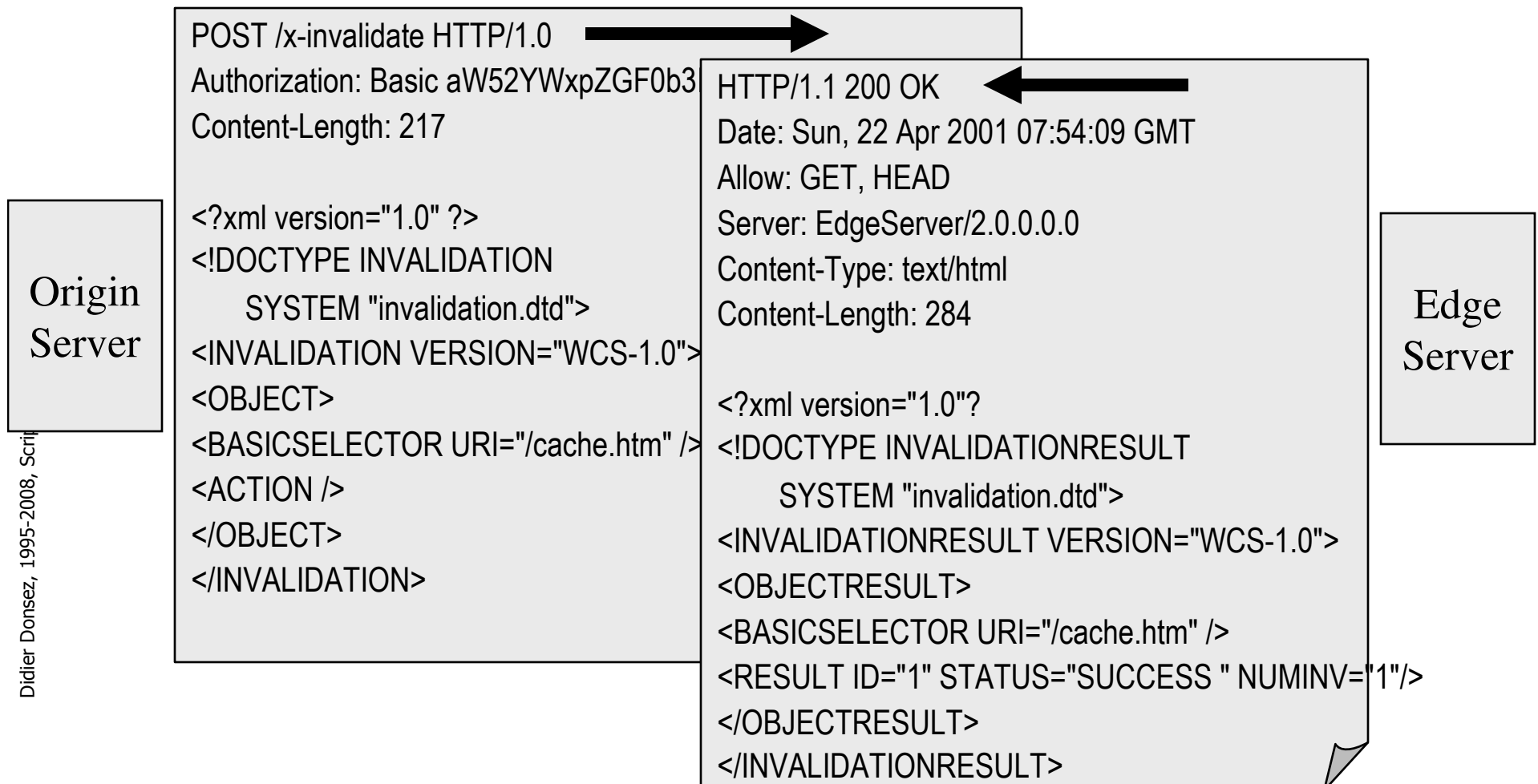
Taglib JESI - <http://www.esi.org/>

- Basé sur les TagLib JSP (Spec J2EE 1.3)

| JSP TagLib | Purpose |
|---------------------------------------|---|
| <code><jesi:include></code> | Used in a “template” page to indicate to the ESI processor how the fragments are to be assembled (the tag generates the tag). |
| <code><jesi:control></code> | Assign an attribute (e.g., expiration) to templates and fragments. |
| <code><jesi:template></code> | Used to contain the entire content of a JSP container page within its body. |
| <code><jesi:fragment></code> | Encapsulate individual content fragments within a JSP page. |
| <code><jesi:codeblock></code> | Specify that a particular piece of code needs to be executed before any other fragment is executed (a database connection established, user id computed, etc.). |
| <code><jesi:invalidate></code> | Explicitly remove and/or expire selected objects cached in an ESI processor. |
| <code><jesi:personalize></code> | Insert personalized content into a page where the content is placed in cookies and inserted into the page by the ESI processor. |

Edge Side Includes (ESI) Protocole d'invalidation

- Requêtes d'invalidation
 - Message XML (DTD) + transport sur HTTP / POST (port 4001)



Les outils accessibles aux scripts

- Base de Données
 - ODBC, ADO, JDBC
- Moniteurs Transactionnelles
 - MTS, JTS
- Messages inter-application
 - Queueing
- Composants Métiers
 - CORBA, DCOM+
 - EJB : Enterprise Java Beans
- Moteurs d'Indexation

Paramètres et Sécurité

- Vérifier l'intégrité de paramètres passées aux scripts
 - Pas seulement coté client (contrôle JavaScript sur les entrées d'un FORM)
 - présence de caractères d'échappement dans la valeur du paramètre
 - Risque pour le service
 - Risque pour le client d'un service (login/password)
- Exemple
 - PERL, SH
 - `demo.sh?p=toto;/bin/rm%20-fr%20.`
 - Requête SQL
 - `addproduct.jsp?ref=12345;DROP%20TABLE%20PRODUCT;`
 - Cross Site Scripting
 - `search.jsp?q=toto<script>pwd=prompt("Entrez votre mot de passe");</script>`

Scripting Serveur et Bases de Données

- But
 - Accès à des données dans une base de données
 - Formatage du résultat en HTML, XML, WML, ... PDF
- Accès à la Base
 - Accès Natif
 - ☺ fonctionnalités natives étendues ☹ portabilité
 - Accès via ODBC/JDBC
 - Via des Enterprise Beans
 - ☺ pas d'écriture de JDBC pour le CMP ☺ un tiers de plus
- Type de Connexion
 - « Transient »
 - transaction commit ou rollback à la fin du script, connexion close
 - ☹ performance
 - Pool de connexions
 - une connexion est partagée successivement par plusieurs invocations de scripts

Outils

- Les versions « entreprise » des principaux outils de développement permettent le développement intégré des Servlet, JSP ou ASP
 - MS Dev Studio
 - IBM VisualAge, Inprise Jbuilder, ...
- Test unitaire
 - HTTPTest, Cactus/JUnit (Jakarta), Simple Test (PHP), ...

Quelques applications standards sur Apache

- **JetSpeed (<http://jakarta.apache.org>)**
 - Jetspeed is an Open Source implementation of an Enterprise Information Portal (or Web Based Groupware).
 - Jetspeed attempts to consume information from multiple resources on the Internet and helps the user manage large amounts of data. This information can come from multiple content types from XML to SMTP to newer protocols such as iCalendar.
 - There are similar commercial products available and range from full blown Portals (my.netscape.com, my.yahoo.com, www.excite.com) to conventional Groupware packages (Lotus Notes, Microsoft Exchange, etc).
- **Java Apache Jyve (<http://jakarta.apache.org/jyve>)**
 - Jyve is a Java Servlet based FAQ-O-Matic system built on top of the Turbine framework. It uses Element Construction Set for Html generation and Village for JDBC access. Currently, Jyve directly supports the MySQL database, but others can be easily support by creating custom schema's for the appropriate databases (interested parties should subscribe to the mailing list and post code there).

Bibliographie

- Beaucoup de Guides, Tutoriels, Manuels
 - <http://ds.internic.net/rfc/rfc2068.txt>
 - <http://search.yahoo.fr/search/fr?p=HTTP>
 - <http://www.w3.org>
 - <http://java.sun.com>
 - <http://java.apache.com>
- Des livres
 - *Attention, ca change très vite !*
 - *Voir Amazon.com*
 - *La traduction en français a au moins 1 an de retard sur la version anglaise*

Bibliographie - Servlets et JSP

- [Hunter] Jason Hunter with William Crawford , « Java Servlet Programming », 1st Edition November 1998, ISBN 1-56592-391-X, Ed : O'Reilly
 - *très nombreux exemples (l 'édition française est chez Oreilly, ISBN 2-84177-082-6)*
- Christopher Taylor, Timothy Kimmitt , « Core Java Web Server (Core Series) », (Novembre 1998) Ed Prentice Hall Computer Books; ISBN: 0130805599
- Dustin R. Callaway, « Inside Servlets : Server-Side Programming for the Java Platform », May 1999, Ed Addison-Wesley Pub Co; ISBN: 0201379635
- Tutorial Java de Sun sur les Servlet (*voir l 'exemple BookStore*) et celui sur J2EE
 - <http://java.sun.com/j2ee/tutorial/>
- De nombreux sites proposent les sources de JSP/Servlet !!
 - Fouillez un peu avant de redévelopper la roue !
 - <http://www.servlet.com>

Bibliographie - Servlets et JSP

- Andrew Patzer, Sing Li, Paul Houle, Mark Wilcox, Ron Phillips, Danny Ayers, Hans Bergsten, Jason Diamond, Mike Bogovich, Matthew Ferris, Marc Fleury, Ari Halberstadt, Piroz Mohseni, Krishna Vedati, Stefan Zeiger , "Professional Java Server Programming: with Servlets, JavaServer Pages (JSP), XML, Enterprise JavaBeans (EJB), JNDI, CORBA, Jini and Javaspaces ", 1121 pages 1st edition (August 1999), Wrox Press Inc; ISBN: 1861002777
- Andrew Patzer , "Programmation Java côté serveur : Servlets, JSP et EJB", Ed Eyrolles-Wrox, 2000, ISBN 1-861002-77-7 (sources des exemples sur www.wroxfrance.com)
- Karl Avedal, Danny Ayers, Timothy Briggs, George Gonchar, Naufal Khan, Peter Henderson, Mac Holden, Andre, Lei, Dan Malks, Sameer Tyagi, Stephan Osmont, Paul Siegmann, Gert Van Damme, Steve Wilkinson, Stefan Zeiger, John Zukowski, Ari Halberstadt, Carl Burnham, John Timney, " Professional JavaServer Pages ", 1st edition (January 15, 2000), Wrox Press Inc; ISBN: 1861003625
- Bill Dudley, Jonathan Lehr, « Jakarta Pitfalls: Time-Saving Solutions for Struts, Ant, JUnit, and Cactus (Java Open Source Library) », Pub. Wiley, July 2003, ISBN: 0-471-44915-6
- Bill Dudley, Stephen Asbury, Joseph K. Krozak, Kevin Wittkopf, J2EE AntiPatterns, Pub Wiley, August 2003, ISBN: 0-471-14615-3

Bibliographie -ASP

- A. Keyton Weissinger, « ASP in a Nutshell, A Desktop Quick Reference », 1st Edition February 1999, ISBN 1-56592-490-8, Ed : O'Reilly
- Shelley Powers, « Developing ASP Components », 1st Edition April 1999, ISBN 1-56592-446-0, Ed : O'Reilly
- Francis, « Active server pages 2.0 », 1999, Ed. Eyrolles, ISBN 2-212-09041-2 (en francais)
- ASP.NET
- De nombreux sites proposent les sources d 'ASP !!
 - Fouillez un peu avant de redévelopper la roue !

Bibliographie -PHP

- Castagnetto , "PHP Professionnel", Editions Eyrolles - 10/2000, ISBN: 2-212-09235-0
- Craig Hilton, Jeff Willis, « Building Database Applications on the Web Using PHP3 » , Ed Addison & Wesley - 12/1999, ISBN: 0-201-65771-6
- Leon Atkinson , « Core PHP Programming » , Ed Prentice Hall - 08/2000, 768 pages, ISBN: 0-13-089398-6