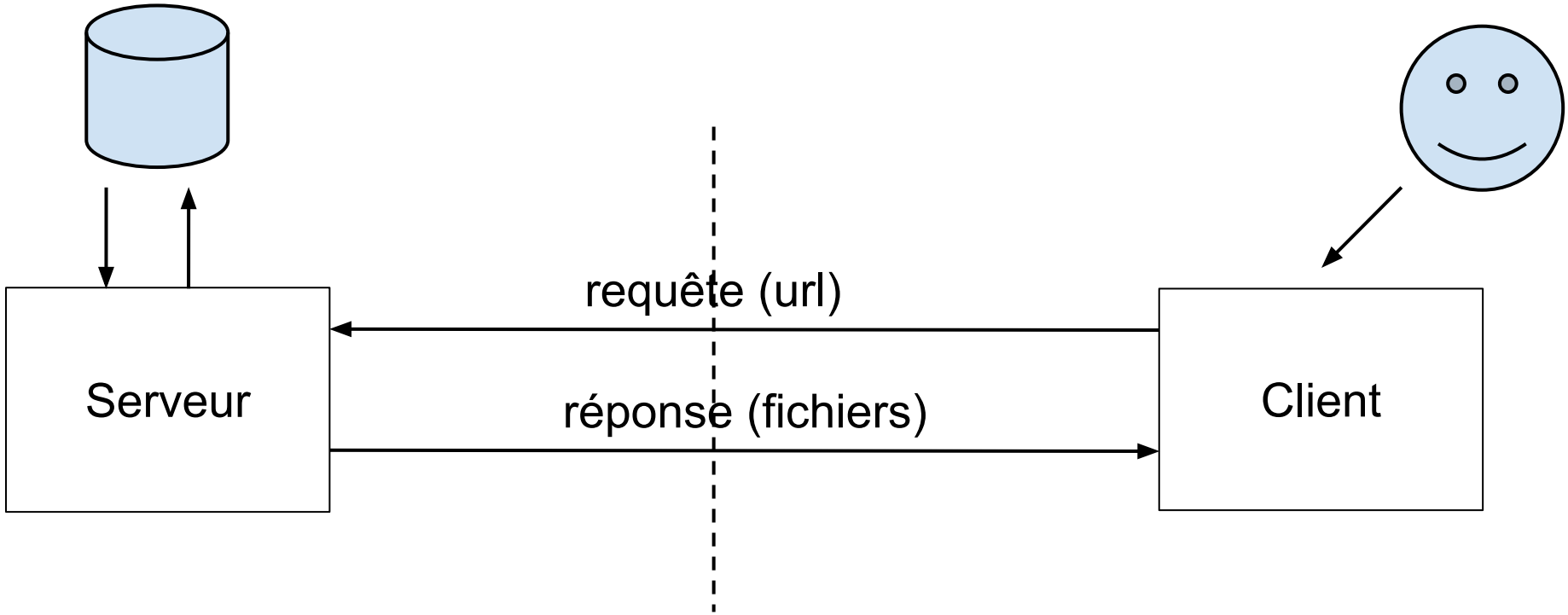
Web dynamique

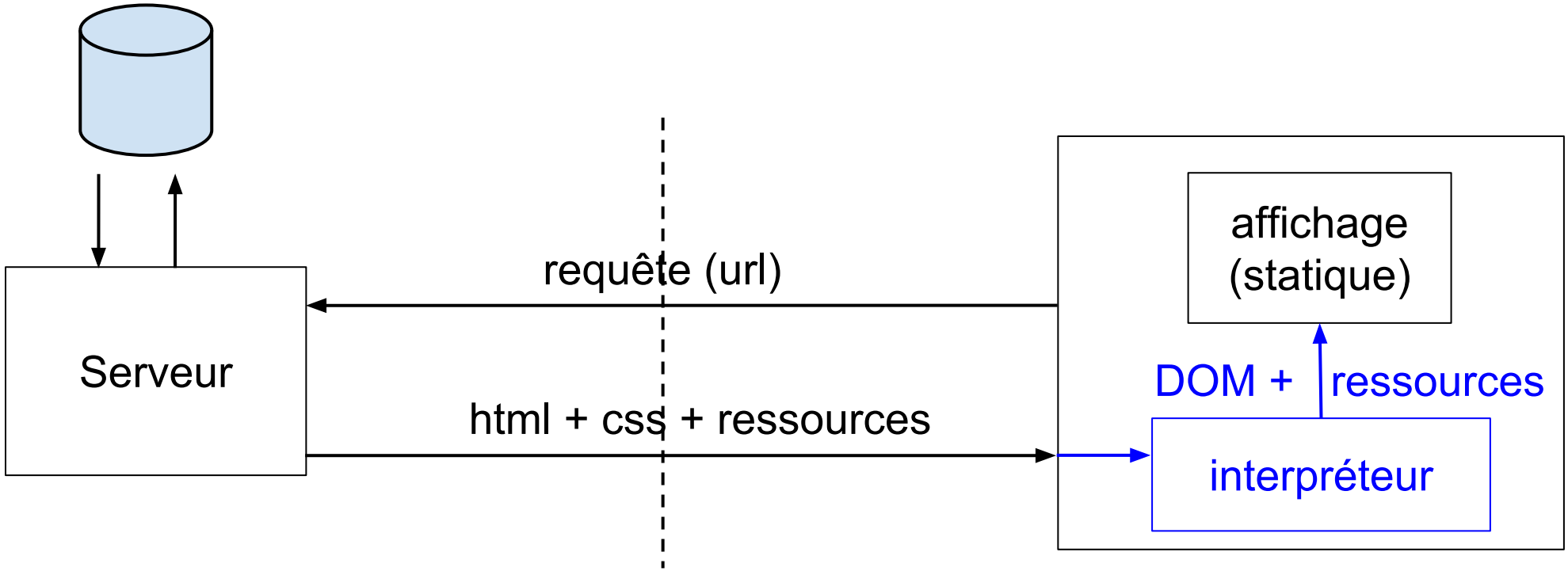
approche client/serveur

Client/serveur



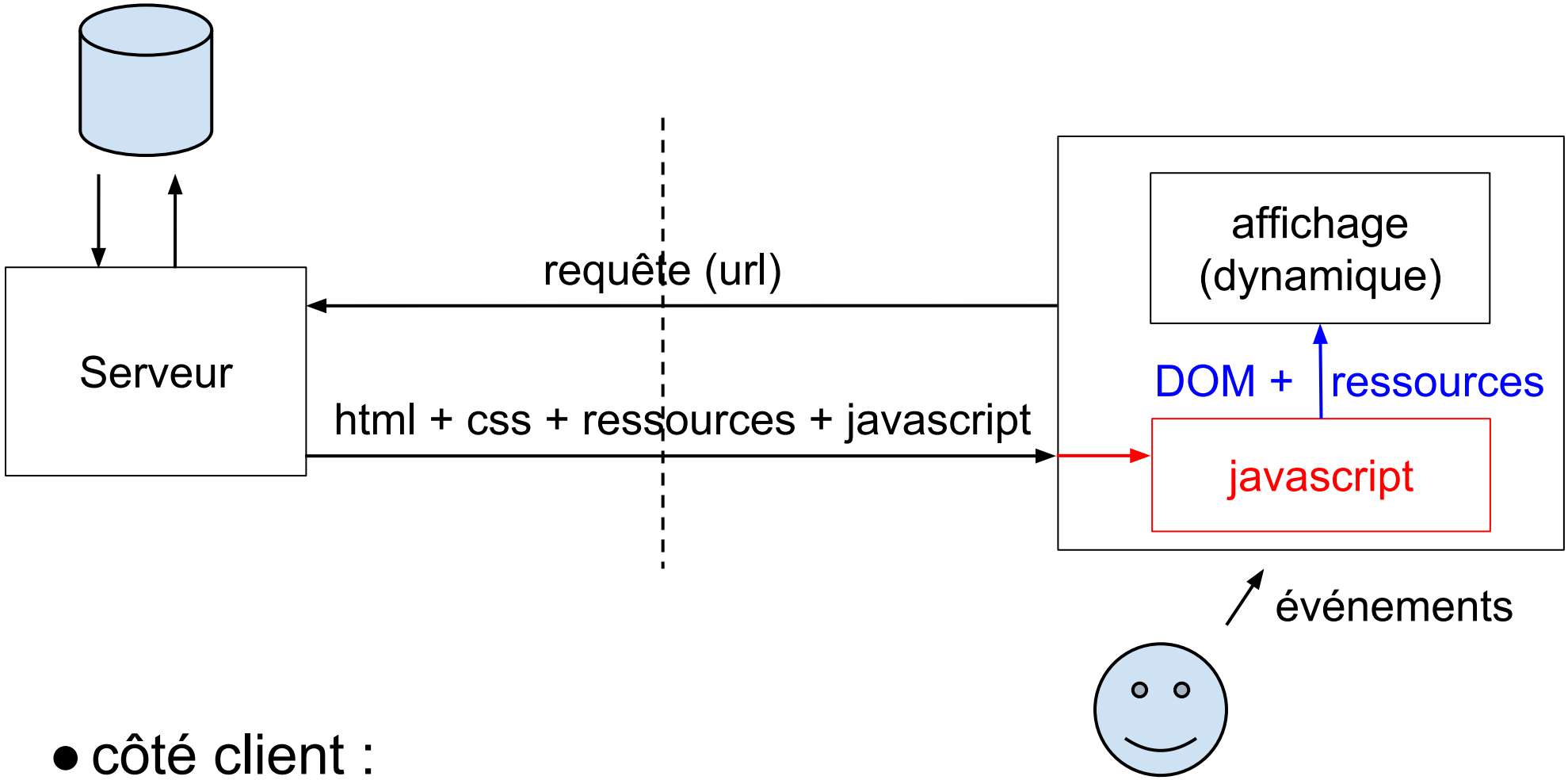
- url = adresse IP + /chemin/vers/fichier
- réponse = fichier (lu sur le DD du serveur)
- Le client gère la mise en page.

html + css



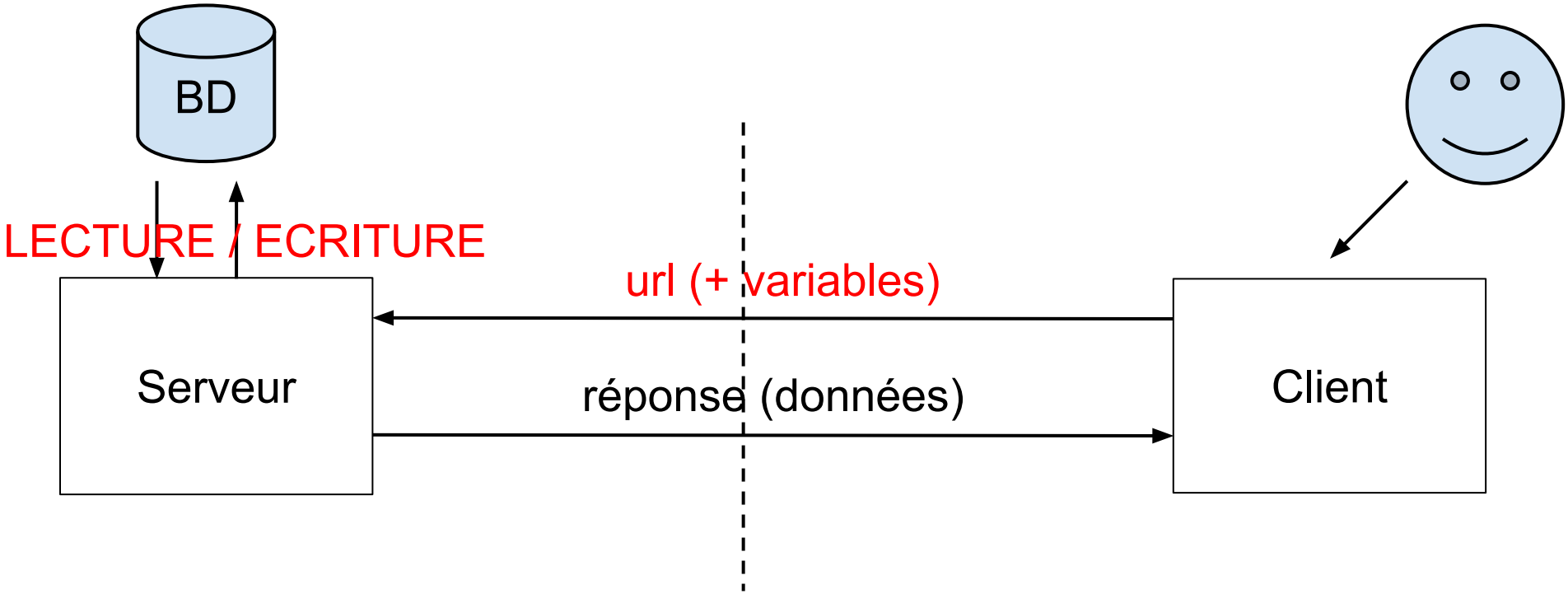
- côté client :
  - construction de l'arbre DOM => bloc = objet
  - affichage (flux)

Pages dynamiques



- côté client :
  - modification du DOM selon les événements produits par l'utilisateur

Web dynamique



- CRUD : Create / Read / Update / Delete
- réponse = données mises en forme au niveau du serveur

Transmission de données "en clair"

- Les variables sont inscrites dans l'URL transmise au serveur:

```
http//mon.adresse.com/mon_site.php?nom=Pignon&prenom=Francois
```

adresse ressource variables

- Le serveur exécute le script (en général du php mais aussi python, java, ...), c'est à dire :
  - traite les variables
  - exécute des opérations de lecture/écriture
  - transmet un contenu au client (en général html mais aussi xml ou json...)

Transmission de données

- variables "POST" (n'apparaissent pas dans l'URL)
- utilisation de Formulaires HTML :
  - le fichier cible est défini comme attribut du formulaire :

```
<FORM method="post" action="cible.php">
    ...
</FORM>
```

  - les balises INPUT définissent les variables à transmettre:

```
<INPUT type="text" name="var1" />
```

  - l'INPUT de type submit lance la requête : la page cible est chargée et remplace la page courante:

```
<INPUT type="submit" value="Envoyer" />
```

php

- Langage de script interprété côté serveur
- Le fichier contenant du php doit posséder l'extension **`.php`**
- Le script est inséré dans du html à l'aide du conteneur :

```
<?php
...
?>
```

- en entrée : les variables transmises (méthode GET ou POST)
- en sortie : du texte :
  - **`echo "Salut la Terre!";`**
  - **`echo "J'ai $mon_age ans! <BR/>";`**
  - **`echo "{nom : $nom, liste : [$a, $b, $c]}"`**
- remarque :
  - le client reçoit un fichier html où les zones de script sont remplacées par le texte généré par echo
  - ne pas oublier d'insérer des balises html pour la mise en forme côté client

Exemple 1

```
<html>
<head>
    <title> Salut! </title>
</head>
<body>
<?php
echo "Salut la Terre!";
// echo Salut la terre!"
?>
</body>
</html>
```

Variables php

- Les variables
  - commencent par $
  - sont interprétées dans les chaînes : `"J'ai $age ans!";`

- Variables externes : celles qui sont transmises
  - par url: `http://bla.bla/site.php?nom=Pignon&prenom=Francois`

```
-->  $nom = $_GET['nom'];
     $prenom = $_GET['prenom'];
```

  - ou formulaire :

```
$var1 = $_POST['var1'];
```

- Variables internes : déclarées dans le script :

```
$n = 100;
for ($i=0; $i<$n; $i++) { ... }
```

Exemple 2

formulaire.html :

```
<form action = "bonjour.php" method ="post">
Votre nom : <input type="text" name="nom"/> <br/>
Votre prénom : <input type="text" name="prenom"/> <br/>
<input type="submit" value="Envoyer">
</form>
```

bonjour.php

```
<?php
   echo "Bonjour, ".$_POST["prenom"]." ".$_POST["nom"]."
!!!";
?>
```

Exemple 3 : table de multiplications

```
<?php
echo '<table width="100%">';
for ($i=1;$i<=10;$i++){
    echo "<tr>";
    for ($j=1;$j<=10;$j++){
        echo "<td>". $i * $j ."</td>";
        }
    echo "</td>";
    }
echo '<caption> Table de multiplication </caption>';
echo '</table>';
?>
```

Session php

Voir sessions php

- Tableau `$_SESSION` persistant d'une page à l'autre.
- ... pour toutes les pages possédant l'en-tête `session_start` :

```
<?php
session_start();
?>
<!DOCTYPE  ...
```

- pour tester une valeur :

```
if (!isset($_SESSION['pseudo'])){
...
```

- pour effacer le tableau :

```
session_destroy()
```

Pour conserver l'information plus longtemps...

- Les informations fournies par l'utilisateur peuvent être stockés dans un fichier (ex : fichier dbm).

  voir la recette

- La plupart du temps, on utilisera des *bases de données* :
  - plus robuste
  - plus sécurisé
  - plus rapide

php + base de données!

- C'est la recette du succès!

- Le **web dynamique** repose sur :
  - la *mise à jour* en continu d'informations
  - gérées par une *base de données*
  - tout se passe côté *serveur*

- php/mysql est la combinaison la plus utilisée pour cette fonctionnalité

Exemple avec SQLite3

un simple fichier situé dans le répertoire courant sert à stocker les informations:

Connexion à la base:

```
$db = new SQLite3("ma_base.db");
```

où `ma_base.db` est située dans le même dossier que le script php

attention :
le fichier ma_base.db doit être accessible en lecture/*écriture* :

```
chmod 777 ma_base.db
```

Exécuter une requête

- Requête lecture/écriture : INSERT, UPDATE ou DELETE

```
$db->exec("INSERT INTO Utilisateur VALUES
('Joe','Dalton')");
```

retourne un booléen

- Requête SELECT : retourne une liste de tuples

```
$reponse = $db->query('SELECT * FROM Livre');
```

Exemple

```
<?php

$db = new SQLite3('biblio.db');

$reponse = $db->query('SELECT * FROM Livre');

while ($tuple = $reponse->fetchArray())
  {
    echo " Titre : $tuple[2] - ";
    echo " Editeur : $tuple[3] <BR/>";
  }

?>
```

Exemple : poster un message

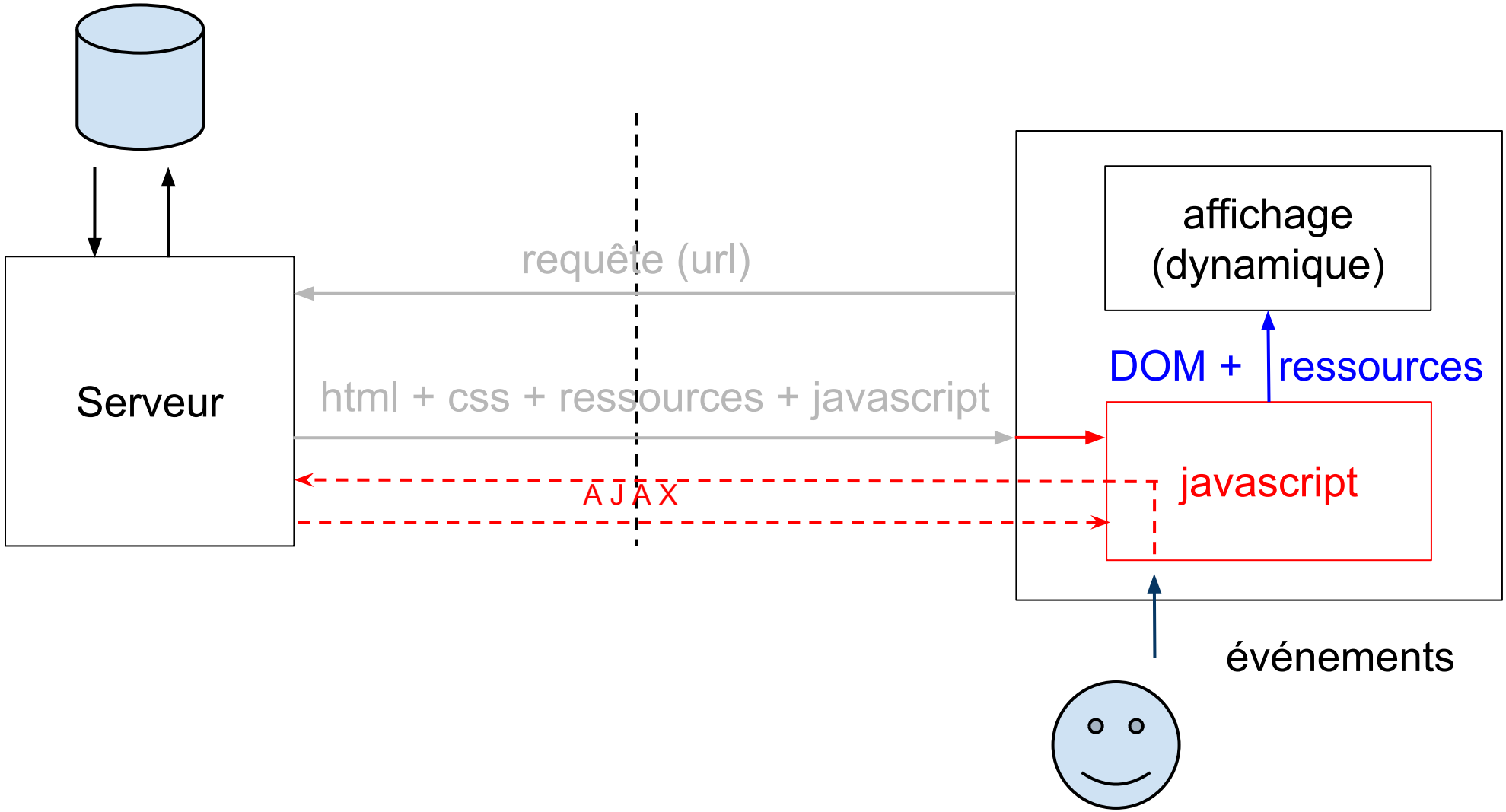
formulaire.html :

```
<form action = "insere.php" method ="post">
Votre pseudo : <input type="text" name="pseudo"/> <br/>
Votre message : <textarea name="message"/> <br/>
<input type="submit" value="Envoyer">
</form>
```

insere.php:

```
<? php
    $db = new SQLite3("message.db");
    $auth = $_POST['pseudo'];
    $mes =  $_POST['message'];
    $db->exec("INSERT INTO Message values ('$auth','$mes')");
   $reponse = $db->query('SELECT * FROM Message');
   echo "<h3>Liste des messages :</h3>";
   while ($tuple = $reponse->fetchArray())
    {
    echo " <b>Auteur</b> : $tuple['auteur'] <BR/> ";
    echo " <b>Texte</b> : $tuple['texte'] <BR/>";
    }
?>
```

jquery + ajax


Serveur
requête (url)
html + css + ressources + javascript
AJAX
affichage
(dynamique)
DOM +
ressources
javascript
événements

jquery + ajax

http://babylon-design.com/apprendre-et-comprendre-jquery-3-3/
http://api.jquery.com/category/ajax/

```
$.ajax({
   type: "POST",
   url: "test.html",
   success:
     function(retour){
      alert("Données retournées : " + retour );
   }
});
```

Requête sur un lien

```
$("a.test").click(function() {
   $.ajax({
      type: "POST",
      url: $(this).attr("href"),
      success: function(retour){
         $("#recipient").empty().append(retour);
      }
   });
   return false;
});
```

Requête sur un formulaire

```
$("form.test").submit(function() {
s = $(this).serialize();
$.ajax({
       type: "POST",
       data: s,
       url: $(this).attr("action"),
       success: function(retour){
          $("#recipient").empty().append(retour);
       }
    });
return false;
 });
```