

COGNITIS Group

WebSphere et VisualAge Pacbase

Présentation de STRUTS

Bruno CLEMENT
IBS
bclement@cognitis.fr

WebSphere software



Sommaire

- ❑ Cognitis et VisualAge Pacbase/IBM
- ❑ Les architectures n-tiers
- ❑ Le modèle MVC
- ❑ Présentation du projet Struts
- ❑ Pourquoi Struts
- ❑ Les outils autour de Struts
- ❑ Etude de cas - CSM

Cognitis et VisualAge Pacbase/IBM

Expertise avérée sur VisualAge Pacbase depuis plus de 10 ans

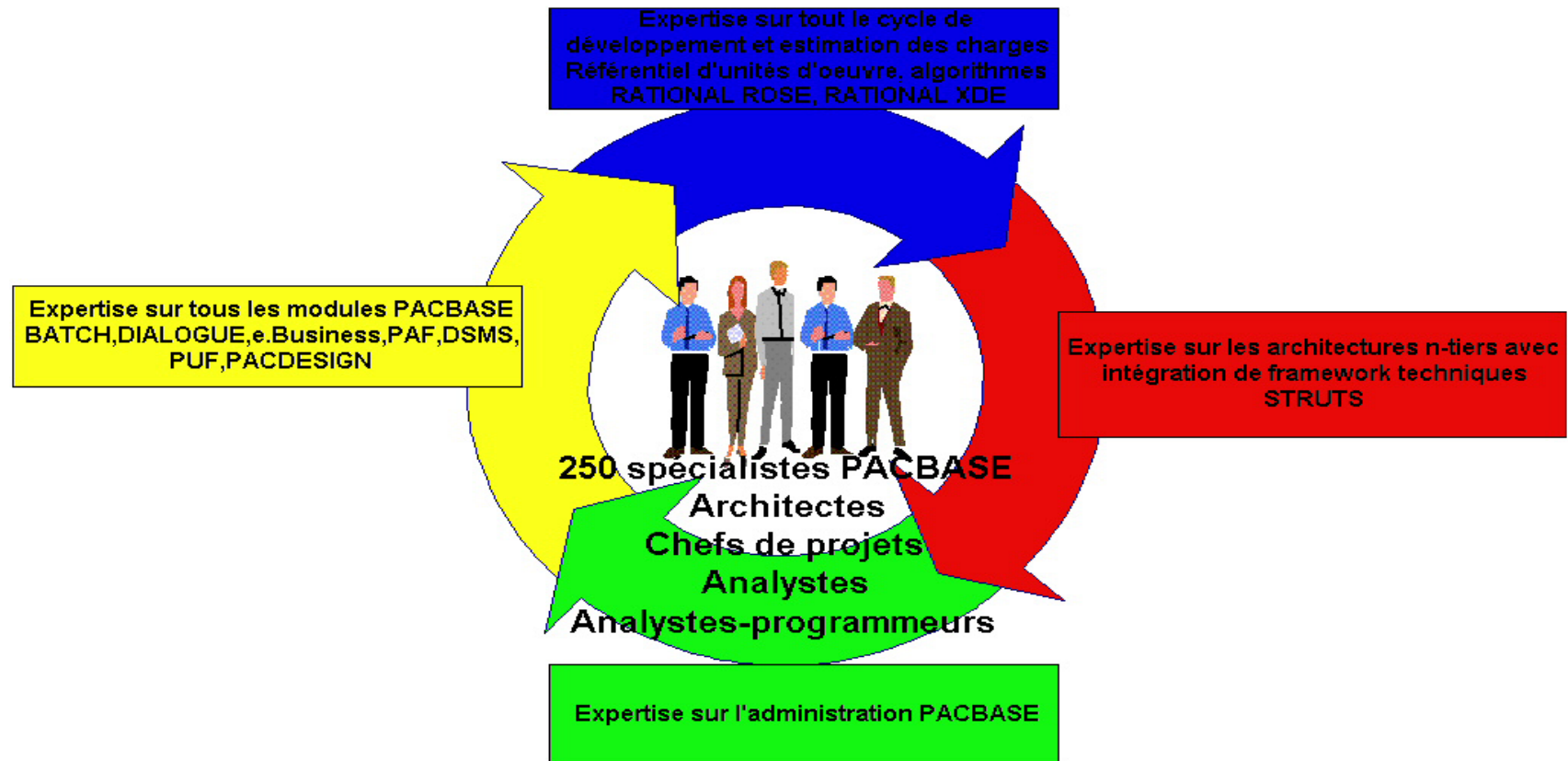
Partenaire « PREMIER » d'IBM de longue date

Relation privilégiée avec le laboratoire IBM Paris

IBM Training Provider sur VisualAge Pacbase

Hotline VisualAge Pacbase en Espagne, Portugal, USA, Canada, Mexique

Partenaire de référence sur la version VisualAge Pacbase V3 et le module e-business



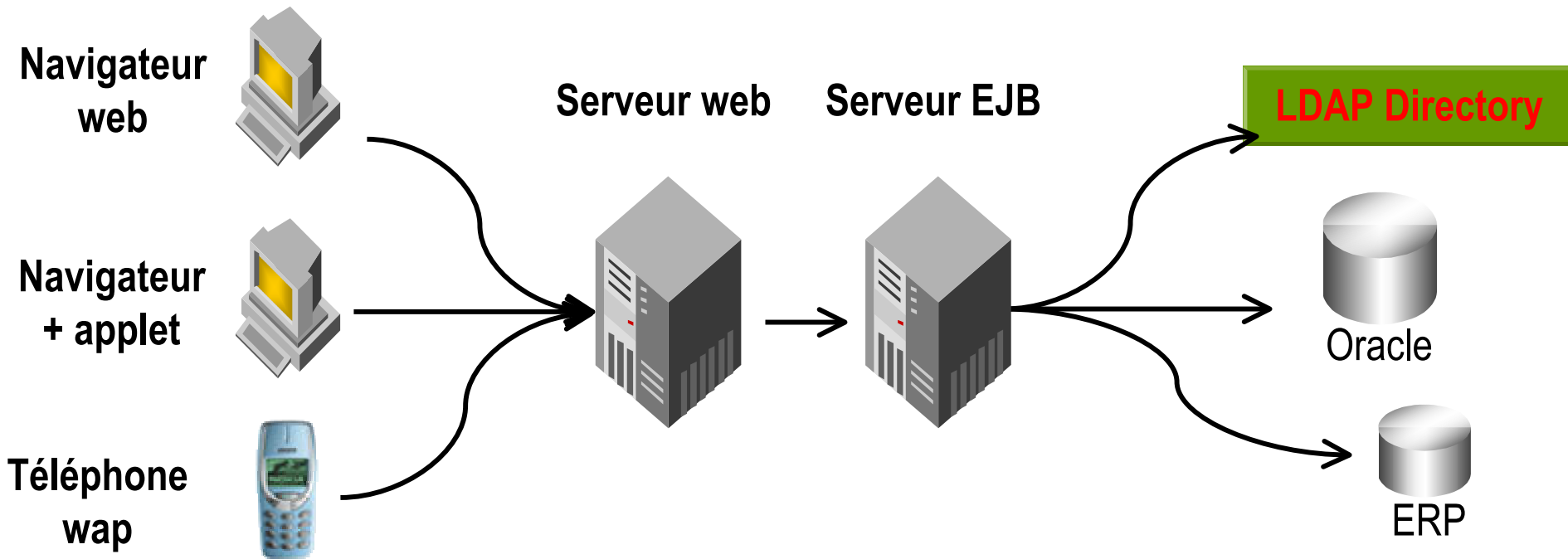
Les Architectures n-tiers

Vue globale de l'architecture

Présentation

Traitements

Données



Les Architectures n-tiers

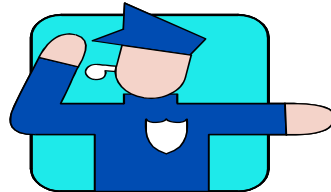
Détail des couches

Interface



• *interface
utilisateur*

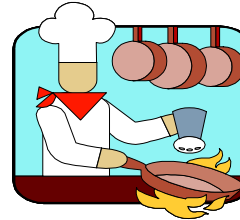
Présentation
Navigation



*gestion de
contexte
appels de
services

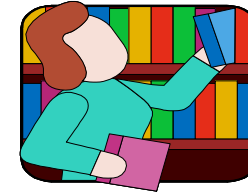
contrôles
locaux de saisie*

Logique
Applicative



• *règles de
gestion*

Données



• *stockage et
gestion des
données*

Le modèle MVC

Principes

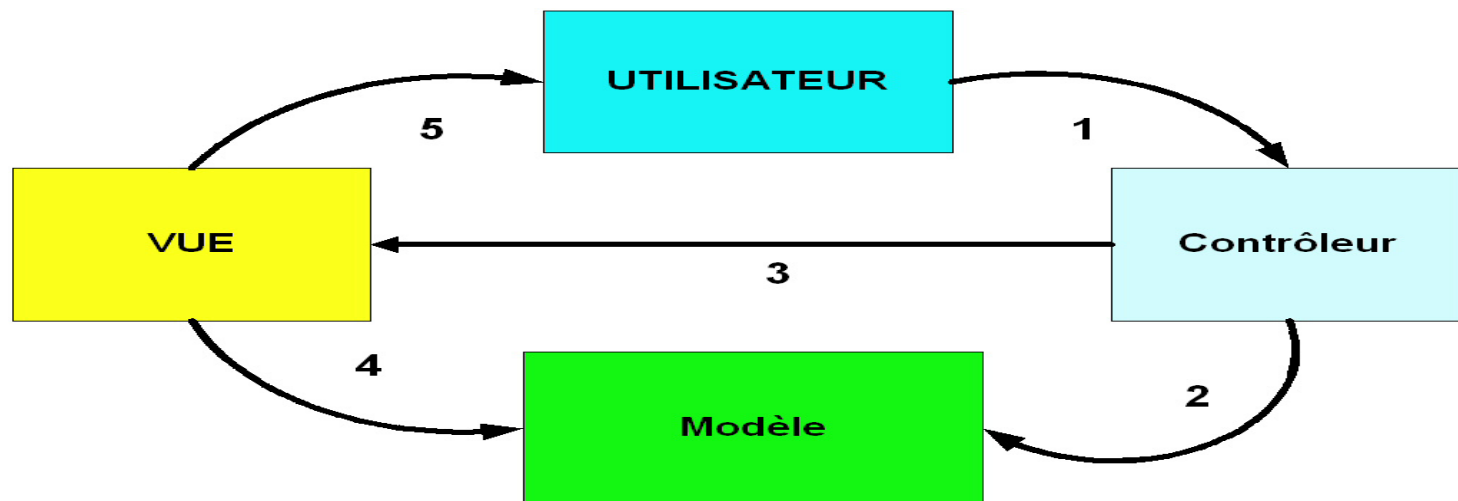
Ce paradigme **Modèle/Vue/Contrôleur** a pour objectif de faire collaborer deux équipes:

- les infographistes (interface utilisateur)
- Les informaticiens (cinématique)

MVC est un modèle de programmation qui propose:

- ✓ Le modèle contenant la logique et l'état de l'application
- ✓ La vue représentant l'interface utilisateur
- ✓ Le contrôleur gérant la synchronisation entre la vue et le modèle

CETTE SEPARATION FAVORISE LE DEVELOPPEMENT ET LA MAINTENANCE DES APPLICATIONS

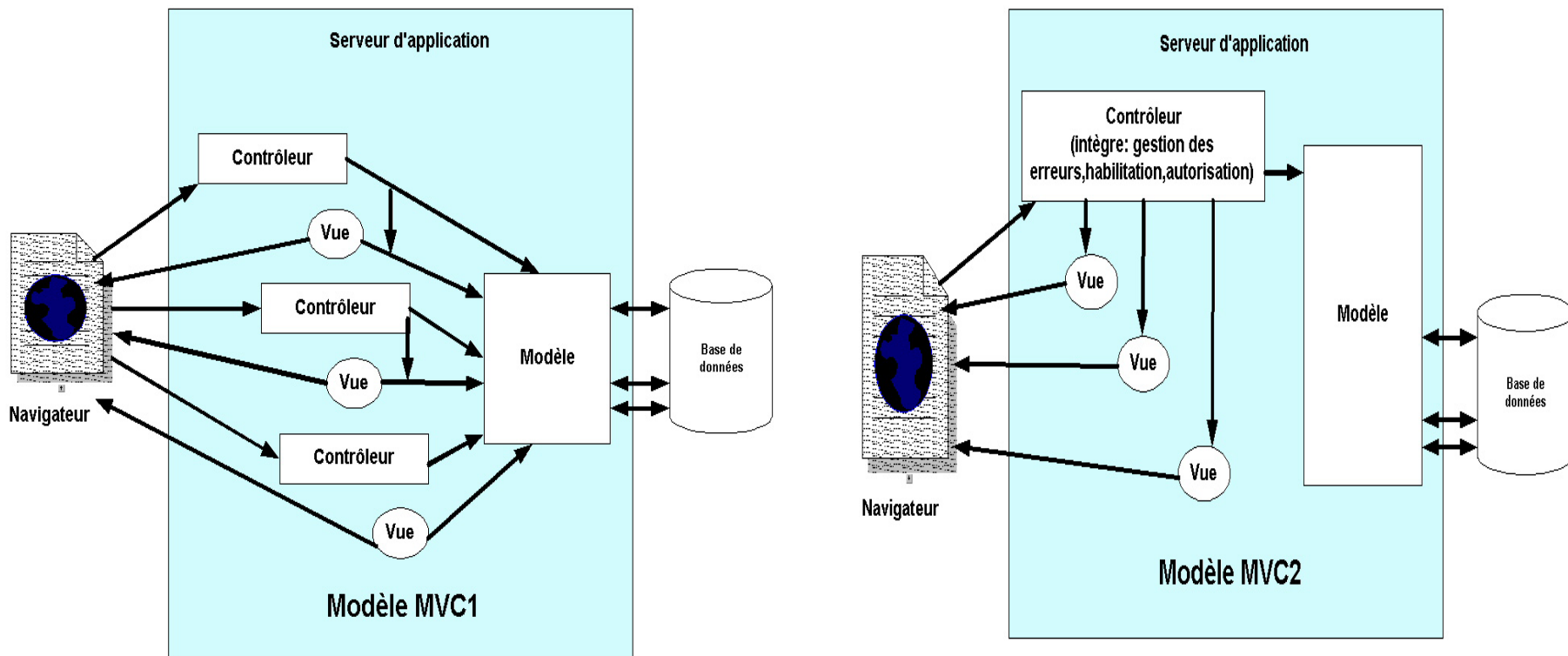


Le modèle MVC Evolution

Le modèle **MVC1** n'est cependant pas encore idéal car oblige à écrire une multitude de servlets qui sont autant de points d'entrée dans l'application.

APPARITION DE FRAMEWORK composé d'un seul servlet (donc d'un seul contrôleur) appelé **MVC2**.

BARRACUDA – HAMMOCK – TAPESTRY – WEBWORK - STRUTS



Le Projet STRUTS

Le monde Open Source



- STRUTS désigne un framework technique développé en open source dans le cadre du projet JAKARTA de l'ASF (Apache Software Foundation)
- Un framework est une structure d'accueil composée de boîtes à outils (bibliothèques de composants réutilisables) et de modèles de conception « design patterns »
- Un framework technique apporte les concepts, entités et mécanismes qui permettent de s'abstraire d'un certain nombre de problématiques conceptuelles et techniques récurrentes.

Le Projet STRUTS

Son contenu



- Un framework composé de classes java, de bibliothèques de tags JSP et de servlets :
- un contrôleur configurable permettant d'associer des actions à des requêtes HTTP (STRUTS CONFIG.XML)
- des bibliothèques de tags spécifiques pour créer des vues (JSP)
- Un digester pour parser un fichier XML et d'en récupérer les informations voulues
- des utilitaires pour remplir automatiquement des champs et supportant plusieurs langages

Le Projet STRUTS

Les TAGLIBS

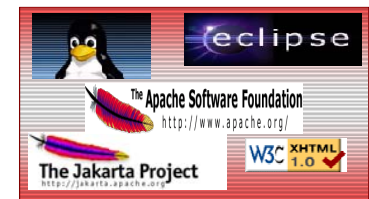


Les vues sous STRUTS sont des pages JSP, les Taglibs permettent la création de pages JSP de manière aisée **sans code java** pour le designer

- HTML
- BEAN
- LOGIC
- TEMPLATES

Le Projet STRUTS

TAG HTML




Beer For All - Microsoft Internet Explorer provided by BellSouth

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Media History Mail Print

Address http://localhost:8080/beer4all/action/checkout?method=checkout

 *Beer for All*

Powered by **Struts**

Would you like to sign in?
 Items in shopping cart: 1
 Current Total: \$5.25

! To order this beer, enter a quantity below and click on the "Add to Cart" button.

Shipping Address

Address:

City:

State:

Postal:

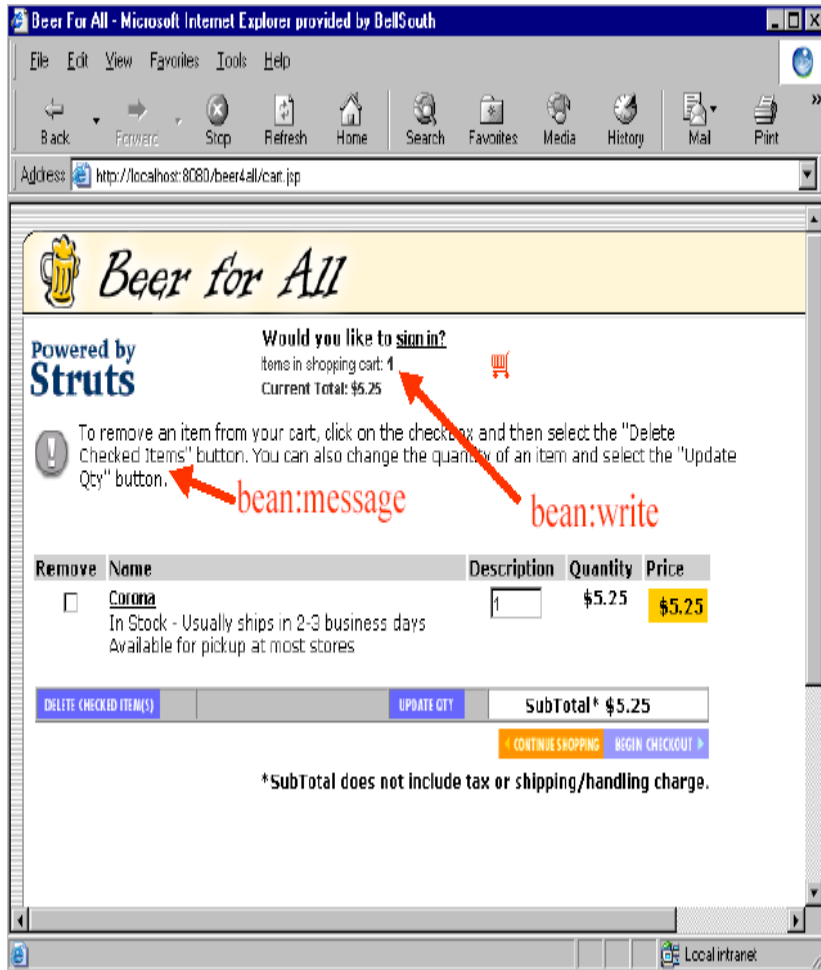
Local intranet

← **html:text**

```
<table>
  <tr><b>Shipping Address</b></tr>
  <tr><td>Address:</td>
    <td><html:text name="checkoutForm" property="shippingAddress" />
  </td></tr>
  <tr><td>City:</td>
    <td><html:text name="checkoutForm" property="shippingCity" />
  </td></tr>
  <tr><td>State:</td>
    <td><html:text name="checkoutForm" property="shippingState" />
  </td></tr>
  <tr><td>Postal:</td>
    <td><html:text name="checkoutForm" property="shippingPostal" />
  </td></tr>
</table>
```

Le Projet STRUTS

TAG BEAN



<bean:write name="UserContainer" property="cart.size" scope="session"/>

Current Total:

\$<bean:write name="UserContainer" format="#,##0.00"
property="cart.totalPrice" scope="session"/>

Le Projet STRUTS

TAG LOGIC



```
<logic:iterate id="cartItem" scope="session" name="UserContainer" property="cart.items">
```

```
...
```

```
<html:link
```

```
page="/action/viewitemdetail?method=viewItem"
```

```
paramName="cartItem"
```

```
paramId="id"
```

```
paramProperty="id">
```

```
<bean:write name="cartItem" property="name"/>
```

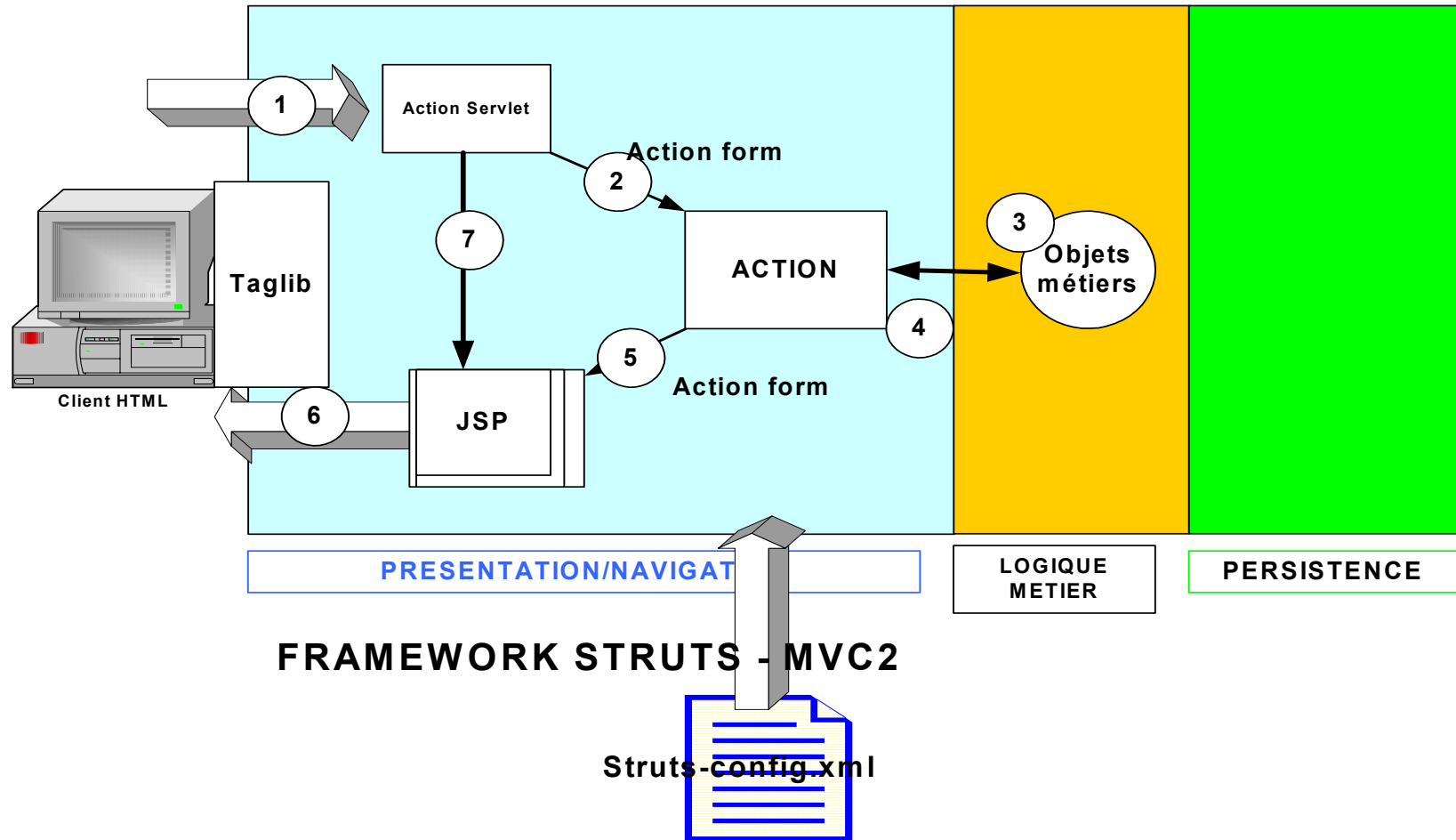
```
</html:link>
```

```
...
```

```
</logic:iterate>
```

Le Projet STRUTS

Son Architecture



Pourquoi STRUTS

- Evite d'avoir à insérer du code Java

- `<html:text property="id_employe"/>`

au lieu de :

```
<input type="text" name ="id_employe"
      value="<%= addMemberForm.getId_employe() %>" />
```

- Inclut le multilinguisme

- Les tags `<bean :message>` avec les attributs « key » permettent d'aller rechercher les messages dans les fichiers de ressources selon diverses langues (_fr,_en,_es, etc...)

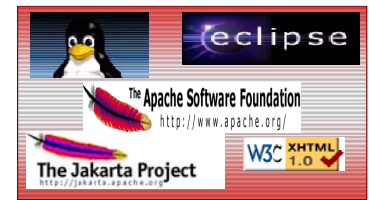
- Inclut le traitement d'erreurs

- Les tags `<html :errors>` permettent d'afficher les éventuels messages d'erreur qui ont été stockés dans la session par les composants métiers

- Nombreux assistants

- Ex: Easy Struts, Struts-Layout

Le Projet STRUTS – Outils autour



STRUTS-LAYOUT (IMPROVE)

Mise en page automatique des formulaires par les tags sans écriture de code HTML.
Avec la bibliothèque Struts-Layout, possibilité de changer de façon dynamique

EASY-STRUTS (SOURCEFORGE)

Fournit un éditeur de fichier de configuration via un plug-in J2EE compatible avec WSAD 5

PAC-STRUTS (COGNITIS) :

Framework technique fournissant un lien entre STRUTS et la Proxy PACBASE - **CSM**

WAM (IBM)

Nouveau générateur proposé par IBM construit sur STRUTS et alimenté par différentes sources de données (SGBD, Proxy Pacbase)

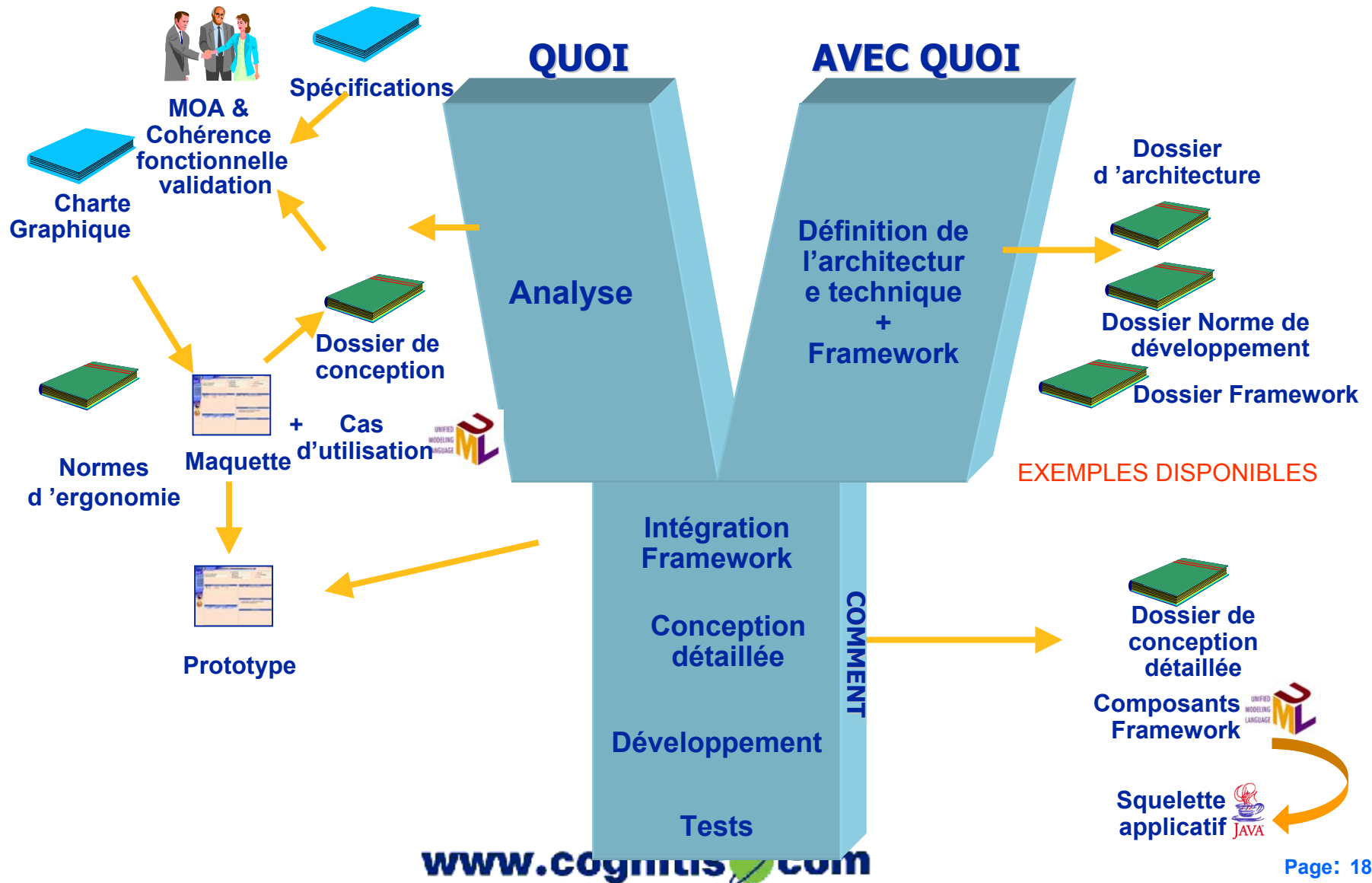
Etude de cas - CSM



- Utilisation de VisualAge Pacbase V3 e-business
 - 3 dossiers
 - 5 serveurs élémentaires
 - 8 tables DB2 6.1
 - z/OS, moniteur transactionnel CICS TS1.3
- Utilisation de WSAD, Rose 2000 + pont
- Projet pilote ODM : 6 cas d'utilisation dont 3 utilisant le module e-business
- Planning: Semaine 46 – Semaine 51 (1 mois)

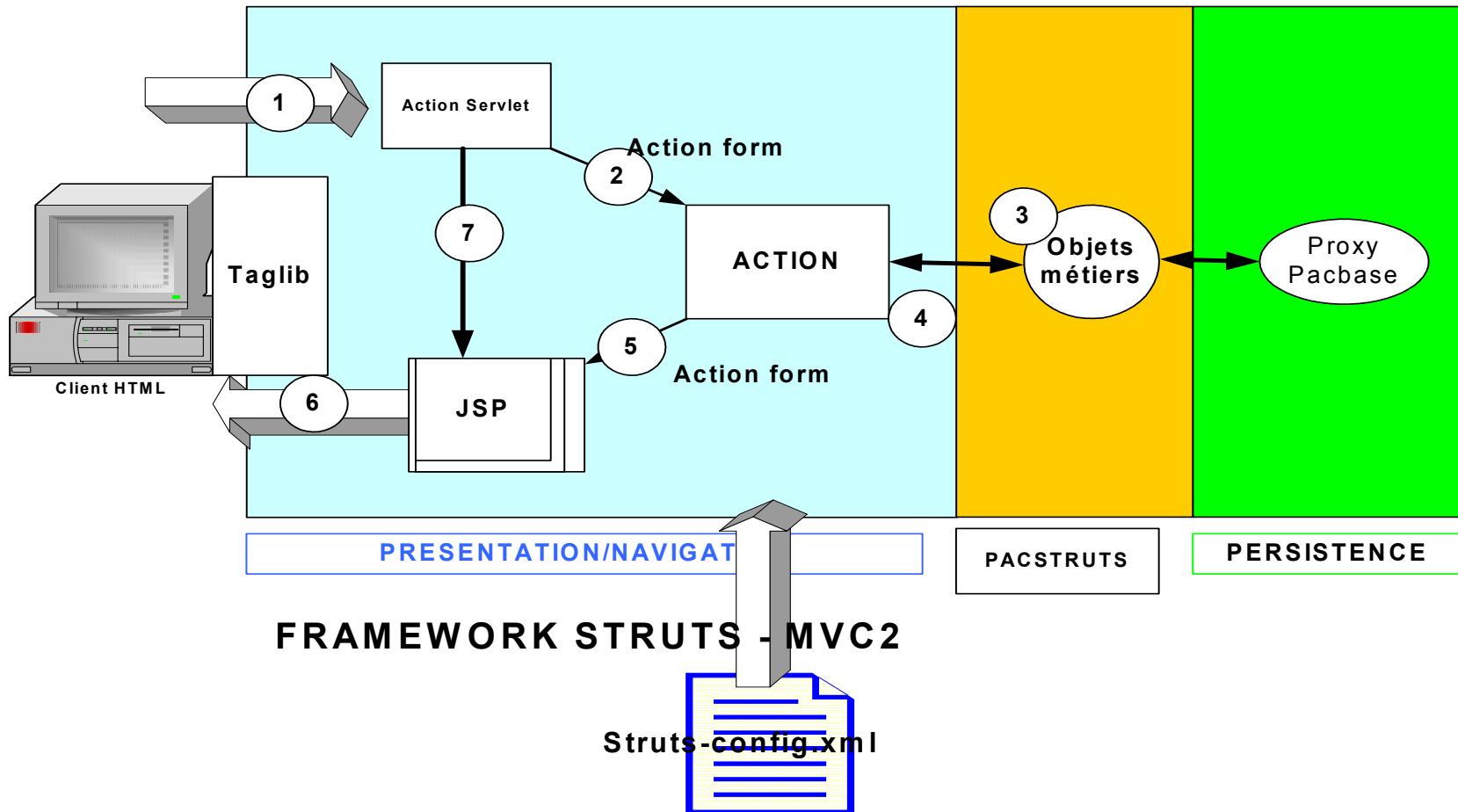
Etude de cas – CSM

Démarche projet utilisée



Etude de cas - CSM

Architecture Logique générale



Pré-Bilan

Le Pré-Bilan



- **Points positifs**
 - Réutilisabilité
 - Maintenance facilitée
 - Appropriation de STRUTS
 - Planning respecté

- **Points à améliorer**
 - Pas assez automatisé
 - ➔ vers WAM

Merci de votre attention



111, rue du Château des Rentiers 75013
Paris

Tel 01 53 61 64 00 Fax 01 53 61 64 01

WebSphere software

IBM