

APPLICATION MULTIMODULE

Refactoring

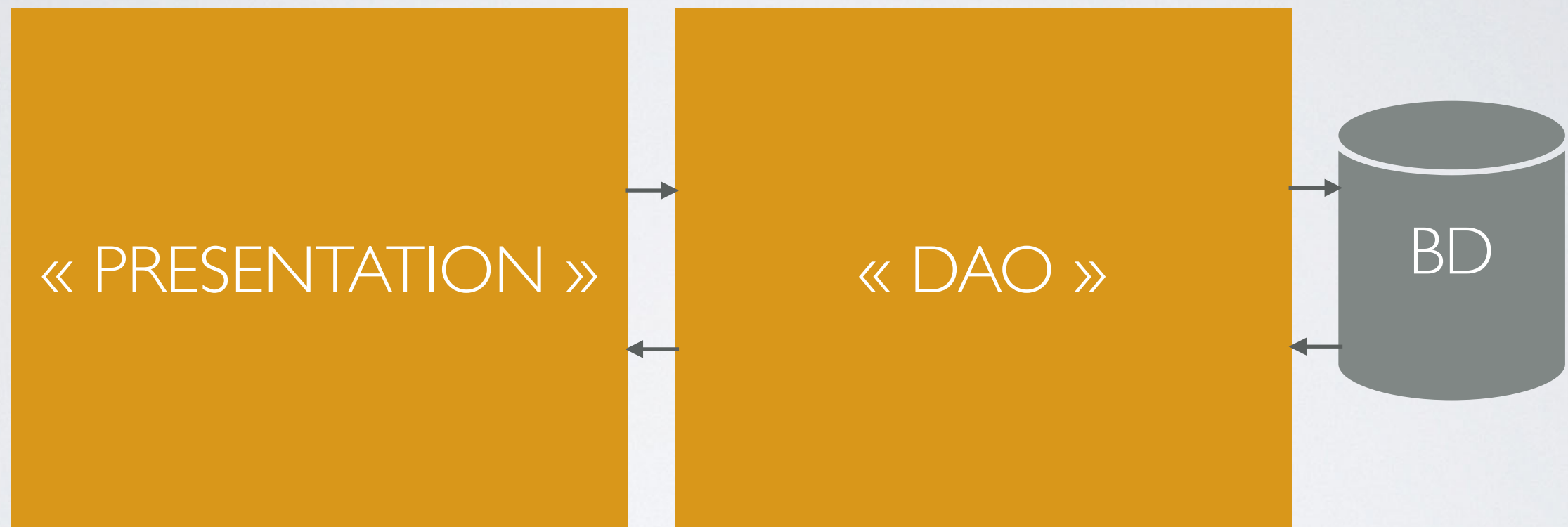
PRÉSENTATION

Cette application simple permet la saisie de nouvelles formations par l'utilisateur. Les données sont injectées dans en base et apparaissent dans la partie inférieure de la fenêtre.



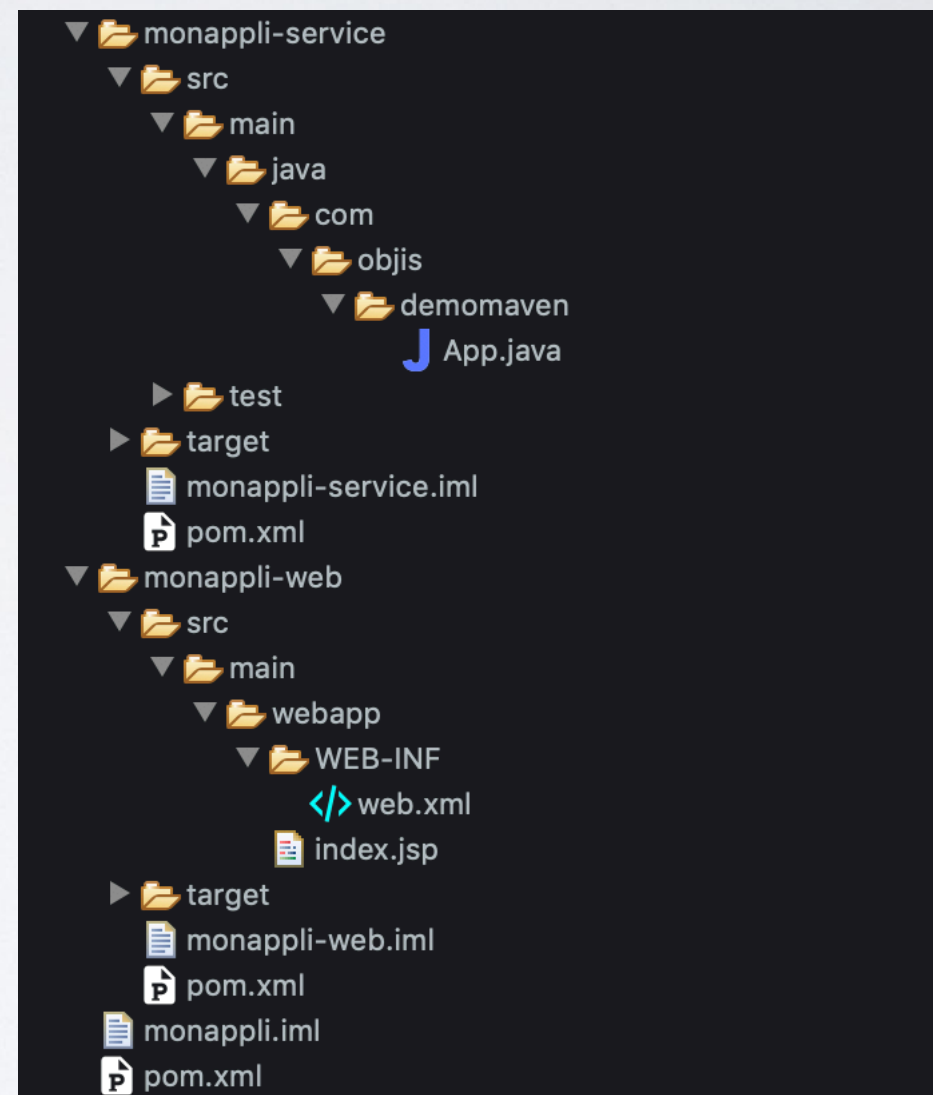
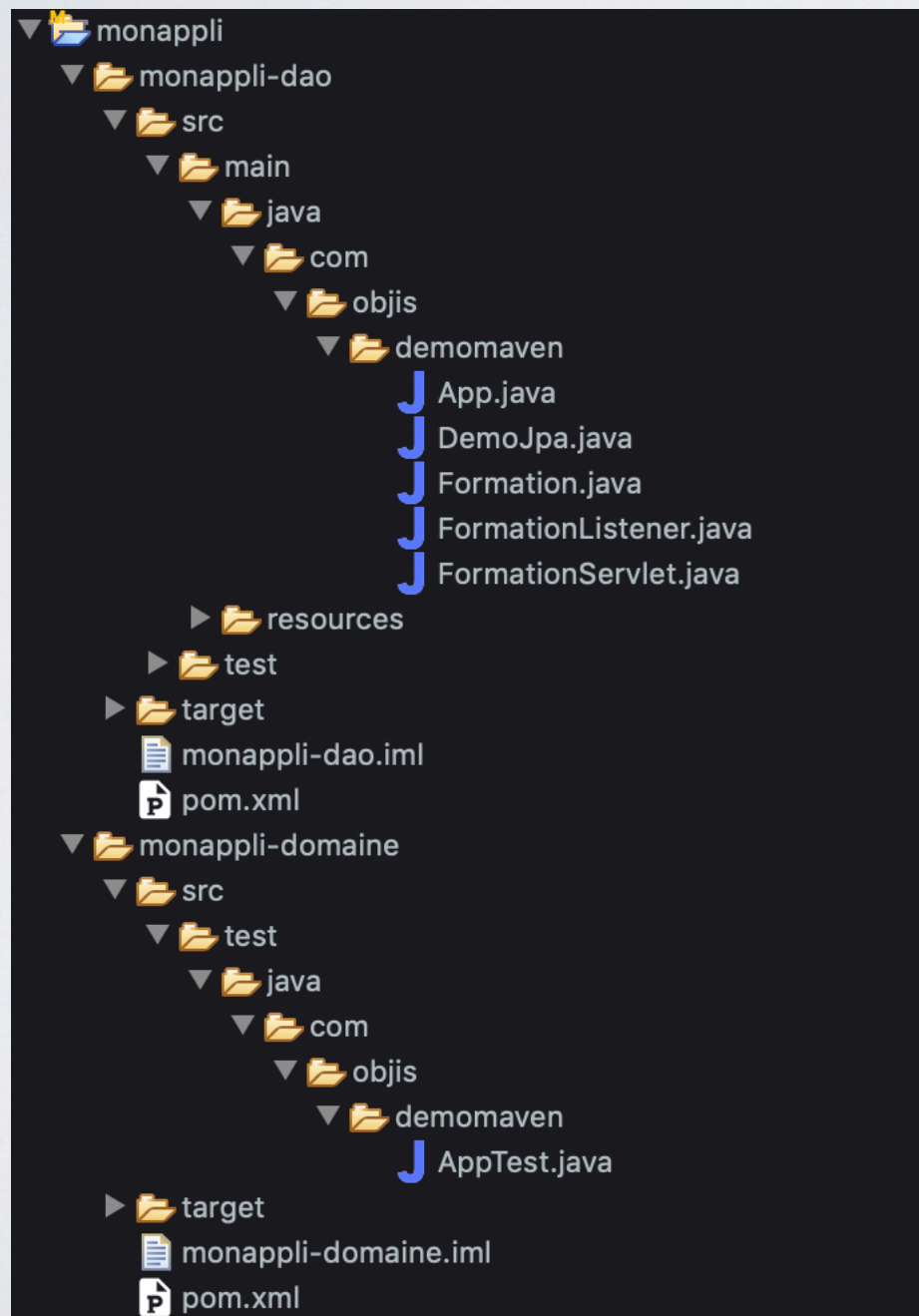
The screenshot shows a web application interface. At the top, there is a label "Thème Formation:" followed by a text input field and a button labeled "Ajouter". Below this, there are two horizontal lines, likely representing a table or list structure. The rest of the page is empty, suggesting a list of added items or a confirmation message.

ARCHITECTURE ACTUELLE



Toutes les classes se situent dans la couche DAO, même des classes de test. Seule la page d'accueil de l'application (index.jsp) se situe dans Présentation.

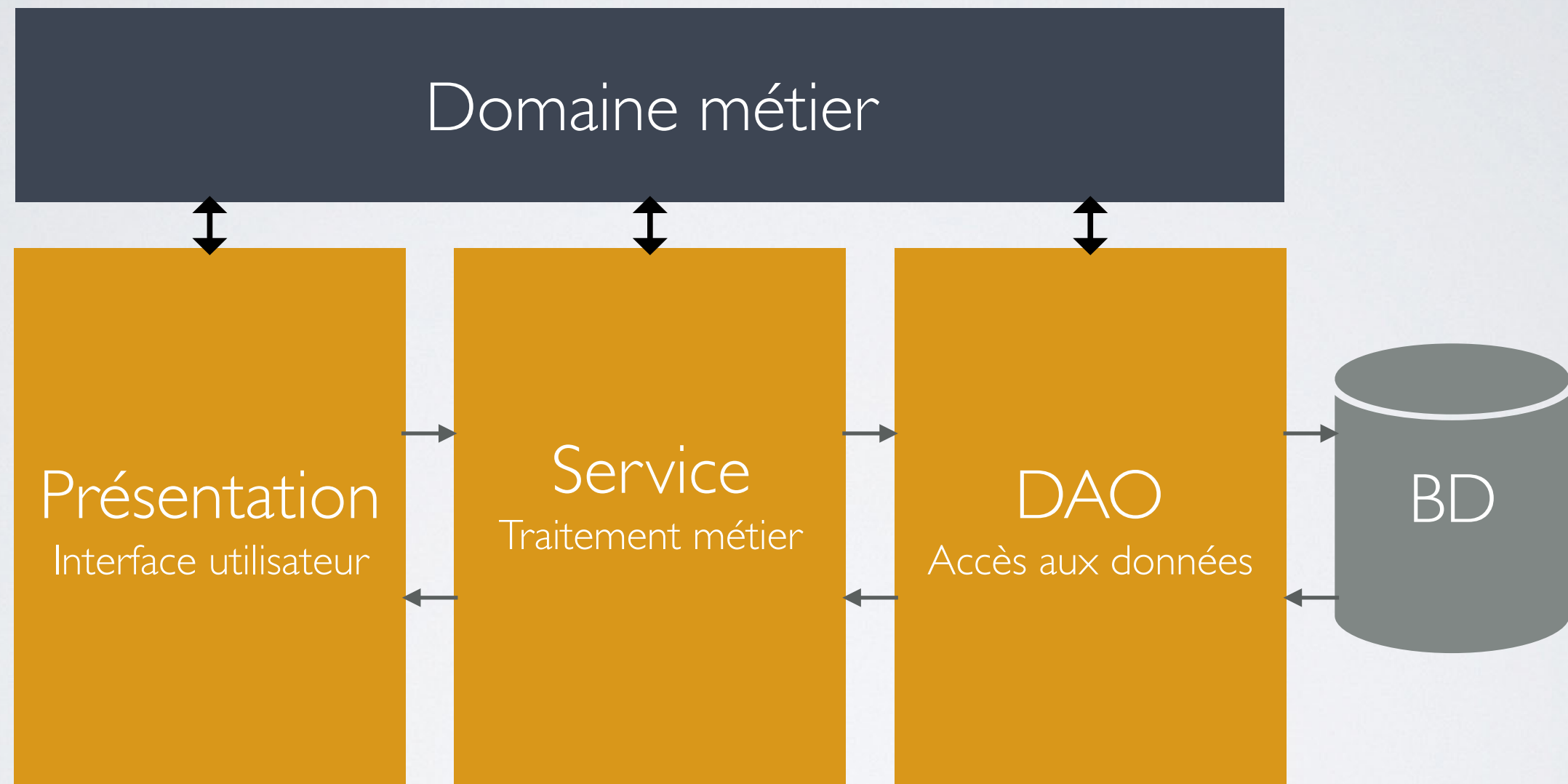
ARCHITECTURE ACTUELLE



ARCHITECTURE ACTUELLE

Avantage	Inconvénients
<i>Fonctionnel</i>	<i>Évolutivité</i>
	<i>Maintenance</i>
	<i>Sécurité</i>

ARCHITECTURE CIBLE



OBJECTIFS

Présentation
Interface utilisateur

- Décoder les paramètres envoyés par le client
- Envoyer les réponses aux services pour travailler les informations
- Encoder les réponses des services pour qu'elles puissent être lues ou affichées par le client.

OBJECTIFS



Service
Traitement métier

- Correspond à la partie fonctionnelle de l'application, celle qui implémente la logique métier, et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche de présentation.
- Valide les paramètres qui sont transmis à la couche d'accès aux données

OBJECTIFS

DAO

Accès aux données

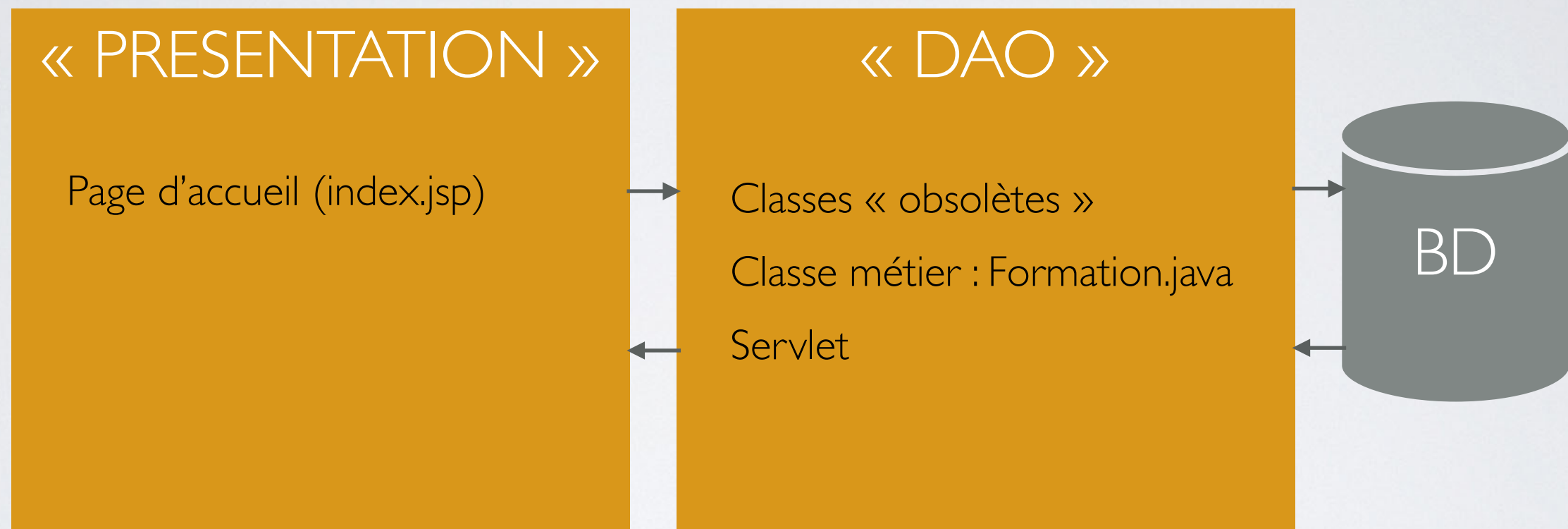
- Correspond à la partie gérant l'accès aux données de l'application. Ces données peuvent être propres à l'application, ou gérées par une autre application.
- La couche Service n'a pas à s'adapter à ces deux cas, ils sont transparents pour elle, et elle accède aux données de manière uniforme (couplage faible).

OBJECTIFS

Domaine métier

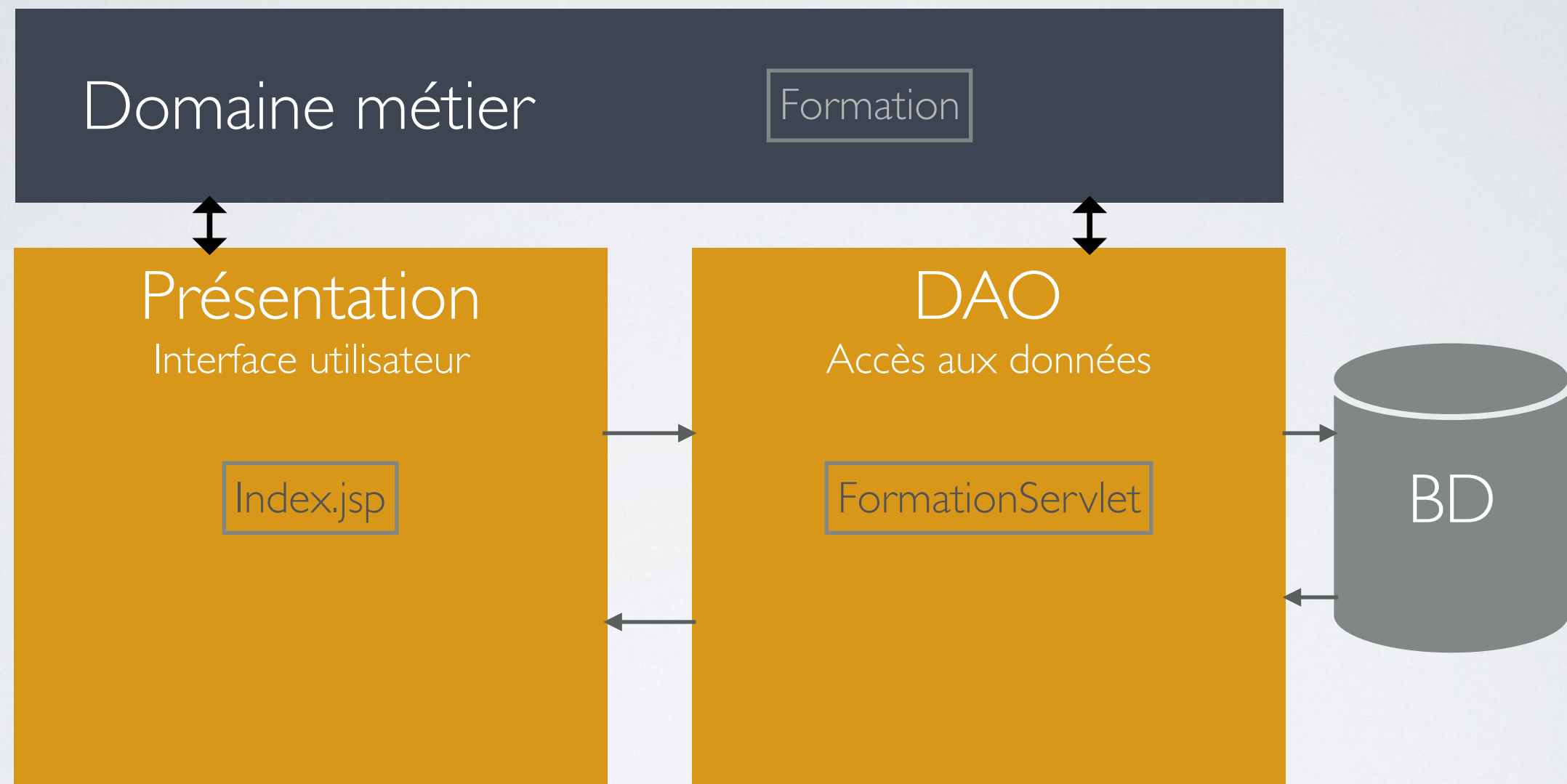
- Représente les objets du domaine, c'est à dire l'ensemble des entités persistantes de l'application

ARCHITECTURE - V1



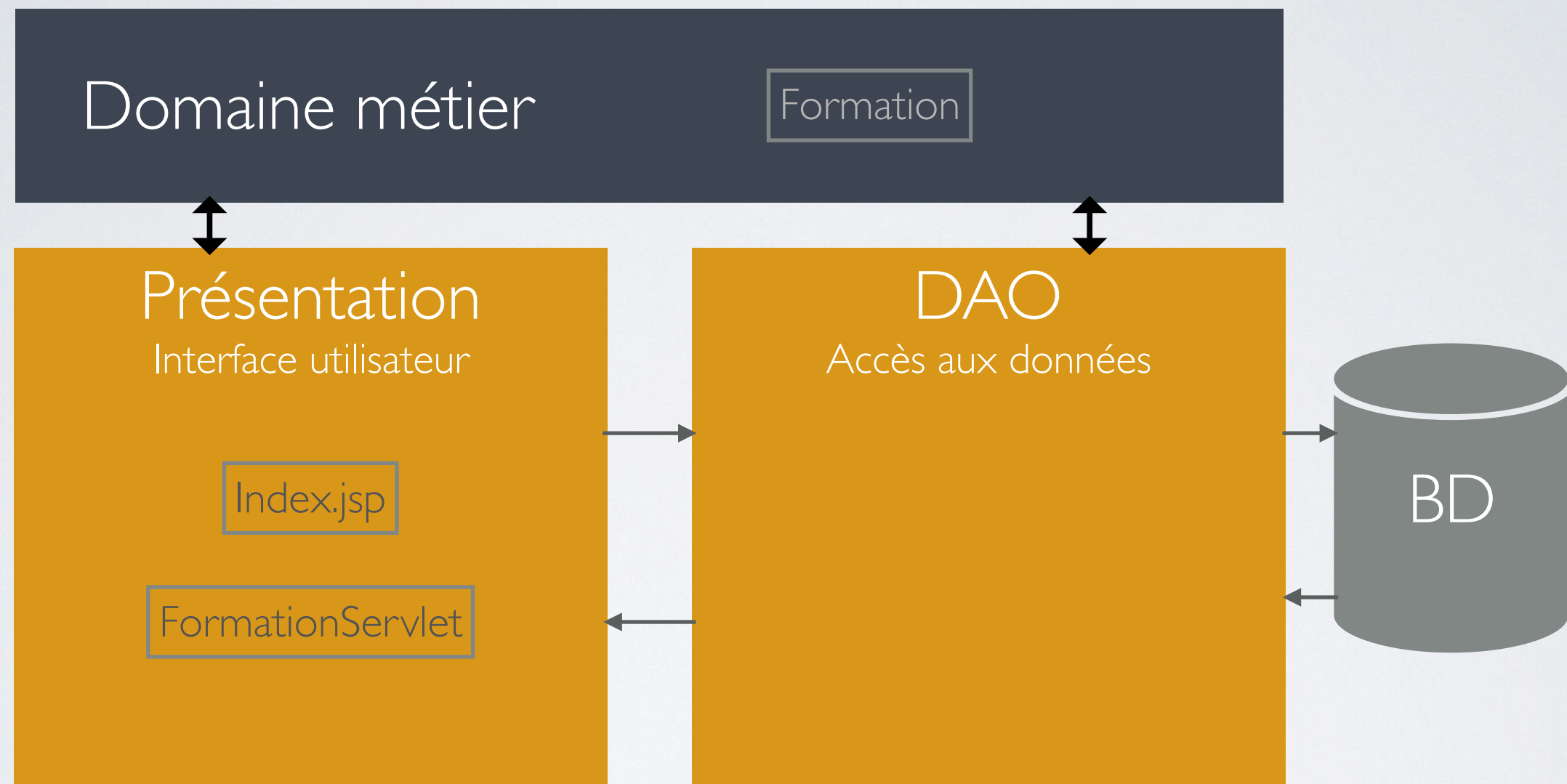
Dans un premier temps il va être nécessaire de supprimer les classes qui ne nous servent plus, puis de commencer à mettre en place une réelle architecture logicielle.

ARCHITECTURE - V2



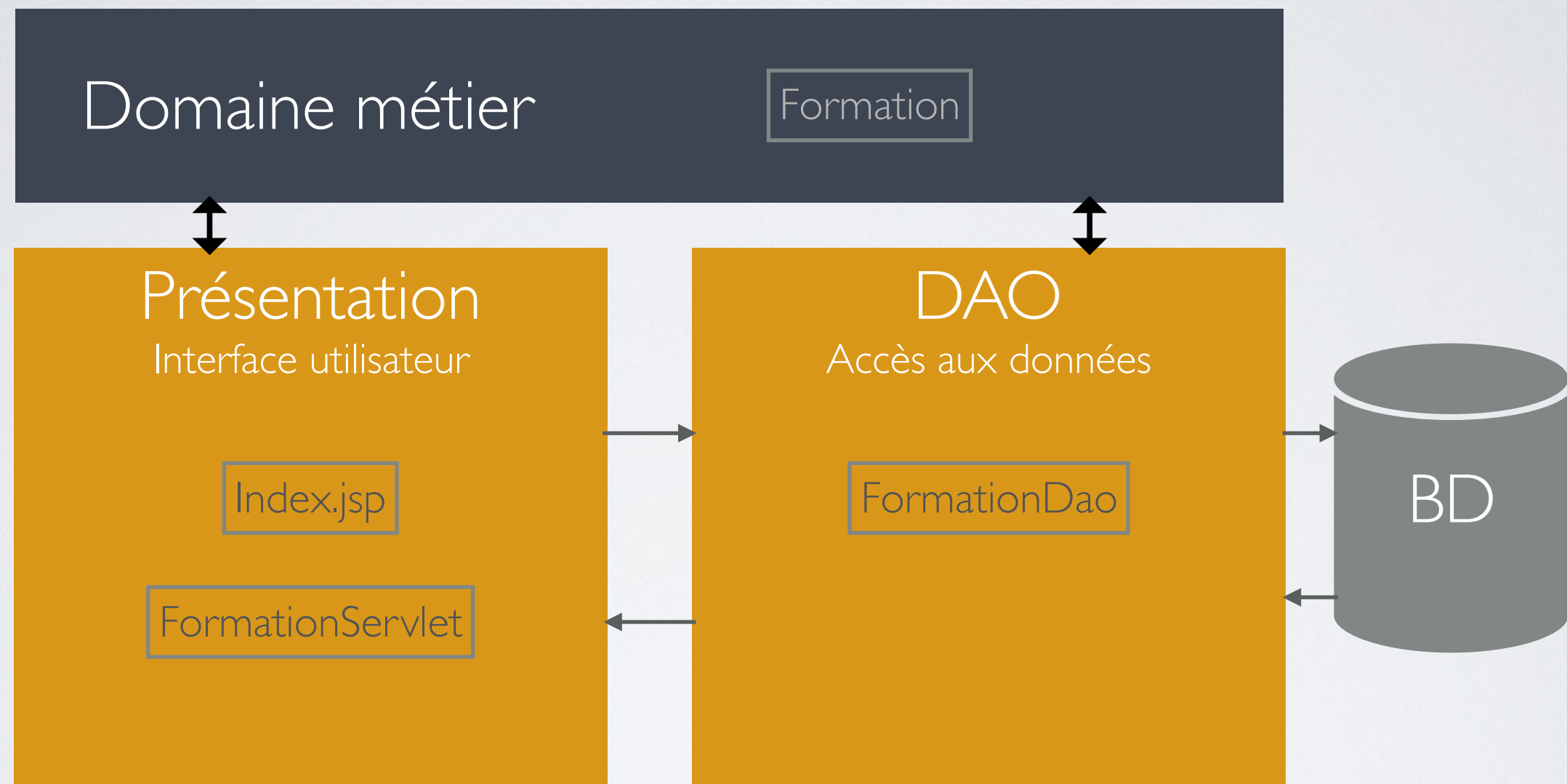
La couche domaine récupère la classe métier Formation. Les classes obsolètes sont supprimées.

ARCHITECTURE - V3



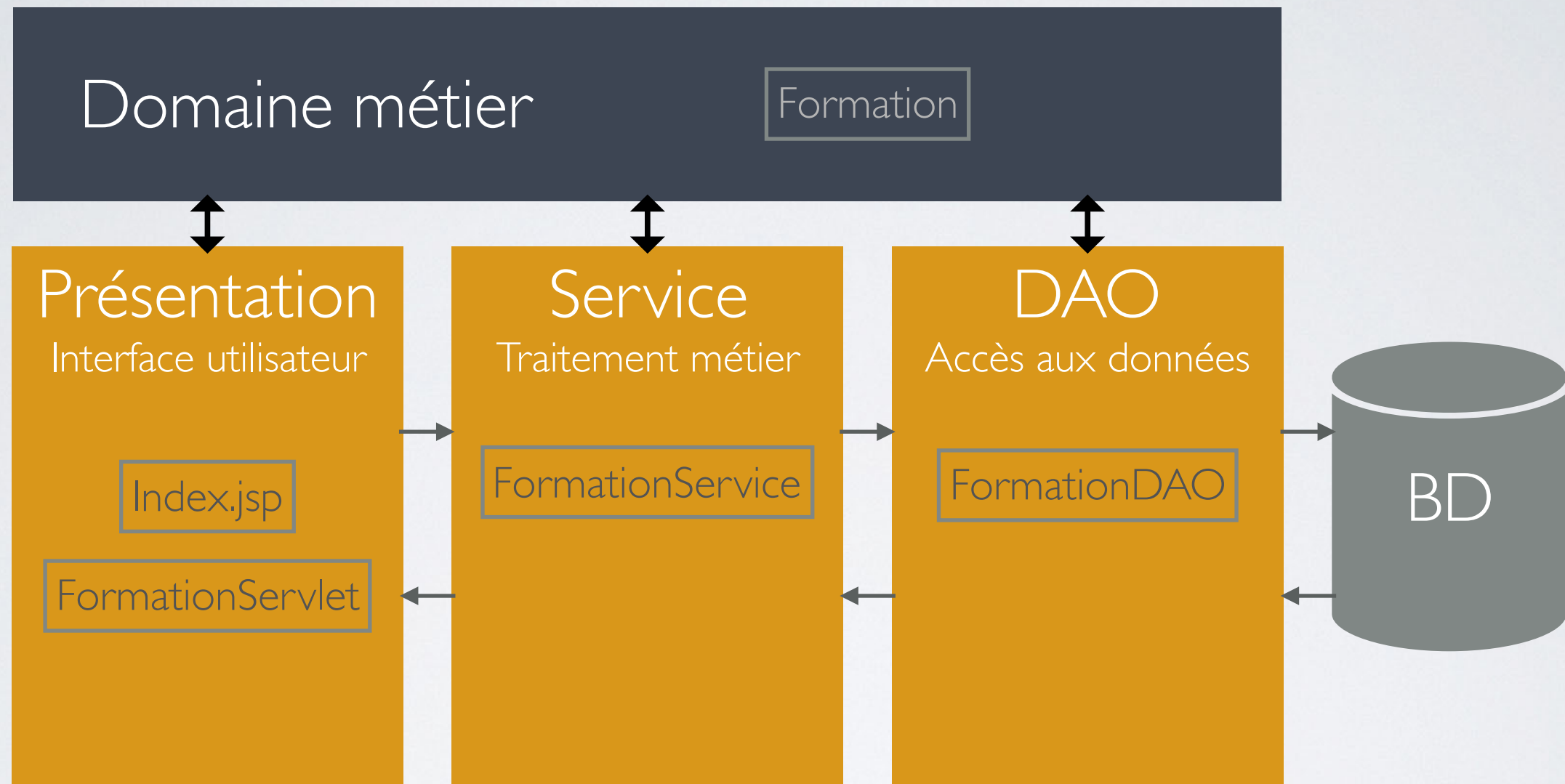
La classe FormationServlet est déplacée dans la couche présentation.

ARCHITECTURE - V4



Une partie du code de la classe `FormationServlet` a été déplacé dans une nouvelle classe `FormationDao` car elle permettait l'accès aux données de la base.

ARCHITECTURE - V5



Une partie du code de la classe `FormationServlet` a été déplacée dans une nouvelle classe `FormationService`. Cette classe interprétera les données envoyées par l'utilisateur depuis la couche présentation et validera les données avant d'atteindre la couche DAO.

EXCEPTIONS - V6

Le fichier web.xml permet de configurer notre application web.

Cela donne la possibilité de rediriger l'utilisateur vers des pages d'erreur personnalisées.

```
1  <!DOCTYPE web-app PUBLIC
2  "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
3  "http://java.sun.com/dtd/web-app_2_3.dtd" >
4
5  <web-app>
6    <display-name>Archetype Created Web Application</display-name>
7    <error-page>
8      <error-code>404</error-code>
9      <location>/WEB-INF/error/errorpage.jsp</location>
10   </error-page>
11
12   <error-page>
13     <exception-type>java.lang.Throwable</exception-type >
14     <location>/WEB-INF/error/errorthrow.jsp</location>
15   </error-page>
16
17   <error-page>
18     <exception-type>javax.servlet.ServletException</exception-type >
19     <location>/WEB-INF/error/errorthrow.jsp</location>
20   </error-page>
21
22   <error-page>
23     <exception-type>java.io.IOException</exception-type >
24     <location>/WEB-INF/error/errorthrow.jsp</location>
25   </error-page>
26 </web-app>
27
```


EXCEPTIONS - V6

Dans notre application nous avons créé un package exception dans chaque « couche ».

Ils contiennent une ou plusieurs classes java représentant chacune une exception.

Cela permet d'avoir un message d'erreur personnalisé en fonction de la couche dans laquelle l'erreur se produit.

