



Java avancé

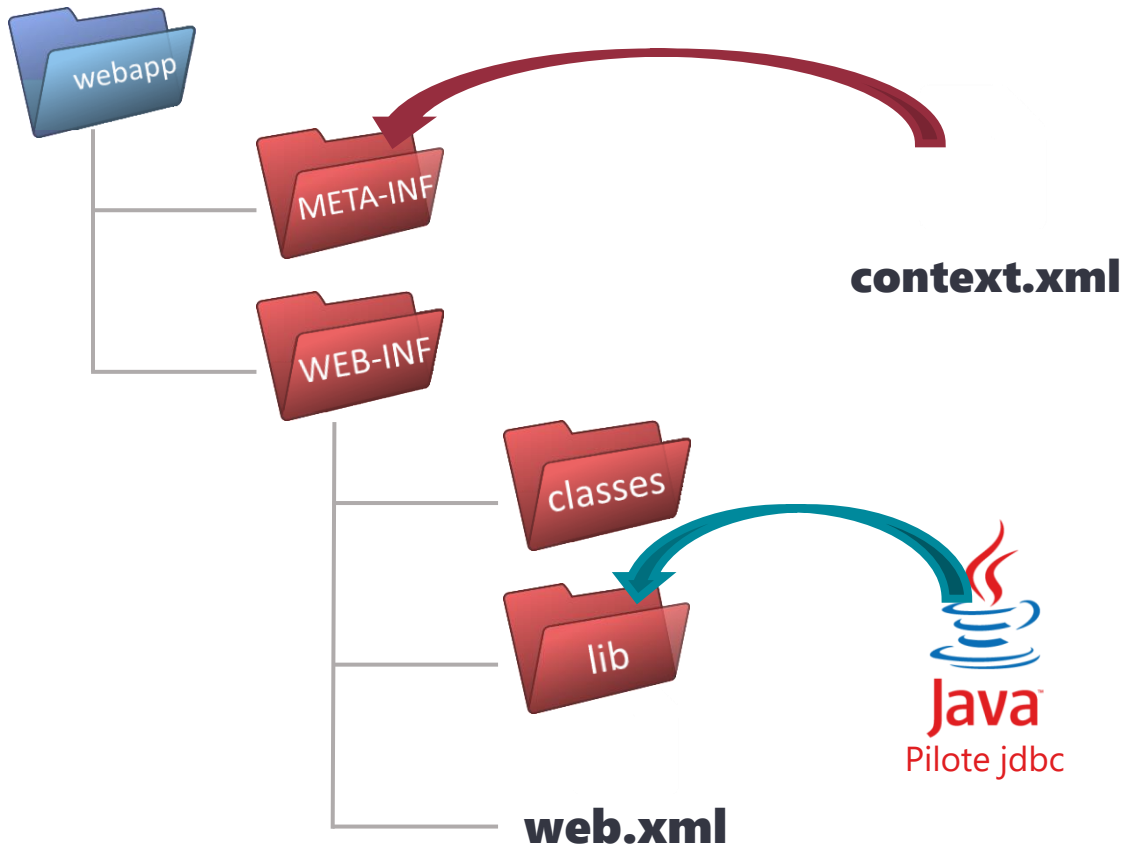
L'accès aux données

Objectifs

- Savoir configurer un pool de connexions
- Savoir intégrer un pool de connexions dans le design pattern DAO



La configuration d'un pool de connexions



Paramètres de configuration

- **name**
- **driverClassName** chargé de faire le lien entre BDD et
- **type**
- **url** localhost/8080/.../phpMyAdmin... (url de la BDD)
- **username** id de l'appli
- **password** pw de l'appli
- **maxTotal** ex: 30: pas plus de 30 connexions en simultanée sur ma BDD (les autres sont en chargement)
- **maxIdle** Temps durant lequel un utilisateur est inactif (on dégage un utilisateur s'il est inactif trop longtemps) => normalement inutile car on doit fermer la connexion après chaque requête
- **maxWaitMillis**

requête qui ne se termine pas: **timeOut** (ex 30sec, l'utilisateur est éjecté au bout de 30 sec)



L'exploitation d'un pool de connexions

- L'obtention d'une connexion

```
Context context = new InitialContext();  
DataSource dataSource = (DataSource)context.lookup("java:comp/env/"+name);  
Connection cnx = dataSource.getConnection();
```

name: nom du context context.XML
fondamental

- La fermeture d'une connexion

```
cnx.close();
```

La connexion n'est pas
fermée mais remise
dans le pool

Défini dans
le fichier
context.xml

permet de redonner la possibilité
aux autres utilisateurs de se connecter



Demo

La configuration et l'exploitation
d'un pool de connexions



Le pattern DAO

Le pattern DAO (Data Access Object) permet de faire le lien entre la couche métier et la couche persistante, ceci afin de centraliser les mécanismes de mapping entre notre système de stockage et nos objets Java. Il permet aussi de prévenir un changement éventuel de système de stockage de données (de PostgreSQL vers Oracle par exemple).

La couche persistante correspond, en fait, à notre système de stockage et la couche métier correspond à nos objets Java, mapper sur notre base. Le pattern DAO consiste à ajouter un ensemble d'objets dont le rôle sera d'aller :

- lire ;
- écrire ;
- modifier ;
- supprimer ;

dans notre système de stockage.
Cet ensemble d'objets s'appelle la couche DAO.

BO

Ces classes représentent les données de la DB, afin de les manipuler en Java

Bin= grain

Java=café

BO: Bin Object



Le pattern DAO

JSP

Spécialisée dans
l'affichage

*Réalise exclusivement
les tâches d'affichage*

Servlet

Spécialisée dans la
récupération et la
transformation des
données

*Fait le lien entre jsp
et bll*

BO

Ces classes représentent les données de la DB, afin de les manipuler en Java



Le pattern DAO

JSP

Spécialisée dans
l'affichage

*Réalise exclusivement
les tâches d'affichage*

Servlet

Spécialisée dans la
récupération et la
transformation des
données

*Fait le lien entre jsp
et bll*

BLL
(Business Logic
Layer)

Spécialisée dans le
contrôle de la
donnée

*Vérifie que les
données fournies
respectent les règles
de l'application*

BO

Ces classes représentent les données de la DB, afin de les manipuler en Java



Le pattern DAO

vue

JSP

Spécialisée dans
l'affichage

*Réalise exclusivement
les tâches d'affichage*

contrôleur

Servlet

Spécialisée dans la
récupération et la
transformation des
données

*Fait le lien entre jsp
et bll*

Model

BLL
(Business Logic
Layer)

Spécialisée dans le
contrôle de la
donnée

*Vérifie que les
données fournies
respectent les règles
de l'application*

Model

DAL
(Data Abstraction
Layer)

Spécialisée dans la
communication avec
la BDD

*Met à disposition les
différentes méthodes
permettant
d'interagir avec la
BDD*

BO

Ces classes représentent les données de la DB, afin de les manipuler en Java



Le pattern DAO

JSP

Spécialisée dans
l'affichage

*Réalise exclusivement
les tâches d'affichage*

Servlet

Spécialisée dans la
récupération et la
transformation des
données

*Fait le lien entre jsp
et bll*

BLL
(Business Logic
Layer)

Spécialisée dans le
contrôle de la
donnée

*Vérifie que les
données fournies
respectent les règles
de l'application*

DAL
(Data Abstraction
Layer)

Spécialisée dans la
communication avec
la BDD

*Met à disposition les
différentes méthodes
permettant
d'interagir avec la
BDD*

DB
(Database)

Spécialisée dans le
stockage de la
donnée

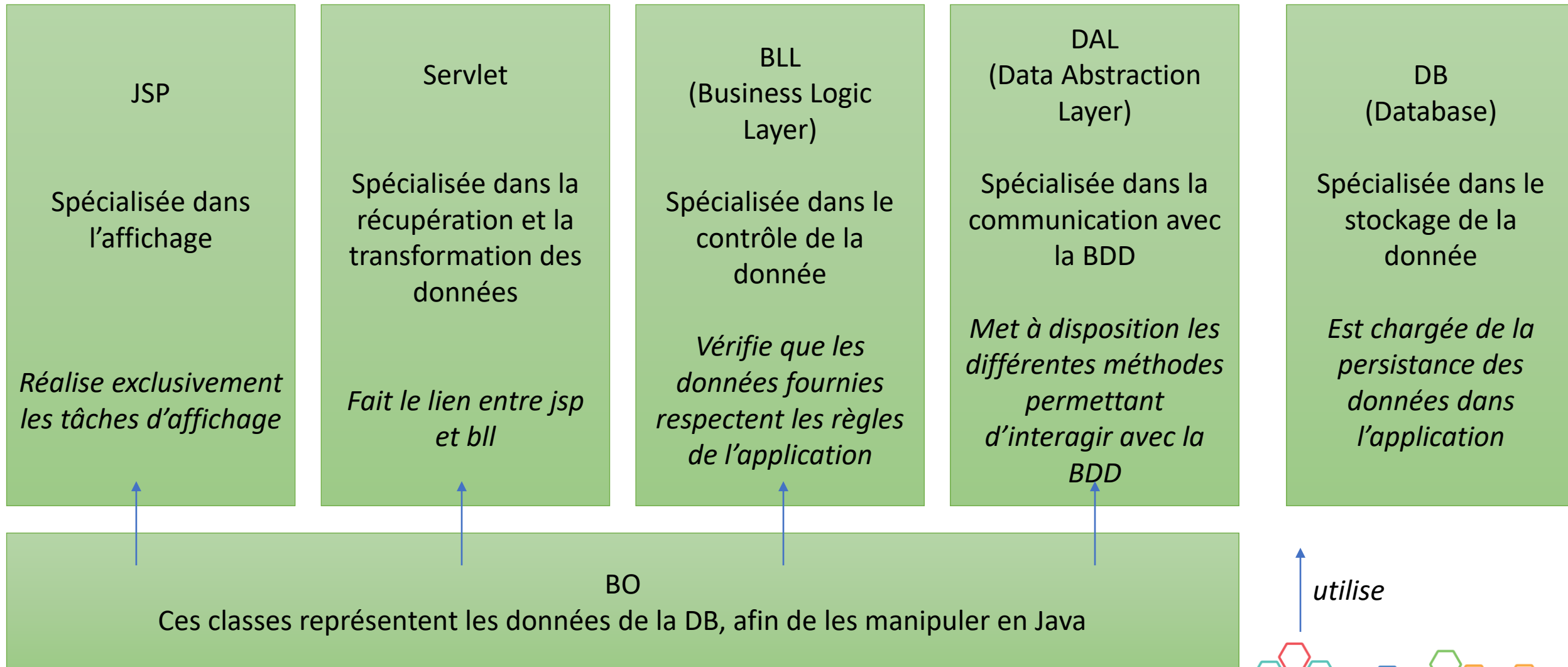
*Est chargée de la
persistance des
données dans
l'application*

BO

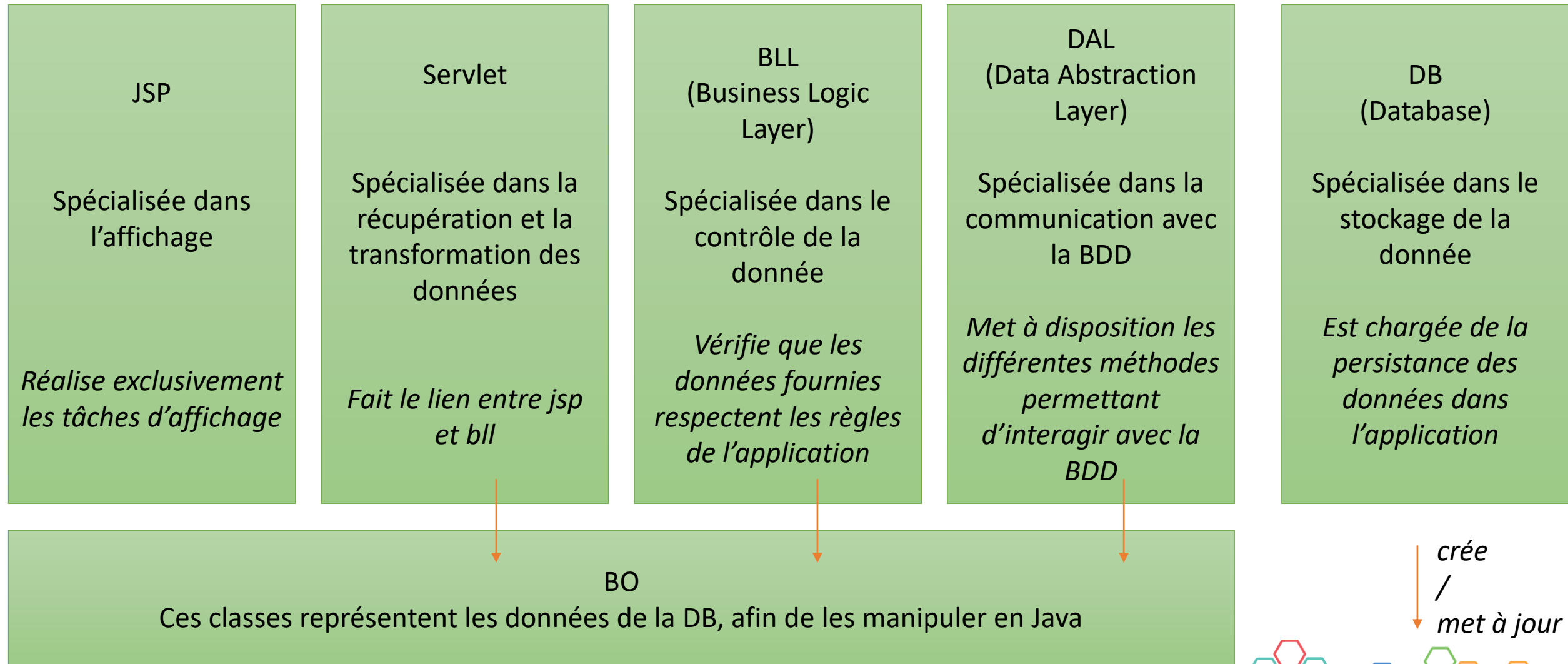
Ces classes représentent les données de la DB, afin de les manipuler en Java



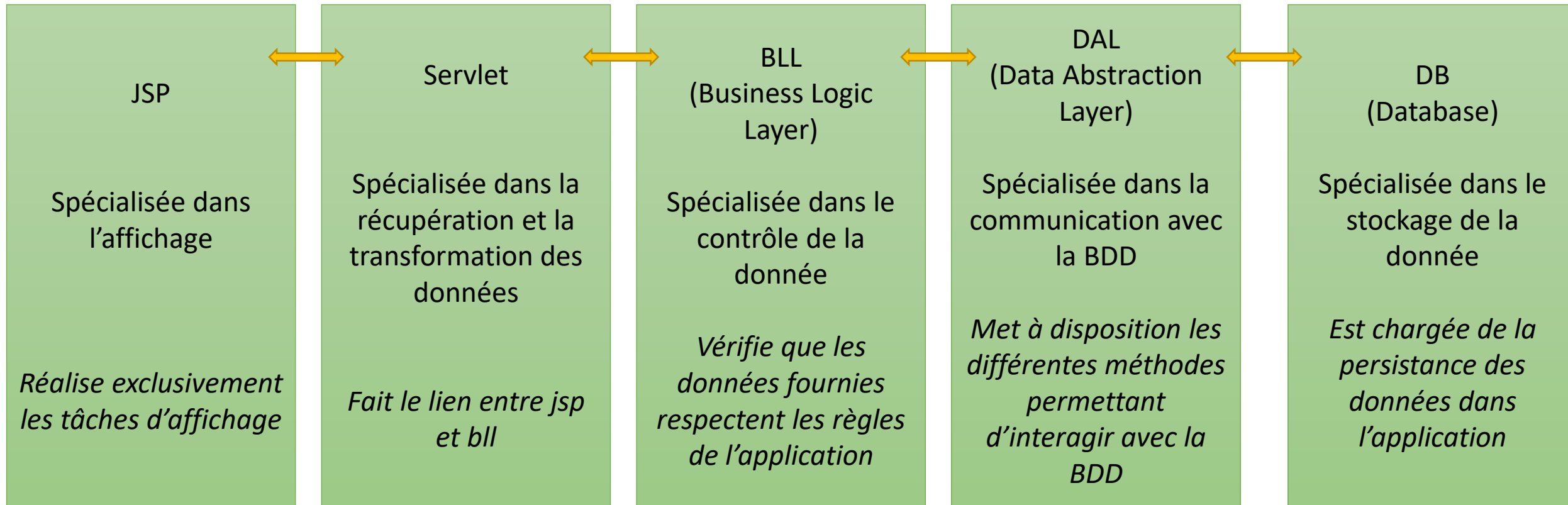
Le pattern DAO



Le pattern DAO



Le pattern DAO

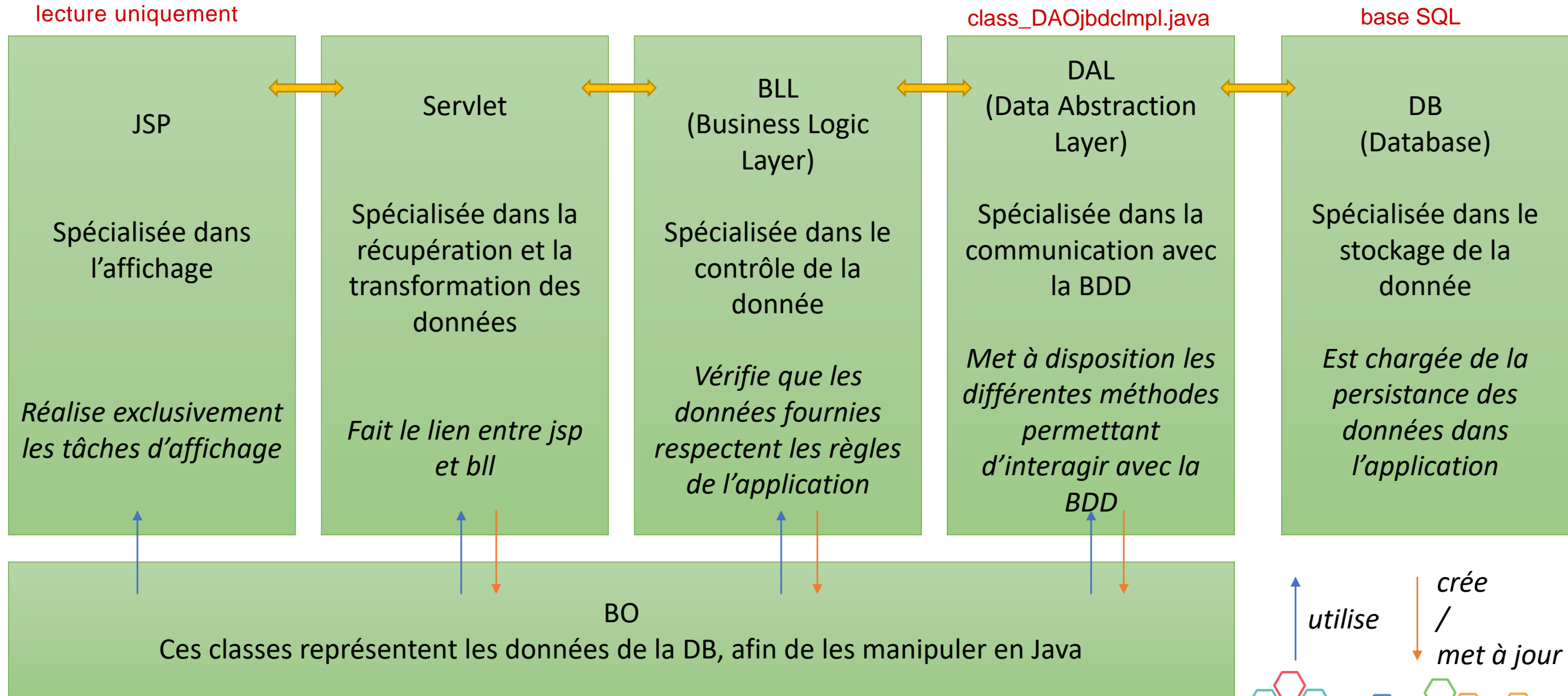


BO
Ces classes représentent les données de la DB, afin de les manipuler en Java

Communication entre les différents packages

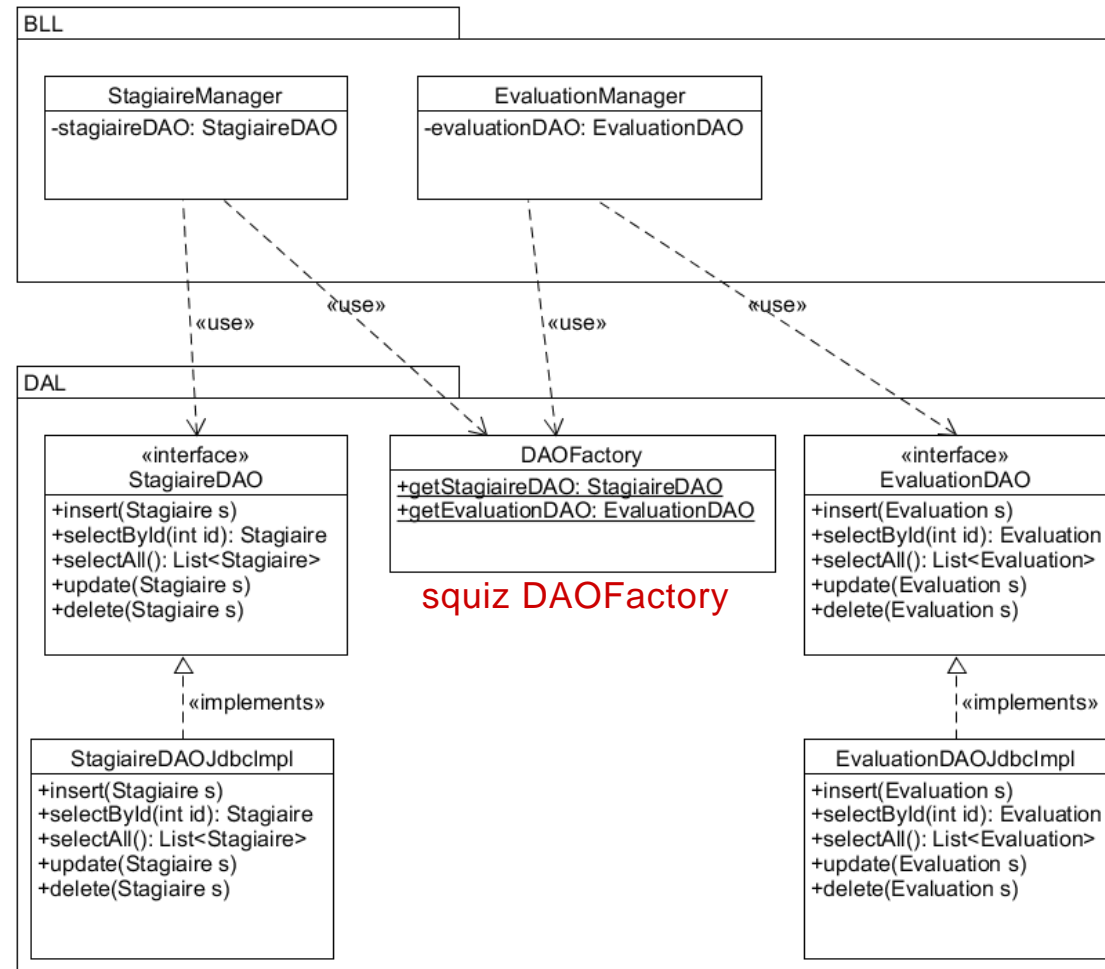


Le pattern DAO

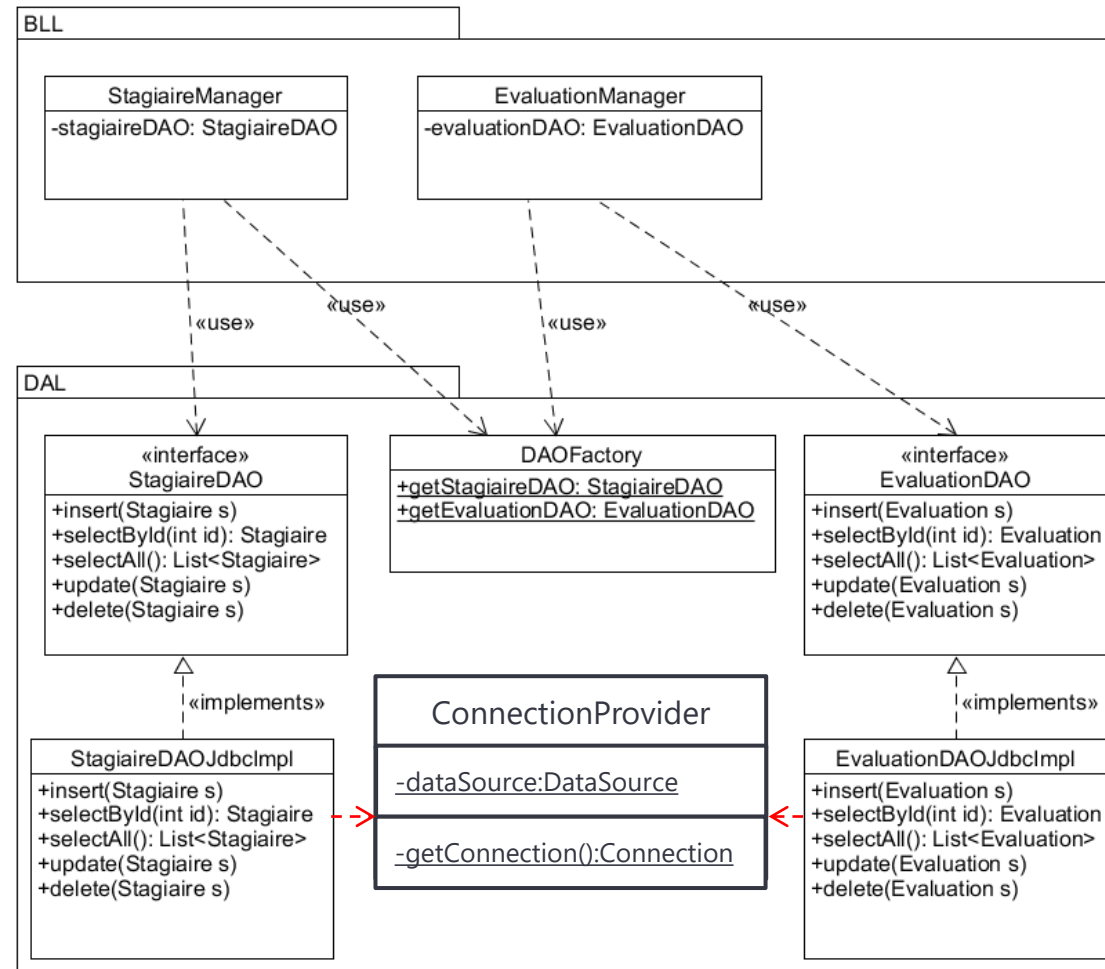


Le pattern DAO

tests unitaires uniquement ici



L'intégration dans le pattern DAO



Demo

Mise en place d'un CRUD avec le pattern DAO



Atelier Annuaire pro

Atelier3-annuaire-bis.pdf





Java avancé

L'accès aux données