

Design Pattern Java EE – L'essentiel

By Yannick



• Sommaire

Tour d'horizon de ce que l'on va aborder...

Intro



Design Pattern dans l'archi JEE

Une vue d'ensemble des design de Sun

DP



Les design pattern

MVC

Business Delegate

Service Locator

Transfert Object

Intercepting Filter

Front Controller

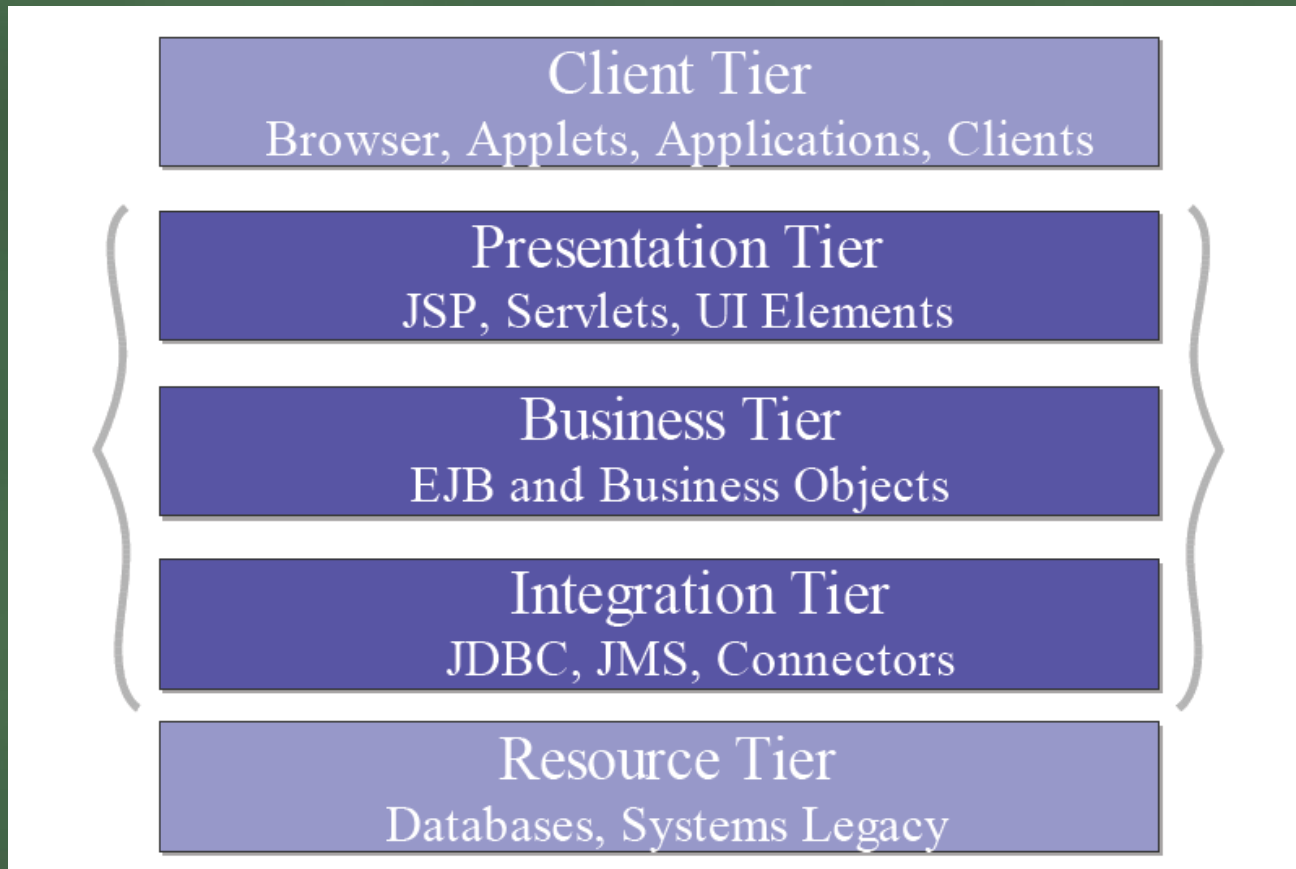
Quizz



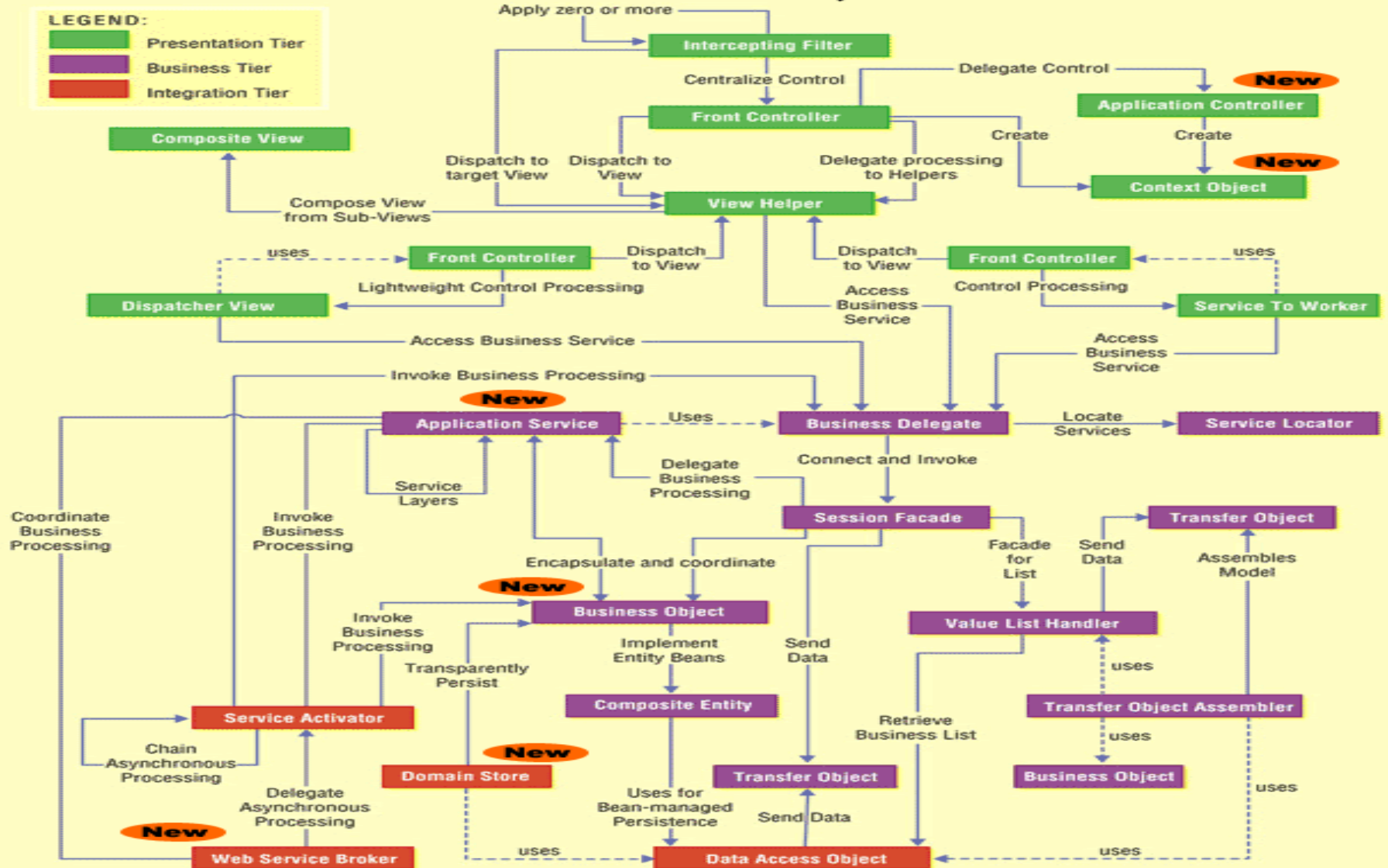
Petit test à chaud !!

Ou comment voir si on a suivi ^^

• Architecture n-tiers



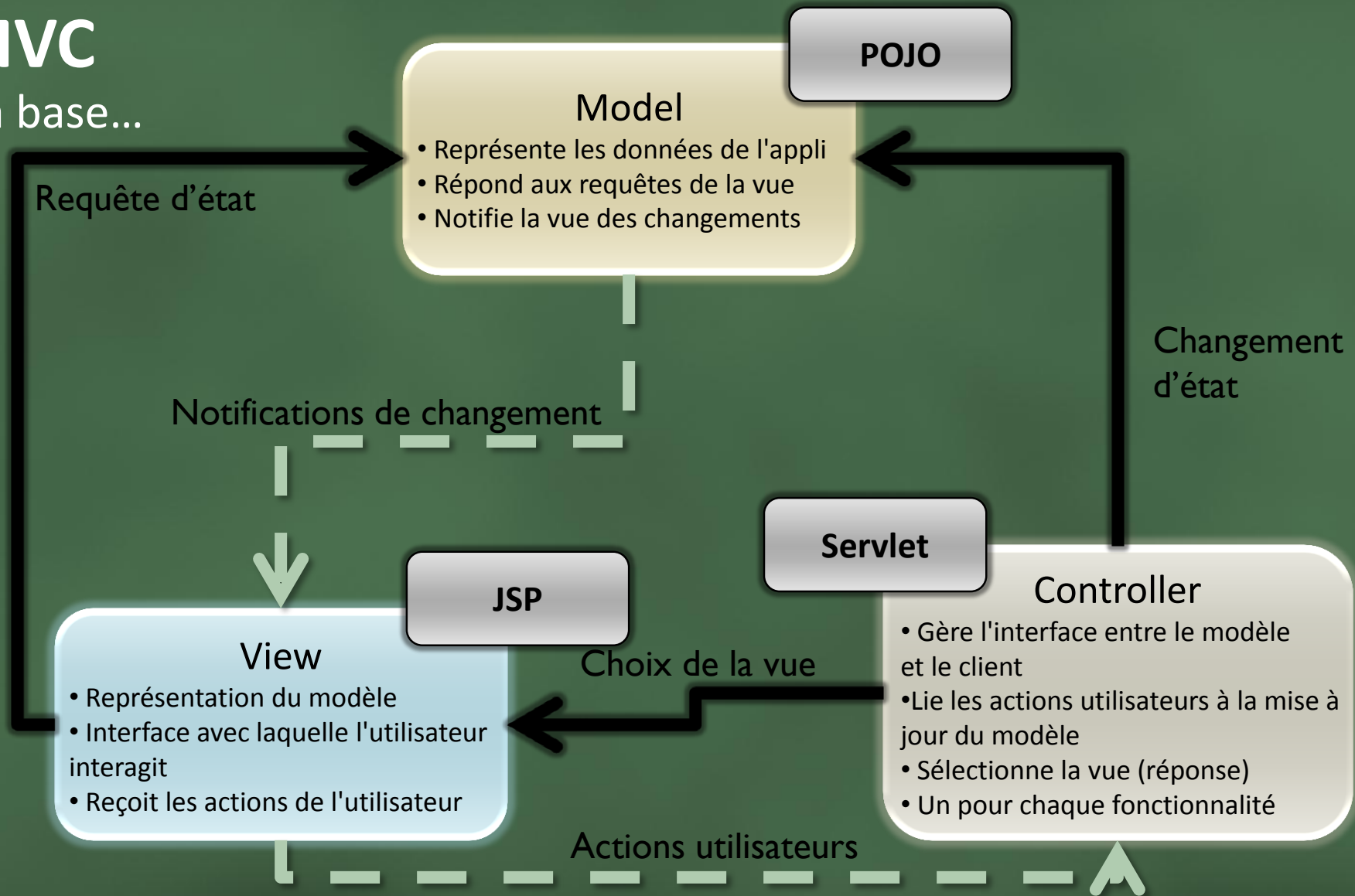
Core J2EE Patterns, 2nd Edition



(c) 2003 corej2eepatterns.com. All Rights Reserved.

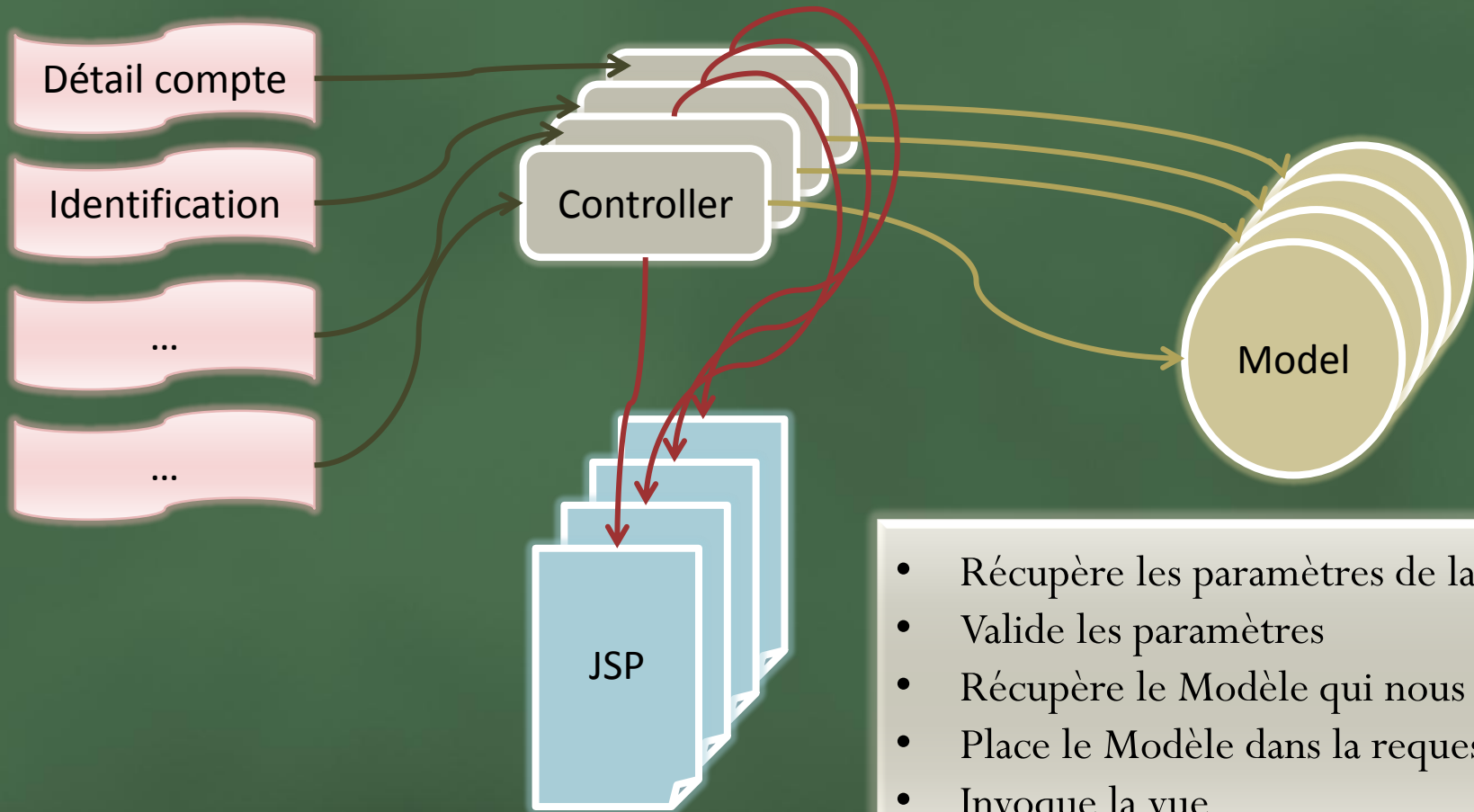
• MVC

La base...



• MVC, dans une appli web « Réelle »

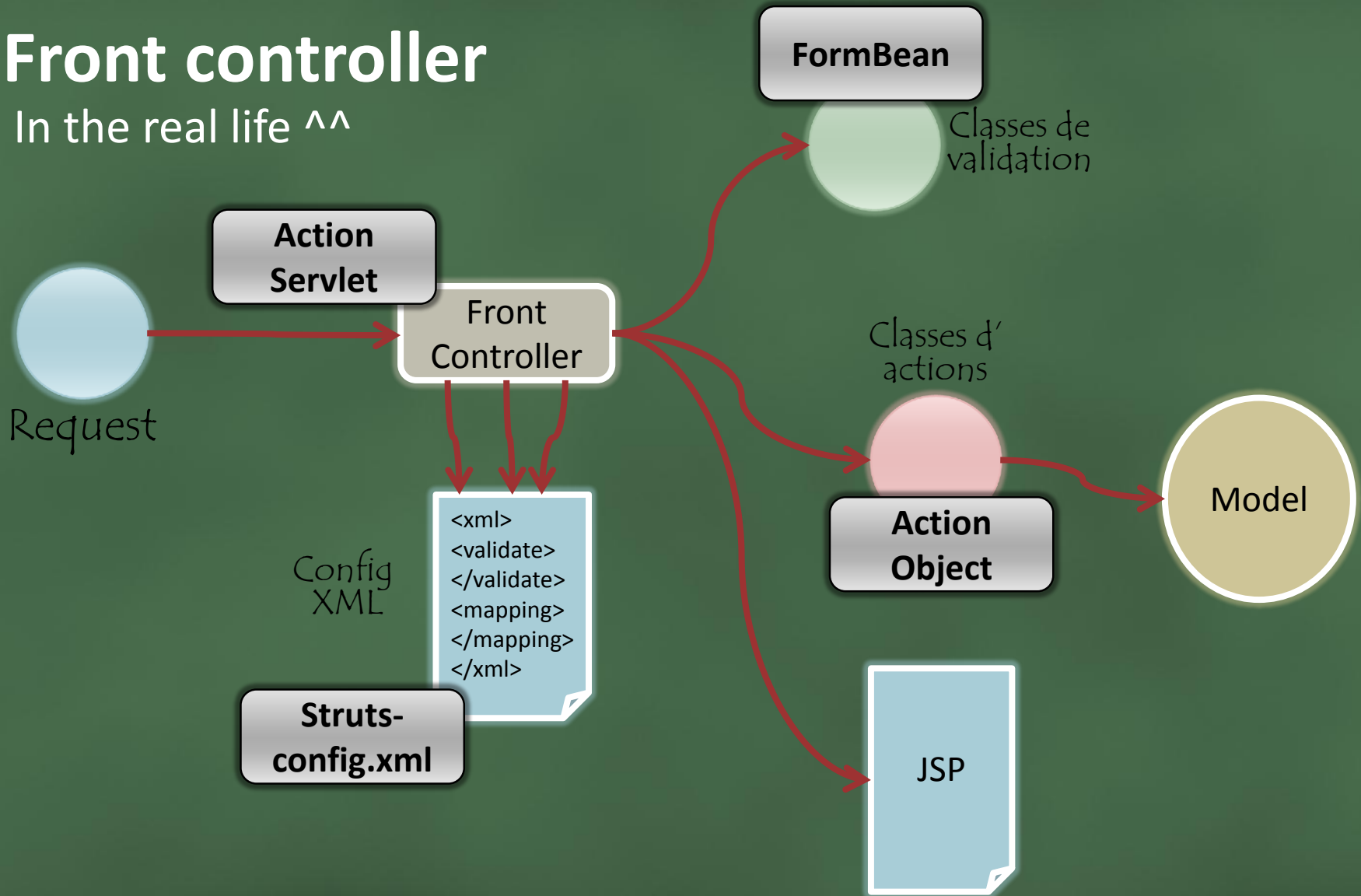
Ou comment gérer le code dupliqué...



- Récupère les paramètres de la request
- Valide les paramètres
- Récupère le Modèle qui nous intéresse
- Place le Modèle dans la request
- Invoque la vue

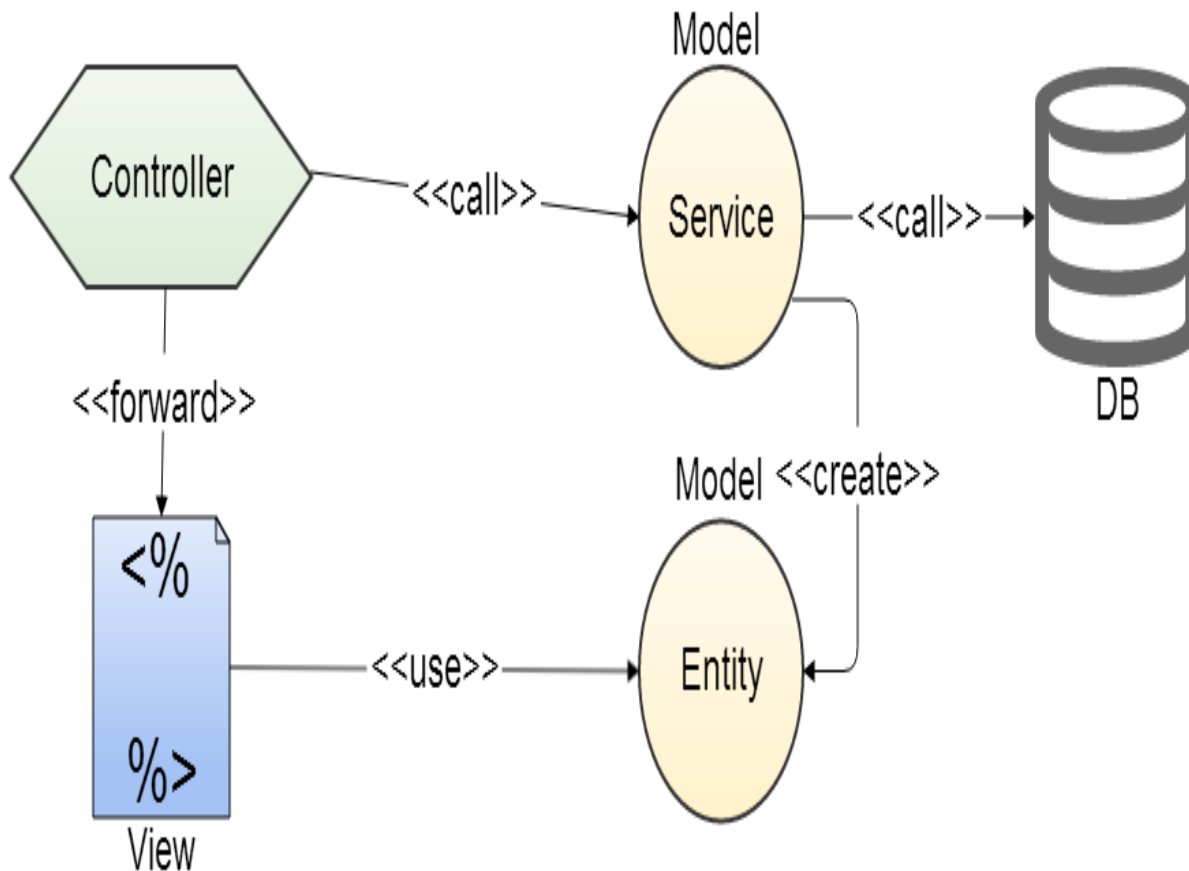
• Front controller

In the real life ^^



•MVC

Quand on est que sur une VM



Oui mais voilà,
et quand le
modèle
n'est pas
sur le
même serveur ???

• Les outils du « remote »

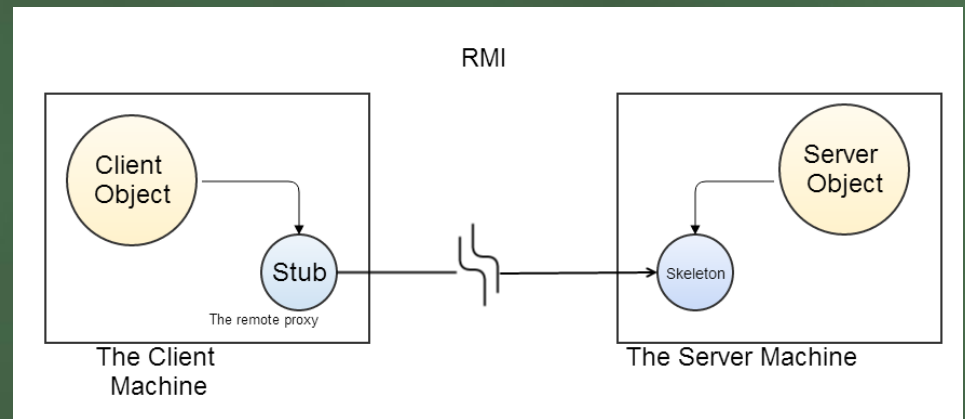
Ou les outils pour comprendre les pattern « remote »

JNDI

- Java Naming and Directory Interface
- Le but c'est de rajouter un élément à l'architecture pour centraliser la location des objets (un annuaire)
- Si on veut partager un objet on l'enregistre sur l'annuaire
- Si on en cherche un, on regarde dans l'annuaire

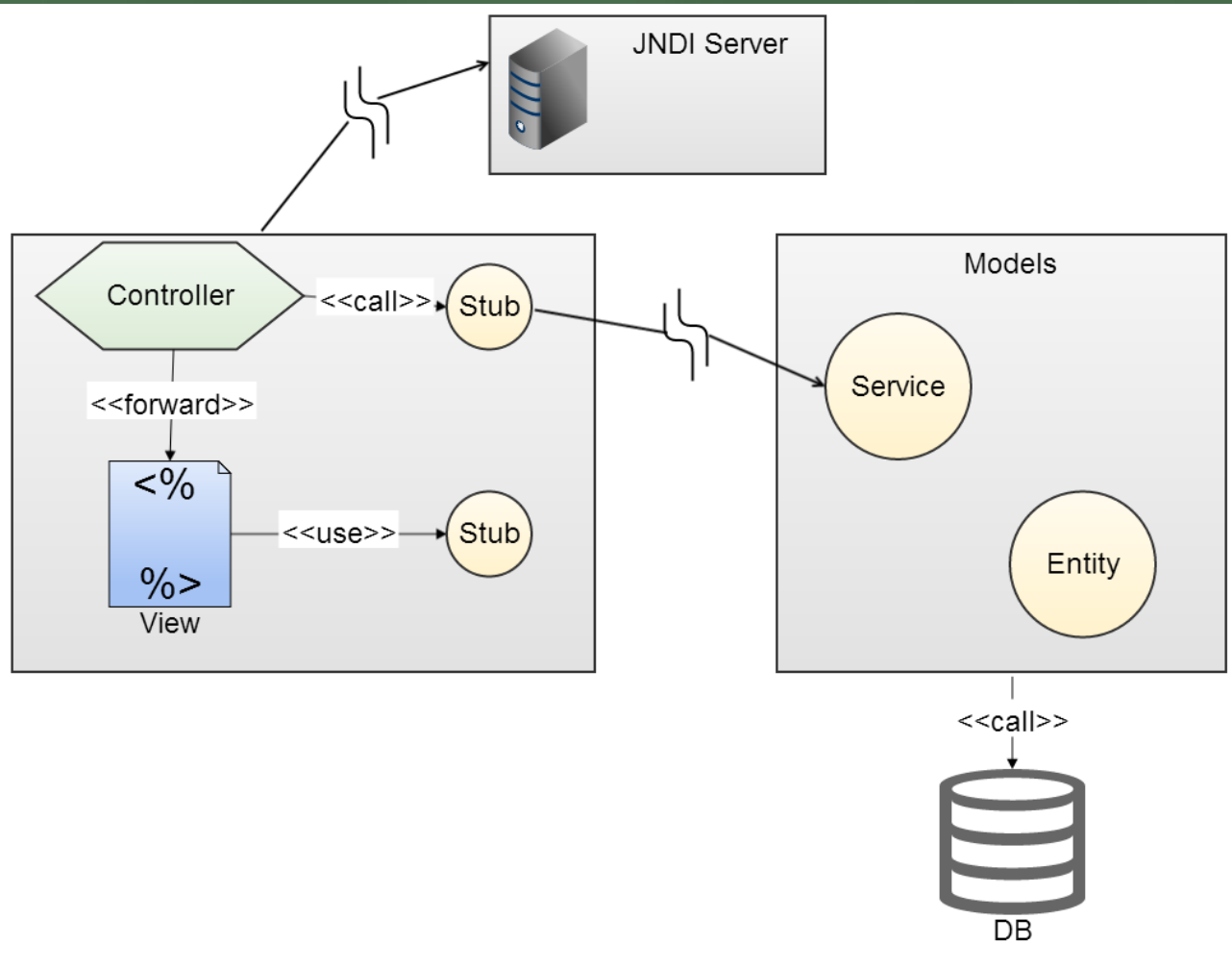
RMI

- Remote Method Invocation
- Le but est de récupérer un objet au travers du réseau



• MVC avec Model « remote »

Un premier jet

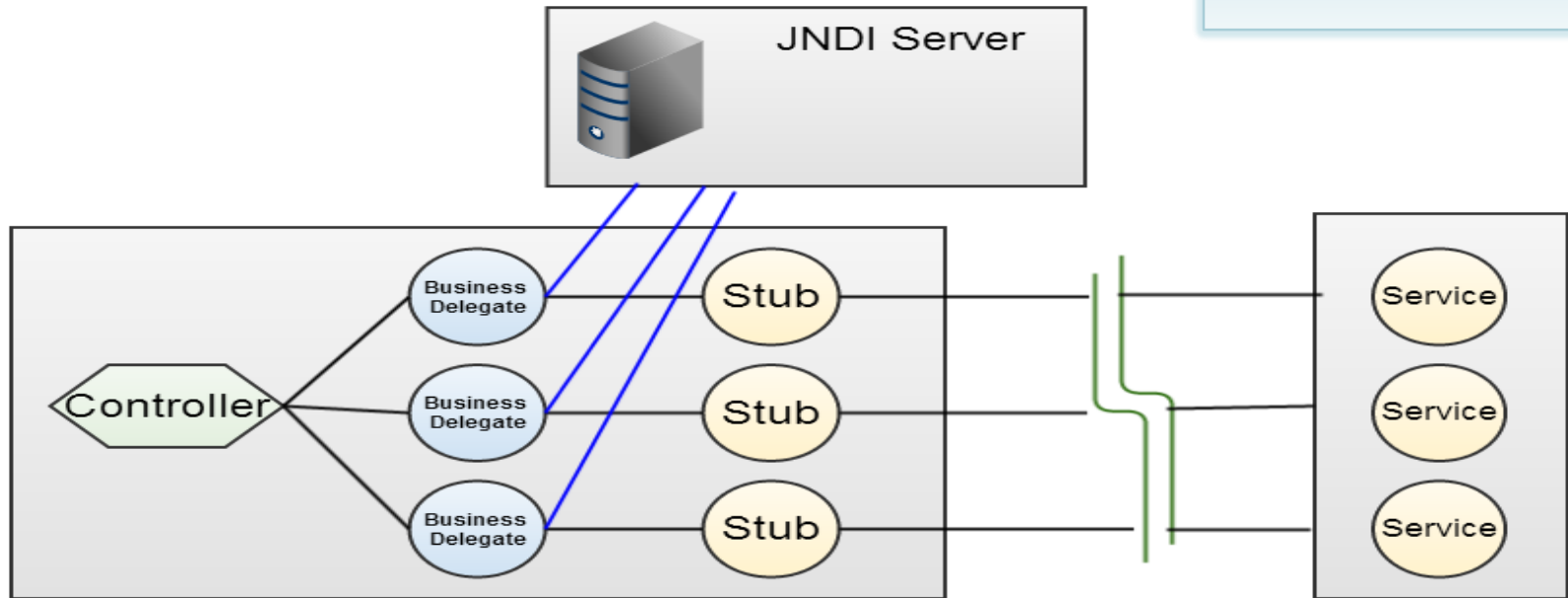


Oui mais voilà,
des fois on est
« remote »
Et des fois pas
😊

• Le pattern Business Delegate

Ou comment cacher la complexité du JNDI et RMI

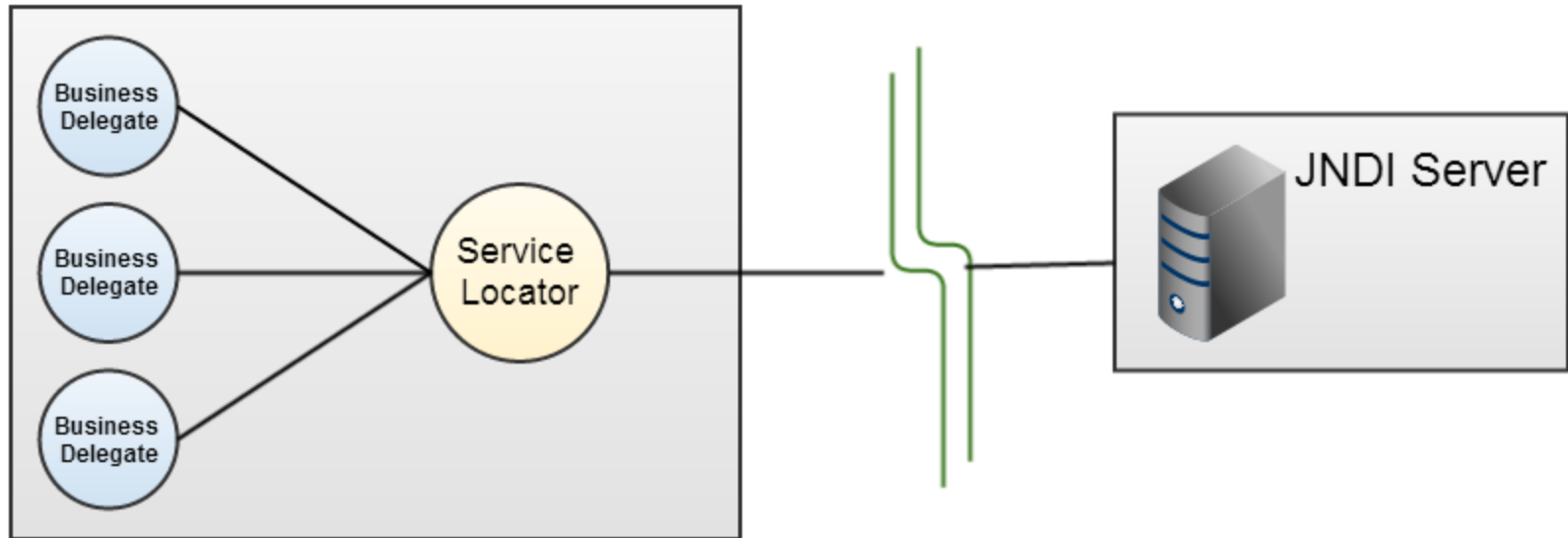
Et cette fois
il est où
le code dupliqué?



- Réduit le couplage entre les couches présentation et Business
- Encapsule la recherche dans l'annuaire
- Proxy pour le client

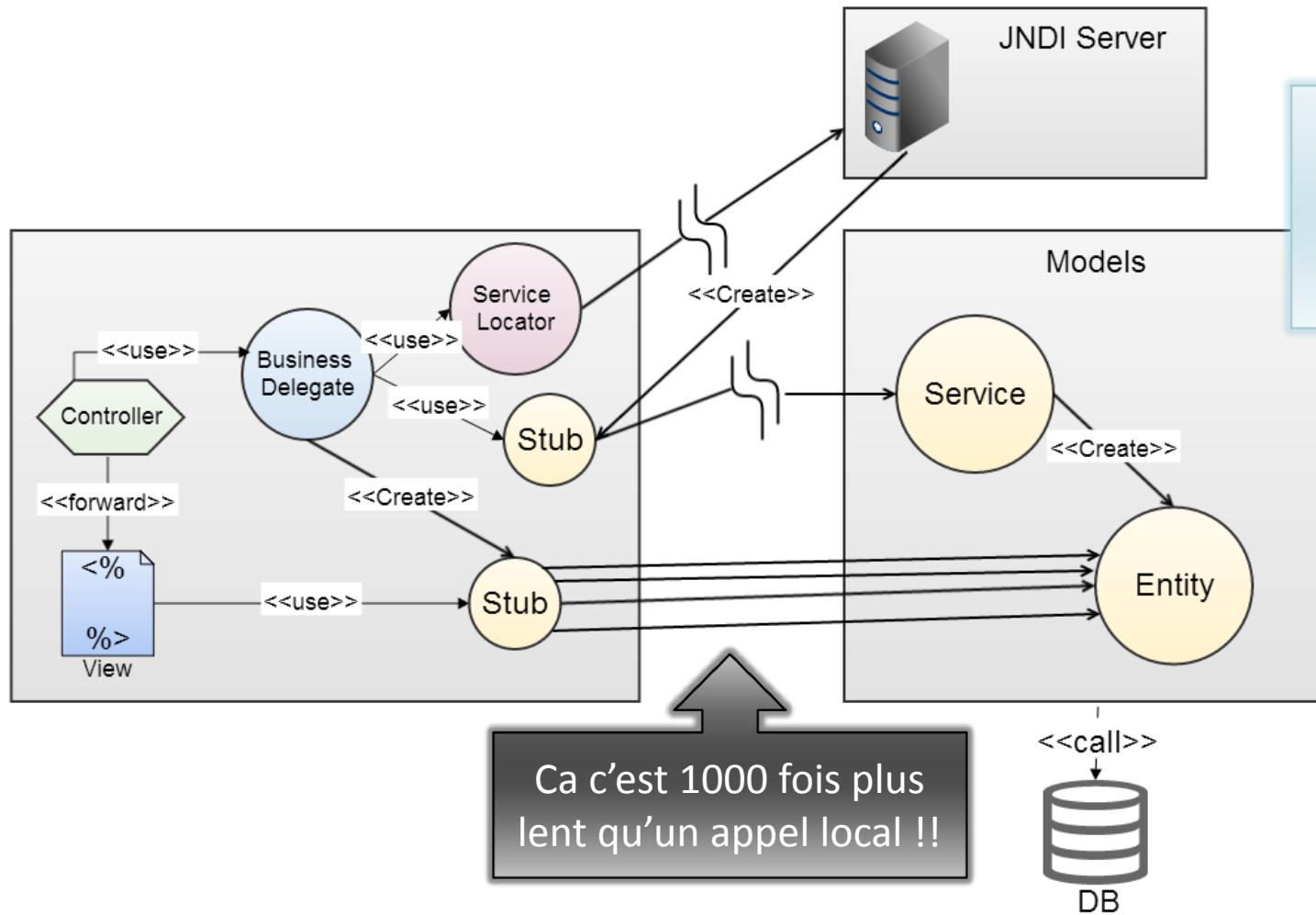
• Le Pattern Service Locator

Ou comment éviter de dupliquer le code du « lookup »



• MVC avec Model « remote »

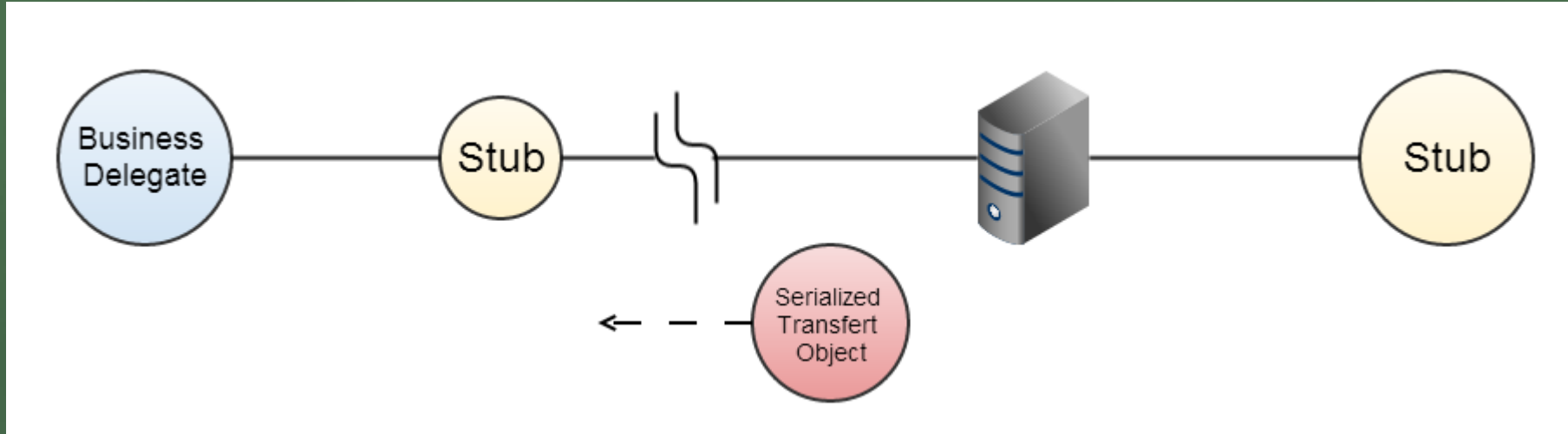
Deuxième jet



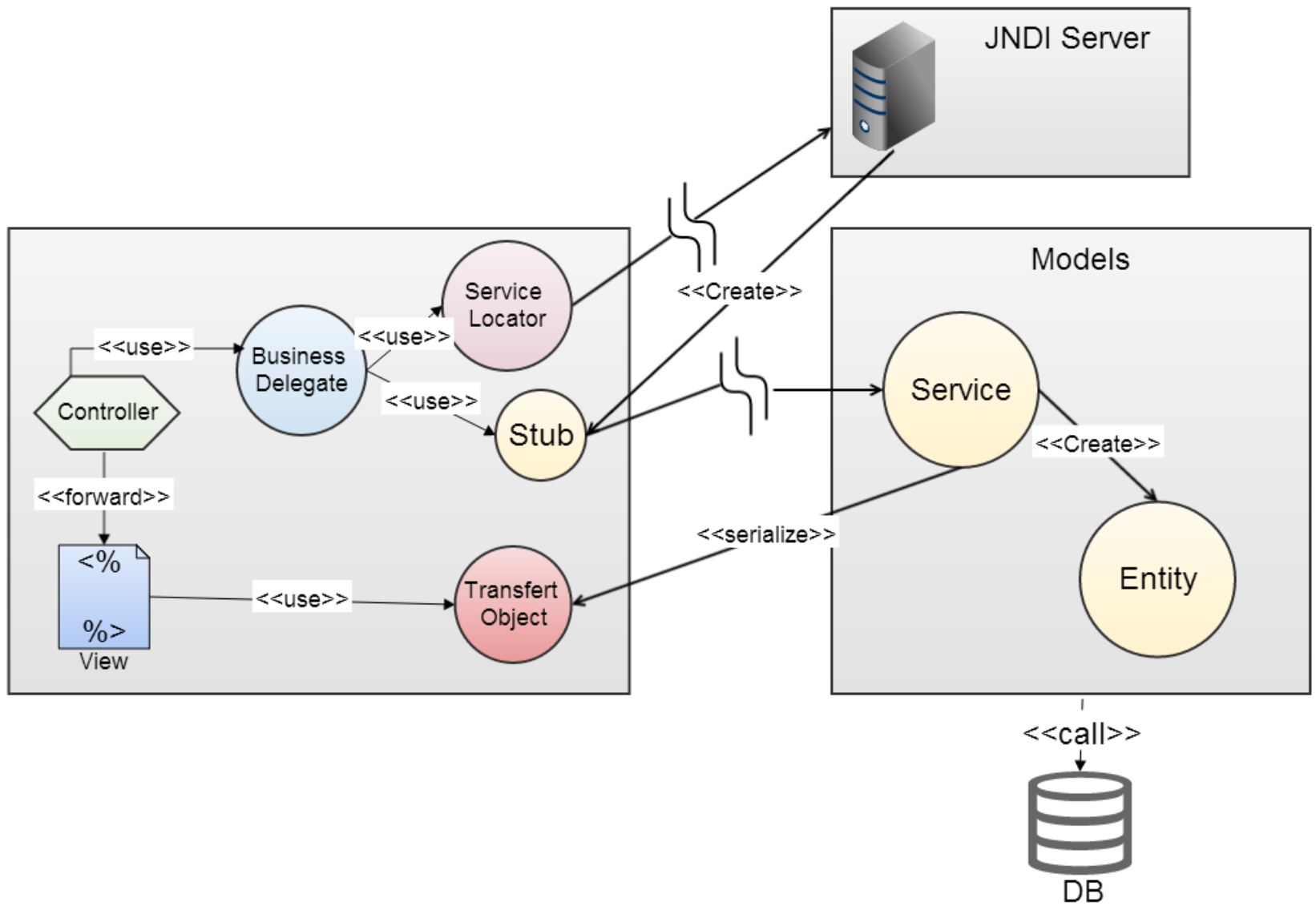
Où est le problème ?

• Le pattern DTO (Data Transfert Object)

Ou à quoi sert la sérialisation...

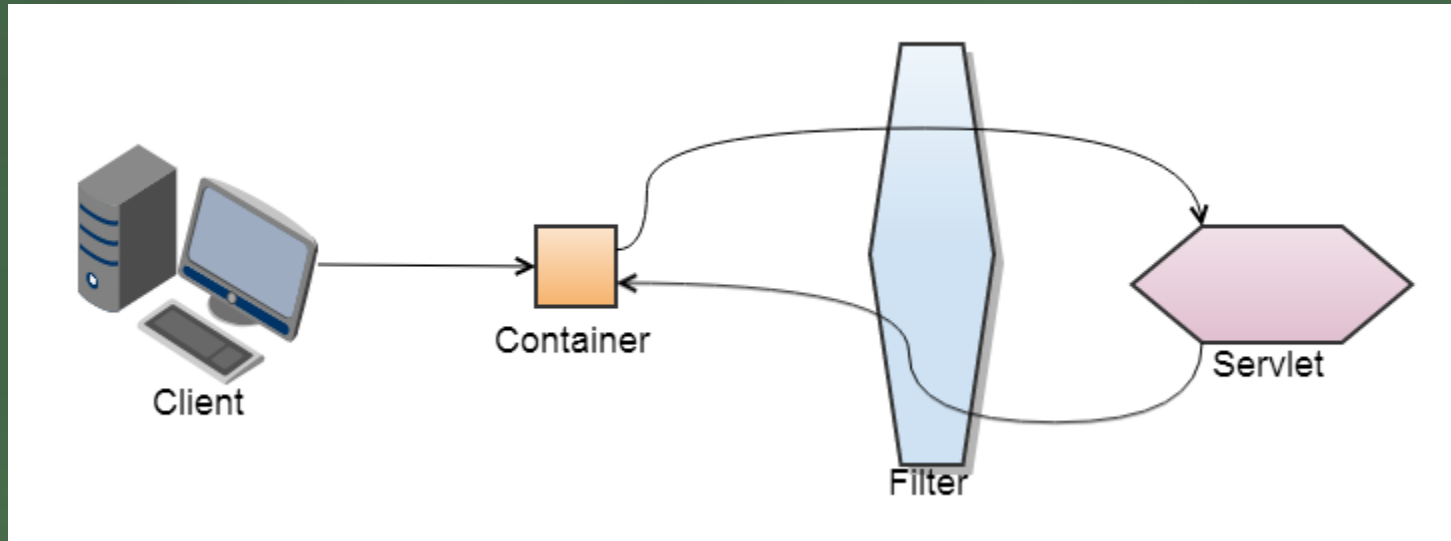


```
try {  
    TransfertObject o = serviceStub.serviceMethod(param);  
} catch (RemoteException e) {  
    throw new ApplicativeException();  
}
```

• Le Pattern Intercepting Filter

Ou comment rajouter de l'information partout sans tout changer...



- Peut intercepter et/ou modifier la requête avant qu'elle arrive à la servlet
- Peut intercepter et/ou modifier la réponse avant qu'elle soit retournée au client
- Déclarée dans le web.xml
- Les filtres peuvent être chaîné