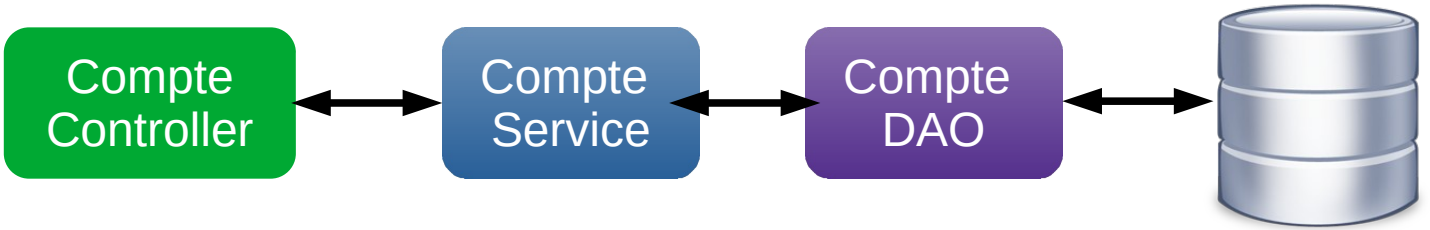
# Spring Aspect-Oriented Programming (AOP)

# AOP – Problème à résoudre



# Dans la Dao on va trouver cette méthode :

```
public void addCompte (Compte compte, String userId){
    Session session = sessionFactory.getCurrentSession() ;
    session.save(compte) ;
    }
```

# New Requirement - Logging

Votre Patron vous interpelle et vous explique

- Nous avons besoin d'ajouter la fonctionnalité logging à nos méthodes présentes dans la DAO

  ajouter des instructions de logging avant de démarrer, au déclenchement de chaque méthode

- On va en ajouter (du logging) probablement à d'autres endroits, masi commence déjà par là

# Dao code

```
public void addCompte (Compte compte, String userId){
    // ici je vais ajouter mon logging
    Session session = sessionFactory.getCurrentSession() ;
    session.save(compte) ;
    }
```

# Nouveau requirement - Sécurité


- Nous devons ajouter du code pour sécuriser la DAO

  On doit s'assurer que l'utilisateur est autorisé à utiliser les méthodes dans la dao

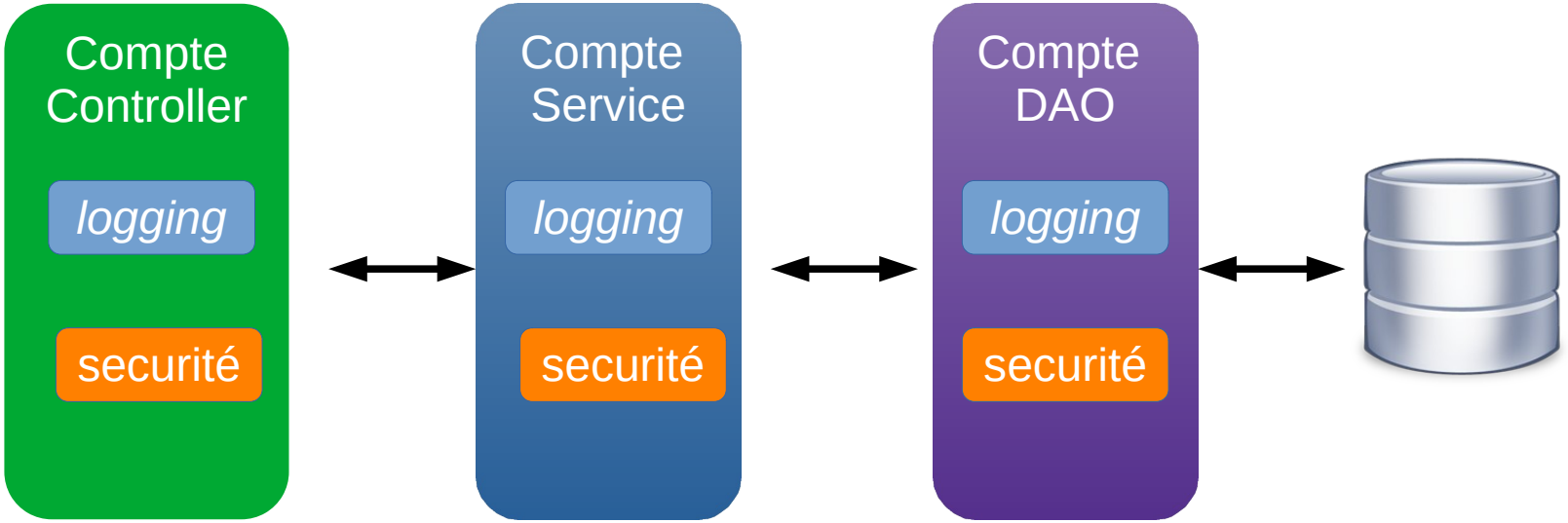
# Dao code

```
public void addCompte (Compte compte, String userId){
    // ici je vais ajouter mon code de logging
    // ici je vais ajouter mon code de sécurité
    Session session = sessionFactory.getCurrentSession() ;
    session.save(compte) ;
    }
```

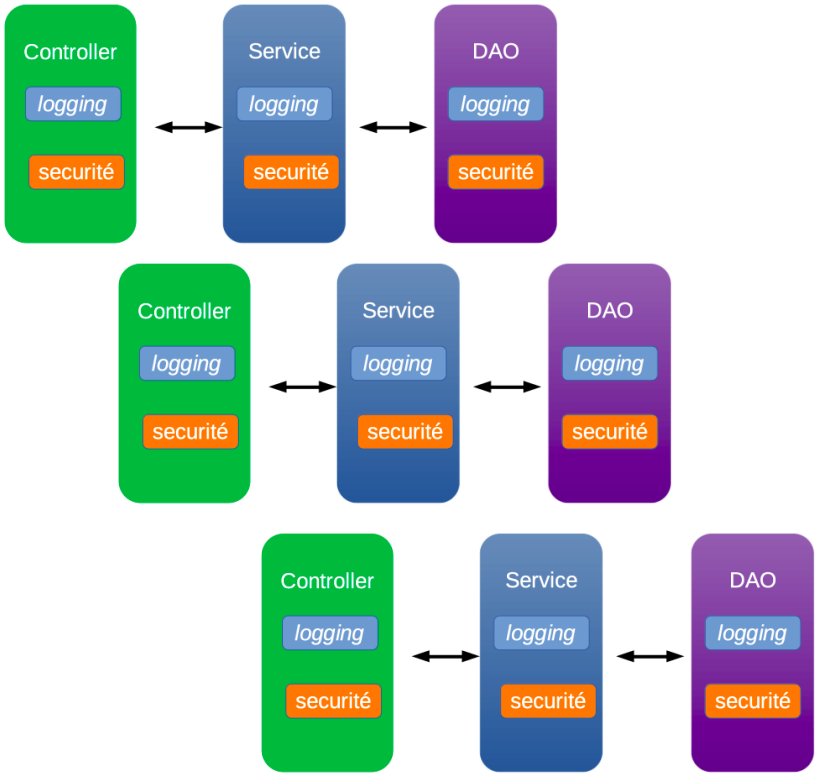
# AOP – Problème à résoudre

Votre Patron vous interpelle et vous explique

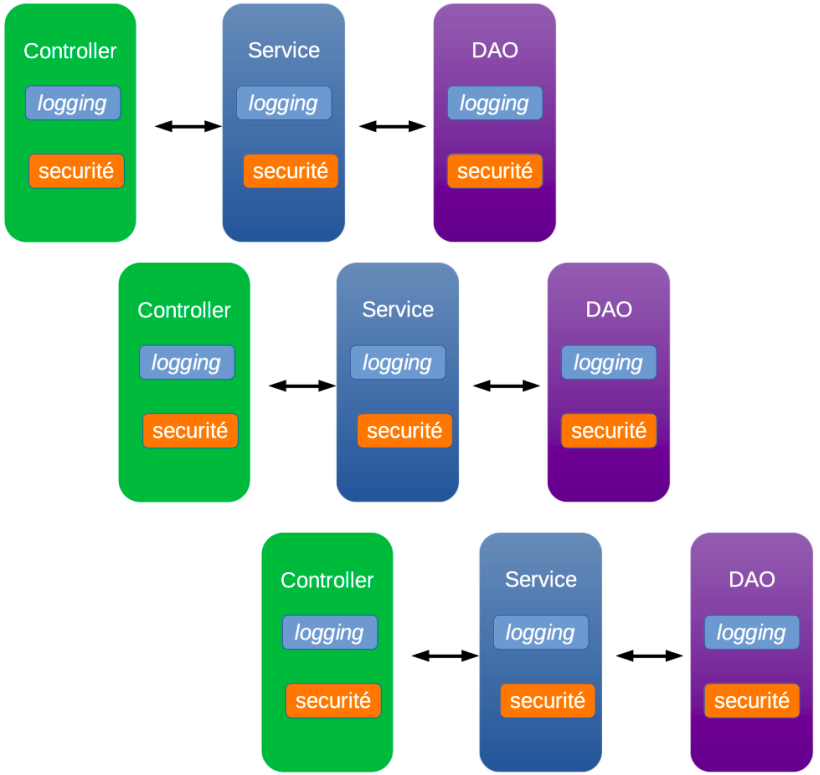
- On va  ajouter du logging et du code de sécurité à chaque couche logicielle

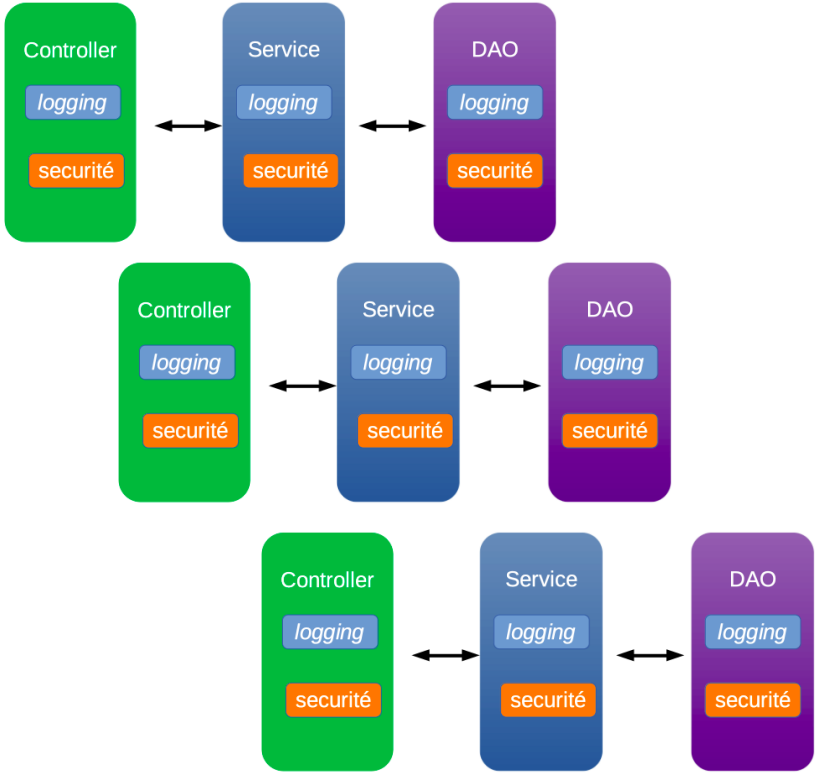


# On va avoir du boulot ...

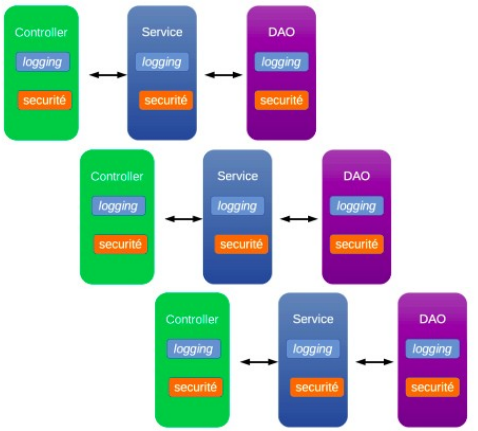


# On va avoir du boulot ...



# Deux principaux problèmes




- On va perdre en lisibilité : pour chaque méthode on va avoir une tâche de logging et une tache de sécurité à définir. On va mélanger code technique et code métier, ce qui n'est pas souhaitable.
- Ensuite on va avoir un code éparpillé , si on doit modifier le code de logging ou celui lié à la sécurité , on va devoir intervenir sur plusieurs composants logiciels. On va devoir mettre à jour toutes les classes.

# Quelles autres solutions

- ?

# Les options qui s'offrent ...

- L'héritage
- Déléguation

# Les options qui s'offrent ...

- L'héritage
- Déléguation
- Programmation Orientée Aspect

# Définition : AOP

- Il s'agit d'une technique de programmation basée sur le concept d'Aspect.
- Un aspect va encapsuler la logique transverse.

# Définition : AOP

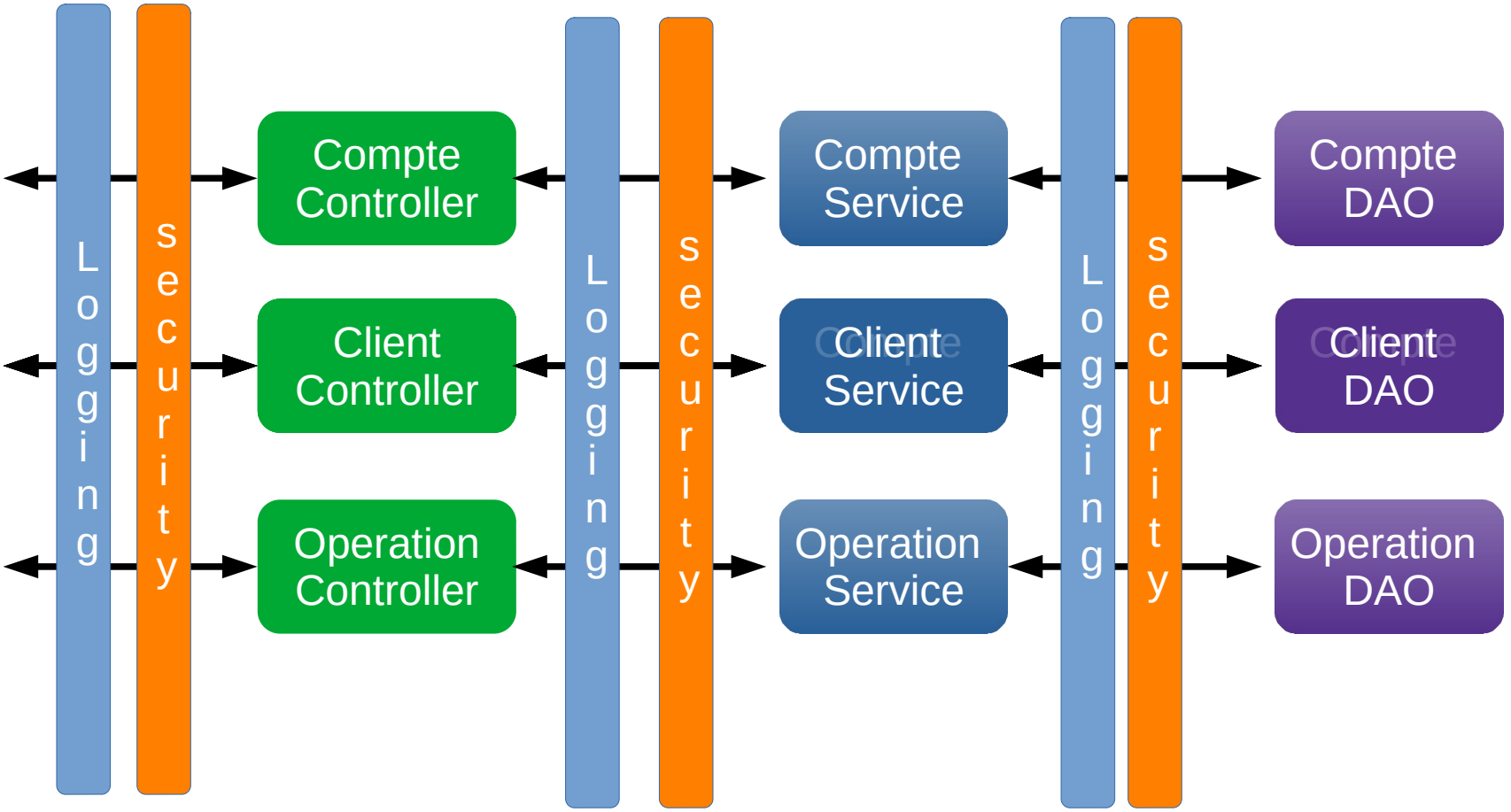
- Il s'agit d'une technique de programmation basée sur le concept d'Aspect.
- Un aspect va encapsuler la logique transverse.

**Alerte buzz-word : Cross-cutting Concerns**

"concerns" signifie logique/ fonctionnalités.

"Cross-cutting" signifie transversal

# Cross-cutting concerns



# Aspects

Aspect peut être réutilisé à plusieurs endroits
Le même aspect / classe .. applicable selon une configuration