Structure d'une application

Architecture n-tiers

REST

Diviser pour mieux régner

Le sens apporté à une classe permet de lui associer une fonction spécifique plutôt que de créer des Objets “fourre tout” responsables de toute l’application.

C’est une extension du **“Principe de responsabilité unique”** introduit par la POO. “Une classe ne doit changer que pour une seule raison”.

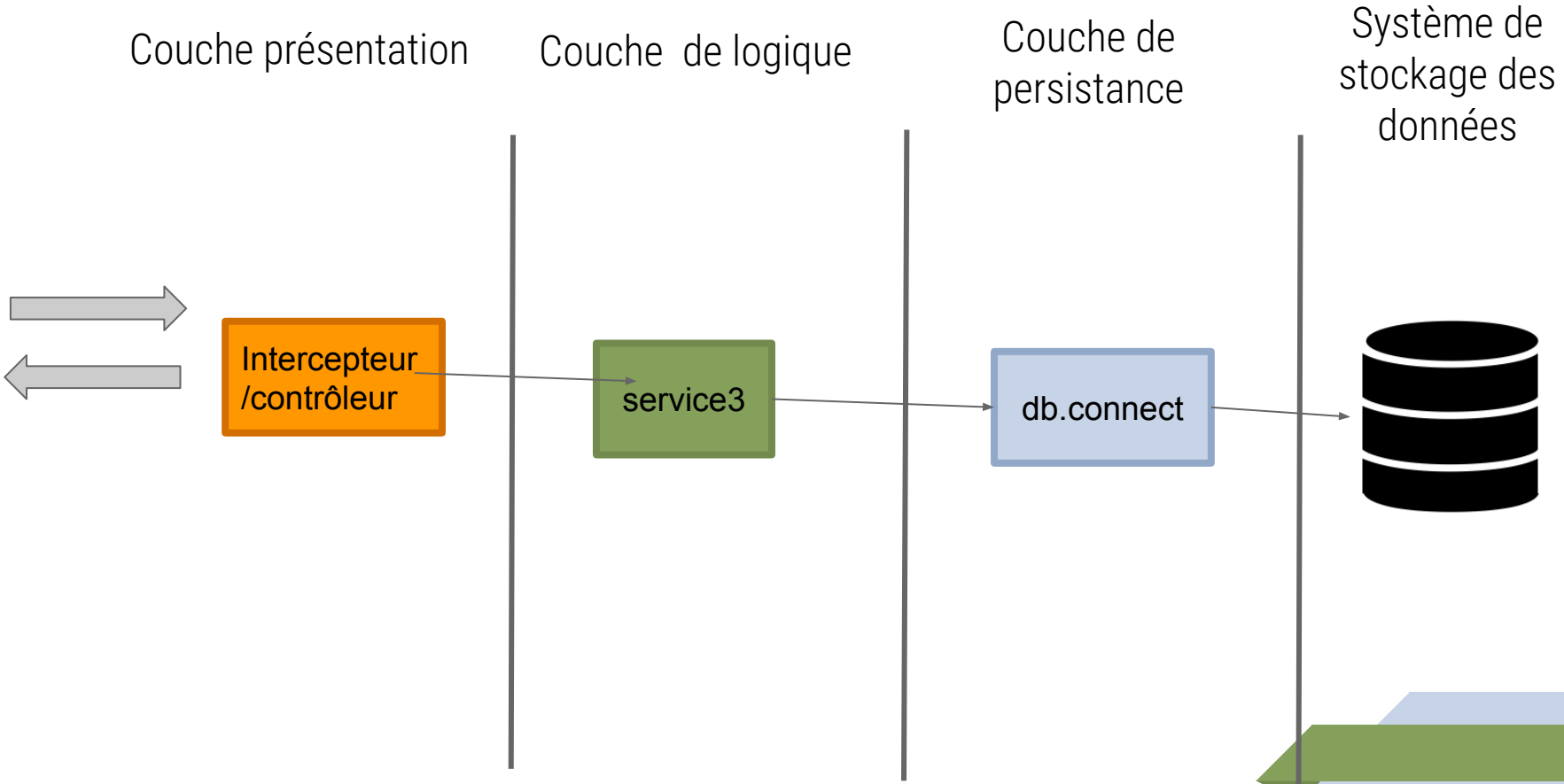
Modèle n-tiers

Plusieurs strates pour autant d'usages

L'une des conceptualisations de cette séparation des tâches au sein des classes d'une application a été théorisée par la création du modèle n-tiers.

Dans cette architecture, l'application est divisée en au moins trois couches:

- La présentation
- La logique
- La persistance des données


Couche présentation
Couche de logique
Couche de persistance
Système de stockage des données
Intercepteur /contrôleur
service3
db.connect

Modèle n-tiers

Couche présentation

Cette couche est responsable de l'interaction de l'application avec l'utilisateur ou le client. Elle peut prendre de nombreuses formes:

- Une page Web
- Une API Rest
- Une application android
- ...

Modèle n-tiers

Couche logique

Cette couche est porteuse de l'intelligence de l'application, de son domaine métier. On y trouvera toutes les règles métiers

C'est elle qui fournira les données calculées à la couche de présentation sous la forme de "**Services**" à appeler depuis les classes de la couche de présentation :

- Un service de récupération des élèves d'une classe
- Un service de calcul d'une fiche de paie
- ...