

# Organiser son code en MVC





# Hein ?

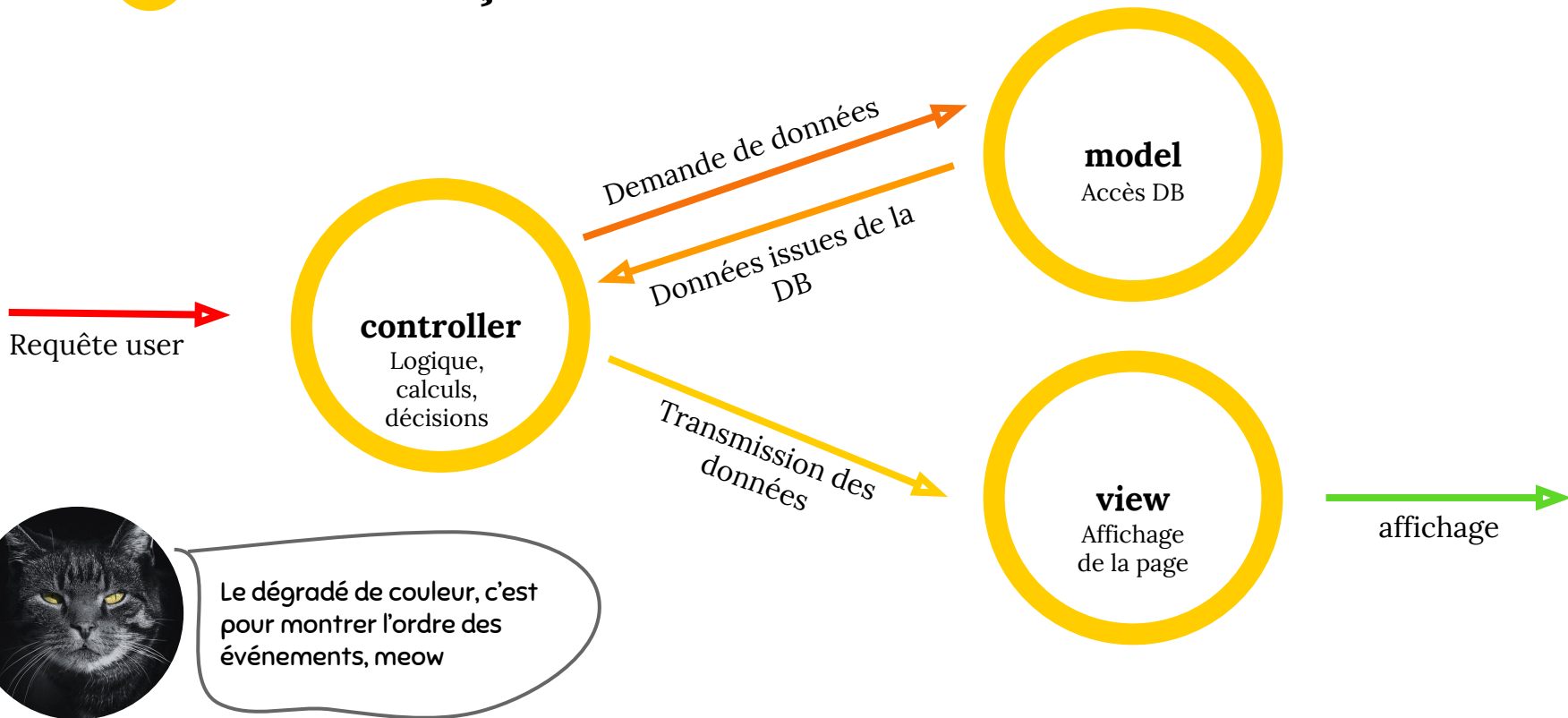
M => Modèle

V => Vue

C => Contrôleur



## Comment ça fonctionne ?





## **Concrètement donc...**

---

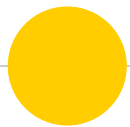
Le contrôleur est l'aiguilleur qui va actionner les bons interrupteurs pour acheminer l'internaute vers sa destination.

Du point de vue du visiteur, il demande une page au contrôleur et il lui renvoie une vue selon certains paramètres (a-t-il l'autorisation pour accéder à cette page par exemple ?).



## Mais pourquoi ?

- Pour mieux **organiser son code de manière stratégique** : la vue ne se charge que de l'affichage, le contrôleur se charge des entrées utilisateur et de les transmettre au modèle.
- Pour respecter une **convention** très utilisée
- Pour pouvoir travailler de manière plus **efficace en équipe**
- Pour **communiquer sous forme d'événements** entre la vue, le modèle et le contrôleur
- Ça permet une structure composite : la vue sera composée de plusieurs éléments superposés les uns aux autres, générés par les interactions entre l'utilisateur et l'application



# Exemple concret

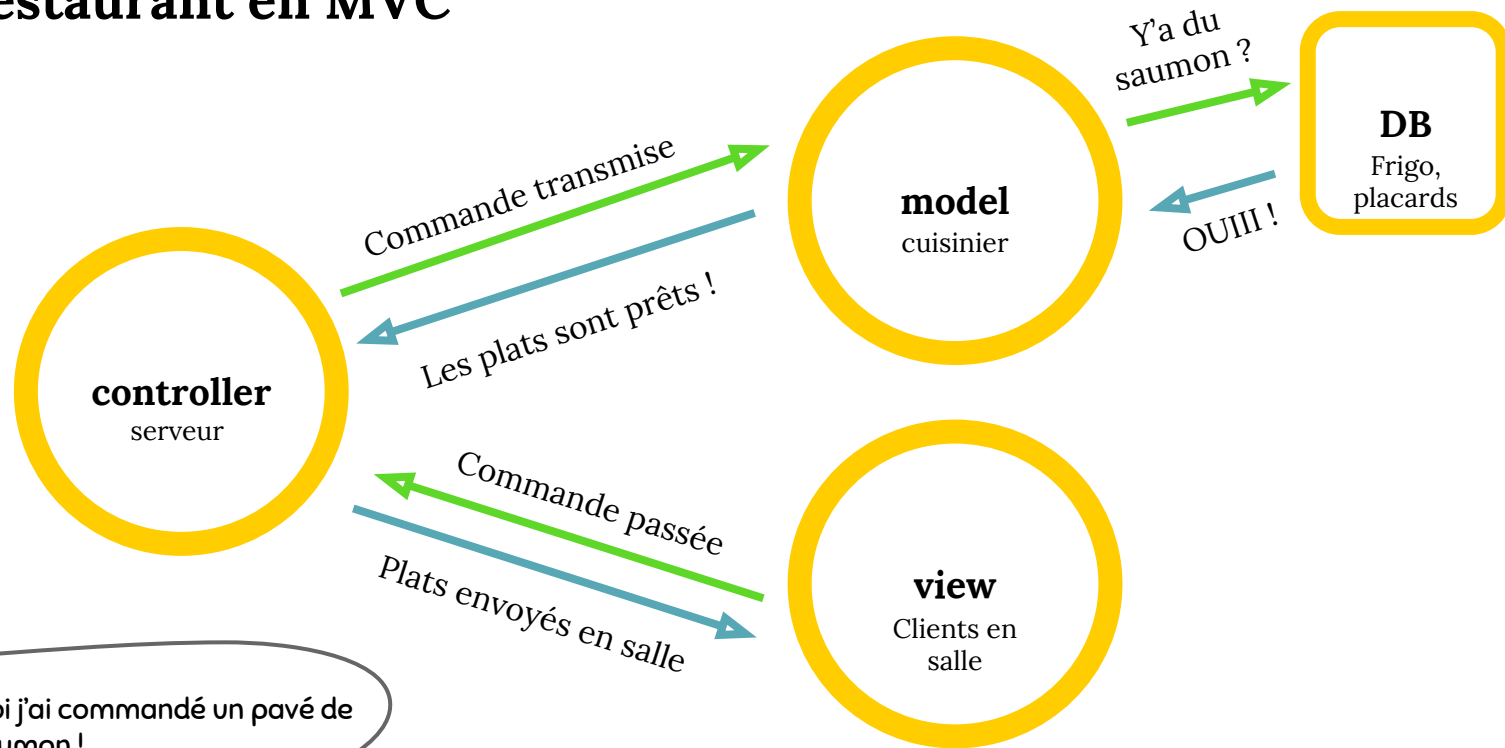
Le modèle MVC transposé sur un restaurant



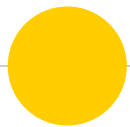
Y'a à manger ?



# Restaurant en MVC



Moi j'ai commandé un pavé de saumon !



# Exemple concret

Le **modèle MVC** transposé sur un **lecteur MP3**



Oui, s'il te plaît !





## Lecteur MP3 en MVC

---

Via l'interface, l'utilisateur peut ajouter des MP3, gérer des playlists, renommer des pistes, modifier le volume.

Le lecteur sauvegarde les informations, les ajouts, les suppressions de pistes,...

Le lecteur lance la lecture, arrête la musique, affiche les informations des morceaux, le volume du son



## Lecteur MP3 en MVC

**Vue** : les différentes interfaces, les différents affichages demandés par l'utilisateur

**Modèle** : là où est stockée la logique de l'application, les états (lecture, pause,...), les données (des morceaux, compositions des playlists,...) pour gérer et lire les morceaux

**Contrôleur** : Lorsque l'utilisateur clique sur le bouton [lecture] :

- le contrôleur demande au modèle de jouer le morceau choisi,
- le modèle informe la vue du morceau qui est en train d'être joué,
- l'interface graphique affiche les informations du morceau qui est en train d'être joué.



## A toi de jouer !

---

Fais un petit site.

Si l'internaute n'est pas connecté, la page d'accueil affiche "*coucou l'inconnu*". Si l'internaute est connecté, la page d'accueil affiche "*coucou*" + le nom de l'utilisateur.

Fais un petit menu avec les boutons "accueil" et "admin".

Si on n'est pas connecté, en cliquant sur "admin", on est redirigé vers l'accueil. Si on est connecté, on arrive sur une page avec le message "*bienvenue dans le secret des dieux !*".

Dans le contrôleur : tu gères les redirections de l'utilisateur (condition de connexion).

Dans le modèle : tu gères les données des variables de session.

Dans la vue : tu gères l'affichage.



## Ressources

---

- [Adopter le MVC](#) avec exemples (Openclassrooms)
- [Bonnes pratiques](#) à connaître (openclassrooms)