

DESIGN OOP

Designer une application avec la
Programmation Orientée Objet



Compétence demandée :
Savoir construire le design d'une
application web

1. Rappel architecture MVC
2. Layered architecture
3. UML : Diagramme de classes
4. Exercices

ARCHITECTURE D'UN APPLICATION WEB

Couper son code en 3 parties
M, V et C

MVC :
Model
View
Controller

MVC :
Model = Entités

MVC :

Model = Entités

View = Code pour le visuel

MVC :

Model = Entités

View = Code pour le visuel

Controller = Le reste

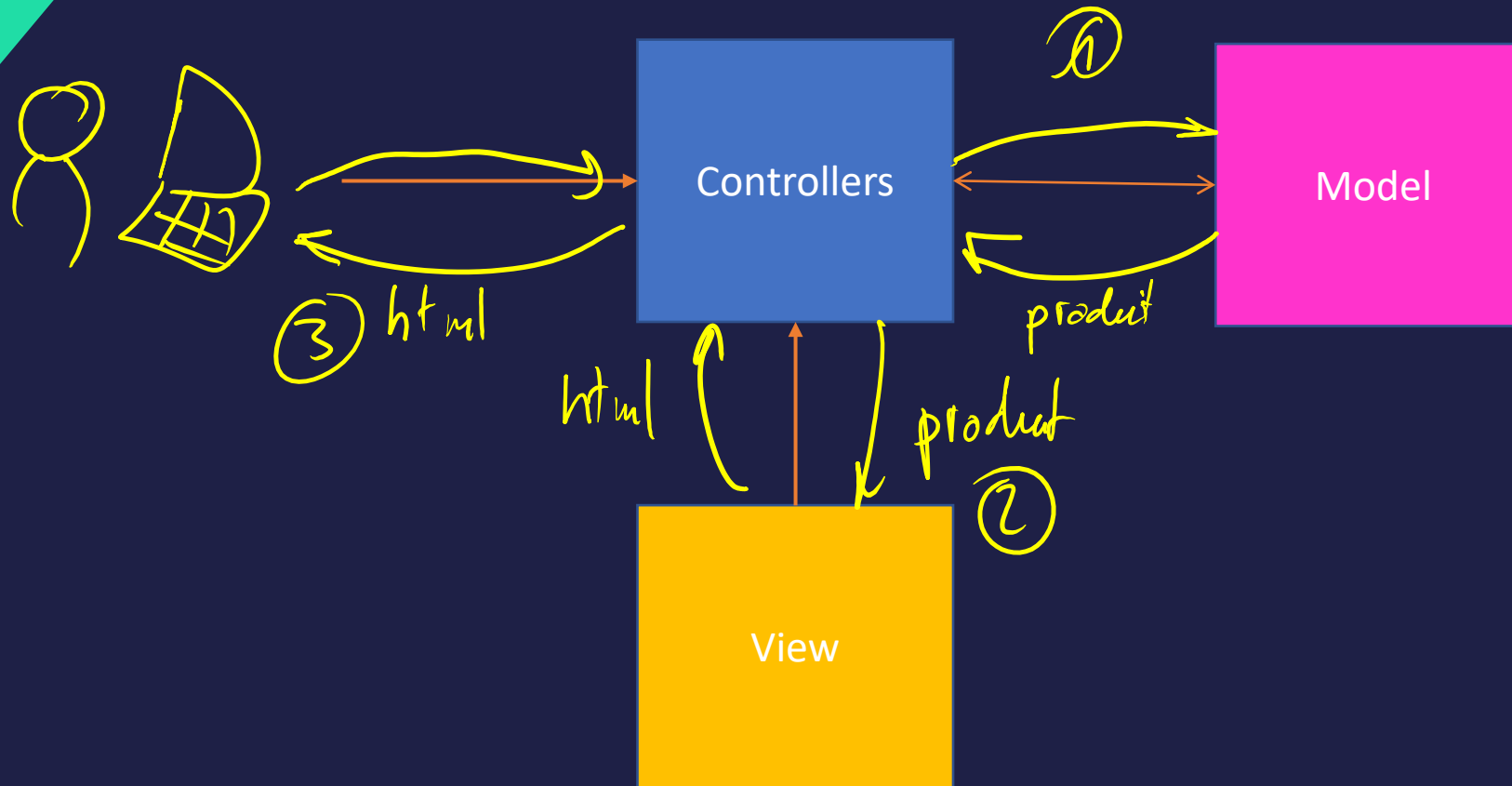
MVC :

Model = Entités

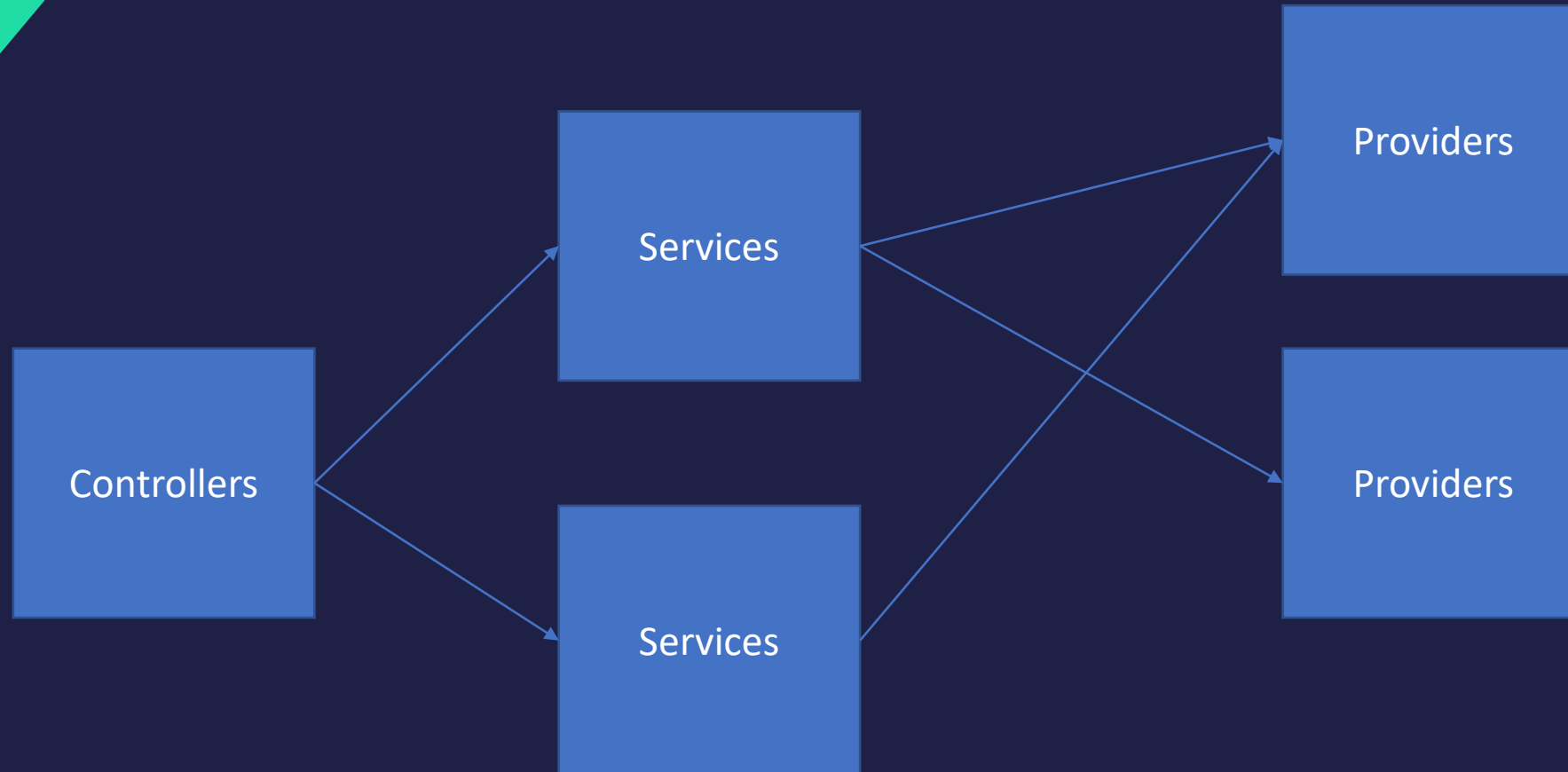
View = Code pour le visuel

Controller = Le reste

Architecture MVC



MVC + Layered architecture



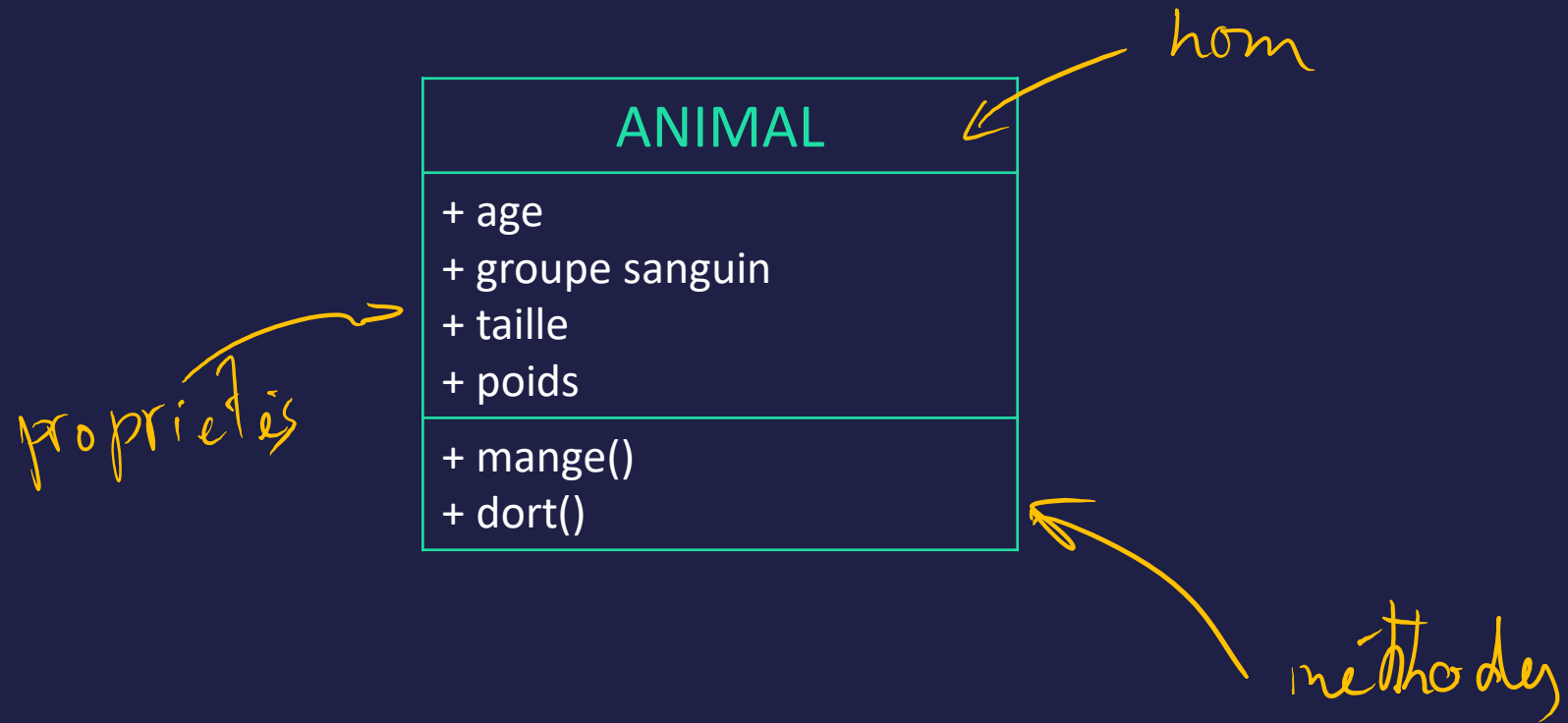
UML : DIAGRAMME DE CLASSES

L'**UML** (Unified Modeling Language), est un **langage de modélisation graphique** à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du **développement logiciel et en conception orientée objet**.

Le **diagramme de classes** est un schéma utilisé **en génie logiciel** pour présenter **les classes et les interfaces des systèmes ainsi que leurs relations**.

Ce diagramme fait partie de la partie statique d'UML, ne s'intéressant pas aux aspects temporels et dynamiques.

Les classes



Les relations

Héritage

Héritage

Héritage Implémentation

Héritage Implémentation Composition

Héritage
Implémentation
Composition
Agrégation



Héritage
Implémentation
Composition
Agrégation

relation
d'inclusion

corps agrégation organes
corps composition familles

<https://refactoring.guru/fr/design-patterns/catalog>

EXERCICES

METHODOLOGIE

1. Développer les modèles
2. Développer les providers (repositories ou managers)
3. Développer les services et/ou controllers

Pour les exercices suivants, nous nous limiterons à 3 entités

X  NATIS

Youtube

Messenger

Doctolib

Spotify

Gmail

Booking.com

Fitbit

AppStore

Amazon

BNP Paribas

