

# Model Driven Engineering

## MDE

Zineb BOUGROUN



# Plan

- Introduction
- Model driven architecture MDA



# Introduction

## Pourquoi faire une modélisation?

- Un nouveau besoin :
  - Les systèmes deviennent plus complexes
  - Volume croissant des données
  - Évolutivité croissante
    - partie plateforme
    - Partie métier
    - Langages de programmation
  - Hétérogénéité
    - Support de données
    - Technologies, langages ...



# Introduction

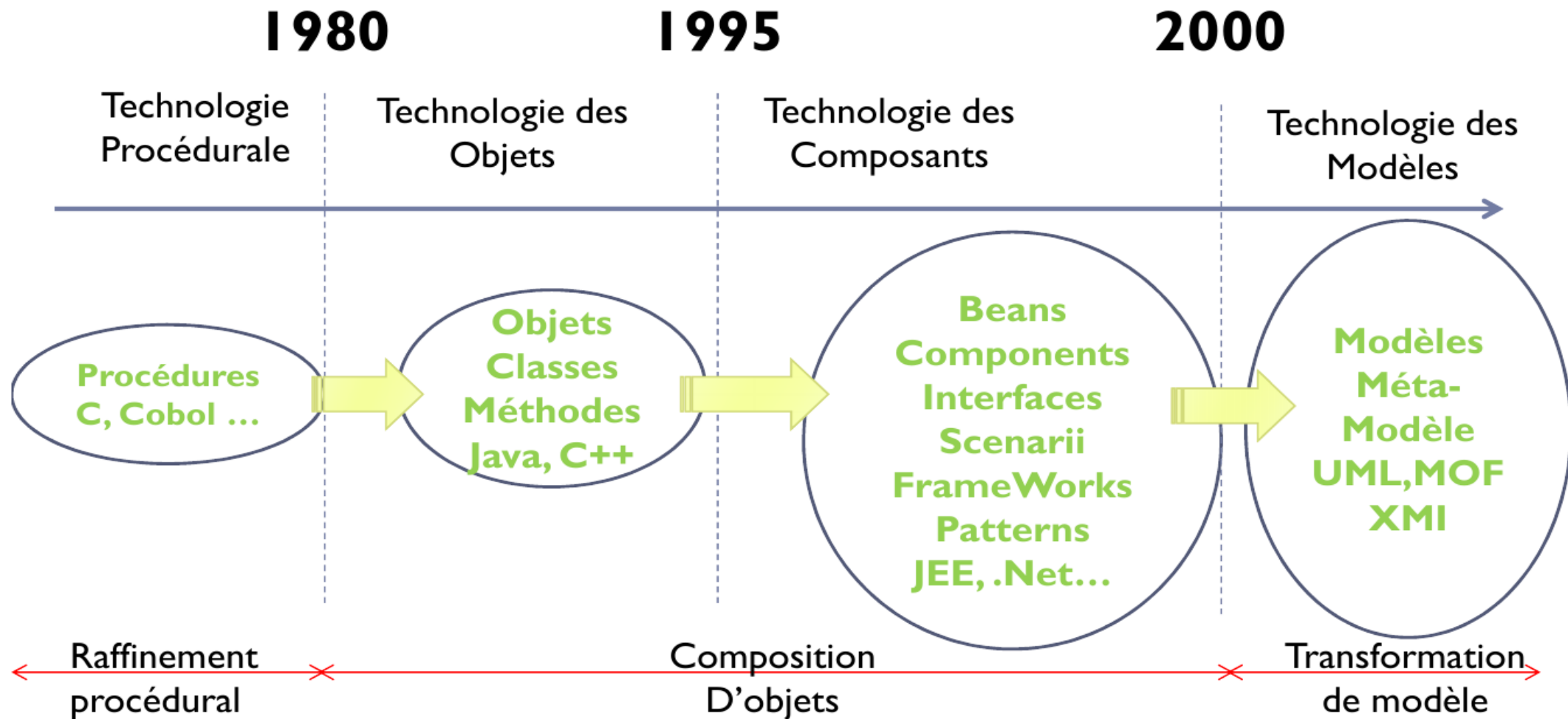
## Pourquoi faire une modélisation?

- Pour une Entreprise, il est crucial de posséder la dernière technologie
- Mais, pour la même application, il faut tout recommencer!
- Les limitation des technologies :
  - Les objets ont échoué dans la recherche de simplicité,
  - l'extensibilité, l'intégration
  - Les services ne sont pas des objets
  - La technologie des composants n'est qu'une fuite en avant et apporte peu d'espoir pour une sortie de crise



# Introduction

Pourquoi faire une modélisation?



# Introduction

## Pourquoi faire une modélisation?

- Afin d'assurer l'extensibilité, et de gagner du temps :

L'élément de base n'est plus l'objet : c'est le modèle!



# Introduction

## Définition de modèle

- Un modèle pourrait être défini comme une représentation abstraite d'une réalité.
- Plusieurs entités distinctes peuvent être représentées par un unique modèle donné.
- Exemple
- Un plan de votre maison est le modèle à partir duquel on a construit la maison que vous habitez. Ce plan regroupe des caractéristiques de dimension que votre maison respecte. Dans le contexte de la modélisation, on dit alors que votre maison est représentée par son modèle (son plan).



# Introduction

## Méta modèle définition

- Le préfixe « méta » exprime l'auto-référence :
  - Méta-physique : la physique de la physique
  - Méta-classe : la classe des classes (`java.lang.Class` en Java)
  - Méta-table : la table des tables (`user_tables` pour Oracle)
  - Méta-modèle : le modèle d'un modèle ?
- Modèle d'un ensemble de modèles
  - Grammaire décrivant un langage de modélisation
  - Définit la syntaxe abstraite des modèles
  - Un métamodèle ne définit que la structure et pas la sémantique.





# Introduction

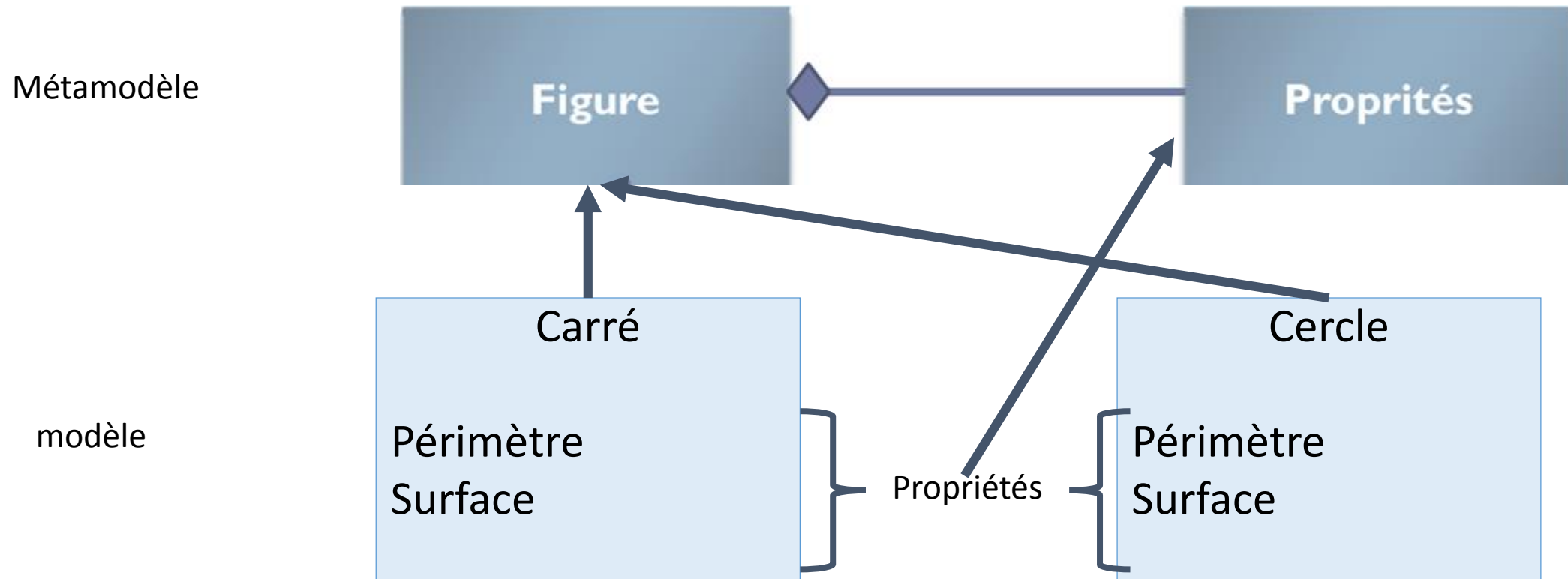
## Pourquoi méta modèle ?

- Pour éditer et valider les modèles.
- Savoir si le modèle est Ok ou non.
- Pour transformer des modèles :
  - Règles de transformation entre méta-modèles



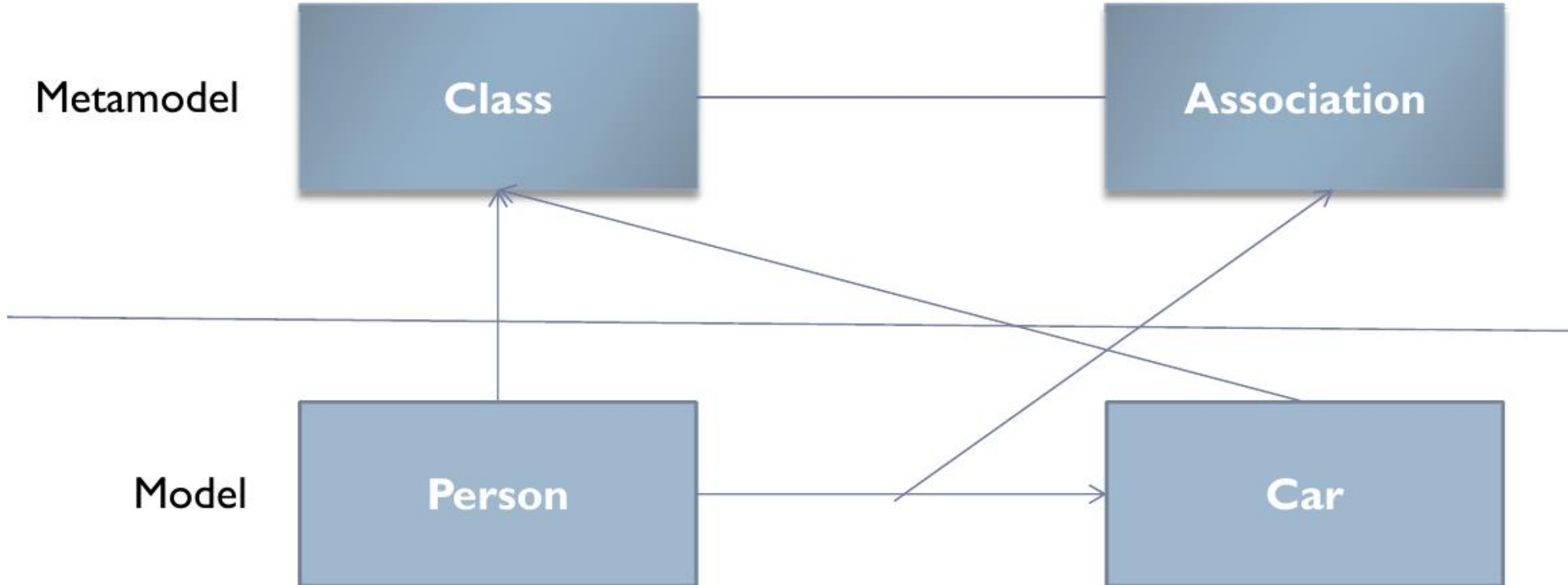
# Introduction

## Exemple



# Introduction

## Exemple



# Introduction

## Méta méta modèle définition

- Modèle d'un ensemble de méta-modèles
- Grammaire décrivant les grammaires décrivant des langages de modélisation
- Définit la syntaxe abstraite des méta-modèles
- C'est le MOF : Méta Object Facilities
  - Le MOF définit le langage permettant de définir des méta modèles
  - Les concepts du MOF sont les concepts de méta-classe méta-attribut méta-association, etc.



# Model Driven Architecture

## OMG?

- Object Management Group (OMG)
  - Organisme international dont les membres se réunissent cinq fois par an pour définir collectivement les orientations des standards et des architectures
- Constats :
  - Il n'y aura jamais de système unique, de langage unique, de système unique, de SGBD unique, de réseau unique, etc
  - La seule caractéristique dont on soit certain c'est l'hétérogénéité
  - La recherche de l'interopérabilité est une activité qui ne peut devenir que de plus en plus importante



# Model Driven Architecture

## Définition

- Model Driven Architecture (MDA)
  - nouvelle méthode de génie logiciel initié par l'OMG
  - Propose une séparation systématique et nette dans la logique métier de l'entreprise et la logique d'implémentation

Méthode = Langage + Démarche + Outils



# Model Driven Architecture

## Intention du MDA

- Pérennité des savoir-faire
  - La durée de vie des modèles doit être plus longue que celle du code (modèles abstraits)
- Gains de productivité
  - L'utilisation des modèles ne doit plus être seulement contemplative mais productive (automatisation des opérations sur les modèles et génération de code)
- Prise en compte des plateformes d'exécution
  - La prise en compte des plateformes doit être explicitée par des modèles



# Model Driven Architecture

## Intention du MDA

- Modèle des exigences
  - Computation Independant Model (CIM)
- Modèle métier
  - Platform Independant Model (PIM)
- Modèle technique
  - Platform Specific Model (PSM)





# Model Driven Architecture

## Modèle des exigences

- Computation Independant Model (CIM)
  - Il est souvent considéré comme un modèle d'entreprise ou de domaine, vu qu'il utilise un vocabulaire familier aux experts en la matière (PME).
  - Il présente exactement ce que le système est censé faire, mais cache toutes les technologies de spécifications et reste indépendant de la façon dont ce système sera (ou est en cours) mis en œuvre.



# Model Driven Architecture

## Modèle métier

- Platform Independant Model (PIM)
  - Il permet l'extraction du concept commun de l'application indépendamment de la plateforme cible permettant sa cartographie sur une ou plusieurs plateformes.
  - Ceci est généralement réalisé par la définition d'un ensemble de services d'une manière qui fait abstraction des détails techniques.



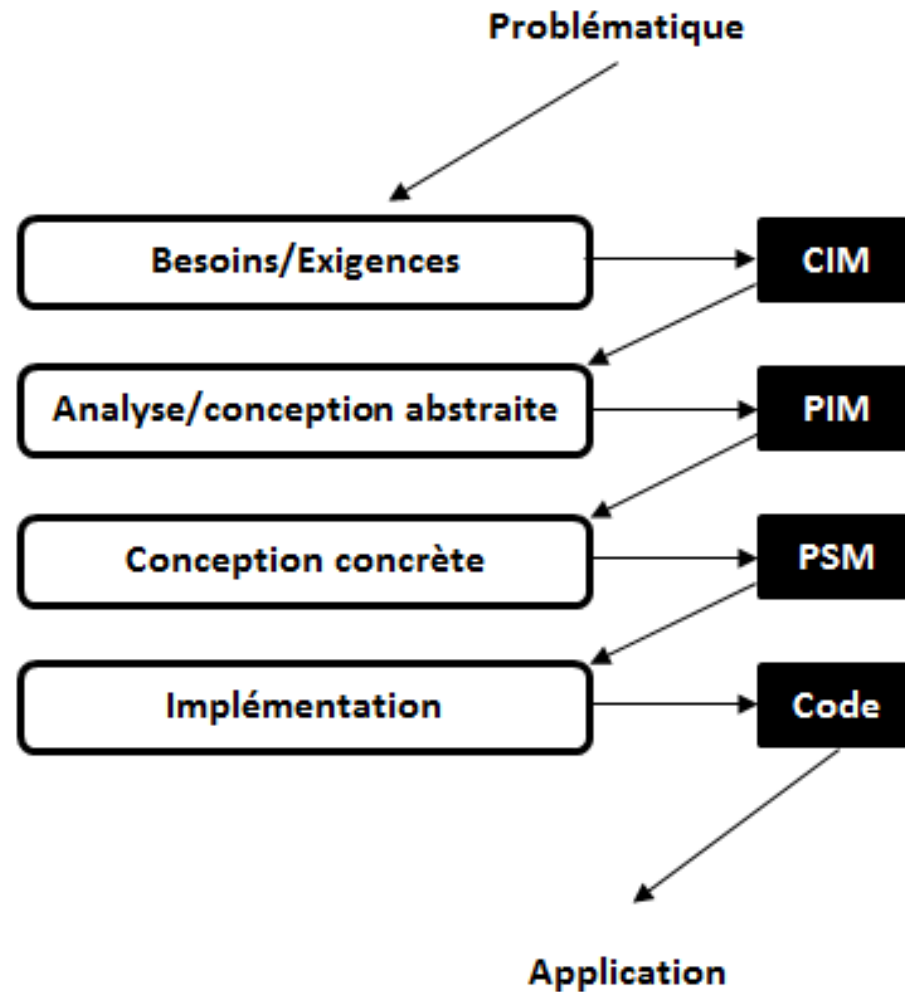
# Model Driven Architecture

## Modèle technique

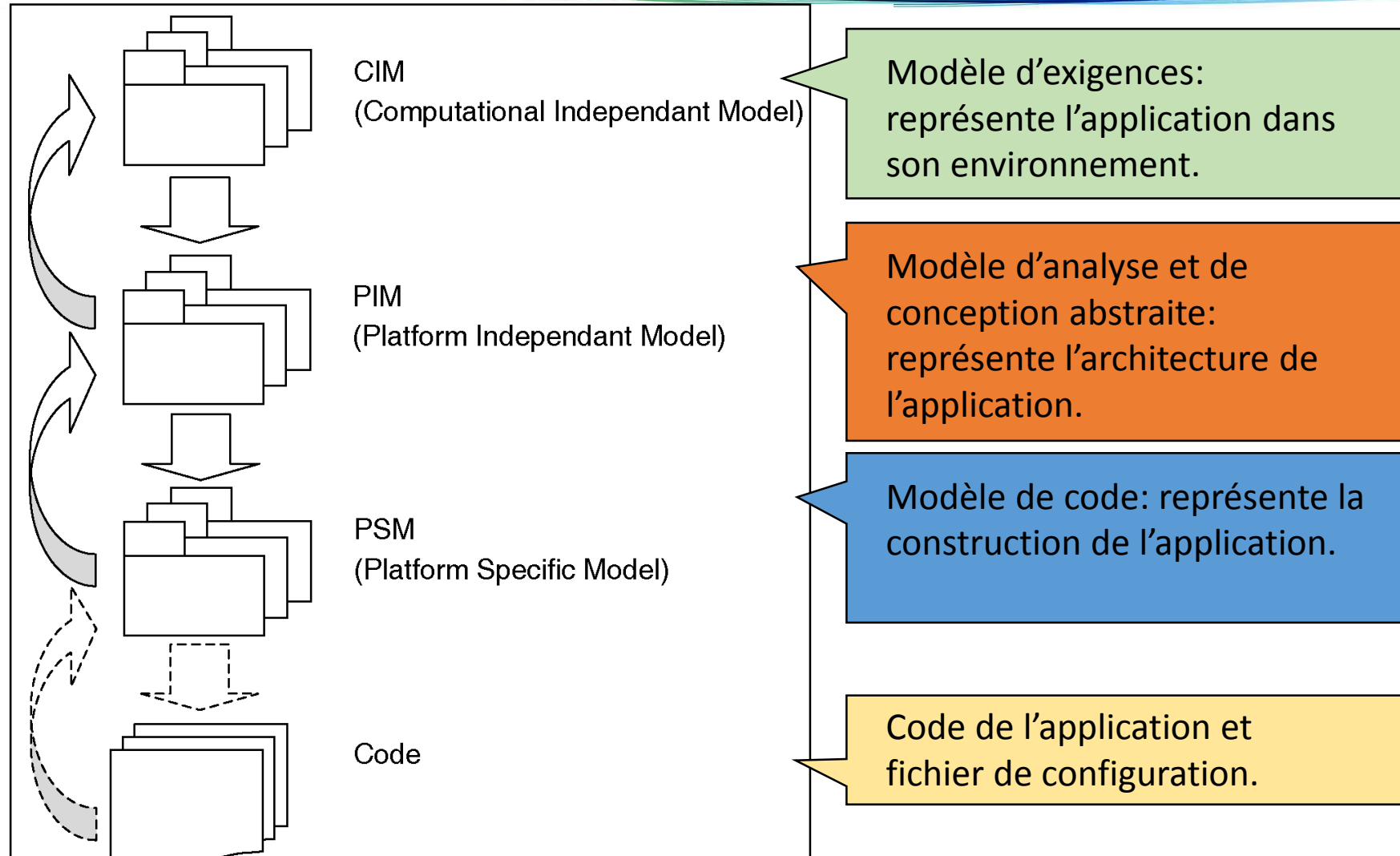
- Platform Specific Model (PSM)
  - Il combine les spécifications dans le PIM avec les détails nécessaires pour préciser la façon dont un système utilise un type particulier de plate-forme.
  - Il est appelé aussi appelé modèle de code ou de conception concrète et définit une base de la génération de code



# Model Driven Architecture

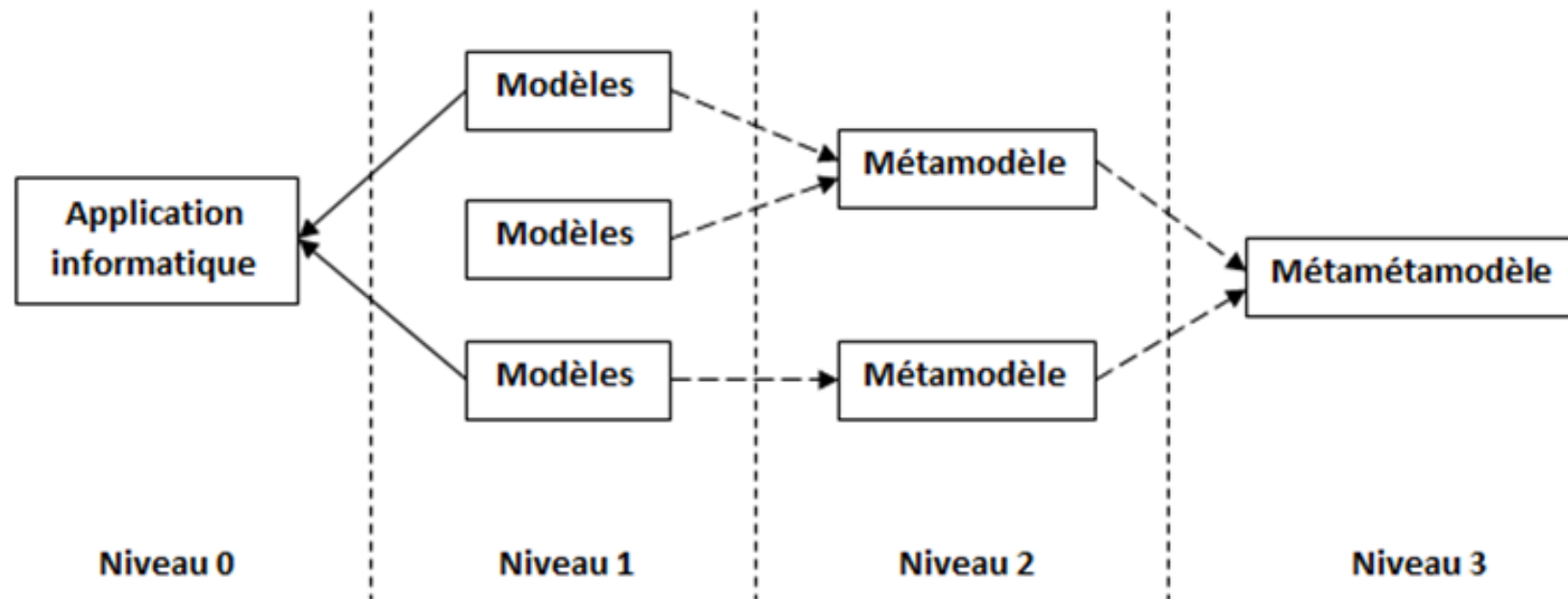


# Model Driven Architecture



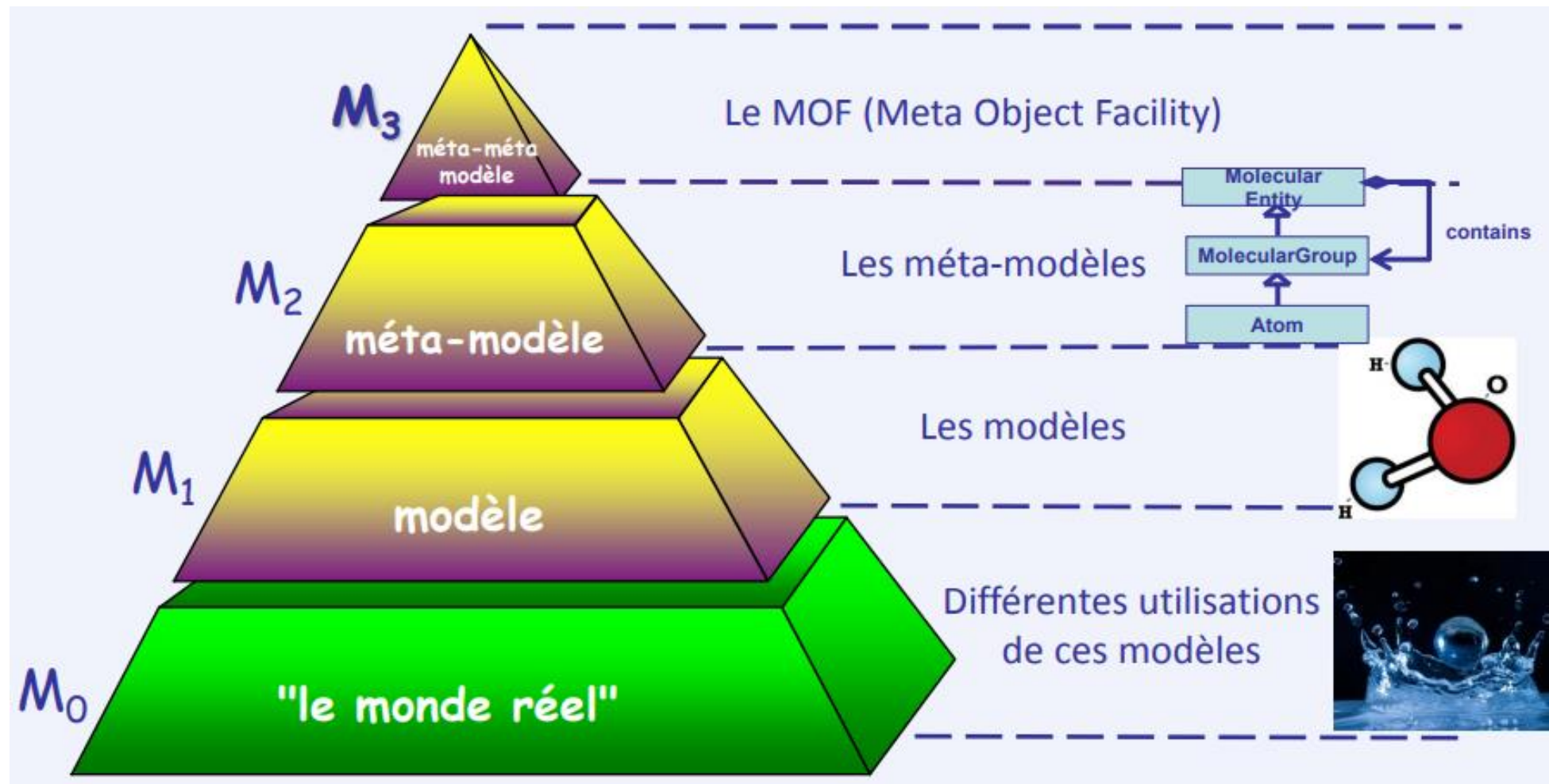
# Model Driven Architecture

L'architecture à 04 niveaux de MDA



# Model Driven Architecture

L'architecture à 04 niveaux de MDA : exemple



# Model Driven Architecture

## Architecture MDA

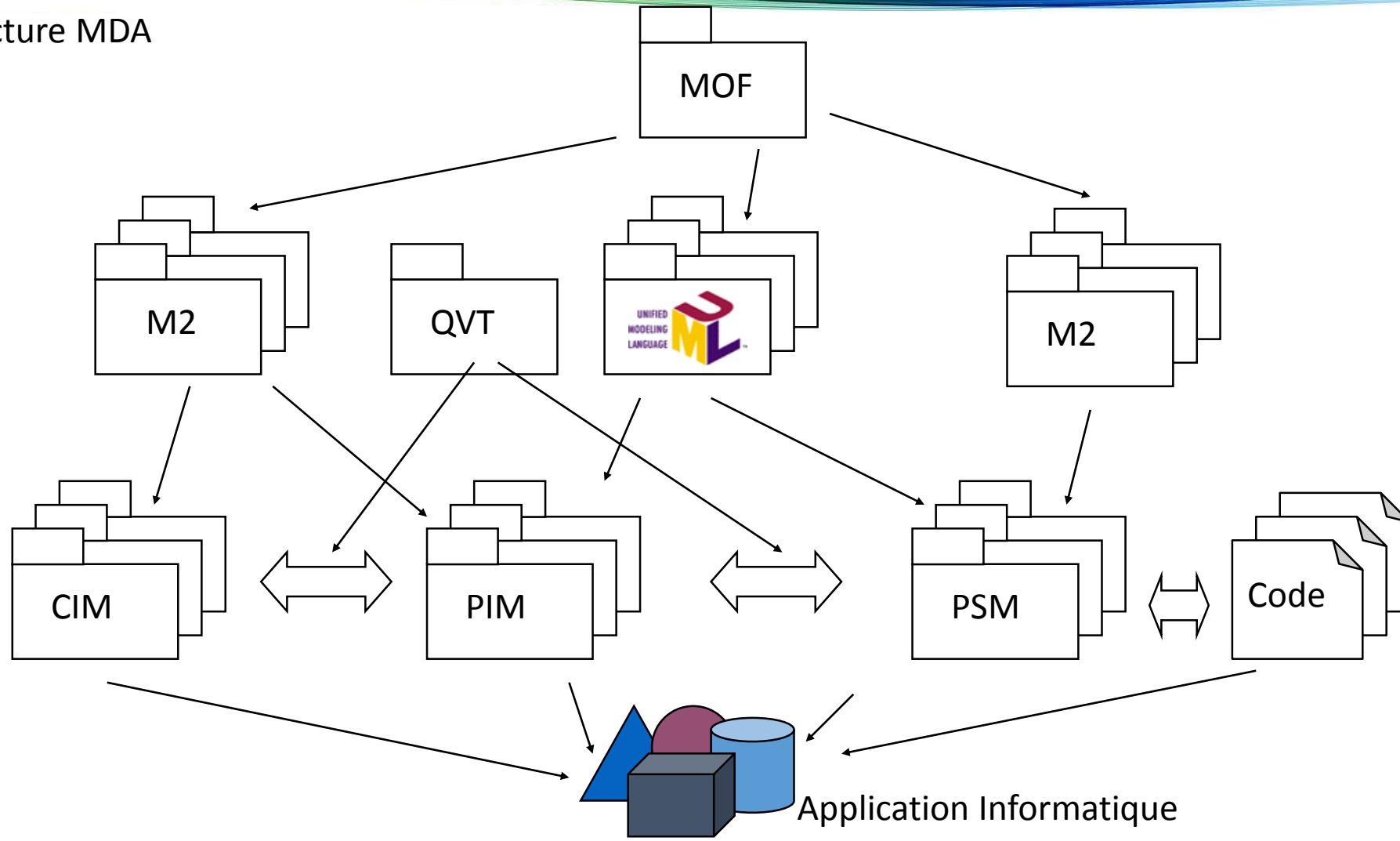
- M3 : méta-méta-modèle ou modèle du MOF
  - décrit les éléments d'un langage de modèles UML, Merise...
- M2 : méta-modèle d'UML
  - décrit les éléments d'un langage de modélisation
- M1 : modèle
  - décrit un modèle de système
- M0 : instance d'un modèle (d'UML) ou système
  - instances du système





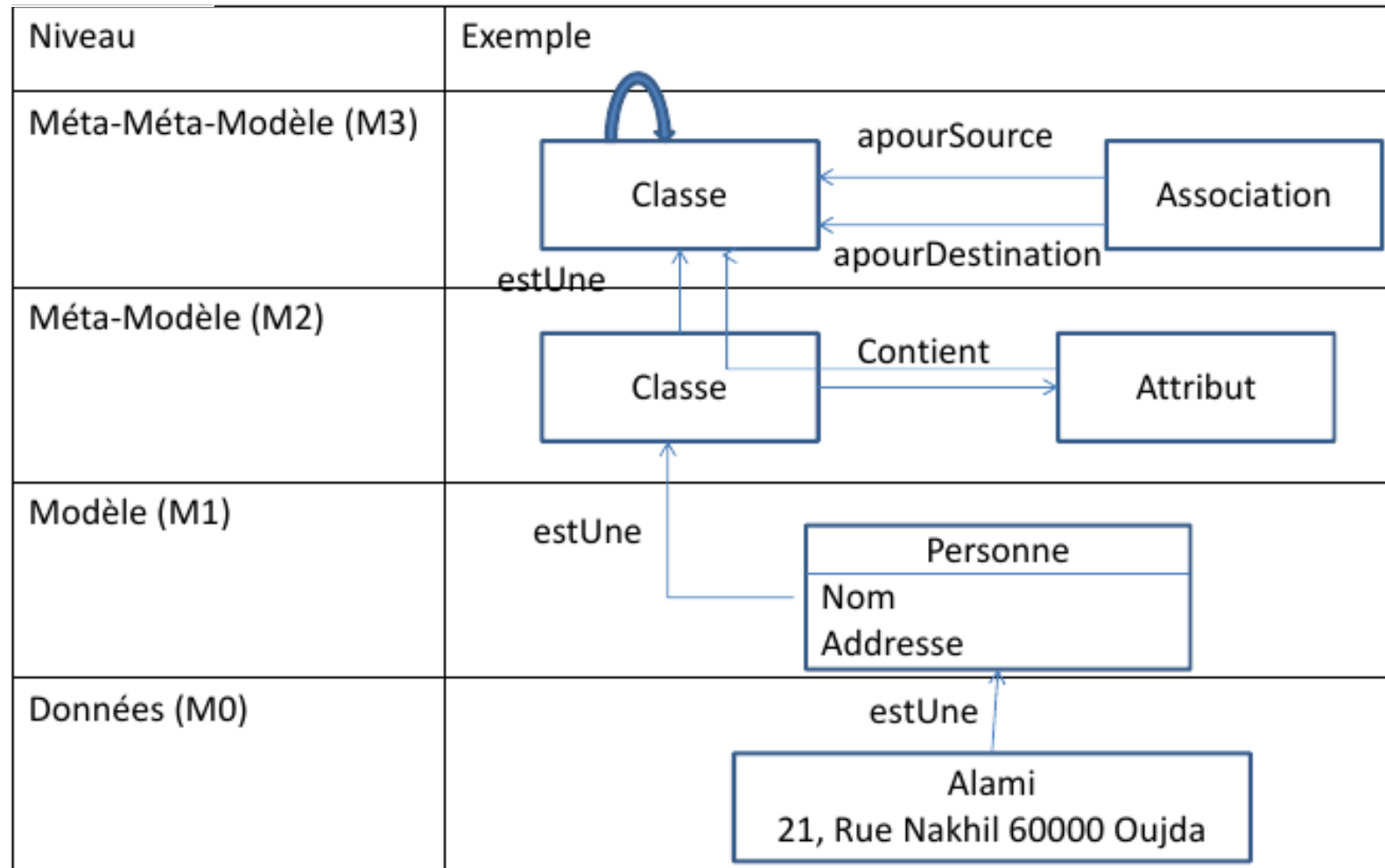
# Model Driven Architecture

Architecture MDA



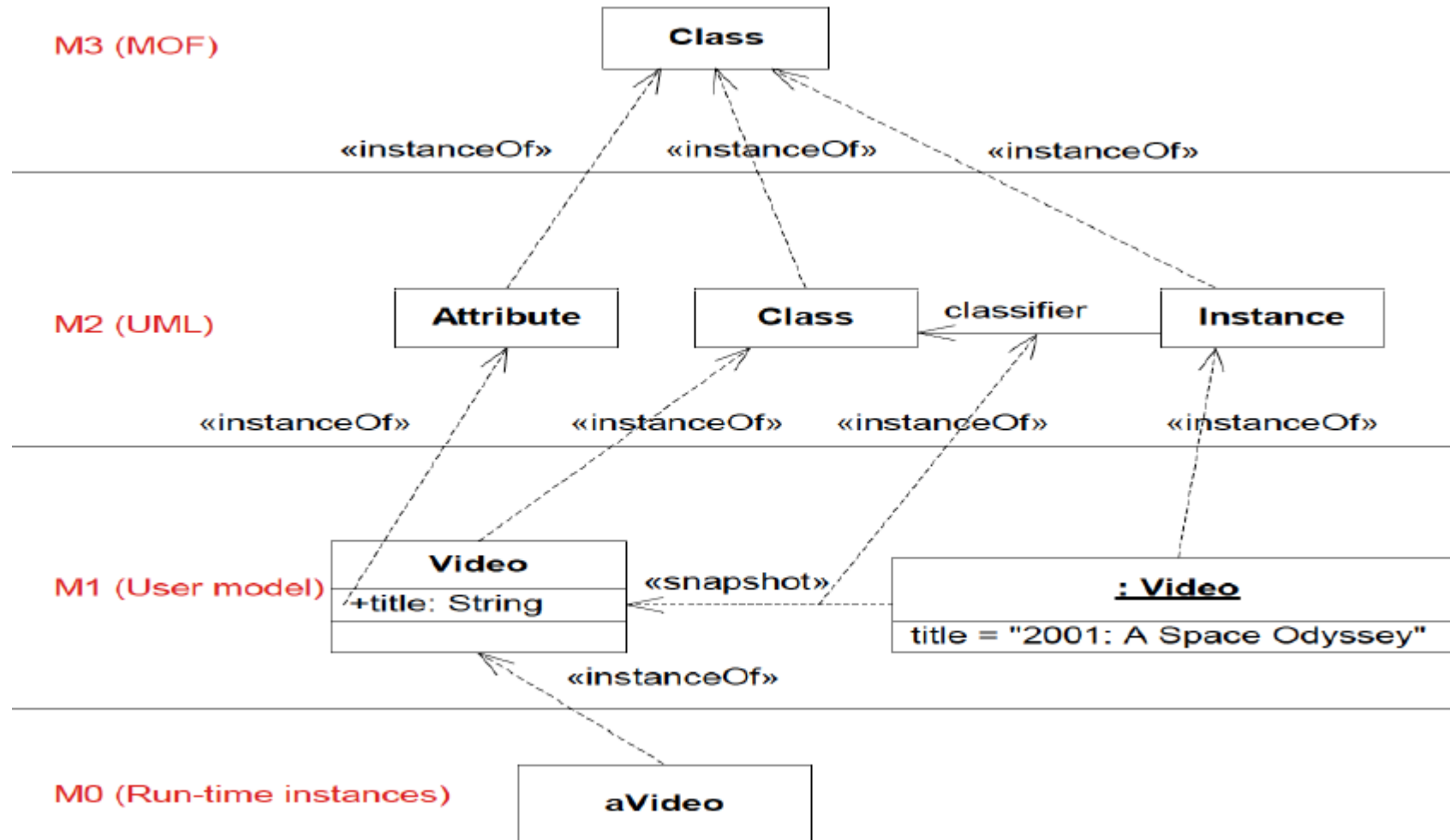
# Model Driven Architecture

Architecture MDA : exemple



# Model Driven Architecture

Architecture MDA : exemple



# Model Driven Architecture

## MDA et UML

- UML est actuellement le métamodèle le plus important de l'approche MDA.
- C'est le métamodèle qui définit la structuration des modèles des applications informatiques
- UML2.0 supporte différents niveaux d'abstractions et différents points de vue :
  - Cas d'utilisation, Séquences, Structuration Interne, Etats,
  - Déploiement, etc.



# Model Driven Architecture

## MDA et UML

- UML permet aussi de représenter une application dans son environnement afin d'en faire ressortir les exigences (cas d'utilisation)
- UML peut être utilisé pour les CIM (Computational Independent Model)
  - Les modèles UML utilisés pour les modèles d'exigence sont : le diagramme de cas d'utilisations et le diagramme de séquences.
  - Ces diagrammes permettent d'exprimer les besoins de l'utilisateur et sont indépendants des plateformes d'exécution.



# Model Driven Architecture

MDA et UML

- UML permet principalement de construire des modèles d'applications informatiques indépendants des plates-formes d'exécution
  - UML est le métamodèle pour les PIM (Platform Independent Model)
  - Les modèles UML utilisés pour la phase analyse et conception abstraite sont : le diagramme de classes, le diagramme d'activités, le diagramme d'états et le diagramme de séquences.



# Model Driven Architecture

## MDA et UML

- UML permet aussi de construire des modèles d'applications informatiques en respectant l'architecture des plates-formes d'exécution
  - UML est le métamodèle pour les PSM (Platform Specific Model)
  - Les modèles UML utilisés pour la conception concrète sont : le diagramme de classes et le diagramme de composants. MDA conseille l'utilisation des profils UML pour l'élaboration des PSM.



# Model Driven Architecture

MDA : pratique

- Cas d'utilisation : définition
- Un diagramme de cas d'utilisation contient des acteurs, un système et des cas d'utilisation
  - Un acteur a un nom et est relié aux cas d'utilisation
  - Un acteur peut hériter d'un autre acteur
- Un cas d'utilisation a un intitulé et peut étendre ou inclure un autre cas d'utilisation
- Le système a lui aussi un nom, et il inclut tous les cas d'utilisation





# Model Driven Architecture

MDA : pratique

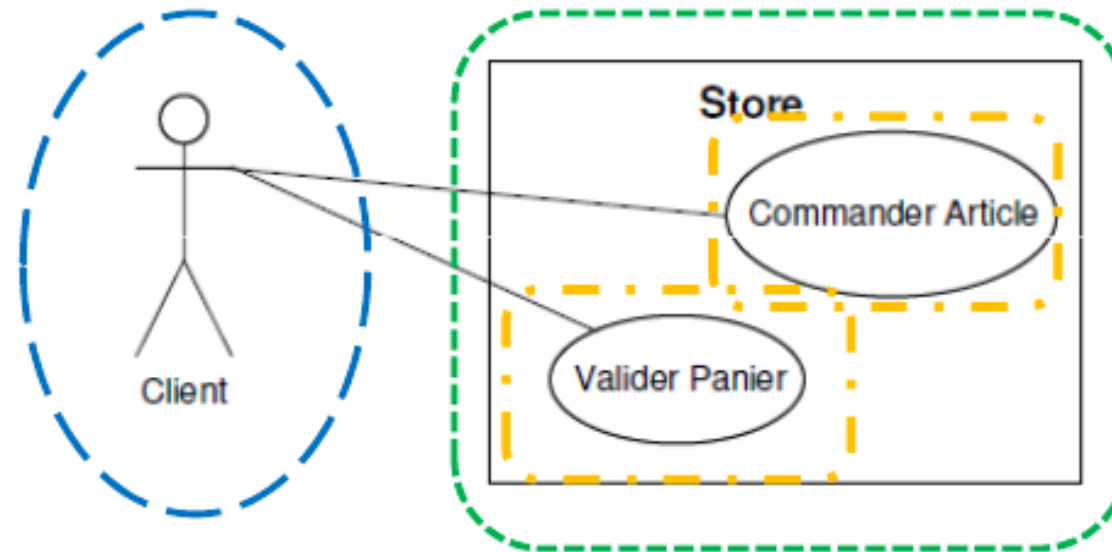
- Cas d'utilisation : M0
  - Dans le système Store,
    - le client X commande l'article A1 et valide son Panier P1
    - Le client Y commande les articles A2, A3 et valide son panier P2
    - ...



# Model Driven Architecture

MDA : pratique

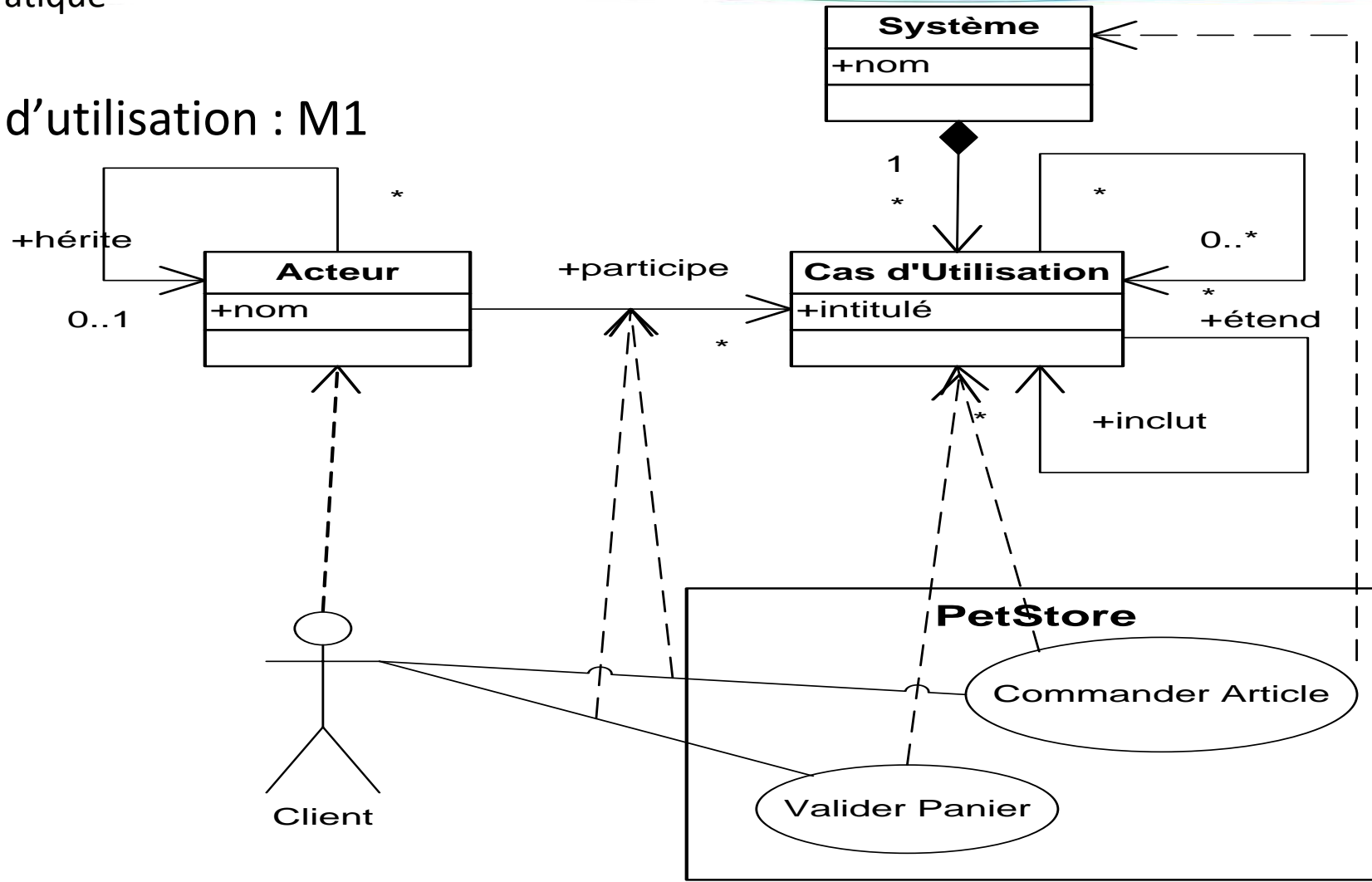
- Cas d'utilisation : M1



# Model Driven Architecture

MDA : pratique

- Cas d'utilisation : M1



# Model Driven Architecture

MDA : pratique

- Proposer un méta modèle du diagramme de classe



# Model Driven Architecture

MDA : pratique

