

LA MANU

ÉCOLE SUPÉRIEURE
DES MÉTIERS DU NUMÉRIQUE



UML

CONFIDENTIEL

Propriété de NOVEI FORMATION, reproduction et utilisation interdite sans accord écrit préalable

20/11/2023

Séance - UML

Définition

Langage de modélisation unifié. C'est un **langage de modélisation graphique** à base de pictogrammes conçu pour fournir une **méthode normalisée** pour visualiser la **conception d'un système**.

Il est couramment utilisé en conception orientée objet.

La version actuelle est UML 2.5

Type de diagrammes

Il existe 14 diagrammes dont 7 structurels et 7 comportementaux.

Diagrammes structurels :

- Diagramme de classes
- Diagramme d'objets
- Diagramme de composants
- Diagramme de déploiement
- Diagramme des paquets
- Diagramme de structure composite
- Diagramme de profils

Diagrammes comportementaux :

- Diagramme des cas d'utilisation
- Diagramme états-transitions
- Diagramme d'activité
- Diagramme de séquence
- Diagramme de communication
- Diagramme global d'interaction
- Diagramme de temps

Diagrammes les plus utilisés

- Diagramme des cas d'utilisation
- Diagramme de séquence
- Diagramme de classes
- Diagramme de déploiement

Diagramme de cas d'utilisation

Cas d'utilisation : Définition (1)

Objectif : Comprendre les **besoins du client** pour rédiger le cahier des charges fonctionnel

3 questions:

1. A quoi sert-il ? (Utilisations principales)
2. Qui va l'utiliser ou interagir avec lui ? (environnement du système)
3. Où s'arrête sa responsabilité ? (Limites du système)

Cas d'utilisation : Définition (2)

- **Ensemble de scénarios** réalisant un **objectif** de l'utilisateur
- **Fonctionnalités principales** du système du point de vue **extérieur**

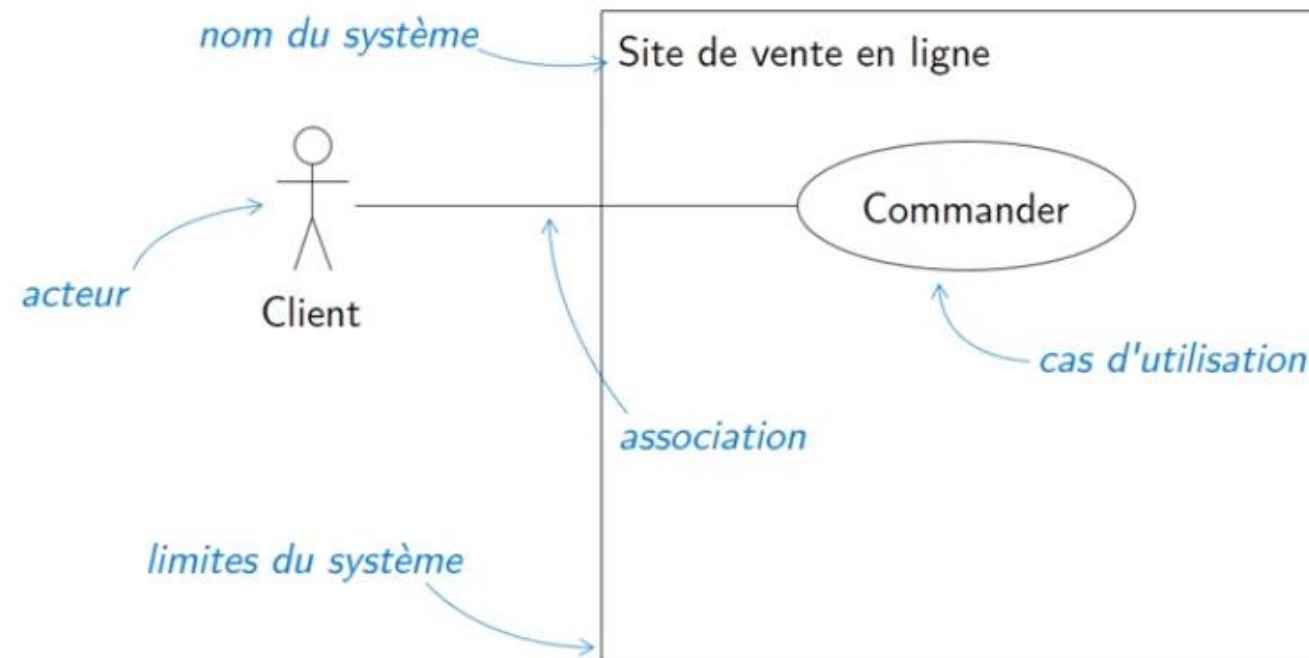
Acteur : Entité qui **interagit** avec le système

- Personne, chose, logiciel, **extérieur au système** décrit
- Représente un **rôle** (plusieurs rôles possibles pour une même entité)
- Identifié par le **nom du rôle**

Cas d'utilisation : **Fonctionnalité** visible de l'extérieur

- Action **déclenchée** par un acteur
- Identifié par une **action** (verbe à l'infinitif)

Cas d'utilisation : Définition (3)

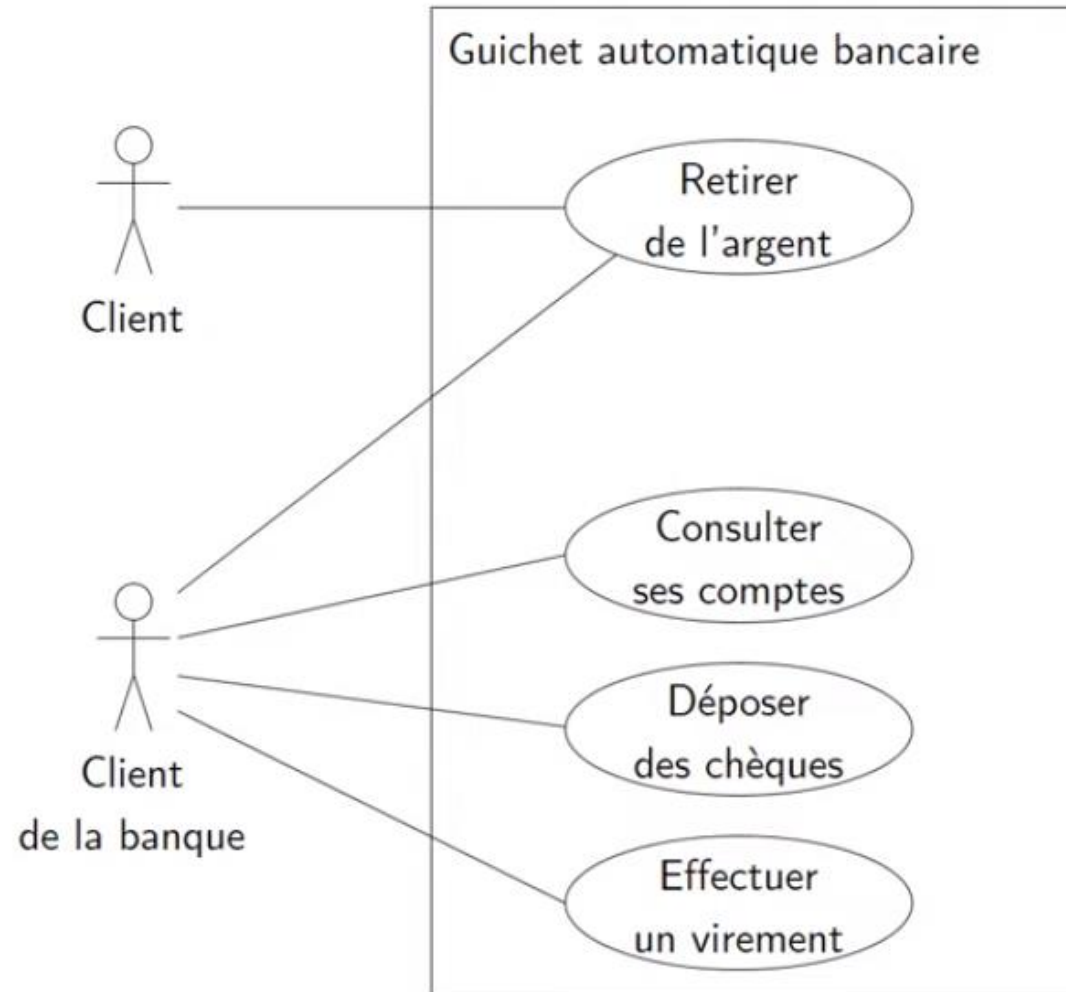


Association :

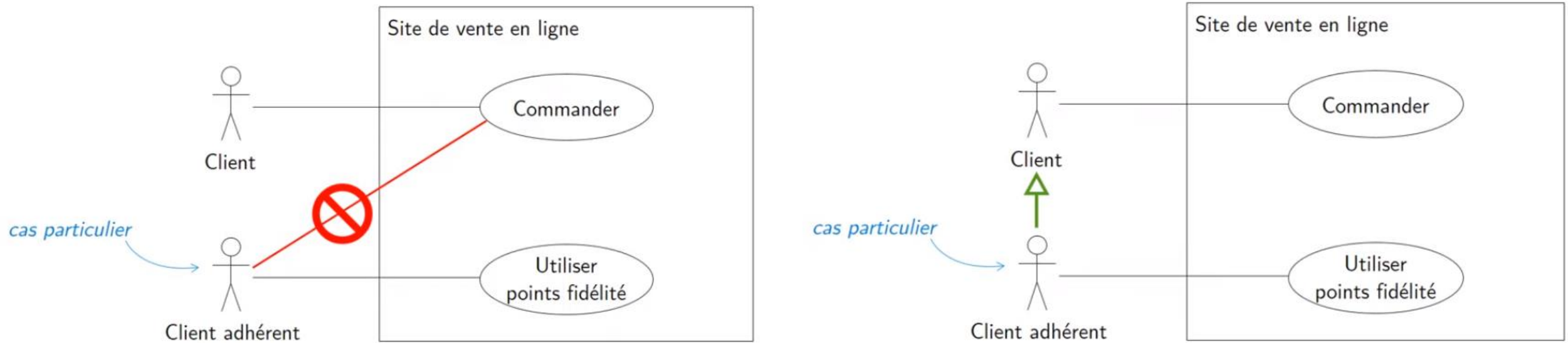
Relation entre **acteurs** et **cas d'utilisation**

Représente la possibilité pour l'acteur de **déclencher** le cas

Cas d'utilisation : Exemple



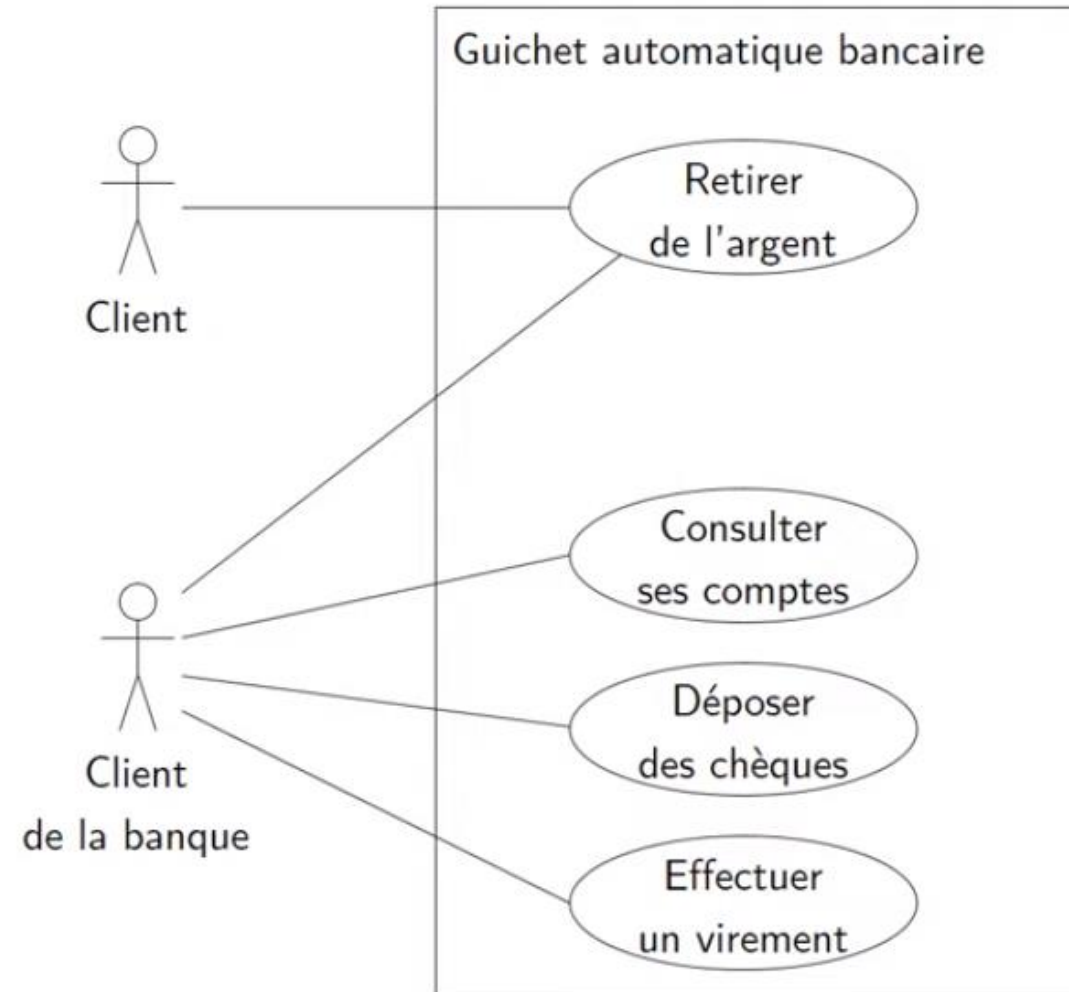
Cas d'utilisation : Généralisation de rôle



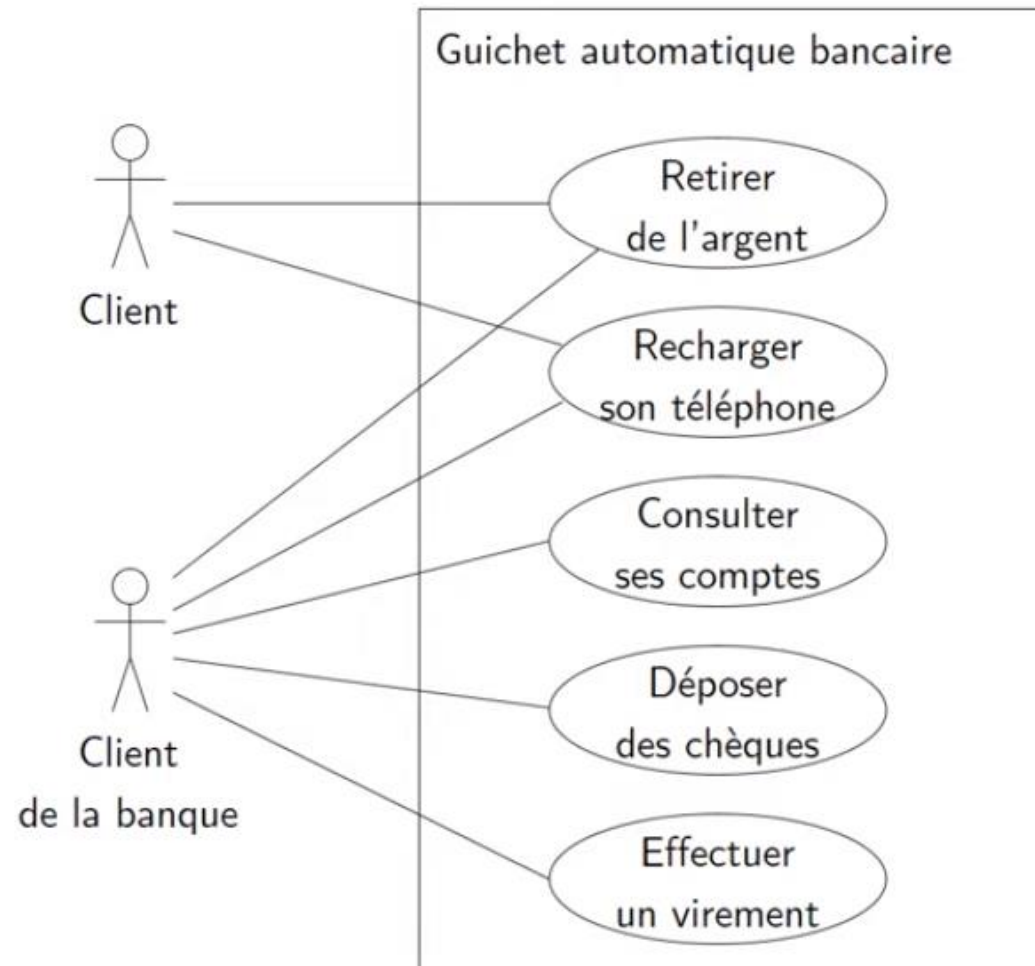
Situation : Y peut faire tout ce que fait X

Modélisation : Faire apparaître Y comme un cas particulier de X
(ou X généralisation de Y)

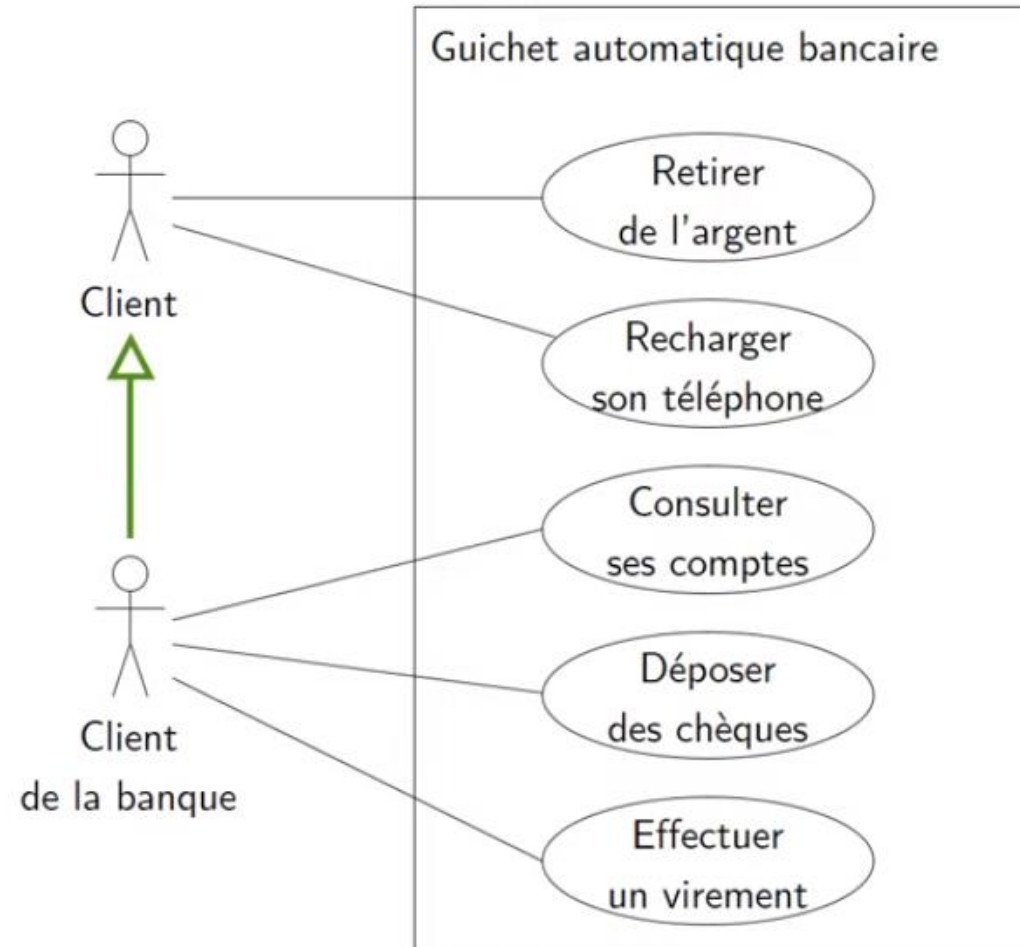
Cas d'utilisation : Exemple (1)



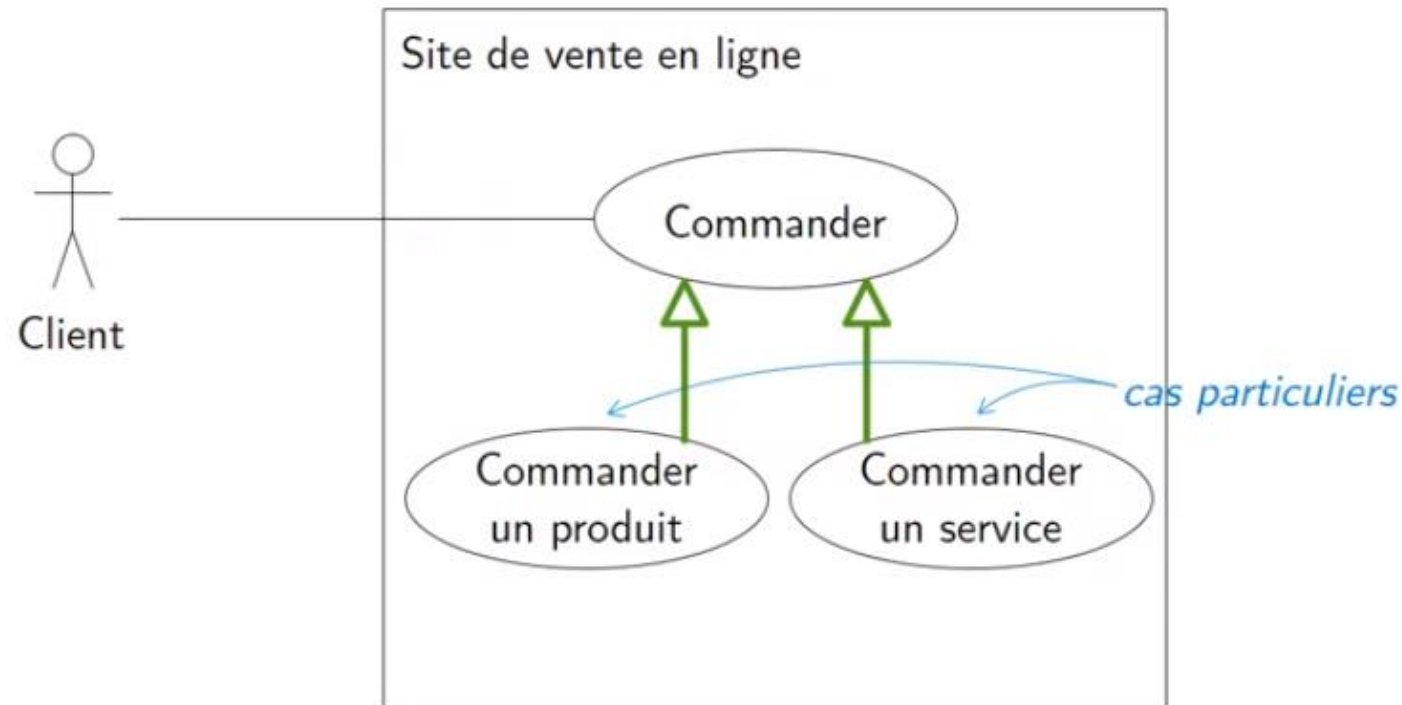
Cas d'utilisation : Exemple (2)



Cas d'utilisation : Exemple (3)

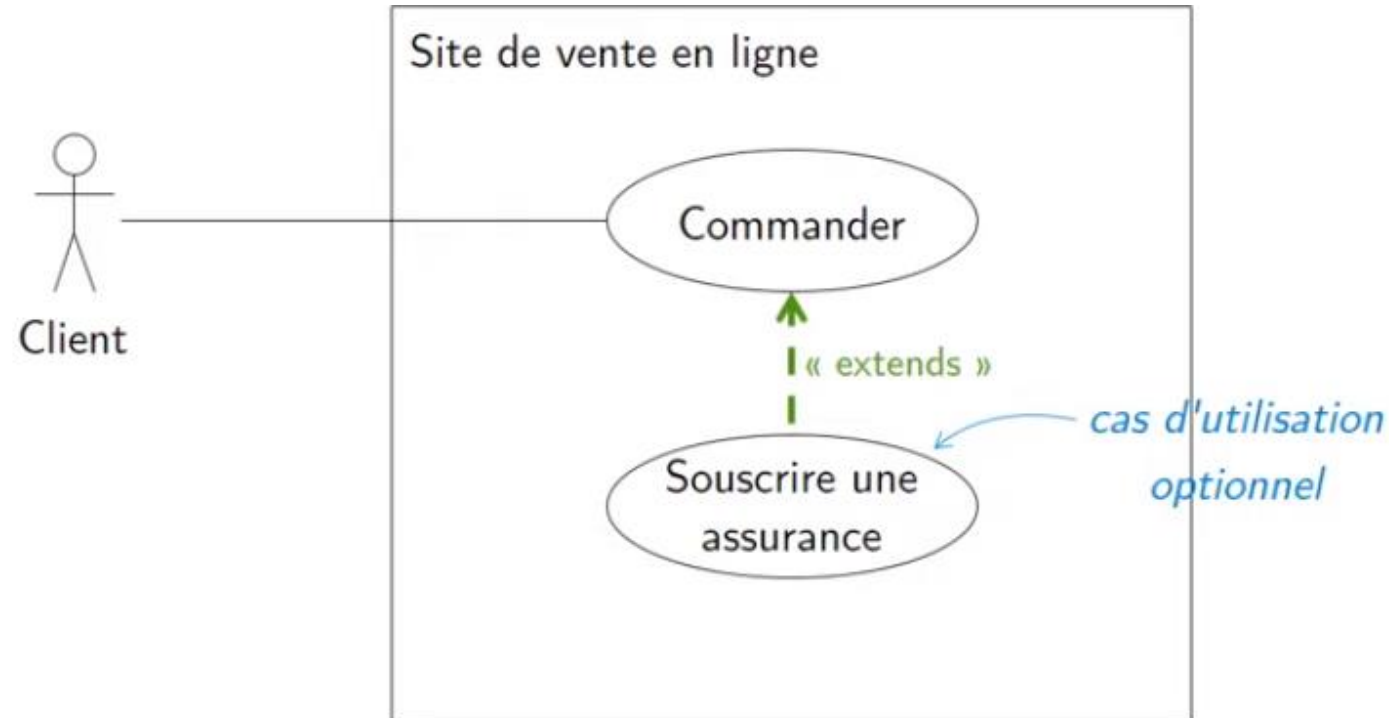


Cas d'utilisation : Relation entre cas d'utilisation (1)



Généralisation : X est un cas particulier de Y
Tout ou partie du scénario de Y est spécifique à X

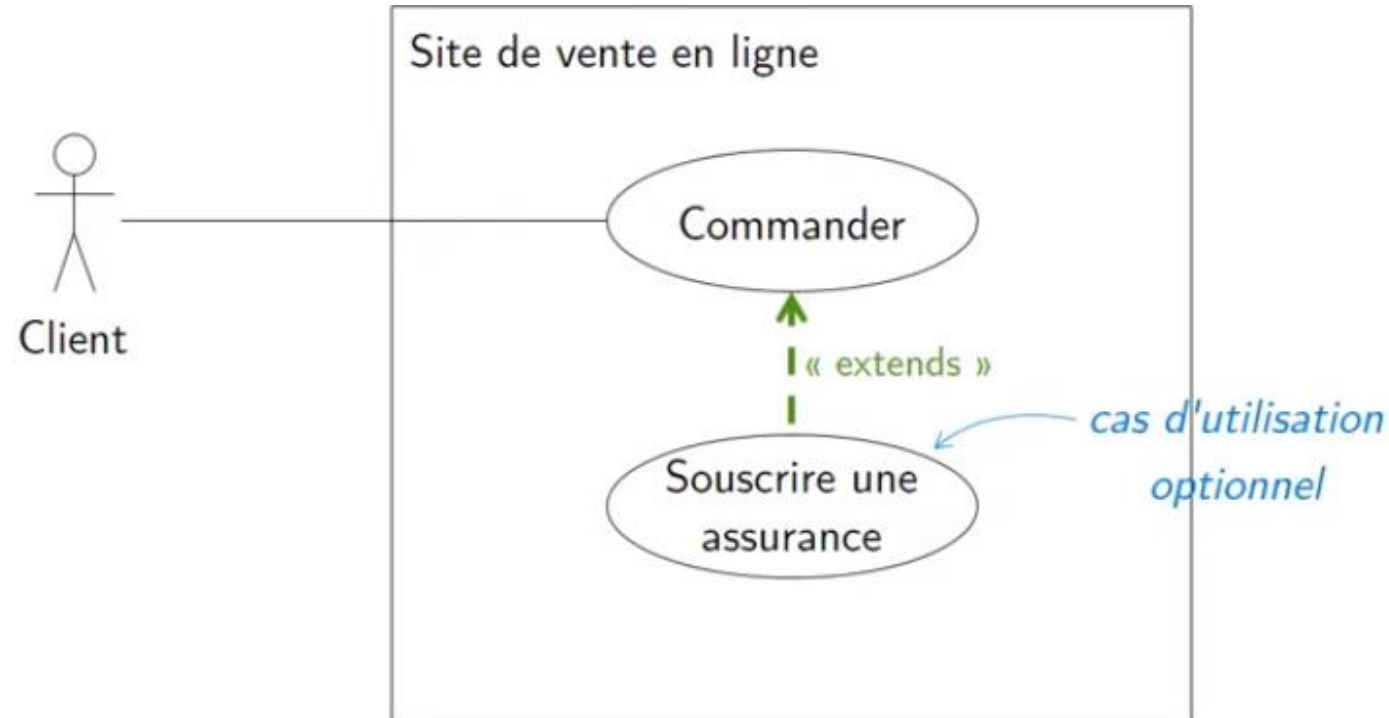
Cas d'utilisation : Relation entre cas d'utilisation (2)



Extension : X « extends » Y

- Cas d'utilisation X peut être déclenché au cours du scénario de Y
- X est optionnel pour Y

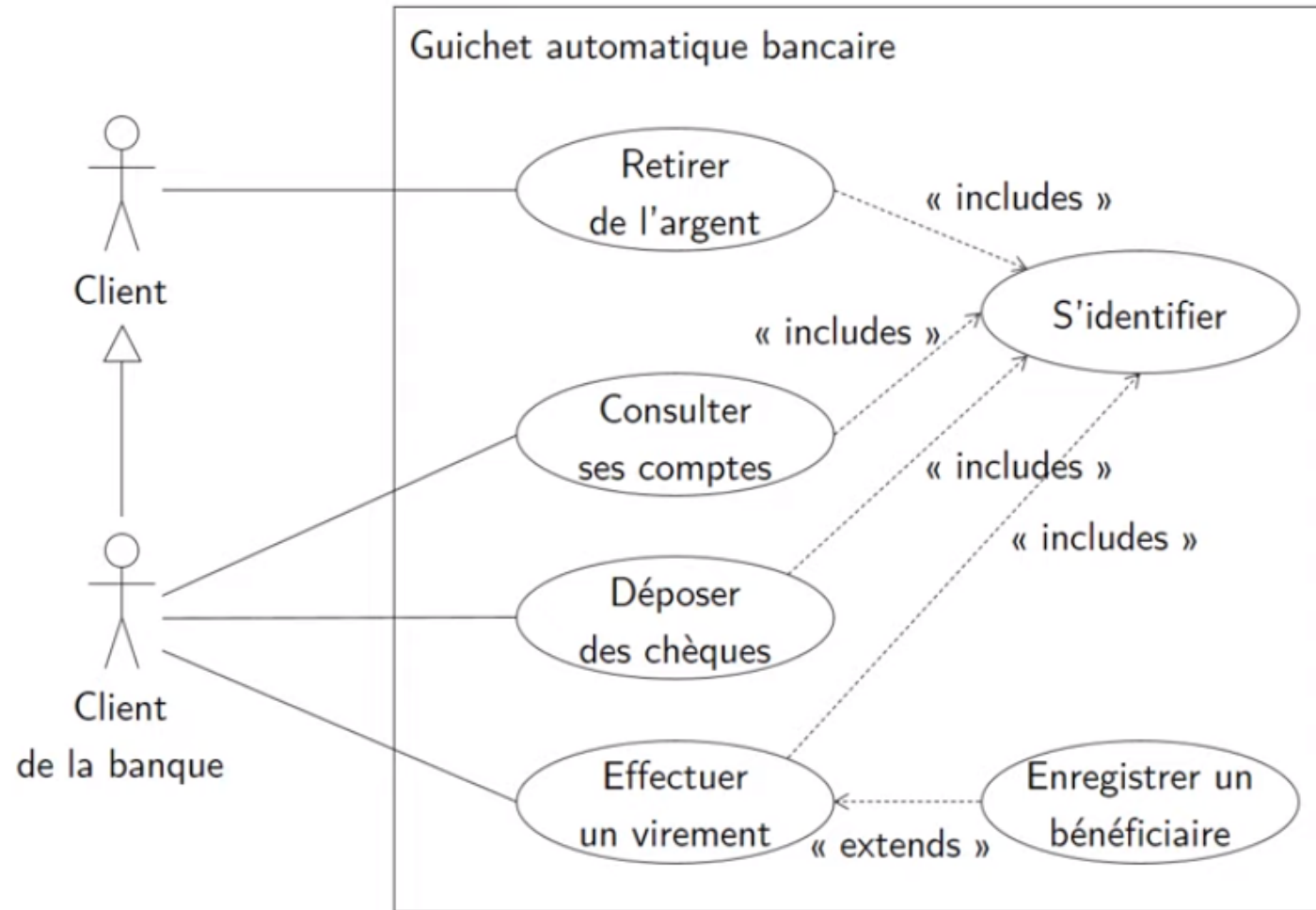
Cas d'utilisation : Relation entre cas d'utilisation (3)



Inclusion : X « includes » Y

- Scénario de Y inclus dans le scénario X
- Cas d'utilisation Y déclenché au cours du scénario de X

Cas d'utilisation : Exemple



Cas d'utilisation : conseils

Reste lisible :

- Pas plus de 6 ou 8 cas dans un diagramme
- Au besoin, faire plusieurs diagrammes (si cas disjoints entre acteurs, pour détailler un cas...)
- Relations entre cas seulement si nécessaire et pas trop lourdes

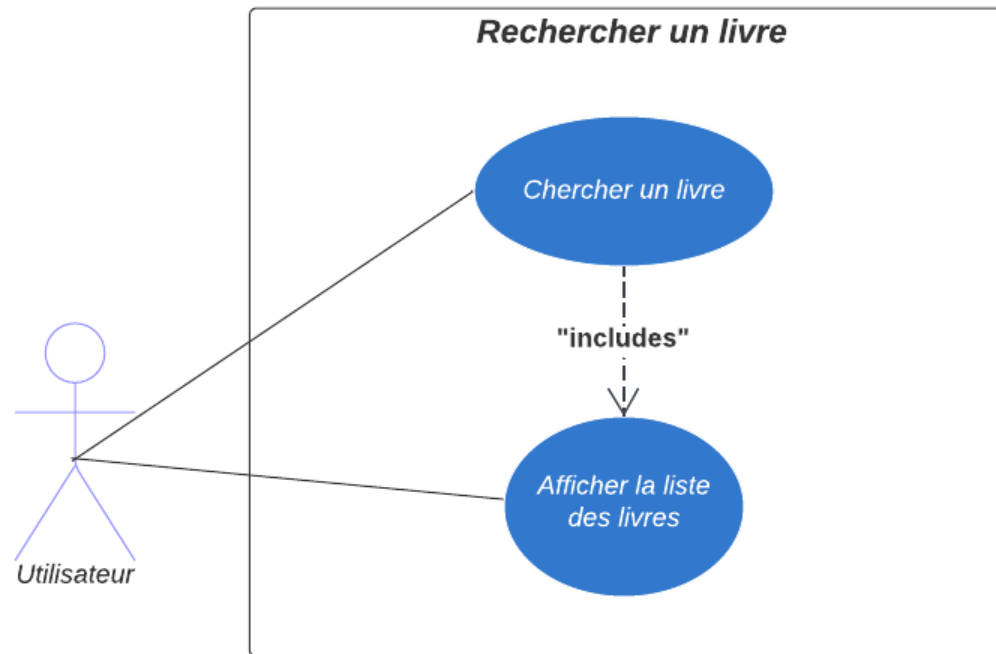
A Toi De Jouer : énoncé

Créer un diagramme de cas d'utilisation pour un utilisateur de la bibliothèque.

Trouver au moins 4 cas d'utilisation

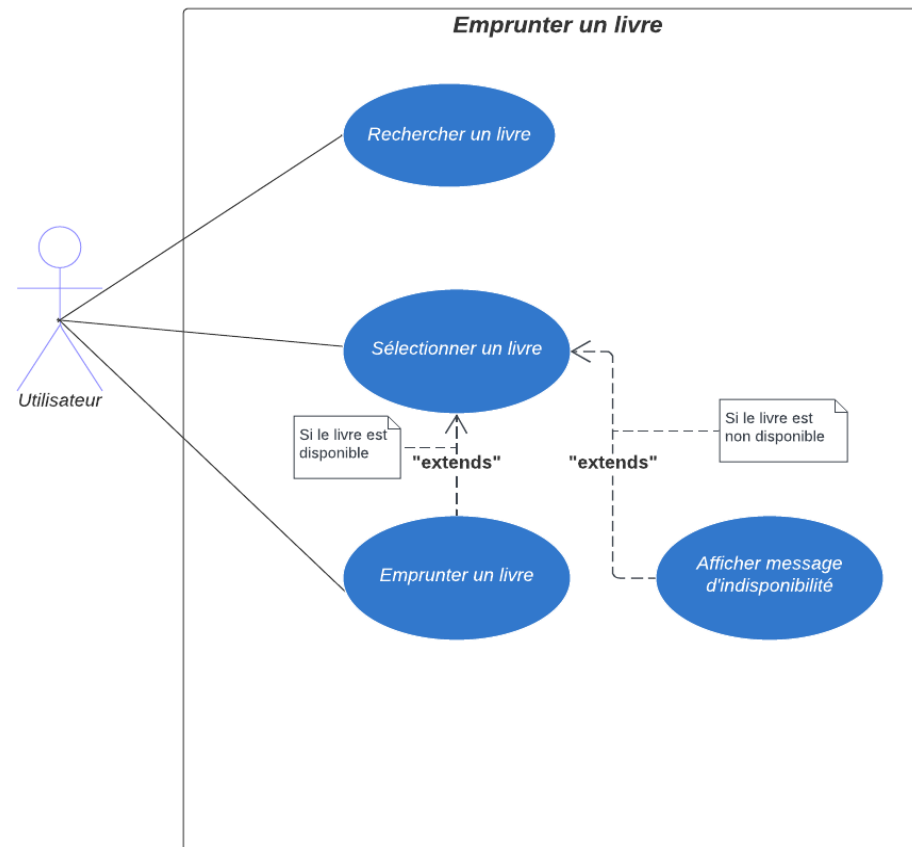
A Toi De Jouer : Recherche un livre

L'utilisateur peut rechercher des livres dans le catalogue de la bibliothèque en utilisant différents critères de recherche.



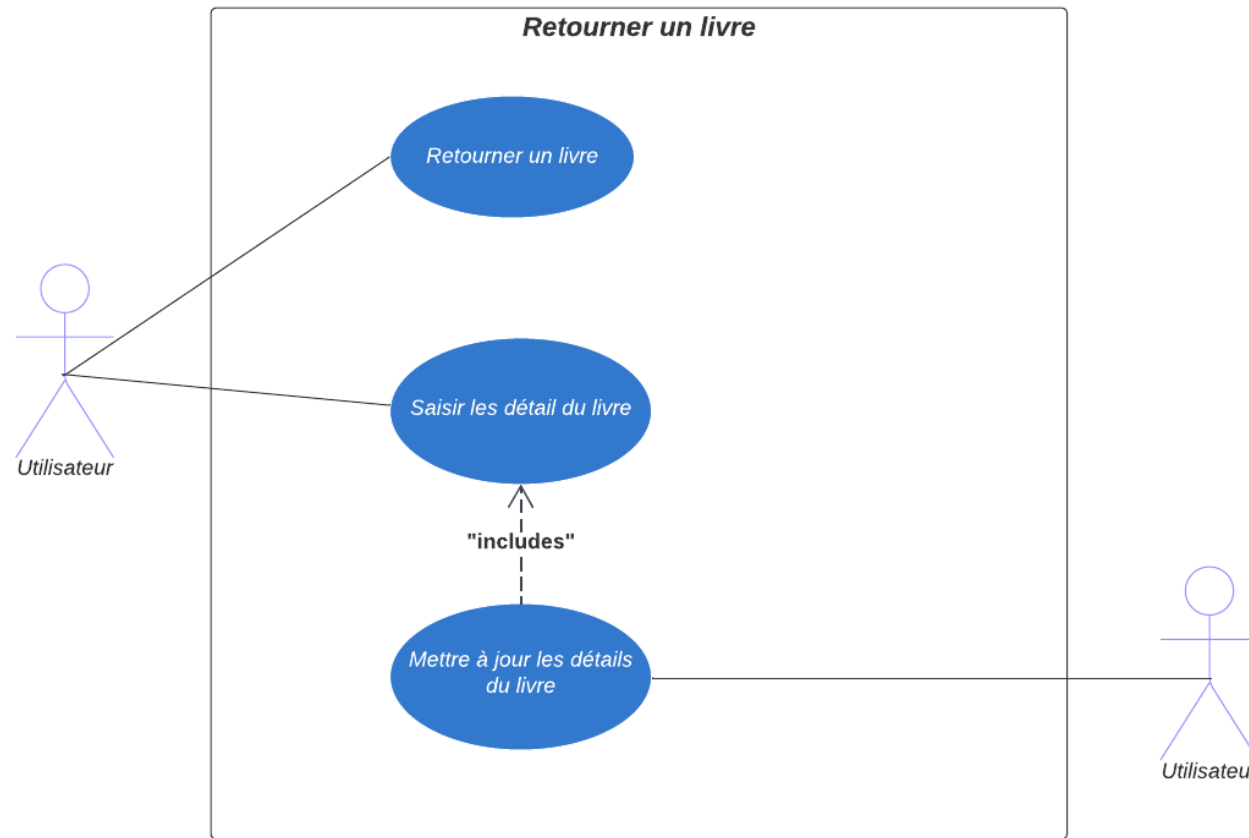
A Toi De Jouer : Emprunter un livre

L'utilisateur peut emprunter un livre disponible dans la bibliothèque.



A Toi De Jouer : Retourner un livre

L'utilisateur peut retourner un livre emprunté à la bibliothèque.



A Toi De Jouer: gérer son compte

L'utilisateur peut gérer son compte, y compris la modification des informations personnelles et la consultation des livres empruntés.

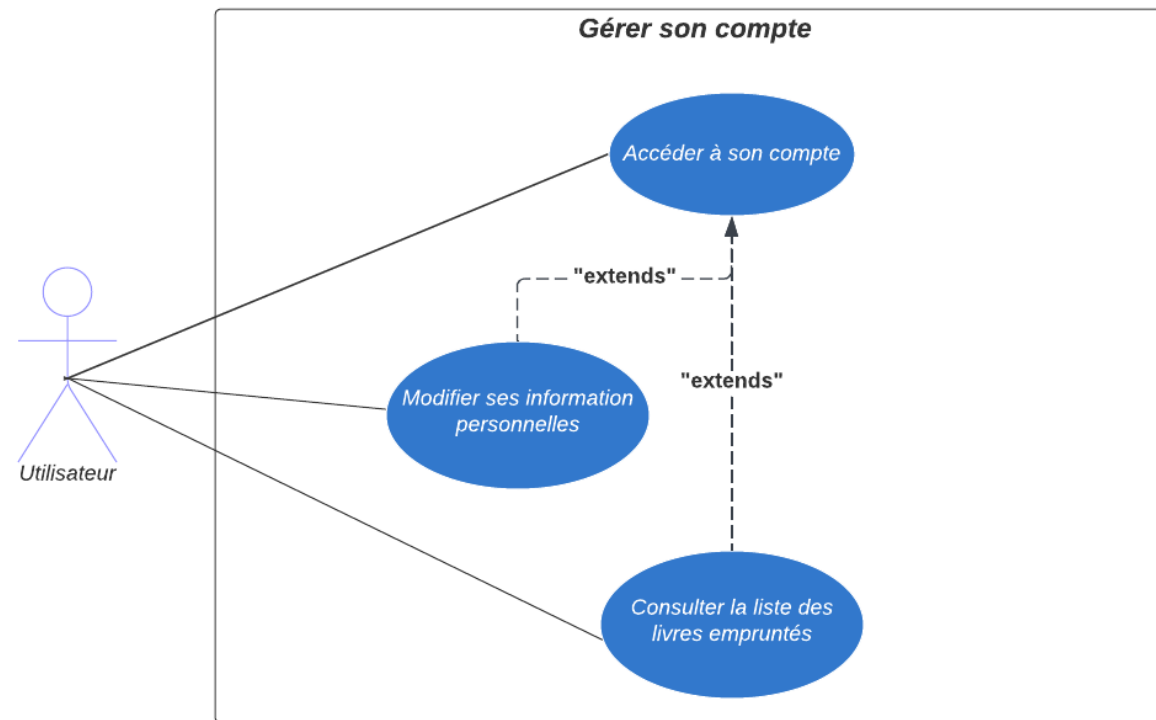


Diagramme de séquence

Séquence : Définition

Objectif : Représenter les communications avec et au sein du logiciel

- Représentation **temporelle** des interactions entre les objets
- **Chronologie** des messages échangés entre les objets et avec les acteurs

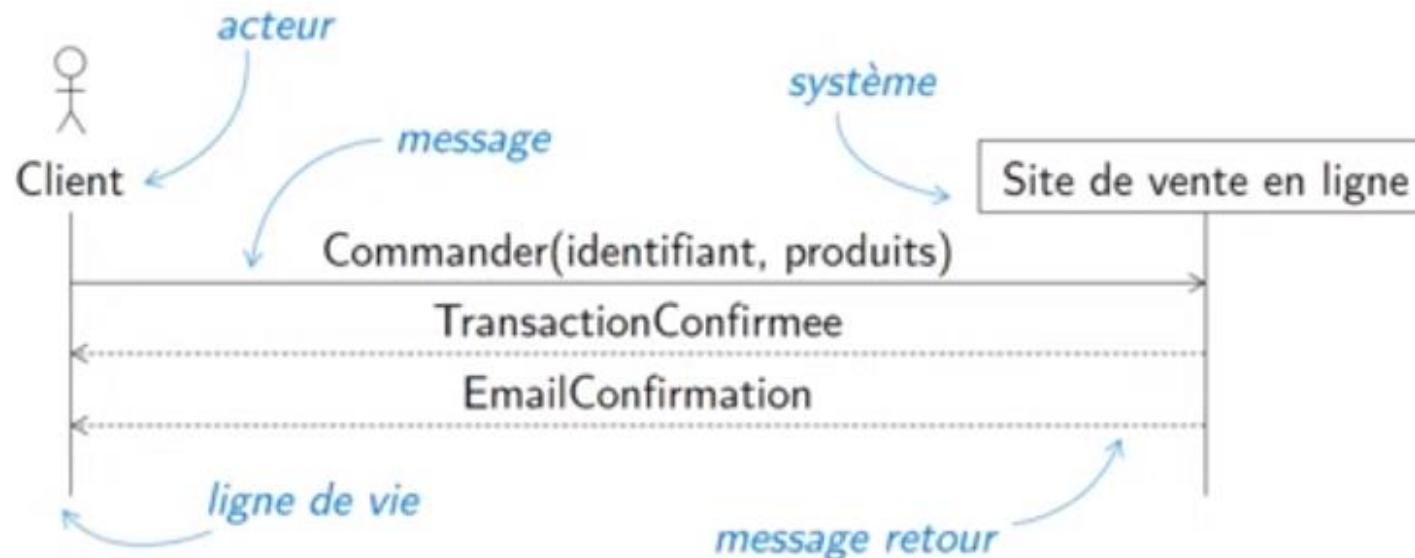
En conception : décrire la réalisation des cas d'utilisation sur le système représenté par le diagramme de classes

- Point de vue **interne** sur le fonctionnement du système
- Description au niveau de **l'instance** (état du système à un instant)
- Description de **scénarios** particuliers
- Représentation des **échanges de message** entre les acteurs et le système, entre les objets du système de façon chronologique

Séquence : Conception (1)

Éléments du digramme de séquence :

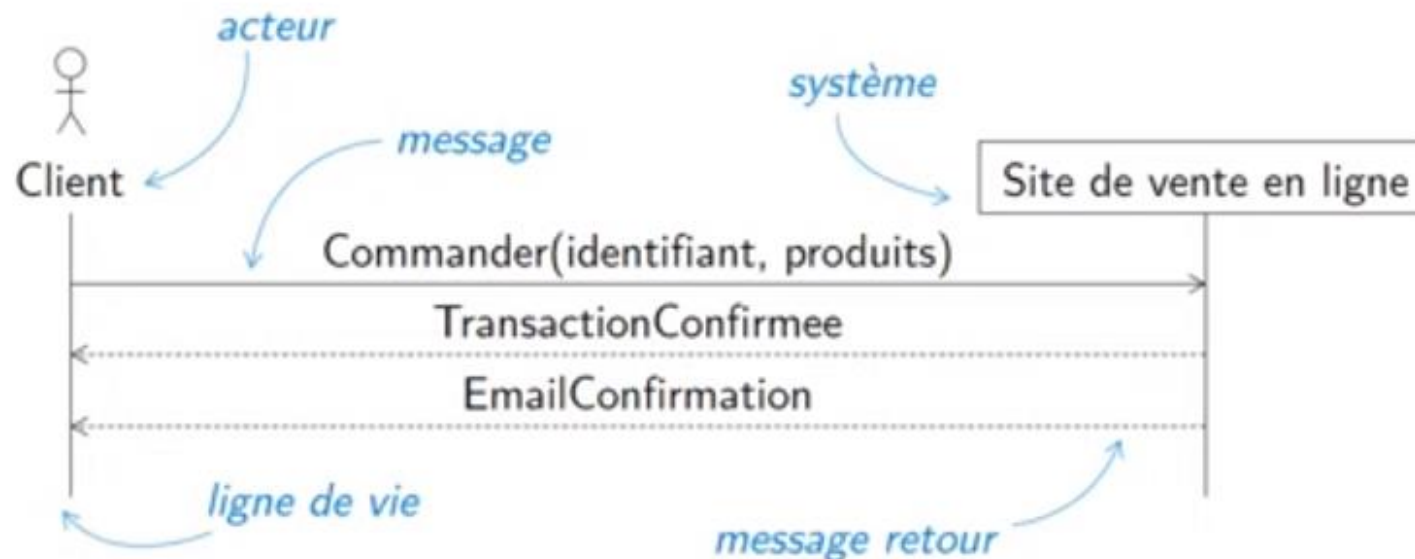
- Acteurs
- Objets (instances)
- Messages (cas d'utilisation, appel d'opération)



Séquence : Conception (2)

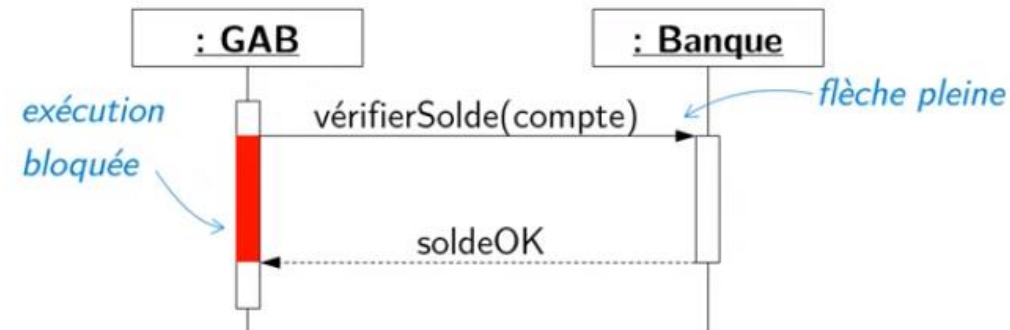
Principes de base : Représentation graphique de la **chronologie** des **échanges de messages** avec le système ou au sein du système

- « Vie » de chaque entité représentée verticalement
- Echanges de messages représentés horizontalement

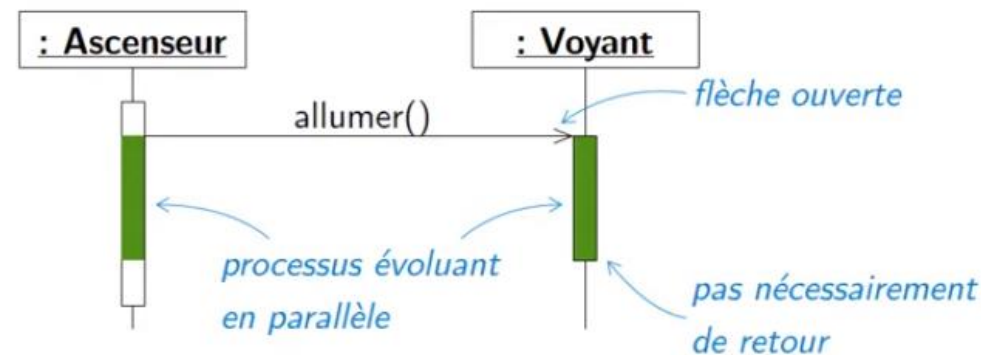


Séquence : Types de messages

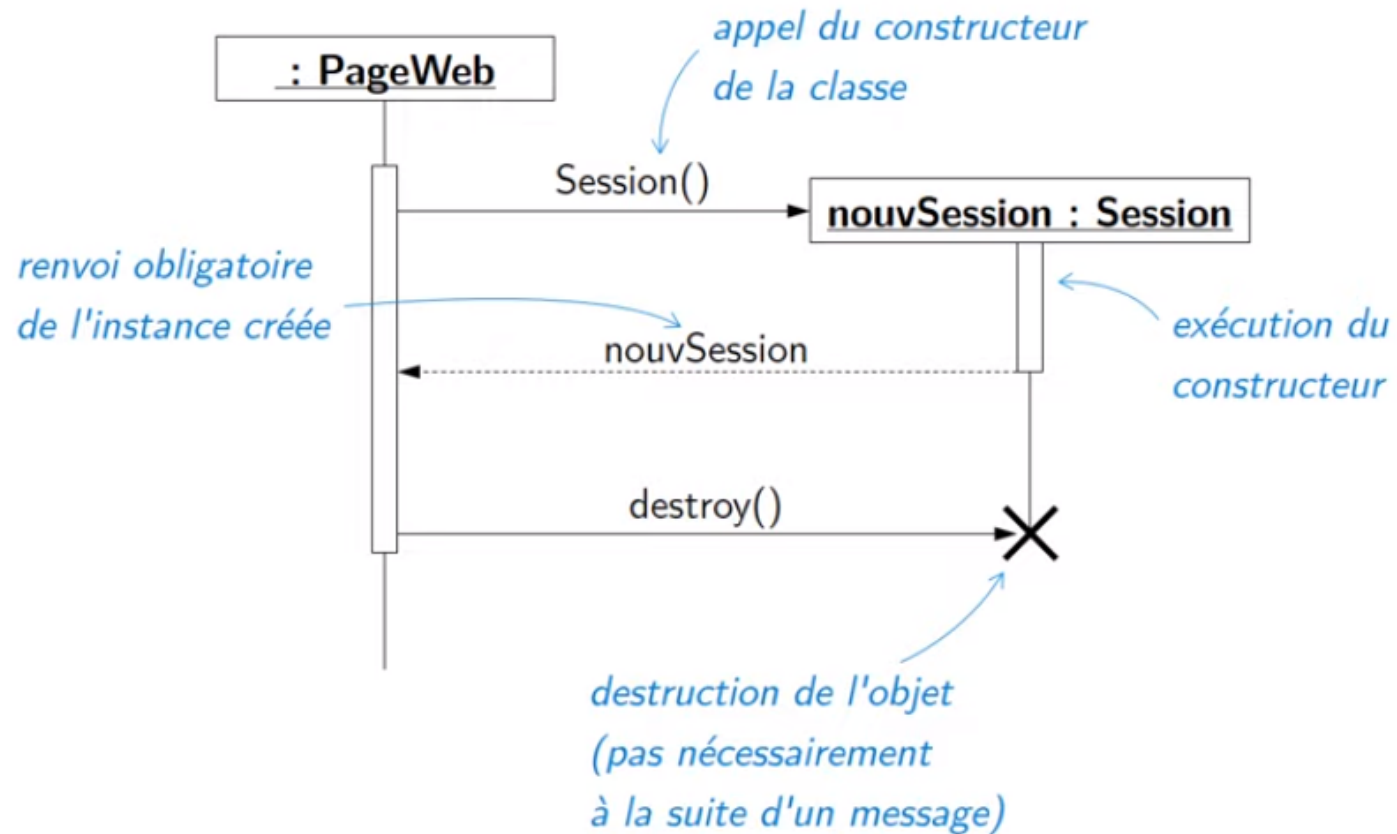
Message synchrone : émetteur **bloqué** en attente du retour



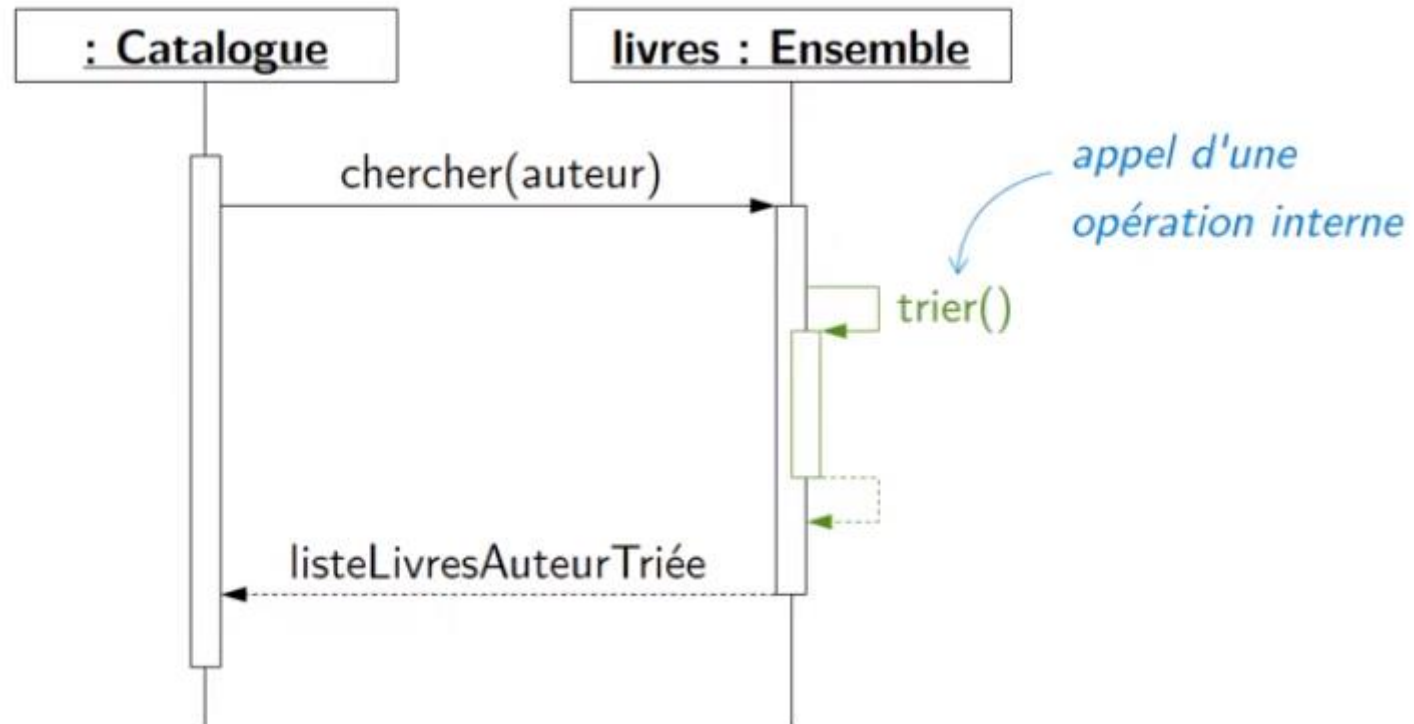
Message asynchrone : émetteur **non bloqué**, continue son exécution



Séquence: création et destruction d'objet



Séquence : message réflexif

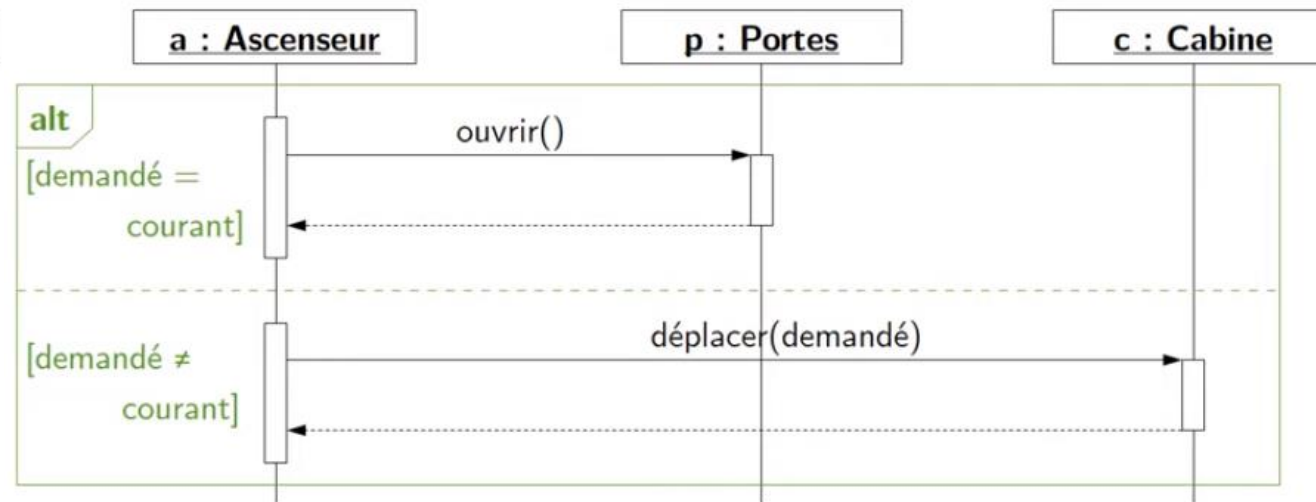
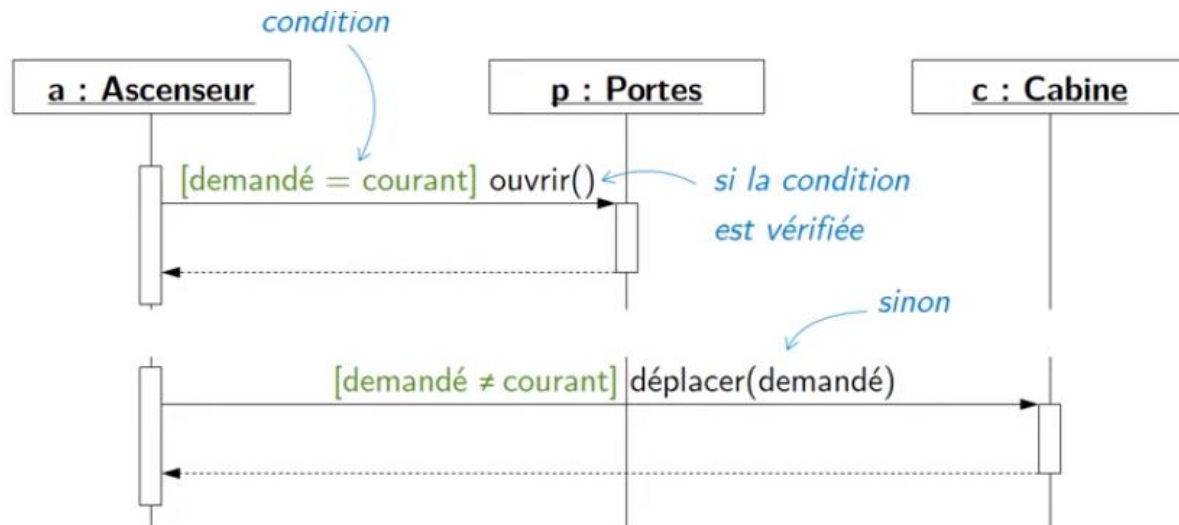


Séquence : Alternative

Principe : Condition à l'envoi d'un message

Notation :

- 2 diagrammes
- Bloc d'alternative **alt**

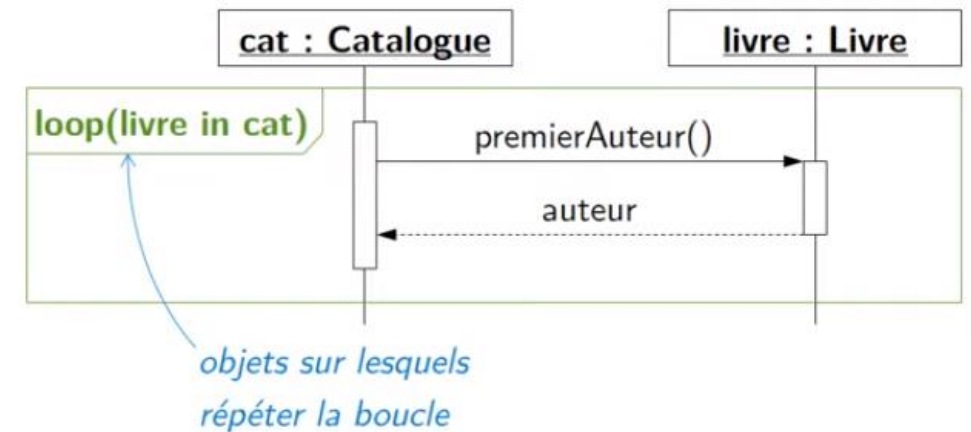
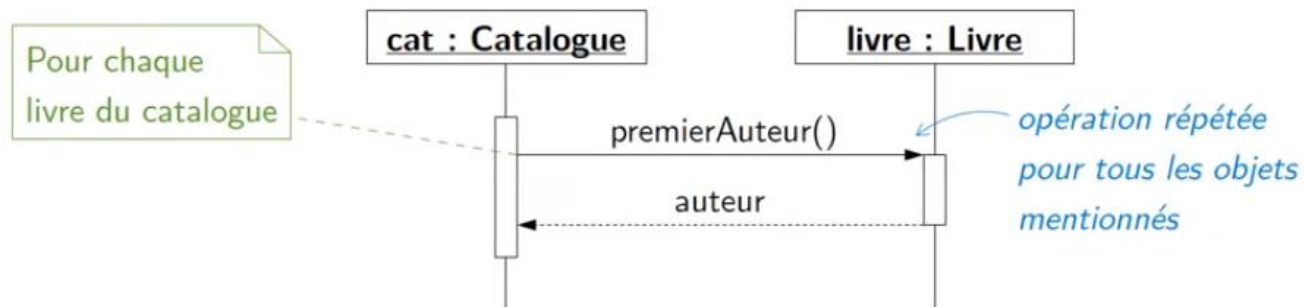


Séquence : Boucle

Principe : Répéter un enchaînement de message

Notation :

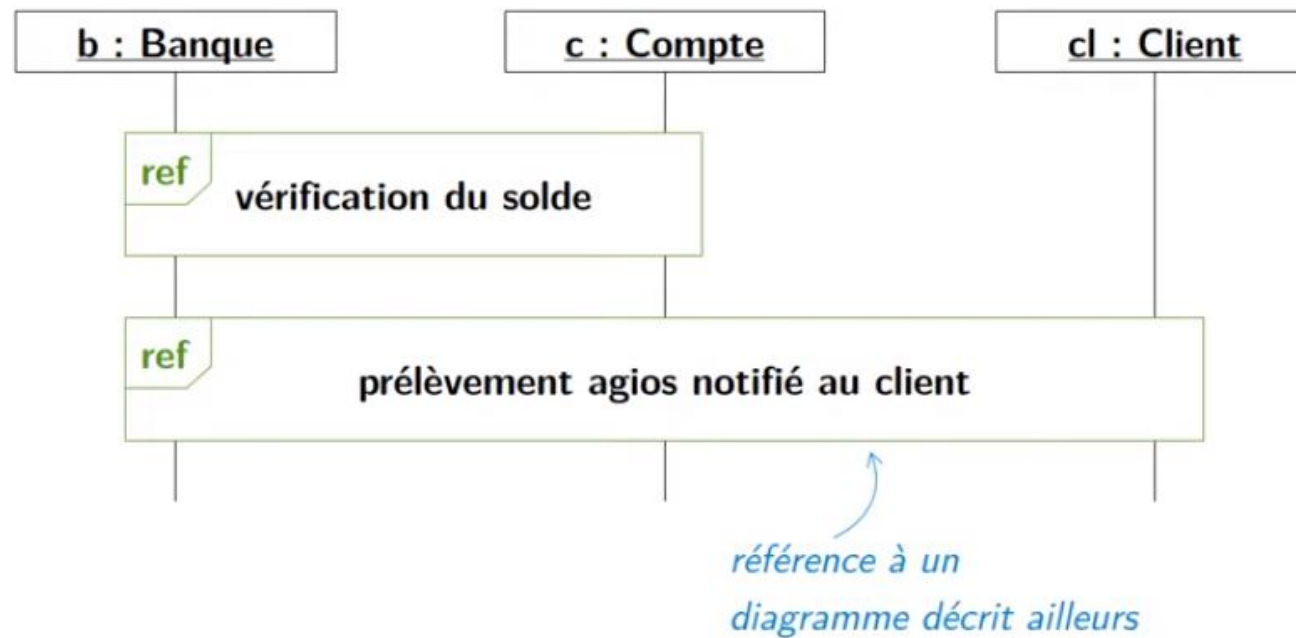
- Note
- Bloc de boucle **loop**



Séquence : Référence

Principe : Faire référence à un autre diagramme

Notation : Bloc de référence **ref**



Séquence : Quelques règles

Messages entre acteurs et interface :

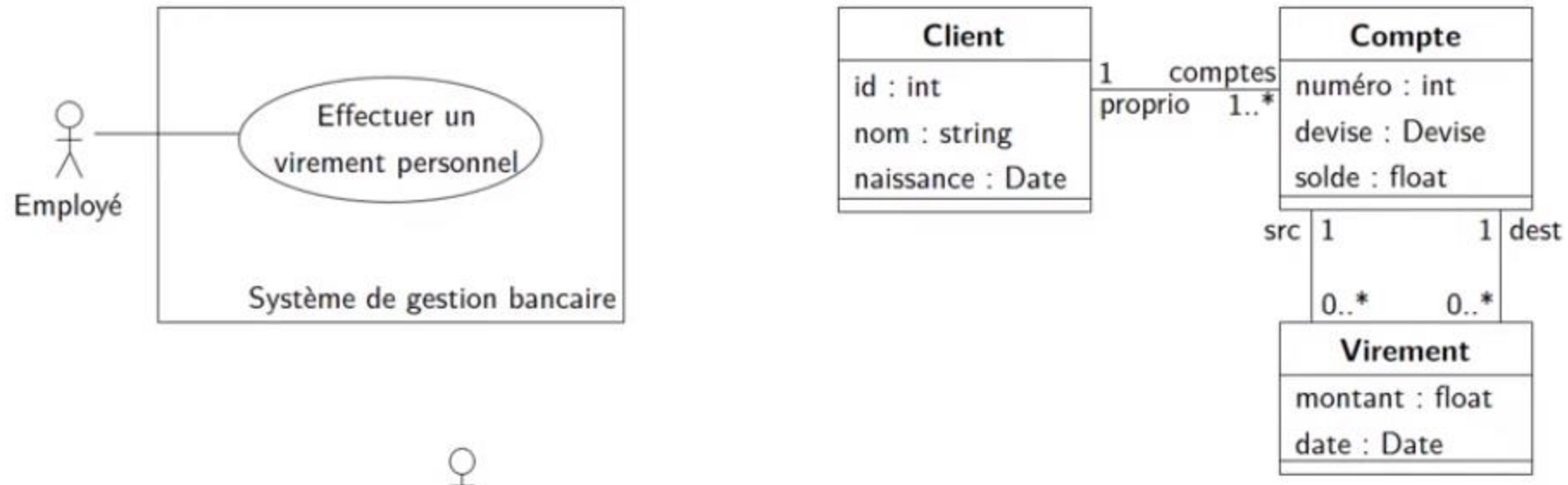
- « Fausses » opérations liés au **cas d'utilisation** (même nom)
- Arguments (saisis) et valeurs de retours (affichées) **simples** :
texte, nombre

Messages au sein du système

- **Opération** du diagramme de classes
- Si message de **objA:ClasseA** vers **objB:ClasseB**, alors opération du
messag dabs ClasseB

Séquence: A Toi De Jouer

A partir de ce cas d'utilisation et de son diagramme de classe, réaliser le diagramme de séquence



Séquence : A Toi De Jouer

Correction

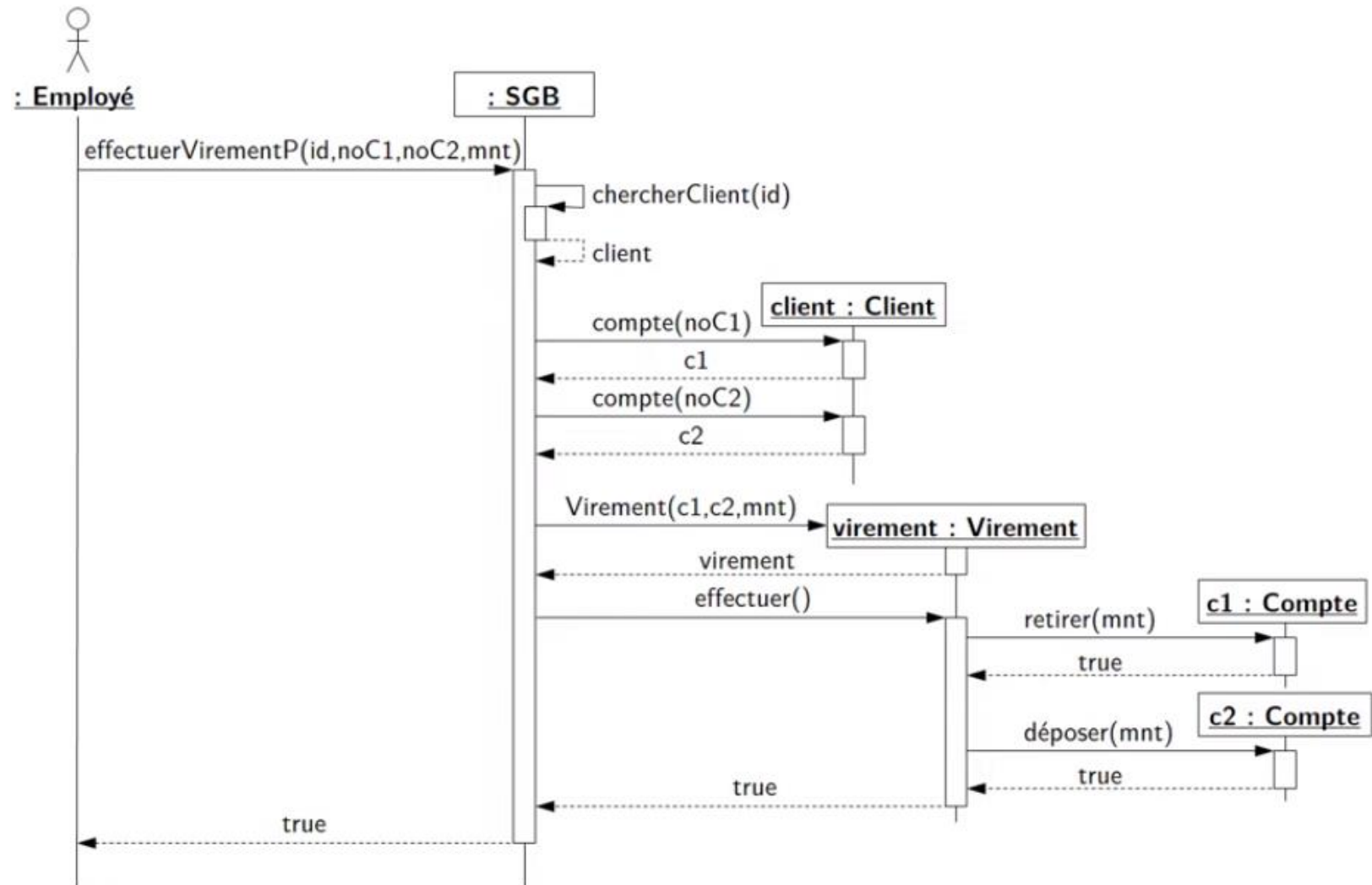


Diagramme de classes

Classe : Définition

Conception orientée objet : Représentation du système comme un ensemble d'objets interagissant

Diagramme de classes

- Représentation de la **structure interne** du logiciel
- Utilisé surtout en conception mais peut être utilisé en analyse

Diagramme d'objets

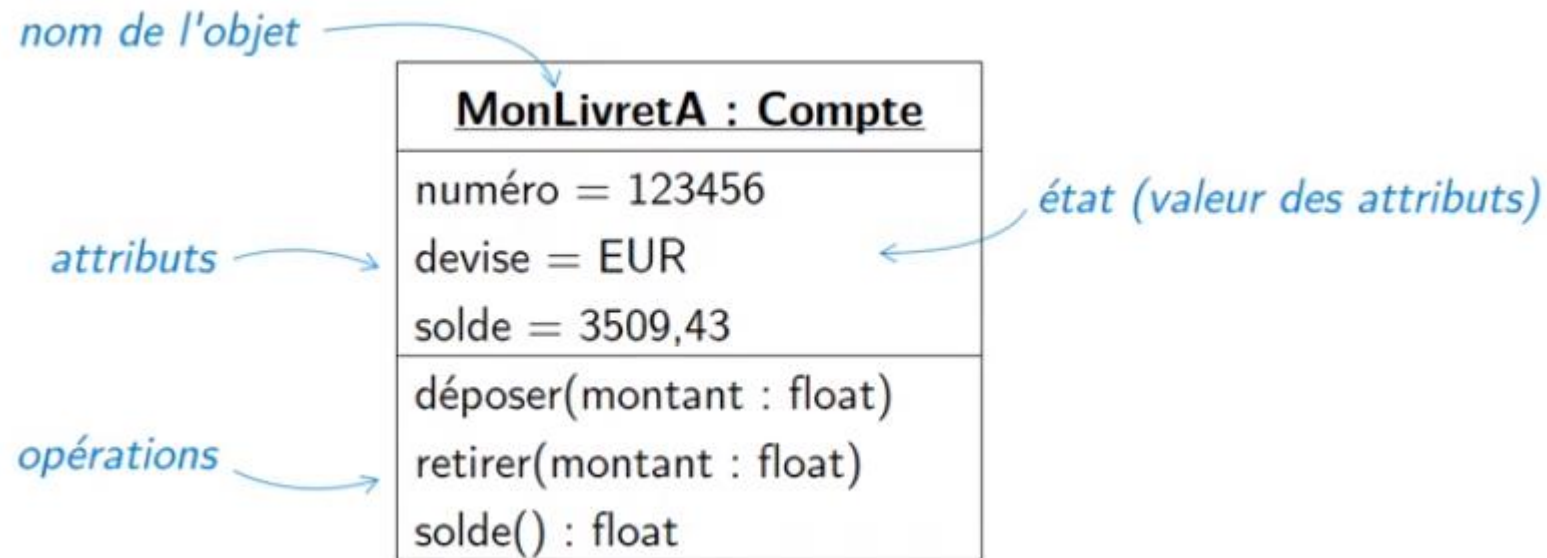
- Représentation de l'état du logiciel (objet + relations)
- Diagramme évoluant avec l'exécution du logiciel

Classe : Objet et classes

Objet

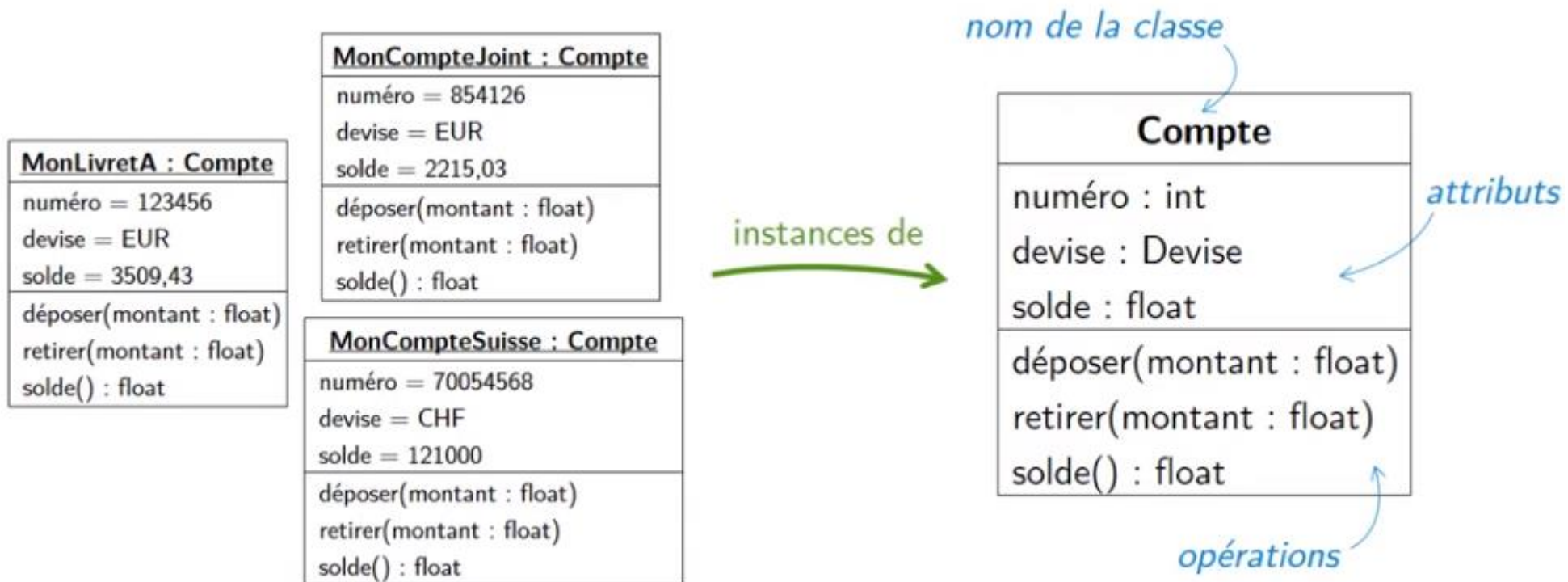
Entité concrète ou abstraite du domaine d'application

Décrit par : identité + état + comportement



Classe : Objet et classes

Classe : Regroupement d'objets de la même nature (même attributs + mêmes opérations). **Un objet est une instance d'une classe.**



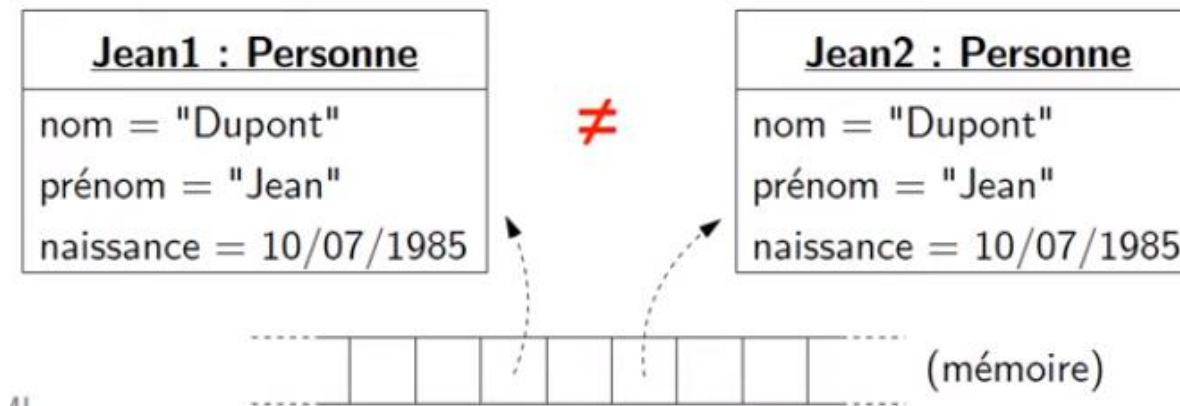
Classe : Déclaration (1)

Attributs

- **Caractéristiques partagées** par tous les objets de la classe
- Associe à chaque objet une **valeur**
- **Type associé simple** (int, bool...), primitif (Date) ou énuméré

Valeur des attributs : Etat de l'objet

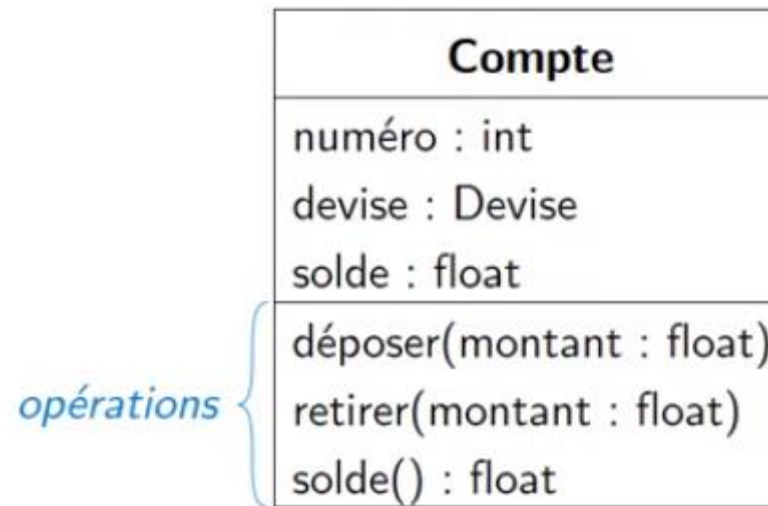
Objets différents (identités différentes) peuvent avoir les mêmes attributs.



Classe : Déclaration (2)

Opérations

- **Service** qui peut être demandé à tout objet de la classe
- **Comportement commun** à tous les objets de la classe

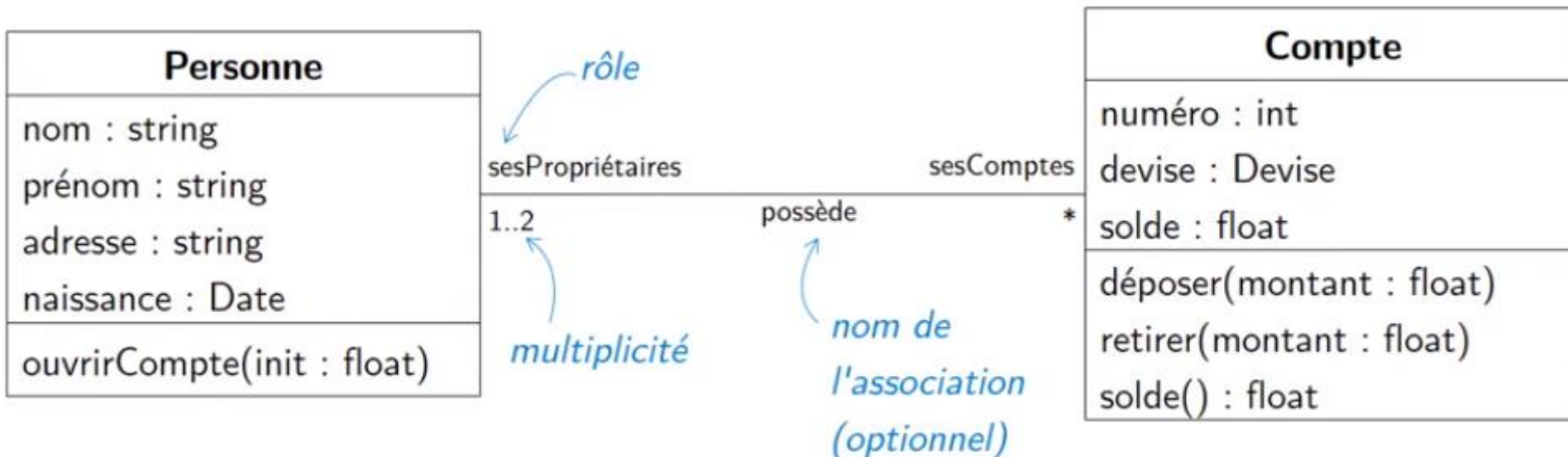


Classe : Relations

Association entre classes : Relation binaire (en général)

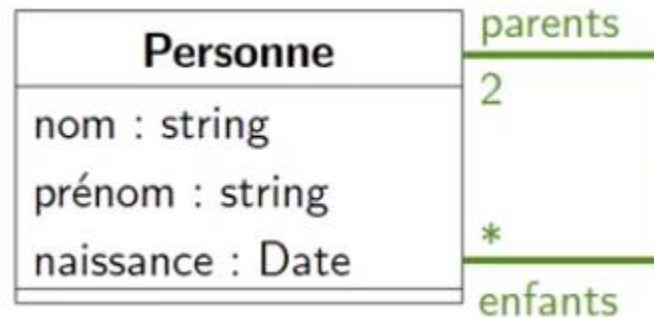
Rôle : Nomme l'extrémité d'une association, permet **d'accéder aux objets liés** par l'association à un objet donné

Multiplicité : Contraint le **nombre d'objets liés** par l'association



Classe : Associations particulières (1)

Association réflexive : Association d'une classe vers elle-même

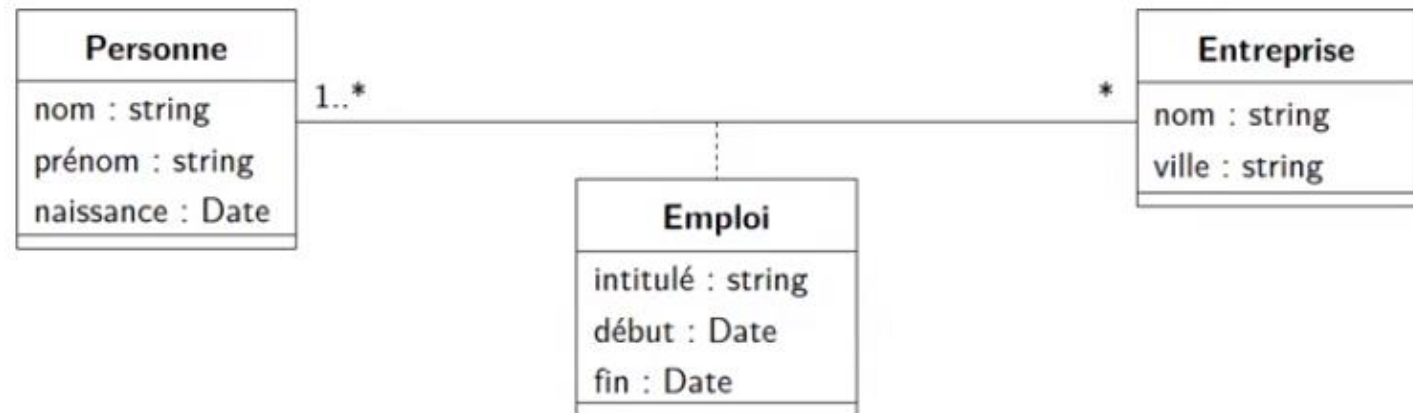


Association multiple : plusieurs associations entre 2 classes

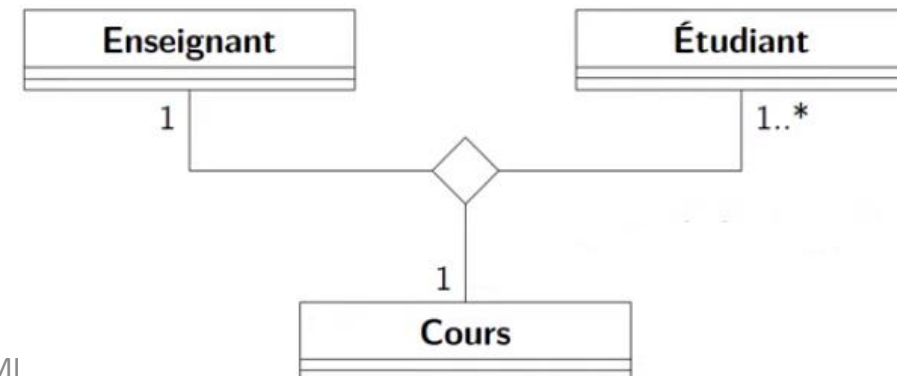


Classe : Associations particulières (2)

Classe-association : permet de paramétrer une association entre 2 classes par une classe



Association n-air : association reliant plus de 2 classes



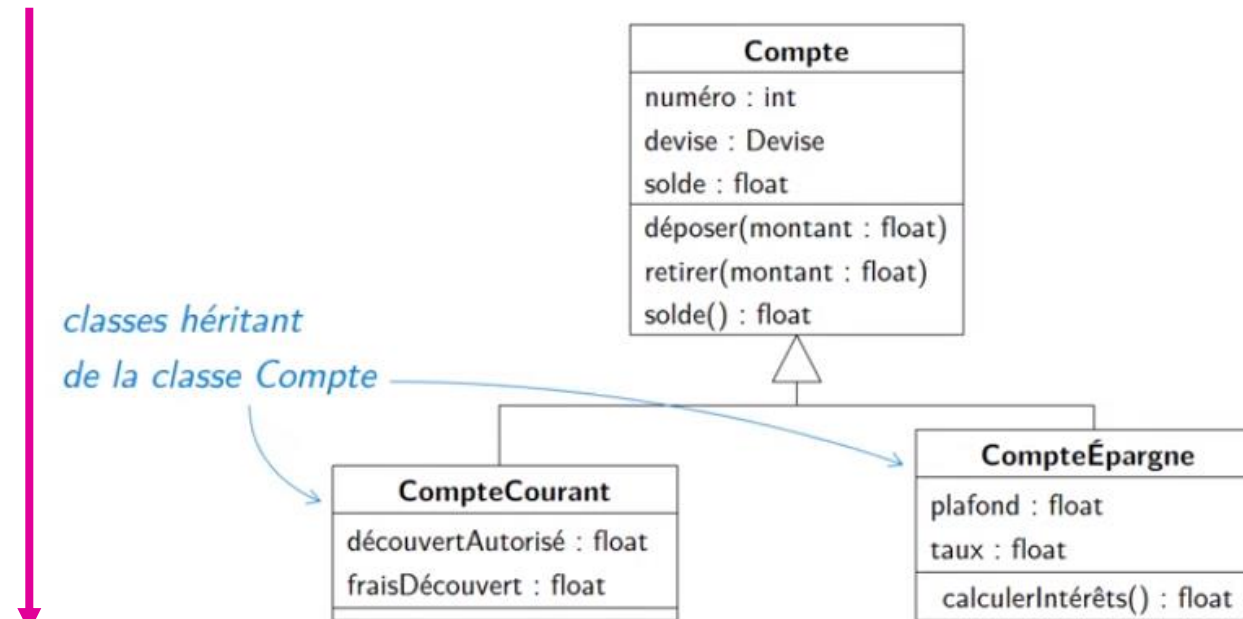
Classe : Hiérarchie de classe

Principe : Regrouper les classes partageant des attributs et des opérations et les organiser en arborescence

Spécialisation : raffinement d'une classe en une **sous-classe**

Généralisation : **abstraction** d'un ensemble de classes en **super-classe**

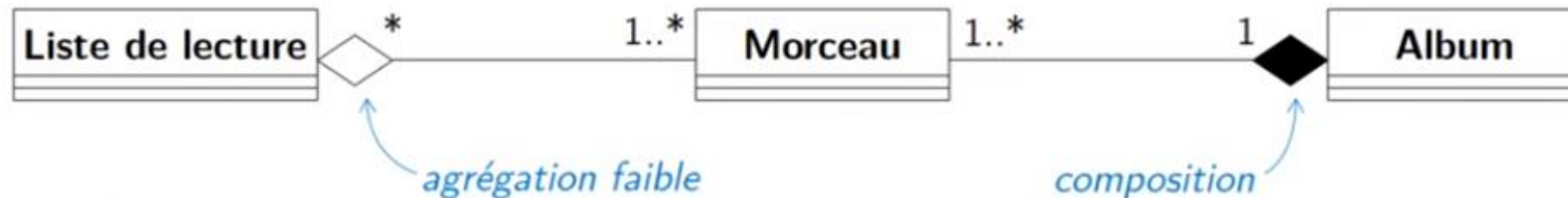
Spécialisation



Généralisation

Classe : Agrégation faible et Composition

- L'agrégation faible fait référence à ses composantes : composite
- La création ou destruction du composite est indépendant de la création ou destruction de ses composants
- Un objet peut faire partie de plusieurs composites à la fois
- Le composite contient ses composants
- La création ou destruction du composite entraîne la création ou destruction de ses composants
- Un objet ne fait partie que d'un composite à la fois



Classe : Contraintes

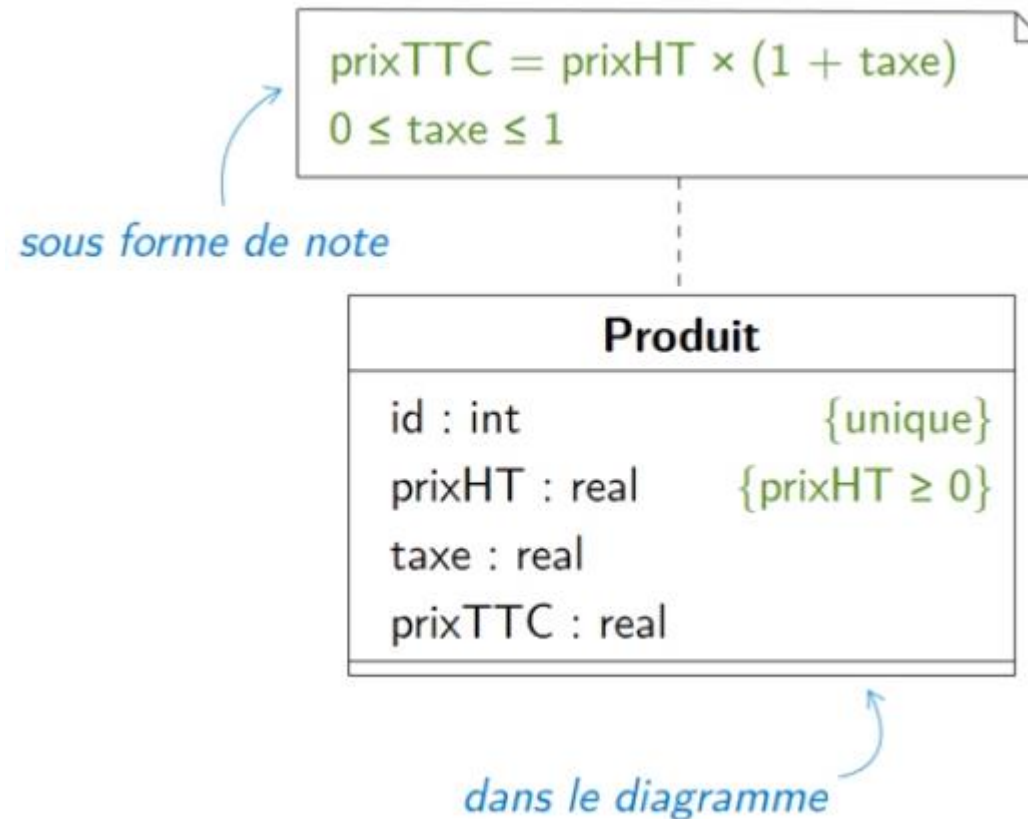
Objectif : contraindre l'implémentation pour qu'elle réponde au cahier des charges

Types de Contraintes :

- Contraintes sur les attributs
- Contraintes sur les associations
- Contraintes sur les multiplicités

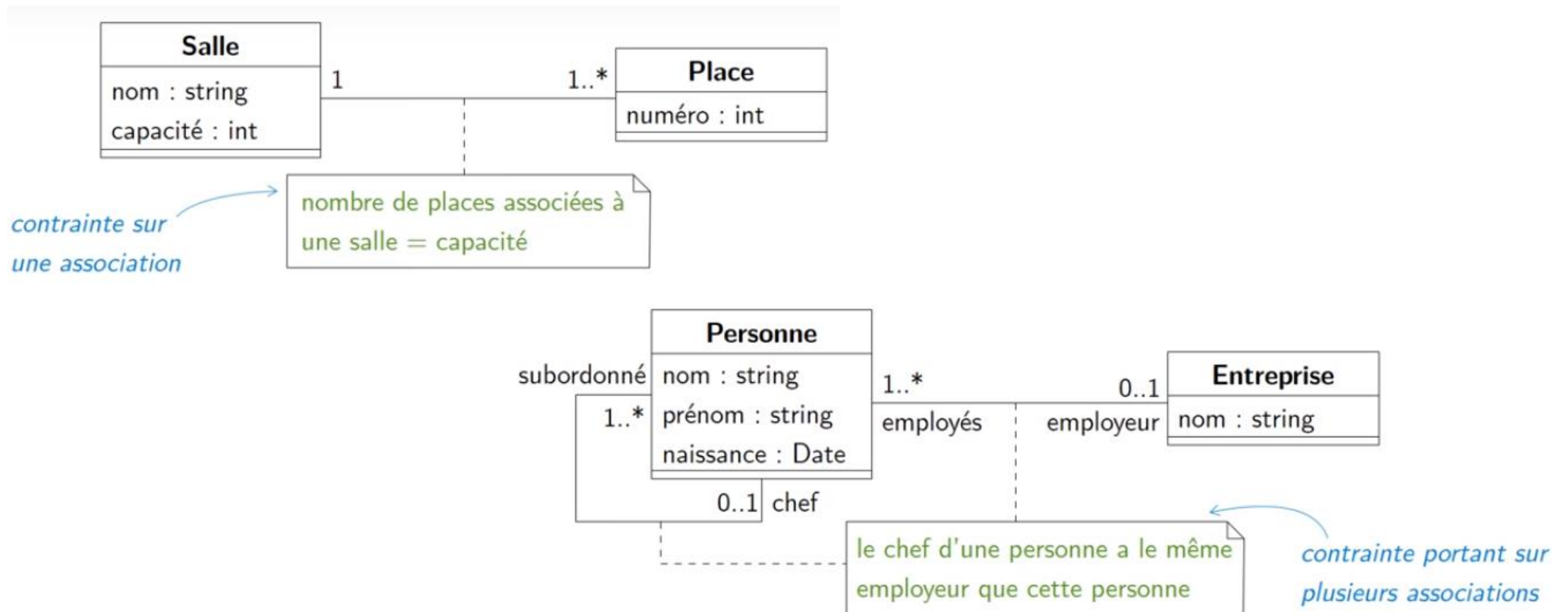
Classe : Contrainte sur attribut

Elles peuvent être déclarées directement sur les attributs entre accolades ou sous la forme de note



Classe : Contrainte sur association

Elles sont déclarées sous forme de note sur une ou plusieurs associations



Classe : Contrainte sur multiplicité

Elles sont déclarées directement sur la multiplicité concernée entre accolades

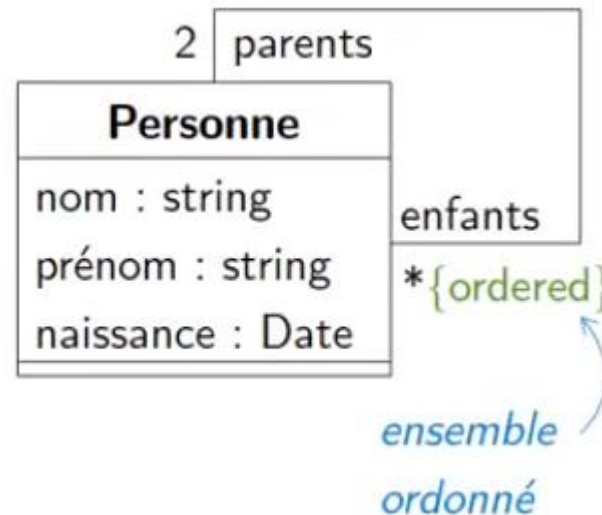


Diagramme de déploiement

Diagramme de déploiement

Le diagramme de déploiement fait partie des **diagrammes structuraux** (statique), il représente :

- **La disposition physique des ressources matérielles** qui constituent le système et montre la répartition des composants (élément logiciels) sur ces matériels
- **La nature des connexions de communication** entre les différentes ressources matérielles

Eléments du diagramme de déploiement

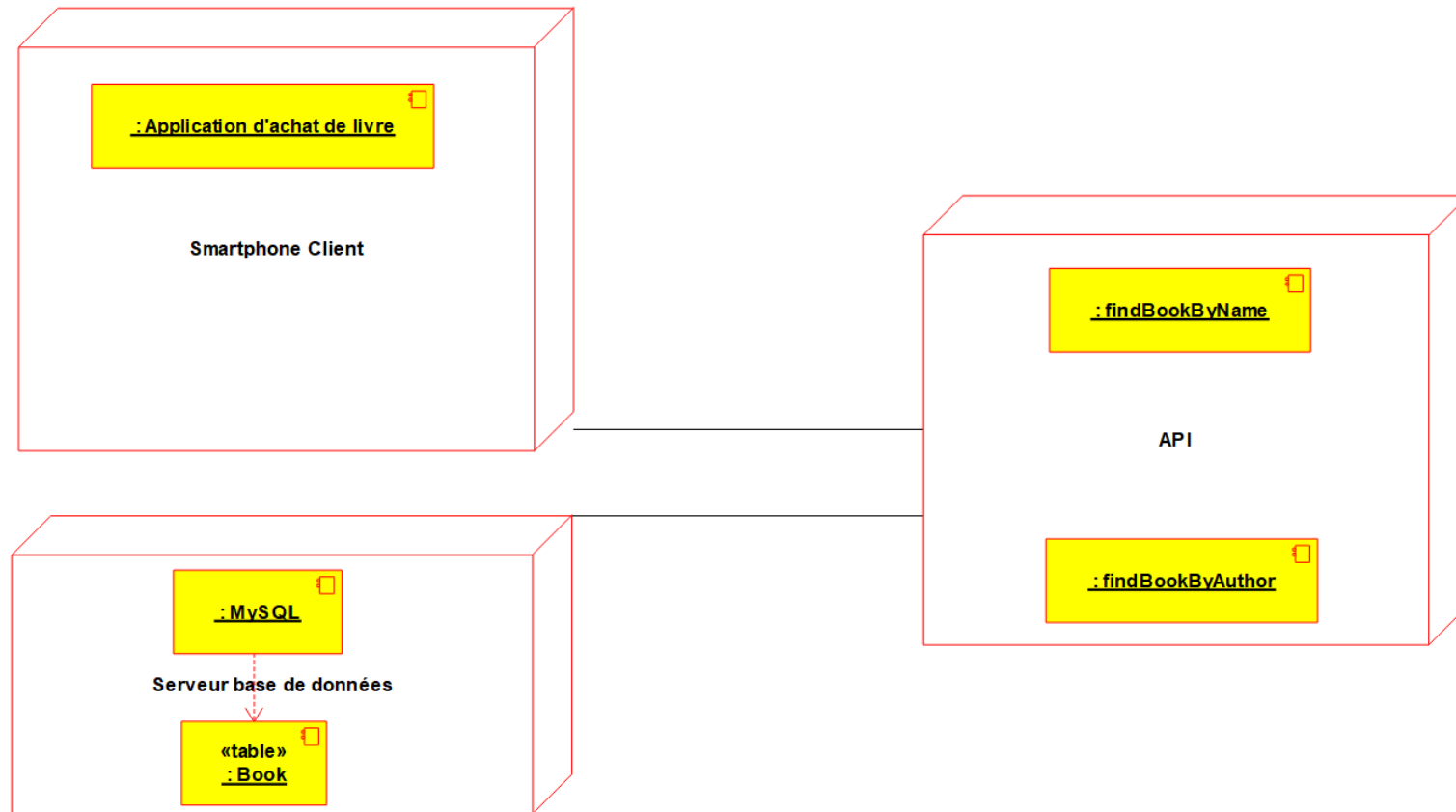
Les nœuds : un nœud est une ressource matérielle du système. En général, cette ressource possède au minimum de la mémoire et parfois aussi des capacités de calcul

- Un nœud est représenté par un parallélépipède rectangle dans lequel figure son nom
- Un nœud possède des attributs
- Pour montrer qu'un composant est affecté sur un nœud, il faut placer le composant dans le nœud ou en le reliant à l'aide d'une relation de dépendance (flèche en pointillées)

Les chemins de communications : les différents nœuds qui apparaissent dans le diagramme de déploiement sont connectés entre eux par des lignes qui symbolisent un support de communication.

Les artefacts : l'artefact est le terme générique qui désigne n'importe quel élément produit du travail, c'est un élément concret et existant dans le monde réel (document, exécutable, fichier, base de donnée...).

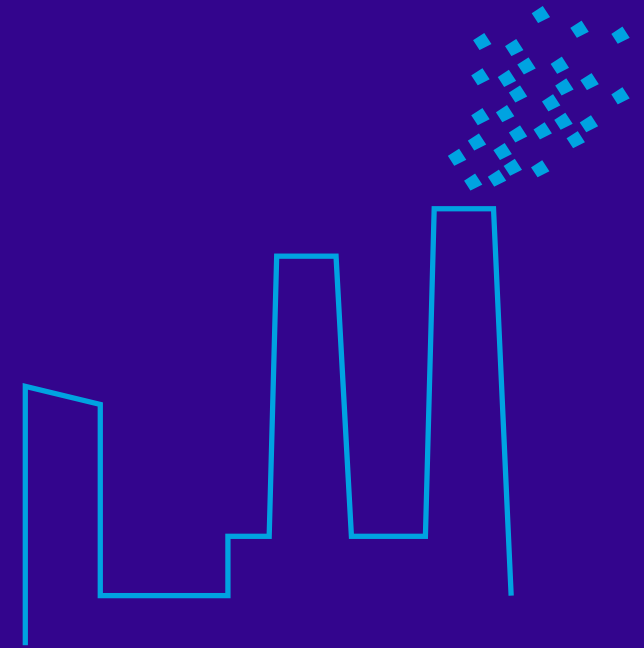
Exemple d'un diagramme de déploiement



Rendez-vous sur lamanu.fr

contact@lamanu.fr

09 86 27 17 04



Campus Versailles

contact-versailles@lamanu.fr

<https://lamanu.fr>

143 rue Yves le Coz, 78000 Versailles

09 86 27 17 04



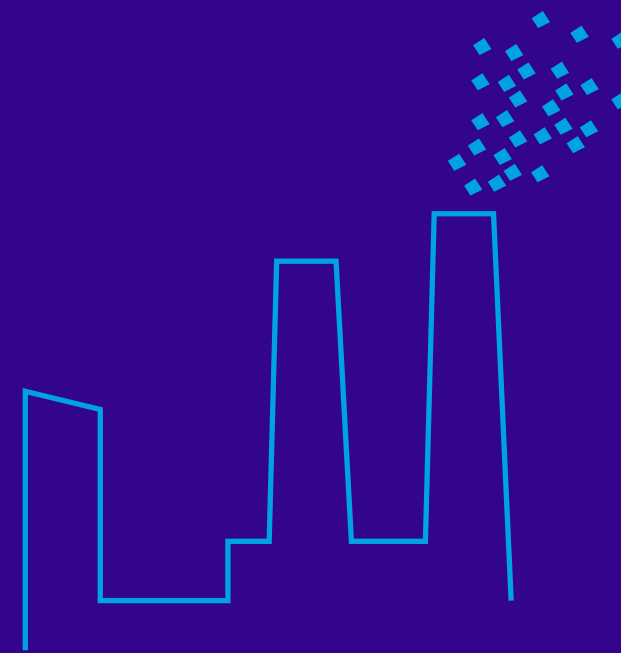
Campus Le Havre

contact-lehavre@lamanu.fr

<https://lamanu.fr>

10 place Léon Meyer, 76600 Le Havre

09 86 27 17 04



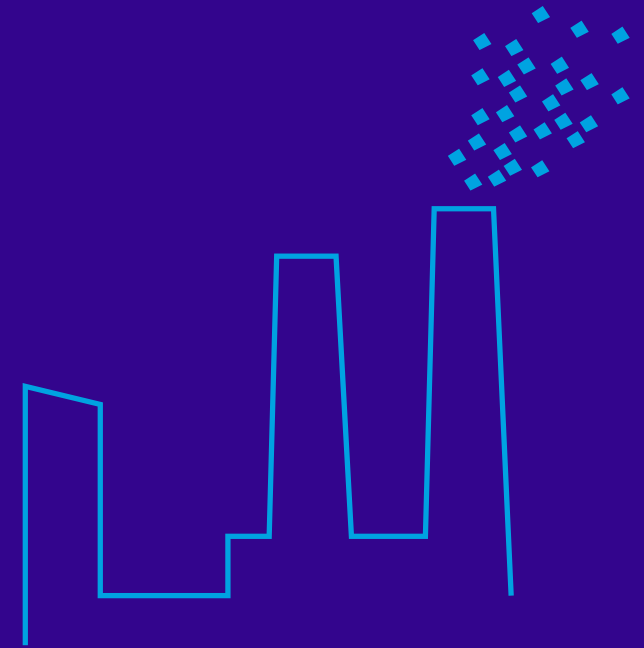
Campus Amiens

`contact-amiens@lamanu.fr`

`https://lamanu.fr`

`70 rue des Jacobins, 80090 Amiens`

`09 86 27 17 04`



Campus Compiègne

contact-compiegne@lamanu.fr

<https://lamanu.fr>

Rue Robert Schuman – La Croix-Saint-Ouen | 60200 Compiègne
Entrée par le 41 rue Irène Joliot Curie (Bâtiment
Millenium II)

09 86 27 17 04

