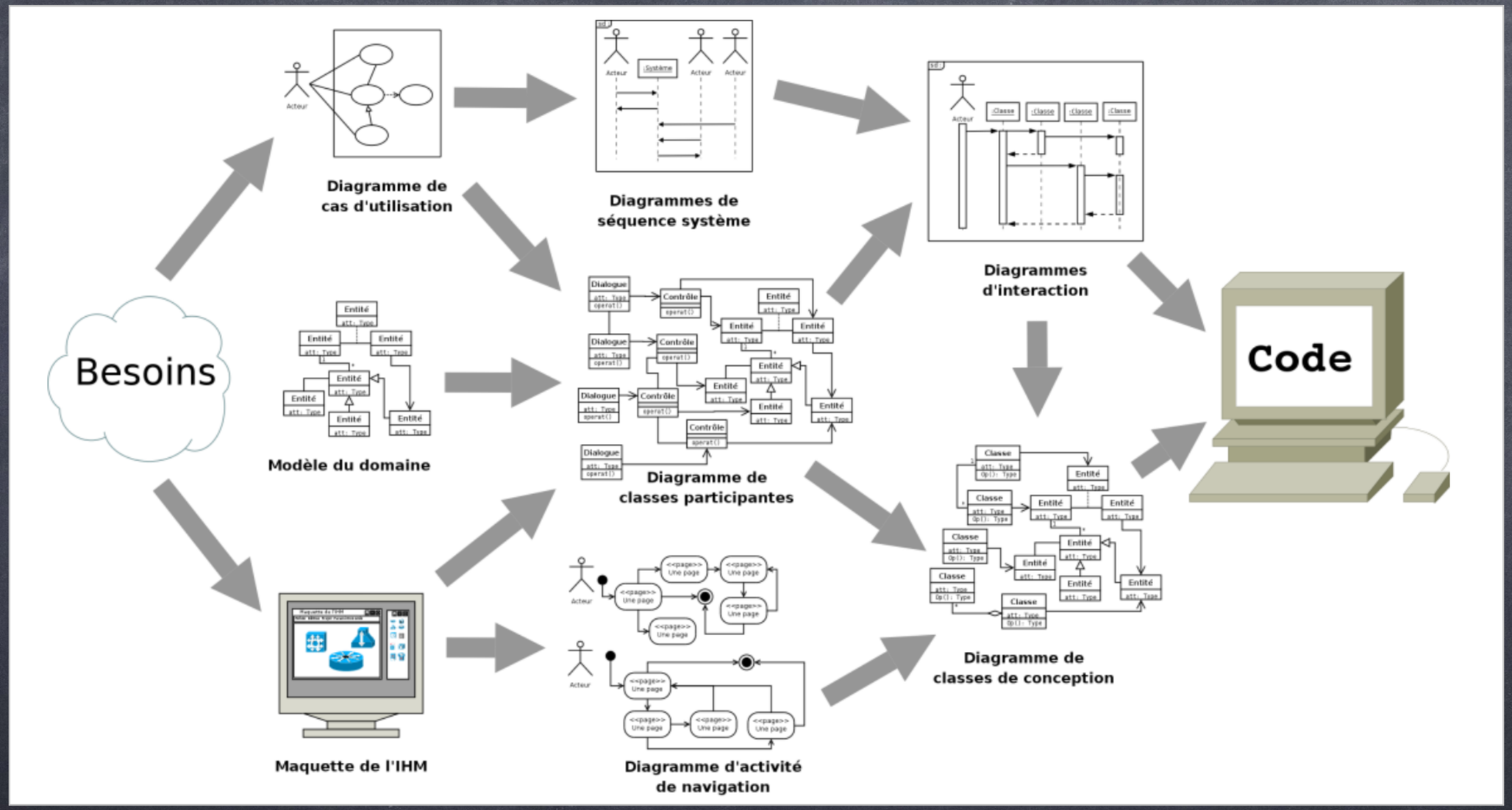
Conception d'application

UML Le cheminement

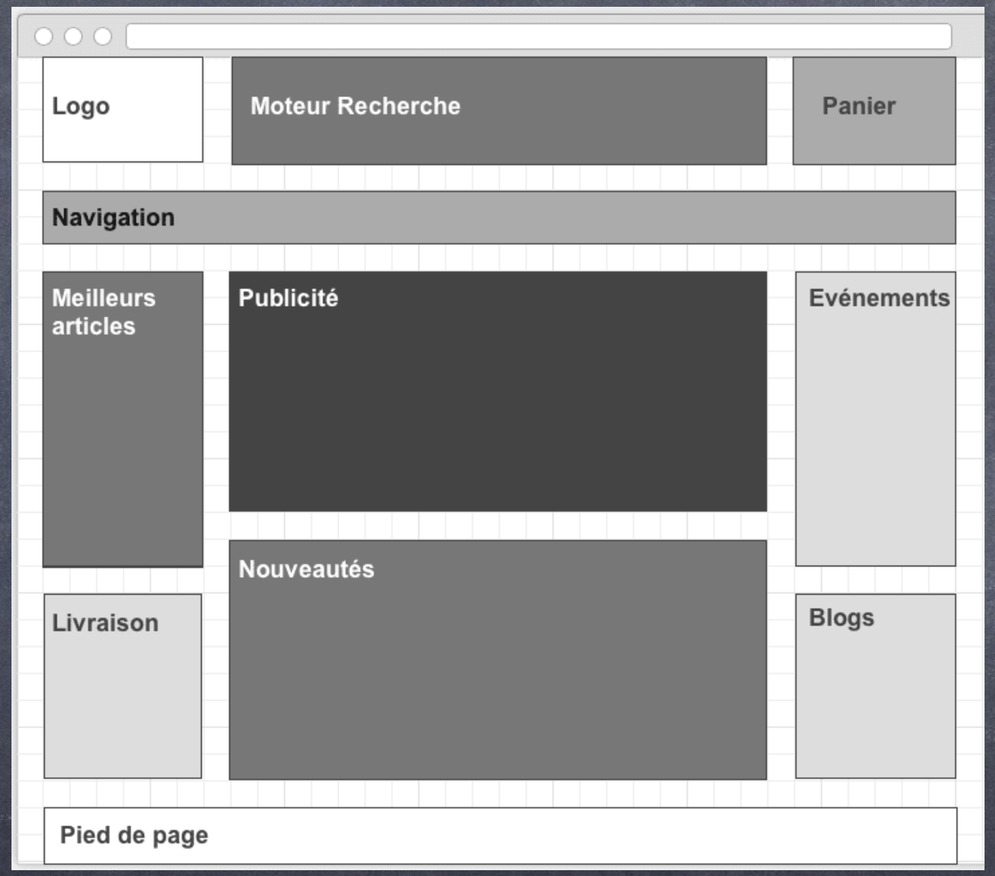


Conception de l'IHM

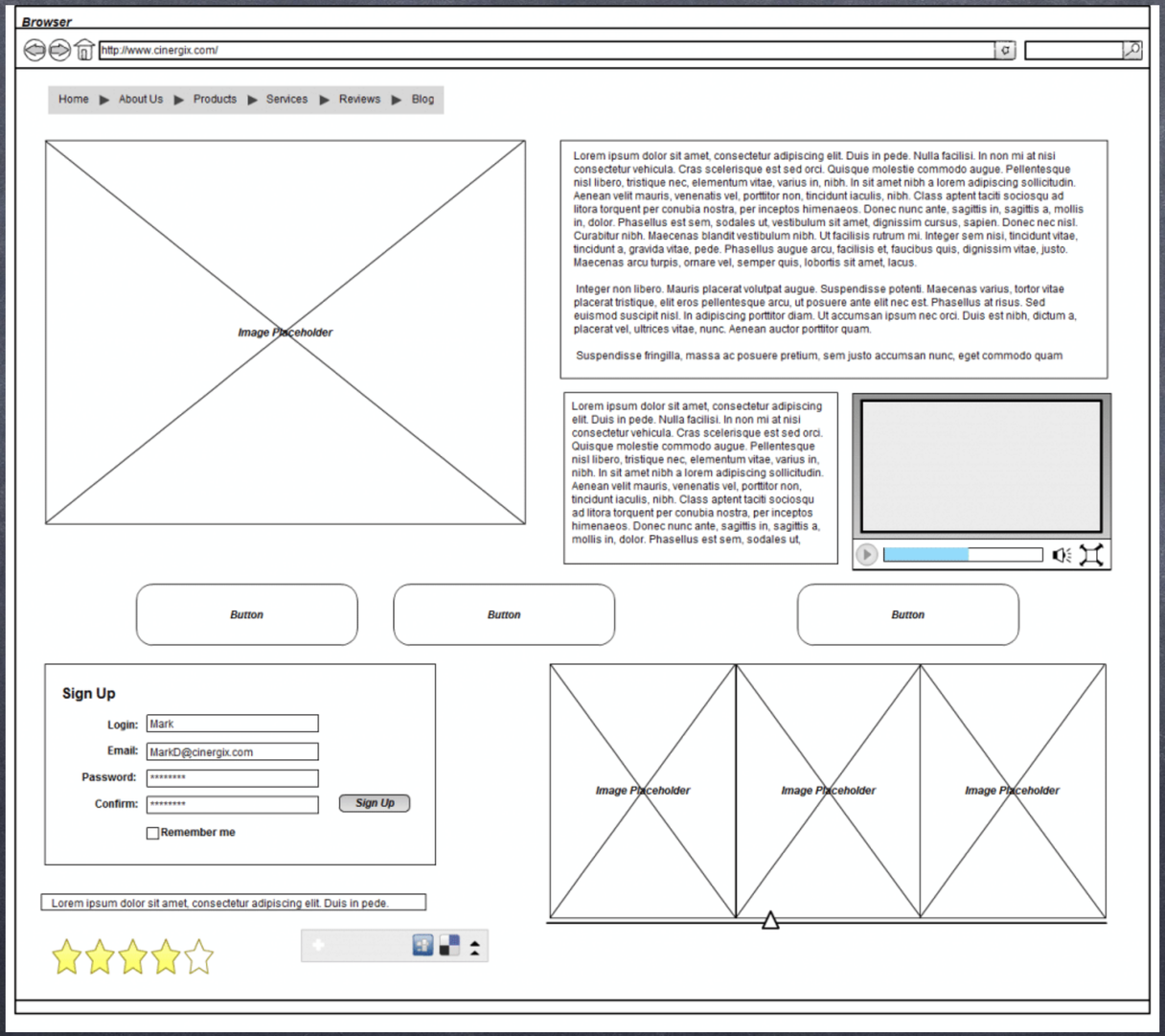
Interface Homme Machine

Graphical User Interface (GUI)

UML Le zoning



UML Maquette



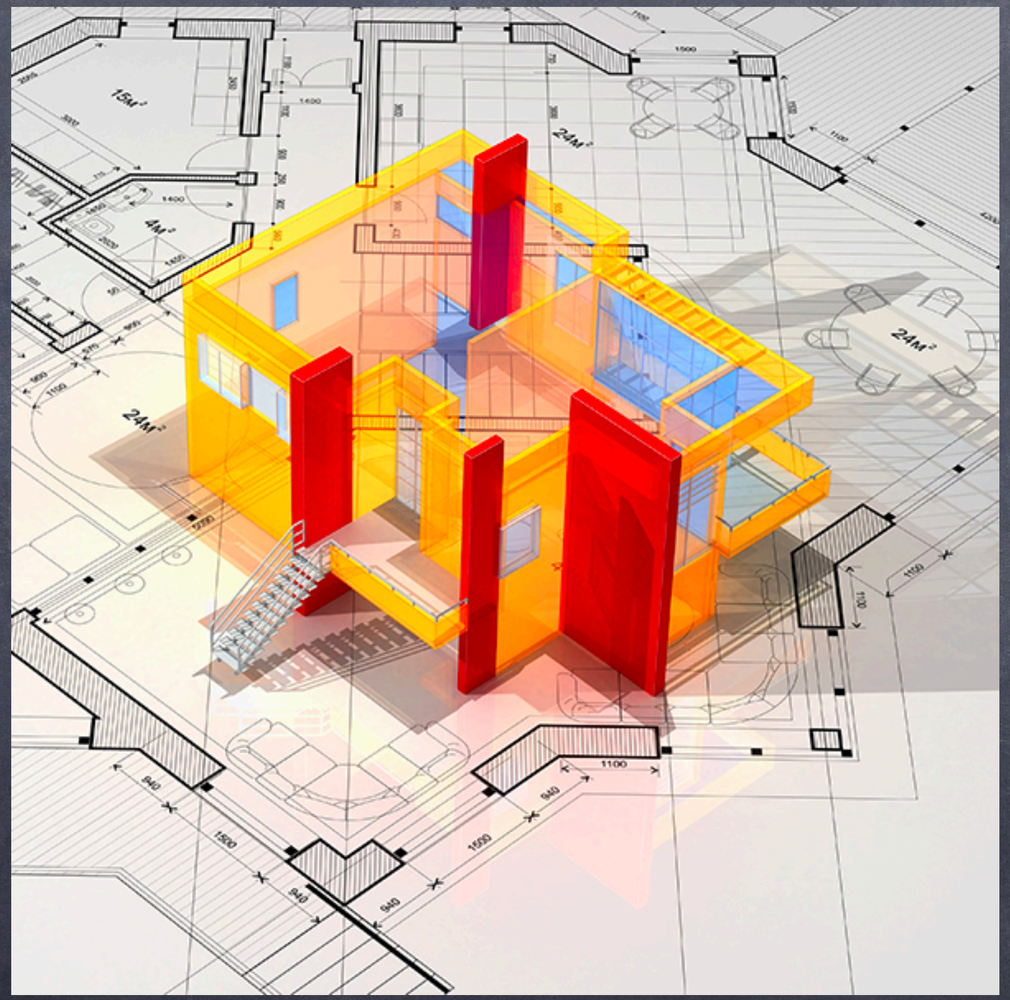
Les outils de zoning

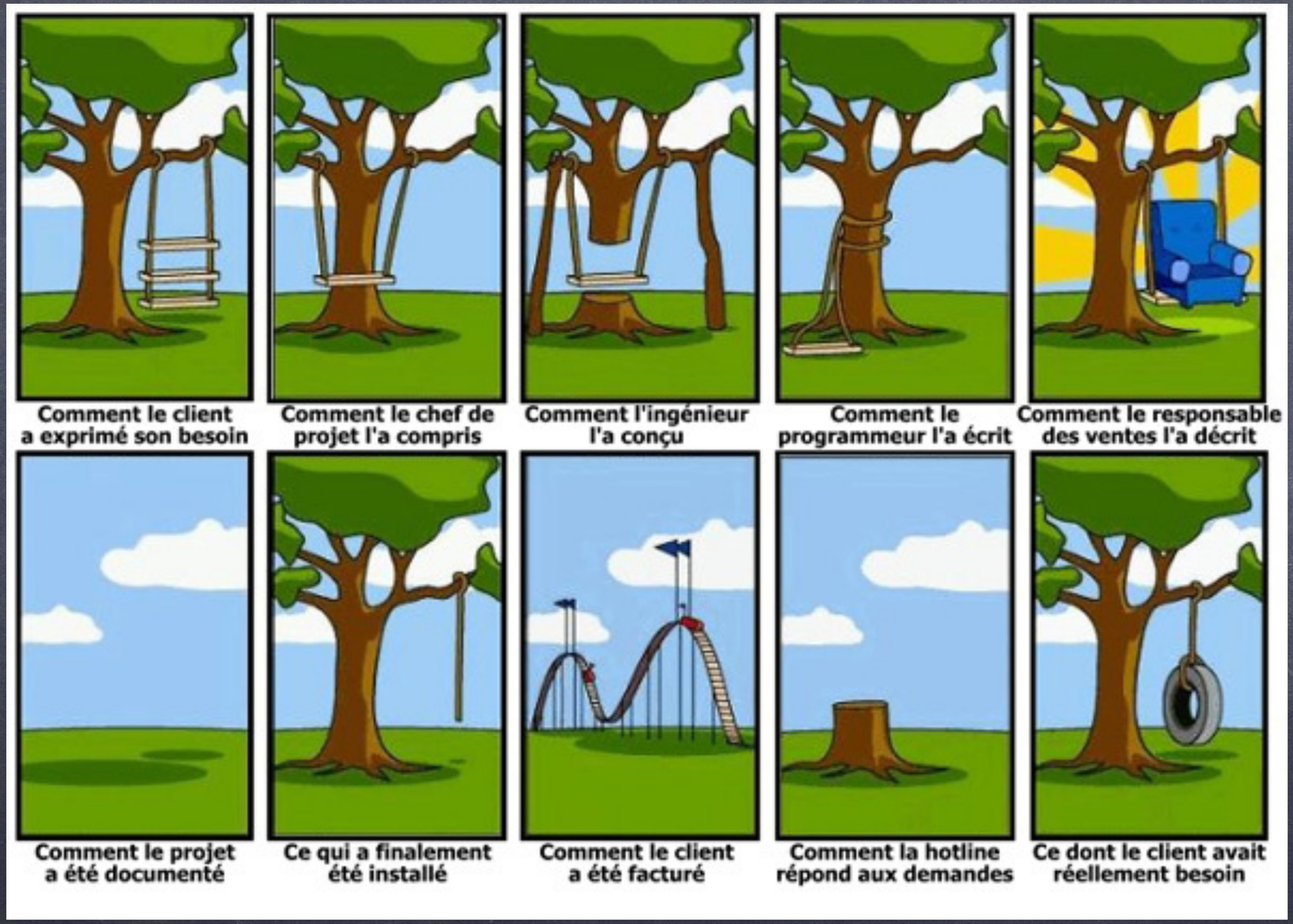
- ✓ Un crayon, un carnet et un téléphone
- ✓ N'importe quel logiciel de dessin, vectoriel de préférence
  - ✓ Inkscape
  - ✓ draw.io

Les outils de maquette

- ✓ Un crayon, un carnet et un téléphone
- ✓ penpot
- ✓ draw.io
- ✓ mydraft
- ✓ pencil

Modélisation




Comment le client a exprimé son besoin
Comment le chef de projet l'a compris
Comment l'ingénieur l'a conçu
Comment le programmeur l'a écrit
Comment le responsable des ventes l'a décrit
Comment le projet a été documenté
Ce qui a finalement été installé
Comment le client a été facturé
Comment la hotline répond aux demandes
Ce dont le client avait réellement besoin

Modèle

Représentation approchée d'une réalité (phénomène, produit...)

exemples

plan de l'architecte,

modélisation mathématique
(pandémie, climat, économie)

Plusieurs utilisation

- ✓ Descriptive : le modèle décrit une réalité existante
- ✓ Prescriptive : le modèle décrit une réalité espérée
- ✓ Prospective : le modèle décrit une évolution future

Descriptif

- ✓ La carte géographique
- ✓ Le plan du métro

Prescriptif

- ✓ Le plan de l'architecte
- ✓ Le schéma de montage d'un meuble
- ✓ Le dessin industriel

Prospectif

- ✓ Le modèle d'évolution du climat du Giec
- ✓ Les modèles d'évolution d'une pandémie

Dans tous les cas

- ✓ La carte n'est pas le territoire
- ✓ Le modèle est une simplification
- ✓ Le modèle possède une intention

ISLANDE
NORVÈGE
SUÈDE
FINLANDE
RUSSIE
ESTONIE
LETTONIE
LITUANIE
Kaliningrad
DANEMARK
IRLANDE
ROYAUME-UNI
BIÉLORUSSIE
POLOGNE
PAYS-BAS
ALLEMAGNE
BELGIQUE
UKRAINE
LUX.
RÉP. TCHÈQUE
SLOVAQUIE
FRANCE
LIEC.
MOLDAVIE
SUISSE
AUTRICHE
HONGRIE
ROUMANIE
SLOVÉNIE
CROATIE
CRIMÉE
ITALIE
BOSNIE
SERBIE
ANDORRE
MONACO
ST-MARIN
ESPAGNE
MONTÉNÉGRO
BULGARIE
PORTUGAL
VATICAN
MACÉDOINE
ALBANIE
TURQUIE
GRÈCE
© Jacques Leclerc 2019
MALTE
CHYPRE

Islande
Mer de Norvège
Mer Blanche
Suède
Finlande
Russie
Norvège
Stockholm
Helsinki
Saint-Pétersbourg
Estonie
Moscou
Lettonie
Riga
Kazakhstan
OCÉAN
Mer du Nord
Danemark
Copenhague
Lituanie
Vilnius
Mer Baltique
Royaume-Uni
Dublin
Irlande
Minsk
Biélorussie
Pays-Bas
Amsterdam
Berlin
Pologne
Varsovie
Kiev
Mer Celtique
Allemagne
Londres
Bruxelles
Ukraine
Mer Caspienne
Manche
Prague
Rép. Tchèque
Slovaquie
Moldavie
Paris
Vienne
Autriche
Budapest
Hongrie
Roumanie
Bucarest
ATLANTIQUE
France
Mer Noire
Belgrade
Serbie
Bulgarie
Sofia
Iran
Italie
Rome
Ankara
Portugal
Madrid
Turquie
Espagne
Syrie
Irak
Chypre
Mer Méditerranée
Malte
Algérie
Tunisie
Typologie urbaine
interCarto

Question

Pourquoi modéliser
dans le cadre d'un projet informatique ?

UML

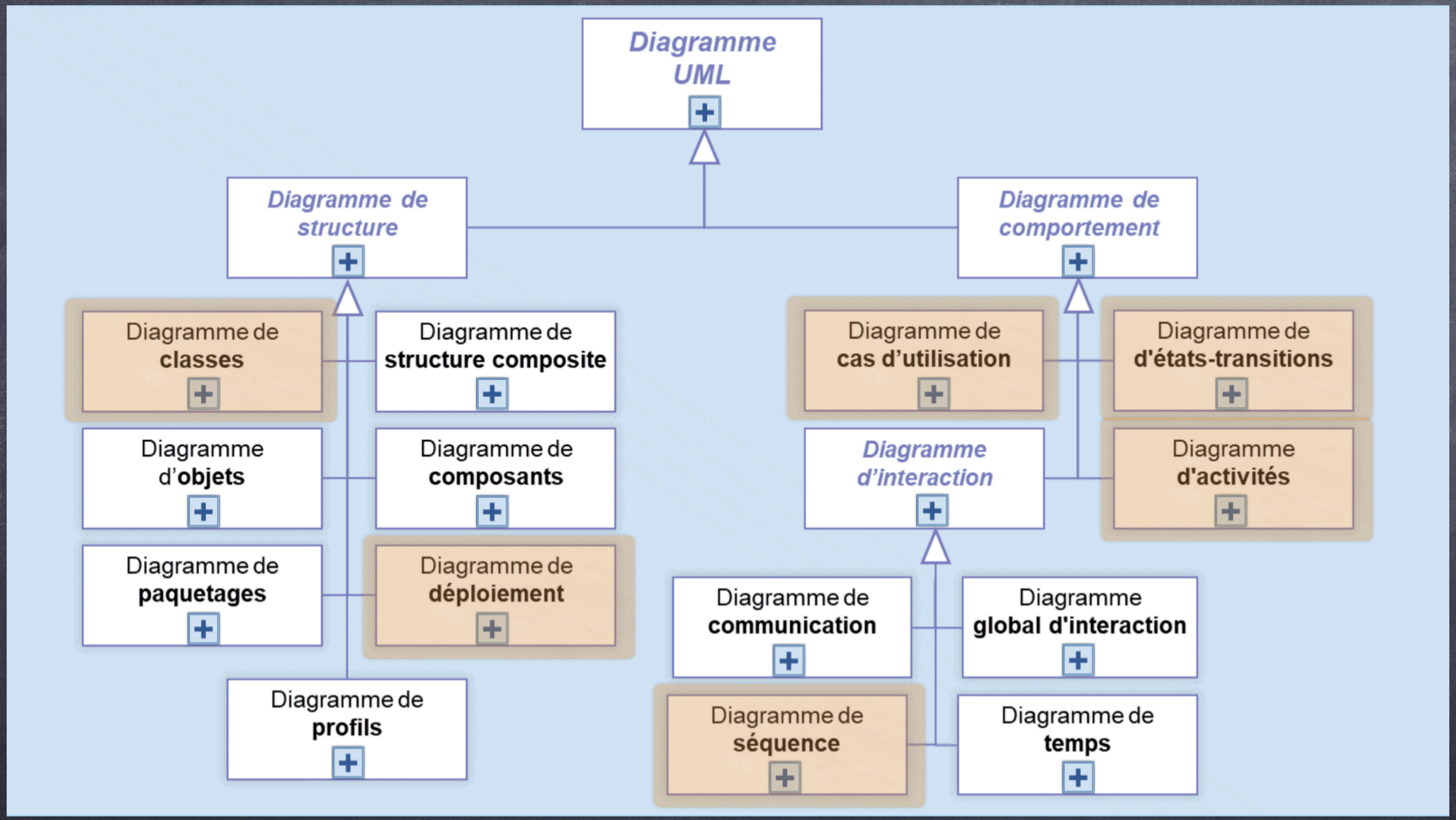
Définition

Formalisme graphique pour décrire
les composants d'un système,
leurs associations et leurs interactions

Historique

- ✓ Créé en 1997
- ✓ Standard géré par l'OMG
  (Object Management Group)
- ✓ Unification de plusieurs autres formalismes
  (Booch, OMT, OOSE)

Les diagrammes



UML pour

- ✓ Communiquer
  chaque symbole graphique a une sémantique connue des autres développeurs
- ✓ Spécifier
  de manière précise et complète, sans ambiguïté,
- ✓ Construire
  certains diagrammes peuvent servir à générer automatiquement du code
- ✓ Documenter
  l'architecture de l'application apparait graphiquement
  sans avoir à lire le code

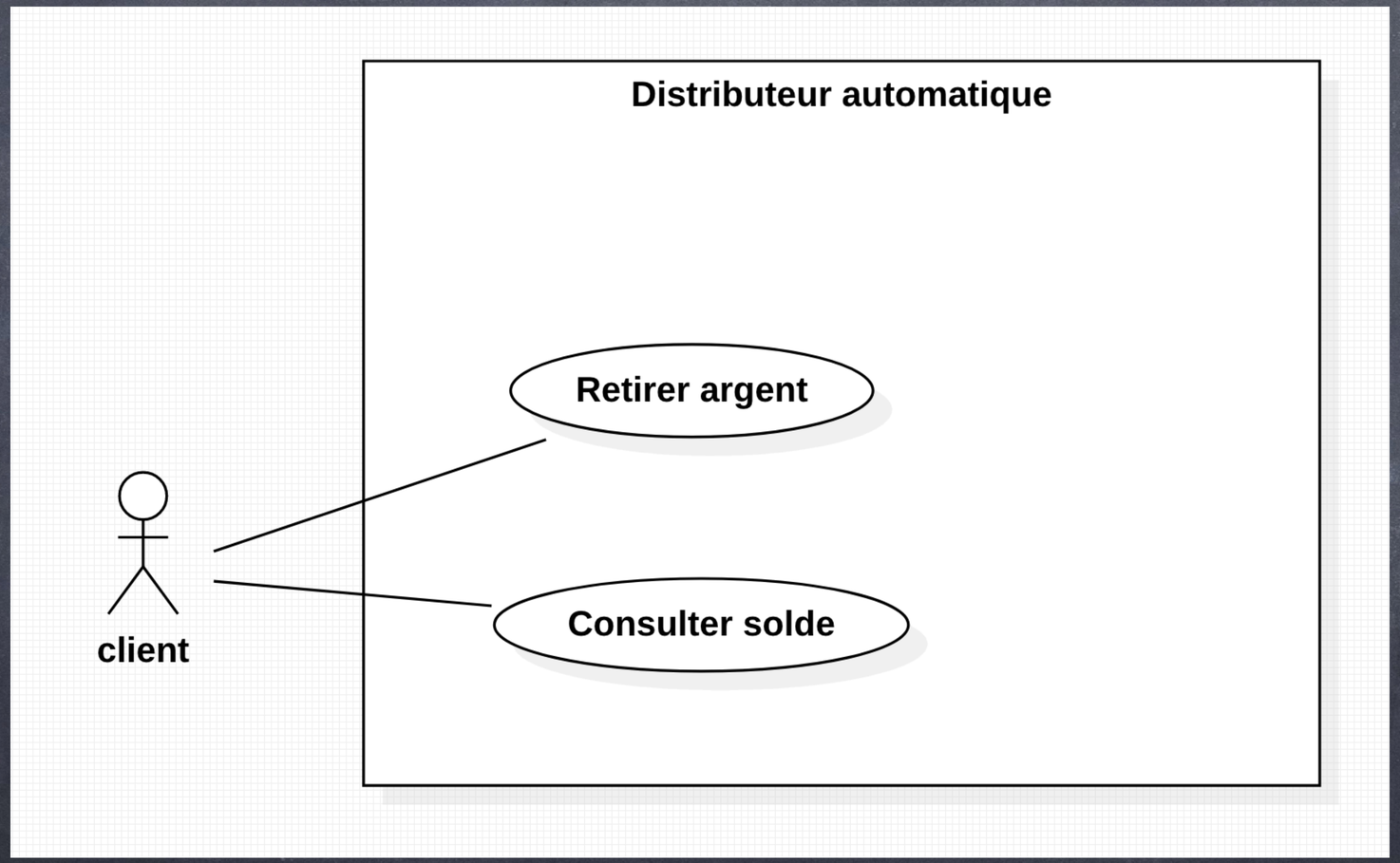
Les outils

- ✓ Visual Paradigm Online
- ✓ Star UML
- ✓ Visual Paradigm CE

Diagramme de cas d'utilisation

Les fonctionnalités d'un système
du point de vue de l'utilisateur

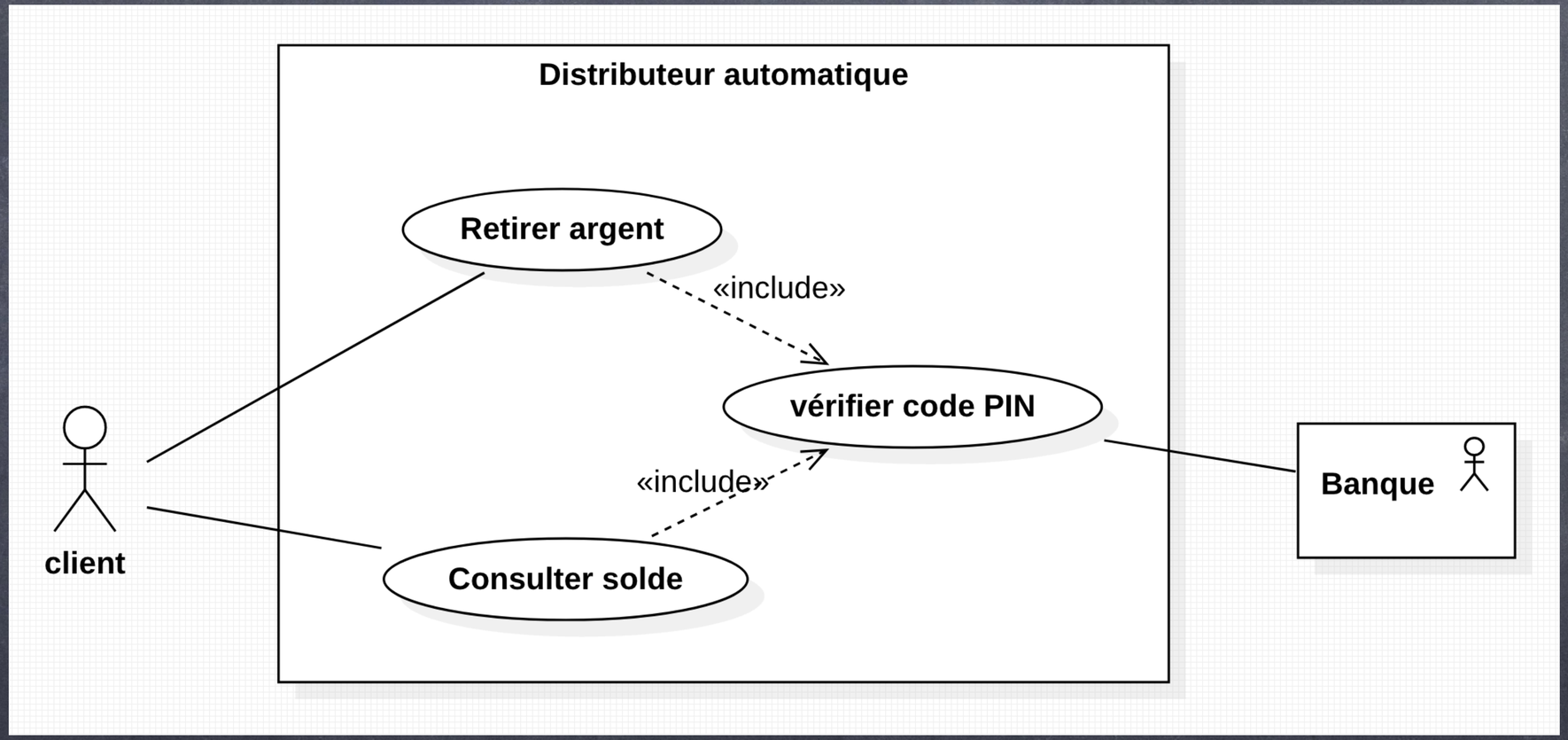
Exemple


Distributeur automatique
Retirer argent
Consulter solde
client

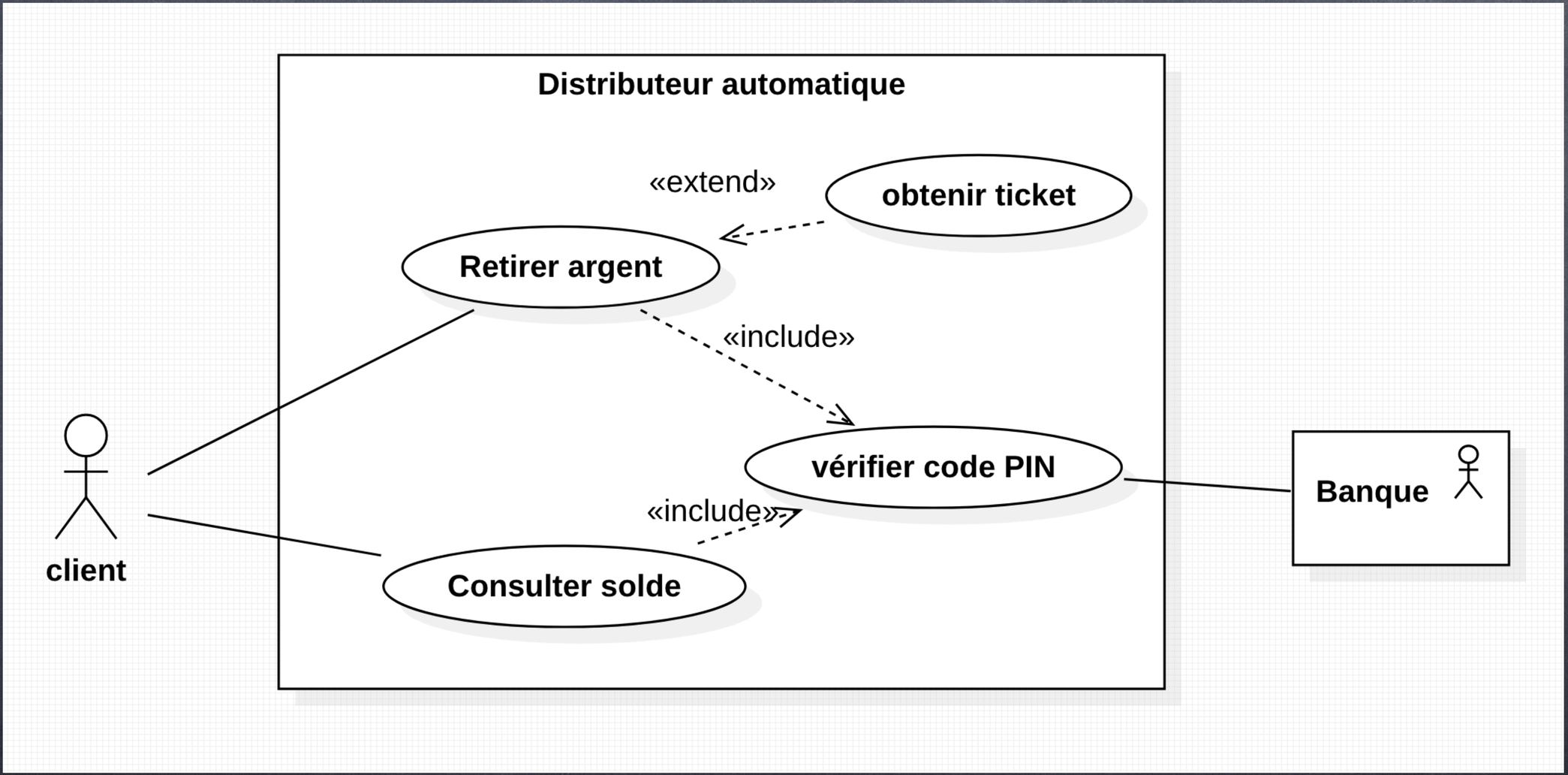
Pourquoi ?

- ✓ Visualiser le périmètre fonctionnel du système (ce que l'on peut faire)
- ✓ Identifier les acteurs et leur rôles
- ✓ Identifier les dépendances entre les fonctionnalités

UML Include


Distributeur automatique
Retirer argent
«include»
vérifier code PIN
«include»
Consulter solde
client
Banque

UML Extend


Distributeur automatique
«extend»
obtenir ticket
Retirer argent
«include»
vérifier code PIN
«include»
Consulter solde
client
Banque

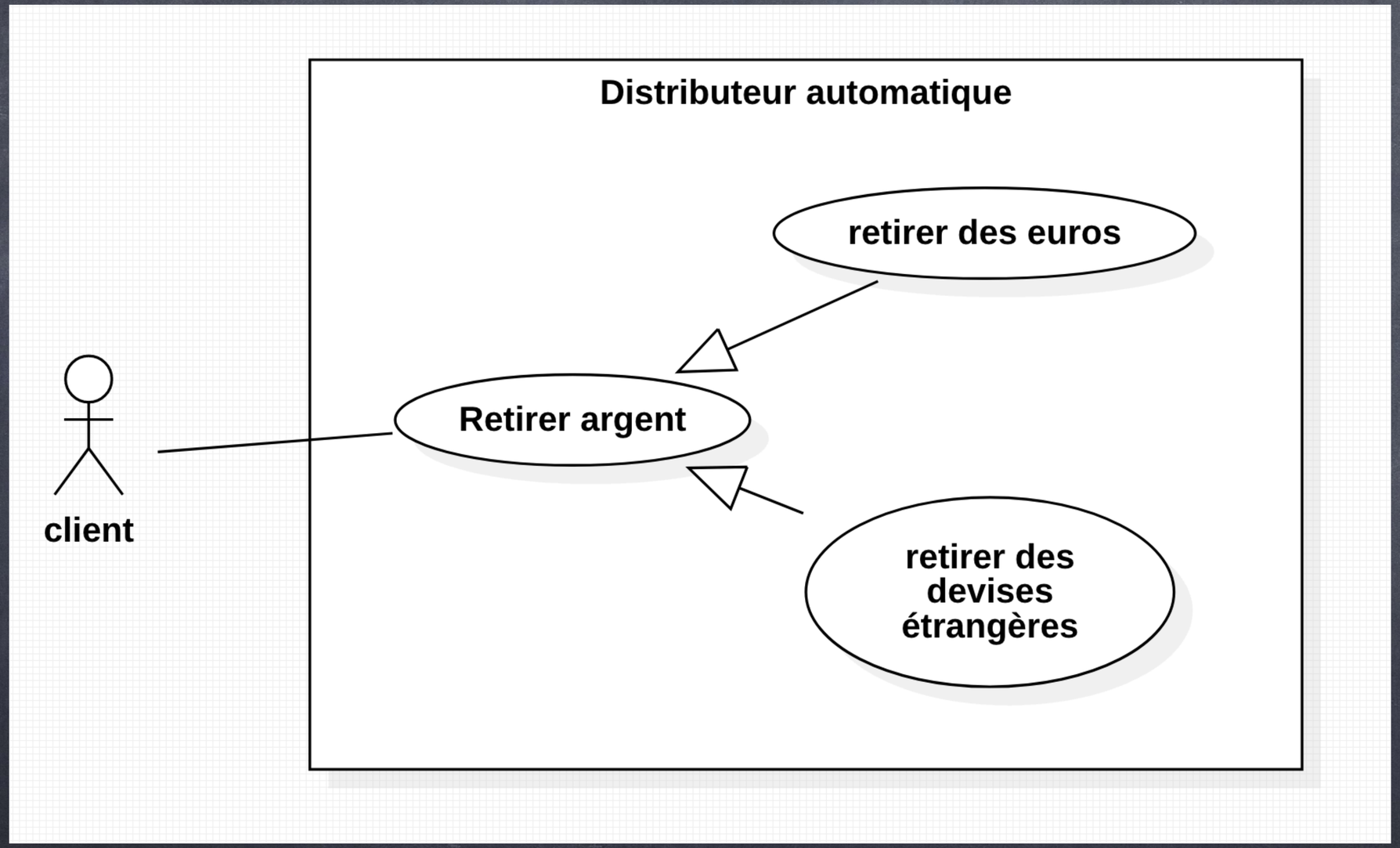
Include ou extend

- ✓ Include : Le cas inclus se déroule obligatoirement lorsque le cas hôte est invoqué. Par exemple il est nécessaire de vérifier le code PIN pour retirer de l'argent
- ✓ Extend : un cas peut se jouer de façon optionnelle lorsqu'un autre cas est invoqué; Par exemple le client peut demander l'impression d'un ticket

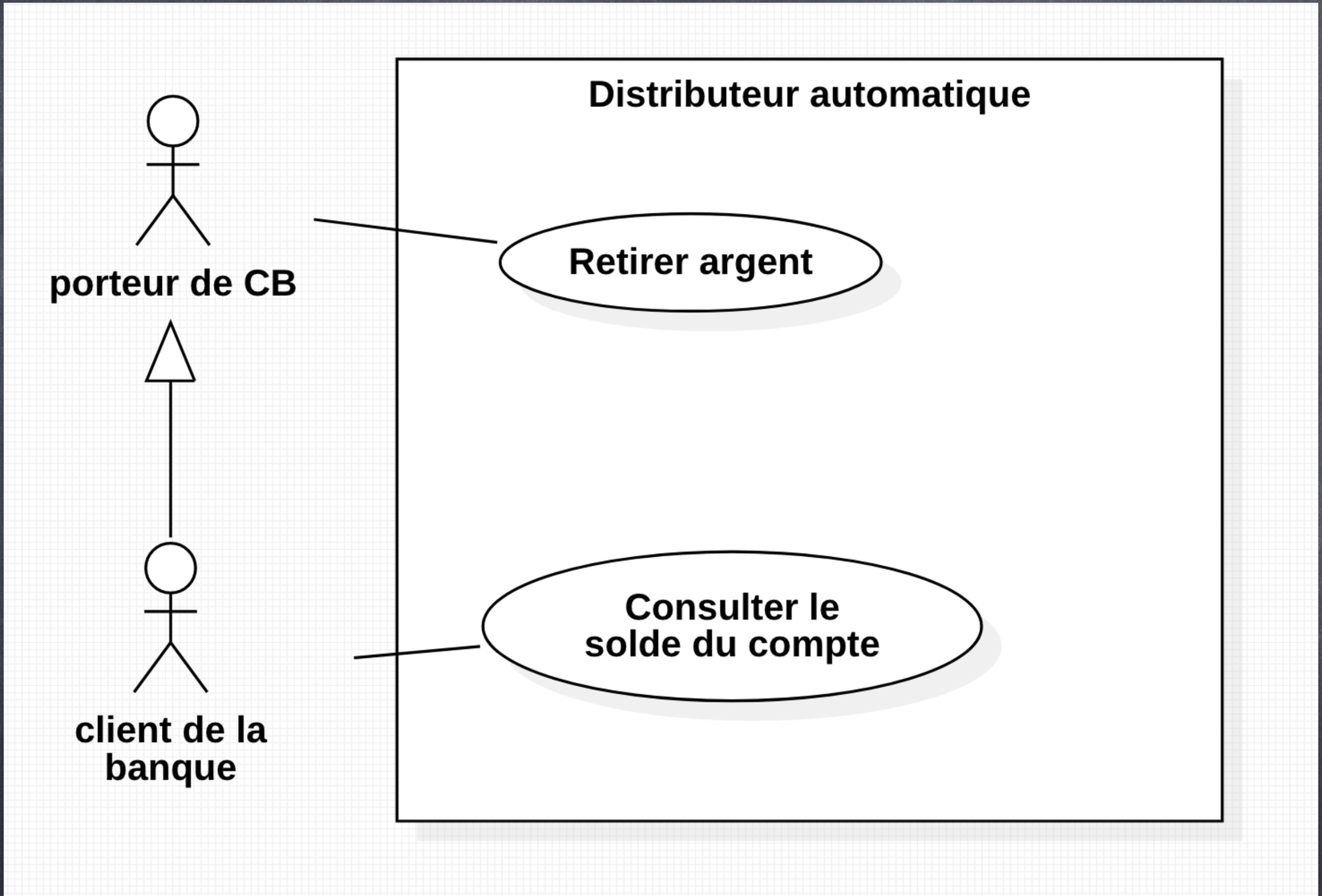
Les acteurs

- ✓ Les acteurs principaux sont ceux qui initient une action. Ils sont généralement représentés à gauche du système
- ✓ Les acteurs secondaires sont ceux qui réagissent aux actions des acteurs principaux. Ils sont généralement représentés à droite du système

Généralisation des cas



Généralisation des acteurs



Conseils

- ✓ Faire simple
- ✓ Avoir une intention
- ✓ Ne pas hésiter à faire plusieurs diagrammes
- ✓ Trouver le niveau de détail pertinent

Conseils

- ✓ Utiliser des verbes explicites qui indiquent l'action (donc éviter les verbes faire, avoir, être..)
- ✓ Utiliser le vocabulaire voire le jargon du métier
  ex : "retirer des devises" plutôt que "prendre des sous"
- ✓ Placer les cas les plus importants en haut à gauche du diagramme
- ✓ Nommer les acteurs de façon précise et en utilisant si possible le champ sémantique du domaine, privilégier un nom qui indique le rôle plutôt que la nature de l'acteur

Conseils

- ✓ Les acteurs n'ont pas d'interactions entre eux, ils n'interagissent qu'avec les cas
- ✓ Pour représenter un cas planifié on peut utiliser un acteur nommé chrono ou temps

Conclusion

- ✓ Les cas d'utilisation présentent une synthèse de ce qu'il est possible de faire avec un système
- ✓ Ils ne détaillent pas le scénario de chaque fonctionnalité pour ça il y a :
  - ✓ le diagramme d'activité
  - ✓ le diagramme de séquence
  - ✓ le détail textuel du scénario

Description textuelle

- ✓ Nom du cas
- ✓ Auteur
- ✓ Dates de création et de modification
- ✓ Pré-conditions et données en entrée
- ✓ Cas nominal ou scénario principal
- ✓ Cas marginaux ou scénarii alternatifs
- ✓ Post-conditions et données en sortie
- ✓ Contraintes non fonctionnelles

Exemple de cas nominal et marginaux

	étapes	Acteur	Système
cas nominal	1	Insère la carte	
	2		vérifie la validité de la carte
	3	Entre le code	
	4		Vérifie le code
	5		Autorise l'accès
cas marginaux	2a		Carte invalide Affiche un message et éjecte la carte
	4a		Code incorrect Affiche un message et redemande le code (deux fois)
	4b		Code incorrect (troisième fois) Affiche un message et ne rend pas la carte

Autre exemple

Cas d'utilisation : S'authentifier

Acteurs : Administrateur et autres utilisateurs.

Objectif : Il permet à l'acteur de s'identifier en saisissant son login et mot de passe.

Précondition : L'acteur doit être présent dans la base de données.

Postcondition :
- Acteur authentifié.
- La page d'accueil s'affiche.

Scénario nominal :
1. L'acteur ouvre l'application,
2. Le système affiche la page d'authentification,
3. L'acteur saisit le login et le mot de passe,
4. Le système vérifie l'existence des données,
5. Le système affiche la page d'accueil.

Scénarii alternatif :

4a. Erreur d'authentification : login ou mot de passe non valide.
Cet enchaînement démarre au point 4.
5. Le système affiche un message d'erreur.
Le scénario reprend au point 2.

4b. Champs obligatoires vides.
Cet enchaînement démarre au point 4.
Le scénario reprend au point 2.

Exigences non fonctionnelles :

- Le mot de passe doit être haché avec un algorithme moderne
- Le système doit répondre en moins d'une seconde

Exercice

Réaliser le diagramme de cas d'utilisation
d'une application de type Todo List

Exercice

Réaliser le diagramme de cas d'utilisation d'un forum de discussion dans lequel les visiteurs peuvent consulter les articles, les utilisateurs authentifiés peuvent poster des réponses et ouvrir de nouveaux sujets. Enfin, l'auteur d'un sujet a la possibilité de modérer les réactions à ses propres articles et les administrateurs peuvent bannir un membre.

Autre intérêt des cas d'utilisation

- ✓ Chaque cas est une tâche à réaliser
  - ✓ Suivi de l'avancement
  - ✓ Arbitrage en fonction de deux critères:
    - ✓ valeur
    - ✓ risque

Exemple

Cas d'utilisation	Priorité	Risque	Poids	Séquence
Gérer le catalogue	Haute	Haut	6	1
Commander	Haute	Haut	6	2
Rechercher un livre	Haute	Moyen	5	3
Gérer un panier	Moyenne	Moyen	4	4
Suivi des commandes de la part de l'internaute	Basse	Haut	4	5
Gérer le système d'aide	Basse	Bas	2	7
Maintenir le site	Basse	Bas	2	6

Conclusion

Grâce aux cas d'utilisation :

- ✓ Le client sait ce que fera son logiciel
- ✓ Le développeur sait ce qu'il va devoir produire
- ✓ On évite les malentendus, les oublis et les conflits

Diagramme d'activité

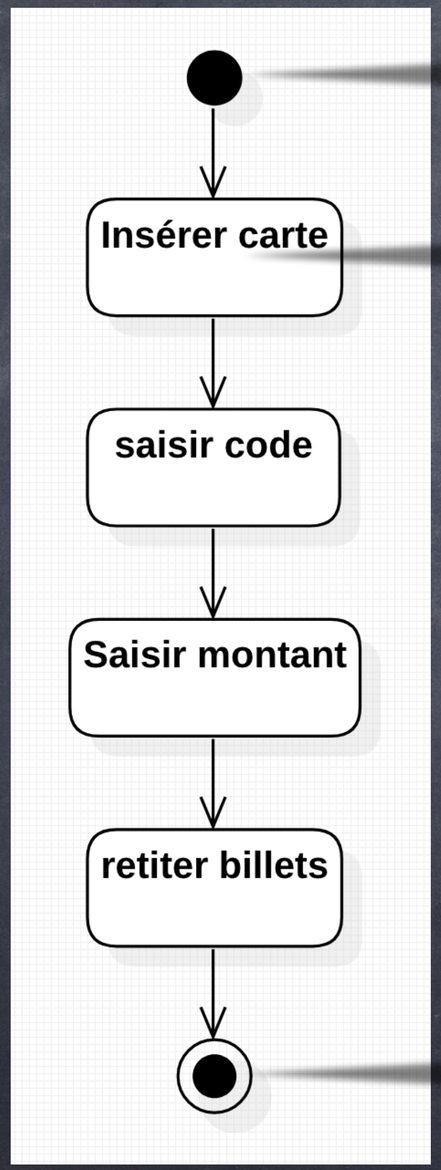
Illustre un processus

et le flux des activités qui le composent

Utilité

- ✓ Représenter les étapes d'un algorithme
- ✓ Représenter tout ou partie du scénario d'un cas d'utilisation
- ✓ Représenter les étapes d'un process métier (pas forcément informatisé)
- ✓ Représenter un plan de navigation (pas prévu dans UML mais préconisé par Pascal Roques)

Exemple

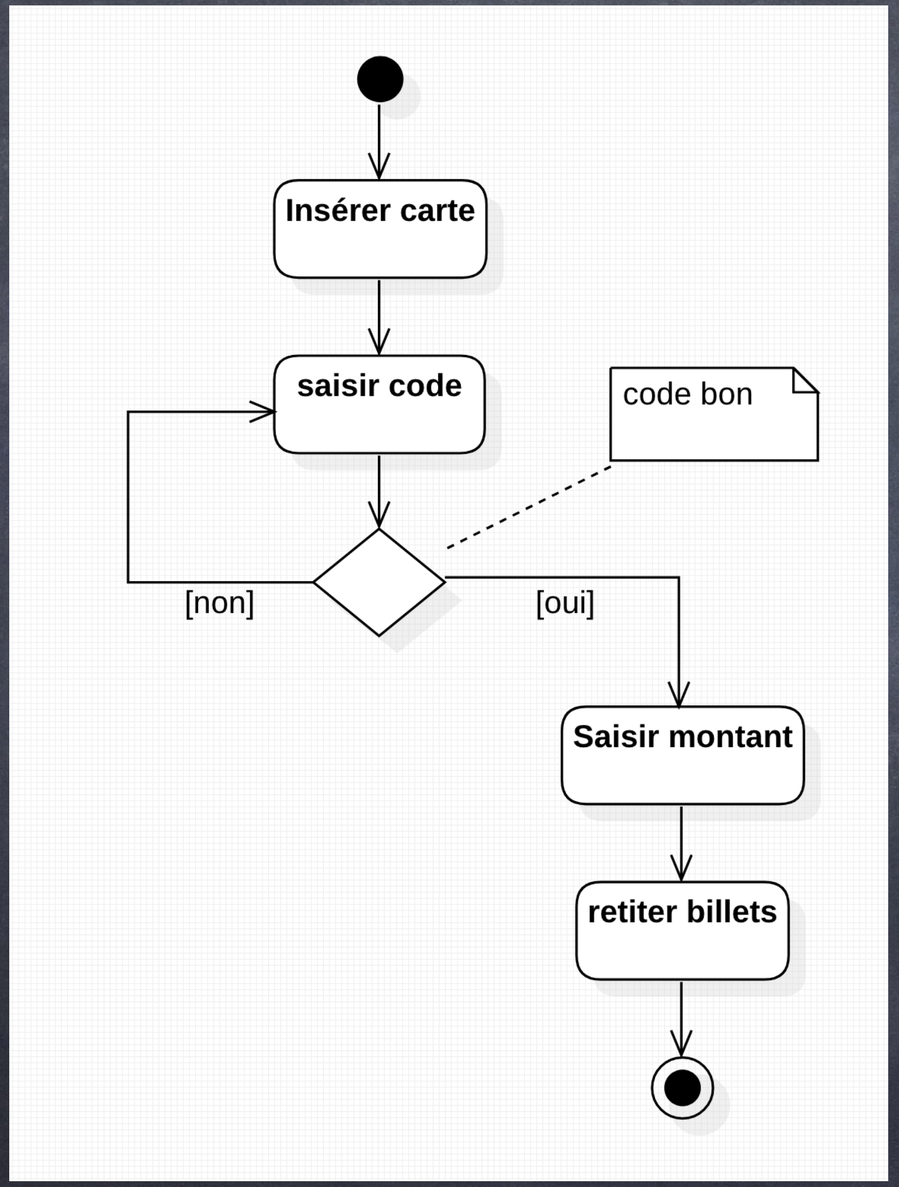


Début

action

fin

UML Décision



Exercice

Faire de diagramme d'activité de la machine à café

Exercice

Faire de diagramme d'activité
de la recherche d'une valeur
dans un tableau

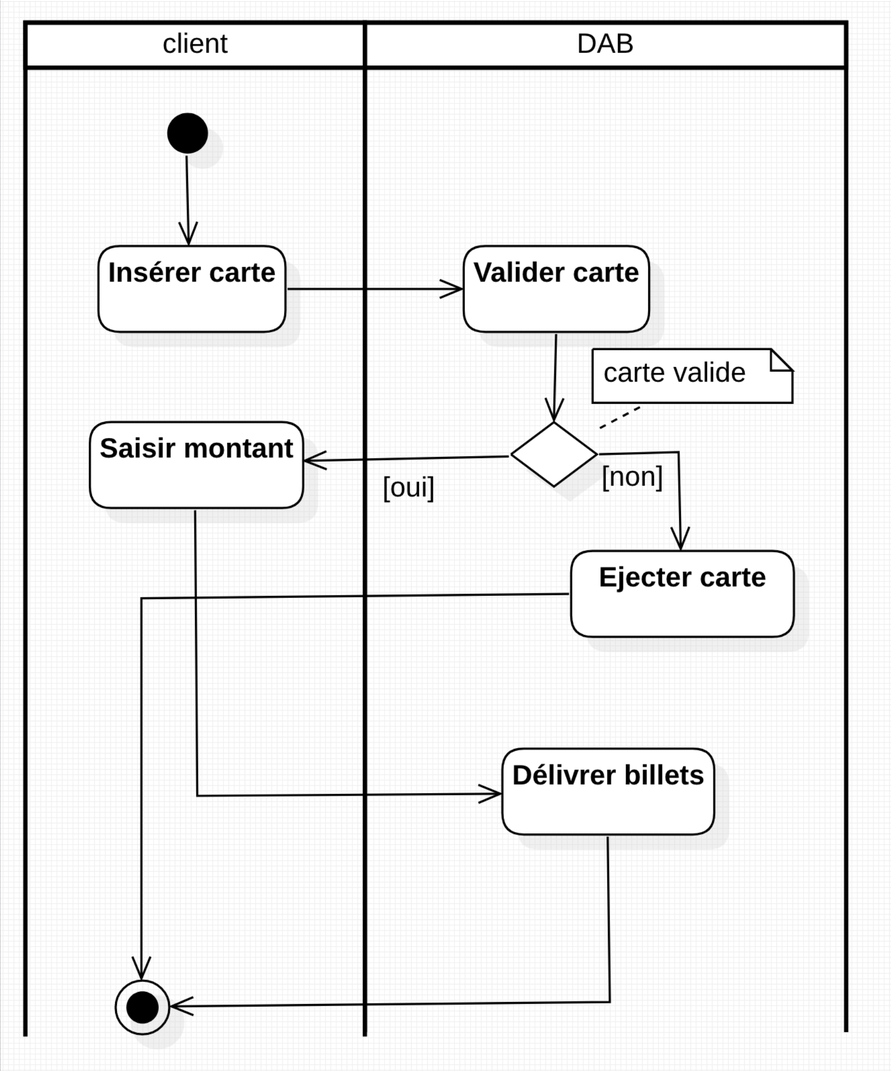
(retourne un booléen)

Exercice

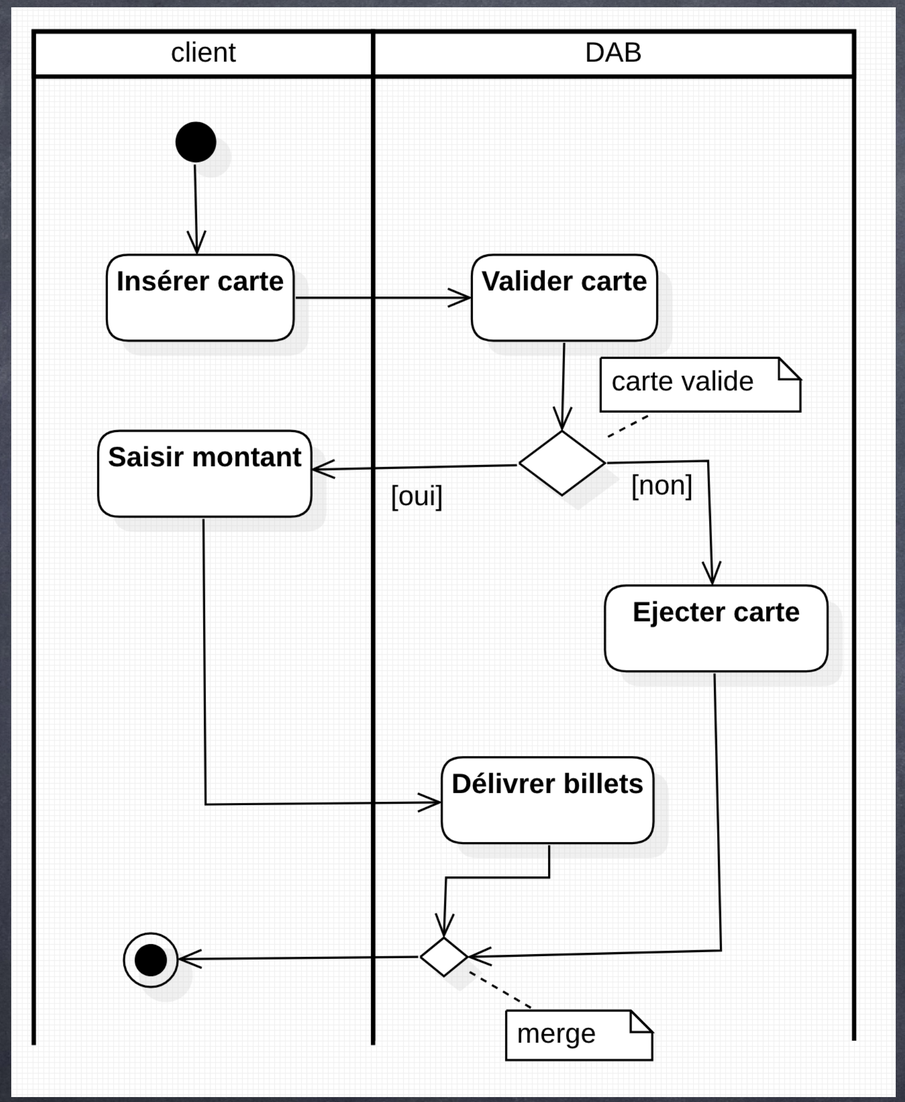
Faire de diagramme d'activité
de la recherche dichotomique
dans un tableau

(retourne un booléen)

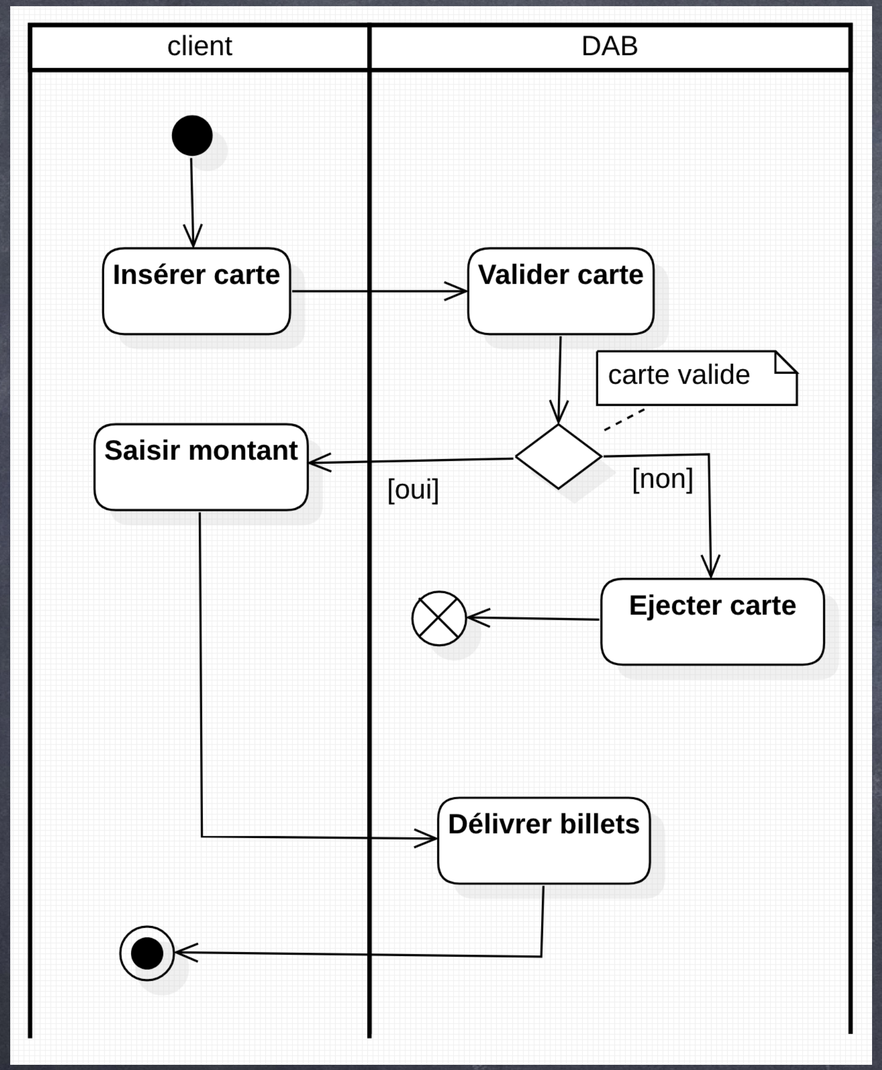
Partitions


client
DAB
Insérer carte
Valider carte
carte valide
Saisir montant
[oui]
[non]
Ejecter carte
Délivrer billets

Variante avec un merge



Simplification avec une sortie d'erreur



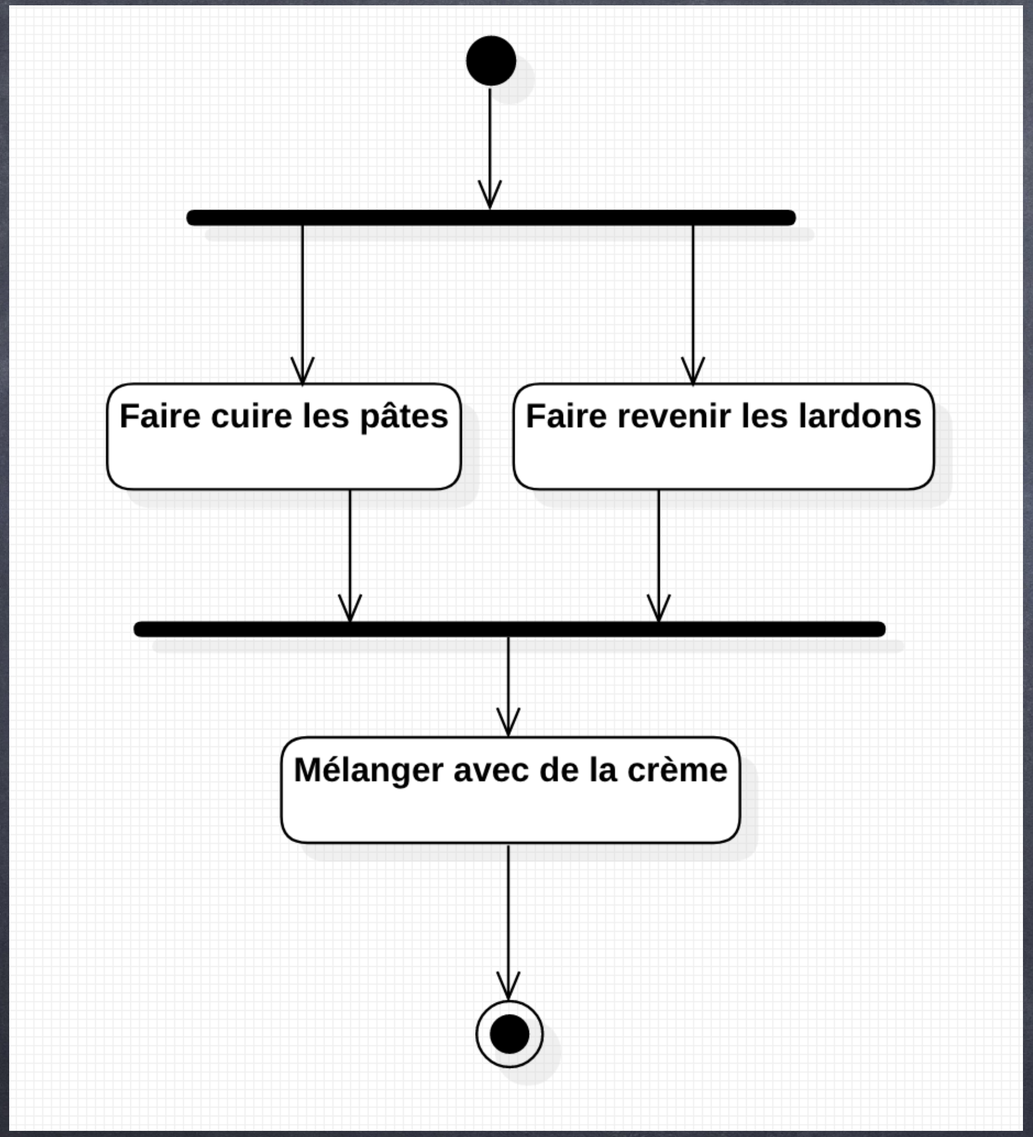
Exercice

Réaliser le diagramme d'activité de l'authentification sur un site web en distinguant les acteurs suivants : client et serveur

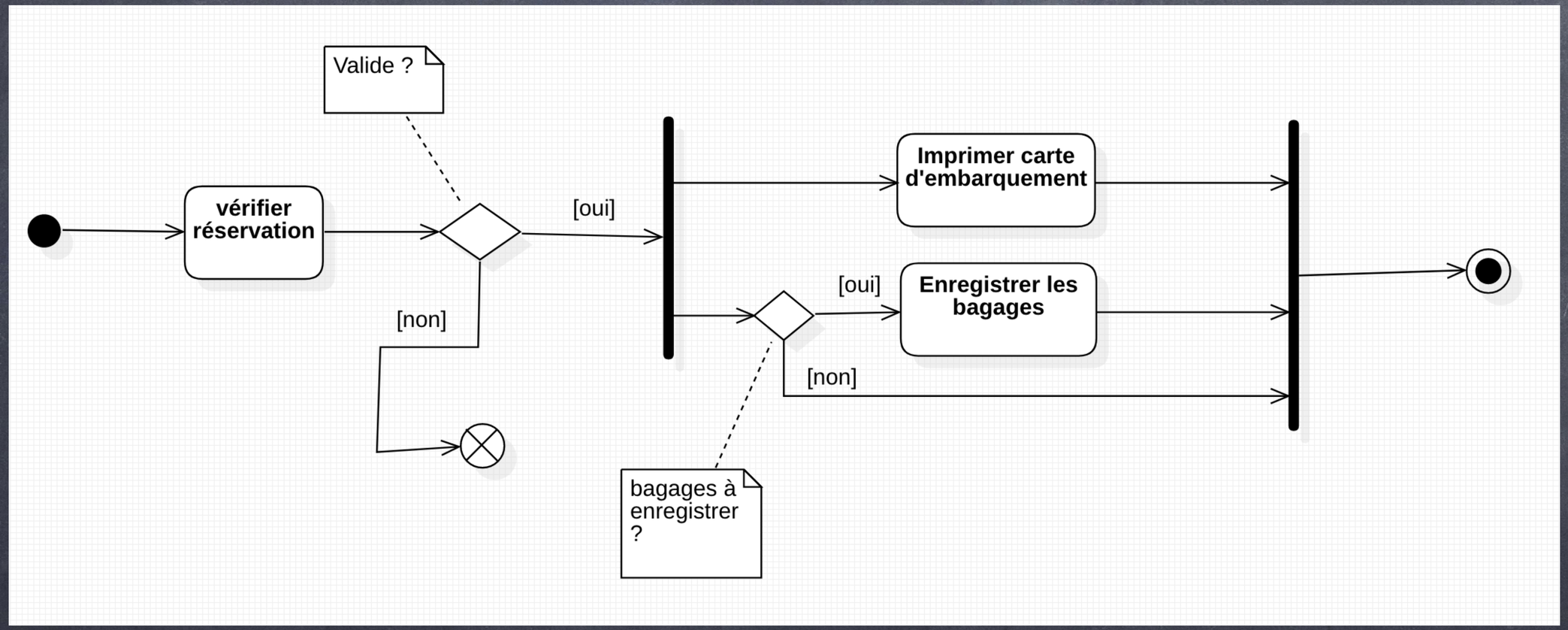
Exercice

Réaliser le diagramme d'activité du retrait d'argent dans un DAB en distinguant les acteurs suivants : le client, le BAB et la banque

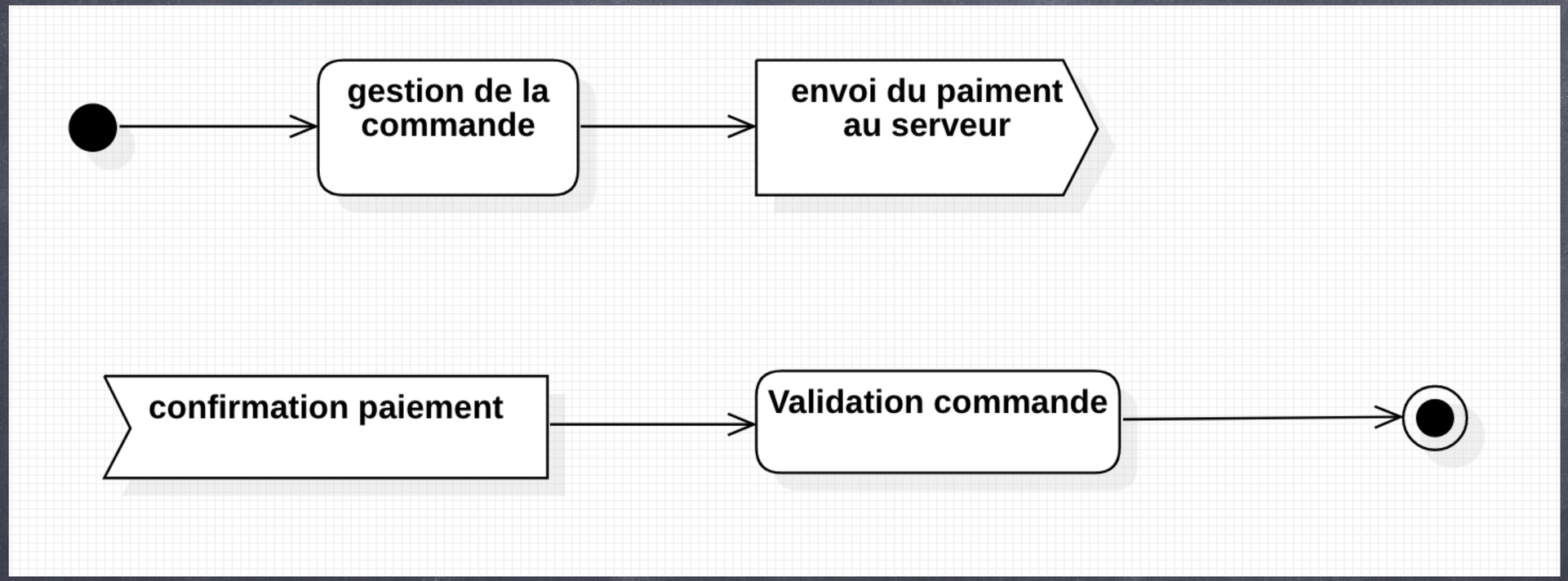
UML Fork et join



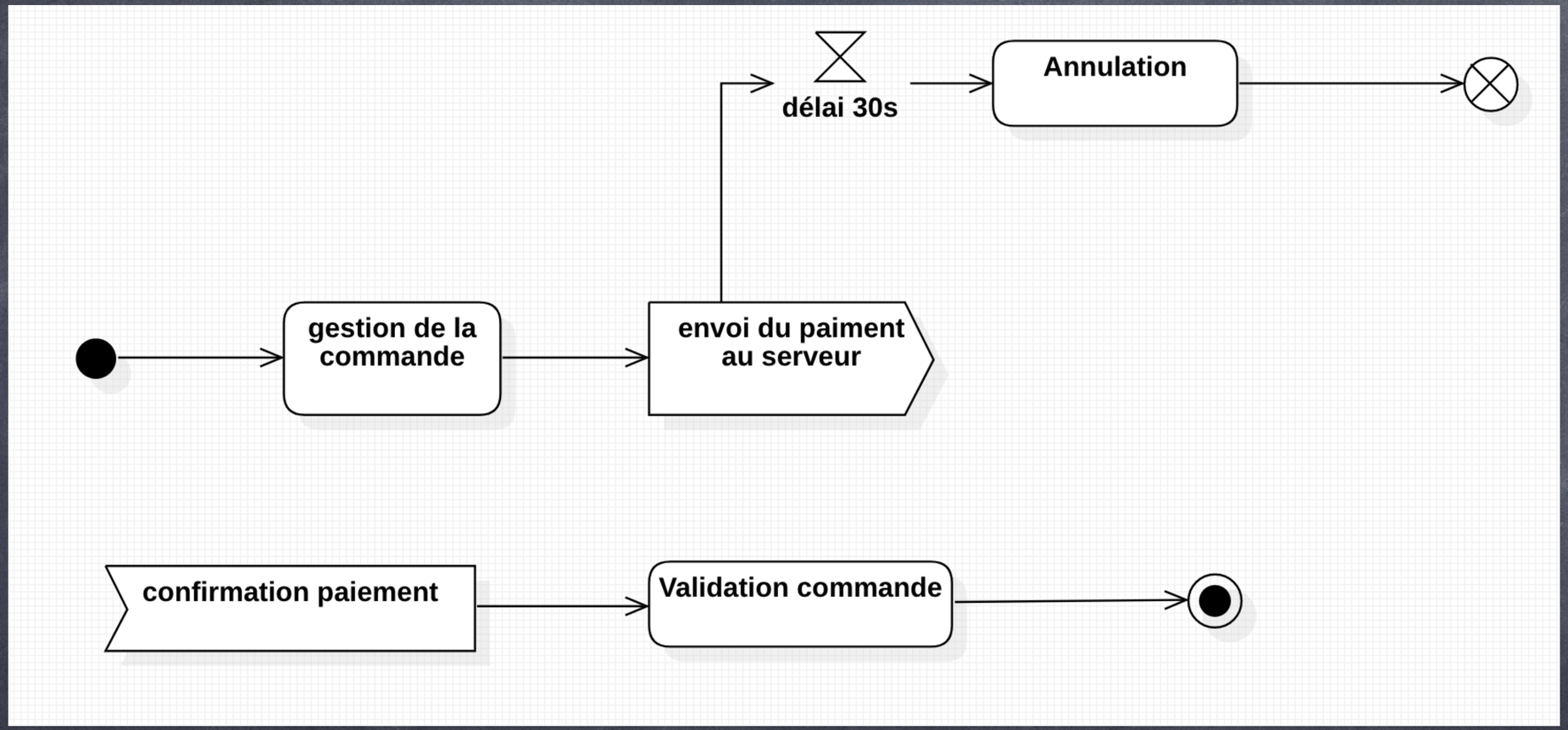
UML Fork et join



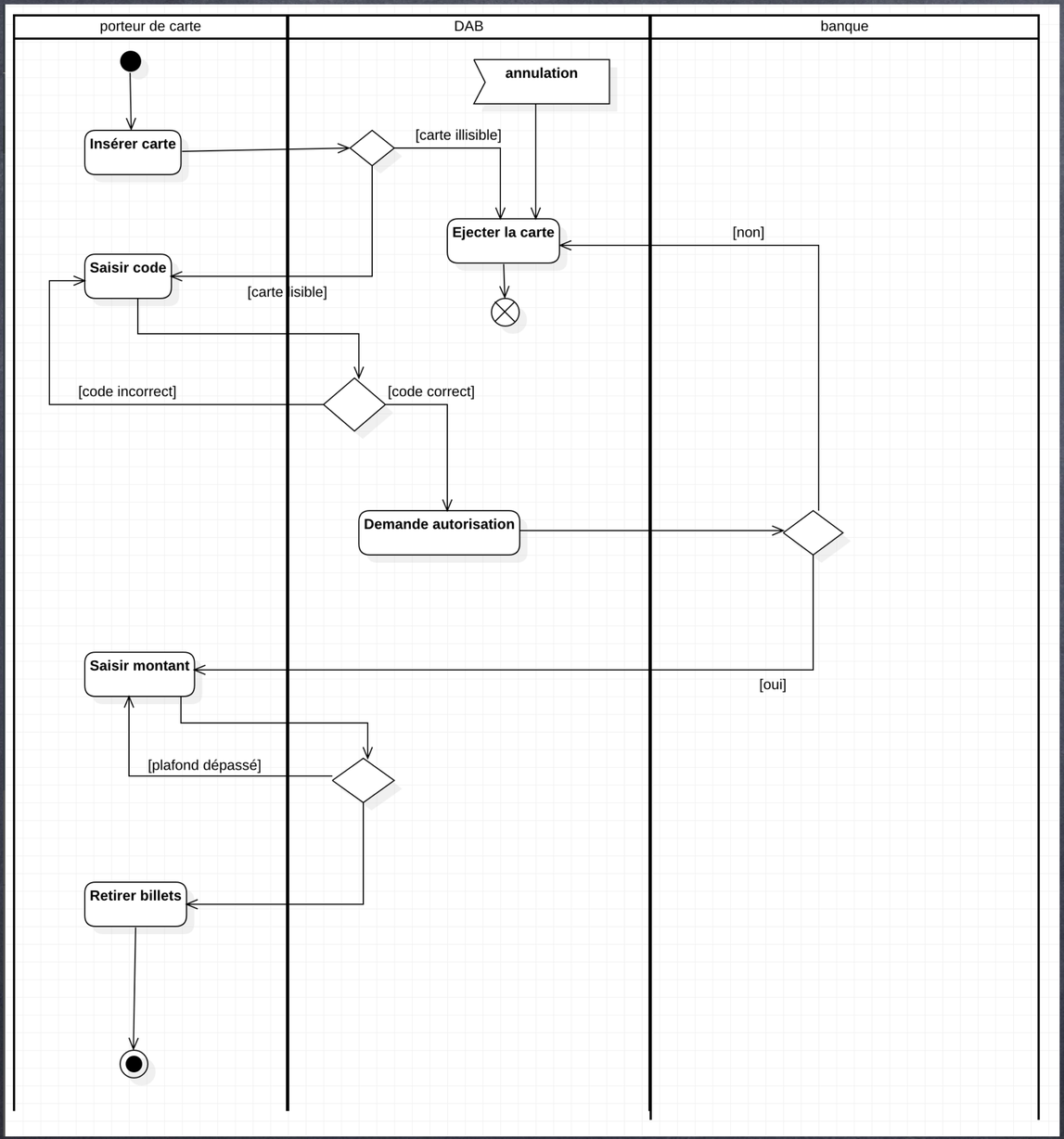
Signaux


gestion de la commande
envoi du paiment au serveur
confirmation paiement
Validation commande

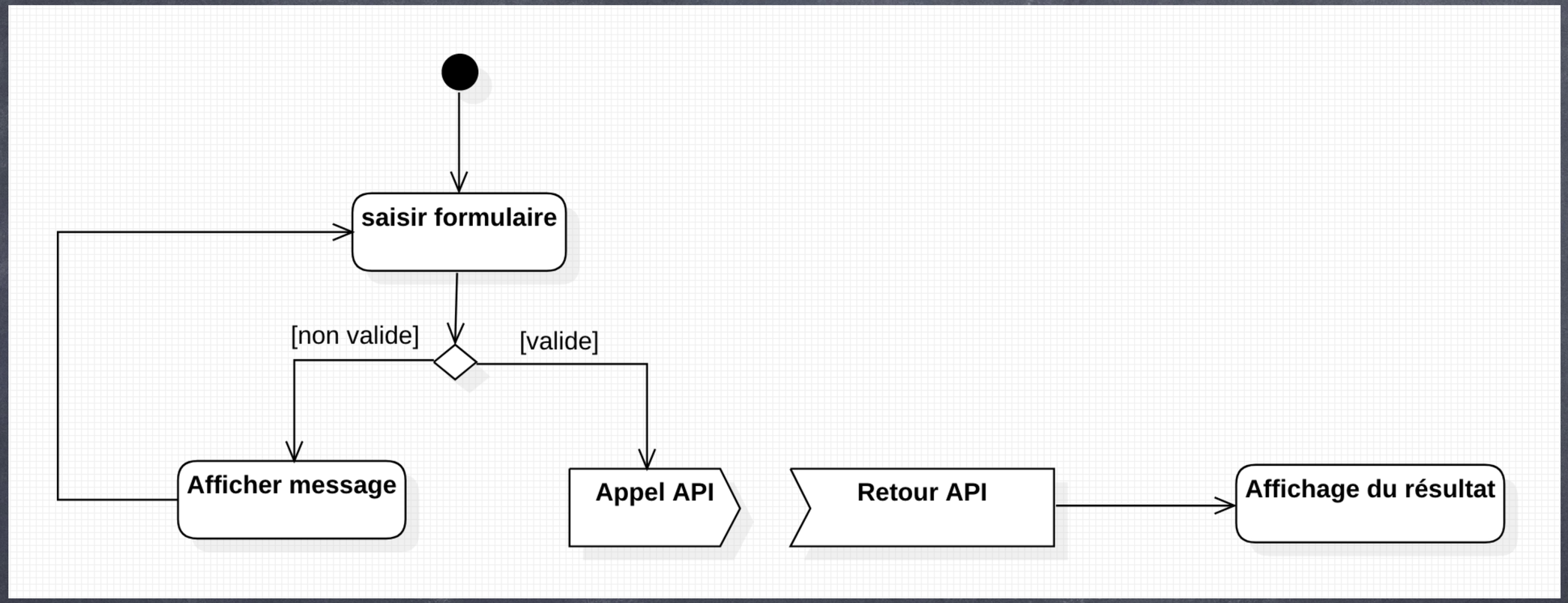
Événement temporel



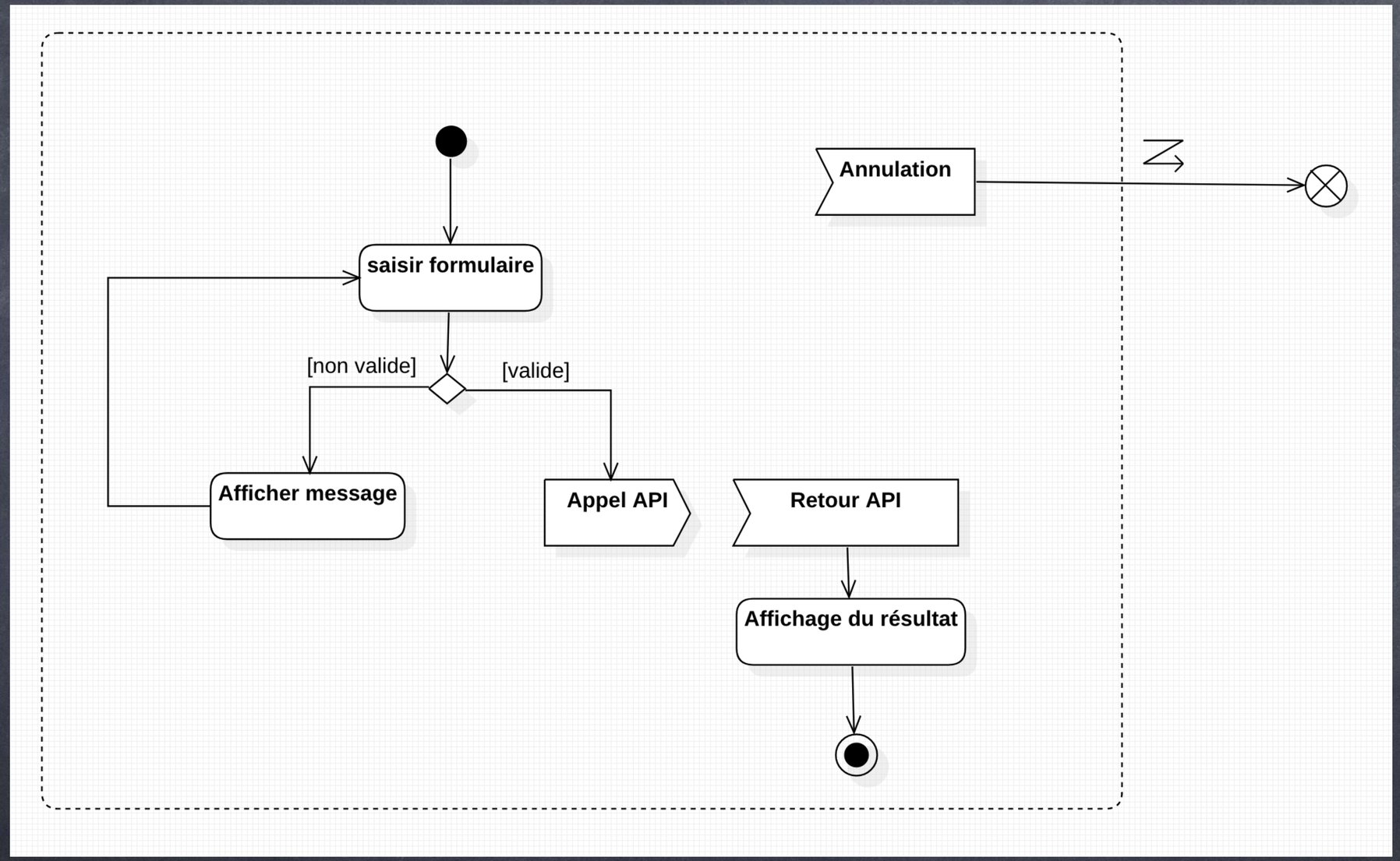
Annulation


porteur de carte
DAB
banque
annulation
Insérer carte
[carte illisible]
Ejecter la carte
[non]
Saisir code
[carte lisible]
[code incorrect]
[code correct]
Demande autorisation
Saisir montant
[oui]
[plafond dépassé]
Retirer billets

Appel API


saisir formulaire
[non valide]
[valide]
Afficher message
Appel API
Retour API
Affichage du résultat

Interruptions



UML Expansion regions

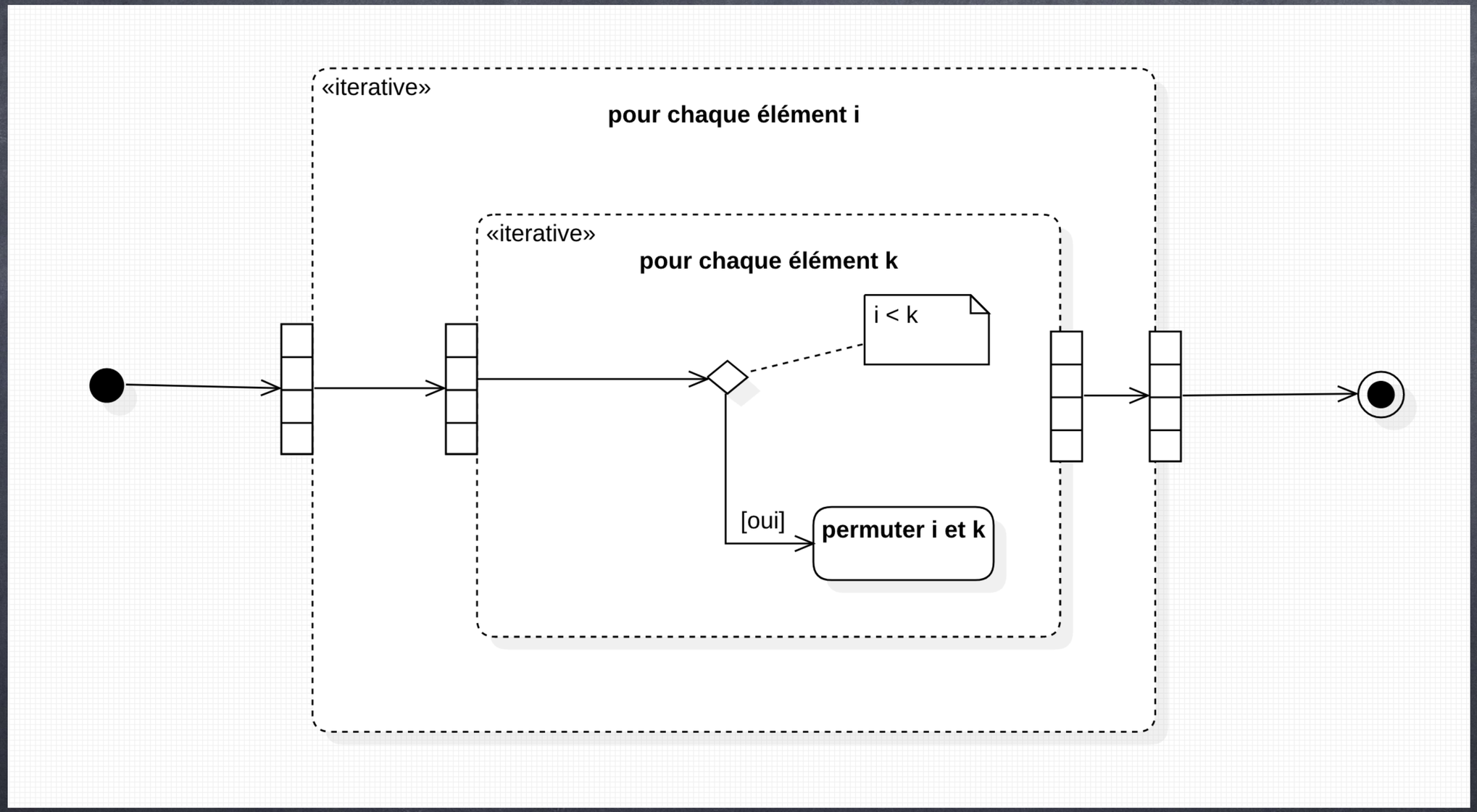


Diagramme de classe

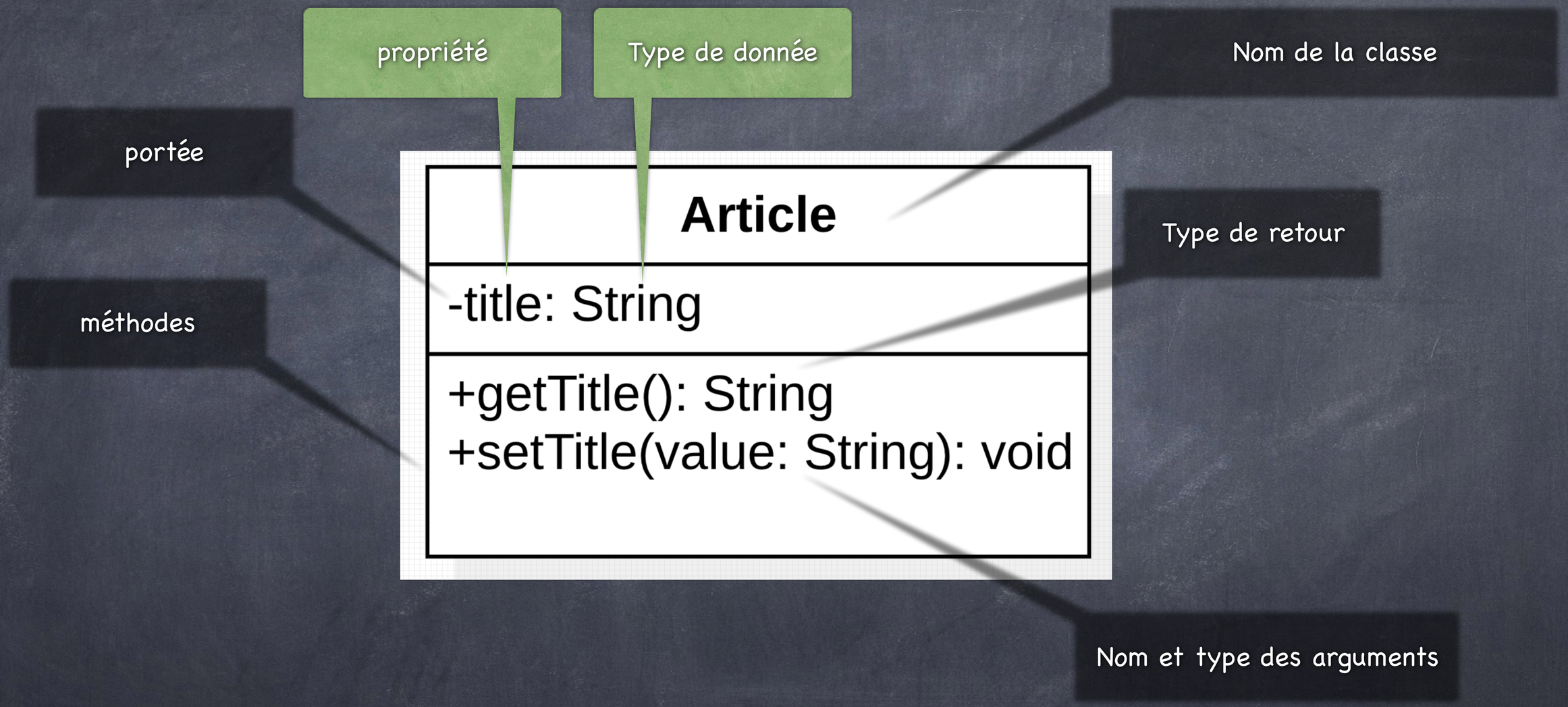
Décrit la structure d'un système orienté objet

Peut également représenter une structure de données

A quoi sert le diagramme de classe

- ✓ Décrire une structure de données
  - > alternative au Modèle Conceptuel de Données
- ✓ Décrire l'architecture statique d'une application codée en orienté objet
- ✓ Décrire un design pattern

UML Anatomie d'une classe



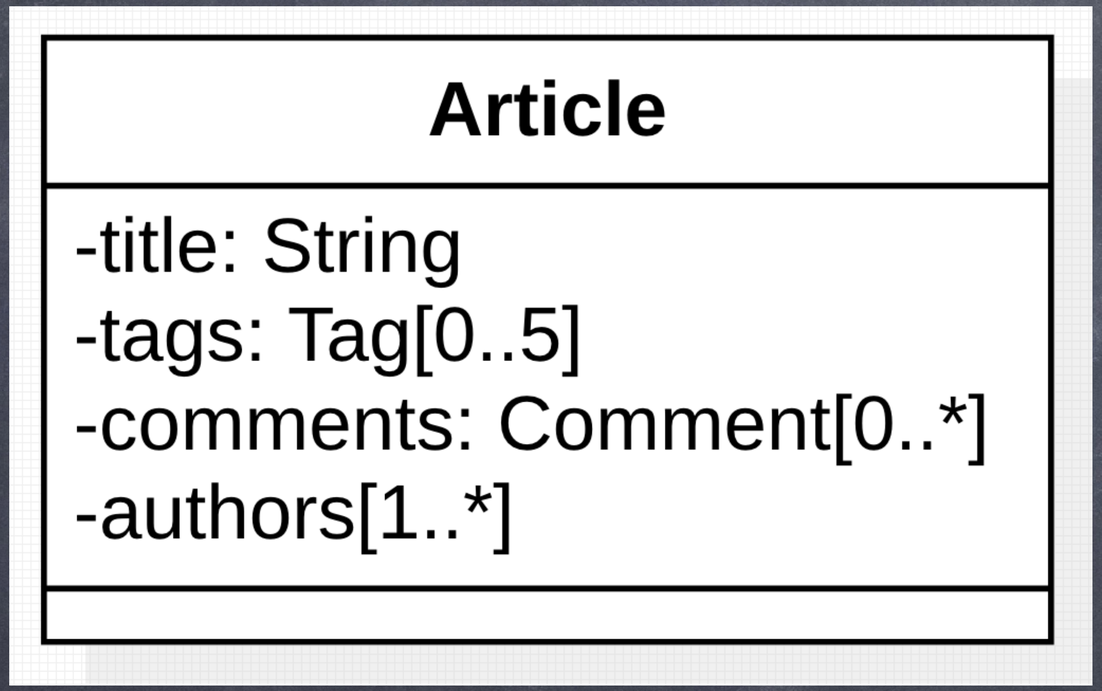
Typage

Type	Description	exemple
Scalaire	Types primitifs ex: String, Integer, Boolean	price: Float name: String
Array	Soit un objet Array soit un type suivit de crochets	tags: String[] students: Array prices: Array<Float>
Classe	Une classe est un type de données	author: Author studentList: Students[]

Contraintes

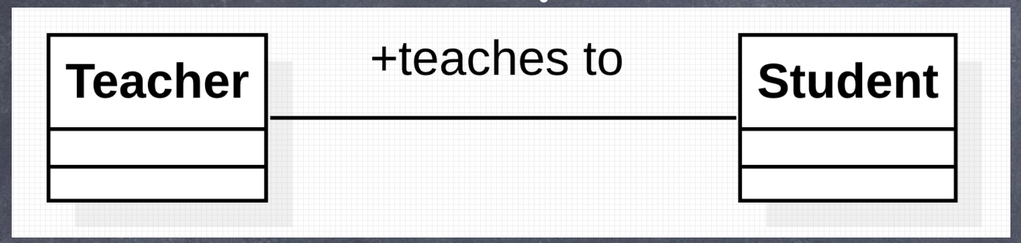
Type	Description	exemple
dérivée	La propriété n'est pas stockée mais dérive d'un calcul	/total: Float
readOnly	Propriété en lecture seule	birthDate: Date {readonly}
ordered	Les éléments d'un collection sont triés	studentList: Students[*] {ordered}

Multiplicité

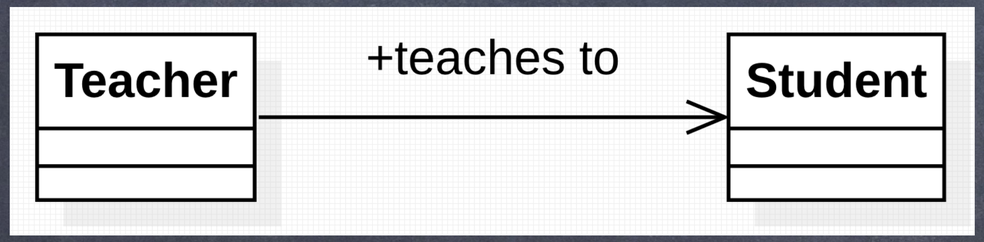


Associations

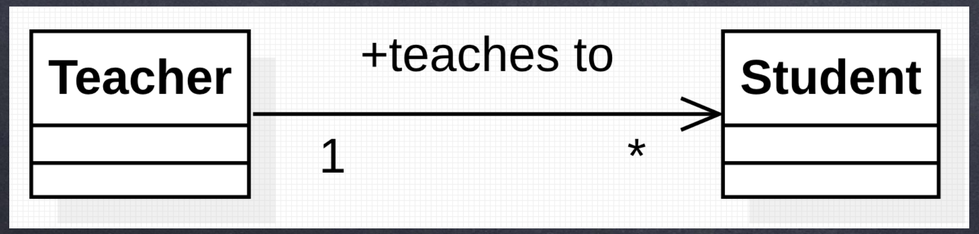
association simple



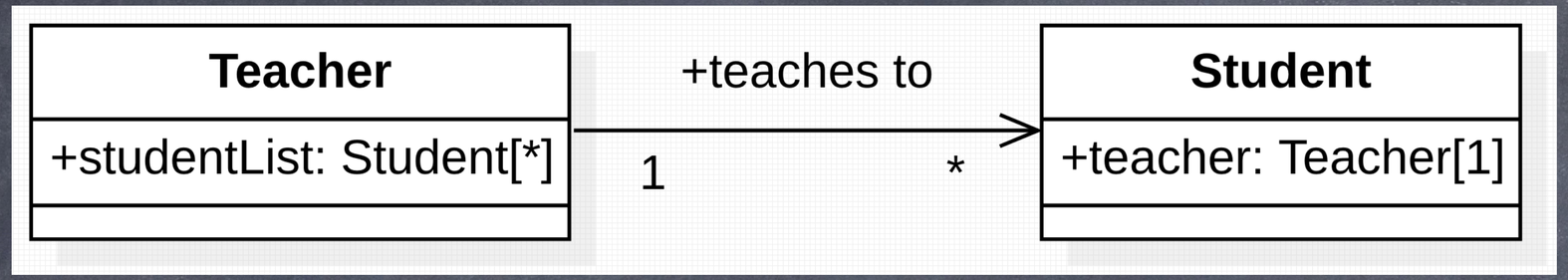
association dirigée

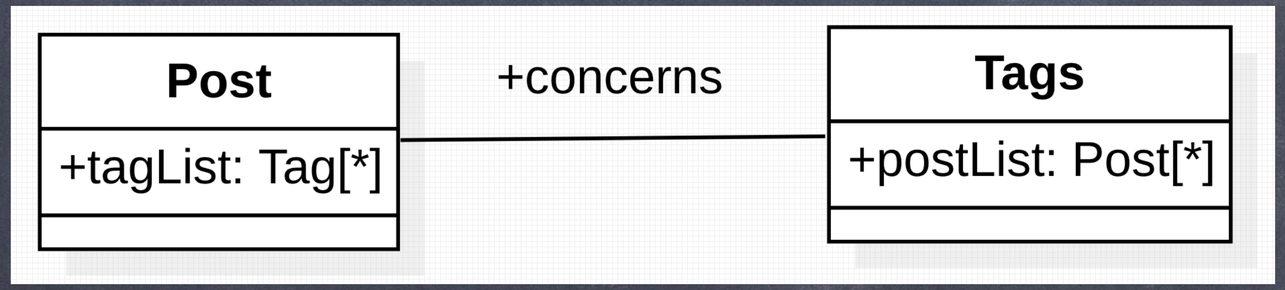


association dirigée avec multiplicités



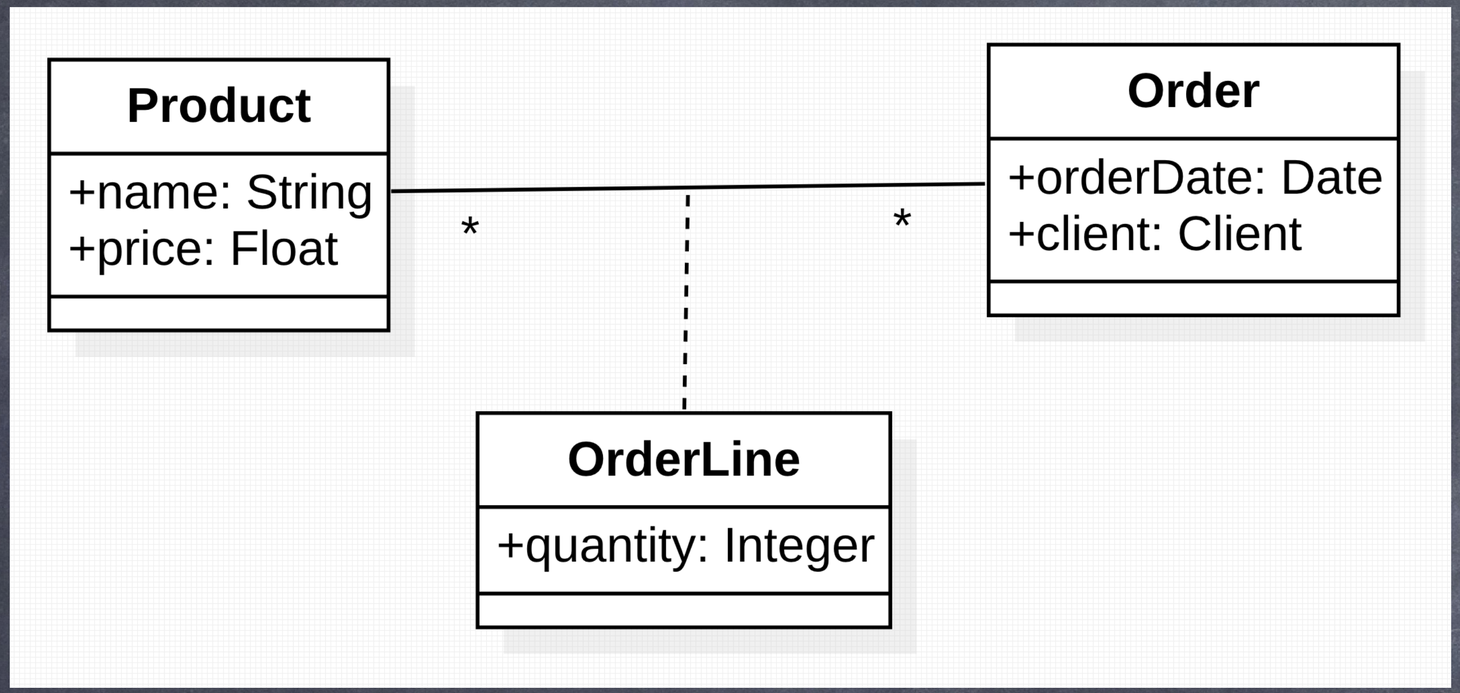
Associations et propriétés





l'association bidirectionnelle n'est pas une obligation, c'est au concepteur de déterminer si elle est judicieuse ou non.

Classe association



Exercice

- ✓ Une entreprise est composée de départements.
- ✓ Les départements sont situés dans un ou plusieurs bureaux.
- ✓ Un bureau fait office de siège social.
- ✓ Chaque département a un responsable qui est recruté parmi l'ensemble des employés.

Dessiner le diagramme de classe correspondant

Exercice

- ✓ Un étudiant peut s'inscrire à un cours.
- ✓ Un cours est donné par un professeur.
- ✓ Pour un cours donné le professeur attribue des devoirs à tous les étudiants inscrits.
- ✓ Pour chaque devoir le professeur attribue une note aux étudiants.

Dessiner le diagramme de classe correspondant

Exercice

- ✓ Un musicien est l'auteur de titres musicaux.
- ✓ Un musicien est l'auteur d'albums qui regroupent plusieurs titres.
- ✓ Un titre peut apparaitre dans plusieurs albums.
- ✓ A une combinaison de titre et d'album correspond un numéro de piste.

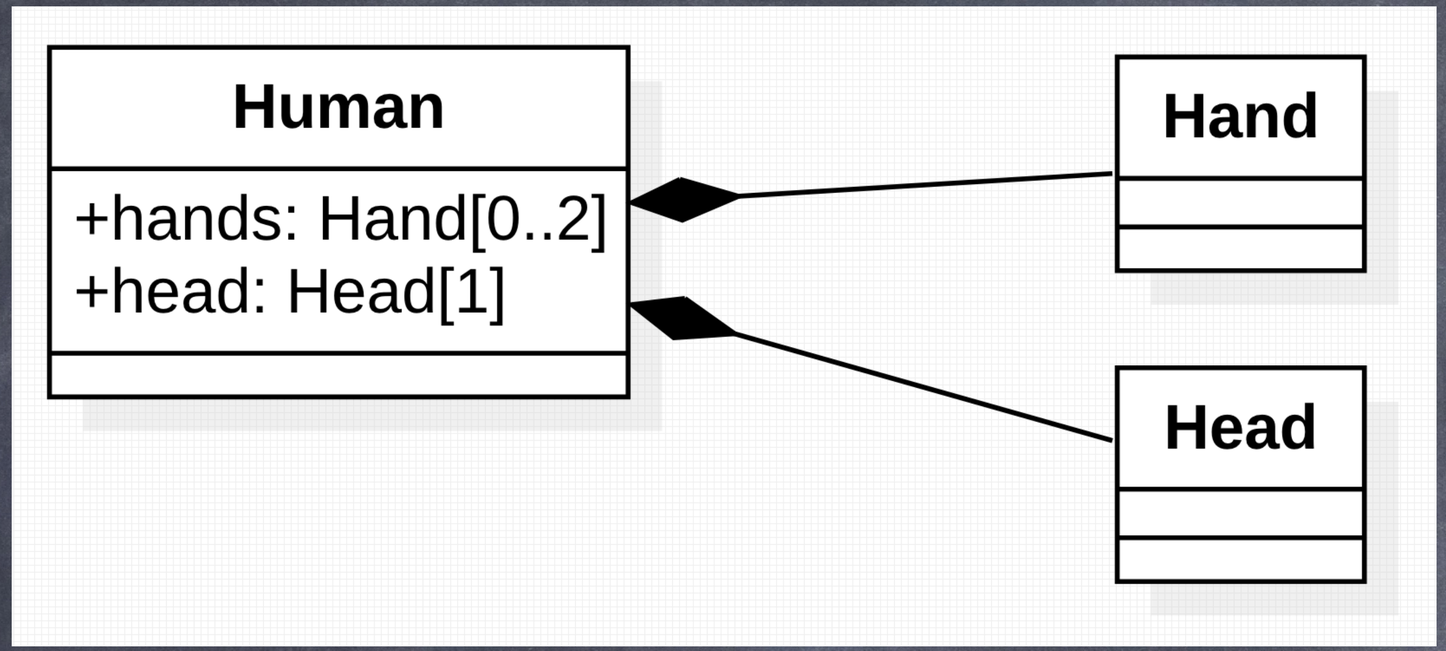
Dessiner le diagramme de classe correspondant

Exercice

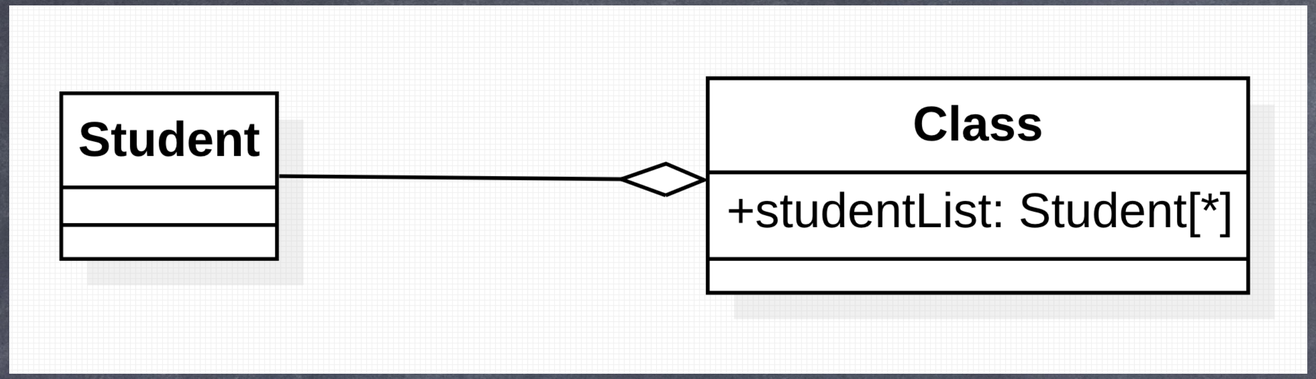
- ✓ Une course hippique a lieu sur un champ de course à une date donnée.
- ✓ Une course comprend un certains nombre de chevaux.
- ✓ Chaque cheval est monté par un jockey pour cette course.
- ✓ Pour une course, à un cheval et un jockey est associé un numéro de dossard.

Dessiner le diagramme de classe correspondant

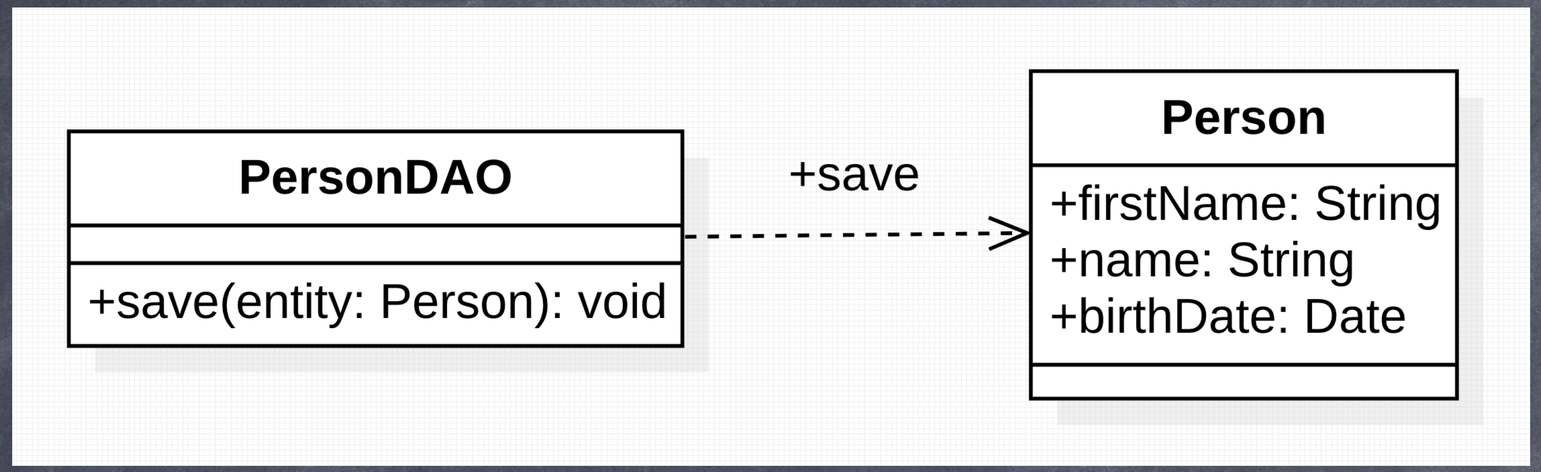
Composition



Agrégation



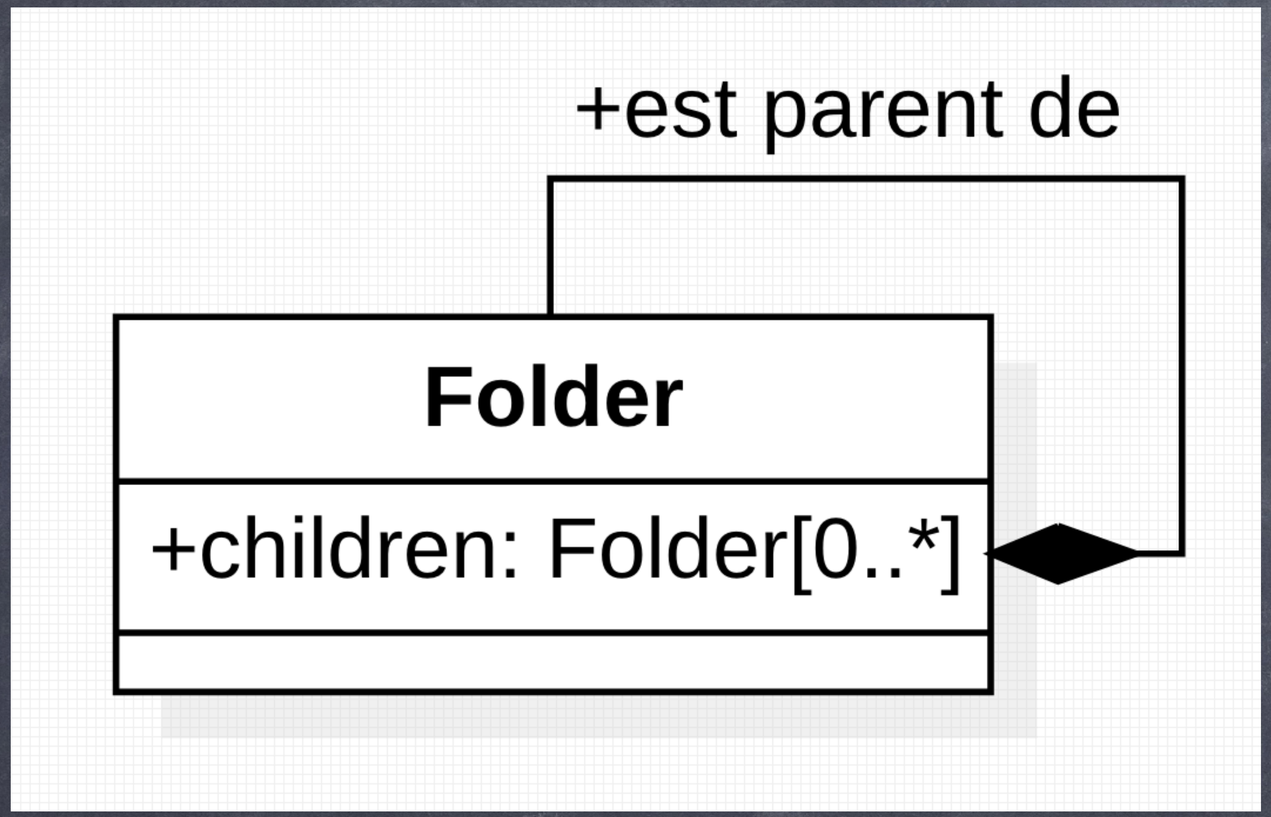
Dépendance



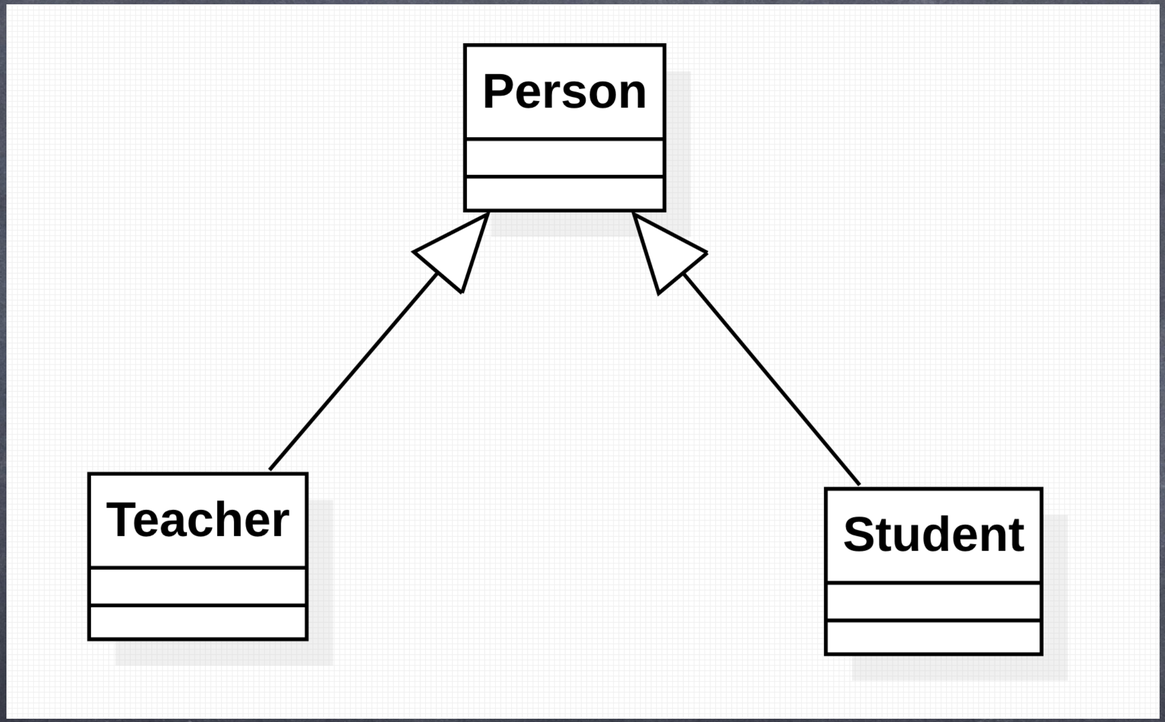
La dépendance n'est ni une composition, ni une agrégation car la classe dépendante ne stocke pas durablement l'instance dont elle dépend.

Ici par exemple, la méthode save a temporairement besoin d'une instance de Person

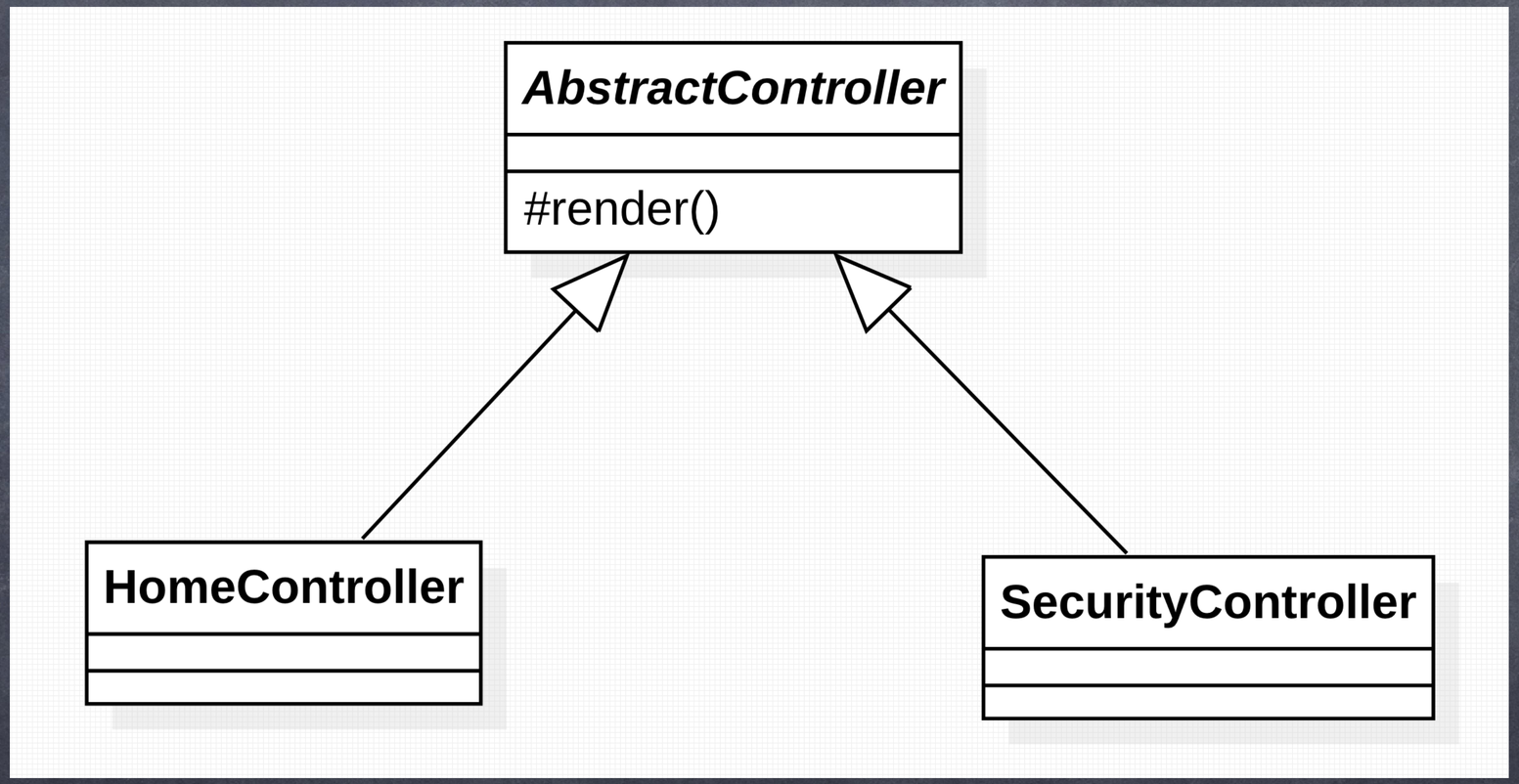
Auto association



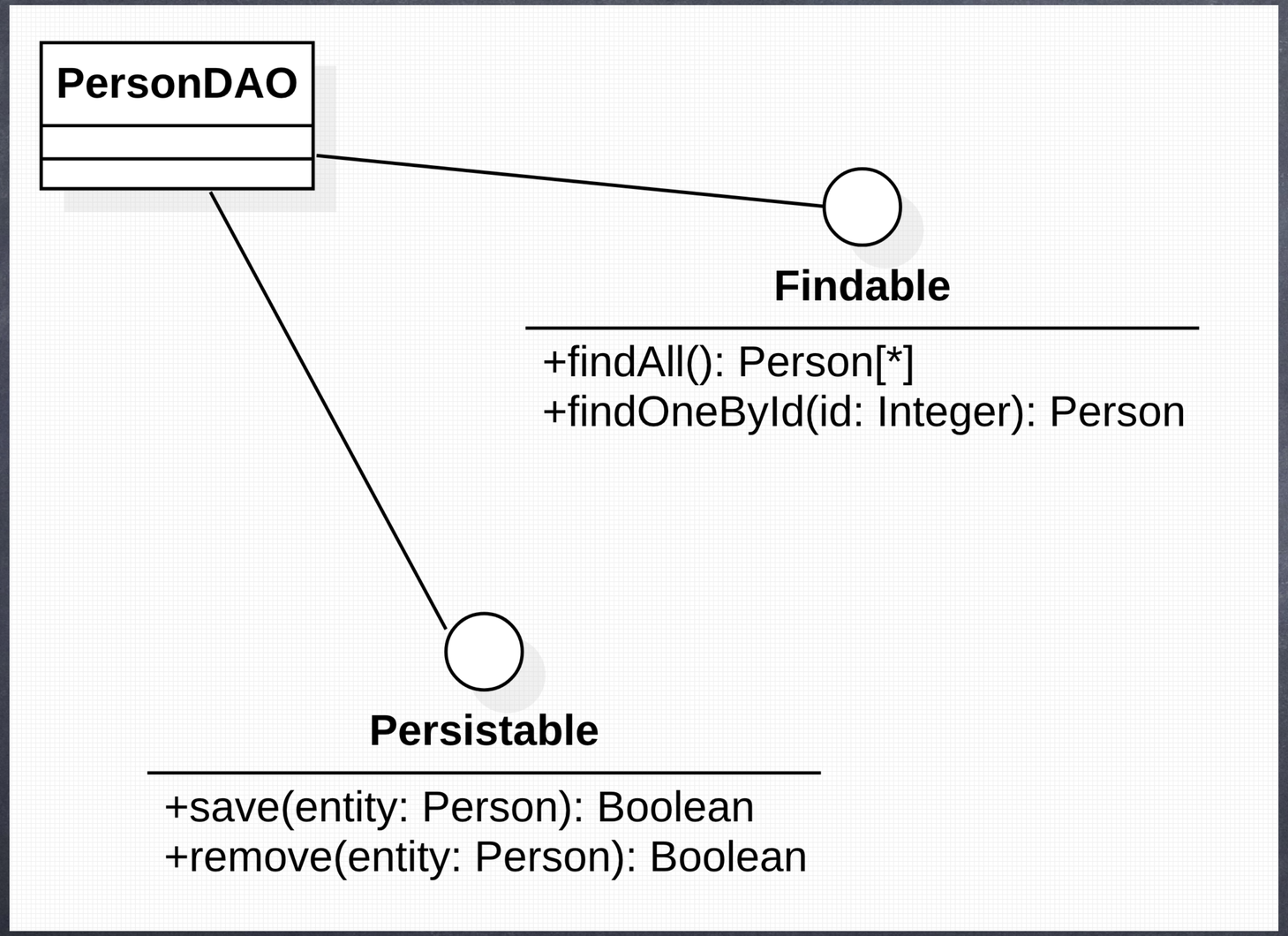
UML Héritage



UML Classe abstraite



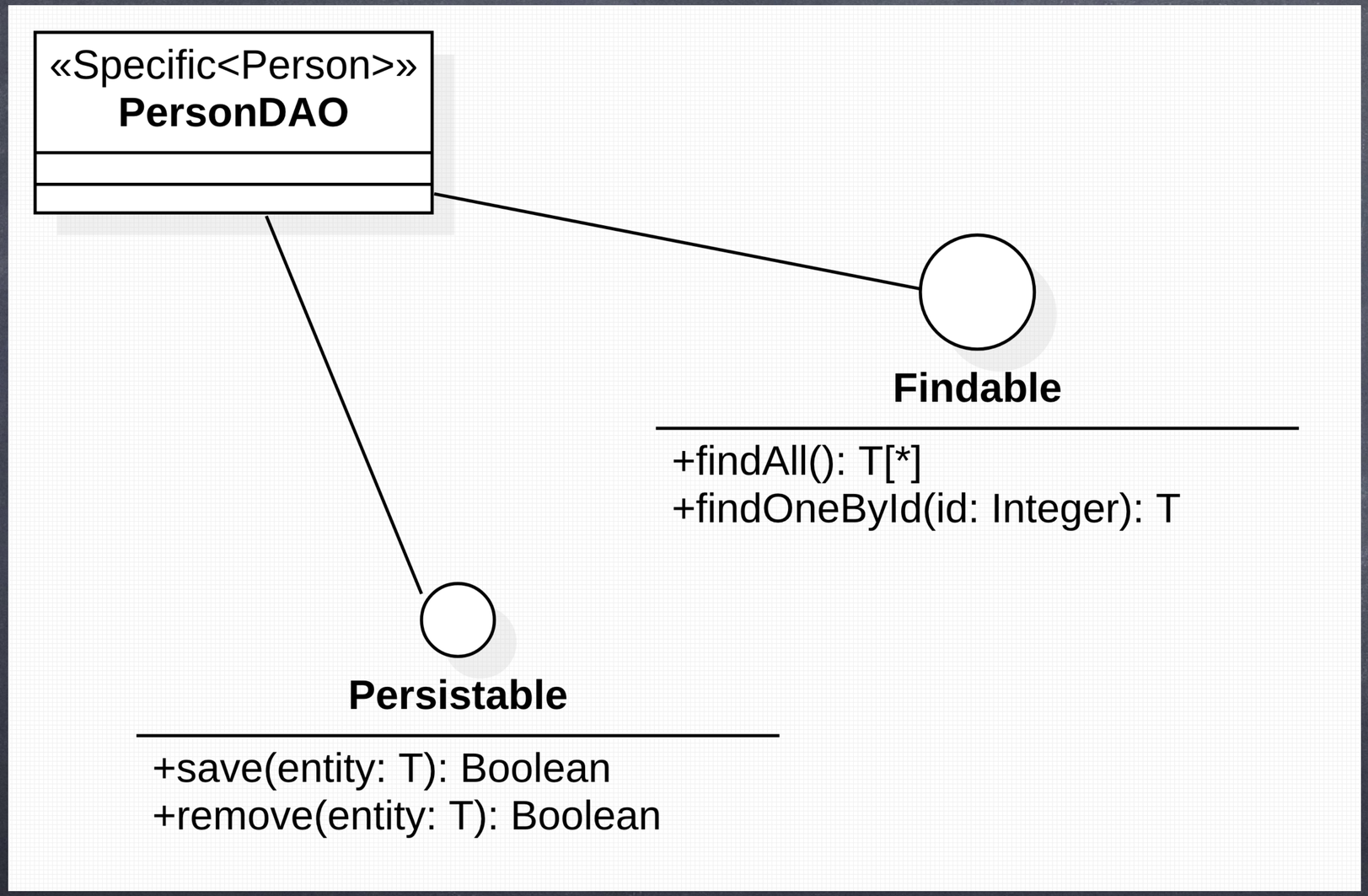
Interface



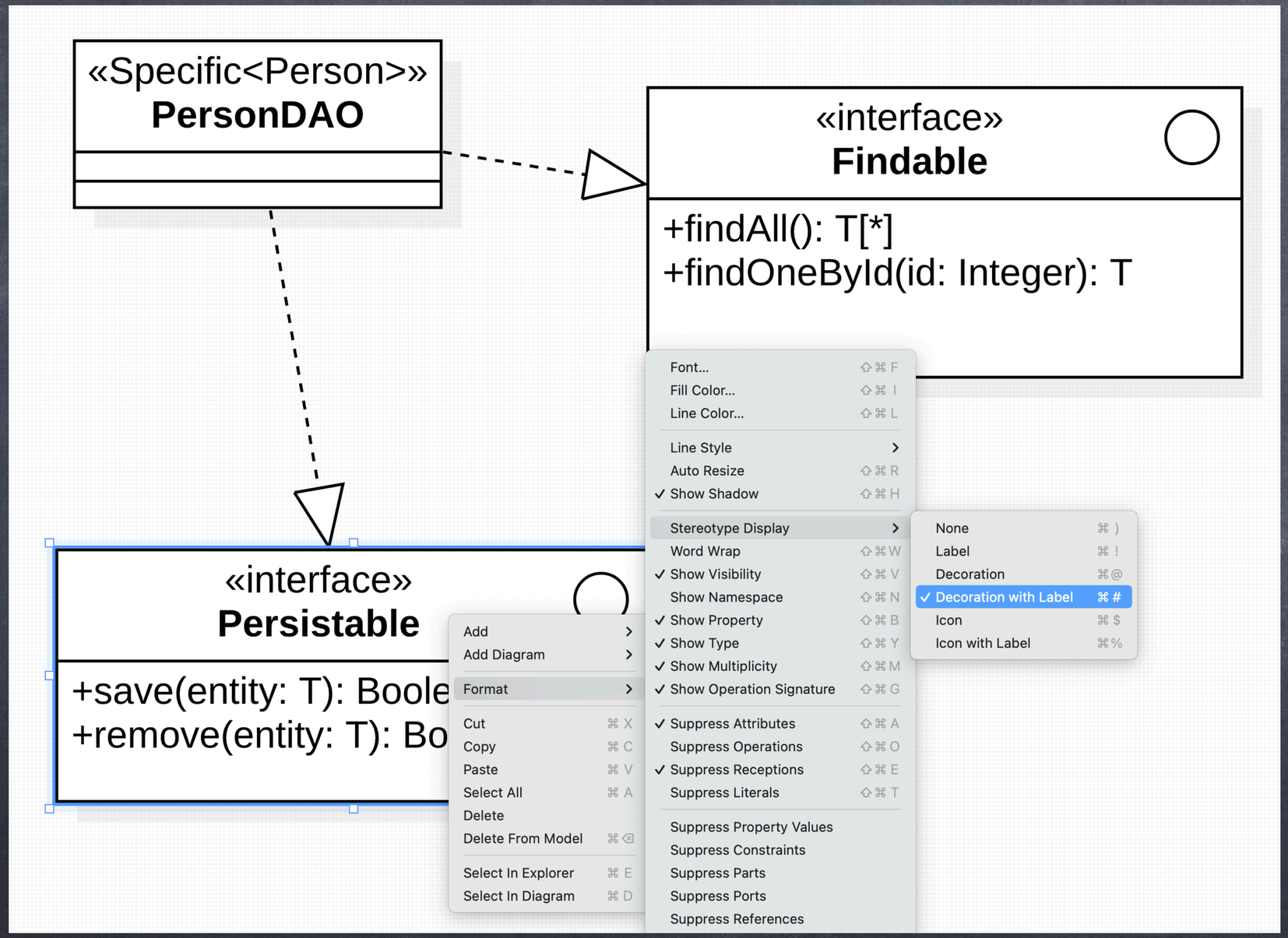
Que penser de ce diagramme ?

N'y a-t-il pas un souci quelque part ?

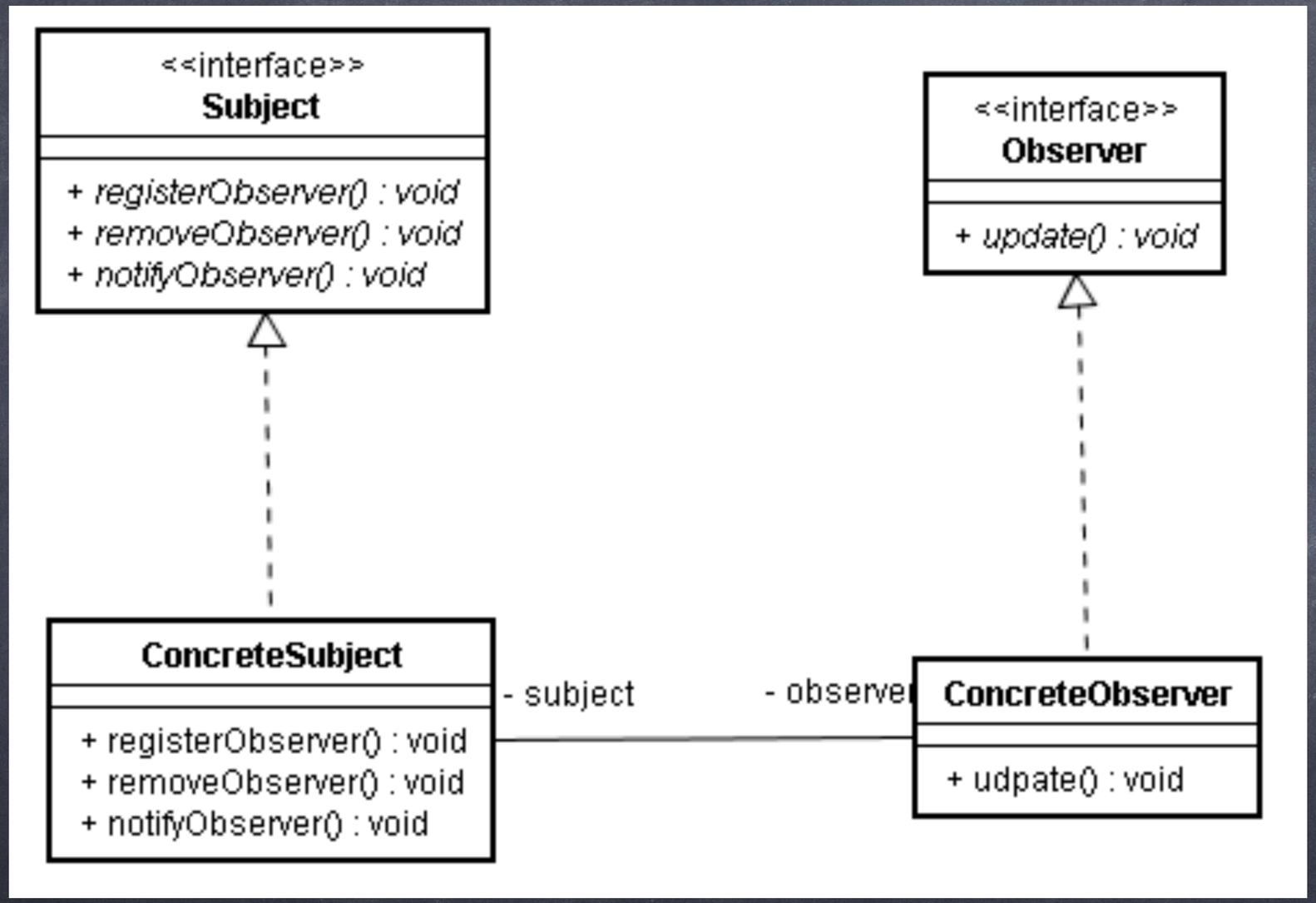
Une solution : Les generics



Notation alternative



Un pattern : Observer



Pattern Decorator

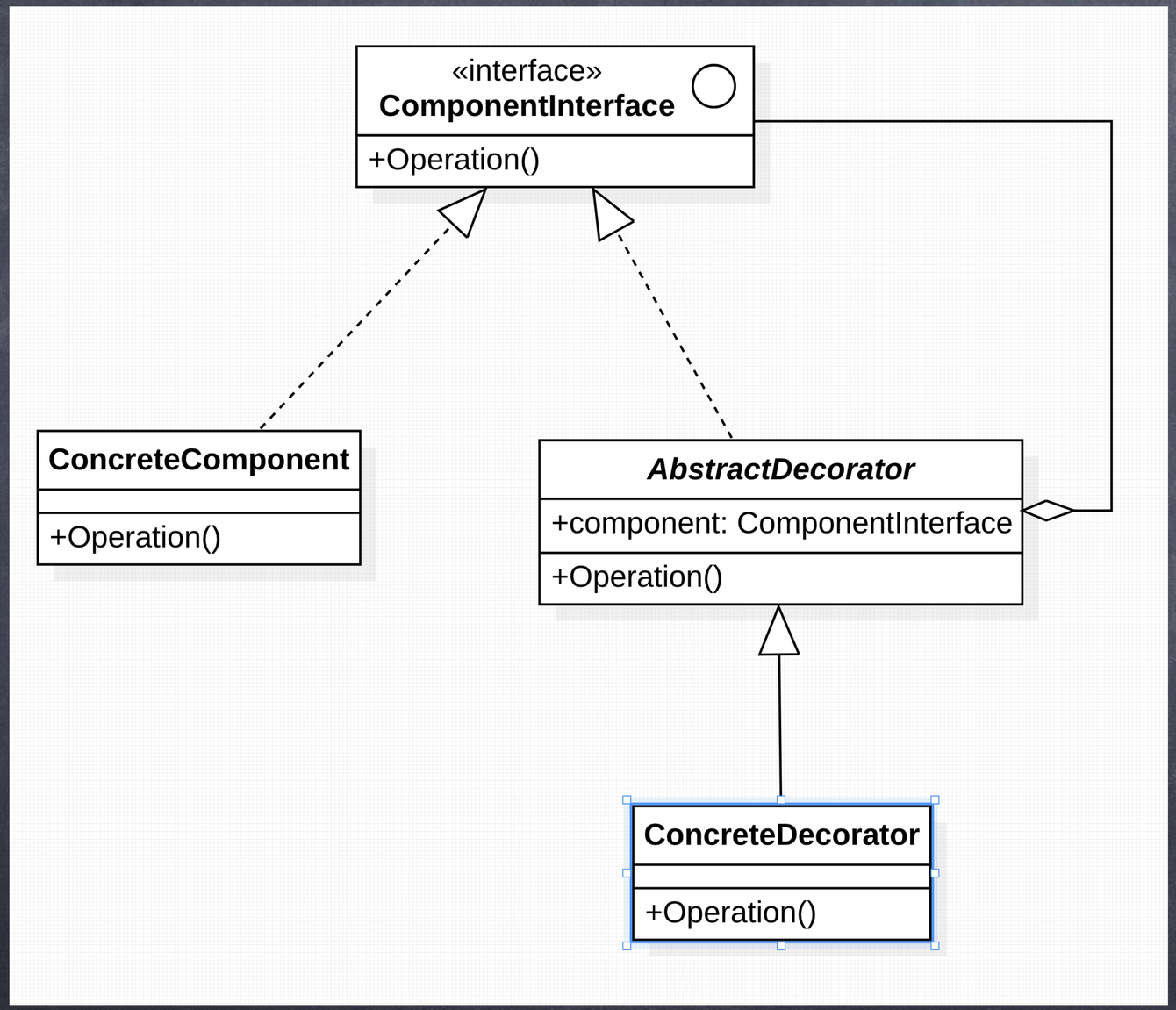
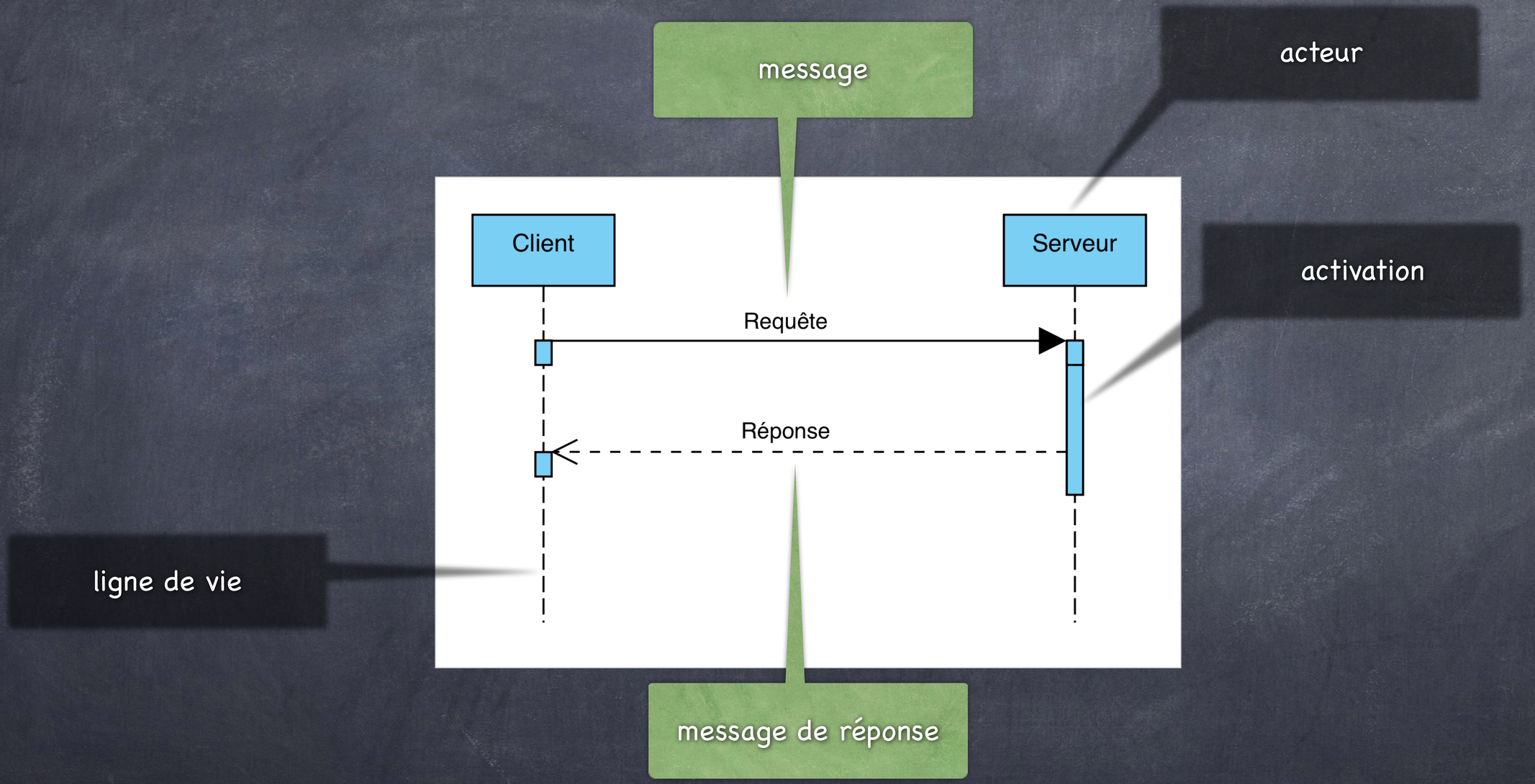


Diagramme de séquence

Illustre les interactions des acteurs d'un système

en se concentrant sur l'ordre des messages échangés

UML Les composants principaux



Deux types de diagrammes de séquence

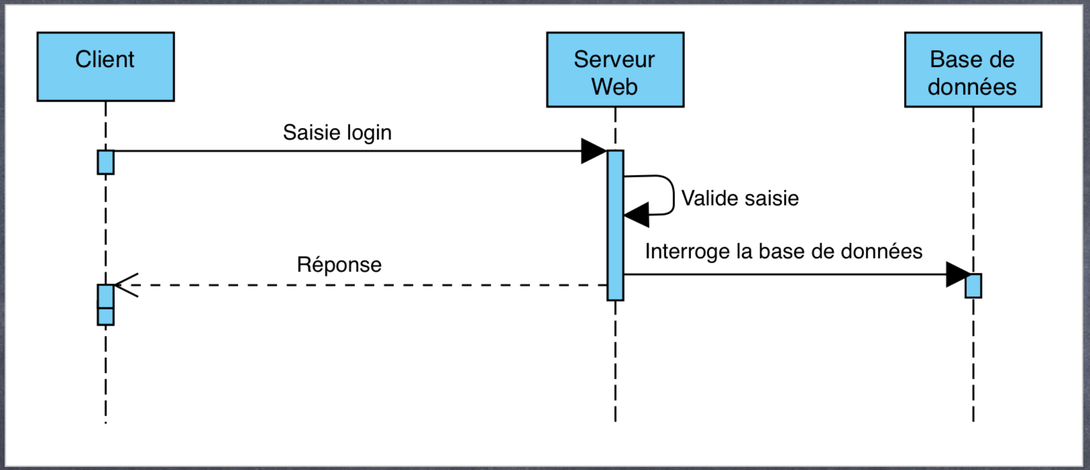
- ✓ Diagramme de séquence système

  Donne une vue d'ensemble des interactions entre les acteurs d'un système,
  le nombre d'acteurs est limité et ils ne sont pas typés

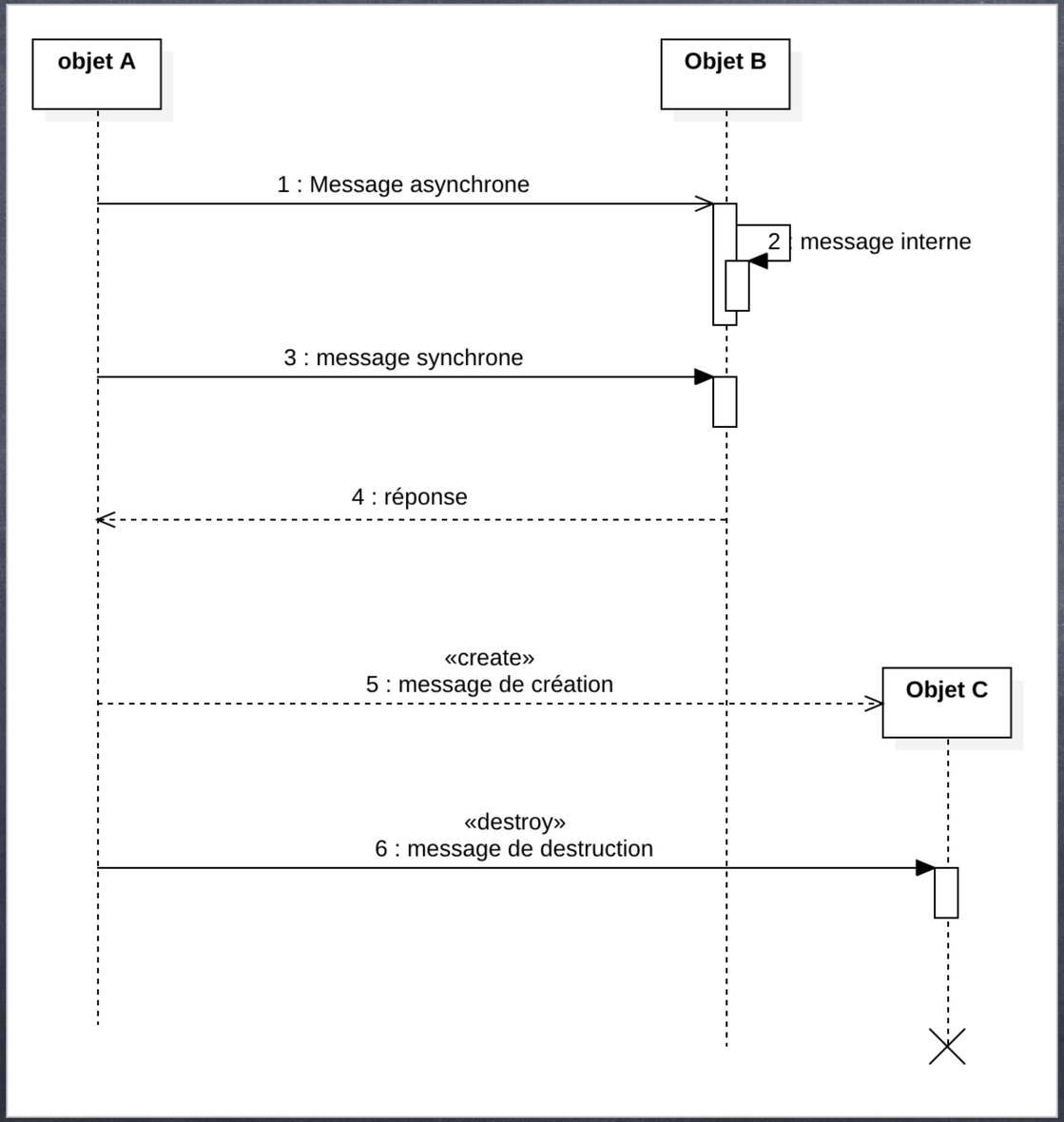
- ✓ Diagramme de séquence détaillé

  Détaille les interactions entre des composants d'un système voire entre des objets,
  dans ce dernier cas les messages sont souvent des appels de méthodes et les
  instances sont typées par leur classe.
  Souvent utilisé pour illustrer tout ou partie d'un cas d'utilisation

Exemple de diagramme système



UML Les messages



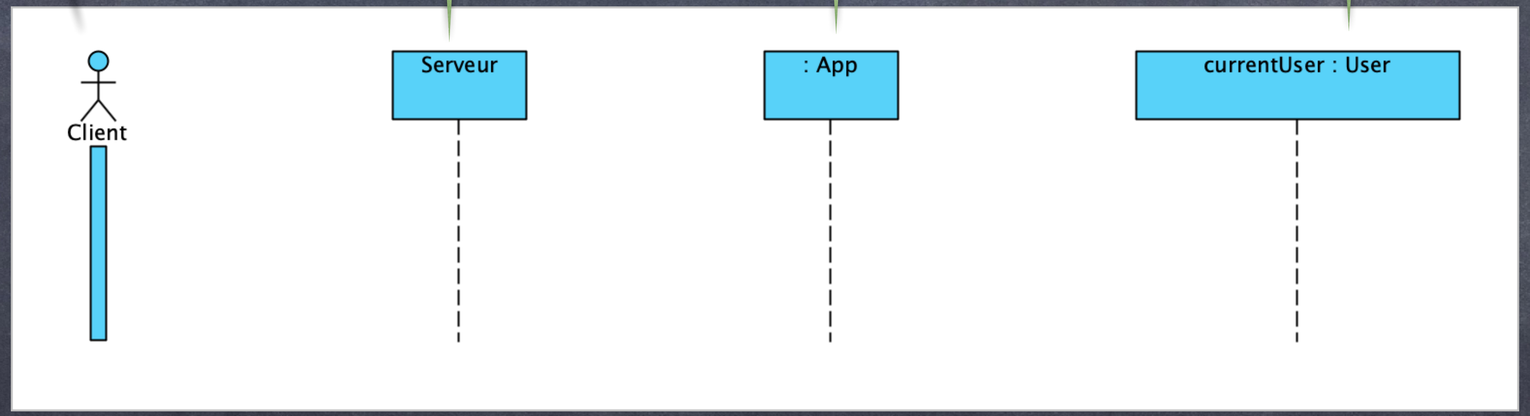
UML Notation des acteurs

un acteur humain

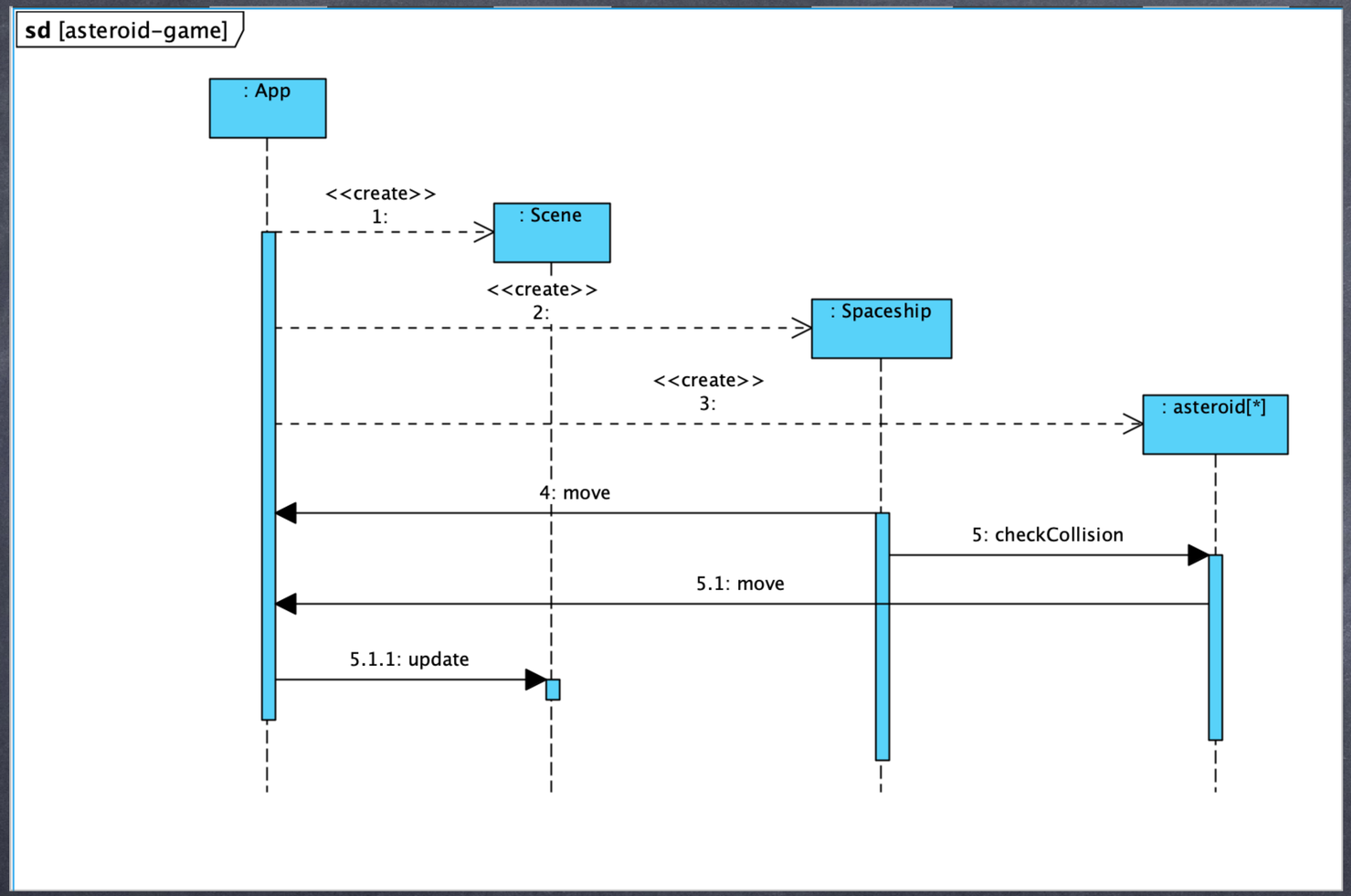
un acteur non typé

une instance anonyme de la classe App

une instance nommée de la classe User



Exemple : Le jeu asteroids



Quelles sont les informations transmises par ce diagramme ?

Exercice

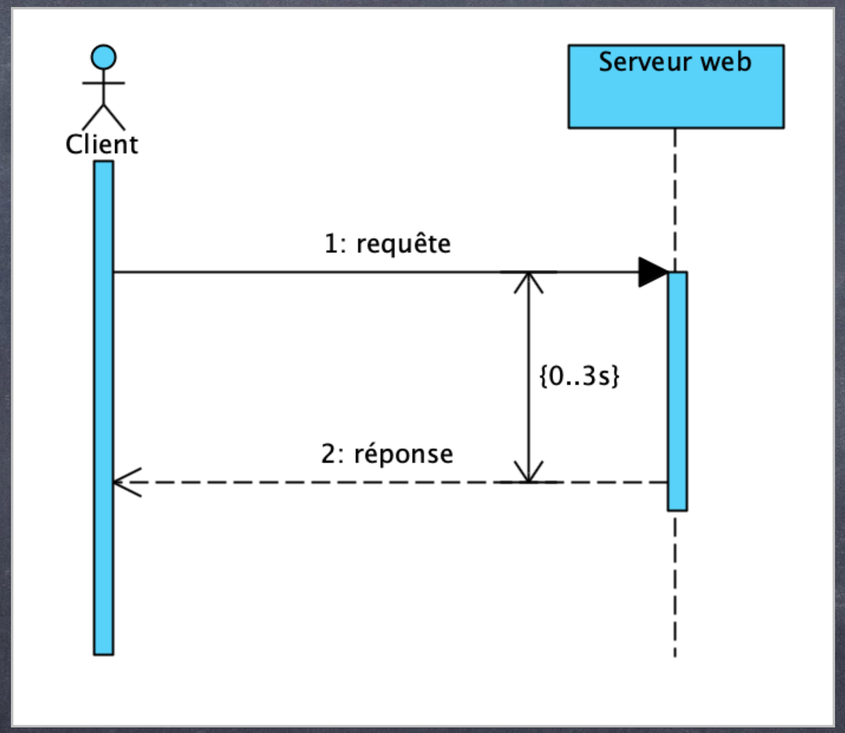
Réalisez un diagramme de séquence pour le scénario suivant :

- ✓ Un client se connecte à une application de commerce en ligne.
- ✓ Il parcourt les articles disponibles et ajoute un article au panier.
- ✓ Il passe la commande en confirmant les articles dans son panier.
- ✓ L'application envoie une demande de paiement au service de paiement.
- ✓ Une fois le paiement confirmé, un e-mail de confirmation est envoyé au client.

Commencez par lister les acteurs puis définissez les interactions
(échange de messages)

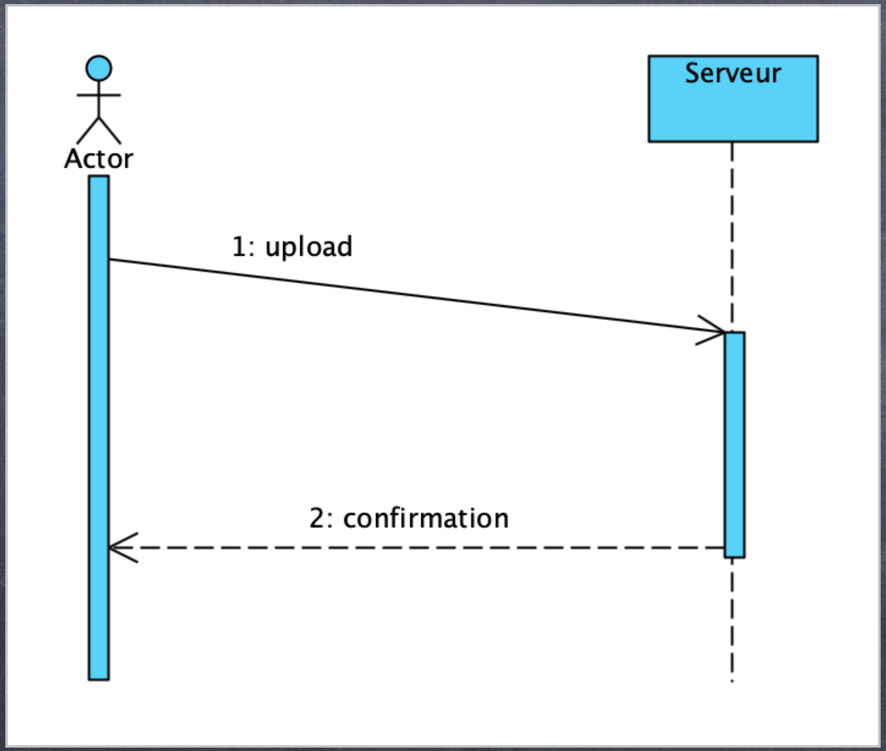
Duration constraint

Attribue une contrainte temporelle entre deux messages



Duration message

Indique qu'un message n'est pas instantané sans préciser de contrainte de temps



Combined fragments

modélise différentes possibilités d'exécution pour améliorer la précision et l'expressivité du diagramme

- ✓ **Alt (Alternative)**
  Utilisé pour représenter un choix entre plusieurs alternatives. Chaque branche d'un fragment "alt" est associée à une condition (un garde), et une seule des branches peut être exécutée. Par exemple, dans un système de validation de mot de passe, le fragment "alt" pourrait représenter deux chemins : un si le mot de passe est correct et un autre si le mot de passe est incorrect.

- ✓ **Opt (Optionnel)**
  Représente un comportement qui peut ou non se produire. Le fragment "opt" possède une condition de garde, et si cette condition est vraie, l'interaction dans le fragment est exécutée ; sinon, elle est ignorée. Cela est utile pour modéliser des cas où une action est facultative, par exemple, une notification facultative envoyée à l'utilisateur.

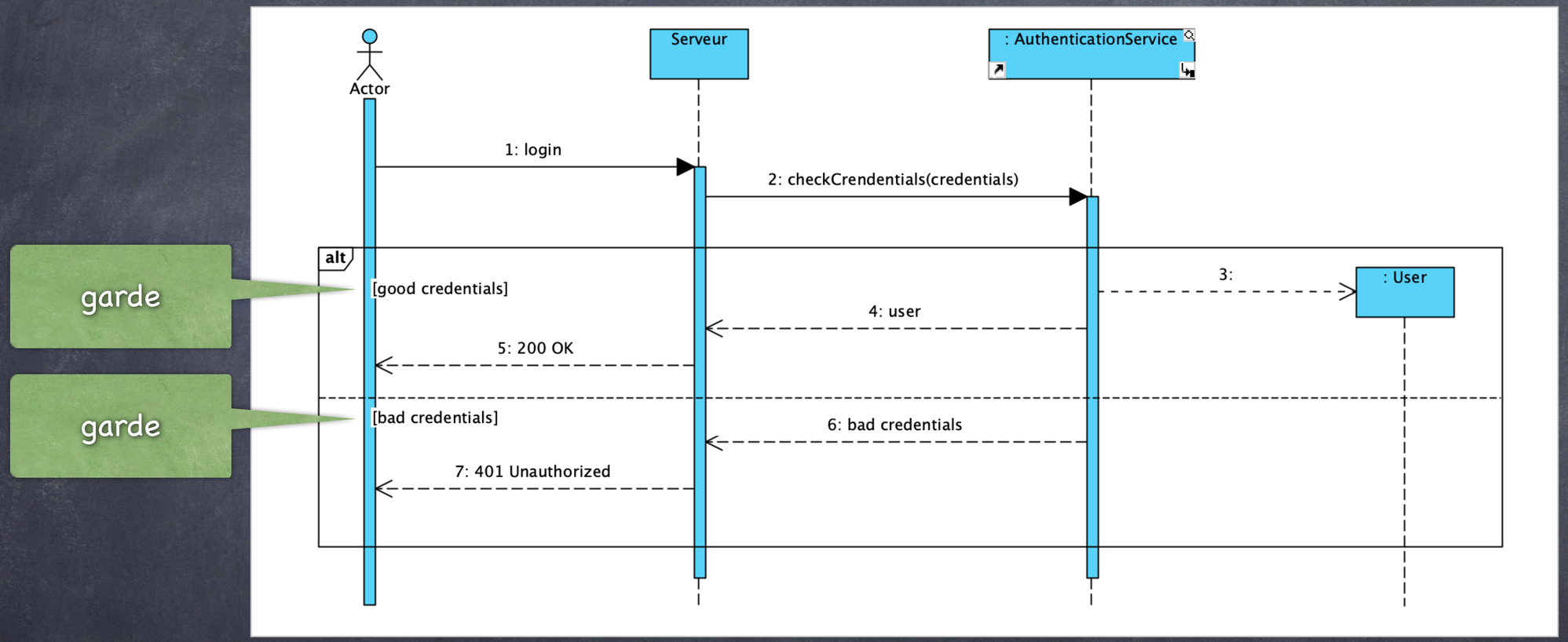
- ✓ **Break (sortie)**
  Représente l'interruption d'une boucle ou de l'ensemble de l'interaction. La condition de garde indique quand cette sortie peut avoir lieue.

- ✓ **Loop (Boucle)**
  Utilisé pour modéliser une séquence d'interactions répétée. La condition de garde spécifie le nombre de répétitions ou la condition sous laquelle le fragment de boucle est exécuté. Par exemple, pour une séquence d'essais de connexion, le fragment "loop" peut représenter plusieurs tentatives jusqu'à un nombre maximal.

- ✓ **Par (Parallèle)**
  Permet de modéliser des interactions qui se déroulent en parallèle. Chaque branche du fragment "par" est exécutée simultanément. C'est utile pour les processus qui s'exécutent en parallèle, comme les téléchargements multiples.

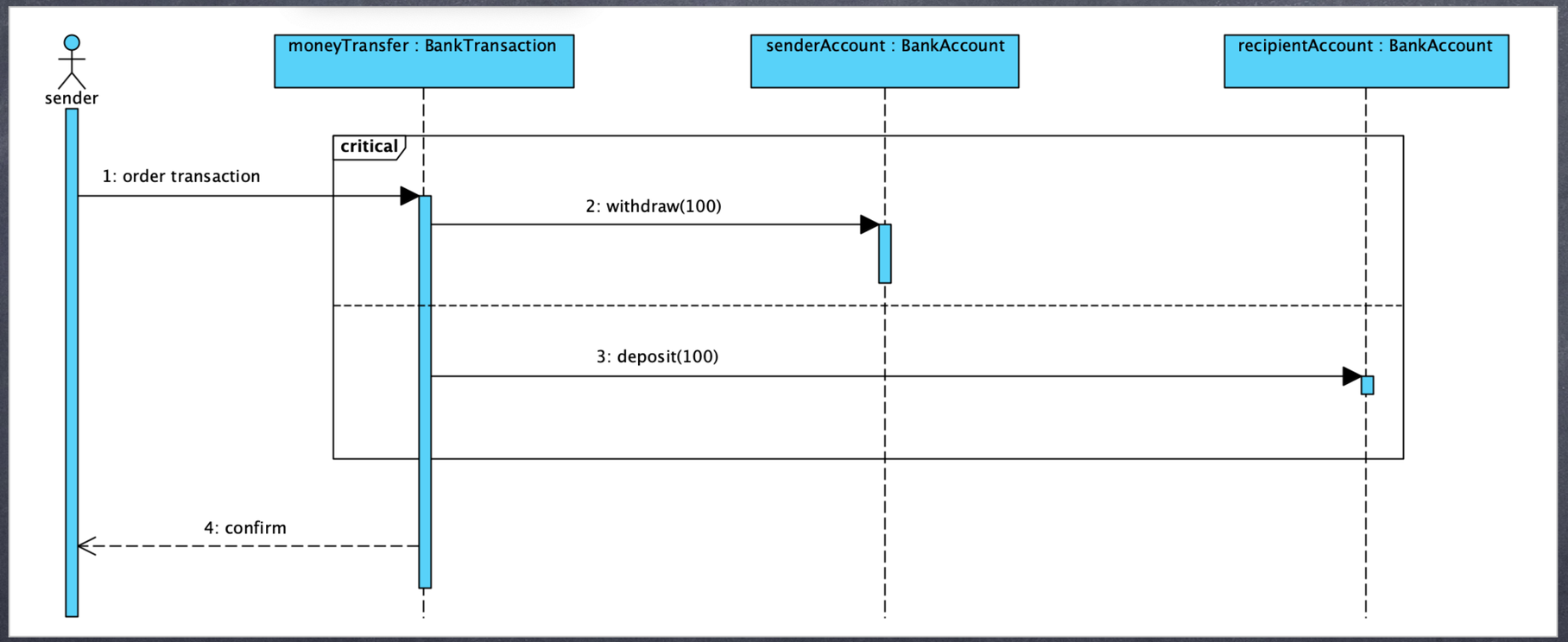
- ✓ **Critical (Critique)**
  Représente une section critique de l'interaction, c'est-à-dire une portion qui doit être exécutée sans interruption. Le fragment "critical" assure que les messages à l'intérieur de ce bloc sont traités de manière atomique.

Exemple d'alternative



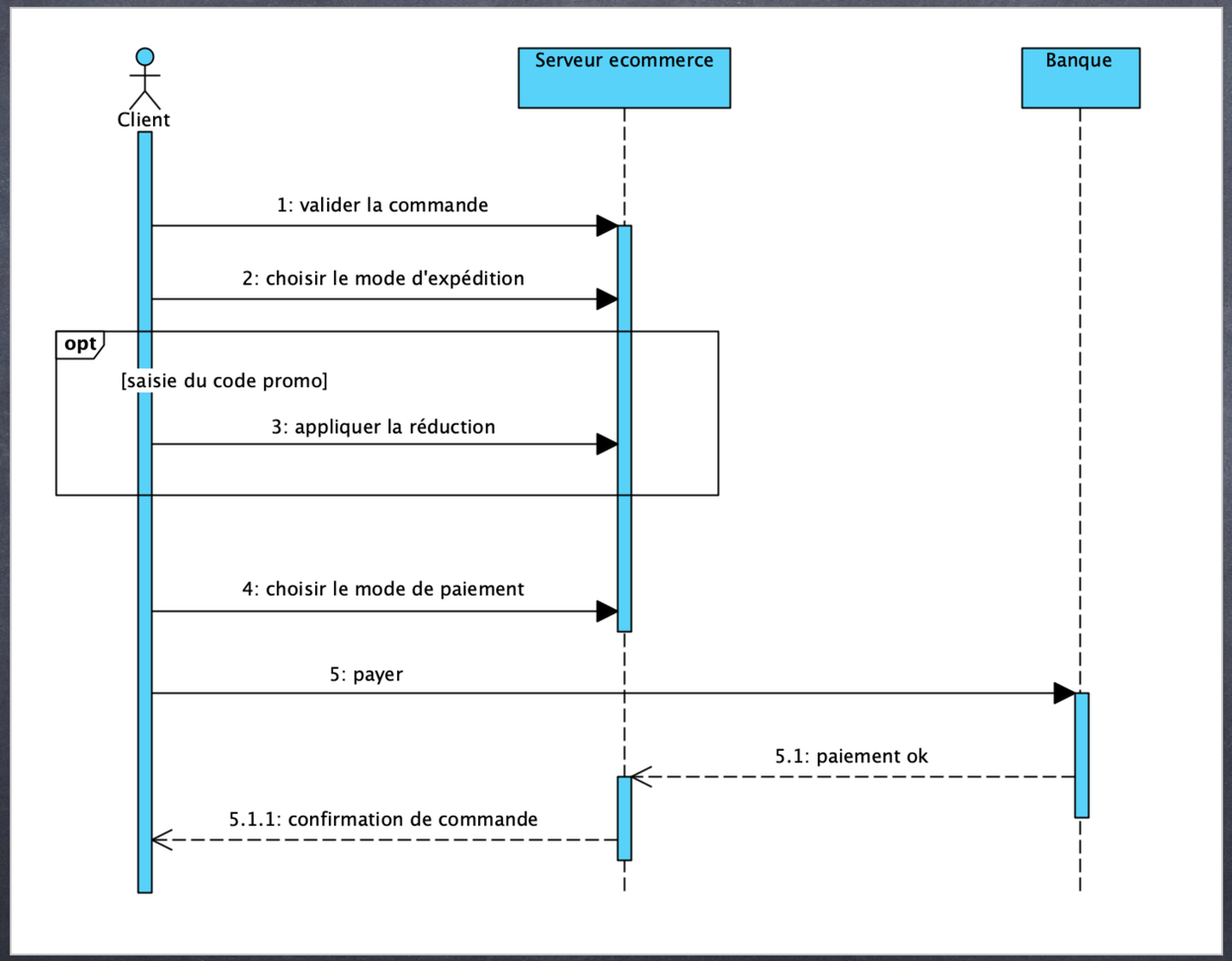
Exemple de séquence critique

Si une opération critique échoue, l'ensemble est annulé

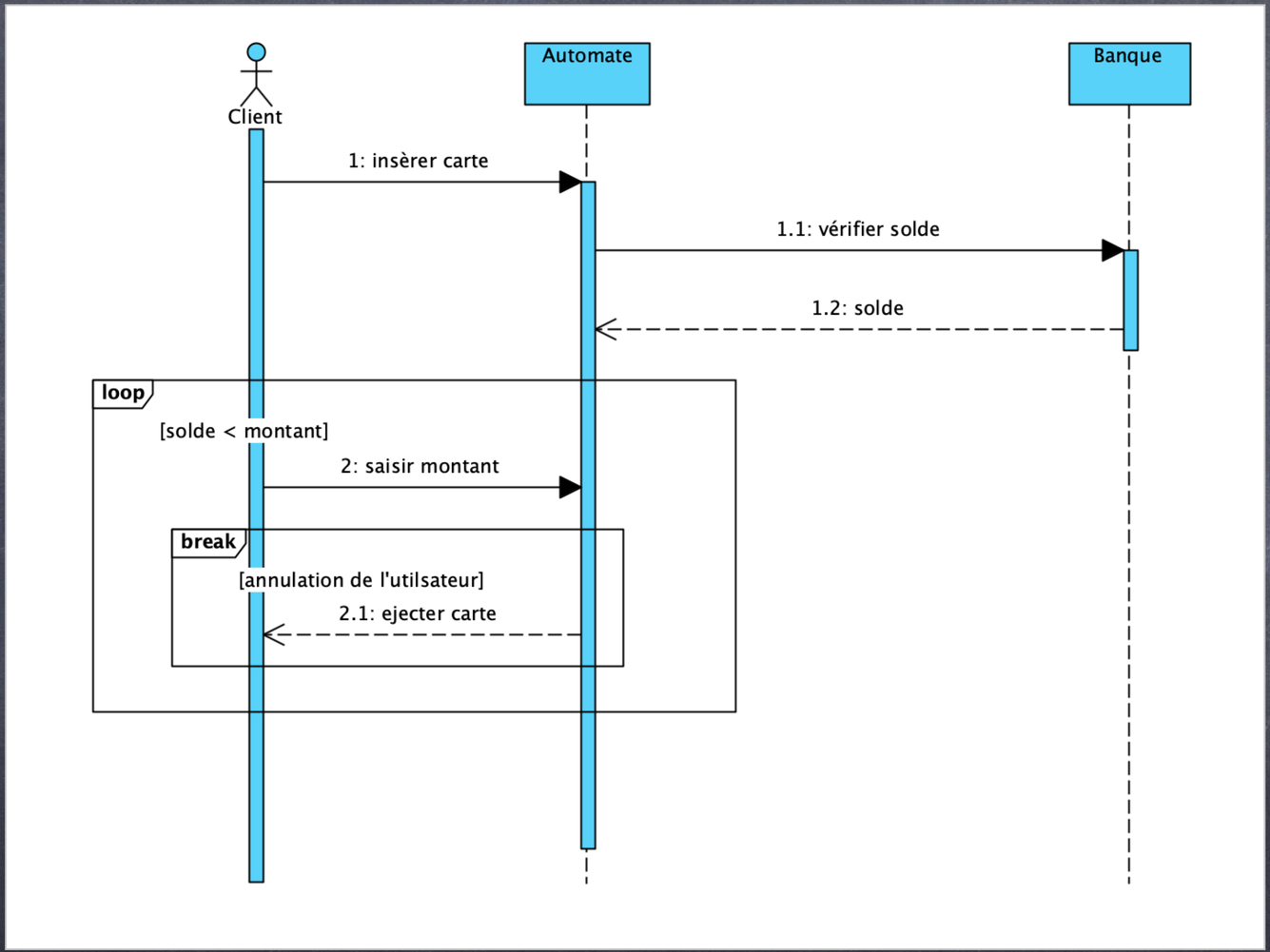


Exemple de séquence optionnelle

Une sorte de alt sans clause else



UML Exemple de séquence itérative avec sortie



Exemple de séquence avec traitements en parallèle

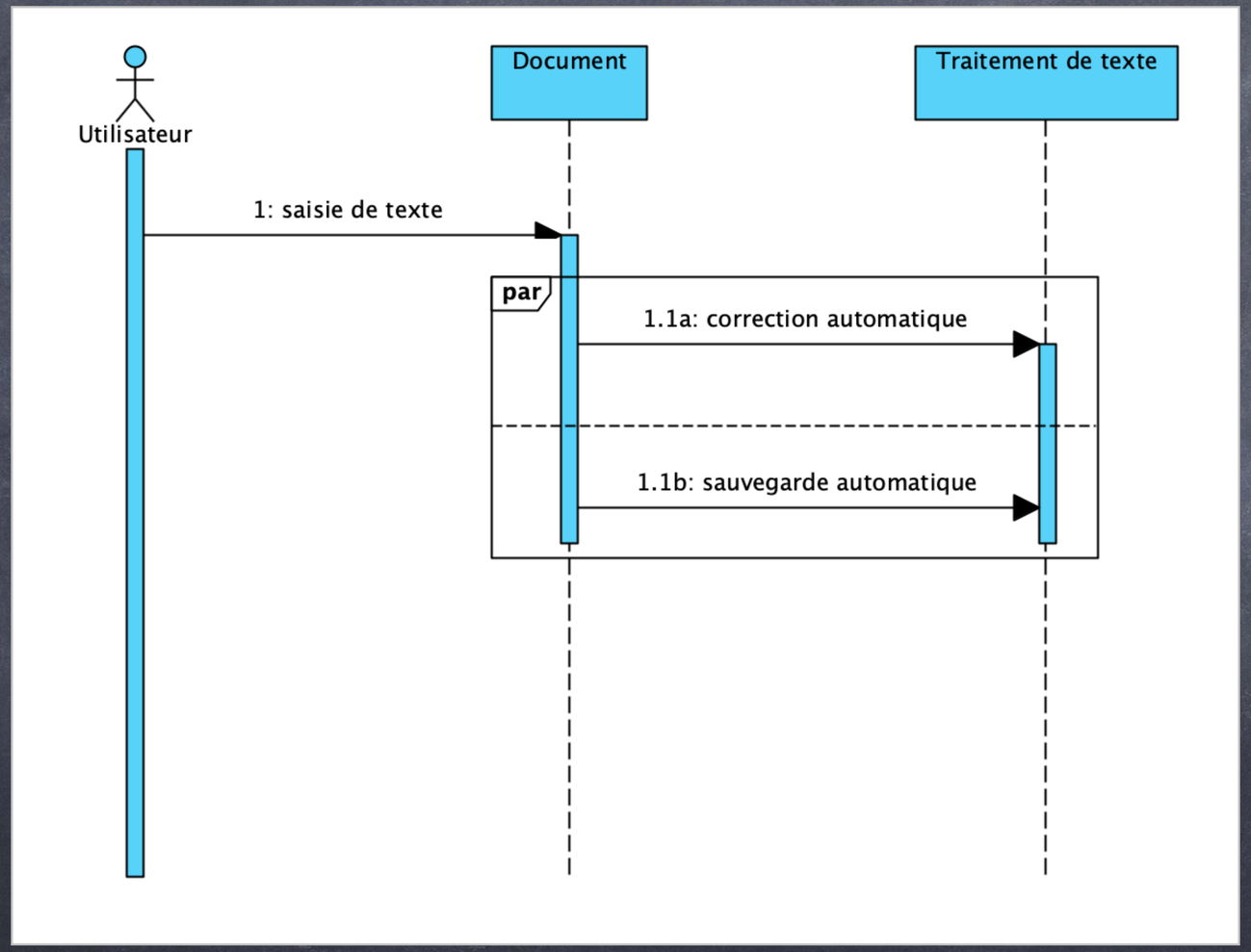
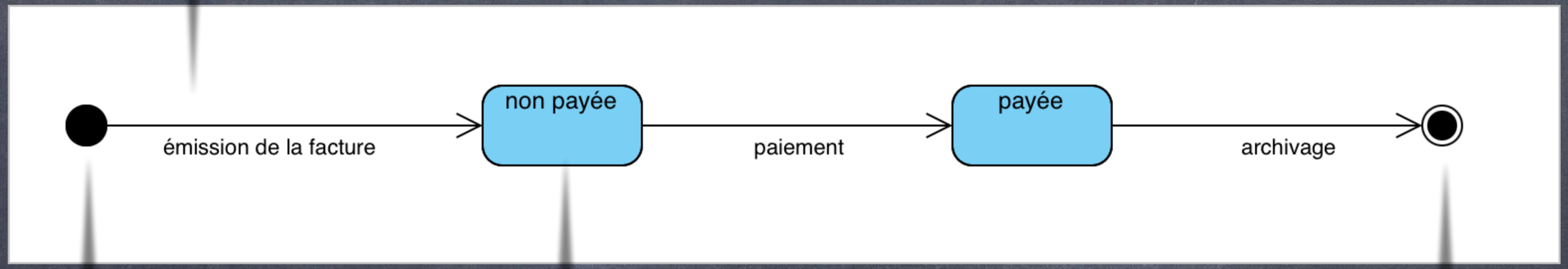


Diagramme d'état-transition

Illustre les différents états d'un système, d'un objet ou d'une variable ainsi que les condition du passage d'un état à l'autre

Un exemple

événement / transition

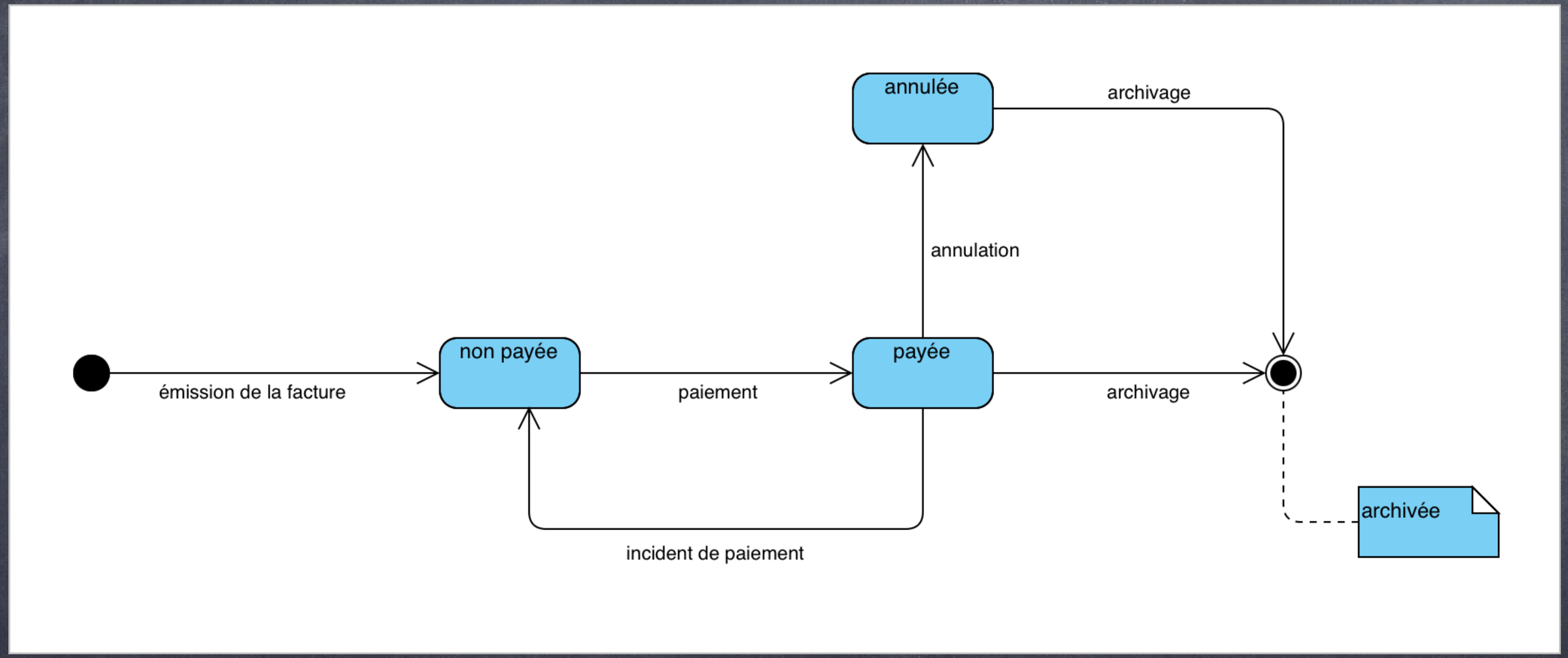


état initial

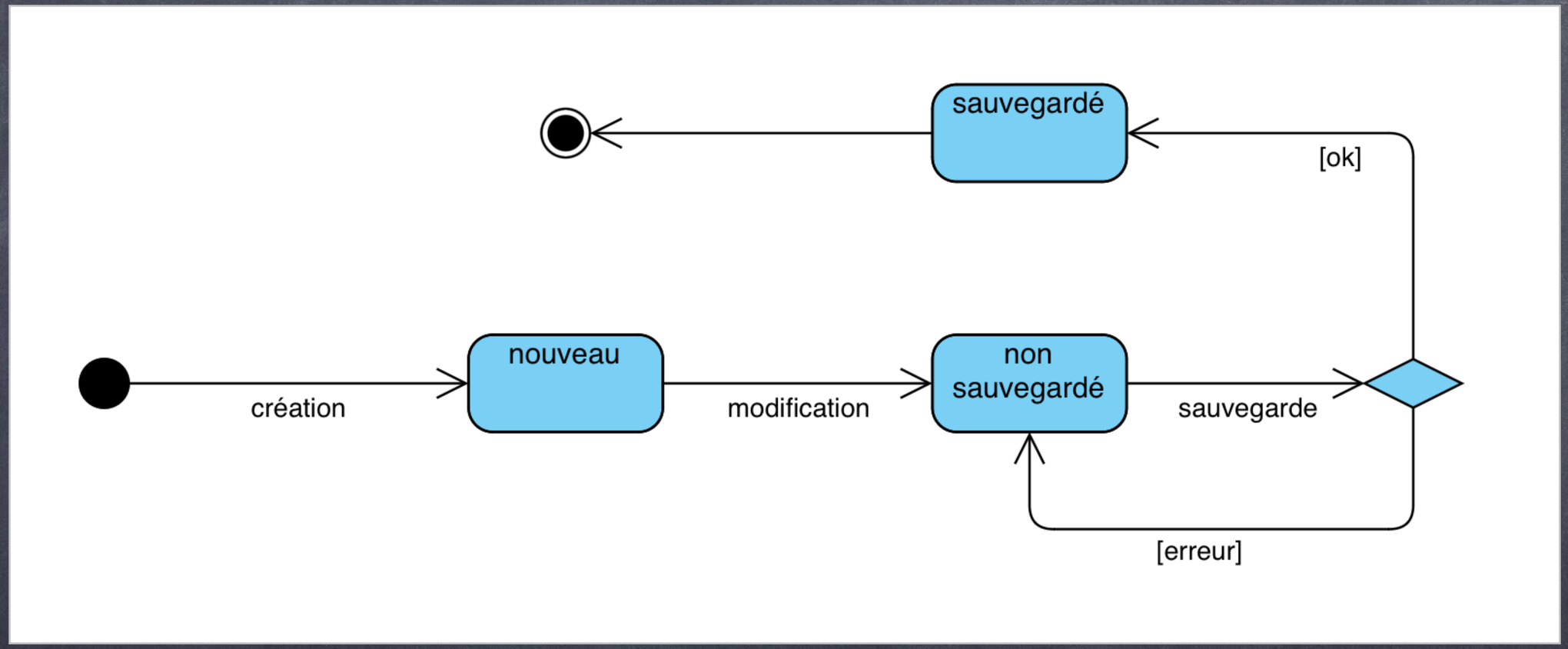
état

état final

Exemple plus complet



Condition


sauvegardé
[ok]
nouveau
non sauvegardé
création
modification
sauvegarde
[erreur]

Les transitions

- ✓ déclencheur (trigger)
- ✓ garde (condition)
- ✓ action

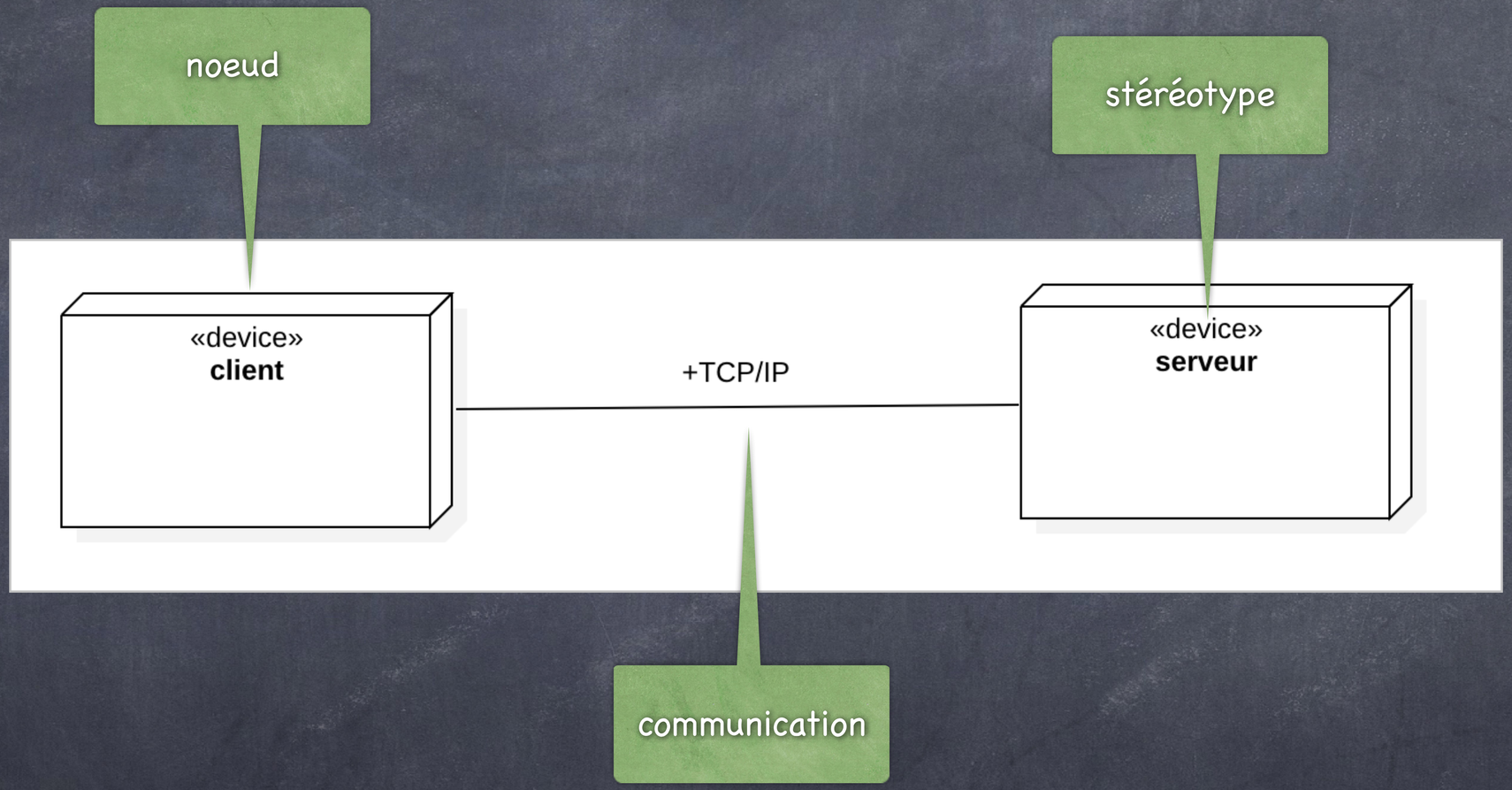
déclencheur [garde] / action

saisie [nombre de caractère > 3]/recherche

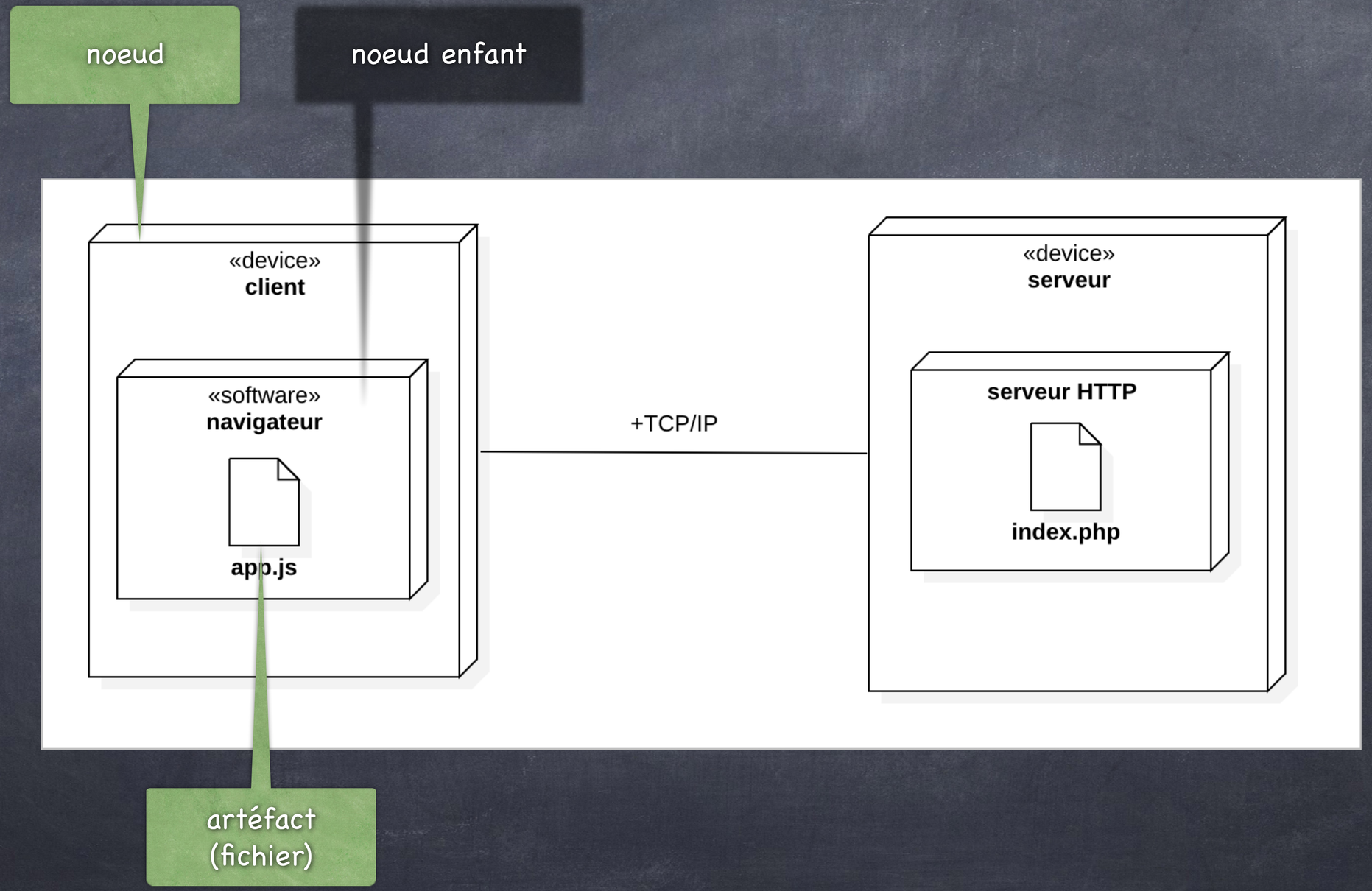
Diagramme de déploiement

Vue de l'infrastructure technique d'exécution d'une application et de la distribution des composants et des artéfacts entre les différents noeuds de cette architecture

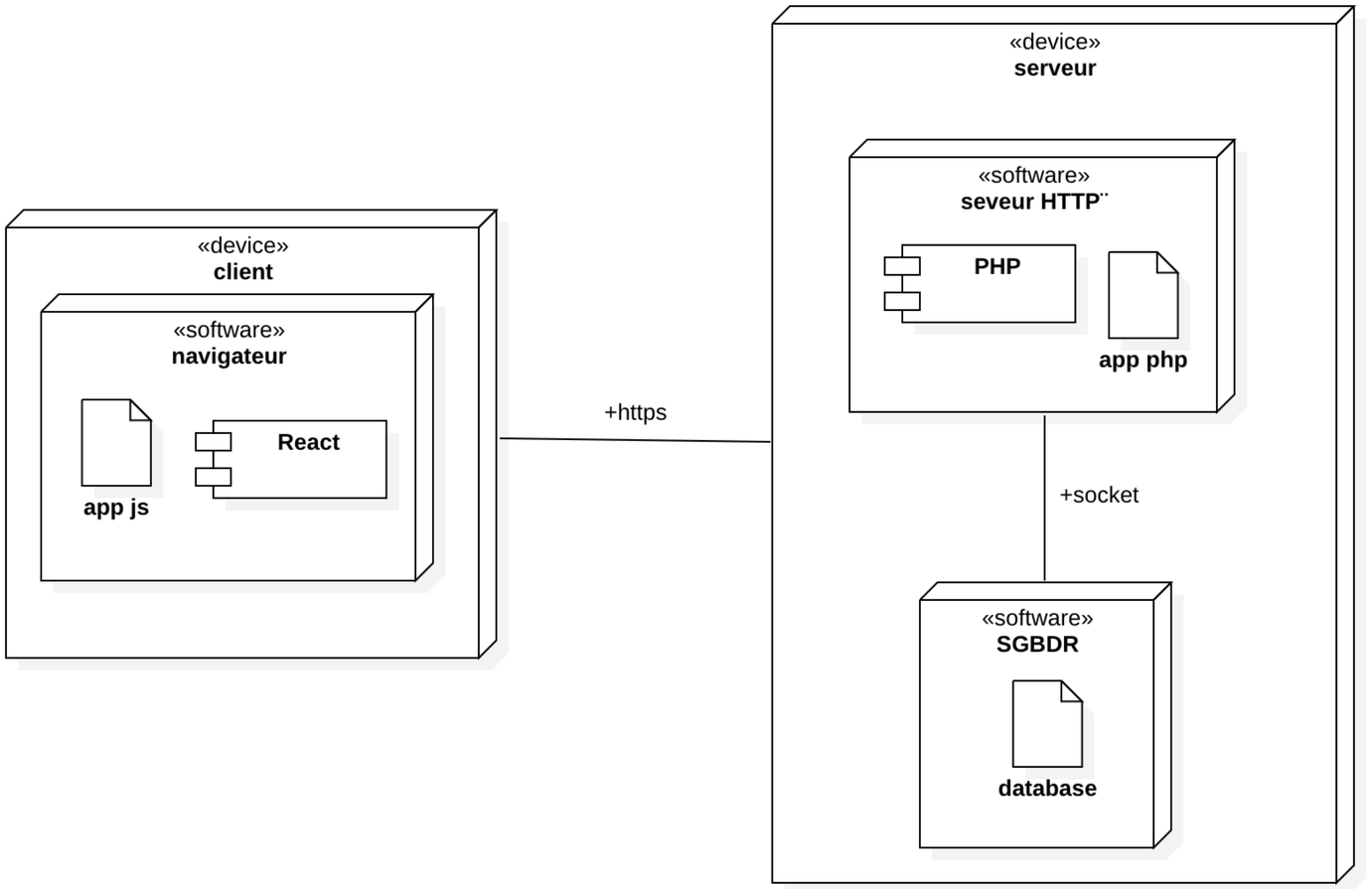
Un exemple simple



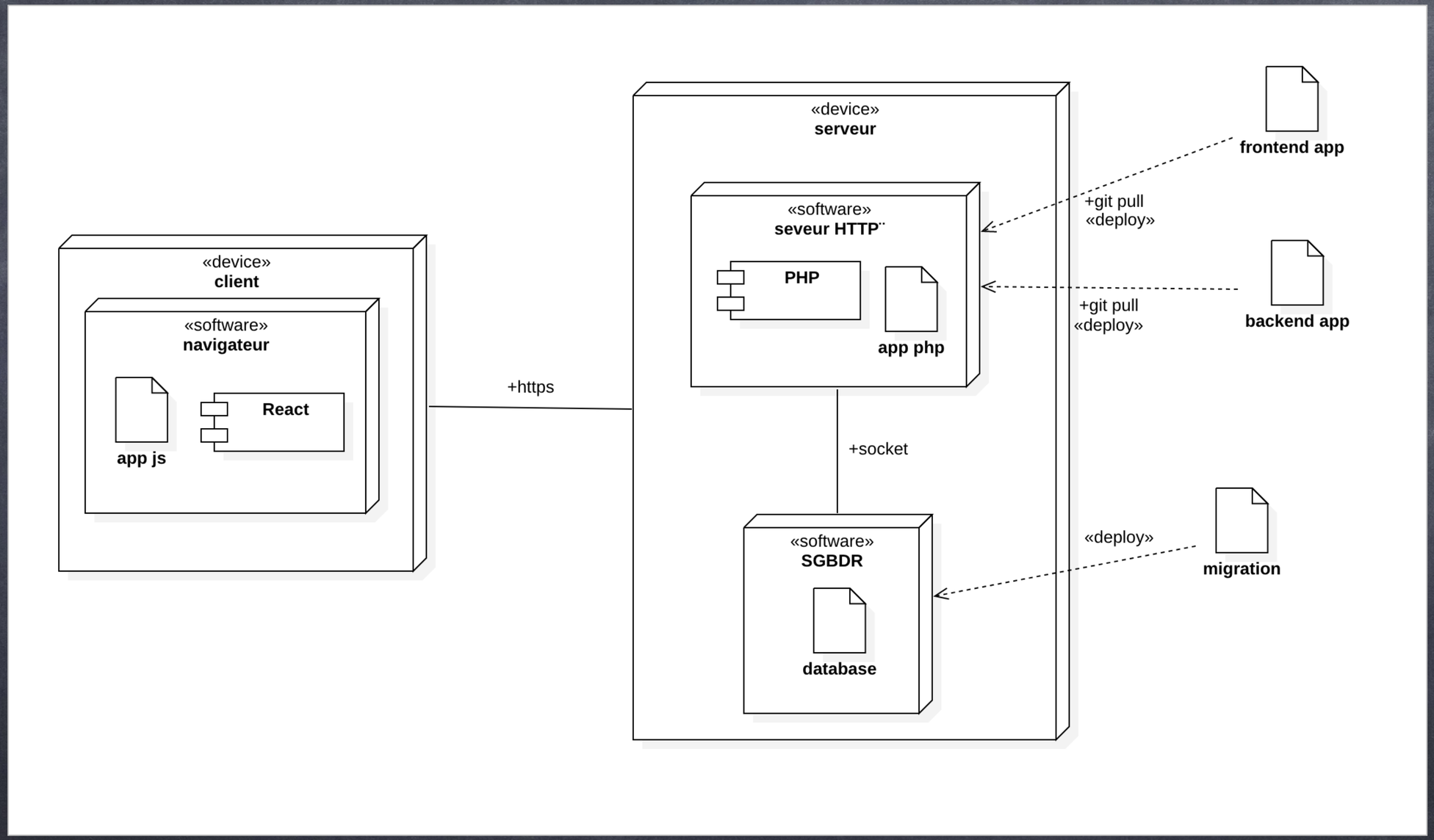
Composition des noeuds



Communication interne



UML Déploiement



UML Les éléments du diagramme

Noeud (node)	Représente un élément physique, un serveur, un routeur. Peut également représenter un logiciel en fonction du stéréotype
Composant (component)	Représente un composant logique, un logiciel, un module ou un service
Artéfact (artifact)	Représente un fichier
Liaison (communication path)	Canal de communication entre deux noeuds ou deux composants
déploiement	Lien de déploiement entre un artéfact ou un composant et un noeud