# UML – Use Case

**Bertrand LIAUDET**

## SOMMAIRE



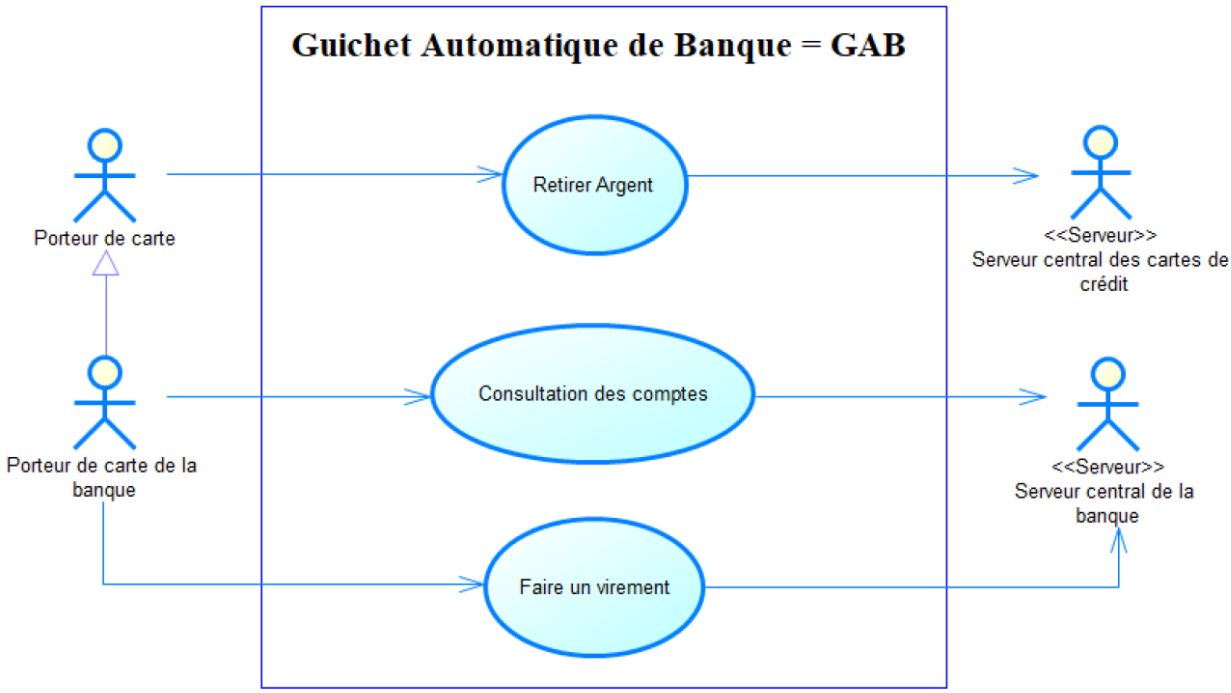
Dernière édition : 2023

# UC - BASES

## 1. La notion de cas d'utilisation : UC – Approche par un exemple – 6p

**Exemple de diagramme de cas d'utilisation : le GAB**

- Un guichet automatique de banque permet aux porteurs de carte de **retirer de l'argent**. Les porteurs de carte de la banque du guichet peuvent en plus **consulter leurs comptes et faire des virements**. Pour retirer de l'argent, le système communique avec le **serveur central des cartes de crédit**. Pour consulter les comptes ou faire un virement, le système communique avec le **serveur central de la banque concernée**.




- Dans ce schéma on trouve un système, des acteurs actifs et passifs, un héritage entre acteurs et 3 cas d'utilisation

**Cas d'utilisation = Use Case = UC**

**Synonyme**

- UC = **usage**

**Définition théorique**

- **UC = fonctionnalité complète** du système (du programme).

**Définition du point de vue de l'utilisateur :**

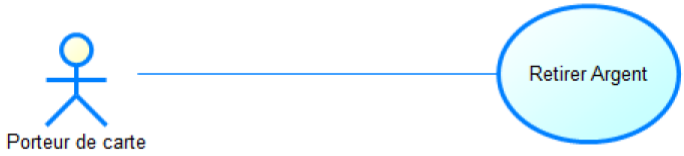
- UC = des actions qui produisent un **résultat intéressant pour un utilisateur**

**Définition du point de vue du système :**

- UC = des actions qui partent d'un système au repos pour arriver à un système au repos.

**Exemple du GAB : retirer de l'argent sur un GAB**

- Dans le système « guichet automatique d'une banque », GAB, **« retirer de l'argent » est un UC**.
- C'est une **fonctionnalité complète** du système qui va de l'insertion de la carte de retrait par le client jusqu'à la récupération de la carte de retrait par le client.



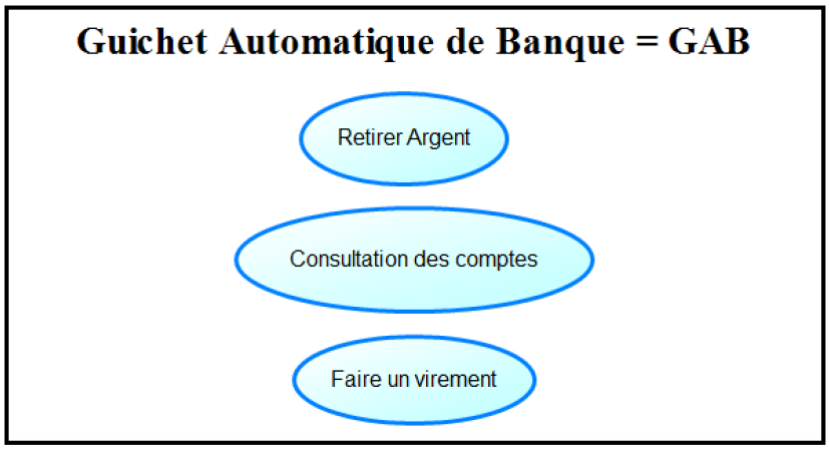
## Le système

### Définition – représentation UML

- Le système c'est **le programme qu'on réalise**. Il est constitué par **la totalité des UC.**
- On le représente par un rectangle qui regroupe les cas d'utilisation.
- Le rectangle matérialise les **frontières du système**.
- **Le système et les UC répondent à la question QUOI ?**

### Exemple du GAB :

Le guichet automatique de la banque propose 3 services qui correspondent à 3 UC :




### Précisions UML : notion de « classeur » : un rectangle de regroupement

- Un **classeur** est un élément de modélisation qui décrit une **unité comportementale ou structurelle**.
- C'est la **forme la plus simple du regroupement**. Un classeur est représenté par un rectangle.
- Le système complet est un classeur.

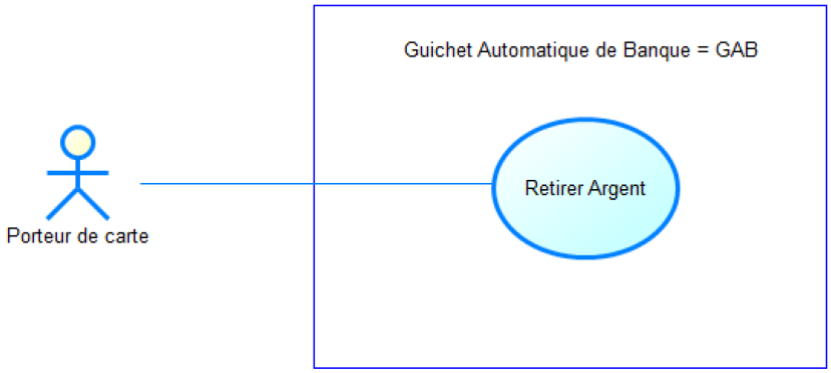
## Les interfaces du système : interactions avec le monde extérieur

**Interface proposée : principalement l'IHM (GUI ou CLI)**

- L'**IHM**, Interface Homme Machine, c'est l'interface avec les utilisateurs. **Le système fournit cette interface** pour communiquer avec les utilisateurs.
- L'interface proposée, ce sont les cas d'utilisation qui la fournisse.
- Les **utilisateurs** sont appelés « **acteurs** ». **Les acteurs sont à l'extérieur** du système. Ils communiquent avec l'interface.
- **Les utilisateurs répondent à la question QUI ?**

➢ ***Exemple du GAB :***

- Le porteur de carte **communique** avec système. Il peut accéder au cas d'utilisation « retirer de l'argent »

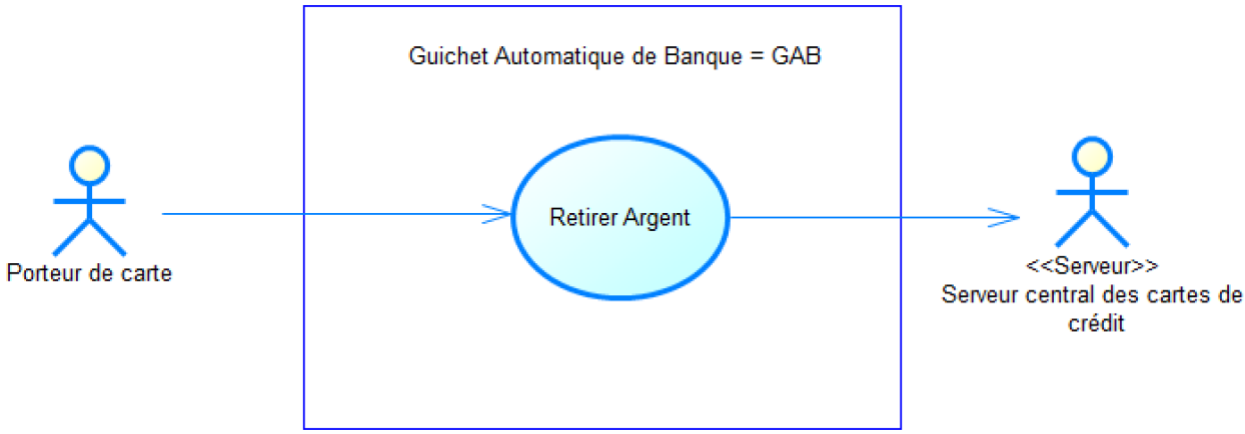


**Interface utilisée : acteur passif**

- Le système peut utiliser des machines, des logiciels, des services externes à lui pour réaliser ses UC.
- Ces **machines**, **logiciels**, services externes sont appelés « **acteur passif** ».
- Le système utilise les interfaces des acteurs passifs qui eux proposent ces interfaces.
- Un acteur passif est sollicité par un UC et communique avec le système mais un acteur passif n'est pas à l'origine du déclenchement du l'UC.
- **L'acteur passif répond à la question : AVEC QUI ? AVEC QUOI ?**

➢ ***Exemple du GAB :***

- Pour retirer de l'argent le système GAB doit communiquer avec le serveur central des cartes de crédit pour vérifier les autorisations.




**Précisions UML : liaison fléchée et stéréotype**

➢ ***Liaison fléchée et acteur passif :***

- La relation entre l'UC et un acteur passif est fléchée vers l'acteur passif.
- On peut flécher ou pas la relation entre un acteur actif et un cas d'utilisation.

➢ ***Notion de Stéréotype UML :***

- L'acteur « Serveur central des cartes de crédit » est **stéréotypé « Serveur »**.
- Un stéréotype est un type qu'on peut donner à tous les éléments graphiques UML pour les distinguer les uns des autres.
- Les stéréotypes sont toujours présentés entre guillemets : « Serveur ».

## Relativité des frontières du système

➢ ***Exemple du GAB :***

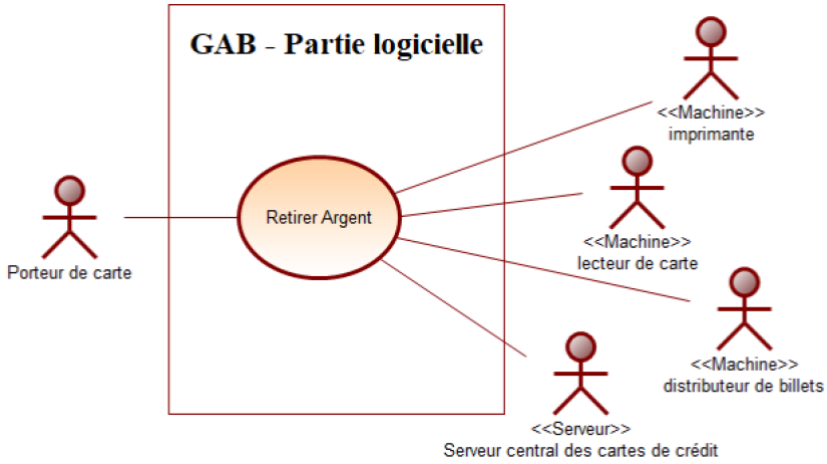
- Pour le développeur de la partie logiciel du GAB (l'interface utilisateur), l'automate de lecture de carte, le distributeur de billets et l'imprimante sont des sous-systèmes.

  ⇒ On peut sortir ces sous-systèmes du GAB et en faire des acteurs passifs.

  ⇒ Dans ce cas, ils ne font plus partie de la responsabilité du développeur de la partie logiciel du GAB.

  ⇒ Le développeur devra connaître les protocoles de communication avec ces acteurs passifs.

➢ ***Exemple : le GAB avec une responsabilité limitée au logiciel et pas aux automates (machines)***



## 2. Les acteurs – 3p

### L'acteur est un type

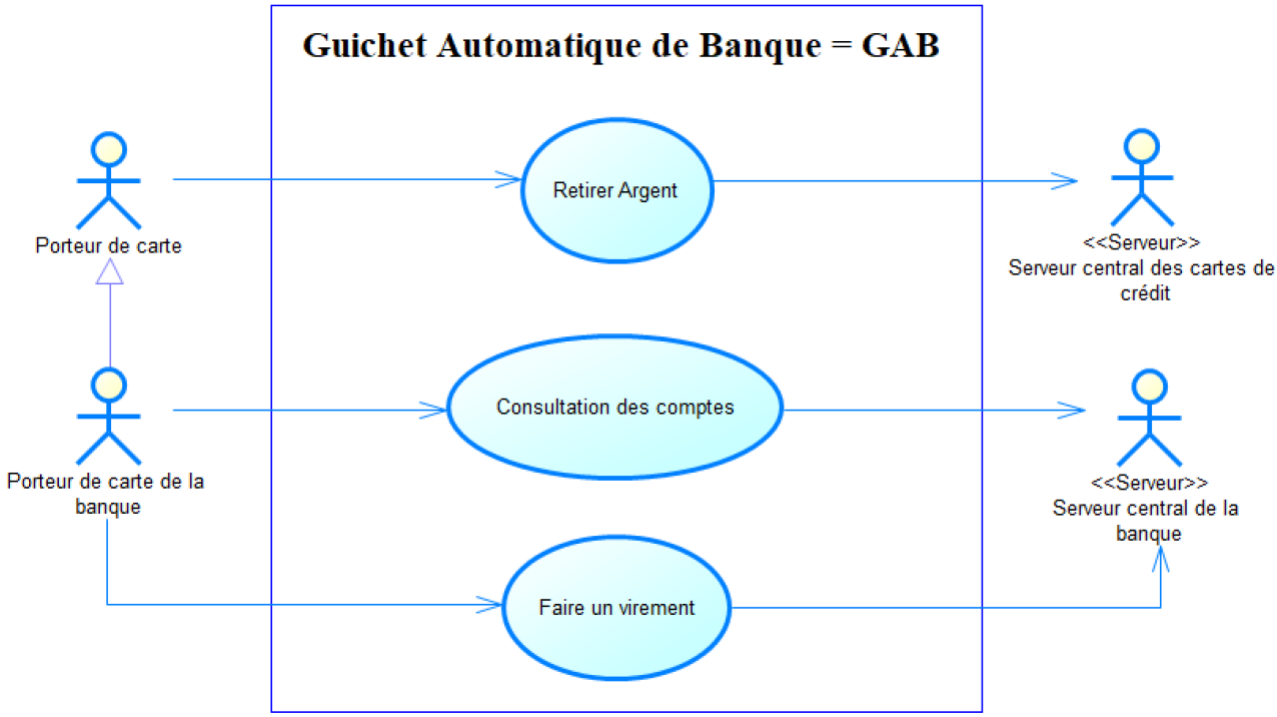
- La notion d'acteur est une abstraction : un type.
- L'acteur « porteur de carte » correspond à l'ensemble de toutes les personnes ayant une carte.
- L'acteur « imprimante » correspond à l'ensemble des machines de type « imprimante ».

### Relation entre les acteurs : l'héritage

- Il n'y a qu'une seule relation possible entre les acteurs : **l'héritage.**
- Comme pour tout héritage, c'est une **relation « est un ».**
- Comme pour tout héritage, ce mécanisme permet de factoriser l'écriture.

➢ ***Exemple du GAB :***

- Seuls les porteurs de carte de la banque peuvent consulter les comptes et faire un virement. Ils utilisent pour cela le serveur central de la banque.
- Les porteurs de carte de la banque peuvent faire tout ce que fait un simple porteur de carte.



**catégories d'acteurs :**

**principal *vs* secondaire**

- **L'acteur principal** : l'utilisateur. Celui pour qui est fait le système.
- **L'acteur secondaire** : l'administrateur du système, etc. (on écrit : << Secondaire >> )

**actif *vs* passif**

- **L'acteur actif** est à l'origine du UC. Il utilise le système.
- **L'acteur passif** n'est pas à l'origine du UC. Il est utilisé par le système.

  ⇒ **Les acteurs passifs sont souvent des machines ou des logiciels** (des serveurs).

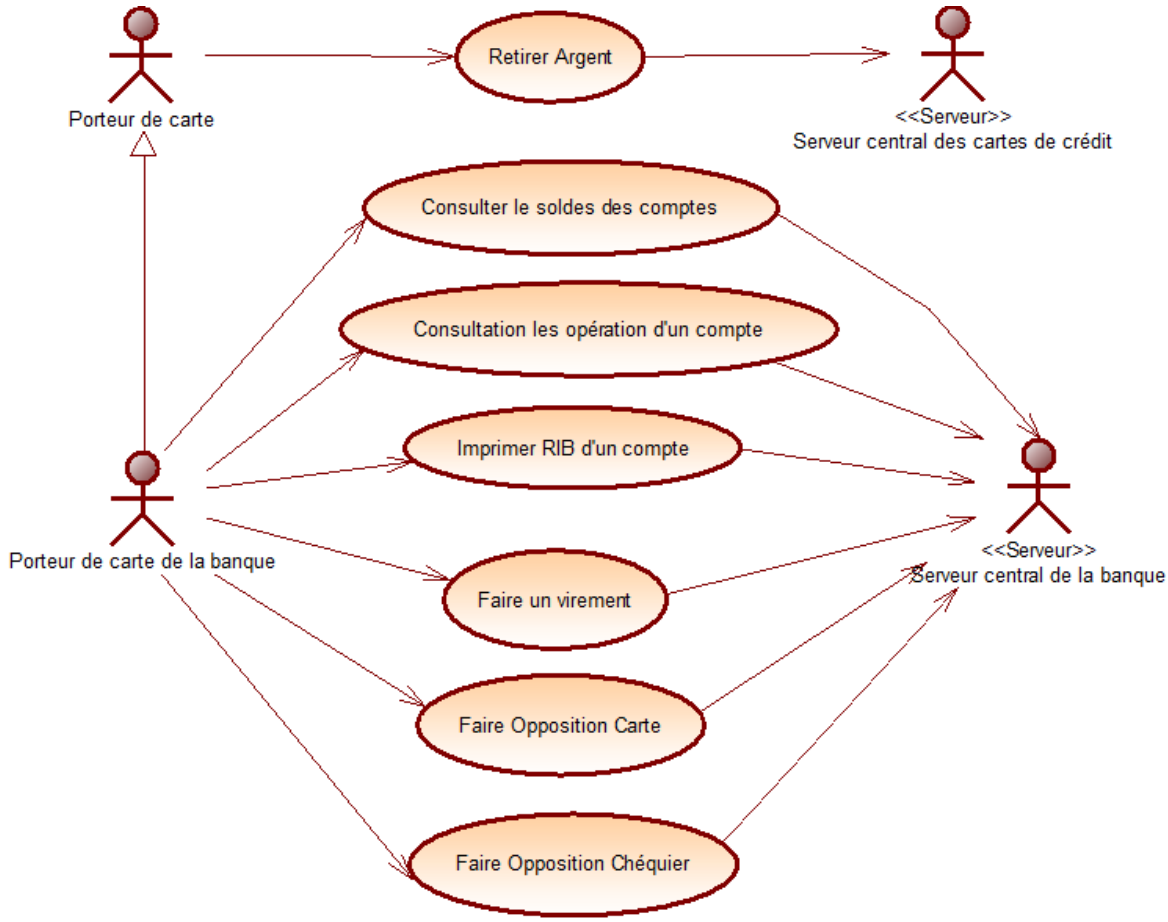
## 3. L'héritage entre UC : technique de regroupement – 3p

**Principes**

- Dès qu'on a plus de 3 ou 4 UC, on a intérêt à **réfléchir à des regroupements** pour clarifier la présentation.
- C'est une **logique de « menu déroulant »** : on met les UC qui vont ensemble dans un UC qui les regroupe et qui correspond à un menu déroulant.
- **Ca rend le modèle plus clair**, d'autant que ça permet aussi de regrouper de lien « acteur – cas d'utilisation » et aussi des liens de composition entre cas d'utilisation (include et extend).
- Attention, **les UC sont « objectifs »** : ils traduisent le cahier des charges. **Les regroupements sont toujours « subjectifs »** : ils traduisent une façon de voir les choses.

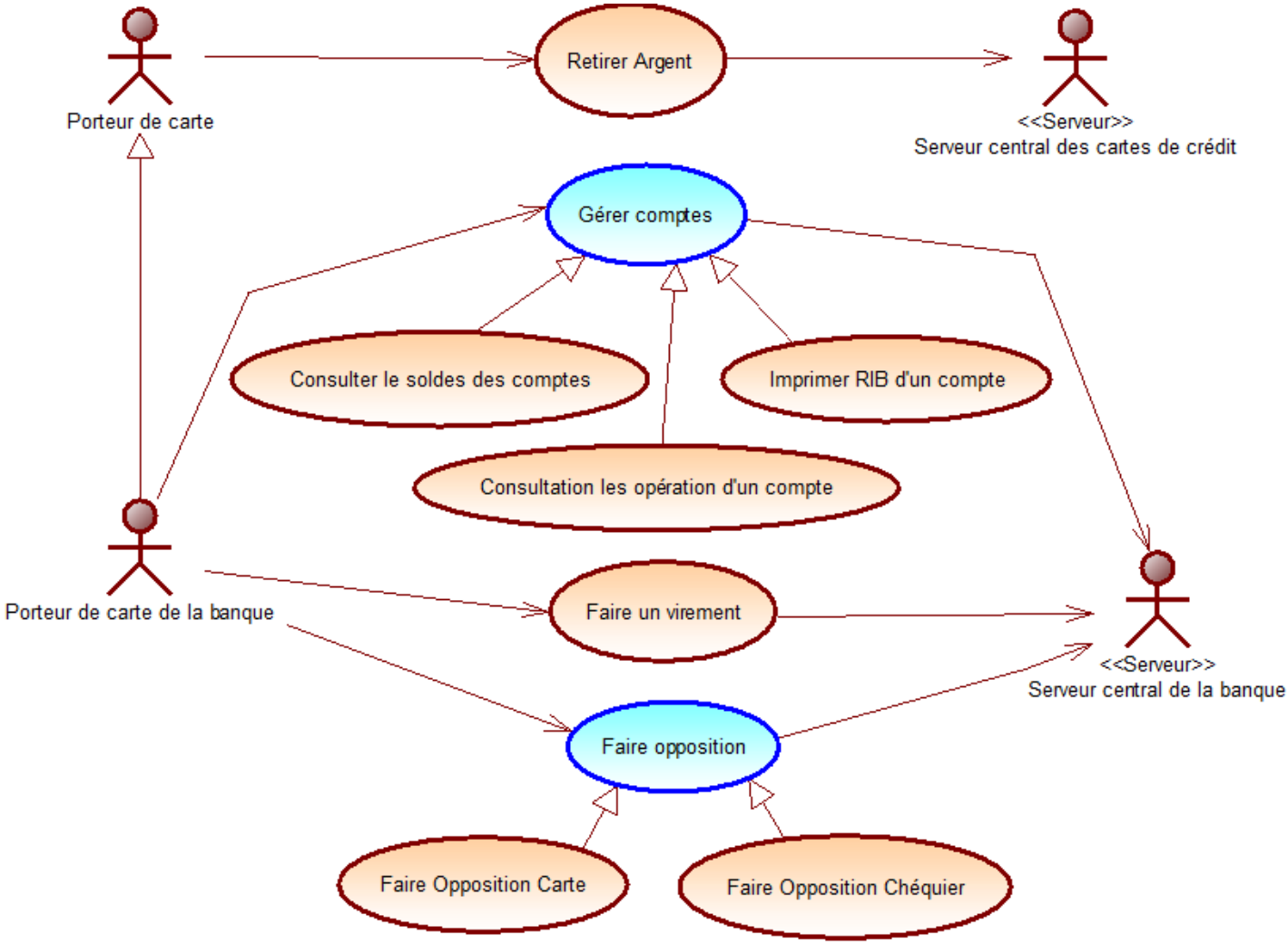
**Exemple : le GAB**

**Version sans généralisation : 7 UC accédés directement par les utilisateurs**



**Version avec généralisation : 4 UC accédés directement par les utilisateurs**

Les UC « Gérer comptes » et « Faire opposition » permettent de faire des regroupements.



## 4. Composant d'un UC : include et extend, techniques pour préciser – 5p

### Principes

- Un UC, c'est une succession d'activités élémentaires qui se déroulent dans le temps.
- Le porteur de carte vient pour retirer de l'argent :

  ⇒ il rentre sa carte,

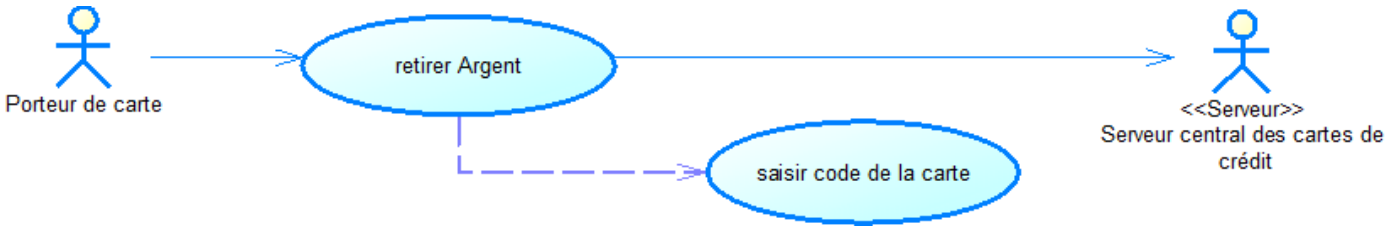
  ⇒ saisit son code,

  ⇒ saisit un montant,

  ⇒ récupère sa carte et

  ⇒ récupère son argent.
- **Les étapes décomposent l'UC : le coupe en tranches (= partie = composant)**
- On peut présenter le ou les composants qu'on souhaite.
- Le but est de clarifier le diagramme.

  Chaque « tranche » (ou partie ou composant) d'un UC est représentée comme un UC et relié à son UC complet (le composé) par une relation d'include ou d'extend. Les include et les extend apportent des précisions sur le contenu d'un UC.
- Exemple : l'UC « retirer Argent » inclut l'UC « saisir code de la carte ».



**Attention !**

## C'est du détail !

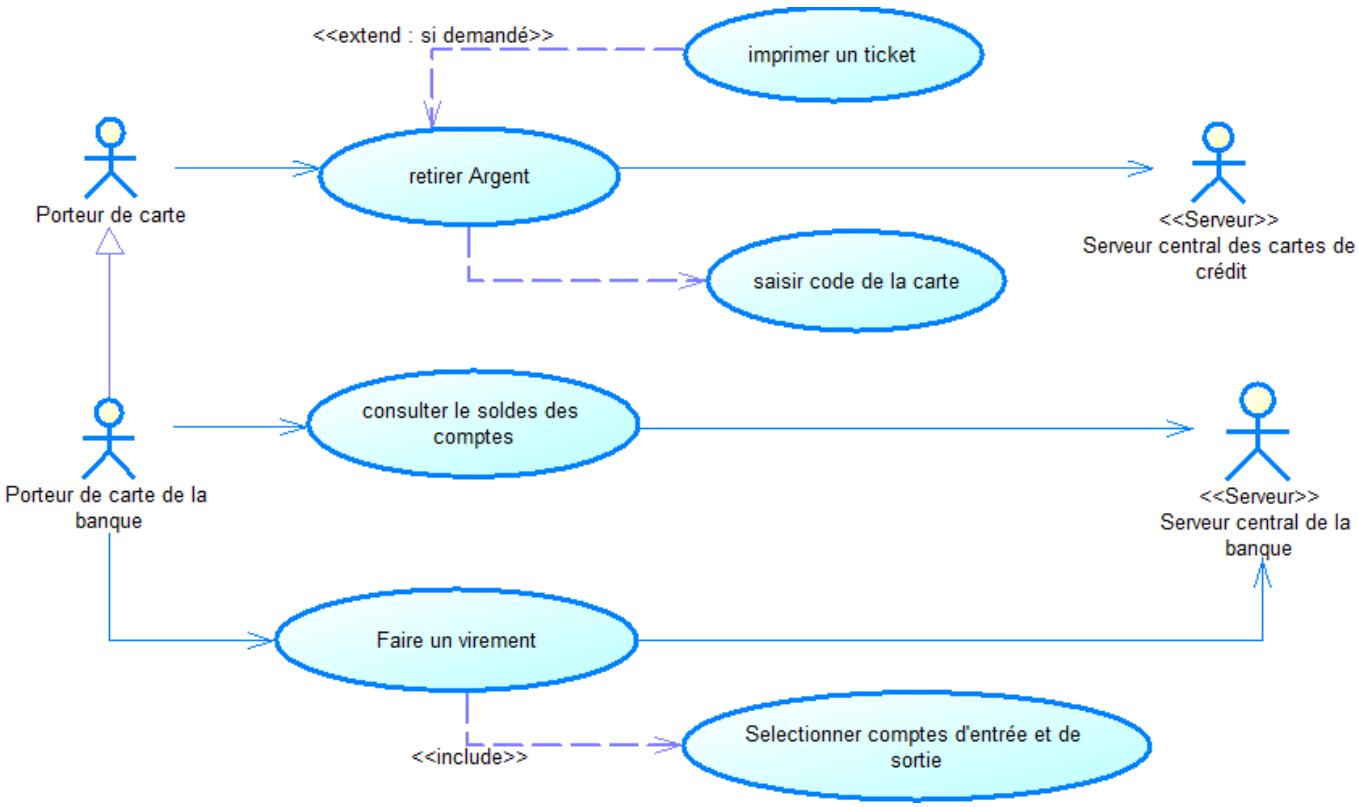
- C'est du détail

  ⇒ Il ne faut pas trop détailler.

  ⇒ Il faut bien doser le niveau de détail : on peut s'en passer, c'est juste pour y voir un peu plus clair.

**Exemple du GAB**

## Diagramme de Cas d'utilisation :



**2 include**

- Retirer de l'argent inclus forcément : saisir le code de la carte.
- Faire un virement inclus la sélection des comptes d'entrée et de sortie.

**1 extend**

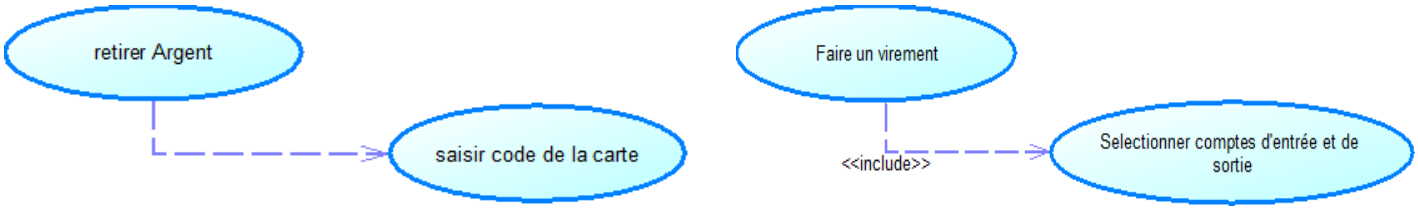
- Retirer de l'argent inclus l'impression du ticket **à condition qu'on l'ait demandé**.

**Distinction « include » et « extend »**

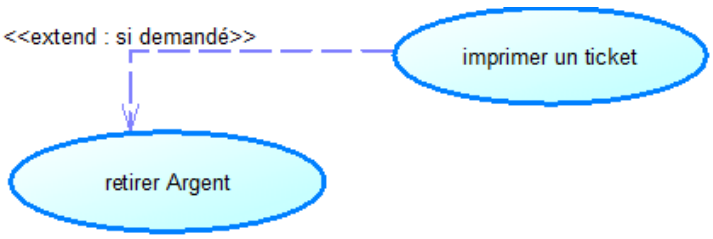
- Quand on réalise l'UC complet (par exemple quand on retire de l'argent) :
  - Un **include** est un composant qui **se réalise forcément** (par exemple : saisir le code de la carte).
  - Un **extend** est un composant qui **se réalise si une condition est vérifiée** (par exemple : imprimer un ticket). En général, la condition est : « si demandé ».

**Notation UML :**

- Le lien de composition entre 2 UC, lien en pointillé, est fléché.
- **Include** : L'UC de départ est le composé, l'UC d'arrivé est le composant. S'il n'y a pas de stéréotype sur un lien de composition entre 2 UC, c'est forcément un include. On peut préciser le stéréotype sur l'include




- **Extend** : L'UC de départ est le composant, l'UC d'arrivé est le composé. Attention ! C'est l'inverse de l'include. On est obligé d'afficher le stéréotype « extend » en précisant la condition (souvent simplement « si demandé »).



## 5. Use case et CRUD – 1p

### CRUD

Dès qu'on gère une BD, il faut faire des CRUD : create (= insert), read (=select), update et delete.

Par exemple, on aura des use cases pour créer, modifier, supprimer, consulter un utilisateur.

### UC « gestion de tel truc »

On peut regrouper ça par le terme : « Gestion utilisateur »

Et on peut, ou pas, montrer 4 sous-UC : insert, upadate, delete, read

Dans ce cas, connexion et déconnexion, sont un autre usage, qu'on peut regrouper dans l'UC « Connexion », avec 2 sous-UC : connexion et déconnexion.

## 6. Outils

### Logiciel de dessin en ligne : Draw.io

On peut utiliser le logiciel dessin en ligne draw-io pour faire de l'UML :

- Draw.io : https://app.diagrams.net/

### Génération automatique textuelle : chatGPT

On peut demander à chatGPT de générer un diagramme de cas d'utilisation textuel et ensuite le transcrire en schéma

- ChatGPT : https://chatgpt.com/

### Modeler UML

Un modeler UML est une application qui permet de faire des diagrammes UML et parfois de faire de la génération automatique de code.

Il existe aussi de nombreuses applications dédiées à cela : LucidChart, Visio, Star UML, SAP Power Designer, etc.