

# *UML - Diagrammes de cas d'utilisation*

*K. Gaber*

# Diagramme des cas d'utilisation

- Décrit, sous forme d'actions et de réactions, le comportement d'un système du point de vue d'un utilisateur.
- Permet de définir les limites du système et ses relations avec l'environnement.
- Sert à modéliser les aspects dynamiques d'un système (Contrairement aux diagrammes de classes).
- Fait ressortir les acteurs et les fonctions offertes par le système.
- Utilisé pour modéliser les exigences (besoins) du client

# Diagrammes des cas d'utilisation

Comportent plusieurs éléments :

- Acteurs
- Cas d'utilisation
- Relations de dépendances, de généralisations et d'associations

# Acteurs

- UML n'emploie pas le terme d'utilisateur mais d'acteur.
- Le terme acteur ne désigne pas seulement des utilisateurs humains mais également les autres systèmes (machines, programmes, ...)
- Un acteur est un rôle joué par une entité externe qui agit sur le système (Comptabilité, service commercial, ...), en échangeant de l'information (en entrée et en sortie)

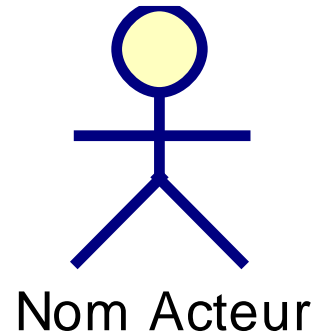
## Remarques

- La même personne physique peut jouer le rôle de plusieurs acteurs (*Chef d'agence est un client de la banque*).
- D'autres part, plusieurs personnes peuvent jouer le même rôle, et donc agir comme un même acteur (*plusieurs personnes peuvent jouer le rôle d'administrateur*).

# Acteurs

Peut être représenté de deux manières différentes :

- Petit personnage



- Classe stéréotypée



# Acteurs

Les acteurs peuvent être de trois types :

- **Humains** : utilisateurs du logiciel à travers son interface graphique, par exemple.
- **Logiciels** : disponibles qui communiquent avec le système grâce à une interface logicielle (API, ODBC, ...)
- **Matériels** : exploitant les données du système ou qui sont pilotés par le système (Imprimante, robots, automates, ...)

# Acteurs

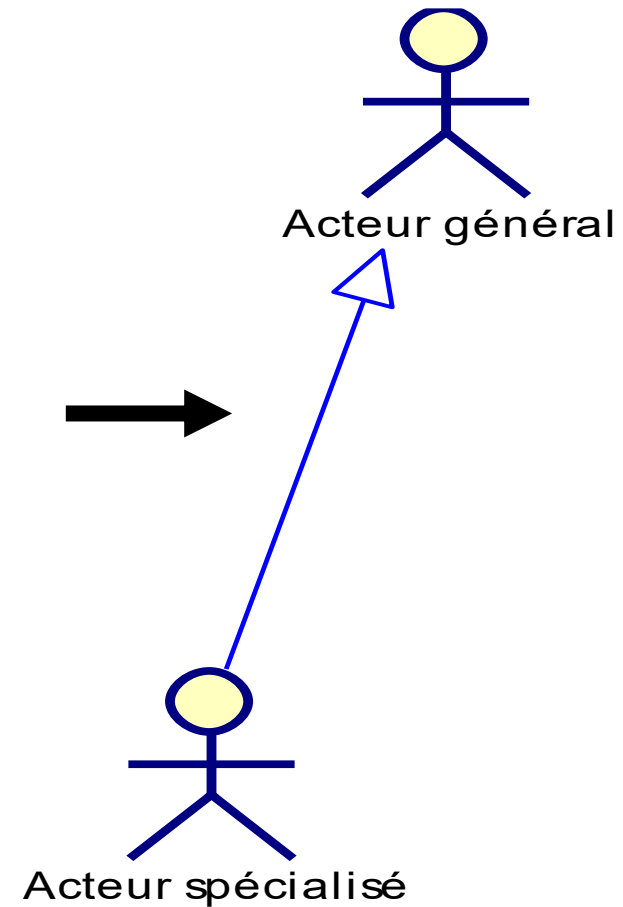
Du point de vue système on distingue deux types d'acteurs :

- Acteurs principaux : utilisent les fonctions principales du système. Par exemple, le client pour un distributeur de billets.
- Acteurs secondaires : effectuent des tâches administratives ou de maintenance. Par exemple, la personne qui recharge la caisse contenue dans le distributeur.

# Acteurs

Un acteur peut être une **spécialisation** d'un **autre acteur** déjà défini.

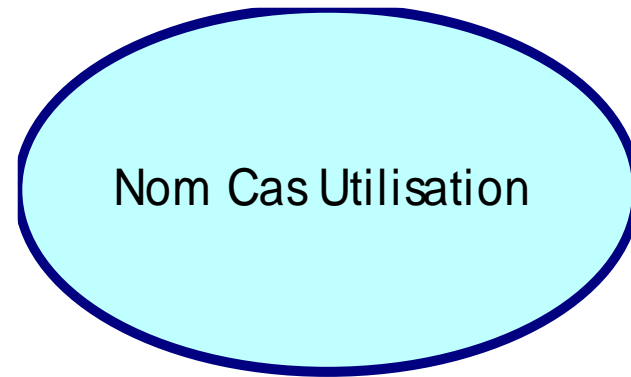
Dans ce cas, on utilise la relation de **généralisation/spécialisation**.





# Cas d'utilisation

Un cas d'utilisation est représenté par une ellipse en trait plein, contenant son nom.



# Relations entre cas d'utilisation

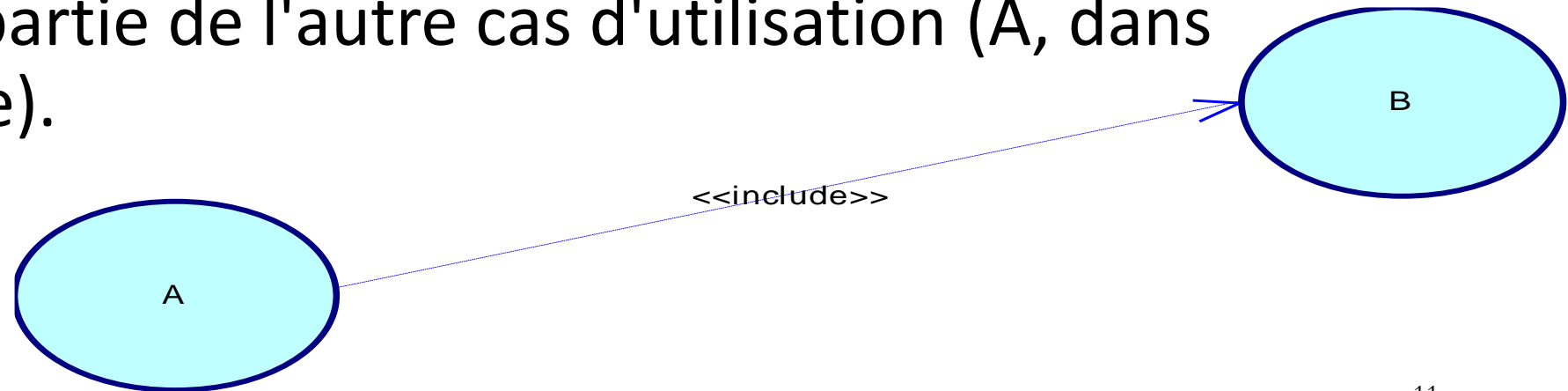
Il y a **trois types de relations** standardisées entre cas d'utilisation :

- Une relation **d'inclusion**, formalisée par la dépendance «include»
- Une relation **d'extension**, formalisée par la dépendance «extend»
- Une relation de **généralisation/spécialisation**

# Relation d'inclusion

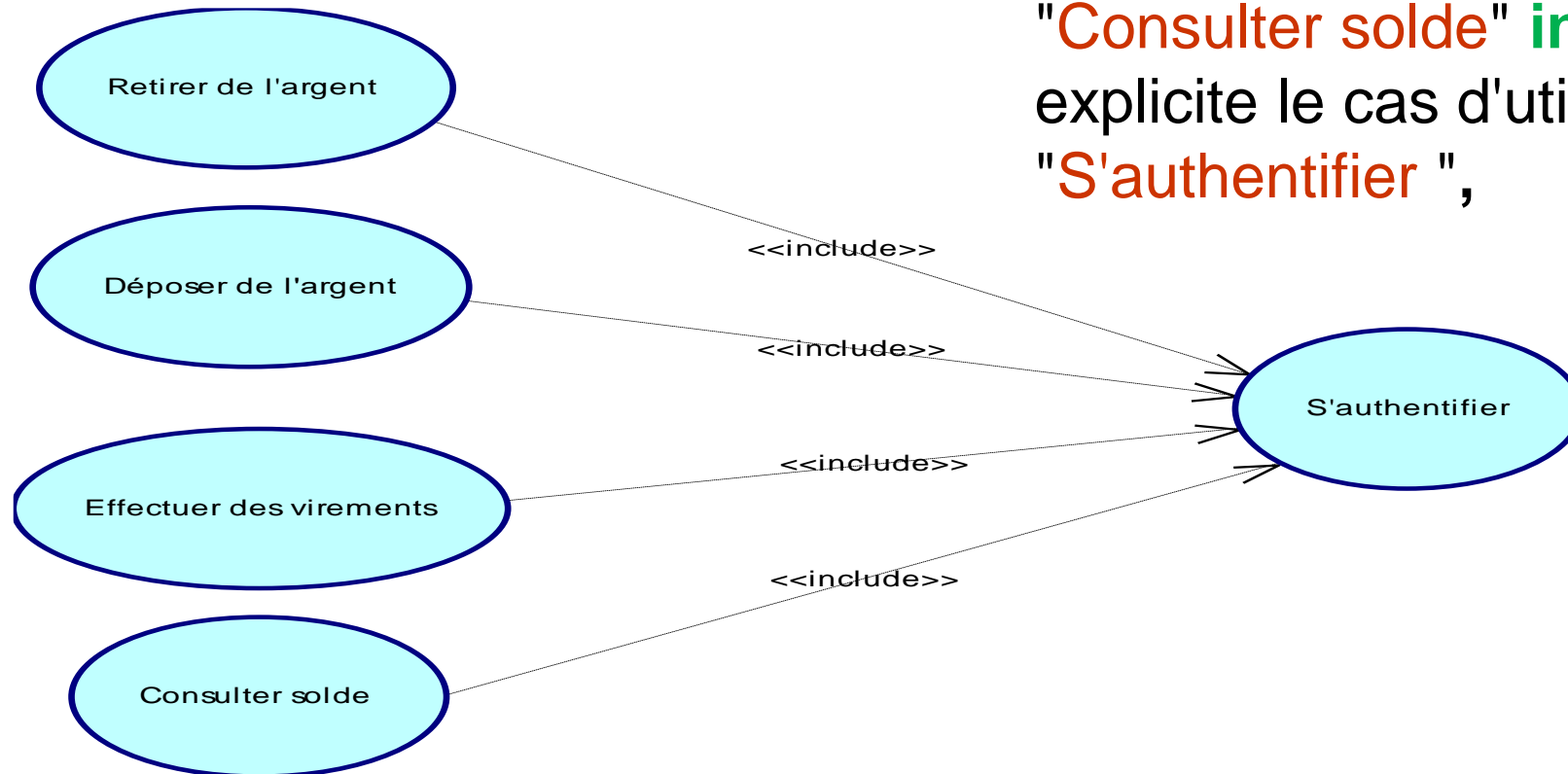
Lors de la description des cas d'utilisation, il existe des sous-ensembles communs à plusieurs cas d'utilisation

- A inclut B : le cas A **inclut obligatoirement** le comportement défini par le cas B; permet de factoriser des fonctionnalités partagées
- Le cas d'utilisation pointé par la flèche (dans notre cas B) est une sous partie de l'autre cas d'utilisation (A, dans notre exemple).



# Relation d'inclusion

Les cas d'utilisation "Déposer de l'argent", "Retirer de l'argent", "Effectuer des virements" et "Consulter solde" **inclut** de façon explicite le cas d'utilisation "S'authentifier",



# Relation d'inclusion

- Une instance du cas *source* *inclut obligatoirement* le comportement décrit par le cas d'utilisation *destination*
- Permet de décomposer des comportements et de définir les comportements partagées entre plusieurs cas d'utilisation  
—► Factoriser

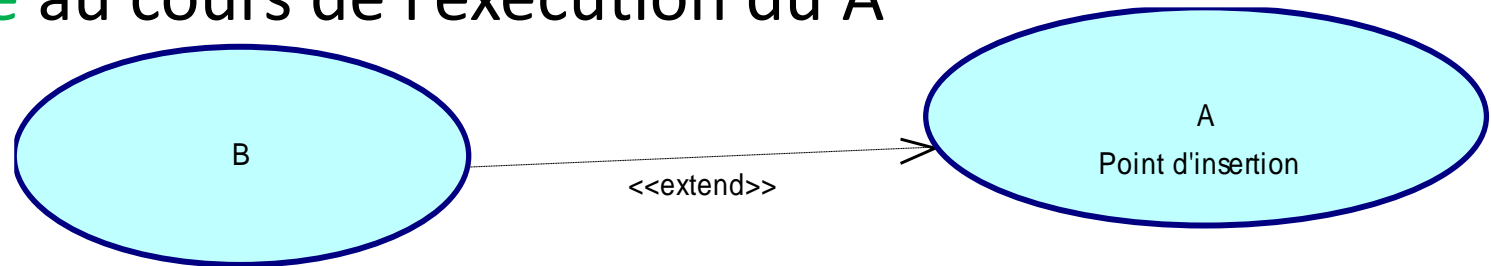
# Relation d'extension

La relation stéréotypée «*extend*» permet d'étendre les interactions et donc les fonctions décrites dans les cas d'utilisation, *mais sous certaines contraintes*.

# Relation d'extension

La relation stéréotypée «**extend**» permet d'étendre les interactions et donc les fonctions décrites dans les cas d'utilisation, **mais sous certaines contraintes**.

- Le Cas **source B** ajoute, sous certaines conditions, son comportement au Cas **destination A**
- le B **peut être appelé** au cours de l'exécution du A



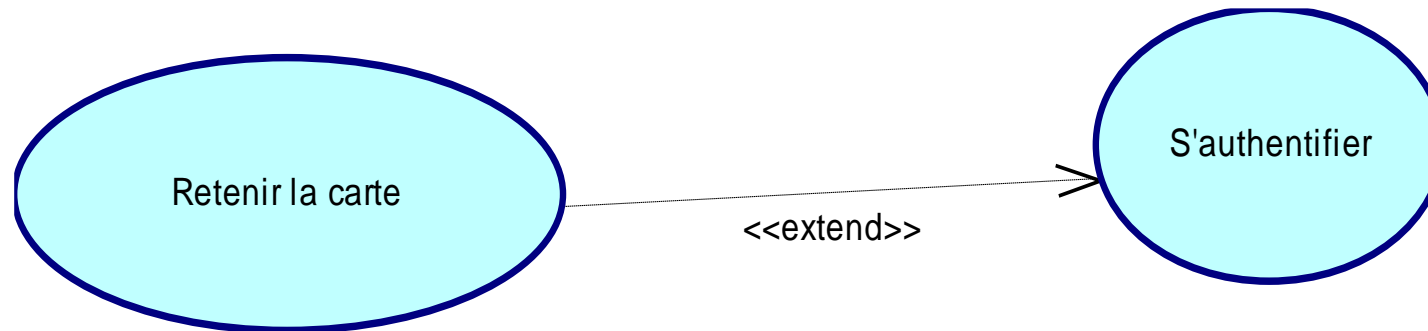
- Le comportement ajouté s'insère au niveau d'un point d'extension définit dans le Cas d'utilisation destination

# Relation d'extension

- Le cas d'utilisation de **destination peut fonctionner tout seul**, mais il **peut** également **être complété** par un autre cas d'utilisation, **sous certaines conditions**.
- On utilise principalement cette relation pour **séparer** le comportement **optionnel** (les variantes) du comportement **obligatoire**.

Exemple :

Au moment de **l'authentification**, il se peut que le guichet **retient la carte**.





# Relation d'héritage

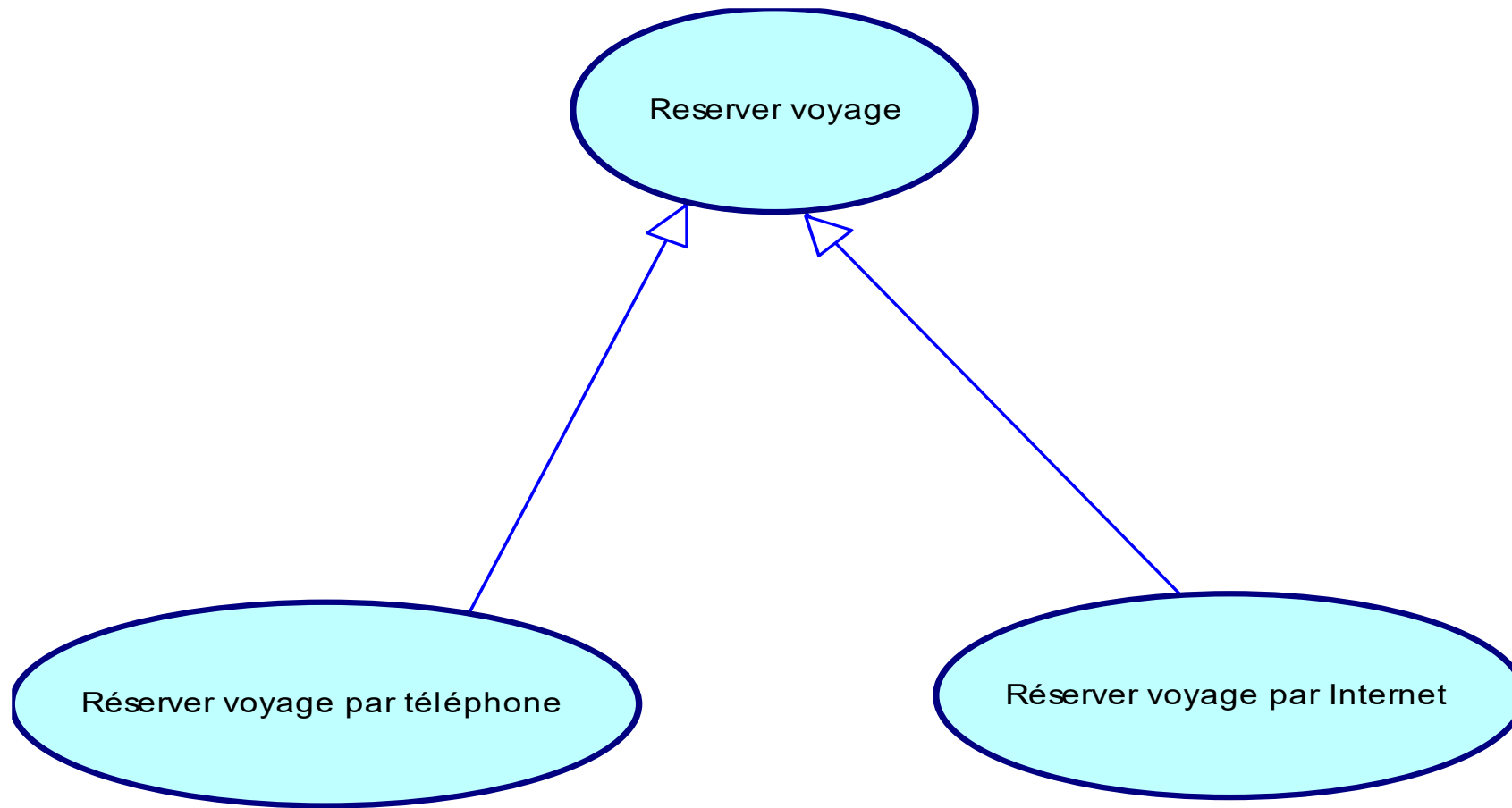
- Il peut également exister une relation d'héritage entre cas d'utilisation.
- Cette relation exprime une relation de spécialisation/généralisation au sens classique.

## Exemple:

Dans un système d'agence de voyage, un acteur "Touriste" peut participer à un cas d'utilisation qui est "Réservation voyage". Une réservation peut être réalisée par téléphone ou par Internet.

- Les deux cas d'utilisation "Réservation voyage" et "Réserver voyage par Internet" sont liés : la réservation par Internet est un cas particulier de réservation.

# Relation d'héritage : Exemple



# Structuration entre cas d'utilisation(résumé)

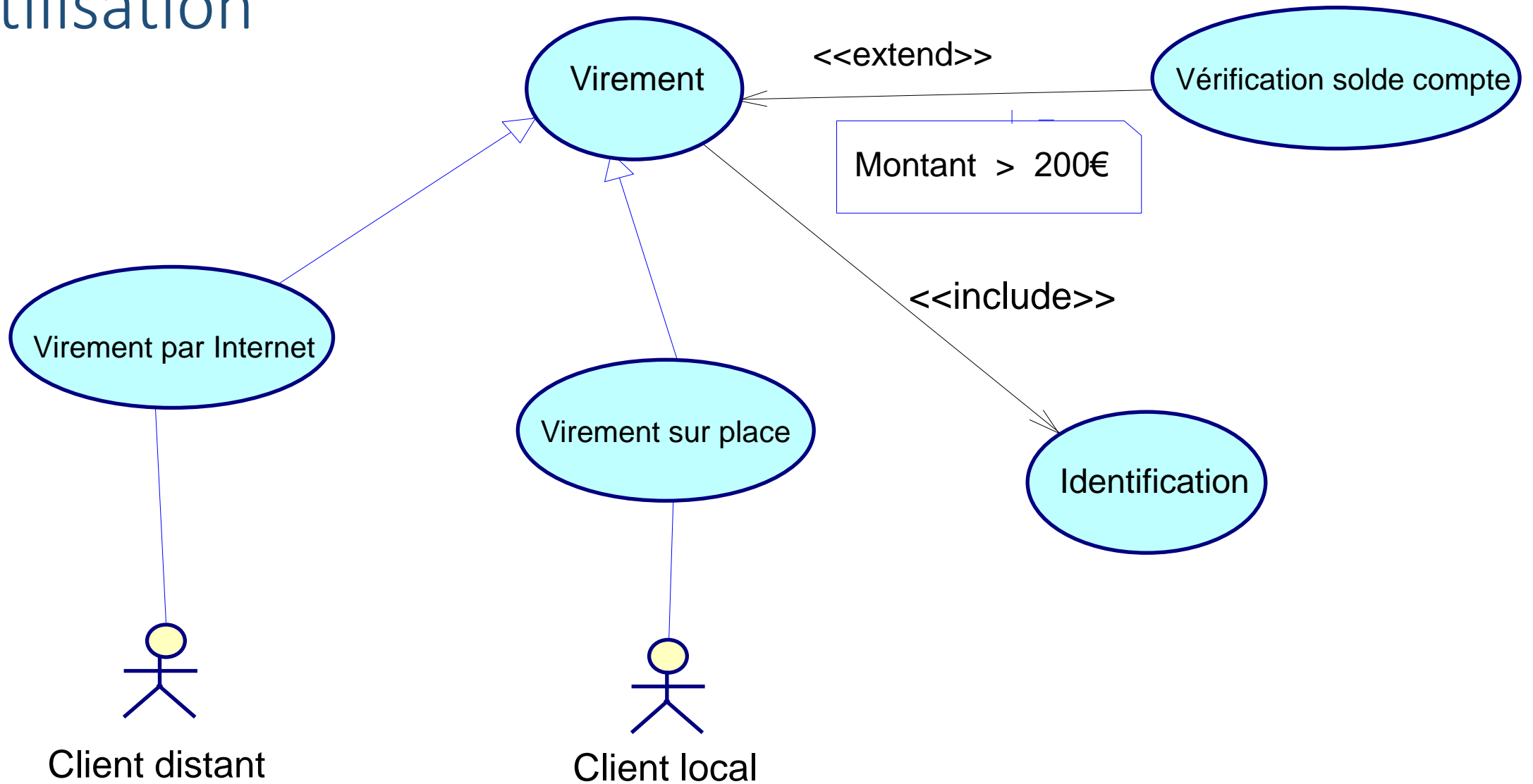
Les cas peuvent être structurées par des relations :

- A **inclut** B : le cas A **inclut obligatoirement** le comportement défini par le cas B; permet de factoriser des fonctionnalités partagées
- A **étend** B : le cas A est une **extension optionnelle** du cas B à un certain point de son exécution.
- A **généralise** B : le cas B est un cas particulier du cas A.

## Relations entre cas d'utilisation : Exemple

Un client peut effectuer un virement bancaire. Le virement peut être effectué sur place ou par Internet. Le client doit être identifié (en fournissant son code d'accès) pour effectuer un retrait, mais si le montant dépasse 200 €, la vérification du solde de son compte est réalisée.

# Relations entre cas d'utilisation



# Description des cas d'utilisation

- Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs.
- Mais il n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation.
- ➔ nécessité de décrire ce dialogue

# Description des cas d'utilisation

Deux façons sont couramment utilisées pour décrire les cas d'utilisation :

- Description textuelle
- Description à l'aide d'un diagramme de séquence (voir chapitre suivant)

# Description des cas d'utilisation (description textuelle)

## Identification

- Nom du cas : retrait d'argent
- Objectif : détaille les étapes permettant à un guichetier d'effectuer des opérations de retrait par un client
- Acteurs : Guichetier (Principal), Système central (Secondaire)

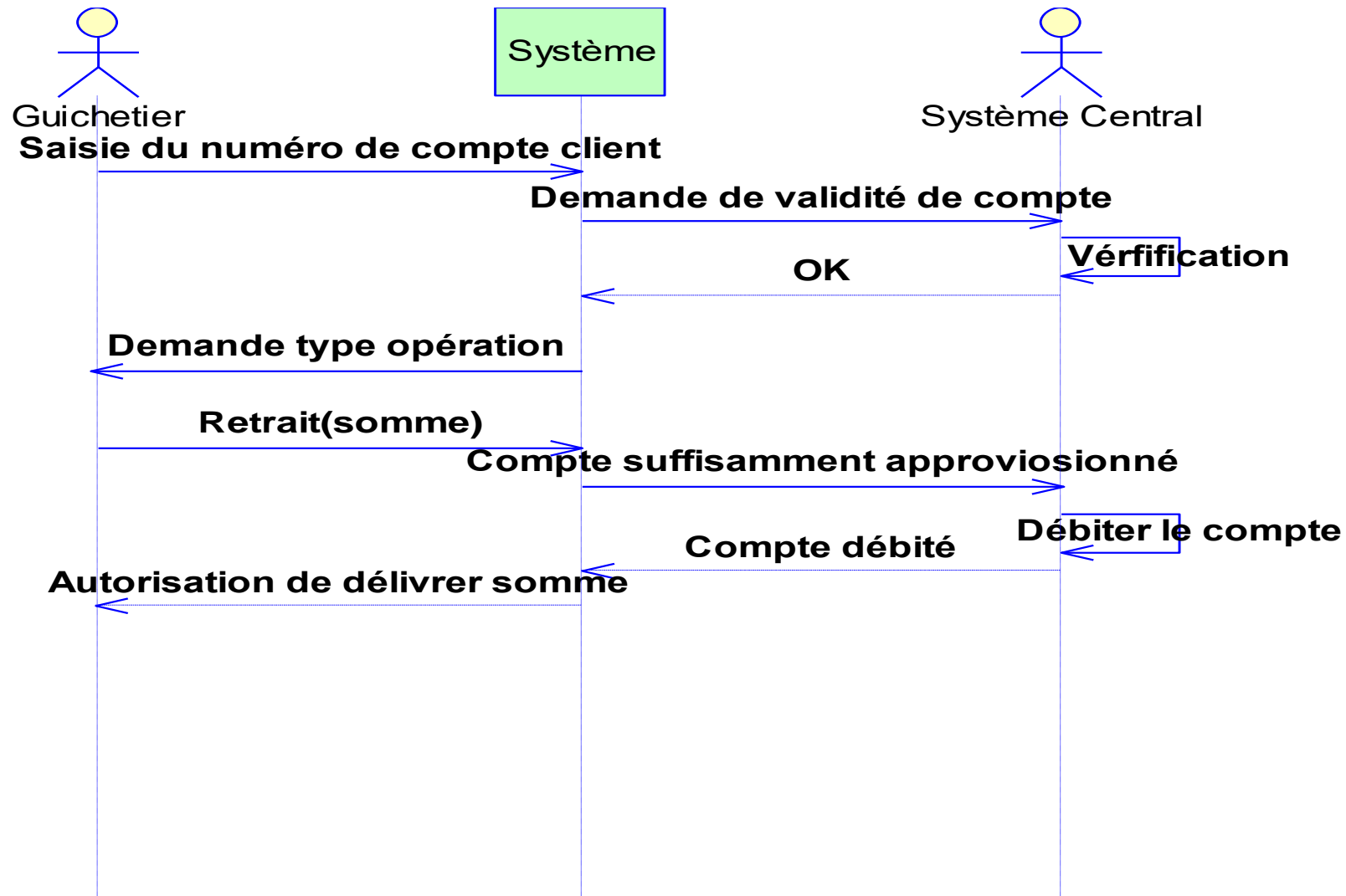


# Description des cas d'utilisation (description textuelle)

## Scénario :

1. Le Guichetier saisit le numéro de compte client
2. L'application valide le compte auprès du *système central*
3. L'application demande le type d'opération au Guichetier
4. Le Guichetier sélectionne un retrait de 200 €
5. Le système interroge le *système central* pour s'assurer que le compte est suffisamment approvisionné, sinon approvisionner
6. Le *système central* effectue le débit du compte
7. Le système notifie au guichetier qu'il peut délivrer le montant demandé

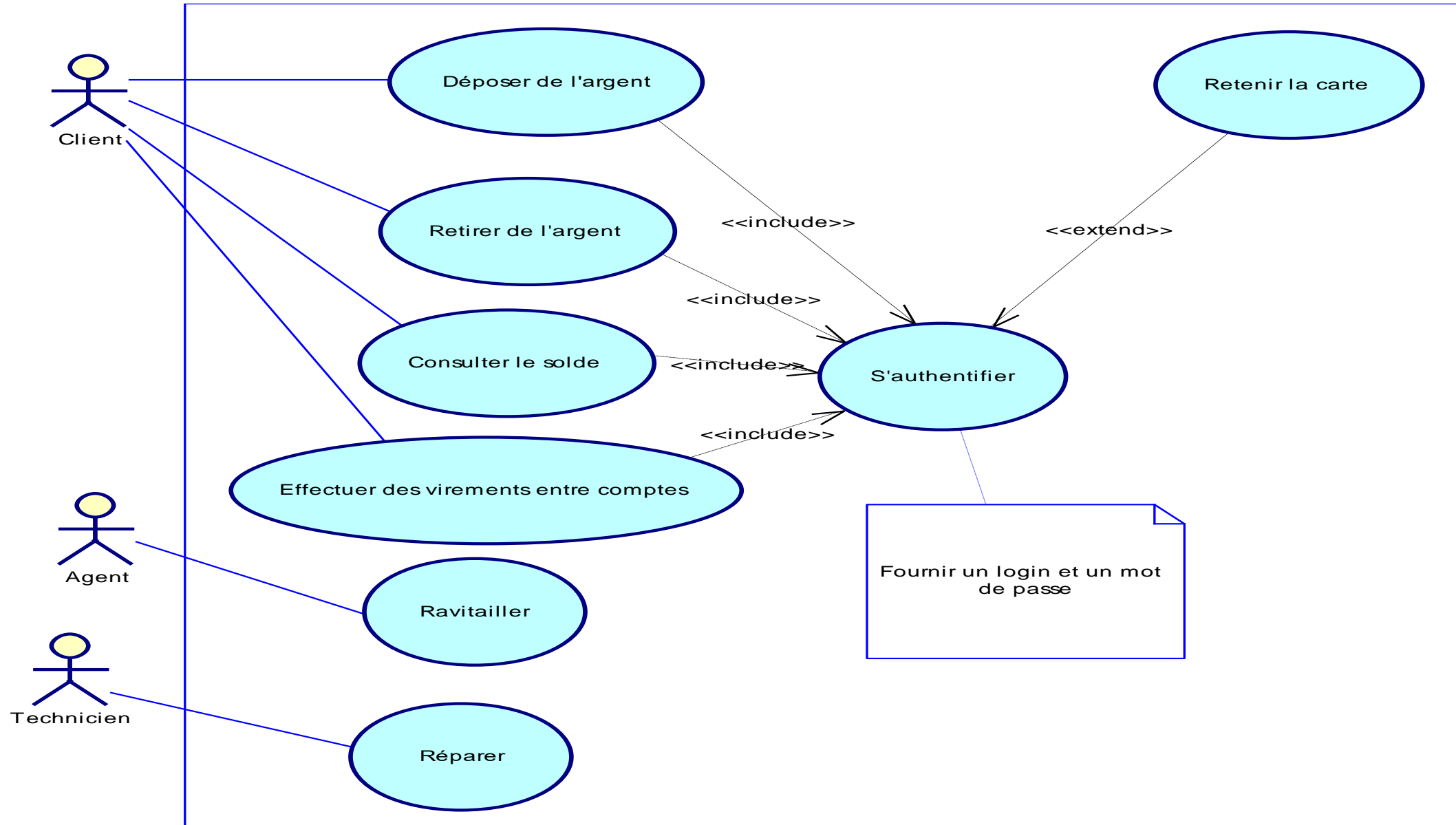
# Description des cas d'utilisation (description par diagramme de séquence)



# Diagramme des cas d'utilisation

- Le diagramme des cas d'utilisation regroupe dans un même schéma les acteurs et les cas d'utilisation en les reliant par des relations. Le système étant délimité par un cadre rectangulaire.
- La représentation de base d'un cas d'utilisation est une ellipse contenant le nom du cas. L'interaction entre un acteur et un cas d'utilisation se représente comme une association. Elle peut comporter des multiplicités comme toute association entre classes.

# Diagramme des cas d'utilisation



# Étapes de construction du diagramme des cas d'utilisation

Pour **modéliser** le diagramme des cas d'utilisation, il faut :

- **Identifier les acteurs** qui entourent le système. Certains acteurs utilisent le système pour accomplir des tâches (acteurs principaux), d'autres effectuent des tâches de maintenance ou d'administration (acteurs secondaires).
- **Organiser les acteurs** selon une hiérarchisation de généralisation/spécialisation
- **Intégrer les acteurs au diagramme** en spécifiant les cas d'utilisation auxquels ils se rapportent
- **Structurer les cas d'utilisation** pour faire apparaître les comportement **partagés** (relation d'inclusion), les cas **particuliers** (généralisation/spécialisation) ou **options** (relation d'extension)