

UML

Diagrammes de séquence

Delphine Longuet
delphine.longuet@lri.fr

Diagrammes de séquence (conception)

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 messages**
 - entre les acteurs et le système, entre les objets du système
 - de façon chronologique

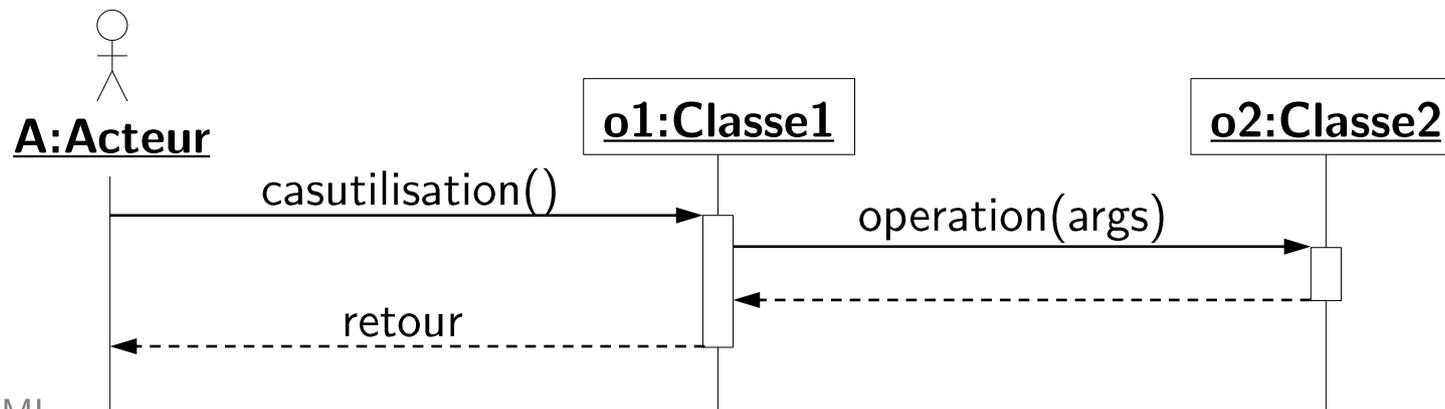
Diagrammes de séquence (conception)

Éléments du diagramme de séquence

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

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
- Échanges de messages représentés horizontalement



Utilisation en phase de conception

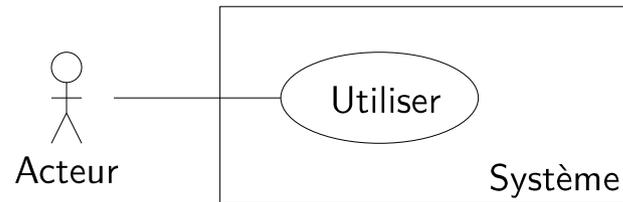


Diagramme de cas d'utilisation

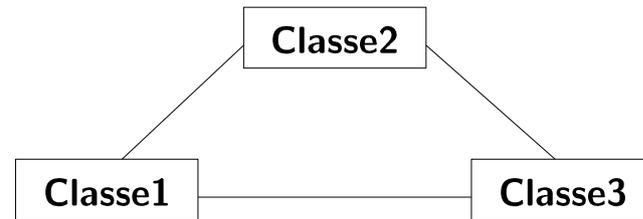


Diagramme de classes du système

Objectif : Description de la réalisation d'un cas d'utilisation sur le système décrit par le diagramme de classes

Problème : Communication entre les acteurs et le système vu comme un ensemble d'objets

Utilisation en phase de conception

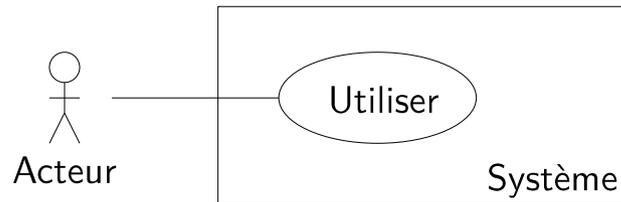


Diagramme de cas d'utilisation

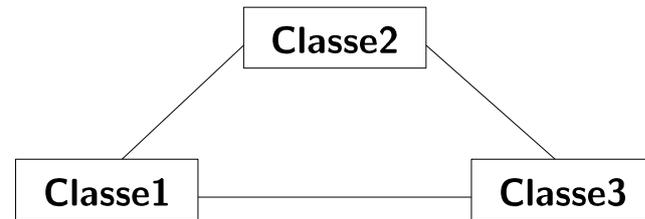
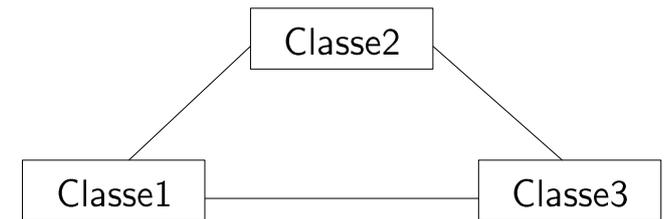
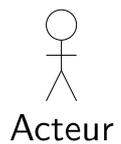


Diagramme de classes du système



Communication entre acteurs et système via une interface (texte, web, physique...)

Utilisation en phase de conception

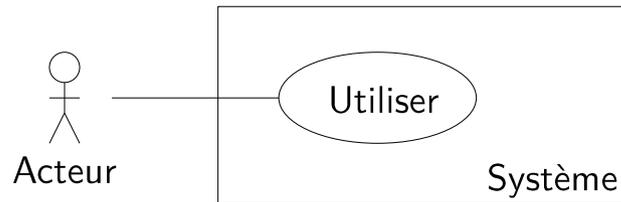


Diagramme de cas d'utilisation

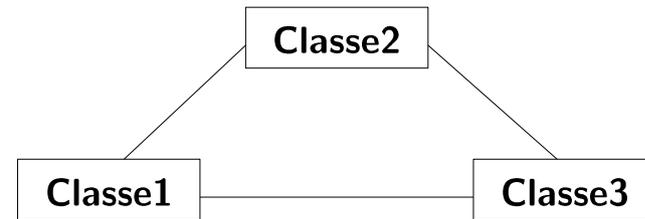
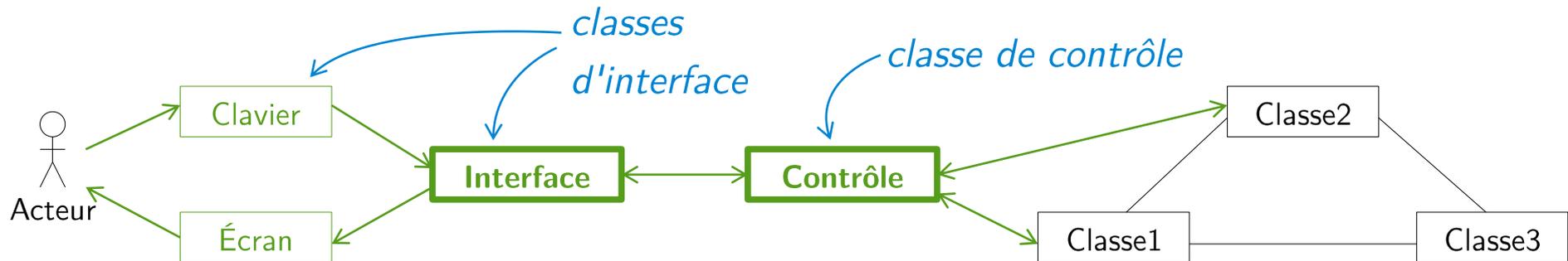


Diagramme de classes du système



Solution : Création de classes de contrôle et de classes d'interface qui :

- gèrent les interactions avec les acteurs
- encapsulent le résultat des opérations

Utilisation en phase de conception

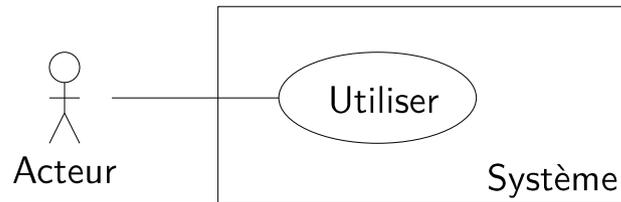


Diagramme de cas d'utilisation

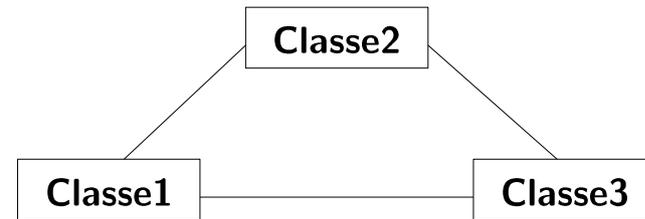


Diagramme de classes

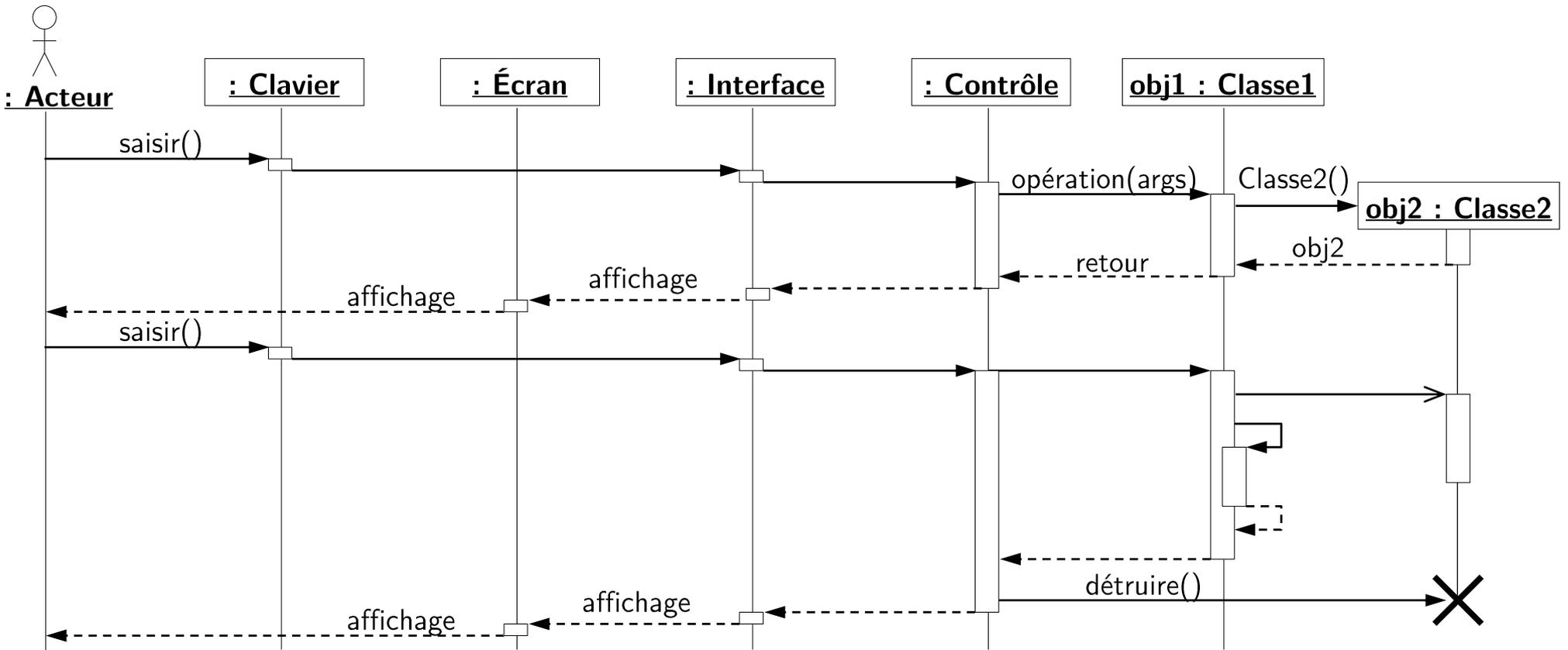
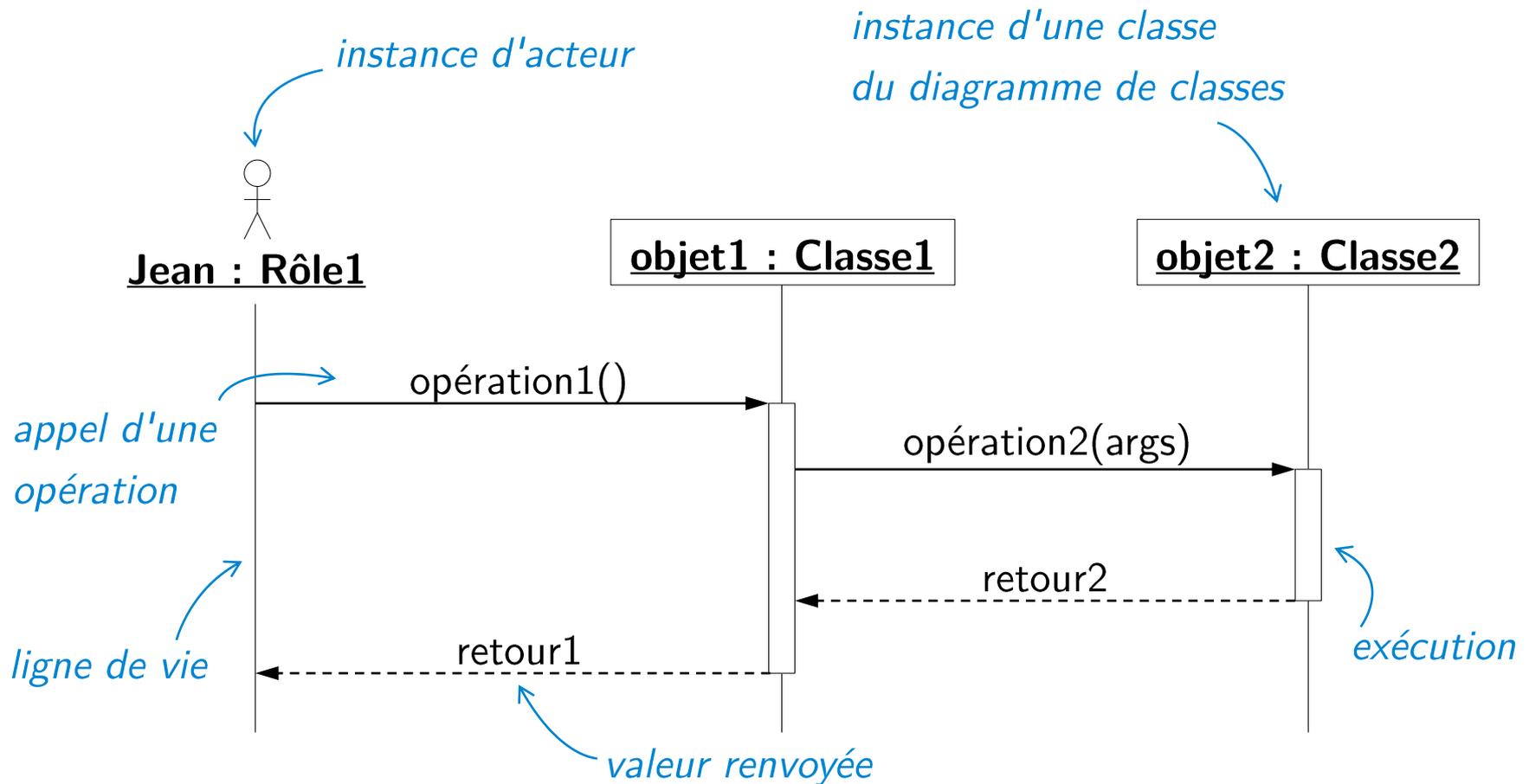


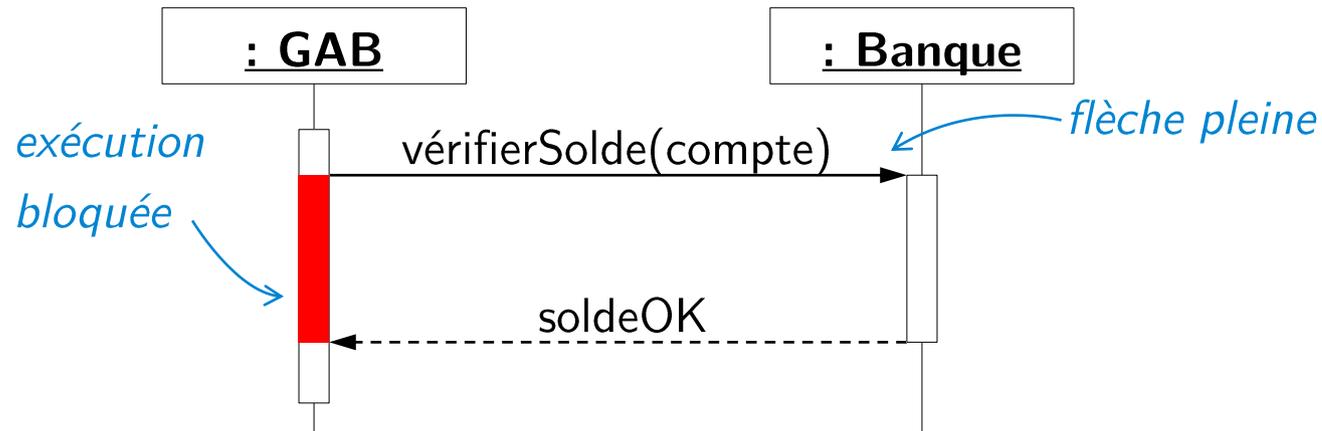
Diagramme de séquence du cas d'utilisation Utiliser

Éléments de base

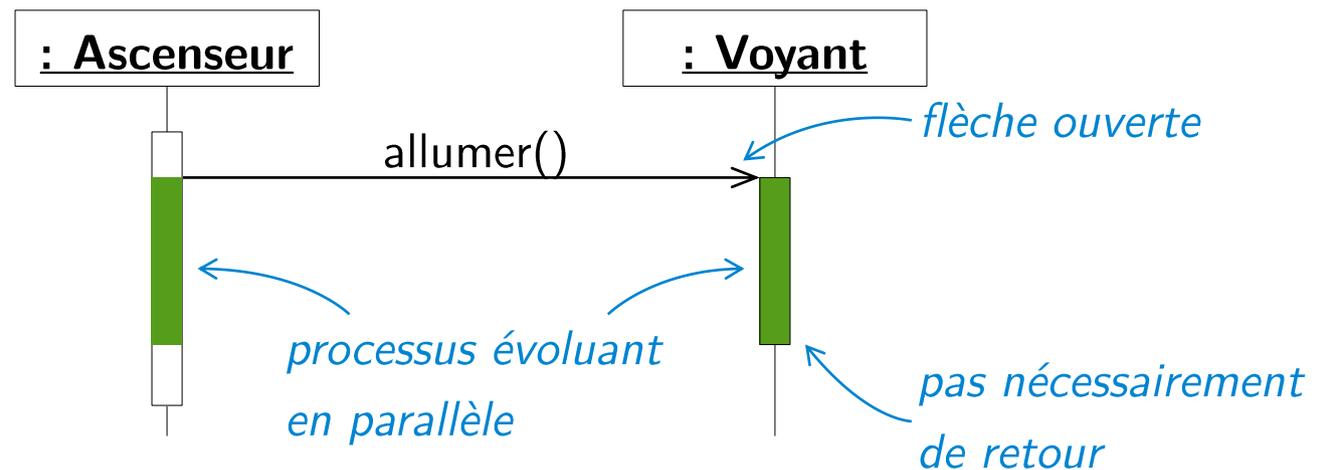


Types de messages

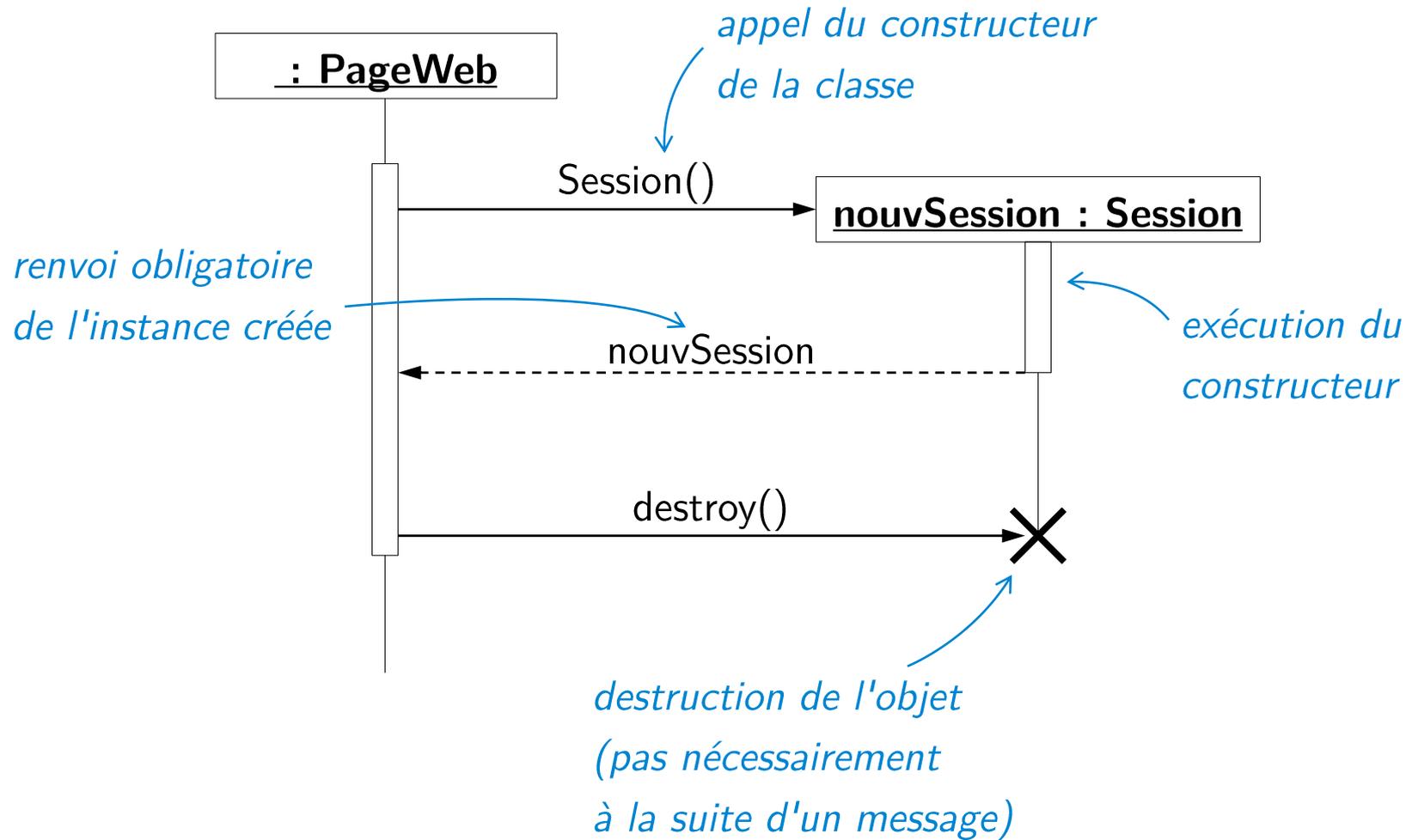
Message synchrone : Émetteur bloqué en attente du retour



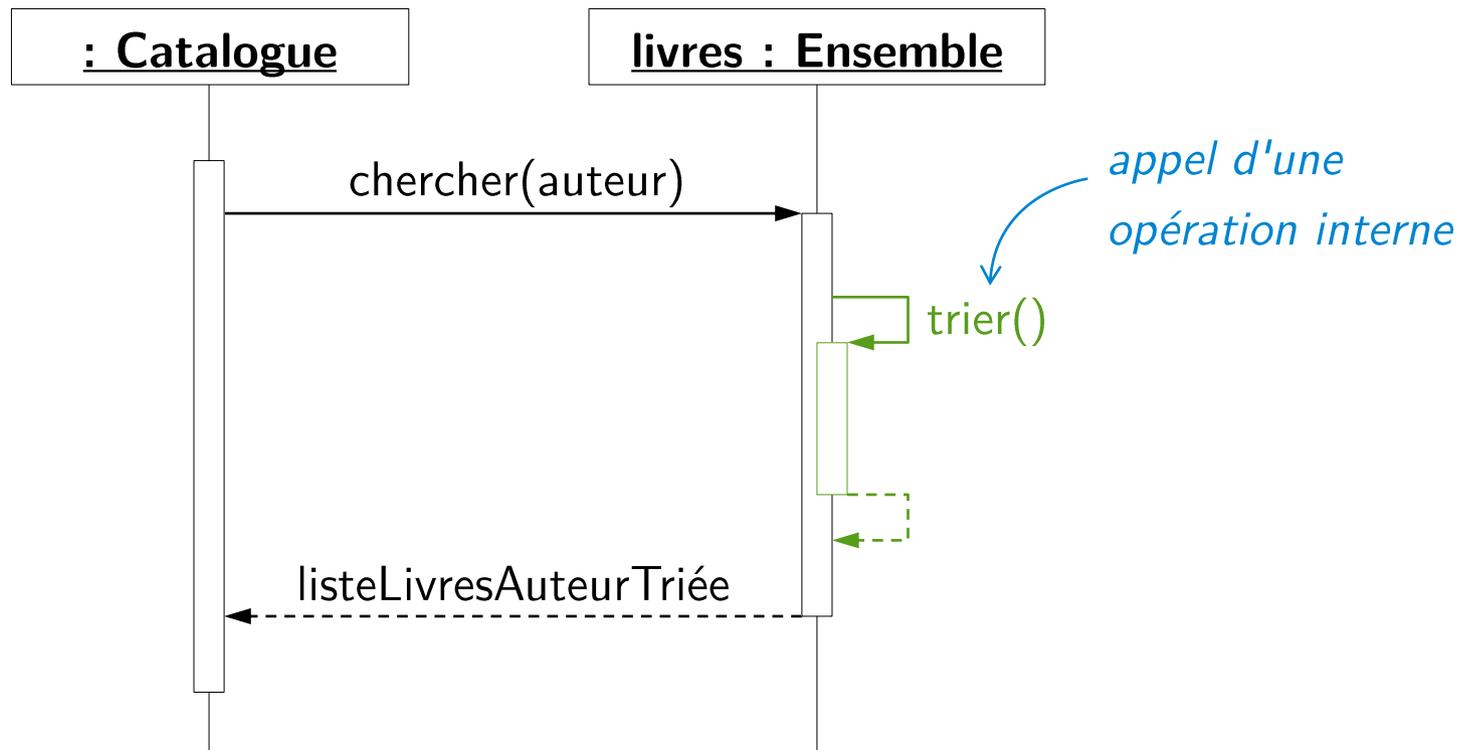
Message asynchrone : Émetteur non bloqué, continue son exécution



Création et destruction d'objet



Message réflexif

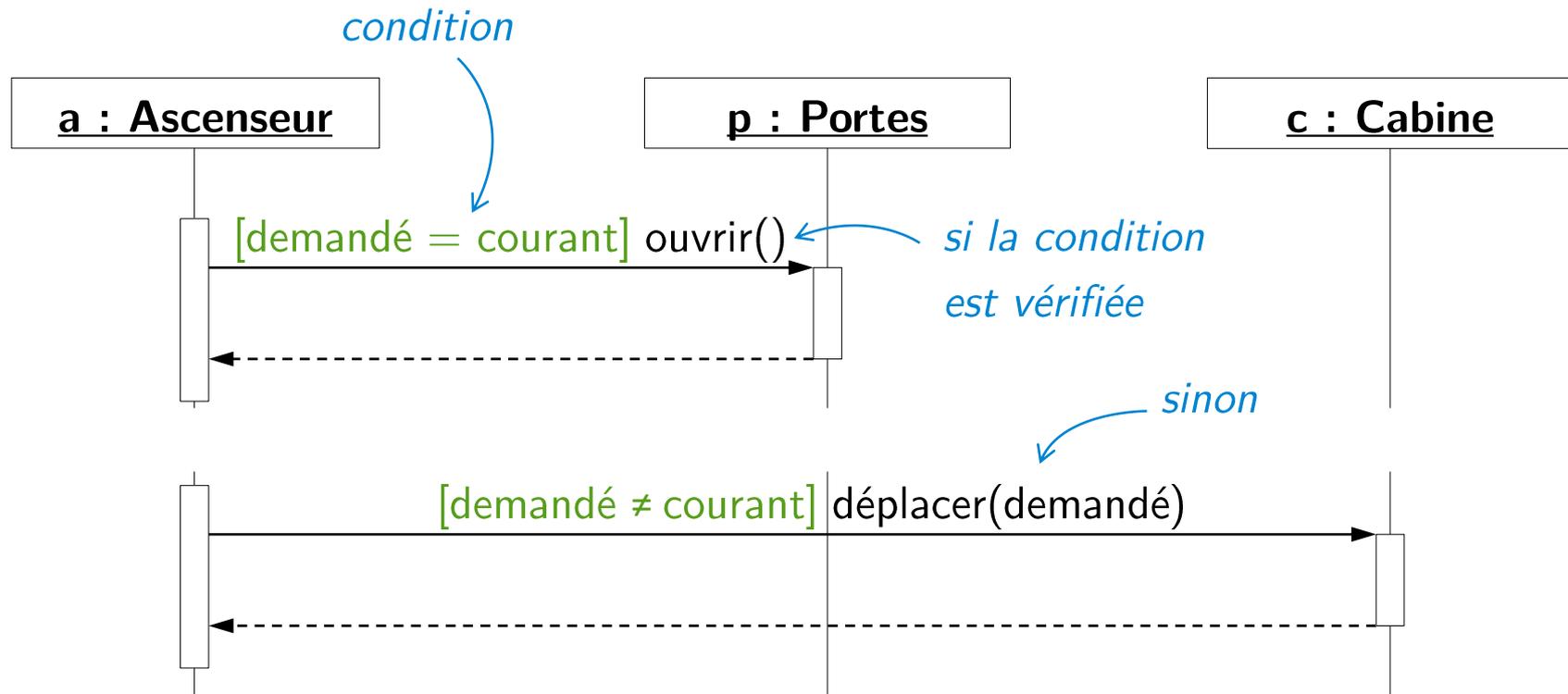


Alternative

Principe : Condition à l'envoi d'un message

Notation :

- Deux diagrammes

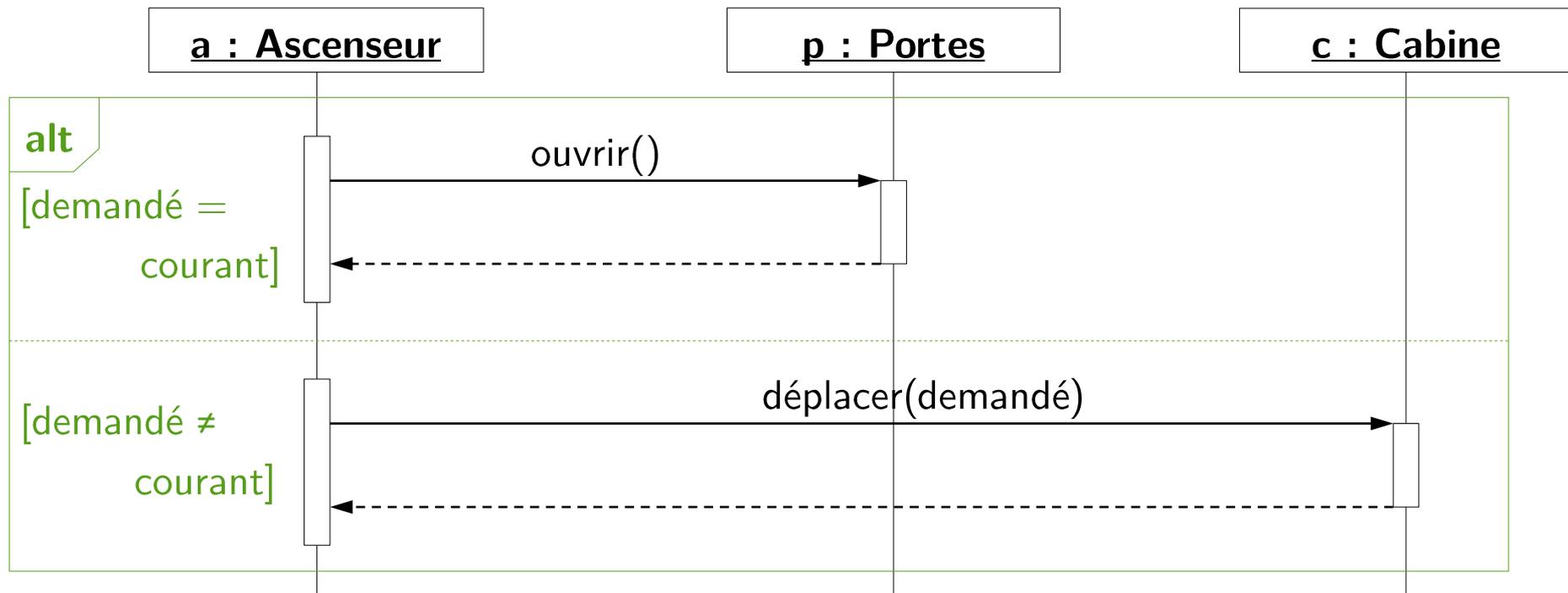


Alternative

Principe : Condition à l'envoi d'un message

Notation :

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

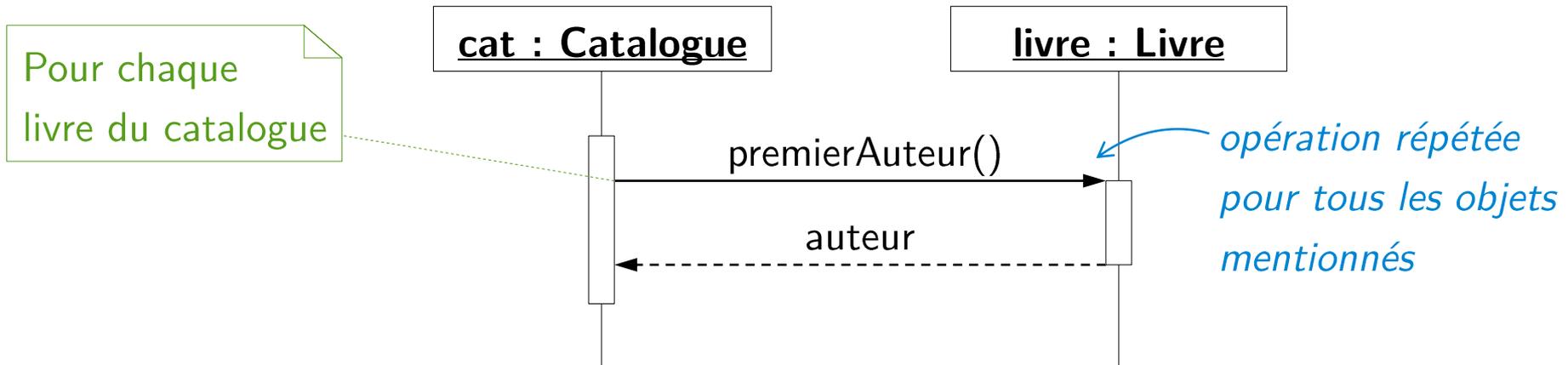


Boucle

Principe : Répéter un enchaînement de messages

Notation :

- Note

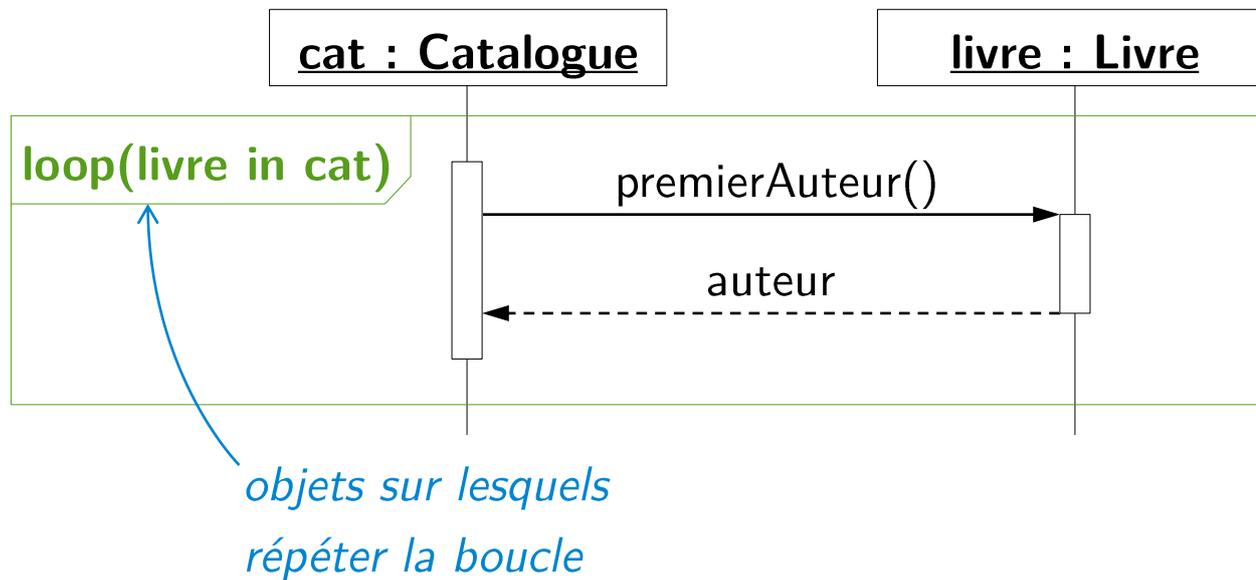


Boucle

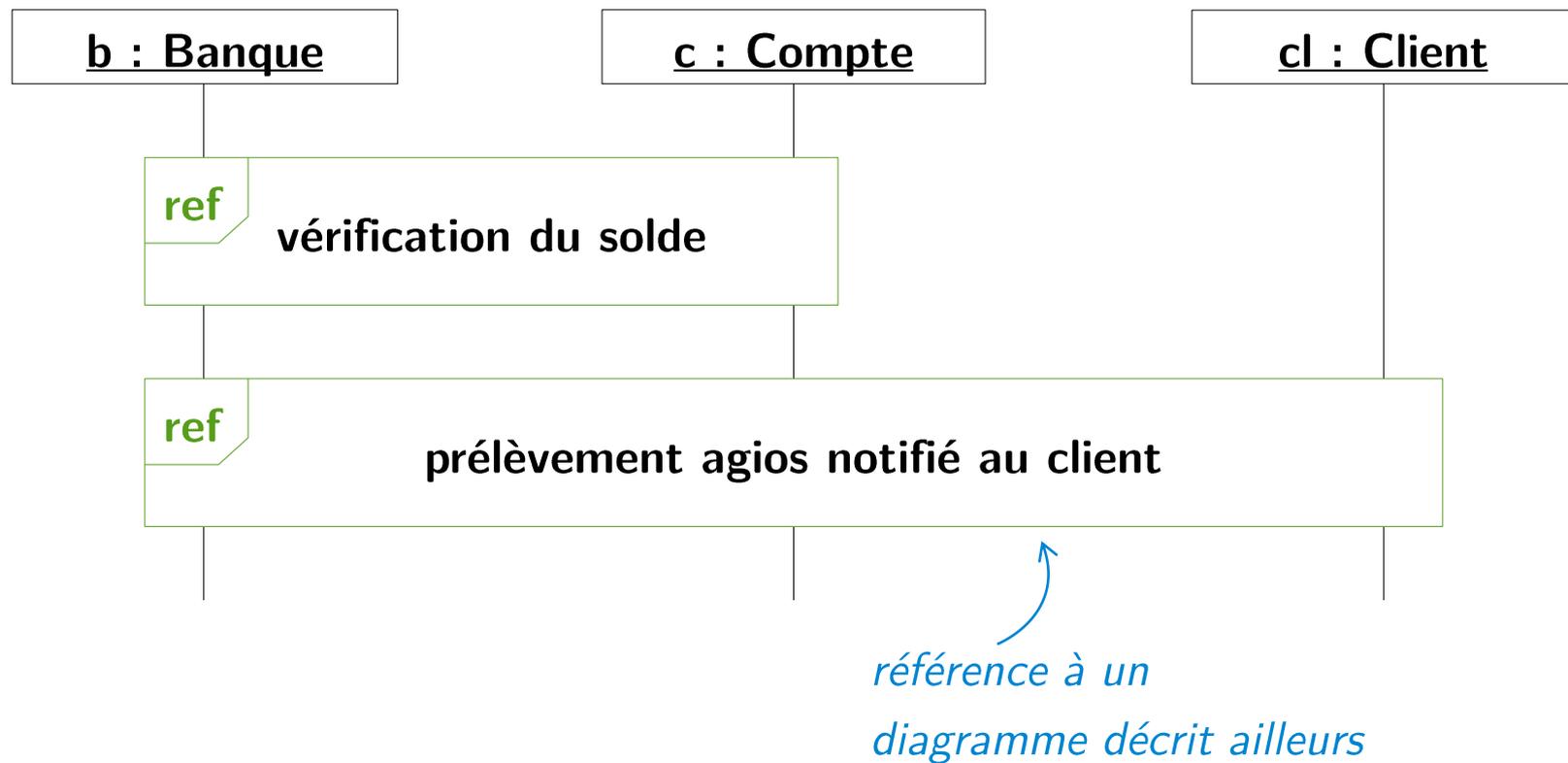
Principe : Répéter un enchaînement de messages

Notation :

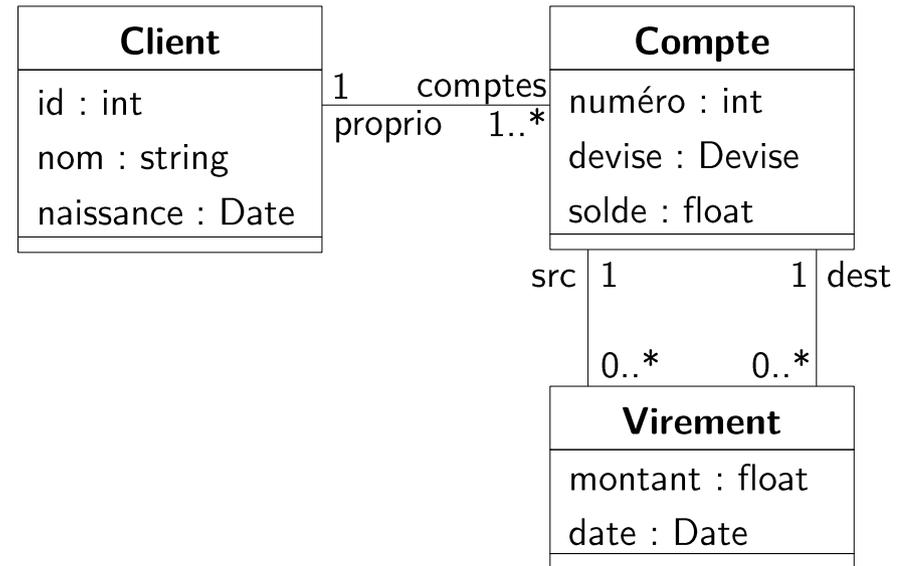
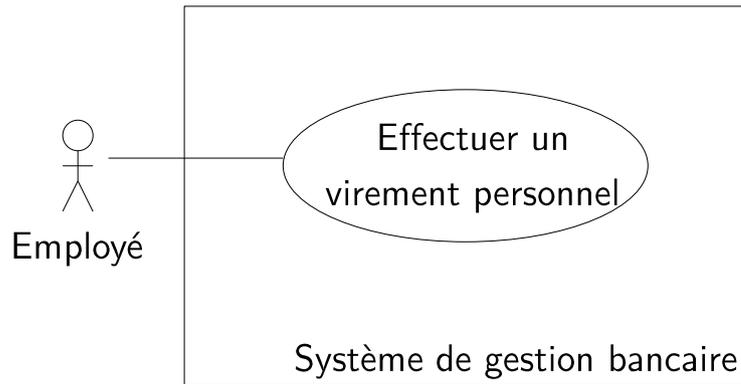
- Note
- Bloc de boucle **loop**



Référence à un autre diagramme

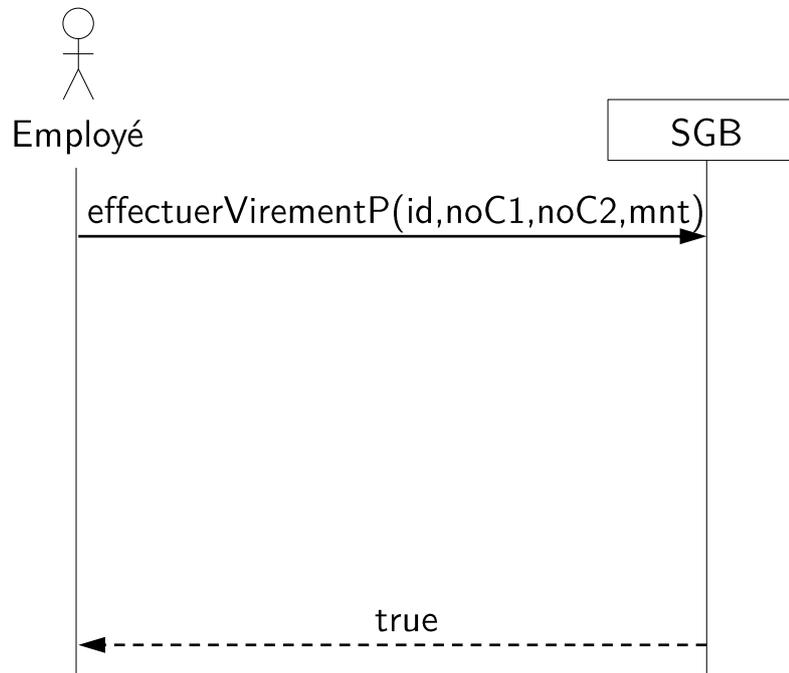


Exemple - Analyse

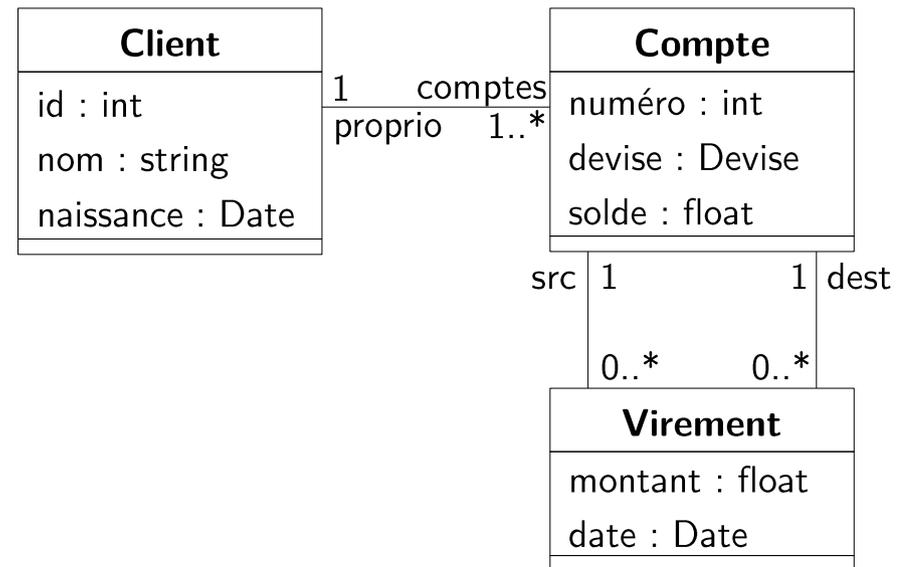
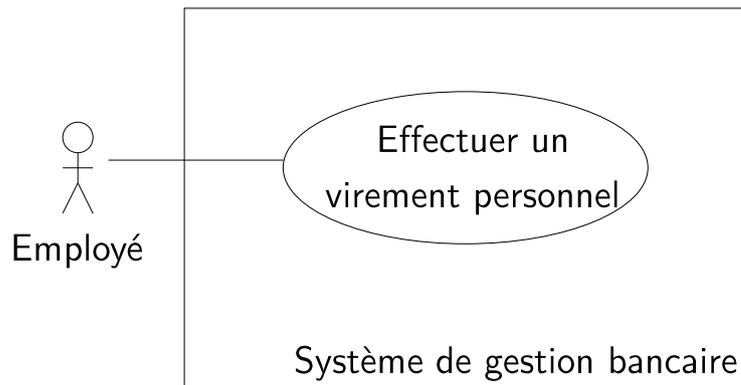


Cas d'utilisation

Effectuer un virement personnel
(niveau analyse)

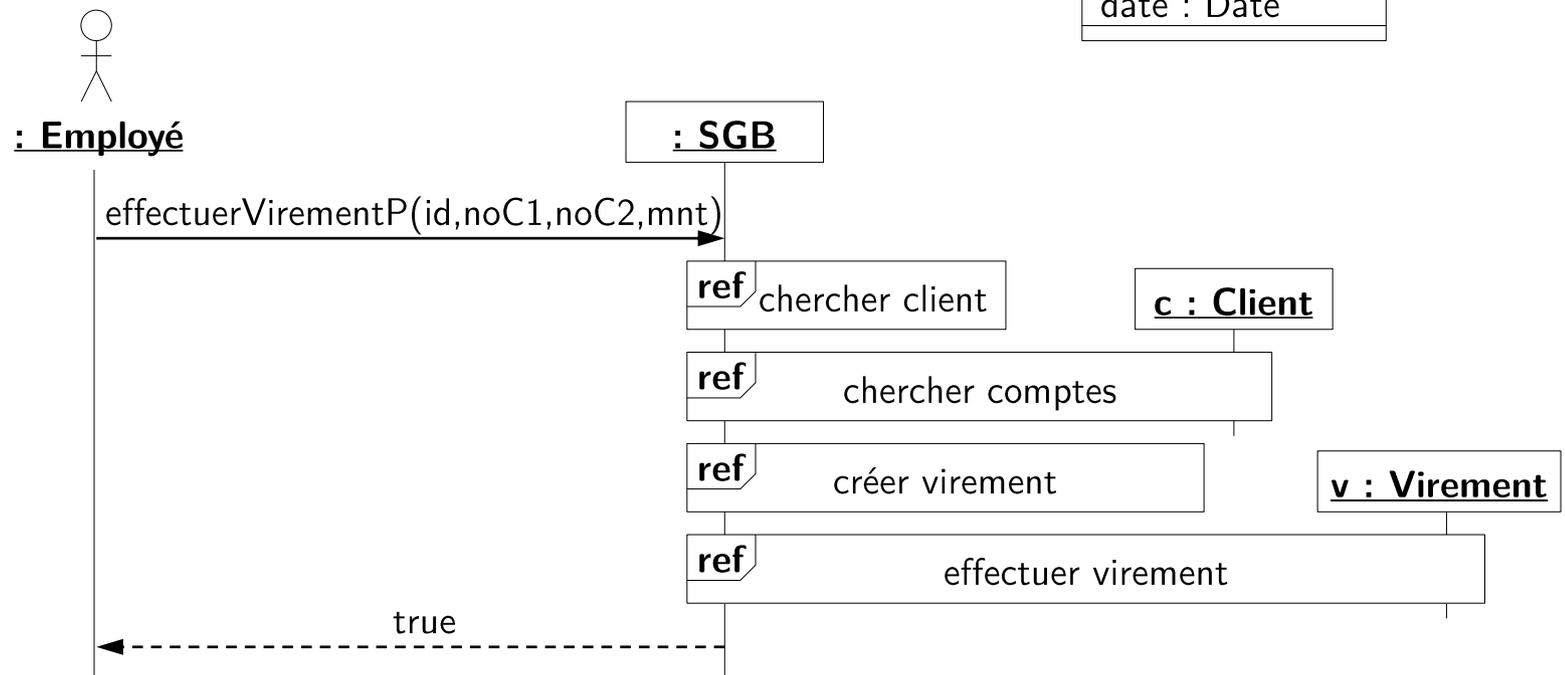


Exemple - Conception

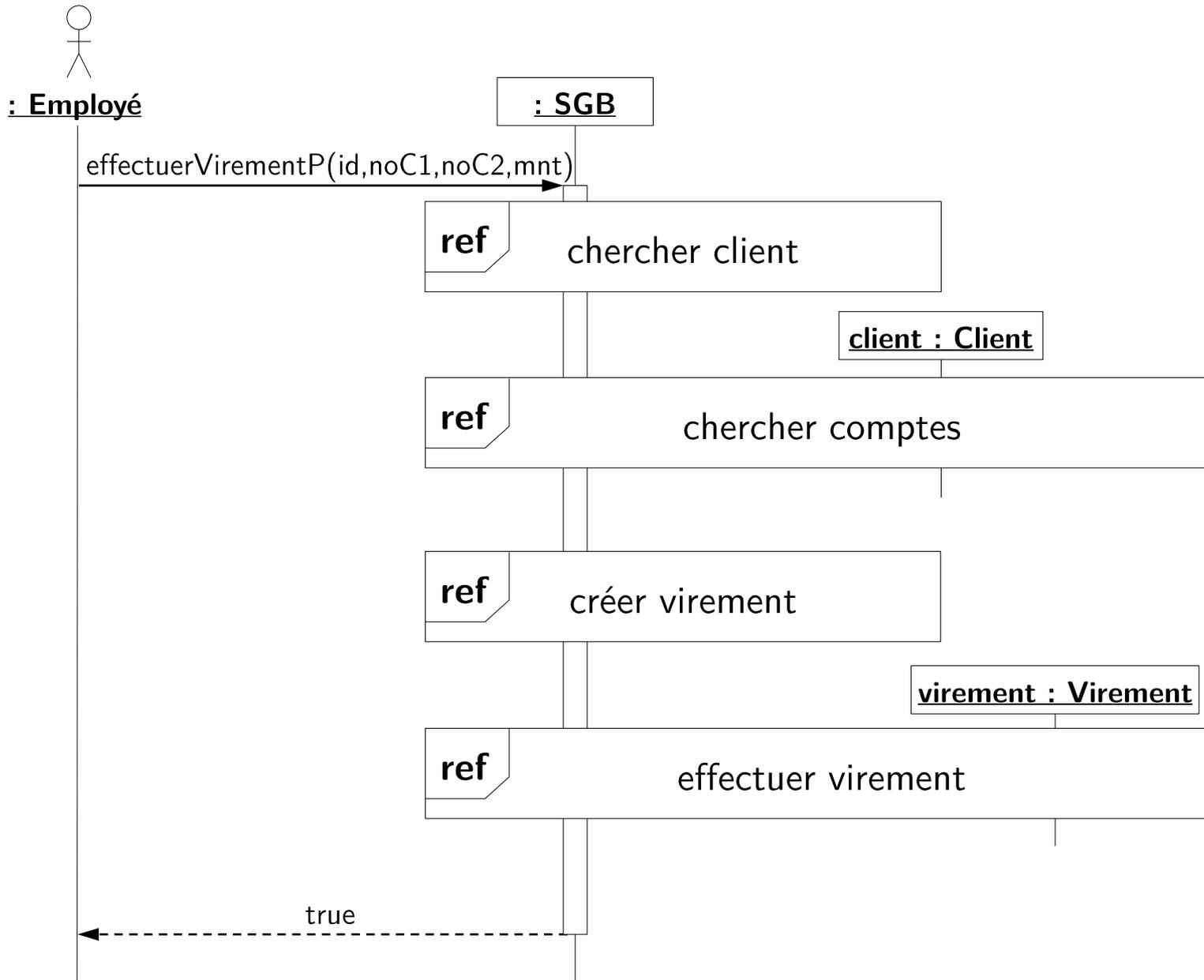


Réalisation du cas d'utilisation

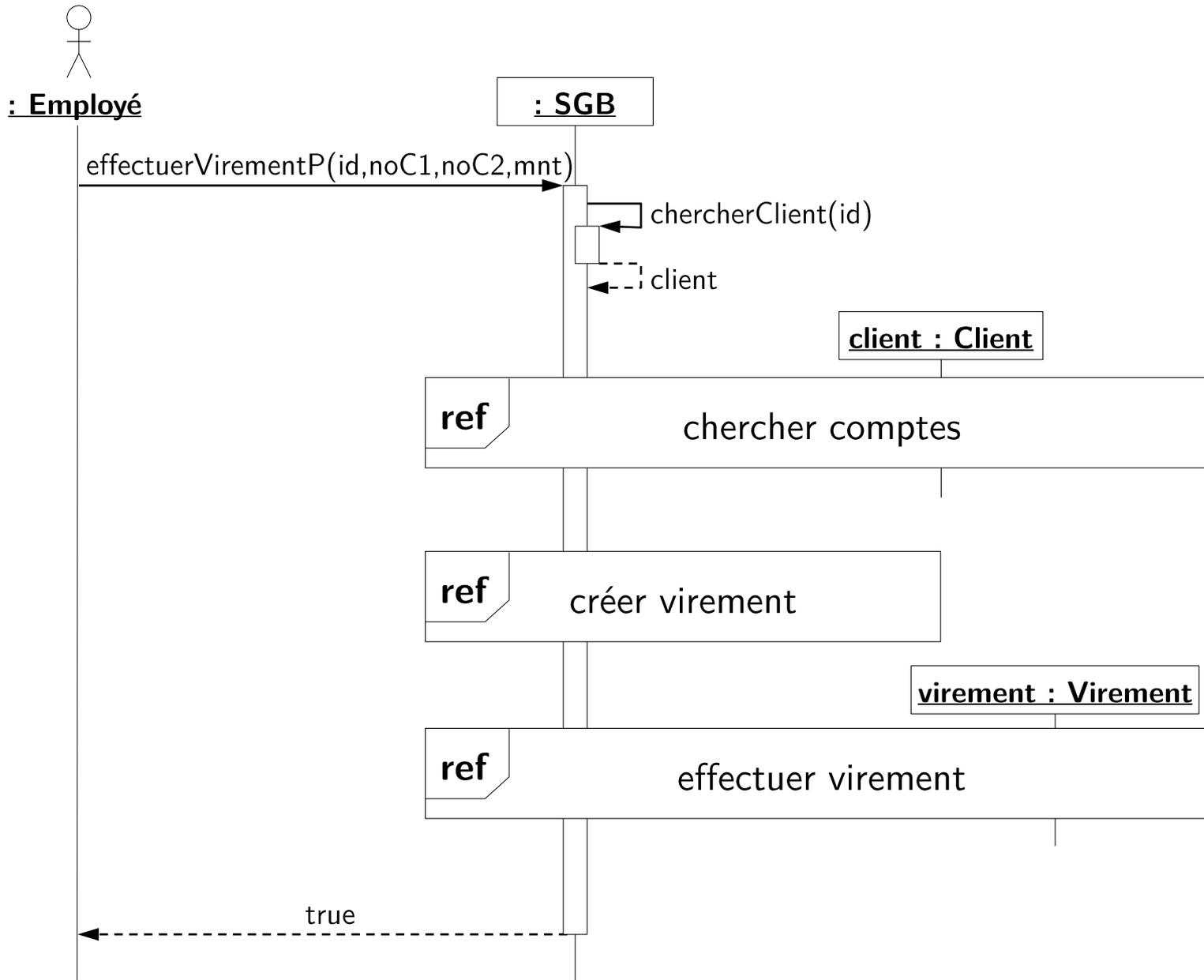
Effectuer un virement personnel
(niveau conception)



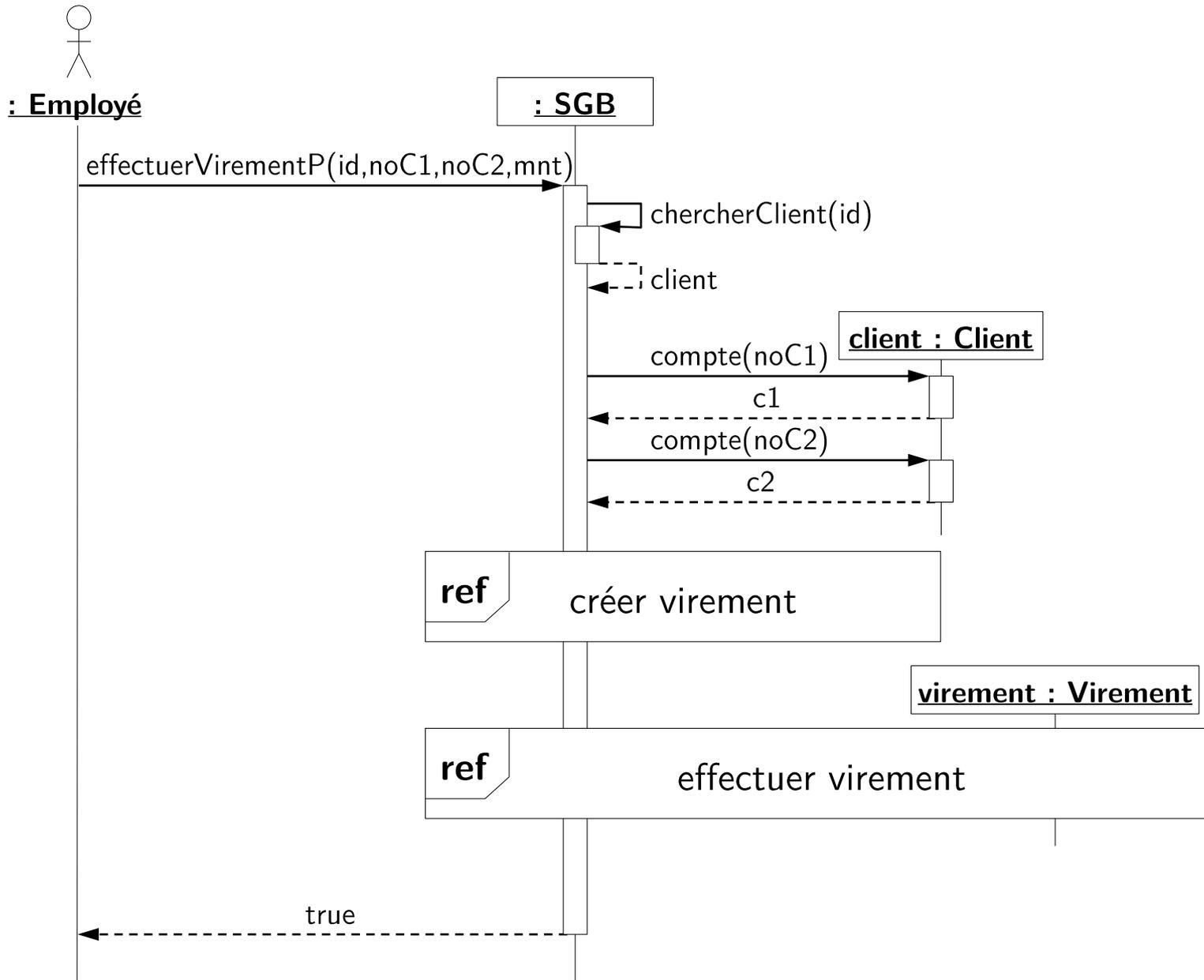
Exemple - Conception



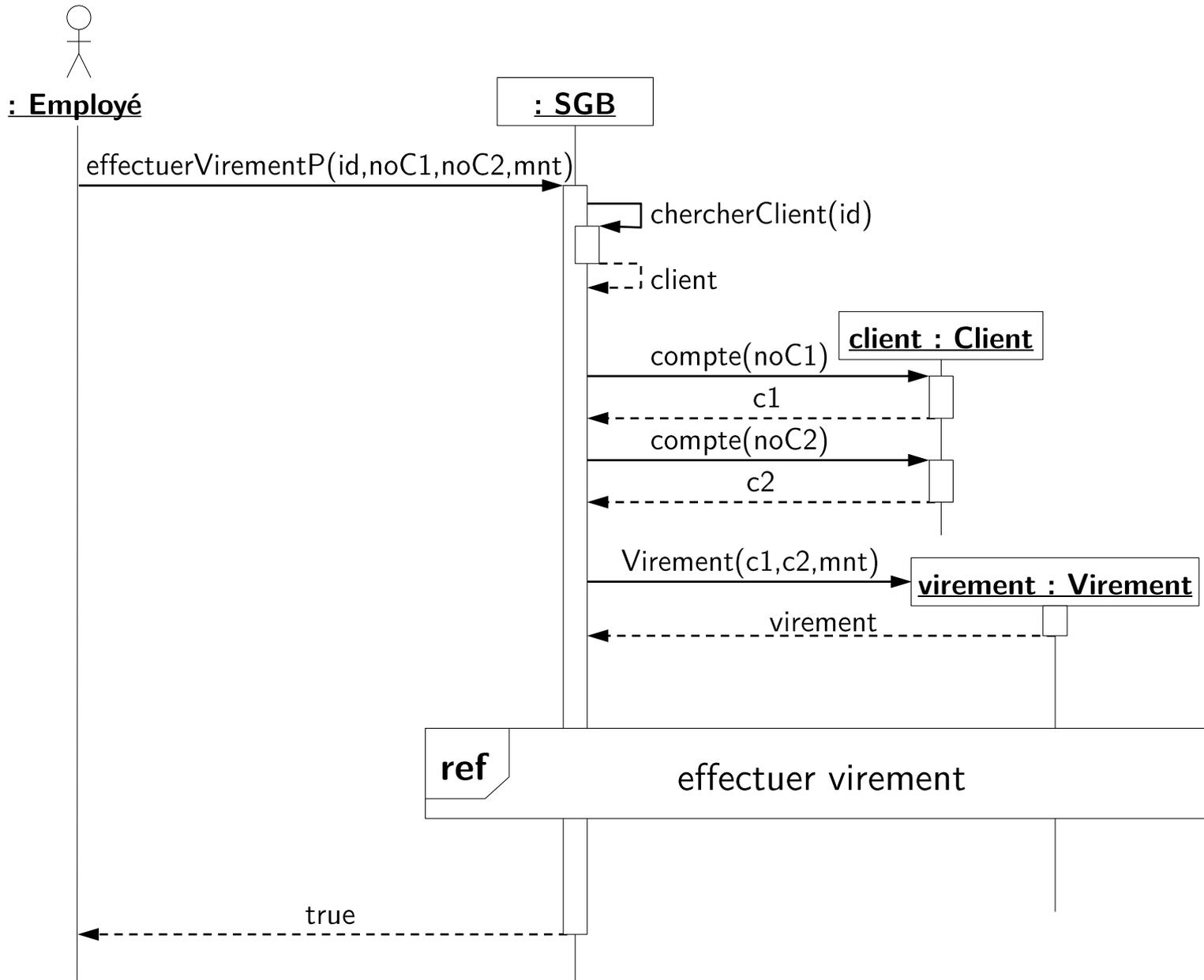
Exemple - Conception



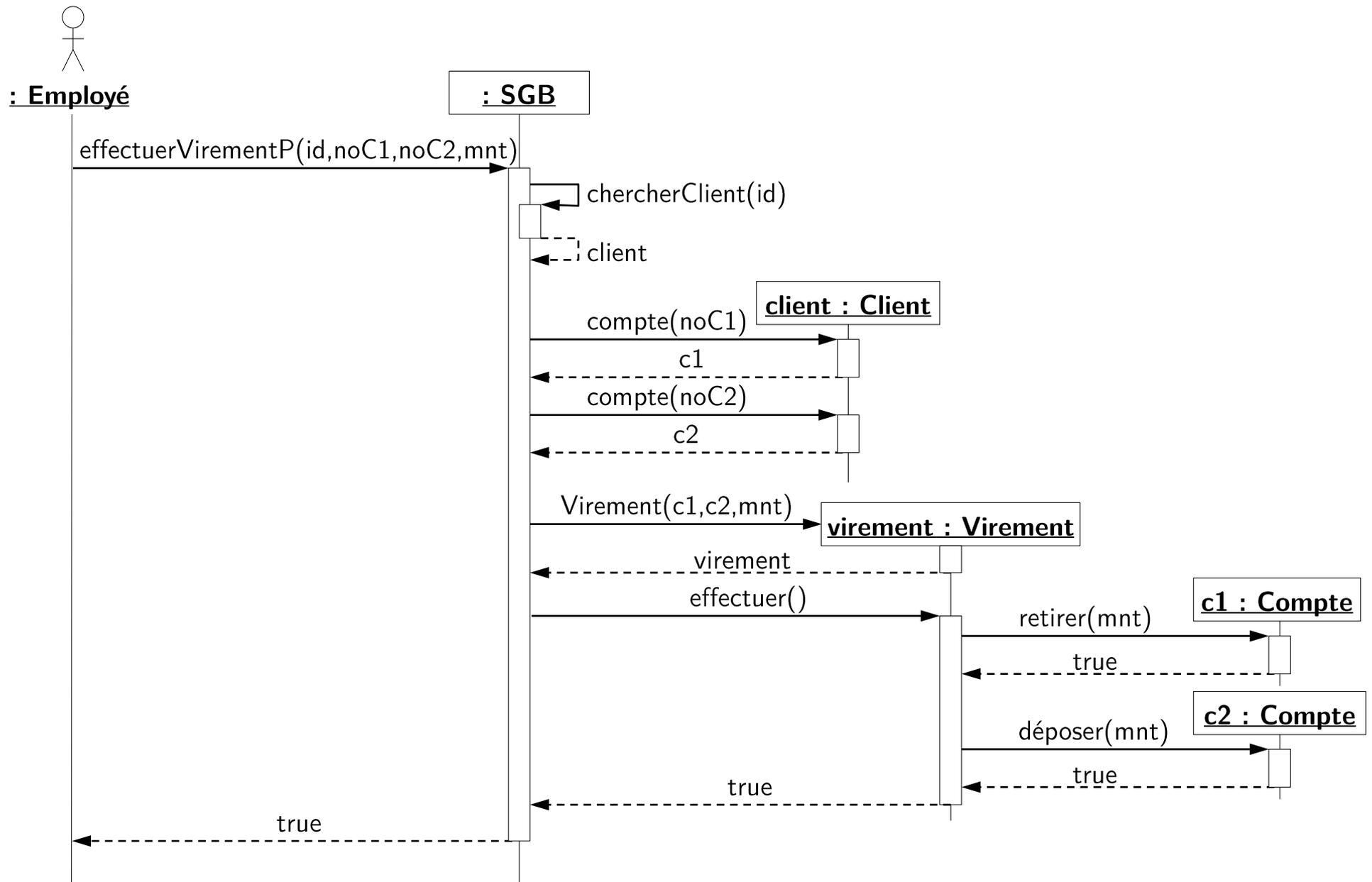
Exemple - Conception



Exemple - Conception

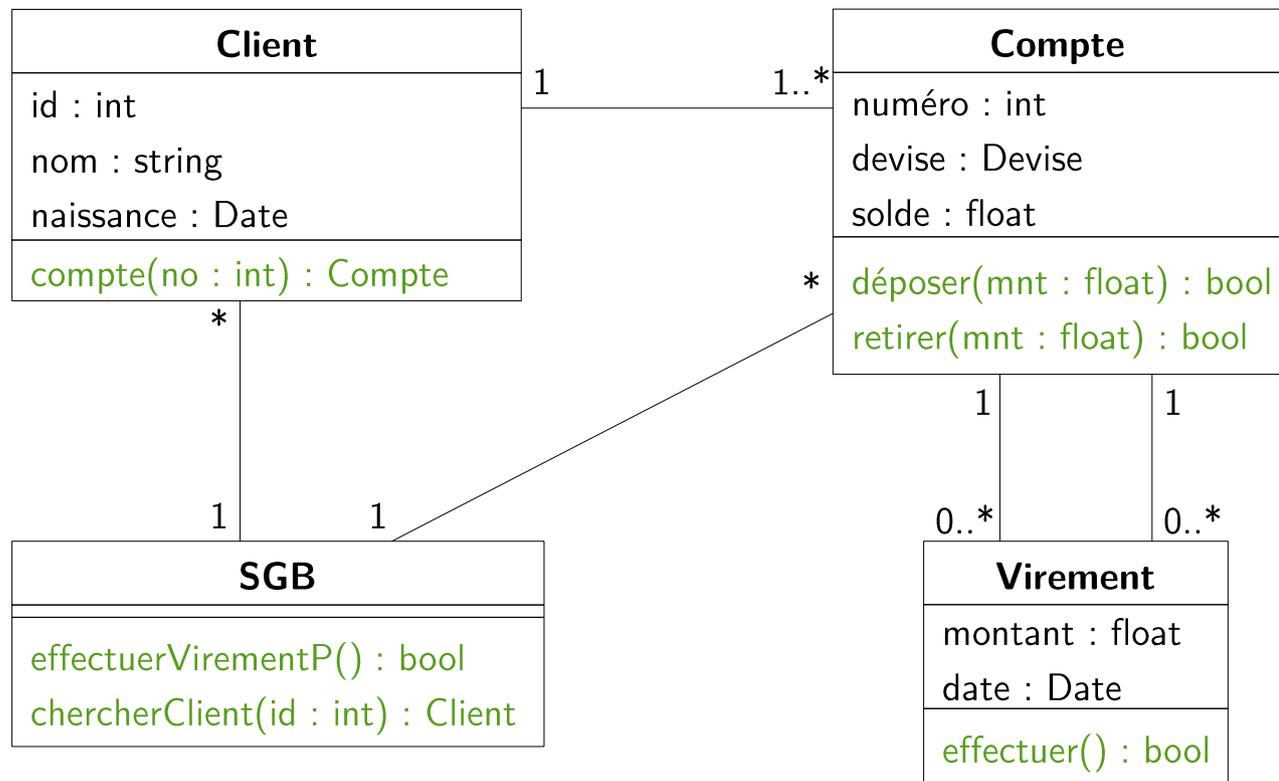


Exemple - Conception



Exemple - Conception

Diagramme de classes **complété** avec les classes techniques et les opérations nécessaires



Quelques règles

Messages entre acteurs et interface

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

Messages au sein du système

- **Opérations** du diagramme de classes
- Si message de **objA : ClasseA** vers **objB : ClasseB**, alors opération du message dans ClasseB