

DIAGRAMME DE SÉQUENCE

DÉFINITION

- Le diagramme de séquence sert à 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

IL COMPREND

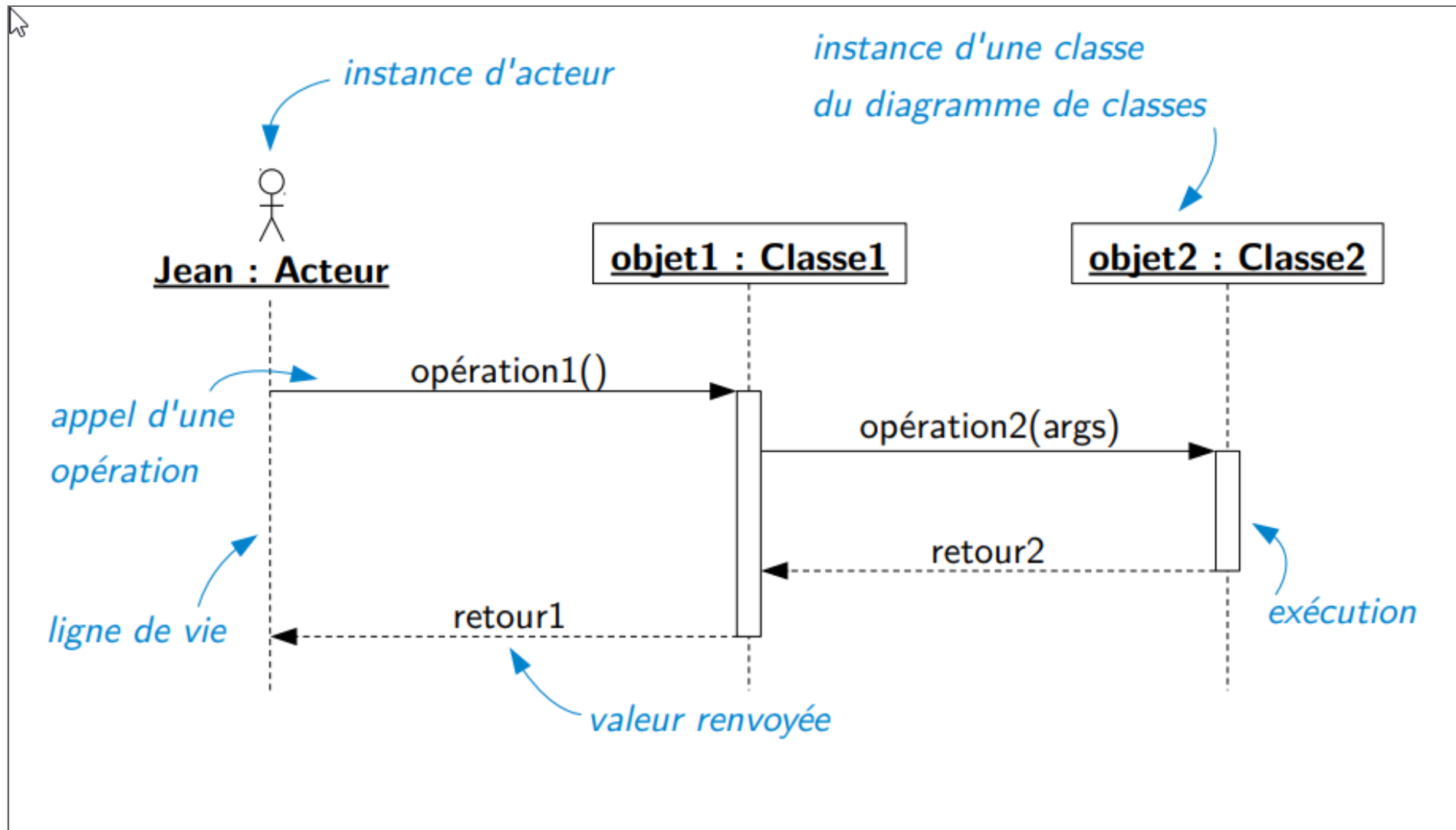
ÉLÉMENTS DU DIAGRAMME DE SÉQUENCE

- 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
- Echanges de messages représentés horizontalement

EXEMPLE



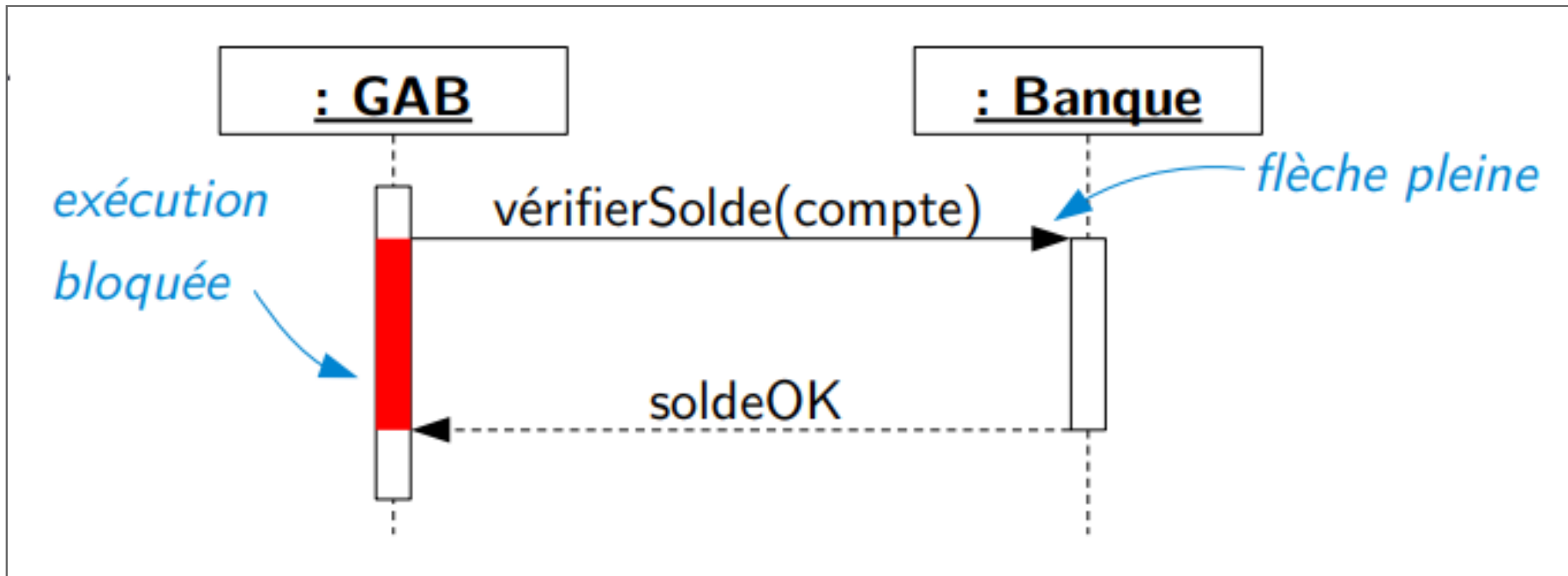
MESSAGES

- Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie :
 - Ils sont représentés par des flèches
 - Ils sont présentés du haut vers le bas le long de la ligne de vie, dans un ordre chronologique
- Un message définit une communication particulière entre des lignes de vie (objets ou acteurs)

- Plusieurs types de messages existent, dont les plus courants:
 - l'envoi d'un signal;
 - l'invocation d'une opération (appel de méthode);
 - la création ou la destruction d'un objet.
- La réception des messages provoque une période d'activité (rectangle vertical sur la ligne de vie) marquant le traitement du message (spécification d'exécution dans le cas d'un appel de méthode).

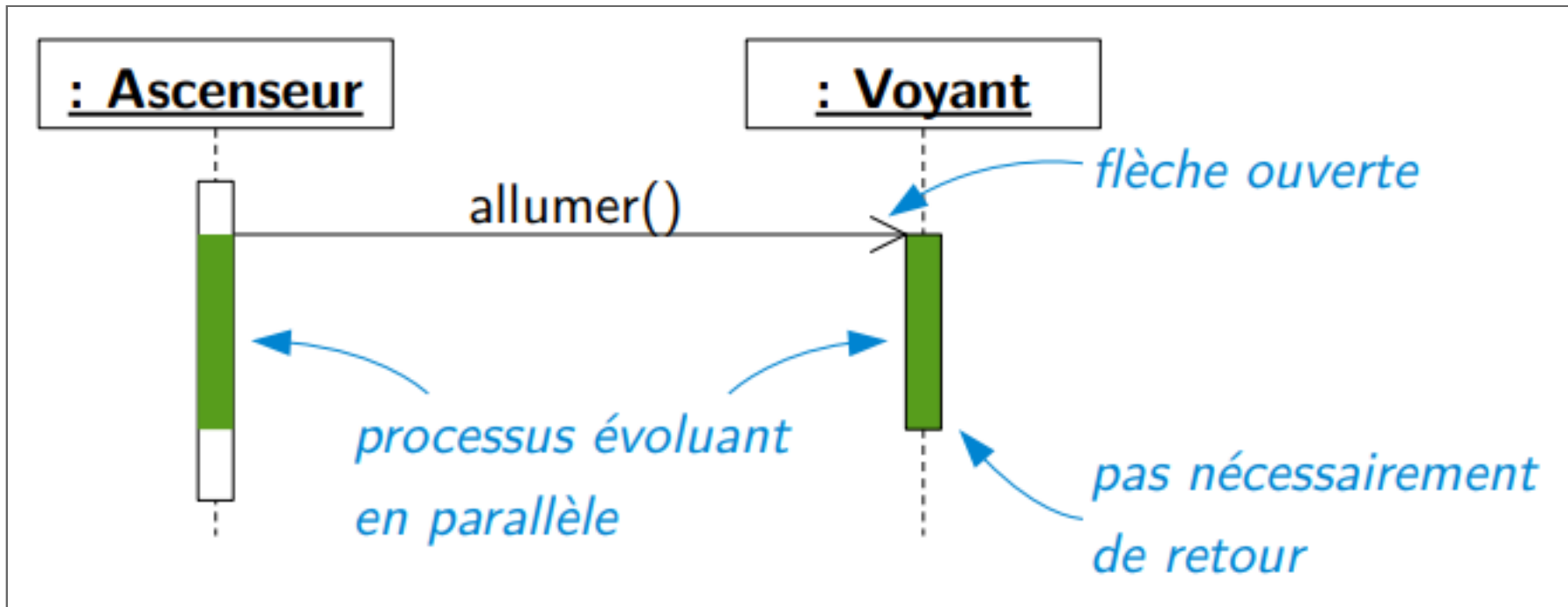
MESSAGES SYNCHRONES

- Un message synchrone bloque l'expéditeur jusqu'à la réponse du destinataire. Le flot de contrôle passe de l'émetteur au récepteur.
- Si un objet A envoie un message synchrone à un objet B, A reste bloqué tant que B n'a pas terminé.



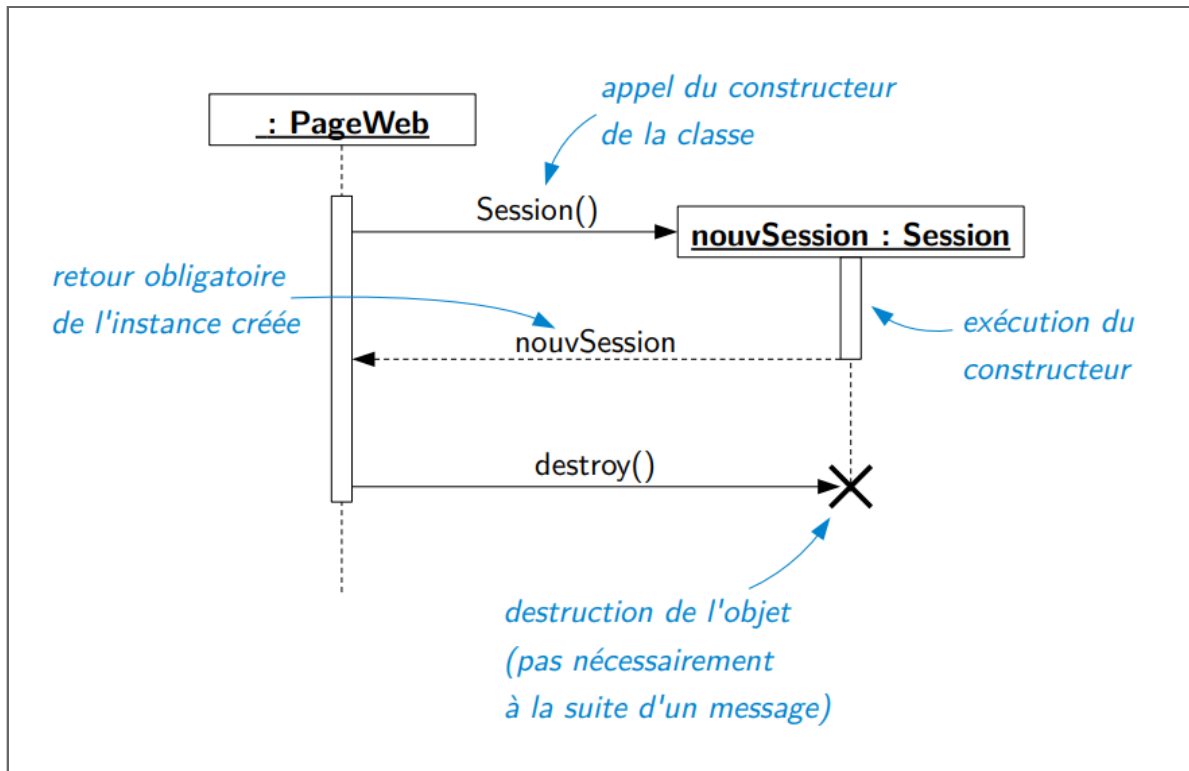
MESSAGES ASYNCHRONES

- Un message asynchrone n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré.

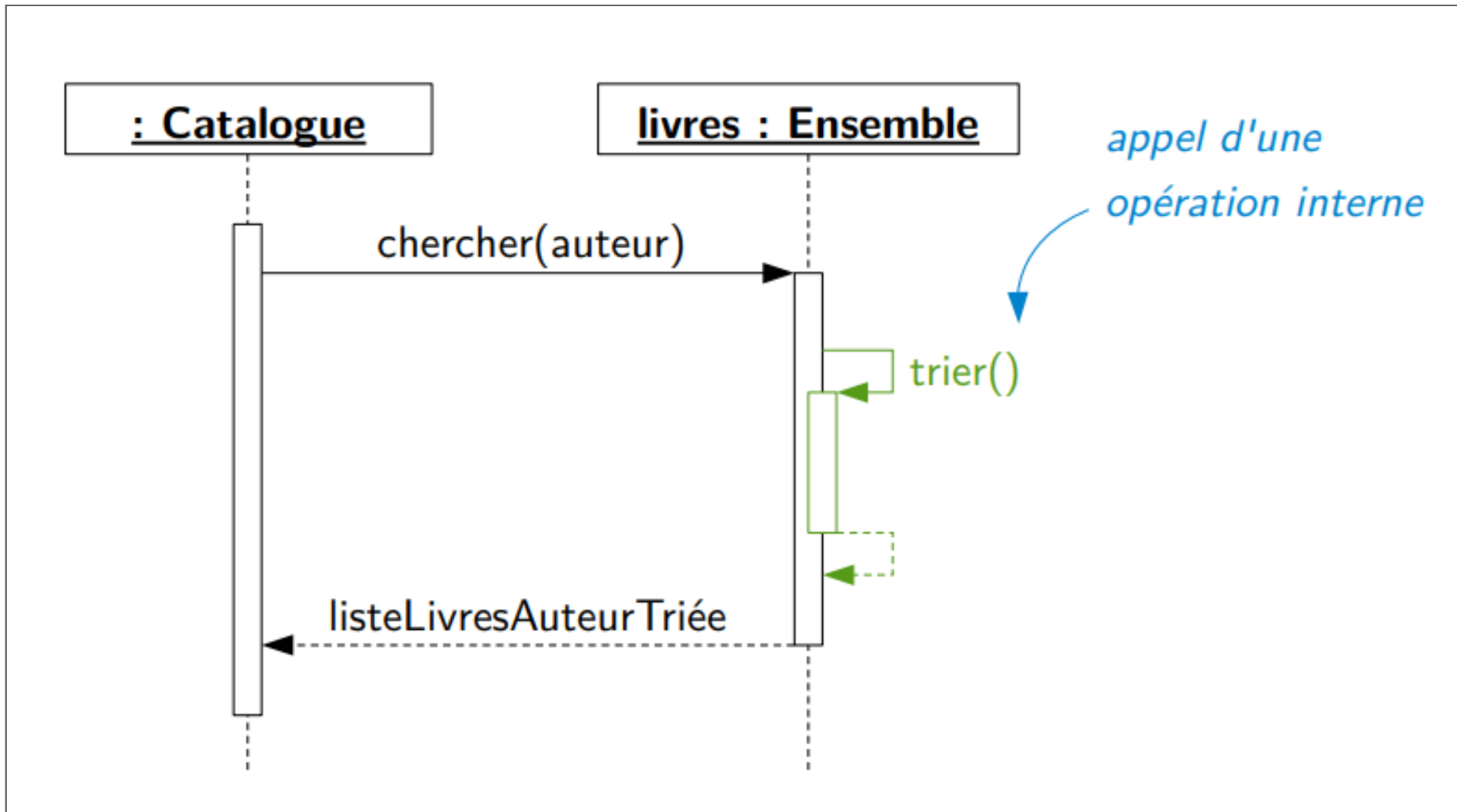


CRÉATION ET DESTRUCTION D'OBJETS

- Création : message asynchrone stéréotypé << create >> pointant vers le rectangle en tête de la ligne de vie
- Destruction : message asynchrone stéréotype << destroy >> précédant une croix sur la ligne de vie



MESSAGE RÉFLEXIF

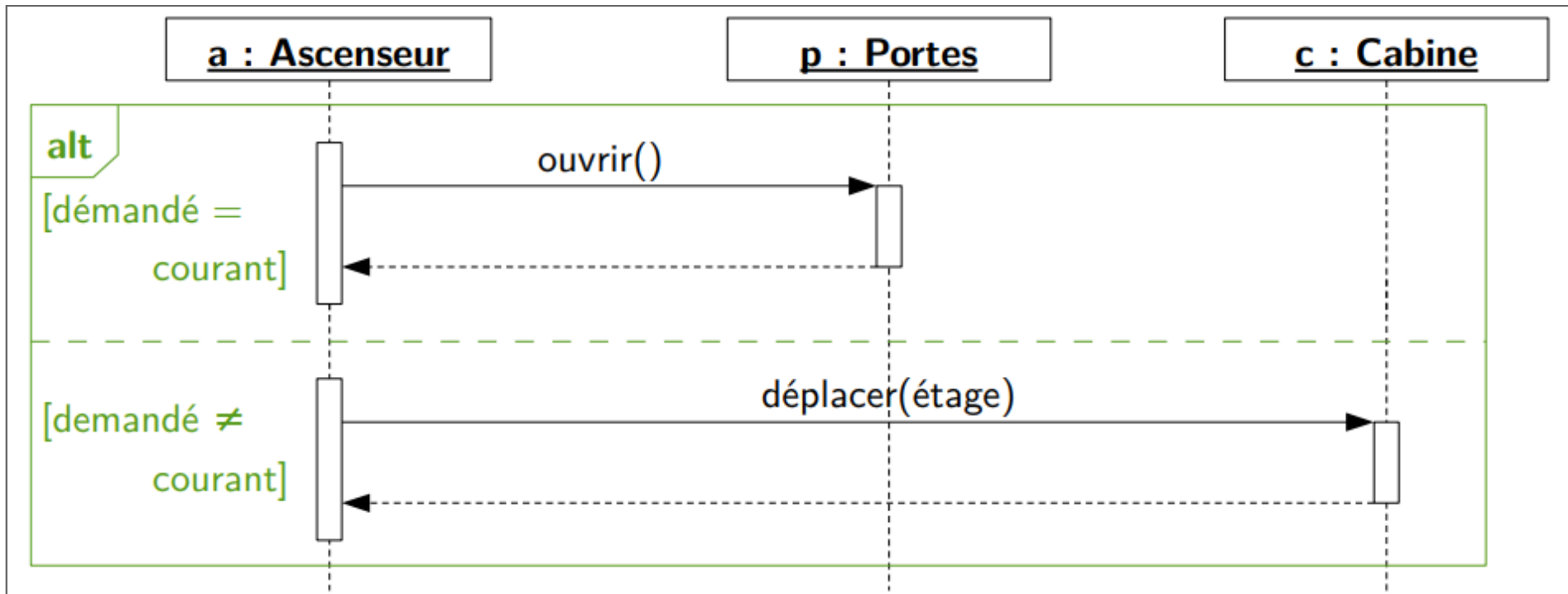


FRAGMENT COMBINÉ

- Un fragment combiné permet de décomposer une interaction complexe en fragments suffisamment simples pour être compris.
- Un fragment combiné se représente de la même façon qu'une interaction. Il est représenté un rectangle dont le coin supérieur gauche contient un pentagone.
- Dans le pentagone figure le type de la combinaison (appelé opérateur d'interaction).

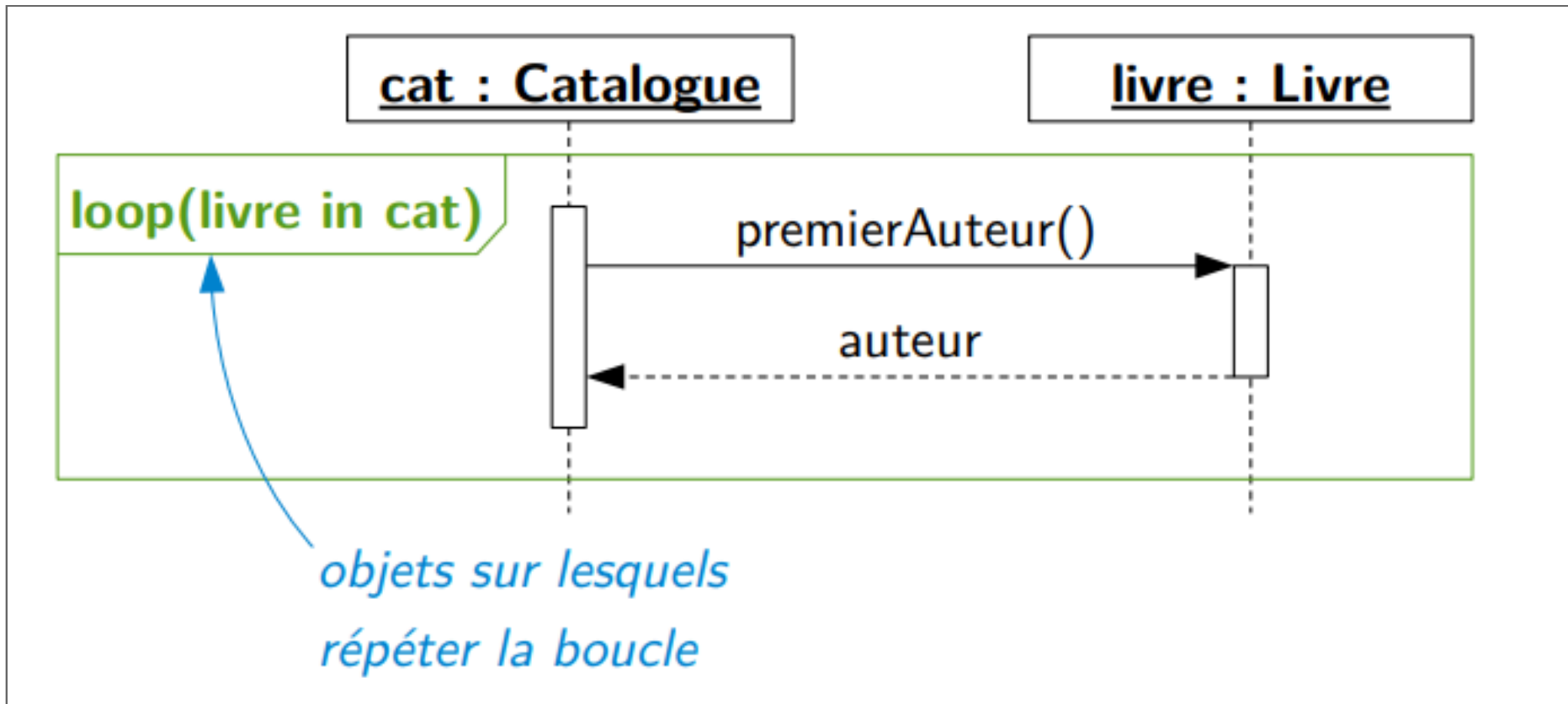
FRAGMENT ALT : OPÉRATEUR CONDITIONNEL

- Les différentes alternatives sont spécifiées dans des zones délimitées par des pointillés.
- Les conditions sont spécifiées entre crochets dans chaque zone.
- On peut utiliser une clause [else]



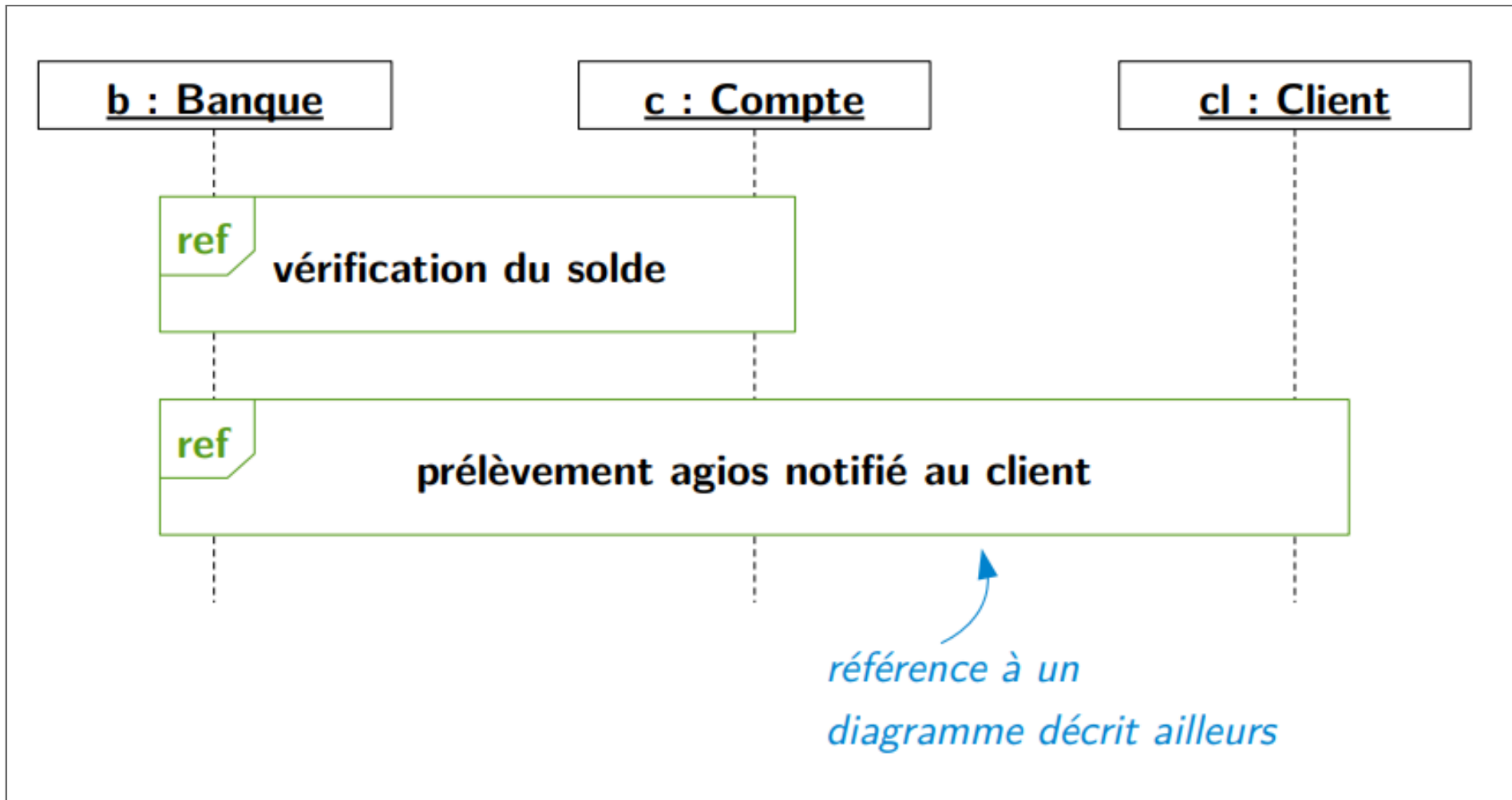
FRAGMENT LOOP : OPÉRATEUR D'ITÉRATION

- Le fragment **loop** permet de répéter ce qui se trouve en son sein.
- On peut spécifier entre crochets à quelle condition continuer.



FRAGMENT REF : RÉUTILISATION DE SÉQUENCES

- Un fragment **ref** permet d'indiquer la réutilisation d'un diagramme de séquences défini par ailleurs.



Speaker notes