

MCPO

Diagrammes de séquence

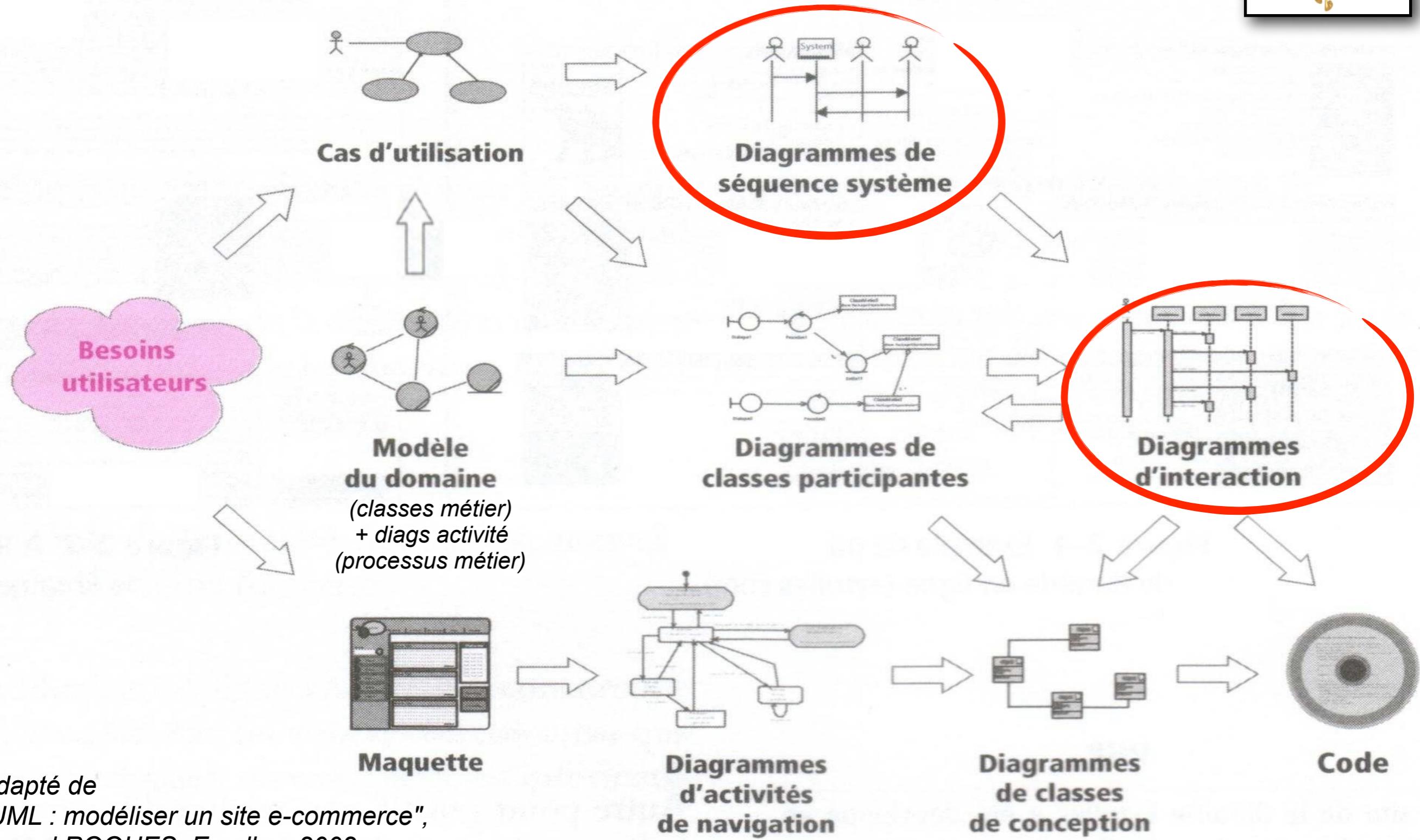
Ileana Ober

Université Paul Sabatier
IRIT

<http://www.irit.fr/~Ileana.Ober/>

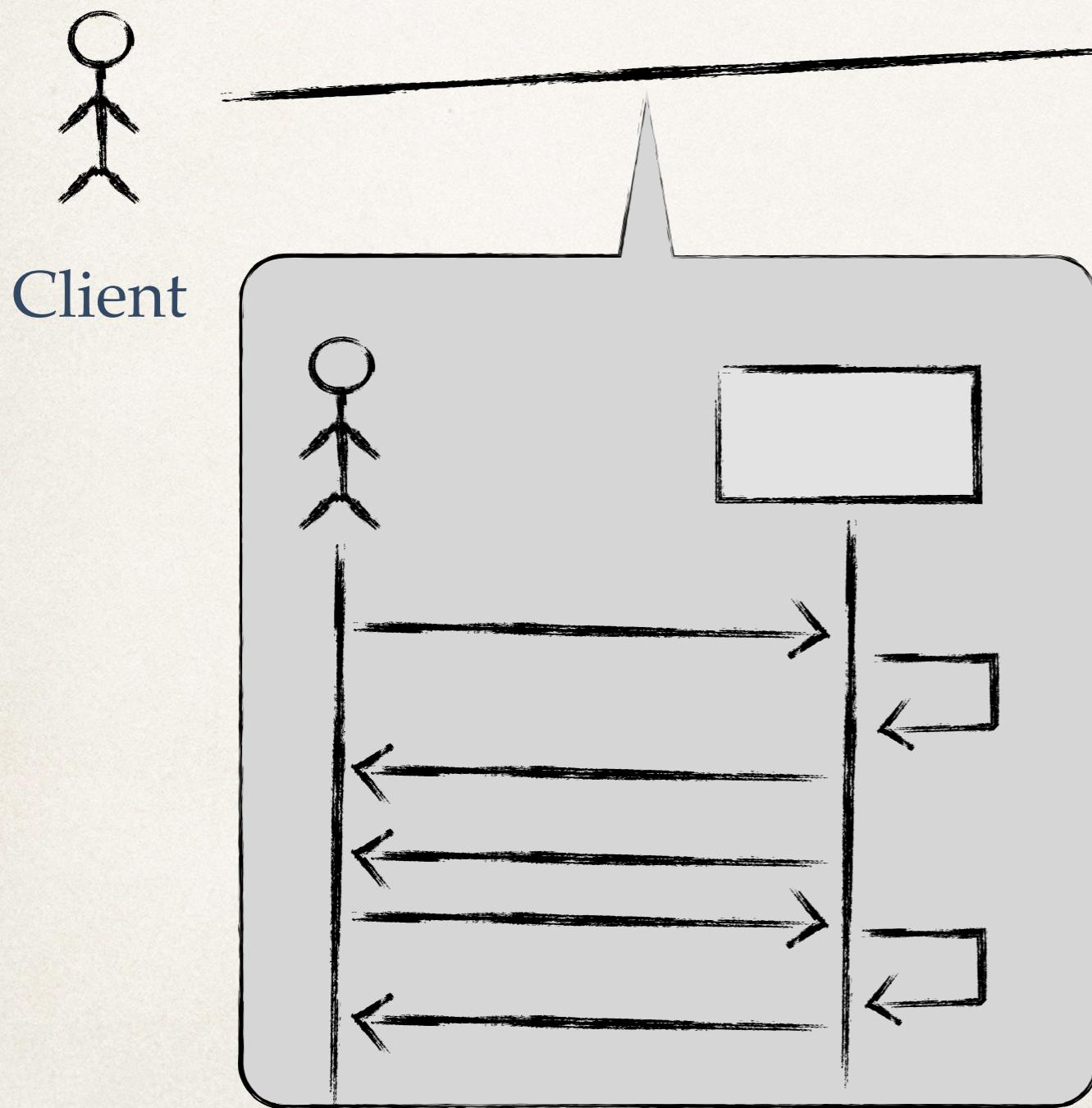


Une démarche



Adapté de
"UML : modéliser un site e-commerce",
Pascal ROQUES, Eyrolles, 2002

Description de l'interaction



RetirerDeLArgent
AuDistributeur

A travers des
diagrammes de séquences
système

Les diagrammes séquence

sequence diagram

- ❖ But :
 - ❖ détailler l'**interaction** entre plusieurs objets / acteurs
 - ❖ donner des **scénarios d'exécution** significatifs
- ❖ Mettent en évidence
 - ❖ les **lignes de vie** des objets / acteurs
 - ❖ les **échanges** de messages et leur **ordre**
 - ❖ **chronologie** des messages échangés entre objets (et acteurs)

DS système / DS détaillé



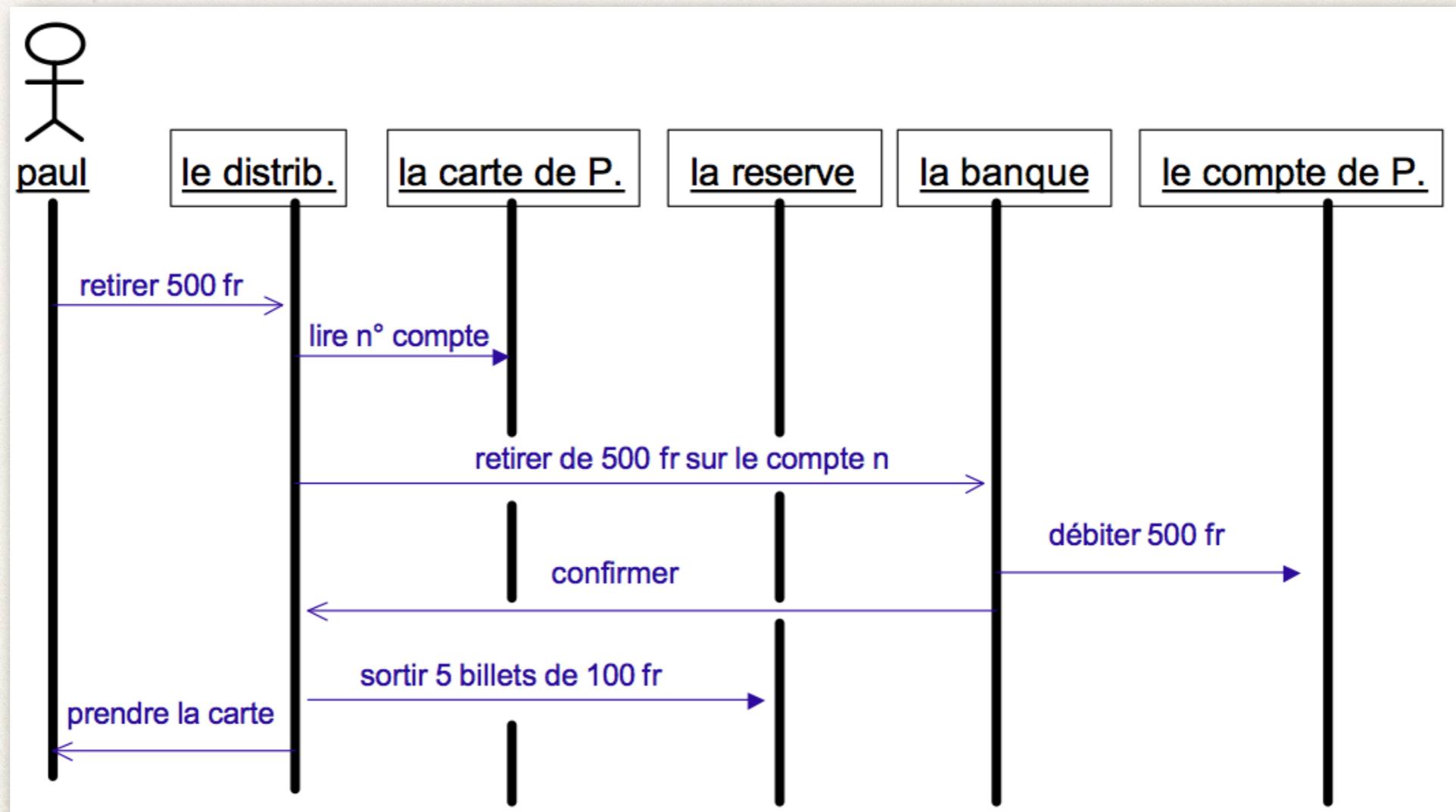
- ❖ En fonction de l'étape à laquelle le diagramme est élaboré on a
 - ❖ **DS système** spécifie les interactions entre le système (vu comme un tout) et un (des) acteur(s) externe(s)
 - ❖ **DS détaillé** spécifie aussi les interactions entre objets internes au système
- ❖ Les deux détaillent un cas d'utilisation, mais à un **grain** différent
- ❖ La **syntaxe** du DS ne dépend pas de son type

Plan

- ❖ Diagramme de séquence parmi les autres diagrammes UML
- ❖ Notions de base
- ❖ Utilisation

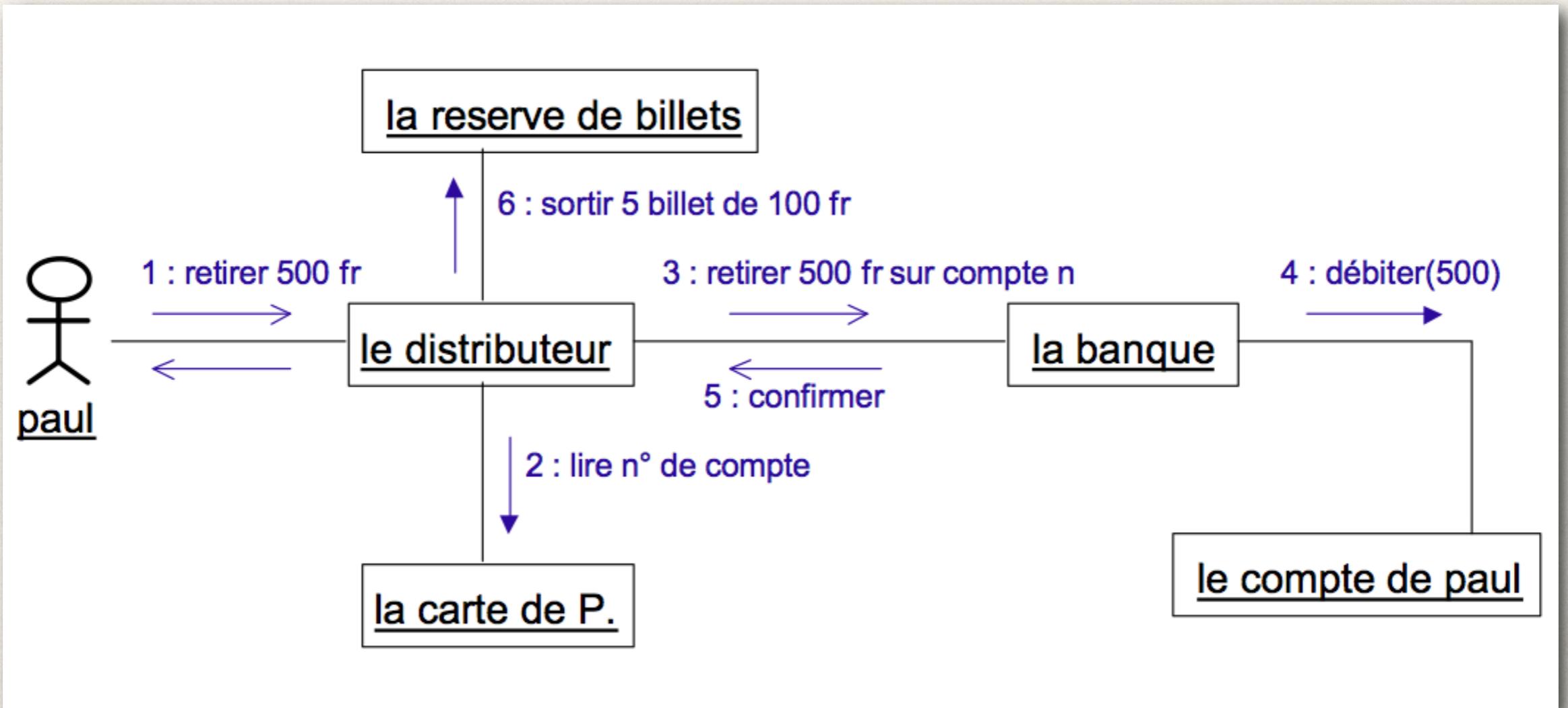
Spécification de l'interaction

Diagramme séquence

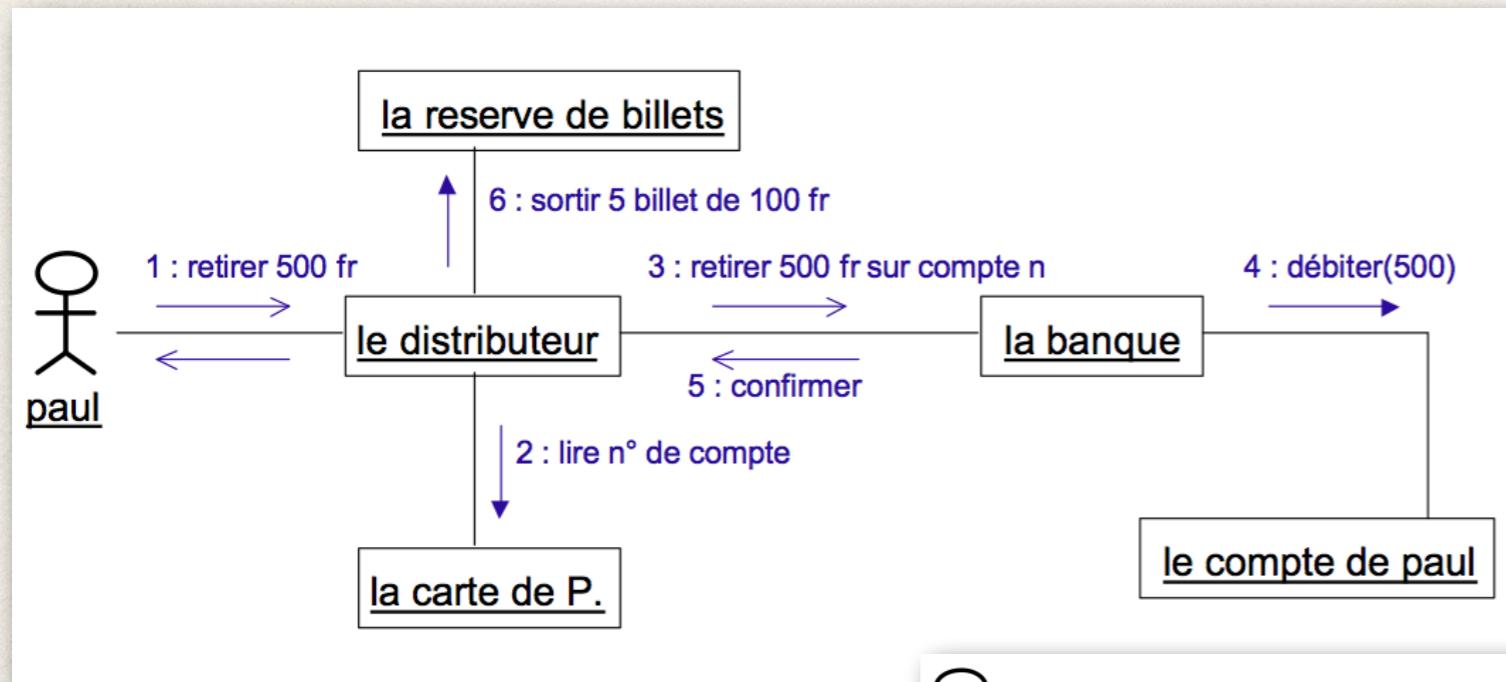


Spécification de l'interaction

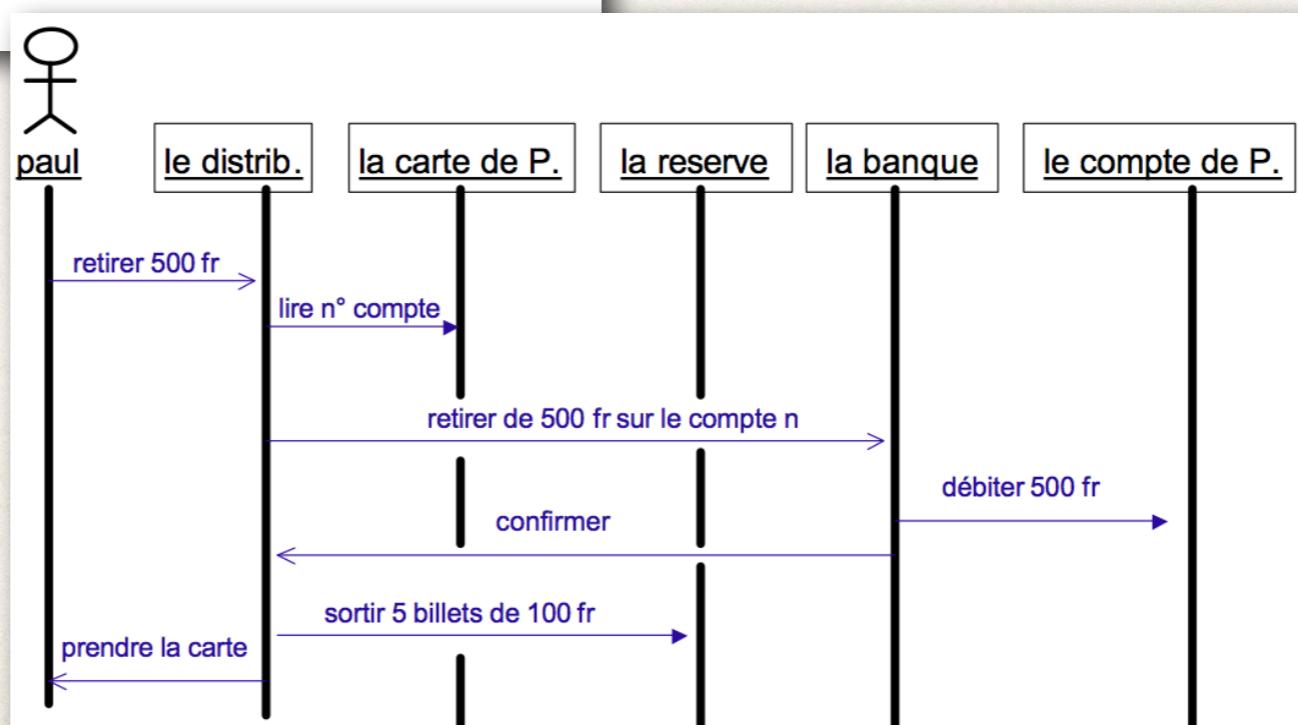
Diagramme collaboration



Les deux diagrammes d'interactions sont équivalentes



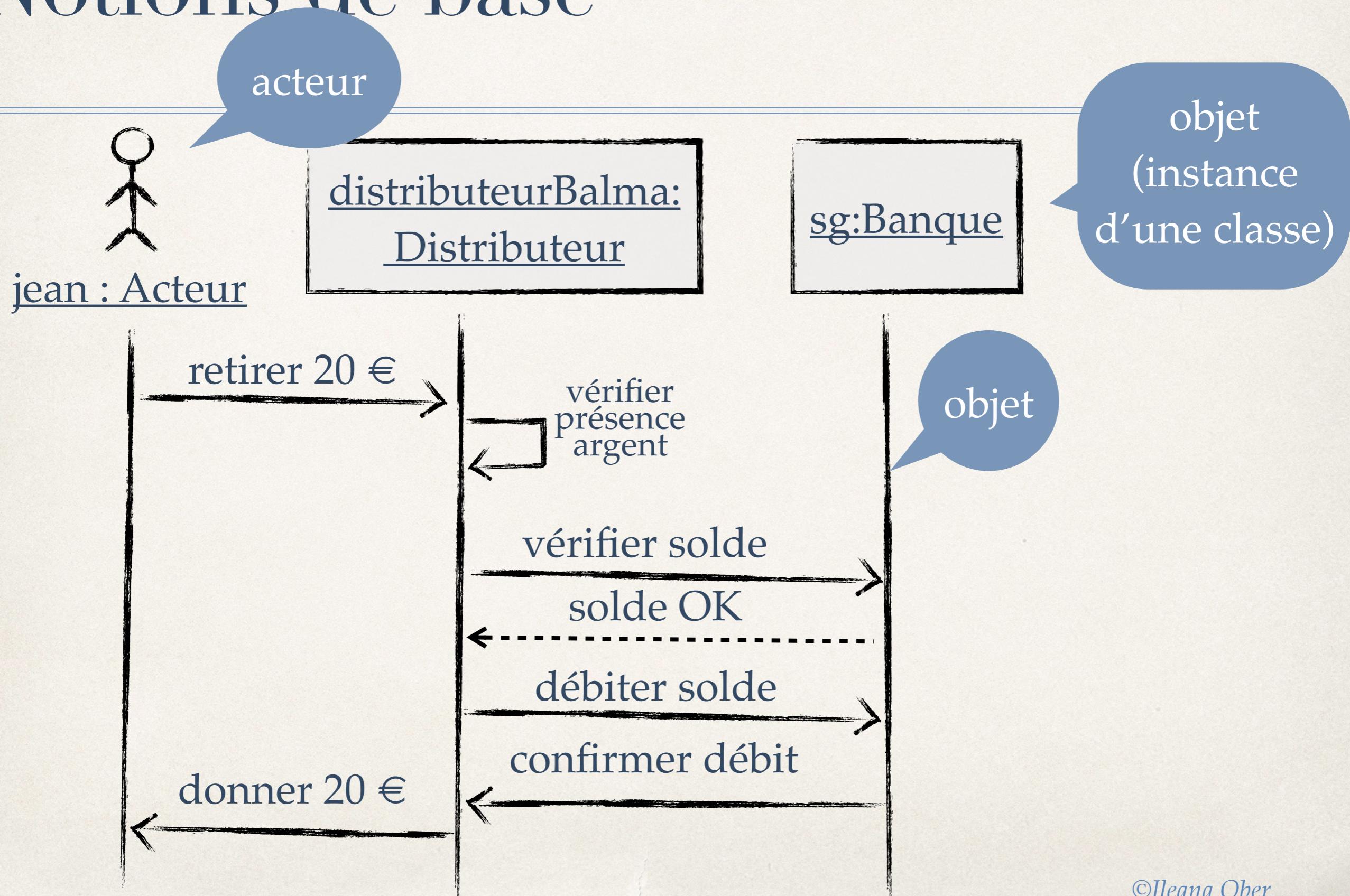
les outils de modélisation proposent souvent de passer automatiquement d'un à l'autre



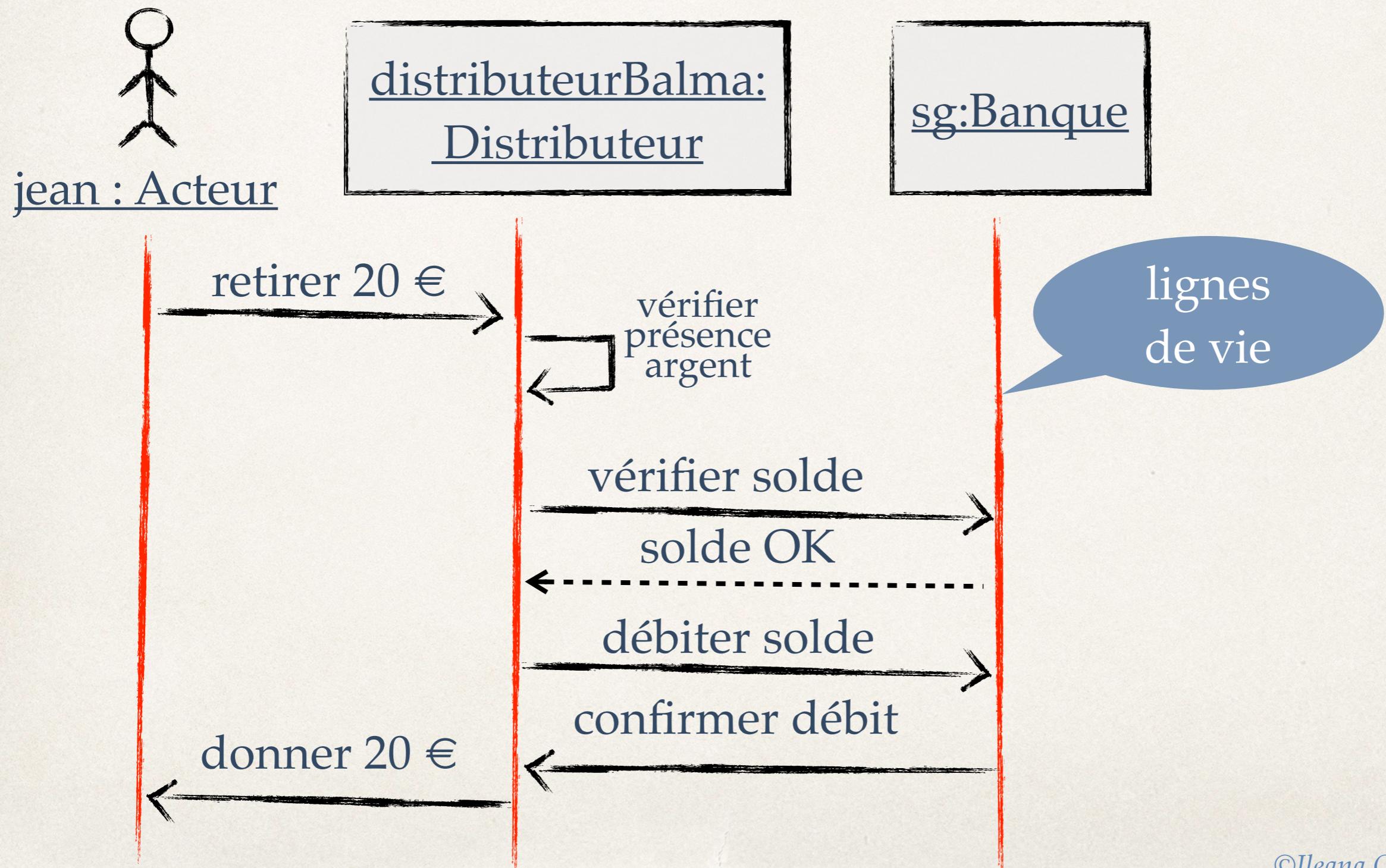
Plan

- ❖ Diagramme de séquence parmi les autres diagrammes UML
- ❖ **Notions de base**
- ❖ Utilisation

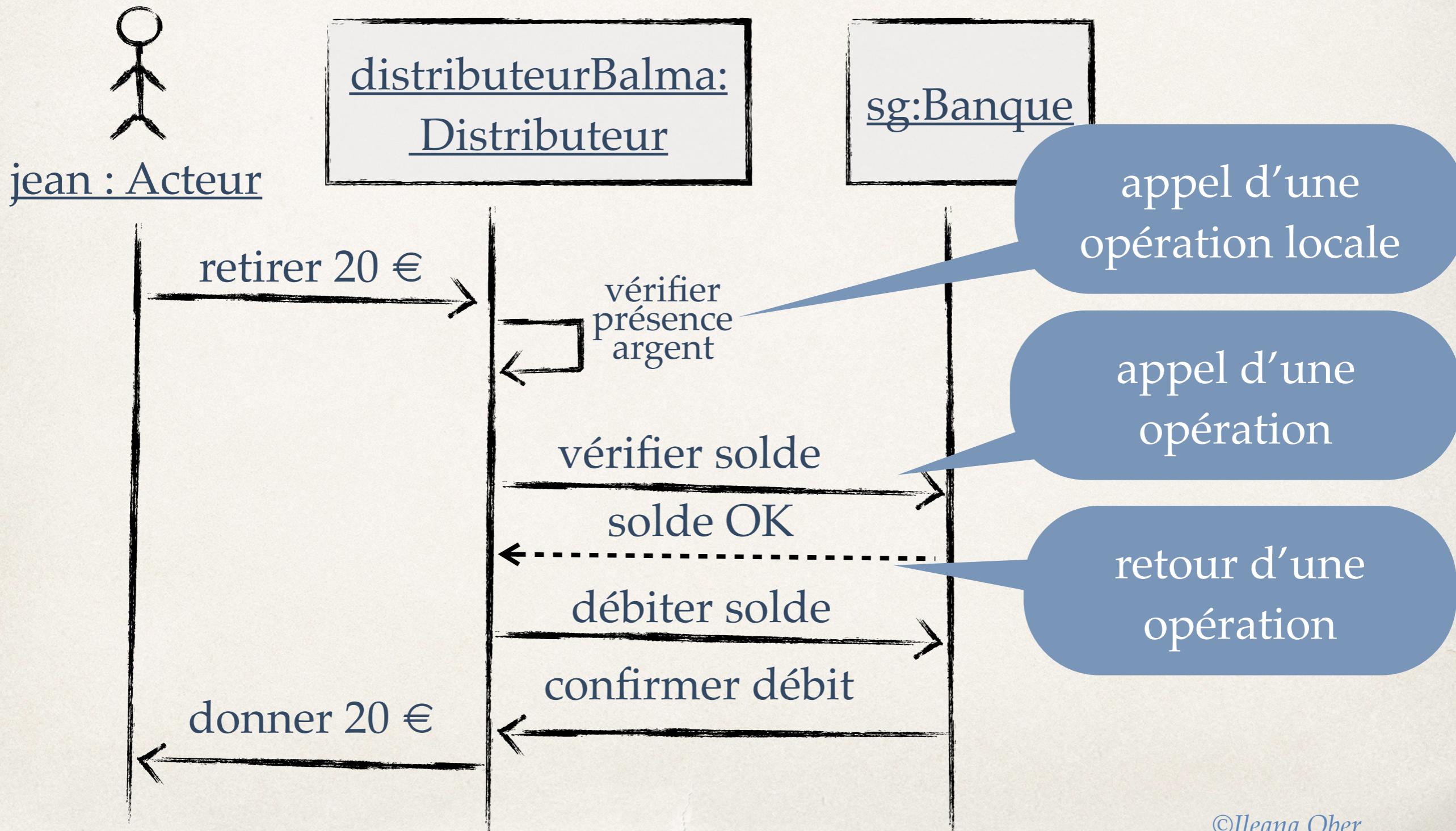
Notions de base



Notions de base



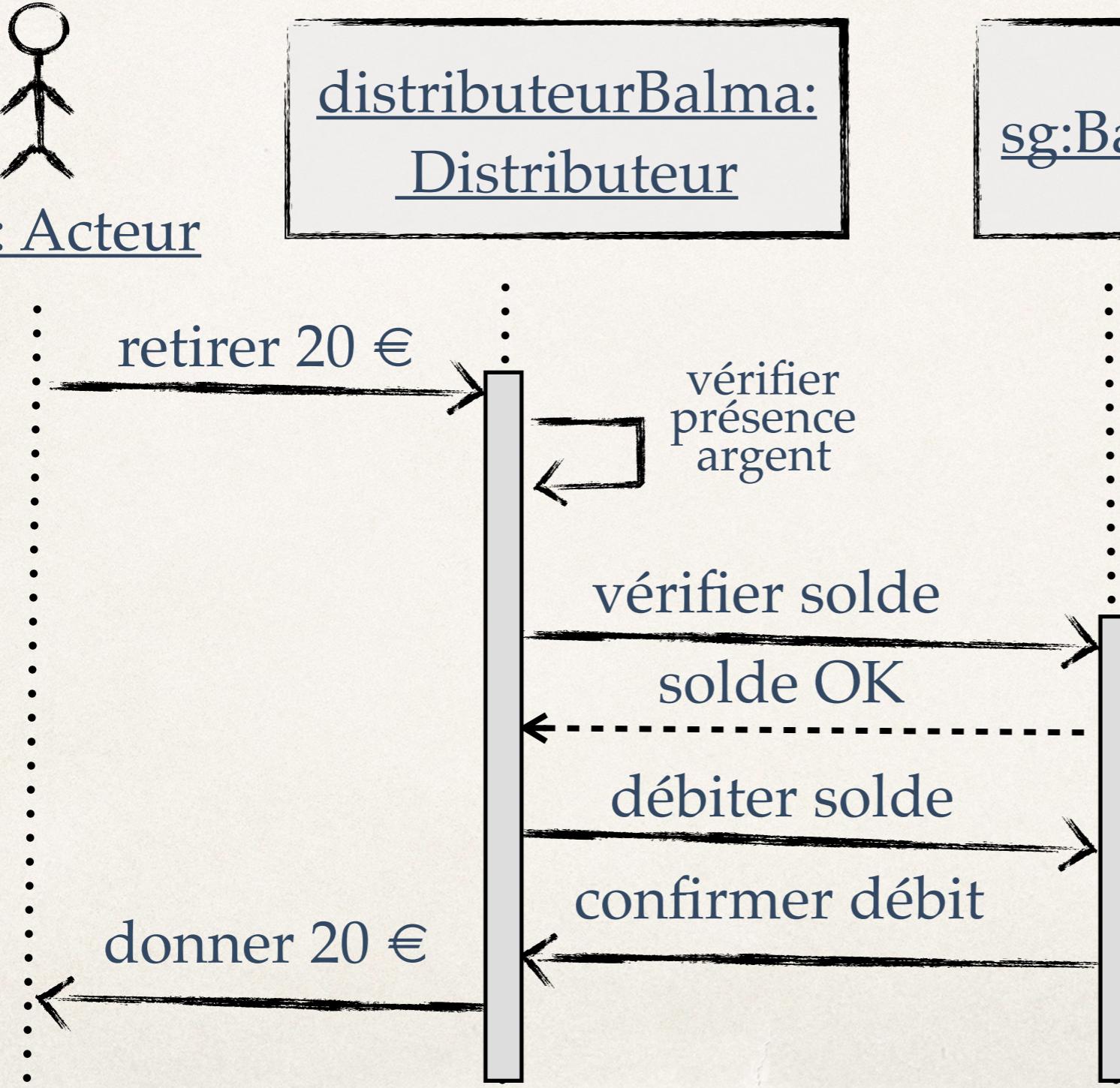
Notions de base



Ligne de vie

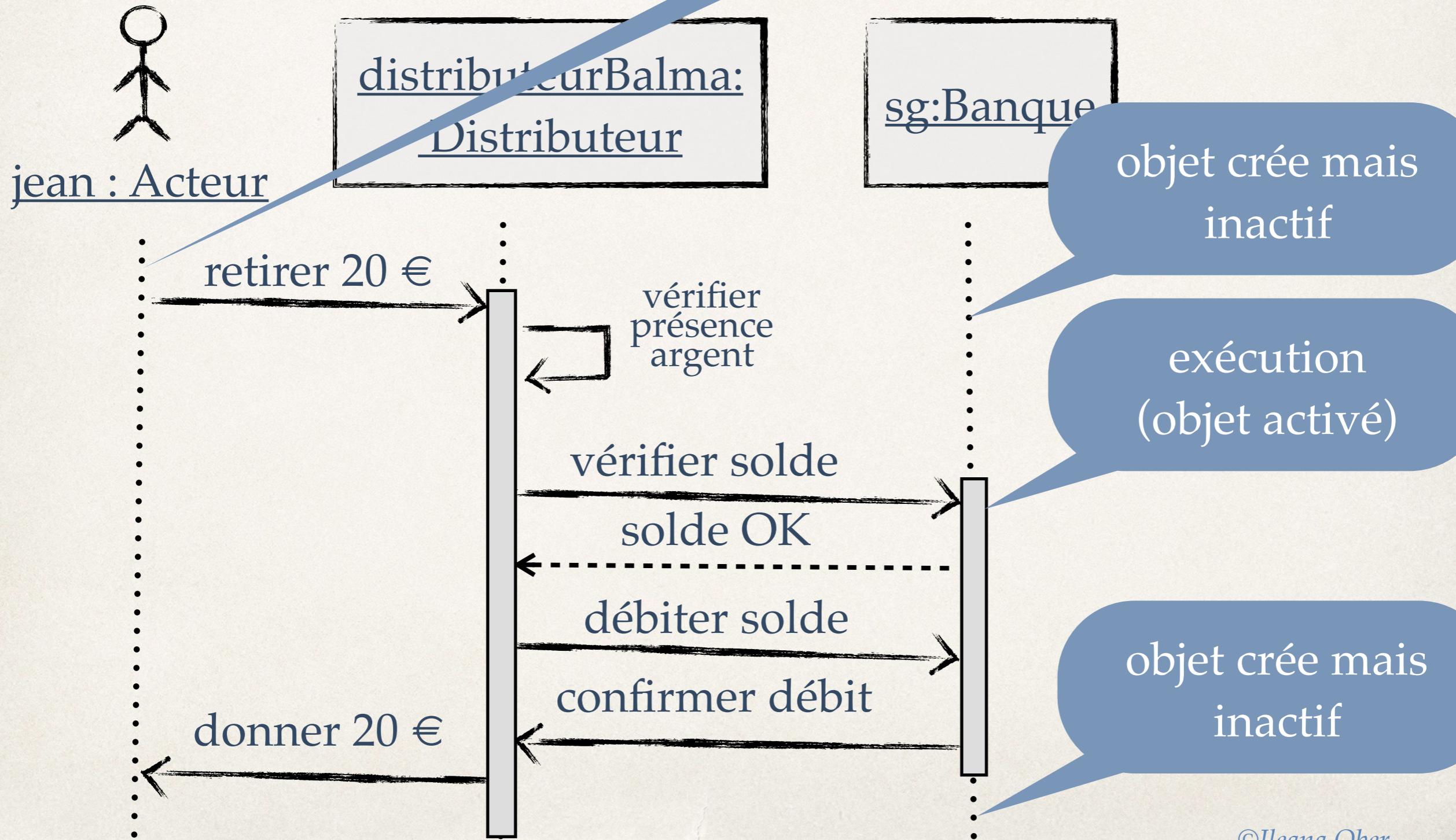
permet de mettre en évidence les périodes d'activation

jean : Acteur



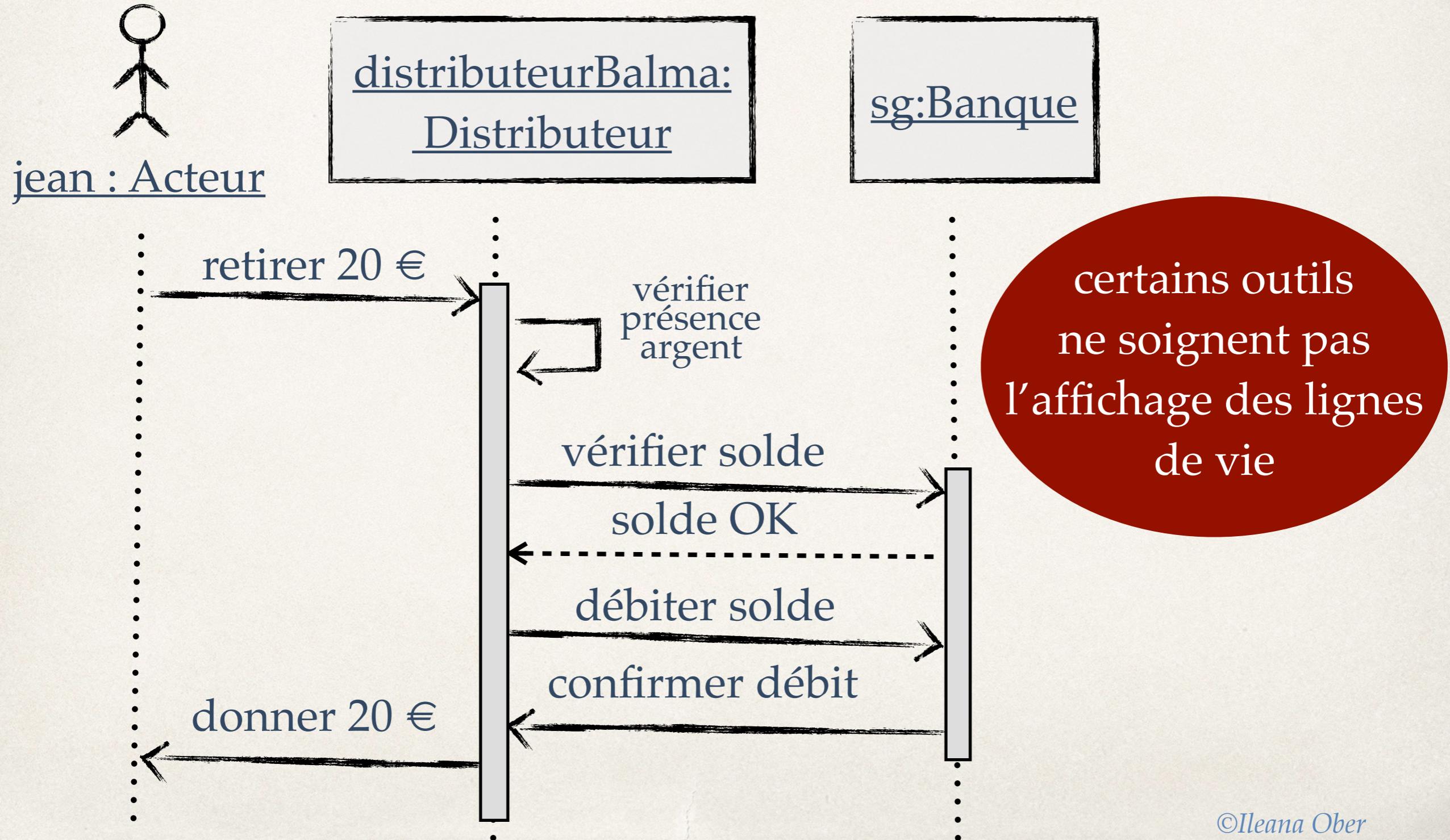
exécution
(objet activé)

Ligne de vie



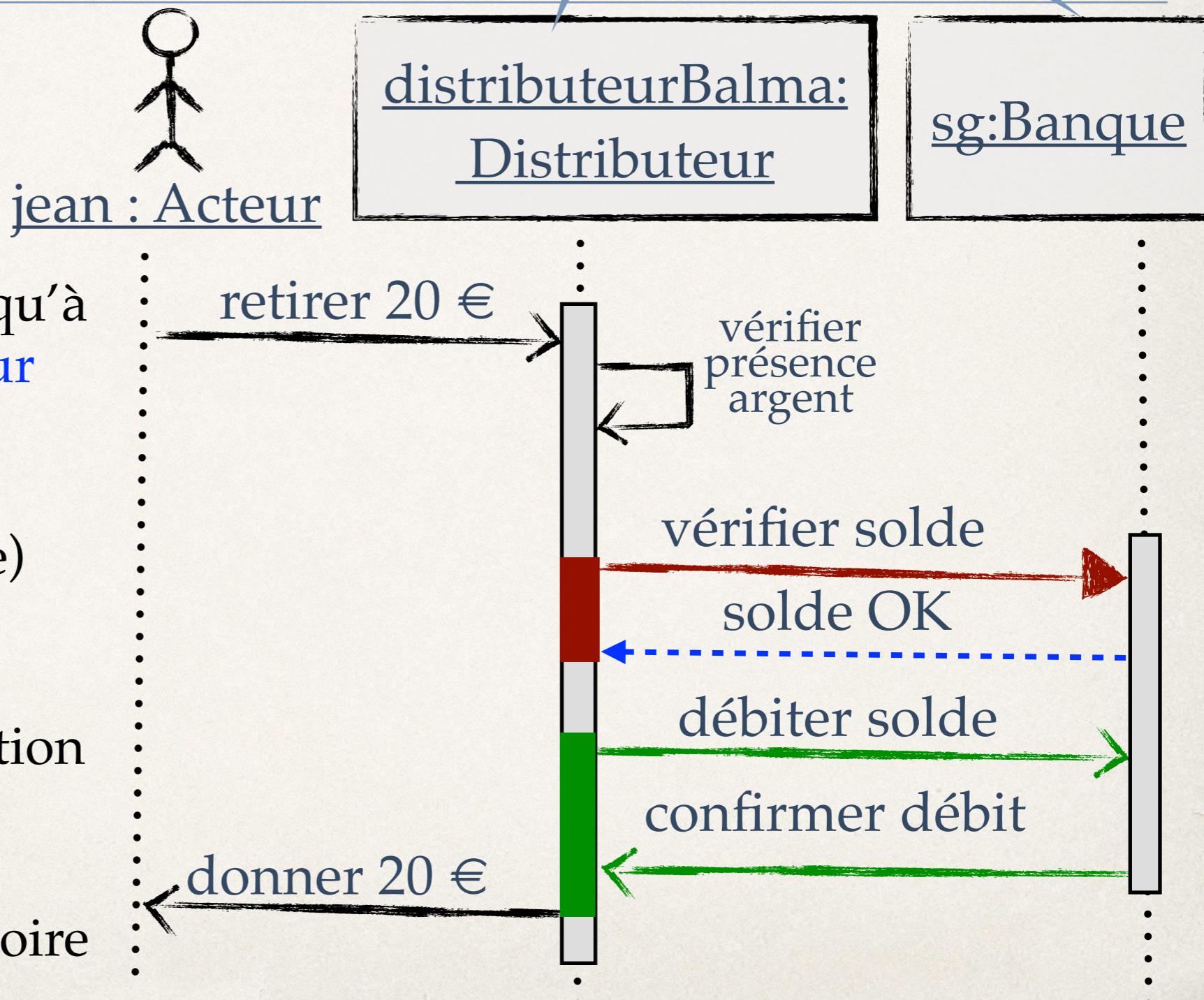


Ligne de vie



Messages

- ❖ **synchrone**
(type appel opération)



- ❖ **asynchrone**
(type envoi de message)

- ❖ l'appelant peut continuer son exécution
- ❖ le retour n'est ni immédiat, ni obligatoire

Messages - notation



- **synchrone**
(type appel)

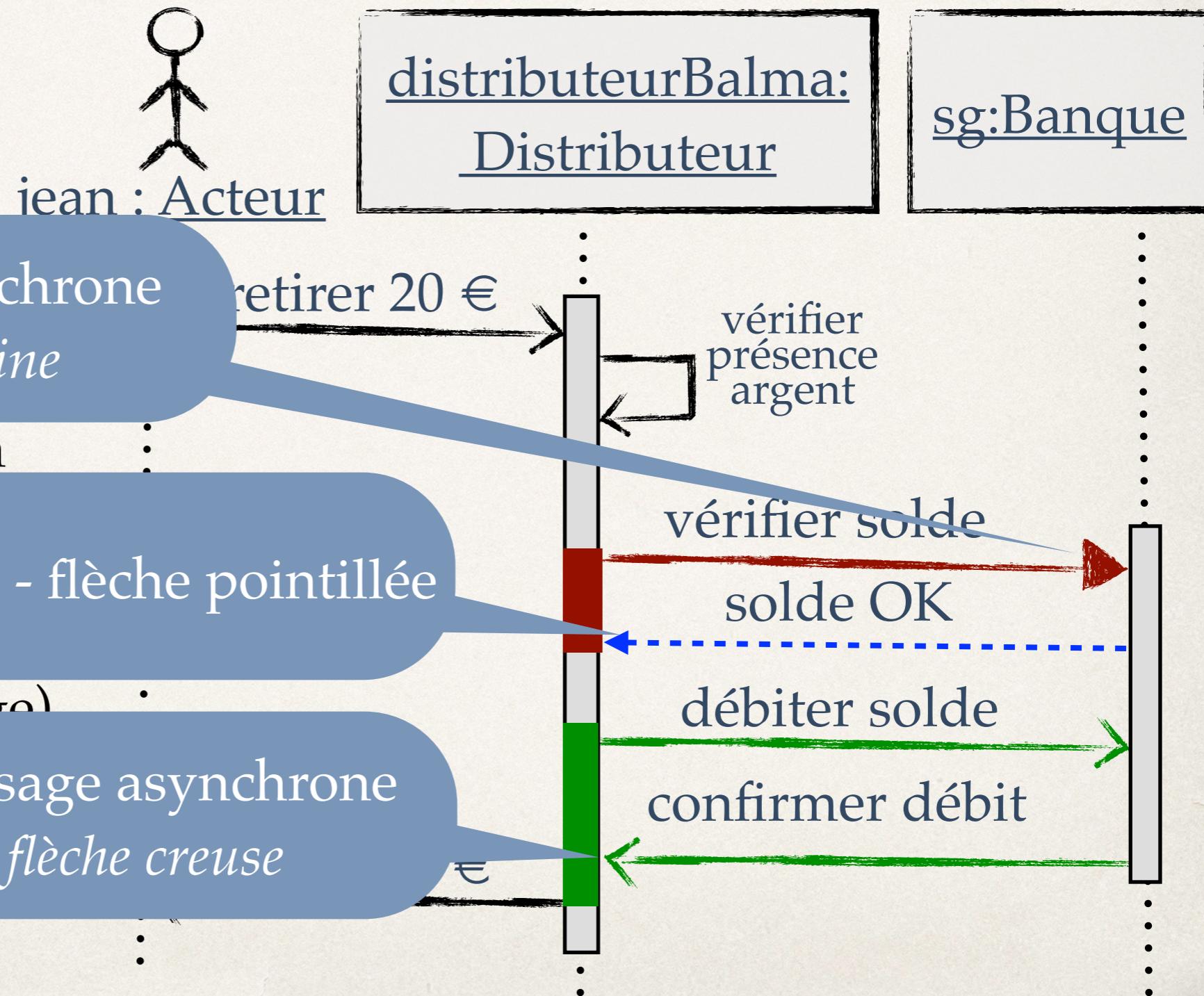
message synchrone
flèche pleine

appelé jusqu'à la réception
du **retour**

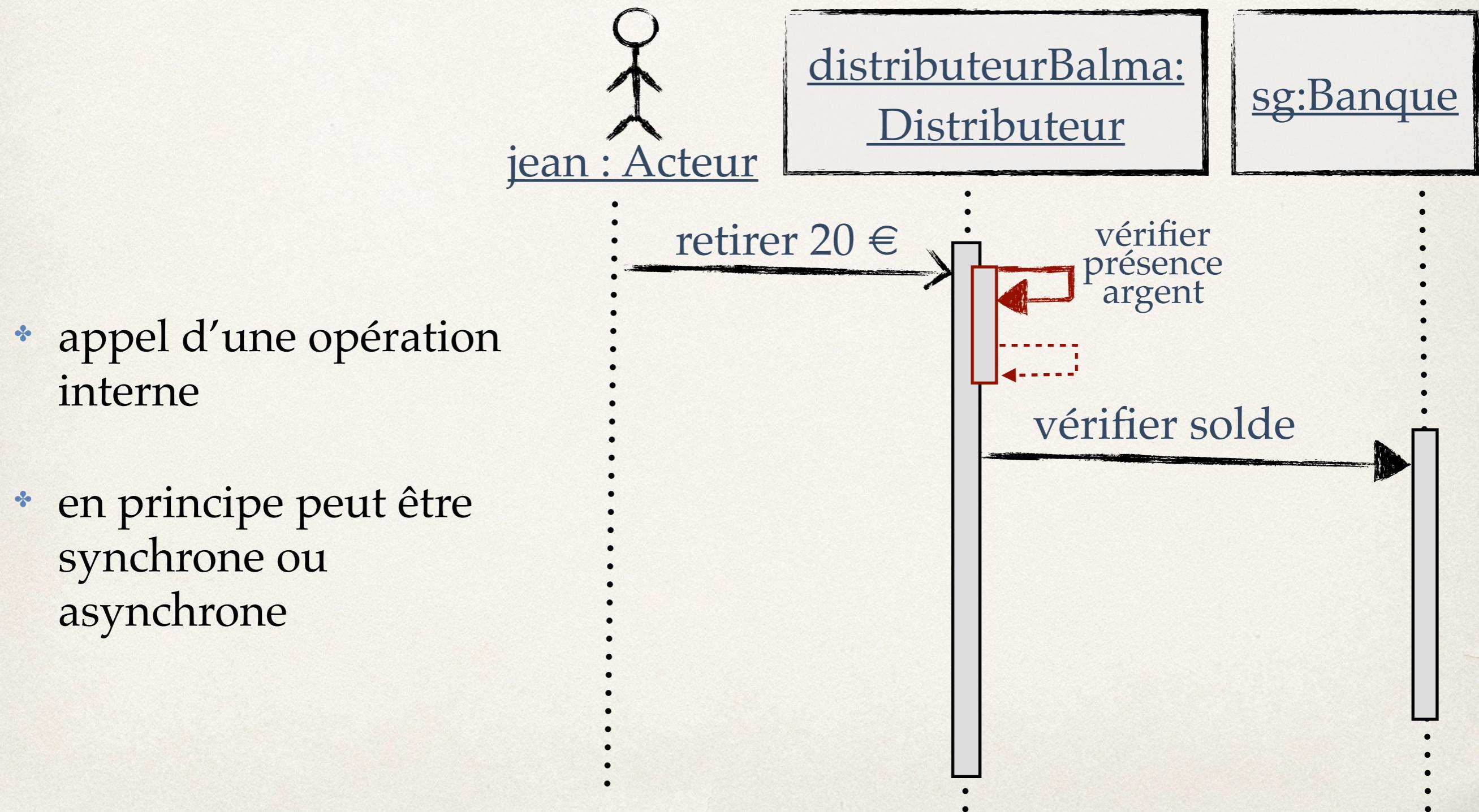
- **asynchrone**
(type envoi de message)

message asynchrone
flèche creuse

l'appelant peut continuer son exécution



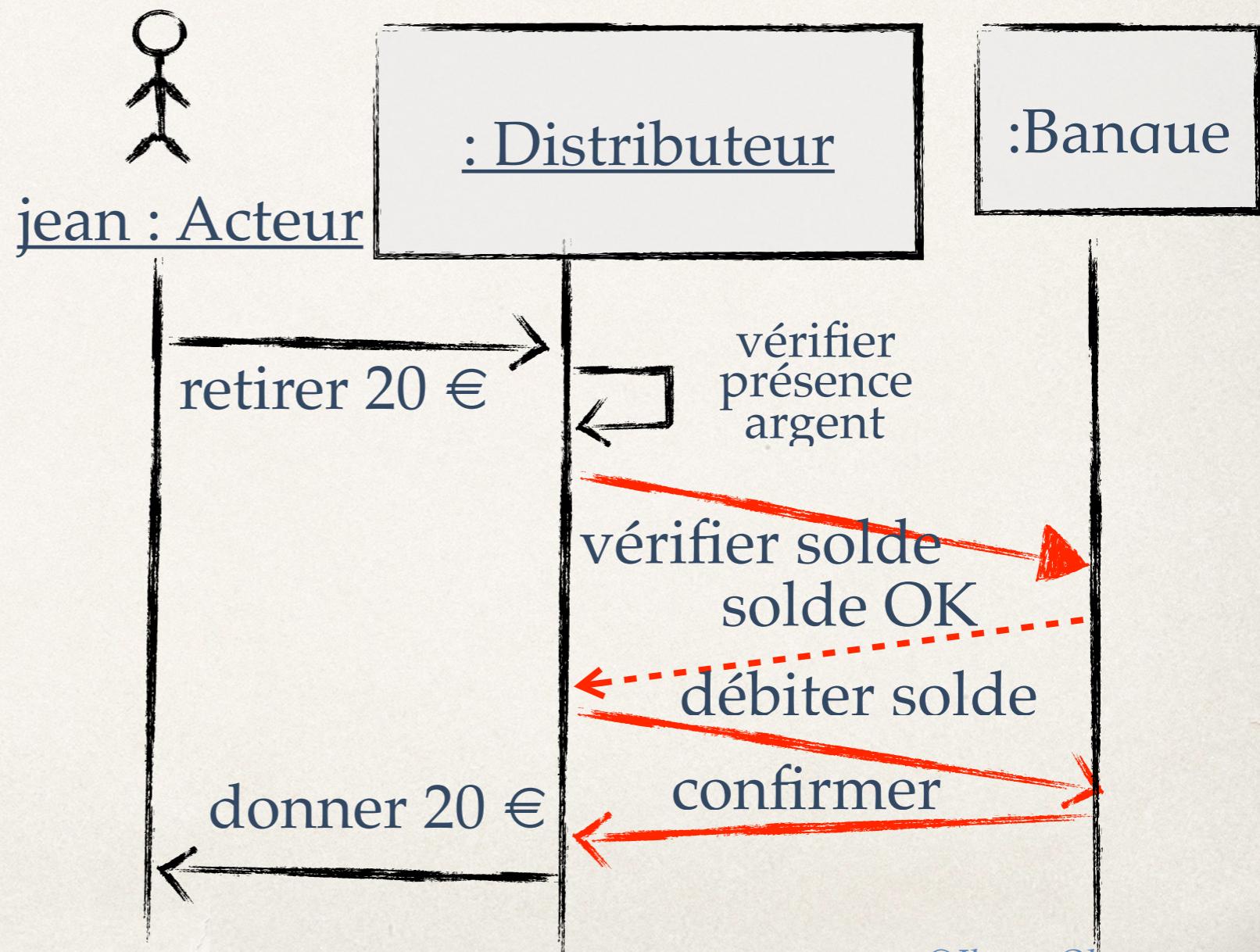
Message réflexif



Le temps dans le diagramme de séquence

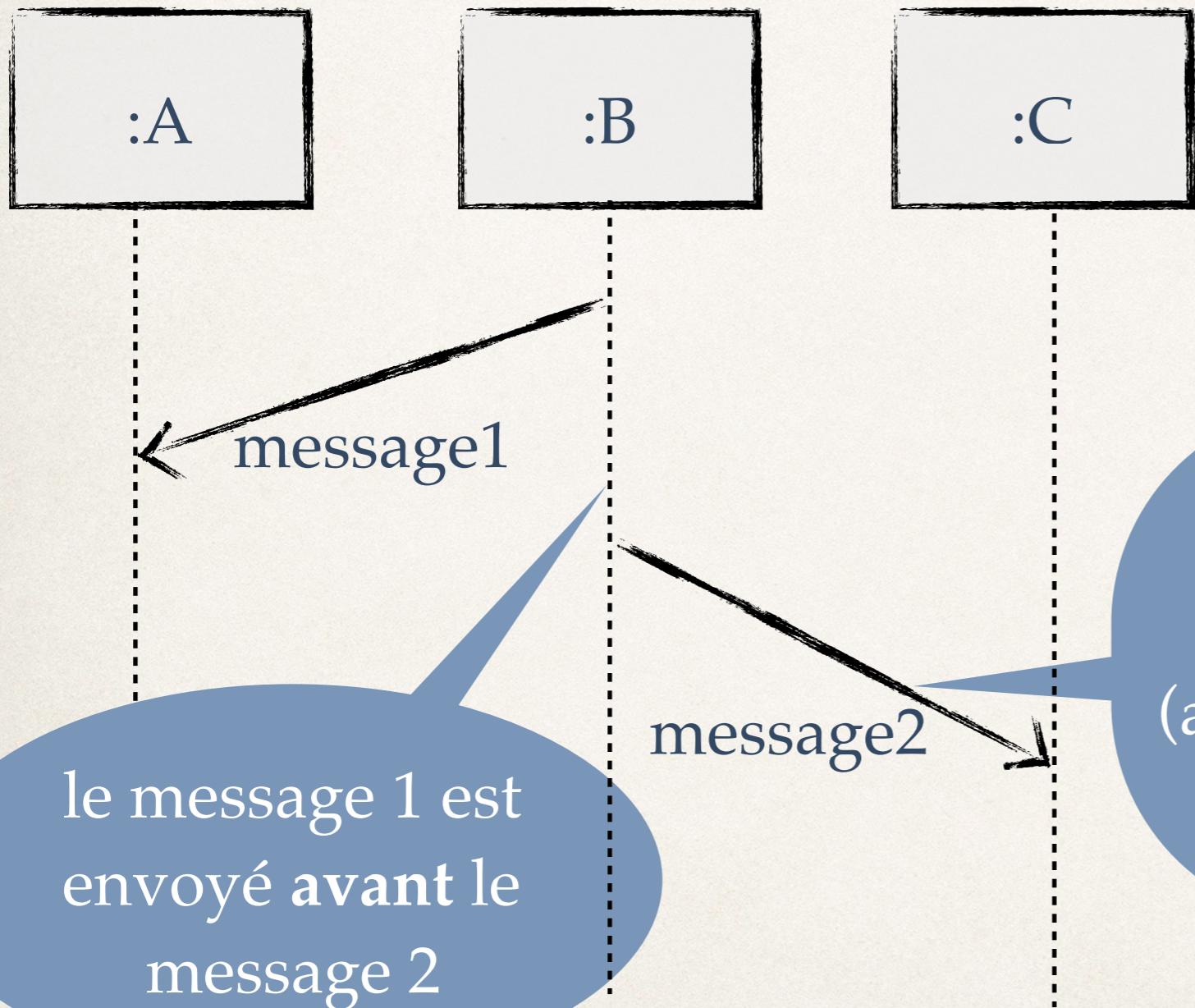


- * Dans un diagramme de séquence le temps s'écoule de haut en bas
- * pour montrer qu'un échange de message prends du temps, la flèche de message peut être **oblique**



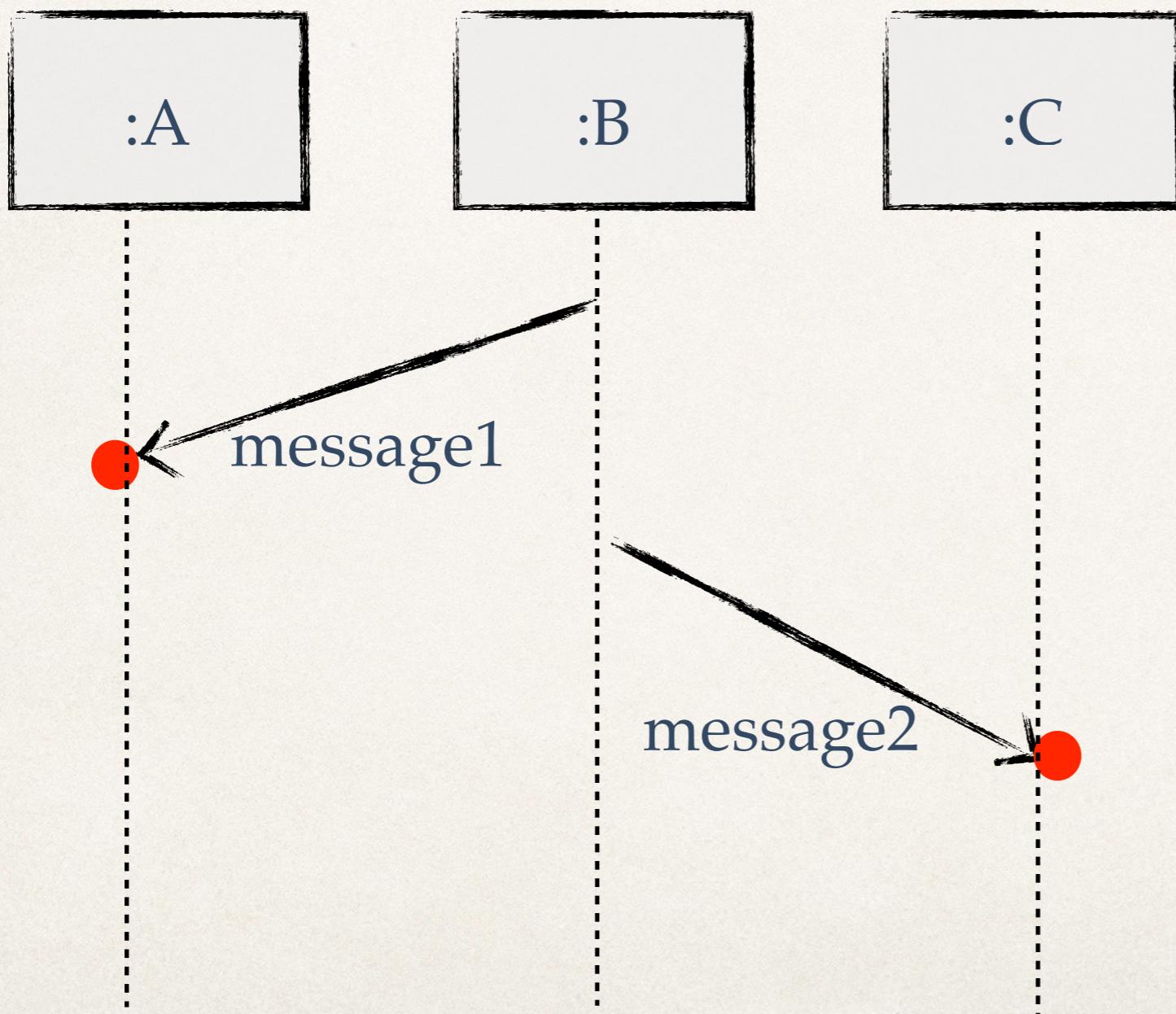


Le temps dans le diagramme de séquence



Exercice

?

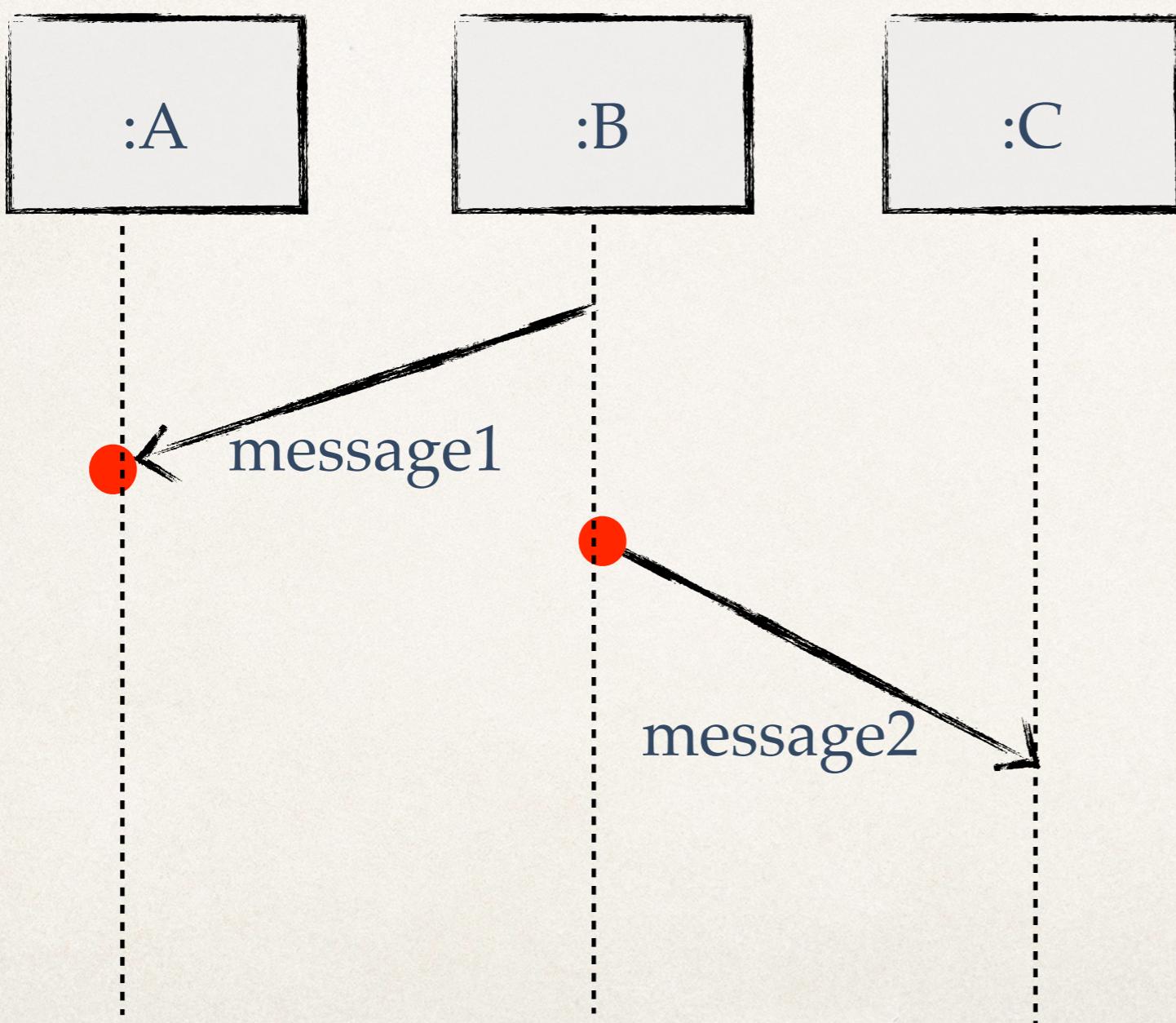


Lequel des messages 1
et 2 arrive en premier?

On ne sait pas !!!

Exercice

?

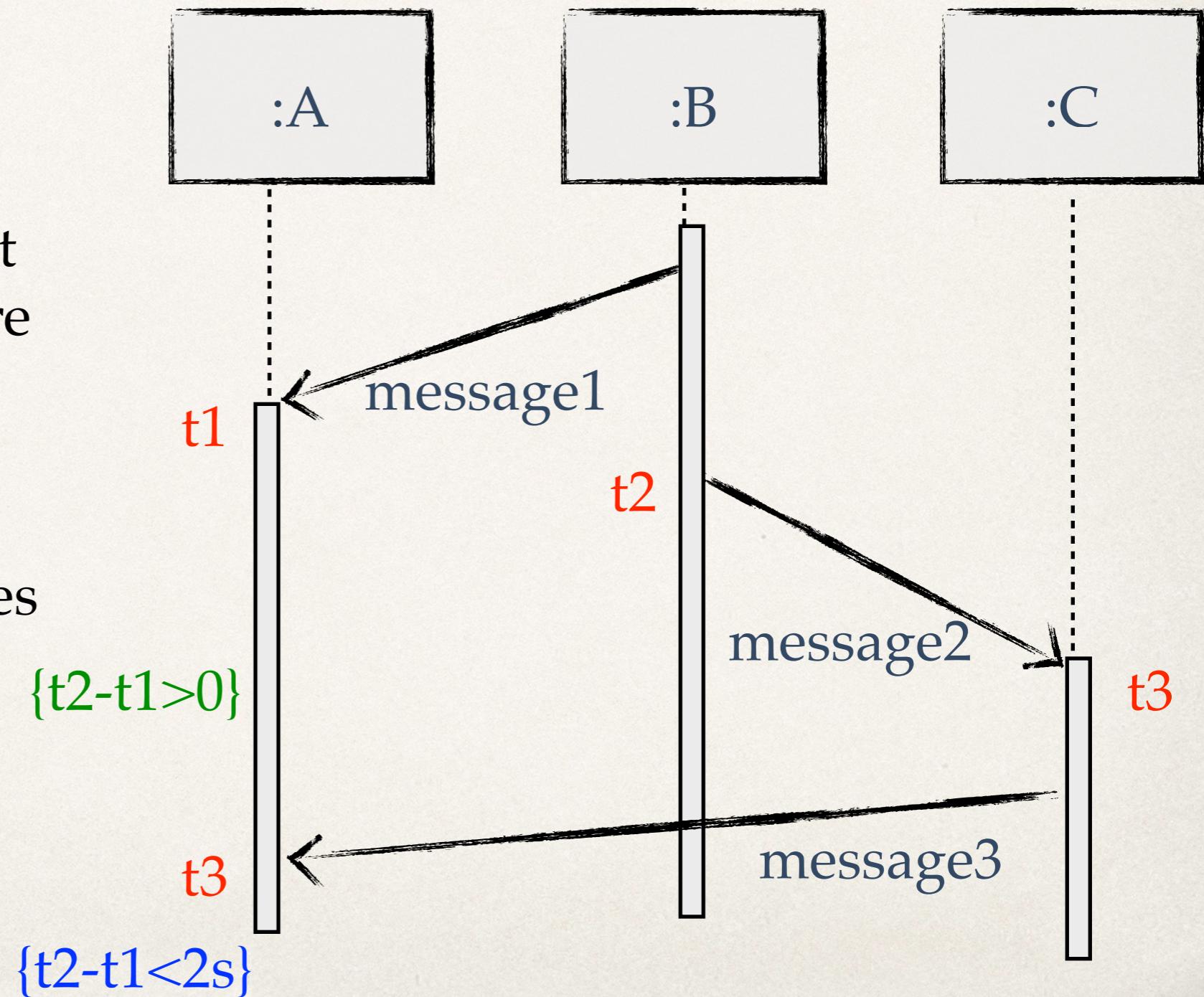


Est-ce que le message 1 est arrivé lorsque le message 2 part?

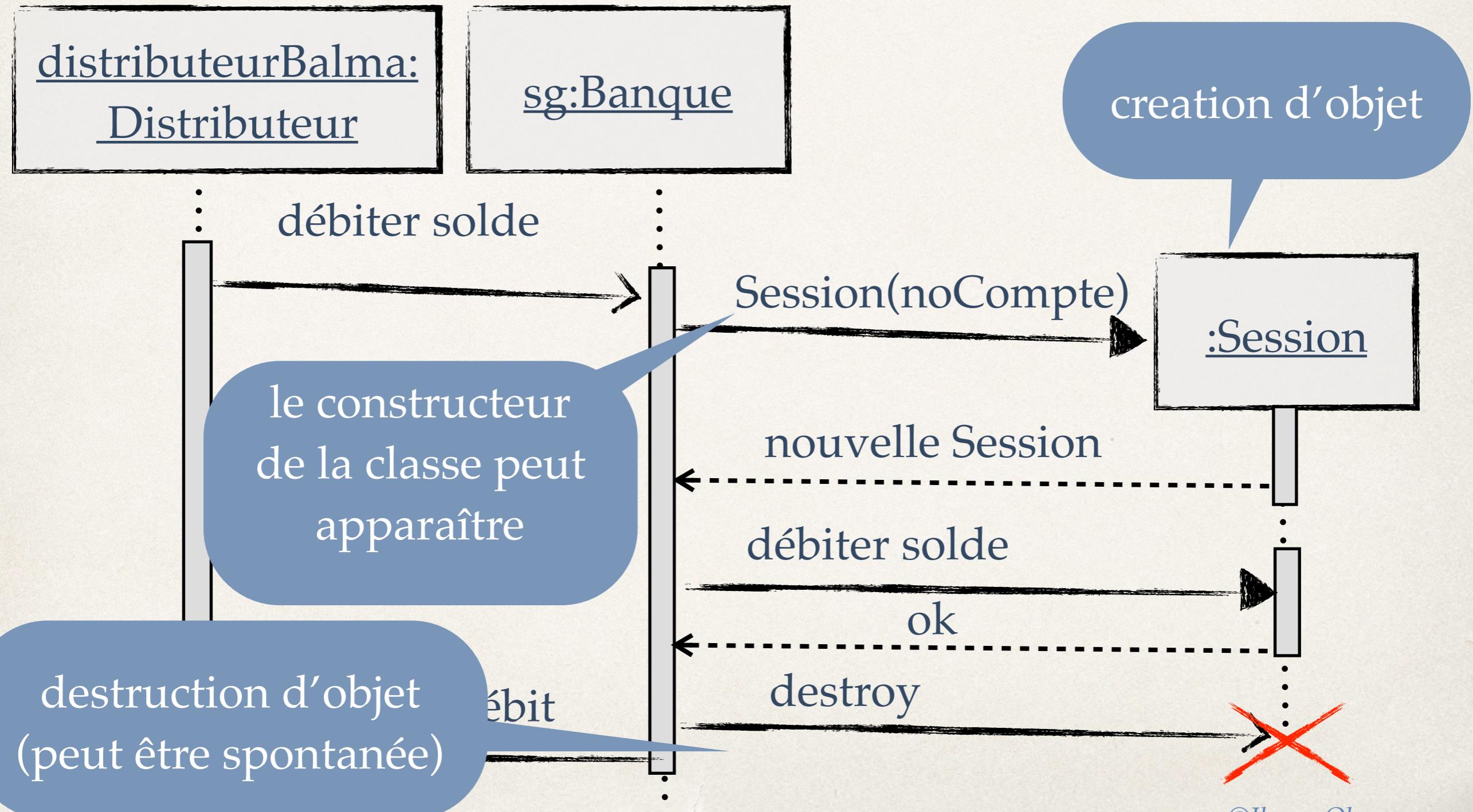
On ne sait pas !!!

Contraintes temporelles

- Si on a besoin de préciser qu'un message arrive avant un autre il faut le dire explicitement
- On rajoute des étiquettes temporelles
- On peut mettre des contraintes **relatives** ou **absolues**



Création / destruction objet



Alternative

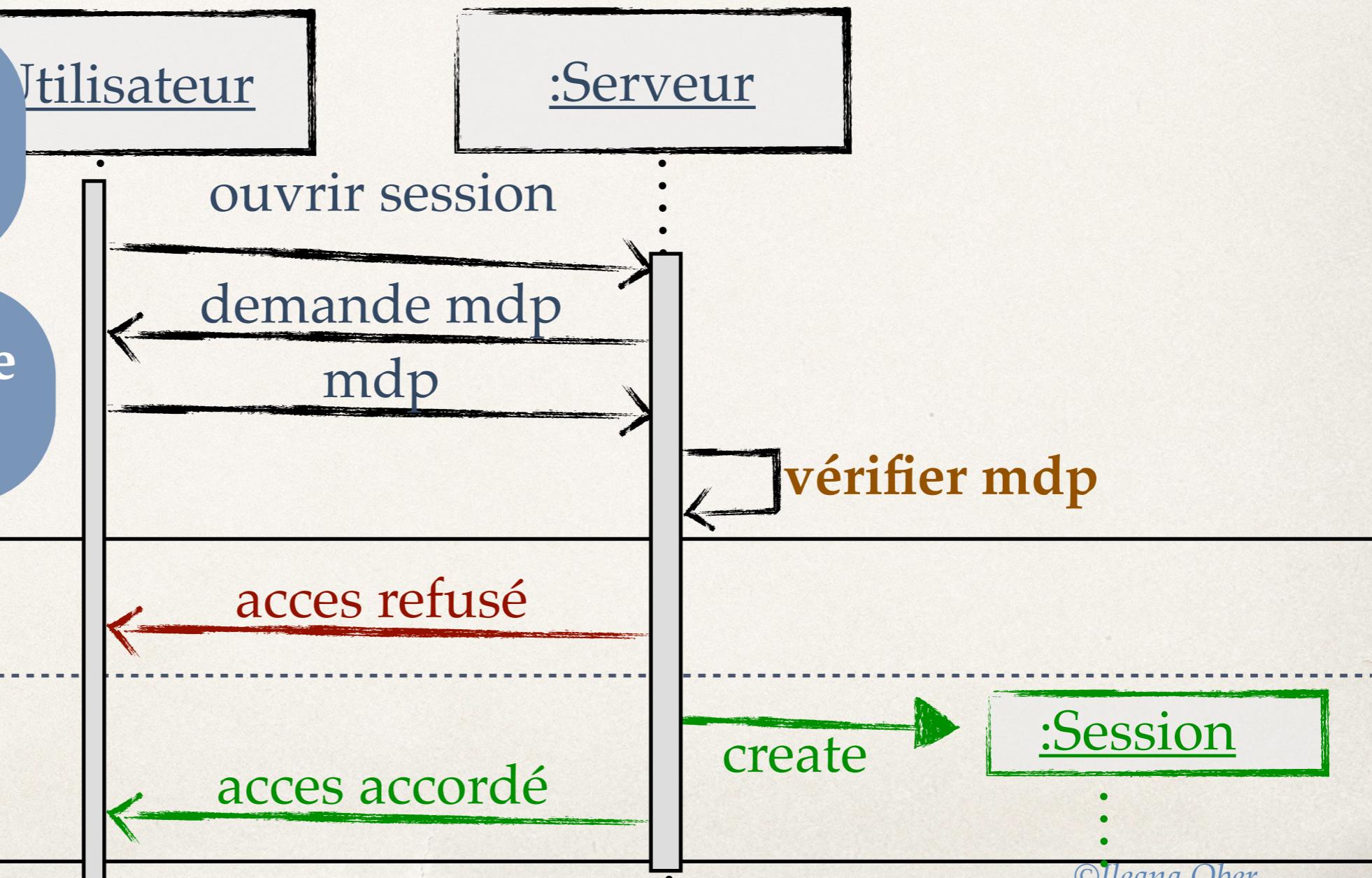
- * principe: condition à l'envoi d'un message
- * deux diagrammes alternatives

on peut avoir plus de 2 alternatives

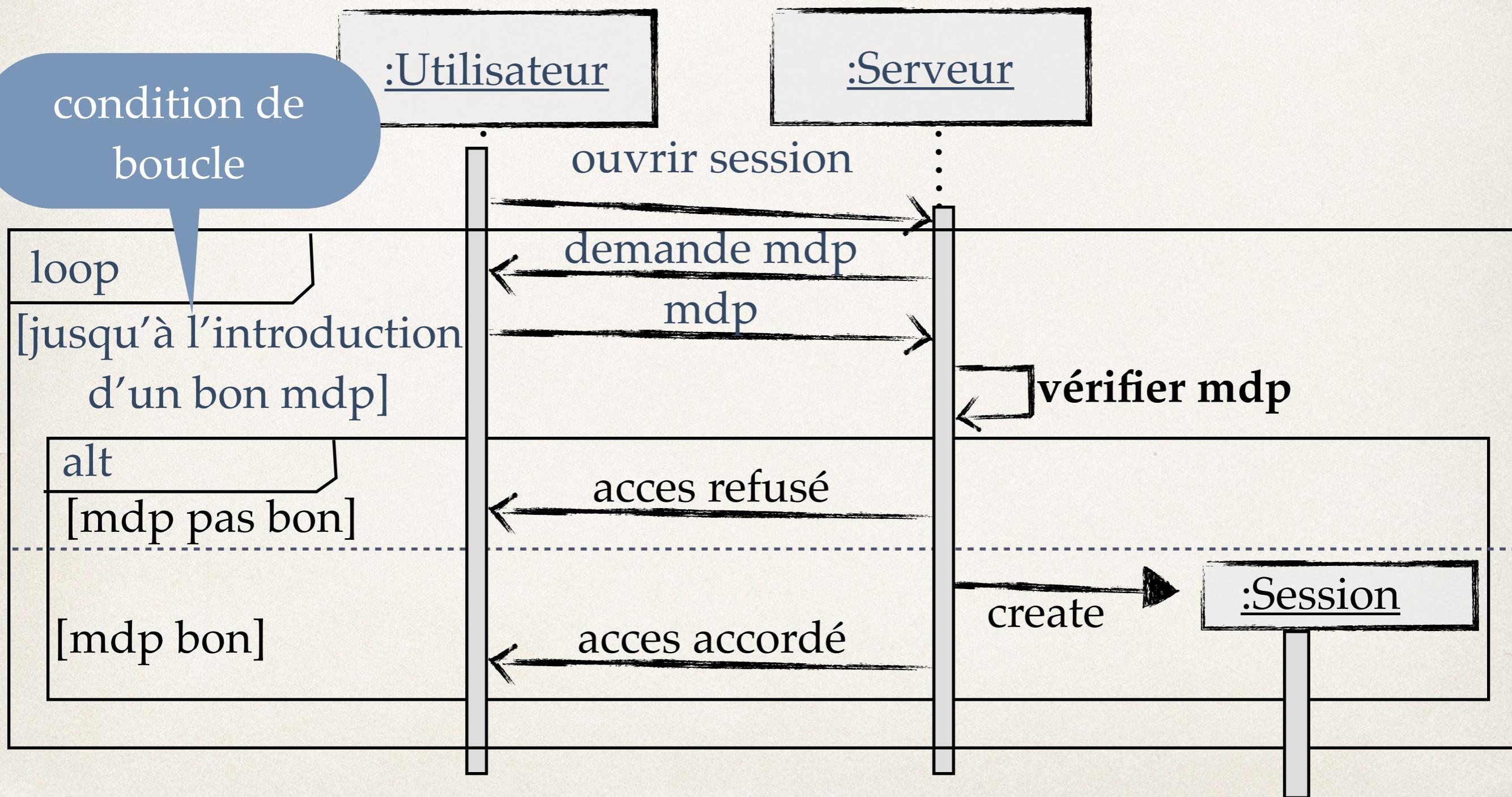
on peut mettre else comme garde

alt
[mdp pas bon]

[mdp bon]



Boucle



Structuration

- ⊕ En général un diagramme de séquence décrit un scénario
- ⊕ Pour éviter les gros diagrammes qui nuisent la lisibilité ⇒ structurer
- ⊕ Mécanismes de structuration
 - ⊕ **alt** fragment alternatif - condition dans les [gardes]
 - ⊕ **loop** fragment à répéter jusqu'à ce que la condition [garde] est vraie
 - ⊕ **opt** fragment optionnel - à exécuter si [garde] est vraie
 - ⊕ **ref** passage à un autre diagramme de séquence

Plan

- ❖ Diagramme de séquence parmi les autres diagrammes UML
- ❖ Notions de base
- ❖ **Utilisation**

Utilisation

- ❖ Description des **scénarios** significatifs
(ex. réalisation d'un cas d'utilisation)
scénario typique, scénario d'erreur, scénario réputé complexe
- ❖ Utilisation en ultérieure en test (automatique ou manuel)
- ❖ Description de **protocoles**
 - ❖ IHM - échanges d'informations utilisateur - application
 - ❖ systèmes communicants - échanges de messages
- ❖ Description des échanges entre objets pour la réalisation d'une tâche

Réalisation d'un cas d'utilisation

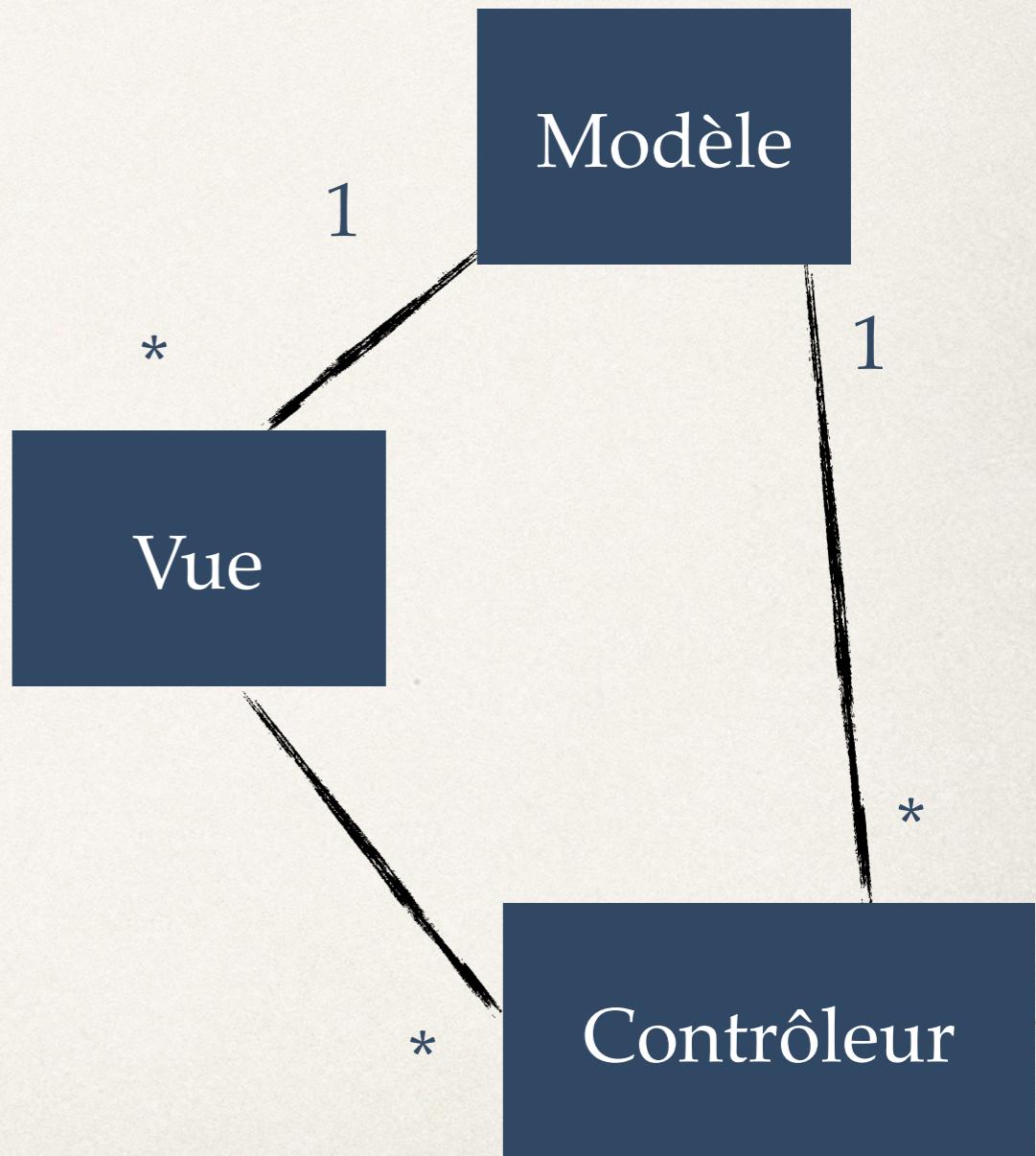
- ✿ Objets:
 - ✿ Acteurs (extraits du cas d'utilisation)
 - ✿ *Dans le cas des DS système*
le diagramme comporte uniquement 2 *objets* (acteur + Système)
 - ✿ *Dans le cas des DS détaillé*
le DS contient comme objets des *instances des classes d'analyse*
- ✿ Messages:
 - ✿ échanges entre les objets identifiés en haut, tels que spécifiés par le cas d'utilisation
 - ✿ le niveau de détail dépend de si DS système ou détaillé

Identification des classes d'analyse à partir d'un cas d'utilisation

- ❖ Au moins **une classe «boundary» par couple (acteur, cas d'utilisation)**
- ❖ Au moins **une classe «control» par cas d'utilisation**
- ❖ Les classes «entity» sont issues, en général, du **modèle des classes du domaine**

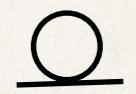
Modèle-Vue-Contrôleur

- ❖ Paradigme (patron, architecture) permettant la **séparation** entre les *données* du programme (*modèle*) et la partie graphique affichant les données
- ❖ *La vue* correspond à la présentation (IHM)
- ❖ *Le contrôleur* assure la partie logique
- ❖ S'utilise aussi en modélisation



Classes d'analyse

- * <<entity>> (correspond à une donnée élémentaire)
objets métier manipulés
en général persistants



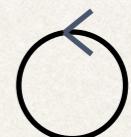
NomEntityClass

- * <<boundary>> (correspond à la vue) - moyens
d'interaction avec le système
écrans, formulaires de saisie (mais peut aller au delà
des IHM)
durée de vie du cas d'utilisation



NomBoundaryClass

- * <<control>> (correspond au contrôle) - logique
applicative

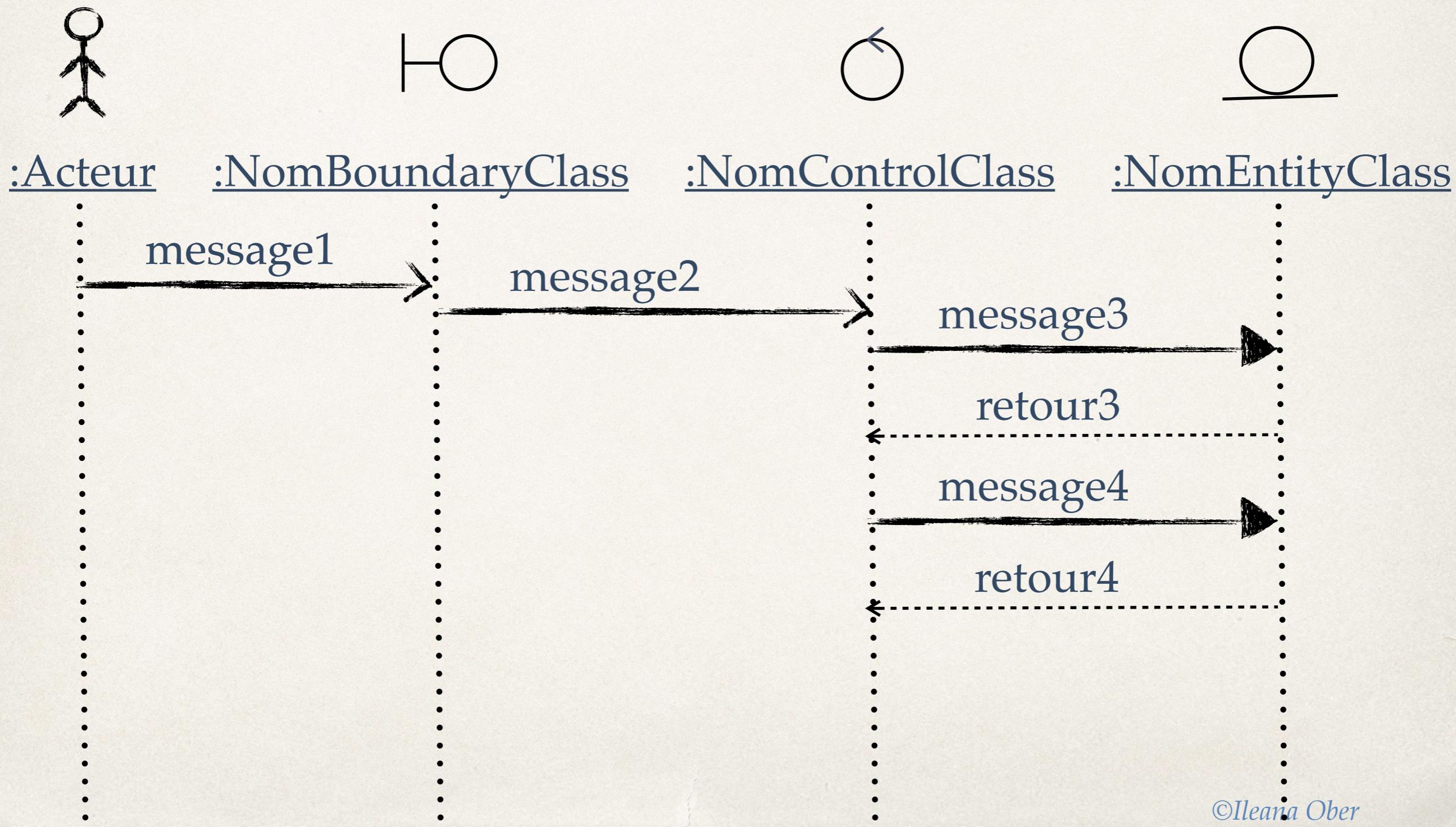


NomControlClass

Règles d'interaction

	Acteur	Boundary	Control	Entity
Acteur	Non	Oui	non	non
Boundary	Oui	rare	oui	non
Control	Non	oui	oui	oui
Entity	Non	non	non	oui

Diagramme de séquence détaillé



DS système / DS détaillé



- ❖ En fonction de l'étape à laquelle le diagramme est élaboré on a
 - ❖ **DS système** spécifie les interactions entre le système (vu comme un tout) et un (des) acteur(s) externe(s)
 - ❖ **DS détaillé** spécifie aussi les interactions entre objets internes au système
- ❖ Les deux détaillent un cas d'utilisation, mais à un **grain** différent
- ❖ La **syntaxe** du DS ne dépend pas de son type

Conclusion

- ❖ Diagramme utile en spécification et conception objet préliminaire
- ❖ Bonne vision des interactions
- ❖ Il faut adapter le niveau de détail