

- 3.3 -

Modélisation interactions (2)

Exemples dans le cours

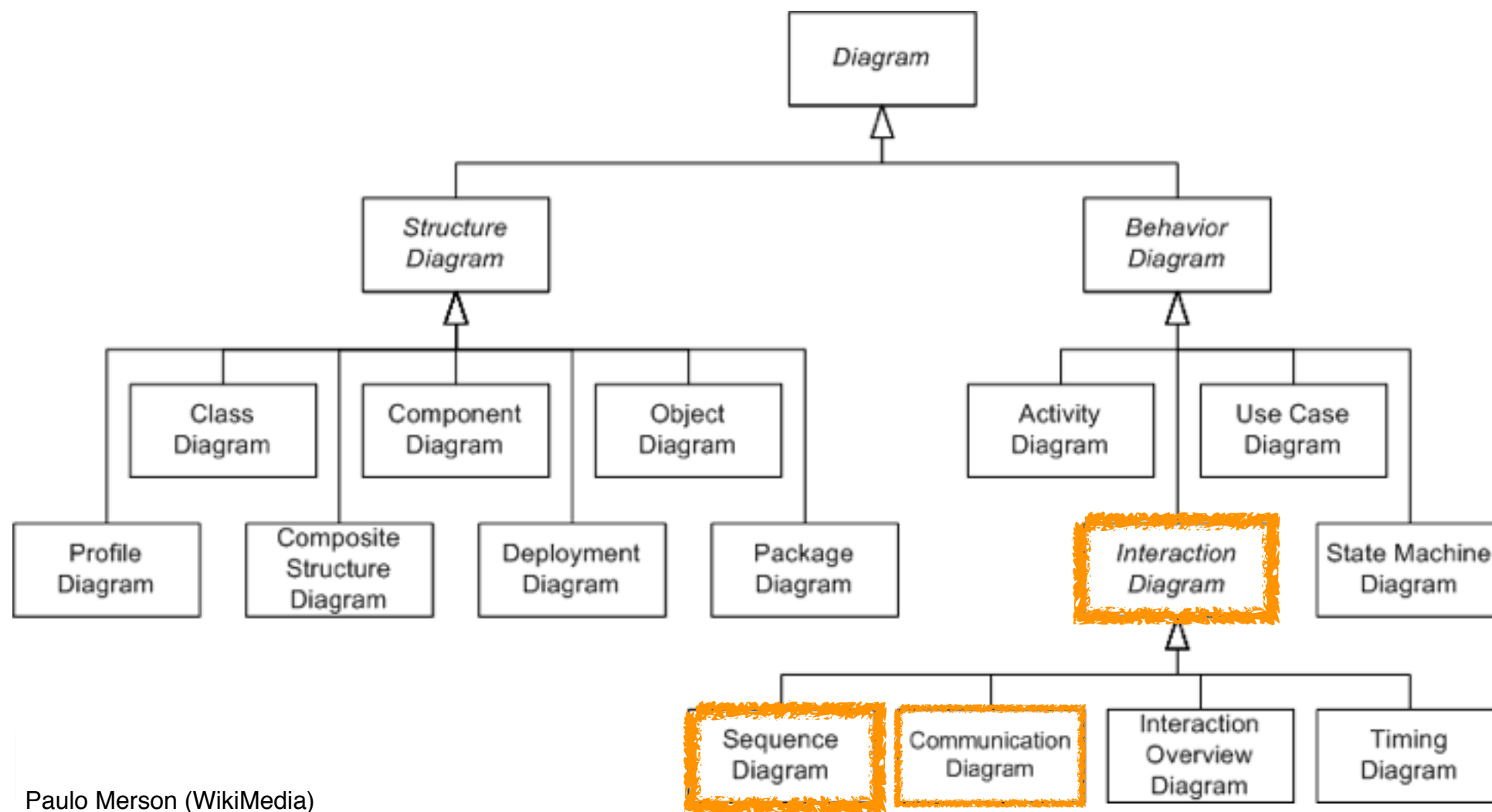
Une partie inspirés de

- Software Engineering, 9th ed. - Ian Sommerville
- Object-Oriented Software Engineering: Practical Software Development using UML and Java - Timothy C. Lethbridge, Robert Laganière
- UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd ed. - Martin Fowler
- UML 2 - de l'apprentissage à la pratique - Laurent Audibert, Ellipses, 2009

Perspectives

- *externe*
 - ▶ modéliser le contexte (l'environnement)
- *interaction*
 - ▶ modéliser les interactions avec l'environnement et entre les composants
- *structurelle*
 - ▶ modéliser l'architecture et les données
- *comportementale*
 - ▶ modéliser la dynamique

Diagrammes d'interaction



Paulo Merson (WikiMedia)

Diagrammes d'interaction

- aident à visualiser comment le système exécute ses tâches
- souvent construit à partir d'un diagramme de classes ou d'un use-case

Interactions

Un *diagramme d'interactions* montrent comment un ensemble d'objets et d'acteurs *communiquent* ensemble afin :

- ▶ de réaliser un use-case
- ▶ de réaliser une certaine fonctionnalité

L'ensemble des différentes étapes à accomplir s'appelle une *interaction*.

Messages

Un diagramme d'interaction montre différents types de communications entre objets, acteurs et sous-systèmes:

- ▶ appel à des méthodes,
- ▶ information envoyés sur un réseau,
- ▶ ...

appelées **messages**.

Éléments d'un diagramme d'interaction

Instances de classes

- Représentées par des rectangles

Acteurs

- Représentés par des stickmen

Messages

- Représentés par des flèches horizontales

Types de diagrammes

- Diagramme de séquence
 - ➔ chronologie des messages
- Diagramme de communication
 - ➔ organisation structurelle des objets

Diagramme de séquence

- Documentation des cas d'utilisation
- Représentation précise des interactions entre objets et répartition du flot de contrôle

Diagramme de séquence

Séquence temporelle d'*échanges de messages* entre l'*acteur* et les *objets*

- Les objets sont disposés horizontalement
- L'acteur qui initie l'interaction se trouve généralement à l'extrême gauche
- La dimension verticale représente le temps

Diagramme de séquence

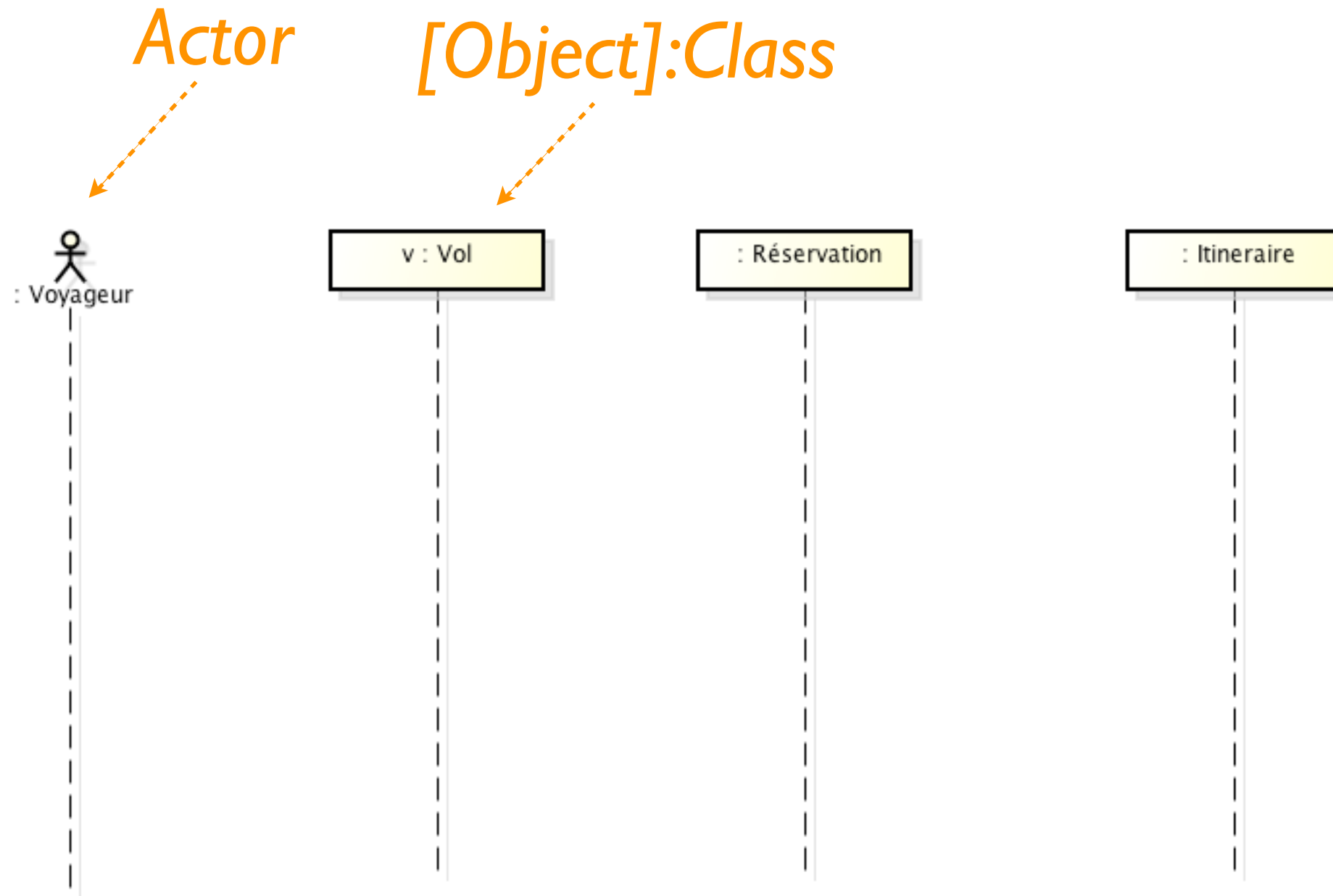
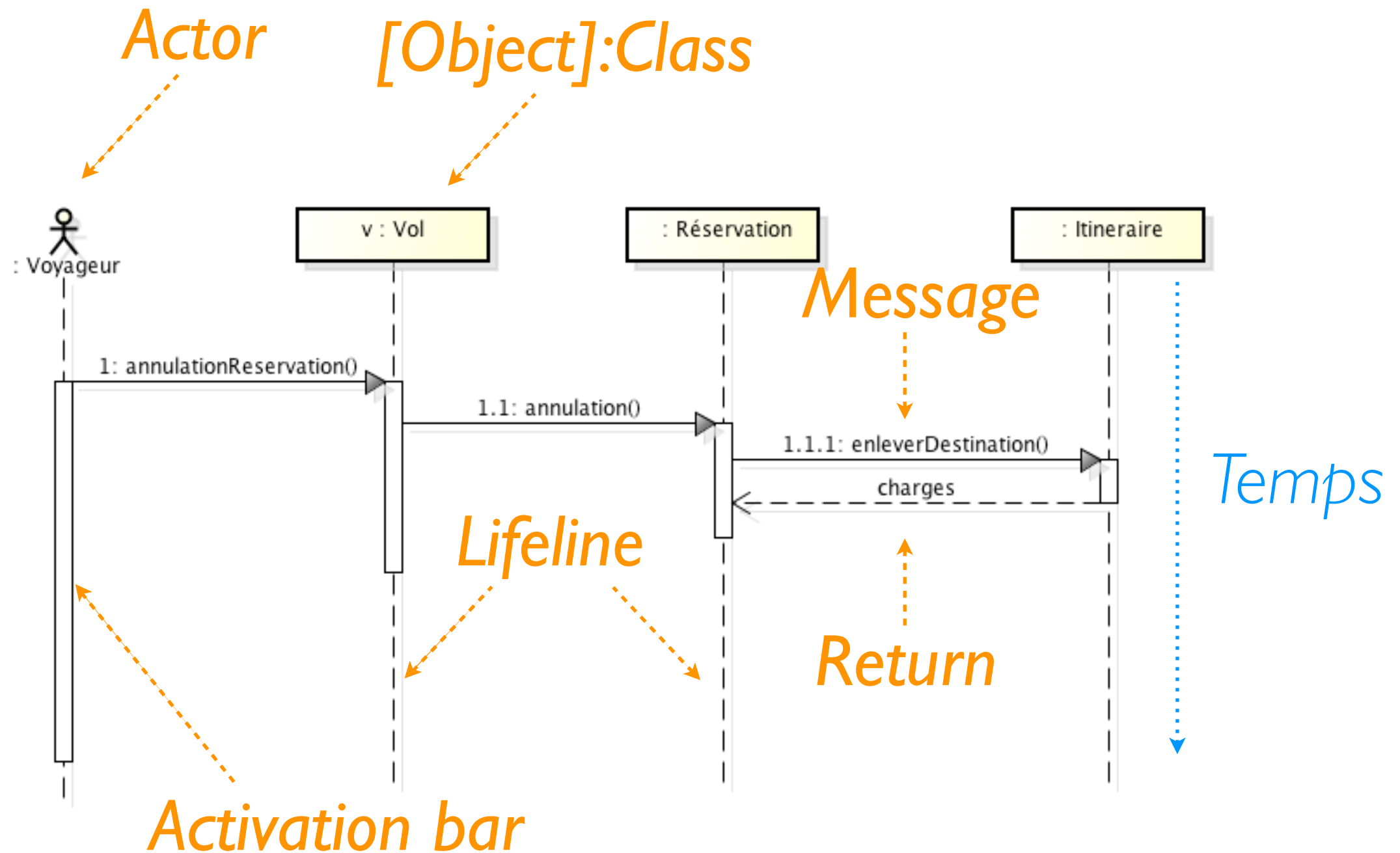


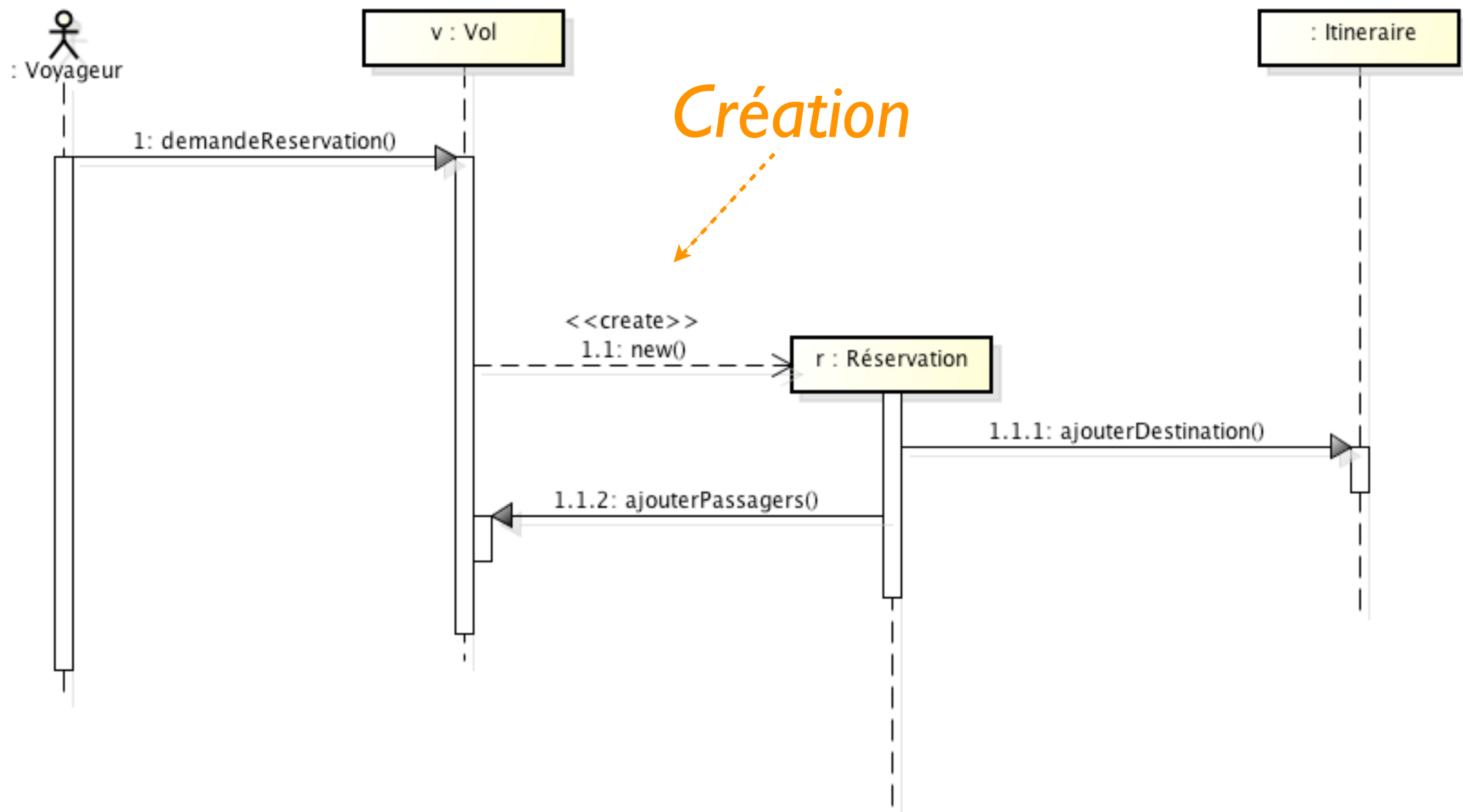
Diagramme de séquence

- Une ligne verticale, appelée *ligne de vie*, est accrochée à chaque objet ou acteur
- La ligne de vie s'épaissit pour devenir une *boîte d'activation* lorsque l'objet est actif
- Un *message* est représenté par une flèche joignant deux boîtes d'activation.
- Un *message* a un nom et peut aussi avoir une liste d'arguments et une valeur de retour.

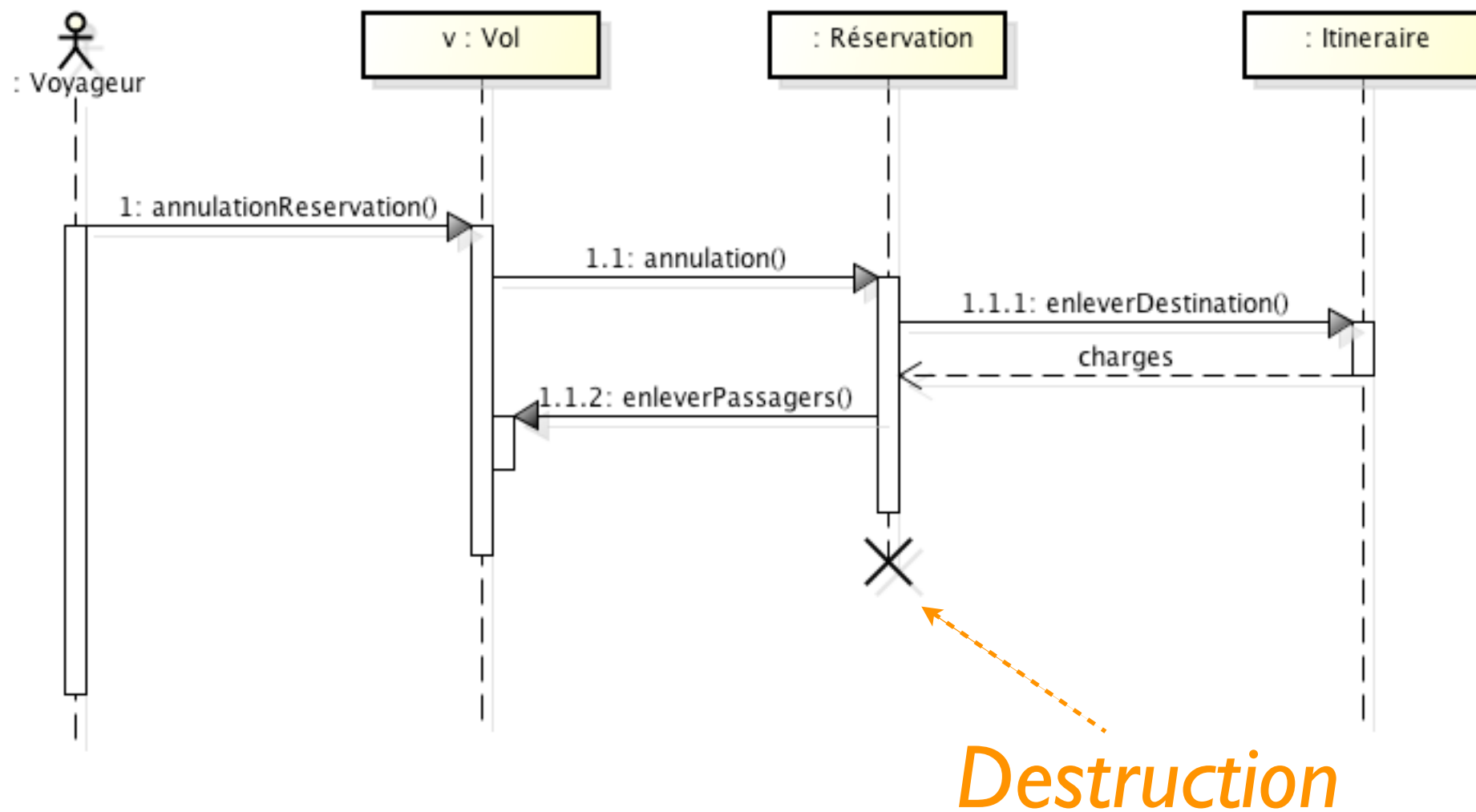
Diagramme de séquence



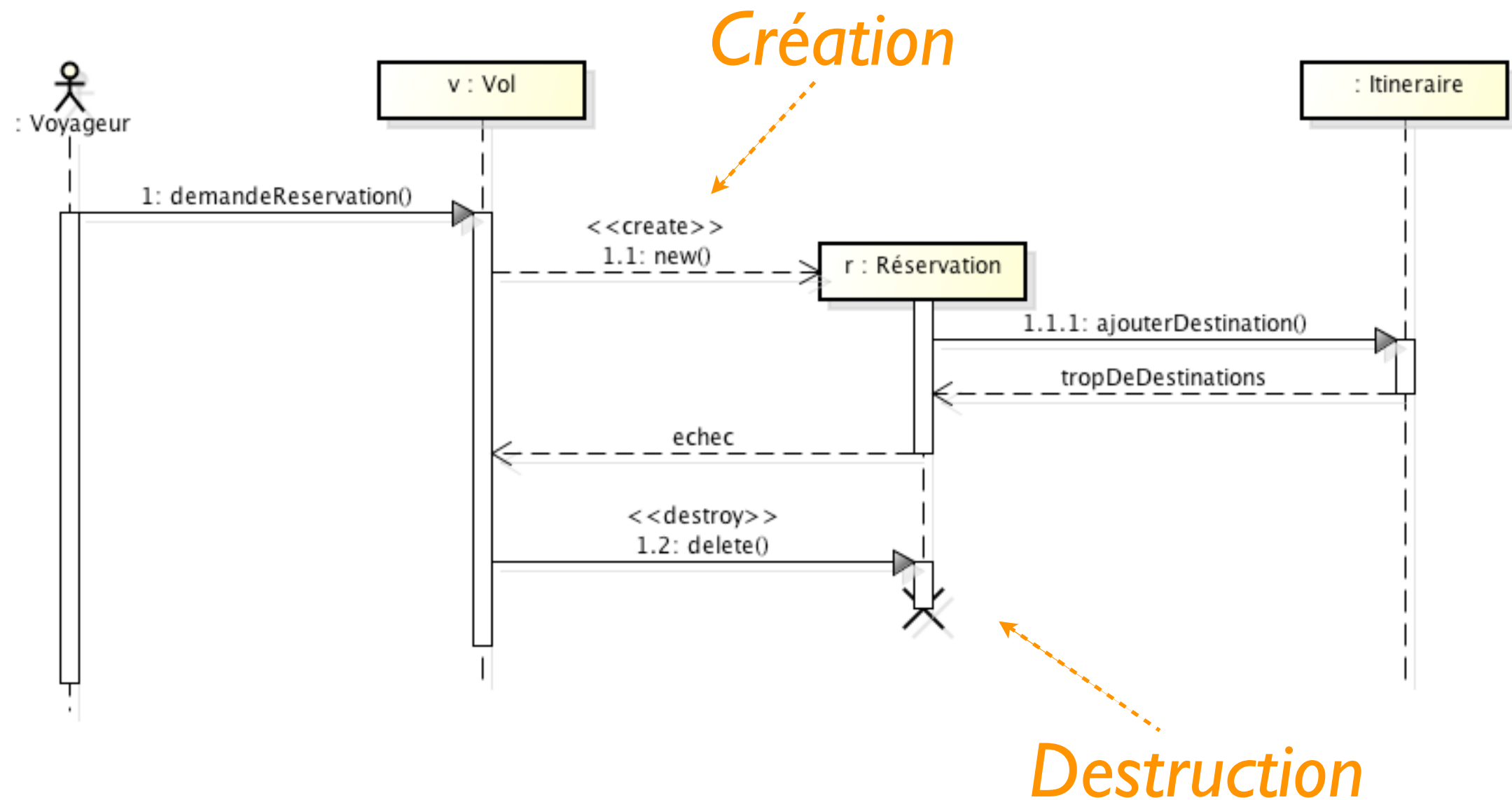
Création d'objets



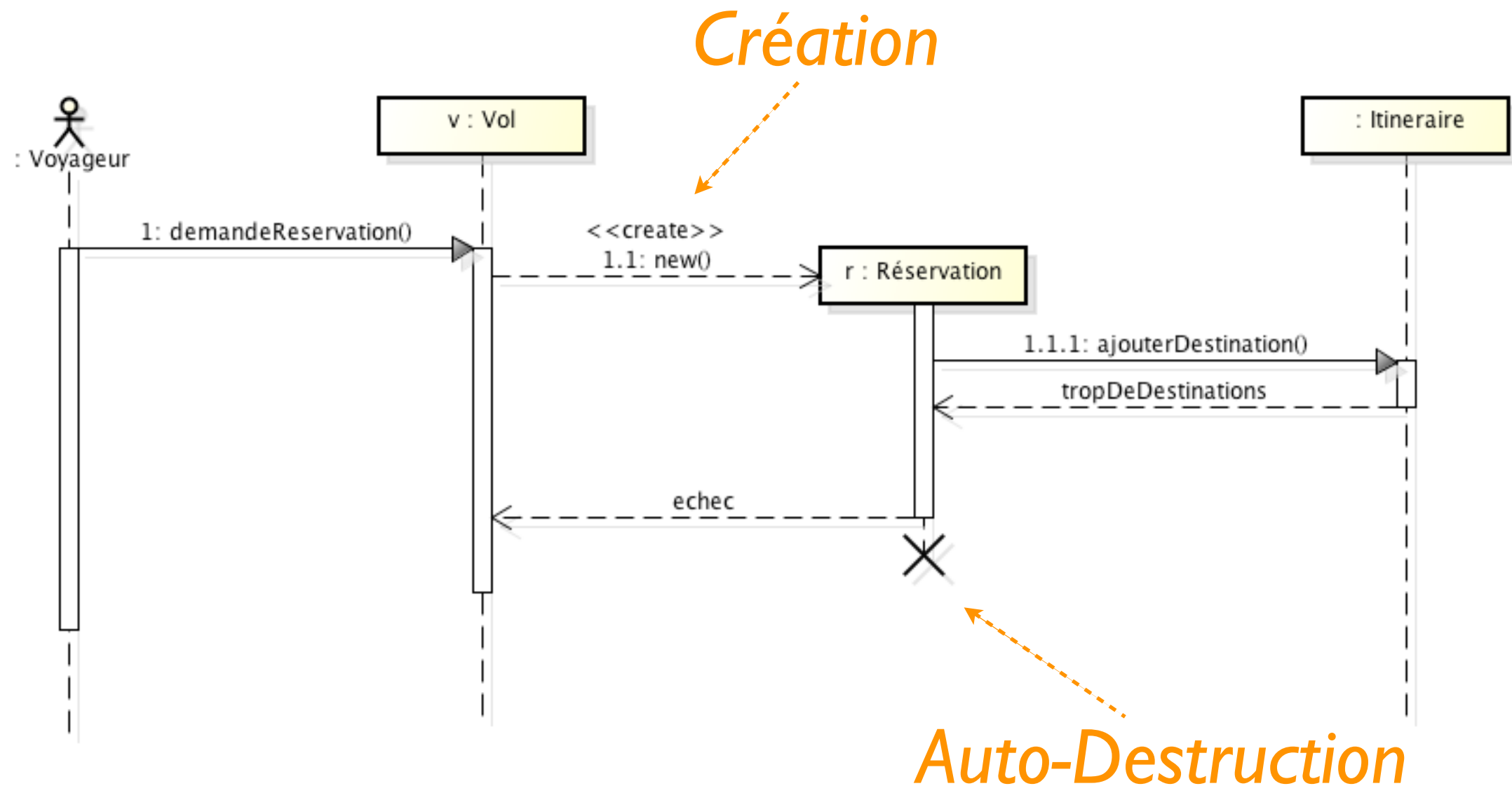
Destruction d'objets



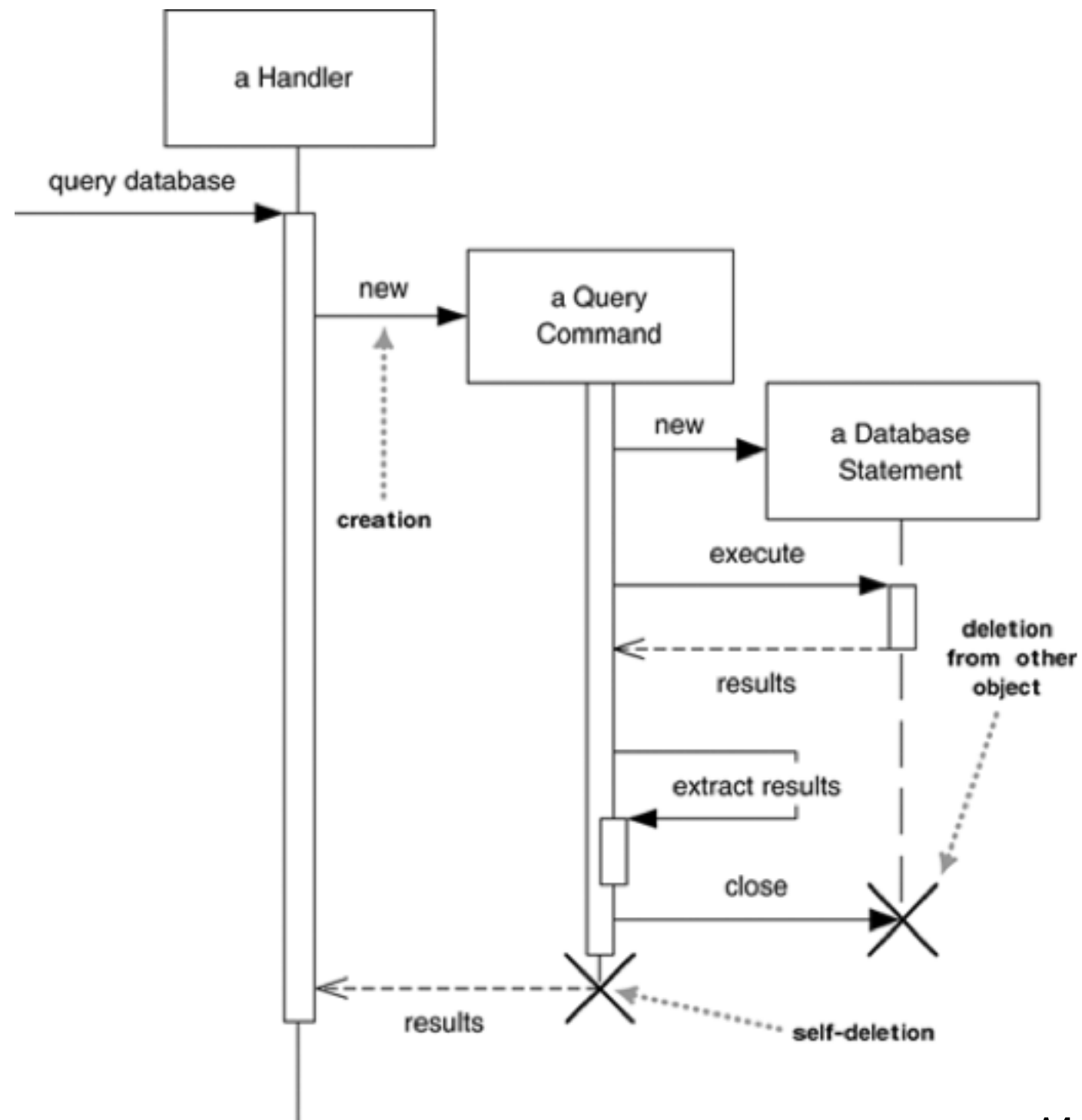
Création/(auto-)destruction d'objets



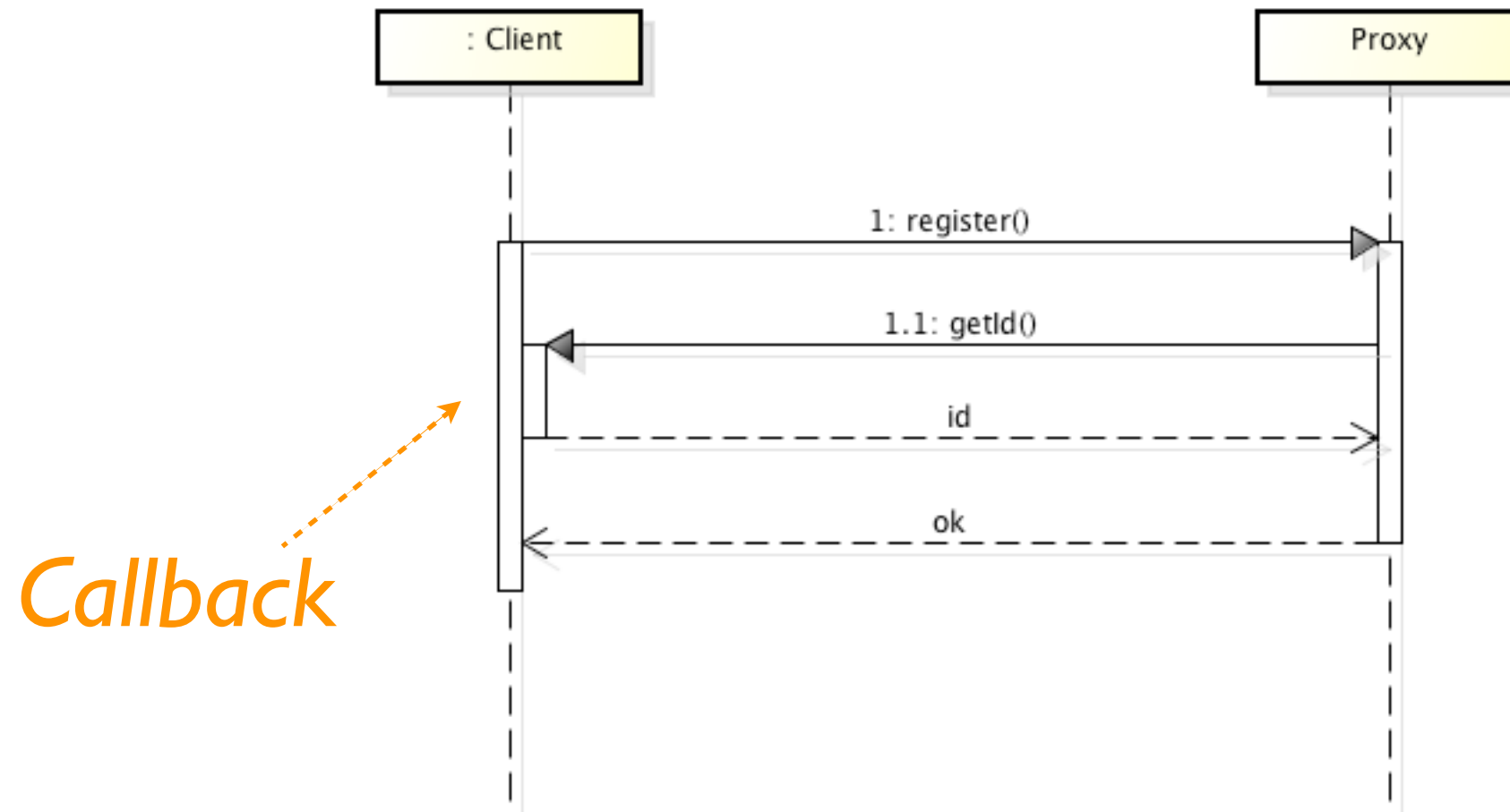
Création/(auto-)destruction d'objets



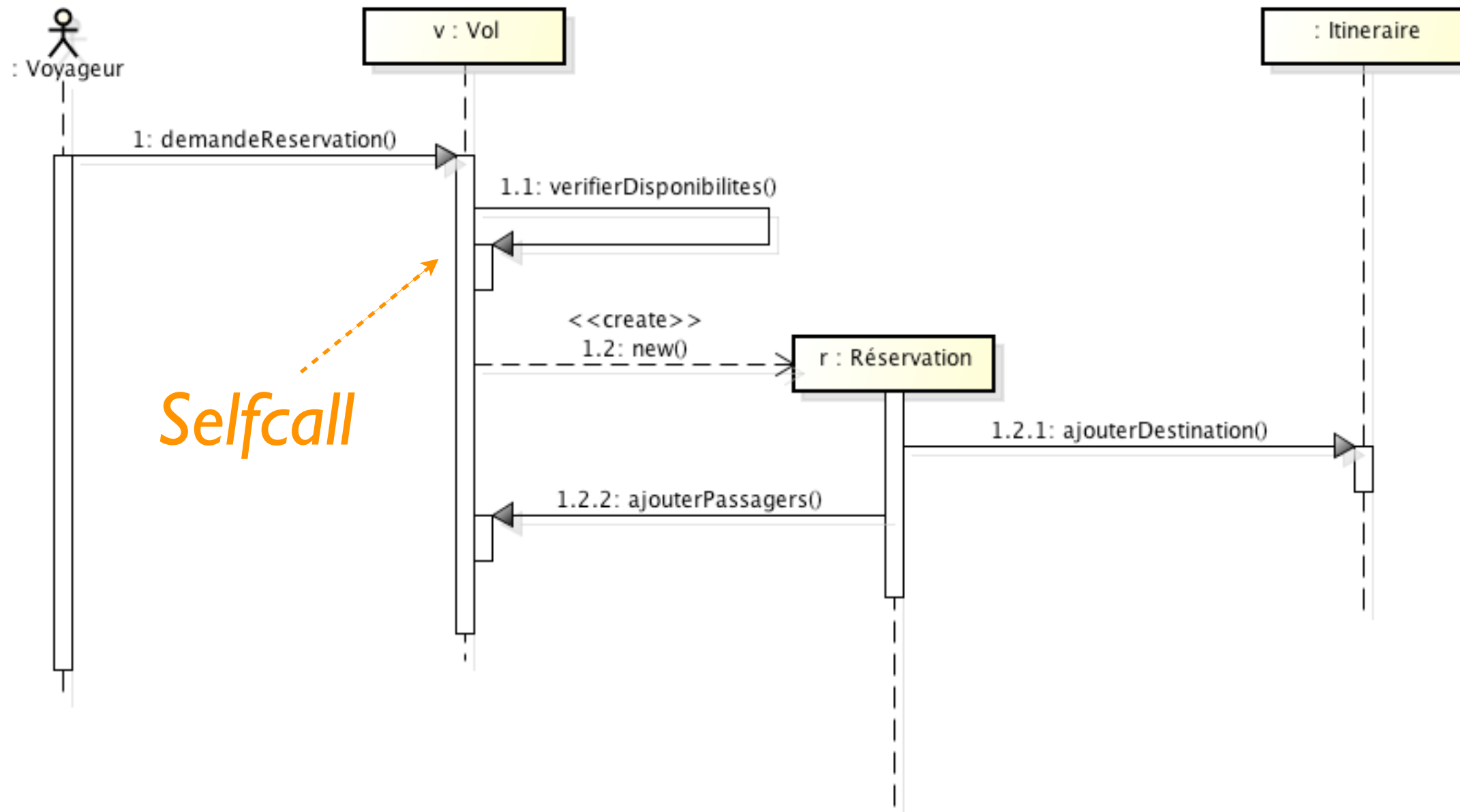
Création/destruction



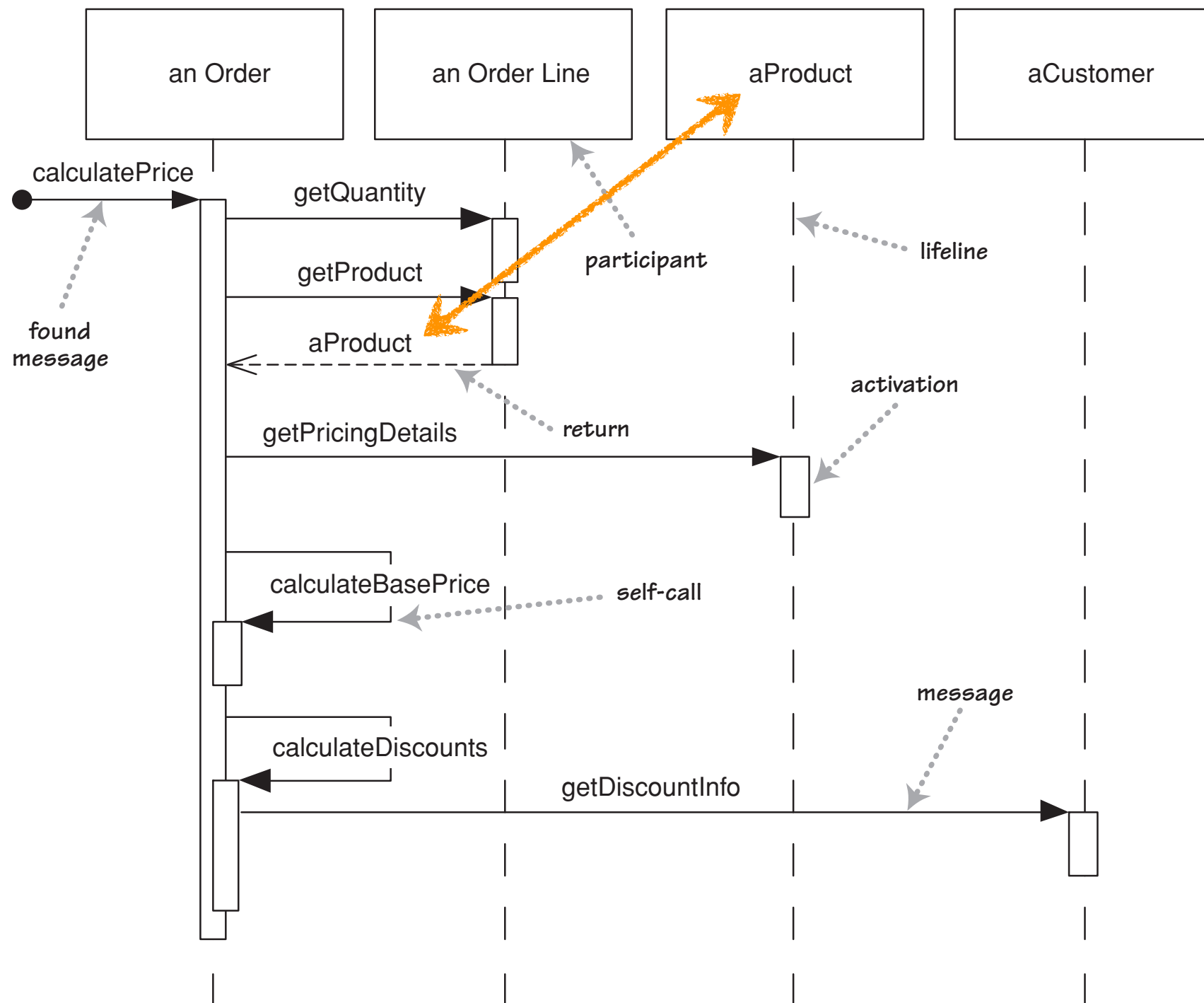
Callback



Selfcall



Example

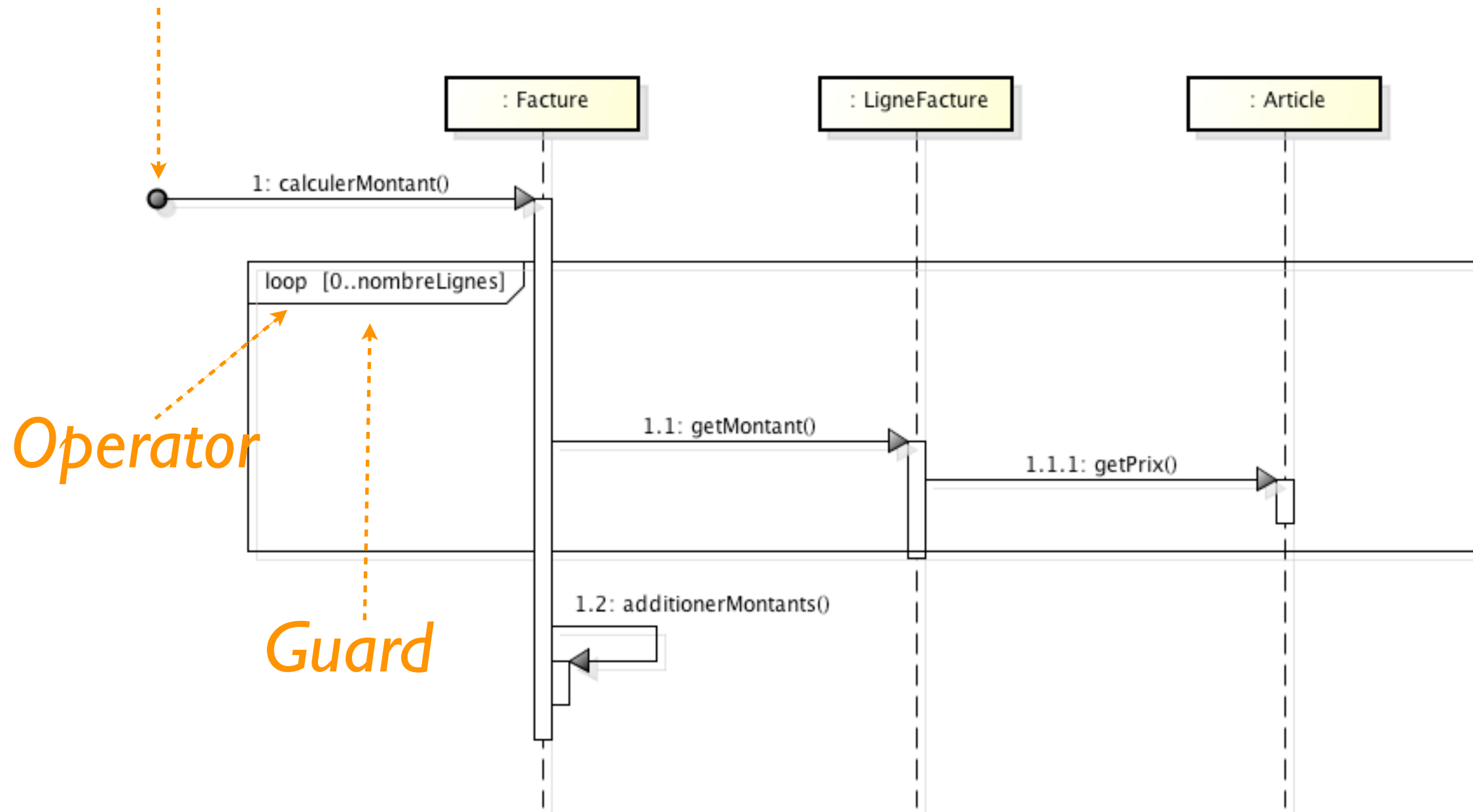


Les opérateurs

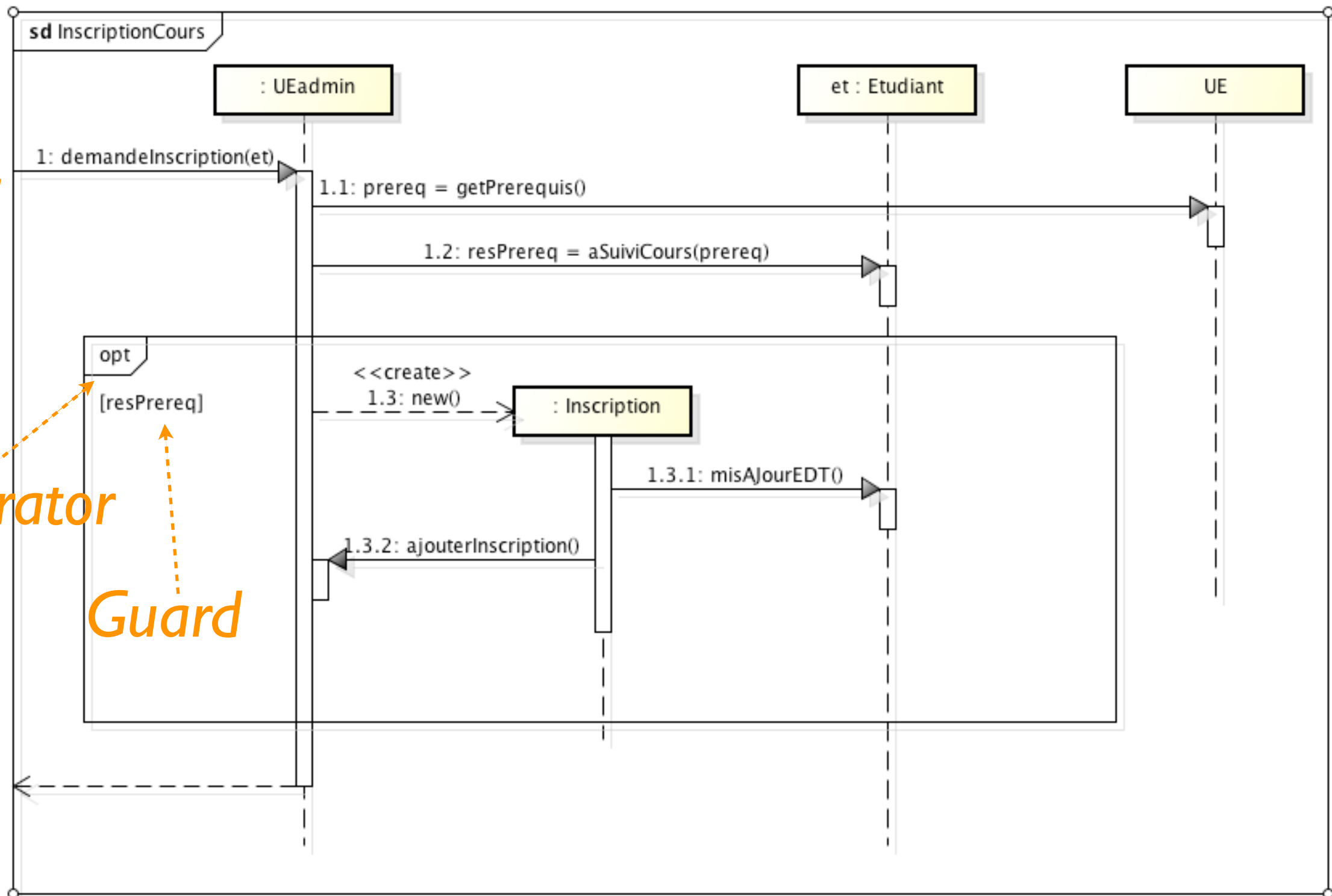
- *alt* : ne s'exécute que si une condition est remplie
- *opt* : ne s'exécute que si une condition est vraie
- *loop* : le fragment peut s'exécuter plusieurs fois
- *par* : les fragments s'exécutent en parallèle
- *ref* : référence à un autre fragment
- ...

Itérations

Found message



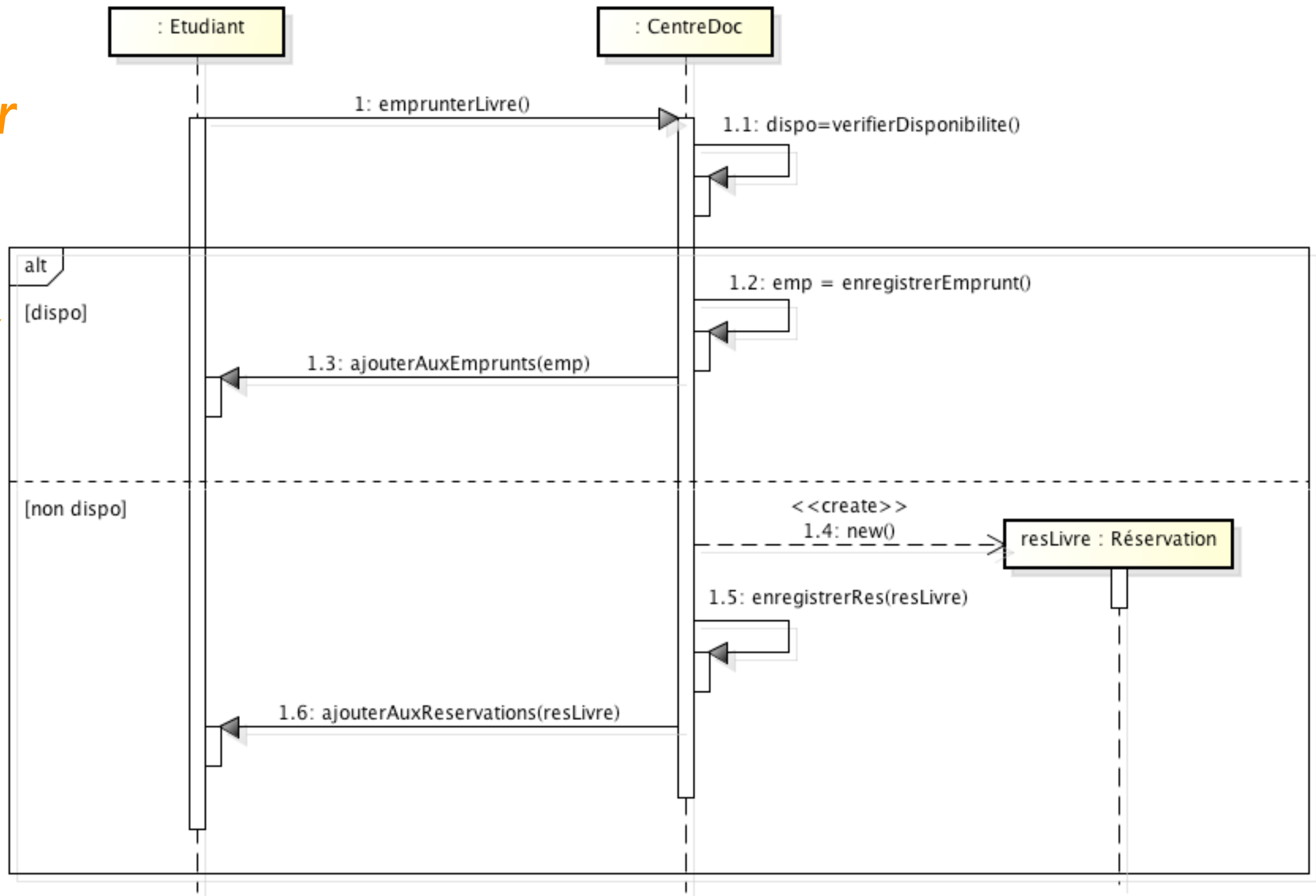
Optionnelle



Conditionnelle

Operator

Guards



Example

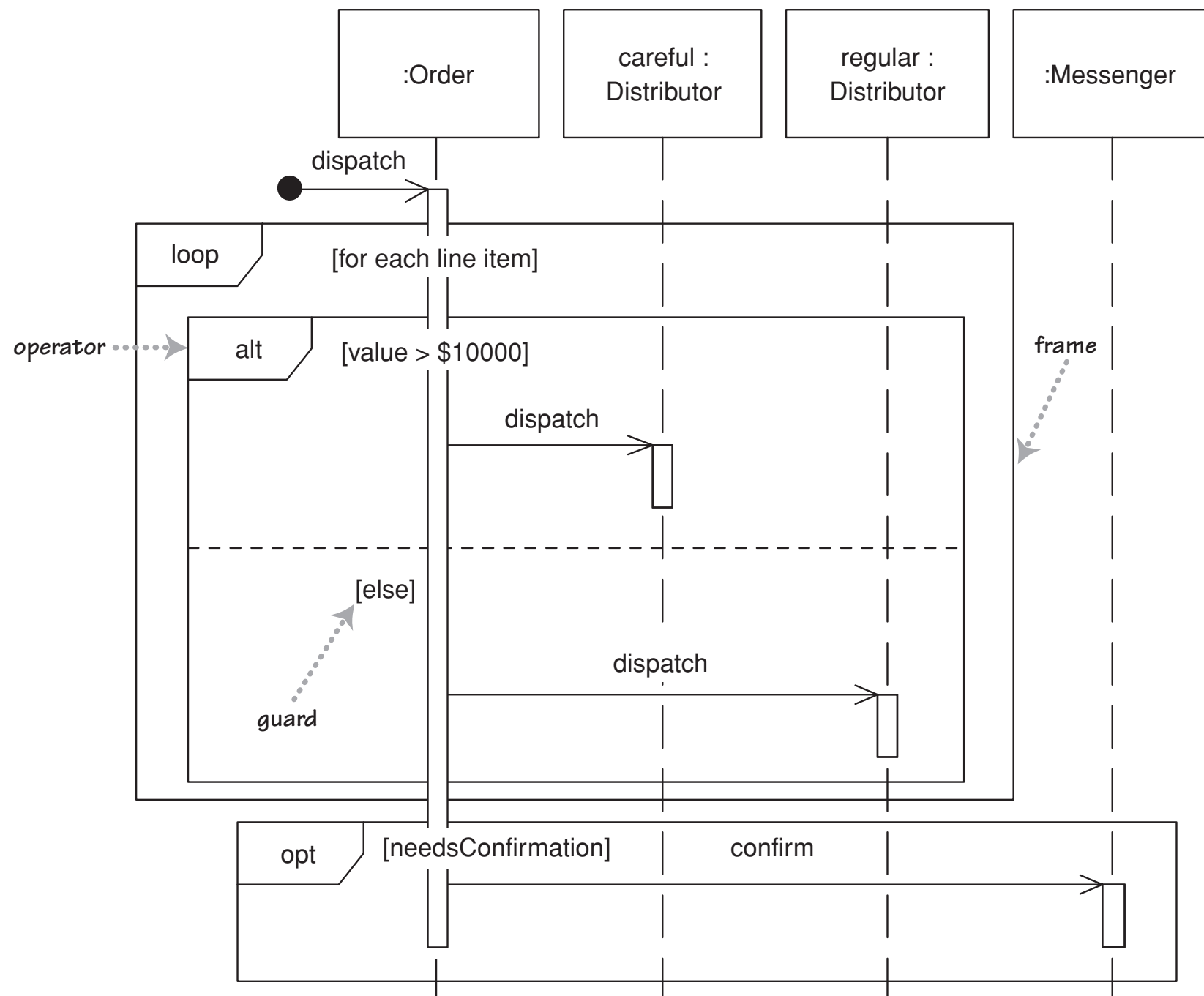


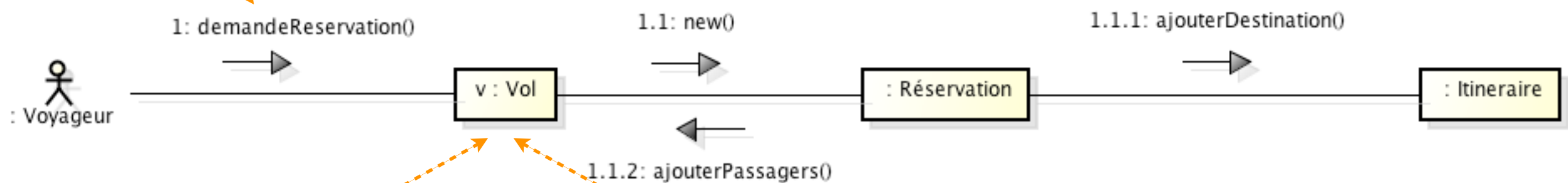
Diagramme de communication

Illustre comment les objets coopèrent dans la réalisation d'une interaction

- un graphe dont les objets sont les sommets
- des liens de communication
- les messages (représentés par des flèches) sont associés à ces liens
- l'ordonnancement temporel est indiqué à l'aide d'une numérotation

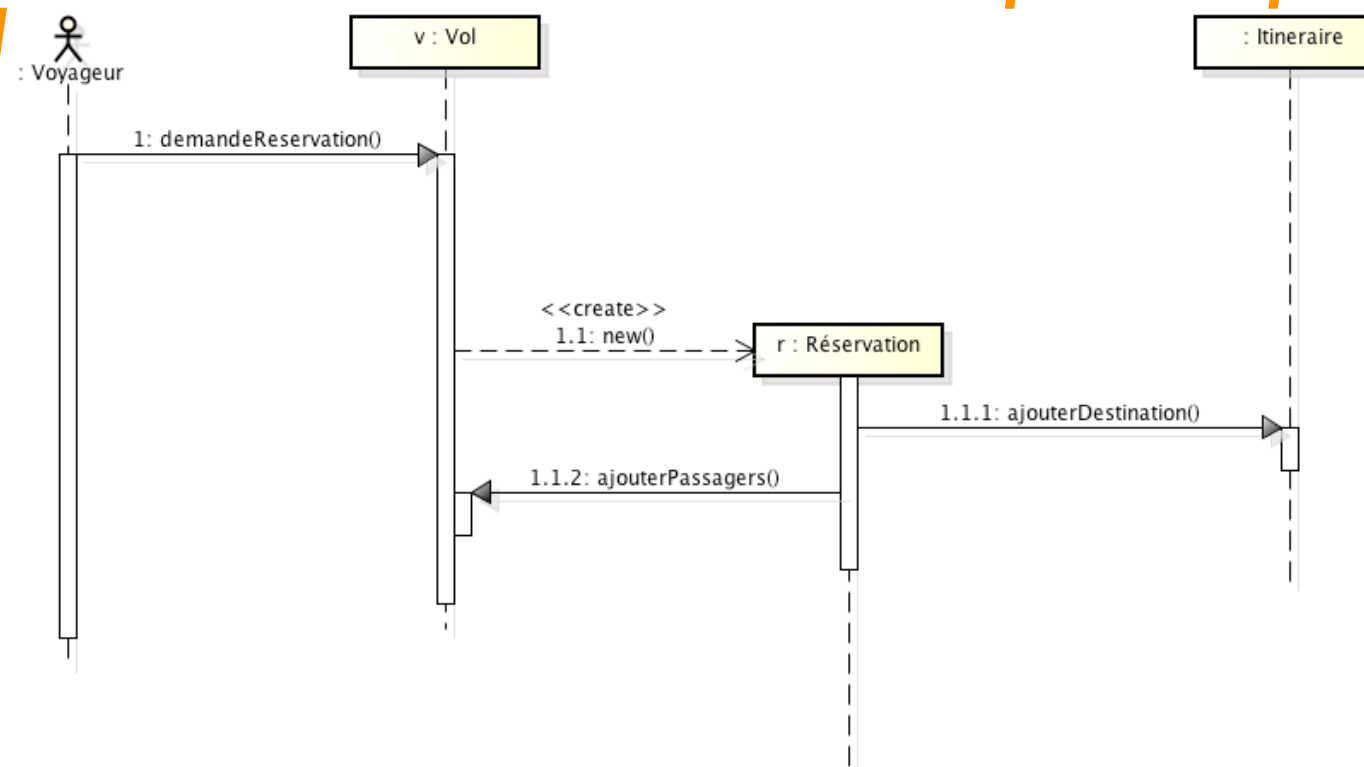
Diagramme de communication

Message

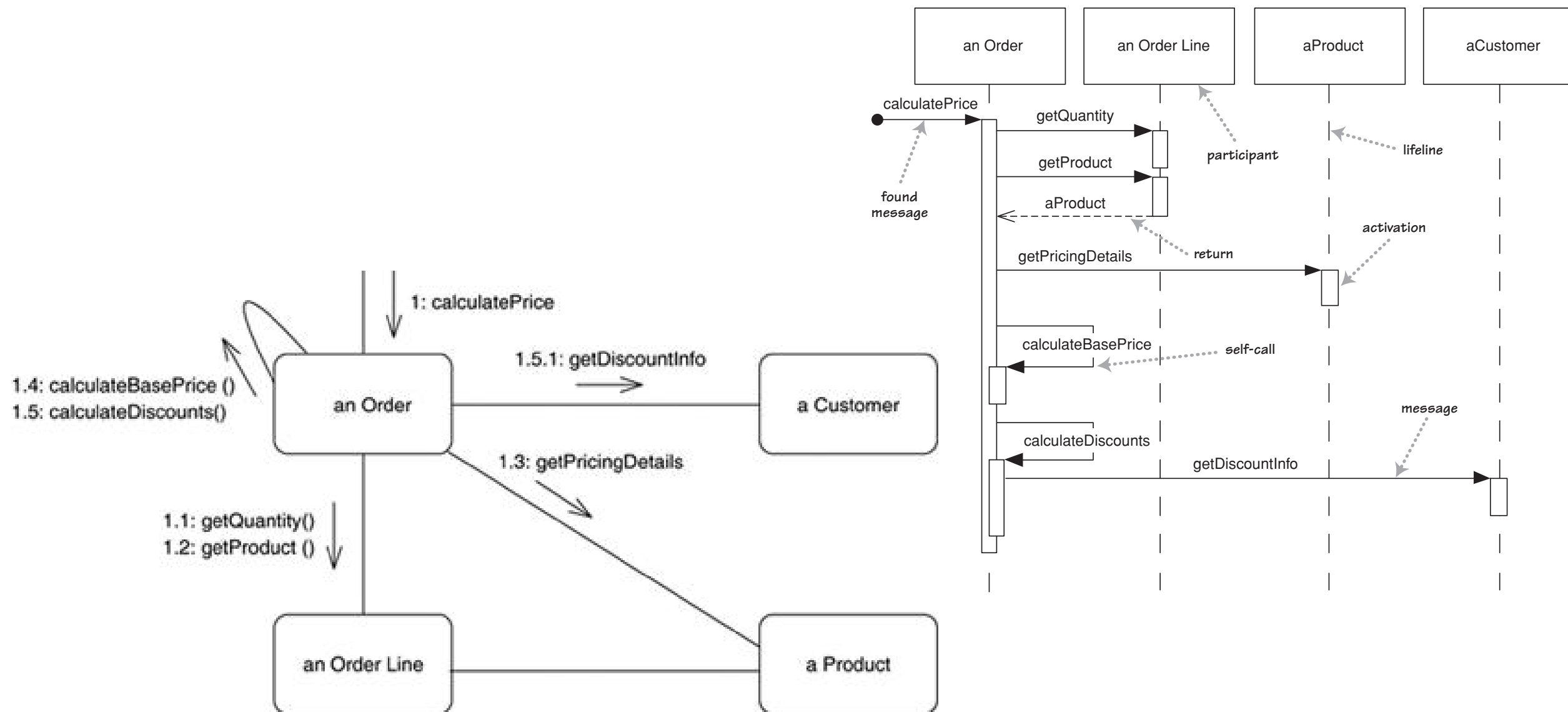


Lifeline
“v” de classe Vol

Individual participant



Example



Comment choisir

Les diagrammes de communication

- Peuvent être vus comme la prolongation d'un diagramme de classes
 - préférable lorsque l'interaction est déduite du diagramme de classes
 - sont aussi très utiles pour valider des diagrammes de classes

Comment choisir

Les diagrammes de séquence

- Rendent explicite la séquence temporelle dans l'interaction
 - choix naturel lorsque l'interaction est construite à partir d'un use-case
- Facilite une écriture détaillée des messages
 - moins d'espace est généralement disponible dans un diagramme de communication

Questions ?

2h