

# Modélisation / Conception de logiciel

## Chapitre 6 – Rappels UML DSQ – Diagramme de Séquences



**Véronique DESLANDRES**  
Licence DEVOPS, IUT de LYON

# Plan de ce cours

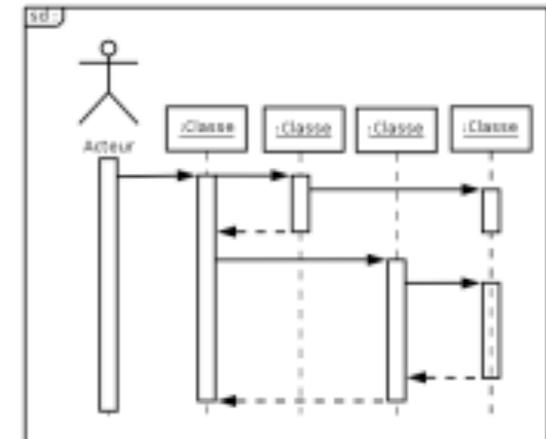
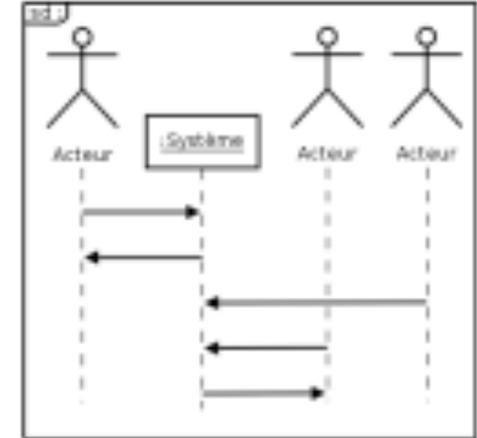
- Diagrammes de séquence (DSQ) système - s5
- Diagrammes de séquences objets - s12
- Exemples de DSQ - s27
  
- Démarche de modélisation de la dynamique d'un système - s35
  
- Diagr. Activités vs. Séquences - s38
  
- Fiche Je retiens DSQ - s39



# 2 Utilisations : 2 types de DSQ

Diagr: Laurent Audibert, UML 2 - Developpez.com

- Portée système
  - Système vu comme une boîte noire
  - Appellation : « **DSQ système** »
- Portée Analyse détaillée ou Conception
  - On modélise les interactions **entre les objets** du système
  - Appellation : « **DSQ Objet** », « DSQ d'analyse », « **DSQ d'interaction** » ou « DSQ de Conception »

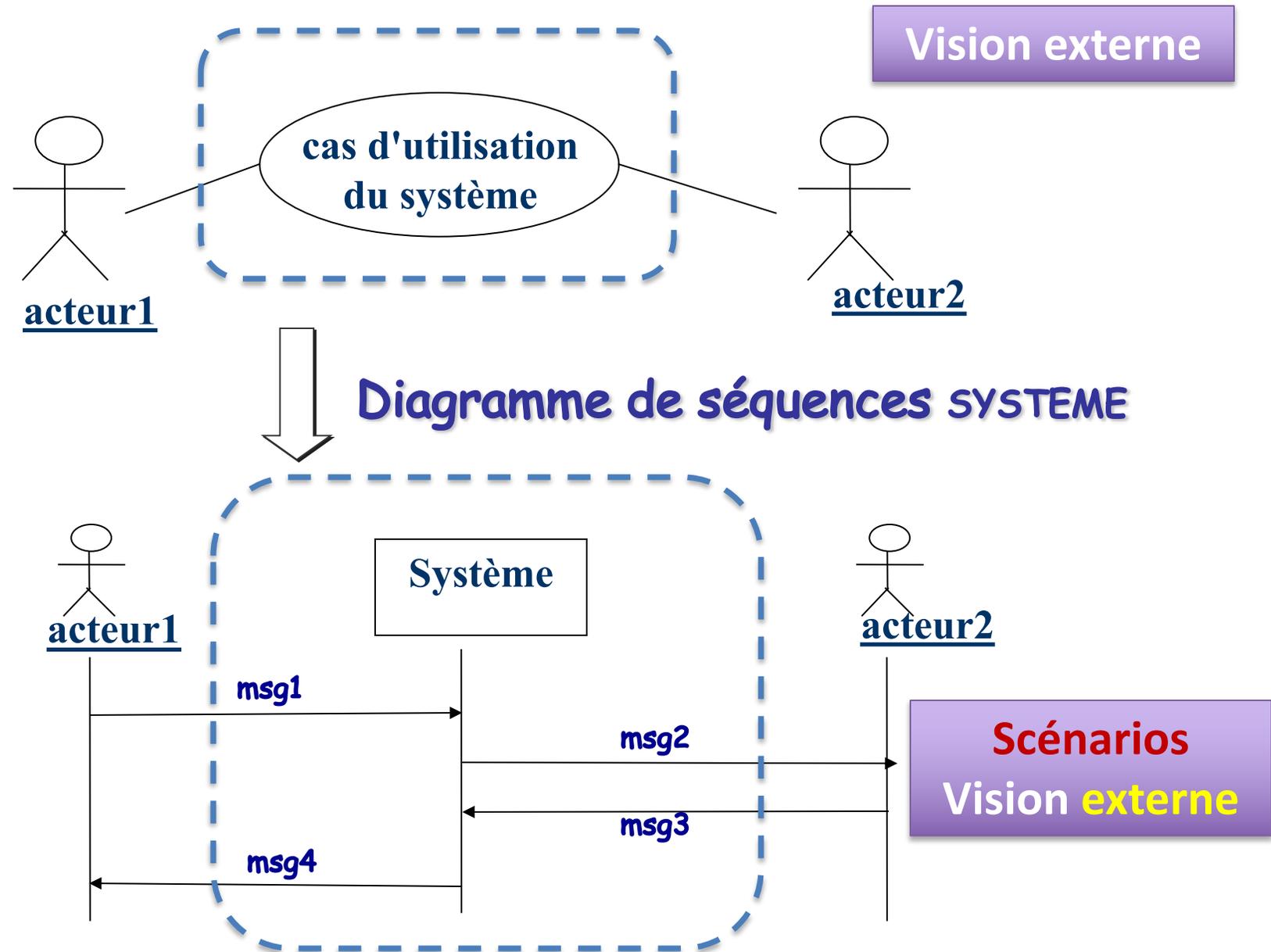


# Diagrammes de Séquences système

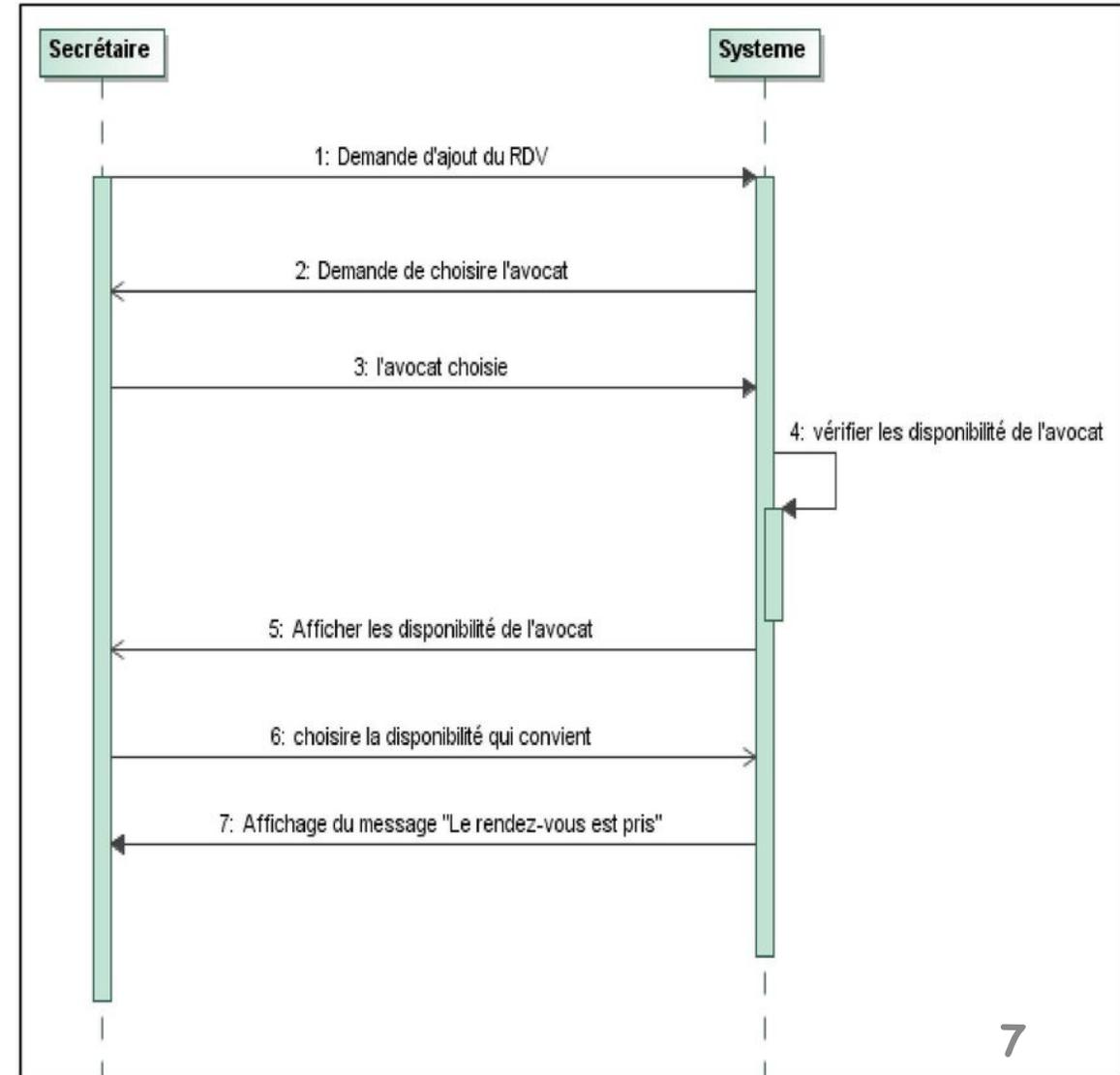
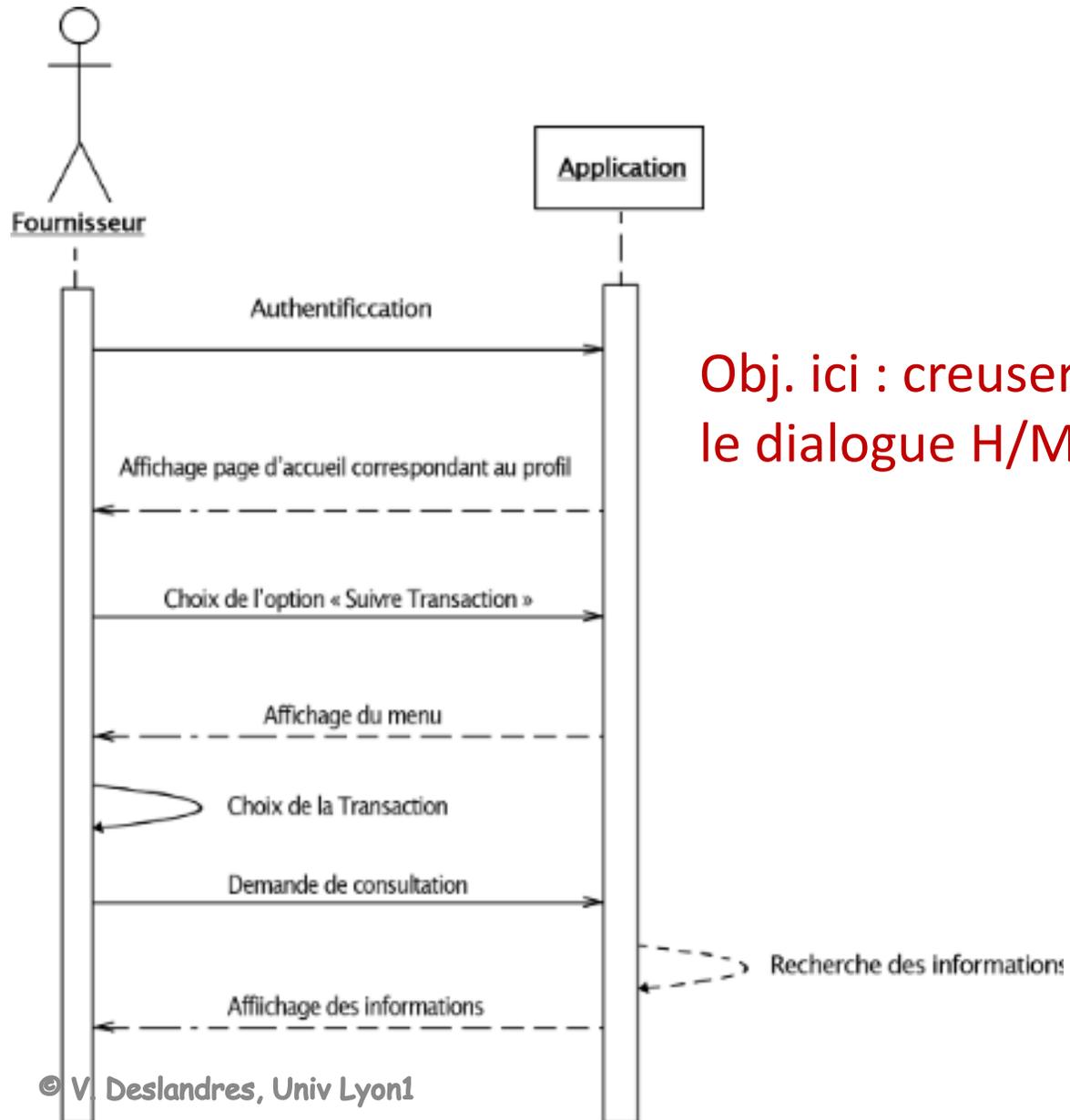
# Objectifs du DSQ Système

- Montrer les interactions des acteurs externes (obtenus avec le Diagramme des Cas d'Utilisation) avec le système
  - Sans détailler les objets qui participent aux processus (**système = boîte noire**, vision **interaction**)
  - Vision globale de tout ce que le système peut faire ou détail d'un processus particulier
- Usages
  - Définir les **interactions entre les différents composants logiciels** (serveurs, BD, framework, services web, etc.)
    - pour différents **processus utilisateurs**
  - Définir les **bases du dialogue H/M**

# On part des Cas d'Utilisation...

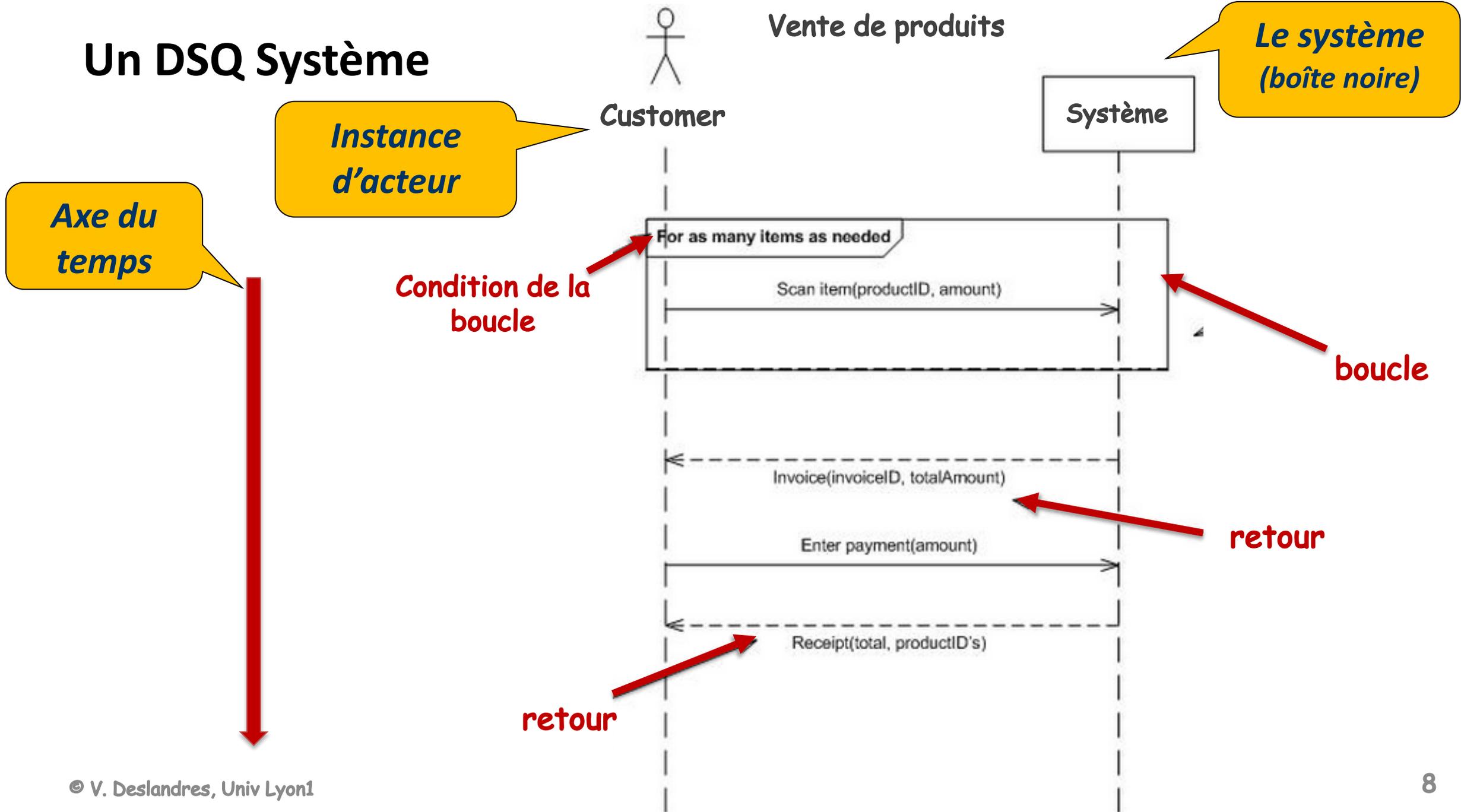


# Ex. de DSQ système

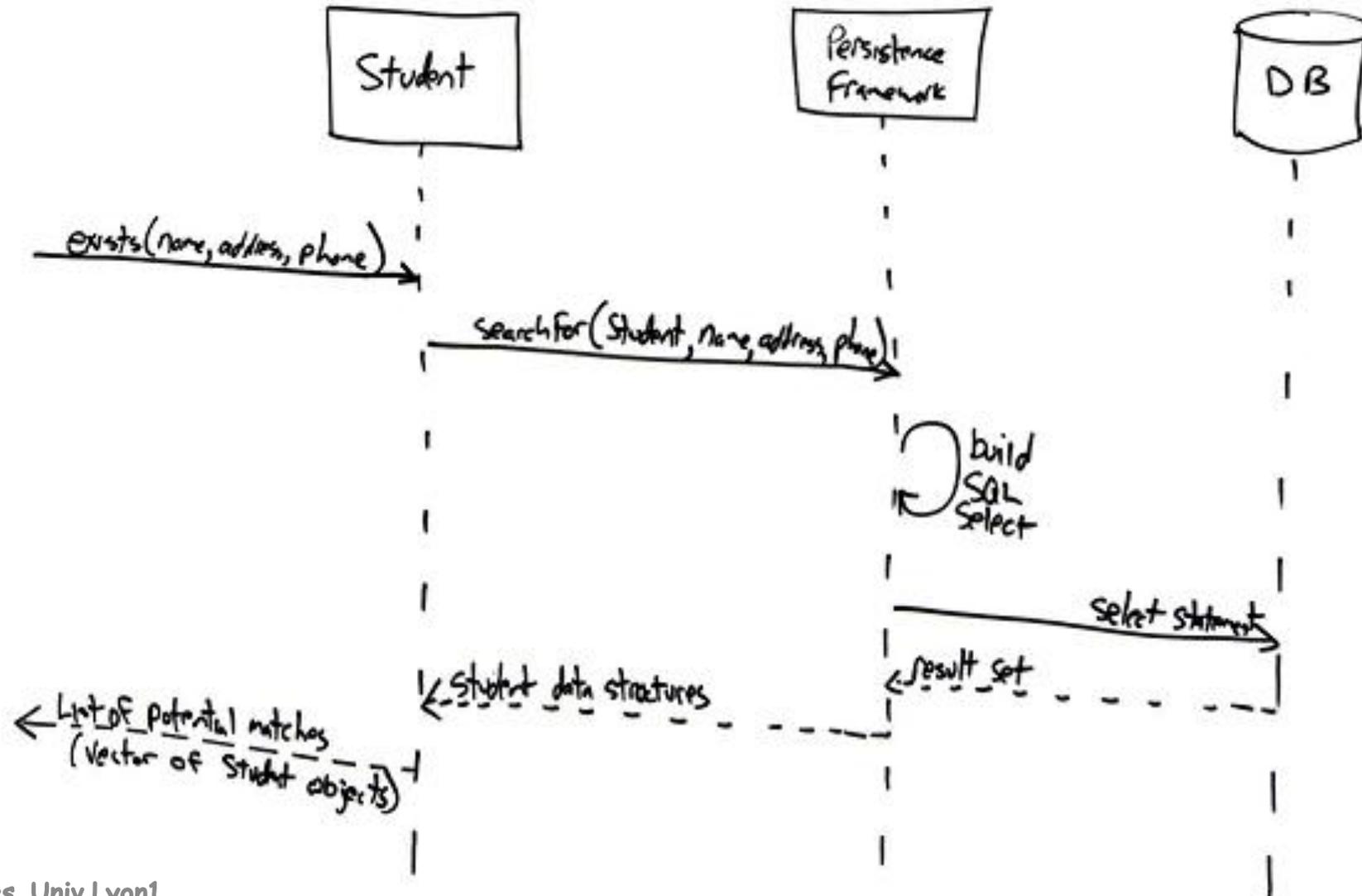


# Un DSQ Système

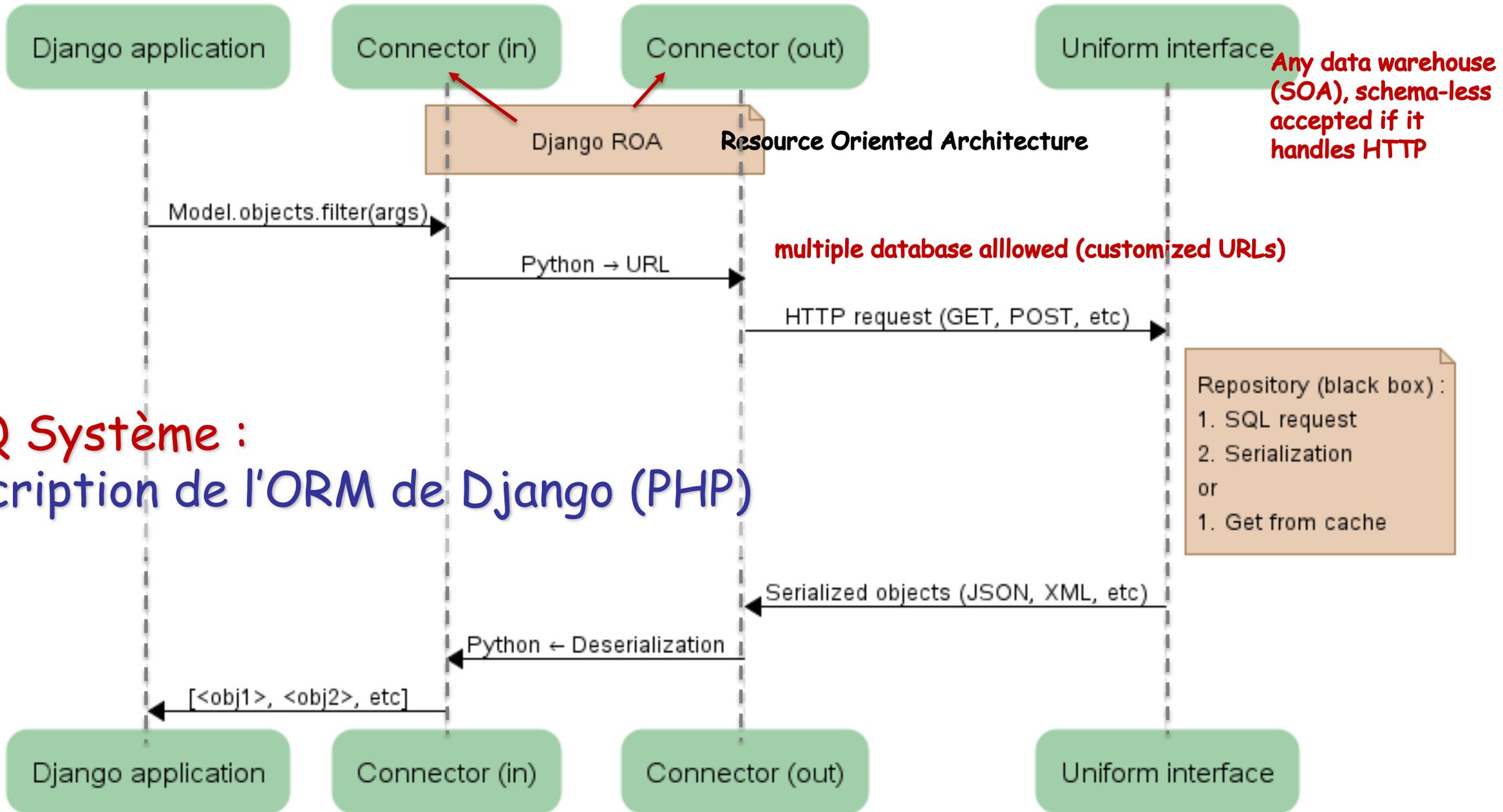
Vente de produits



# DSQ Système le plus fréquent : montrer les relations entre les 'sous-systèmes'



# DSQ Système : Description de l'ORM de Django (PHP)



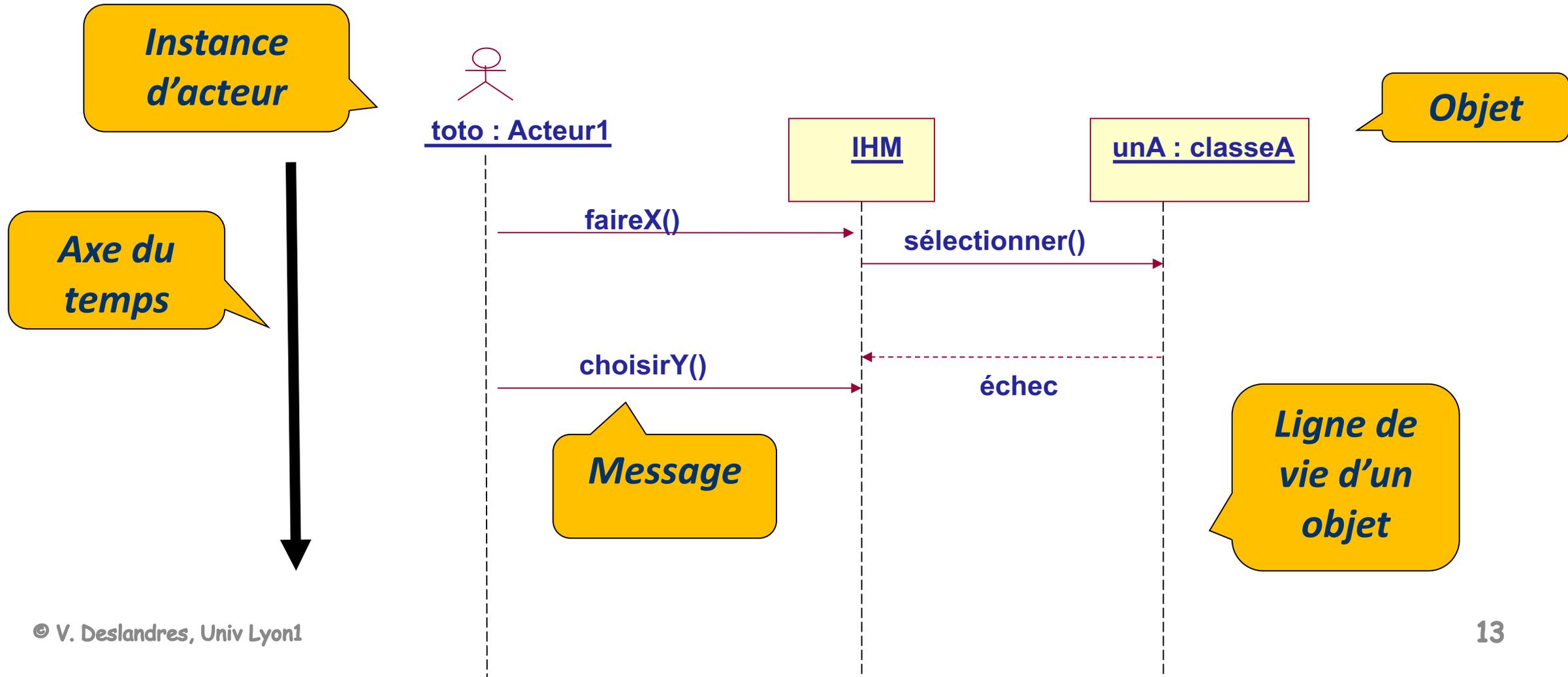
# Diagrammes de Séquences objets

# Diagramme de séquence « objet » (ou « Diagramme d'interaction »)

- Représentation des interactions entre un acteur et les objets
  - Vision **interne**
- **Modélisent des scénarios Utilisateur**
- Vision temporelle d'une interaction
- Souvent utilisé pour représenter le déroulement d'un **cas d'utilisation**
- Ou en **conception**, pour expliciter les choix techniques

# Diagramme de séquences objets en UML

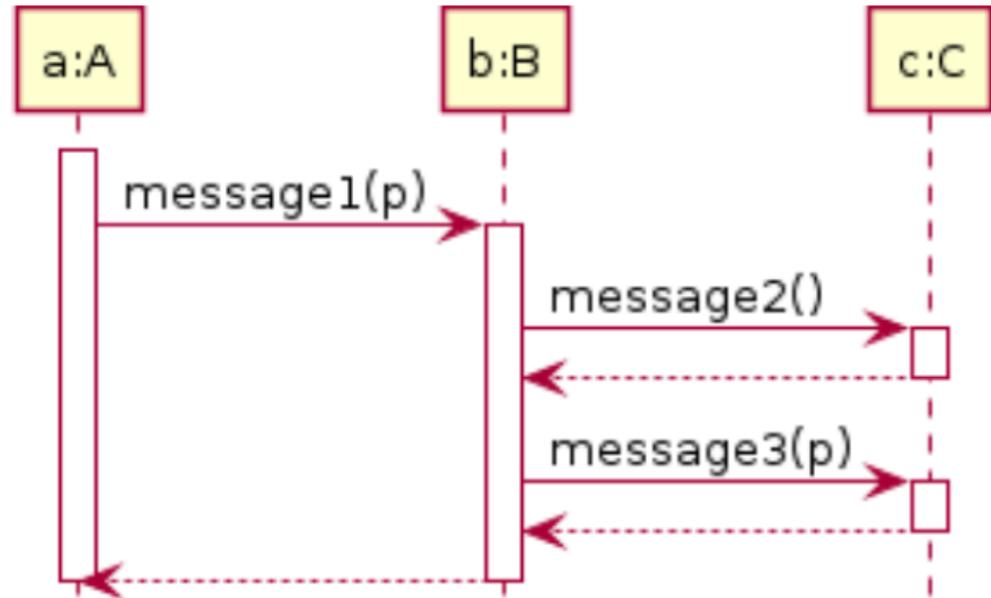
Vision interne



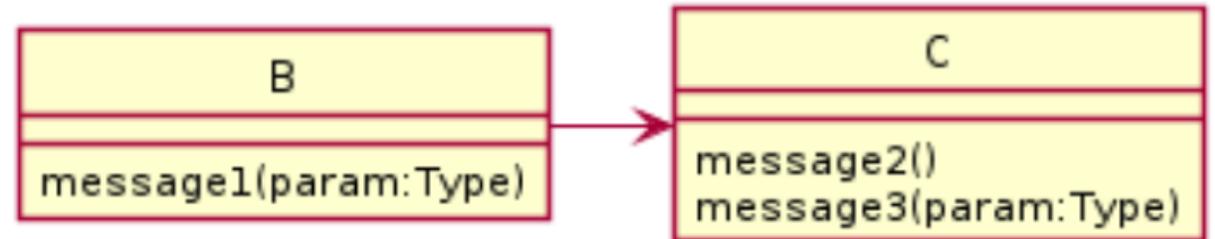
# Sémantique du DSQ Objet

- Les **messages** correspondent généralement à **l'appel d'une méthode** de l'objet
  - Arrêter(unProcessus), modifier(salaire)
- Ou à des **consultations / sélections** de listes d'objets :
  - Lire(lesArticles)
  - Sélectionner(filters)

# Sémantique des messages



Les messages correspondent à l'**appel de méthodes** de la classe **récepteur**



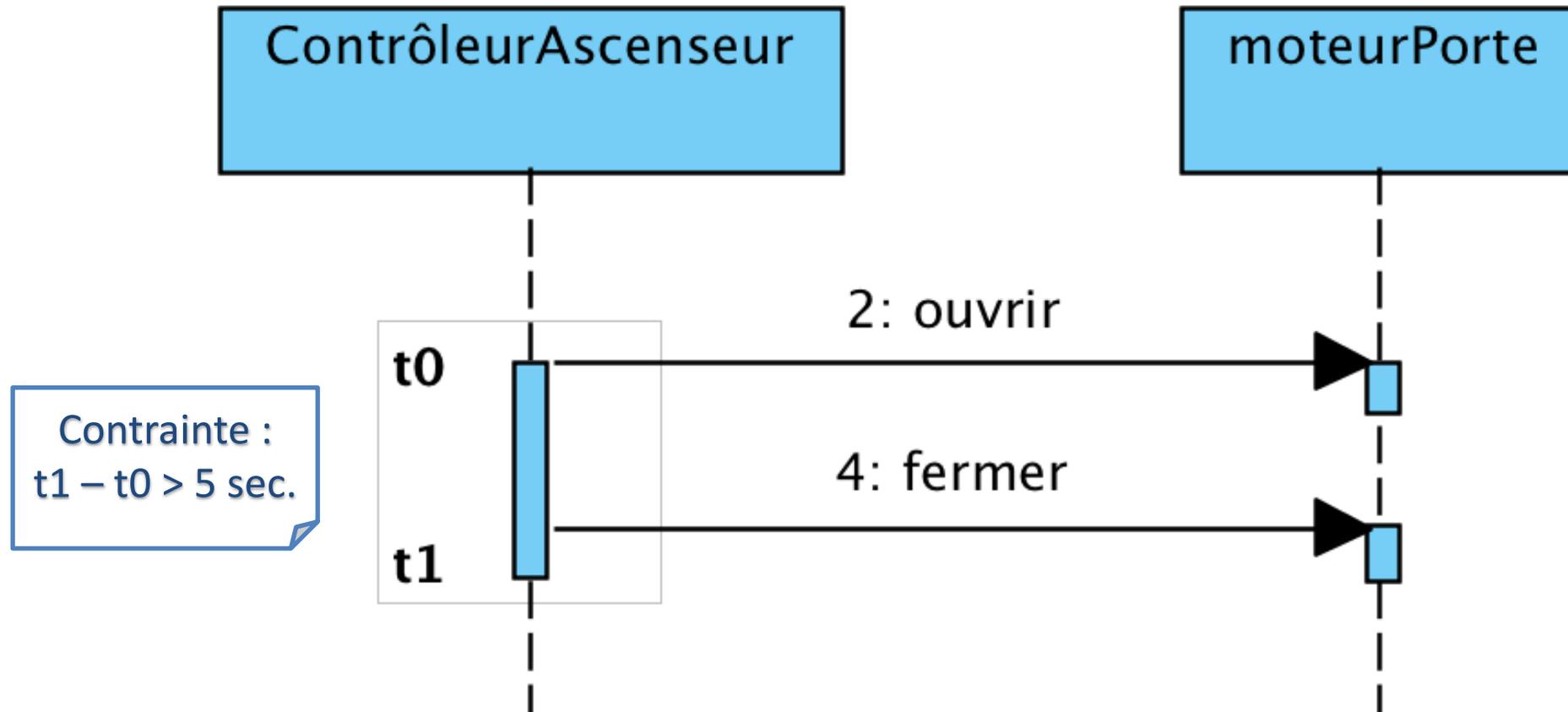
```
class B {
    C unc;
    message1(p:Type) {
        unc.message2();
        unc.message3(p);
    } ...
}
```

```
class C {
    message2() {
        ...
    }
    message3(var:Type) {
        ...
    } ...
}
```

# Sémantique du DSQ Objet (2)

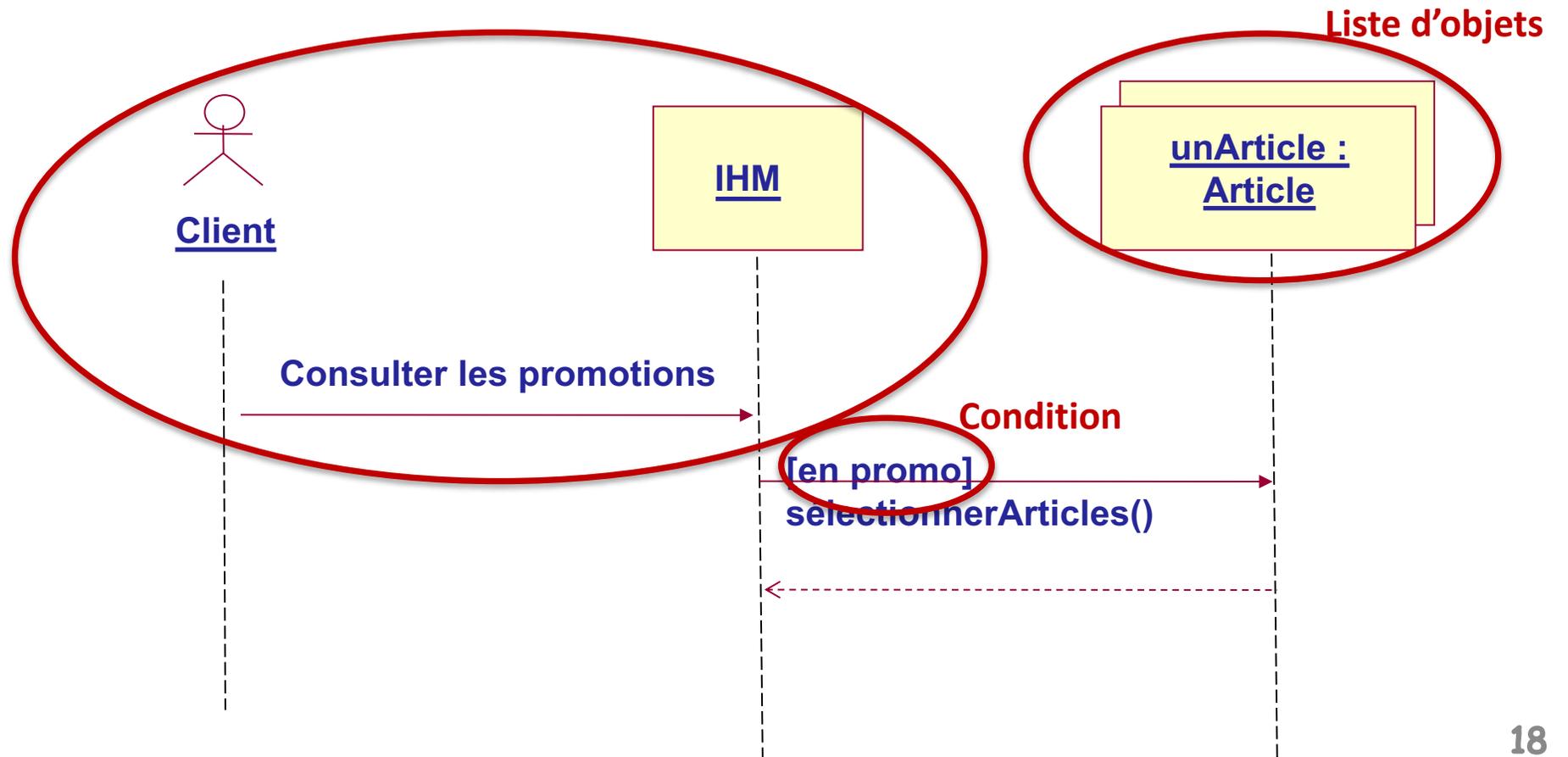
- On peut mentionner explicitement des **objets multiples** (= listes d'objets)
- Les **conditions** sur les messages, s'il y en a, figurent en **prédicat** UML :
  - [en promo] [si  $x > 200$ ]
- On peut aussi définir des contraintes de **temporisation**

# Ex. temporisation : porte ascenseur



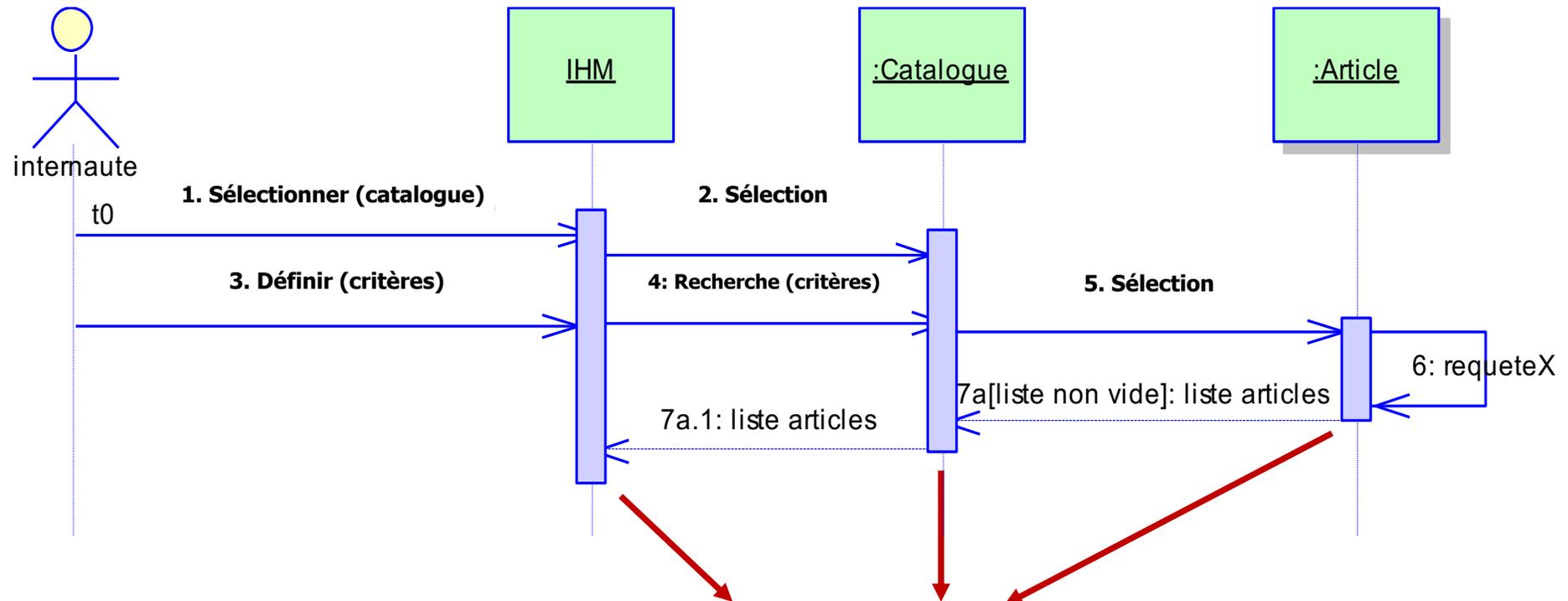
# DSQ Objet : convention graphique

- Un **diagramme de séquences objet** démarre toujours par un acteur qui sollicite un objet interface :



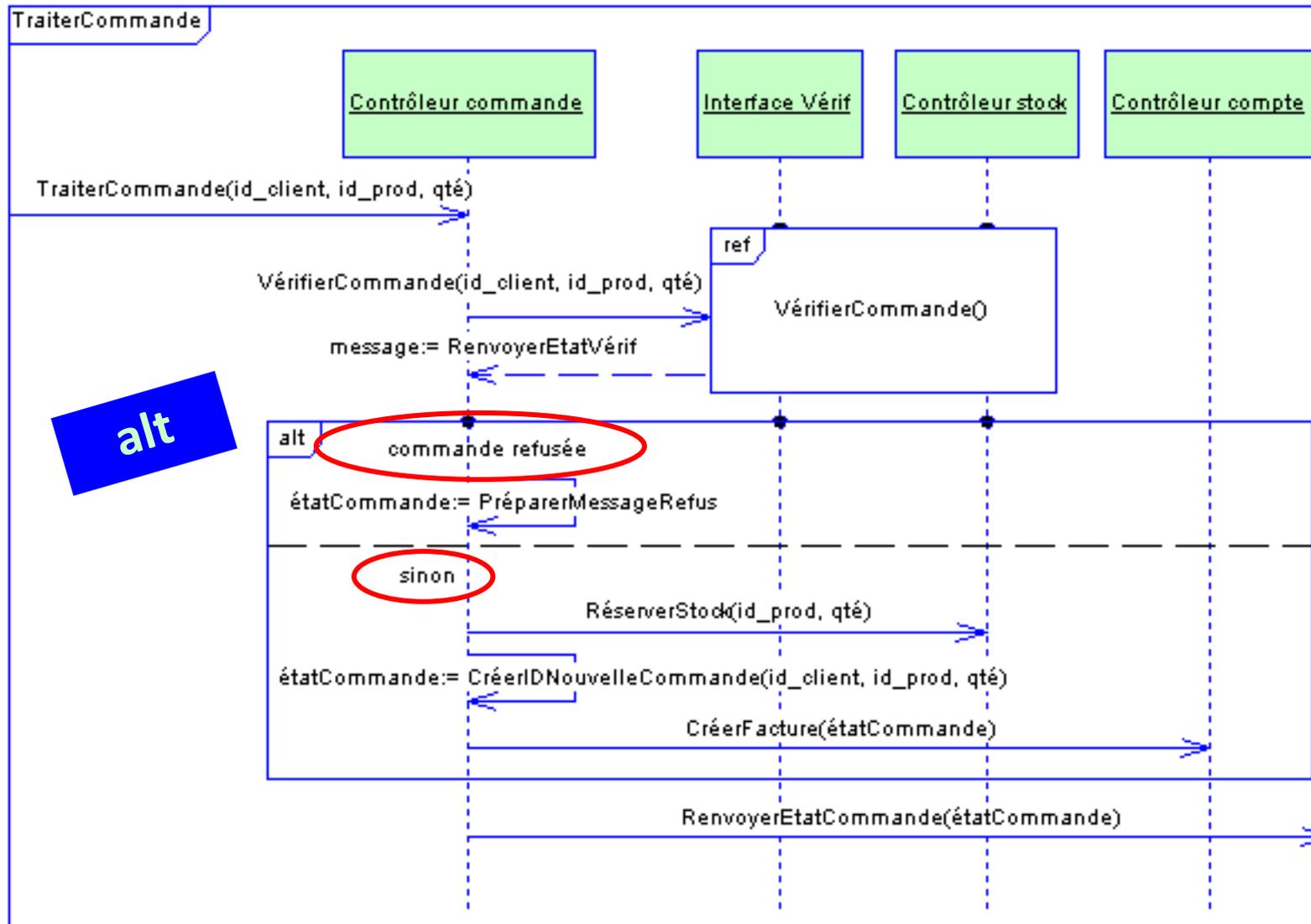
# Notion d'activation d'objet

Permet de représenter une séquence :



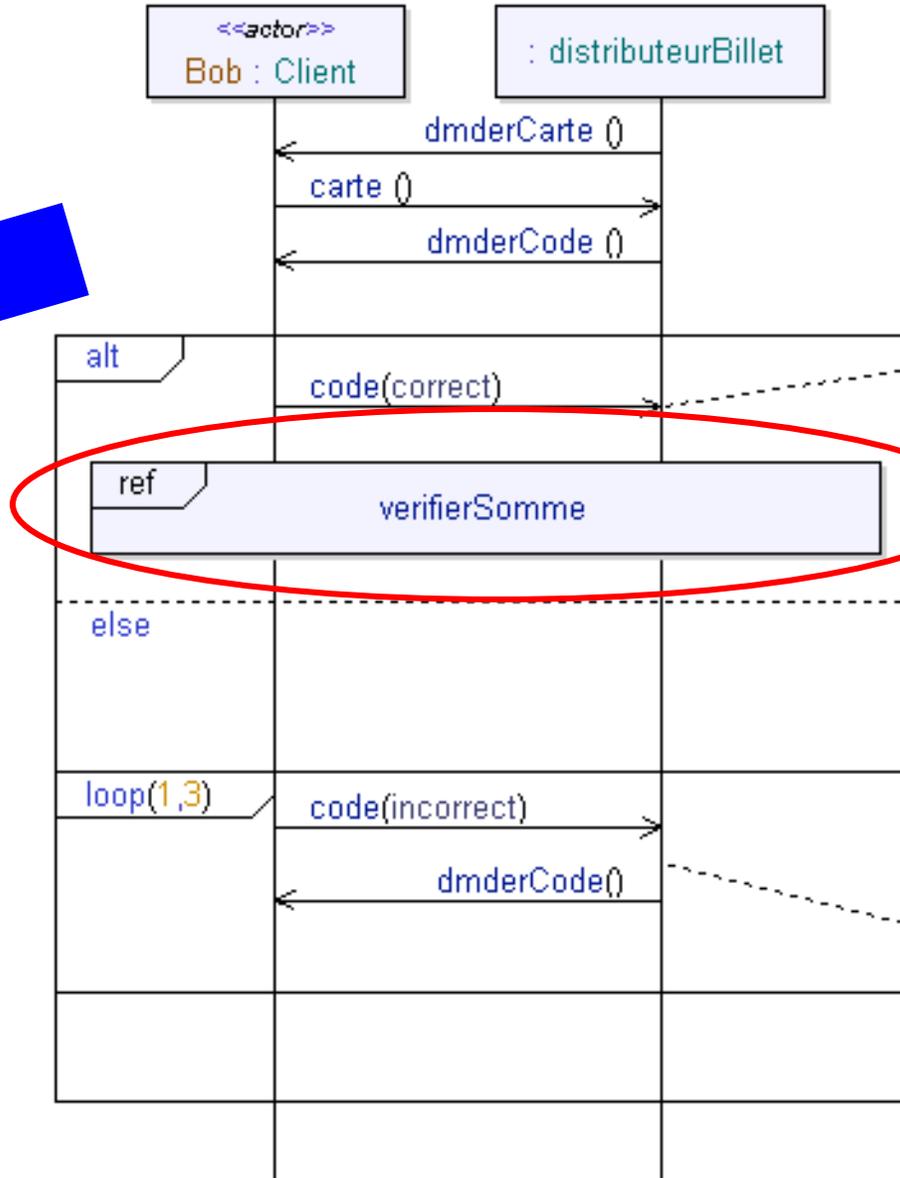
**Symbole d'activation de l'objet : continuité du flux d'activités**

# Bloc d'alternative



## DSQ : référence à un autre diagramme de séquence

ref



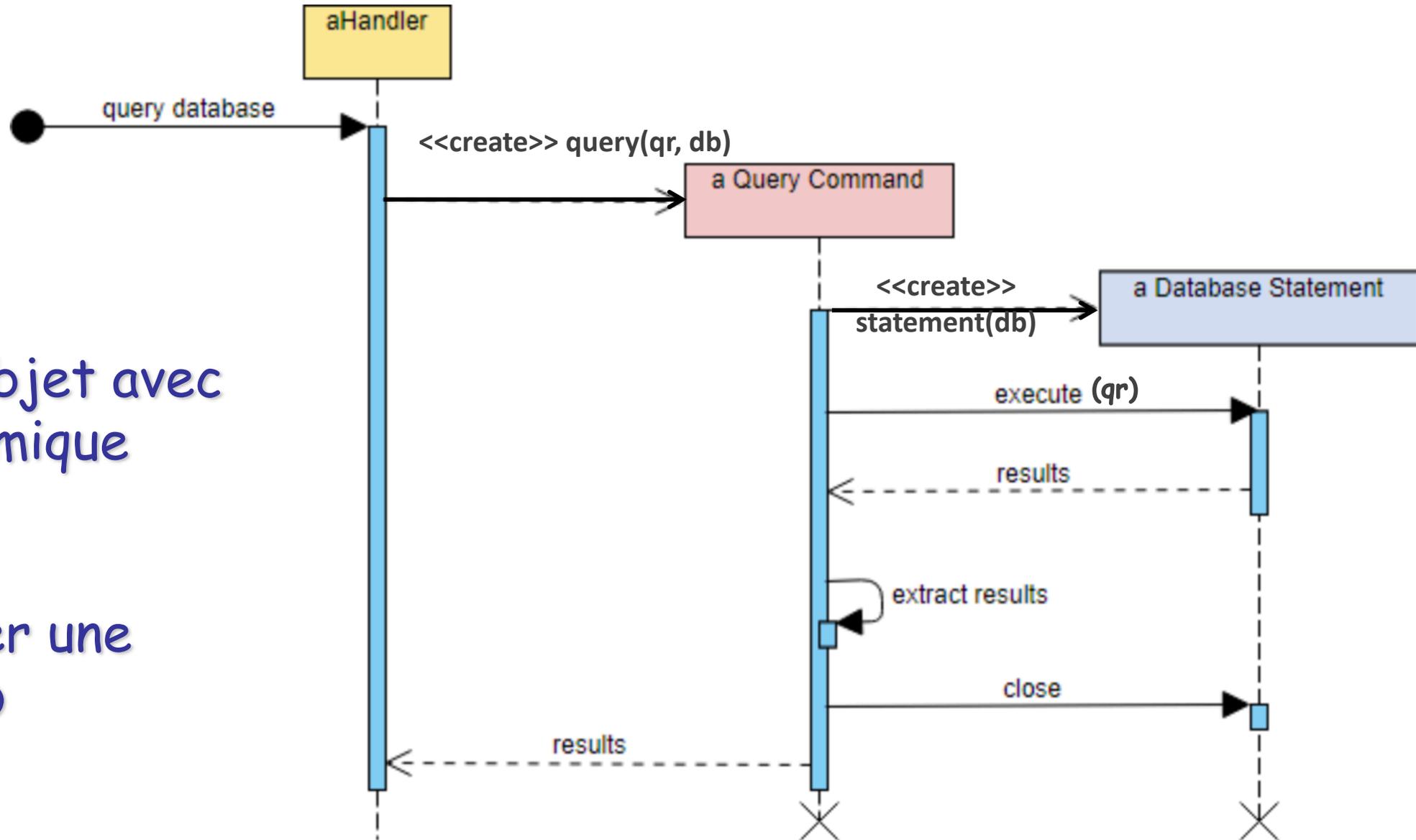
Code rentré correct.  
On fait appel au  
sequence diagramme  
du Use Case verifierSomme.

Pour alléger un diagramme, on  
peut faire référence à un autre  
DSQ

3 essais de code incorrect  
  
L'automate  
garde la carte du client  
( pas de signal rendreCart  
et se remet en attente  
d'un autre client

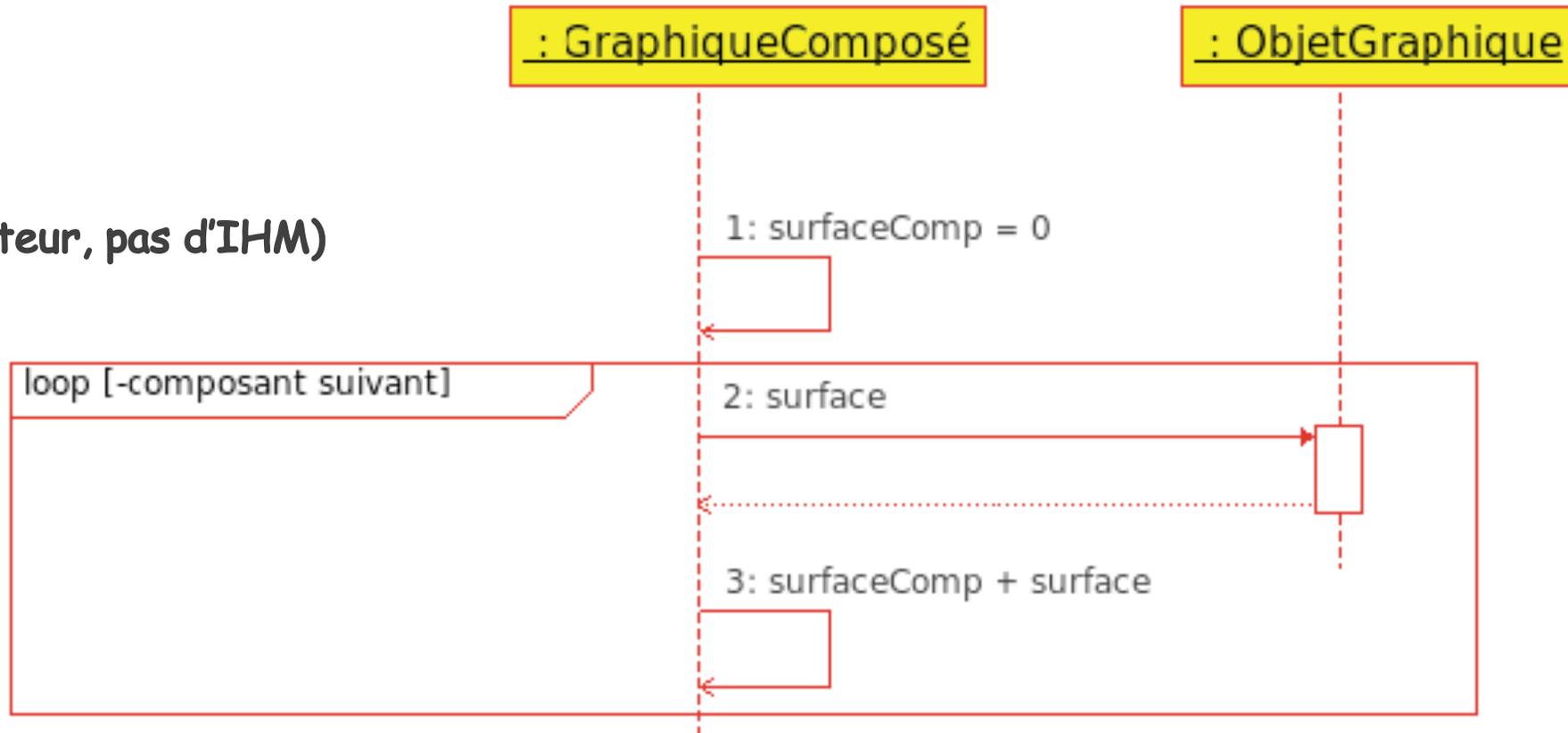
Ex. de DSQ objet avec  
création dynamique  
d'objets

SCN Effectuer une  
requête en BD



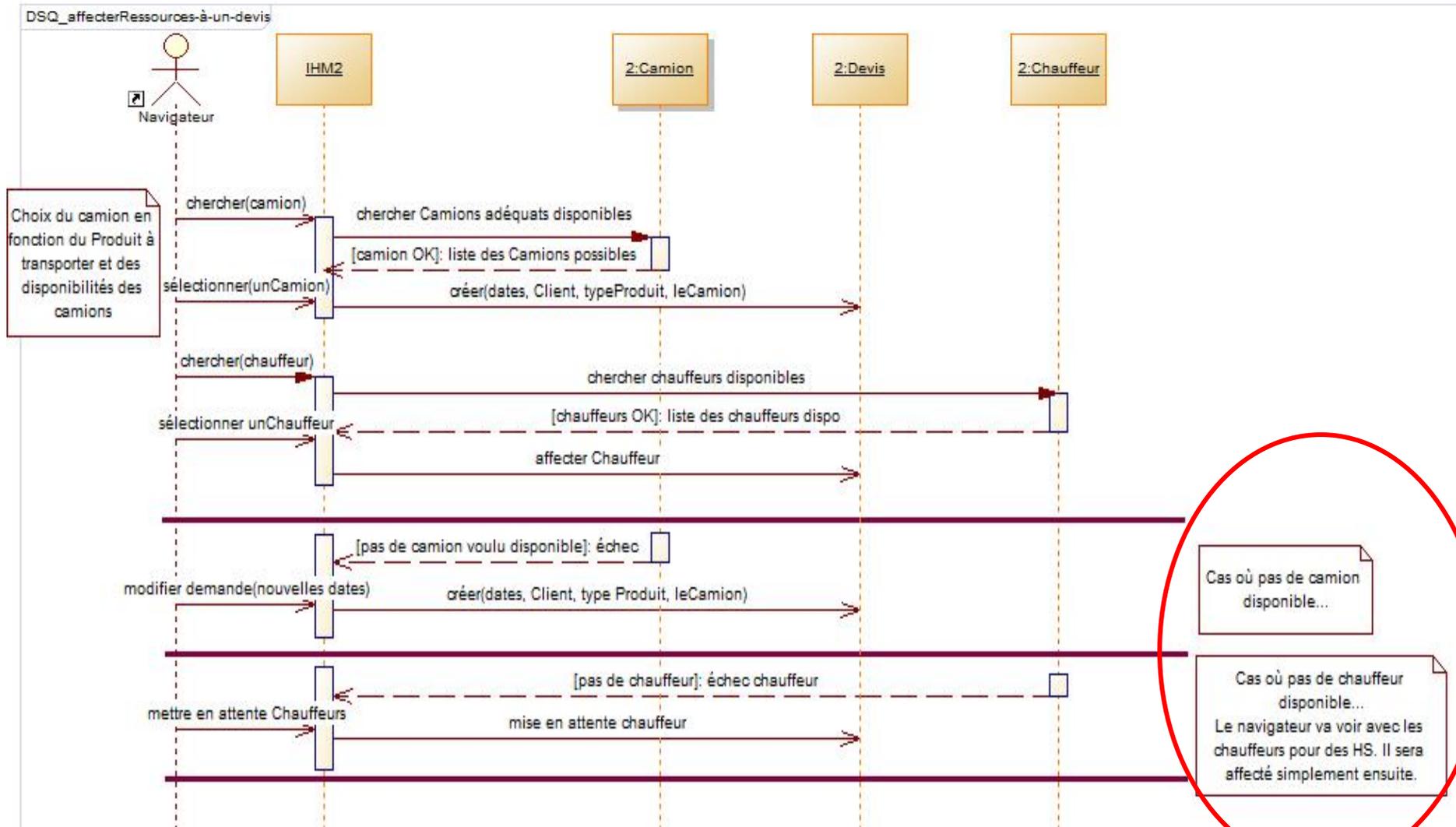
# Ex. DSQ Objet en Conception : détail d'une méthode

(pas d'acteur, pas d'IHM)



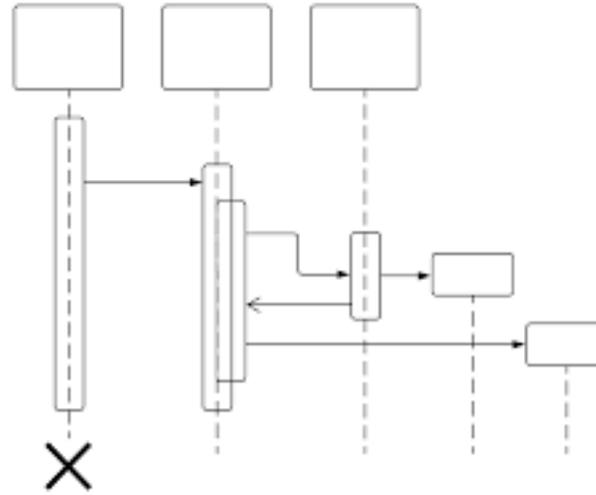
**Illustration du calcul de surfaces pour un objet composé d'autres objets**

# DSQ : représenter les scénarios d'exception



# Rappel : scénario

- Un ***cas d'utilisation*** regroupe une famille de scénarios
- Un scénario = une **séquence d'interactions** entre le système et ses acteurs
- Typologie :
  - Scénario **nominal** (1) ou scénario **d'exception** (souvent :  $n$ )
  -
- Le **diagramme d'activités** suffit généralement en analyse
  - Vision externe
- Le **diagramme de séquences** fournit un outil efficace pour approfondir les scénarios
  - Montrer les classes sollicitées (vision interne)



DSQ système et Objets

# ILLUSTRATIONS DIAGRAMMES DE SÉQUENCES

# DSQ système : Mécanisme RPC

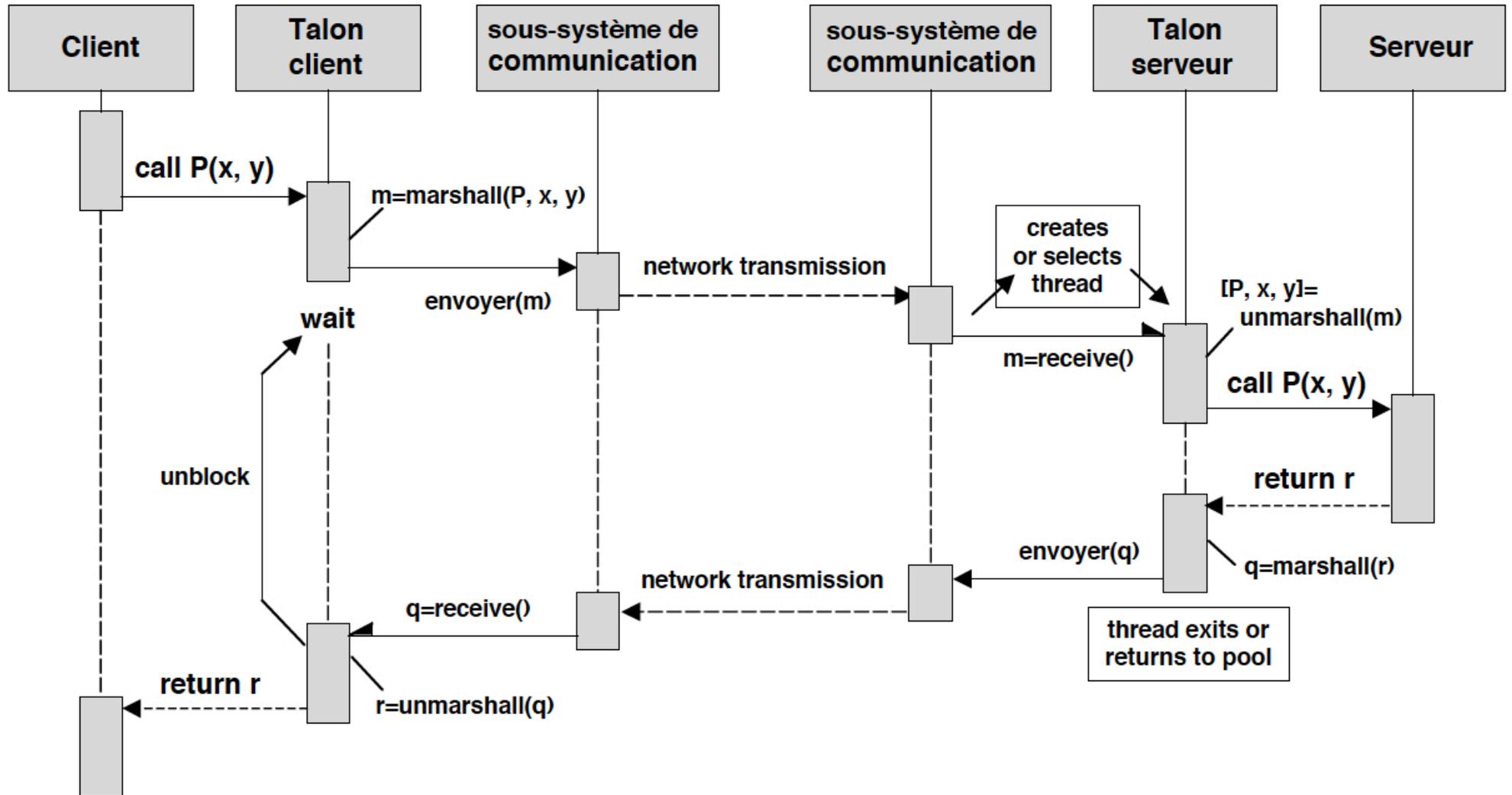


Figure 1.9 – Flot d'exécution dans un appel de procédure à distance

*RPC : détail*

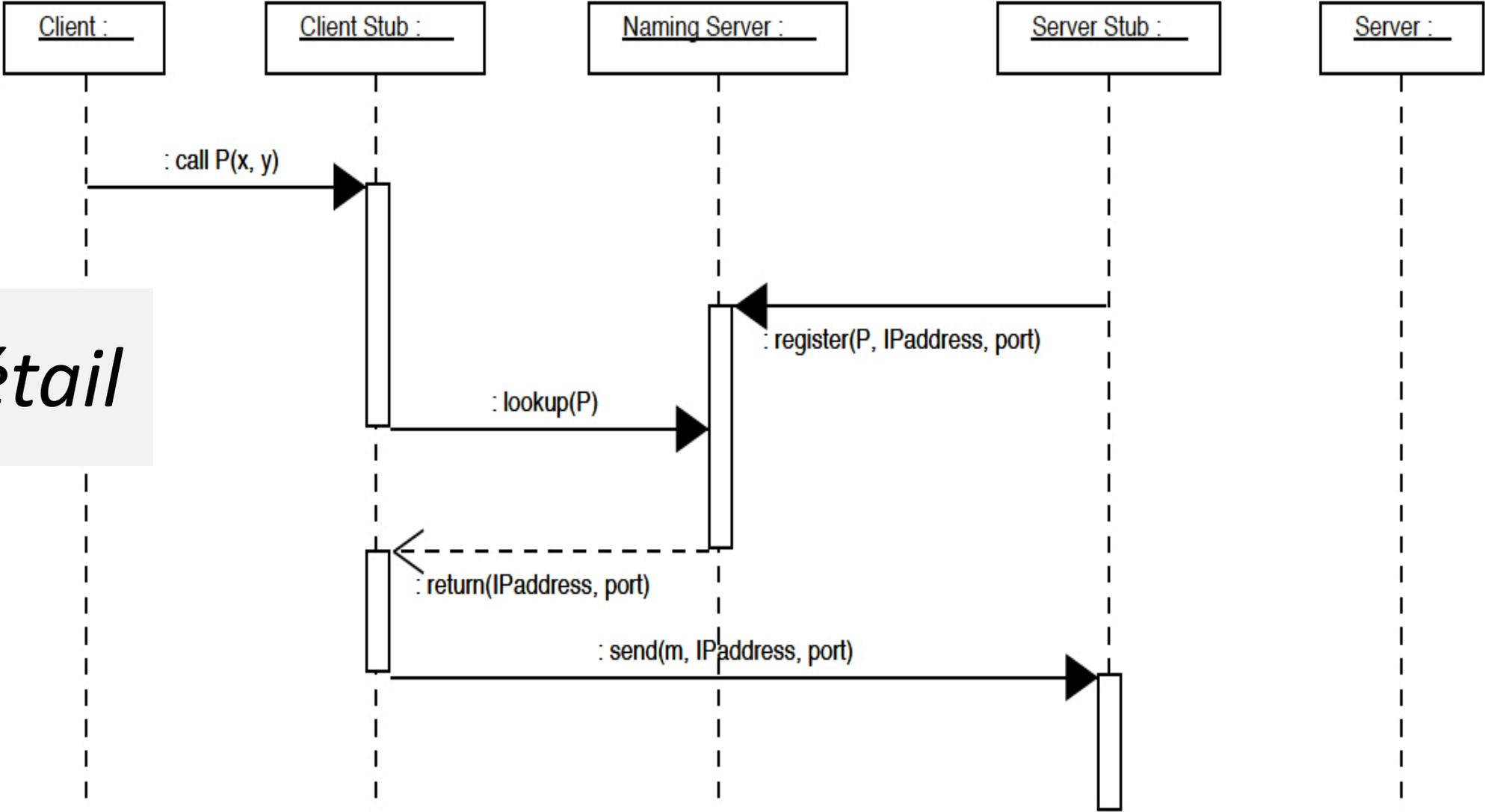
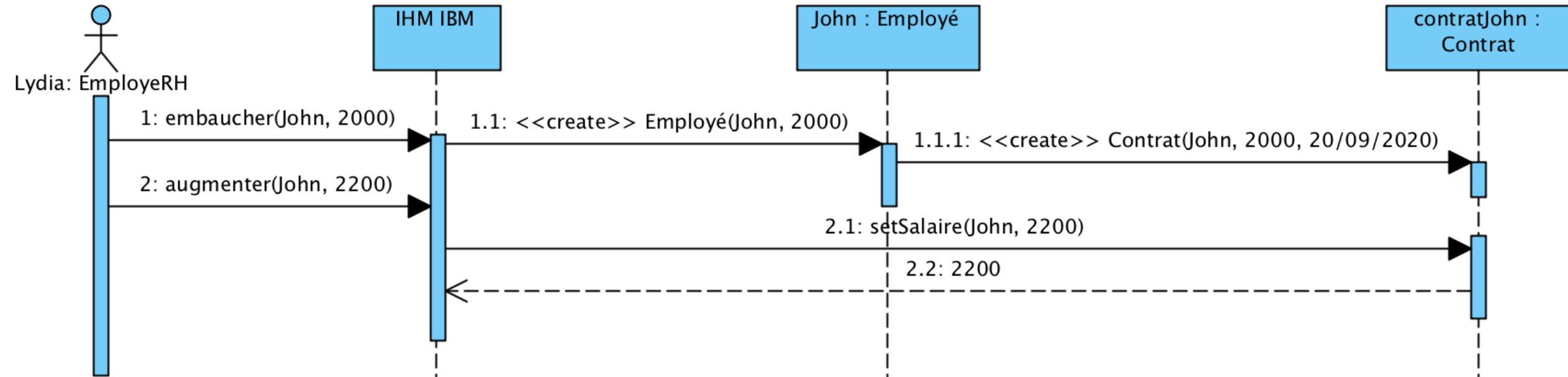


Figure 1.10 – Localisation du serveur dans l'appel de procédure à distance.

# DSQ objets pour 2 scénarios : **embaucher()** et **augmenter()**

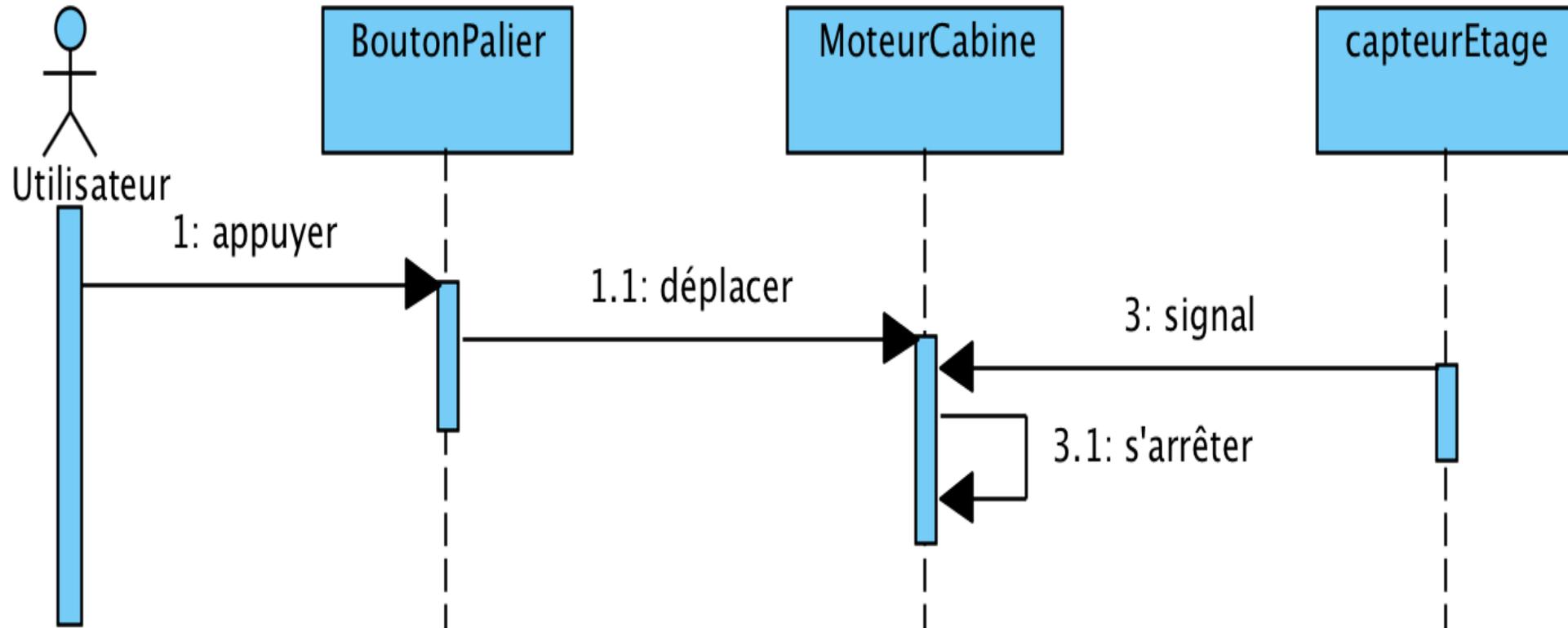
On voit qu'on ne peut augmenter un salarié que s'il a déjà un contrat (!...) et que **augmenter()** sera dans Contrat et pas Employé



# DSQ objets du système de pilotage d'un ascenseur

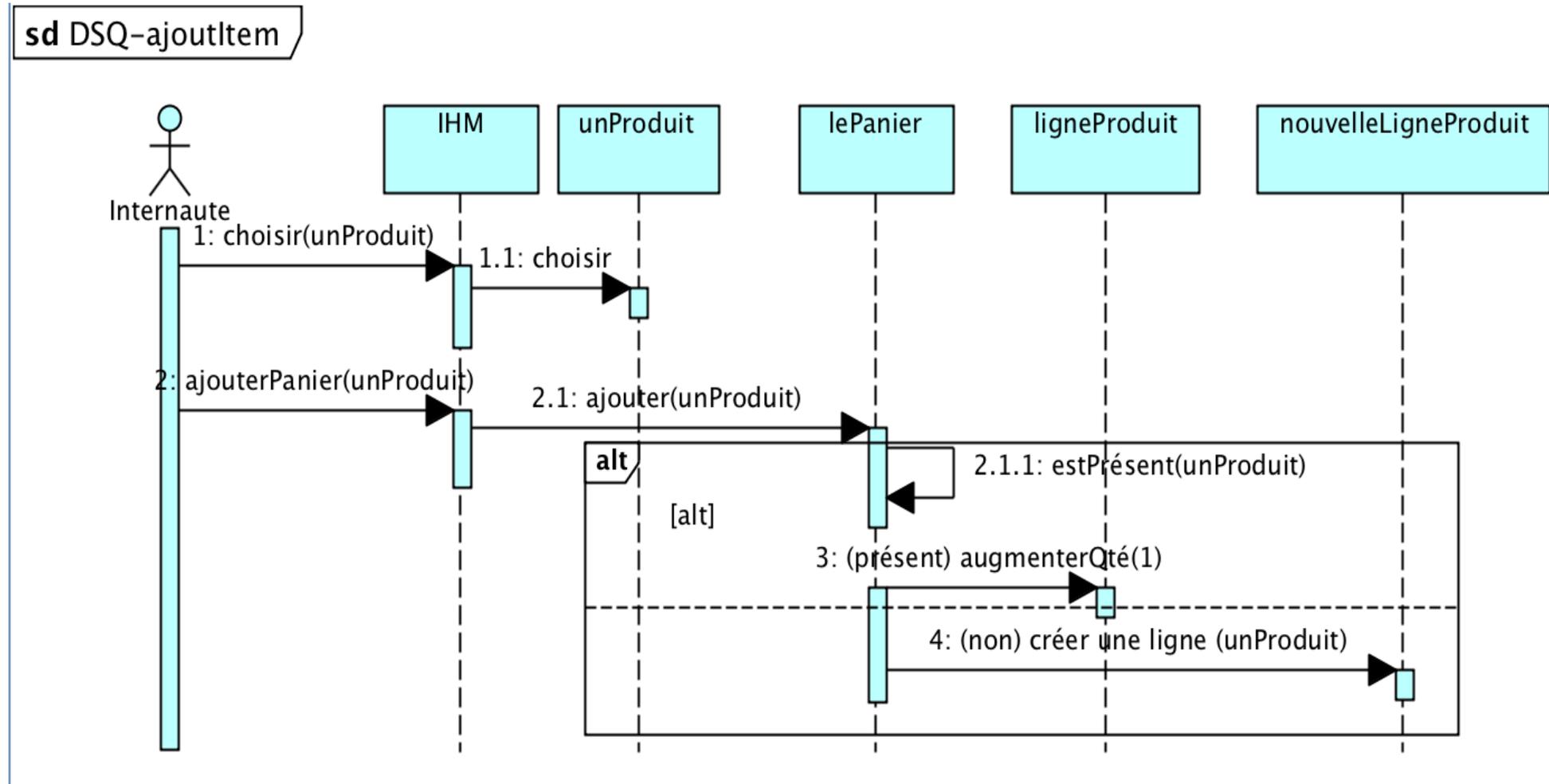
## SCN : appel depuis un palier

sd cabine-ascenseur

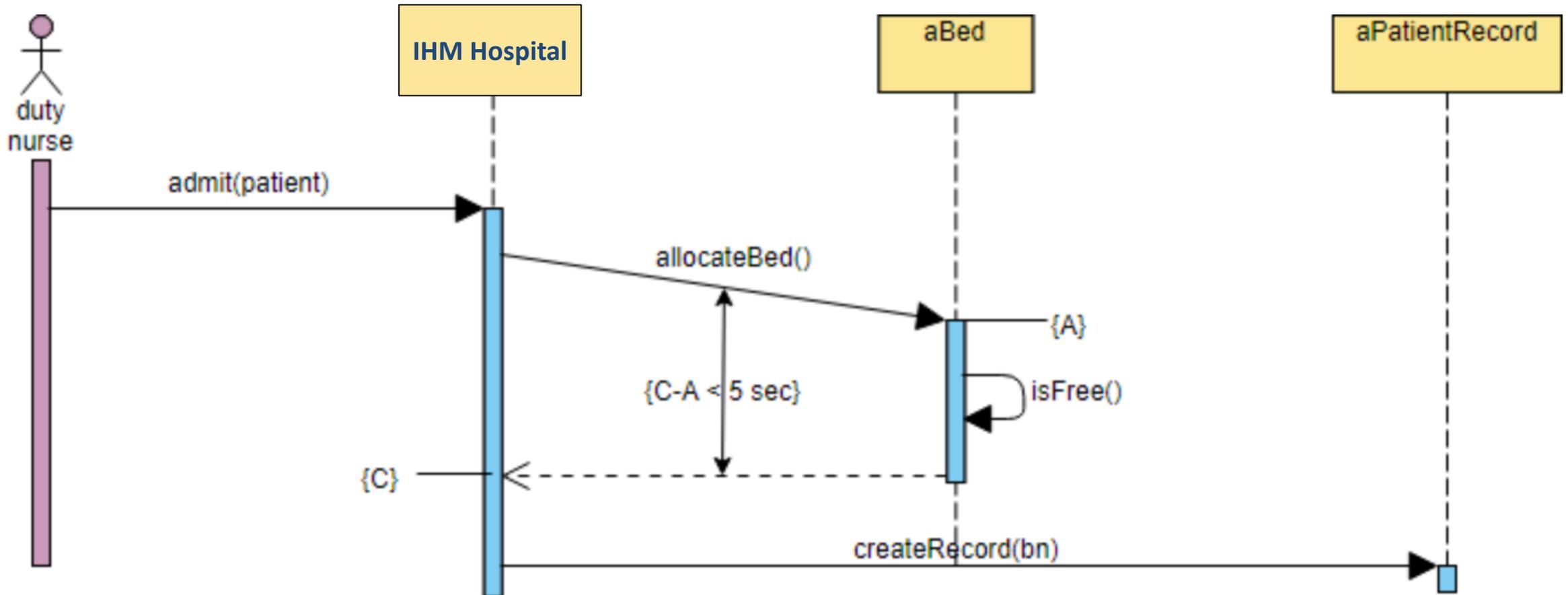


# DSQ objets : site eCommerce

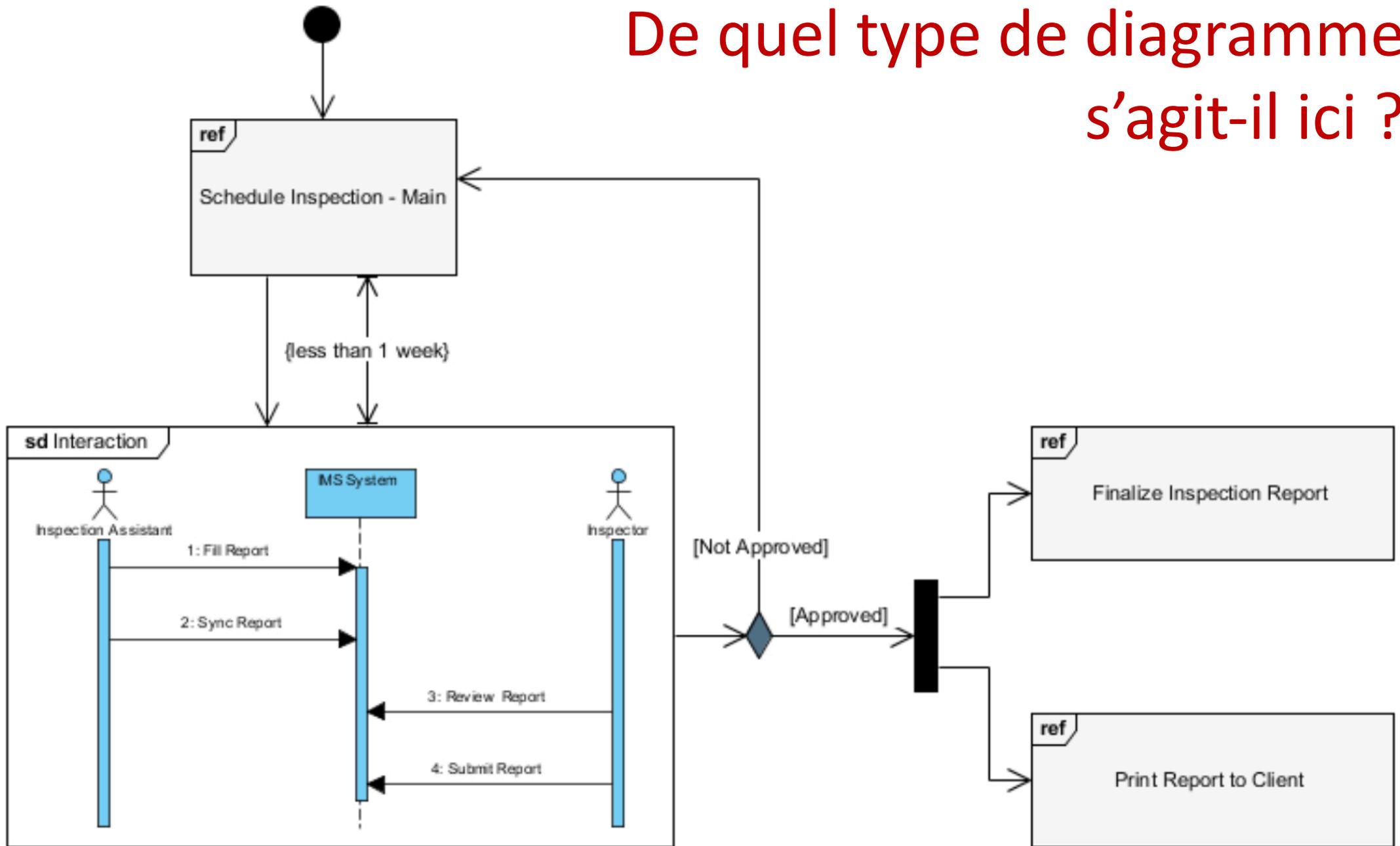
## SCN « Ajout d'un produit au panier »



# DSQ Objet : admission d'un patient



De quel type de diagramme s'agit-il ici ?



# DSQ : Bonnes pratiques

- Ce sont des diagrammes complexes :
  - à élaborer (avec un AGL)
  - à lire
- En Analyse, ils sont utilisés dans la découverte des besoins pour **identifier les méthodes** des classes
- En Conception, on ne les crée que :
  - lorsqu'on est face à une **logique complexe** qu'il faut creuser
  - Lorsqu'on veut **communiquer** sur des choix de conceptions complexes
- Si le SCN est simple, le DSQ objet n'aura aucune plus-value : on pourra **coder directement**

# Démarche de modélisation dynamique



# Démarche d'élaboration du modèle dynamique UML

## 1. Identifier les scénarios (Analyse)

Partir des cas d'utilisation : **un cas = une famille de scénarios**

ex.: Gérer les clients

Lister les scénarii intéressants; définir les enchaînements nominaux et alternatifs.

- Ex.: Créer un nouveau client (SCN alternatif : un client de même nom est déjà présent dans la BD → on fait quoi ?)
- Ex.: supprimer un Client (donner la raison et l'archiver; s'il a des encours : ?)

# Démarche Dynamique (suite)

## 2. Modéliser les principaux scénarios (Analyse détaillée)

avec des **Diagrammes d'activité** (vision utilisateur)

3. Pour les processus complexes, identifier les objets mis en œuvre à l'aide de **Diagrammes de Séquences objets**

## 4. Valider le modèle dynamique

On vérifie la **cohérence des éléments communs** (événements, opérations, attributs) aux différents diagrammes

5. **En Conception**, faire des **DSQ système** pour montrer les composants logiciels qui interviendront dans certains processus utilisateur

# Diagr. d'Activités vs. de Séquences

- Rappel : diagramme d'ACTIVITÉS (DIT)
  - Description d'un **processus**, peu de liens avec le monde Objet (à part les *nœuds d'objets* impactés par les activités)
- POUR MODÉLISER une OPÉRATION COMPLEXE ?
  - Un **diagramme d'activités** sera préféré quand il s'agit de décrire une méthode pour laquelle on a **un petit nombre de classes** et de **nombreuses alternatives**.
  - Un **diagramme de séquences** quand la méthode fait intervenir **beaucoup de classes** avec **peu d'alternatives**.



# Diagramme de séquences : je retiens

- 2 types : **DSQ système (vision externe)** et **DSQ d'objets (vision interne)**
- La notion de **scénario**
  - Typologie : SCN nominaux, d'erreurs
  - Famille de scénarios
- Un **message** = appel d'une méthode de l'objet en général
- **DSQ d'objets** : on en fait autant qu'il y a de processus *importants* à détailler
- Un **DSQ d'objets** démarre tjrs par un **acteur et un objet IHM**
- **DSQ système** : utiles pour construire le dialogue H/M ou pour montrer les relations entre les composants physiques ou logiciels