

# UML

## Unified Modeling Language



[www.uml.org](http://www.uml.org)



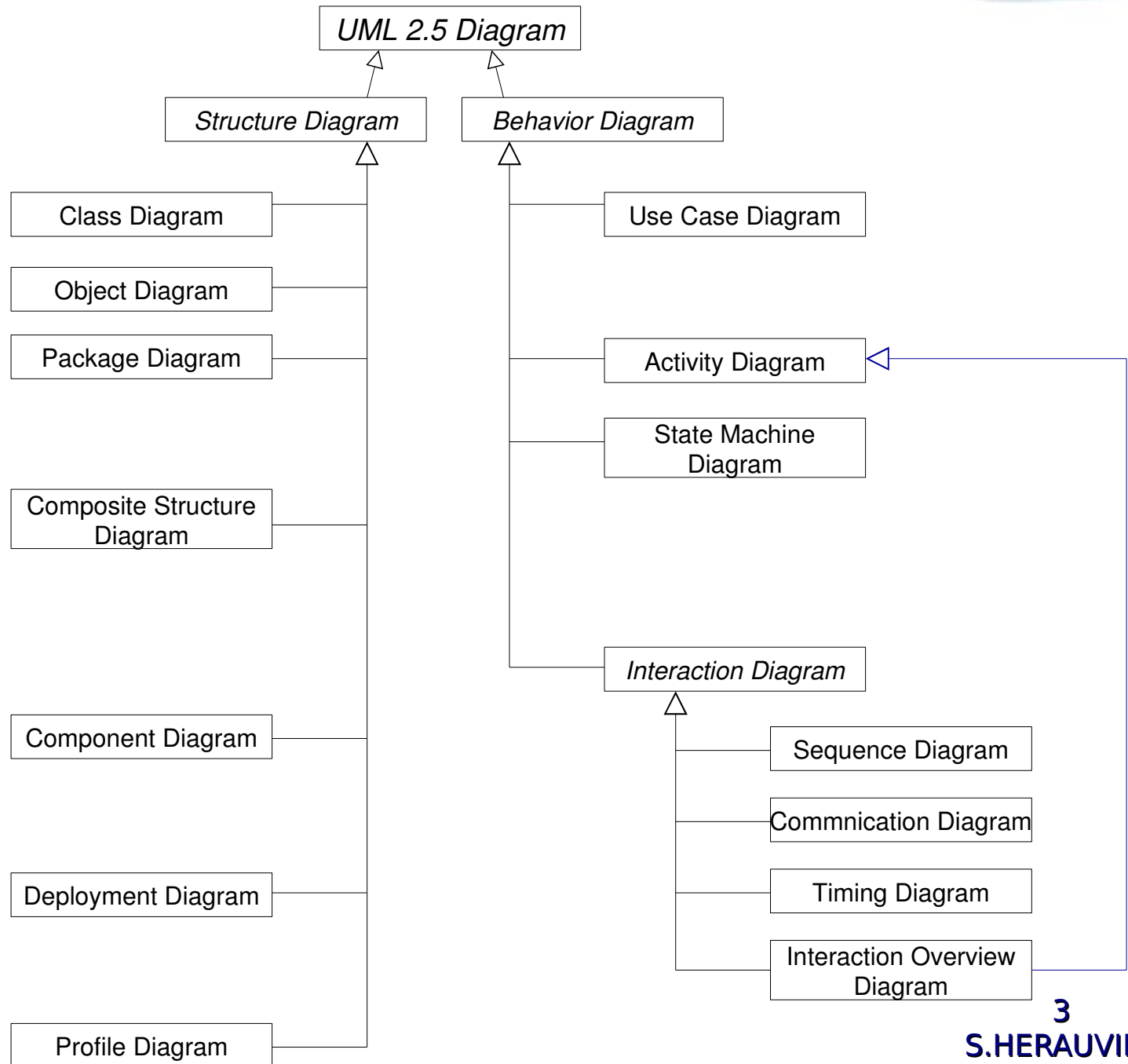
[www.omg.org](http://www.omg.org)

# UML

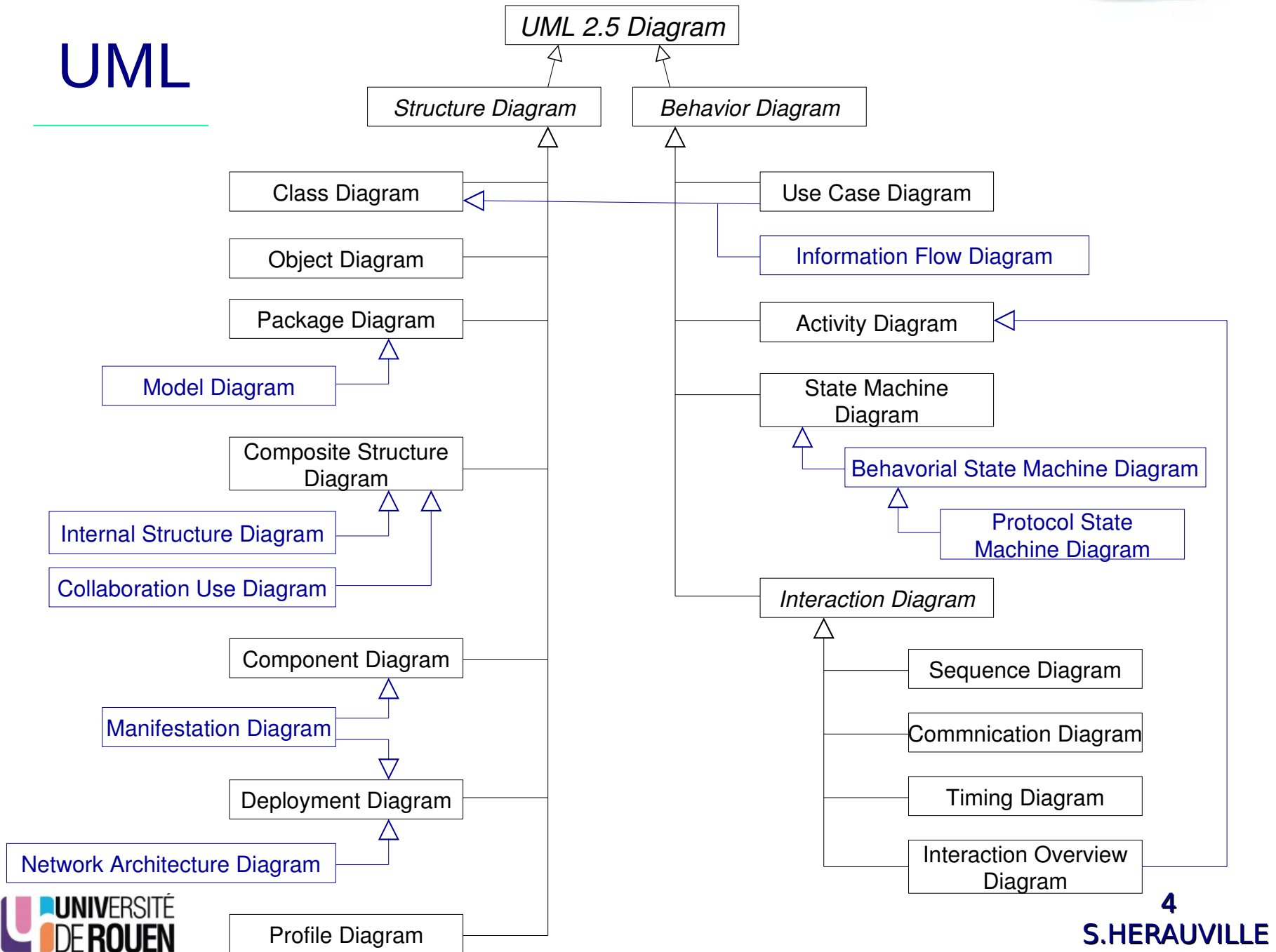
---

## DIAGRAMMES UML

# UML



# UML



# UML

---

# DIAGRAMME d'OBJETS

# Diagramme d'objets

---

## Objectif

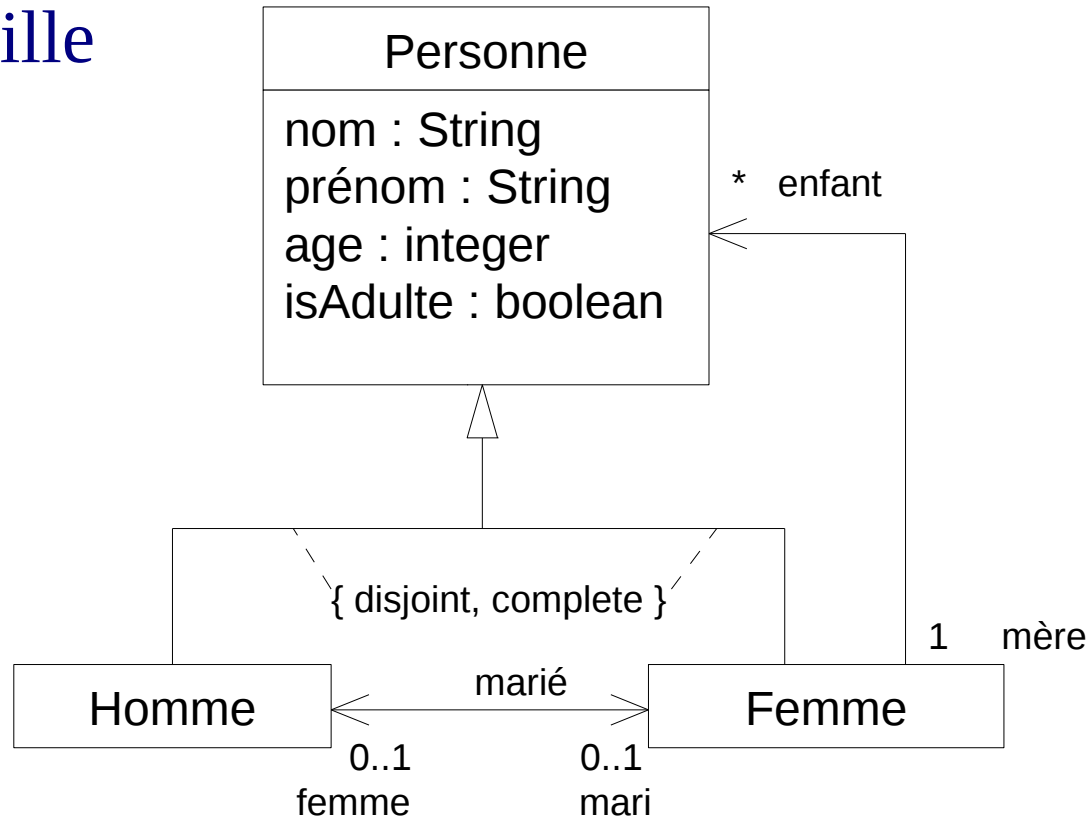
- Vue des objets d'un système à un moment donné
- Représentation des liens entre objets  
(en complément du diagramme de classe)

! spécification d'instance  $\neq$  instance réelle

# Diagramme d'objets

## Diagramme de classe : Structure d'une entité

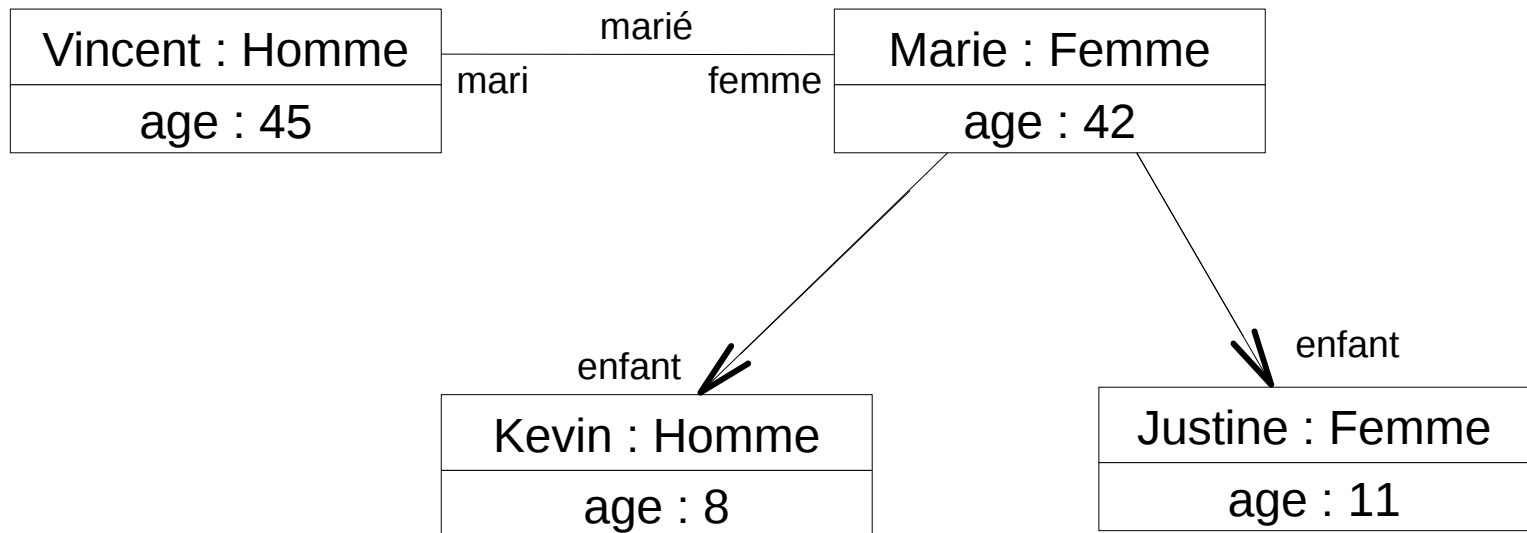
→ famille



# Diagramme d'objets

## Diagramme d'objet : Exemple d'instance d'une entité

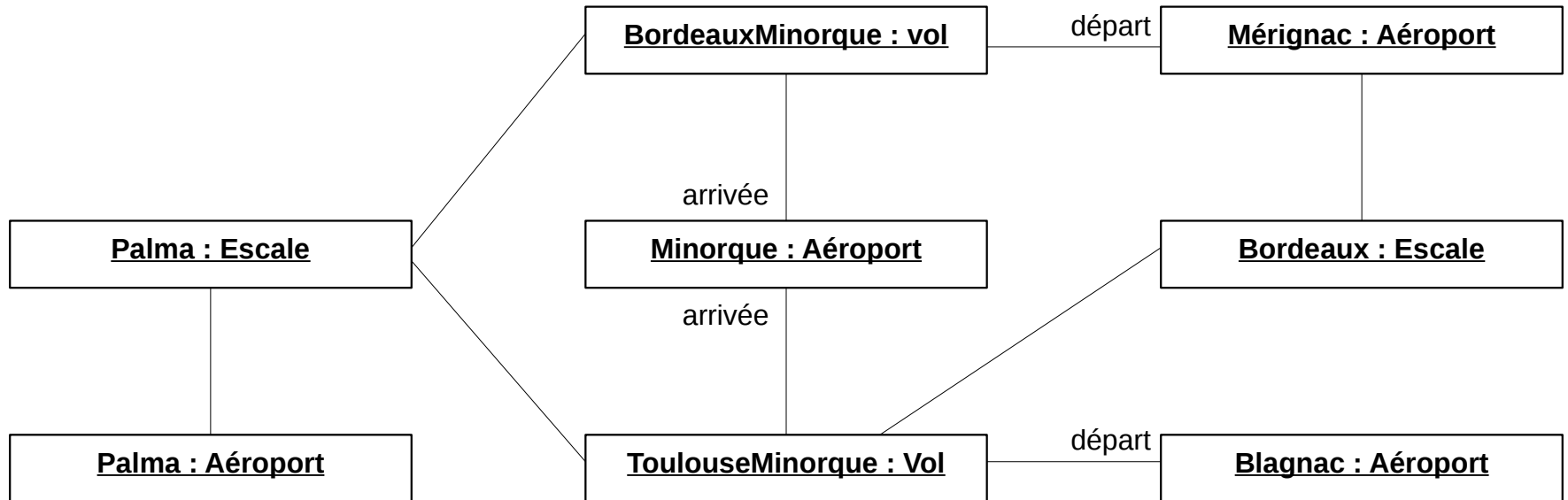
→ famille





# Diagramme d'objets

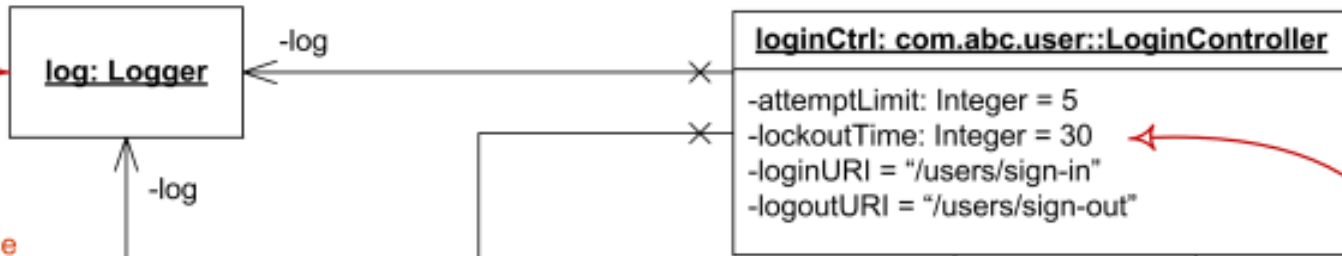
## Diagramme d'objet : Compagne aérienne



# Diagramme d'objets

© uml-diagrams.org

instance specification  
log of class Logger



link, non-navigable  
backward

anonymous instance  
of UserManager interface

link, navigable  
forward

5 {ordered,  
unique}

collection of 5 unique  
anonymous instances  
of unknown classifier

slots with value  
specification

named instance with  
value specification

slots with value  
specification

link

## **DIAGRAMME de COMPOSANTS**

# Diagramme de composants

---

Composant :

- Sous-ensemble du système plus ou moins complexe
- Élément autonome
- Contenu accessible via interfaces  
(généralement associée à des ports)

Composant logique    *Business, Process, ...*

Composant physique    *Corba, EJB, COM+, .NET, WSDL, ...*

*Le diagramme de package est une extension des diagrammes de composants*

# Diagramme de composants

---

Découpage en composants ?

- Solution technique → Evolutivité / Modularité
- Solution commerciale

Usage :

- CDB : Component-Based Development
- SOA : Service-Oriented Architecture

! Pléthore de composants → DLL hell

# Diagramme de composants

---

Stéréotypes applicables aux composants :

<<subsystem>>	Découpage hiérarchique en sous-ensemble
<<process>>	Basé sur les transactions
<<service>>	Composant fonctionnel
<<specification>>	Définition d'un domaine (interface)
<<realization>>	Domaine + implémentation physique
<<implement>>	Implémentation d'un domaine existant

# Diagramme de composants

Composant

<<component>>  
Order

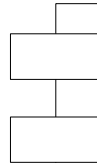
UML 1.4

<<component>>  
Order



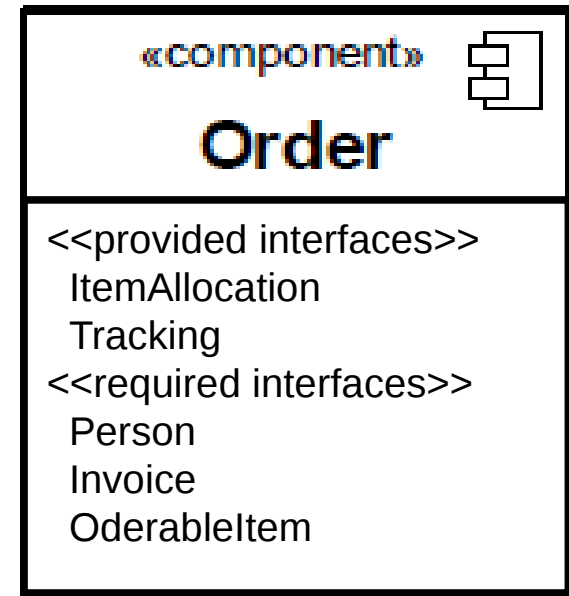
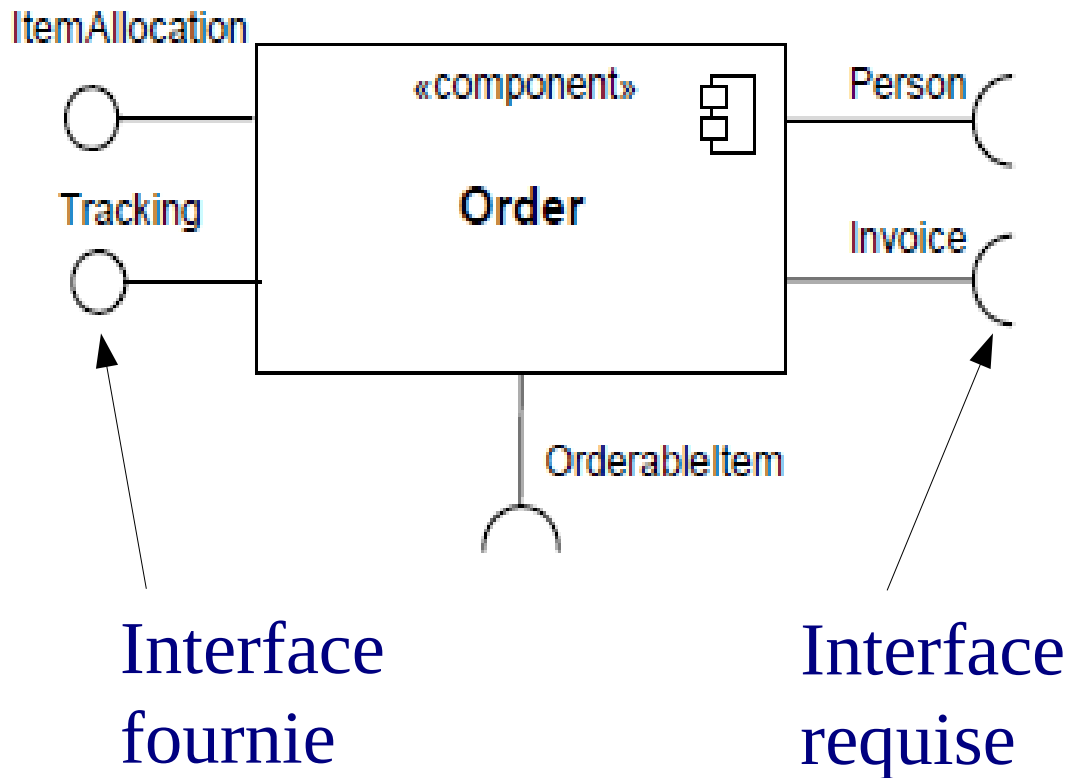
UML 1.4

<<component>>  
Order



# Diagramme de composants

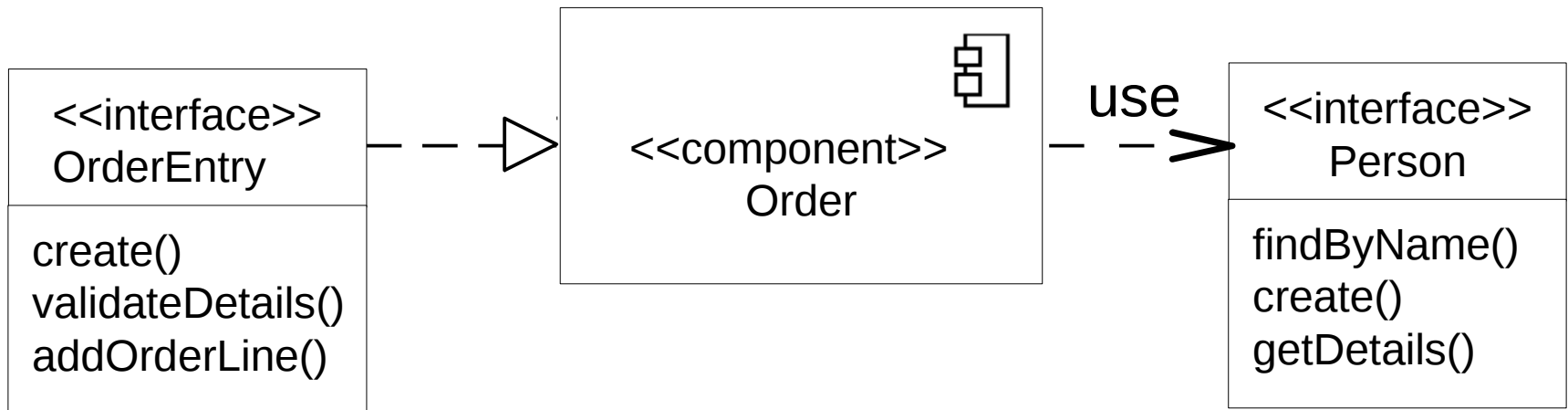
## Composant avec interface





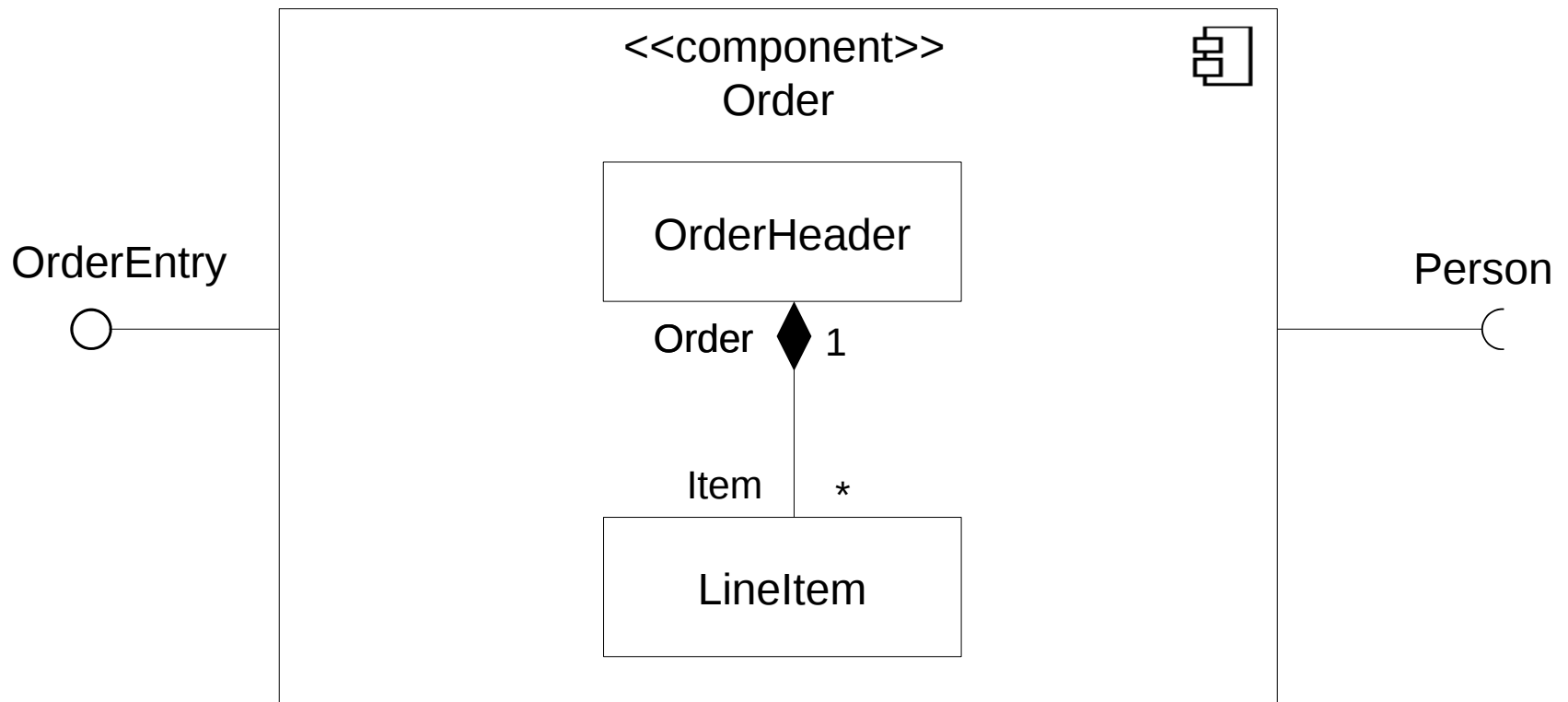
# Diagramme de composants

## Réalisation



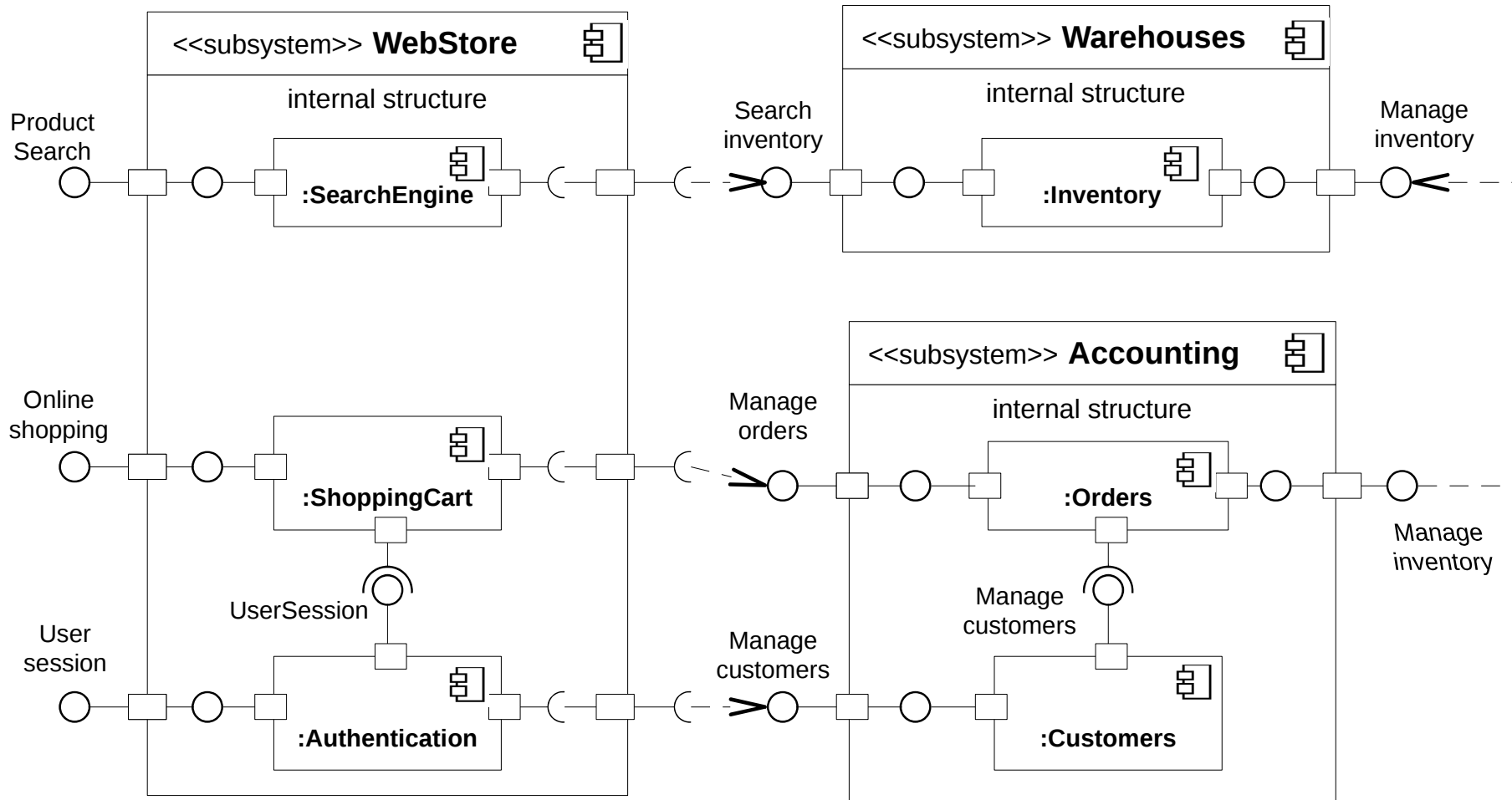
# Diagramme de composants

## Structure d'un composant



# Diagramme de composants

## Composant avec ports



# UML

---

## DIAGRAMME de PACKAGE

# Diagramme de package

---

## Objectif

Représentation d'un groupement de classe

- 1 classe appartient à 1 package
- 1 package peut-être défini dans un autre package

=> Structure hiérarchique

# Diagramme de package

---

## Éléments intégrables à un package

Format

Classe

Use case

Composant

Package

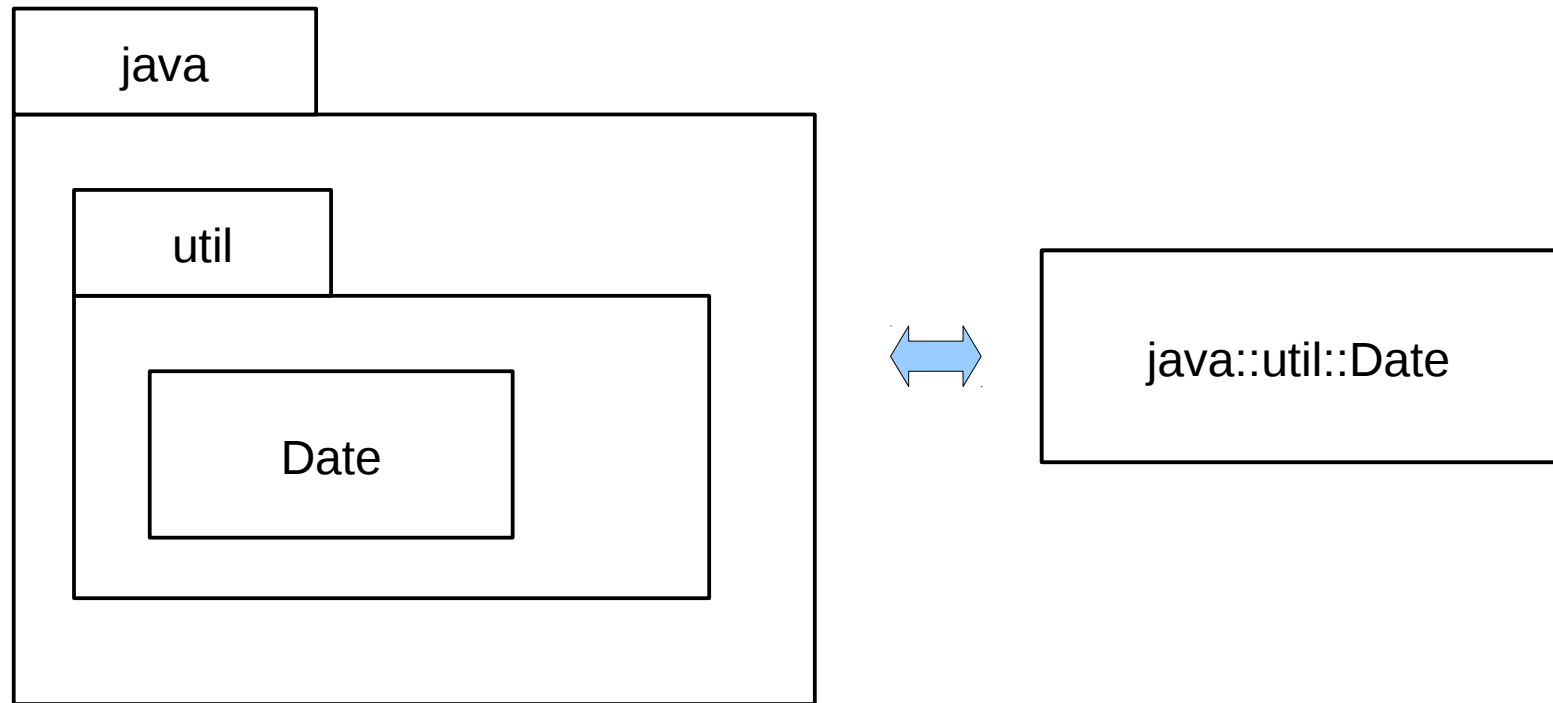
Contrainte

Dépendance

Evènement

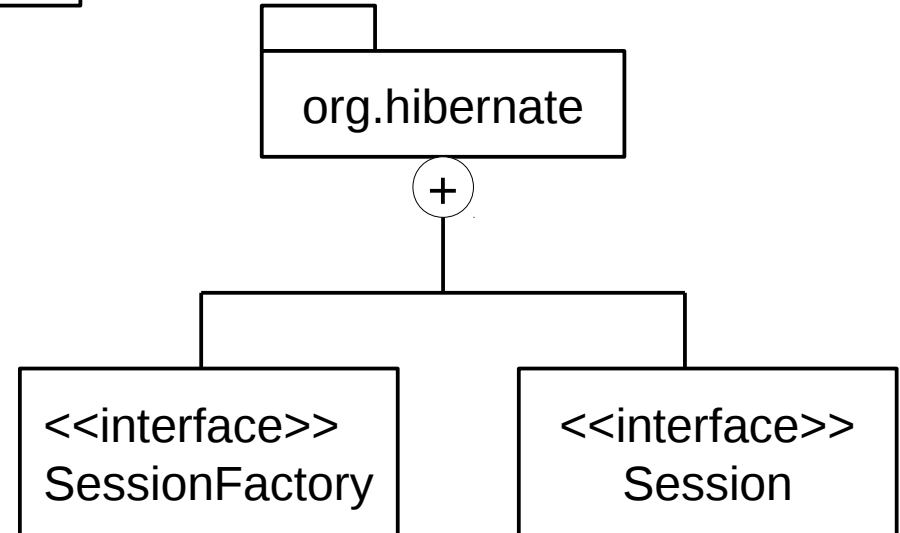
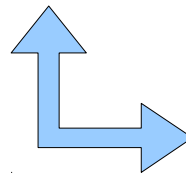
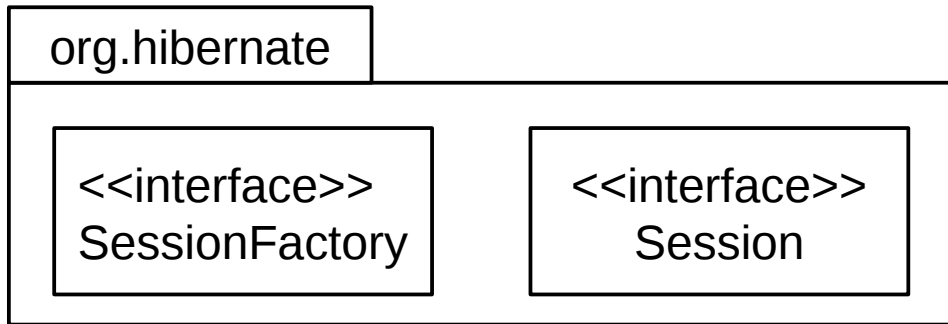
# Diagramme de package

**Package => Espace de noms**



# Diagramme de package

## Membres d'un package



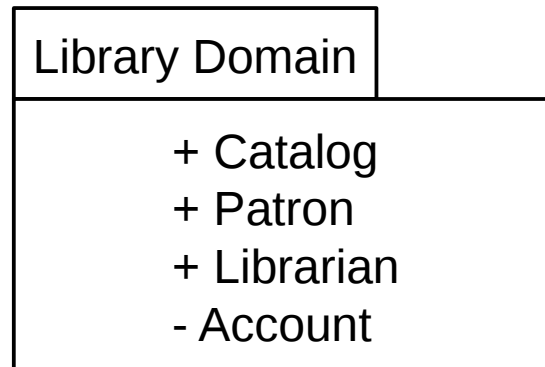


# Diagramme de package

---

## Visibilité

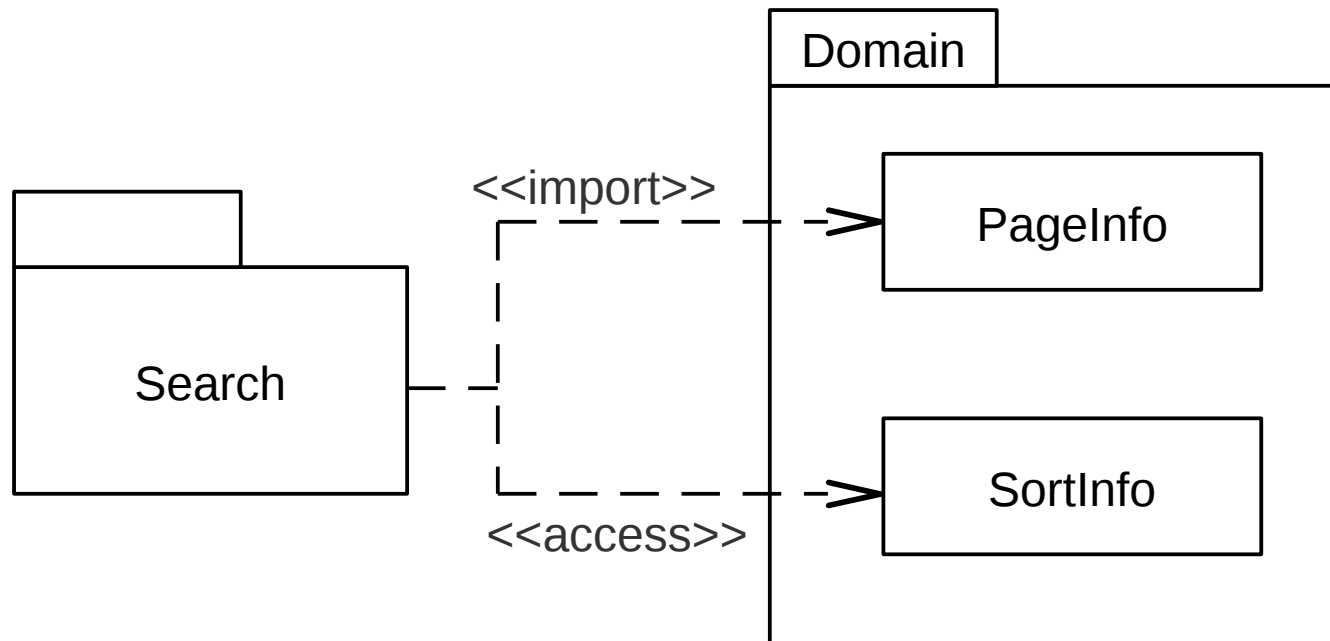
Publique ou Privée (autres choix non autorisés)



*Remarque : voir pattern Classes façades*

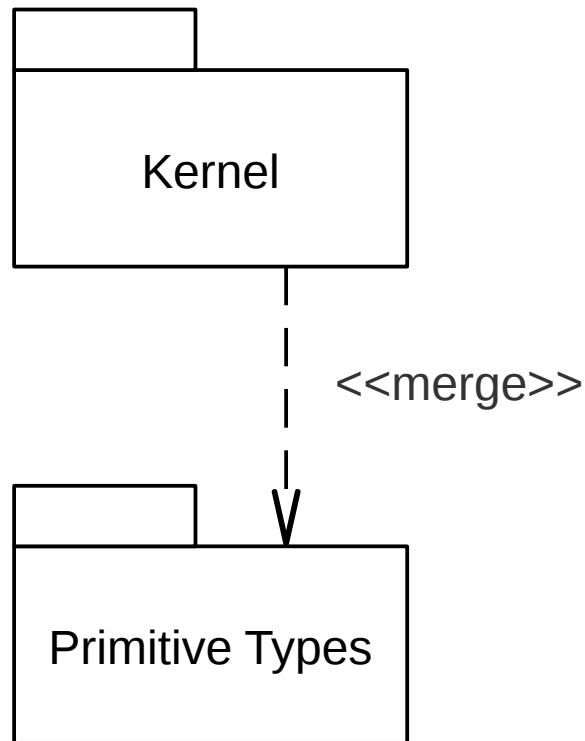
# Diagramme de package

## Import

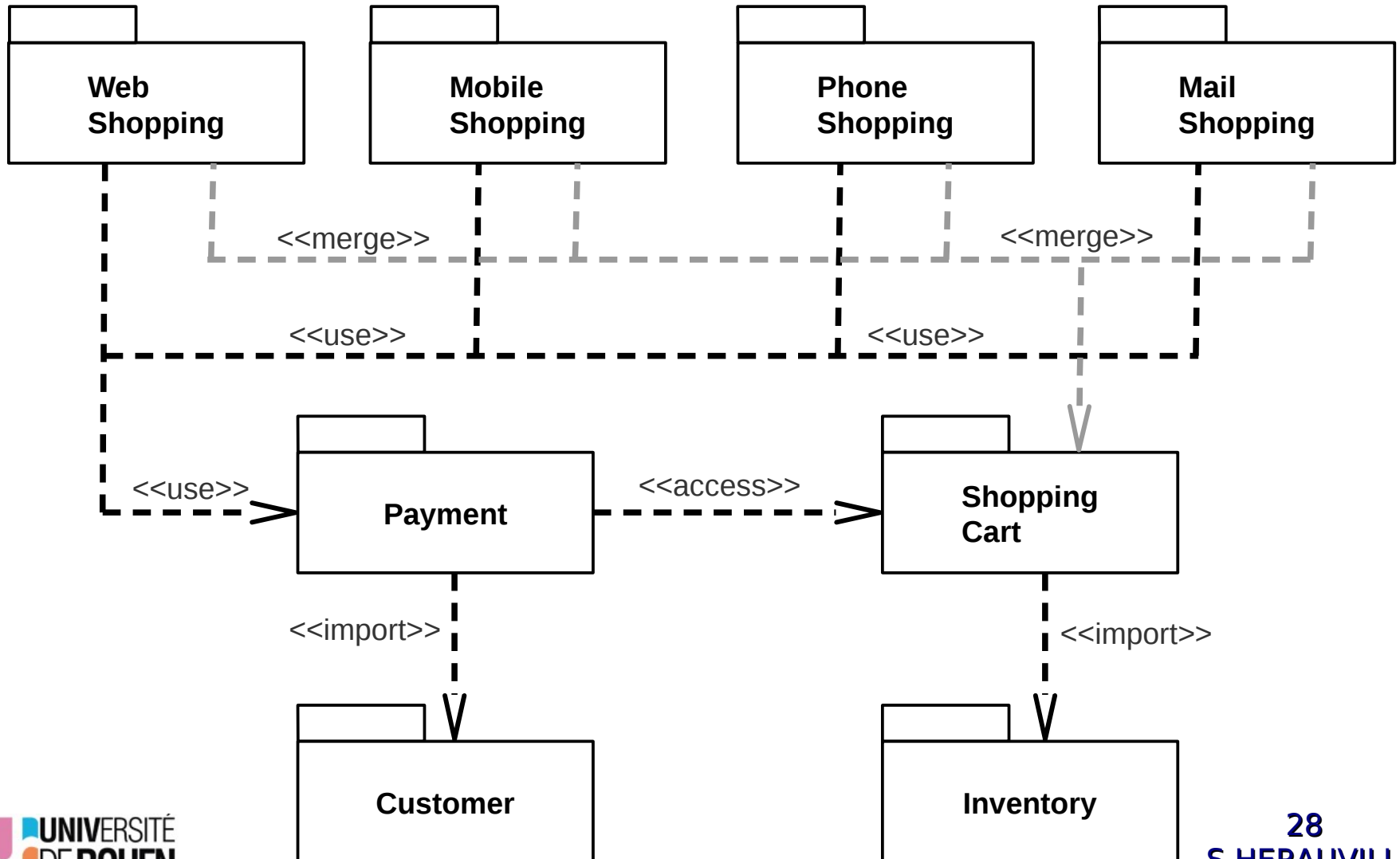


# Diagramme de package

## Merge

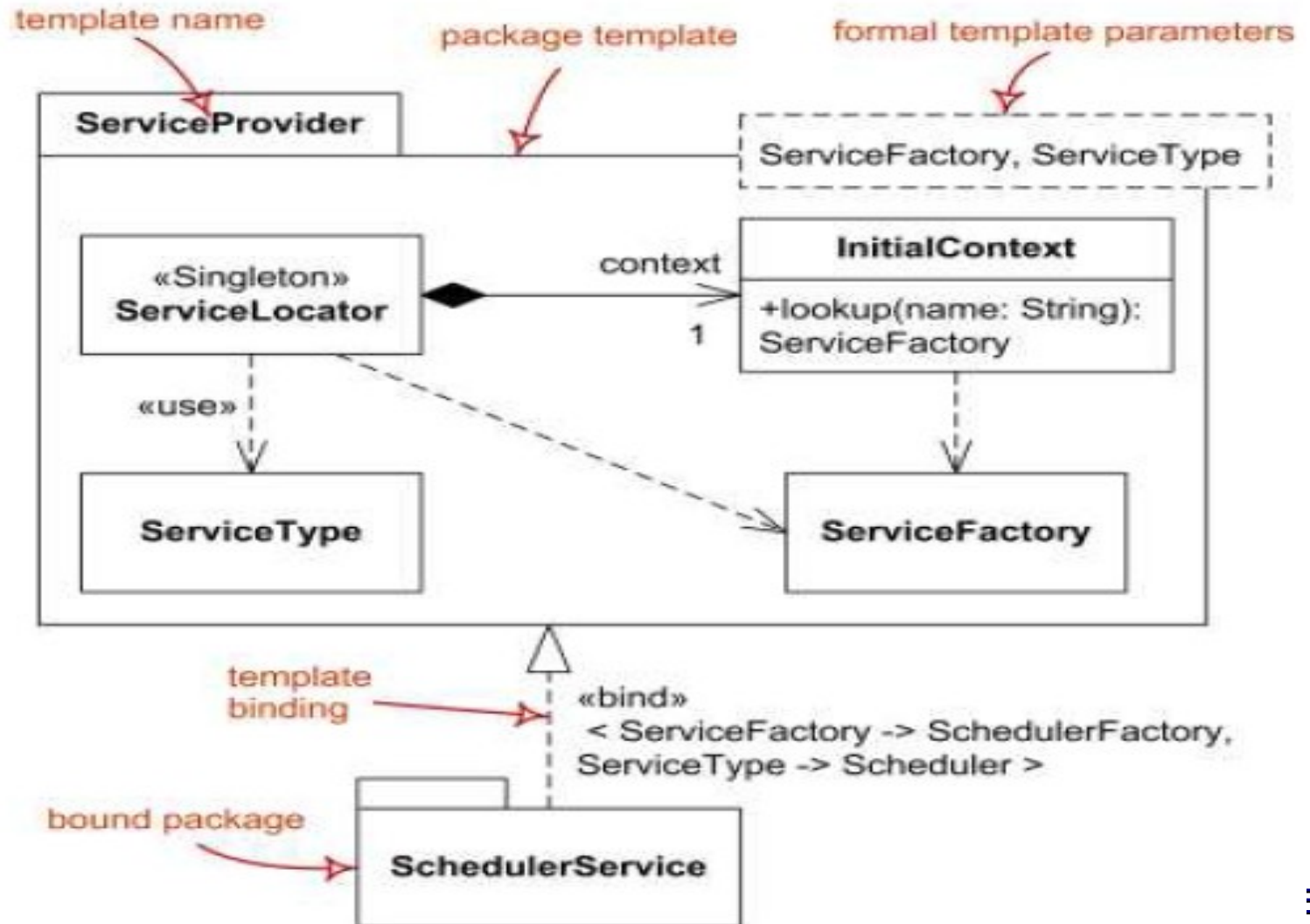


# Diagramme de package



# Diagramme de package

## Package template



# UML

---

## DIAGRAMME de DEPLOIEMENT

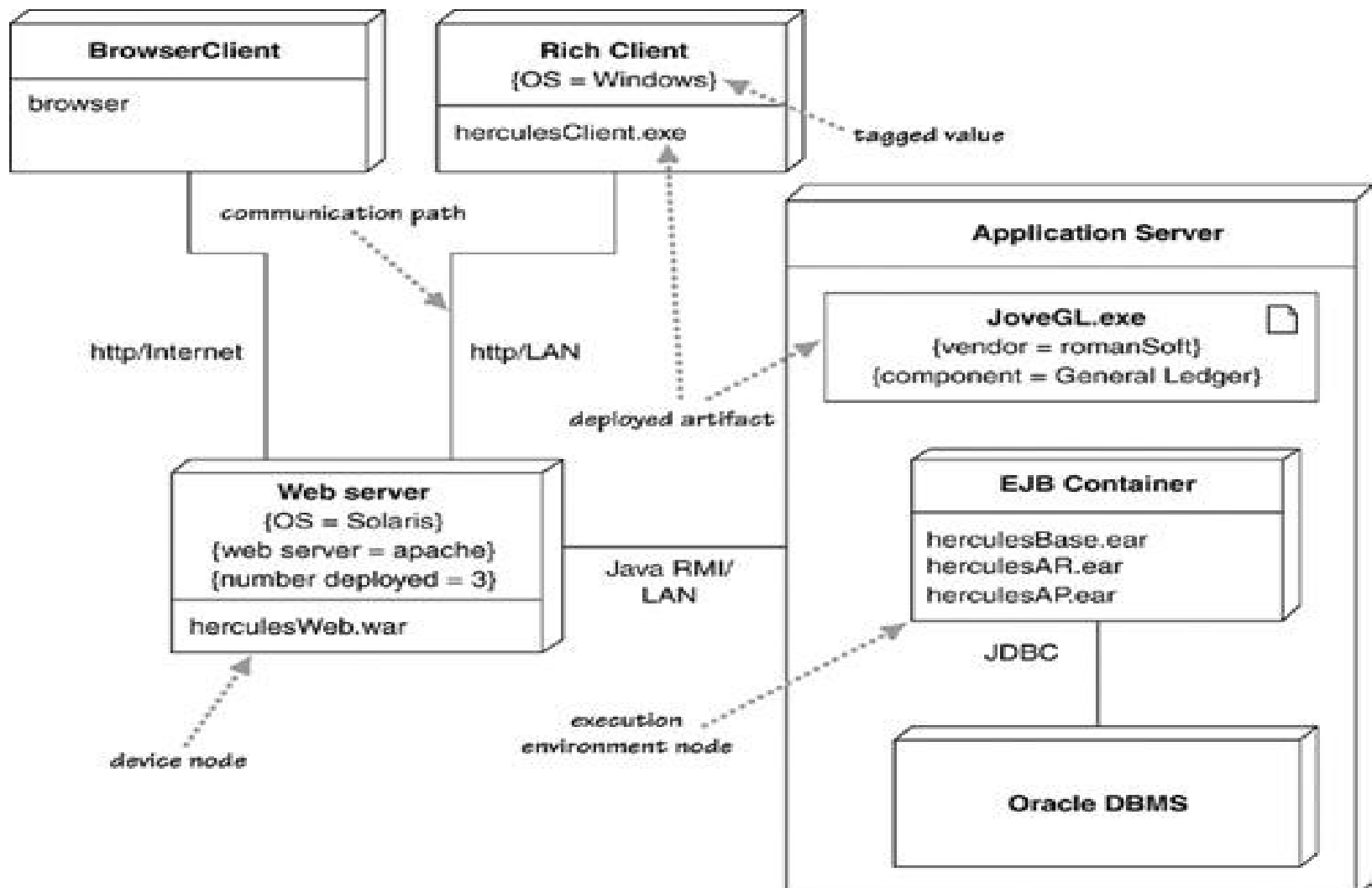
# Diagramme d'objets

---

## Objectif

- Représente le déploiement des parties du systèmes
- Noeud                      Environnement logiciel  
                                  ou      Espace physique

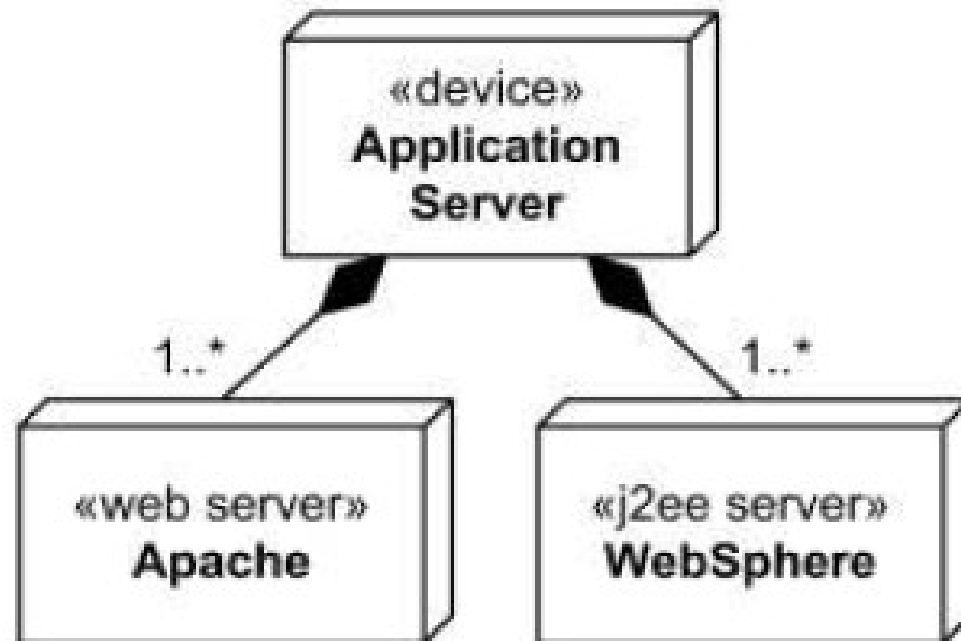
# Diagramme de déploiement





# Diagramme de déploiement

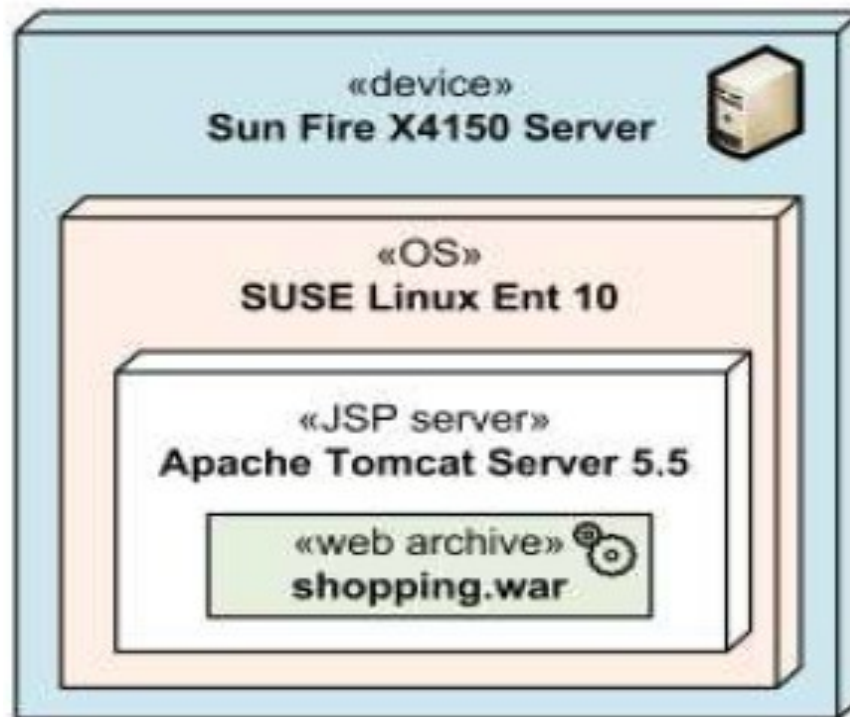
## Noeud hiérarchique



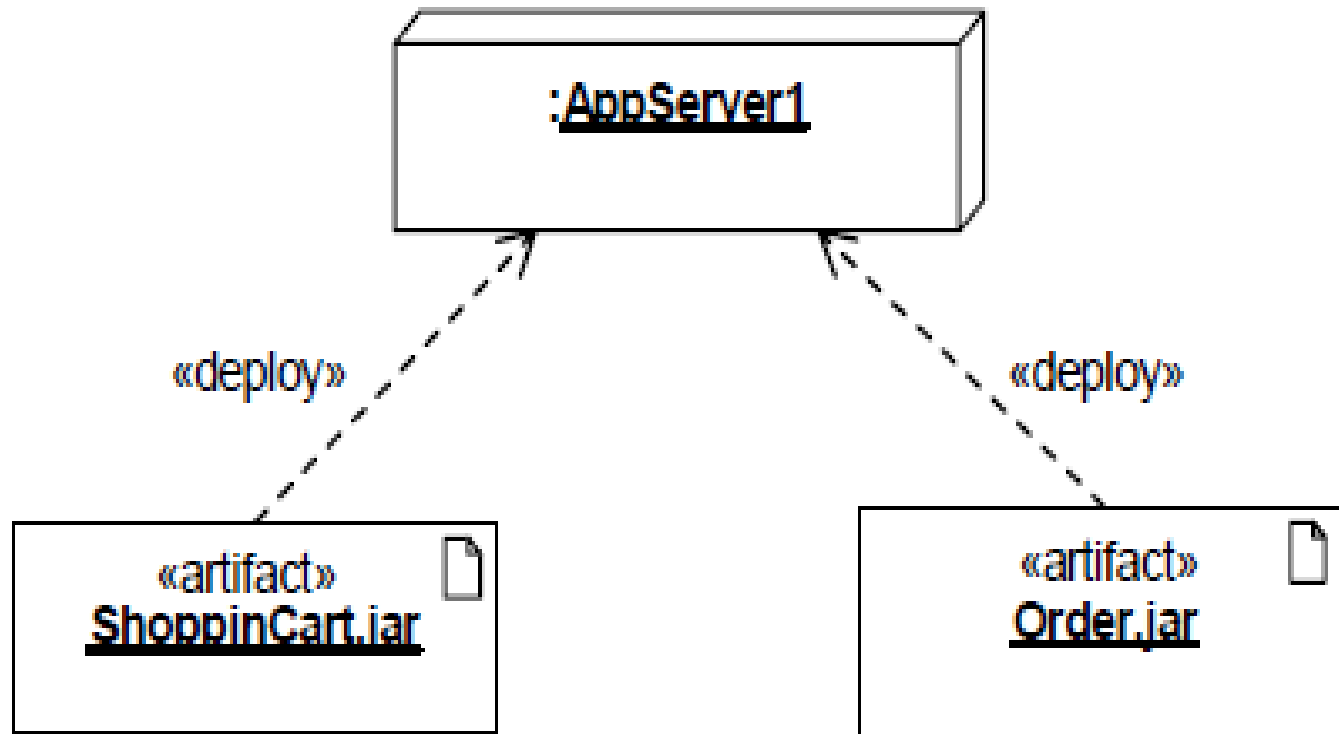
# Diagramme de déploiement

---

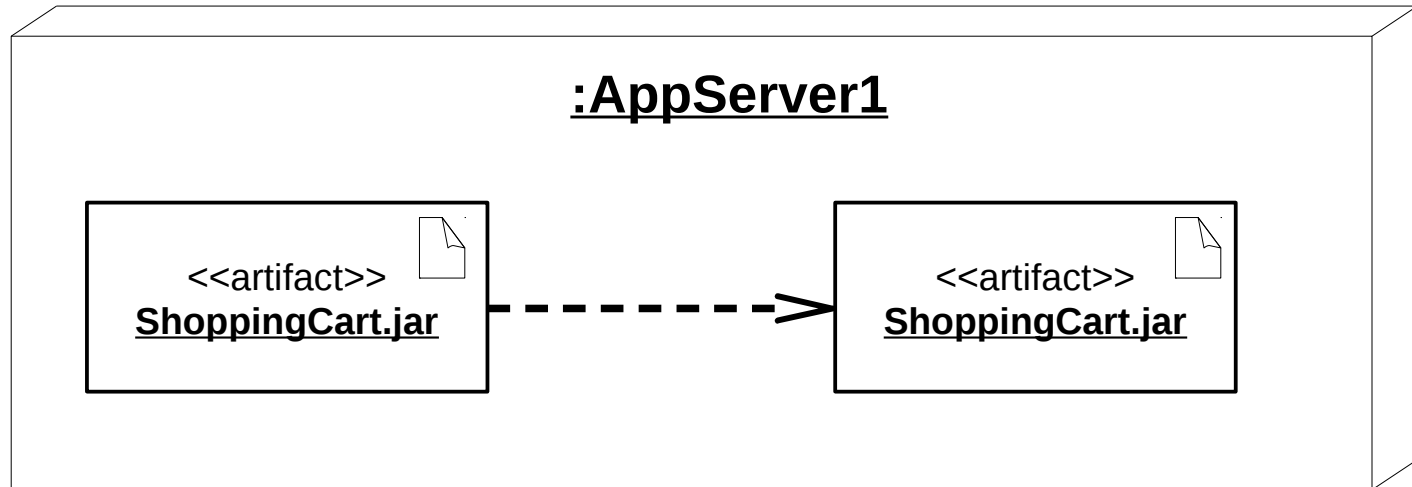
## Environnement d'exécution



# Diagramme de déploiement



# Diagramme de déploiement



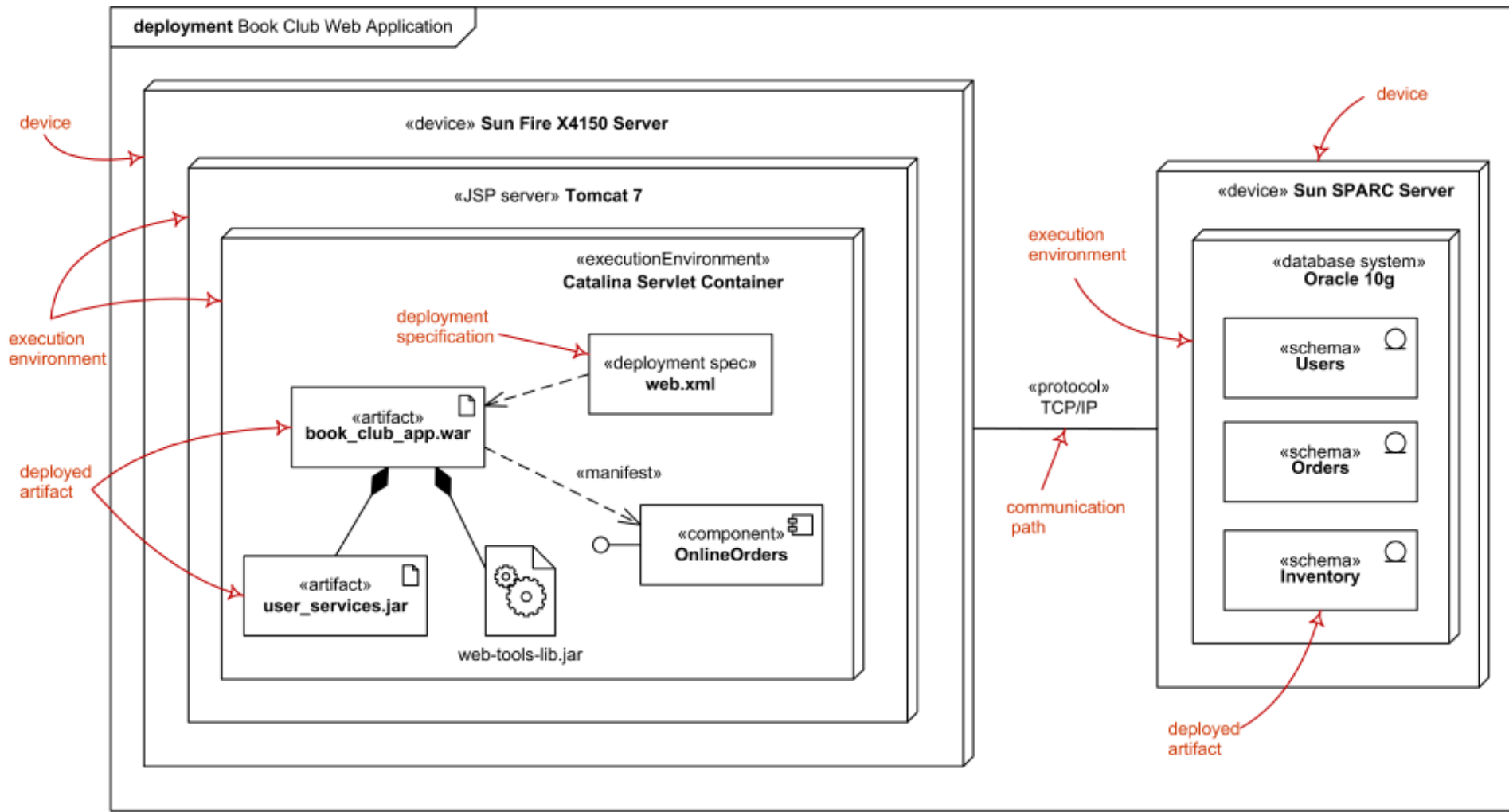
# Diagramme de déploiement

---

## Représentation textuelle



# Diagramme de déploiement



## **DIAGRAMME de MACHINES d'ETATS**

# Diagramme d'objets

---

## Objectif

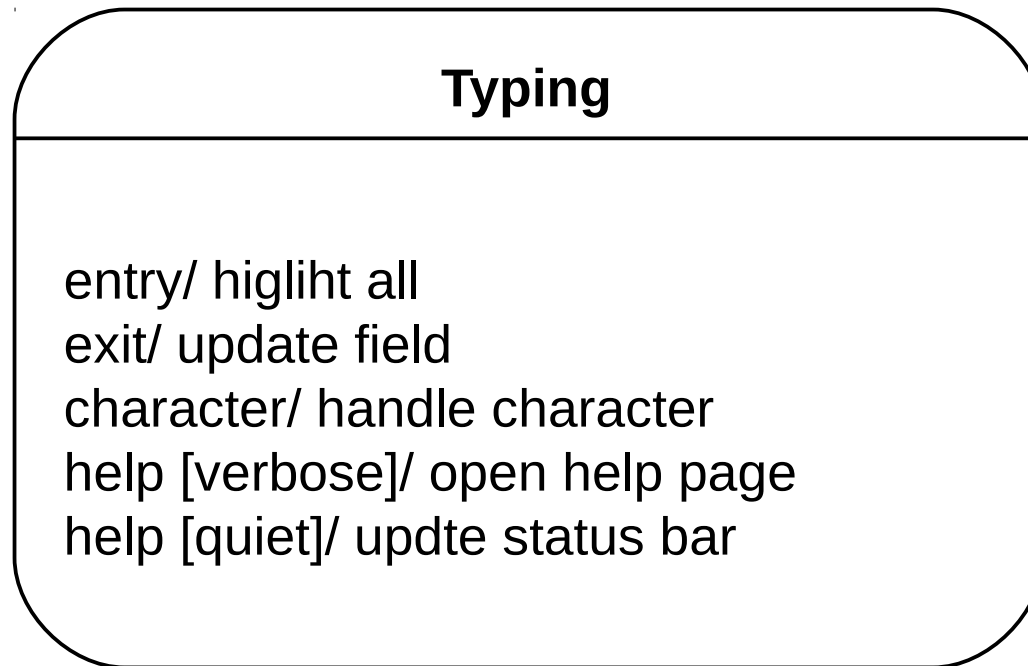
- Description des transitions entre états du système et des conditions d'évolution
- Machines d'état → Automate à états finis



# Diagramme de machines d'états

---

## Activités internes



# Diagramme d'objets

---

## Activités internes

Entry : Exécutée à l'entrée dans l'état

Exit : Exécutée à la sortie de l'état

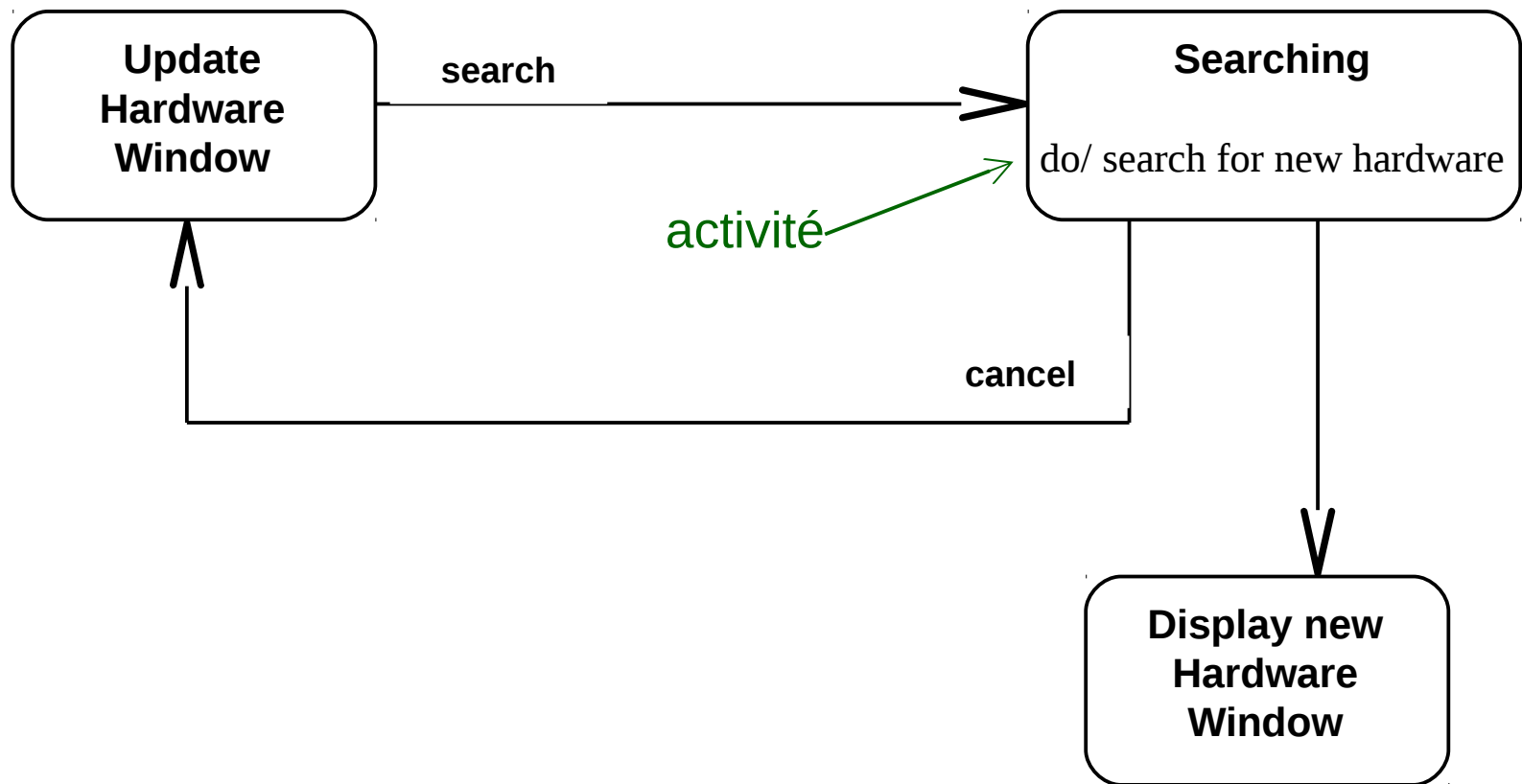
Do : Exécutée après Entry / jusqu'à Exit

Include : Invoque un sous-diagramme

*Evenement* : Exécutée si événement survient  
→ *Evènement[Garde]/Activité*

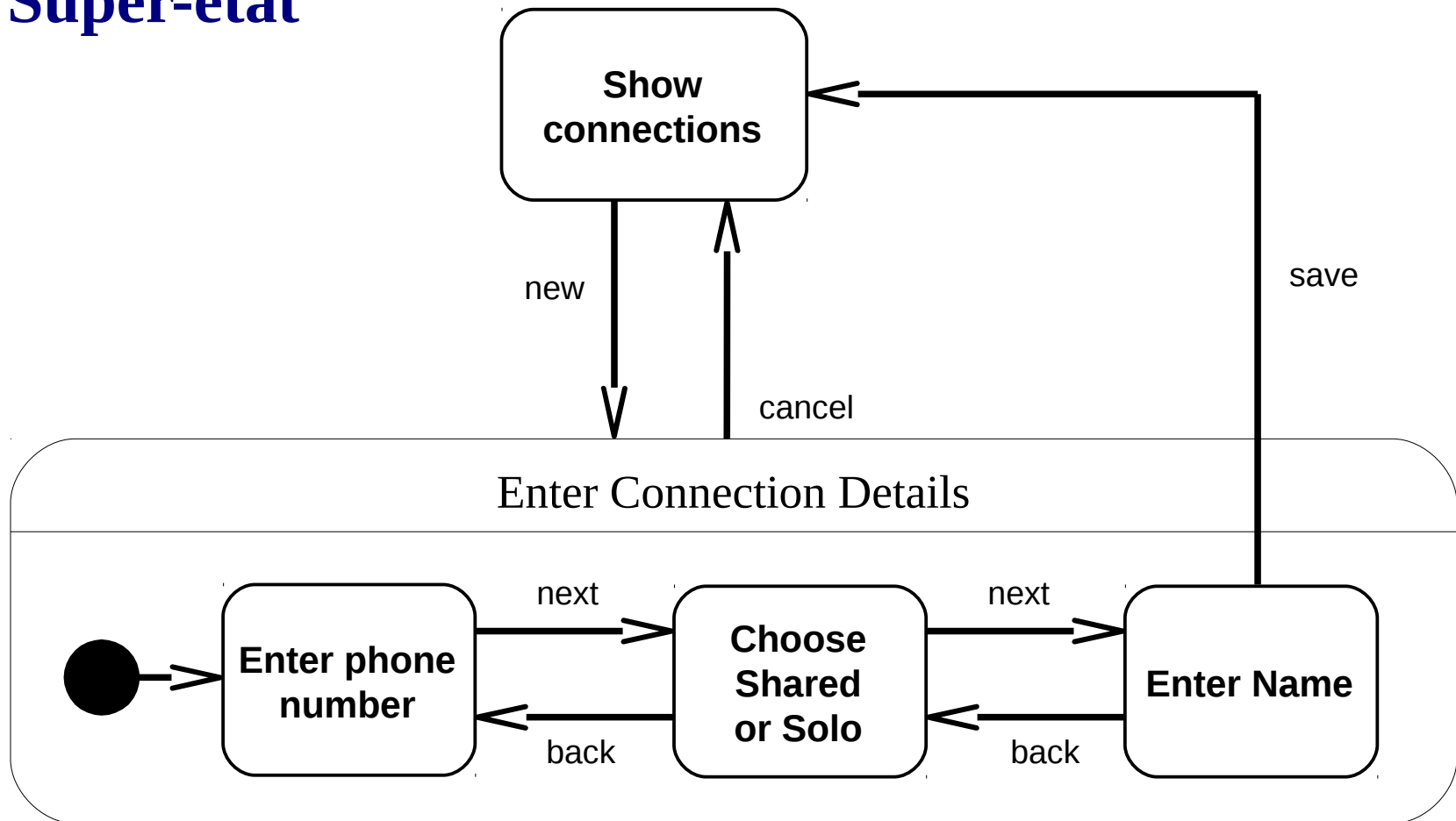
# Diagramme de machines d'états

## Etats d'activités



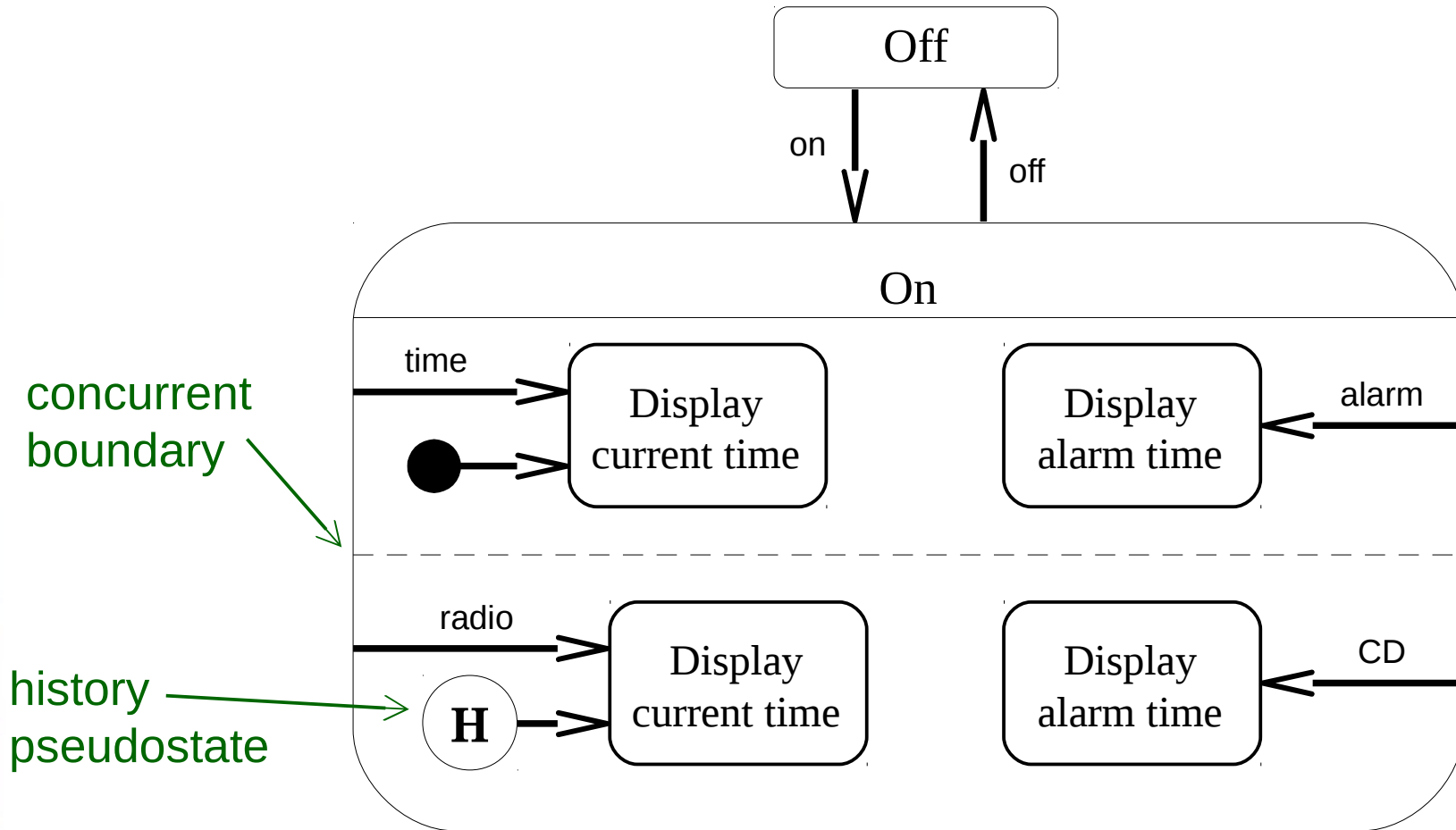
# Diagramme de machines d'états

## Super-état



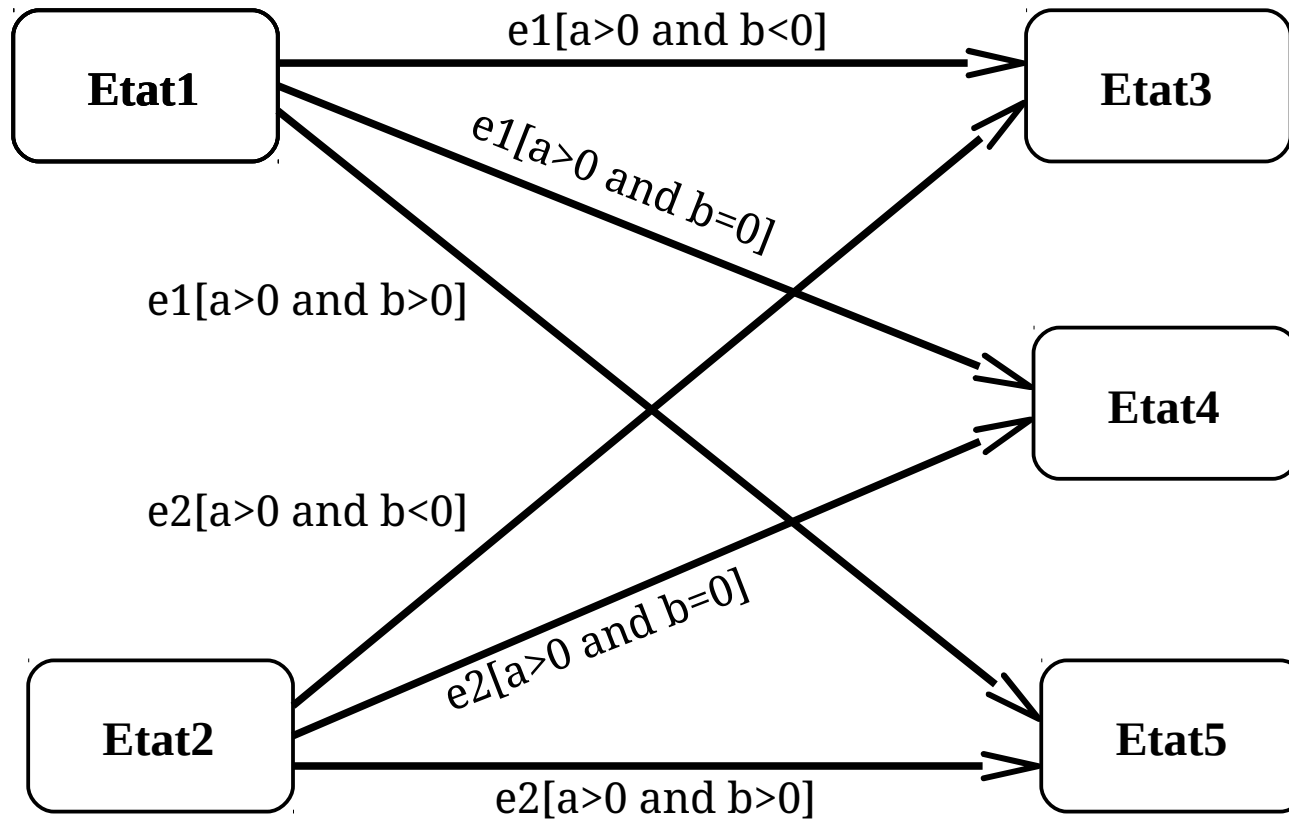
# Diagramme de machines d'états

## Etats concurrents



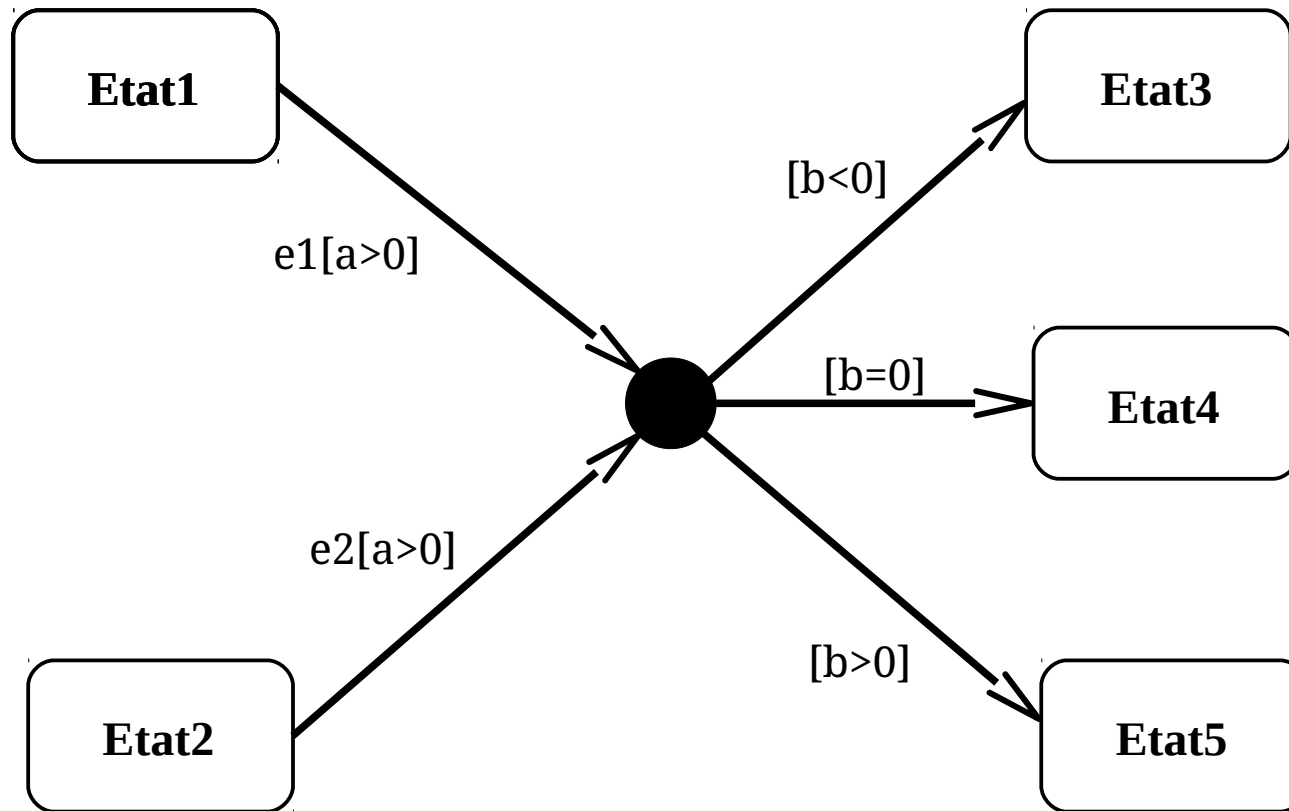
# Diagramme de machines d'états

## Point de jonction



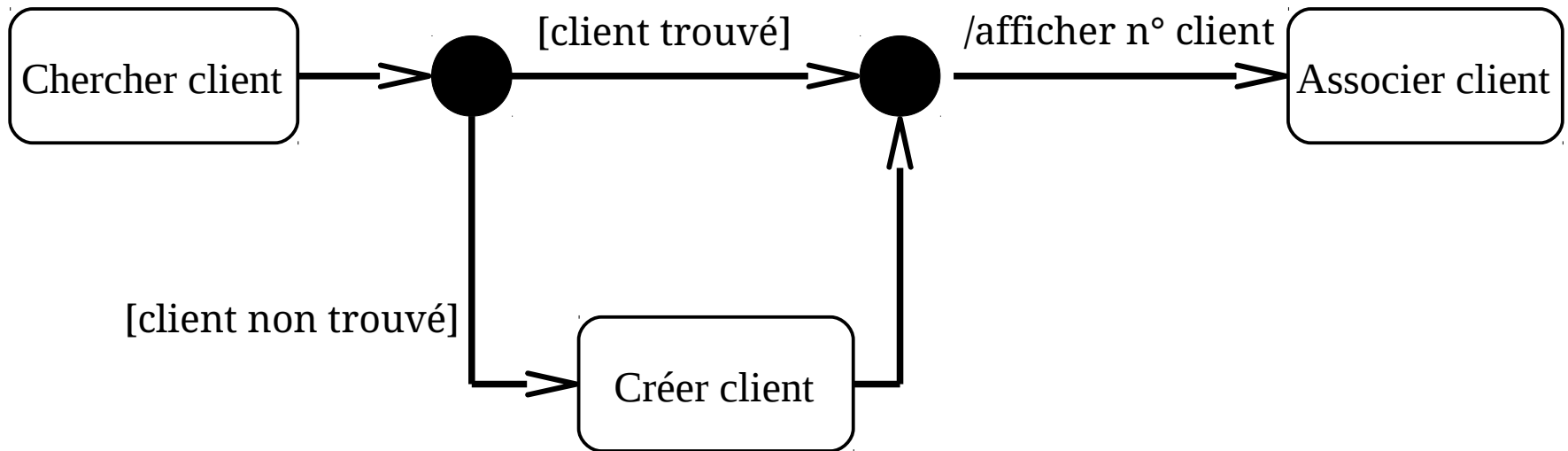
# Diagramme de machines d'états

## Point de jonction



# Diagramme de machines d'états

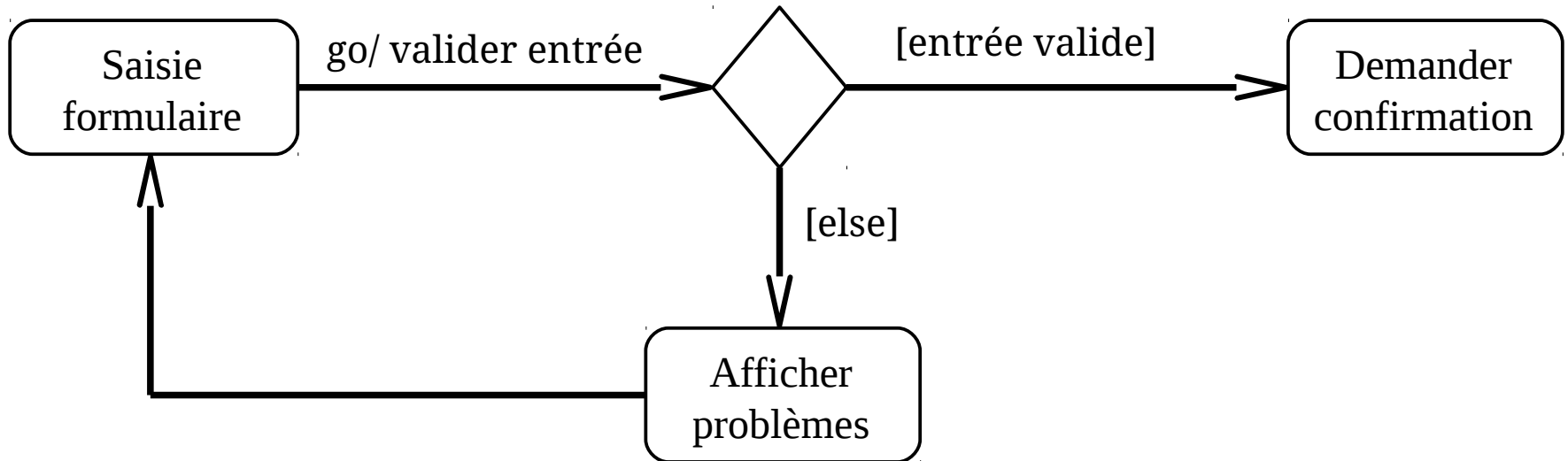
## Point de jonction / Alternative





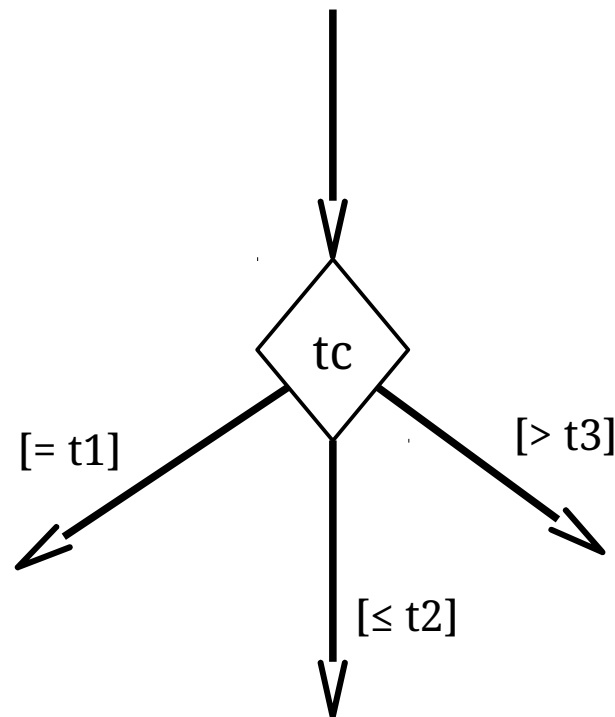
# Diagramme de machines d'états

## Point de décision



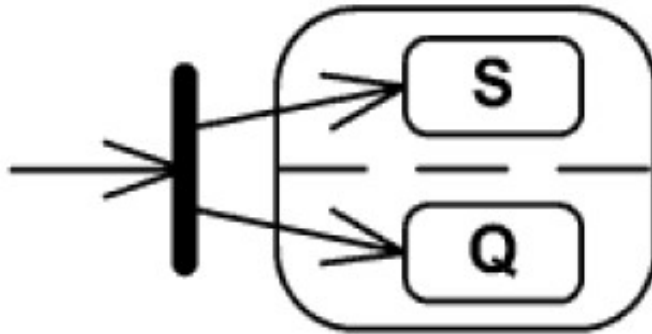
# Diagramme de machines d'états

## Décisions multiples

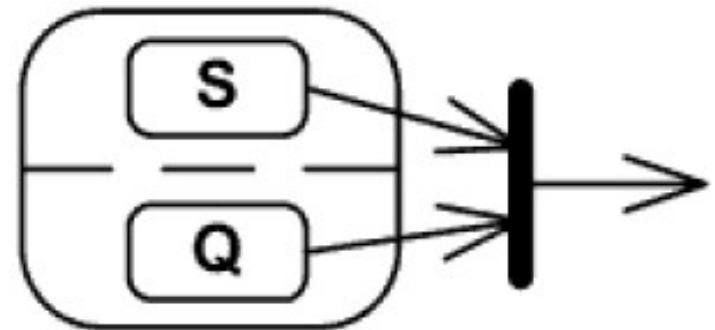


# Diagramme de machines d'états

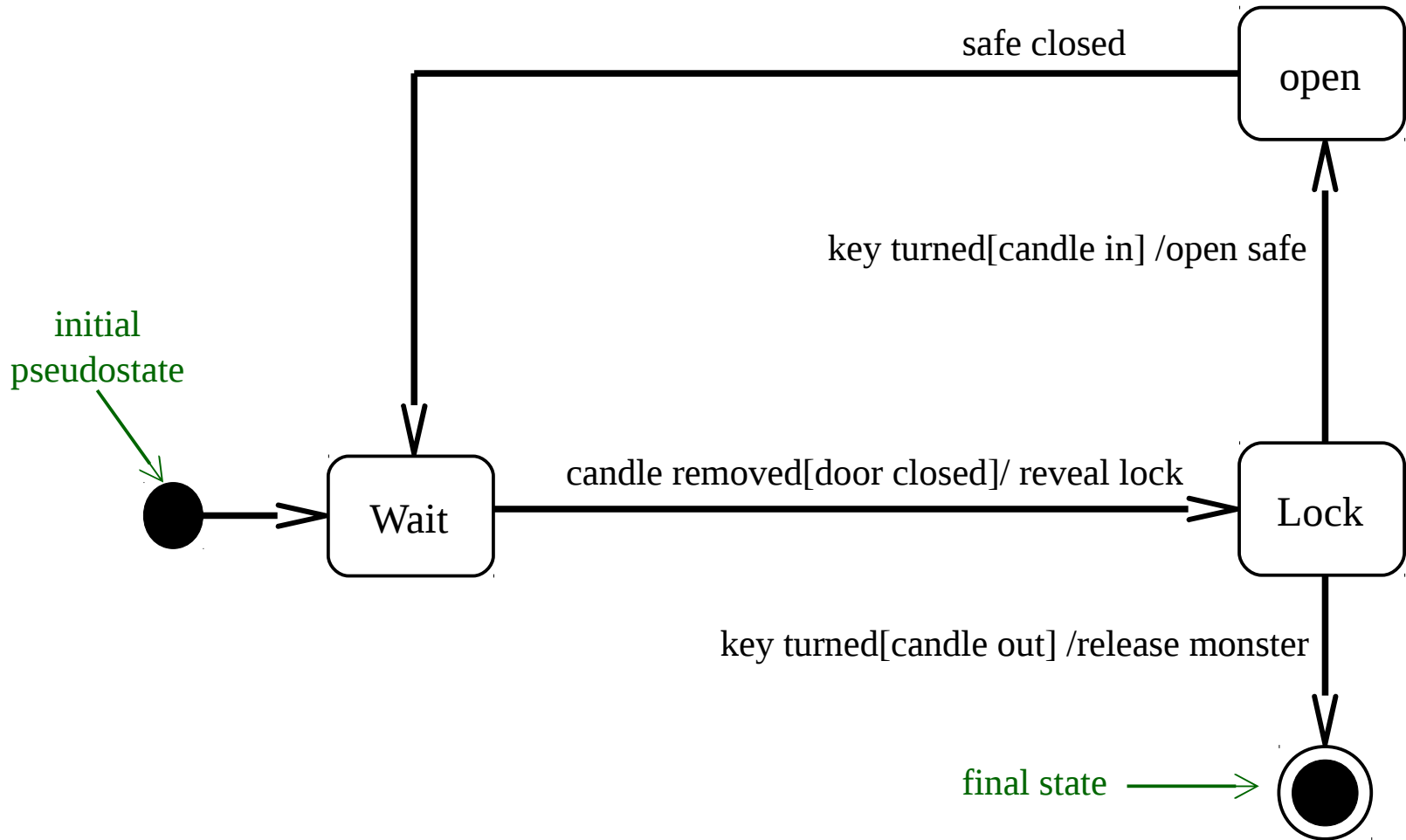
**Fork**



**Join**

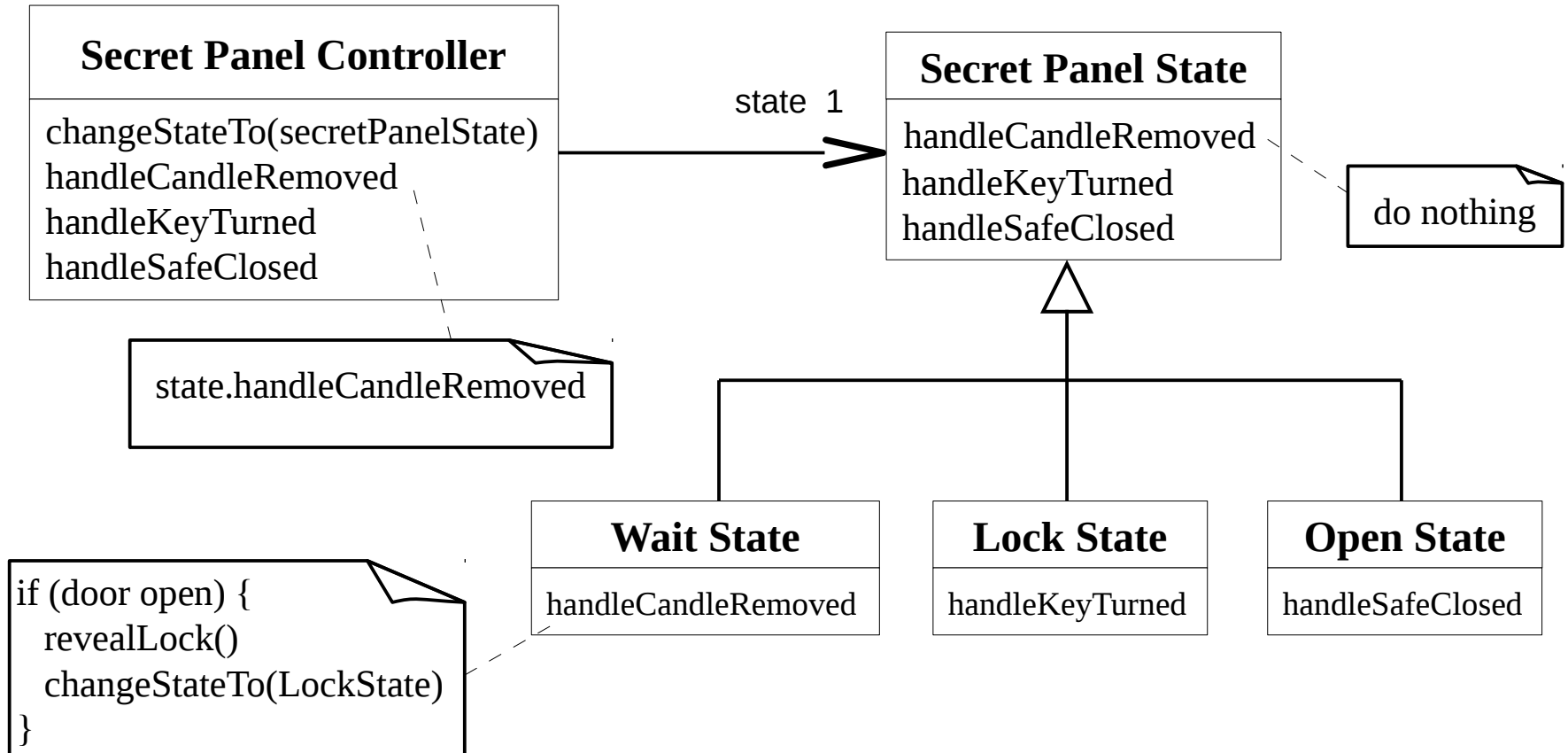


# Diagramme de machines d'états



# Diagramme de machines d'états

## Implémentation par pattern d'état



# Diagramme de machines d'états

## Implémentations par table d'états

Etat source	Etat cible	Evènement	Garde	Procédure
Attente	Fermé	Chandelle otée	Porte ouverte	Montrer serrure
Fermé	Ouvert	Clé tournée	Chandelle dedans	Ouvrir coffre
Fermé	Final	Clé tournée	Chandelle dehors	Libérer monstre
Ouvert	Attente	Coffre fermé		

UML

---

# DIAGRAMME d'ACTIVITES

# Diagramme d'activités

---

## Objectif

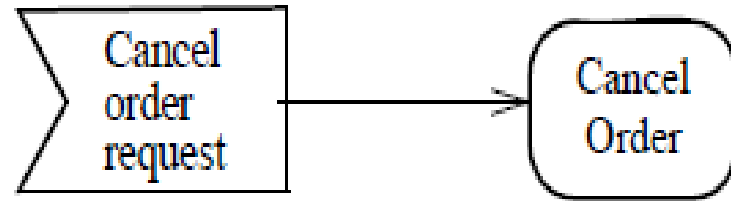
- Description d'un comportement défini par des enchainements d'actions



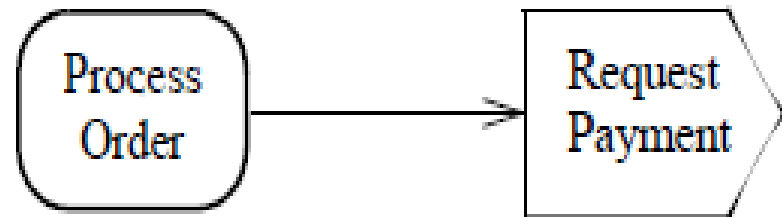
# Diagramme d'activités

Description des évènements :

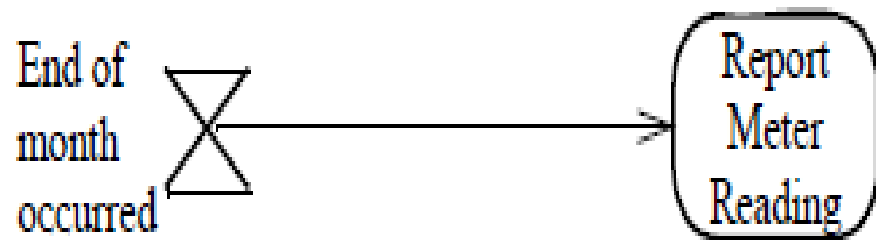
- Réception d'un signal



- Emission d'un signal

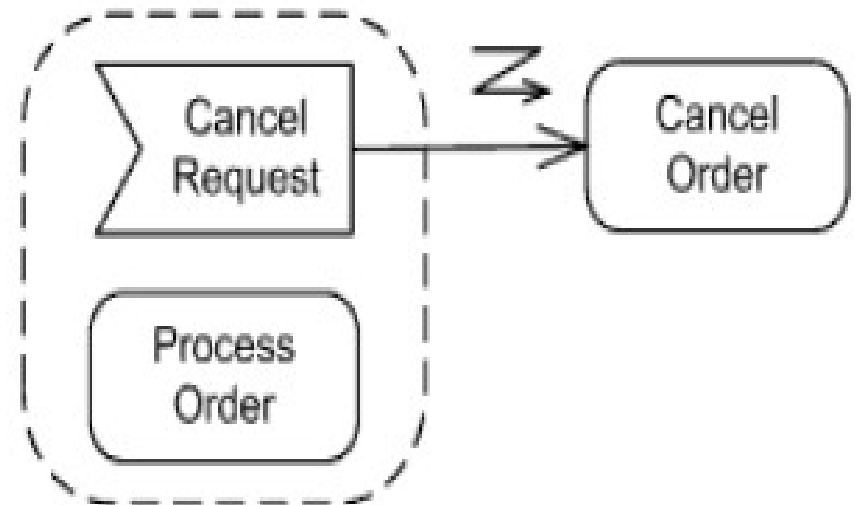
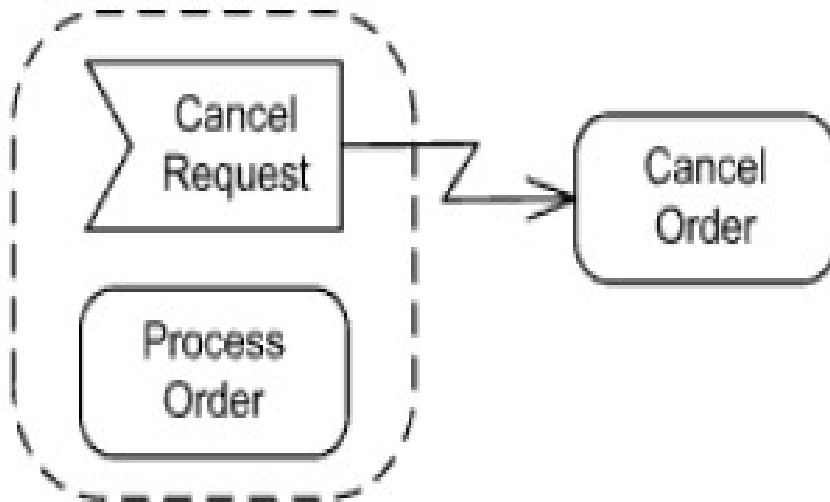


- Evènement répétitif



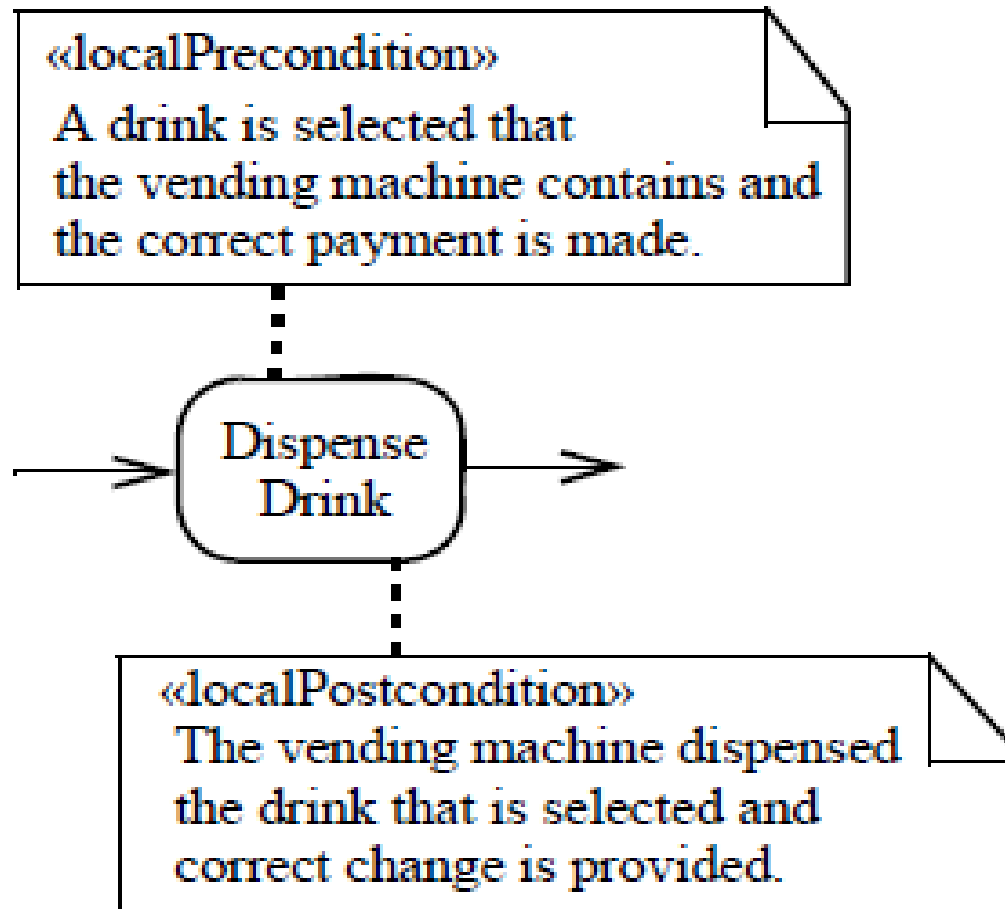
# Diagramme d'activités

## Signal d'interruption



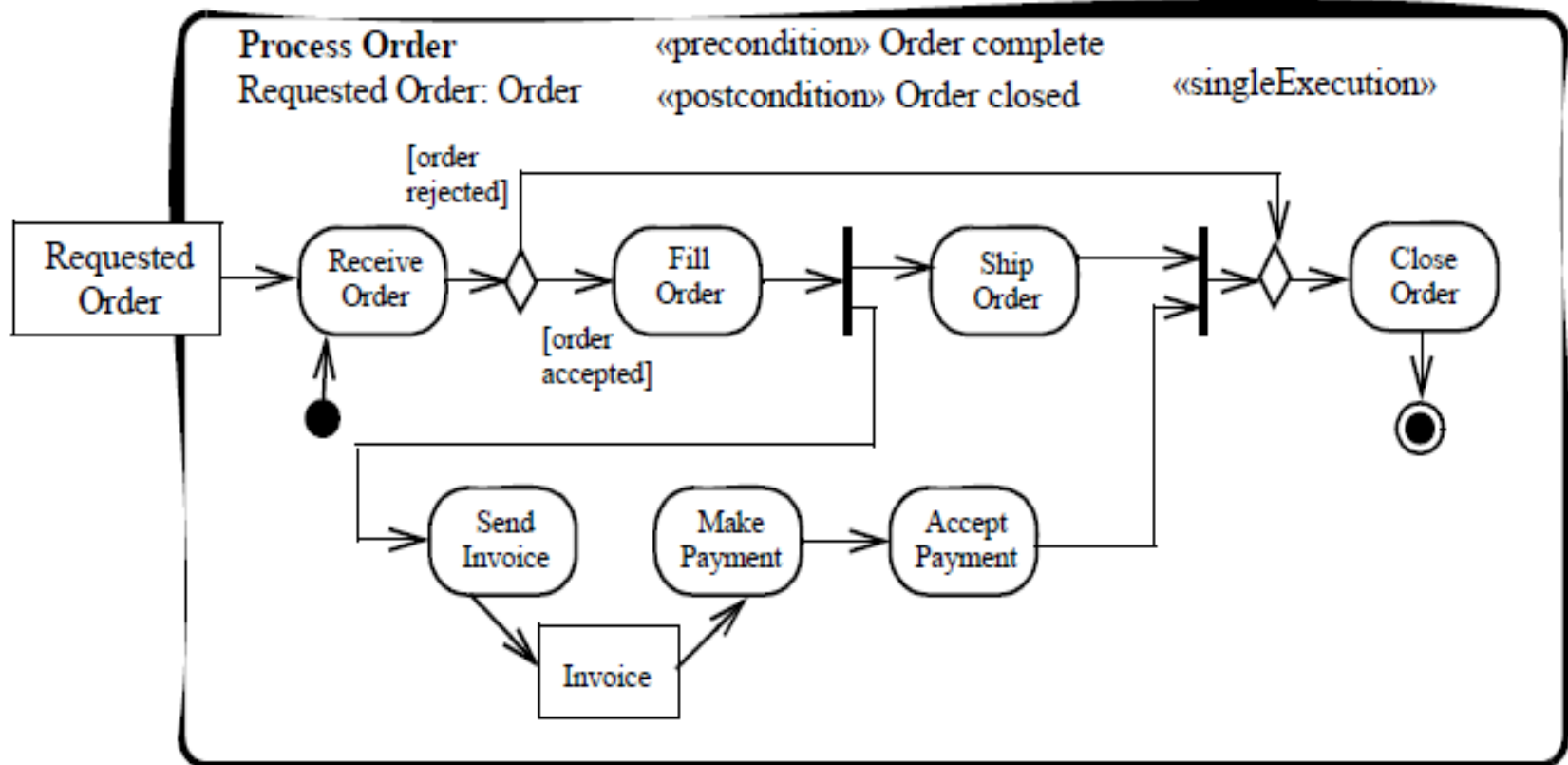
# Diagramme d'activités

## Action avec pré et post-conditions

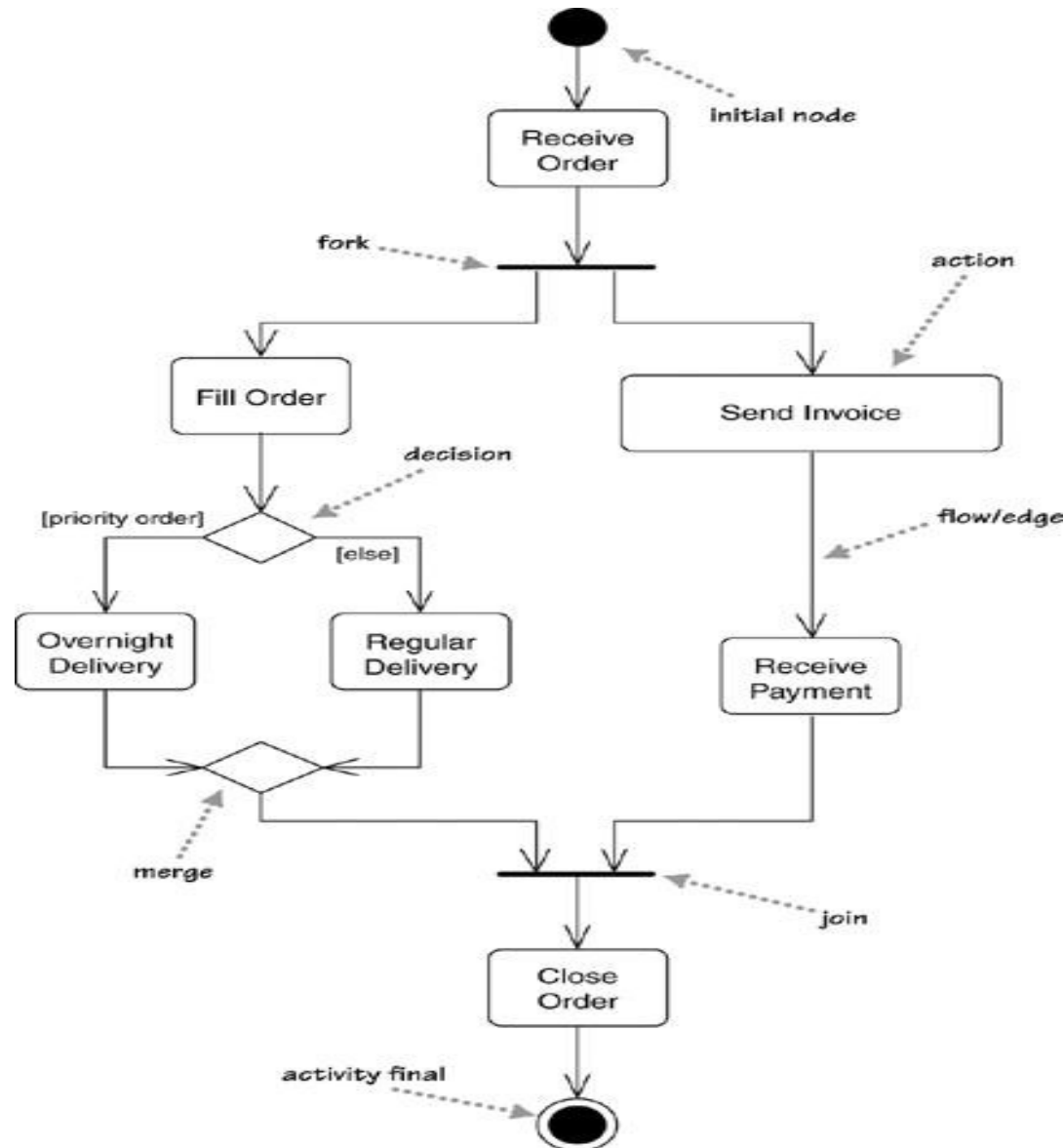


# Diagramme d'activités

## Activité avec paramètre d'entrée

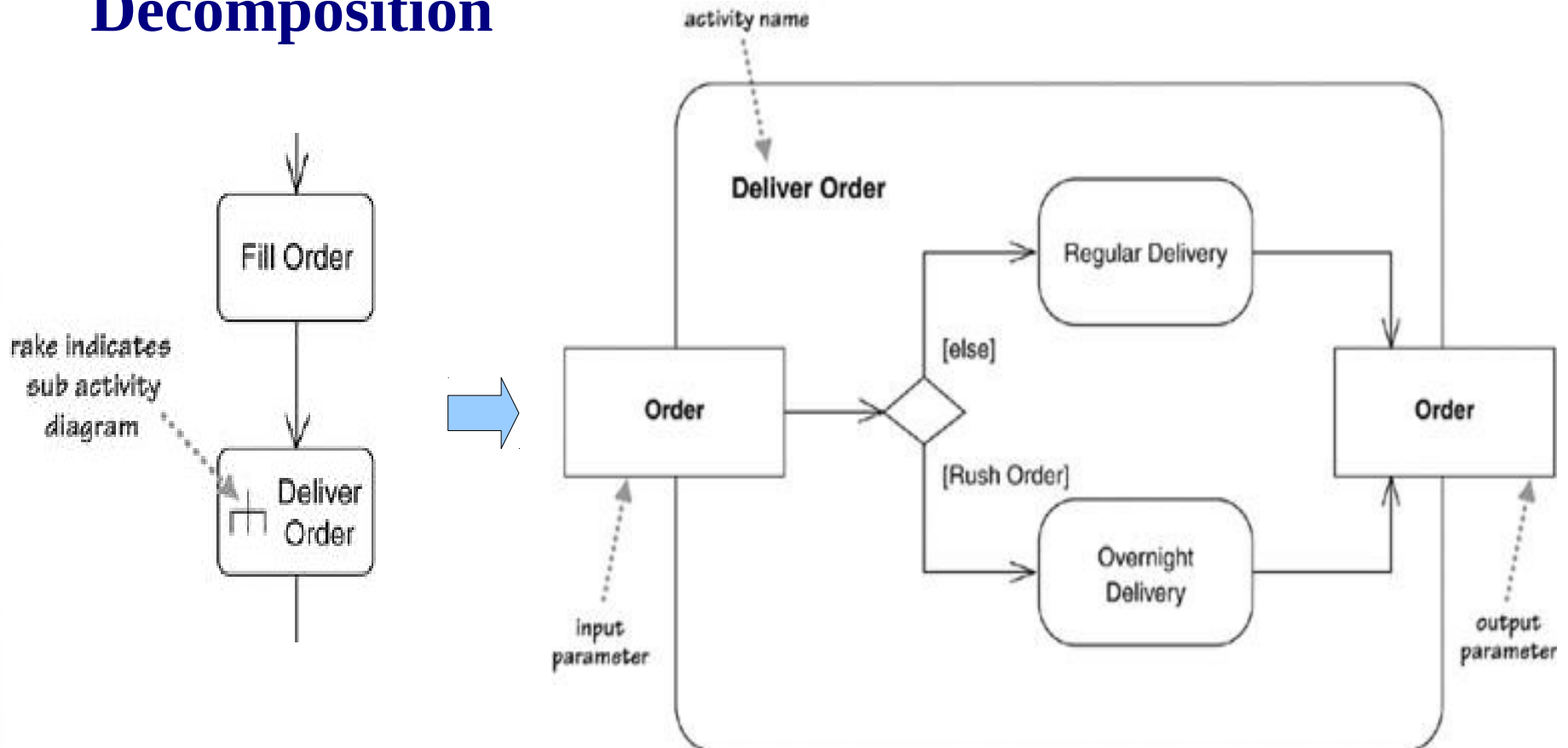


# Diagramme d'activités



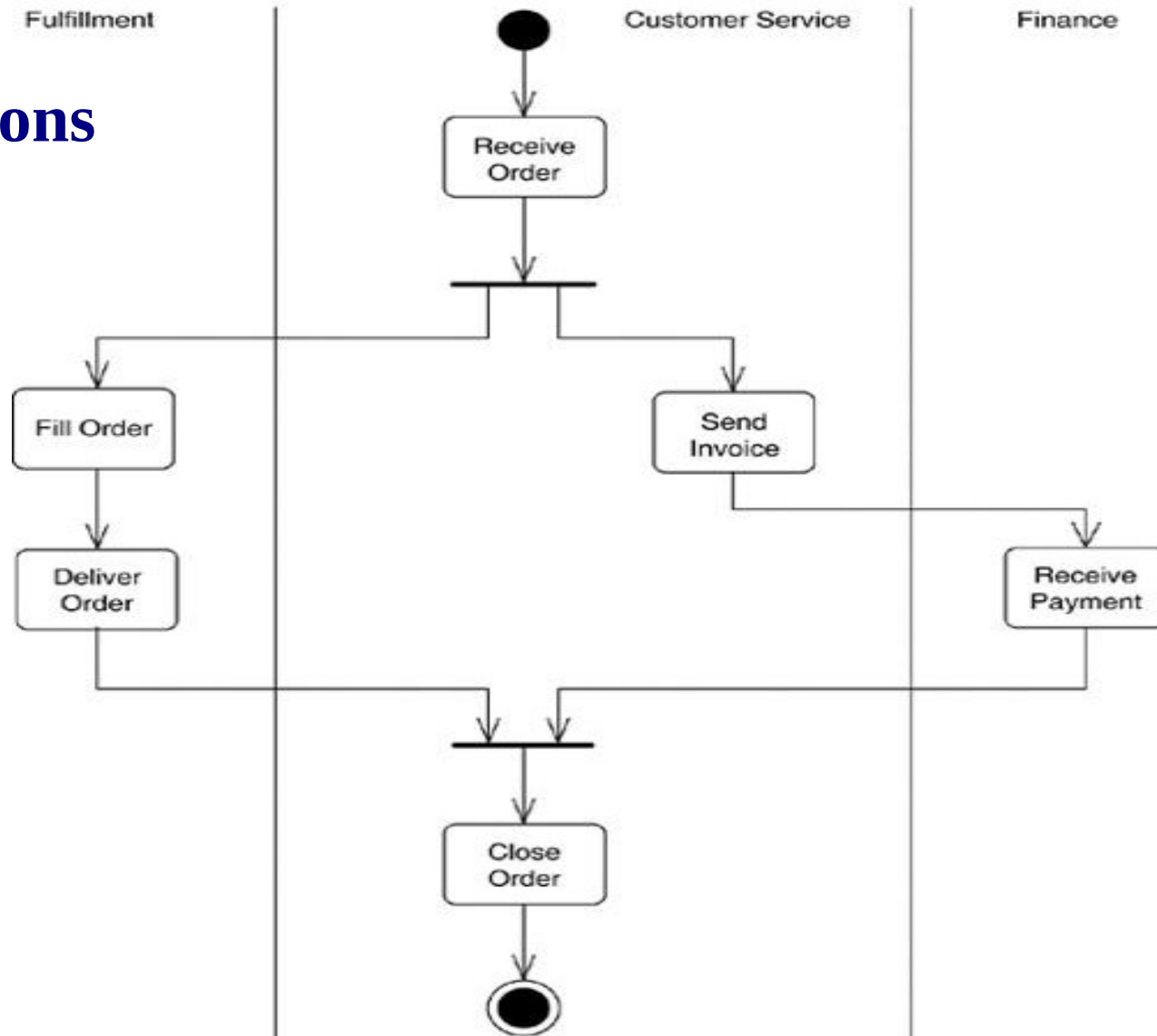
# Diagramme d'activités

## Décomposition



# Diagramme d'activités

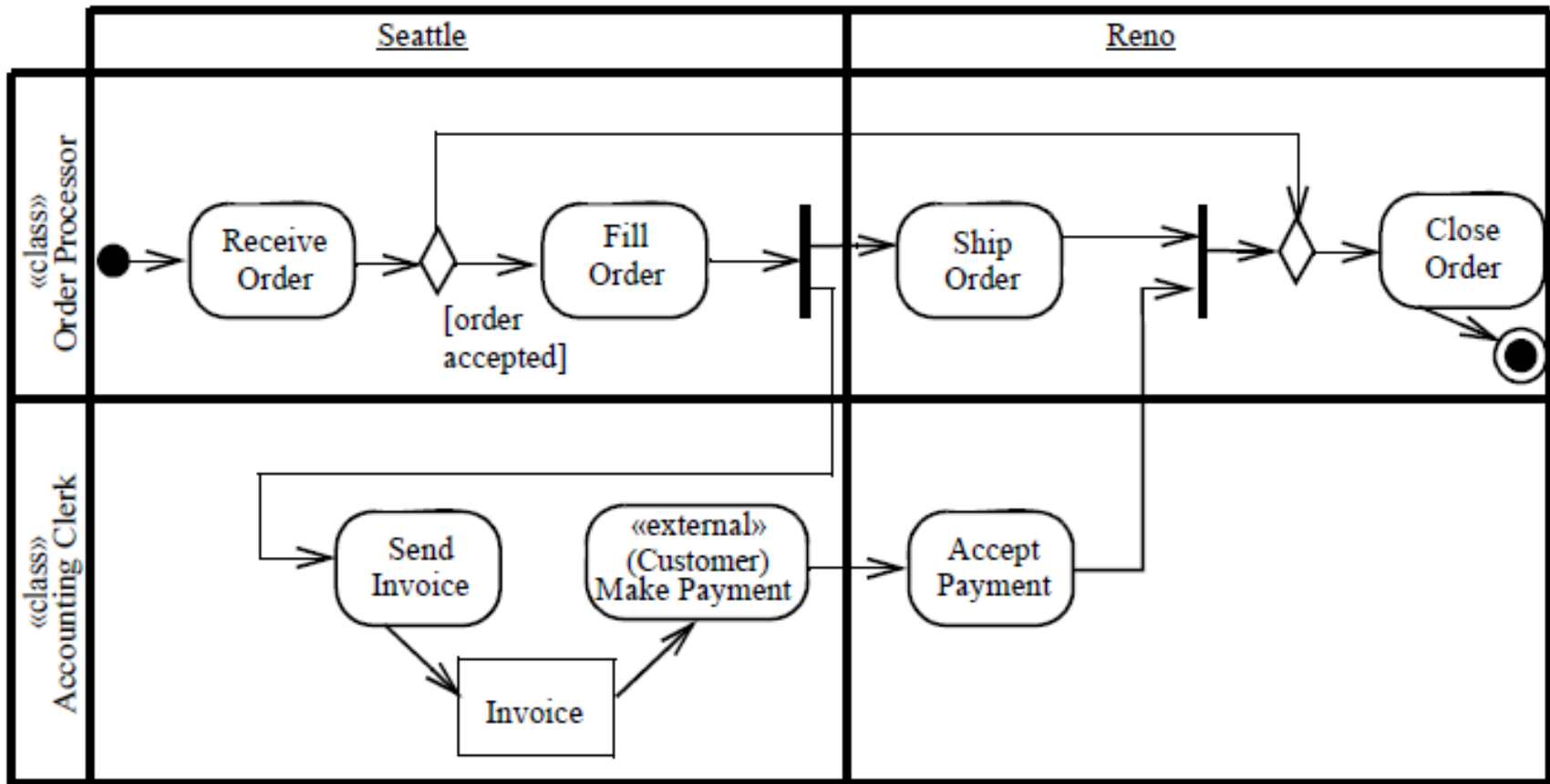
## Partitions



# Diagramme d'activités

## Partitionnement multiple

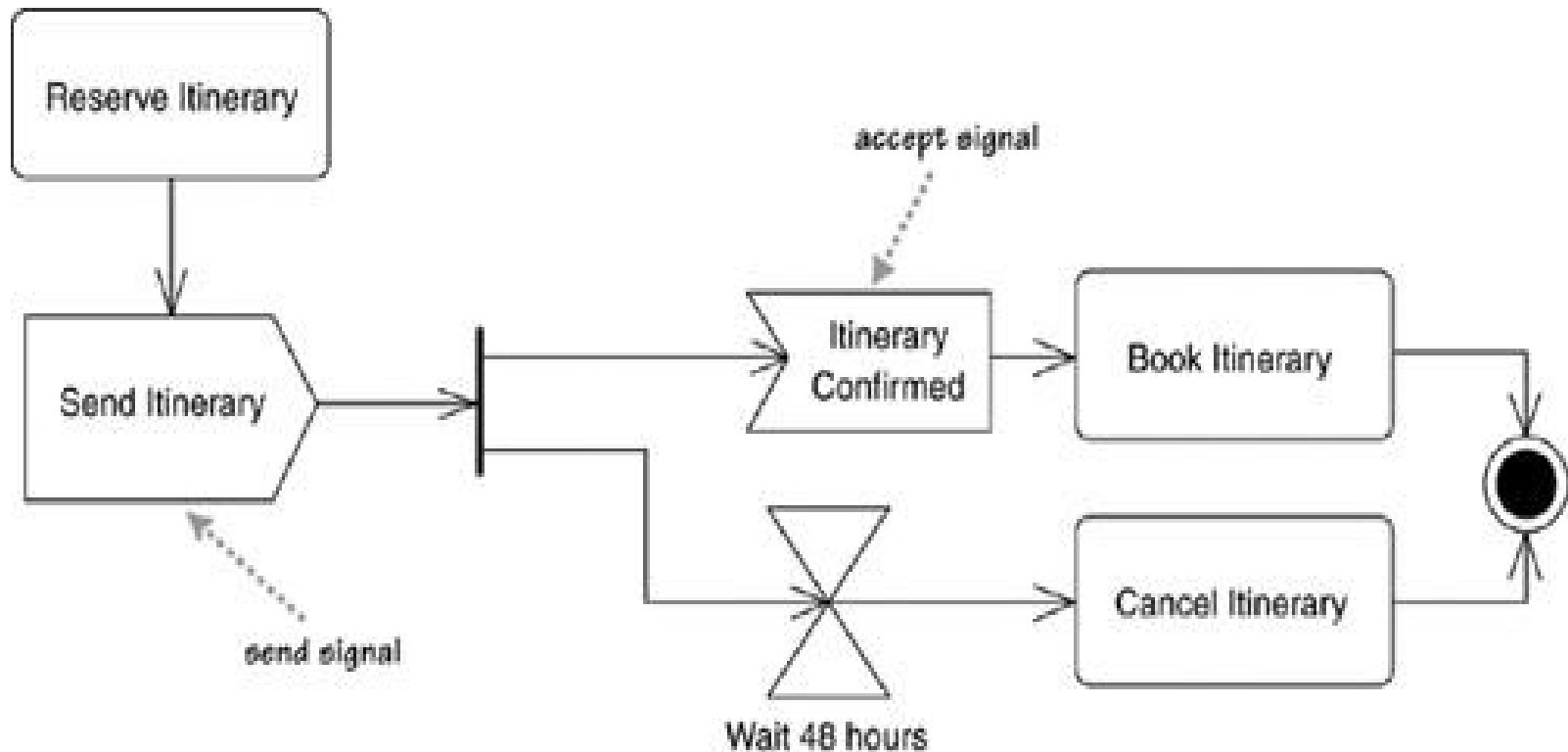
«attribute» performingLocation:Location





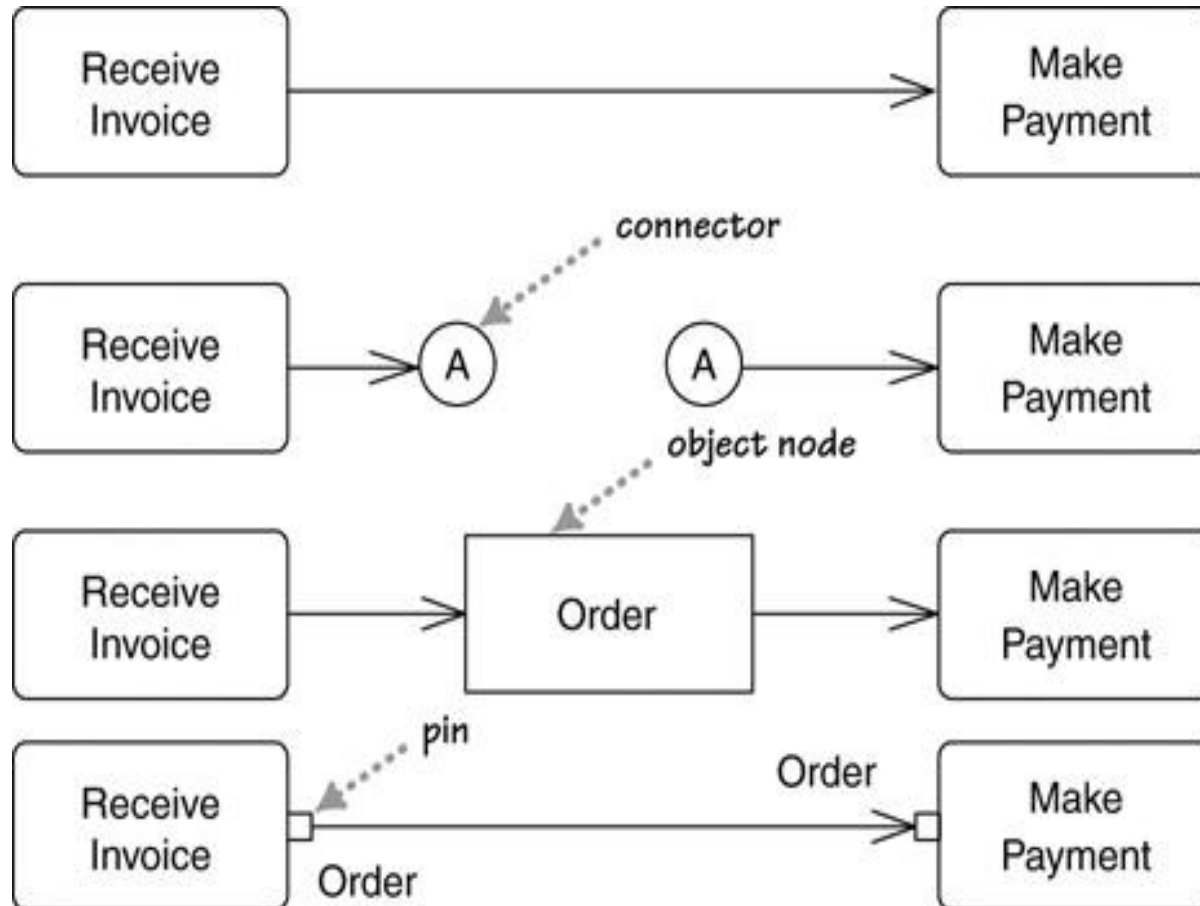
# Diagramme d'activités

## Signaux



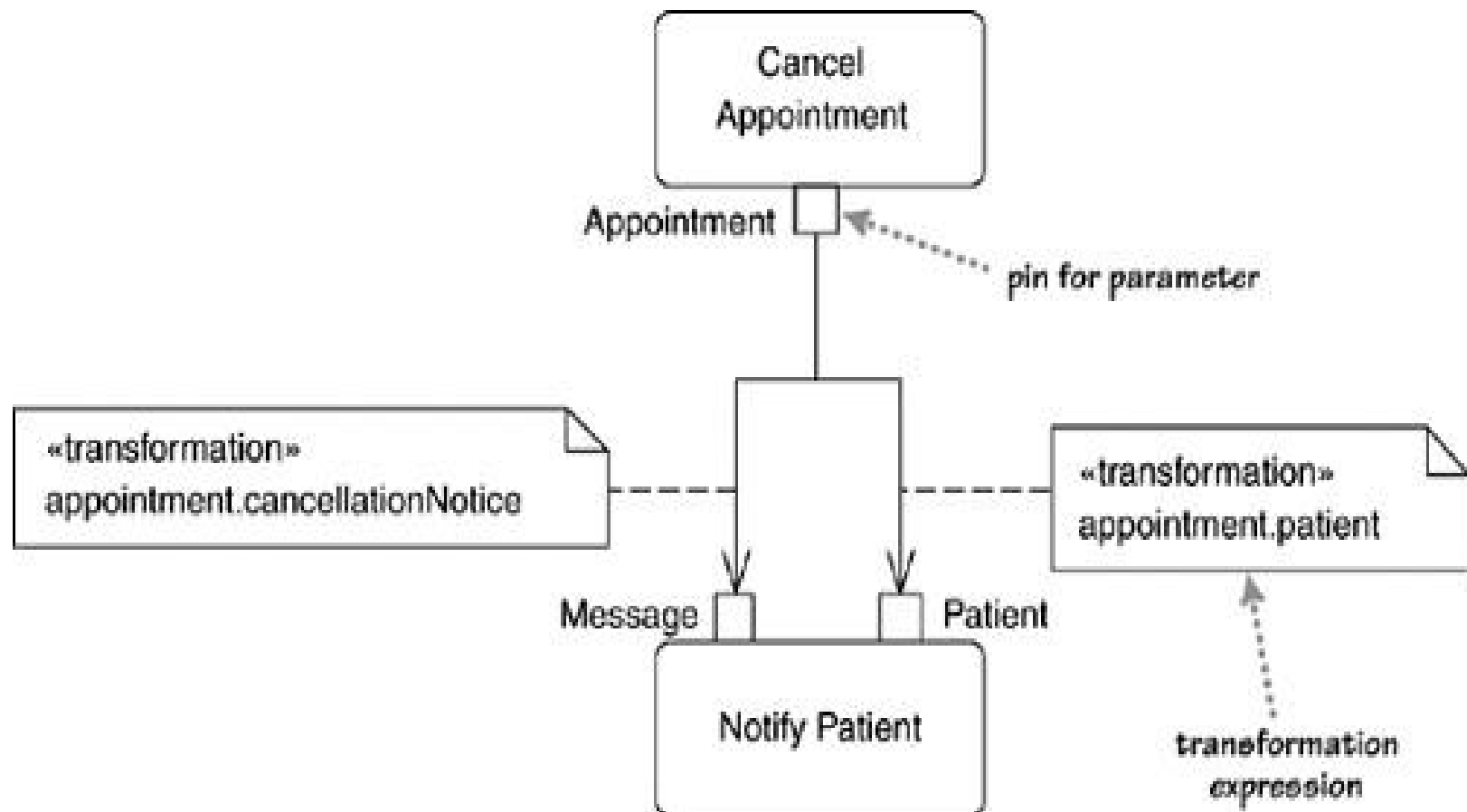
# Diagramme d'activités

## Edge and Flow



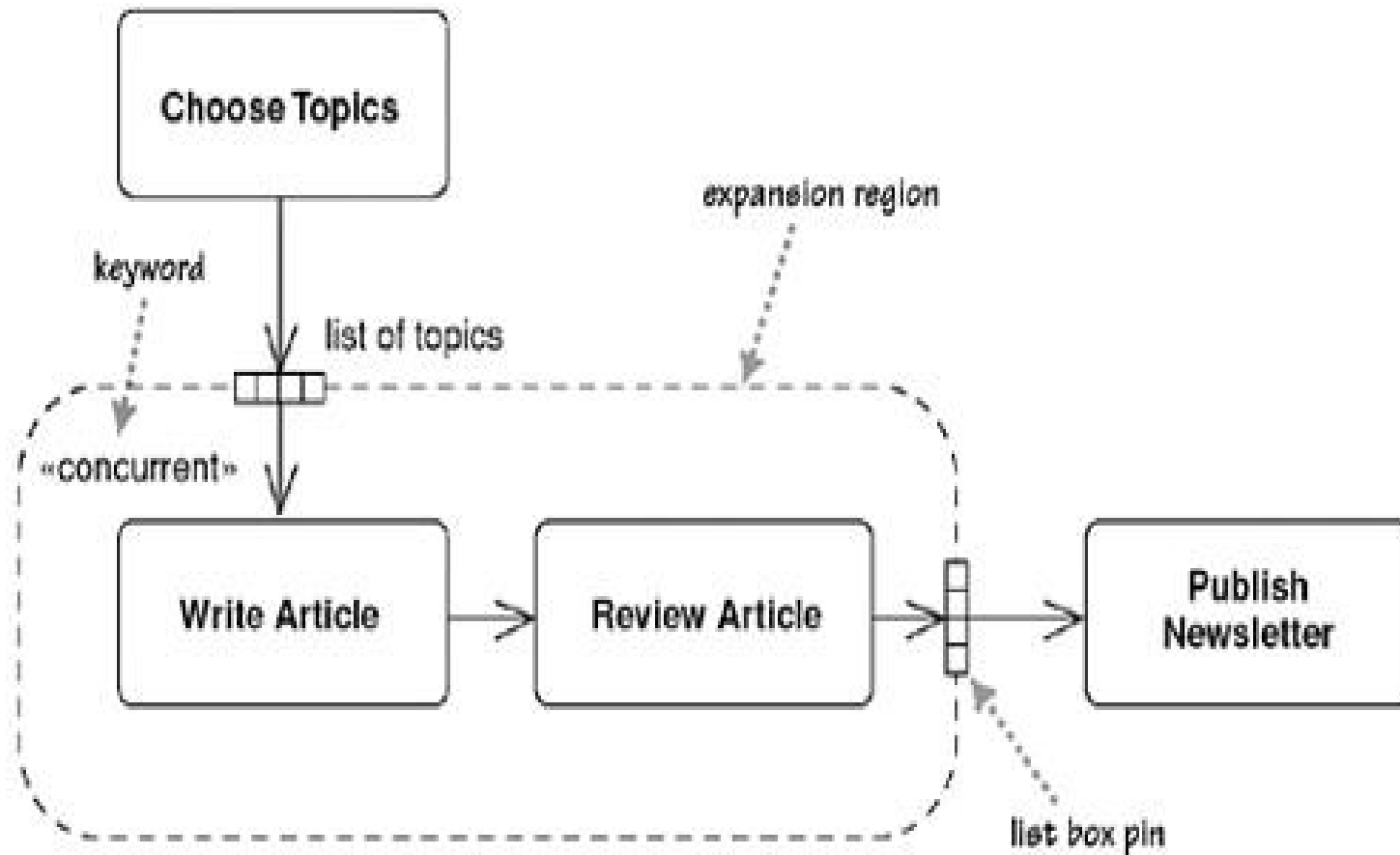
# Diagramme d'activités

## Connecteurs et transformations



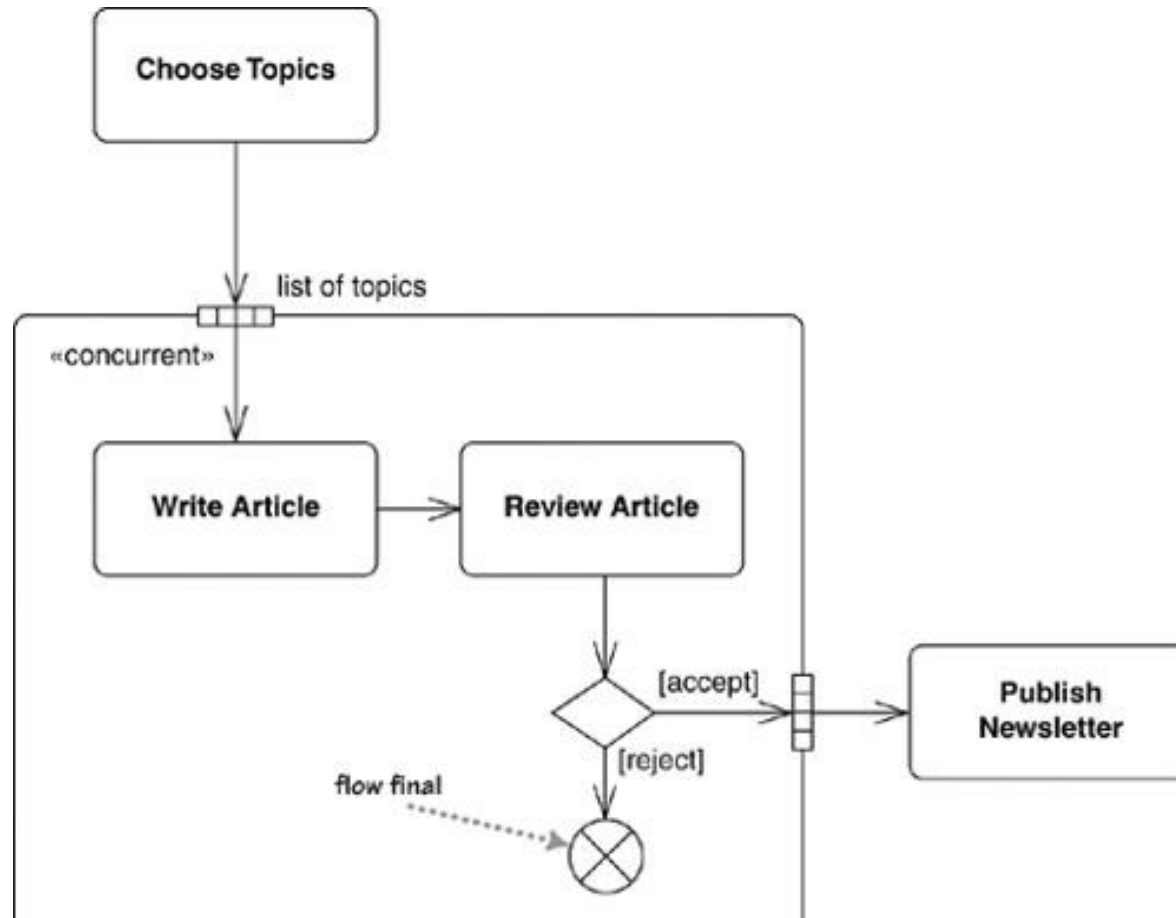
# Diagramme d'activités

## Région d'expansion



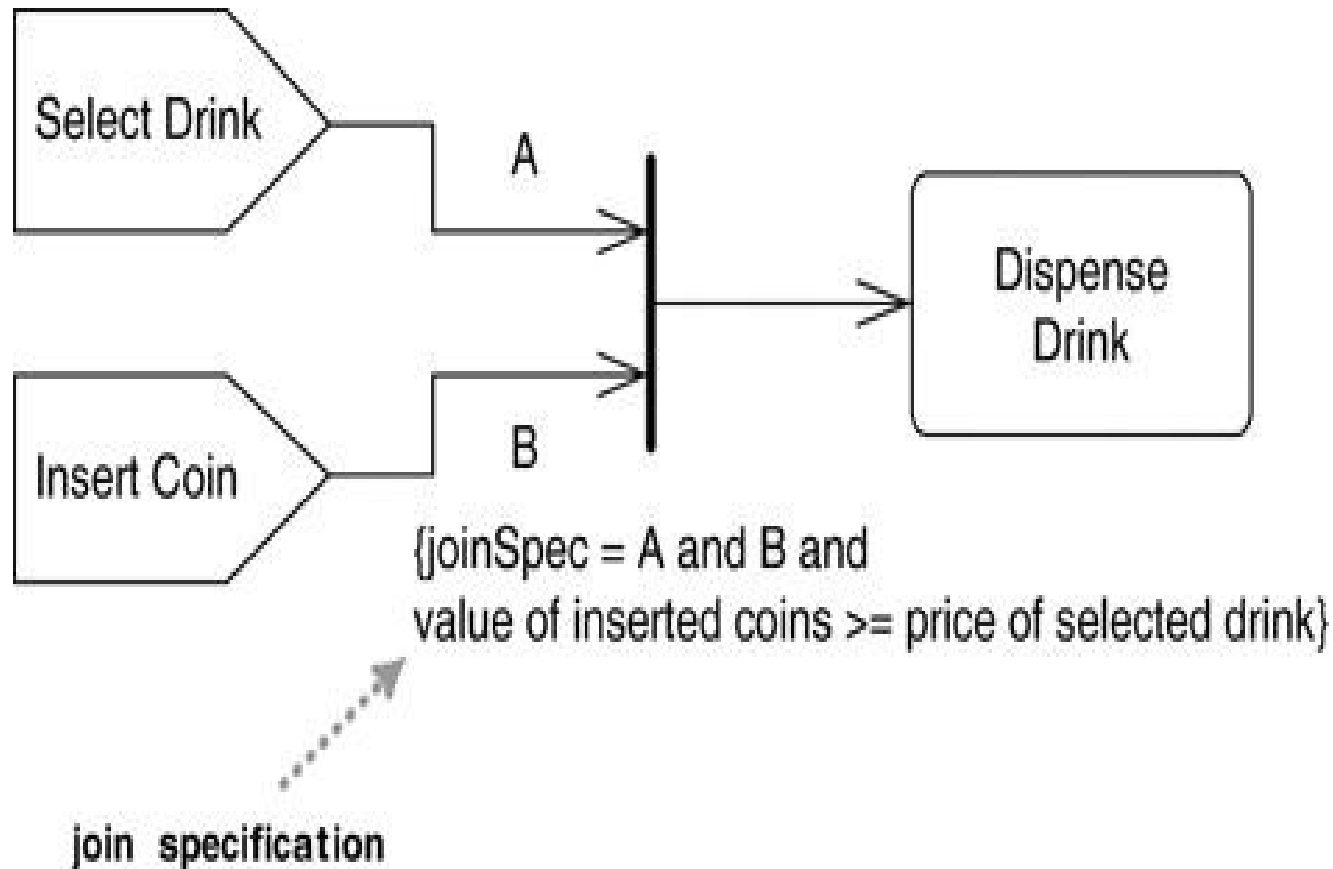
# Diagramme d'activités

## Terminaison de flot



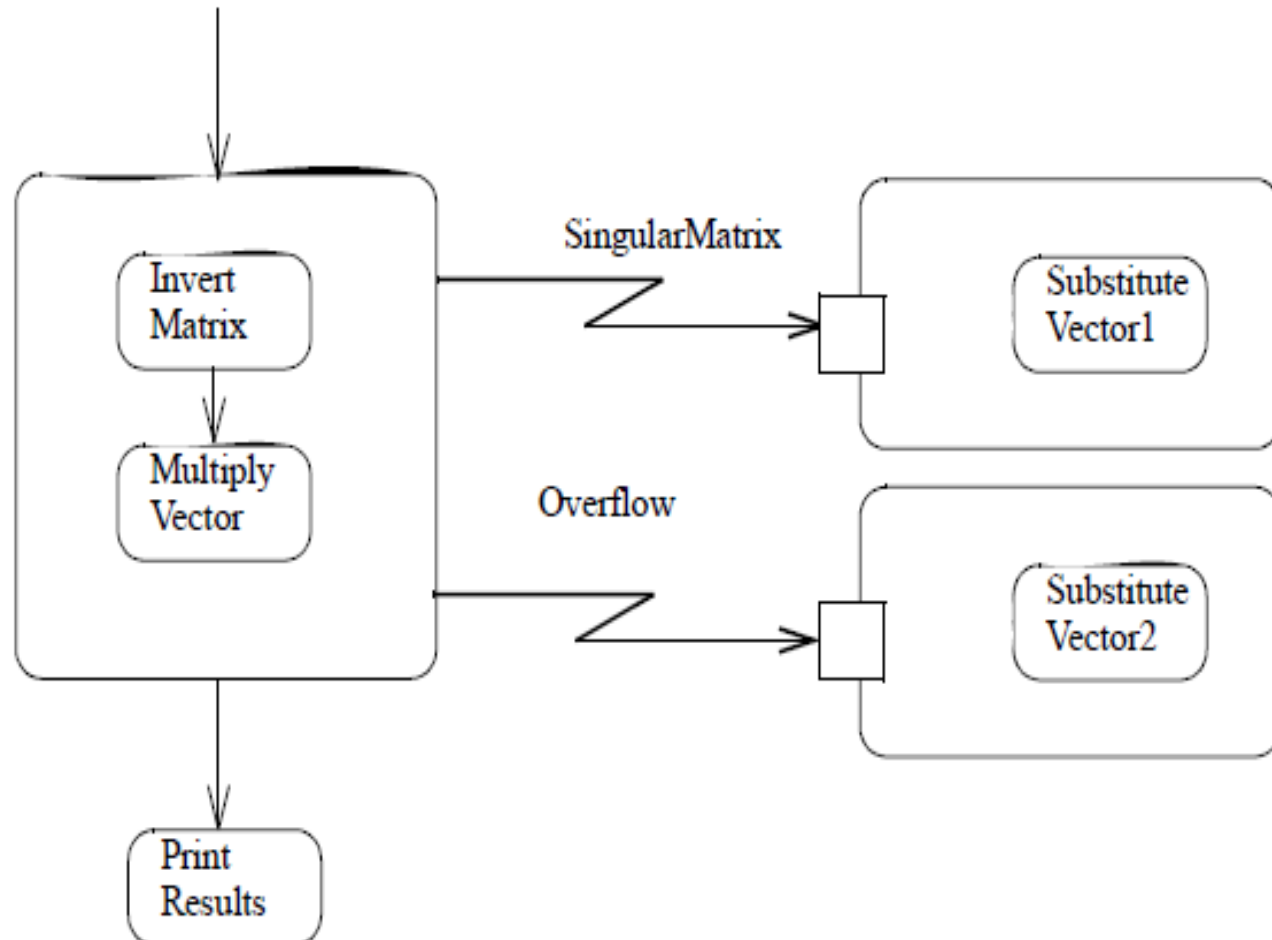
# Diagramme d'activités

## Spécification des jonctions



# Diagramme d'activités

## Exemple de gestionnaire d'interruption



UML

---

# DIAGRAMME de COMMUNICATION



# Diagramme de communication

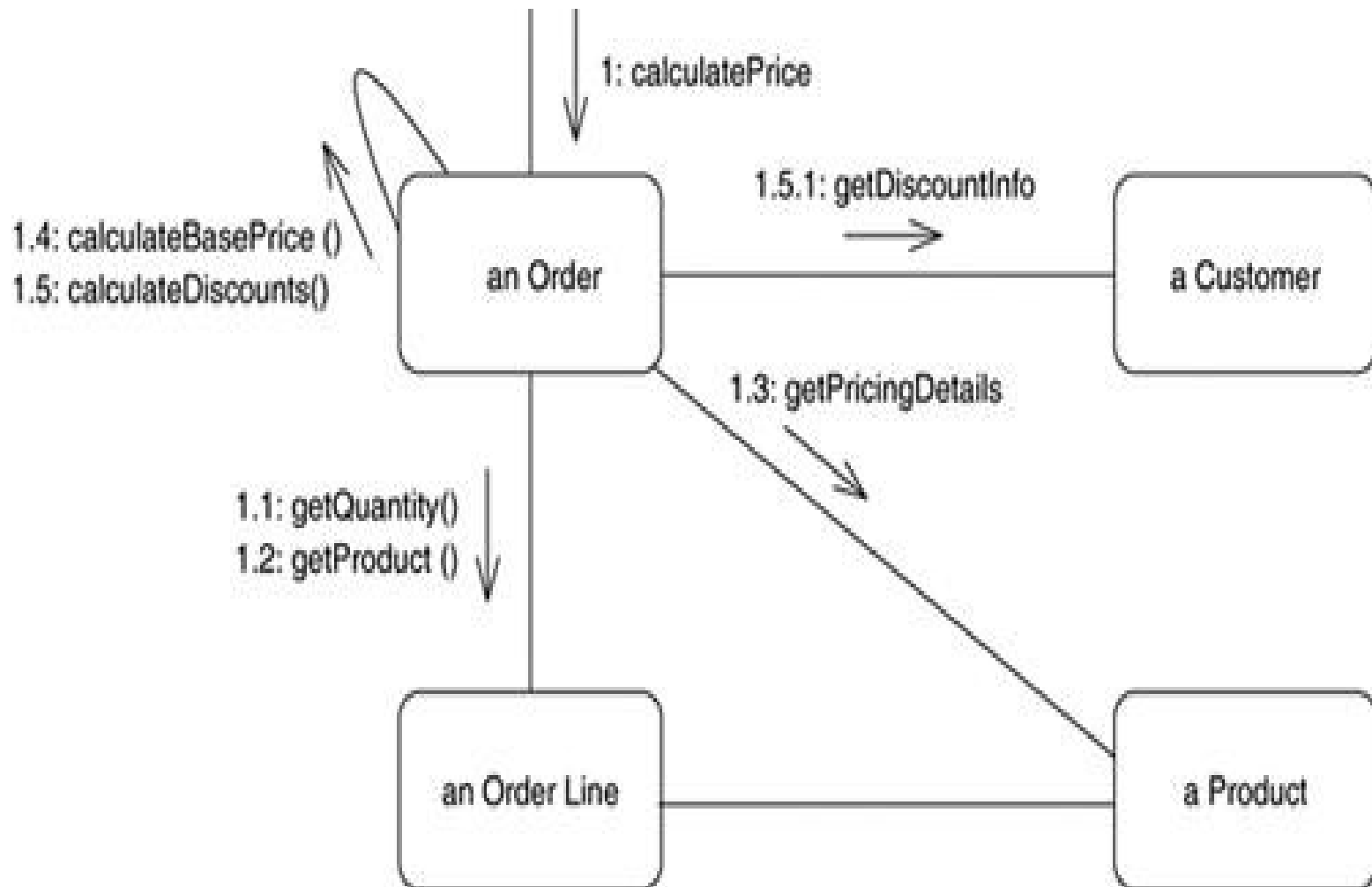
---

## Objectif

- Mise en évidence des liens entre classe
- Informations dynamiques (séquence)

*Remarque : UML 1 → Appelé diagramme de collaboration*

# Diagramme de communication



# Diagramme de communication

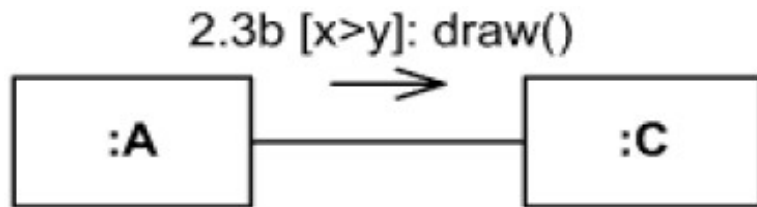
---

## Numérotation des messages

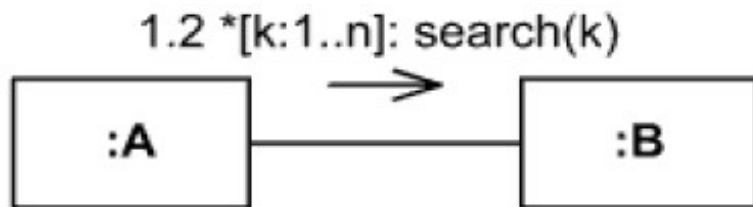
- Numérotation hiérarchique  
1.5 précède 1.5.1 ...  
! Limiter le nombre de niveau (lisibilité)
- Numérotation des messages concurrents  
1a1 en parallèle de 1b1

# Diagramme de communication

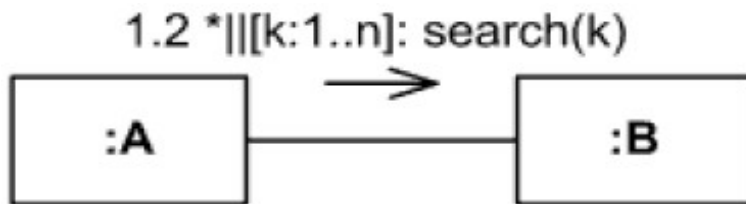
## Messages conditionnels



Exécution si  $x > y$



Emission de 'n' messages successifs



Emission de 'n' messages simultannés

# Diagramme de communication

---

## Remarque

- Présentation des principes dynamiques des classes
- Faible capacité de représentation des séquences, boucles, choix

Faible capacité de structuration et  
Faible capacité d'interférencement

## **DIAGRAMME des FLUX d'INFORMATIONS**

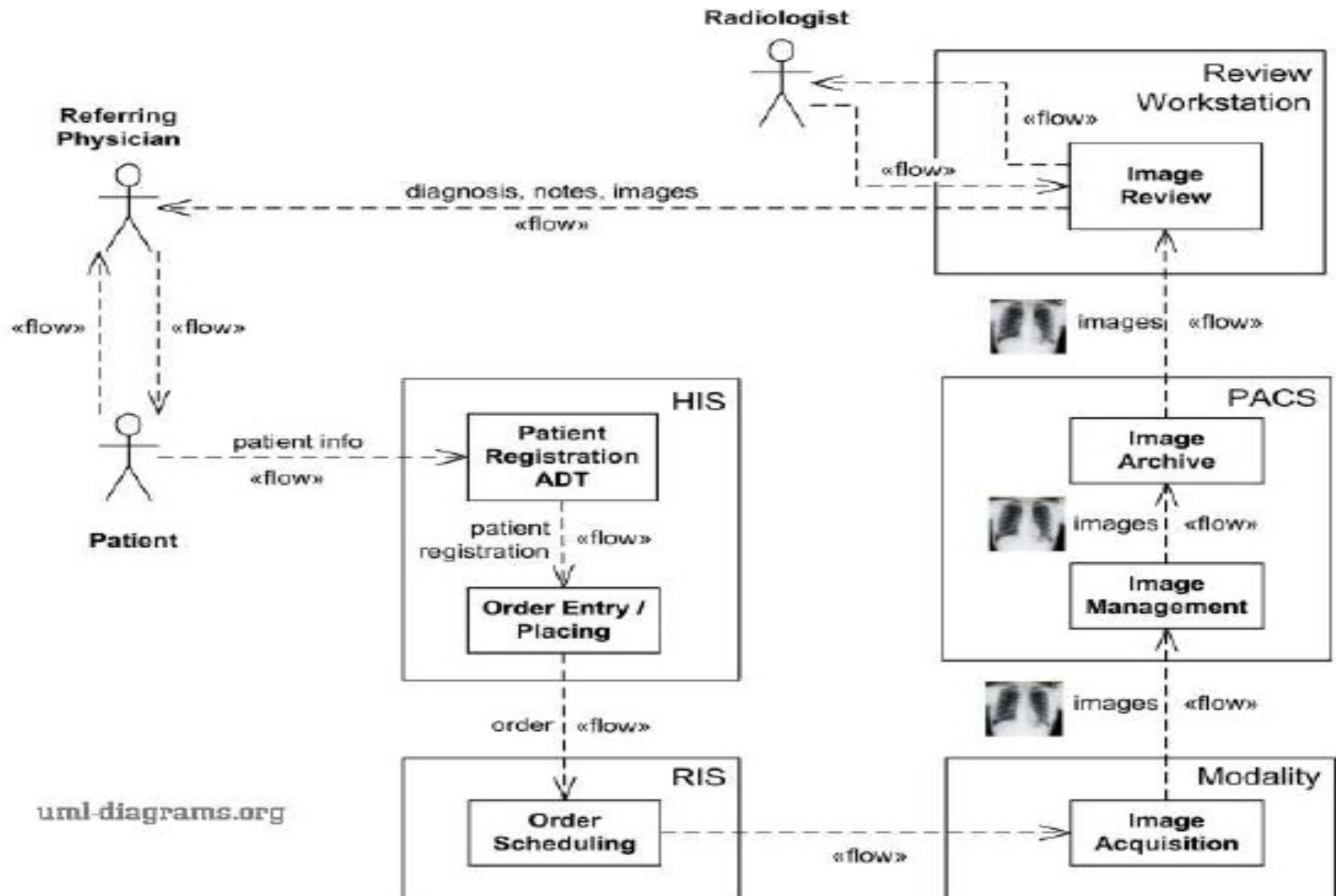
# Diagramme des flux d'informations

---

## Objectifs

- Présentation des échanges d'informations entre les entités
- Description des circulations de données

# Diagramme des flux d'informations





## **DIAGRAMME de STRUCTURE COMPOSITE**

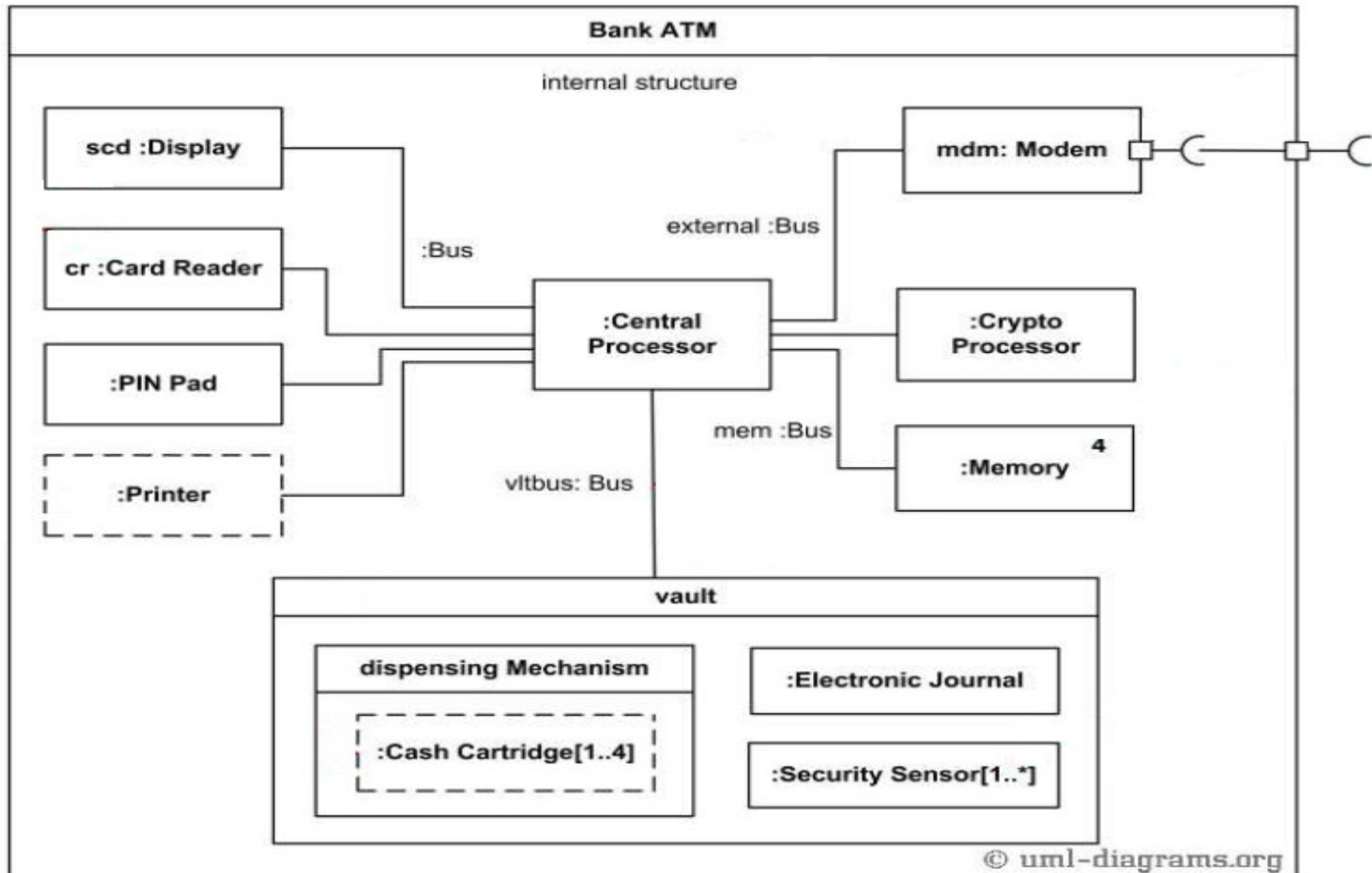
# Diagramme de structure composite

---

## Objectifs

- Diagramme de structure interne
- Interaction avec l'environnement via des ports
- Comportement collaboratif

# Diagramme de structure composite



## **DIAGRAMME de VUE d'ENSEMBLE des INTERACTIONS**

# Diagramme de vues d'ensemble des interactions

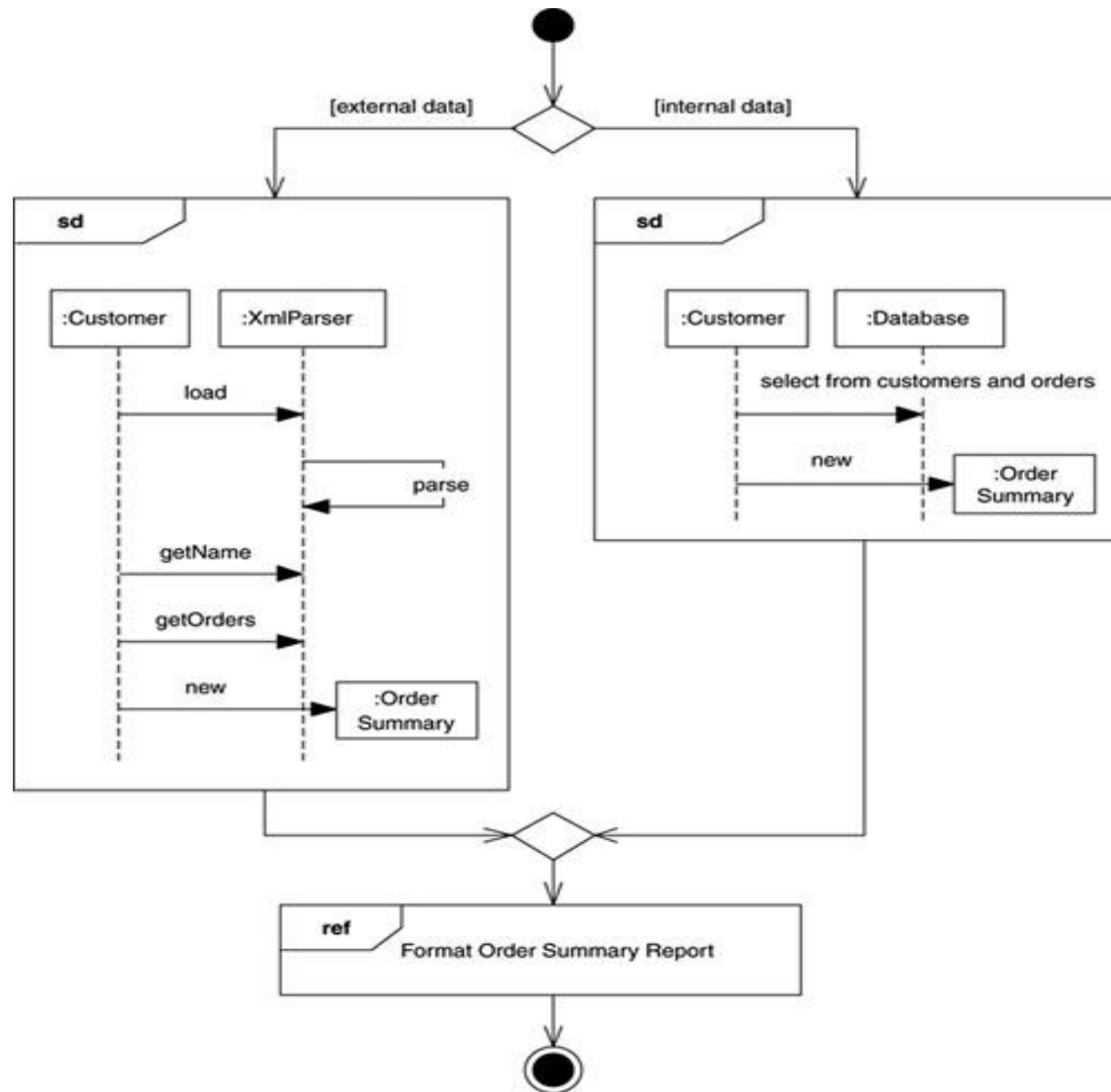
---

## Objectif

- Combinaison des diagrammes d'activités et des diagrammes de séquences

*Nouveauté UML2 → Usage à voir ?*

# Diagramme de vues d'ensemble des interactions



## DIAGRAMME de TIMING

# Diagramme d'objets

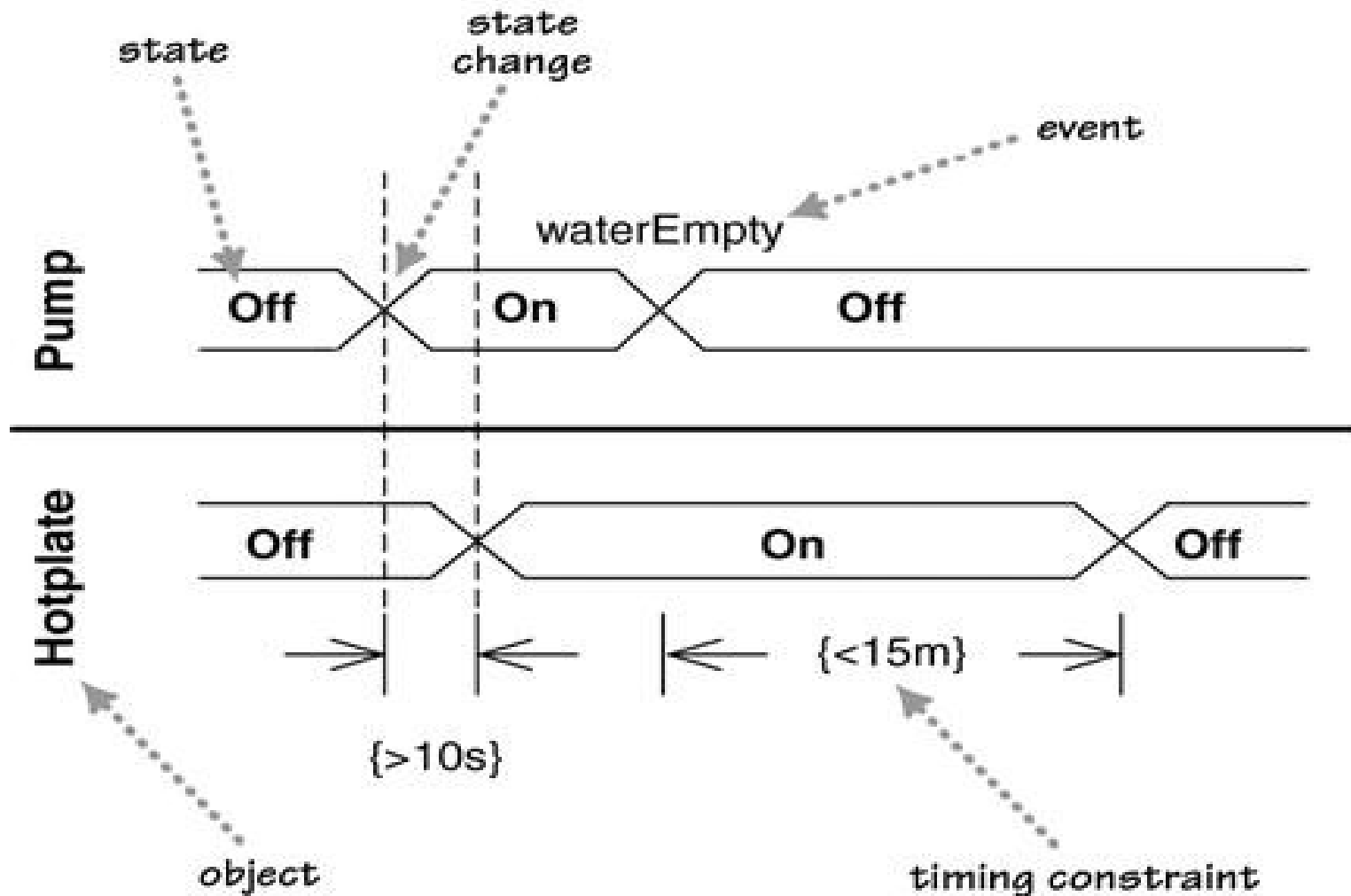
---

## Objectif

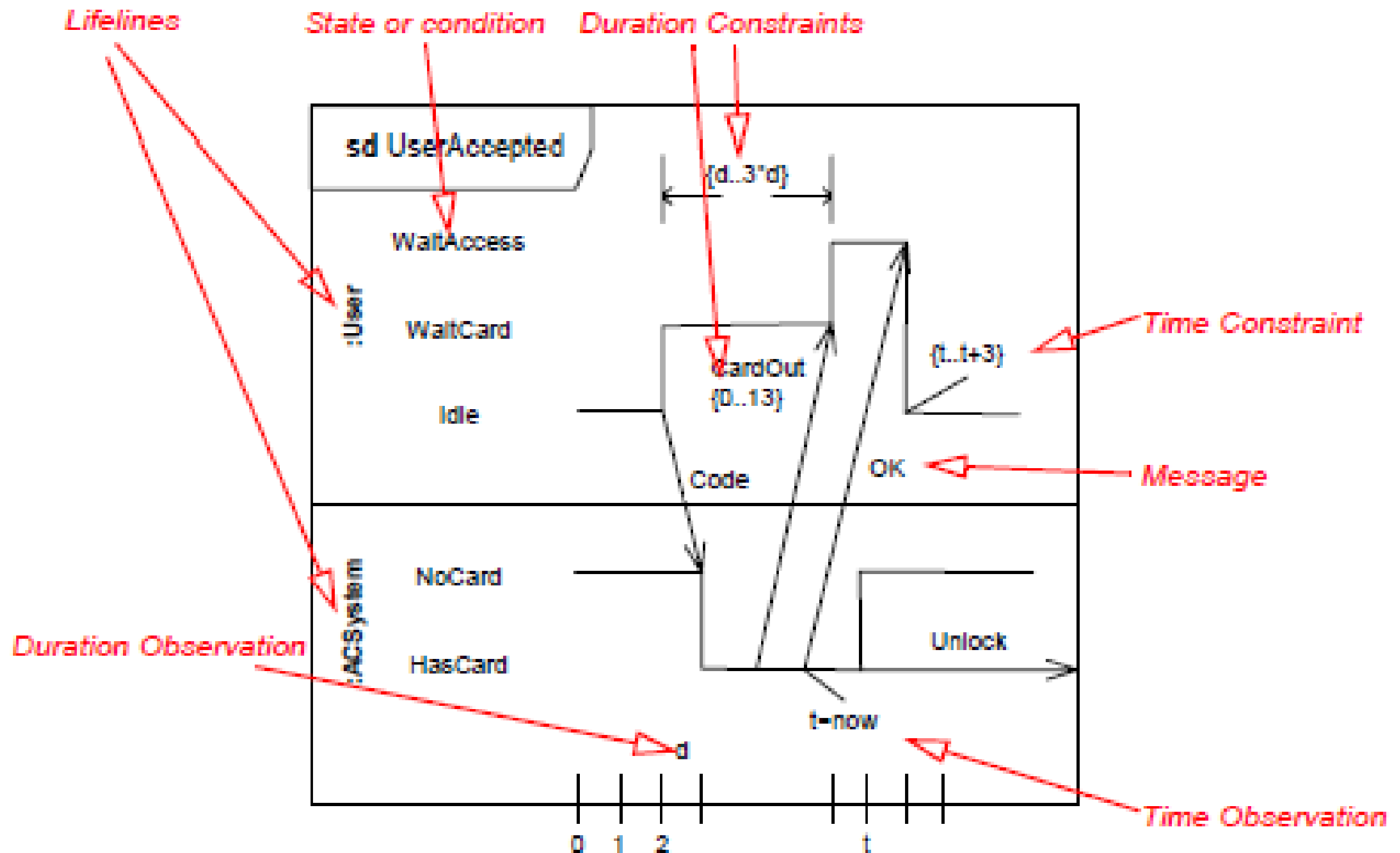
- Description temporelle précise
- Représentation linéaire (échelle)



# Diagramme de timing



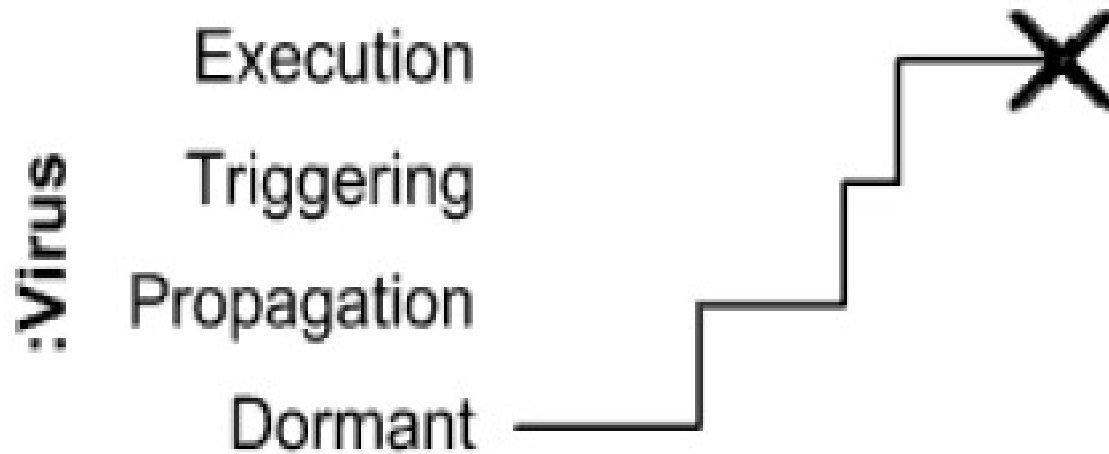
# Diagramme de timing



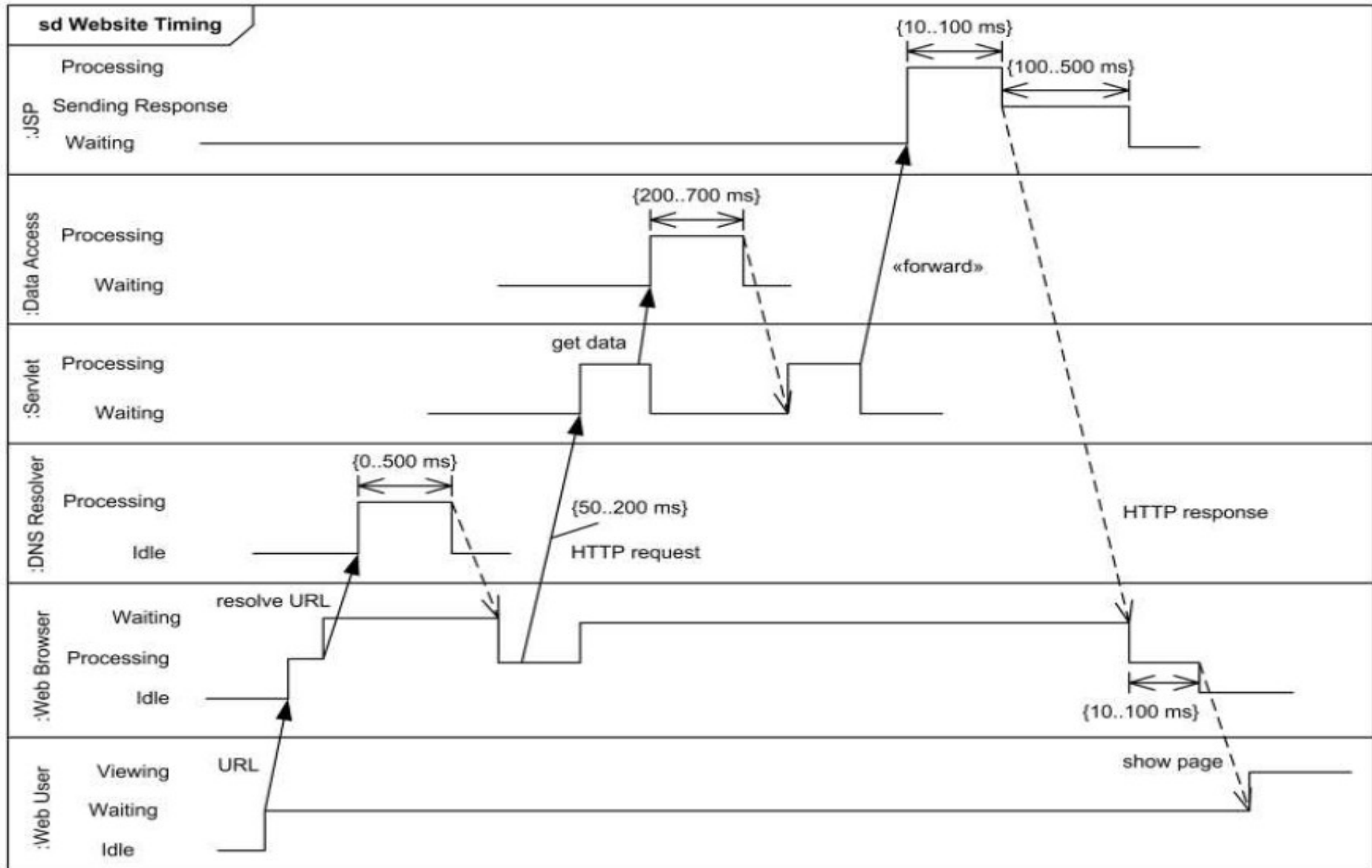
# Diagramme de timing

## Objectif

- Evolution des états
- Destruction



# Diagramme de timing



UML

---

# DIAGRAMMES

## Complémentaires

# Complément

---

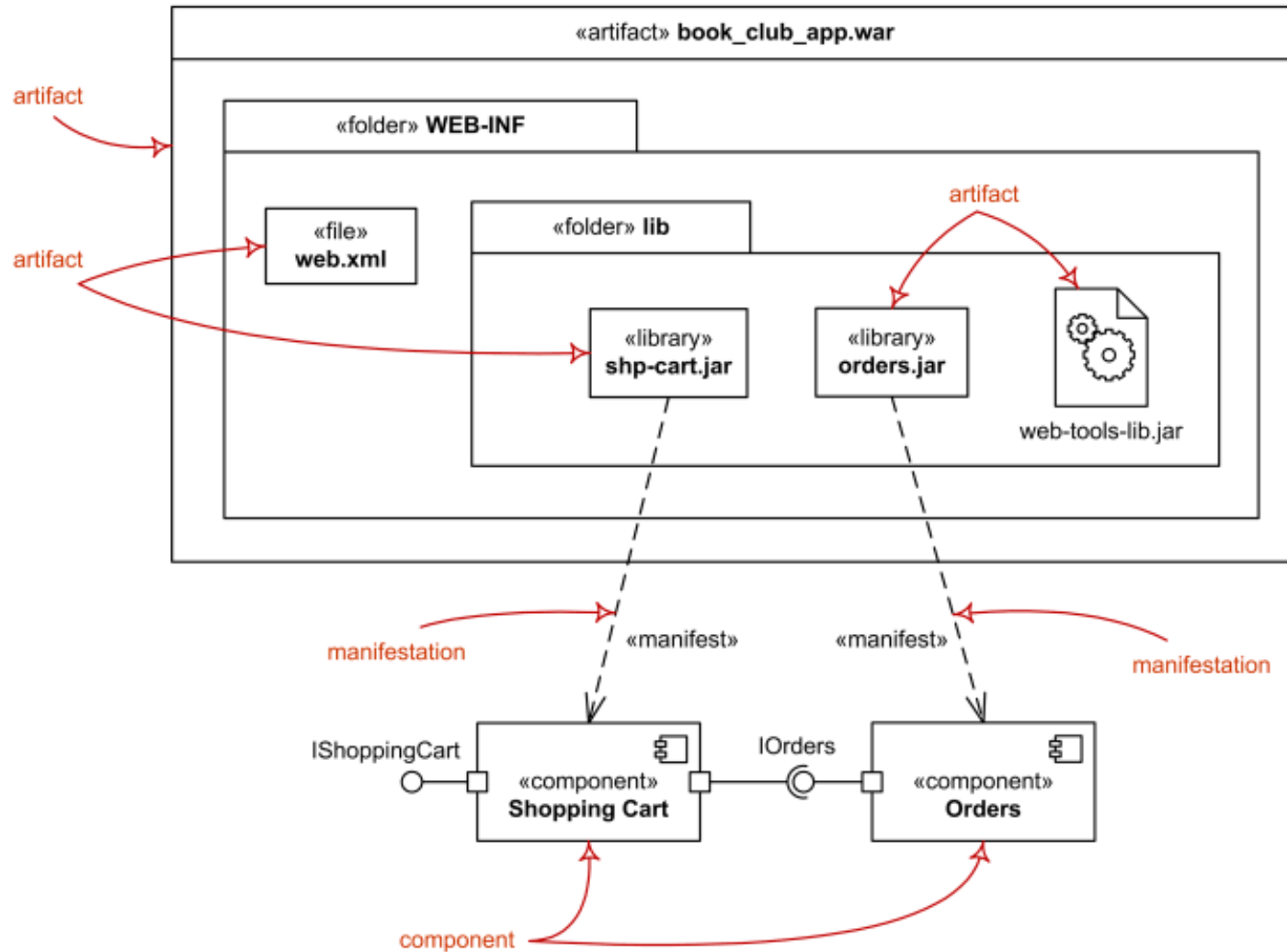
## Diagrammes non définis dans UML :

"Manifestation diagram"

but : Montrer l'implémentation de composants par des artefacts, et la structure interne des artefacts

=> Utiliser un diagramme de composants ou de déploiement

# Compléments



# Complément

---

## Diagrammes non définis dans UML :

"Specification Level Network Architecture"

but : Montrer l'architecture réseau de l'application

=> Utiliser un diagramme de déploiement