

Centre universitaire  
Abdelhafid Boussouf  
Mila

Faculté des sciences et de la  
technologie

Département math et  
informatique



# Génie logiciel

## Chapitre 6

Autres notions et diagrammes UML

Mme. S.HEDJAZ

# Autres notions et diagrammes UML

01

Diagramme de package

02

Diagramme de déploiement

03

Diagramme composant

04

.....





# Diagramme de package

## 1- Différentes façons de subdiviser un système:

- – Un système distribué est divisé en clients et serveurs
- – Un système est divisé en sous-systèmes
- – Un sous-système peut être subdivisé en paquetages
- – Un paquetage est composé de classes
- – Une classe est composée de méthodes



# Diagramme de package

## 2- Diagramme de package:

- UML 2 décrit les paquetages à l'aide d'un diagramme spécifique. Un paquetage est un regroupement d'éléments de modélisation : classes, composants, cas d'utilisation, autres paquetages, etc.
- Les paquetages d'UML sont utiles lors de la modélisation de systèmes importants pour en regrouper les différents éléments. Ce regroupement structure ainsi la modélisation
- Un paquetage est représenté par un dossier (voir figure ci-dessous). Il constitue un ensemble d'éléments de modélisation UML



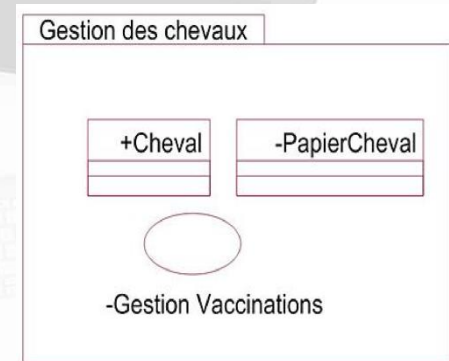
paquetage



# Diagramme de package

## 2- Diagramme de package:

- Le contenu d'un paquetage est décrit par un diagramme de paquetage. Celui ci représente les différents éléments du paquetage avec leur propre représentation graphique. Ceux ci peuvent être des classes, des composants, des cas d'utilisation, d'autres paquetages, etc.
- Chaque élément inclus dans un paquetage peut être accessible à l'extérieur ou encapsulé à l'intérieur de celui ci. Par défaut, un élément est accessible à l'extérieur.





# Diagramme de package

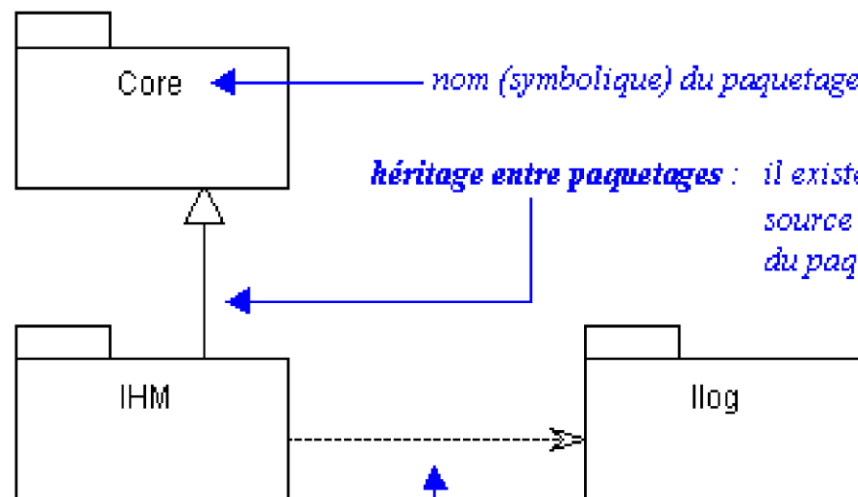
## 2- Diagramme de package:

- Pour qu'un paquetage puisse exploiter les éléments d'un autre paquetage, il existe deux types d'associations :
- L'association d'importation consiste à amener dans le paquetage de destination un élément du paquetage d'origine. L'élément fait alors partie des éléments visibles du paquetage de destination.
- L'association d'accès consiste à accéder, depuis le paquetage de destination, à un élément du paquetage d'origine. L'élément ne fait alors pas partie des éléments visibles du paquetage de destination. Il n'est possible d'importer ou d'accéder à un élément que si celui-ci est spécifié visible dans le paquetage d'origine.



# Diagramme de package

## 3- Exemple diagramme de package:



nom (symbolique) du paquetage

**héritage entre paquetages** : il existe (au moins) un élément du paquetage source qui spécialise (au moins) un élément du paquetage destination.

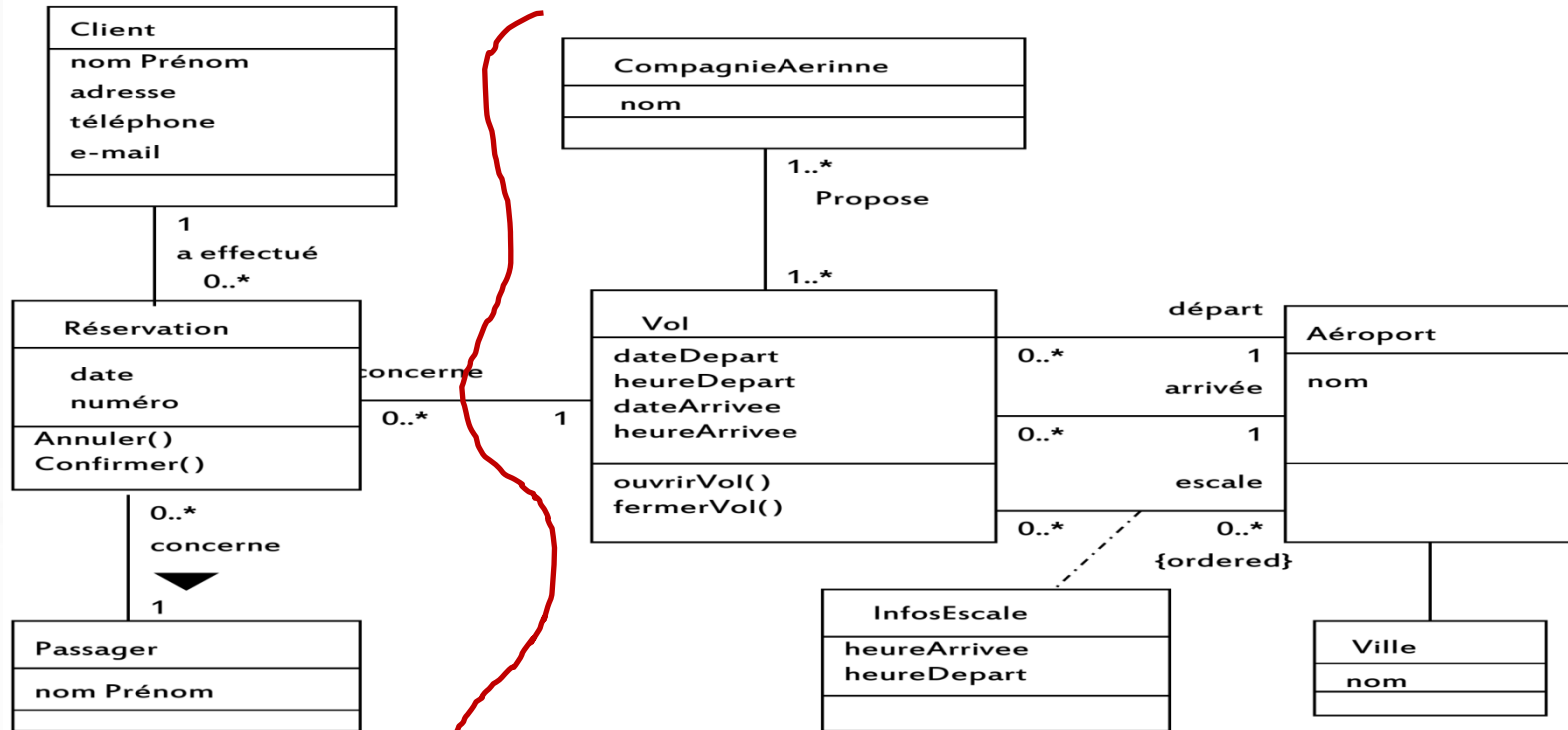
**dépendance entre paquetages**

- au moins un élément du paquetage source utilise les services d'au moins un élément du paquetage destination
- représente une relation d'obsolescence entre les éléments des deux paquetages.



# Diagramme de package

## 4- Exemple diagramme de package:







## Diagramme de déploiement

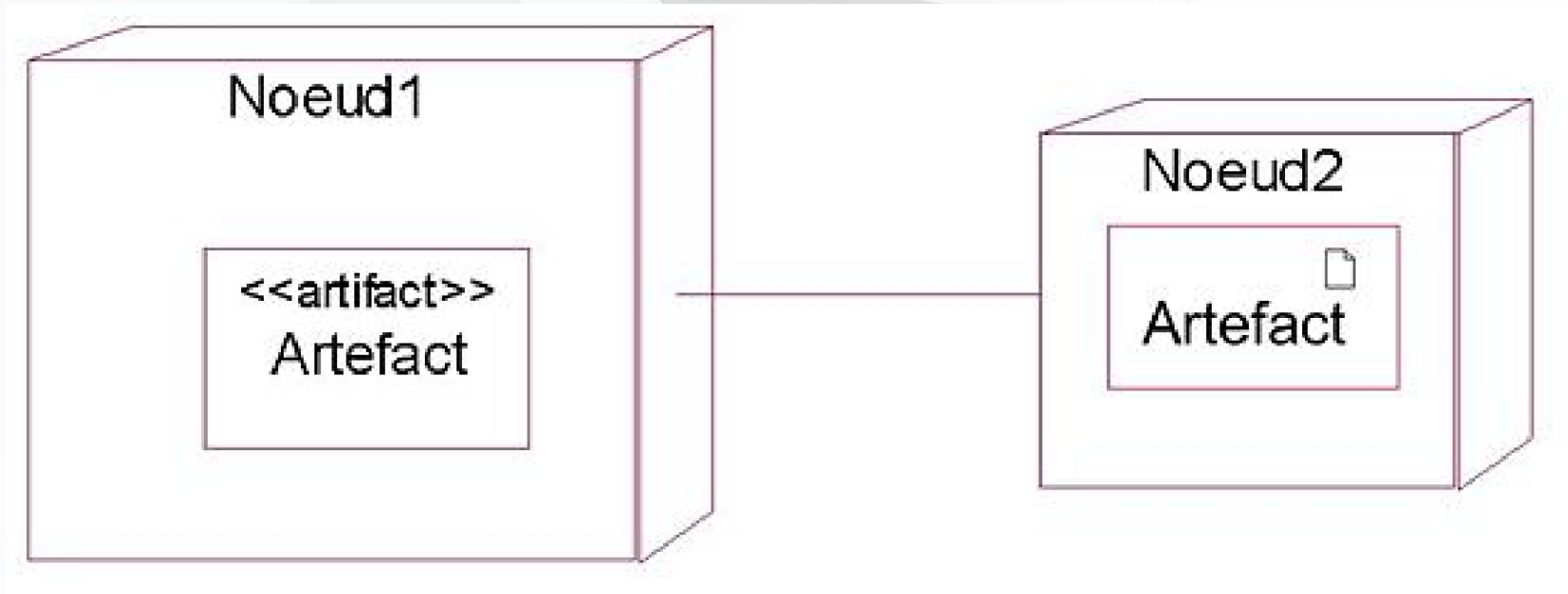
### 1- définition:

- Le diagramme de déploiement décrit l'architecture physique du système. Celui-ci est composé de nœuds. Un nœud est une unité matérielle capable de recevoir et d'exécuter du logiciel. La plupart des nœuds sont des ordinateurs. Les liaisons physiques entre nœuds peuvent également être décrites dans le diagramme de déploiement. Elles correspondent aux branches du réseau.
- Les nœuds contiennent des logiciels sous leur forme physique. Cette dernière est nommée artefact. Un fichier exécutable, une bibliothèque partagée ou un script sont des exemples de forme physique de logiciel.
- Les composants qui constituent l'architecture logicielle du système sont représentés dans le diagramme de déploiement par un artefact qui est souvent un exécutable ou une bibliothèque partagée. La représentation graphique des nœuds, de leurs liens et des artefacts qu'ils contiennent est illustrée à la figure ci-dessous.



## Diagramme de déploiement

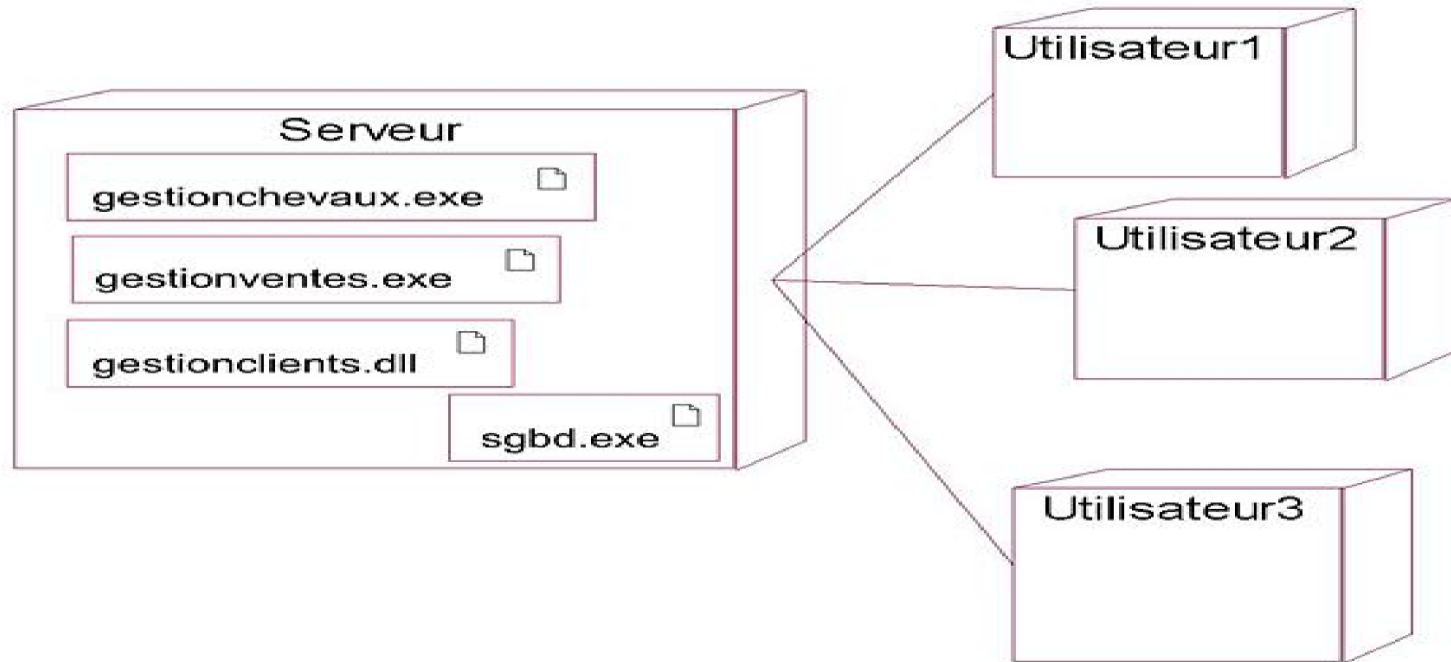
### 1- Définition:





## Diagramme de déploiement

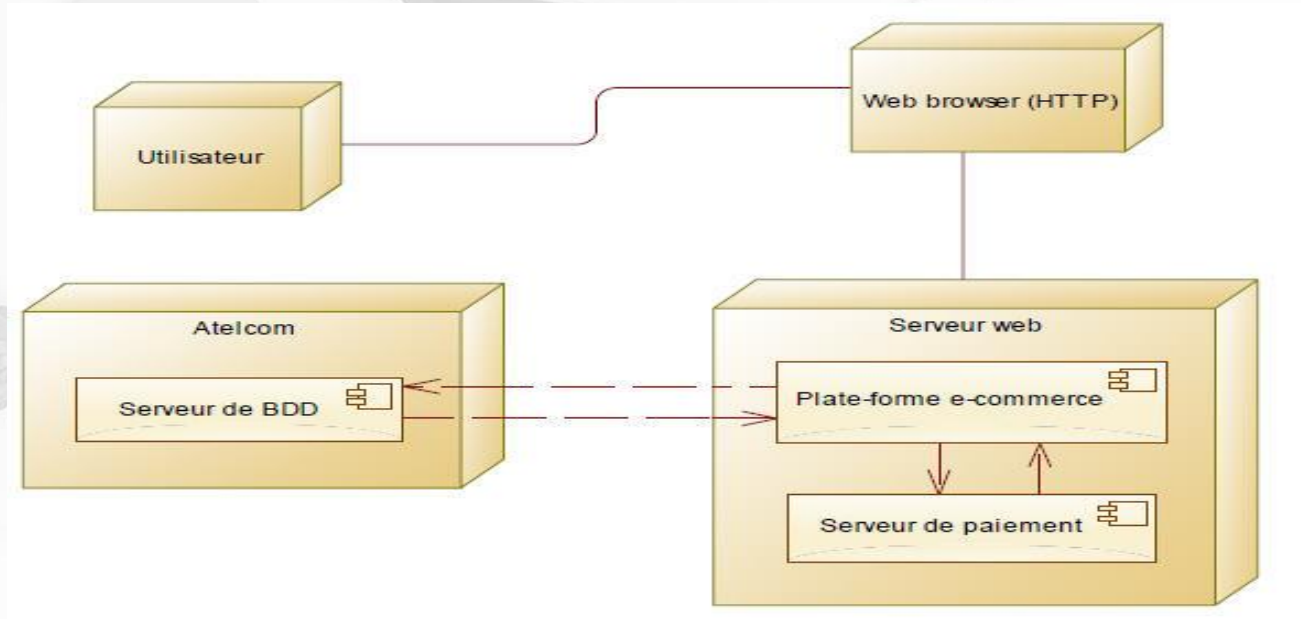
### 2- Exemple1:





## Diagramme de déploiement

### 3- Exemple2:





## Diagramme de composant

### 1- définition:

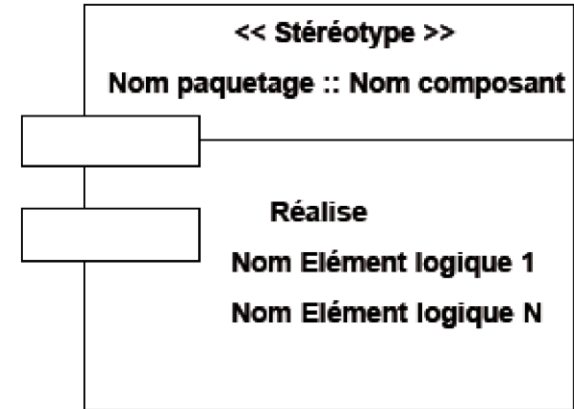
- Les diagrammes de composants représentent l'architecture logicielle du système. Les diagrammes de composants permettent de décrire l'architecture physique et statique d'une application en terme de fichiers sources, librairies, exécutables, etc. Ils montrent les dépendances entre les composants de l'application
- Un composant est l'implémentation physique logicielle d'un ensemble d'éléments logiques (classe ou collaboration)
- Un composant propose un ensemble d'interfaces qu'il se doit de respecter
- Un composant peut être un exécutable, une librairie, un fichier source , un fichier de configuration, un document.
- En java, un composant peut être: Fichiers sources (.jar), JavaBeans, ...etc.



## Diagramme de composant

### 1- définition:

- Un composant est représenté par un rectangle avec des onglets
- Le nom du composant peut être précédé du nom du paquetage qui le contient
- Il est possible de développer le composant de façon à faire apparaître le nom des éléments logiques qu'il implémente

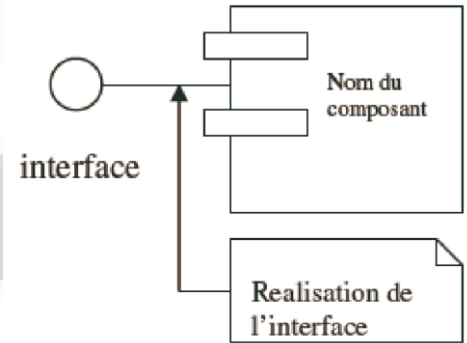
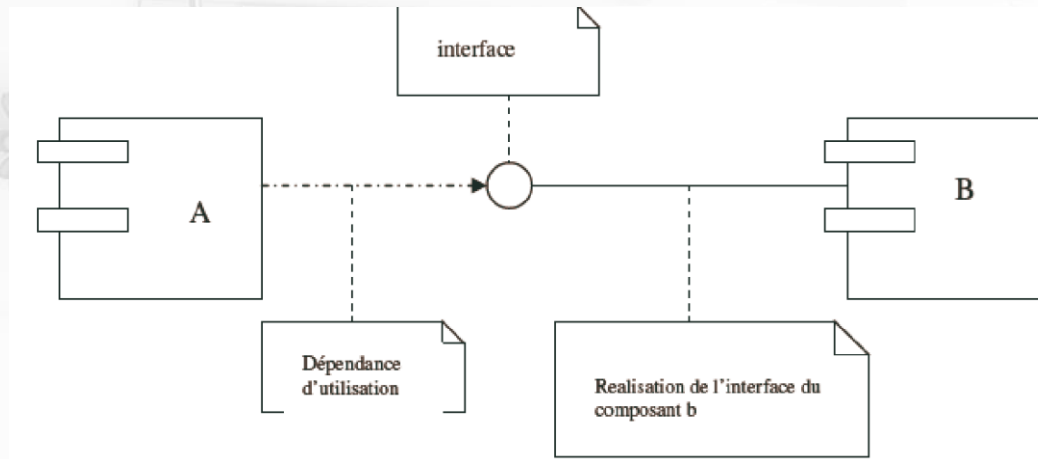




## Diagramme de composant

### 2- Les interfaces d'un composant:

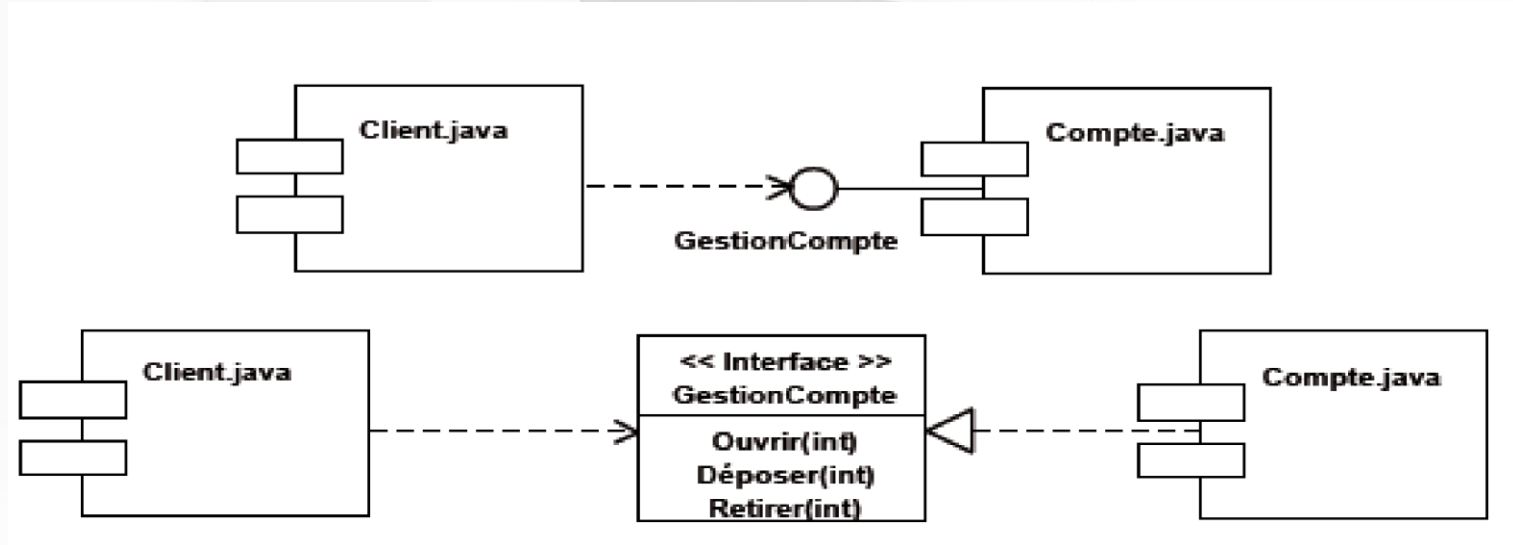
- En UML, un composant est représenté par :
- Un composant A peut faire appel à l'interface d'un composant B. On modélise par :



# 03

## Diagramme de composant

### 3- Exemple1:





# Bibliographies

- **Uml 2 pratique de la modélisation**, Benoît Charroux, Yann Thierry-Mieg, Aomar Osmani  
Ni <https://fr.slideshare.net/nassimamine3994/uml-2-pratique-de-la-modlisation>
- **Uml 2 par la pratique**, Pascal roques
- **Les cahiers du programmeur**, Pascal roques
- **Uml en action**, Pascal roques
- .....