

CHAPITRE 7:

DIAGRAMME DE COMPOSANTS ET DIAGRAMME DE DÉPLOIEMENT

Dr. Ilhem ABDELHEDI ABDELMOULA

Email: ilhemabdelmoula13@gmail.com

Université de Carthage

Enicarthage - École Nationale des Ingénieurs à Carthage

Département informatique

Niveau : 2^{ème} année INFO ING / 1^{ère} Année INFO ING

Semestre : 2

Année universitaire: 2019 - 2020



MODÉLISER LES ASPECTS PHYSIQUES DU SYSTÈME

- **Diagramme de Composants** : se focalise sur l'organisation et les dépendances entre un ensemble de composants
- **Diagrammes de Déploiement** : se focalise sur la **configuration** en temps d'exécution des nœuds de traitement et de composants qui sont actifs

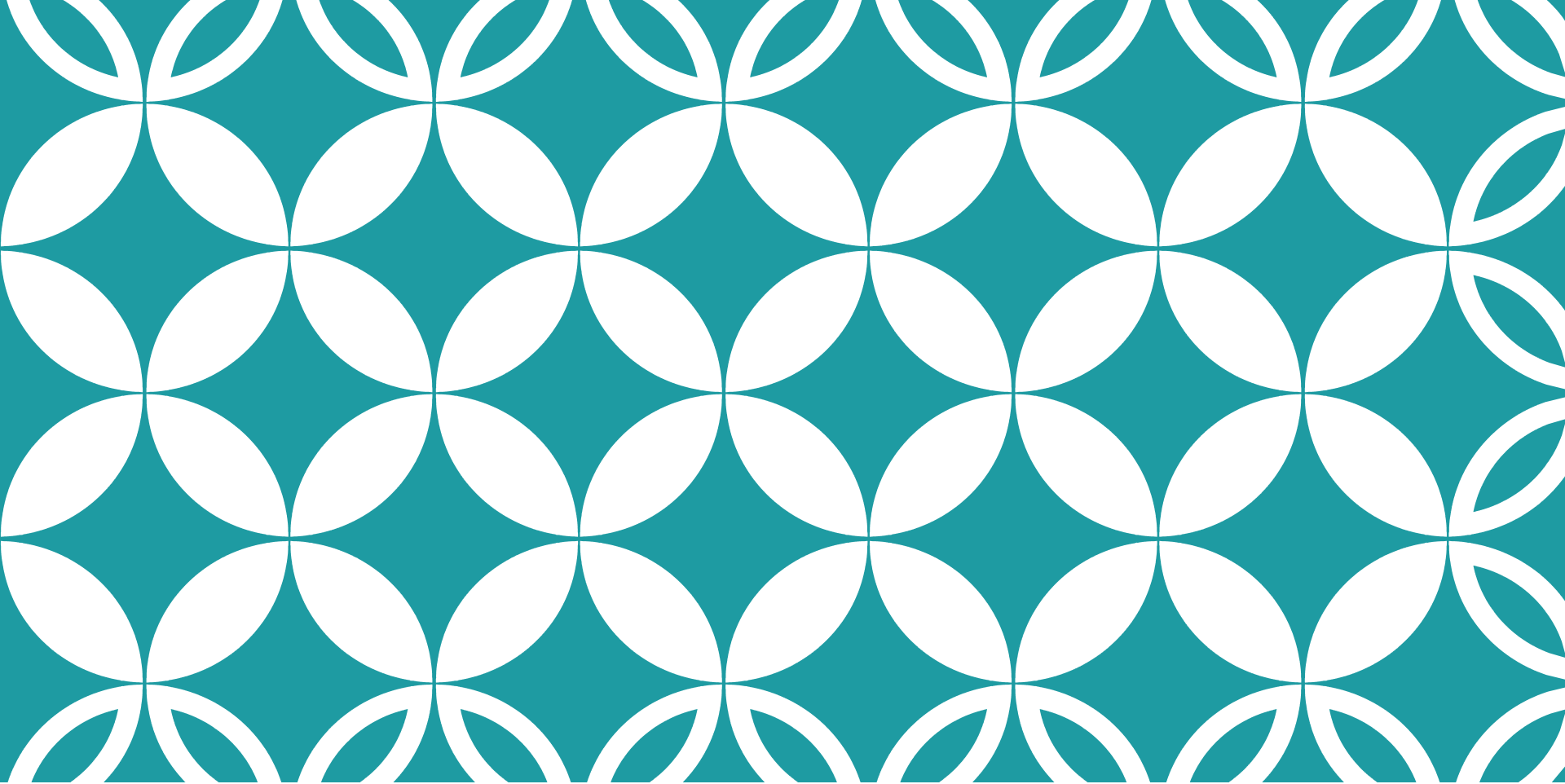


DIAGRAMME DE COMPOSANTS | PARTIE I

INTRODUCTION

Dans le monde de bâtiment, tout ce qui est proposé par l'architecte (plan) constitue une vue logique :

visualiser, spécifier, documenter

Pour **la construction**, on utilise des composants physiques du monde réel :

murs, fenêtres, portes, ...

LES COMPOSANTS

En UML, dans le modèle logique :

*=> visualiser, spécifier et documenter la **structure** et le comportement des objets*

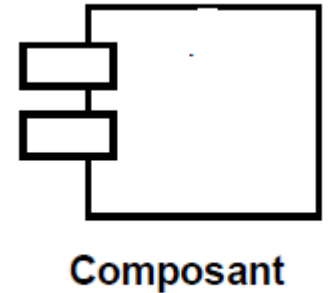
La construction se base sur les **composants** du monde réel de l'ordinateur :

Fichiers, tables, librairies, ...

UTILITÉ DU DIAGRAMME DE COMPOSANTS

- Montre la **mise en œuvre physique** des modèles de la vue logique **dans l'environnement de développement**
- Décrit **l'architecture physique et statique** d'une application en terme de composants :
 - code source, bibliothèques, fichiers exécutables, ...
- Décrit les composants et leurs dépendances dans l'environnement de réalisation
- Spécifie les composants, les interfaces et les relations (dépendance, généralisation, association, réalisation).

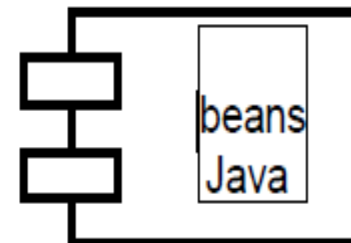
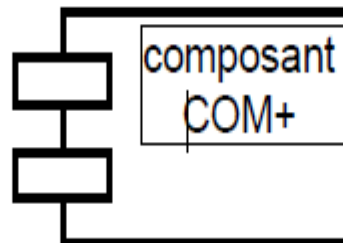
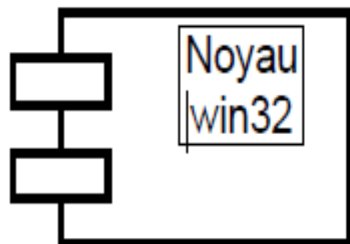
DÉFINITION D'UN COMPOSANT



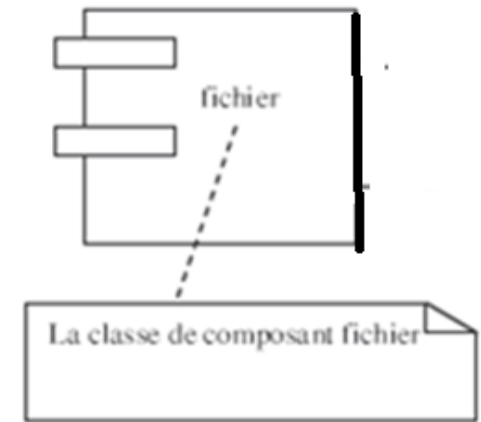
Partie physique (implémentable) du système :

- code, script, fichier de commandes, fichier de données, une table...

Les composants ont des noms distincts et peuvent être organisés en paquetages



COMPOSANTS – INSTANCES



Pour une instanciation de la Classe de composant Fichier, on aurait :



L'objet f1 est une instance de
la classe Fichiers



Un objet générique qui est une
instance de la classe Fichiers

COMPOSANTS – STÉRÉOTYPES

Spécifient un composant qui désigne:

- « *executable* »: un programme pouvant s'exécuter sur un nœud
- « *library* » : une bibliothèque statique ou dynamique
- « *table* »: une table de base de données
- « *file* » : un fichier contenant du code source ou des données
- « *document* » : un document

On peut indiquer une catégorie de composant à l'aide de Stéréotype situé au dessus de nom du composant en UML on dispose de :

<<document>>

<<executable >> ou << exe>>

<<fichier>>

<<table>>

<<bibliothèque>>

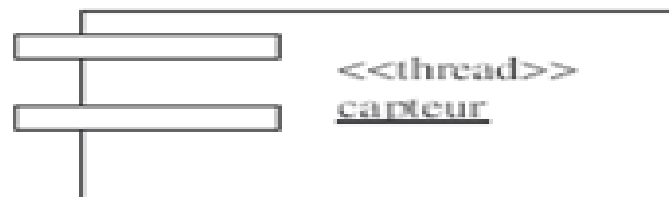
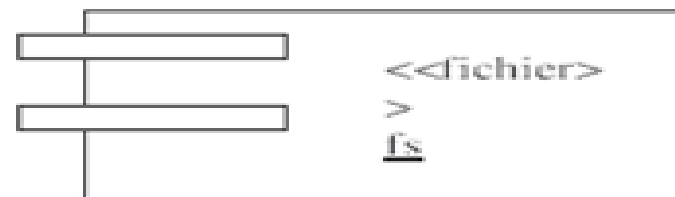
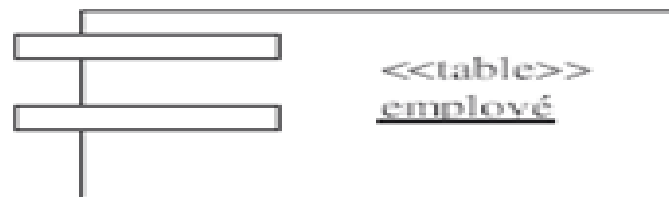
<<DLL>>

et d'icône



pour les BD

exemple :

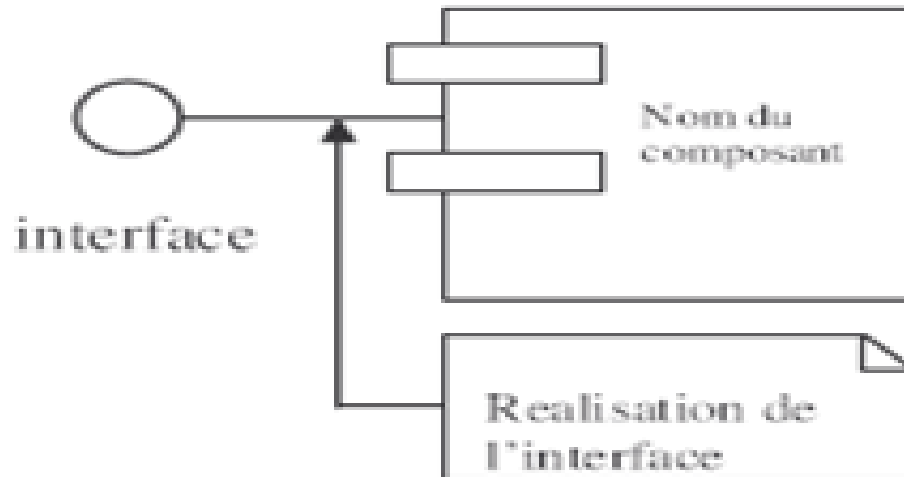


RÉALISATION D'INTERFACES

« *INTERFACE EXPORTÉE* »

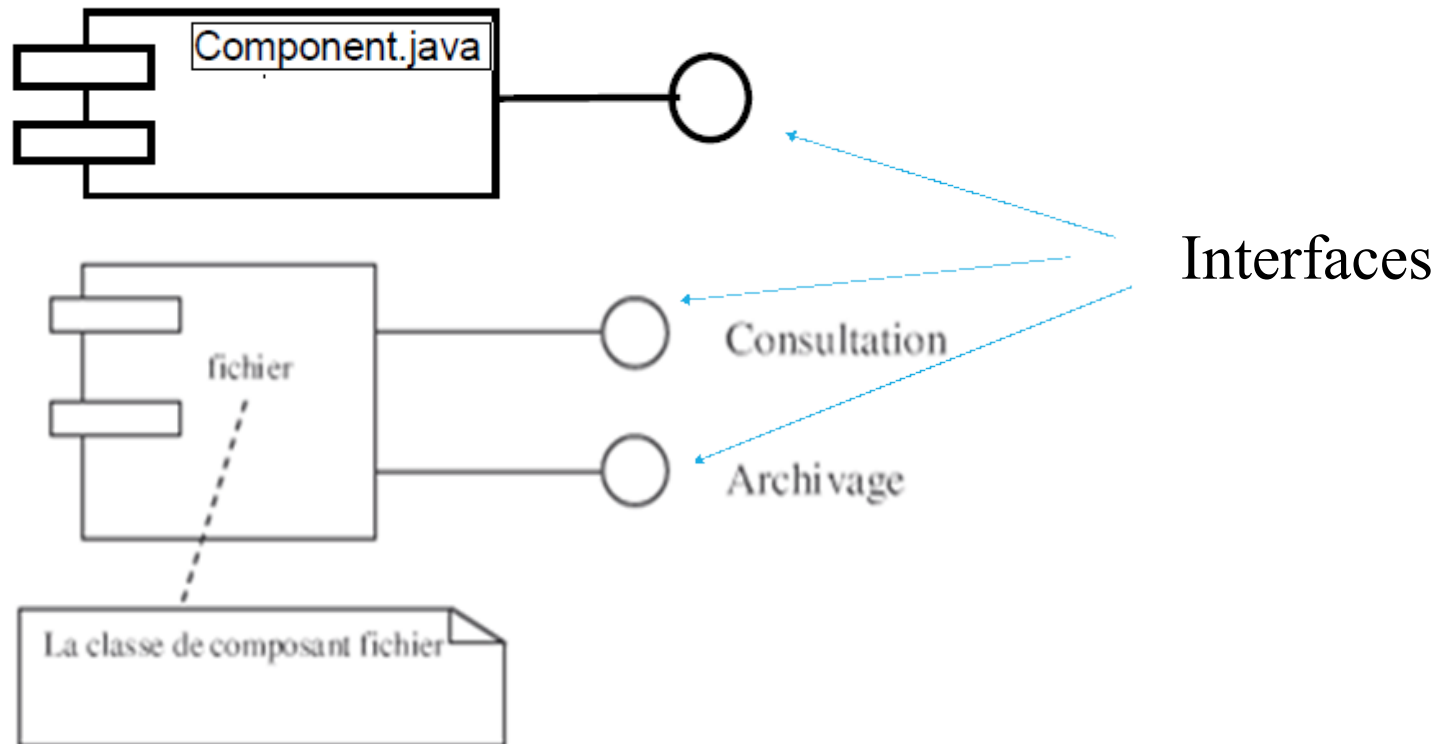
Un composant peut réaliser un ensemble *d'interfaces* qui définissent les services offerts aux autres composants.

- **Interface** : collection d'opérations utilisées pour spécifier les services d'une classe ou d'un composant



COMPOSANTS – INTERFACES

Exemples :



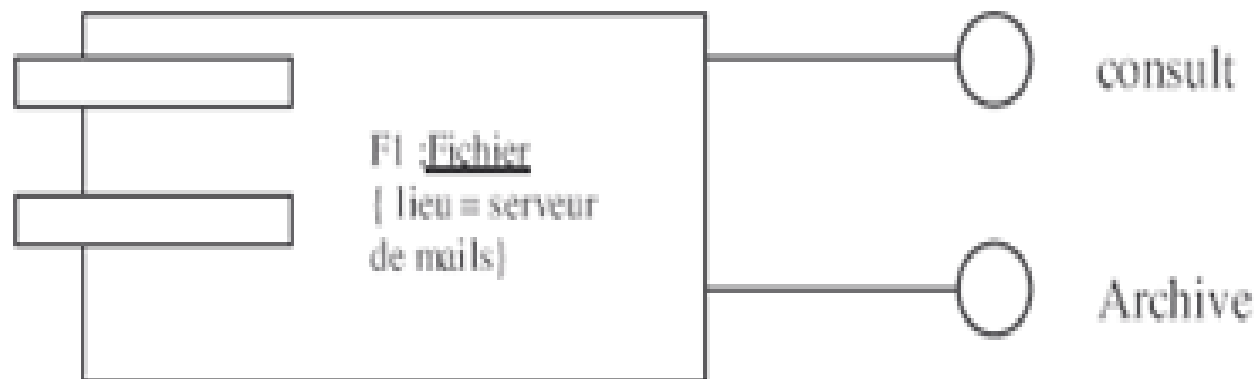
Un composant est décrit comme une sous classe d'un classificateur.

=> un composant pourra avoir

- des attributs
- des opérations

et pourra participer à des relations(association – généralisation, agrégation, composition)

on peut éventuellement noter l'endroit où se situera le composant (voir dial de déploiement).



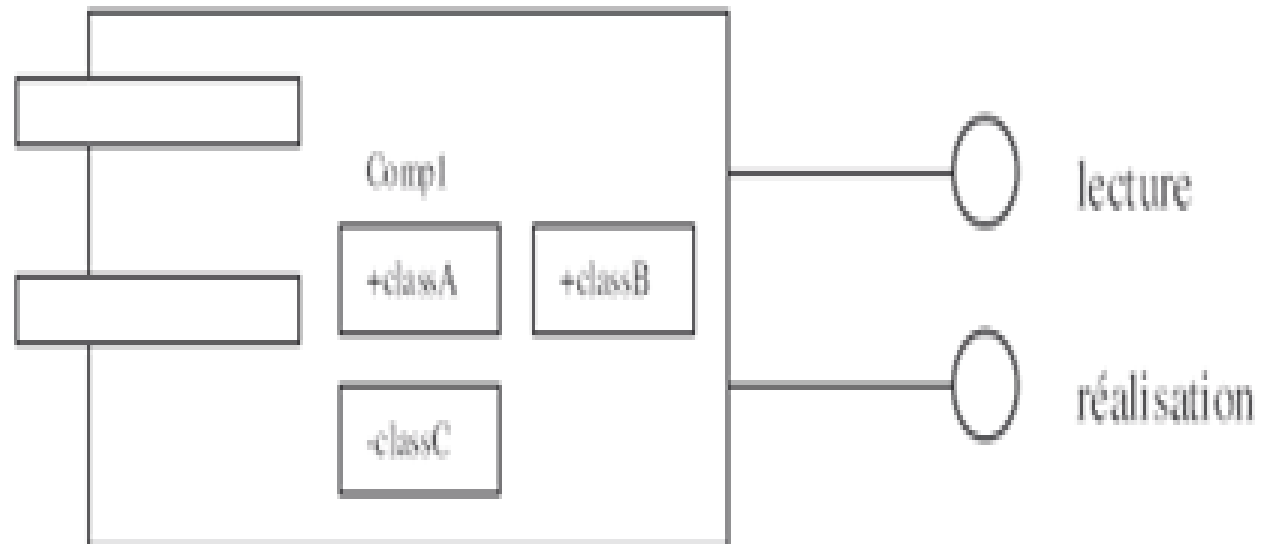
CLASSE DE COMPOSANT

Un composant qui comprend plusieurs classes peut être représenté avec ses classes par des rectangles.

Avec visibilité

+ public

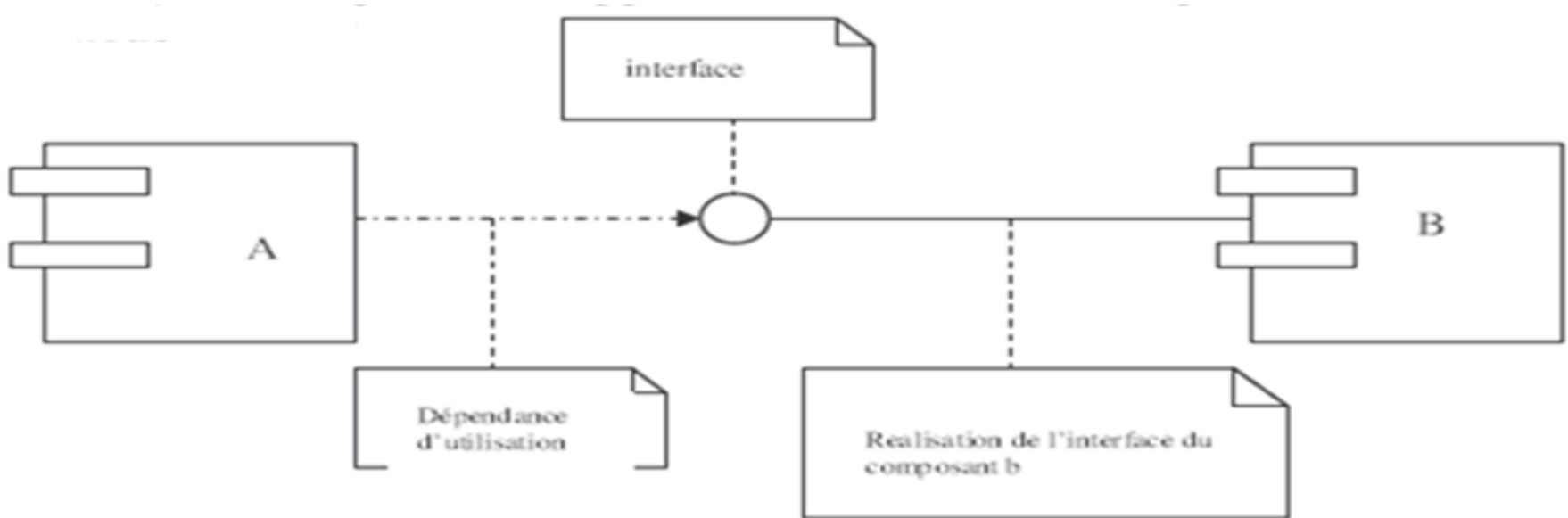
- privé



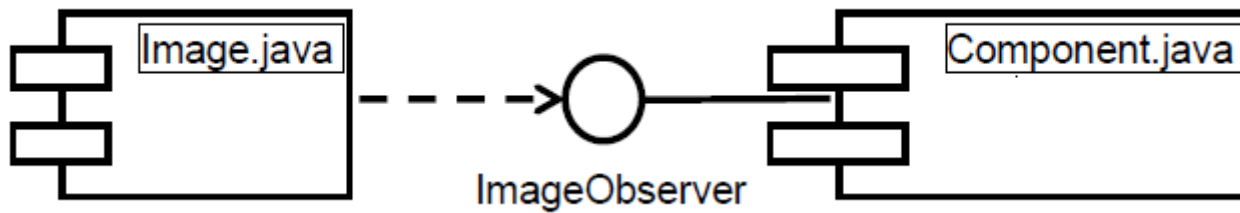
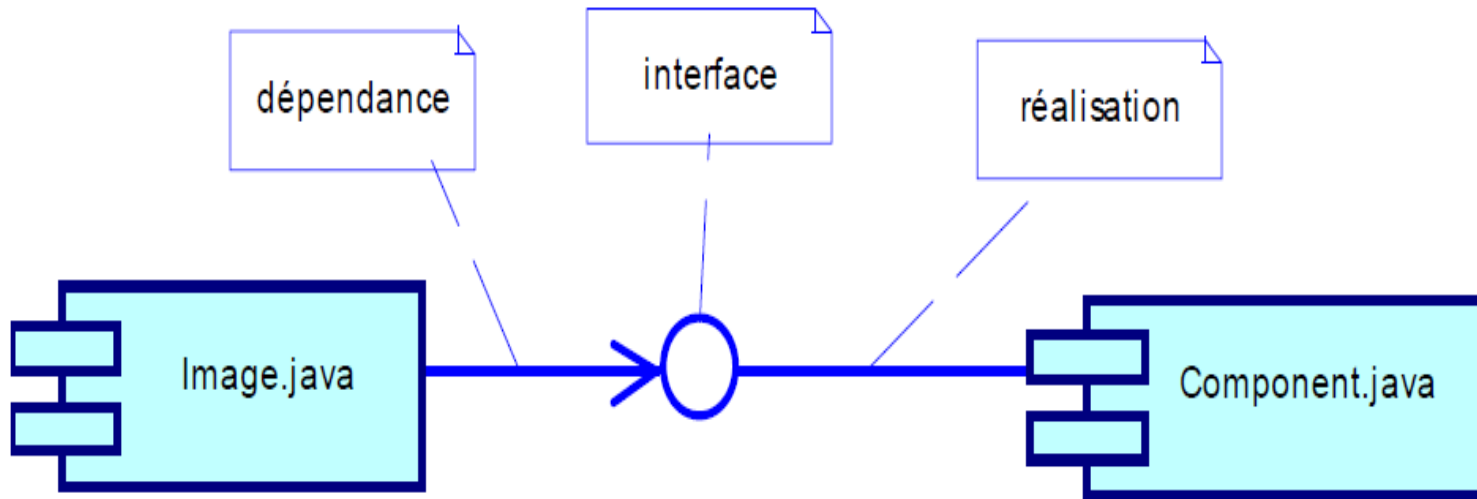
DÉPENDANCES AVEC INTERFACES

« *INTERFACE IMPORTÉE* »

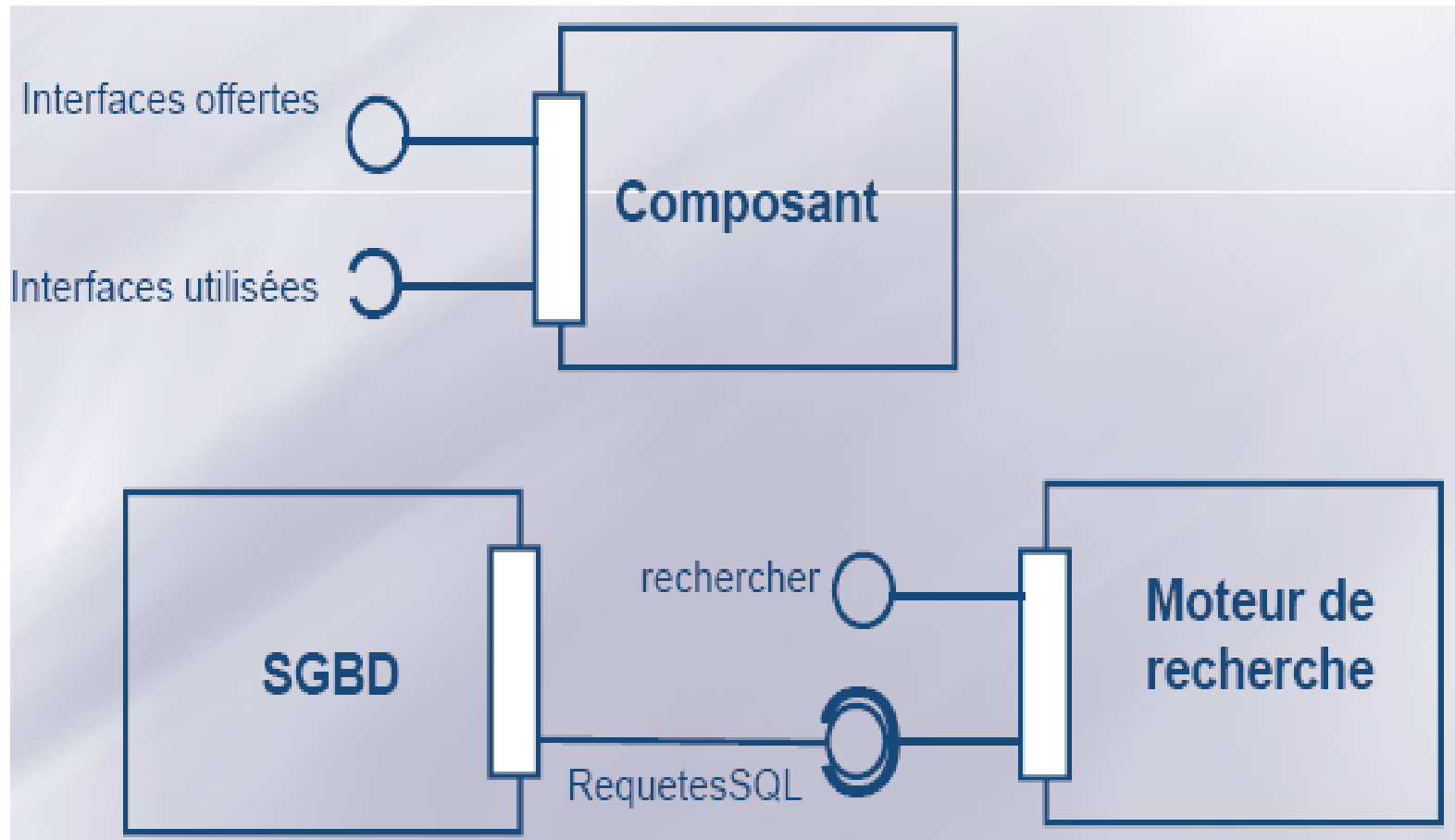
Un composant qui *utilise* les services (les interfaces)
fournis par un autre composant



EXEMPLE

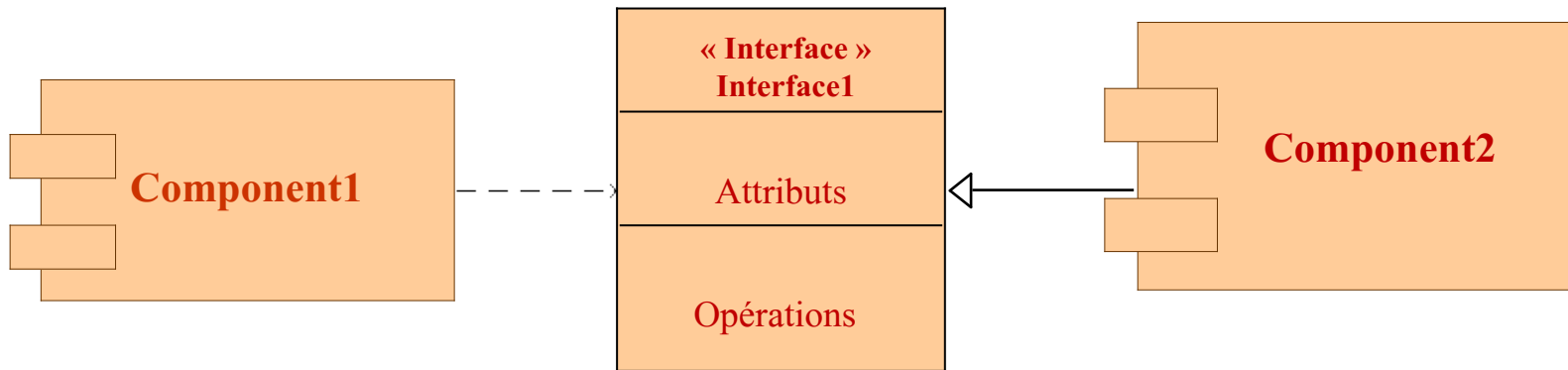


UNE AUTRE NOTATION

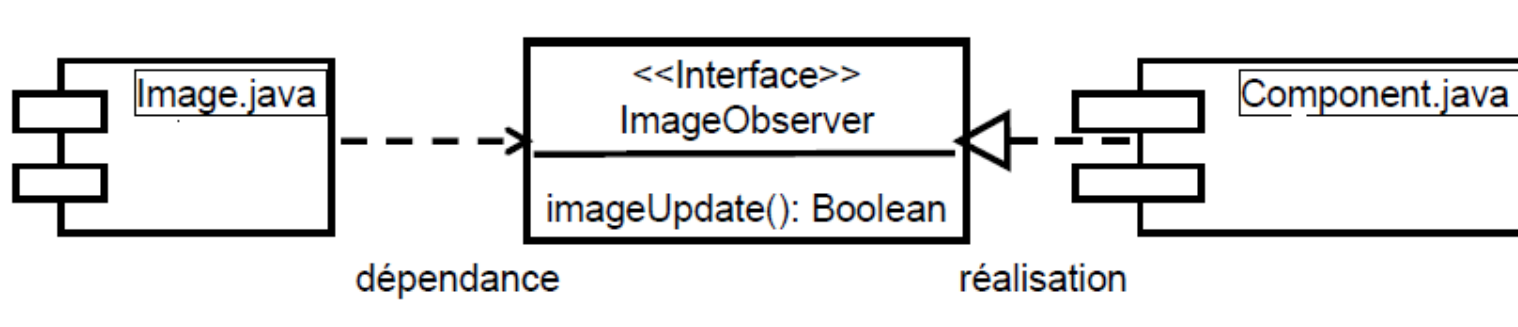


FORME EXPLICITE

Le composant 1 peut faire appel à l'interface du composant 2



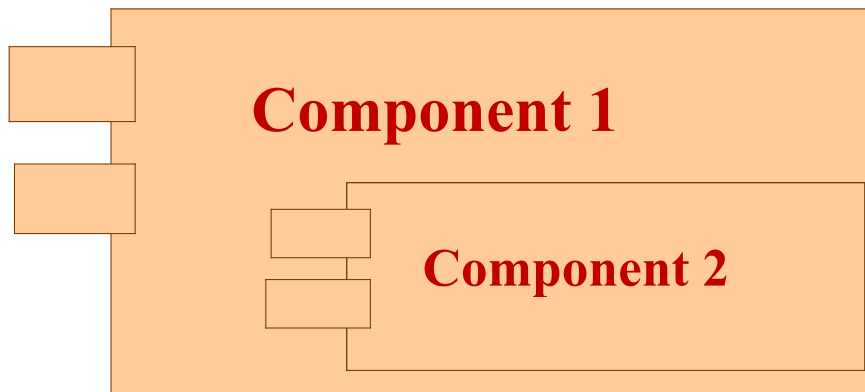
Exemple :



RELATIONS ENTRE LES COMPOSANTS

3. Contenance :

- Un composant peut être contenu dans un autre composant



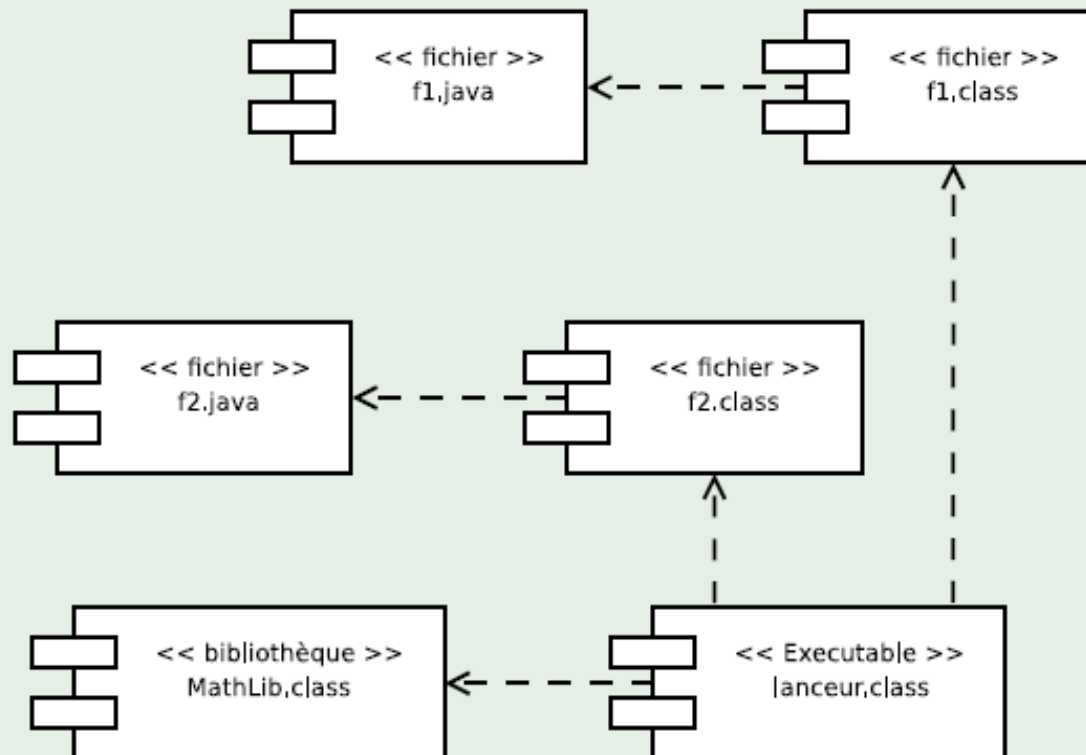
EXEMPLE

Composants d'un serveur Oracle utilisant une base de données servant de catalogue.



EXEMPLE

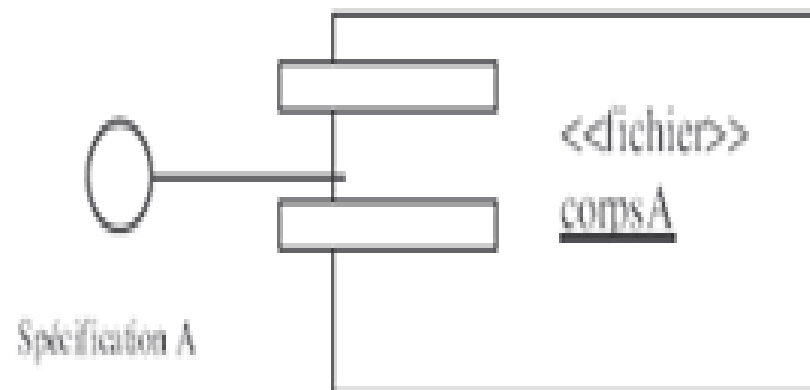
Application construite à partir de deux fichiers source et d'une bibliothèque mathématiques.



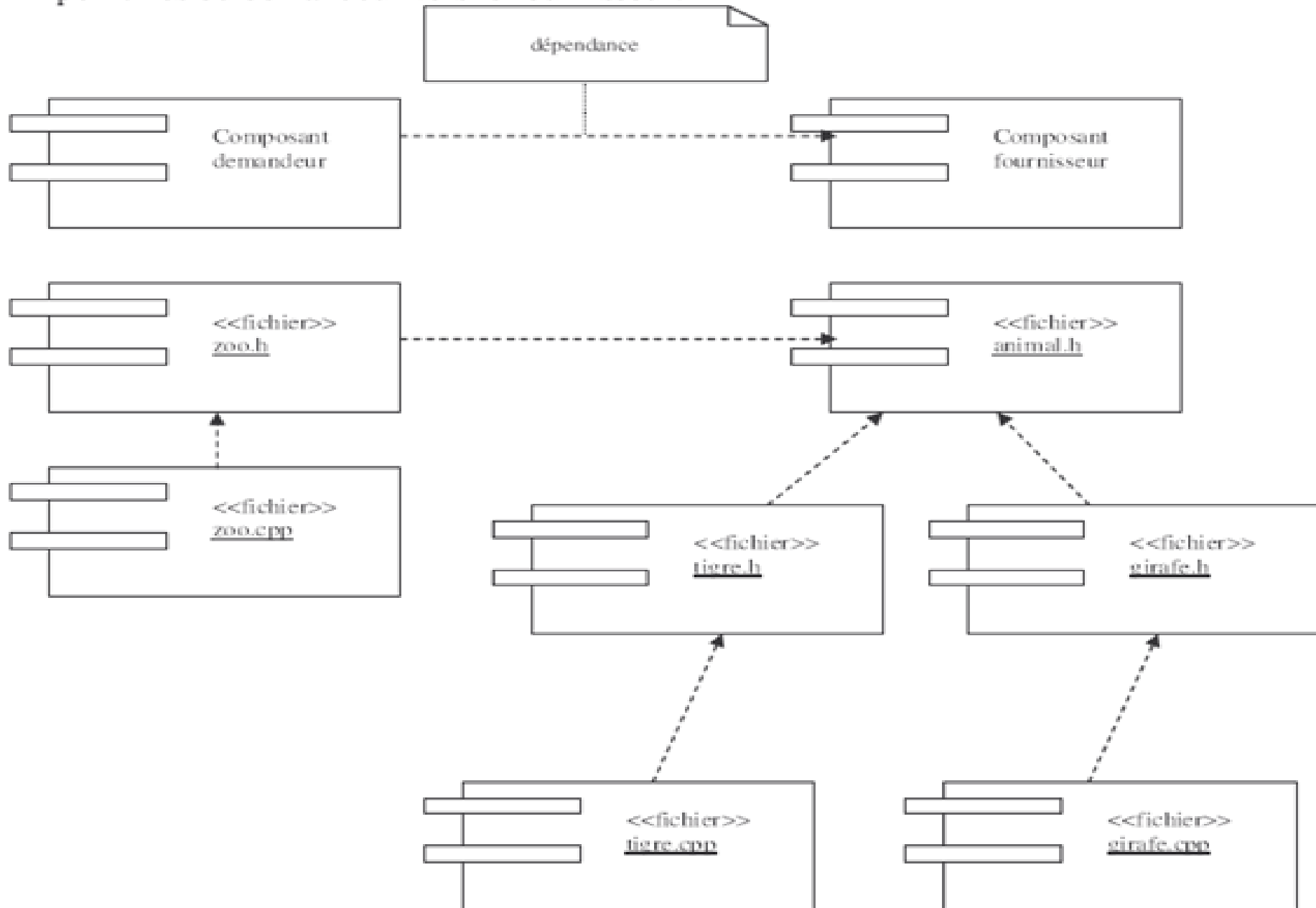
Par défaut chaque classe du modèle logique de l'application est réalisée en deux parties :

- l'interface de la classe = spécification en C++ → fichier .h
- l'implémentation = corps en C++ → fichier .cpp

que l'on peut représenter par



Sont utilisées pour indiquer que l'implémentation d'un composant fait appel à l'implémentation d'un autre composant et cela se note par une flèche en pointillés du demandeur vers le fournisseur.



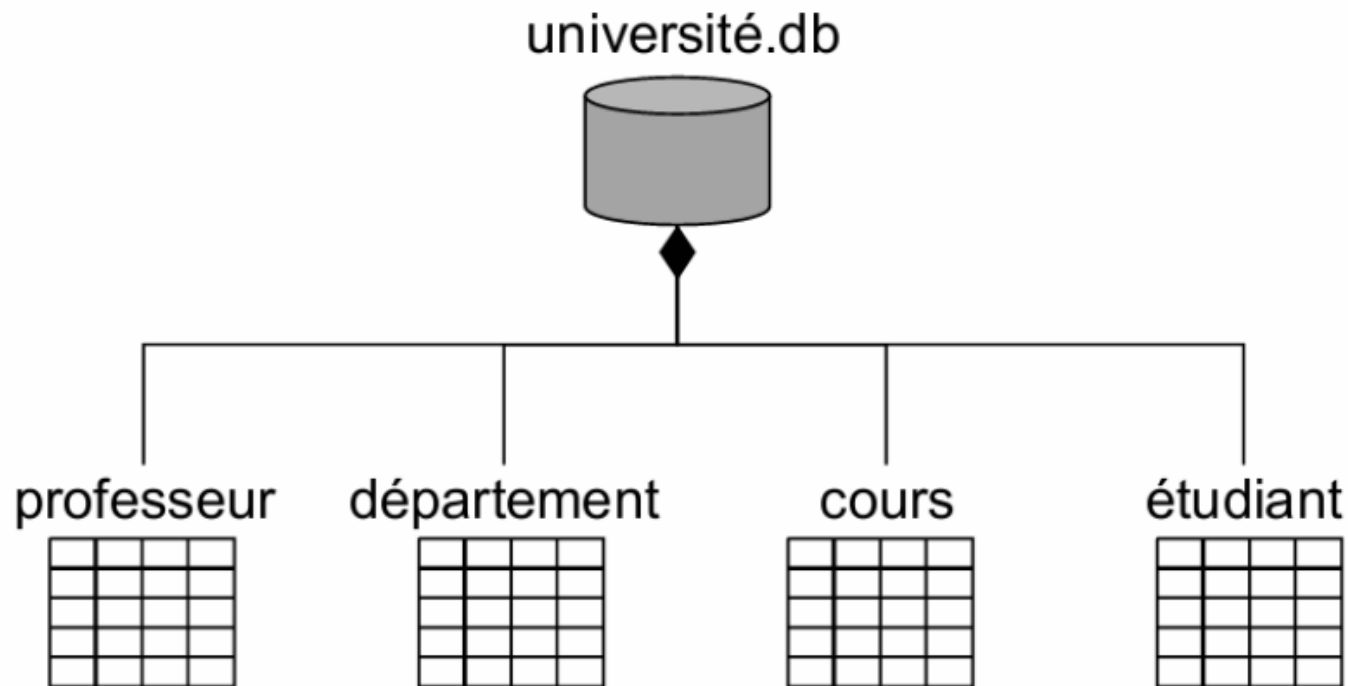
MODÉLISATION DU CODE SOURCE

Ex: Gestion de configuration des fichiers sources

MODÉLISATION DE L'EXÉCUTABLE

La composition d'un exécutable

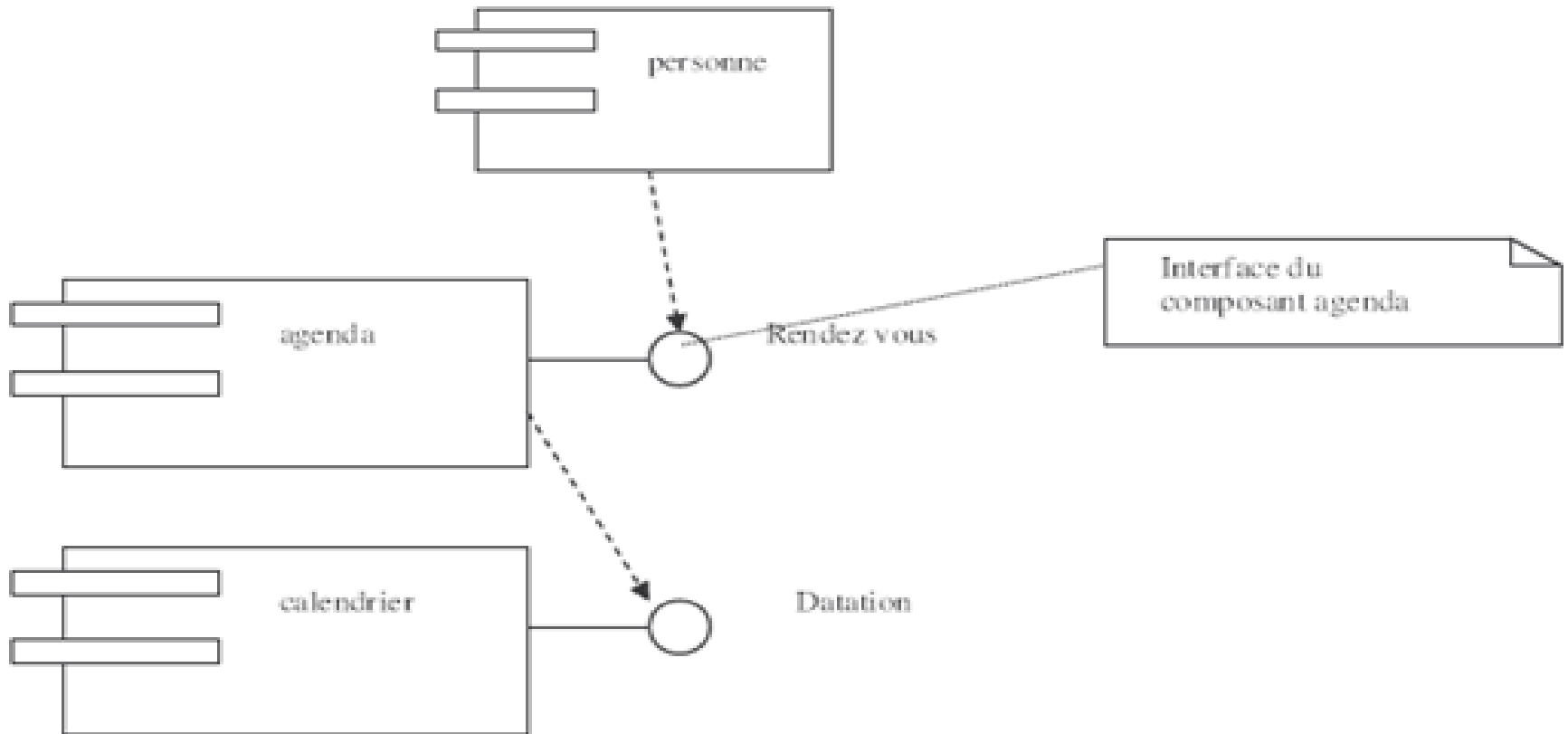
MODÉLISATION D'UNE BASE DE DONNÉES





EXEMPLE

DÉPENDANCES ENTRE COMPOSANTS UTILISATEUR ET DES INTERFACES DE COMPOSANTS FOURNISSEURS



La relation de dépendance peut être spécialisée avec un stéréotype
<<instancié>>
<<ami>>

construire un composant X à partir d'un composant générique ListG et une fonction « amie » de la classe qui se sert des propriétés d'objets X .

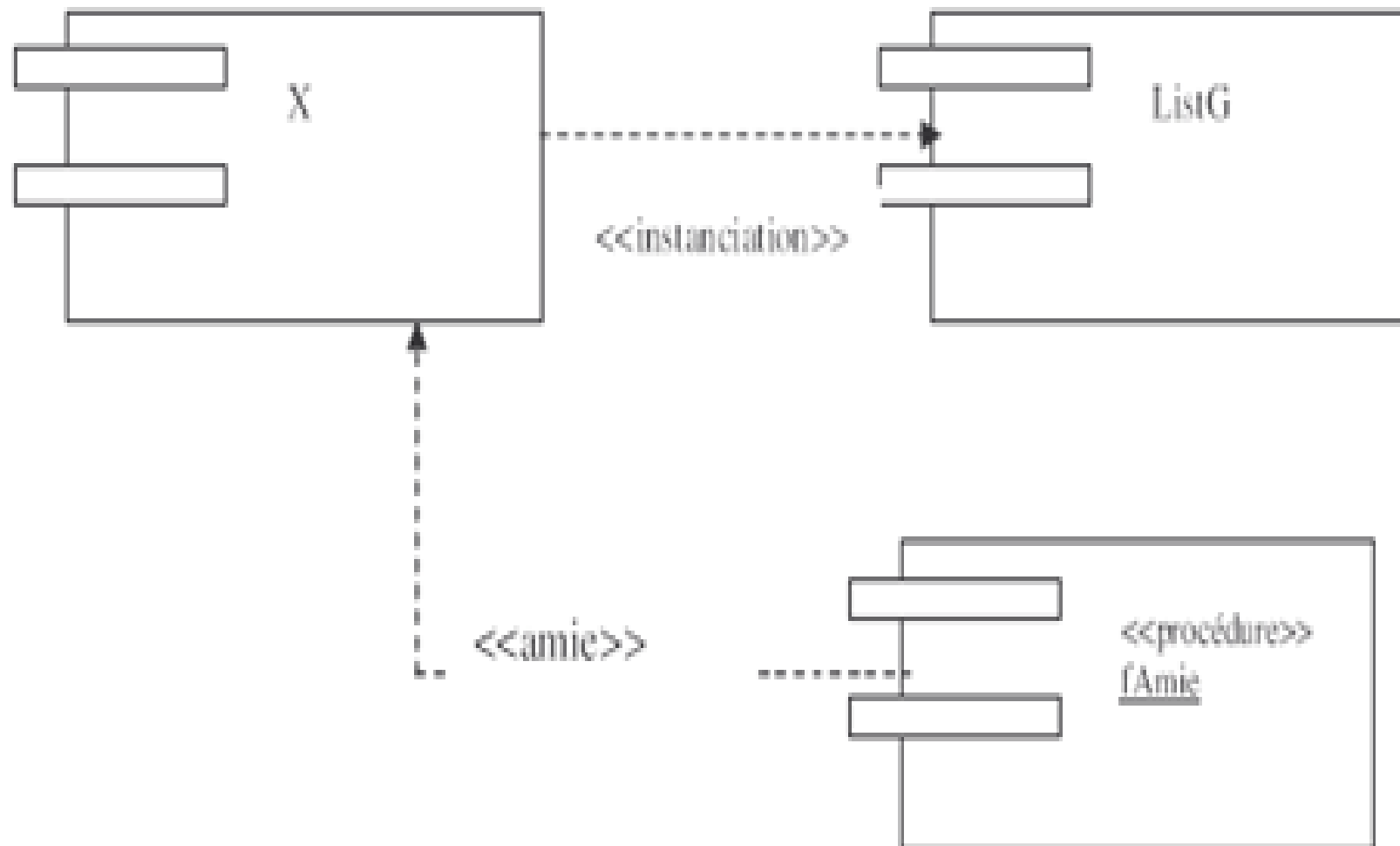
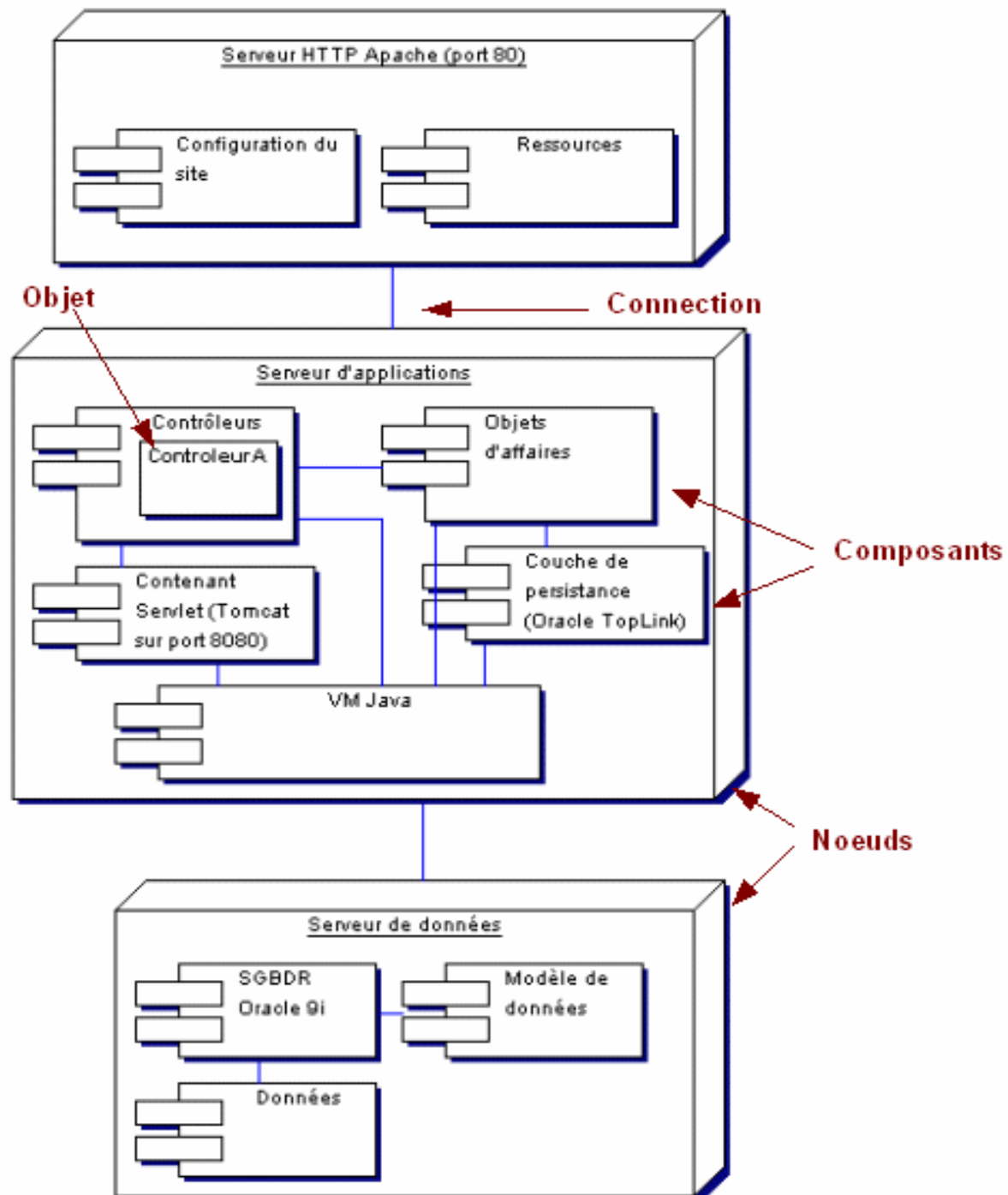


DIAGRAMME DE COMPOSANTS

- Permet de décrire l'architecture statique d'une application en terme de modules (fichiers sources, fichiers compiles, etc.)
- Les dépendances entre modules permettent d'identifier les contraintes de compilation et de mettre en évidence la réutilisation des composants



EXEMPLE



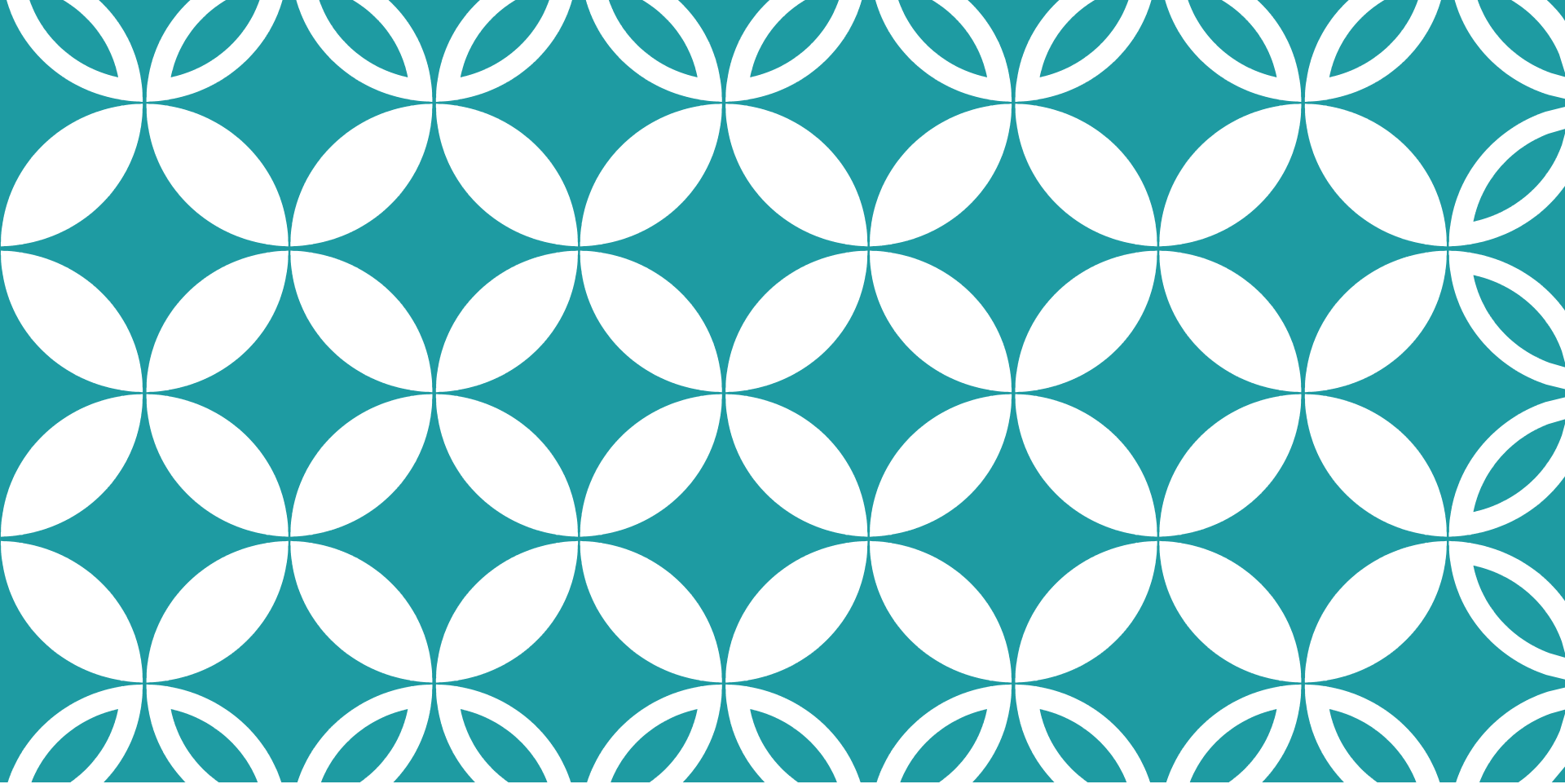
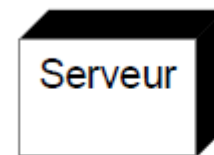


DIAGRAMME DE DÉPLOIEMENT

PARTIE II

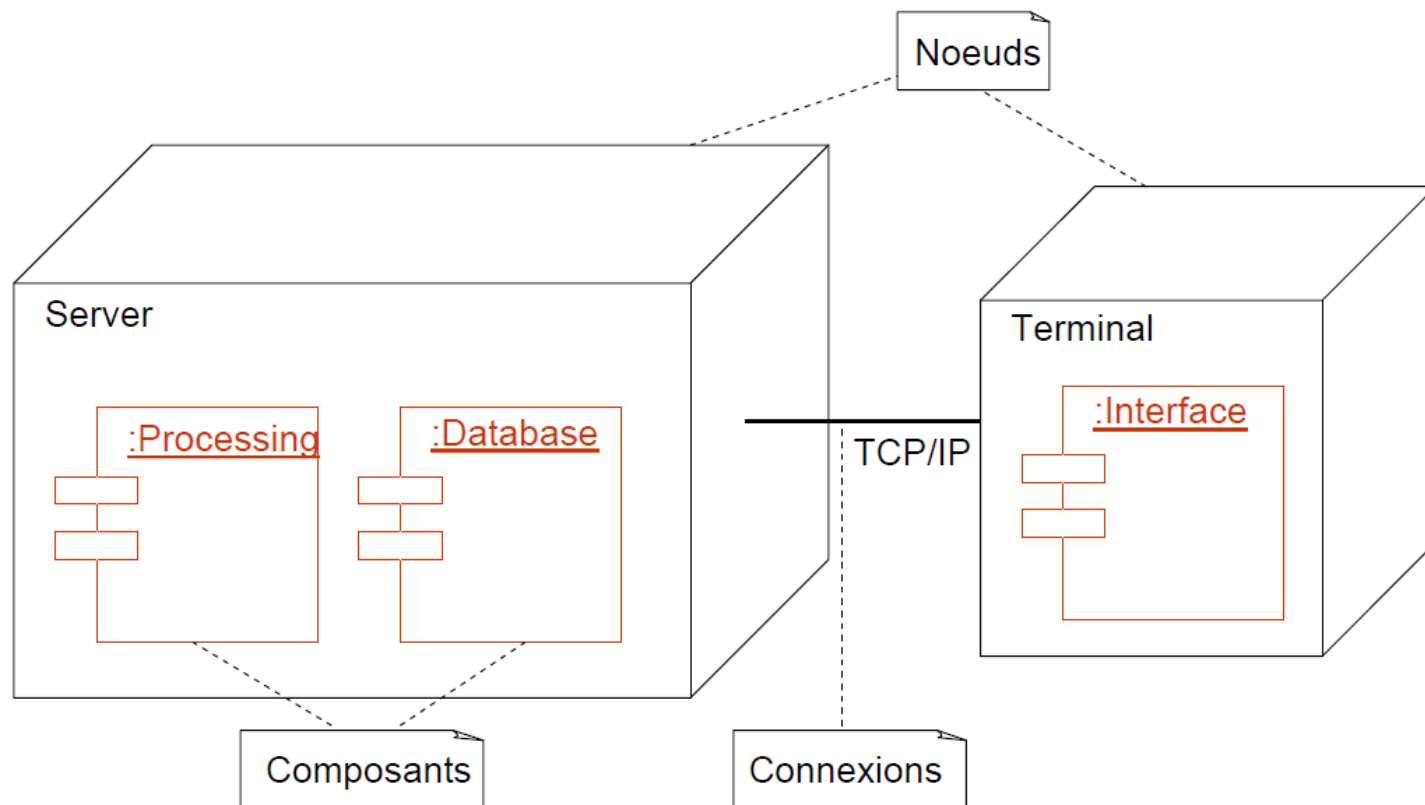
DIAGRAMME DE COMPOSANTS → DIAGRAMME DE DÉPLOIEMENT

- Pour montrer les *instances des composants*, un diagramme de déploiement peut être utilisé.
- Les ressources matérielles sont représentées sous forme de *nœuds*.
- Les nœuds sont connectés entre eux, à l'aide d'un *support de communication*.



Nœud

DIAGRAMME DE DEPLOIEMENT



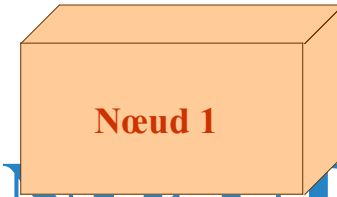
DÉFINITION D'UN DIAGRAMME DE DEPLOIEMENT

Un diagramme de déploiement montre les instances de nœuds (un matériel précis), ou des classes de nœuds.

- Décrit sur quels dispositifs matériels on va déployer les composants logiciels (les différents modules de l'application)
- Décrit la nature des lignes de communication entre les dispositifs matériels

UTILITÉ DES DIAGRAMMES DE DÉPLOIEMENT

- Modéliser l'architecture physique (matérielle) et logique (logicielle) d'un système d'information
- Montrer la disposition physique des différents matériels (les nœuds) qui entrent dans la composition d'un système et la répartition des instances de composants, de processus et d'objets qui « vivent » sur ces matériels.
- Spécifier les Nœuds et les Relations (dépendance, associations)

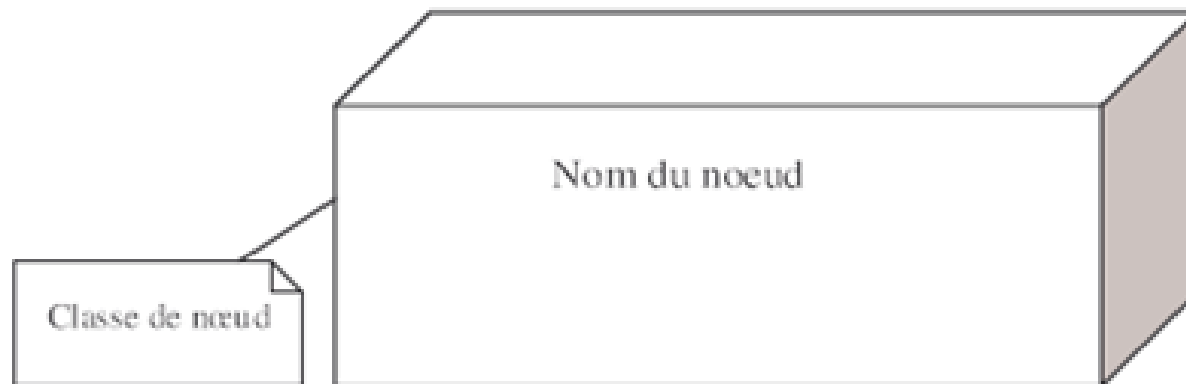


DÉFINITION D'UN NŒUD

- Élément physique ou matériel qui existe pendant l'exécution et représente une ressource informatique
- Possède un nom (simple ou composé du nom de paquetage) pour se distinguer des autres nœuds.
- Possède sa propre mémoire et une capacité de traitement
- Prend en charge l'exécution des composants.
- ***unité de répartition*** = ensemble de composants associés aux nœuds

REPRÉSENTATION D'UN NOEUD

Un Nœud est représenté en UML en 3D par un parallélépipède rectangle



Nom du nœud

Un objet connu

:nom du nœud → objet générique

NŒUDS – STÉRÉOTYPES

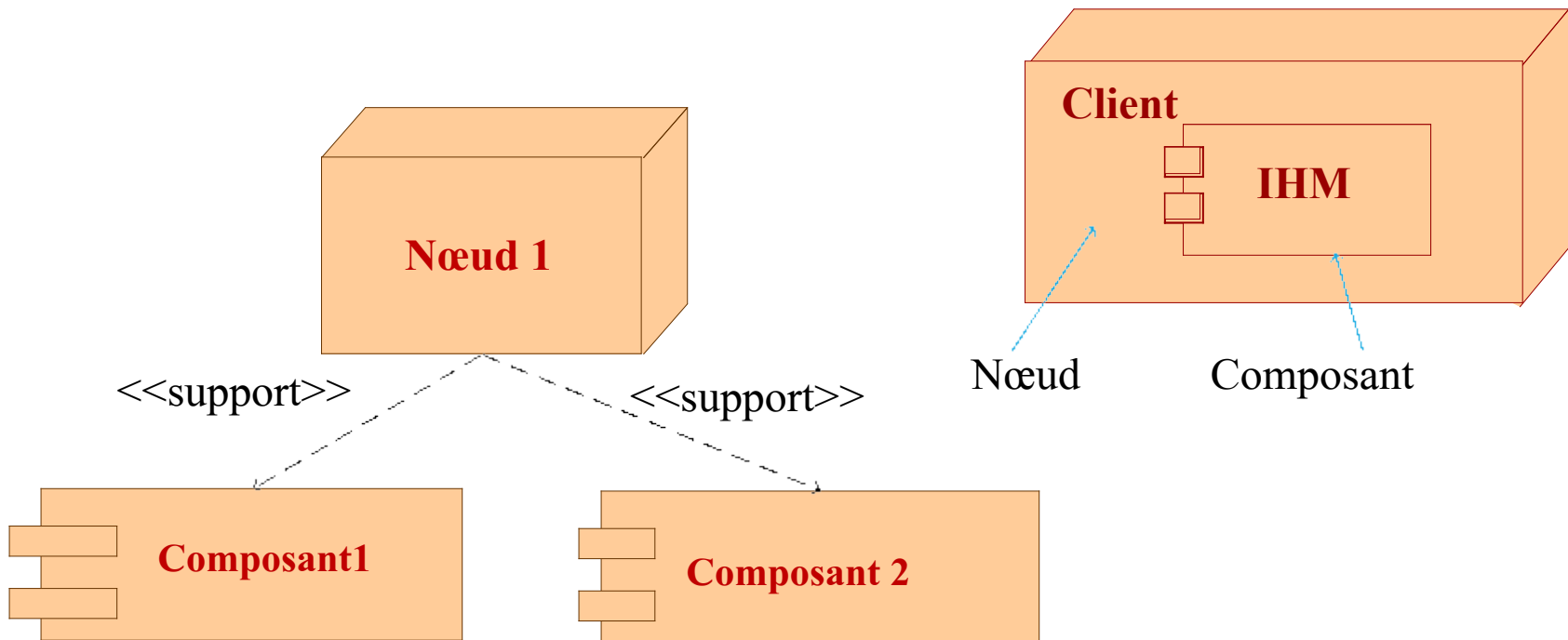
Un nœud peut posséder un stéréotype qui permet de le distinguer des autres types de ressources

Pour spécifier le types de ressources

- « *CPU* » : une unité de calcul
- « *memory* » : une unité de stockage
- « *device* » : un dispositif tel qu'un capteur

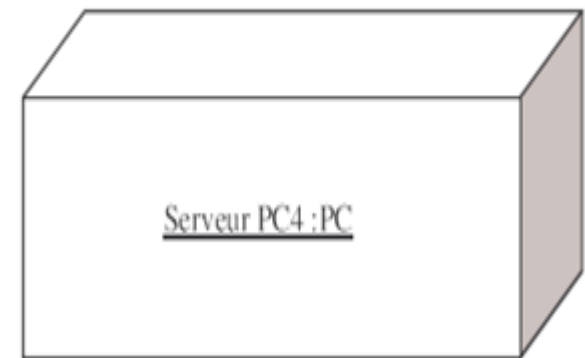
RELATIONS ENTRE NŒUDS ET COMPOSANTS

- 1. Dépendance:** un nœud supporte un composant
=> Des composants qui résident dans un même nœud



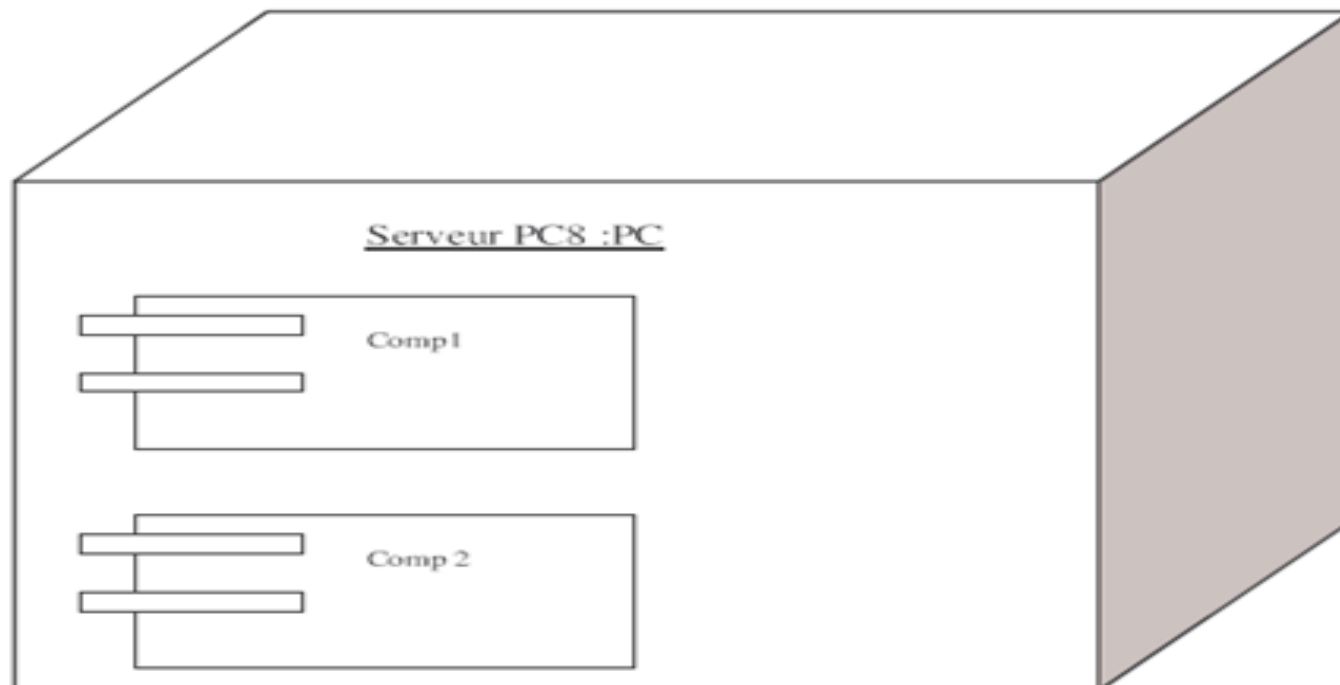
Un nœud est une sous classe d'un classificateur:

- Un nœud peut avoir
 - des attributs
 - des opérations



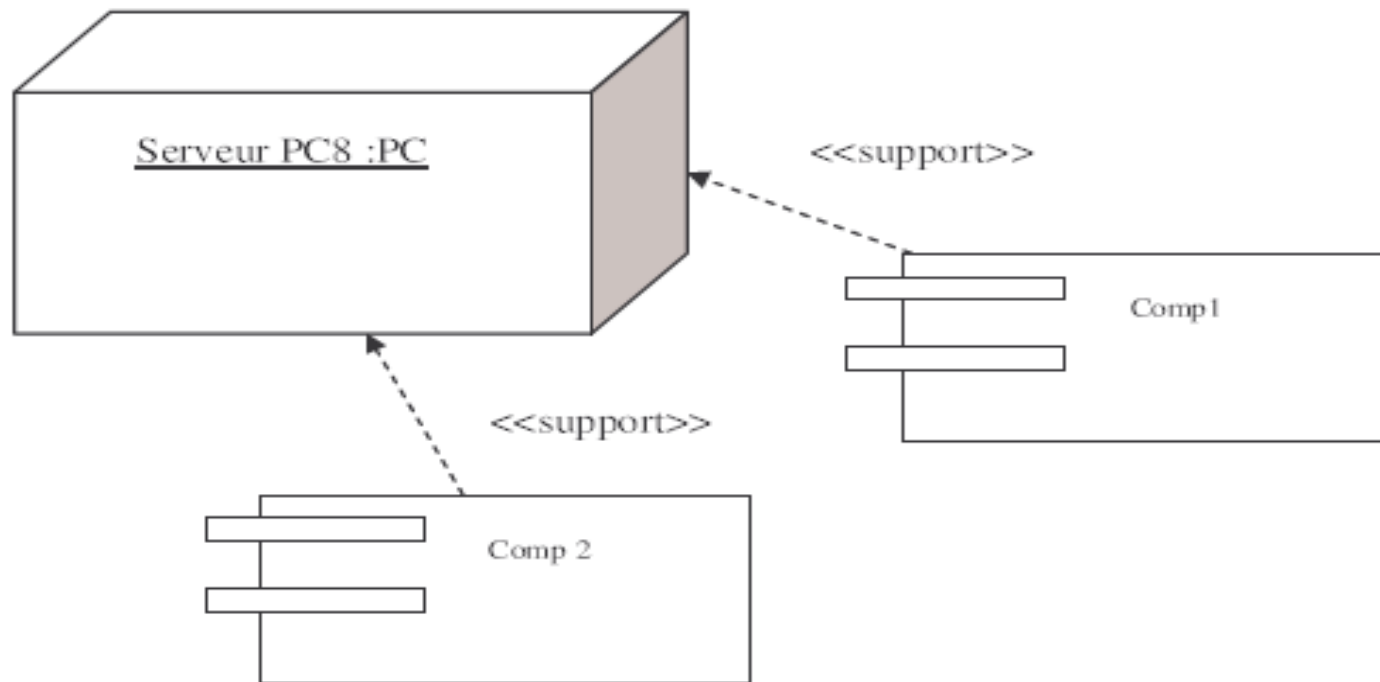
et peut participer à des relations (association, généralisation, agrégation, composition)

A chaque nœud on ajoute le ou les composants résidents sur le nœud (matériel).

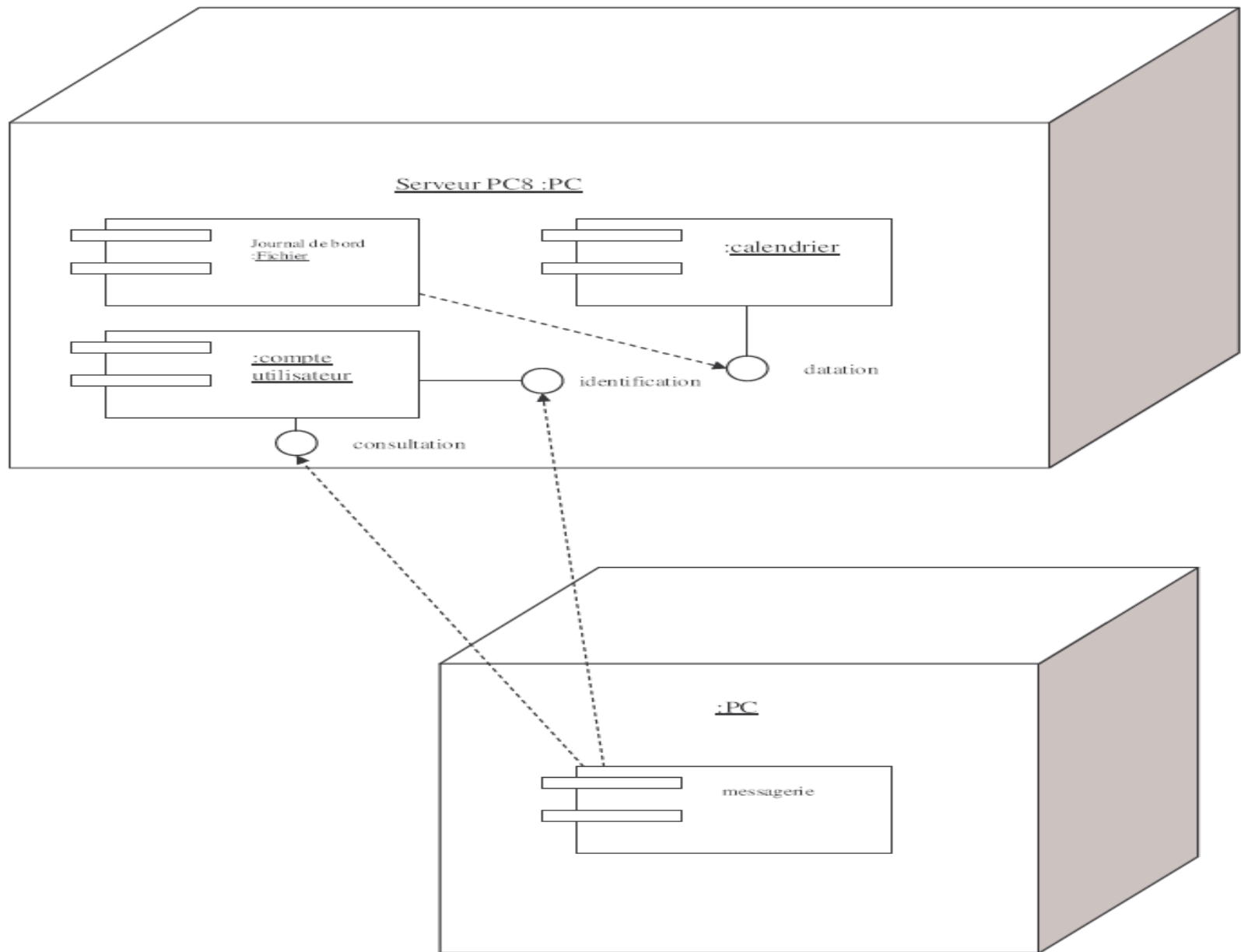


EXEMPLE

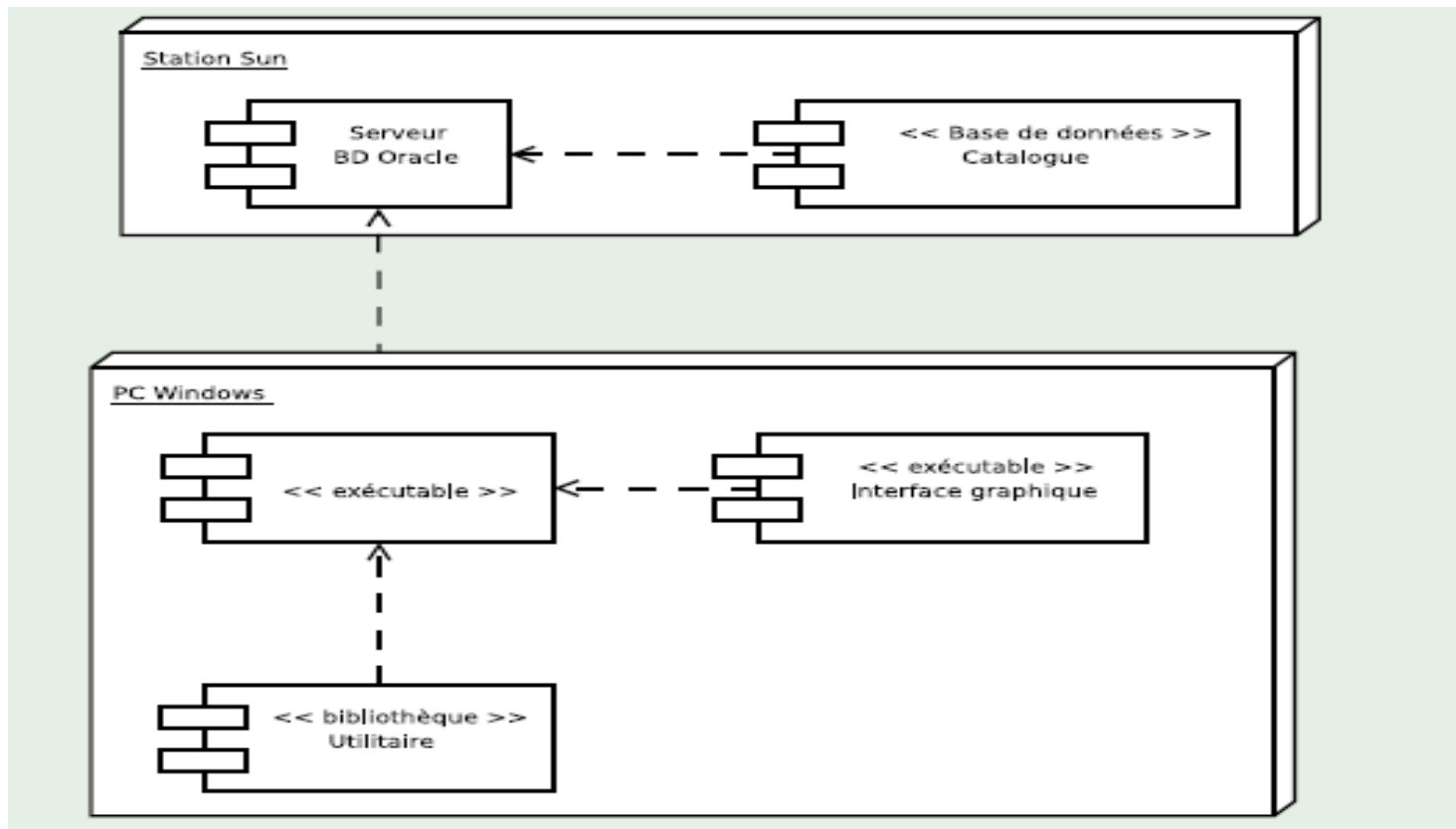
Autre représentation en utilisant la dépendance



Exemple :



EXEMPLE

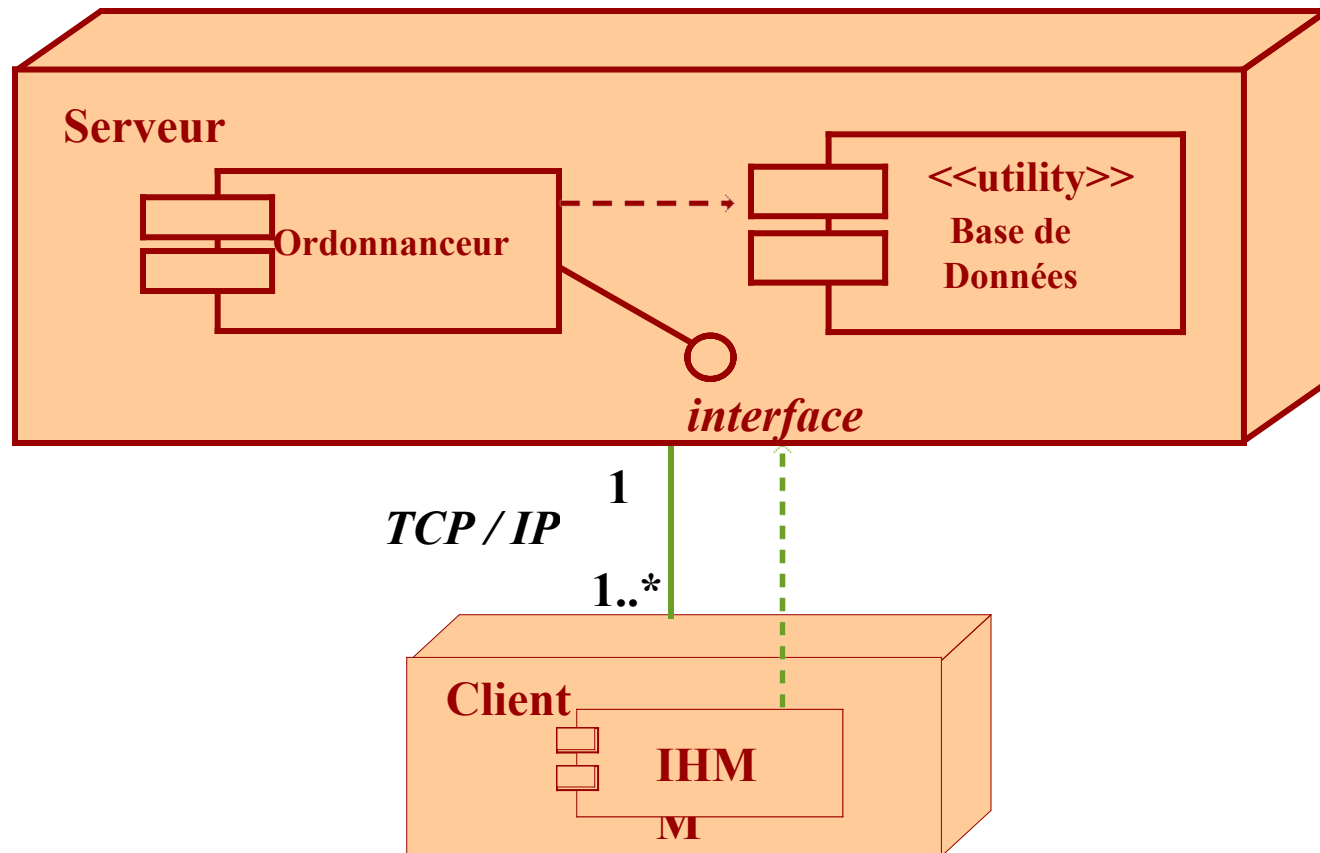


RELATIONS ENTRE DEUX NŒUDS

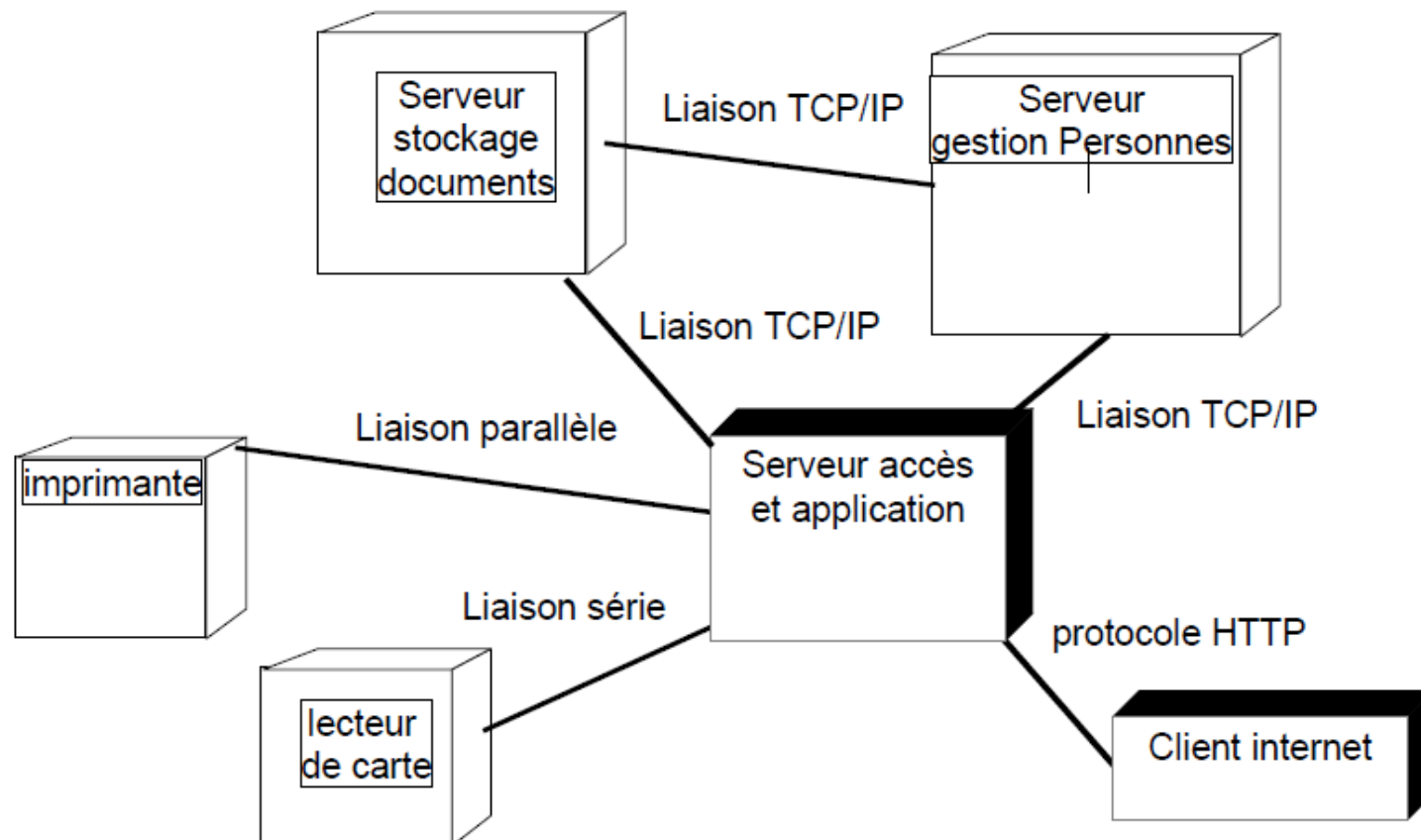
2. **Association** : Indique une voie physique entre deux nœuds
- Une connexion Ethernet
 - Un bus de communication



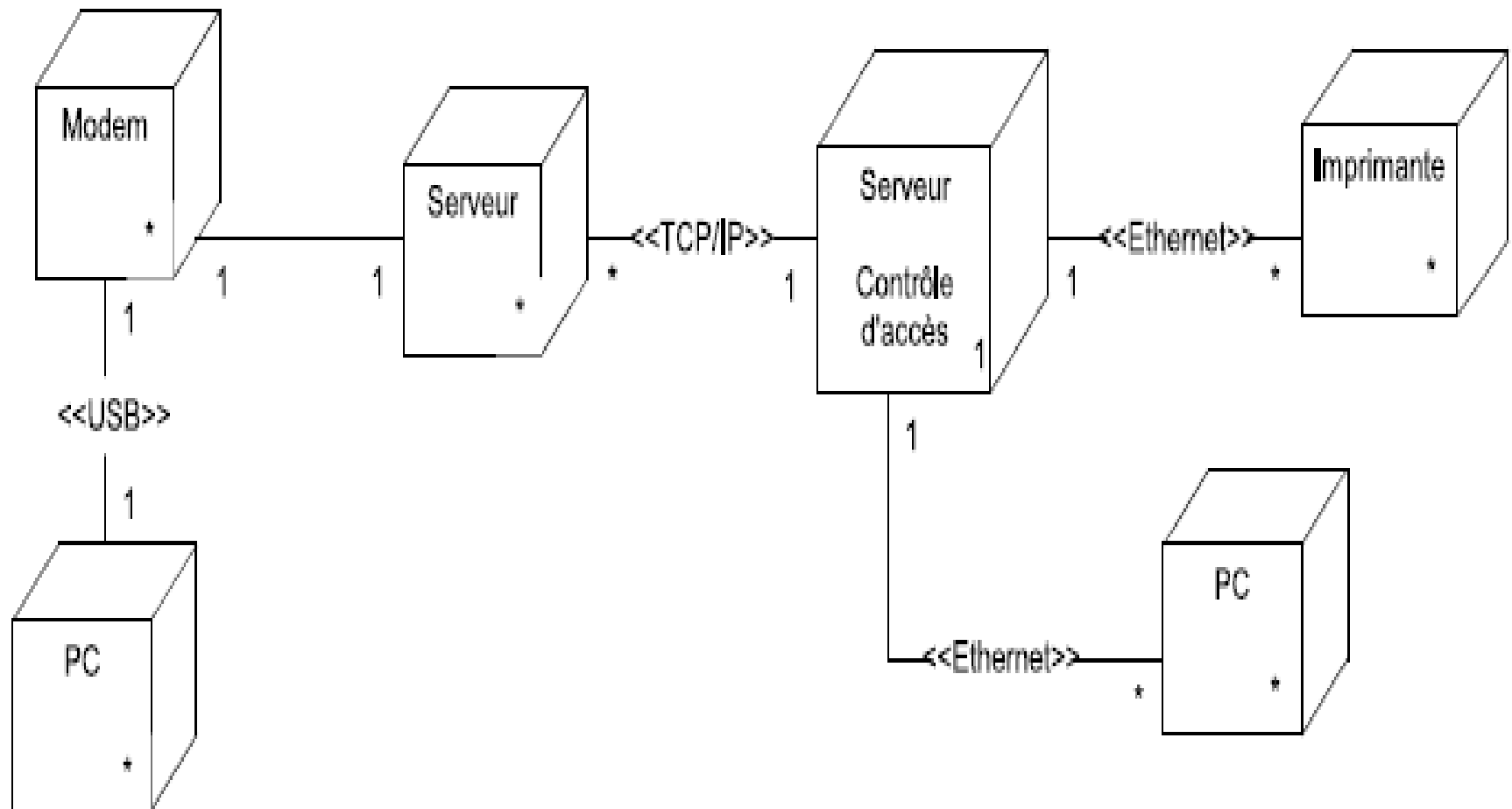
EXEMPLE DE DIAGRAMME DE DÉPLOIEMENT



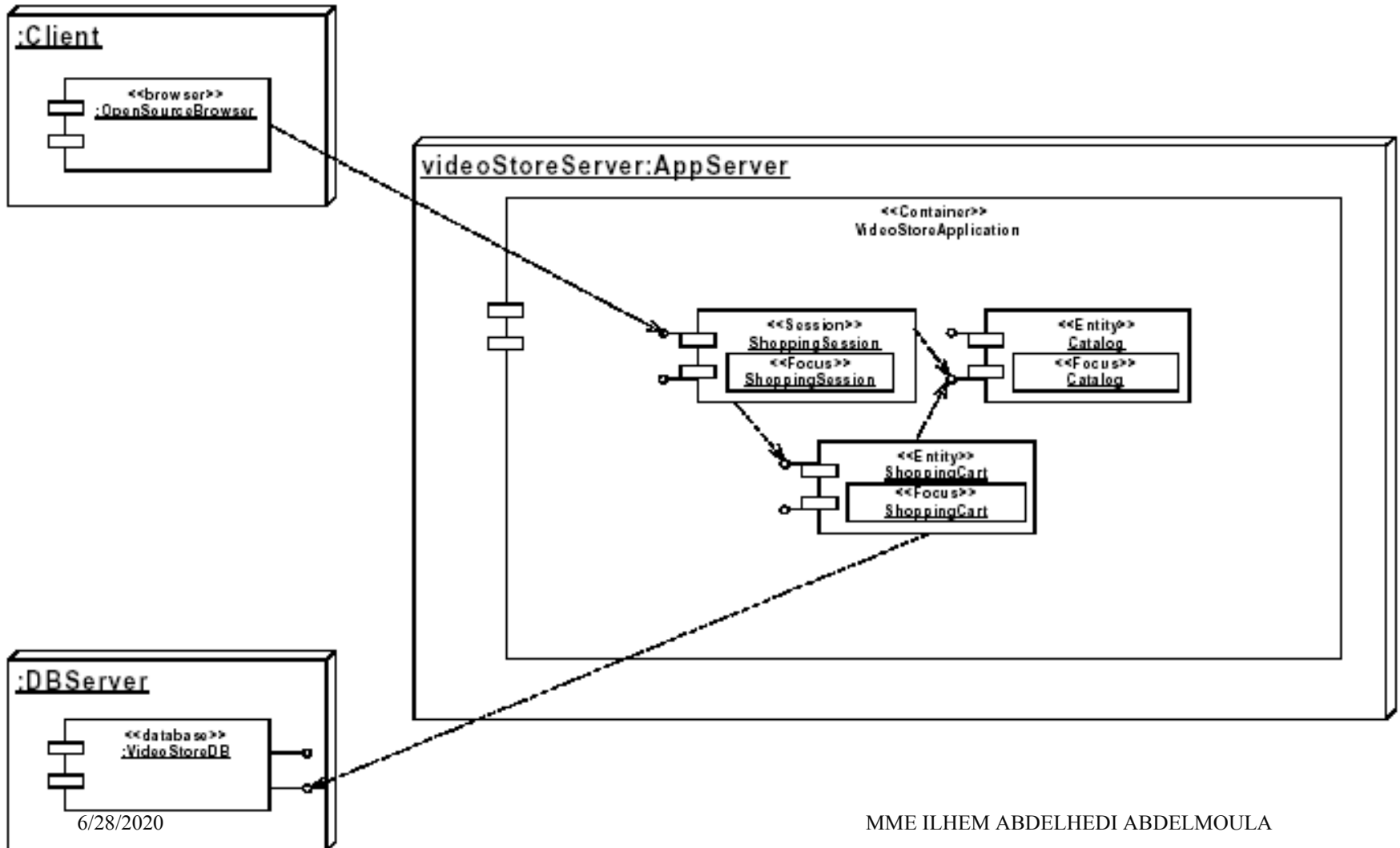
EXEMPLE



EXAMPLE 2



EXEMPLE DE DIAGRAMME DE DÉPLOIEMENT



STÉRÉOTYPES <<DEVIENT>> ET <<COPIE>>

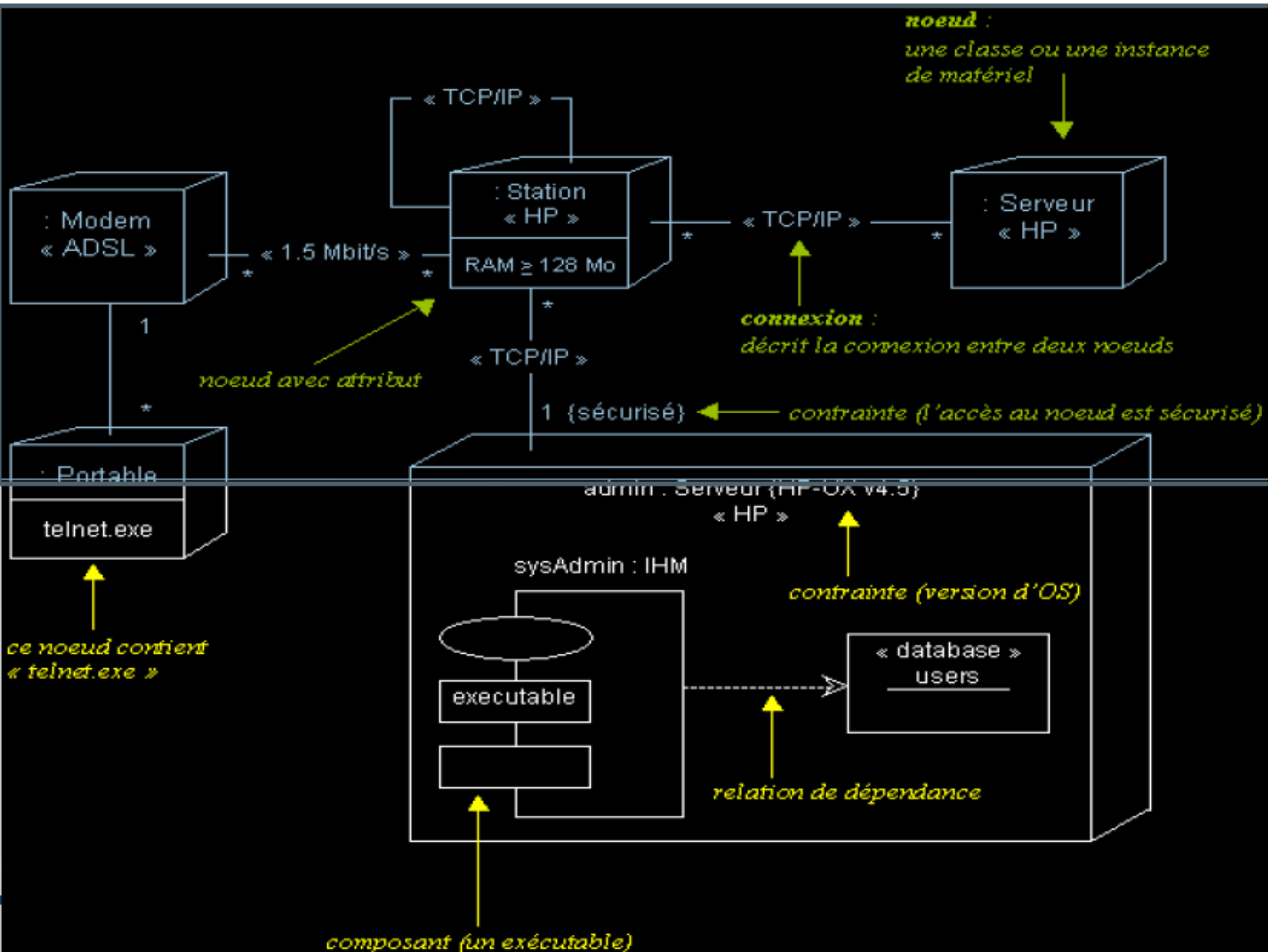
En utilisant le stéréotype <<devient>>

On peut modéliser la migration d'un composant (ou d'un objet) d'un nœud à un autre, ce qui se représente par :



On peut modéliser la réplication c'est à dire la copie d'un composant (ou d'un objet) d'un nœud à un autre.

en utilisant le stéréotype << copie>> dans le schéma précédent

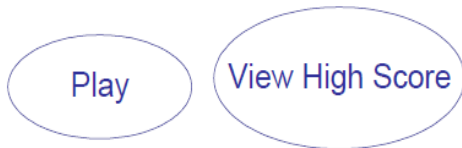


ARCHITECTURE EN COUCHES

Présentation



Applications

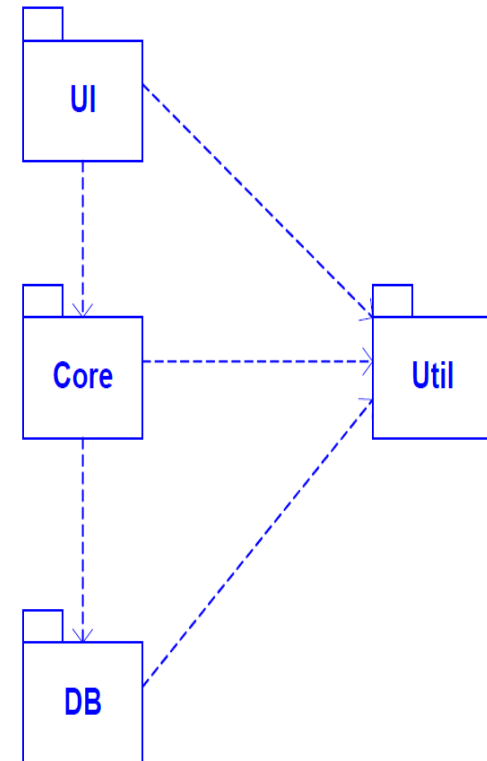


Données
(Persistence)



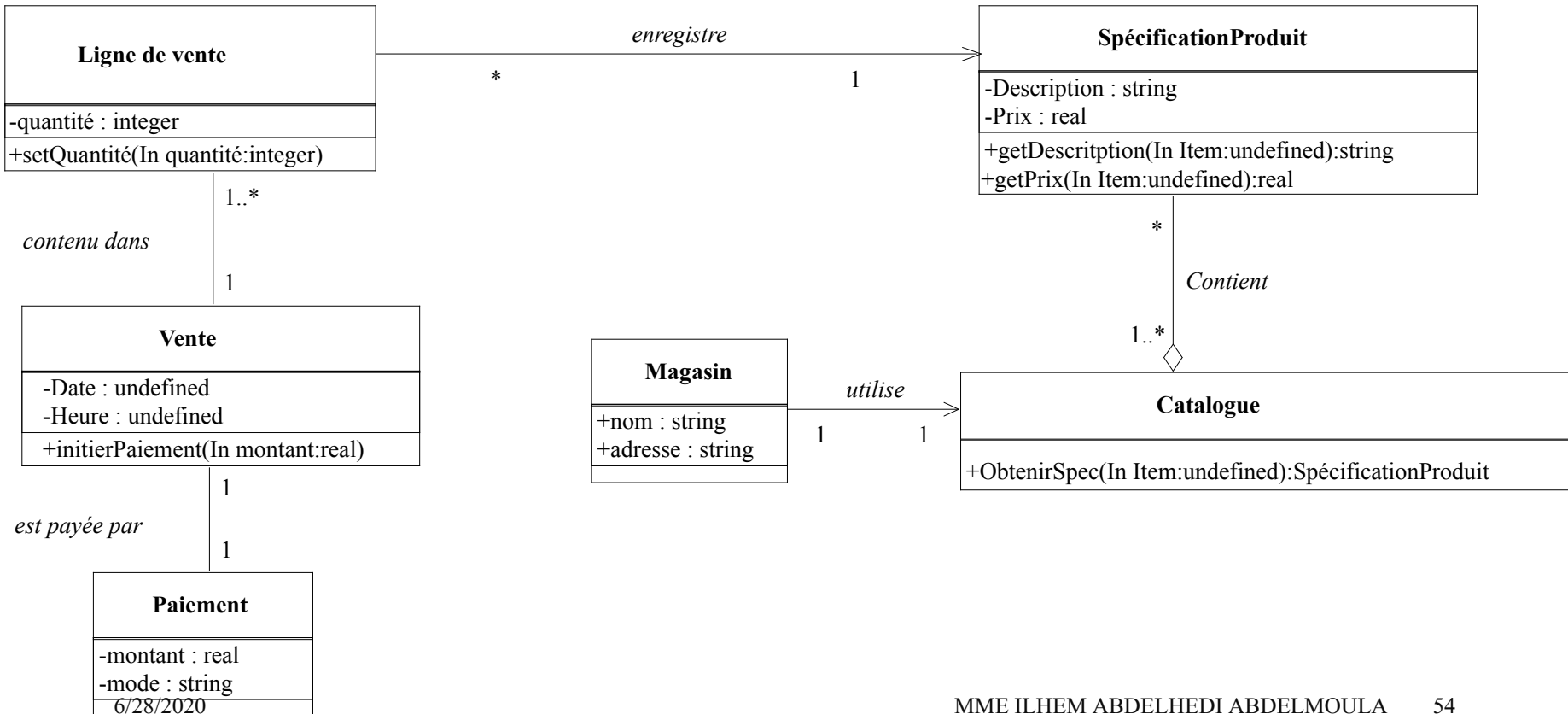
Un diagramme de paquetages correspond à l'architecture

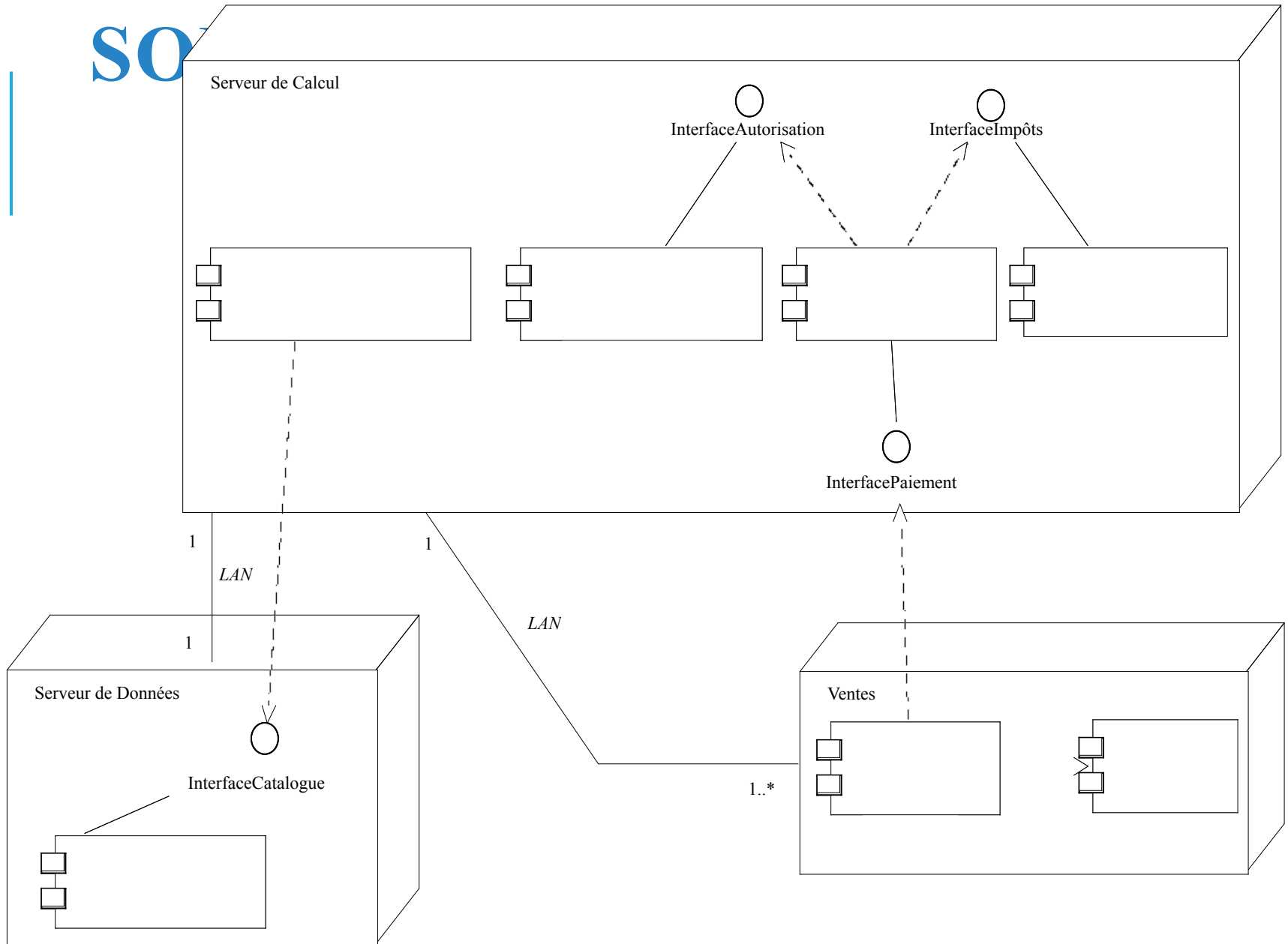
UI : couche de présentation
Core : Couche d'application
DB : Couche de persistance
Util : un sous-système regroupant des services qui ne s'attachent pas clairement aux niveaux



EXERCICE

Réaliser le diagramme de composants et le diagramme de déploiement associés.





EXERCICE D'APPLICATION

But : Modéliser le processus de vente de billets de train.

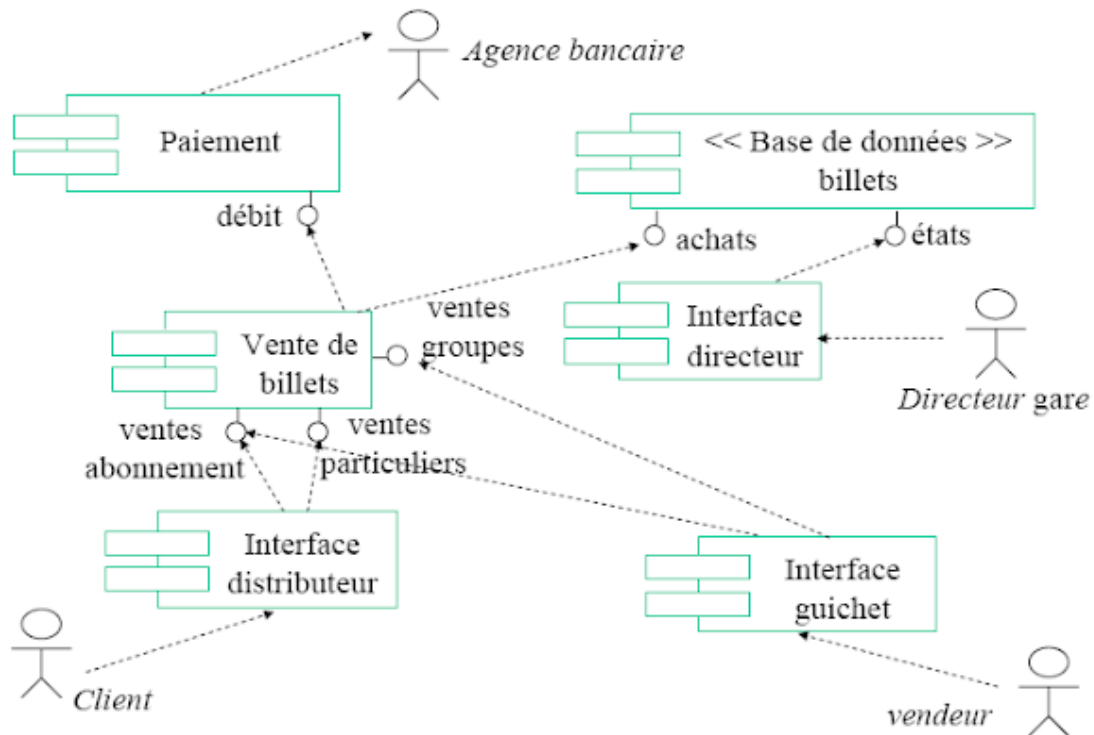
- Une interface distributeur est mise à la disposition des clients soit pour des ventes particuliers ou pour des ventes abonnement.
- Une interface guichet permet au vendeur d'effectuer des ventes groupes ou des ventes abonnement.
- Le paiement des billets s'effectue par carte bancaire.
- Le directeur de la gare peut consulter les états des billets en accédant à la base de données via une interface.

Travail demandé :

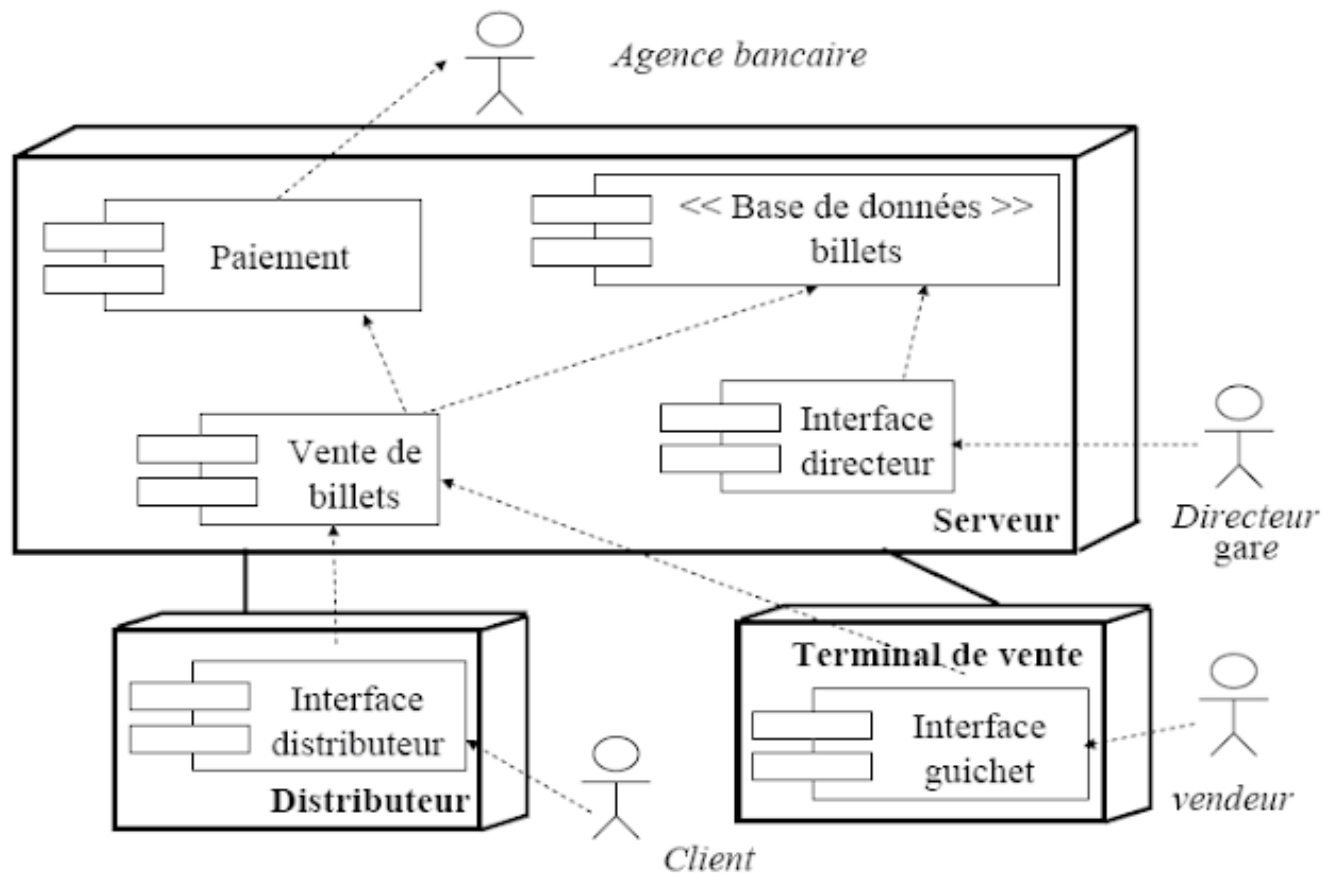
- Établir un diagramme de composants adéquat.
- Établir un diagramme de déploiement propre au processus de vente de billets de train décrit dans le premier exercice

SOLUTION

Diagramme de composants

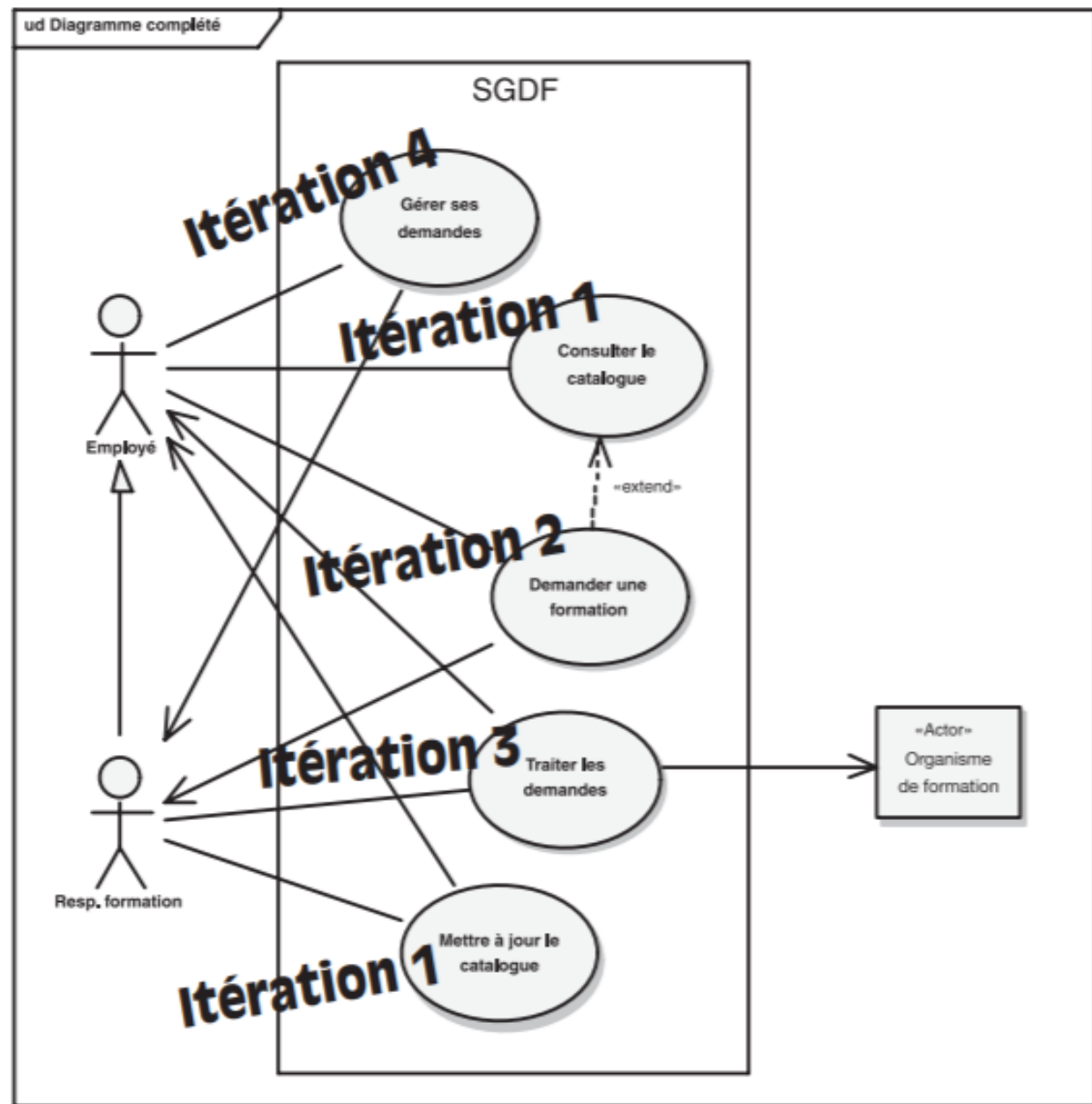


SOLUTION



EXERCI

Proposez un diagramme de déploiement réaliste pour les trois premières itérations du système de gestion des demandes de formation



SOLUTION

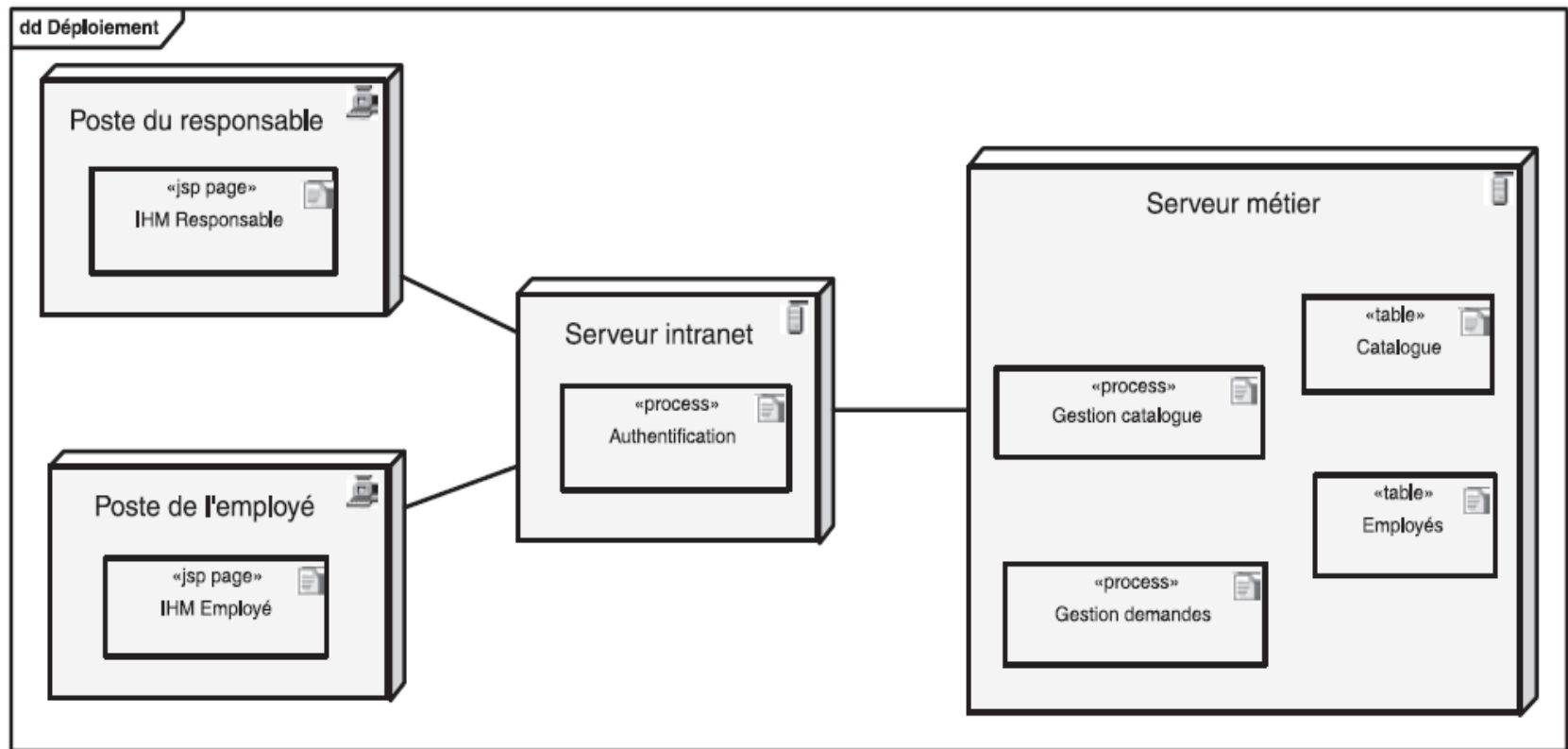
Chaque acteur a son propre poste client qui est un « PC » connecté au serveur intranet de l'entreprise, lequel est lui-même un PC serveur NT. Ce serveur intranet contient en particulier l'application d'authentification.

Chacun des deux acteurs a en outre sa propre interface homme-machine, matérialisée par une page *JSP*. Ces deux *JSP* utilisent un même service d'authentification général, contenu par le serveur intranet. Le catalogue est stocké dans une base de données spécifique, de même que les employés.

Le serveur métier héberge pour sa part les autres applications ainsi que les bases de données. Il s'agit là d'une machine Unix et ce pour des raisons historiques...

Le dessin, réalisé avec l'outil Enterprise Architect qui fournit ses propres icônes pour les différents types de noeud, est donné à titre d'exemple sur la figure suivante

DIAGRAMME DE DÉPLOIEMENT



SYNTHÈSE

1. Diagramme de cas d'utilisation : ce qu'on attend du système; c'est là où on assimile les fonctionnalités demandées par le client
2. Diagramme de classes : les entités du système; le coeur de la conception d'un système
3. Diagrammes de séquence et de collaboration : comment les entités interagissent; indispensable pour comprendre l'interaction entre les classes
4. Diagrammes d'états - transitions et d'activités: les états et les fonctionnalités de chaque entité ; Toute la dynamique du système est là
5. Diagrammes de composants et de déploiement : l'architecture du système

UNE CLASSIFICATION

Vue statique du système

- D cas d'utilisation
- D classes
- D composants
- D déploiement

Vue dynamique du système

- D séquences
- D états - transitions
- D activités