UML 2
Class, Object, Package, Component, Deployment, Composite Structure Diagrams

Zineb BOUGROUN

Diagramme de classe

- Le plus important de la modélisation orientée objet.
- Le seul obligatoire lors d'une telle modélisation.
- Il montre la structure interne.
- Une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système.
- Représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation

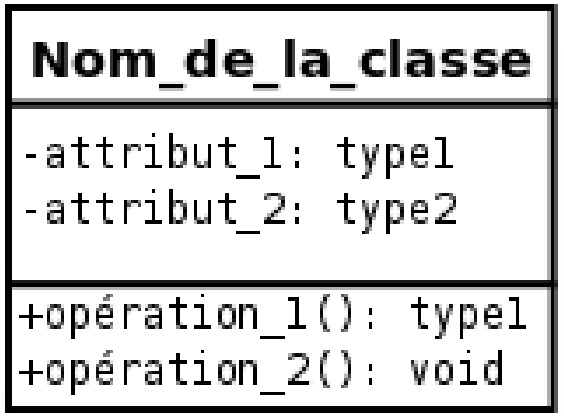
Diagramme de classe

- Une classe est un concept abstrait représentant :
  - des éléments concrets (ex. : des avions),
  - des éléments abstraits (ex. : des commandes de marchandises ou services),
  - des composants d'une application (ex. : les boutons des boîtes de dialogue),
  - des structures informatiques (ex. : des tables de hachage),
  - des éléments comportementaux (ex. : des tâches),
  - …

- Une *instance* est une concrétisation d'une classe :
  - la Ferrari *Enzo* qui se trouve dans votre garage est une instance du concept abstrait *Automobile ;*
  - l'amitié qui lie Jean et Marie est une instance du concept abstrait *Amitié ;*

Diagramme de classe

Représentation d'une classe



Public ou + :

tout élément qui peut voir le conteneur.

Protected ou # :

seul un élément situé dans le conteneur ou un de ses descendants peut voir l'élément.

Private ou - :

seul un élément situé dans le conteneur peut voir l'élément.

Package ou ~ ou rien :

seul un élément déclaré dans le même paquetage peut voir l'élément.

Diagramme de classe

Derived ou / :

un élément calculé à partir d'autres attributs et de formules de calcul

PS : On peut lui associer une note pour documentation

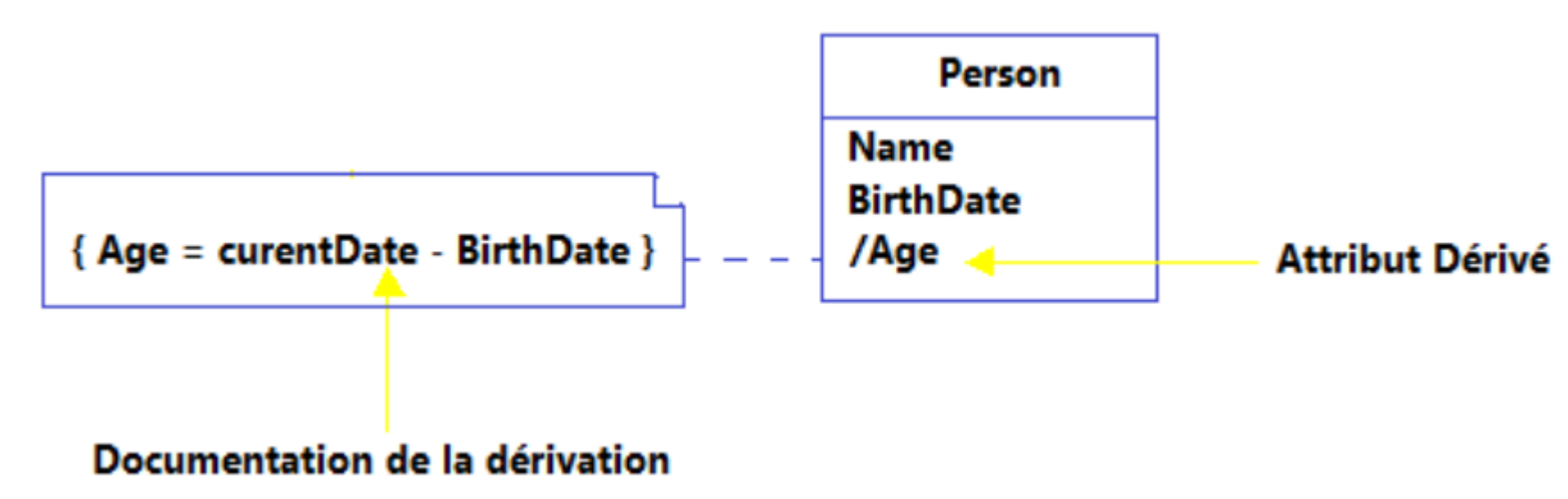


Diagramme de classe

- Syntaxe de déclaration d'attribut :

```
<visibilité> [/] <nom_attribut> : <type> [ = <valeur_par_défaut> ]
```

- Syntaxe de déclaration d'opération:

```
<visibilité> <nom_méthode> ([<paramètre_1>, ... , <paramètre_N>]) : [<type_renvoyé>] [{<propriétés>}]
```

Diagramme de classe

- Association :
  - Décrit les connexions structurelles entre les instances.

- Classe Association :
  - Se connecte à deux ou plusieurs classes et possède également des attributs et des opérations.

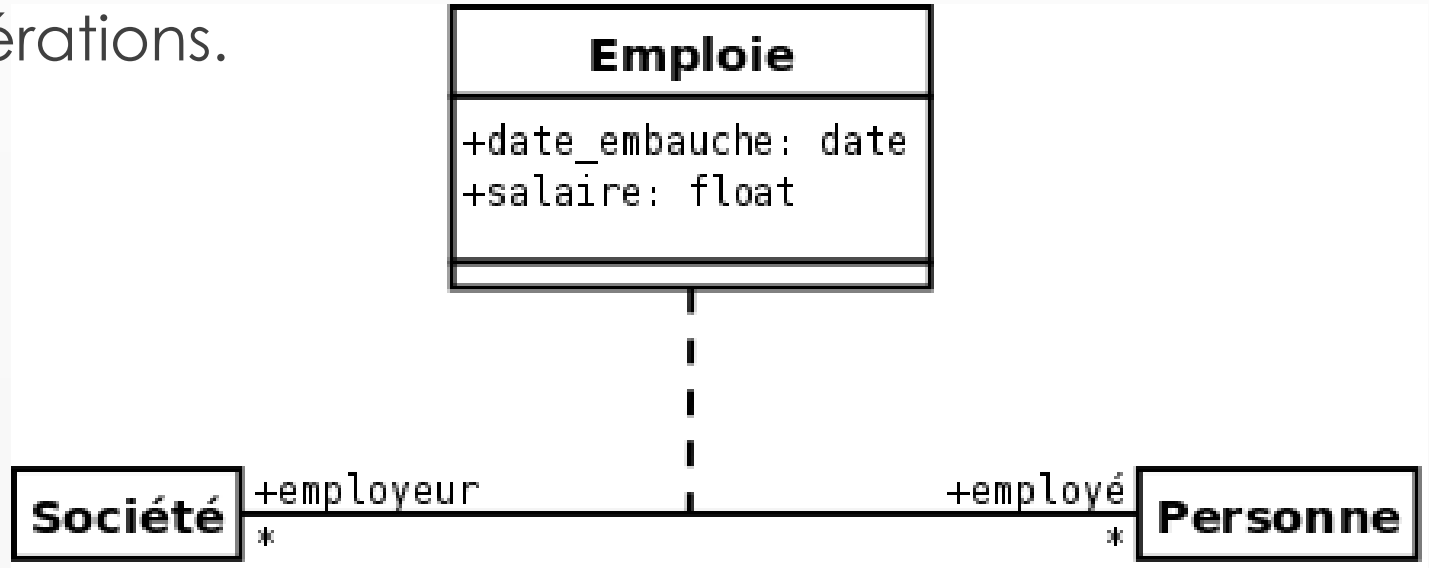
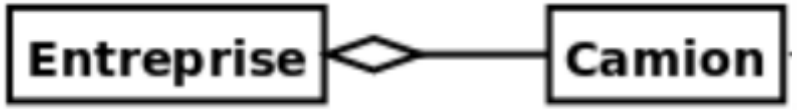


Diagramme de classe

- Agrégation :
  - relation d'inclusion structurelle ou comportementale d'un élément dans un ensemble



  - Composition : décrit une contenance structurelle entre instances

  - Généralisation : relation entre une classe générale (classe de base ou classe parent) et une classe spécialisée (sous-classe)

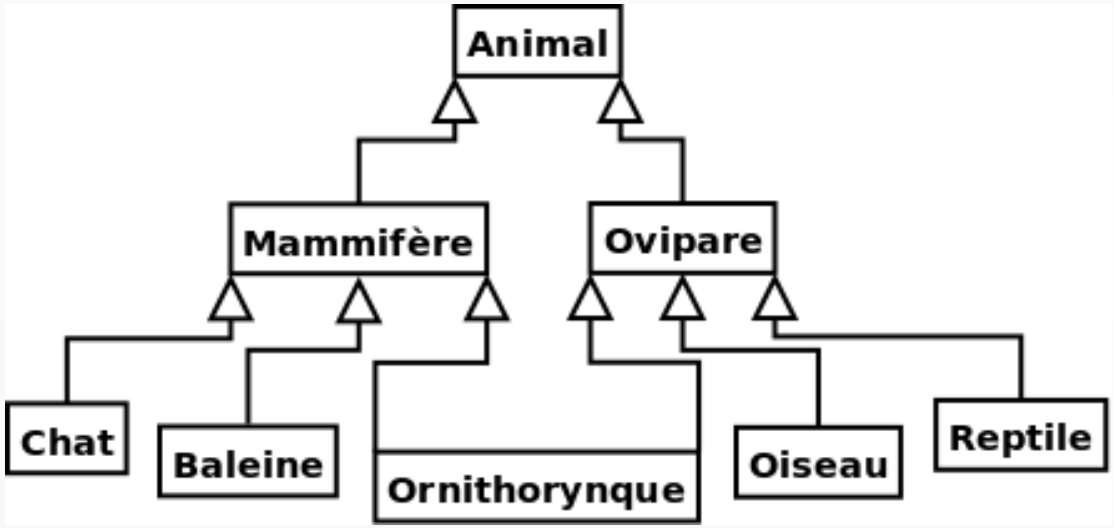


Diagramme de classe

- Chaque Extrémité d'association peut être nommée : C'est le Rôle

- Ils sont nécessaires pour les associations entre deux objets de la même classe ou association multiples

Diagramme de classe : what's new?

- Déclaration d'attribut : multiplicité

[visibilite] nom d'attribut [multiplicité]: type attribut [= valeur initiale]

  - l'attribut représente un ensemble de valeurs; le cas d'un tableau ou une liste

  - Exemple : Parents[1..2] : Personne.

Diagramme de classe : what's new?

- Navigabilité

un produit ne stocke pas les commandes. Mais, chaque objet commande contient une liste de produits

- Association n-aire : lie plus de deux classes

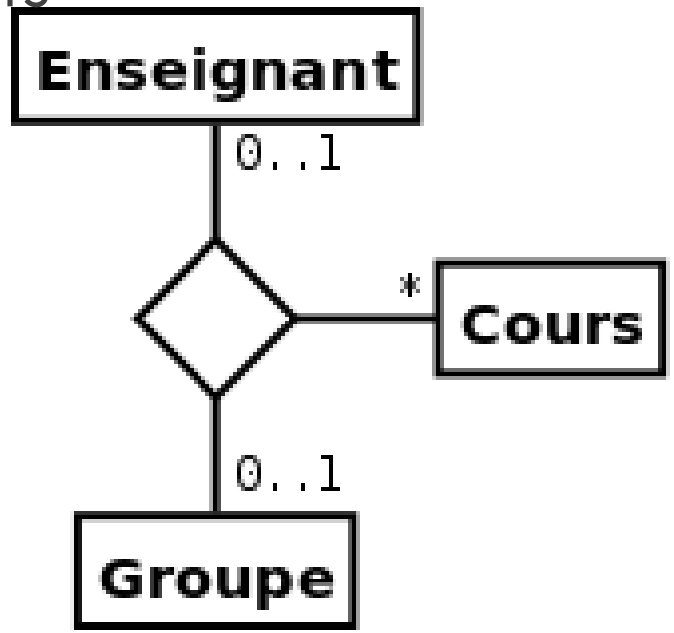


Diagramme d'objet

- Illustrer les interactions concrètes entre instances de classes d'un diagramme de classes

- Vérifier l'adéquation d'un diagramme de classe à différents cas possibles

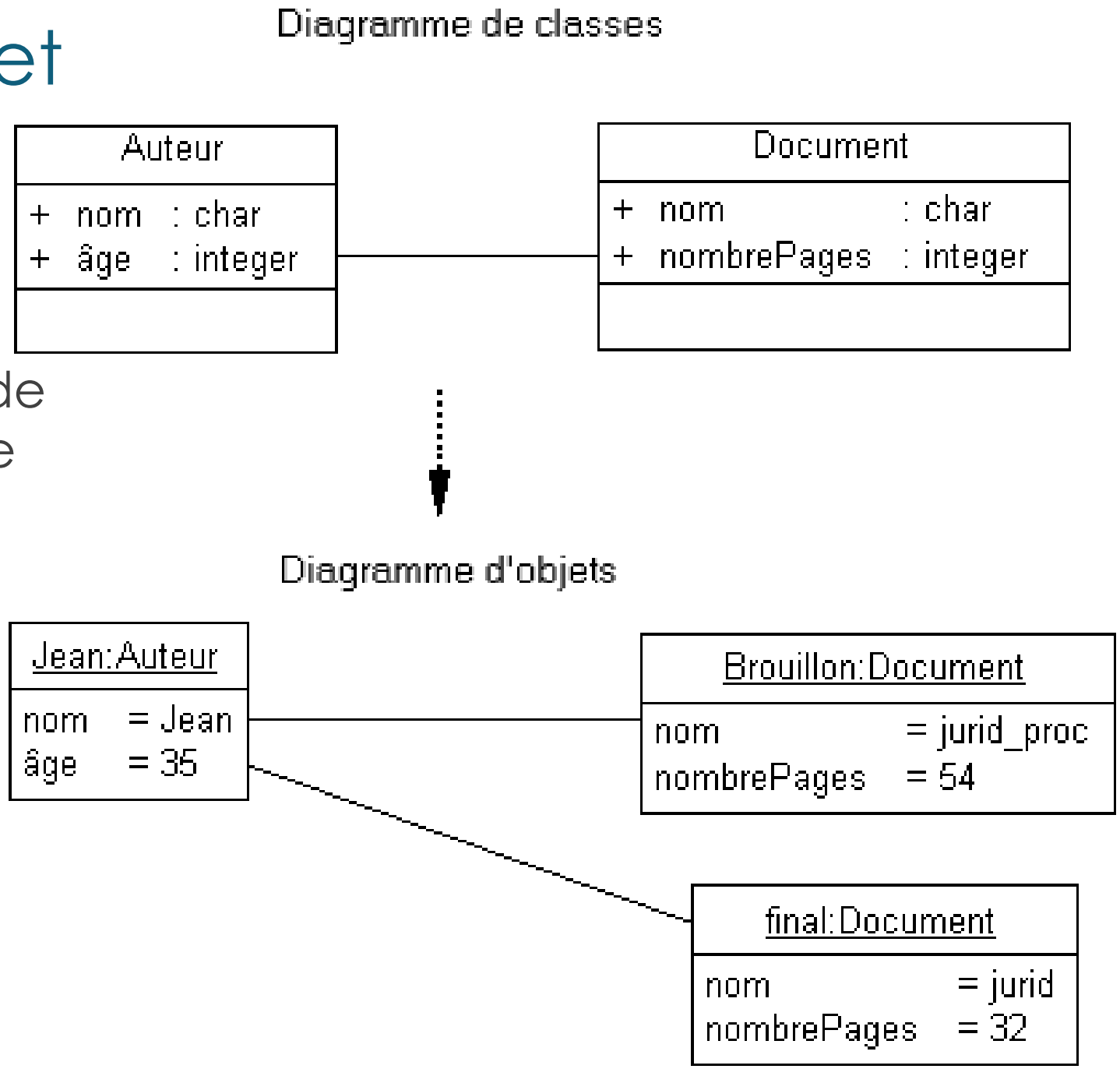
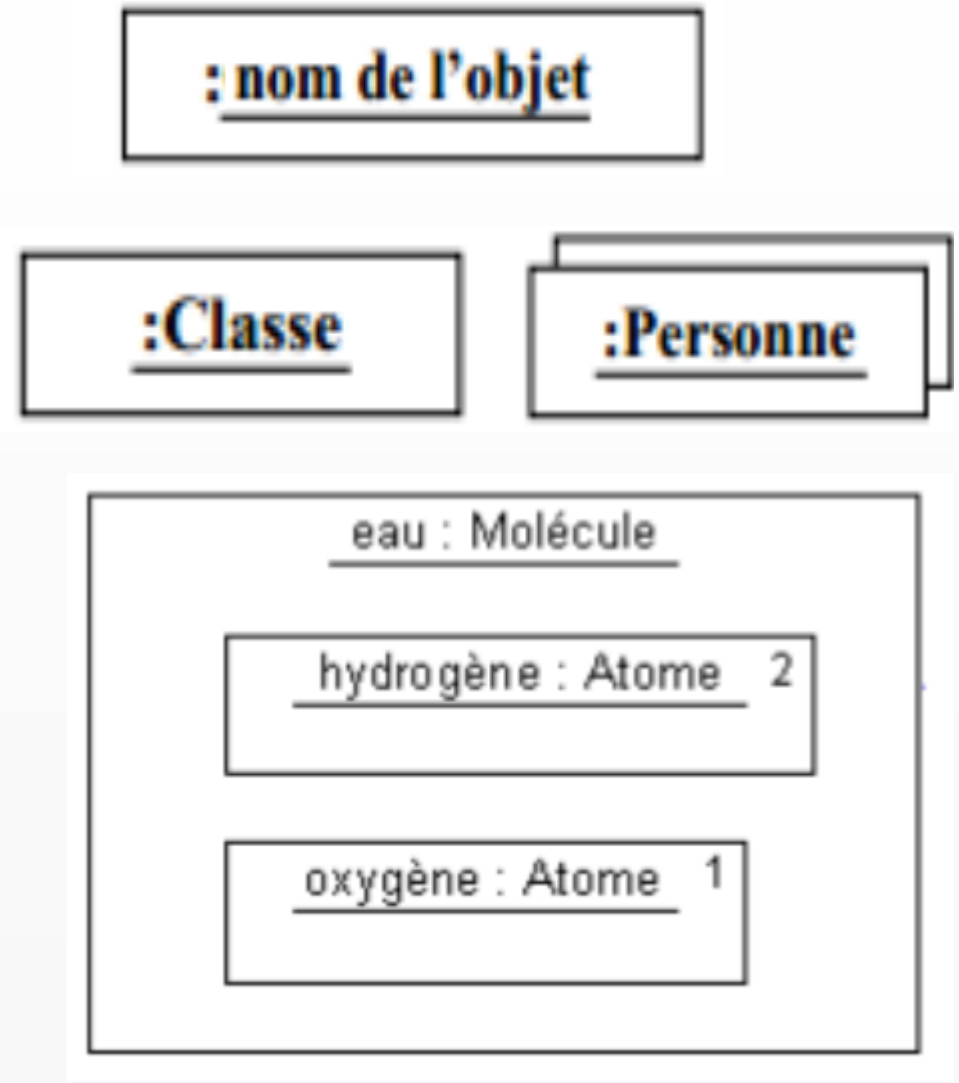
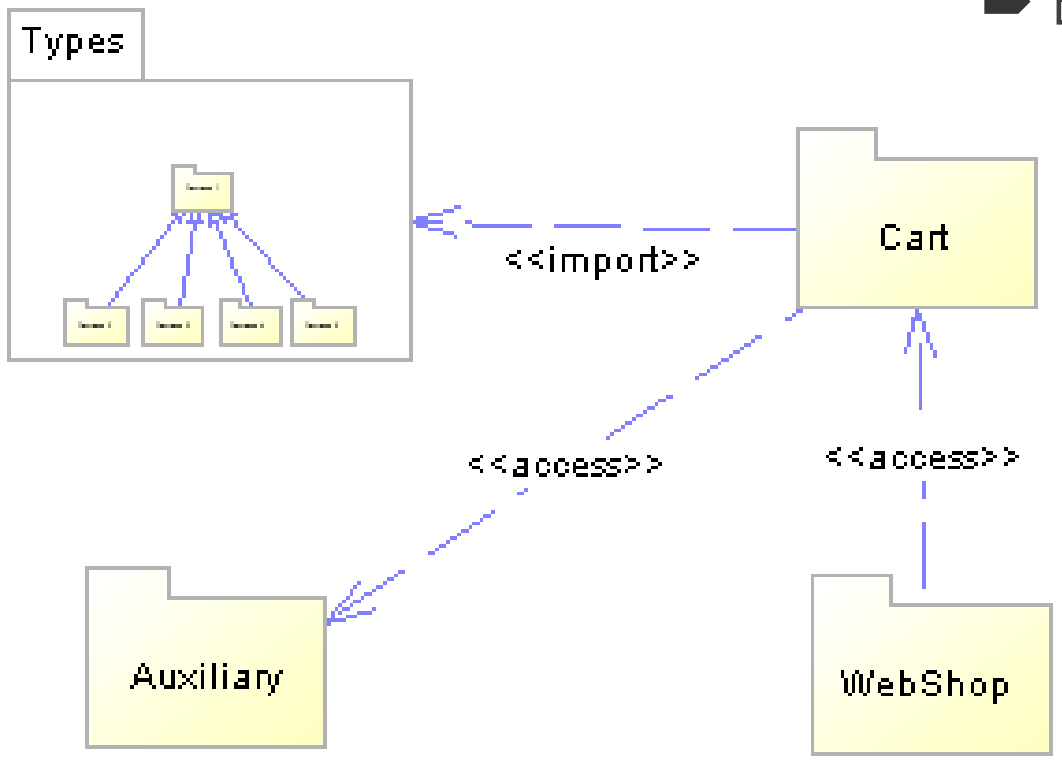


Diagramme d'objet



- le compartiment des opérations n'est pas utile.
- le nom de la classe dont l'objet est une instance est précédé d'un << : >> et est souligné.
- les relations du diagramme de classes deviennent des liens.

Diagramme de package : nouveau



- Élément de modélisation qui :
  - Contient d'autres éléments de modélisation (classes, autres paquetages, ...) ;
  - Possibilité de ne pas représenter tous les éléments contenus
  - Définit un espace de nom (namespace)

Diagramme de package : nouveau

- Dépendances entre paquetages
  - A - - - «use» - - - > B :

    un élément de A nécessite un autre élément de B pour la mise en œuvre complète de son opération

  - A - - - «merge» - - - > B :

    Les éléments de A sont fusinés avec les éléments de B.

  - A - - - «access» - - - > B :

    Tout élément public de B est accessible par son nom complet depuis A

  - A - - - «import» - - - > B :

    Tout élément public de B est accessible par son nom depuis A

Diagramme de package

- Architecture logique
- Définition :
  - un regroupement des classes logicielles en paquetages
  - Point de départ pour un découpage en sous-systèmes
- Objectifs d'une architecture logique
  - Encapsuler et décomposer la complexité
  - Faciliter le travail en équipes
  - Faciliter la réutilisation et l'évolutivité
- Exemples d'architectures logiques :
  - Architecture en couches
  - Architecture Modèle - Vue - Contrôleur (MVC)
  - Architecture Multi-tiers

Diagramme de composant

- **Définition**
- Permet d'illustrer la notion de la réutilisation,
- Il englobe des classes et des interfaces dont les classes sont masquées,
- Un composant doit fournir un service bien précis,
- Un composant est une unité autonome représentée par un classeur structuré, comportant une ou plusieurs interfaces requises ou offertes.

Diagramme de composant

Représentation de package

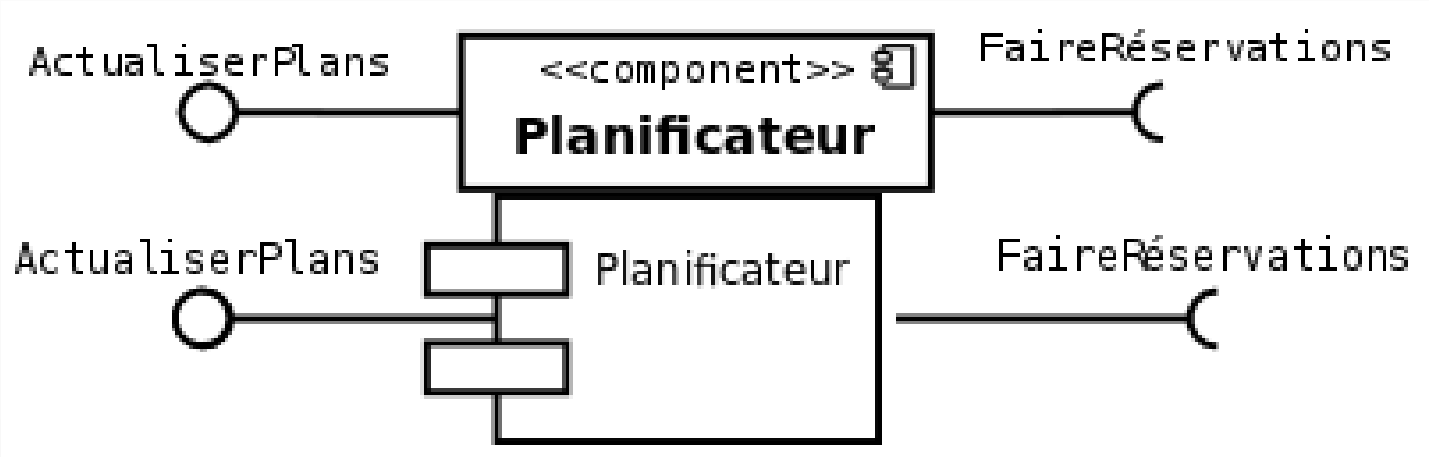


Diagramme de composant

1	Composant
2	Port d'interface fourni
3	Port d'interface requis
4	Dépendance
5	Élément
6	Assembly d'élément
7	Délégation
9	Commande Réduire/Développer

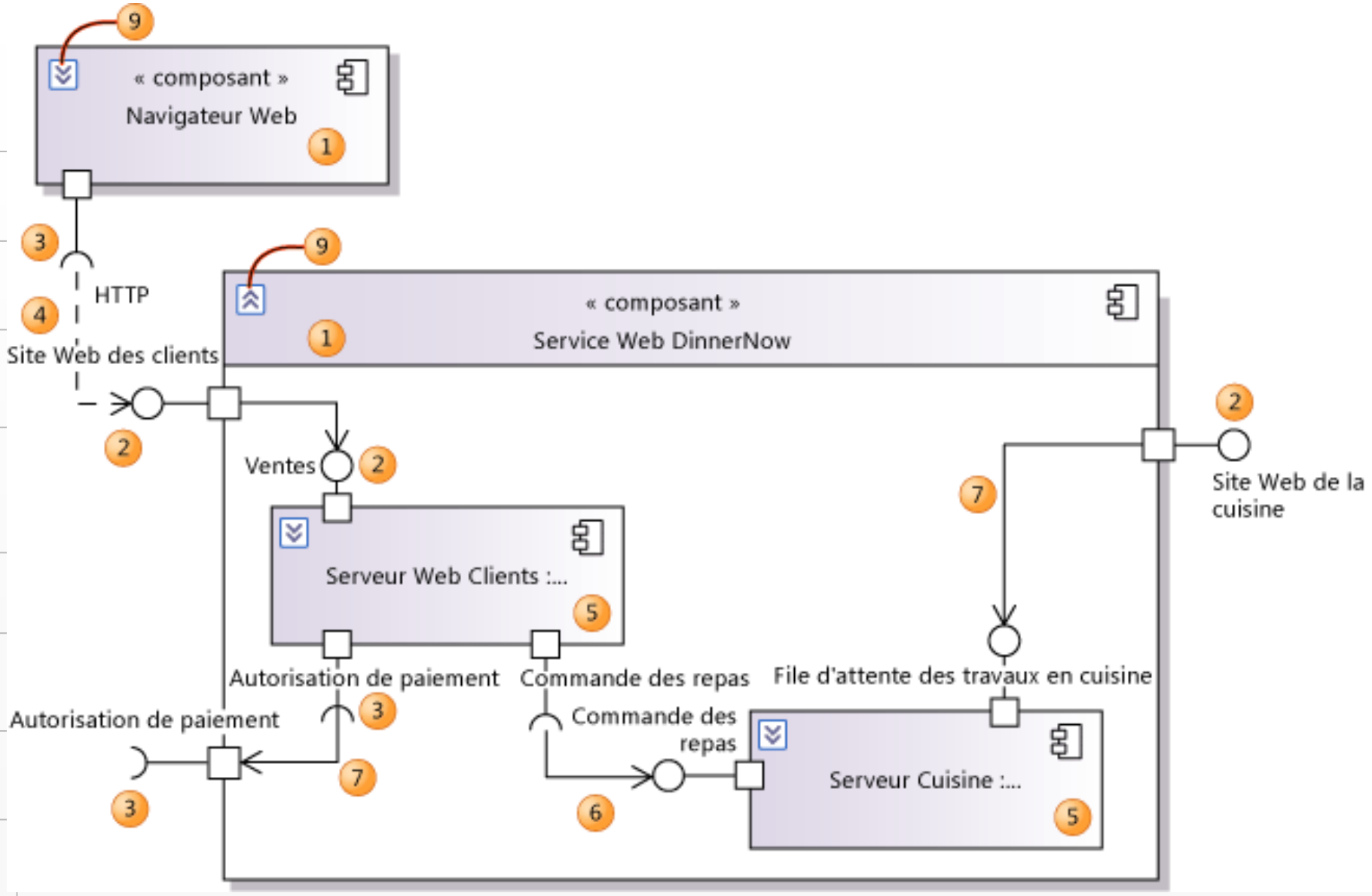


Diagramme de composant

Exemple

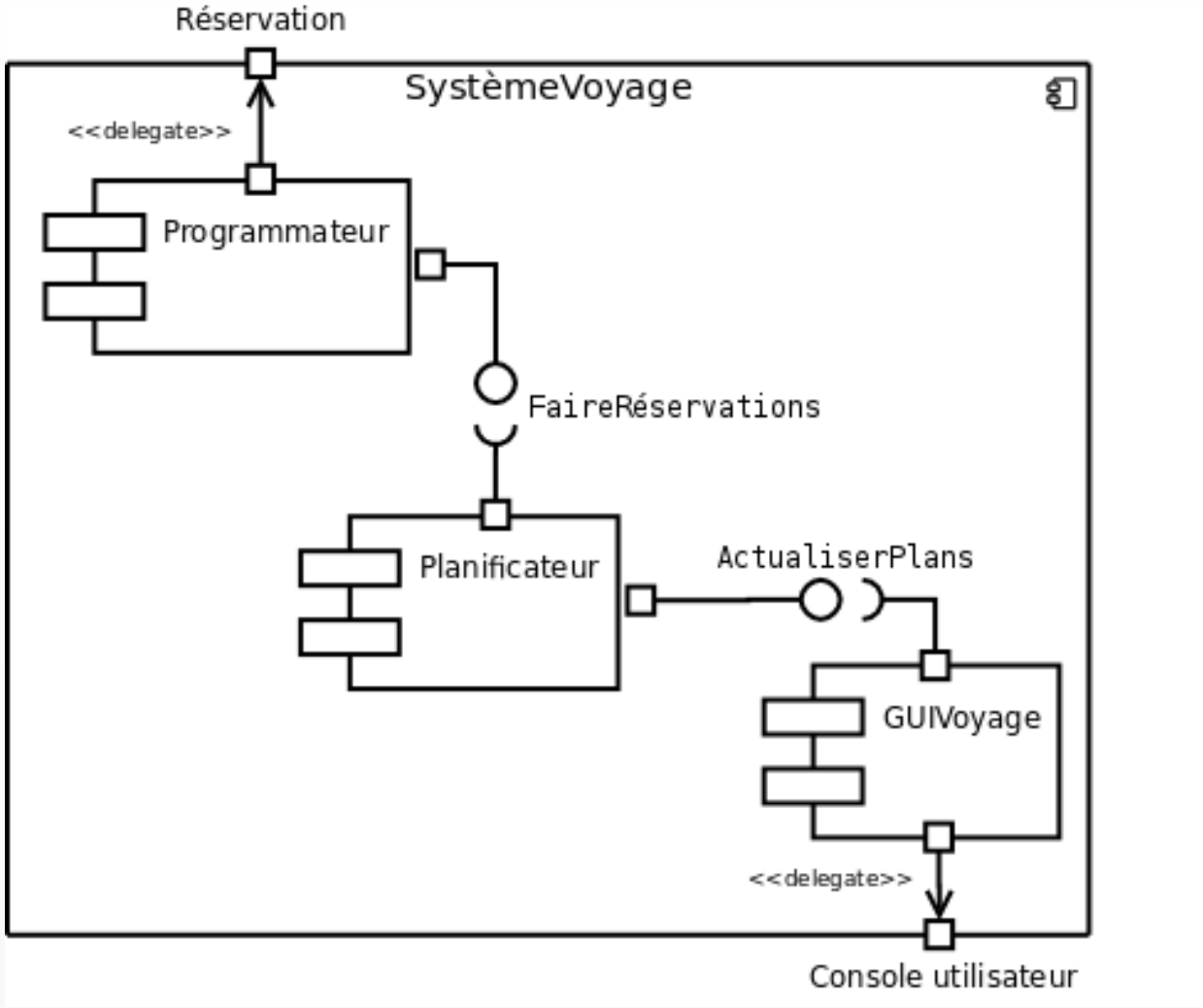


Diagramme de déploiement

- Définition
  - Illustre la disposition physique des différents matériels (ou nœuds) qui entrent dans la composition du système.
  - Illustre la répartition des composants au sein des nœuds.
- Un diagramme est composé :
  - dispositifs physiques (les nœuds),
  - de liens représentants les moyens de communication entre les nœuds (les supports de communication),
  - d'artefact.

Diagramme de déploiement

- Nœud :
  - Un nœud est un classeur et peut posséder des attributs (quantité de mémoire, vitesse du processeur…).
  - Une ressource est matérialisée par un nœud représenté par un cube comportant un nom .

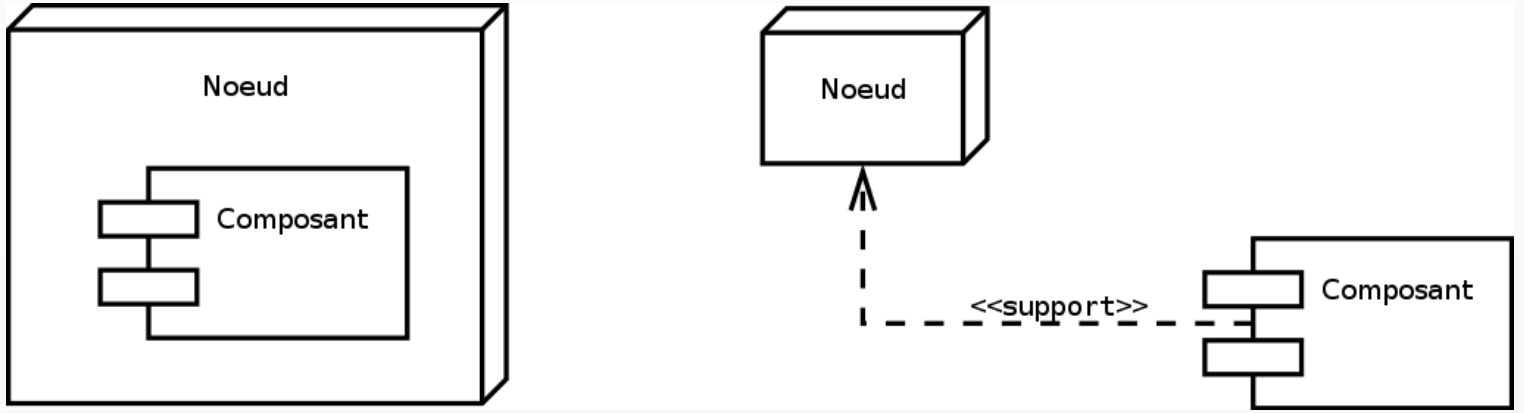


Diagramme de déploiement

- Support de communication:
  - Les supports de communication sont symbolisés par des relations entre les nœuds.

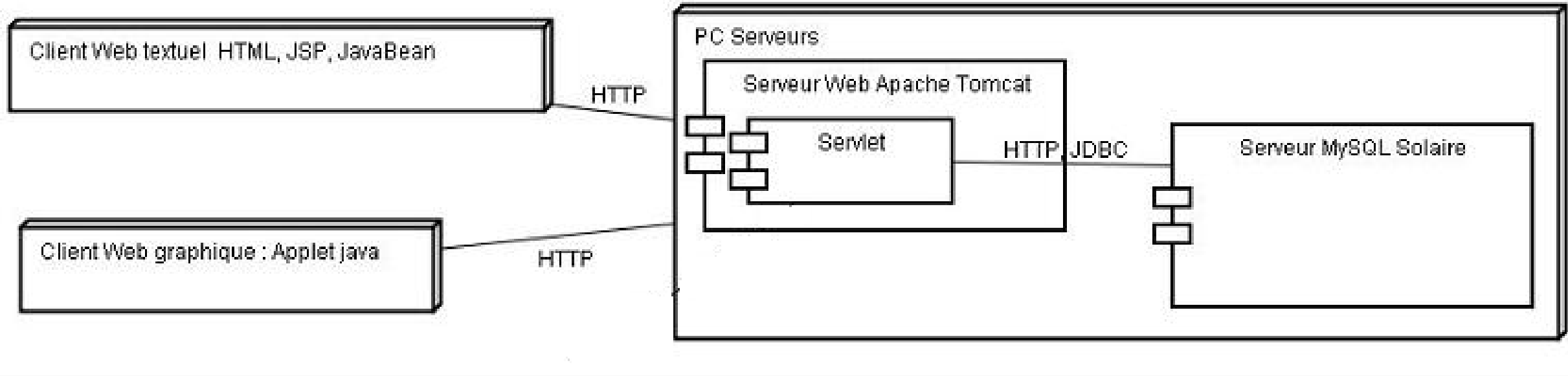


Diagramme de déploiement

- Les *artefacts*
  - Les *artefacts* sont des éléments de modèle qui représentent les entités physiques dans un système logiciel.
  - Les artefacts représentent des unités physiques d'implémentation, telles que des fichiers exécutables, des bibliothèques, des composants de logiciel, des documents, et bases de documents.

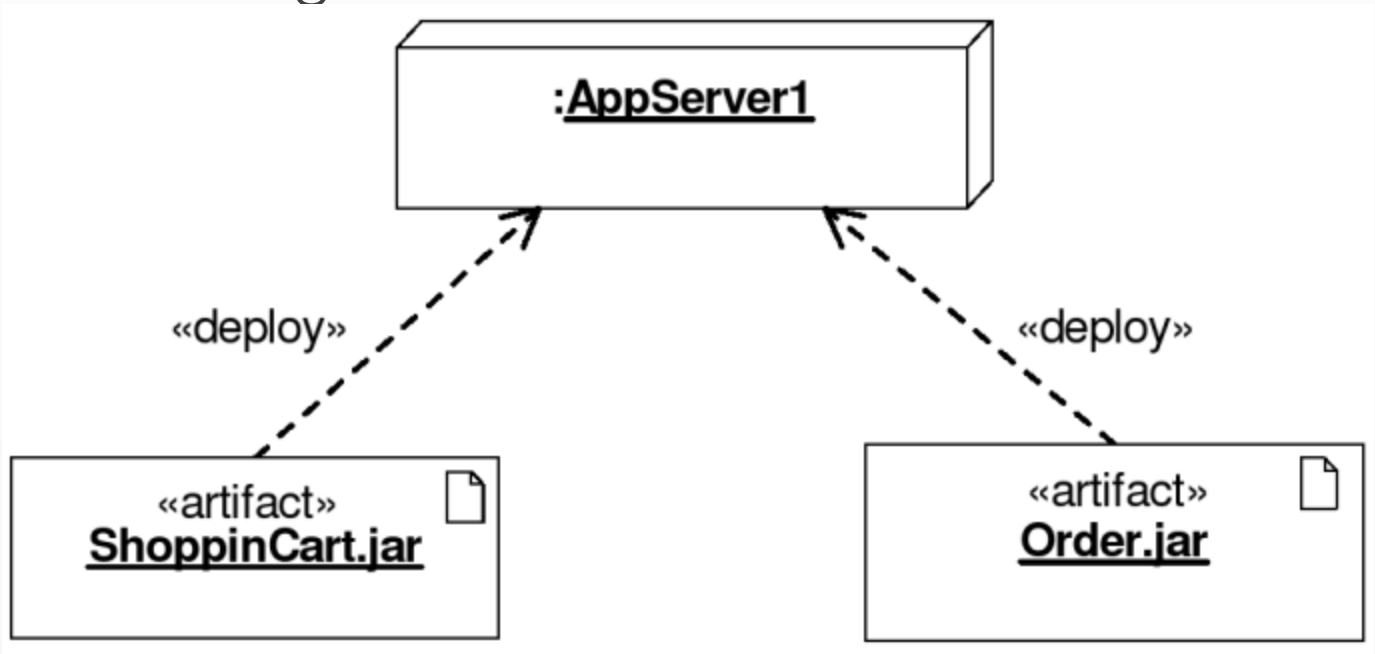


Diagramme de déploiement

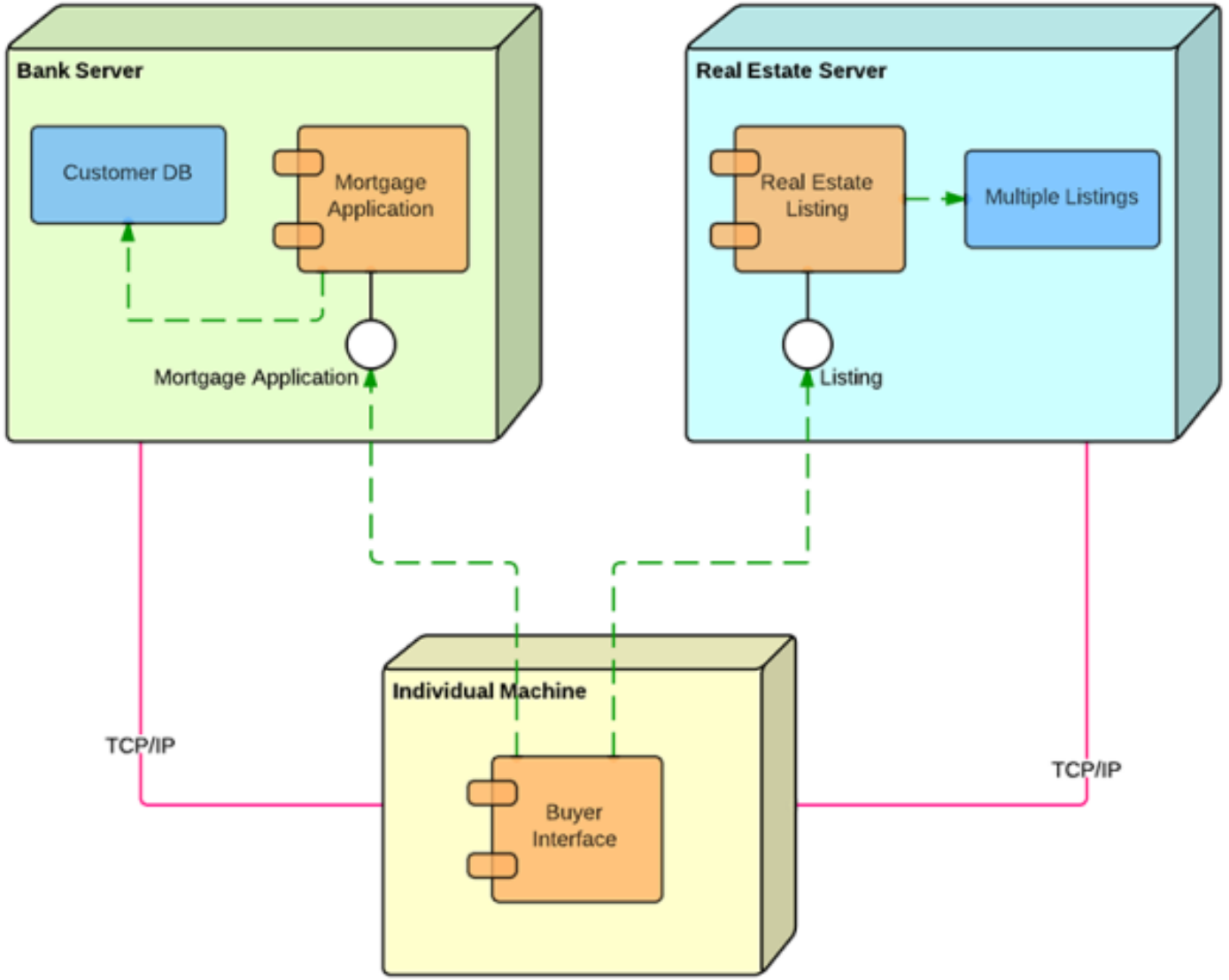


Diagramme de déploiement

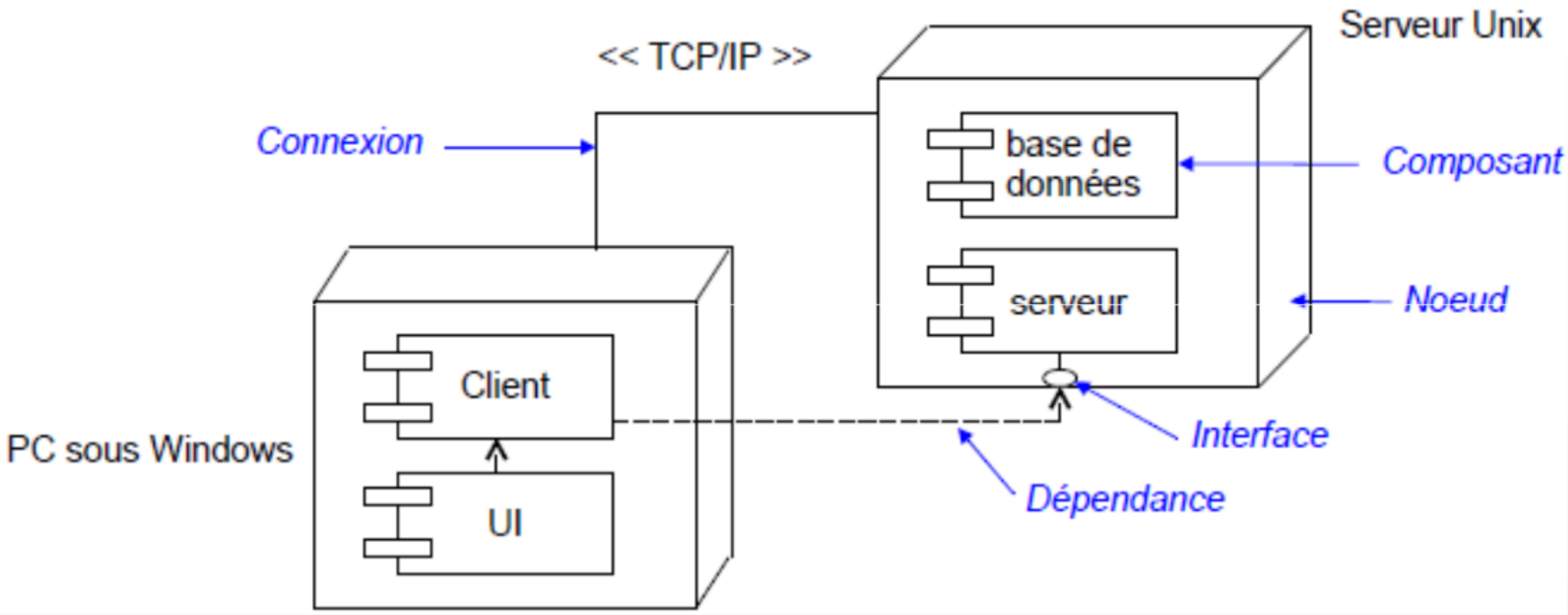


Diagramme structure composite: new

- Un ensemble d'instances qui communiquent et collaborent en Run-Time pour réaliser un objectif commun
- Une collaboration veut décrire un comportement de structure faite par la propriété de la structure
- Doit être raccordé uniquement avec des propriétés qui sont nécessaires pour effectuer son comportement décrit

Diagramme structure composite : new

- Deux notations UML :
  - **Collaboration** :

    Qui définit un ensemble de rôles de co-exploitation utilisés collectivement pour illustrer une fonctionnalité spécifique. Une collaboration ne doit afficher les rôles et les attributs nécessaires pour accomplir sa tâche ou une fonction définie.