Plan

Composant

- Un **composant** logiciel est une unité logicielle autonome au sein d'un système global ou d'un sous-système.
- C'est un classeur structuré particulier, muni d'une ou plusieurs interfaces requises ou offertes.

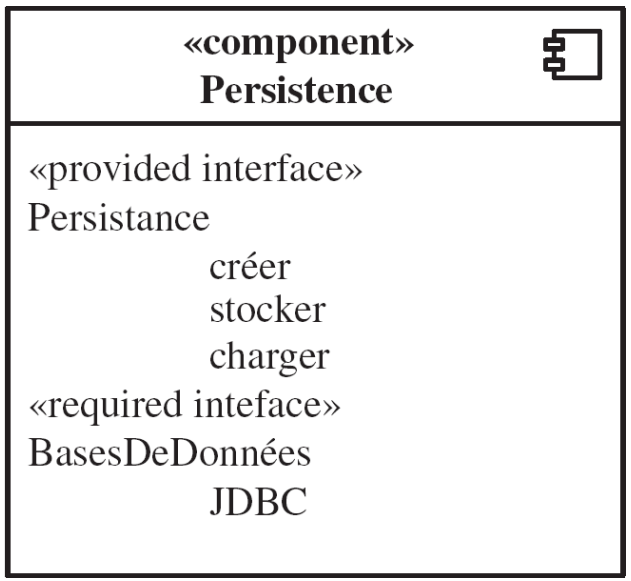
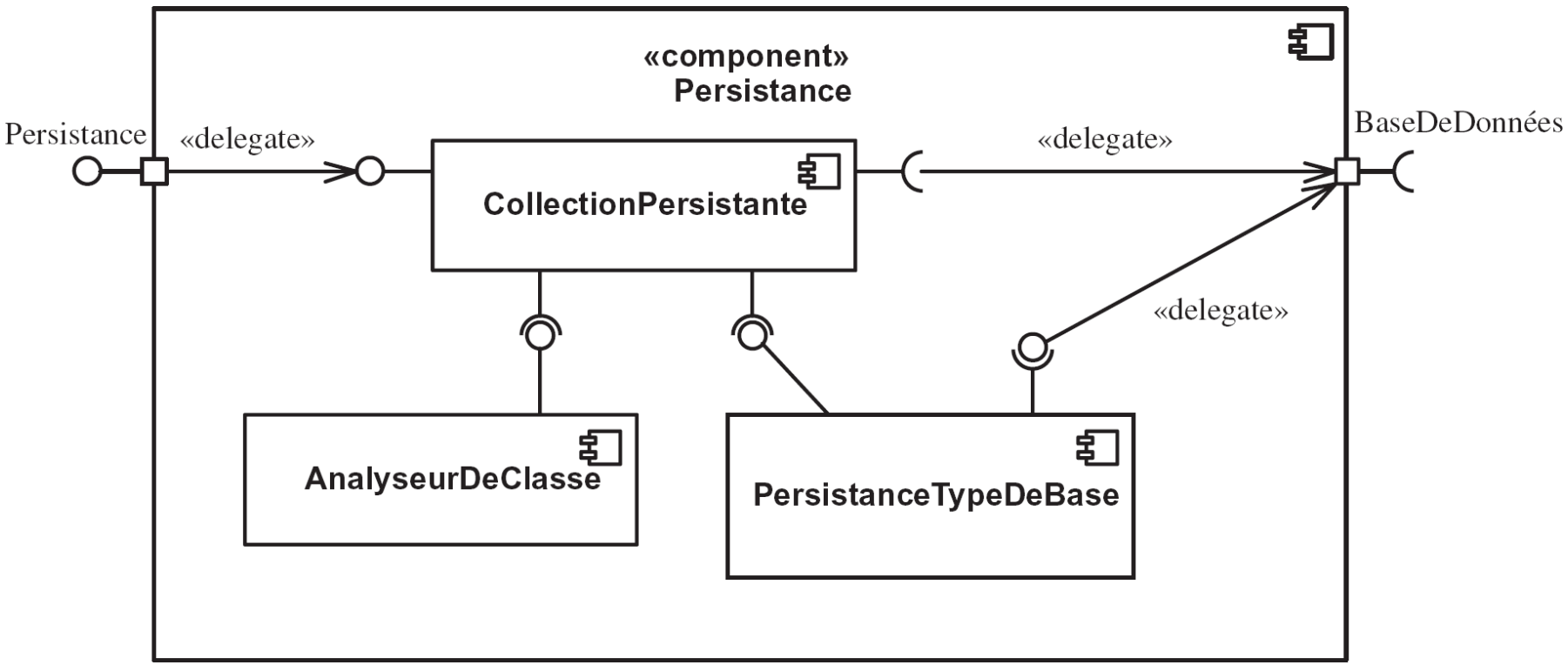


Diagramme de composant

- Les **diagrammes de composants** représentent l'architecture logicielle du système.
  - Les composants peuvent être décomposés en **sous-composants**, le cas échéant.
  - Les liens entre composants sont spécifiés à l'aide de **dépendances entre leurs interfaces**.
    - Le « câblage interne » d'un composant est spécifié par les **connecteurs de délégation**. Un tel connecteur connecte un port externe du composant avec un port de l'un de ses sous-composants internes.

Exemple de modélisation d'un composant



Intérêt des diagrammes de composants

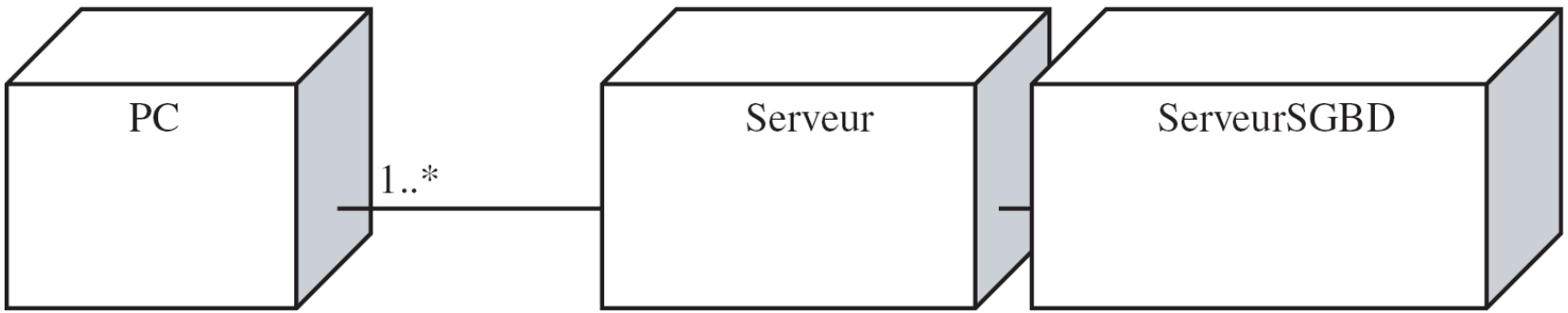
- Les diagrammes de composants permettent de
  - Structurer une architecture logicielle à un niveau de granularité moindre que les classes
    - Les composants peuvent contenir des classes.
  - Spécifier l'intégration de briques logicielles tierces (composants EJB, CORBA, COM+ ou .Net, WSDL...) ;
  - Identifier les composants réutilisables.
- Un composant est un espace de noms qu'on peut employer pour organiser les éléments de conception, les cas d'utilisation, les interactions et les artefacts de code.

Architecture matérielle

- En dernier lieu, un système doit s'exécuter sur des ressources matérielles dans un environnement matériel particulier.
- UML permet de représenter un environnement d'exécution ainsi que des ressources physiques (avec les parties du système qui s'y exécutent) grâce aux **diagrammes de déploiement**.
  - L'environnement d'exécution ou les ressources matérielles sont appelés « noeuds ».
  - Les parties d'un système qui s'exécutent sur un noeud sont appelées « artefacts ».

Noeud

- Un **noeud** est une ressource sur laquelle des artefacts peuvent être déployés pour être exécutés.
- C'est un classeur qui peut prendre des attributs.
  - Un noeud se représente par un cube dont le nom respecte la syntaxe des noms de classes.
  - Les noeuds peuvent être associés comme des classes et on peut spécifier des multiplicités.

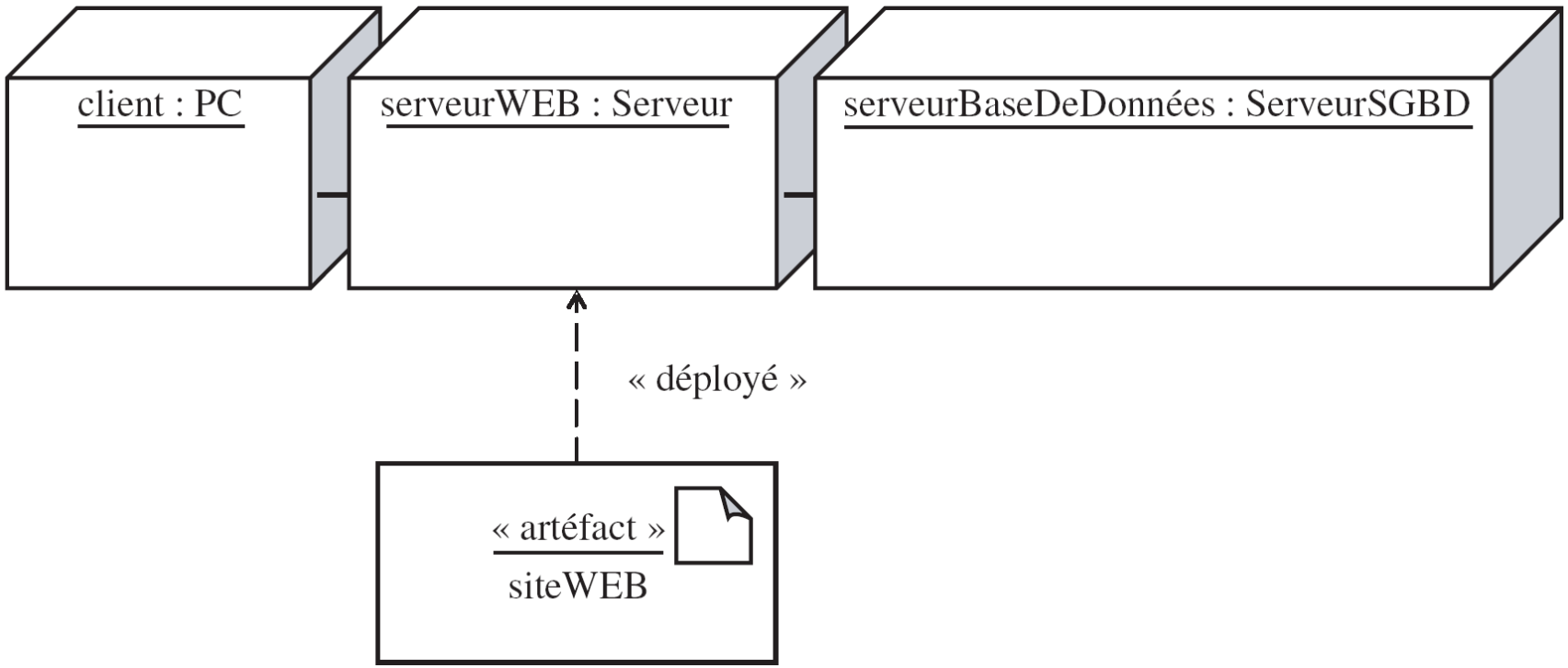


Artefact

- Un **artefact** est la spécification d'une entité physique du monde réel.
- Il se représente comme une classe par un rectangle contenant le mot-clé **artefact** suivi du nom de l'artefact.
  - Un artefact déployé sur un noeud est symbolisé par une flèche en trait pointillé qui porte le stéréotype déployé et qui pointe vers le noeud.
  - L'artefact peut aussi être inclus directement dans le cube représentant le noeud.
- Plusieurs stéréotypes standard existent pour les artefacts : **document**, **exécutable**, **fichier**, **librairie**, **source**.

Instanciation de noeuds et d'artefacts

- Les noms des instances de noeuds et d'artefacts sont soulignés.



Exécution des composants

- On utilise des flèches de dépendance pour montrer la capacité d'un noeud à prendre en charge un type de composant.