

Master mention Informatique  
Spécialité ISI

Génie Logiciel et modélisation  
M1 / 177EN003

C5 – Les packages, les diagrammes de composants,  
les diagrammes de déploiement, MDA

Claudine Piau-Toffolon

# Plan du cours

---

- ❑ Les paquetages (*packages in english*)
  - Définition
  - Visibilité, dépendances
  - Exemples d'utilisation
- ❑ Les diagrammes de composants
  - Définition
  - Dépendances
- ❑ Les diagrammes de déploiement
  - Définition
  - Connections

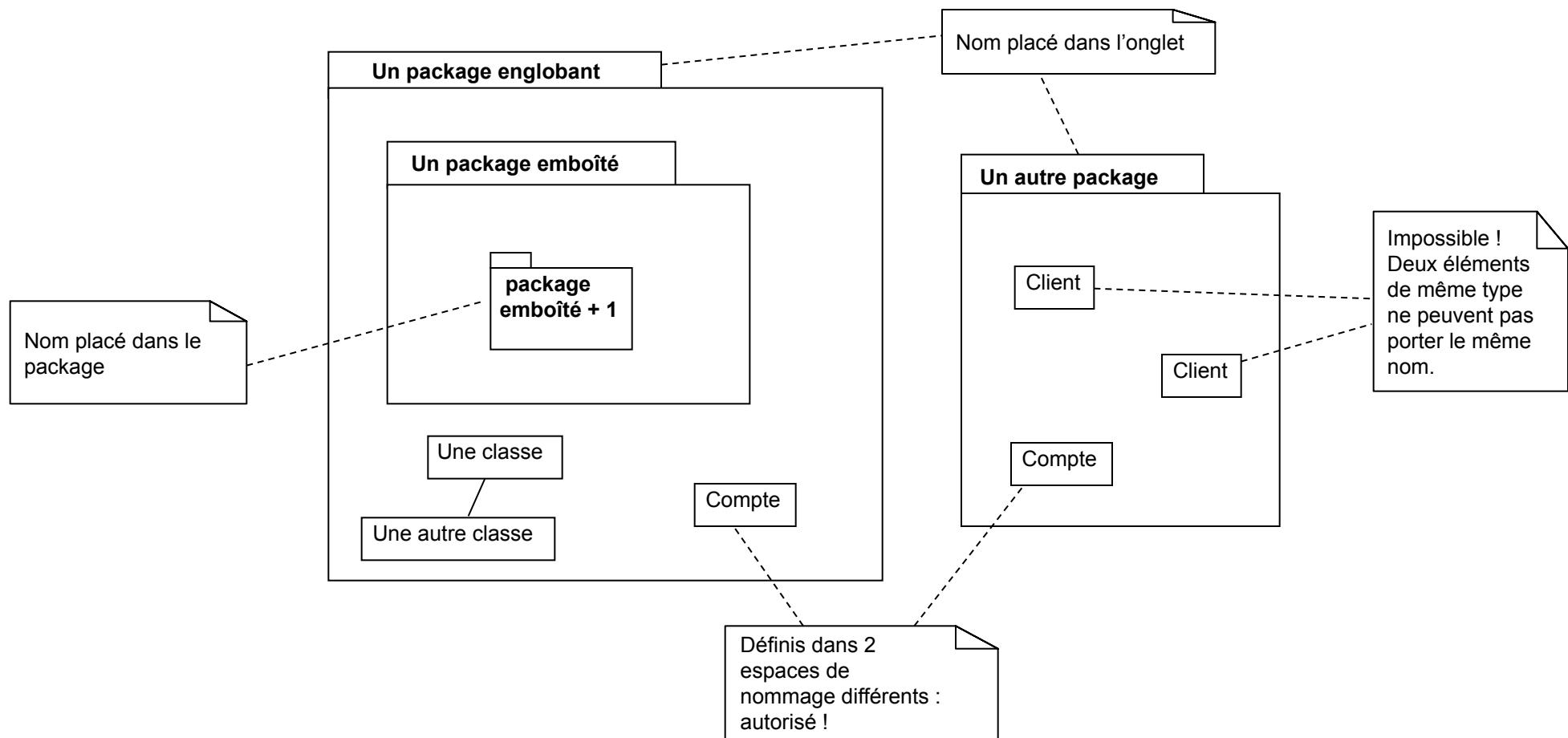
# Les paquetages – Définition (1/2)

---

- Définition : « boîte » englobant des éléments de modélisation tels que
  - des *packages* emboîtés
  - des modèles (sous forme des différents types de diagrammes UML)
  
- A quoi ça sert ? A organiser des modèles, par exemple :
  - un *package* pour les modèles d'analyse contenant
    - un *package* pour les cas d'utilisations,
    - un *package* pour le modèle du domaine,
    - un *package* pour les diagrammes de séquence, etc.
  - un *package* pour les modèles de conception (architecture logique)
  - un *package* pour les modèles de composants (architecture logicielle)
  - un *package* pour les modèles de déploiement (architecture matérielle)
  
- Les éléments d'un *package* :
  - Les éléments d'un package sont liés sémantiquement
  - Un package définit un espace de nommage : les noms des éléments doivent donc être uniques au sein du package
  - Un package définit publics certains de ses éléments et importe des éléments publics d'autres packages

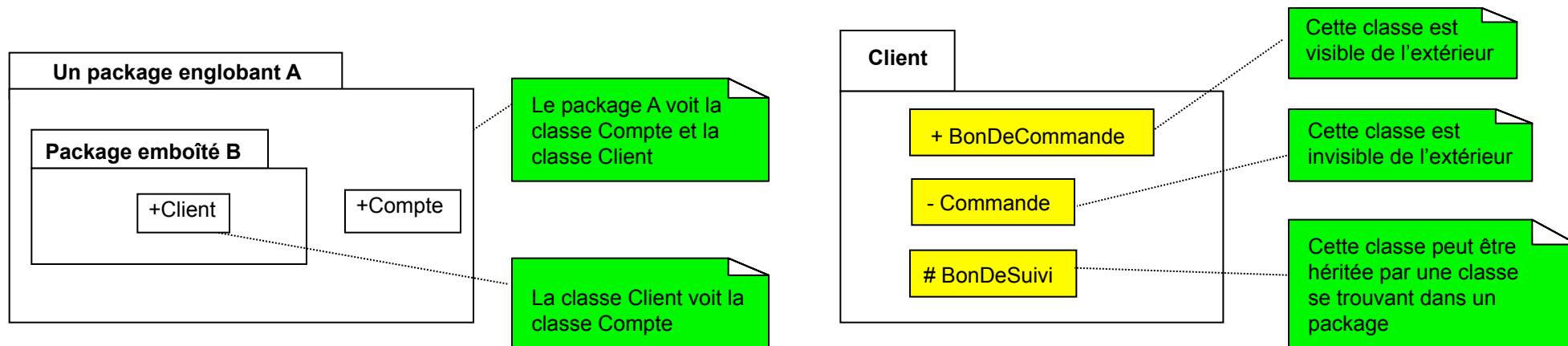
# Les packages – Définition (2/2)

- ❑ Liens entre packages : un package peut faire partie d'un réseau de relation de dépendances.
- ❑ Représentation UML d'un package :



# Les packages – Visibilité

- Un package emboîté voit tout élément du package englobant quelque soit son degré d'emboîtement
- Un package voit tous les éléments publics des packages emboîtés
- On peut préciser la visibilité des éléments contenus dans un package :
  - Un élément public (noté +) est visible pour tout autre élément qui voit le package conteneur (y compris dans le package englobant ou les packages emboîtés)
  - Un élément privé (noté -) est uniquement visible par les autres éléments contenus dans le package conteneur
  - Un élément protégé (noté #) est invisible de l'extérieur sauf dans un package dont un élément hérite de cet élément protégé
  - Un élément de visibilité package (noté ~) est visible uniquement par les éléments du même package



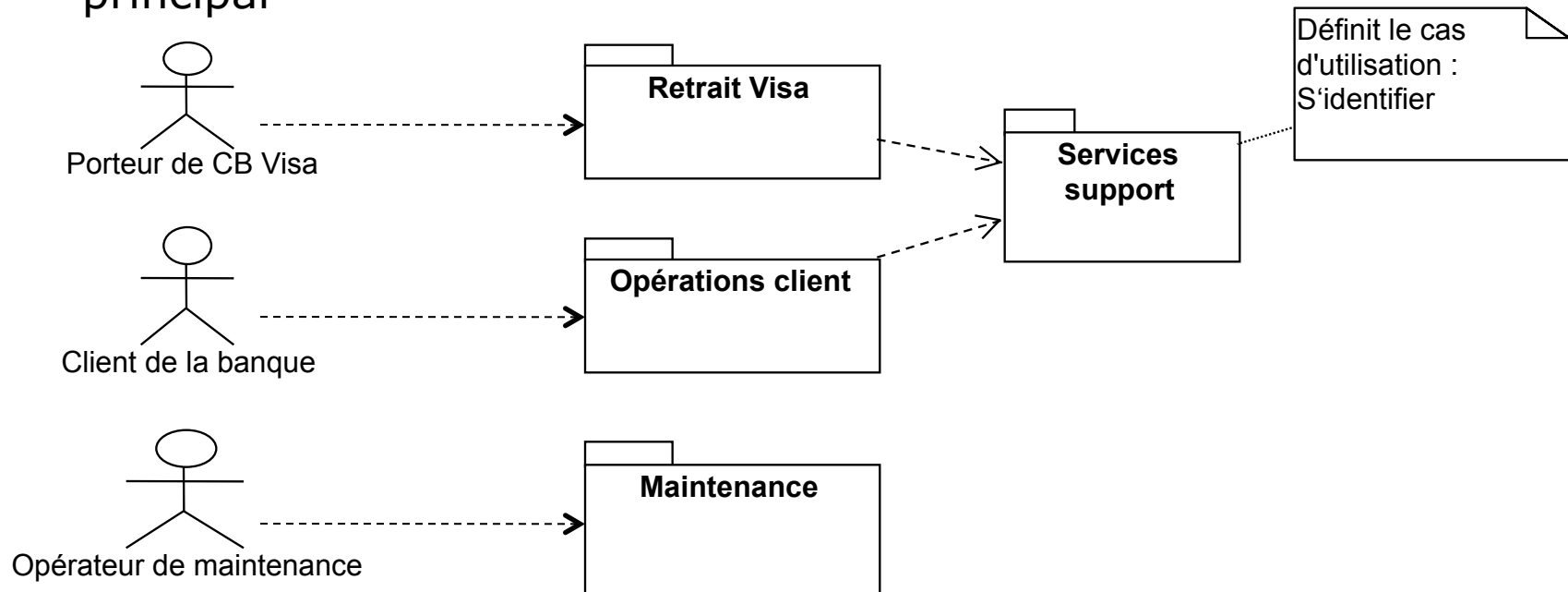
# Les packages – Dépendances

---

- ❑ Rappel : les relations de dépendance modélise des relations unidirectionnelles du type « *utilise* » entre deux éléments modélisés
- ❑ Une relation de dépendance entre deux packages indique qu'au moins un des éléments d'un package utilise un autre élément de l'autre package
- ❑ Les dépendances d'importation :
  - L'importation simplifie l'écriture des noms : il n'est plus nécessaire de qualifier les éléments (publics) pour y accéder
  - « *import* » : ajoute les éléments publics du package importé à l'espace de nommage du package importateur ; les éléments importés sont visibles pour tout élément voyant le package importateur
  - « *access* » : idem mais les éléments publics importés deviennent privés dans le package importateur
  - Les dépendances d'importation ne sont pas transitives : une importation ne permet pas d'importer les relations d'importations mais seulement la partie public du package importé

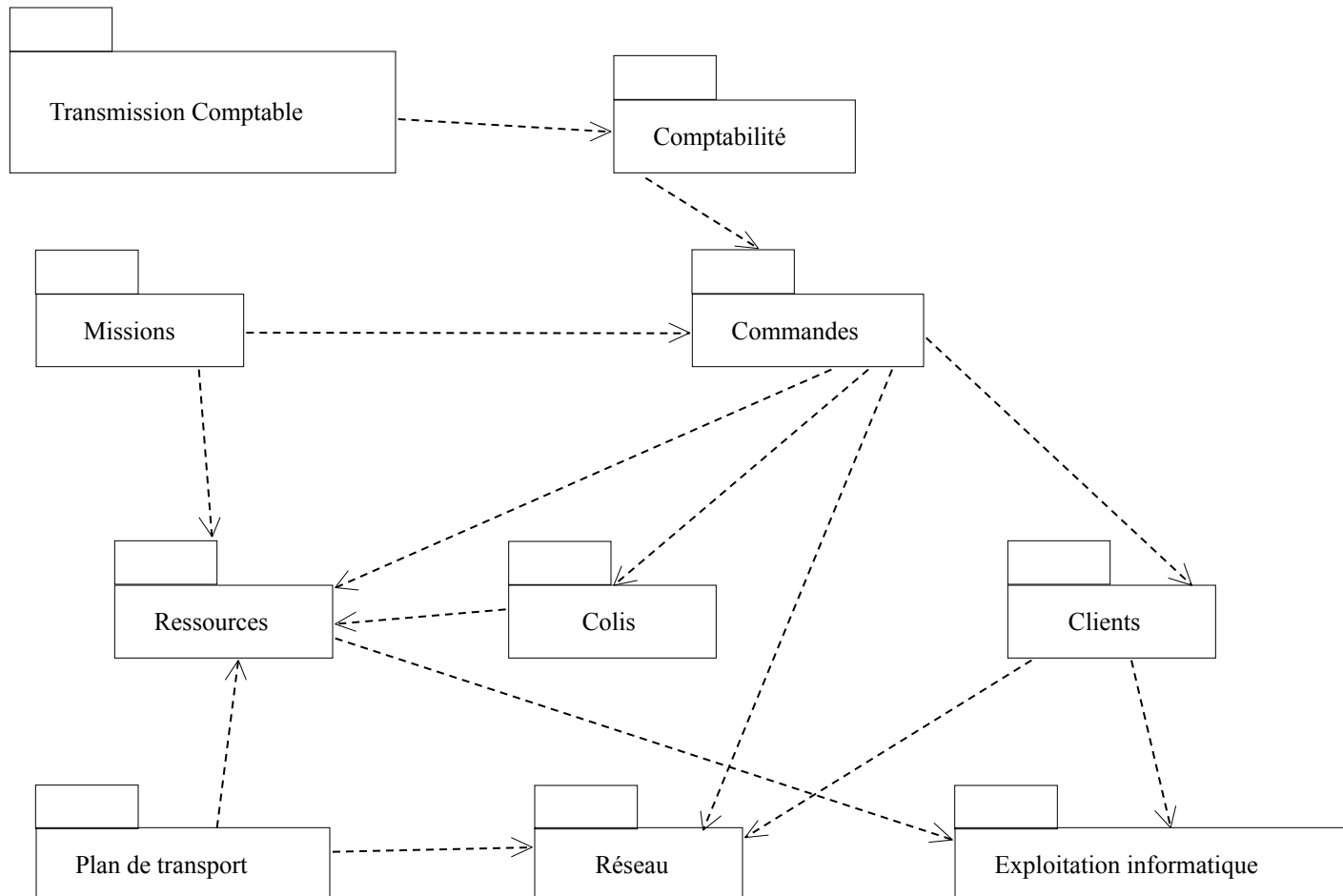
# Les packages – Exemples (1/2)

- Les packages peuvent servir à organiser :
  - Des cas d'utilisation
  - Des diagrammes d'analyse
  - Des diagrammes de conception
  - Etc.
- Exemple du GAB : organisation des cas d'utilisation par acteur principal



## Les packages – Exemples (2/2)

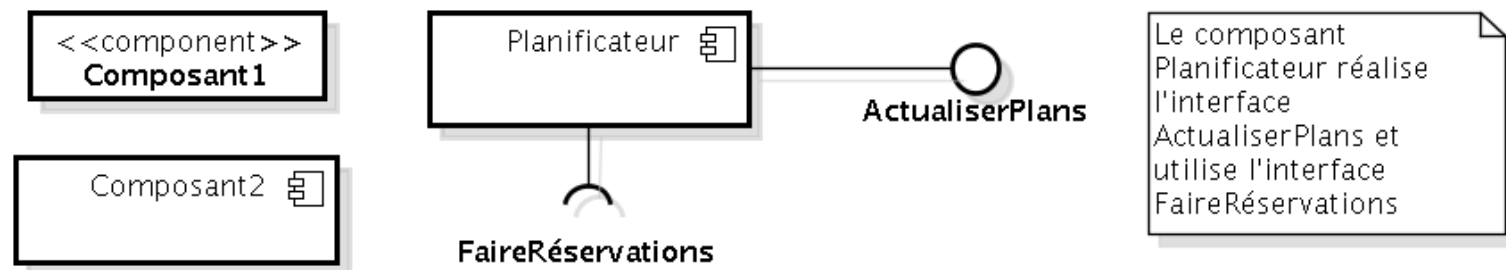
- Exemple d'un réseau de dépendance : il ne doit pas y avoir de cycle entre les packages afin qu'ils soient indépendant les uns des autres



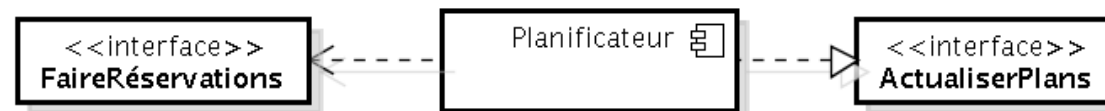


# Les diagrammes de composants – Définition (1/2)

- ❑ Les diagrammes de composants décrivent les **composants** et leurs dépendances
- ❑ Dans UML, un système informatique est un assemblage de composants interchangeables : un composant est une partie logicielle du système
- ❑ Les diagrammes de composants servent donc à représenter la structure logicielle d'un système
- ❑ Un composant peut être lui-même constitué de « sous-composants »
- ❑ Un composant réalise des interfaces et peut utiliser des interfaces réalisées par d'autres composants



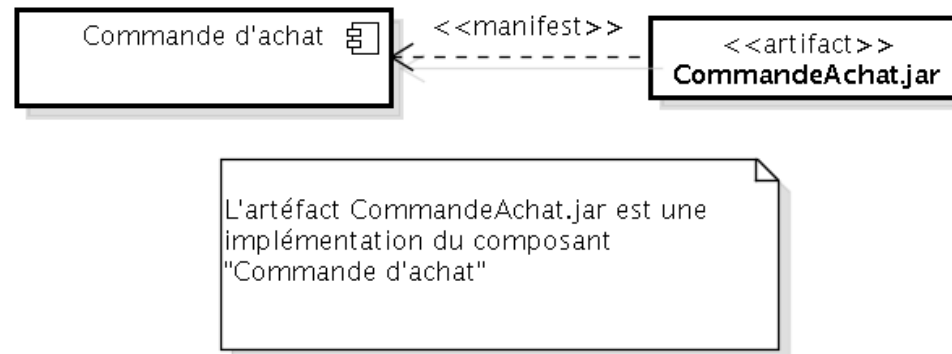
powered by astah\*



powered by astah\*

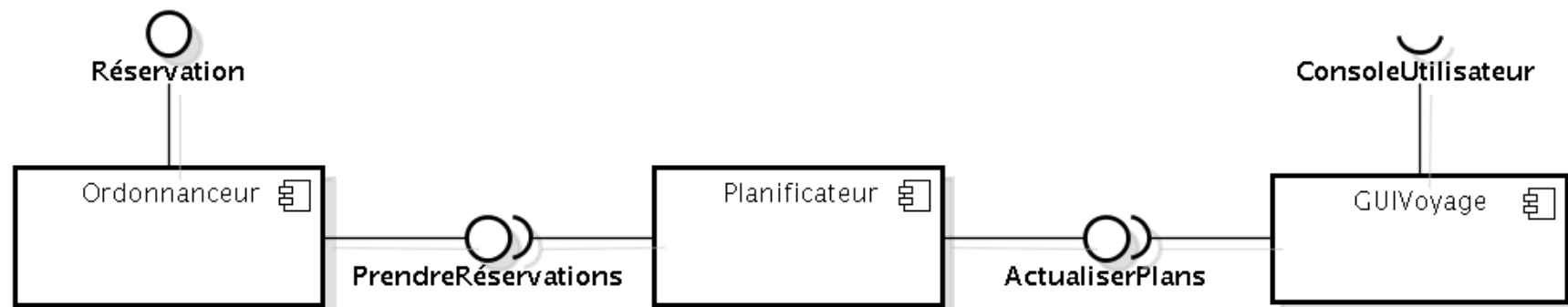
# Les diagrammes de composants – Définition (2/2)

- ❑ Un composant est implémenté par un ou plusieurs artéfacts
- ❑ Un artéfact est un élément physique du monde réel tel que :
  - du code (fichiers source, exécutable ou interprétable)
  - des scripts ou des fichiers de commande
  - un fichier de configuration
  - une table de base de données
  - un fichier de modèle UML (au formalisme XMI par exemple)
  - n'importe quel fichier en général
- ❑ Les instances d'artéfacts implémentant un composant sont déployés dans l'environnement de développement ou d'exploitation



# Les diagrammes de composants – Dépendances

- Les dépendances entre composants :
  - Les relations de dépendances entre composants portent sur les interfaces réalisées ou utilisées par ces composants.



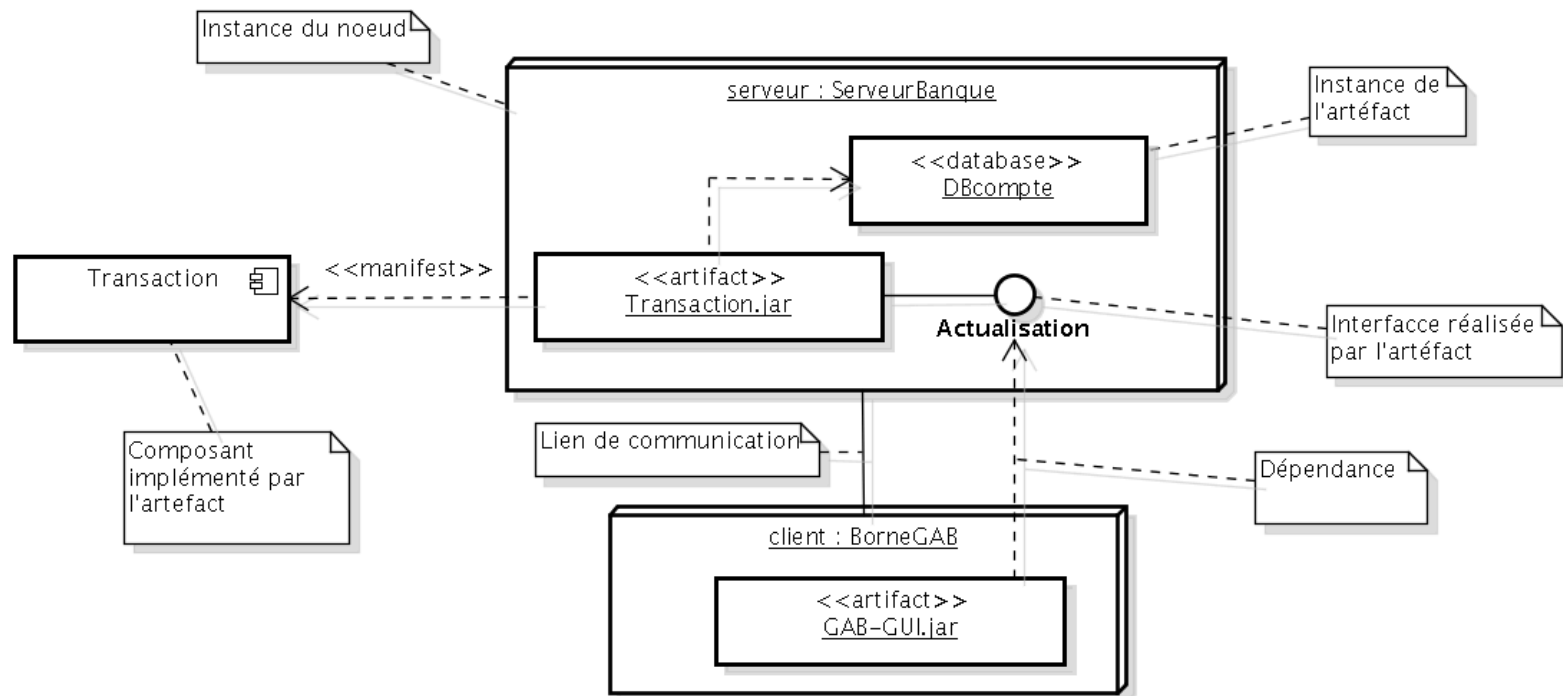
Le composant Ordonnanceur fournit les interfaces Réservation et PrendreRéservations.

Le composant Planificateur fournit l'interface ActualiserPlans et utilise l'interface PrendreRéservations.

Le composant GUIVoyage utilise les interfaces ActualiserPlans et ConsoleUtilisateur

# Les diagrammes de déploiement - Définition

- Le déploiement permet :
  - de décrire les ressources matérielles : **les noeuds**
  - et de décrire la répartition physique des artéfacts dans ces ressources : la **topologie** du système
- Un nœud représente une ressource matérielle qui existe au moment de l'exécution, qui possède des capacités de traitement et qui déploie des instances d'artéfacts :



# Les diagrammes de déploiement - Connections

- Il est possible d'organiser des nœuds en précisant leurs relations de dépendance, de généralisation et d'association.
- Généralement, les nœuds sont reliés (associés) par des chemins de communication (réseaux d'interconnection).

