

# UML : diagrammes de composants et de déploiement

**Achref El Mouelhi**

Docteur de l'université d'Aix-Marseille  
Chercheur en programmation par contrainte (IA)  
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



- 1 Le diagramme de composants
- 2 Le diagramme de déploiement

## Le diagramme de composants

- Un diagramme de structure d'UML (statique)
- Décrit les dépendances physique et statique d'une application en terme de composants : fichiers sources (.java, .cpp, .h, .cs...) librairies (dll, jar...), exécutables...
- Utilisé pour éviter de parler de classes, ou de packages

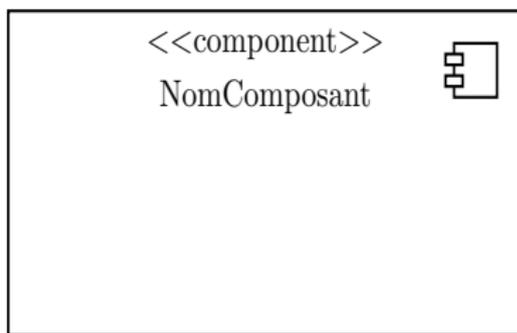
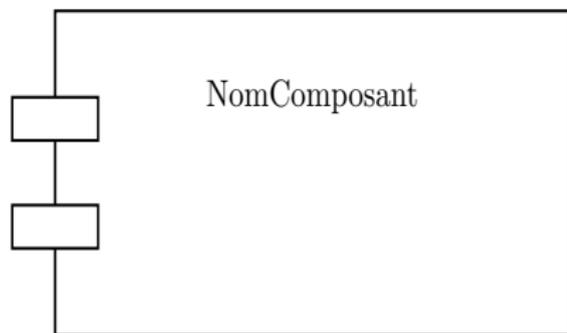
## Mots-clés associés

- Composant
- Interface (identique à celle du diagramme de classe)
- Port
- Dépendance (association identique à celle du diagramme de classe)

## Composant ?

- Unité autonome fournissant un service bien précis
- Représenté en UML par un classeur [+ un stéréotype `<<component>>`]
- Comportant une ou plusieurs interfaces requises ou offertes
- Pouvant contenir d'autres composants
- Composé de deux parties :
  - une partie masquée : ensemble de classes
  - une partie visible : ensemble d'interfaces

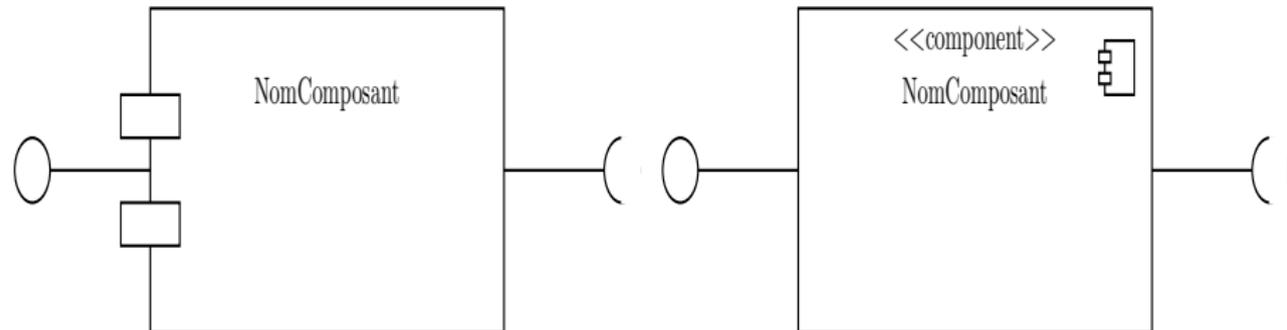
## Deux présentations possibles d'un composant en UML



### Remarque

Pas besoin d'un stéréotype <<component>> dans la première représentation

## Pour visualiser les interfaces requise et offerte

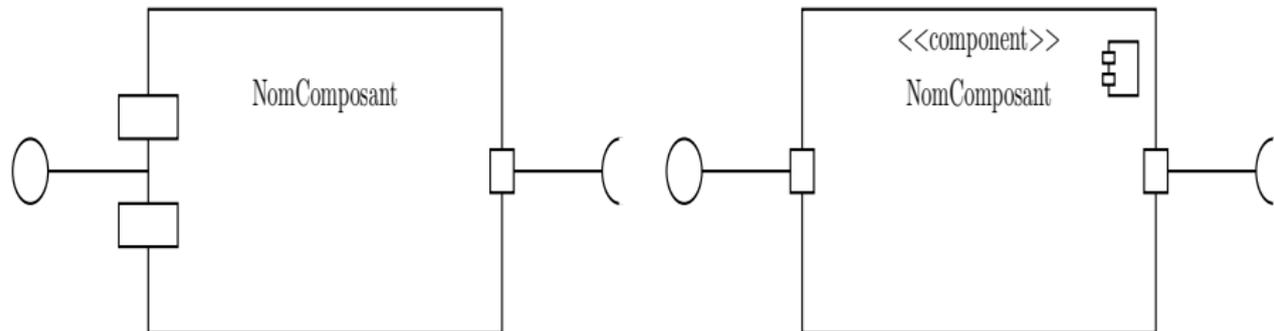


### Explication

- interface requise : représentée par un demi-cercle
- interface offerte : représentée par un cercle

## UML

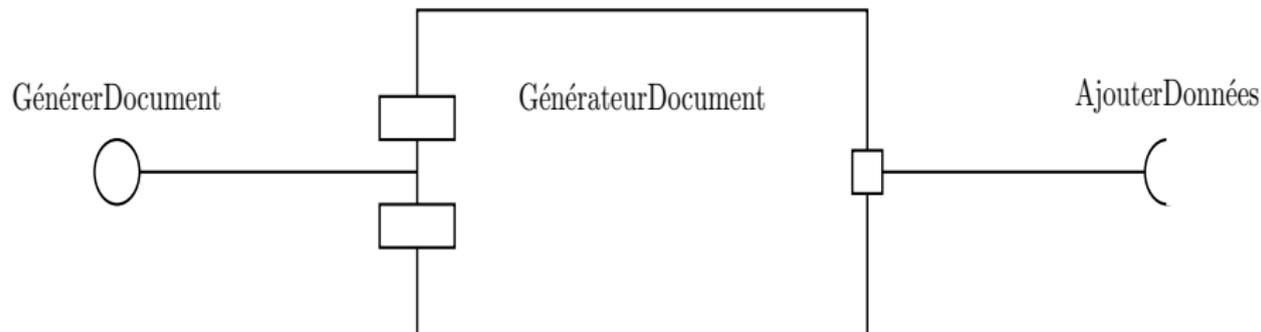
On peut visualiser explicitement les ports (emplacement variable)



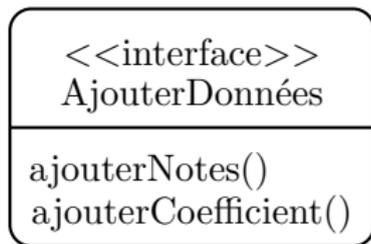
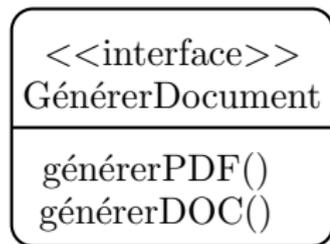
## Port ?

- élément graphique
- point de jonction entre le composant (le classeur) et l'interface (il peut être lié directement à un port du composant englobant)

## Exemple d'un composant permettant de générer de document (diplôme, attestation de réussite...) pour les étudiants



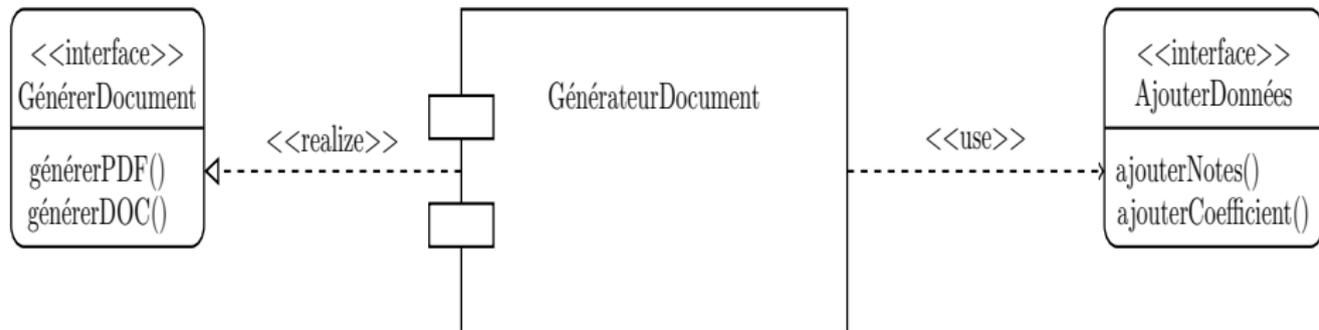
### Les interfaces :



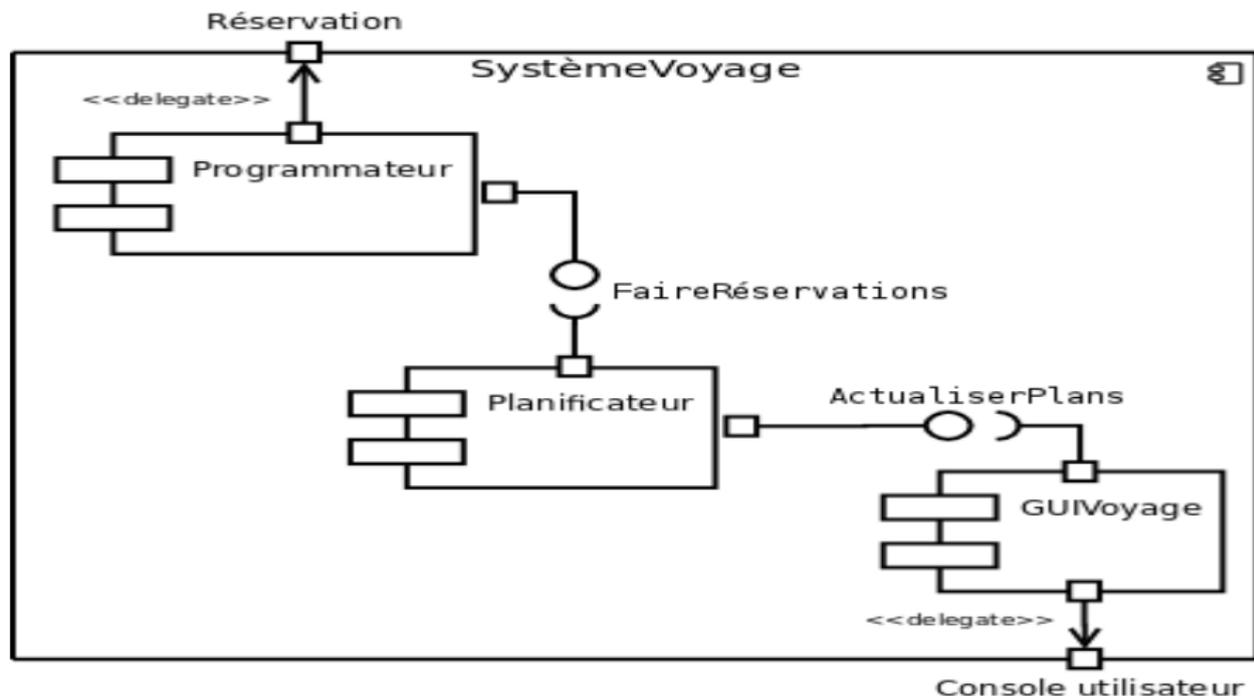
## Explication

- le nom du composant : `GénérateurDocument`
- interface requise : `AjouterDonnées`
- interface offerte : `GénérerDocument`
- `GénérateurDocument` **utilise et dépend de** `AjouterDonnées` : une relation de dépendance
- `GénérateurDocument` **réalise et implémente** les méthodes de `GénérerDocument` : une relation d'héritage

## On peut représenter explicitement la relation avec les interfaces

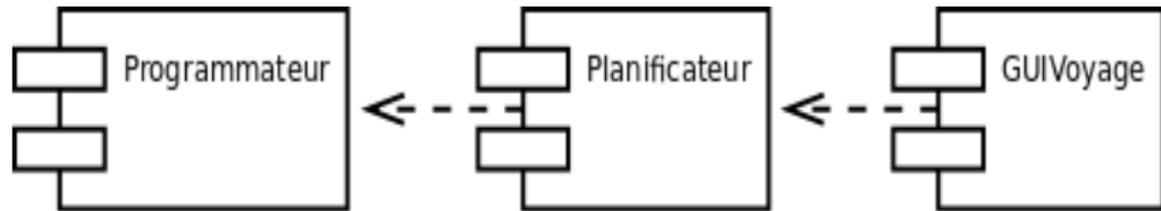


## Un composant peut être composé de plusieurs autres composants



(Source : UML 2 de l'apprentissage à la pratique de Laurent Audibert)

**On peut aussi montrer les dépendances entre les composants sans mentionner les interfaces**



*(Source : UML 2 de l'apprentissage à la pratique de Laurent Audibert)*

## Le diagramme de déploiement

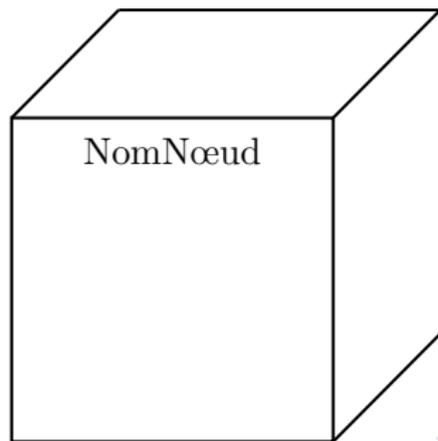
- Un diagramme de structure d'UML (statique)
- Décrivant l'architecture physique ainsi que les relations entre les composants logiciels et matériels d'une application
- Expliquant le déploiement de l'application en terme de réseau et communication

## Mots-clés associés

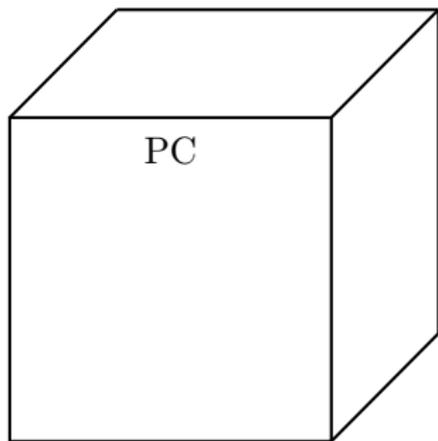
- Nœud
- Lien
- Stéréotype
- Artefact
- Composant

## Nœud ?

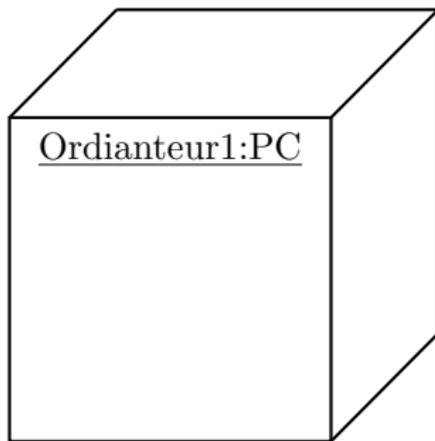
- Composant mécanique : ordinateur, serveur, imprimante, environnement d'exécution...
- Pouvant contenir d'autres nœuds ou artefacts
- Représenté par des cubes



## Exemple d'un nœud et une instance de nœud



Un nœud



Une instance de nœud

# UML

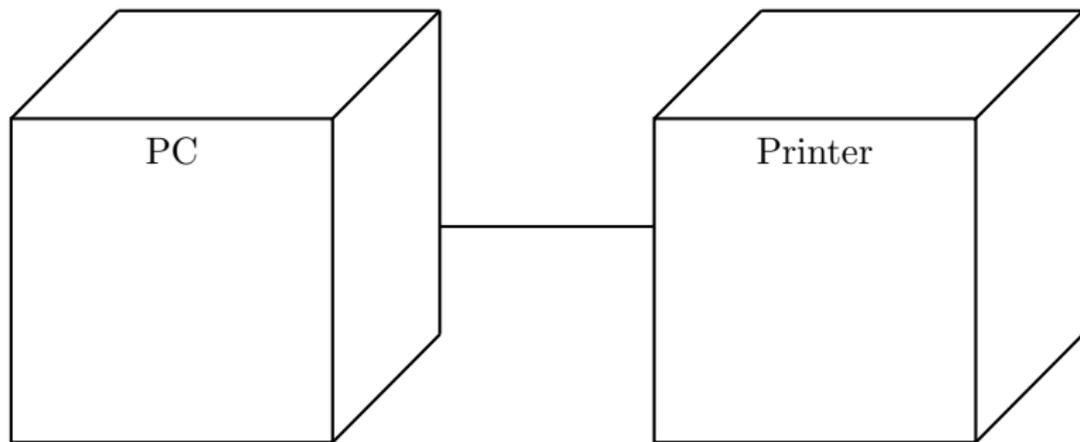
Lien ?

élément permettant de connecter les nœuds

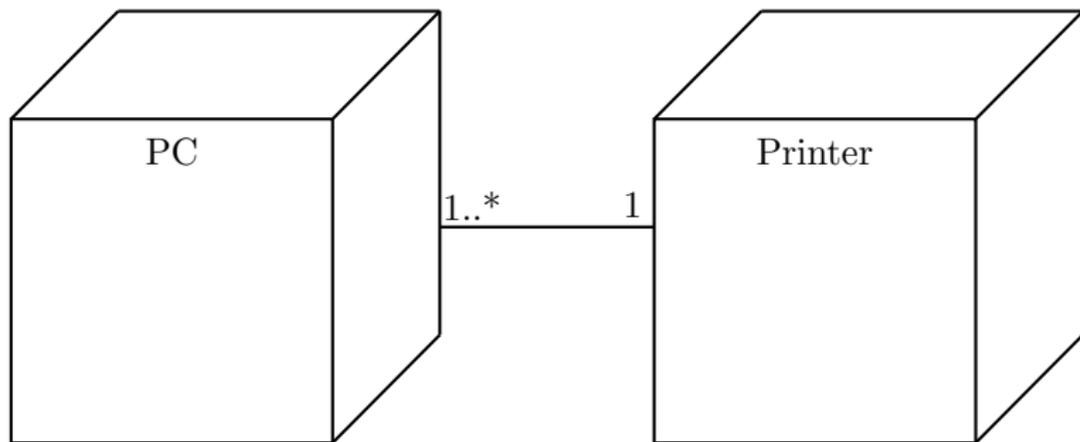
## UML

Lien ?

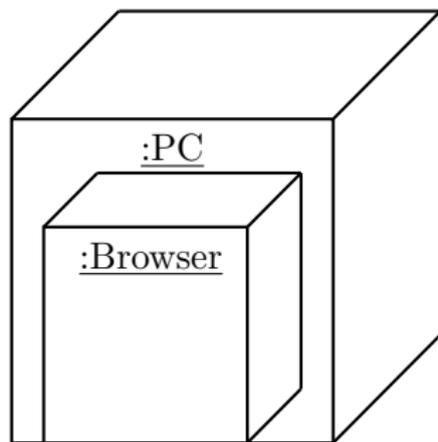
élément permettant de connecter les nœuds



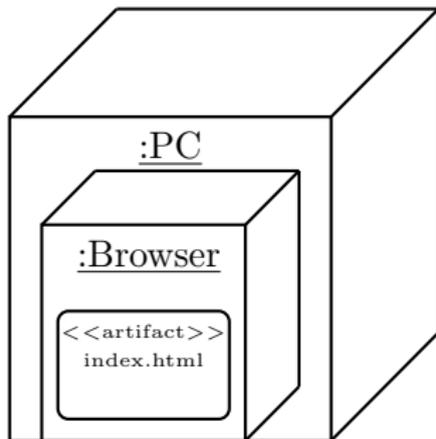
## On peut aussi définir la multiplicité



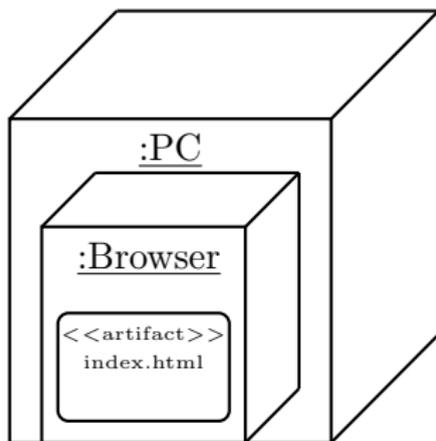
## Un nœud peut contenir d'autres nœuds



## Un nœud peut contenir aussi un artefact



## Un nœud peut contenir aussi un artefact



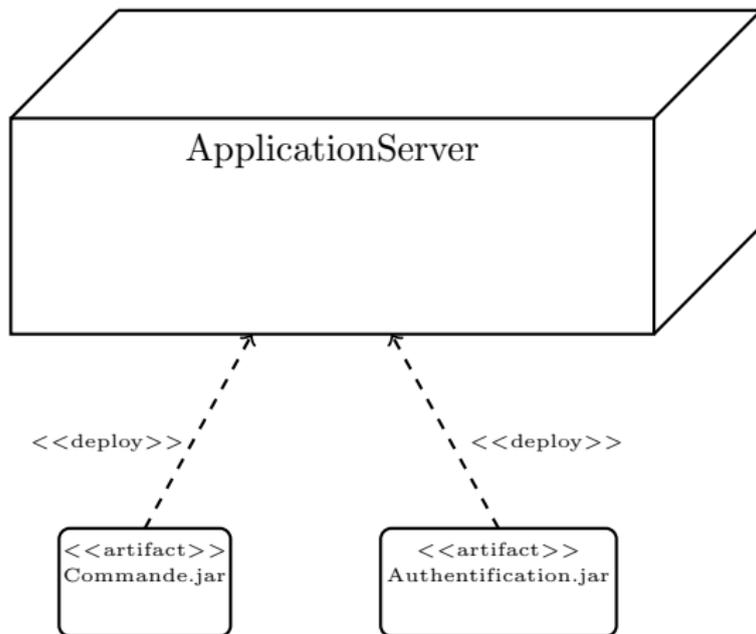
### Artefact ?

- Élément concret de l'application (fichier contenant du code source, table d'une base de données, script...)
- Pouvant `manifest` : résulter et implémenter un élément de modèle.

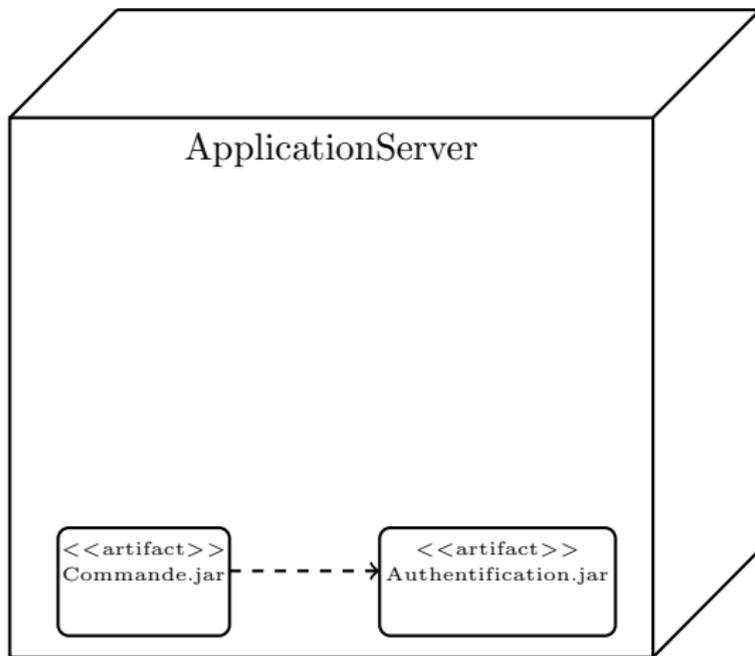
## Un artefact qui manifeste un composant



Un artefact déployé dans un nœud peut être représenté en utilisant le stéréotype `<<deploy>>`



## On peut aussi les représenter à l'intérieur du nœud



## Un exemple de diagramme de déploiement d'une application web

