Design patterns

Catalogue de patrons de conception

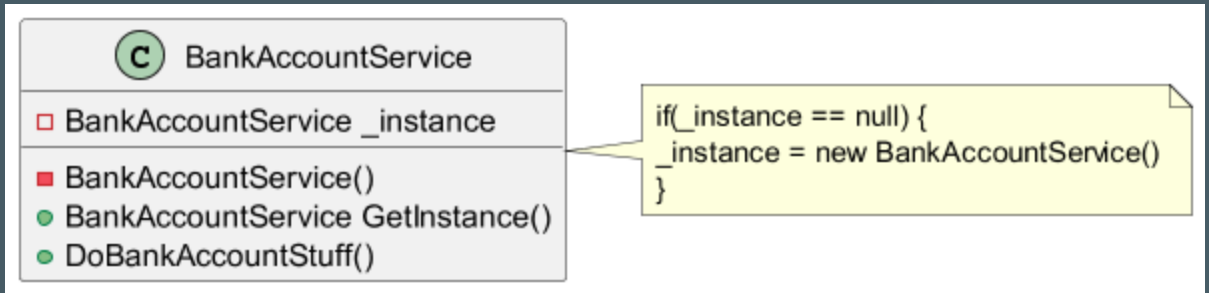
- Design patterns: Elements of reusable object oriented software 1995 - Gang of four
- 23 patrons de conception
- 3 catégories:
  - Création
  - Comportementaux
  - Structure

=> Répond de façon générique à un problème courant

Création: Singleton

Problème: Comment garantir l'unicité de l'instance d'une classe ?

Création: Singleton



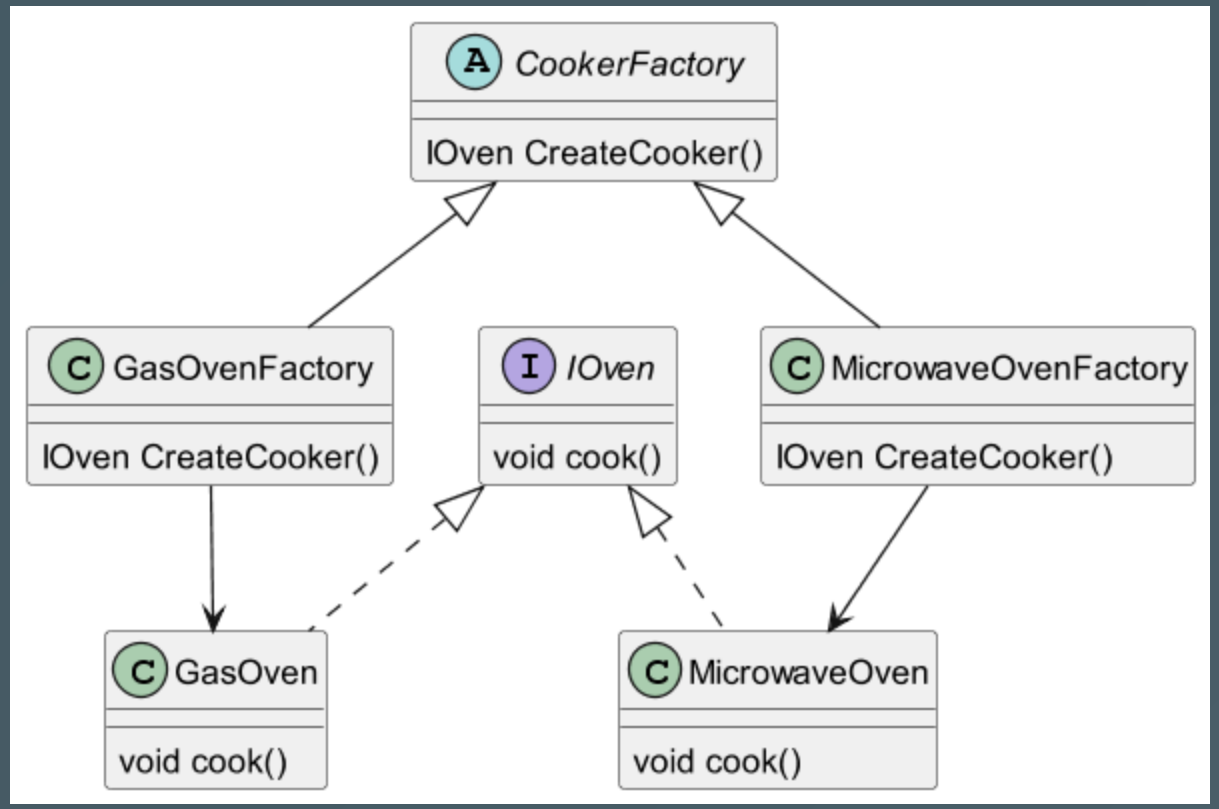
/!\ Multithreading /!\

Création: Factory

Problème: Comment créer des objets différents qui exposent les même comportements?

(ex: Un four et un micro-onde peuvent cuire)

Création: Factory

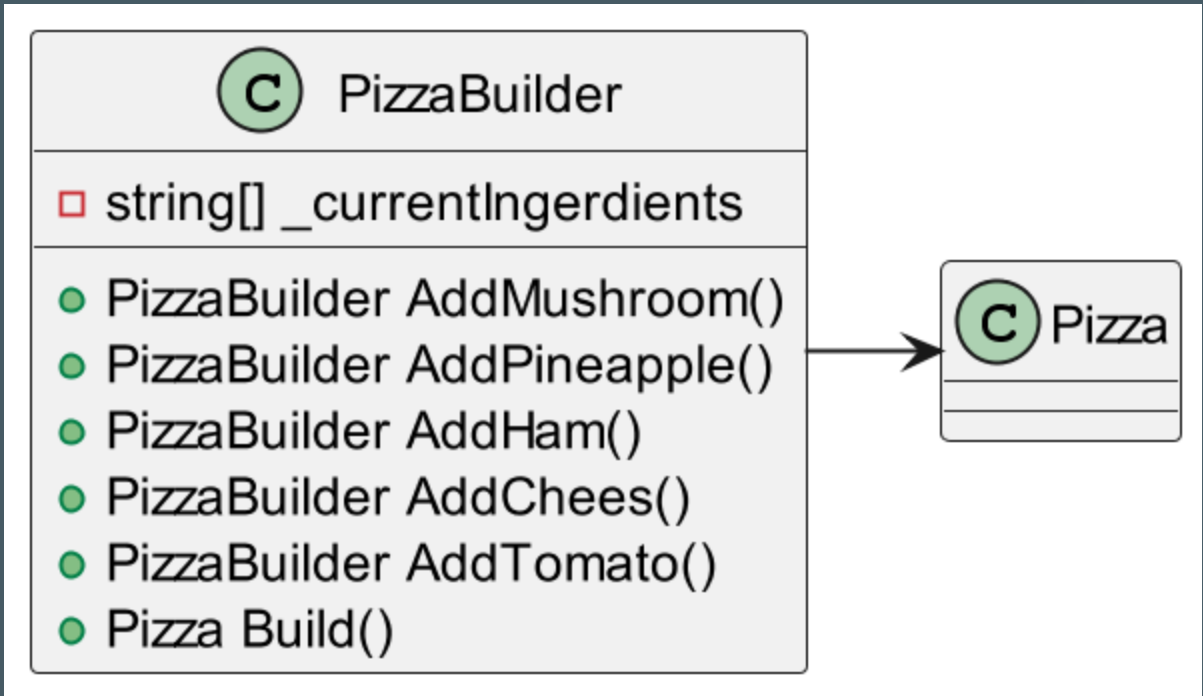


Création: Builder

Problème: Comment créer les même objets qui ont chacun leurs spécificités sans utiliser de conditions ?

(ex: Une pizza reine et une pizza hawaienne)

Création: Builder

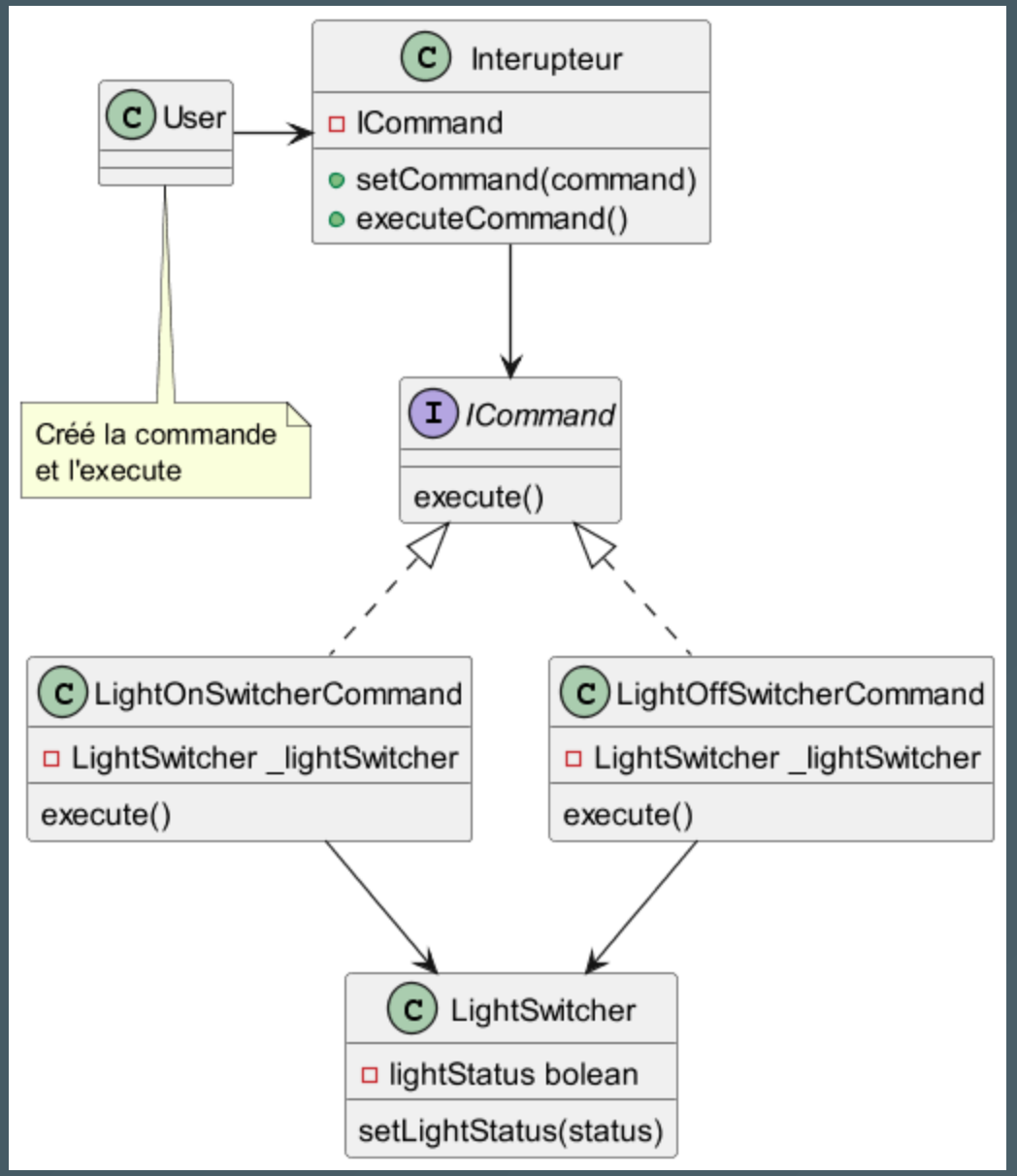


Comportementaux: Commande

Problème: Comment séparer le code initiateur de l'action et l'action elle même ?

(ex: Allumer la lumière via un interrupteur)

Comportementaux: Commande

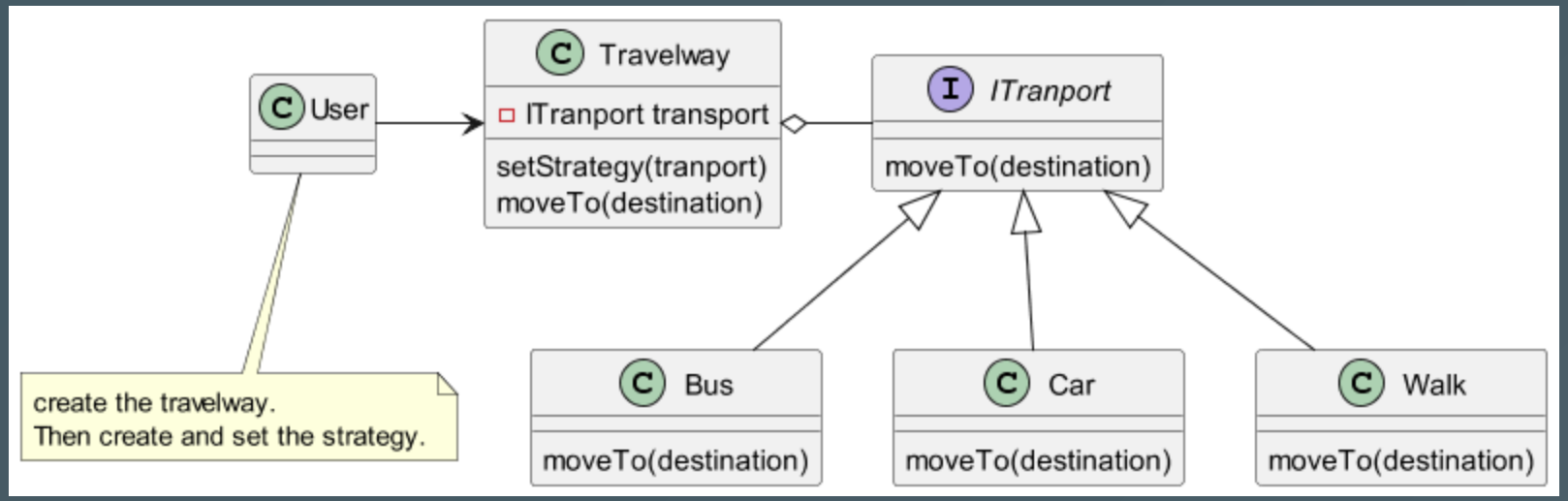


Comportementaux: Stratégie

Problème: Comment utiliser plusieurs stratégies pour résoudre un même de problème de façon différente.

(ex: Aller à la salle de sport)

Comportementaux: Stratégie

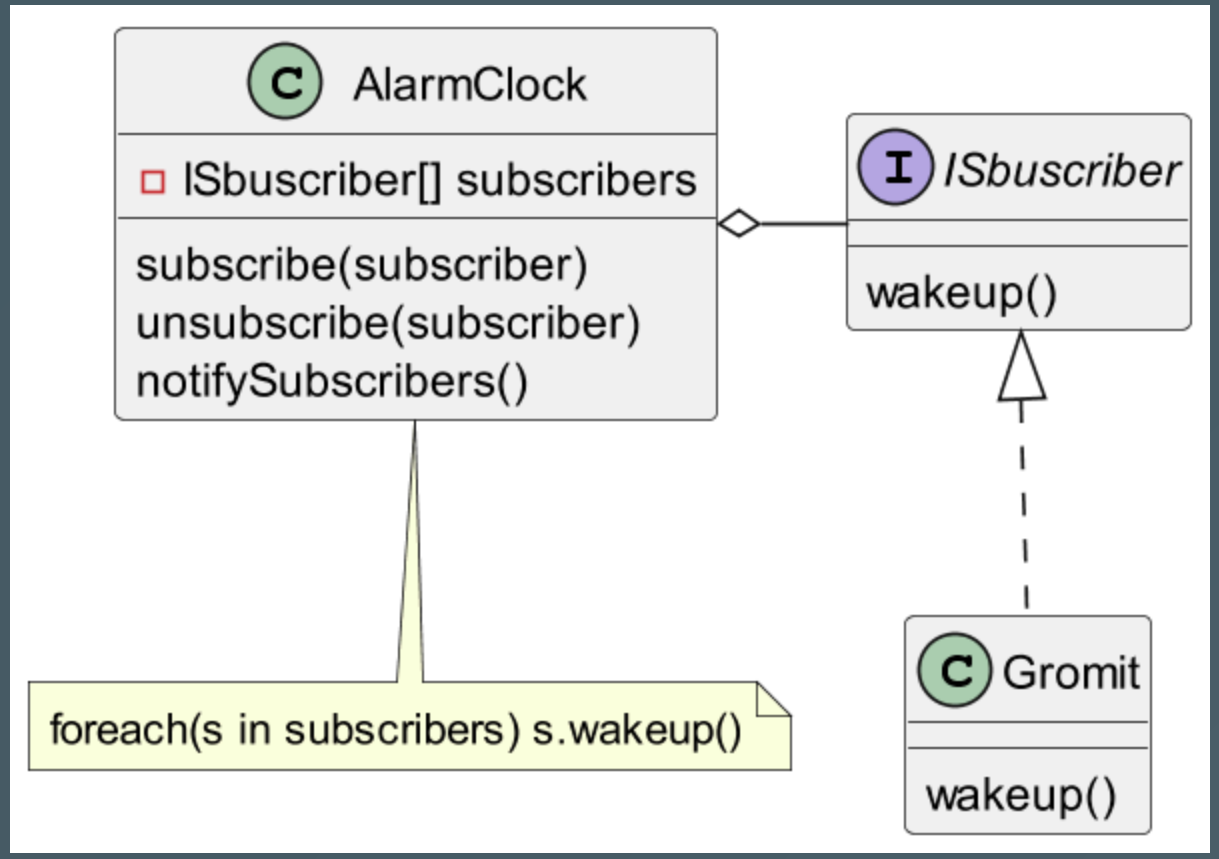


Comportementaux: Observateur

Problème: Comment un objet peut il être informé qu'un autre objet vient d'être modifié ?

(ex: Le reveille sonne et Gromit veut en être informé)

Comportementaux: Observateur

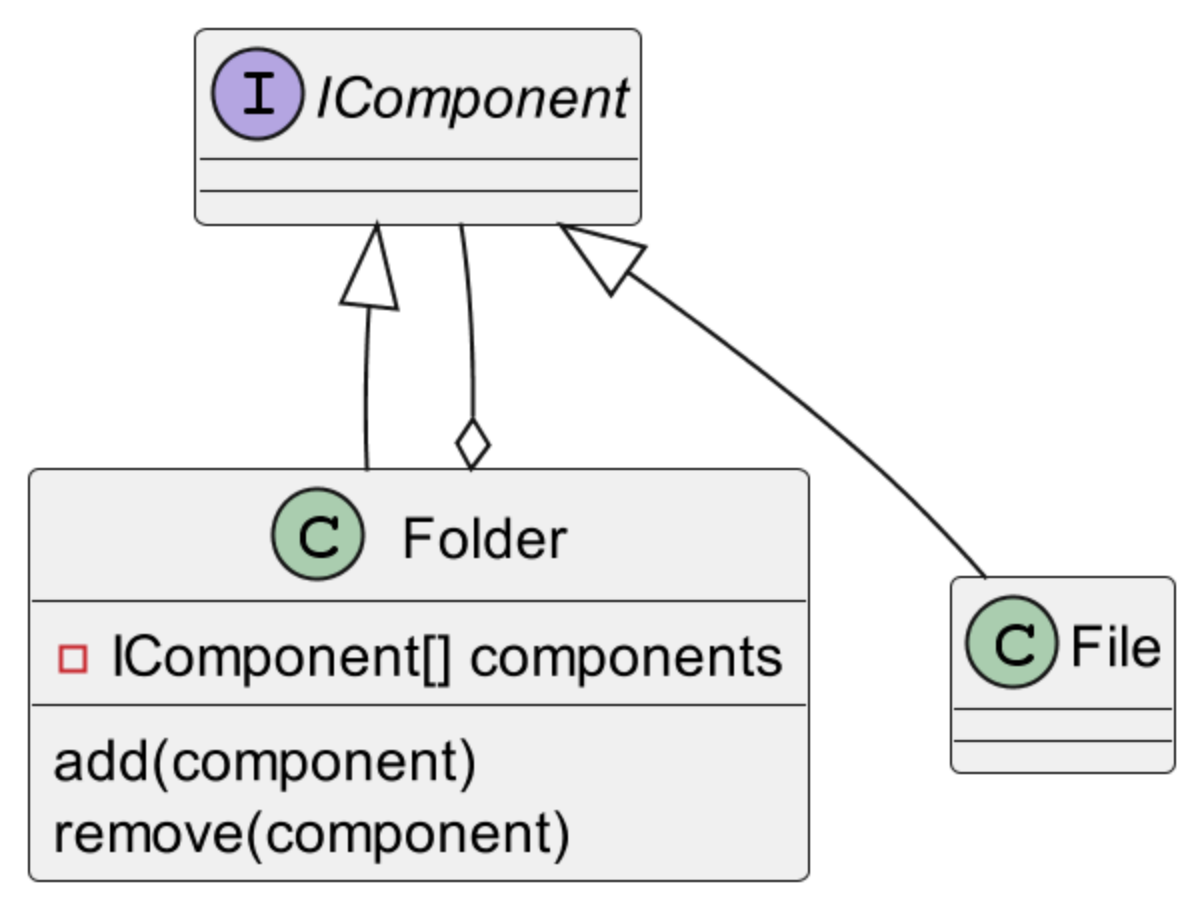


Structure: Composite

Problème: Comment traiter des conteneurs et des contenus ?

(ex: Un dossier de dossier qui contient des fichiers)

Structure: Composite

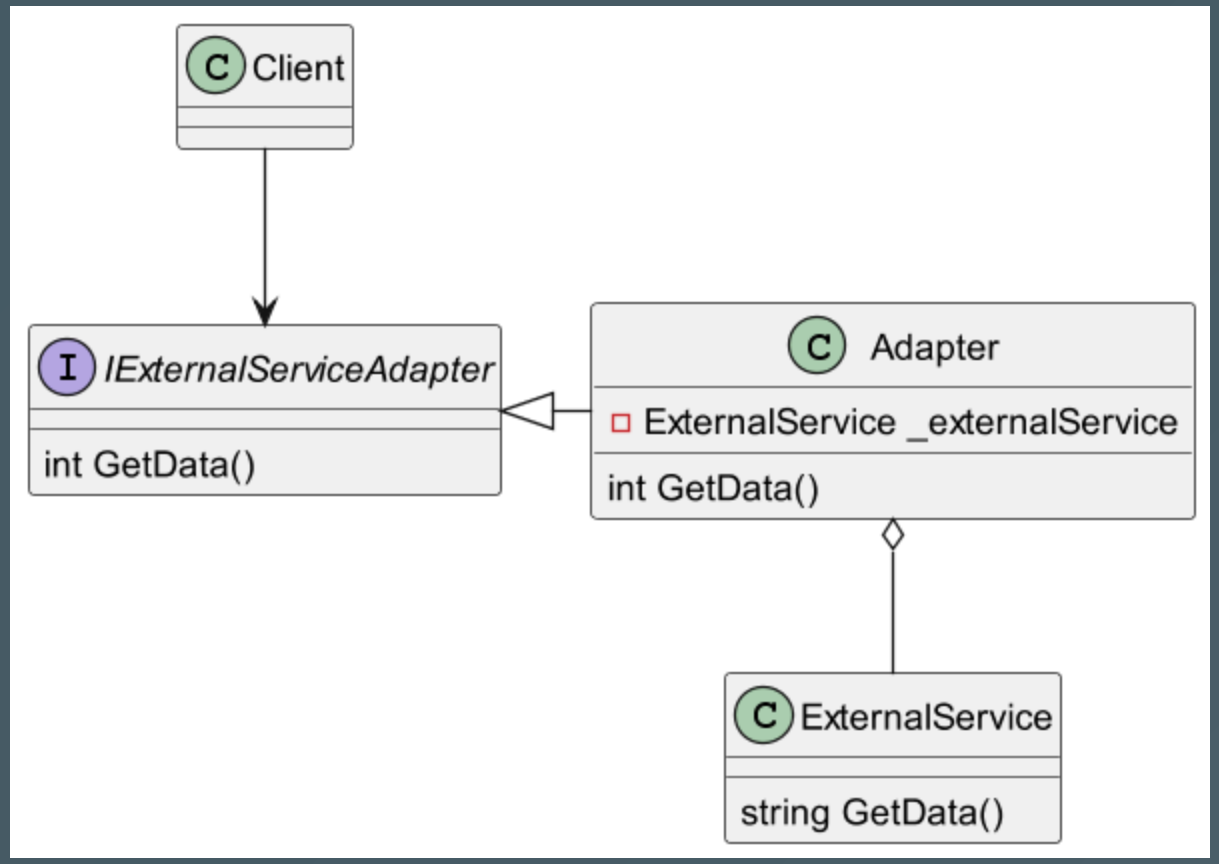


Structure: Adaptateur

Problème: Comment gérer un composant fournit par un système externe qui n'est pas compatible avec notre système ?

(ex: Le composant nous expose une donnée sous forme de chaîne de caractères là ou nous souhaiterions avoir un entier)

Structure: Adaptateur

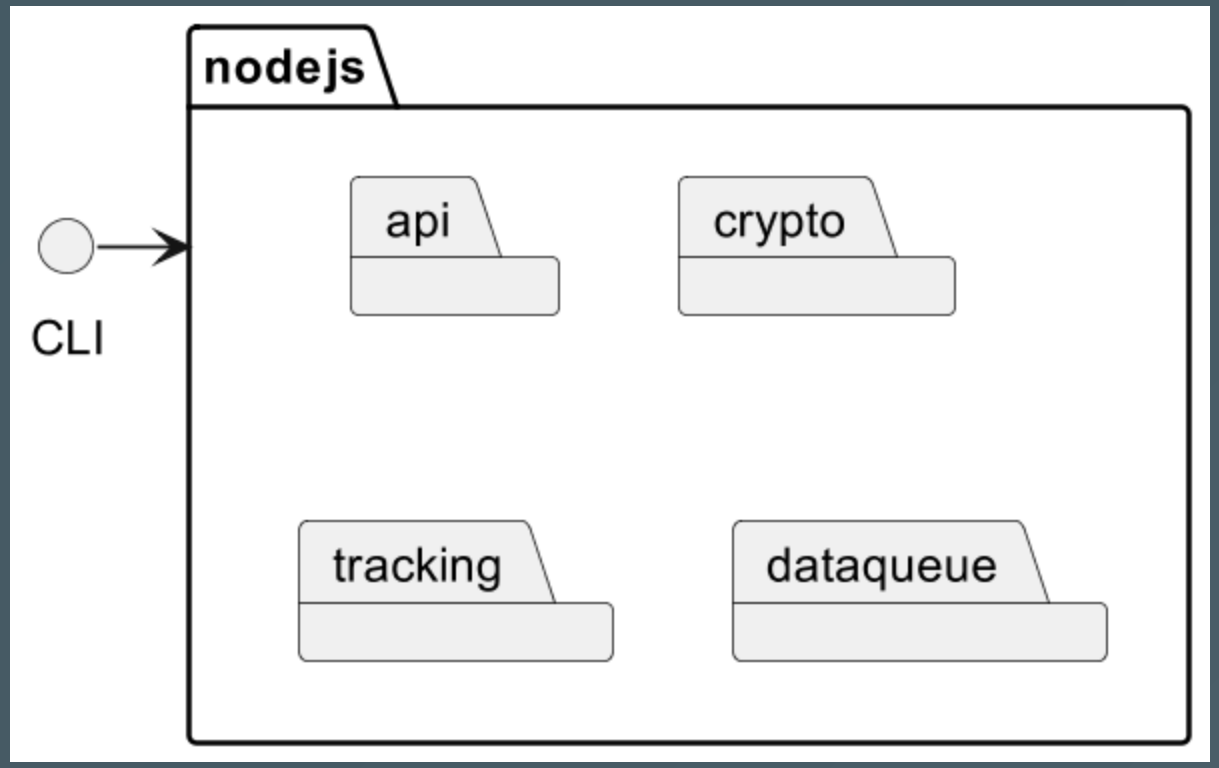


Structure: Façade

Problème: Comment diminuer la complexité d'un ensemble de service pour l'utilisateur ?

(ex: nodeJS)

Structure: Façade



Sources

- Refactoring.guru
- Design Patterns Tête la première
- Design Patterns Elements of Reusable Object-Oriented Software