1. **DESIGN PATERN PROXY**

Un proxy est une classe se substituant à une autre classe. Par convention et simplicité, **le proxy implémente la même interface que la classe à laquelle il se substitue**. L'utilisation de ce proxy ajoute une indirection à l'utilisation de la classe à substituer. **Le proxy sert à gérer l'accès à un objet**, il agit comme un intermédiaire entre la classe utilisatrice et l'objet.

Un proxy est un **cas particulier du patron de comportement**[**État**](https://fr.wikibooks.org/wiki/Patrons_de_conception/%C3%89tat). Un proxy **implémente une et une seule interface**, donc se substitue à une seule classe. Un état peut implémenter un nombre quelconque d'interfaces.

Un proxy est utilisé principalement pour **contrôler l'accès aux méthodes de la classe substituée**. Un état est utilisé pour changer dynamiquement d'interface.

* **Illustration 1**

Contexte

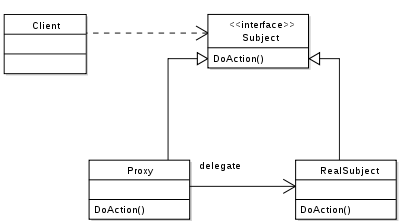
* Le client a besoin d’accéder aux services d’un autre composant (ex. objet, base de données, page html ou image)
* L’accès direct est possible du point de vue technique mais sans être la meilleure solution

Problème

* L’accès direct à un composant n’est souvent pas pratique – des procédures additionnelles de contrôle sont nécessaires (ex. authentification, localisation)
* Le code client doit rester simple et l’accès aux composants transparent et efficace

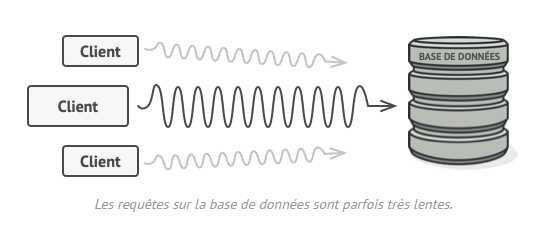
Solution

* Le client communique avec le représentant (proxy) plutôt qu’avec le composant
* Le proxy offre l’interface du composant mais exécute des procédures additionnelles avant (pre) et après (post) l’invocation du composant



* **Illustration 2 :**

**Problème :** Ici les demandes du client passent directement à la base de données. Ce qui ralenti le processus.



**Solution :** Ce Design pattern vous propose de créer une classe procuration qui a la même interface que l’objet du service original. Vous passez ensuite l’objet procuration à tous les clients de l’objet original. Lors de la réception d’une demande d’un client, la procuration crée l’objet du service original et lui délègue la tâche.

