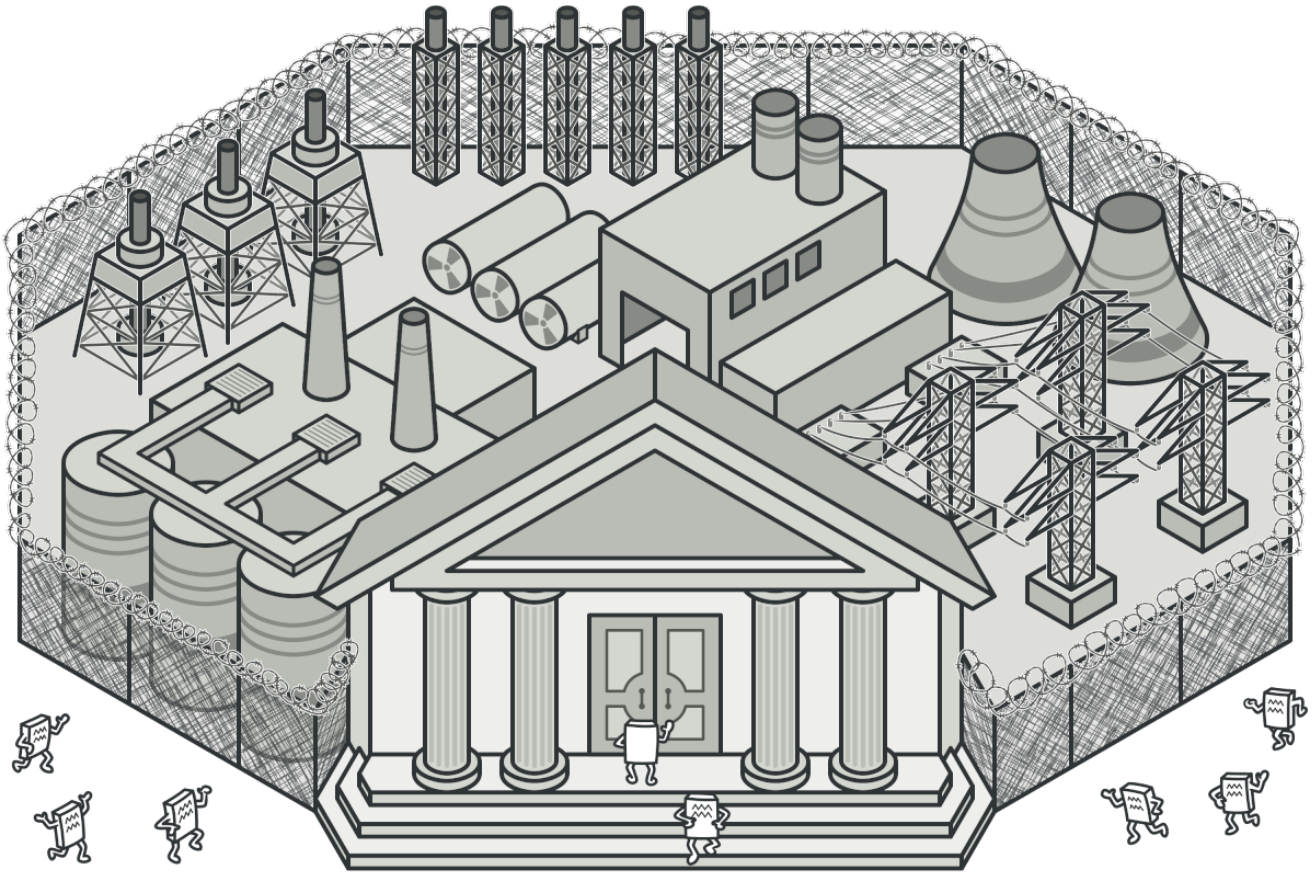


# Façade

Le patron de la façade permet d'avoir un accès simplifié à un ensemble complexe de classes.

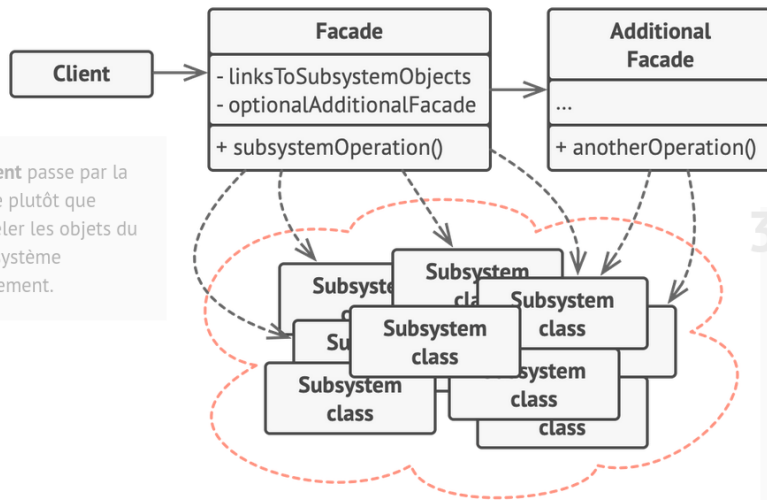


Voici à quoi ressemble le patron de la façade :

1 La **Façade** procure un accès pratique aux différentes parties des fonctionnalités du sous-système. Elle sait où diriger les requêtes du client et comment utiliser les différentes parties mobiles.

2 Une classe **Façade Supplémentaire** peut être créée pour éviter de polluer une autre façade avec des fonctionnalités qui pourraient la rendre trop complexe. Les façades supplémentaires peuvent être utilisées à la fois par le client et par les autres façades.

4 Le **Client** passe par la façade plutôt que d'appeler les objets du sous-système directement.



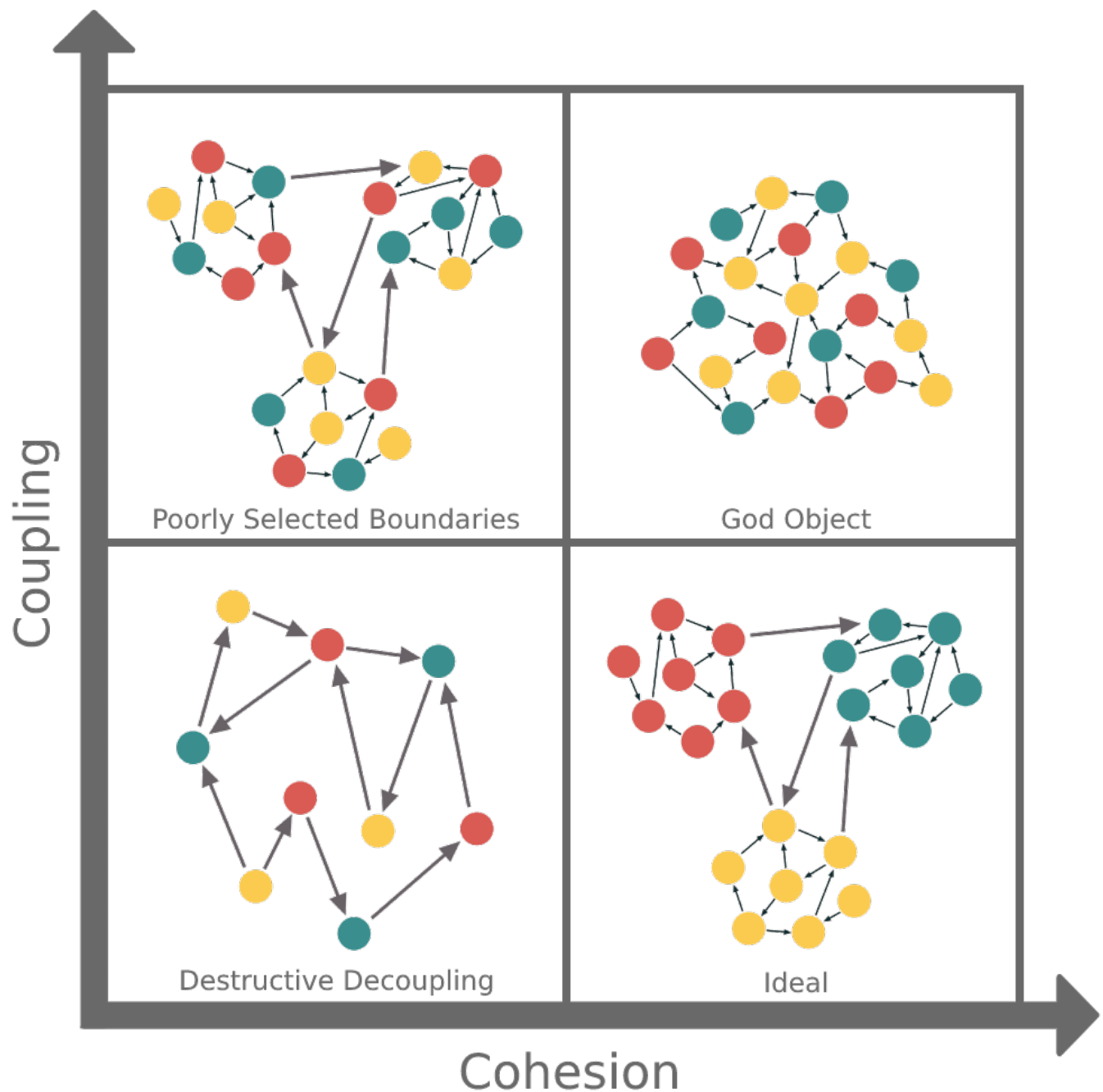
3 Le **Sous-système Complexe** est composé de dizaines d'objets variés. Pour leur donner une réelle utilité, vous devez plonger au cœur des détails de l'implémentation du sous-système, comme initialiser les objets dans le bon ordre et leur fournir les données dans le bon format.

Les classes du sous-système ne sont pas conscientes de l'existence de la façade. Elles opèrent et interagissent directement à l'intérieur de leur propre système.

Le fonctionnement de la facade est que l'on crée une classe facade qui collabore avec les éléments du système pour fournir une interface simple au client. Cependant si la facade devient elle même trop imposante, on peut utiliser faire communiquer une façade avec une autre façade.

Les façades permettent ainsi d'éviter les God objects et les God functions (les objets ou fonctions qui font trop de choses et connaissent trop de classes).

# Cohésion et couplage



Le patron de la façade a également l'avantage d'augmenter la cohésion, c'est à dire d'augmenter le degré d'interconnexion des membres du système du quel la façade fait partie. Une cohésion forte indique que le type représente un concept clair et réutilisable.

Cependant si le couplage est trop élevé, on arrive dans un God Object (précisément ce que l'on souhaite éviter), c'est pour cela qu'il vaut mieux créer plusieurs façades si une façade devient un peu trop omnisciente.

Revision #1

Created 3 October 2023 21:02:16 by SnowCode

Updated 3 October 2023 21:07:06 by SnowCode