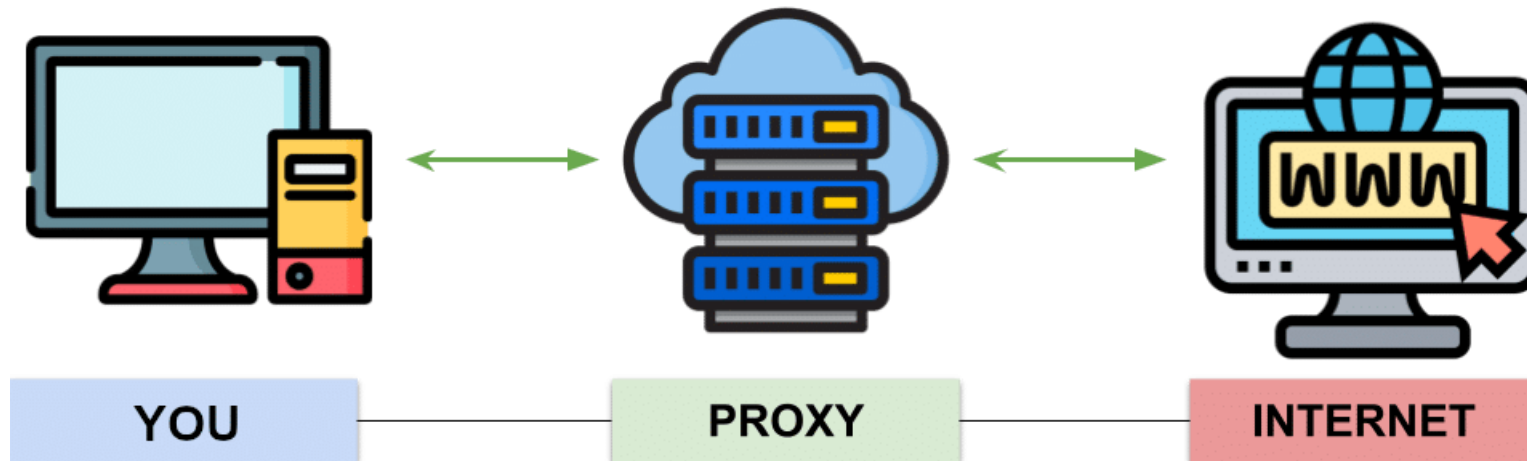


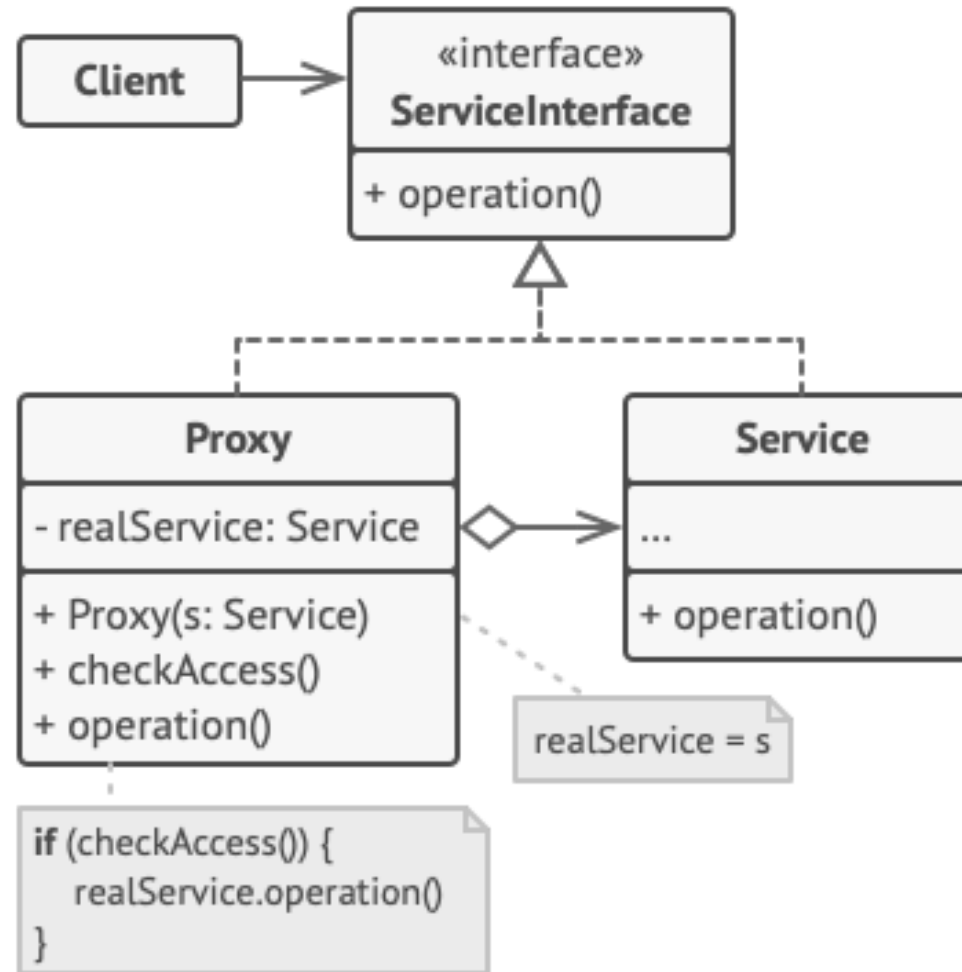
Proxy

Proxy is a structural design pattern that lets us provide a substitute or placeholder for another object. A proxy controls access to the original object, allowing you to perform something either before or after the request gets through to the original object.

“Provide a surrogate or placeholder for another object to control access to it.”
- GOF

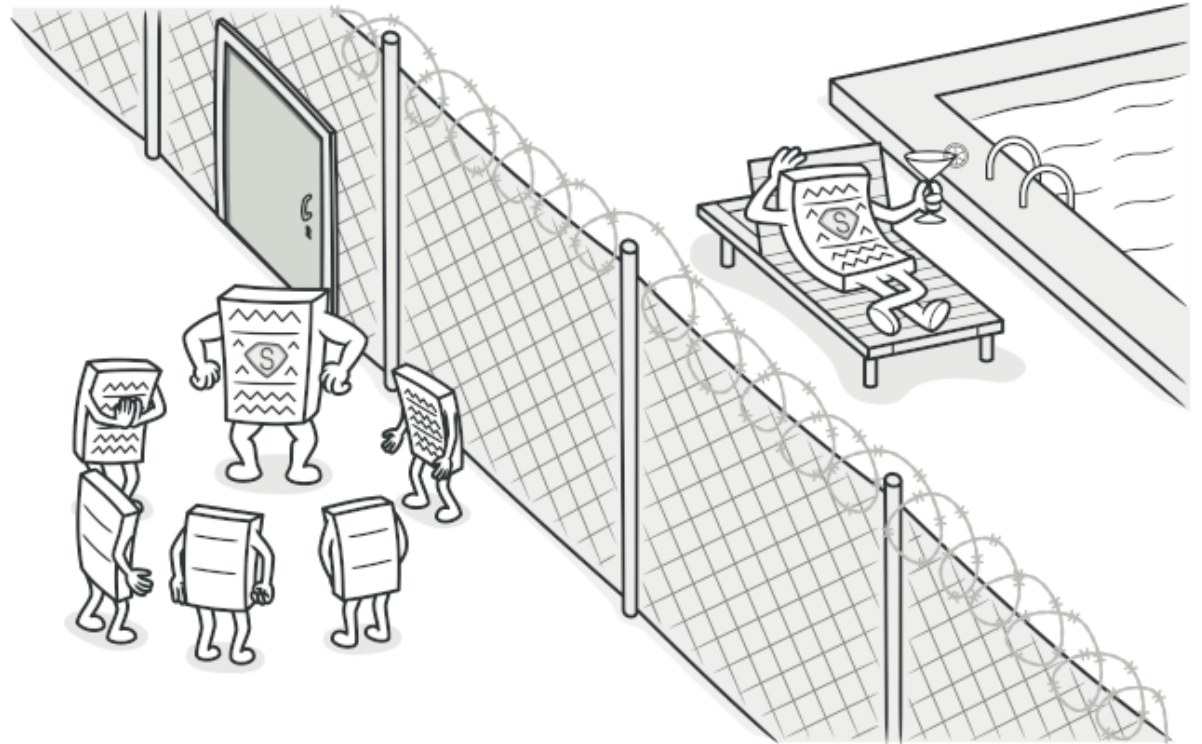


Proxy



Proxy Use Cases

- Lazy Initialization
- Cache
- Logging
- Access Control
- Additional logic



Proxy Use Cases

- **AOP**
- **Transaction Management**
- **Spring Security**
- **Spring Data JPA**
- **Caching**
- **Beans**



Spring Framework

Command

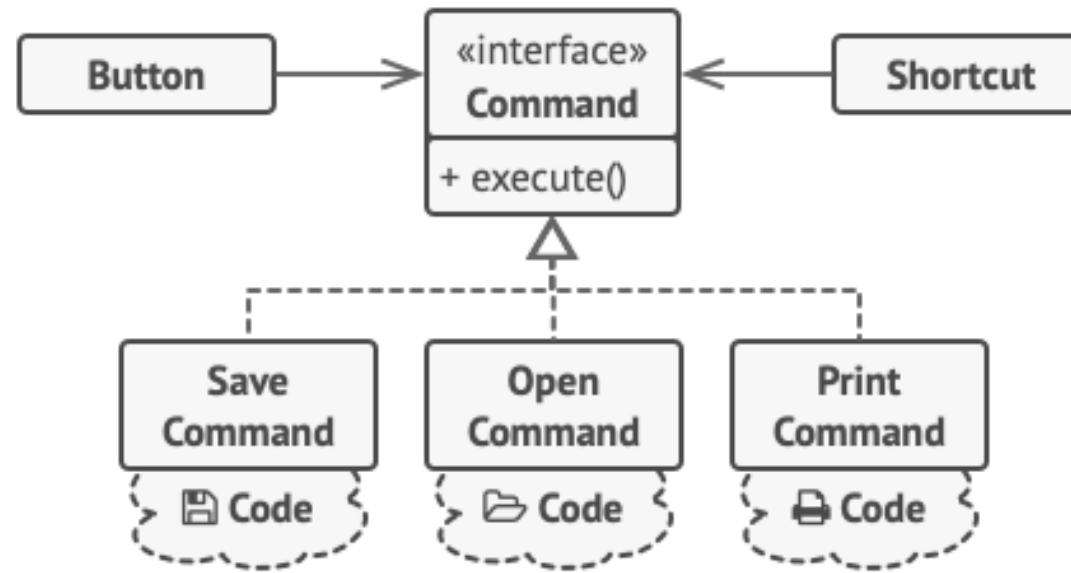
Command is a behavioral design pattern that turns a request into a stand-alone object that contains all information about the request.

This transformation lets us pass requests as a method arguments, delay or queue a request's execution, and support undoable operations.

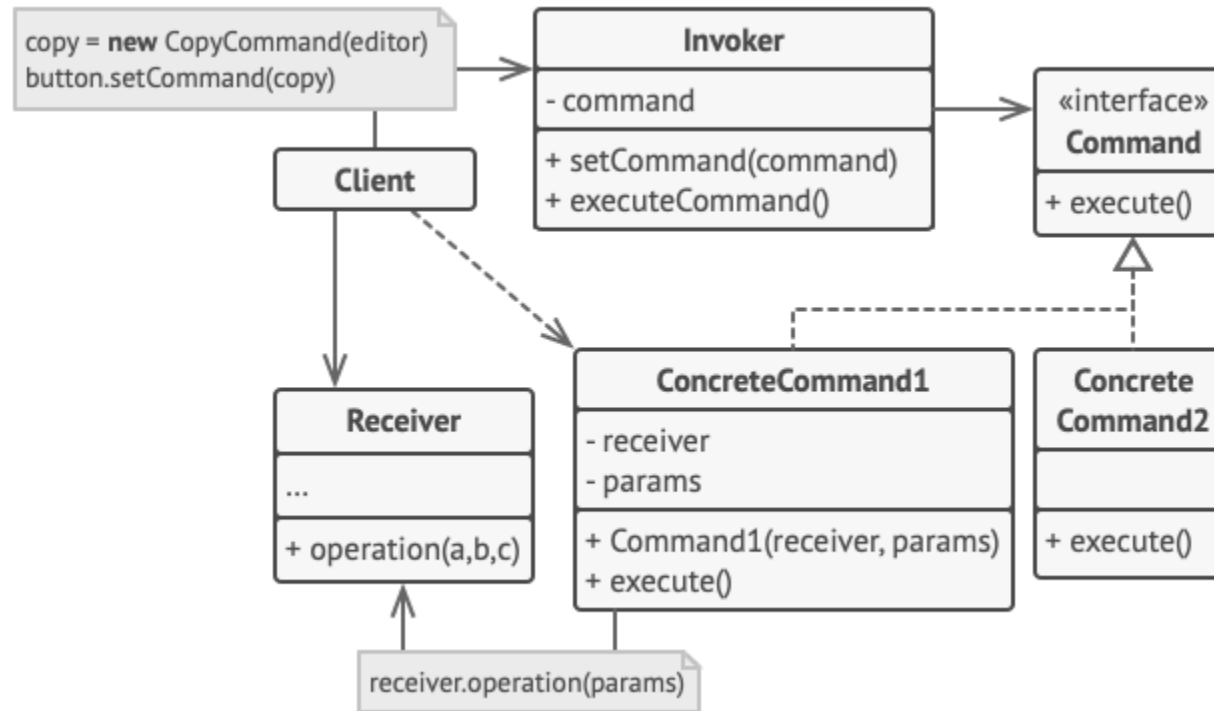


*commando

Command

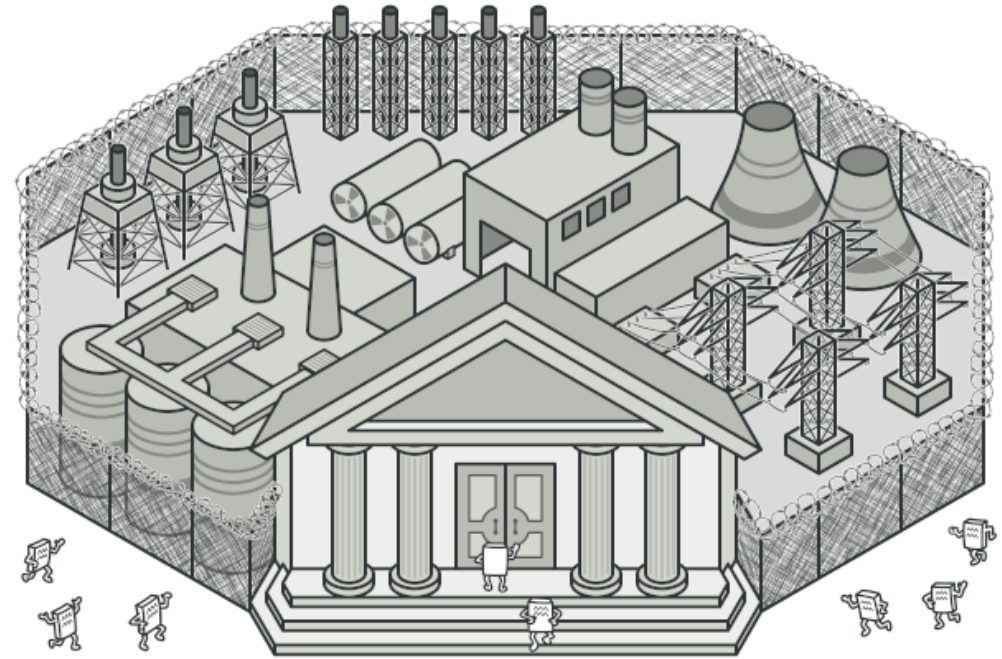


Command

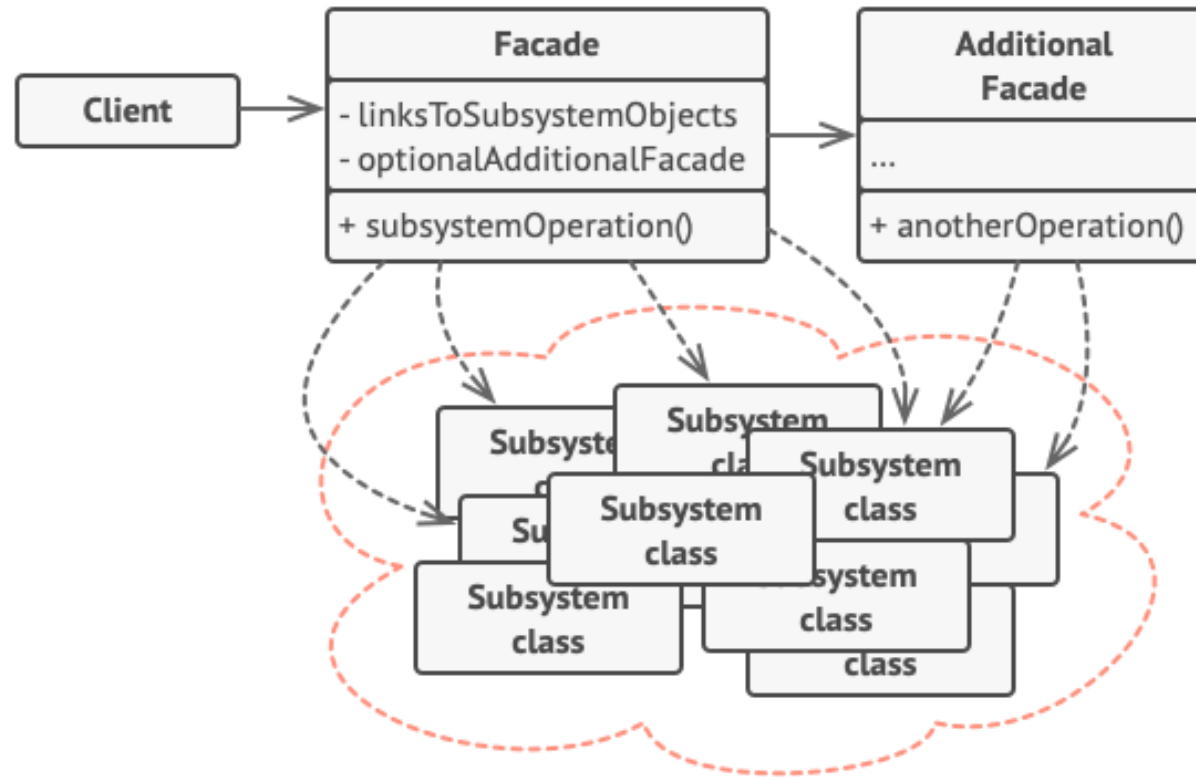


Facade

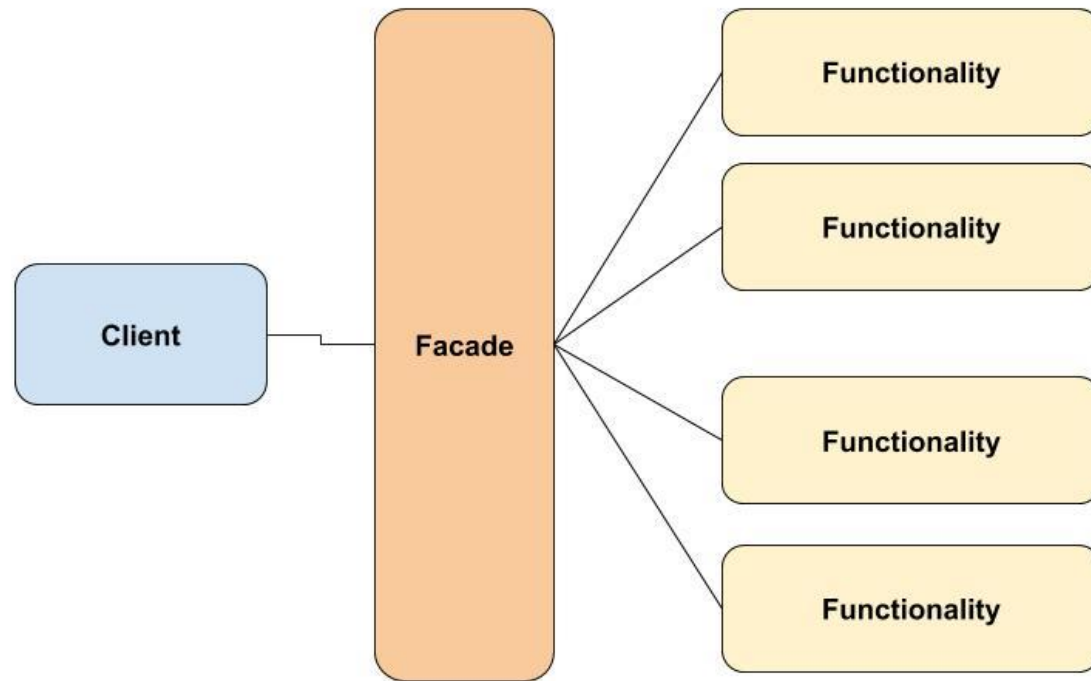
Facade Pattern provides a unified interface to a set of interfaces in a subsystem. Facade defines a higherlevel interface that makes the subsystem easier to use.



Facade



Facade



Facade

