

# The Façade

---

# Façade

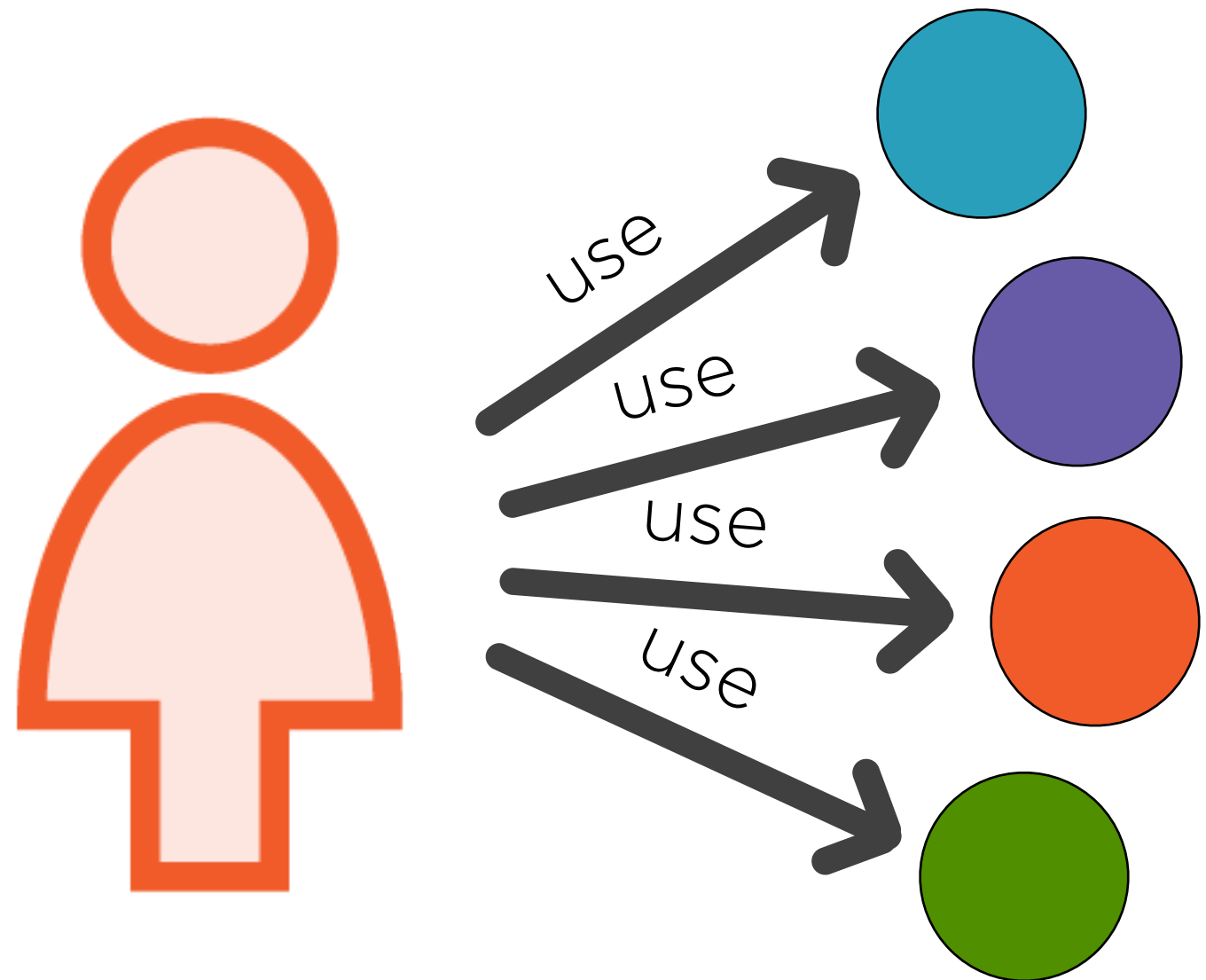
Simplifies the usage of complex subsystems.

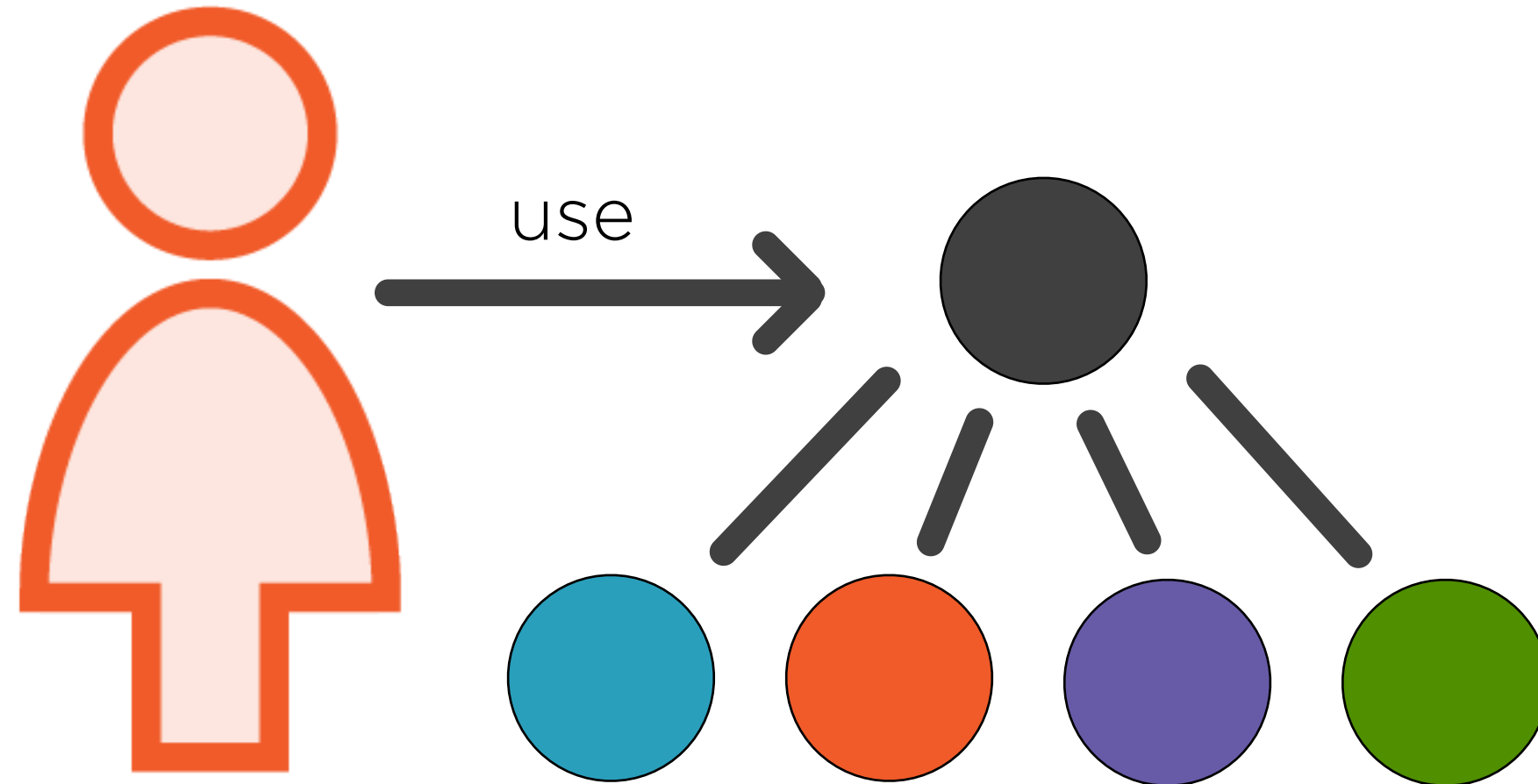
# Façade

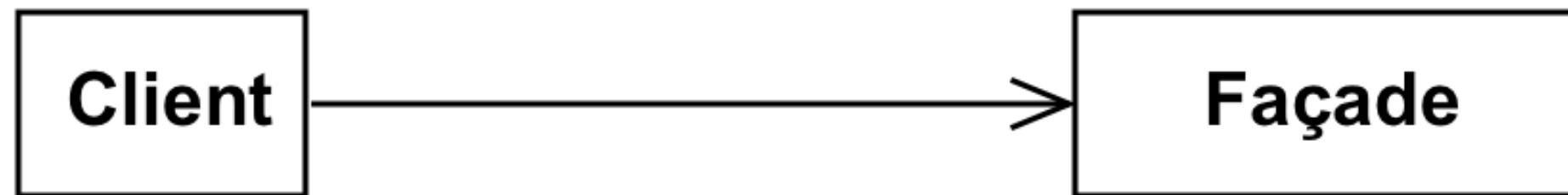
**Hides the complexity of underlying components**

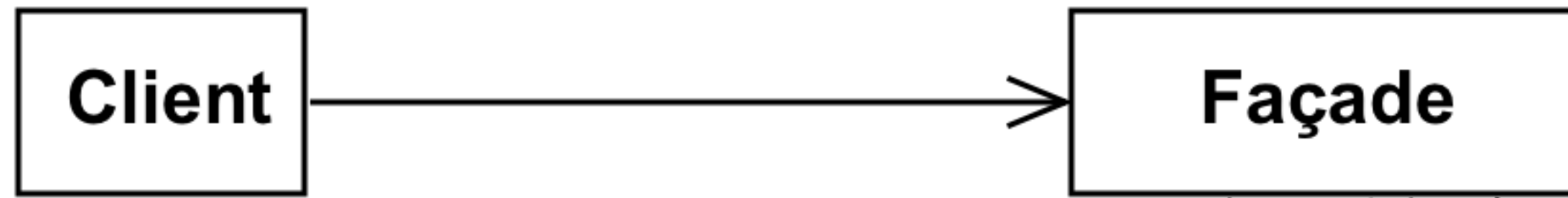
**Consolidates the functionality into a type**

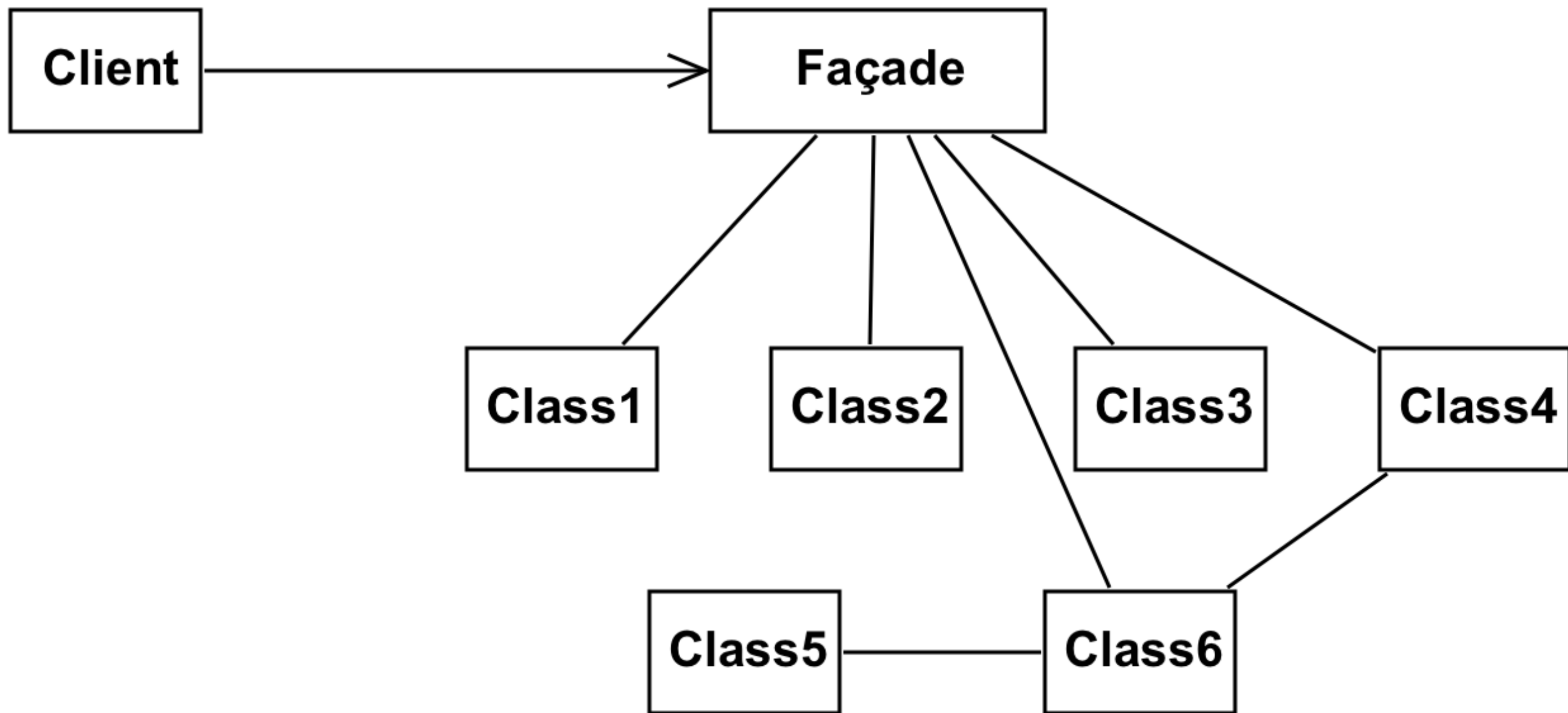
**Clients use the Façade instead of the complex subsystems**













When to use  
the Façade?

**Clients don't need all the features  
of a library/set of types**

**Isolate clients from changes**

**Prevent frequent refactoring**

## Pitfalls

**Avoid adding too many responsibilities**

**Don't expose the underlying types**

# Summary

## **The Façade Pattern:**

- Simplifies the usage of subsystems
- Exposes a unified interface
- Isolates the client from the underlying types

## **Pitfalls**

- Expose details of the underlying types