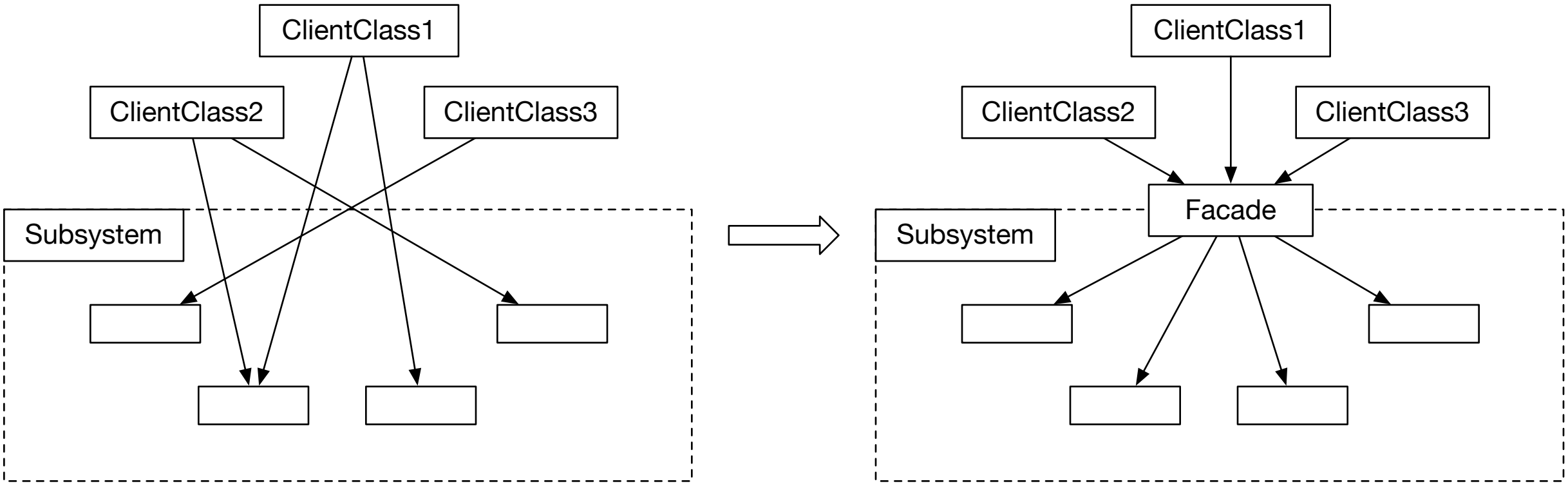
Facade

Provide a unified interface to a set of interfaces in a subsystem. Facade defines a higher level interface that makes subsystems easier to use.

Instead of having the diagram on the left, have one like on the right. Subsystem components don't know about the Facade; the Facade forwards requests to corresponding subsystem components.



Most design patterns result in more and smaller classes, thus creating interfaces easy to customize, but also hard for clients to use. For those clients who don't care, providing a facade may ease their life.
Facade also promotes weak coupling, independence (and subsystem layering). Instead of having clients rely on different interfaces inside your subsystem, they can rely on one, cutting a cleaner line between subsystem and its client.

If there are multiple implementations of subsystems, consider making Facade an interface and derive actual implementation-related facades from the interface.

A class encapsulates state and operations, a subsystem encapsulates classes. It can achieve so using Facade.

Often times only one Facade object needs to exist, thus can be implemented as Singletons.