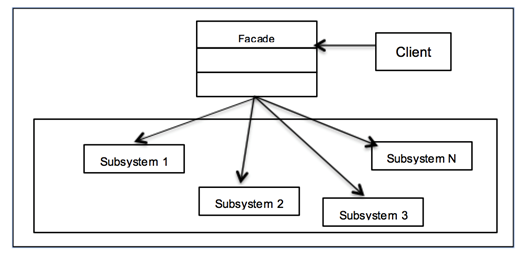
# Intent

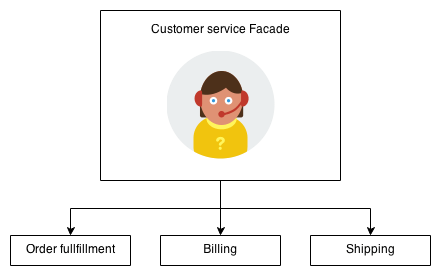
Facade is a structural design pattern that provides a simplified interface to a library, a framework, or any other complex set of classes.

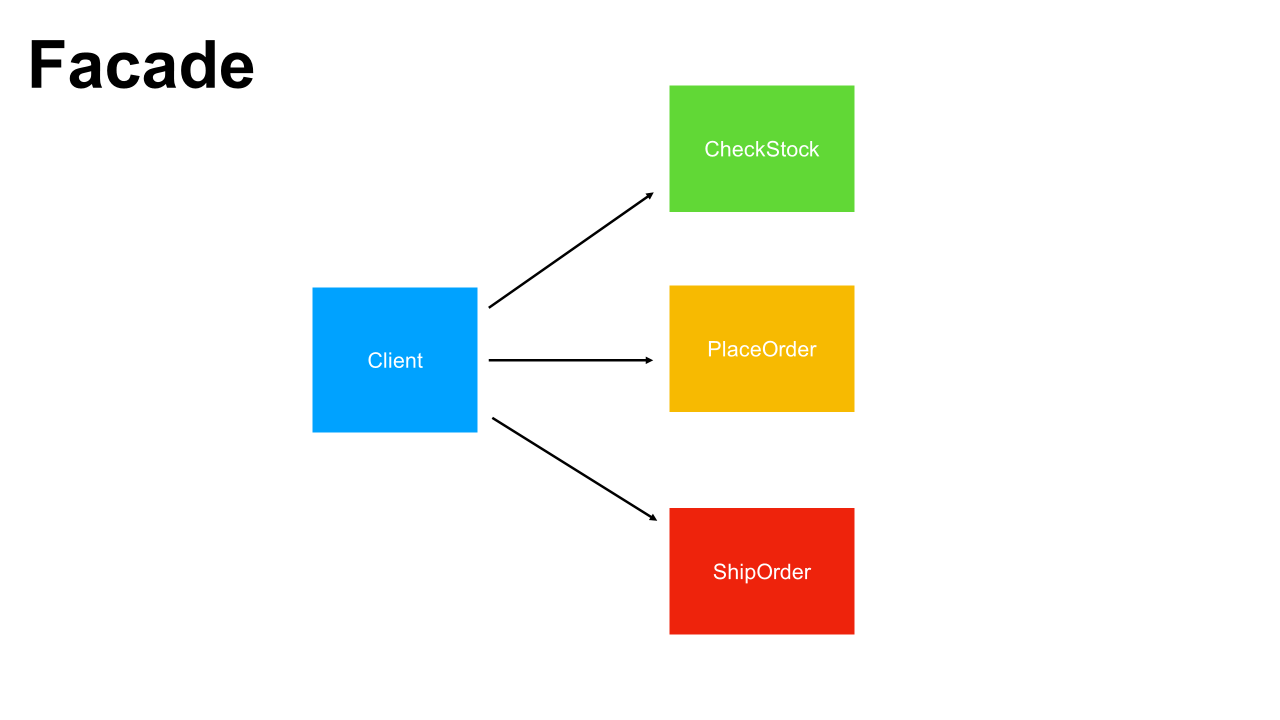
The Facade Pattern hides the complexities of the system by providing an interface to the client from where the client can access the system on a unified interface. Facade defines a higher-level interface that makes the subsystem easier to use. For instance, making one class method perform a complex process by calling several other classes.

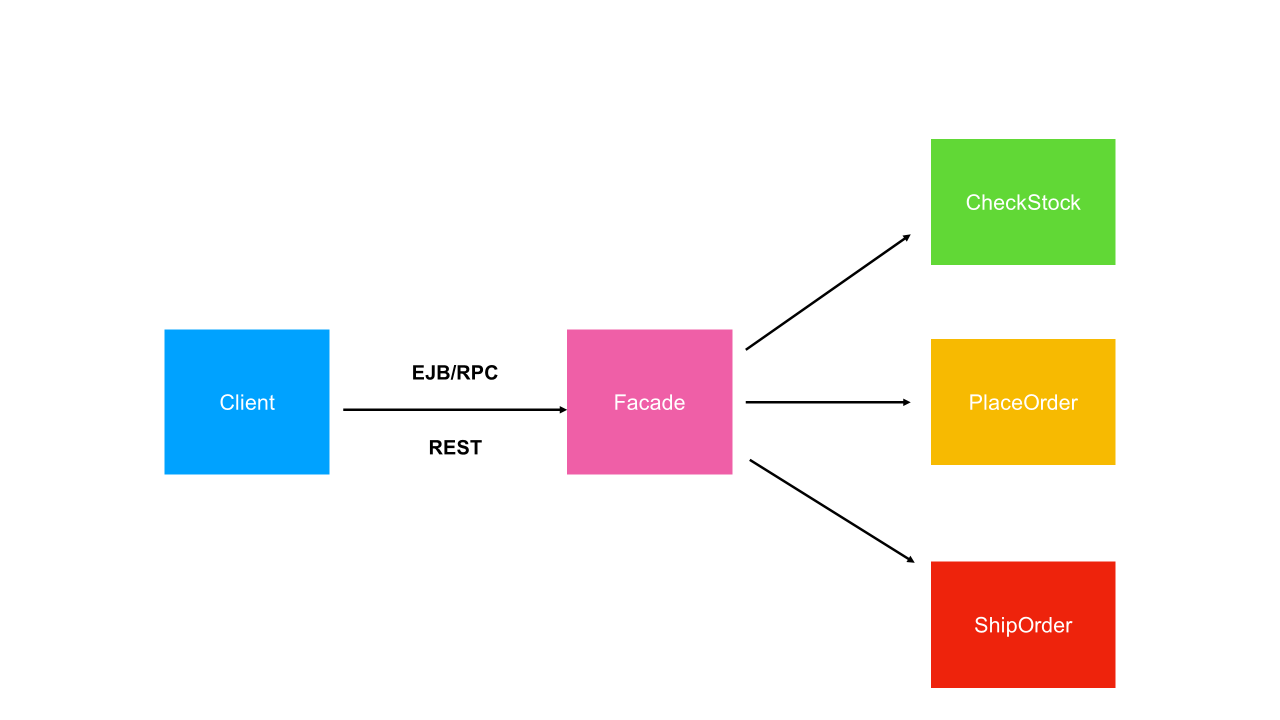
**Problem:**A segment of the client community needs a simplified interface to the overall functionality of a complex subsystem.

**Solution:**Use a facade class and the client only needs to access this class to perform the operations of the subsystems.









* OrderFacade (Facade): Contains members of the type of each subsystem.
* CheckStocks (Subsystem)
* PlaceOredr (Subsystem)
* ShipOrder (Subsystem)
* Client (main function): use OrderFacade to call functions of each subsystem