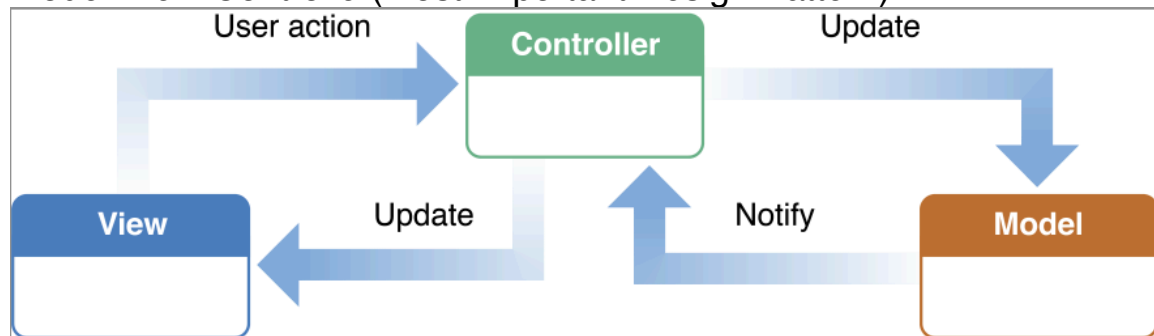


Design Pattern

It's a template for a design that solves a general, recurring problem in a particular context.

Model-View-Controller(Most Important Design Pattern)



Model Object

it encapsulates the data of an app and defines the logic and computation that manipulate and process that data.

View Object

it's an object that user can see, it knows how to draw itself and might respond to user actions.

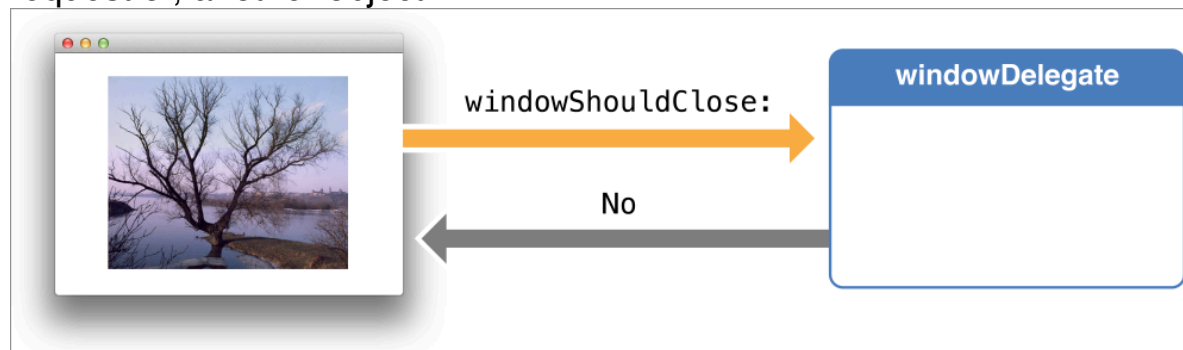
Controller Object

it acts as an intermediary between one or more of an app's view objects. and one or more of its model objects.

Solving Problems with Design Patterns

Delegation(acting on behalf of another object)

In delegation, an object called the delegate acts on behalf of, and at the request of, another object.



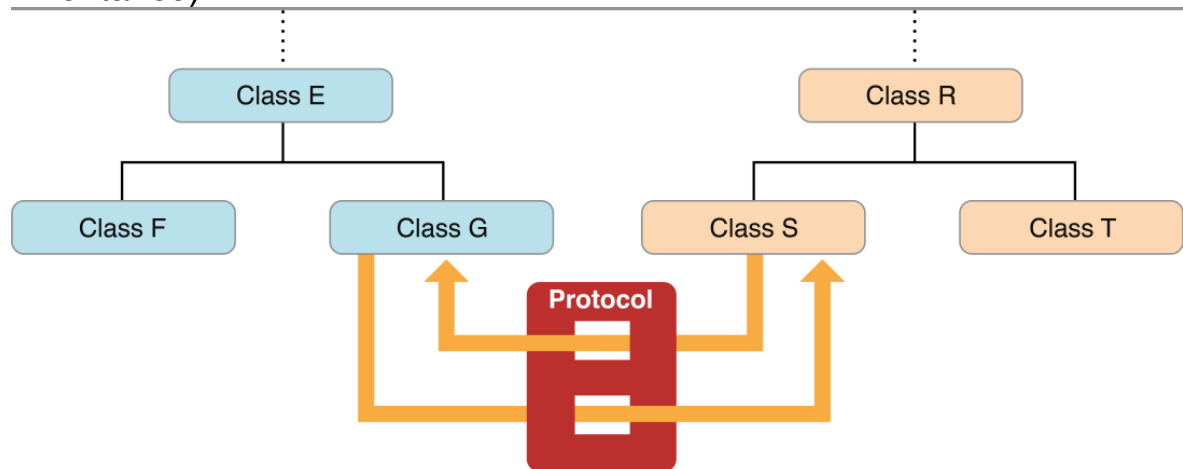
delegating object is a framework object.

delegated object is an instance of custom class.

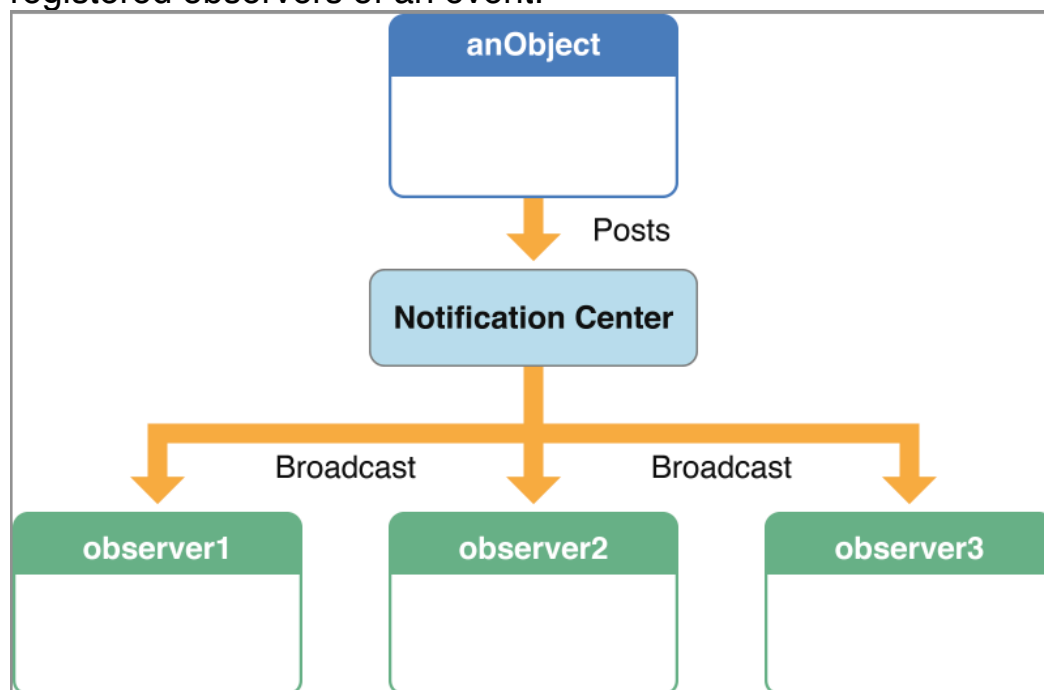
A common design in Cocoa Touch apps uses delegation as a means for allowing a child view controller to communicate some value (typically a

user-entered value) to its parent view controller.

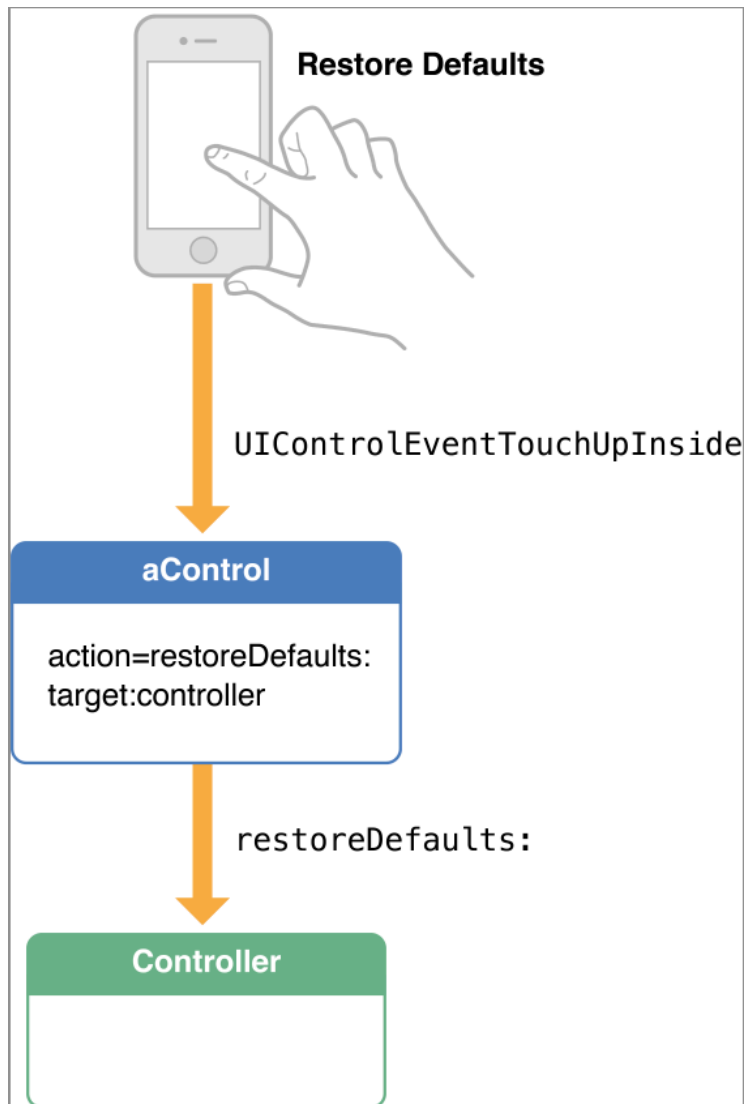
Protocol(enabling communication between objects not related by inheritance)



Notification Center(Notifying interested observers of an event)
a notification center is a subsystem of the Foundation framework that broadcasts a message.(a notification) to all objects in an app that are registered observers of an event.



Target-Action(encapsulating a message to be sent when an event occurs)

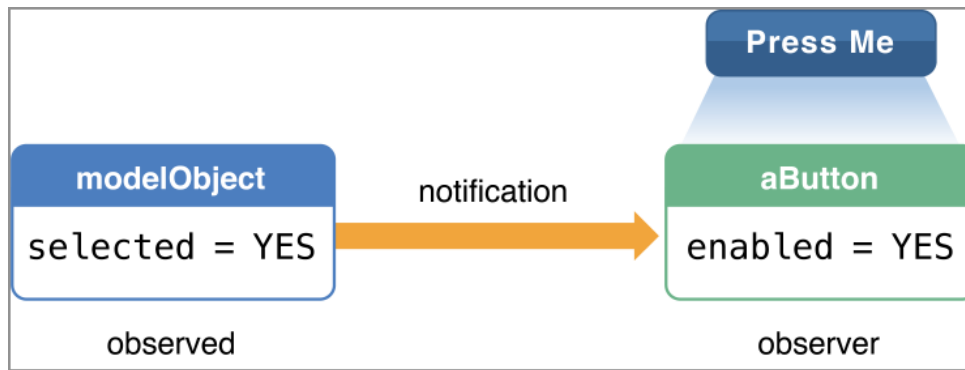


An object stores the elements that make up a message expression and, when a certain event occurs, puts these elements together and sends a message.

The elements are a selector identifying the message (the action) and the object to receive the message (the target).

The target's class implements the method corresponding to the action and the target, and when it receives the message at runtime, it responds to the event by executing the method.

Key-Value-Observing(Notifying an observer when a value changes)



Other Framework Designs Based on Design Patterns

1. View hierarchy
2. Responder chain
3. View controller
4. Receptionist
5. Category