Model View Controller

Advanced Topics in iOS & Swift
11/04/2016

Getting real…

- Creating a small *MoodTracker* app that displays a list of friends in a table view

- each friend has a **`mood`** that can be toggled with a tap on the button that displays the mood symbol (😁😑😡)

- **Goal:** implement a *proper information flow* according to the MVC pattern

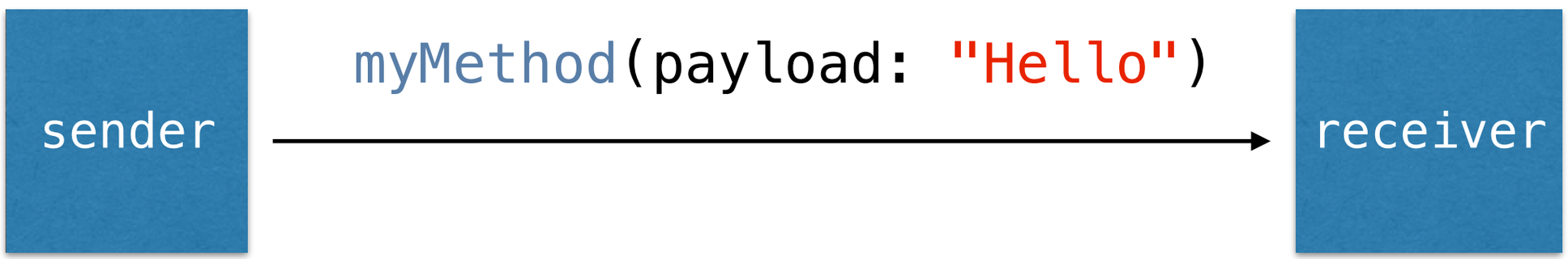
- Note: you'll learn about usage of enums and different communication patterns along the way

Direct Method Invocation

fancy term for **method call**

simplest form to **make objects communicate**

calling method on an object = **send message** to that object



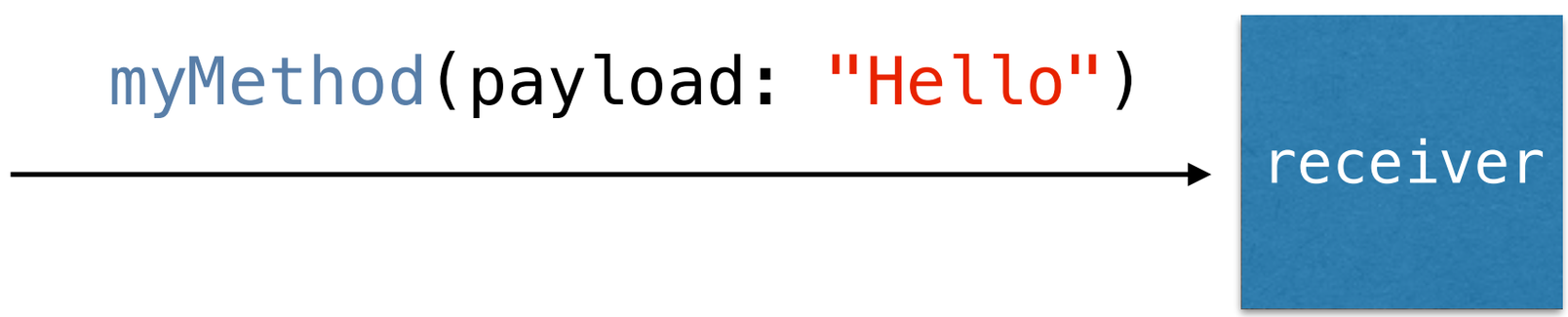
Direct Method Invocation

```
class Receiver {

  func myMethod(payload: String) {
    // ...
  }
}

let receiver = Receiver()
receiver.myMethod(payload: "Hello")
```

<u>NOTE:</u>
The receiver implements the method

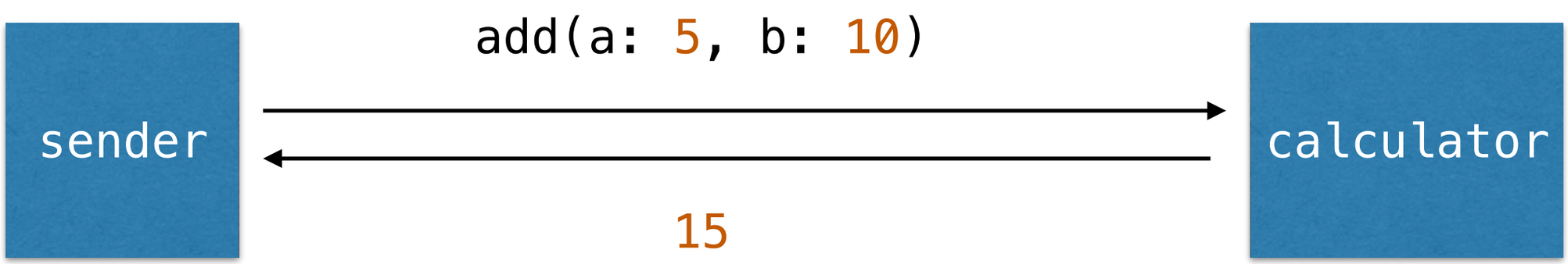


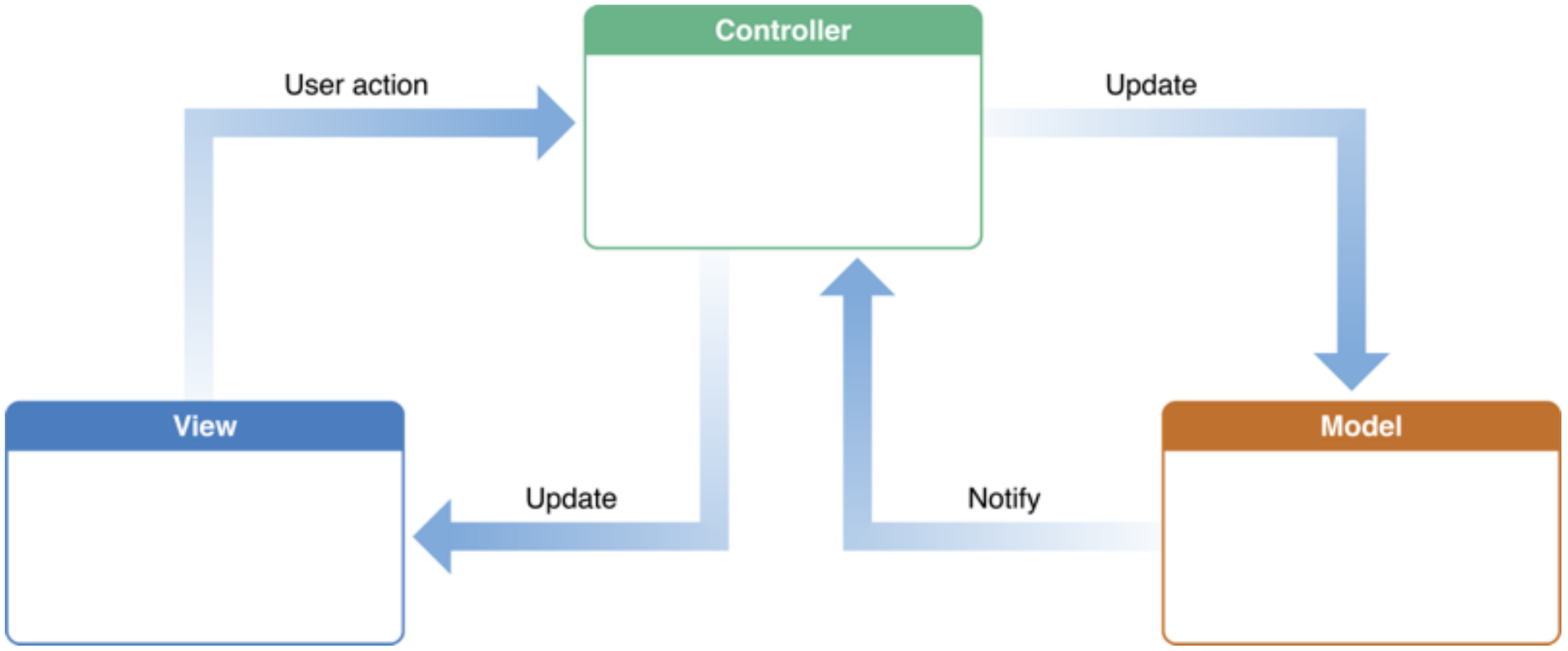
Direct Method Invocation

```
class Calculator {

    func add(a: Int, b: Int) -> Int {
        return a+b
    }

}
```




Controller
User action
Update
View
Update
Notify
Model

Controller

`FriendsTableViewController`

2 OK, I update the `friend`'s `mood`

3 the `friend`'s `mood` changed, `view` needs to look different now

my `friend` got tapped **1**

`Friend`

`FriendTableViewCell`

Model

View