More swift

MVC

Q&A

Link for enrolling to our course on iTunes U

FHM-MLC-WXK

(only from iOS)

Road Map

- MVC
- structures
- enums
- *demo*

Model View Controller

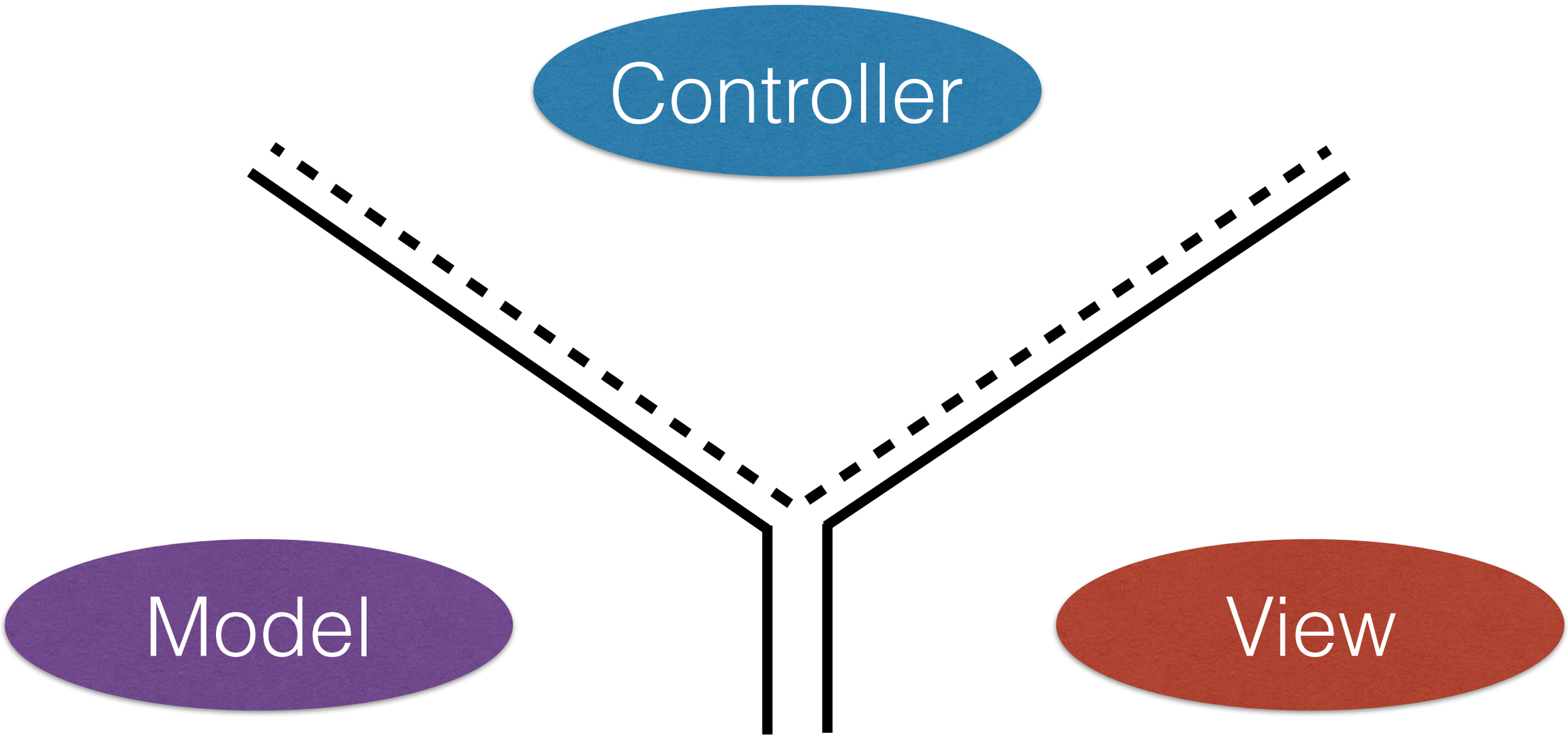
MVC

Controller

Model

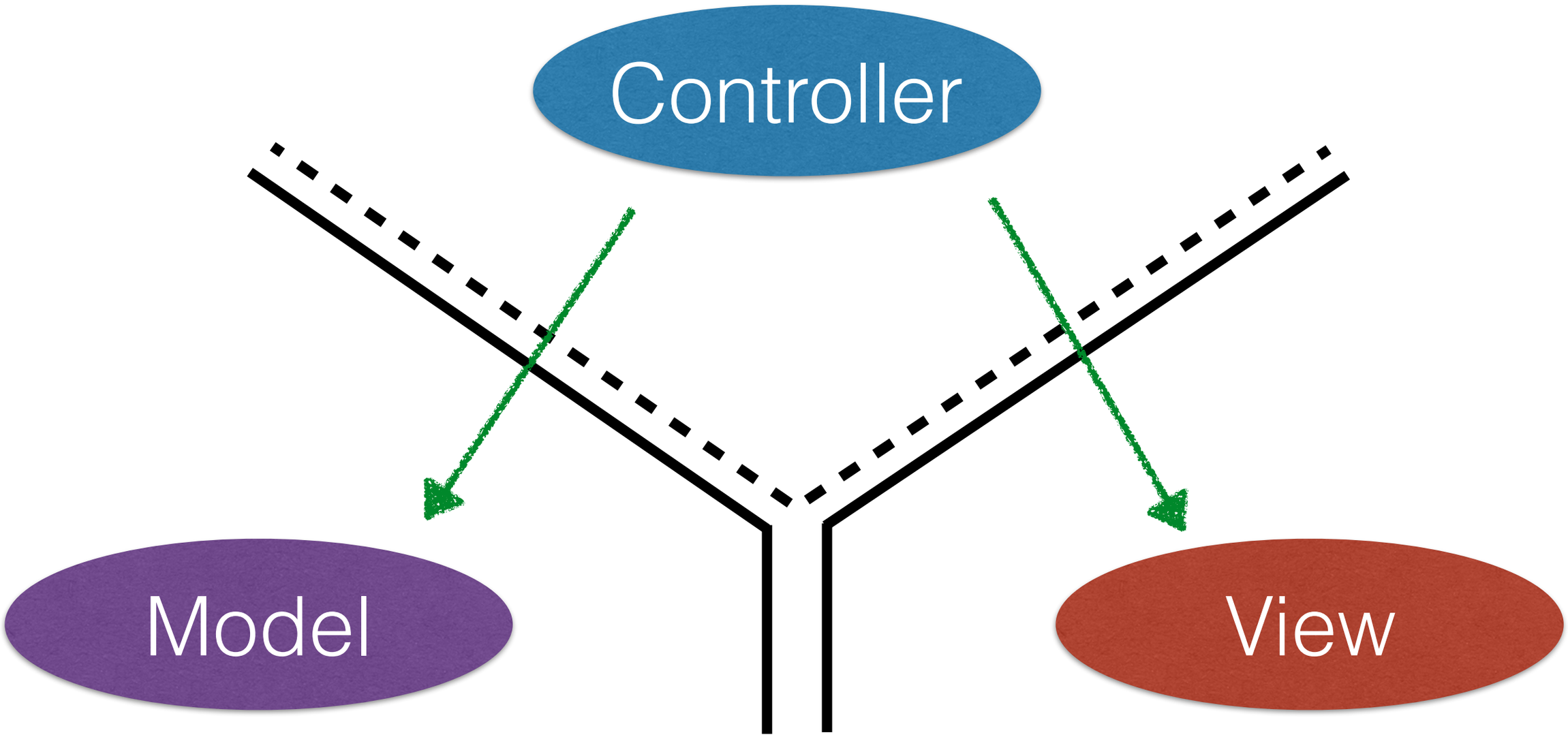
View

MVC

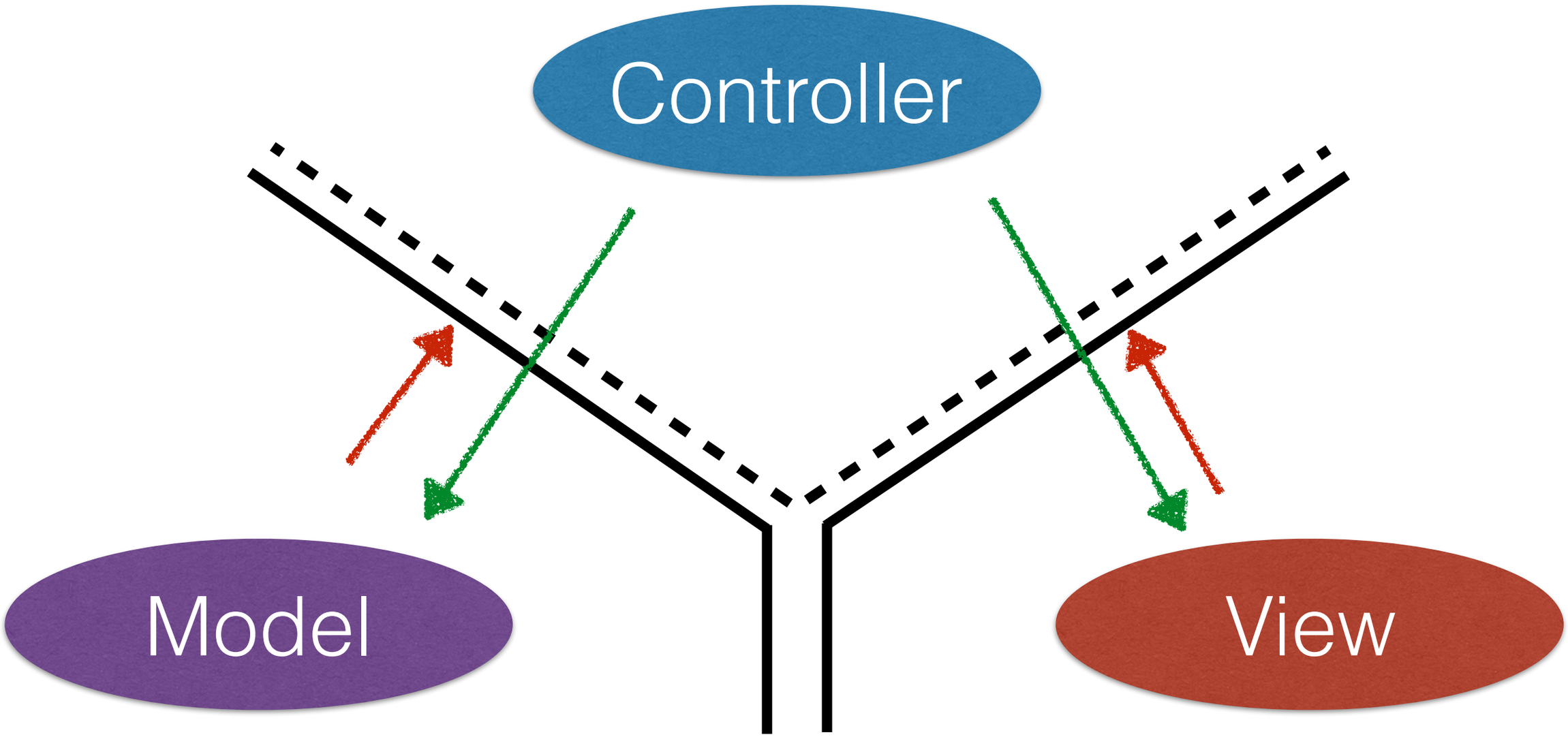


* This example (here and below) is taken from Stanford University CS193P

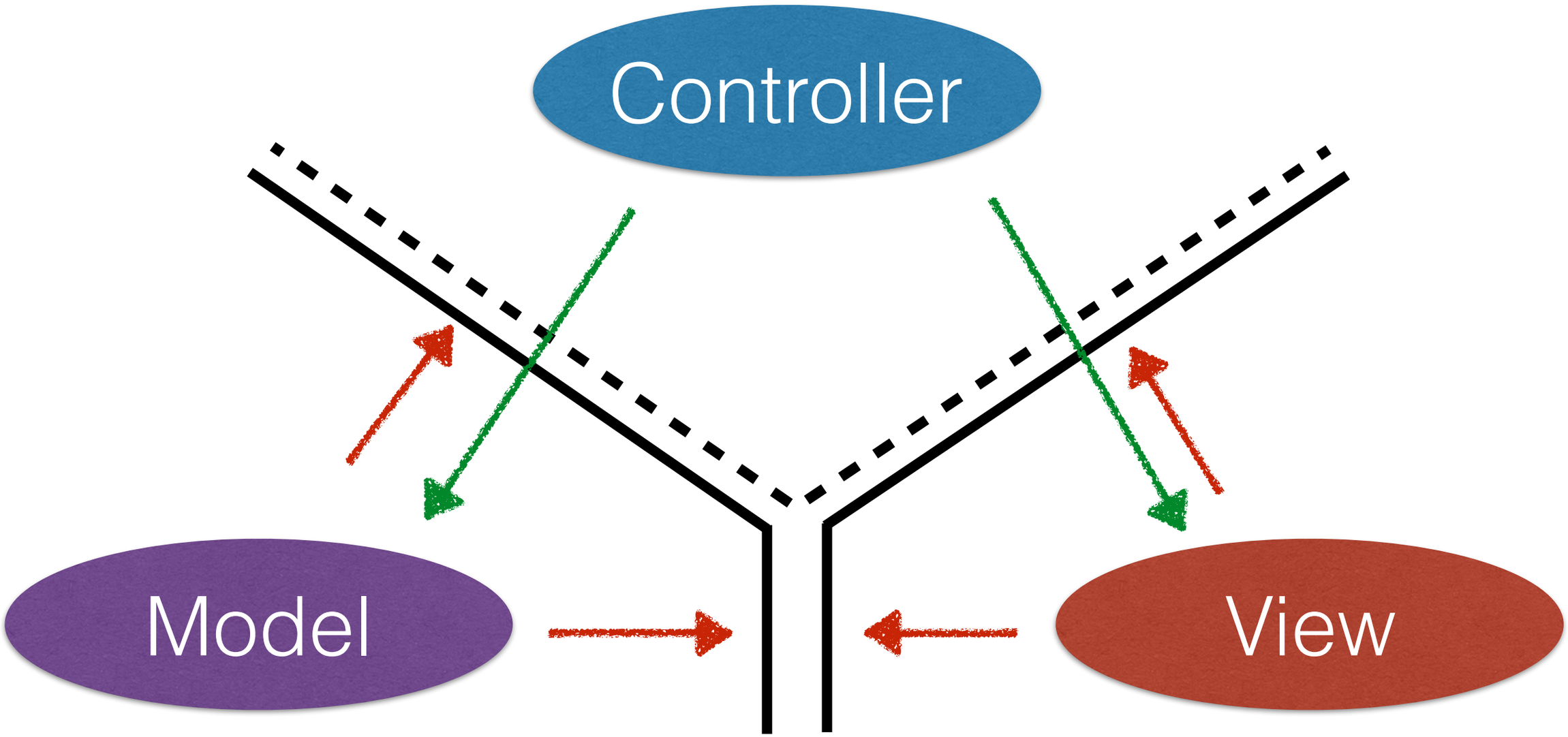
MVC



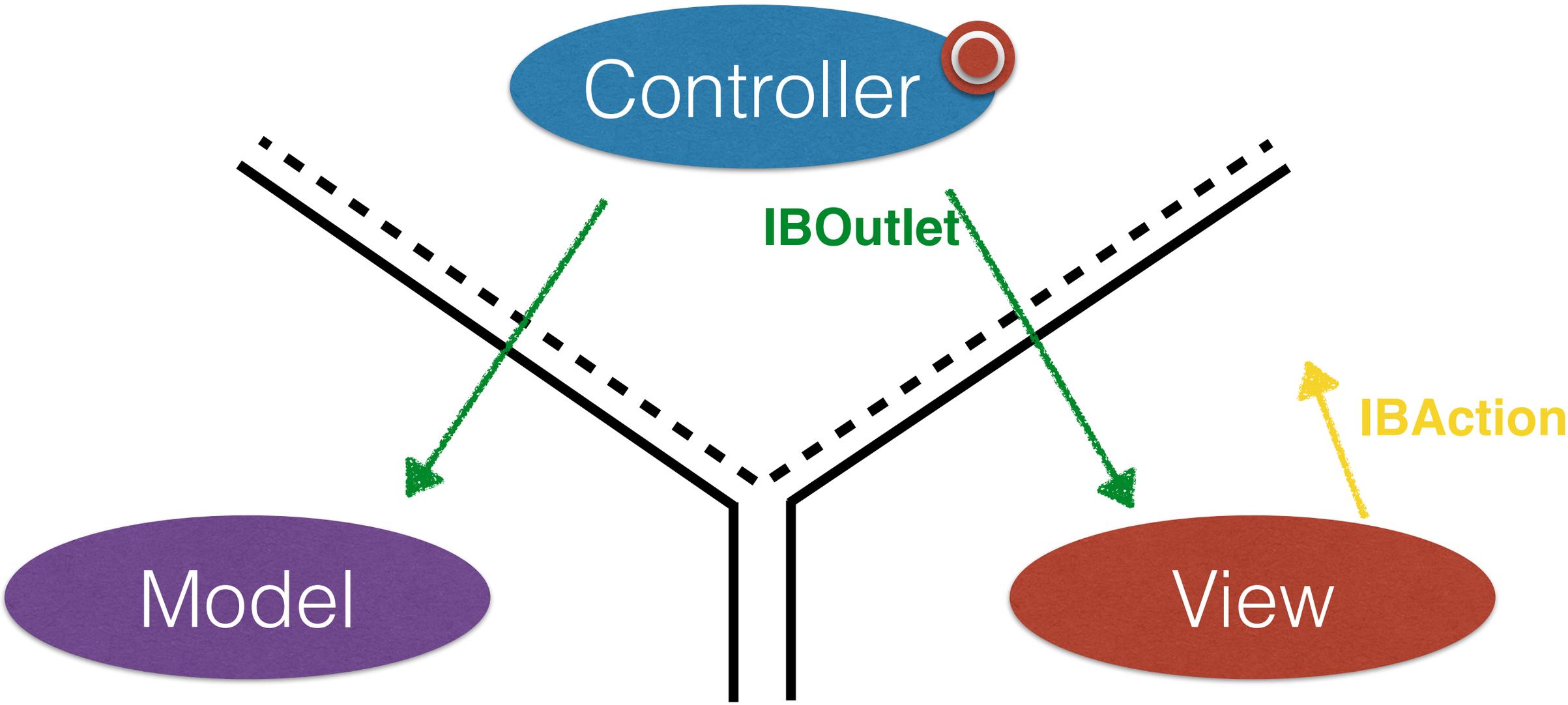
MVC



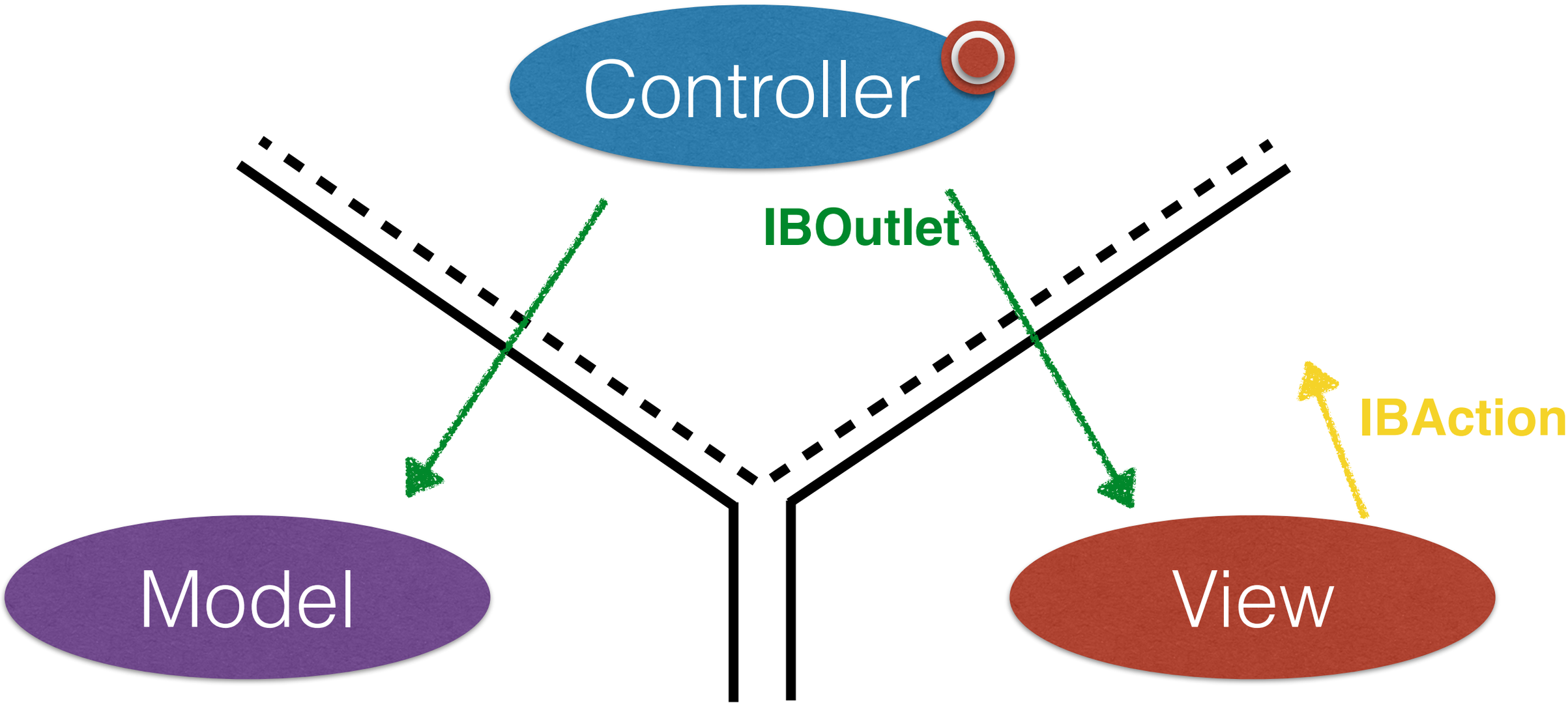
MVC



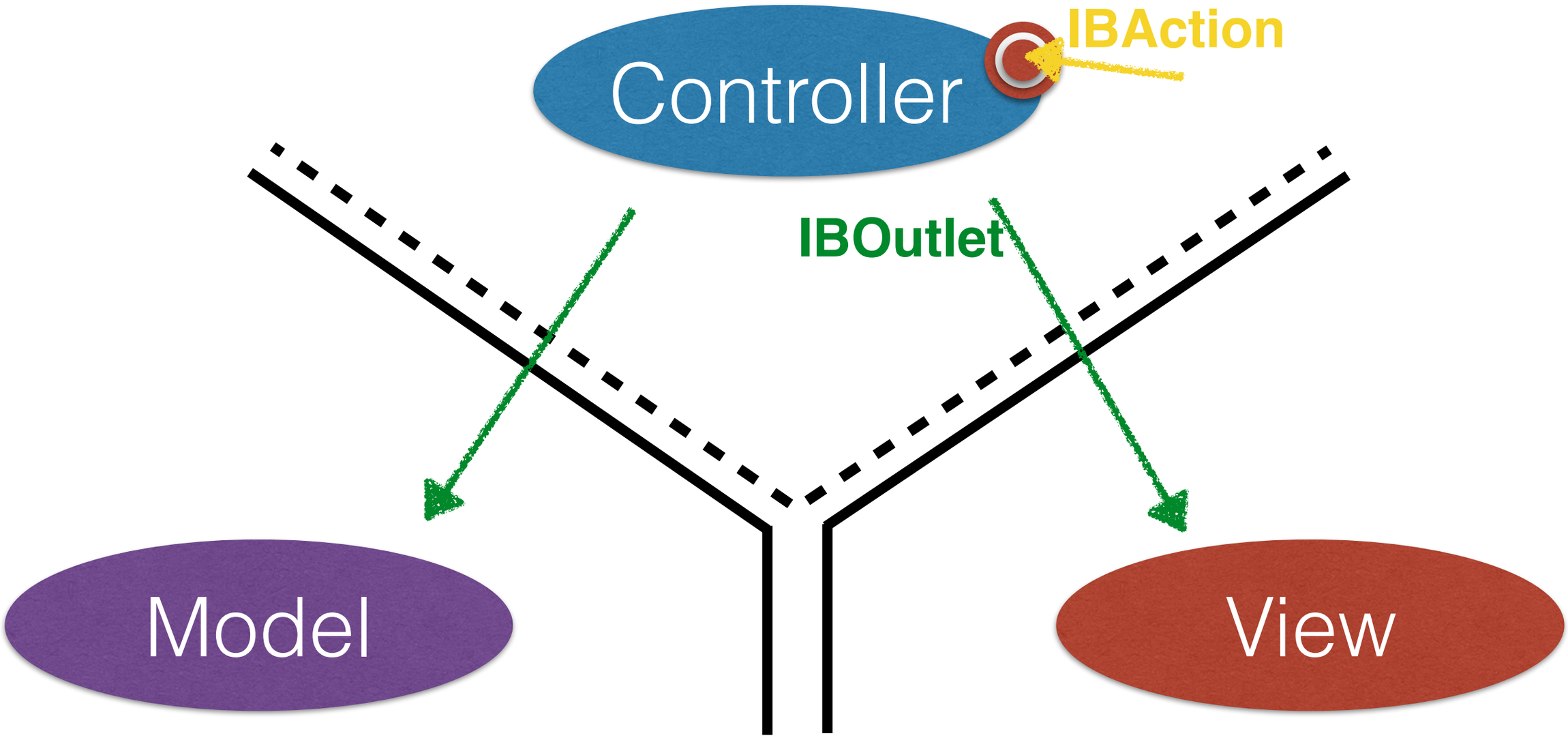
MVC



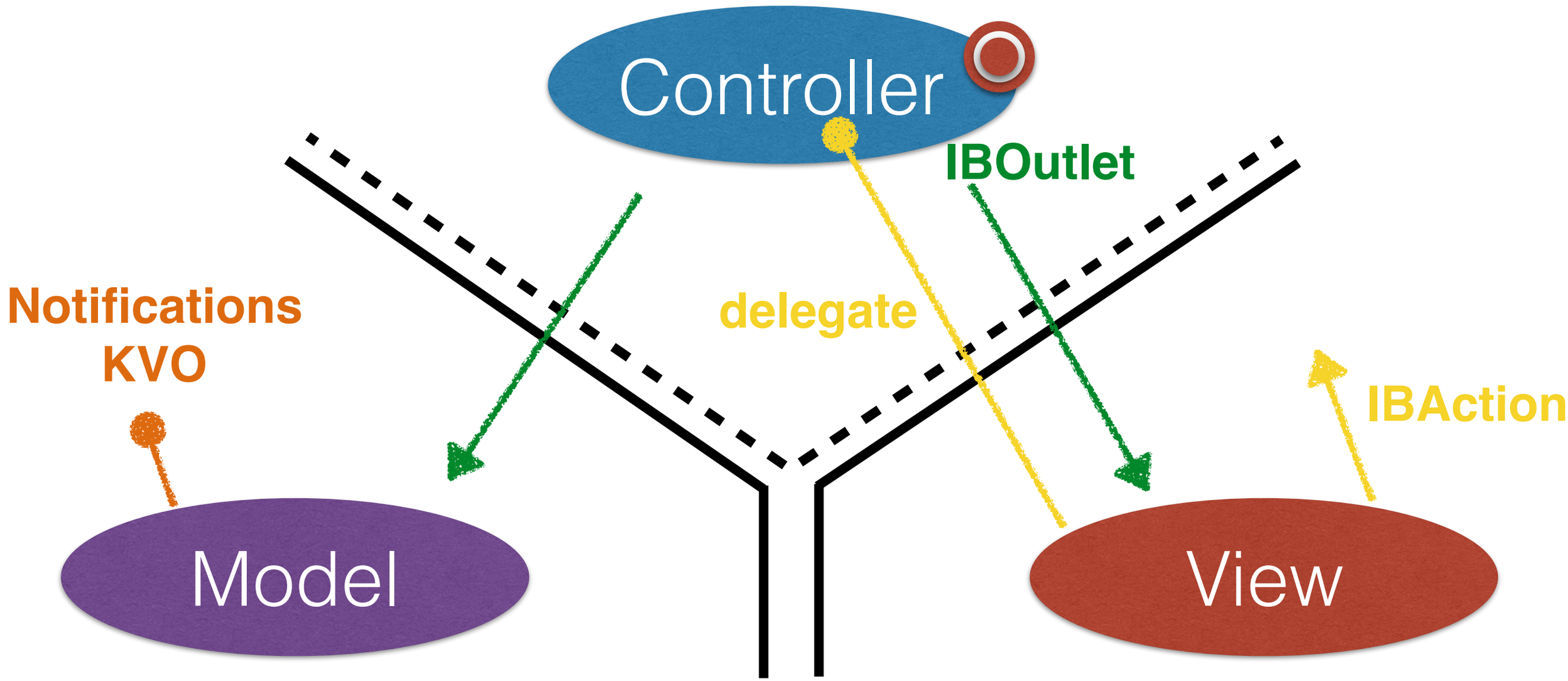
MVC



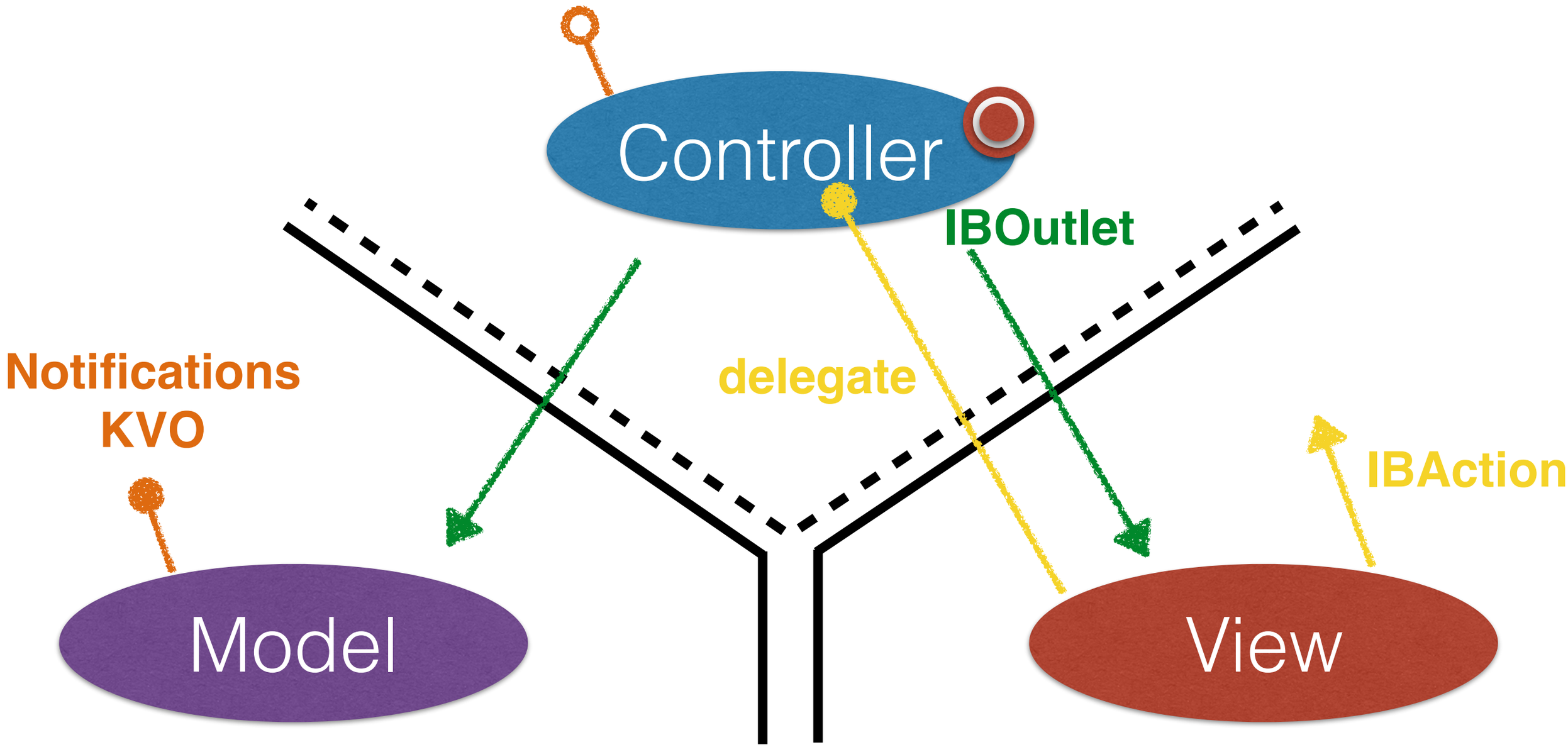
MVC



MVC


Controller
IBOutlet
Notifications
KVO
delegate
IBAction
Model
View

MVC


Controller
IBOutlet
Notifications
KVO
delegate
IBAction
Model
View

Structures and Classes

- Define properties
- Define methods
- Have initialisers
- Can be extended (via extension)*
- Conform to protocols*

* Later in the course

Structures

- Super light-weight classes
- Almost all std types are structures
- Copied by value (=> do not use ARC)
- Near C structures performance
- Default initialiser
- Do not inherit

Structures example

```
struct Location {
    let latitude: Float
    let longitude: Float
}

struct University {
    let name: String
    let acronim: String
    let location: Location

    var fullName: String {
        return "University with \(name), acronim \(acronim)"
    }
}

let ttuLocation = Location(latitude: 59.394868, longitude: 24.661387)
let ttu = University(name: "Tallinn ", acronim: "TTÜ", location: ttuLocation)
```

enum - Swift's superhero

- Computed properties
- Methods

```
struct Location {
    let latitude: Float
    let longitude: Float
}

enum ParallelsDevelopmentOffice {
    case moscow, tallinn, malta

    var location: Location {
        switch self {
        case .moscow: return Location(latitude: 55.756151, longitude: 37.61727)
        case .tallinn: return Location(latitude: 59.436960, longitude: 24.753574)
        case .malta: return Location(latitude: 35.909072, longitude: 14.506489)
        }
    }

    func distance(to location: Location) -> Float {
        //the body here
    }
}
```

enum - Swift's superhero

- Raw values

```
enum ParallelsDevelopmentOffice: String {
    case moscow = "Moscow", tallinn = "Tallinn", malta = "Malta"
}
```

- Associated values

```
enum ServerResponse {
    case error(String)
    case success
}
```

- Default initializer

- Conform to protocols, can be extended, value type

Nested types

- You can declare a type inside another type, which is inside third type
- Resulting type will be GrandParent.Parent.Child

```
struct Layout {

    enum Direction {
        case horizontal, vertical
    }

    let direction: Direction

}
let horizontalDirectionOutside = Layout.Direction.horizontal
```

Demo...