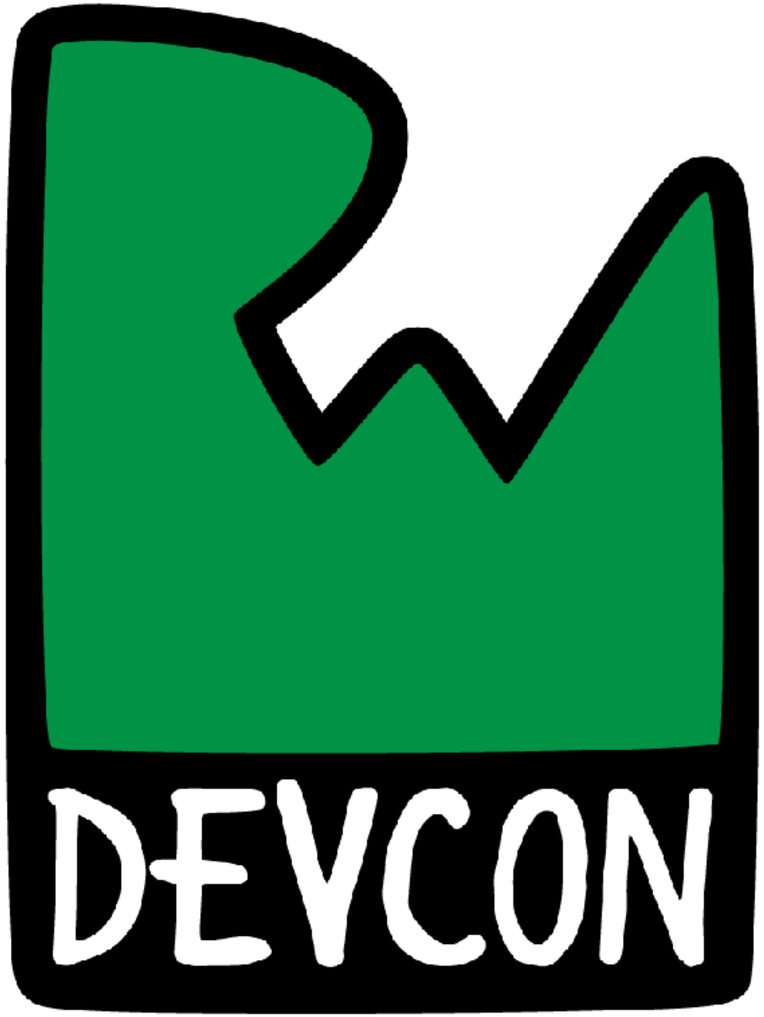
Session 6: Clean Architecture on iOS

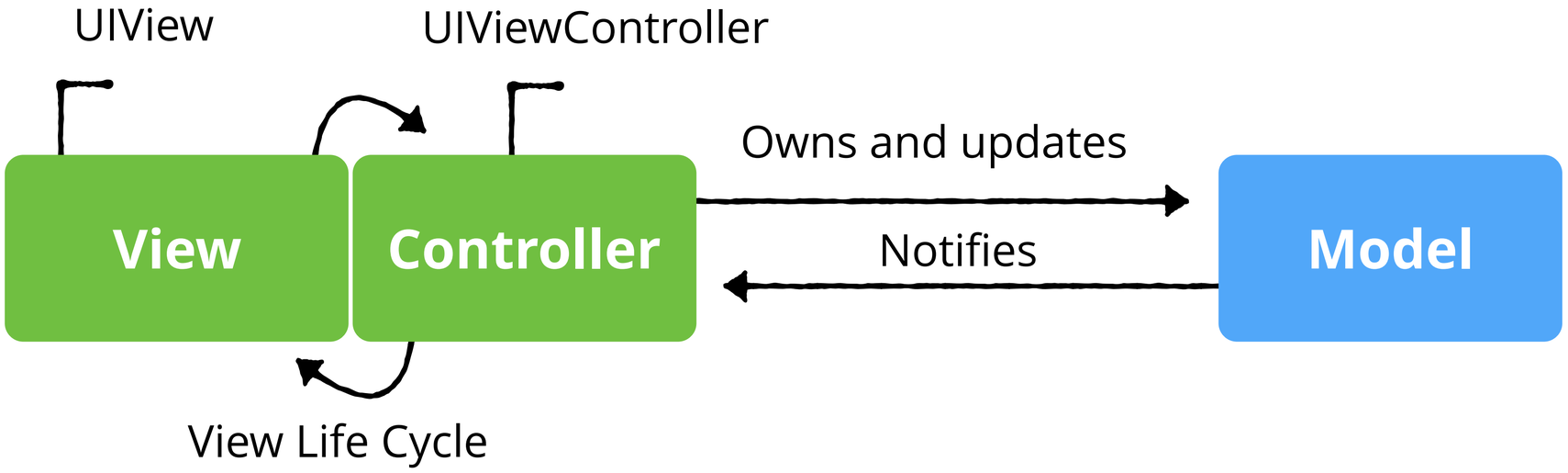


Theory


Carrier
8:49 PM
Todo List
You have no todos.
Relax and continue learning.
DEVCON

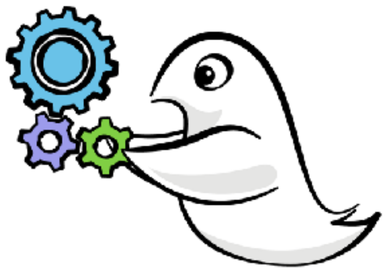
MVC Architecture

- Model View Controller aka Massive View Controller
- Defacto standard



What is Clean Architecture?

- High level guideline
- Flexible, Maintainable, Testable
- Meta Architecture for creating layered architectures
- Separation of concerns



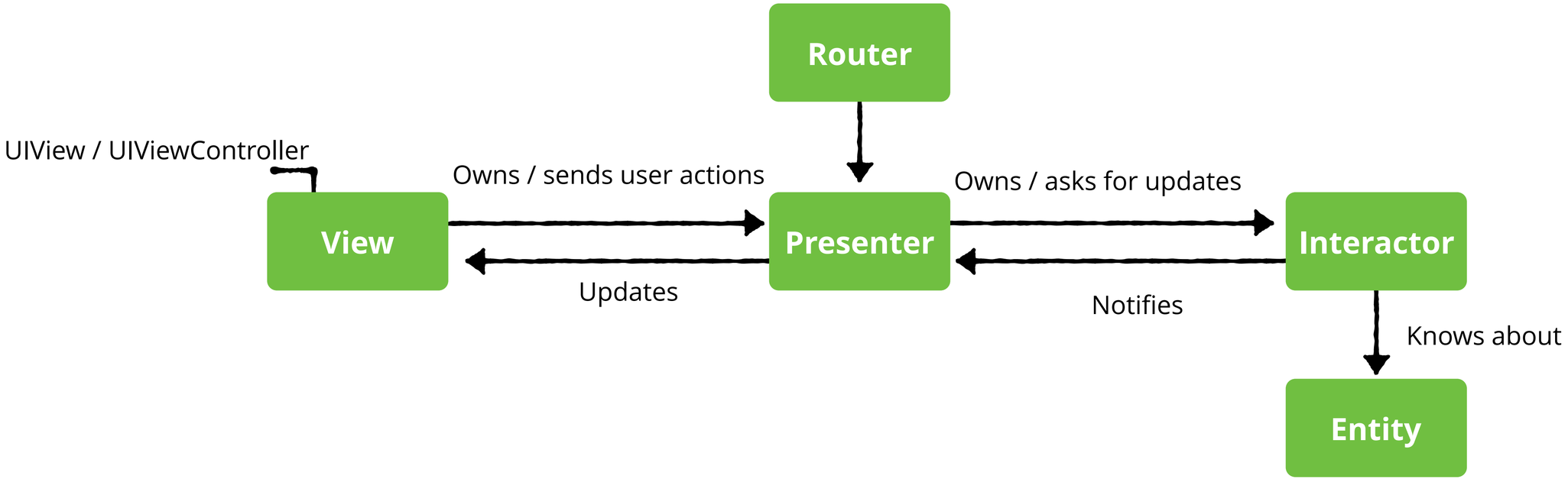
Clean Architecture Guidelines

- Independent of frameworks
- Independent of UI
- Independent of services
- Testable

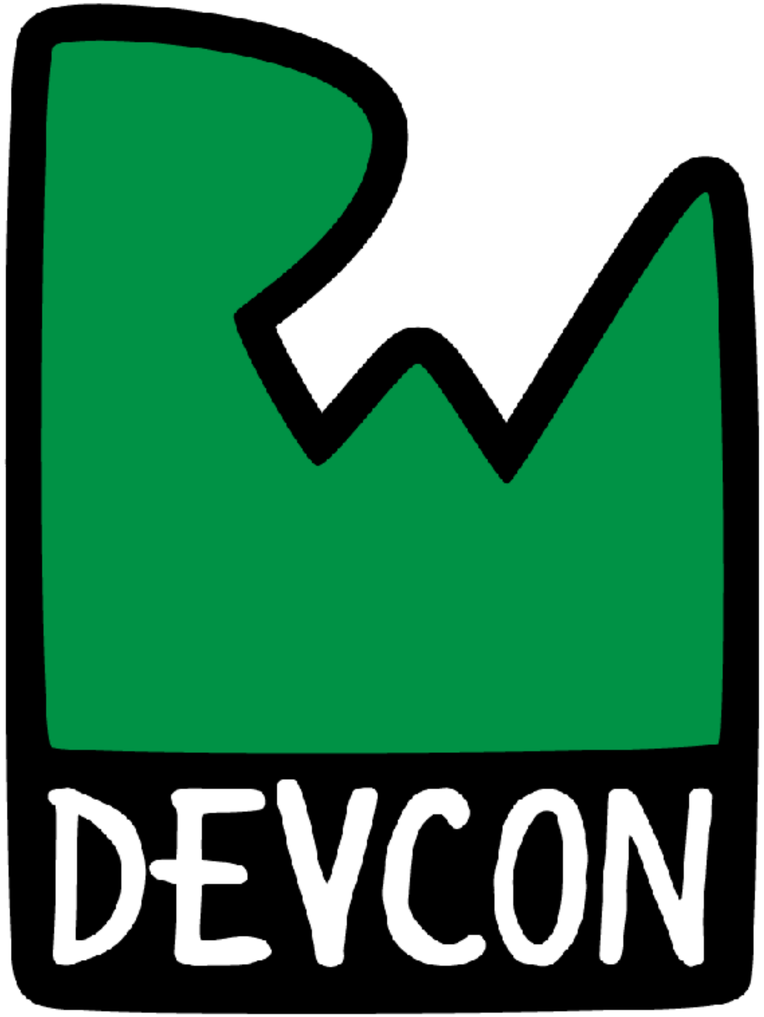
Viper

- View, Interactor, Presenter, Entity, Router
- Created by Mutual Mobile
- Conforms to the Single Responsibility Principle
- Application of Clean Architecture

Viper Diagram



Session 6: Clean Architecture on iOS

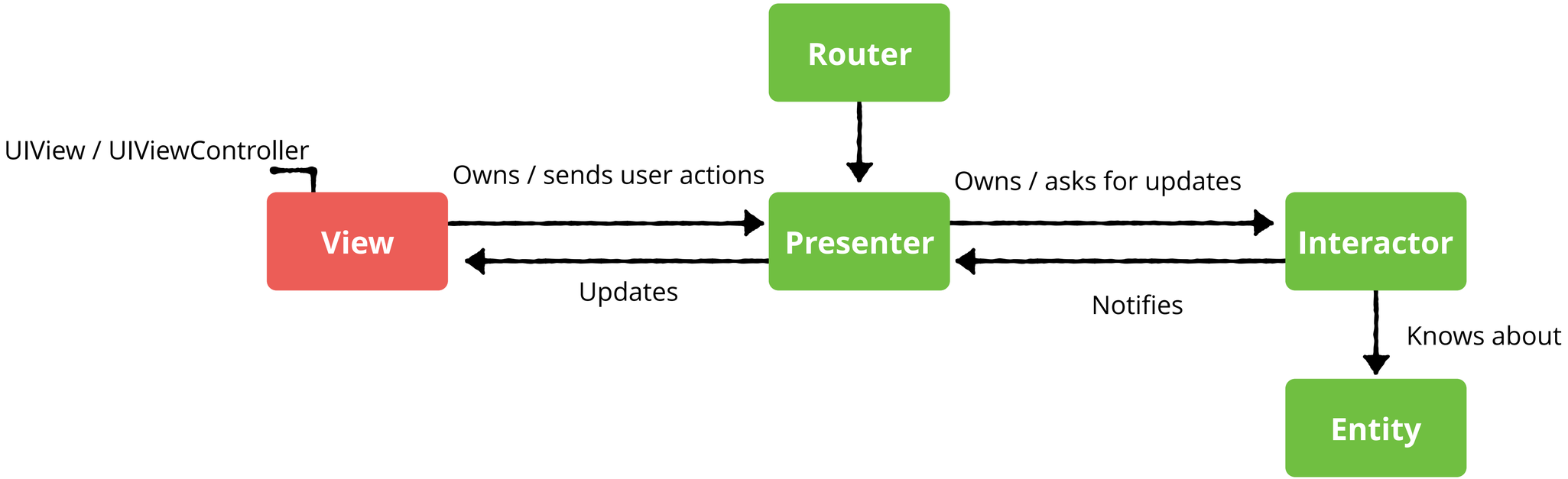


View, Interactor & Entity

View

- A view or view controller
- Told by the presenter which data to display
- Relays user input back to the presenter

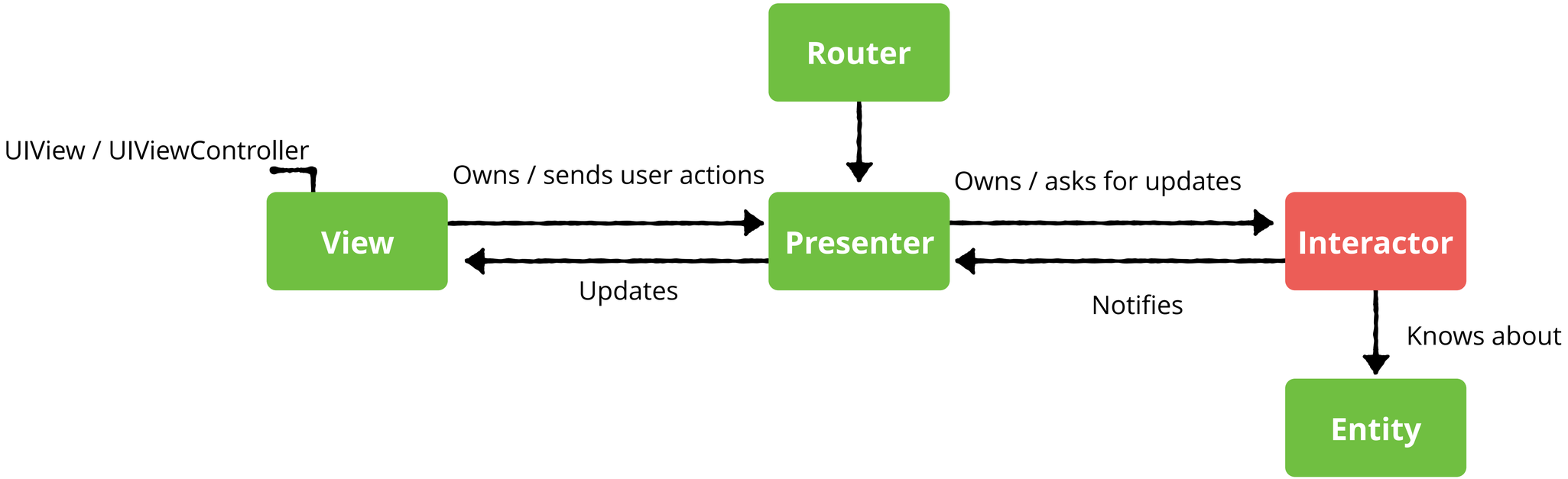
View Diagram



Interactor

- Holds all of the application business logic
- Represents a single use case
- Notifies the presenter of changes from the entity
- PONSO (Plain Old NSObject)

Interactor Diagram



Interactor Example

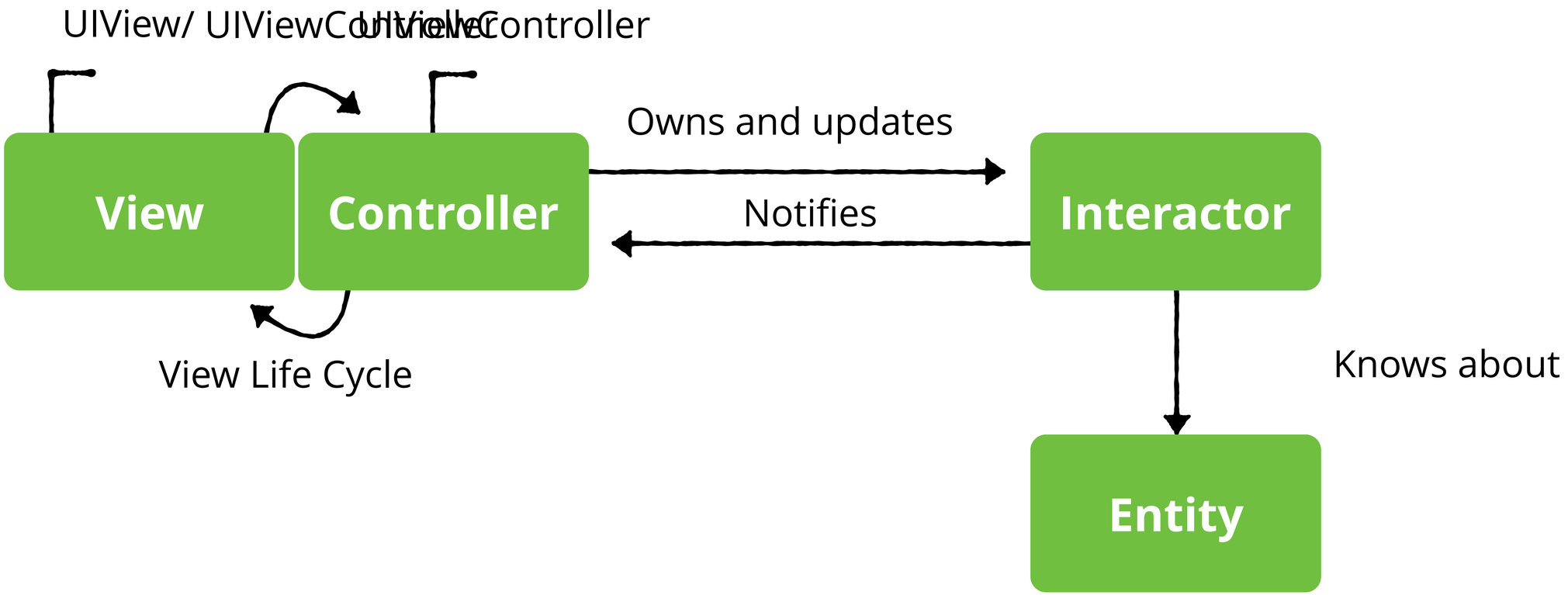
```
func retrieveTodoList() {
    do {
      if let todoItems = try dataManager?.retrieveTodoList() {
        if !todoItems.isEmpty {
          presenter?.didRetrieveTodos(todoItems)
        }

      }
    } catch {
      presenter?.didRetrieveTodos([])
    }
  }
}
```

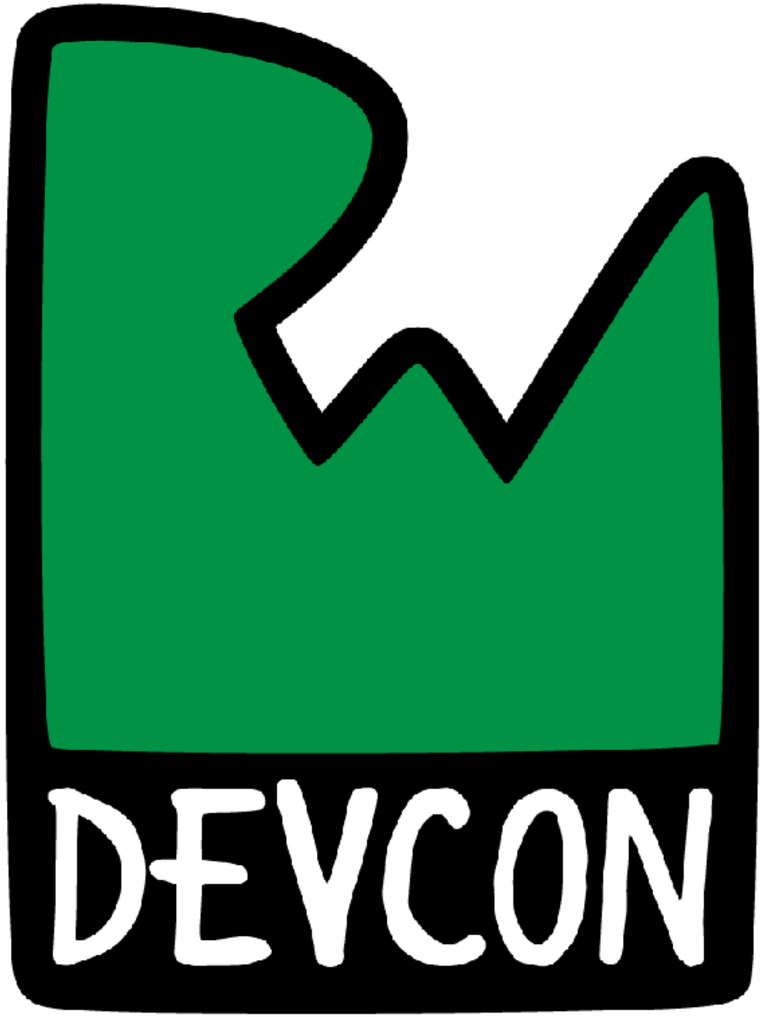
Demo 1

Demo 1 Review

- Simple todo list application built with MVC
- Created the interactor and entity
- The interactor owns all of the business logic
- Notifies the presenter when changes happen in the entity


UIView/ UIViewController
UIViewController
View
Controller
View Life Cycle
Owns and updates
Notifies
Interactor
Knows about
Entity

Session 6: Clean Architecture on iOS

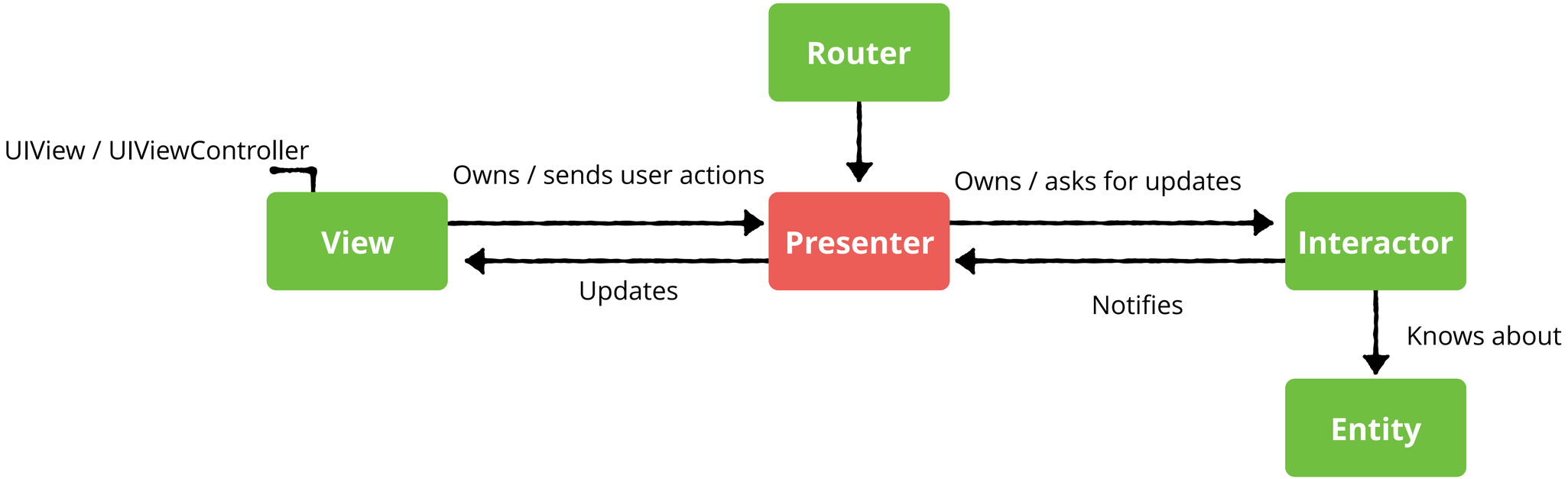


Presenter

Presenter

- ~~A part of a interior architecture called MVP~~
- PONSO (Plain Old NSObject)
- Consists of logic to drive the UI
- The Presenter is decoupled from the view
- Updates the view with data passed from the Interactor
- Reacts to user input, asks for updates upon input

Presenter Diagram



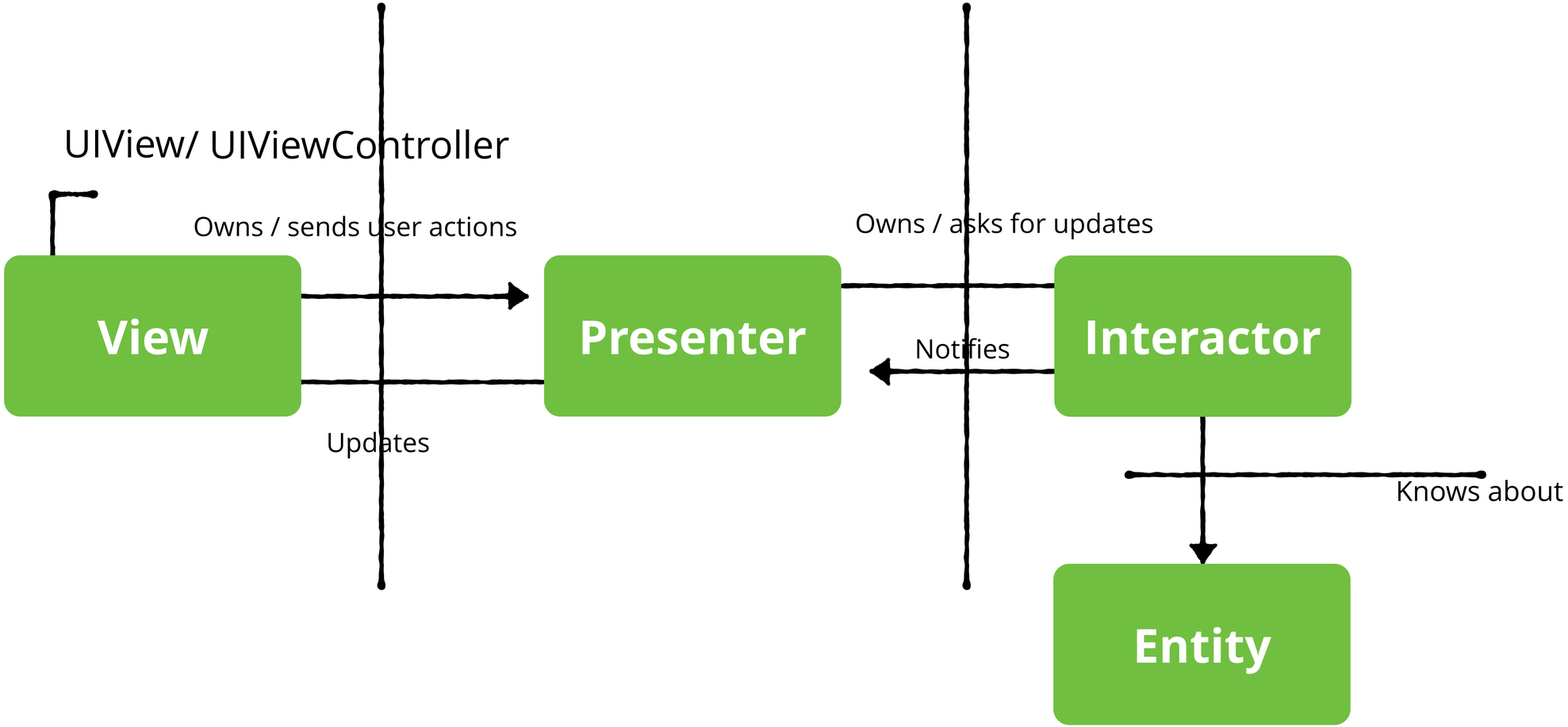
Presenter Example

```
func didRetrieveTodos(_ todos: [TodoItem]) {
    view?.show(items: todos)
  }
```

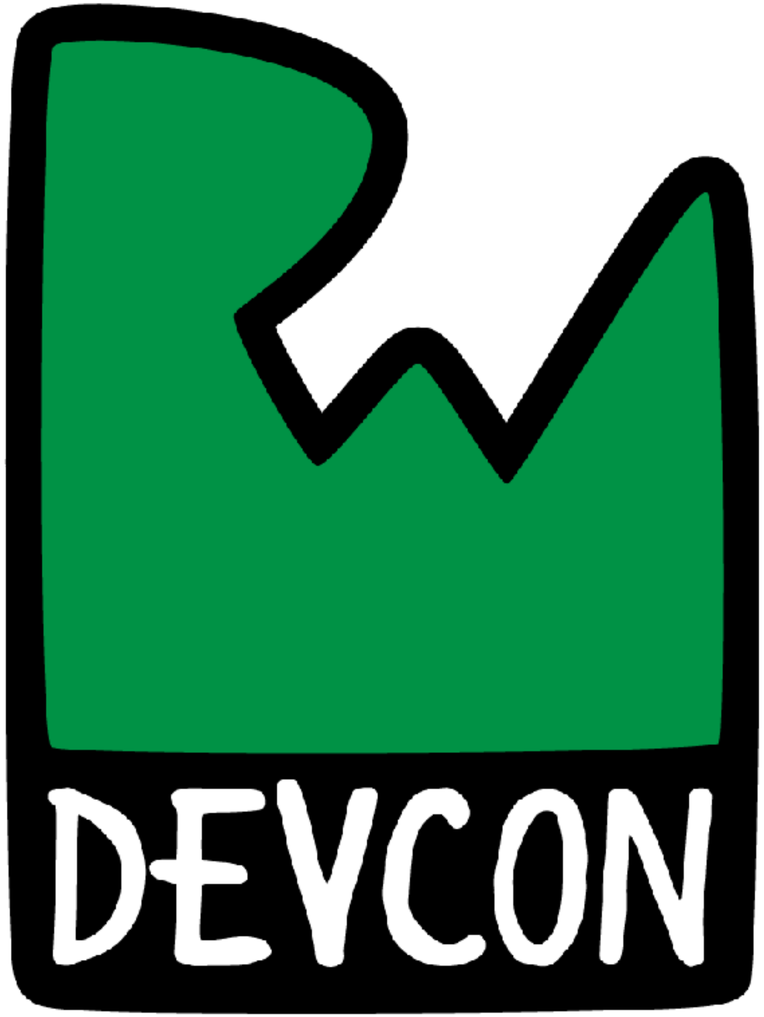
Demo 2

Demo 2 Review

- Created the presenter protocols (use cases)
- Created the presenter
- Updates the view with logic from the interactor
- Reacts to user input, asks for updates upon input
- Cleaned up old MVC code


UIView/ UIViewController
Owns / sends user actions
View
Presenter
Updates
Owns / asks for updates
Notifies
Interactor
Knows about
Entity

Session 6: Clean Architecture on iOS

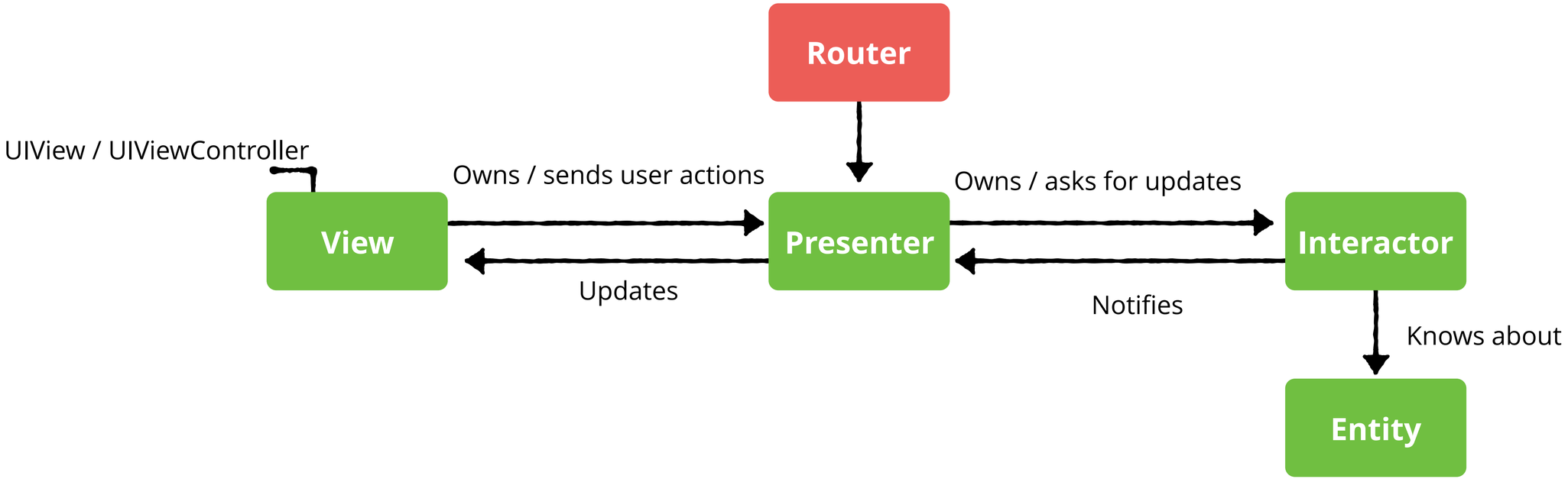


Router

Router

- Handles navigation from one screen to another
- Responsible for creating the interactor and presenter
- Responsible for creating View and ViewControllers and installing them on the window
- This is where you should hold any navigation animations
- Owns UIWindow, UINavigationController, UIViewControlle

Router Diagram



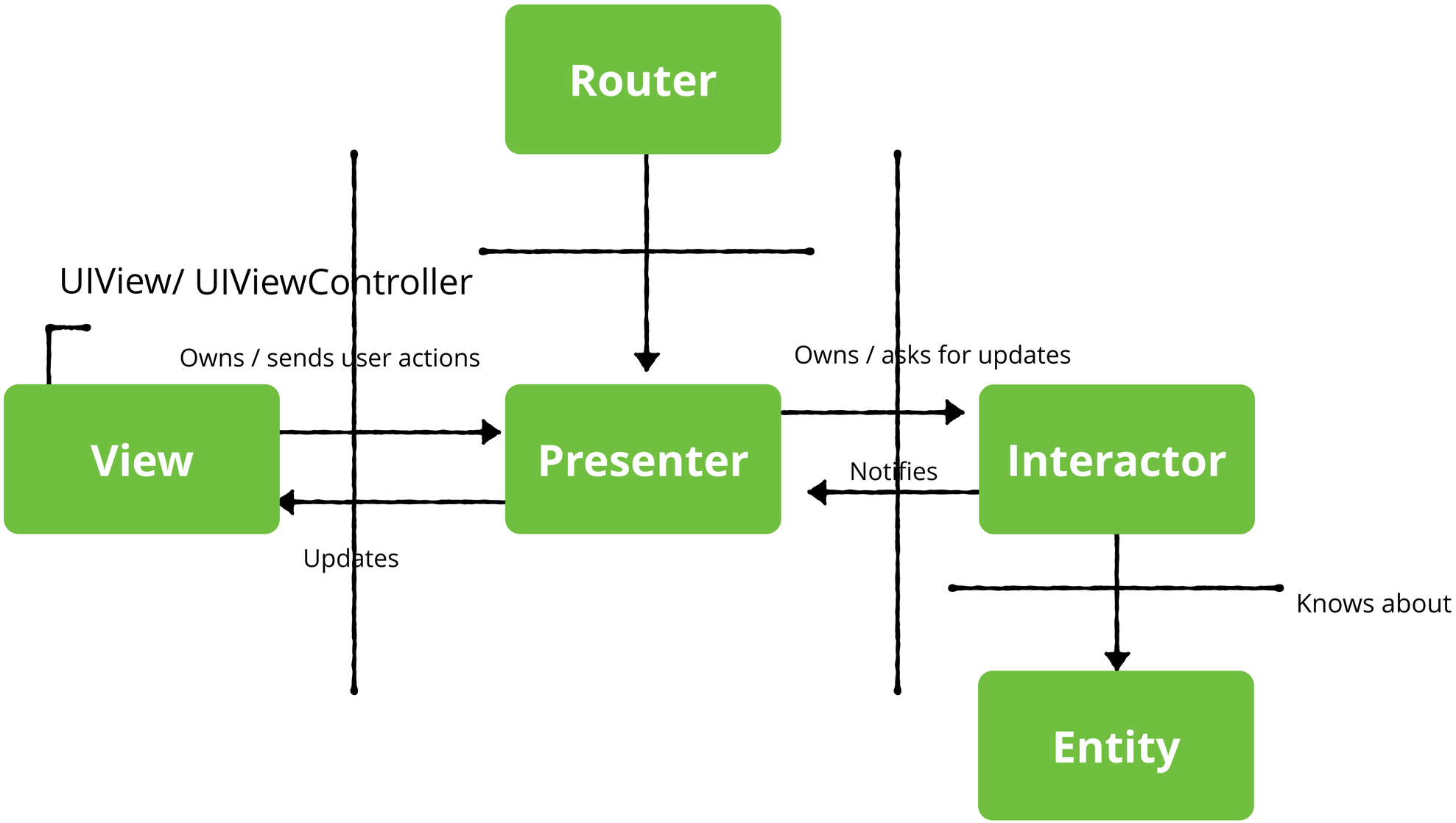
Router Example

```
class func createTodoListController() -> UIViewController {
    let navController = mainStoryboard.instantiateViewController(withIdentifier:
"TodoListNavController")
    guard let todoListView = navController.childViewControllers.first as?
ListViewController else {
      return UIViewController()
    }
```

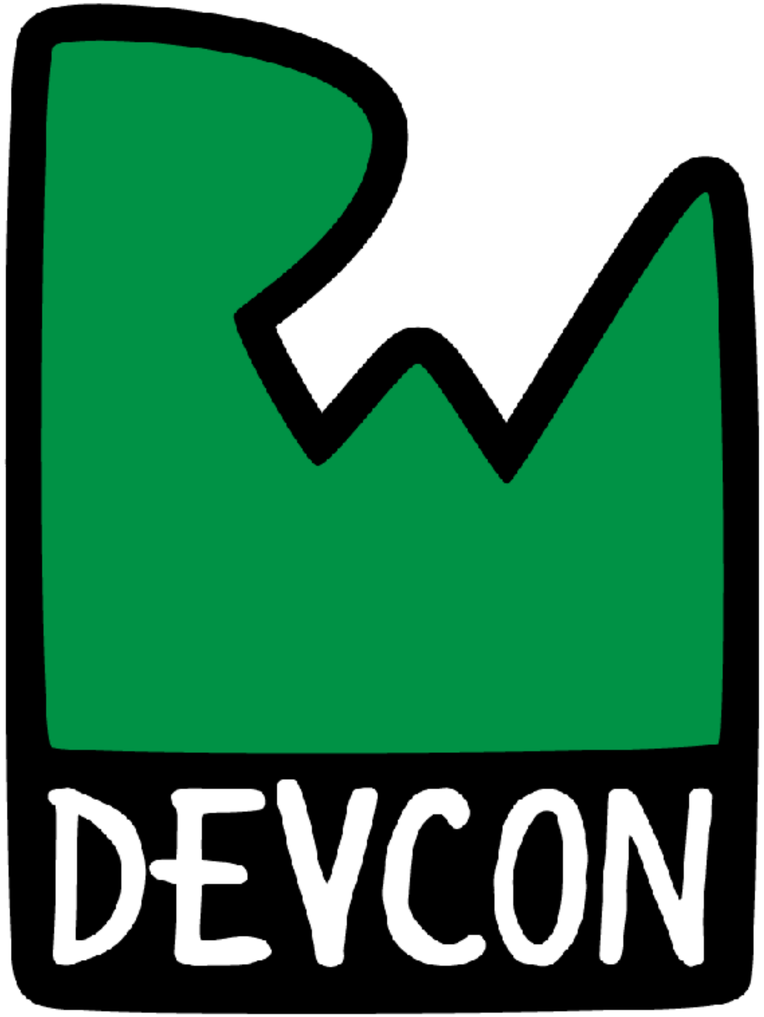
Demo 3

Demo 3 Review

- ~~Created the router / wireframe protocols (use cases)~~
- Created the router
- Configured the application to use the router


Router
UIView/ UIViewController
Owns / sends user actions
View
Presenter
Updates
Owns / asks for updates
Notifies
Interactor
Knows about
Entity

Session 6: Clean Architecture on iOS



Conclusion

What You Learned

- Clean architecture can be done several ways, it is just a set of guidelines.
- Large code bases can be easily decoupled into a clean architecture with some ease.

What You Learned: Demo 1

- Interactor and Entities
- Holds all of the application business logic
- Notifies the presenter of changes from the entity

What You Learned: Demo 2

- Presenters
- Updates the view with logic from the interactor
- Reacts to user input, asks for updates upon input

What You Learned: Demo 3

- Routers
- Decides what screen is displayed at what specific time
- Responsible for creating View and ViewControllers and installing them on the window

Where To Go From Here?

- Free tutorials on our site
- Uncle Bob's Clean Architecture
- Clean Architecture (Android) Edison 8-9:45 tomorrow
- Advanced Unidirectional Architecture. F/G 12:30 - 2:45 tomorrow
- Previous RWDevCon talks
- Twitter: @TheRealLockett