

# Что мы узнаем сегодня?

---



- MVC
- MVC от Apple





# Model View Controller (MVC)

---

# MVC

---



- Model - предоставляет данные и реагирует на команды контроллера, изменяя своё состояние
- View - отвечает за отображение данных модели пользователю, реагируя на изменения модели
- Controller - интерпретирует действия пользователя, оповещая модель о необходимости изменений

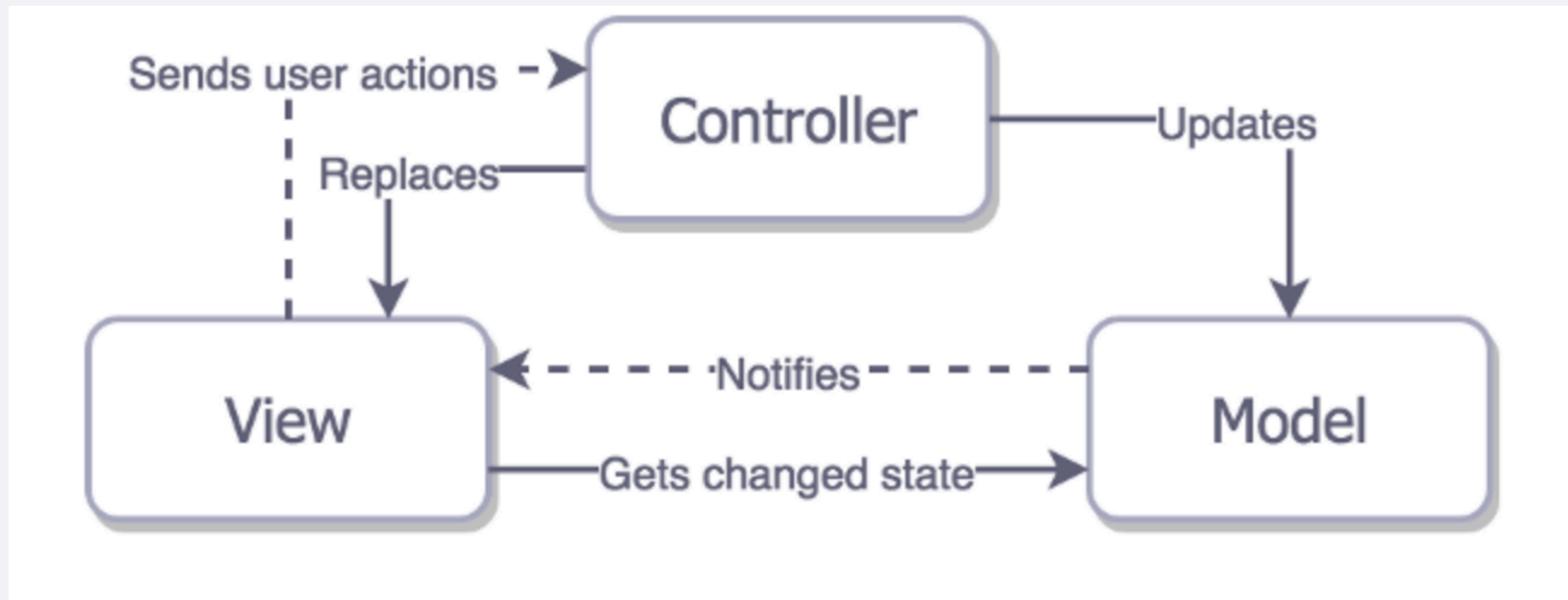
# MVC

---

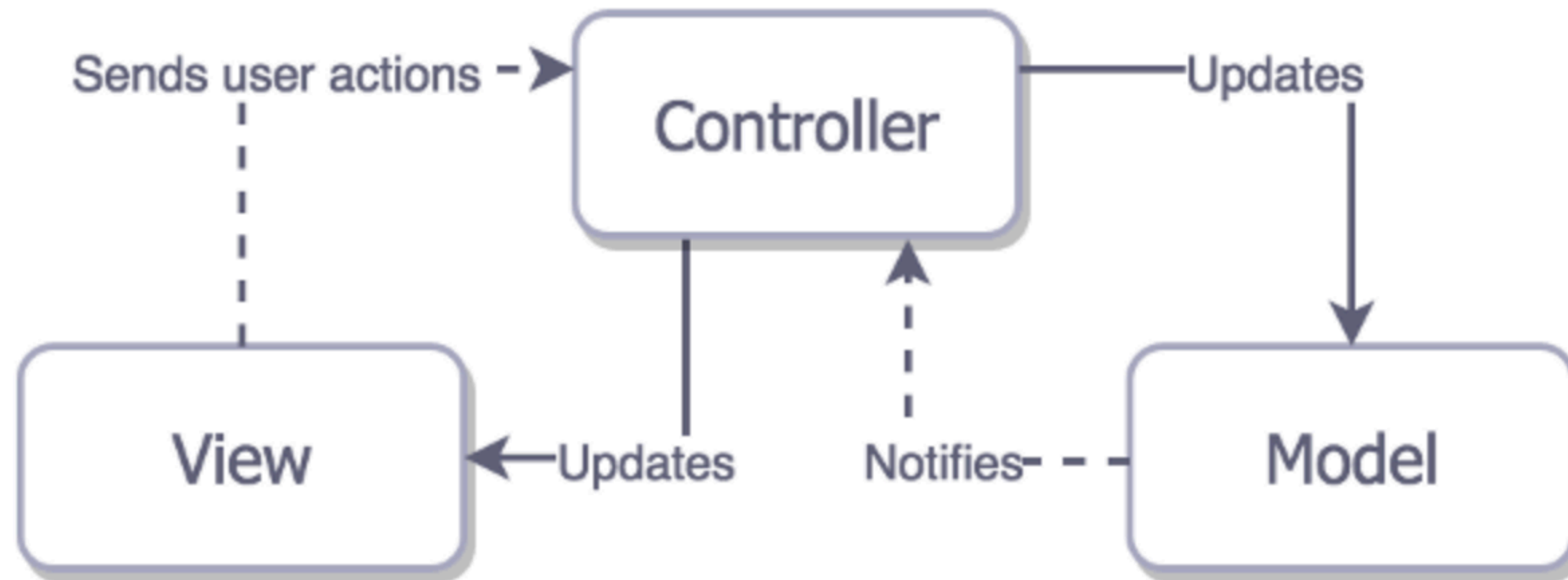


- Схема разделения данных приложения
- Концепция MVC была описана Трюгве Реенскаугом в 1978 году
- Поскольку MVC не имеет строгой реализации, то реализован он может быть по-разному. Нет общепринятого определения, где должна располагаться бизнес-логика. Она может находиться как в контроллере, так и в модели. В последнем случае, модель будет содержать все бизнес-объекты со всеми данными и функциями.

# Традиционный MVC

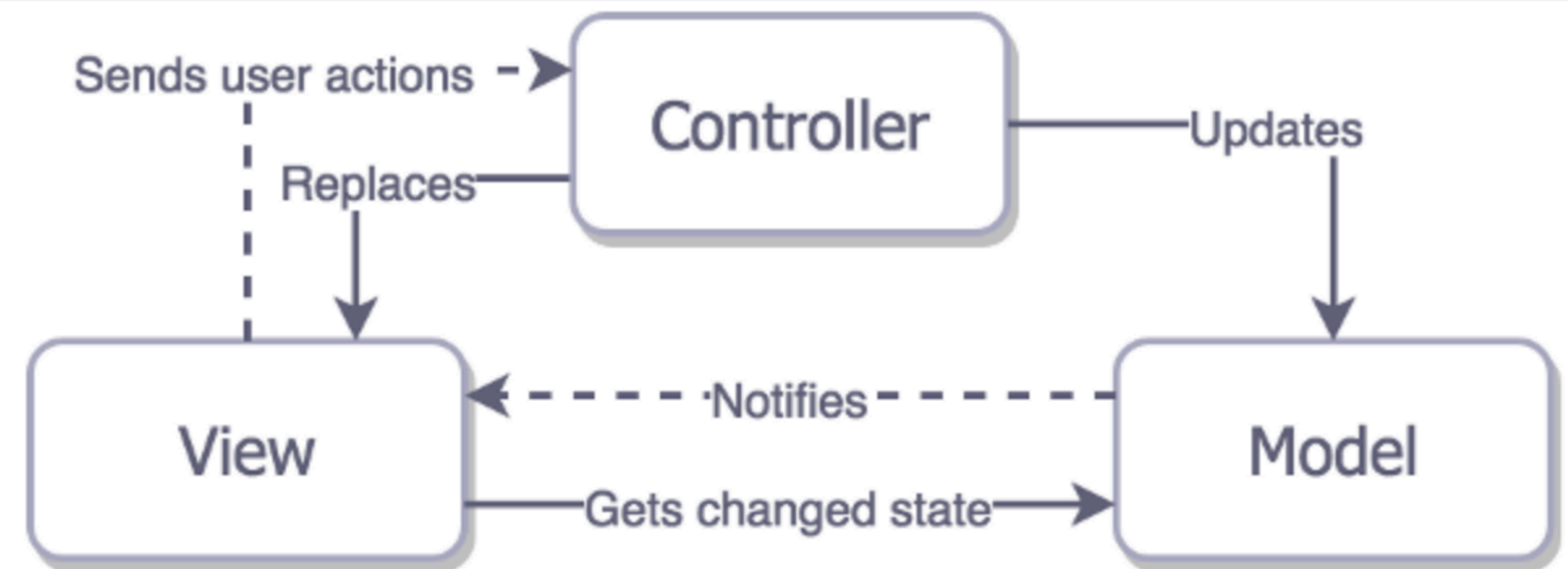
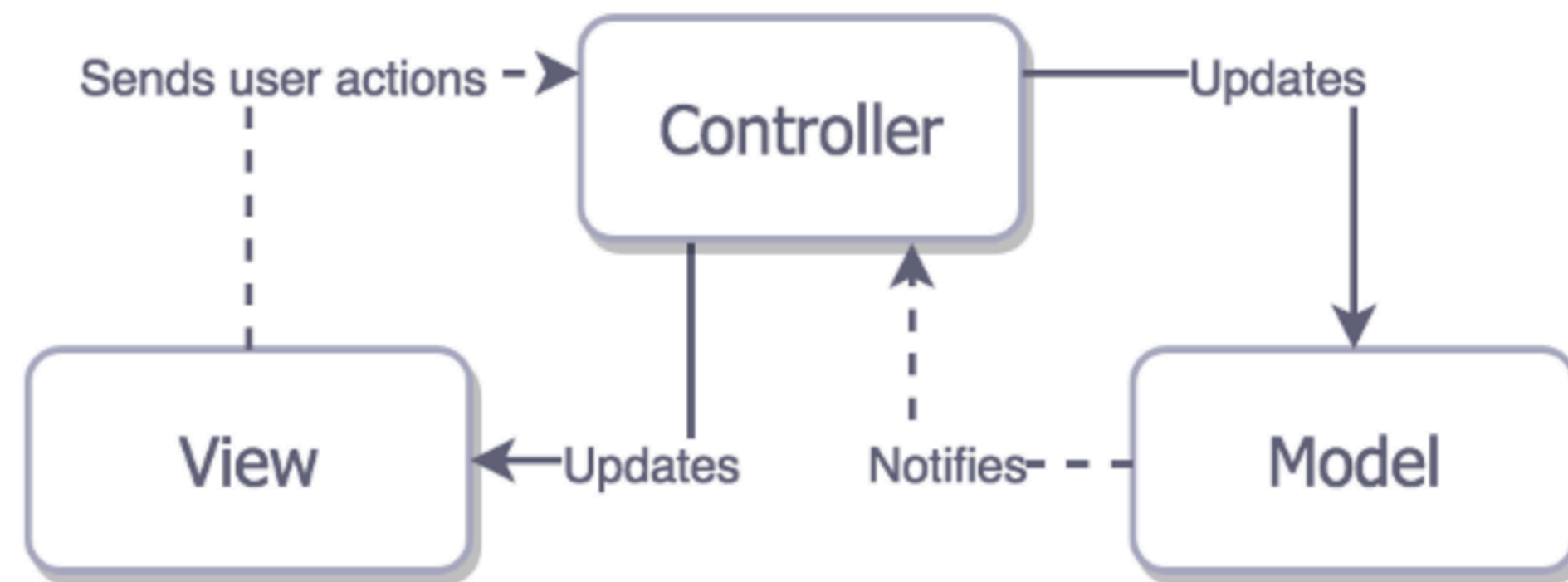


# MVC от Apple



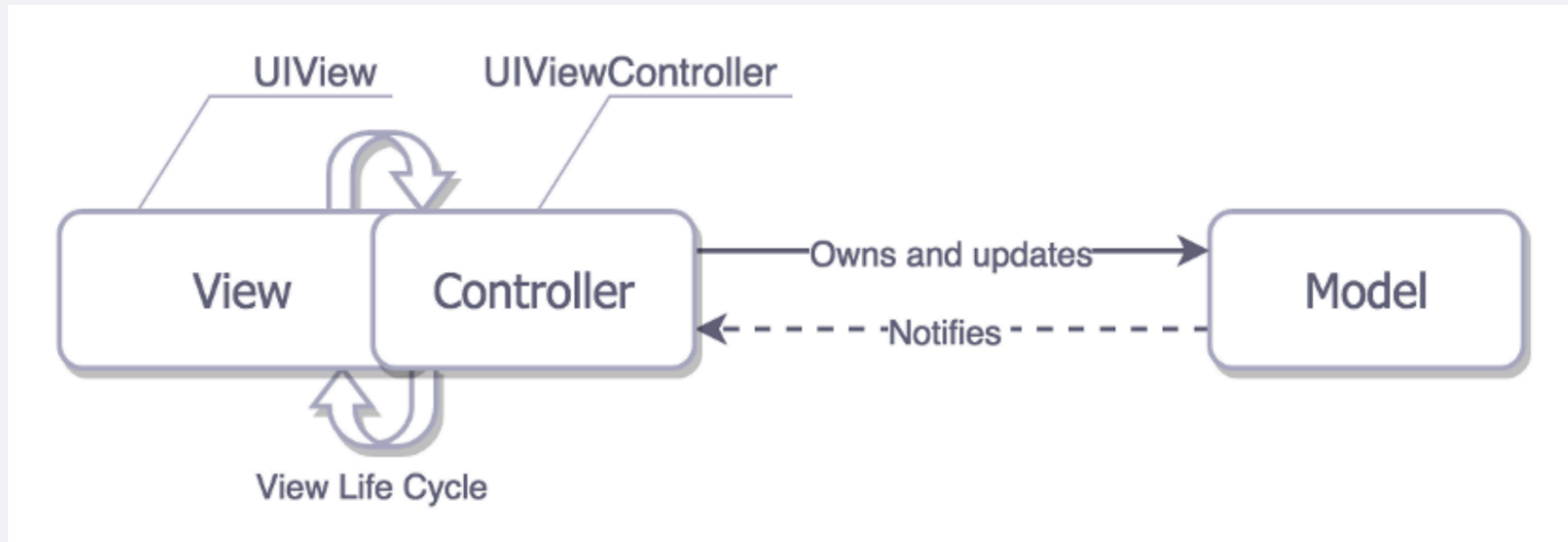


# MVC от Apple vs Традиционный MVC





# Massive ViewController







# MVC: На самом деле

- Самый сложный архитектурный дизайн
- Гибкая и масштабируемая архитектура
- View не связан жестко с Controller
- Можно писать тесты
- MVC подчиняется SOLID
- Архитектура совместимая с MVVM, Flux, VIPER и др.



# SOLID

---

- **S** — The Single Responsibility Principle, SRP, Принцип единственной ответственности
- **O** — The Open Closed Principle, OCP, Принцип открытости/закрытости
- **L** — The Liskov Substitution Principle, Принцип подстановки Барбары Лисков
- **I** — The interface Segregation Principle, ISP, Принцип разделения интерфейса
- **D** — The dependency Inversion Principle, DIP, Принцип инверсии зависимостей



# Православный MVC в iOS SDK

- Content Controllers
- Container Controllers
- Coordinating Controllers
- Model Controllers



# Content Controller

- Ваш UIViewController с переопределенным loadView
- UITableViewController
- UICollectionViewController
- UISearchDisplayController
- AVPlayerViewController

@IBOutlets, constraints, recognizers, создания и настройка view



# Container Controller

---

- UISplitController
- UITabBarController
- UINavigationController
- UIPageViewController



# Coordinating Controllers

- UIStoryboardSegue
- UIDocumentBrowserTransitionController
- UIViewControllerContextTransitioning
- UIViewControllerAnimatedTransitioning



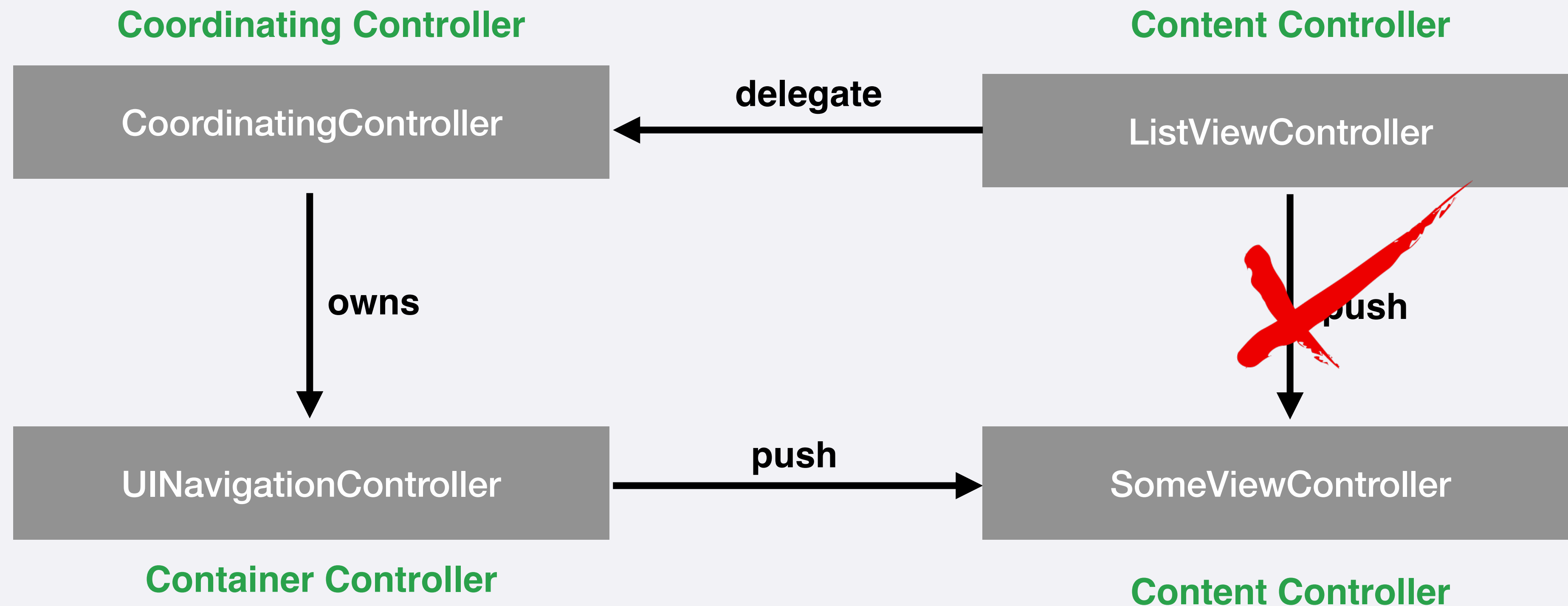
# Model Controllers

- NSFetchResultsController
- UIDocument
- NSArrayController

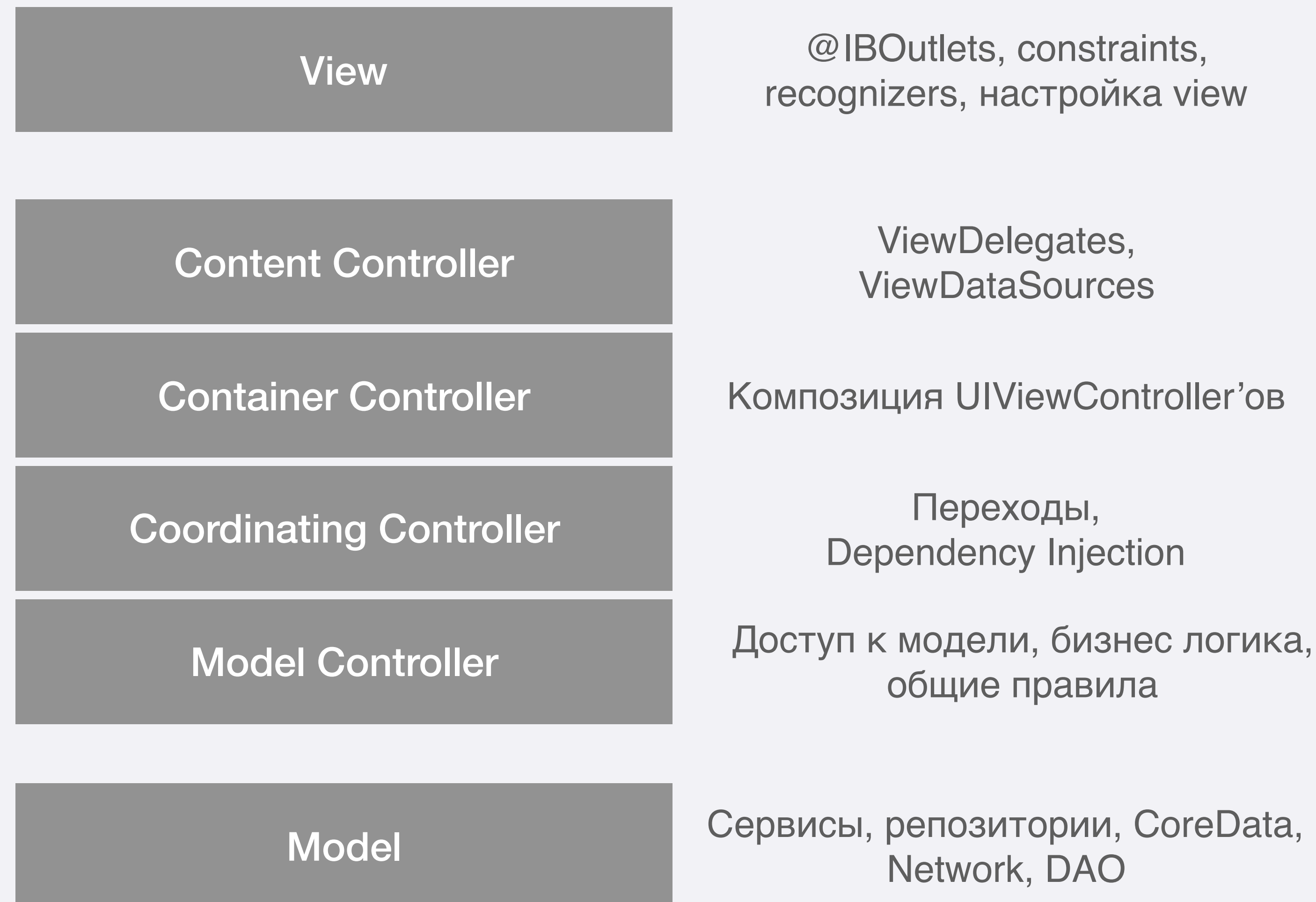
Controller != UIViewController



# MVC: Как разделить ответственность контроллеров

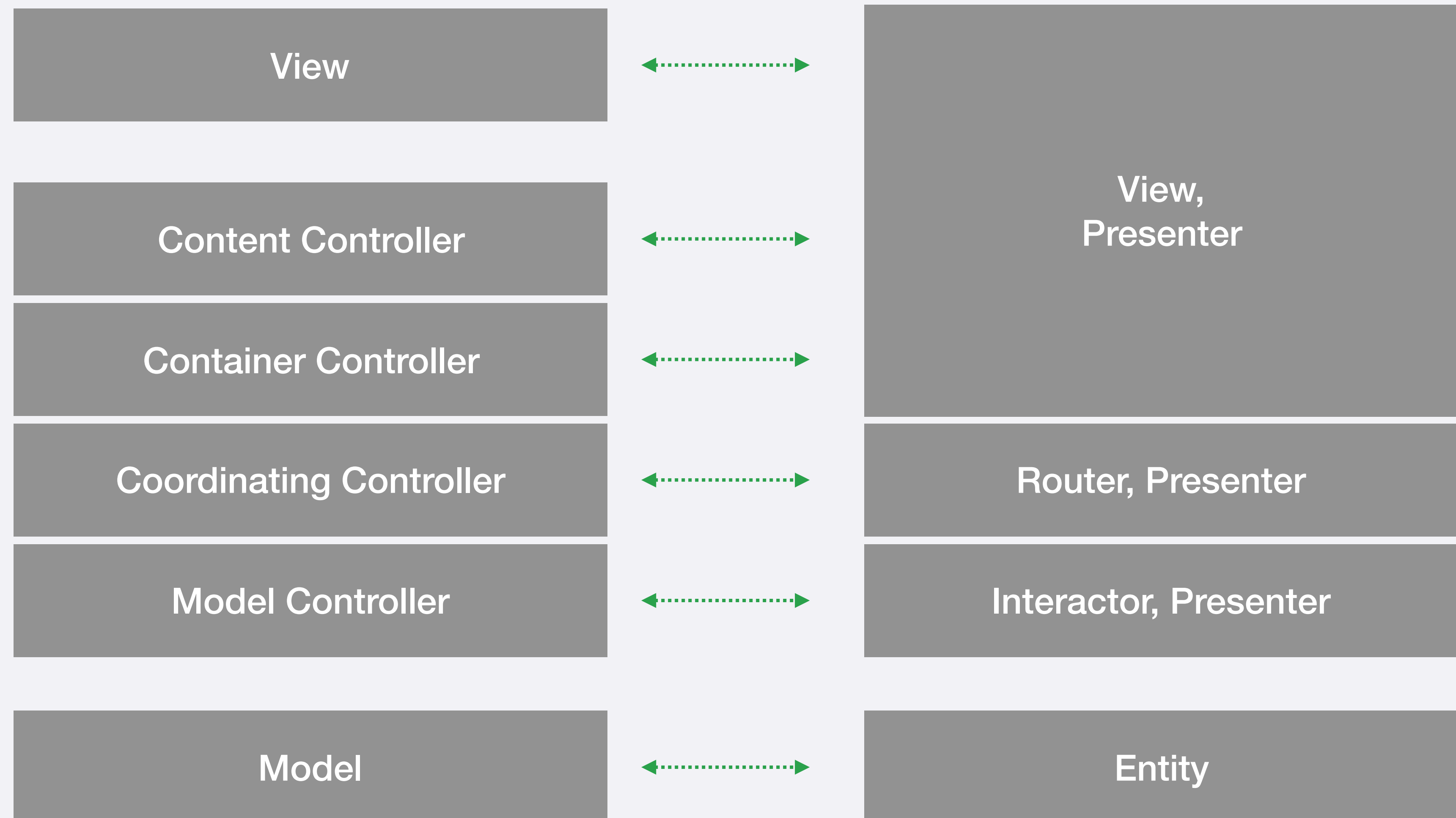


# MVC: Как разделить ответственность контроллеров



DAO — это класс, содержащий CRUD методы для конкретной сущности

# MVC: Как разделить ответственность контроллеров





Let's Code



# Полезные ссылки

---

- <https://ru.wikipedia.org/wiki/Model-View-Controller>
- <https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>
- <https://habr.com/ru/post/324414/>
- <https://habr.com/ru/company/badoo/blog/281162/>
- <https://medium.com/@yo.kononov/разбор-вопросов-на-собеседование-junior-ios-developer-dd969ecfbd41>

# Домашнее задание



- 
- Задание на коллекции с использованием MVC от Apple

# Обратная связь







Вопросы?



Спасибо ❤️