

UIKit

UIKit

UIKit

Core Animation

OpenGL ES / Open GL

Core Graphics

Graphics Hardware

- Использует под капотом Core Animation, а он свою очередь Core Graphics.
- Но UIKit, это не только представления, а еще и обработка пользовательских нажатий
- Самая высокая абстракция
- С iOS 13 - Metal вместо OpenGL

UIKit: UIApplication

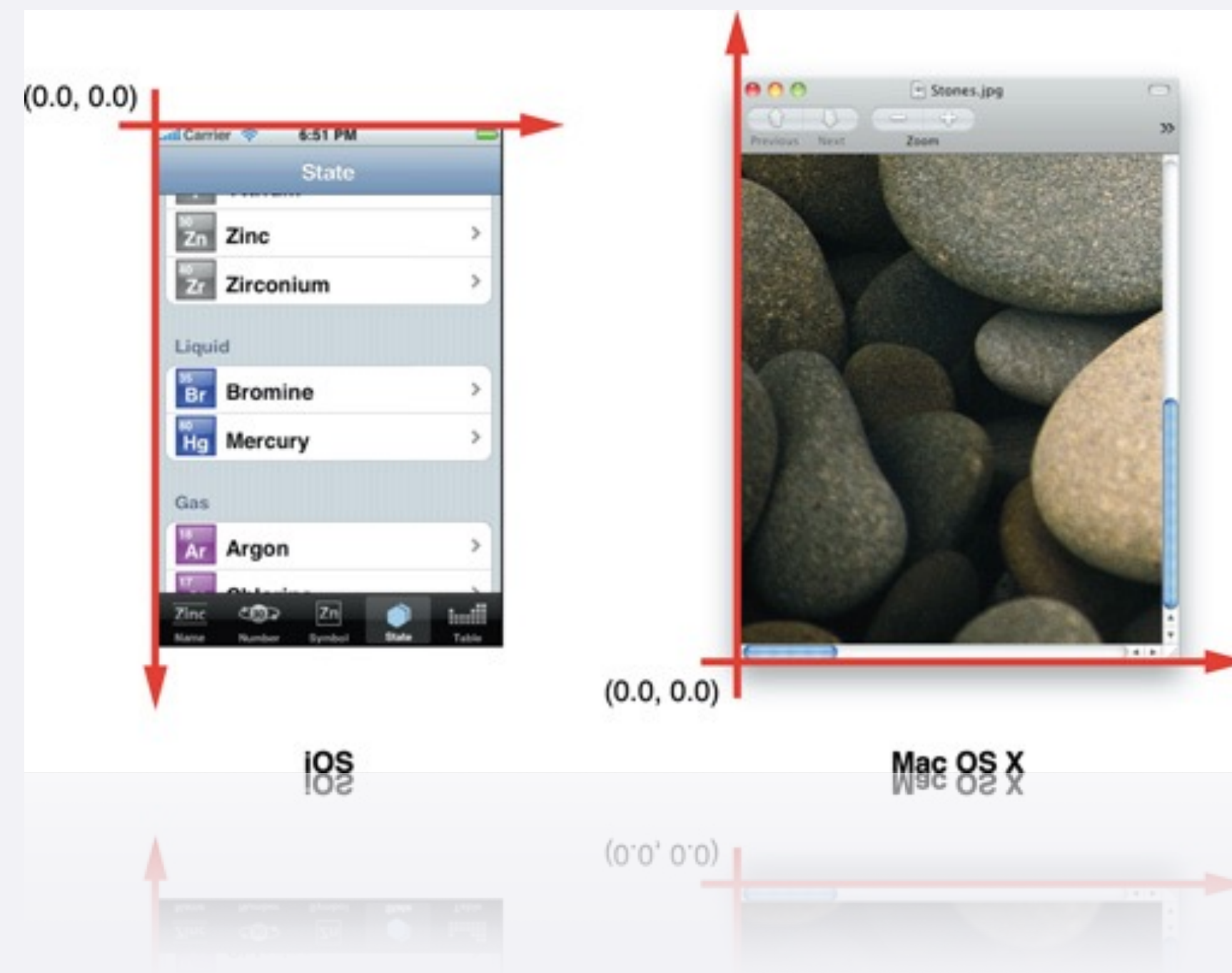
- Сердце вашего приложения
- Обрабатывает нажатия пользователей (конец Responder Chain)
- Управление локальными нотификациями (Local Notifications)
- У него есть delegate, которому сообщаются методы жизненного цикла и другие
- Регистрация на Remote Notifications (Удаленные пуш-уведомления)

UIKit: UIWindow

- Является фоном UI вашего приложения
- Передает пользовательские нажатия представлениям
- Используется в роли main Window, для отображения контента всего приложения
- Для отображения какого-то дополнительного контента
- Только 1 Window в приложении может быть ключевым (`keyWindow`)

Система координат

- Frame - координаты относительно родителя
- Bounds - координаты в своей координатной системе
- Center - центральная точка представления относительно родителя

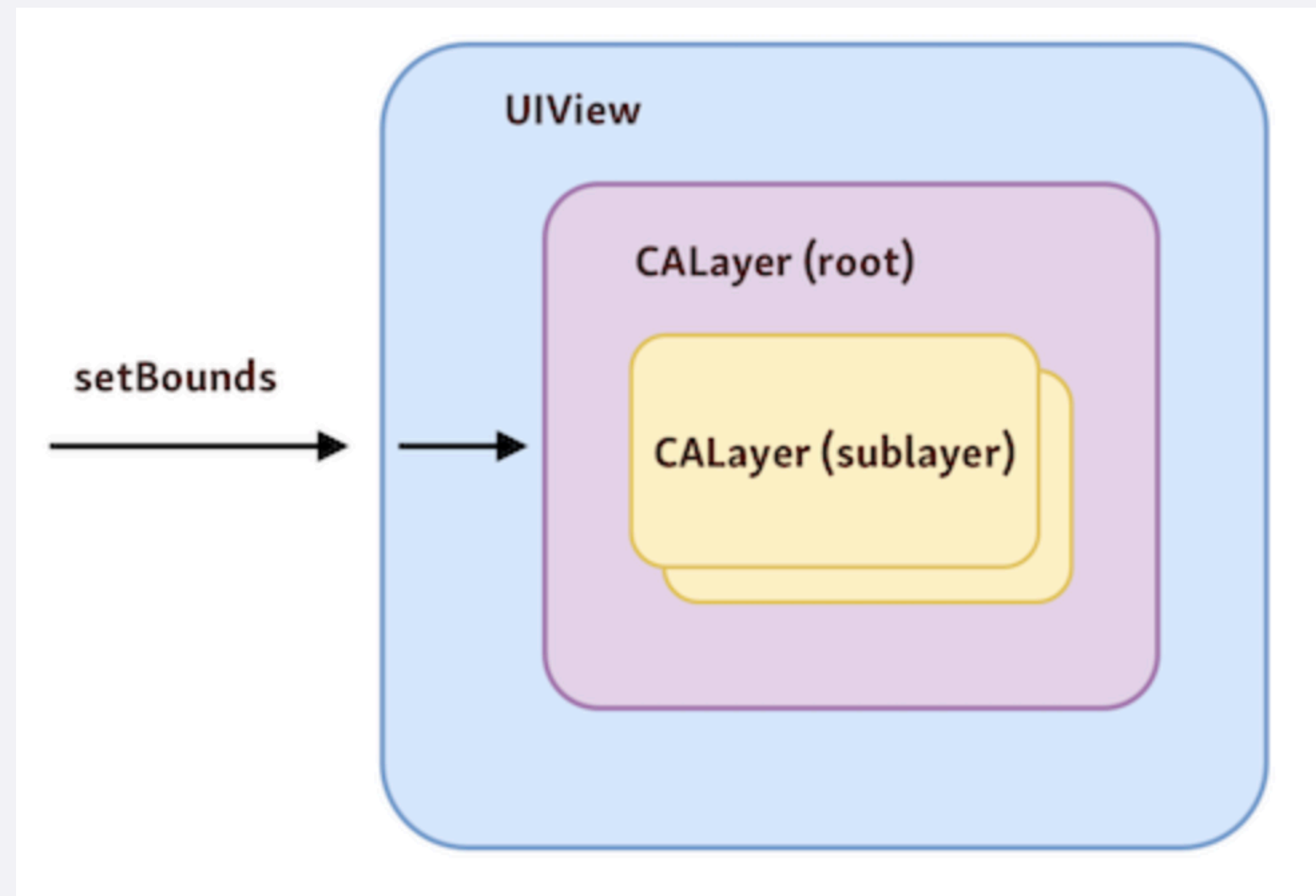


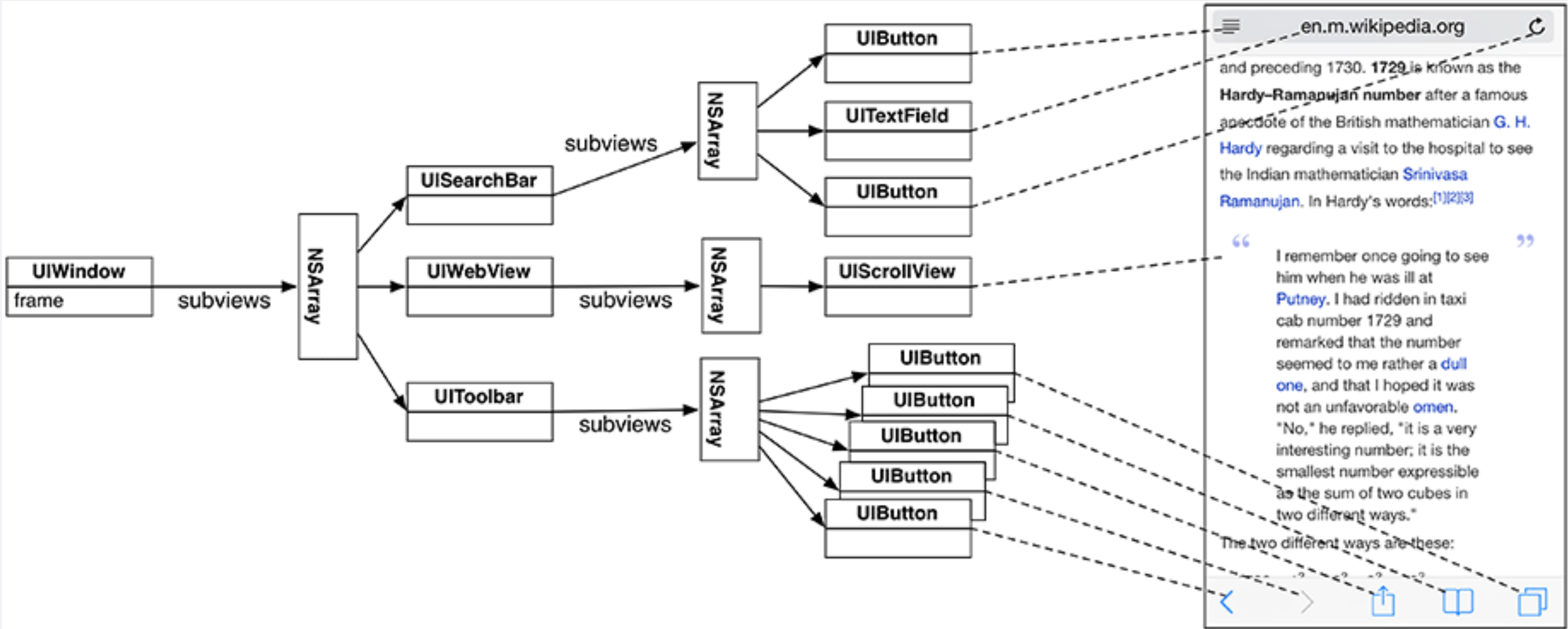
UIView

View - это представление

- Может отражаться в экземпляре окна UIWindow
- Представление находится в иерархии представлений (становится Subview) и может иметь другие представления как subview. Корень этой иерархии - окно приложения (root window)
- Представление обрабатывает события, такие как касания, так как является наследником UIResponder
- Имеет в себе иерархию слоев - CALayer

View и Layer





CALayer

Layer: особенности

- В абстракции View отвечает за отрисовку контента
- Бывают разные, например CALayer, CAGradientLayer.
- Добавление скруглений, теней
- Имеет в себе 2 слоя model, presentation

UIViewController

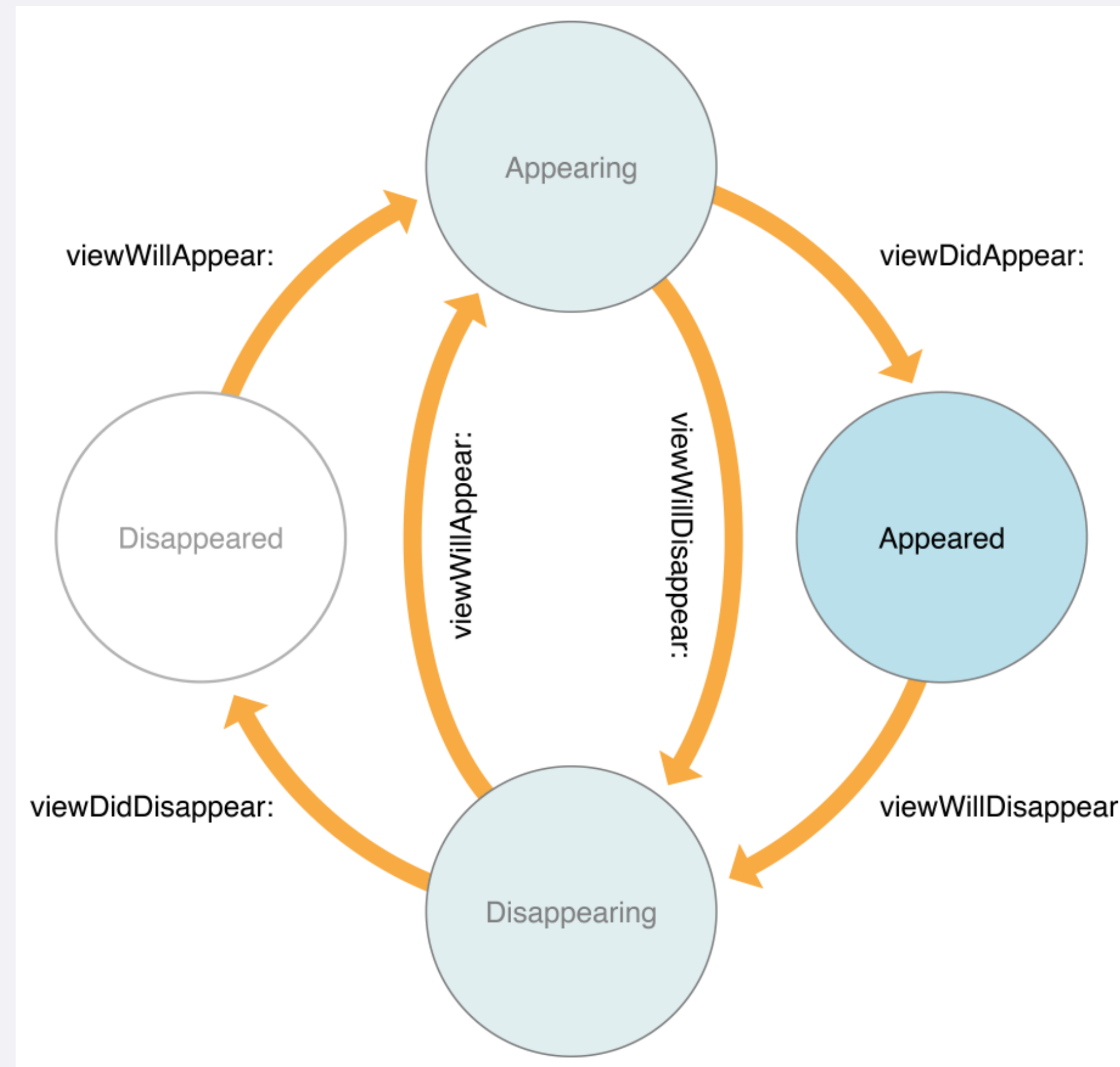
UIViewController

- `.view` - lazy свойство `ViewController`'а. Инициализация `view` происходит в методе - `loadView`
- Как только было обращение к `.view` - представление загрузится.
- Существует метод `isViewLoaded` -> `Bool`
- Может иметь модальное представление
- Знать `frame .view` можем только в методе `viewWillLayoutSubviews`

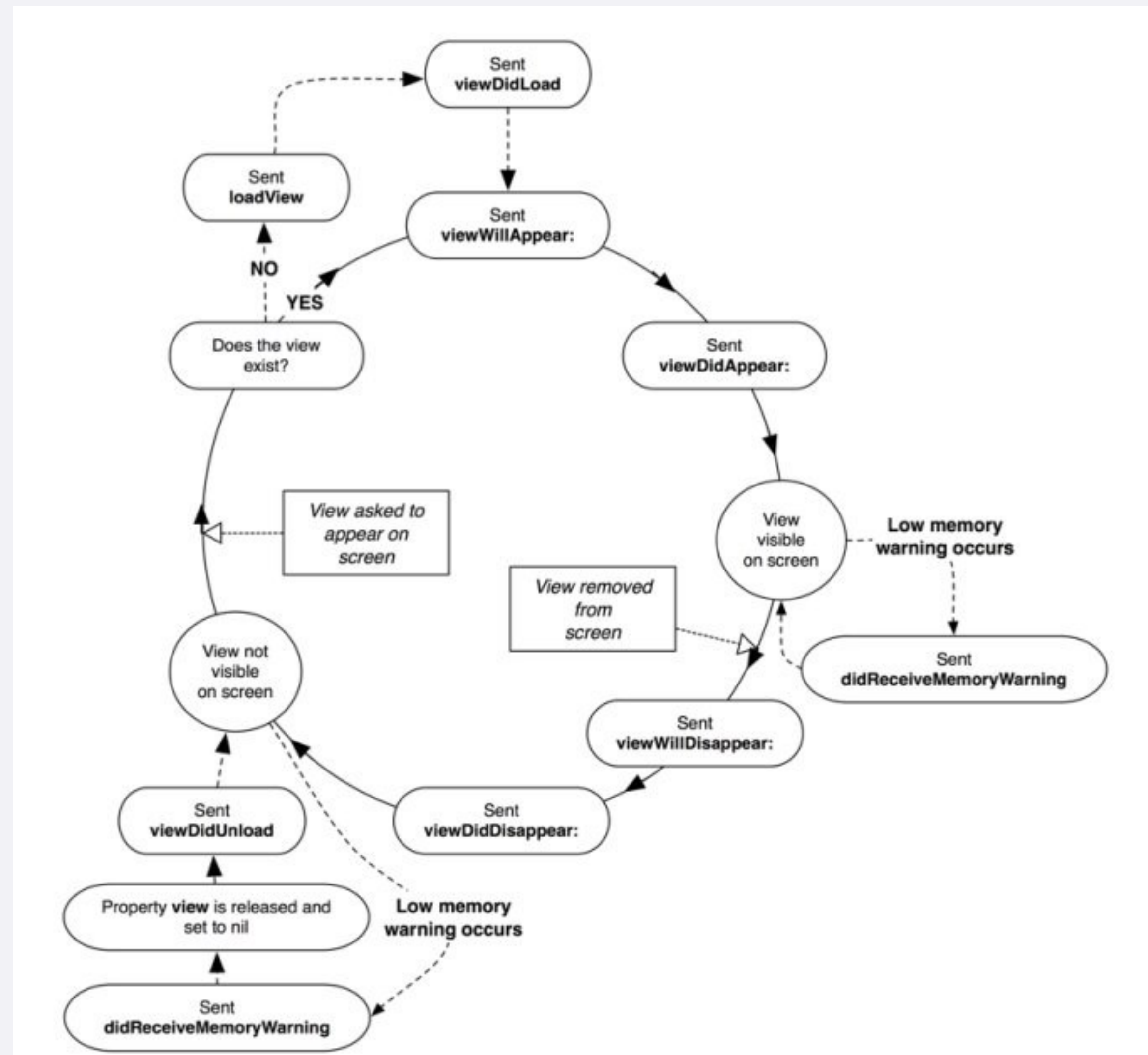
A view controller's main responsibilities include the following:

- Updating the contents of the views, usually in response to changes to the underlying data.
- Responding to user interactions with views.
- Resizing views and managing the layout of the overall interface.
- Coordinating with other objects—including other view controllers—in your app.

UIViewController: Жизненный цикл



UIViewController: Жизненный цикл (чуть подробнее)



UIViewController: Жизненный цикл

1. `init`
2. `loadView`
3. `viewDidLoad`
4. `viewWillAppear:`
5. `viewWillLayoutSubviews`
6. `viewDidLayoutSubviews`
7. `viewDidAppear:`
8. `viewWillDisappear:`
9. `viewDidDisappear:`
10. `didReceiveMemoryWarning`
11. `deinit` (dealloc)
12. `viewWillTransition(to:with:)`

UIGestureRecognizer

UIGestureRecognizer

- Логика по работе с объектами типа UITouch/ UIPress
- Может любую view «научить» распознавать жесты
- Добавляется на конкретную view

UIGestureRecognizer: Типы

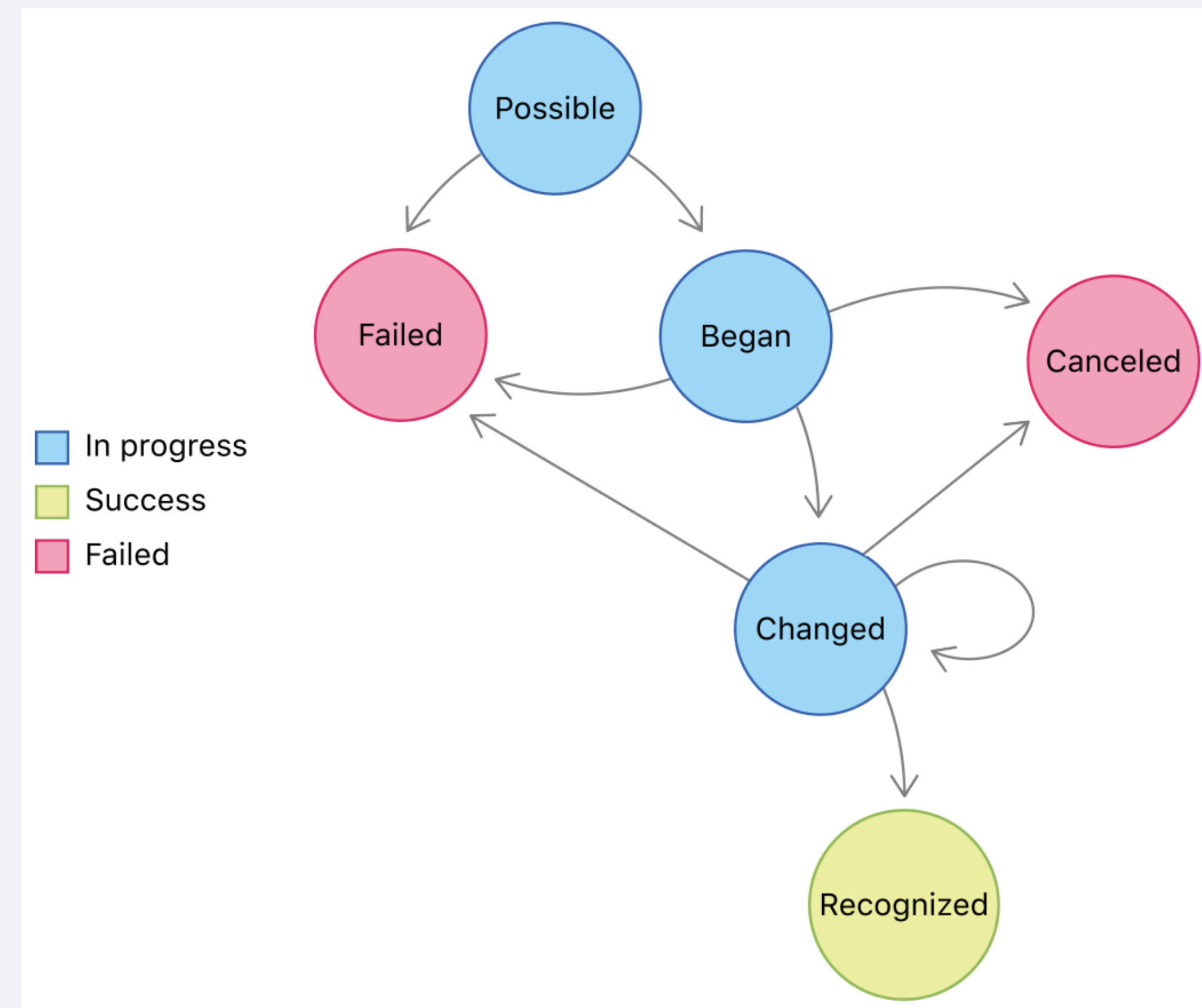
- Дискретные
- Непрерывные

UIGestureRecognizer: Виды

- UITapGestureRecognizer
- UIPinchGestureRecognizer
- UIRotationGestureRecognizer
- UISwipeGestureRecognizer
- UIPanGestureRecognizer
- UIScreenEdgePanGestureRecognizer
- UILongPressGestureRecognizer

UIGestureRecognizer: Цикл

- Possible - готов к работе
- Began - начался распознаваемый жест
- Changed - изменение состояния
- Recognized (Ended) - жест закончился
- Cancelled - отменен
- Failed - жест не был распознан

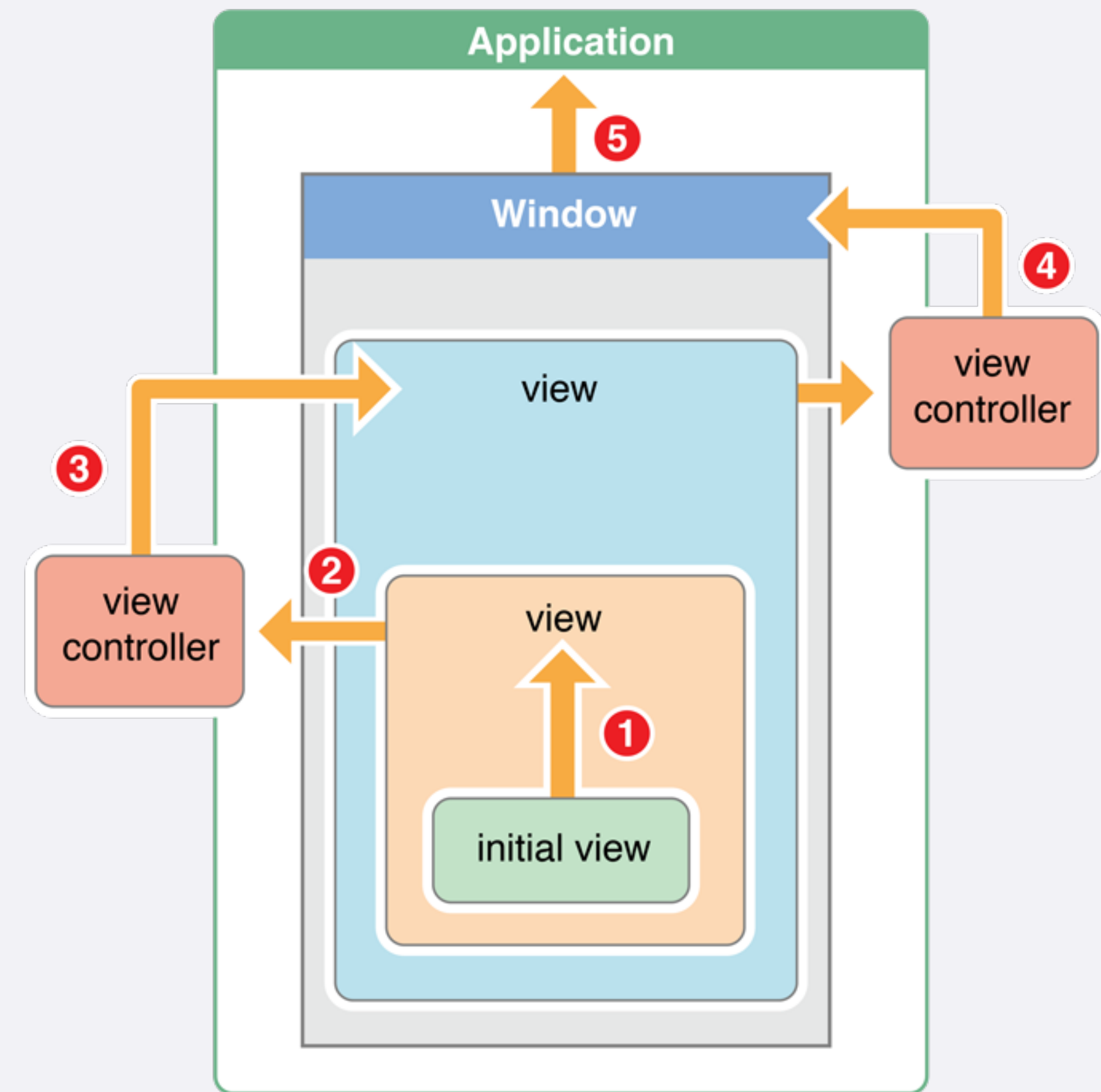
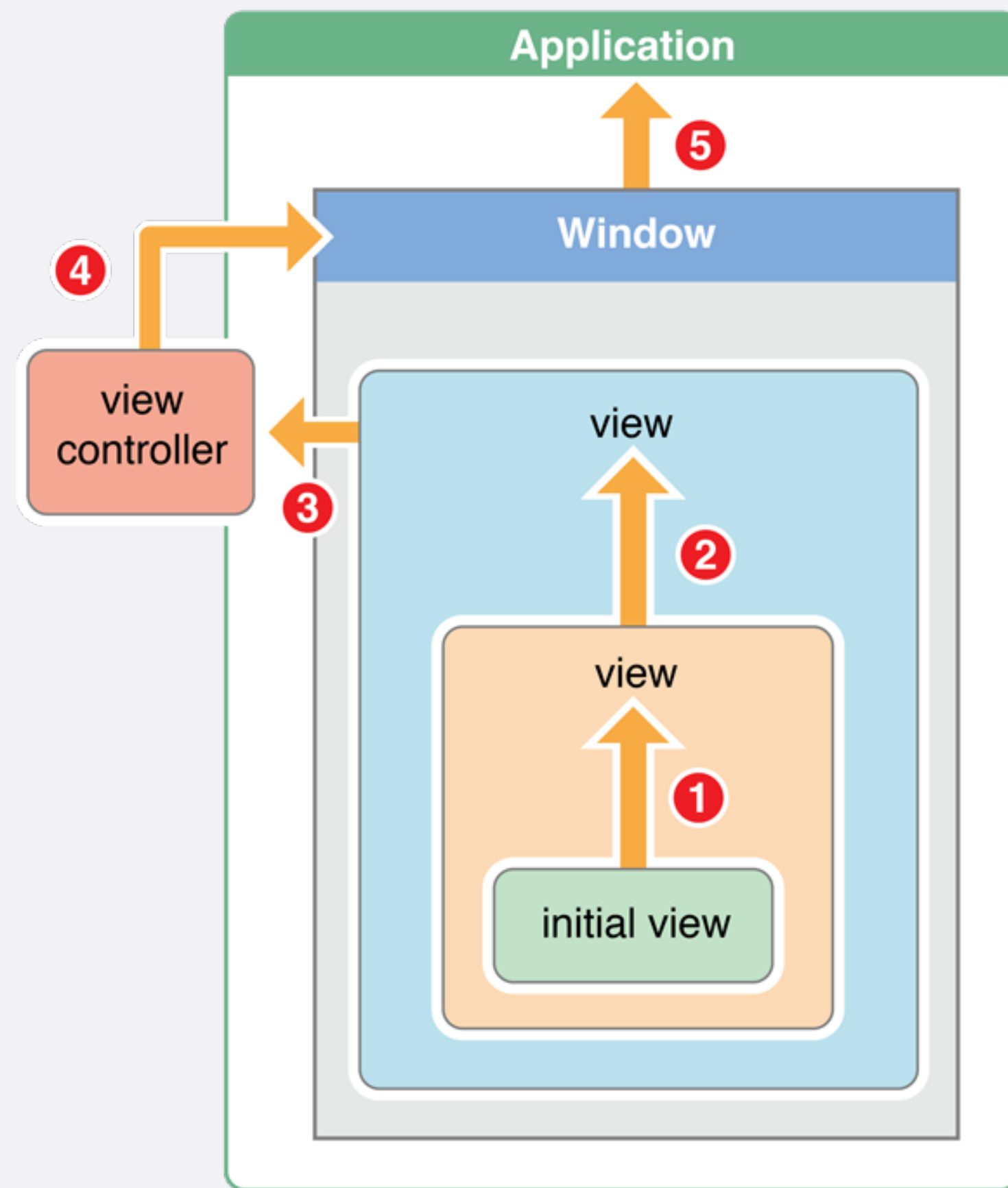


UIResponder

UIKit: Responder Chain

- Поведенческий шаблон проектирования, реализуемый в UIKit
- Все представления, наследники UIResponder, участвуют в цепочке зависимости
- Существует понятие first responder. Его можно переопределить, но обычно это View на которую был совершен tap event
- Next responder - следующее View в иерархии, то есть его родитель
- Цепочка будет продолжаться до View, ассоциируемого с ViewController'ом
- Если данный ViewController - root, то далее по цепочке будет идти UIWindow
- Next responder у UIWindow - это UIApplication
- Последним в цепочке будет AppDelegate

UIKit: Responder Chain



Задание

- Как сделать View с градиентом ? (по командам)
- Работа с firstResponder, hidden, ui -disabled
- Работа с gesture recognizer
- Д/З Сделать бублик и прокидывать нажатие на super view

Литература

- <https://medium.com/ios-os-x-development/understanding-cocoa-and-cocoa-touch-responder-chain-12fe558ebe97>
- <https://habr.com/ru/company/oleg-bunin/blog/423299/>
- <https://developer.apple.com/documentation/uikit>