FROM RXSWIFT PERSPECTIVE

WELCOME TO RX WORLD (ELEMENTRY)

INTRODUCTION

RXSWIFT & RXCOCOA

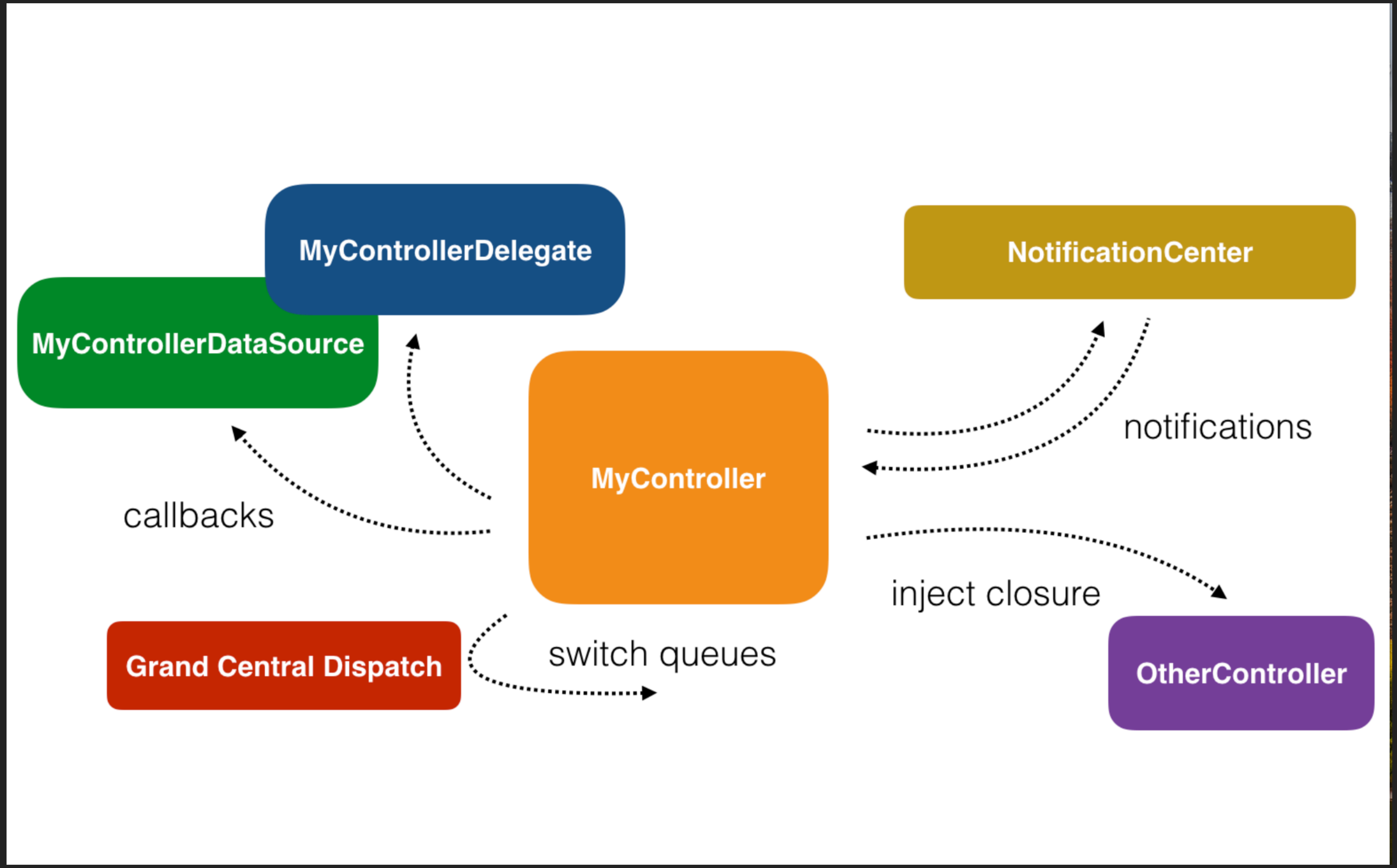
CONTENT

RXSWIFT

RX: REACTIVE EXTENTIONS

HOW TO COMMUNICATE AMONG ASYNC FUNCTIONS

- Closure: 🍤[closurehell.swift]
  - closure hell
  - force caller format
- Delegate🍤[delegate.swift]
  - interaction
  - continue to return back data
- NotificationCenter
  - so weak connection
  - so, we have to keep notification name and selector same.



Is there any better way to coordinate Async functions?

3 CONCEPTS

- ✨Observable, Subject
- ✨Operators
  - filter
  - transform
  - combine
- Scheduler
  - subscribeOn
  - observeOn

OBSERVABLE

- Key words: Cold, Sequence, Status, Dispose
- Methods: of…, create
- Traits: why need them?

🍤[observable]

There is a trouble…

We can NOT modify Observale after defined.

SO… SUBJECT 🍤

- Types: 4 types(Pub, Beh, Rep, Variable).
  - Variable:
    - only 2 methods: value(change), asObservable(subscribe);
  - Pub, Beh, Rep =>
- ✨Snapshot
- ✨Private subject(∵onNext), expose observable(∵ONLY subscribe).

OPERATORS

- Types:
  - Filter 🍤[filter.swift]: filter, throttle, take(_:schedular) …
  - Transform 🍤[tranform.swift]: map, flatMap
  - Combine Video
- ✨Life cycle

RXCOCOA

STEPING INTO UI…

DEAD UI: LABLE

ACTIVE UI: SWITCH BUTTON

ABOUT RXCOCOA

- Active
  - ControlProperty<Bool>: UISwitch+Rx.swift [✨Observable]
  - ControlEvent
- Dead
  - Binder<String?>: UILable+Rx.swift [✨Observer]

BINDER

- UILable+Rx.swift
  - [✨Observer]
  - Binding with underlying logic (show information)
  - Can’t bind error

TRAITS

- Attributes
  - can’t error out;
  - observe, subscribe on main schedular;
- What 📝
  - ControlProperty, ControlEvent
  - Drive

IN PRACTICE

Functional & MVVM

FUNCTIONAL

CONCAT & FLATMAP & RETRYWHEN

CONCAT: HANPPEN IN LINE

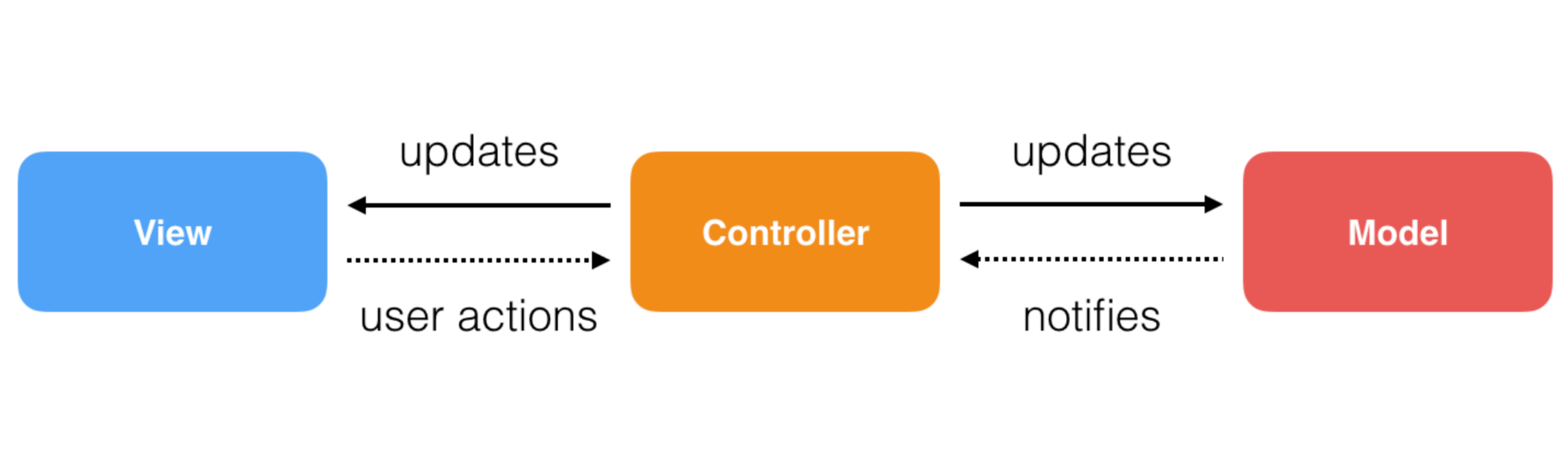
FLATMAP: PRODUCER - CONSUMER

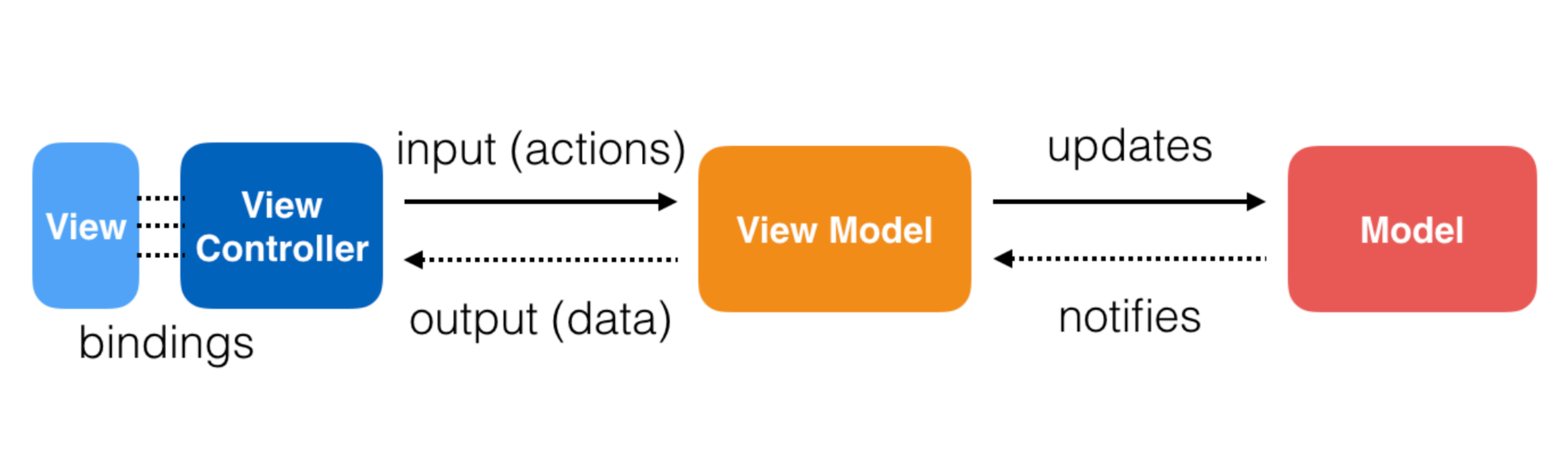
RETRYWHEN: RETRY LOGIC IS BELONGED TO USER

MVVM

MVC = MASSIVE VIEW CONTROLLER

A Joke


updates
updates
View
Controller
Model
user actions
notifies


input (actions)
updates
View
View Controller
View Model
Model
bindings
output (data)
notifies

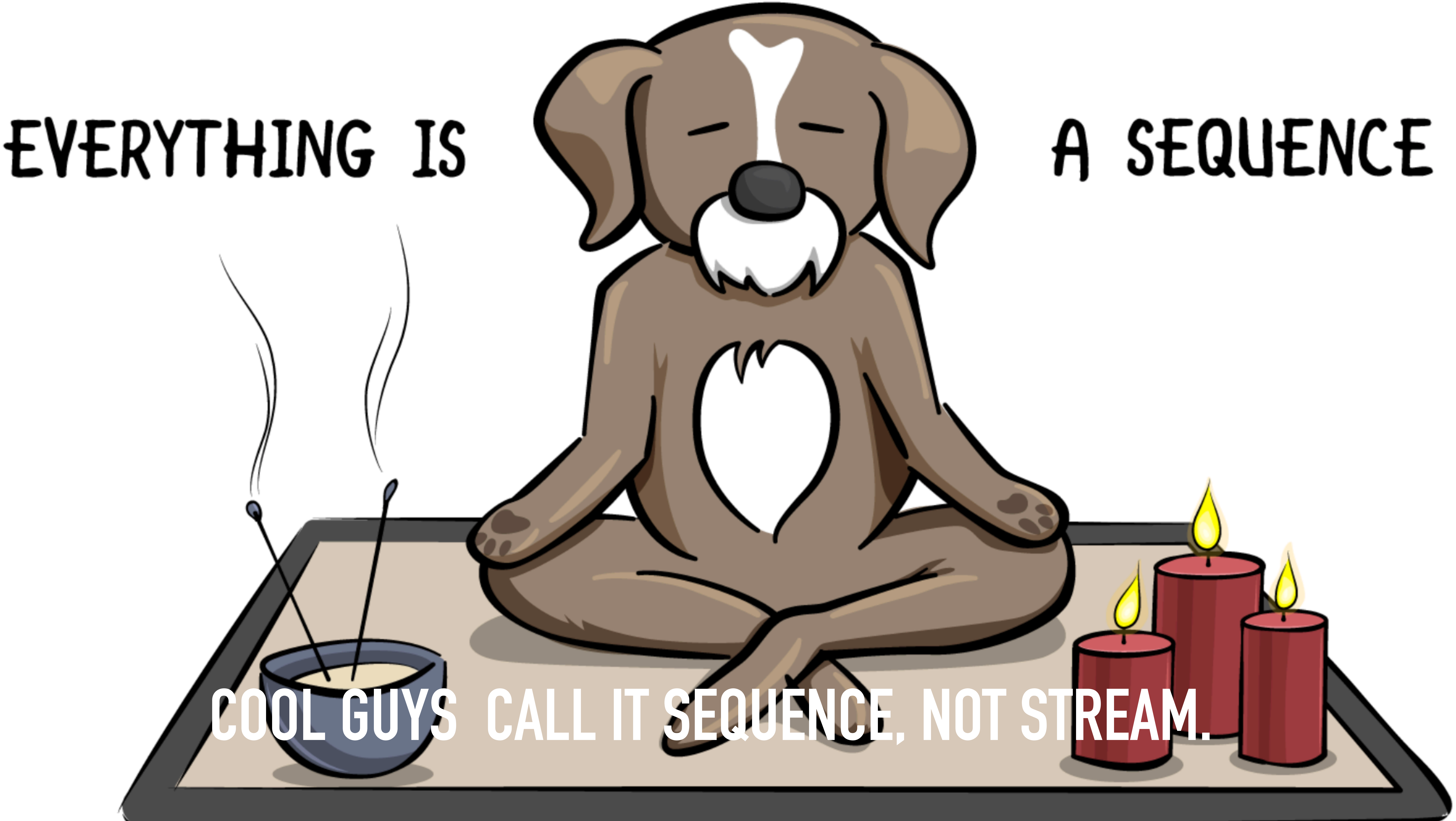
CONCLUDE

TODO:

1. USE MVVM(TEST)
2. SCHEDULER – THREAD(DEBUG)
3. CUSTOM DATASOURCE
4. COORDINATE NAVIGATION…

SUMMERIZE

- Notion about Rx: logo, name, origin
- RxSwift
  - ✨Observable(Subject) ✨Operators ✨Scheduler
- RxCocoa
  - ✨ControlProperty ✨Binder ✨Drive
- IN PRACTICE
  - ✨Functional ✨MVVM



COOL GUYS CALL IT SEQUENCE, NOT STREAM.

REFERENCES

REFERENCES

REFERENCES

- Book:
  RxSwif Reactive Programming with Swift (second)

- App:
  RxMarbles

Up to date for iOS 11, Xcode 9 & RxSwift 4
RxSwift
Reactive Programming with Swift
SECOND EDITION
By the raywenderlich.com Tutorial Team
Florent Pillet, Junior Bontognali, Marin Todorov & Scott Gardner