

FROM RXSWIFT PERSPECTIVE

---

**WELCOME TO RX WORLD  
(ELEMENTARY)**



---

# INTRODUCTION



RXSWIFT & RXCOCO A

---

**CONTENT**





# RXSWIFT

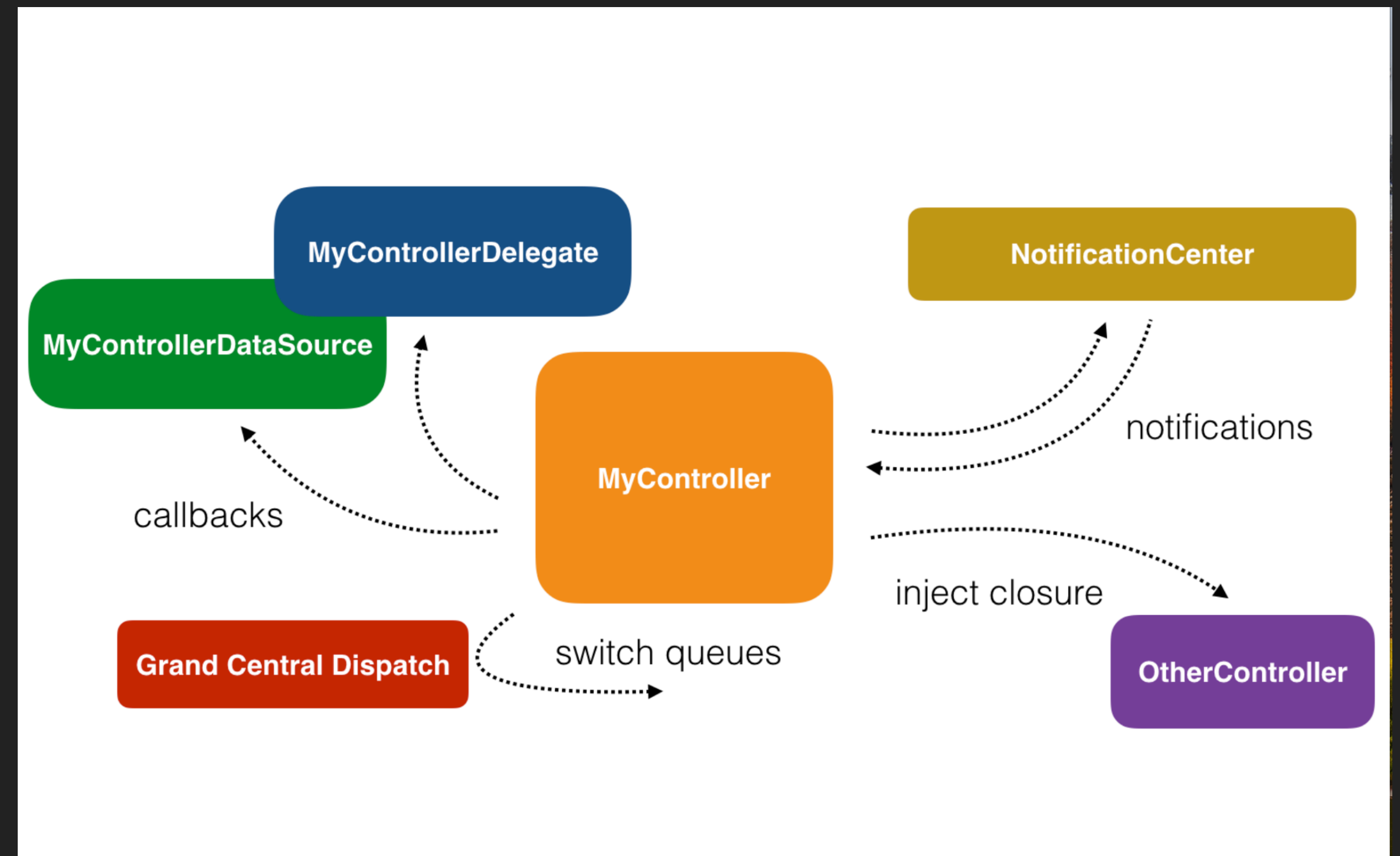
## RX: REACTIVE EXTENTION

### How to compose functions



## HOW TO COMMUNICATE AMONG ASYNC FUNCTIONS

- ▶ Closure: 🍷[closurehell.swift]
  - ▶ closure hell
  - ▶ force caller format
- ▶ Delegate
  - ▶ 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 Observable 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, throttle, take(\_:scheduler) ...

- ▶ Transform 🍤: map, flatMap

- ▶ Combine Video

- ▶ ✨Life cycle



A black and white photograph of three wind turbines. The largest turbine is in the foreground on the left, with its three blades extending upwards. Two smaller turbines are visible in the background, one slightly to the right and one further back. The sky is filled with large, dramatic clouds.

# RXCOCOA

# STEPINGINTO UI...



## ABOUT RXCOCOA

- ▶ `ControlProperty<Bool>: UISwitch+Rx.swift` [ `Observable`]
- ▶ `Binder<String?>: UILabel+Rx.swift` [ `Observer`]

# BINDER

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



# TRAITS

- ▶ Attributes

- ▶ can't error out;
  - ▶ observe, subscribe on main scheduler;

- ▶ What



- ▶ ControlProperty, ControlEvent
  - ▶ Drive



# IN PRACTICE

## Functional & MVVM

# FUNCTIONAL

CONCAT & FLATMAP & RETRYWHEN

**CONCAT: HANPPEN IN LINE**

**FLATMAP: PRODUCER – CONSUMER**

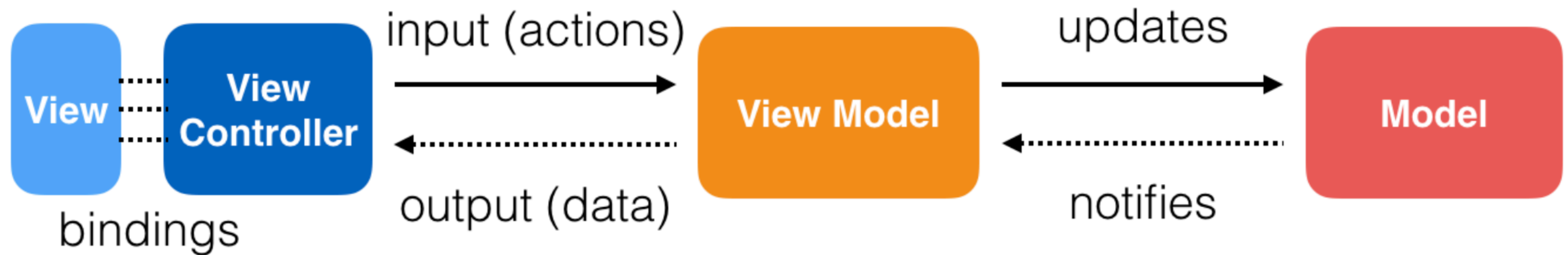
**RETRYWHEN: RETRY LOGIC IS BELONGED TO USER**

**MVVM**



**MVC = MASSIVE VIEW CONTROLLER**

**A Joke**





---

# CONCLUDE

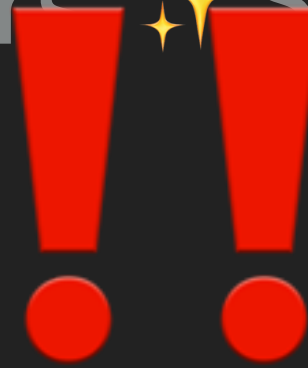


# TODO:

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





EVERYTHING IS

A SEQUENCE



COOL GUYS CALL IT SEQUENCE, NOT STREAM.



---

# REFERENCES