Welcome to Rx World (Elementry)

From RxSwift Perspective

Linda Zhong - June 5, 2018

electric eel

1. Introduction

1. Who knows this animal? And why choose it as logo?

2. What does Rx stand for? History.

2. Content

1. Reflect on existing program style

1. Define functions, call them. However, in real world, there are a lot of relationships among functions. Most normal is call order.
   🐠 reflectOnReal()

2. Core concepts

1. **Observable - Carry immutable snapshots of data.**

   ### 1. LIFE CYCLE

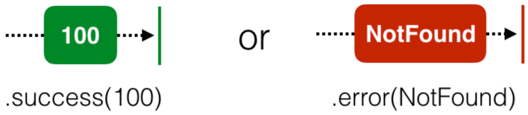
   1. **Activate**: **Cold** signal - Trigger by subscription; <=> Dispose. (**DisposableBag** is really vivid)

   2. **Sequence**: next events

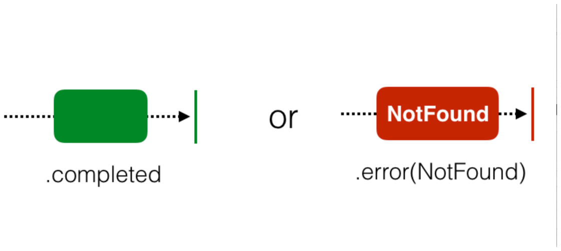
   3. **Dead**: completed, error events(❓ Thinking, how to handle error)

   ### 2. TRAIT

   1. **Single**: like return some value



   2. **Completable**: like bool



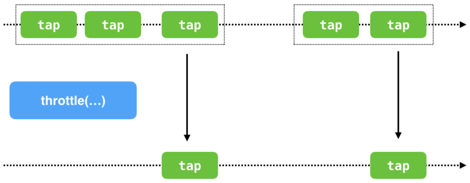
   ### 3. SUBJECT: OBSERVABLE IS STATIC. WE NEED SOME OBJECT CAN EMIT EVENT AT RUNTIME.

   1. **Observer role**: subject.onNext(xx)

2. **Observable role**: asObservable.subcribe(xx)

3. **4 Types**: PublishSubject, BehaviorSubject, ReplaySubject, **Variable(replay last snapshot)**
4. **Private Subject, expose Observable**

2. Operators(compose sequences)

Category	Operator	Function	Scenario
Filter	filter🎃		
	skip🎃(skipWhile, skipUntil)		skip(1)
	distinctUntilChanged🎃		text field input
	take(takeWhile, takeUntil) take(_:scheduler:)🎃		time out
	throttle🎃	 tap tap tap tap tap throttle(...) tap tap	1. text field input 2. navigation bar tap
	ignoreElements	-	Focus on terminated status. e.g. Care about connection status
Transform	map		
Combine			

3. Scheduler (producer-consumer pattern)

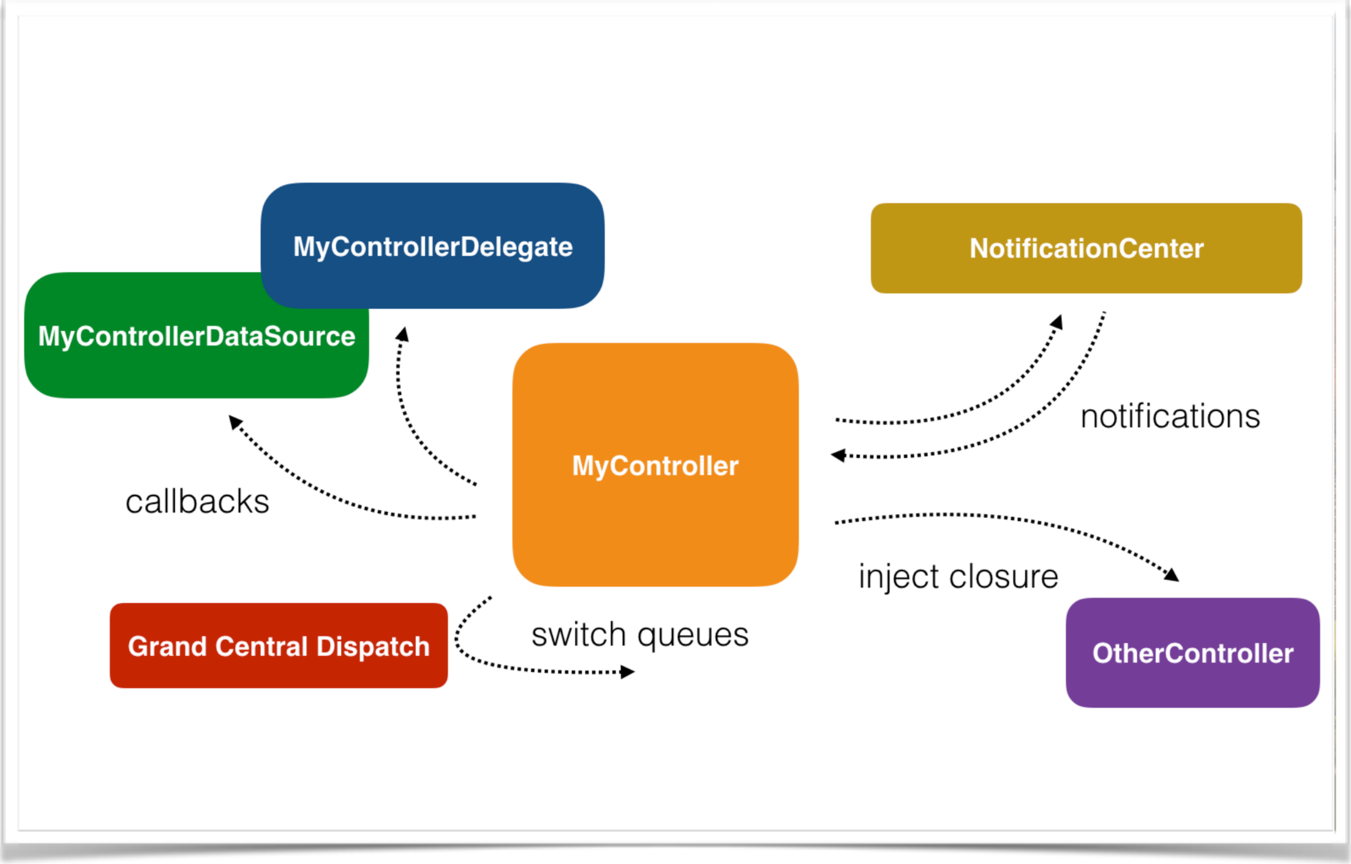
1. SUBSCRIBEON

2. OBSERVEON

3. PITFALLS

3. In practice

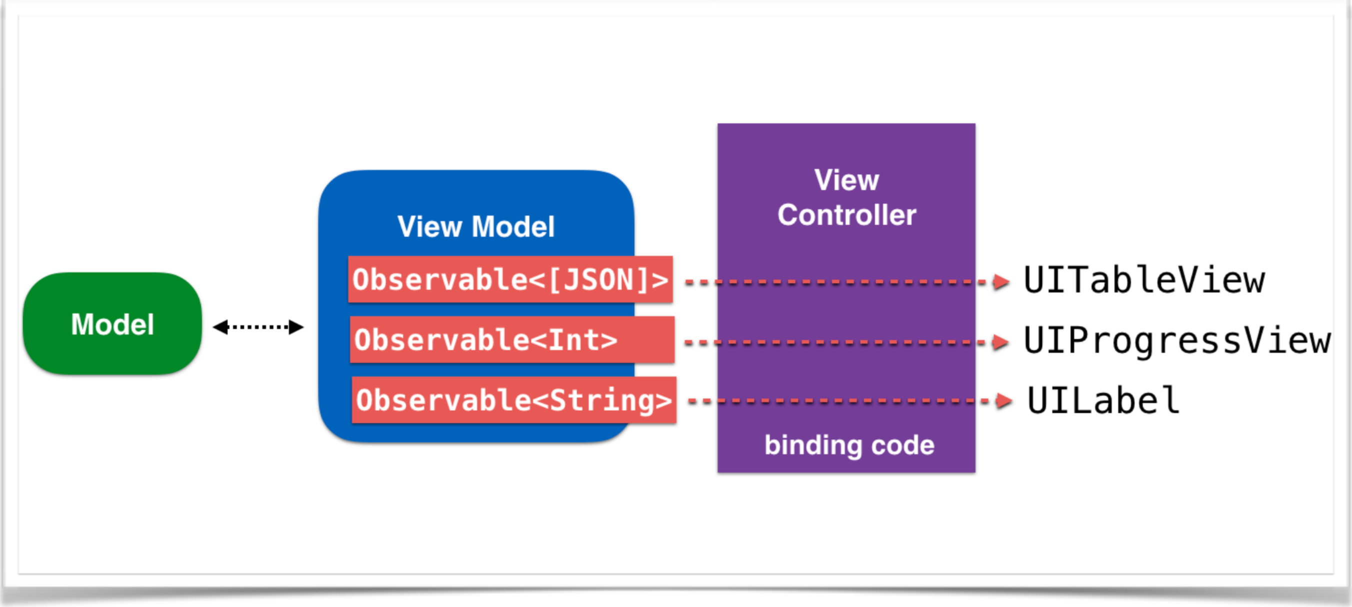
1. Before stepping into Rx World, Let’s have a glance of what the Cocoa world has supported to us?



2. Functional. Because it’s easy. e.g. **retryWhen**, **flatMap**
    1. closure hell

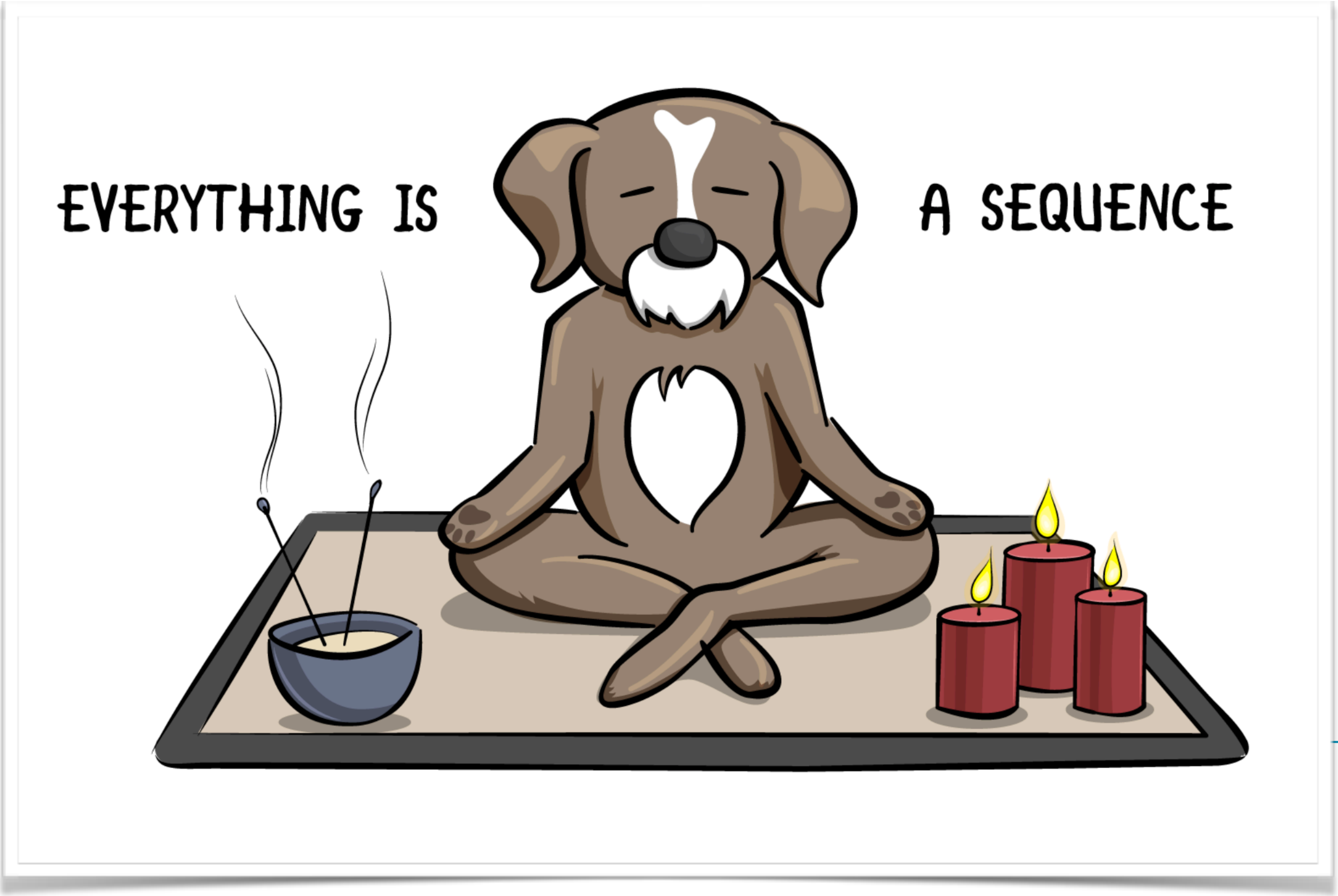
    2. so many ways in codes:

3. MVVM
    1. Data => UI(binding)
    2. UI => Function
4. It’s like a breeze, also be cautious of the effects Rx will play on your project.



3. Concluding

1. Summerize
2. Strengthen “Sequence” - In RxSwift world, cool guys call it sequence, not stream.



4. References:

1. **Book: RxSwift - Reactive Programming with Swift (Second Edition)**

2. **Video: https://www.youtube.com/watch?v=W3zGx4TUaCE&list=PLk4l3klt5ZzRMSbfavQAf65d0SqyAG7D4&index=2 by Marin**

3. **URL: https://leon_lizi.gitbooks.io/rx-/content/chapter1.html Explain what flatMap isApp: RxMarbles An interesting application showing Rx sequence.**