

函数响应式编程之**ReactiveCocoa**框架

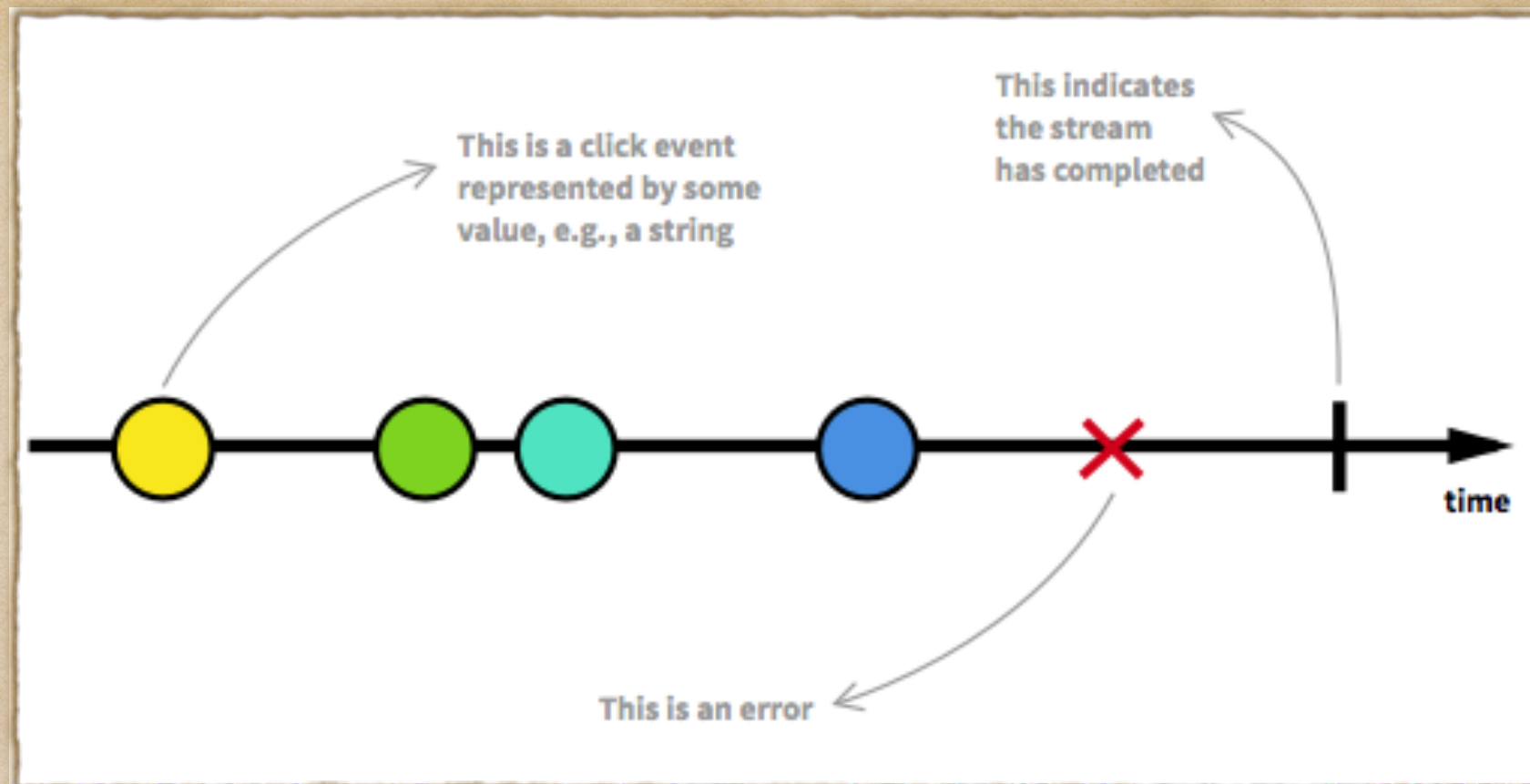
The ReactiveCocoa
Framework of
Functional Reactive Programming

姓名：郭志强

学号：21651160

专业：移动互联网和游戏开发技术

- ReactiveCocoa(RAC)是一个Objective-C的框架,由Github开源的一个应用于iOS和OS开发的新框架, 它的灵感来自函数响应式编程 (FRP)
- 它定义了统一的接口来处理异步事件, 传统的**delegate**方法、**block**回调、**target-action**机制、**notification**和**KVO**都可以用RAC提供的**signal**的方式处理
- 比起传统处理事件的方式, RAC更简洁易懂, 逻辑可以用清晰的管道、流式语法表示



理解RAC的关键-理解RP

抛弃命令式且带状态的编程

RP是一种编程范式，与异步数据流进行交互

几乎任何事物

创建

数据流

过滤
组合

新数据流

RP提高了代码的抽象层级

只需要关注业务逻辑中那些相互依赖的事件

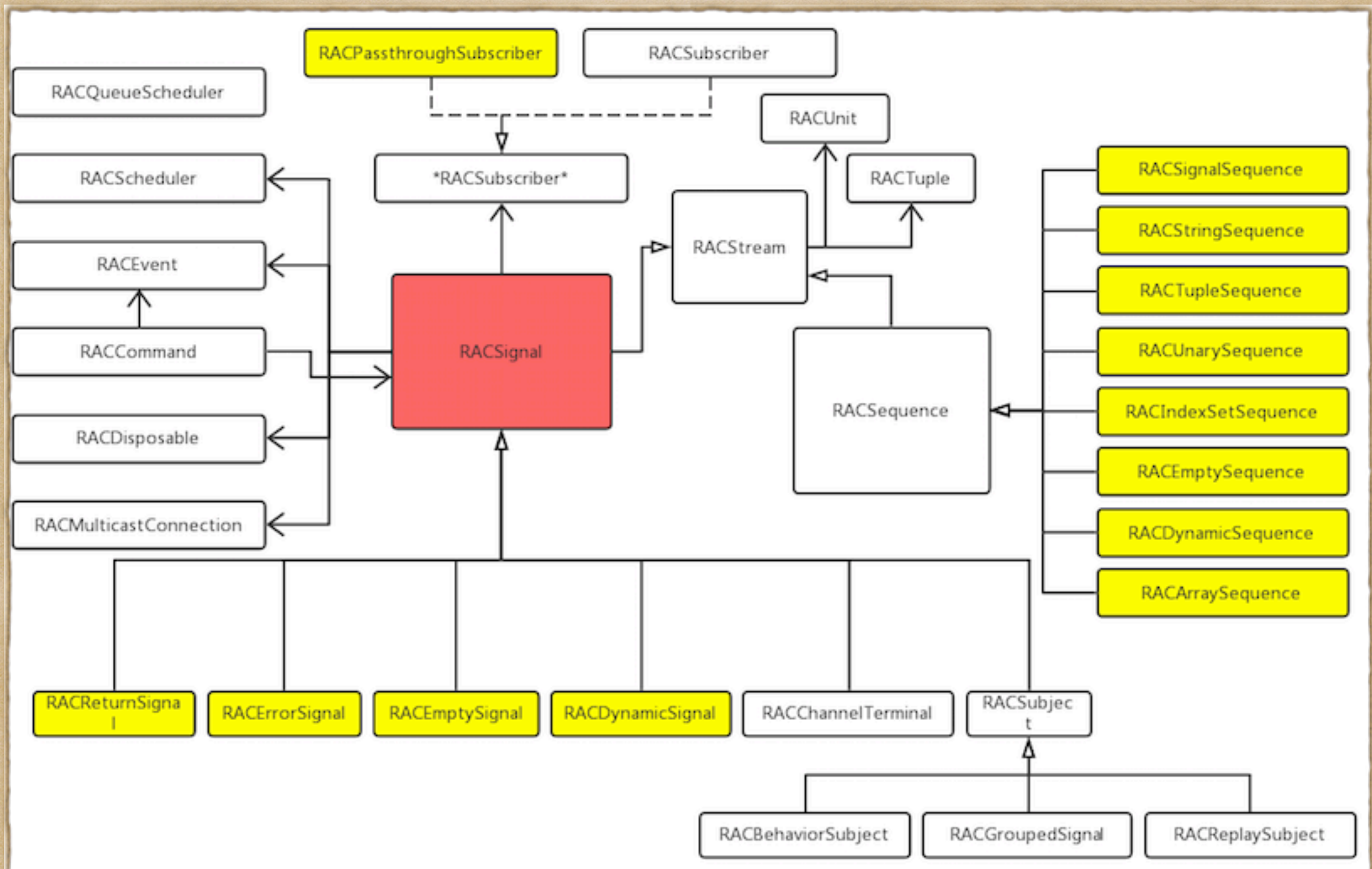
不用纠结于大量的实现细节

RP的代码逻辑更加的清晰、更容易理解

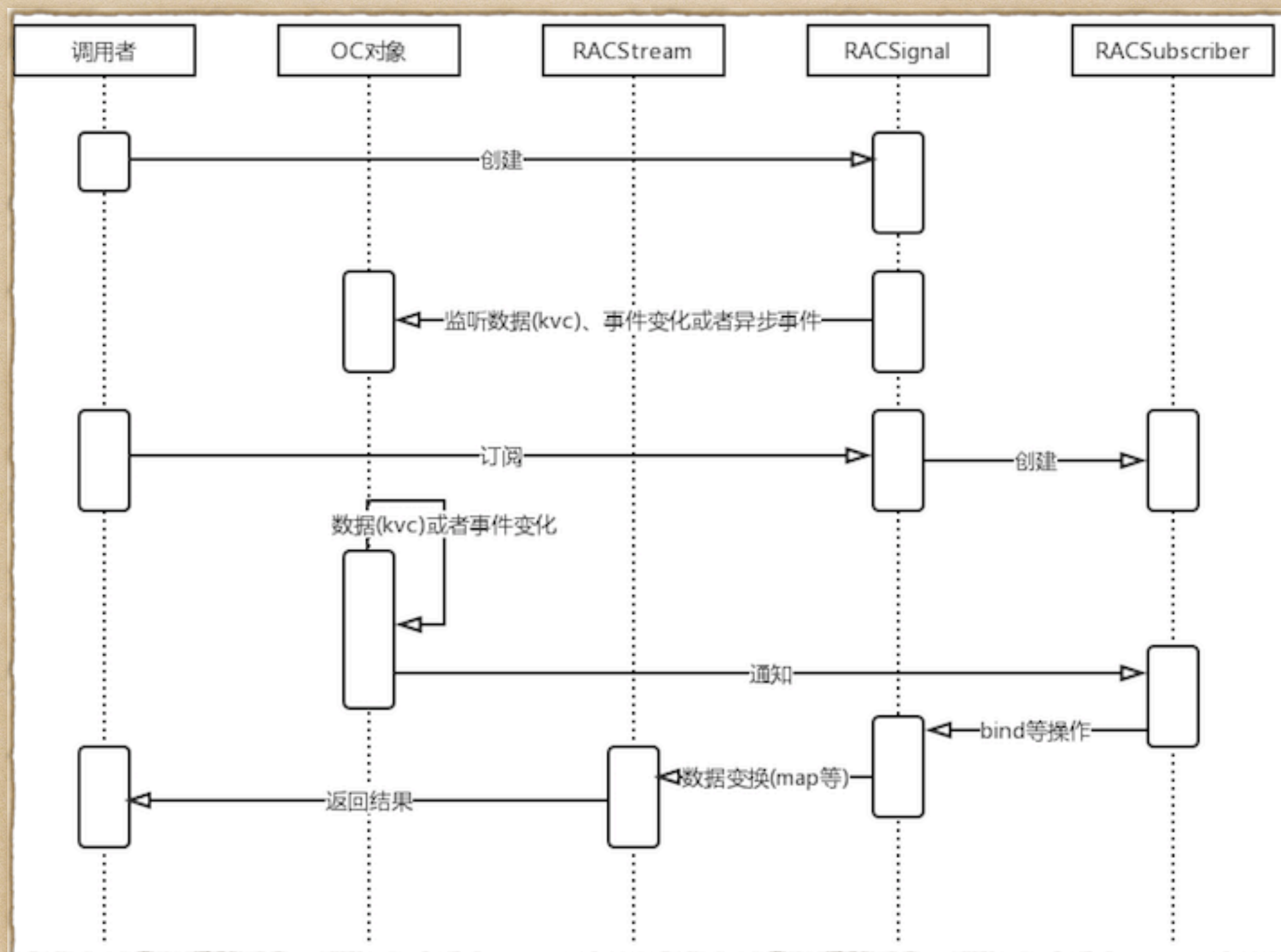
RAC同时也结合了函数式编程思想

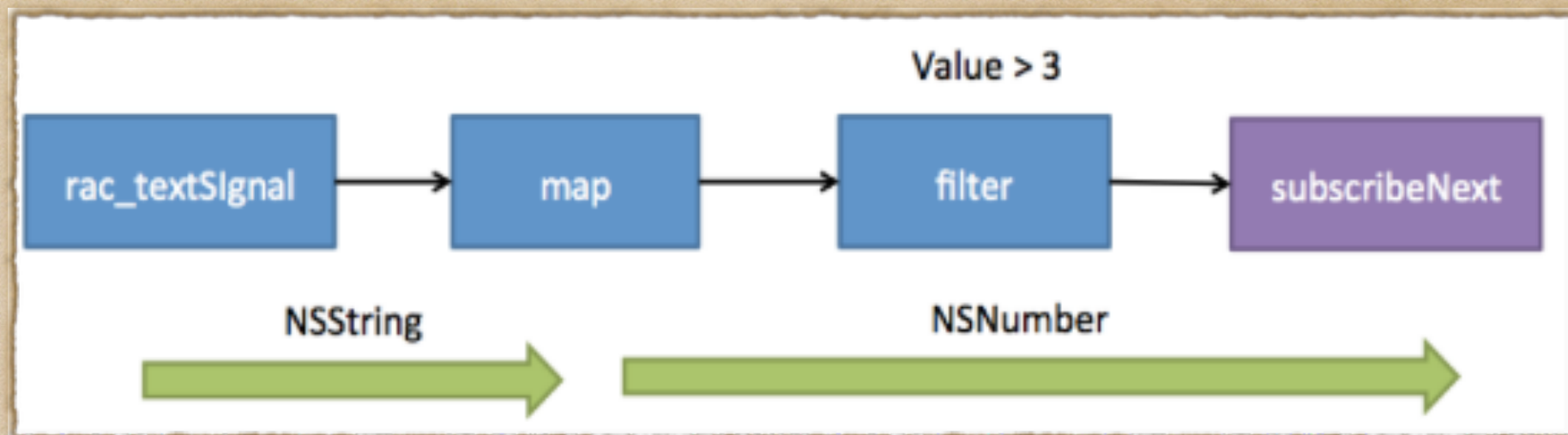
在RP的基础上利用函数、高阶函数进行开发

RAC所有类关系图



RAC工作原理





```
[[[self.usernameTextField.rac_textSignal  
  map:^id(NSString*text){  
    return @(text.length);  
  }]  
  filter:^BOOL(NSNumber*length){  
    return[length integerValue] > 3;  
  }]  
  subscribeNext:^(id x){  
    NSLog(@"%@@", x);  
  }];
```

map操作将文本内容转换为**NSNumber**对象

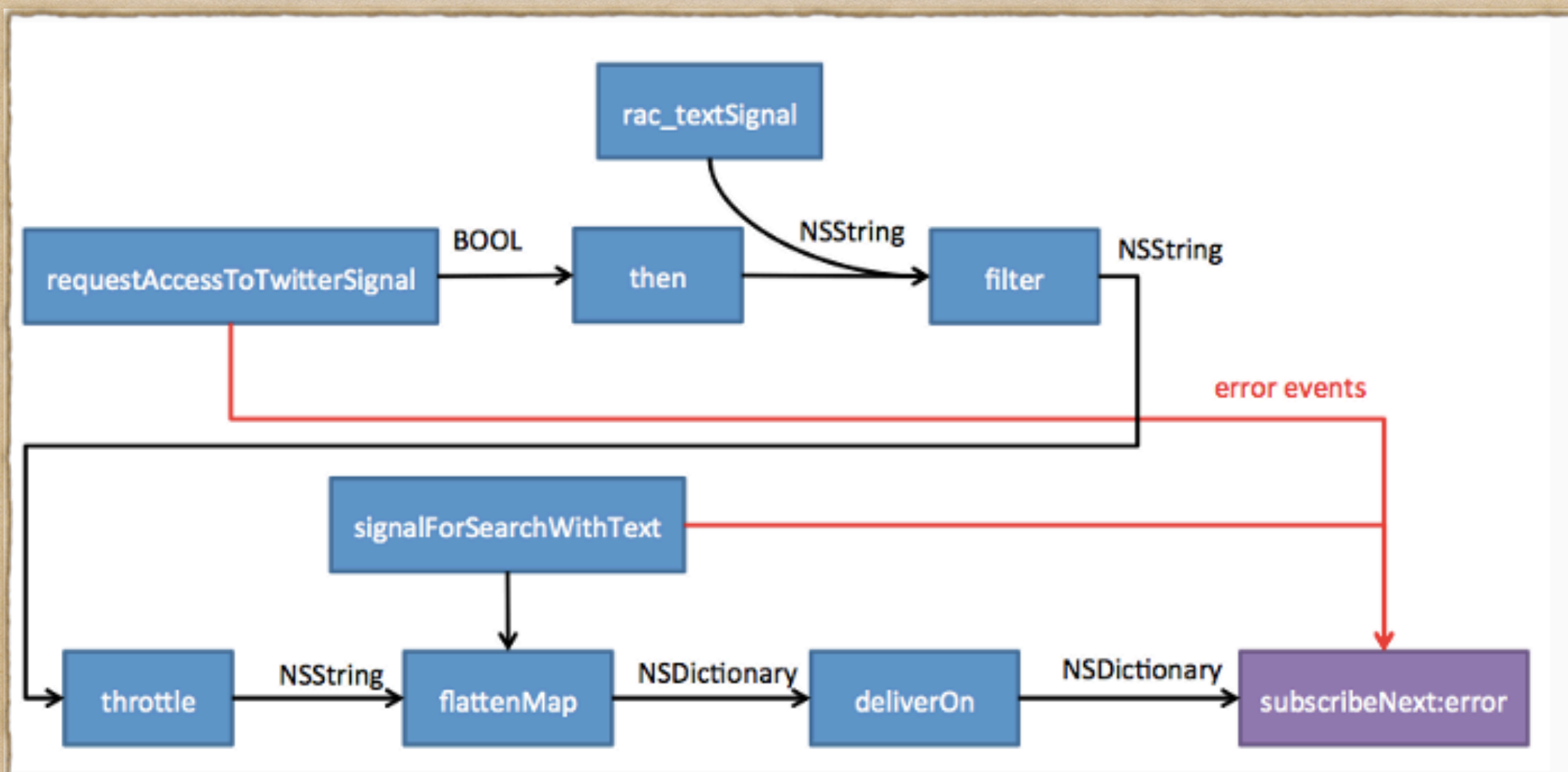
filter保留长度大于3的内容

输入关键字获取用户数据并展示的流程图

代码结构跟这个管道图基本一样

RAC能完成从业务逻辑到代码的快速转换

不用过多关注状态变量和实现细节



Search Text:

kakar

**@kuriea1738**

@zuu_Haul @Vida_kakar i FEEL IT 🥺🥺🥺🥺❤️❤️❤️❤️❤️❤️❤️

**@HM_Kakar**

4 Afghan police officers were killed in a suspected insider attack in s...

**@NaeBaatNews**

KAKAR&gt;

**@SaluhallenSlakt**

Lördagslunch i Saluhallen! Vi har kokat på en Sås Bolognese av Stor...

**@BoririYouth**

Kakar:

**@Thar_Tez_News**

#Kakar

**@MediaOfSindh**

Kakar:

**@TeZz_Tv_News**

Kakar:

**@CityUpdates021**

Kakar:

**@Sindhu_News**

Kakar:

**@Sindhu_plus**

"kakar"

kakaro

kakarotto



RAC主要解决了3个问题

传统iOS开发过程中，状态以及状态之间依赖过多

比如最常见的输入账号、密码时登陆按钮的enable状态问题

试图解决MVC框架的问题

RAC的信号机制很容易将model变量的变化与view关联，简化了

统一消息传递机制

RAC对UI控件事件进行了封装，KVO、target-action等都可以用RAC完成

谢谢！