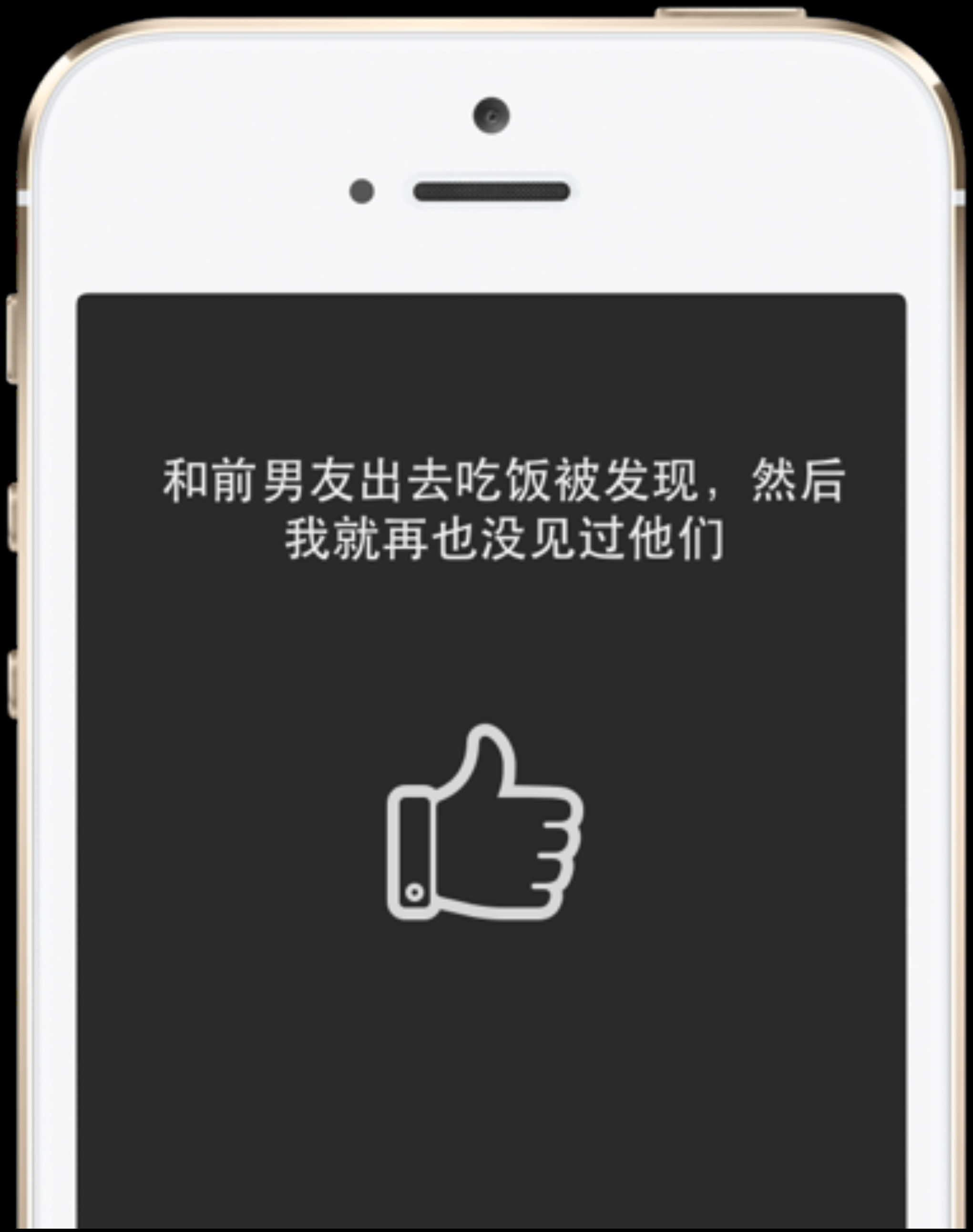
RAC在花瓣客户端的实践

李忠@花瓣

OverView

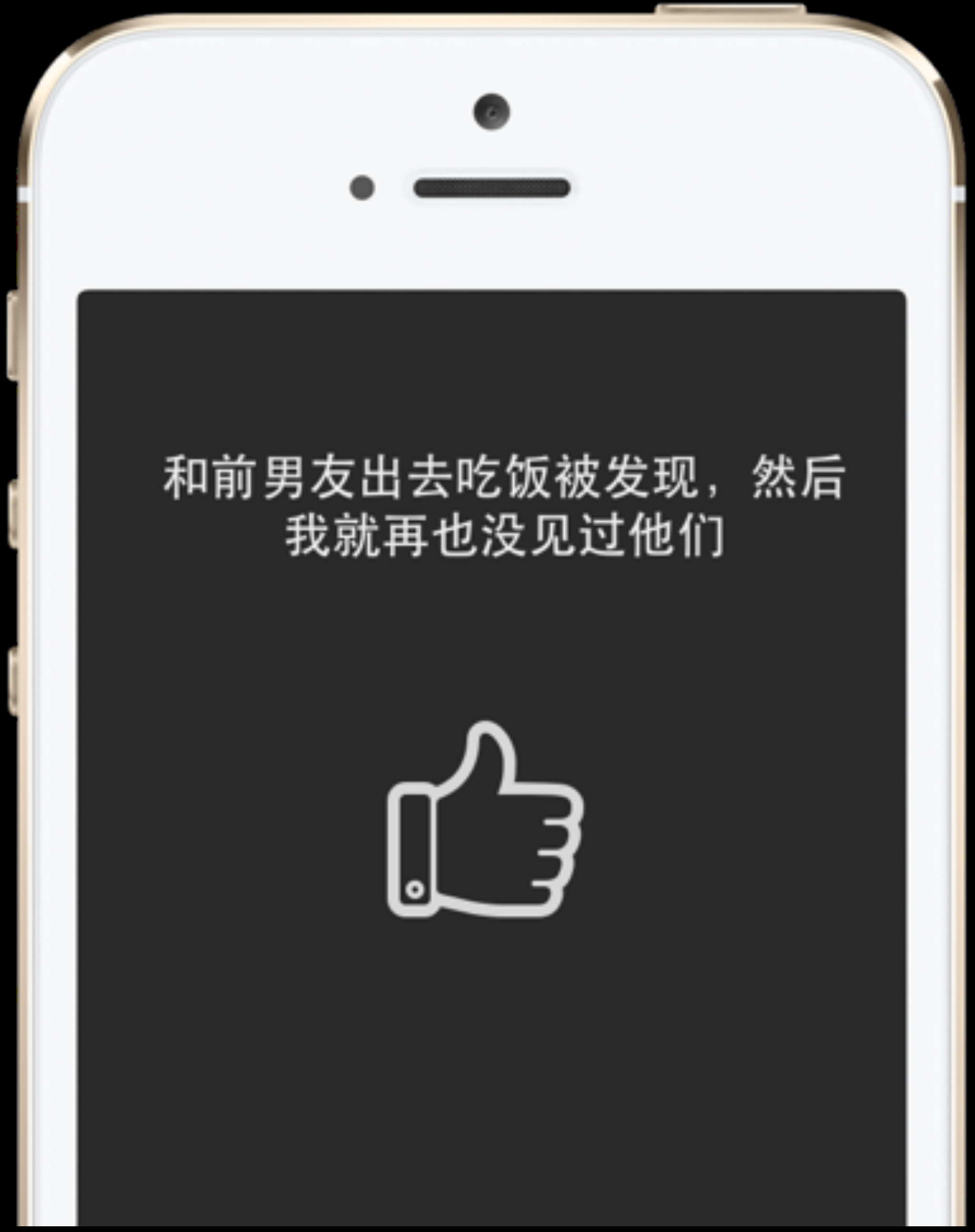
- Cocoa开发现状
- ReactiveCocoa
- MVVM
- Demo
- 使用经验

Cocoa开发现状



和前男友出去吃饭被发现，然后我就再也没见过他们

👍



和前男友出去吃饭被发现，然后
我就再也没见过他们

需要考虑的情况

- 是否已经登录
- 是否已经赞过
- 赞的过程中不能操作
- 与服务端交互完成后改变赞的状态

```
- (void)likeButtonTapped
{
    if (self.hasLoggedIn) {

    } else {
        // popup a login view
    }
}
```

```
- (void)likeButtonTapped
{
    if (self.hasLoggedIn) {
       if (!self.isLiking) {

        }
    } else {
        // popup a login view
    }
}
```

```
- (void)likeButtonTapped
{
    if (self.hasLoggedIn) {
       if (!self.isLiking) {
           if (self.hasLiked) {

            } else {

            }
        }
    } else {
        // popup a login view
    }
}
```

```
- (void)likeButtonTapped
{
    if (self.hasLoggedIn) {
       if (!self.isLiking) {
           if (self.hasLiked) {
               self.isLiking = YES;
                [self.api likeWithComplete:^{
                    self.isLiking = NO;
                    self.likeButton.highlighted = NO;
                }];
            } else {
               self.isLiking = YES;
                [self.api cancelLikeWithComplete:^{
                    self.isLiking = NO;
                    self.likeButton.highlighted = YES;
                }];
            }
        }
    } else {
        // popup a login view
    }
}
```

State

“编程的本质是控制复杂度”

–*The Art of Unix Programming*

State能有效增加复杂度

BOOL visible;	2 states
BOOL enabled;	4 states
BOOL highlighted;	8 states
BOOL selected;	16 states

State是一种缓存

> “There are only two hard things in Computer Science: cache invalidation and naming things”

–Phil Karlton

如何消灭/减少State?

- 搞定产品经理
- 只做简单的App
- 改行
- 换一种编程思路?

ReactiveCocoa

AWARD
gifbin.com

走出舒适区

Are you too busy to improve?
No thanks!
We are too busy
Created by Håkan Forss @hakanforss http://hakanforss.wordpress.com
cc BY NC

“In the beginner’s mind there are many possibilities,
in the expert's mind there are few.”

–Shunryu Suzuki

Input -> Process -> Output

Input

- 键盘输入
- 点击
- 手势操作
- 网络请求
- 磁盘读取

- Delegate

  tableView:didSelectRowAtIndexPath

- Block Callback

  getCategoriesOnComplete:^(NSArray *){}

- Target-Action

  logoutButtonTapped:

- Timers

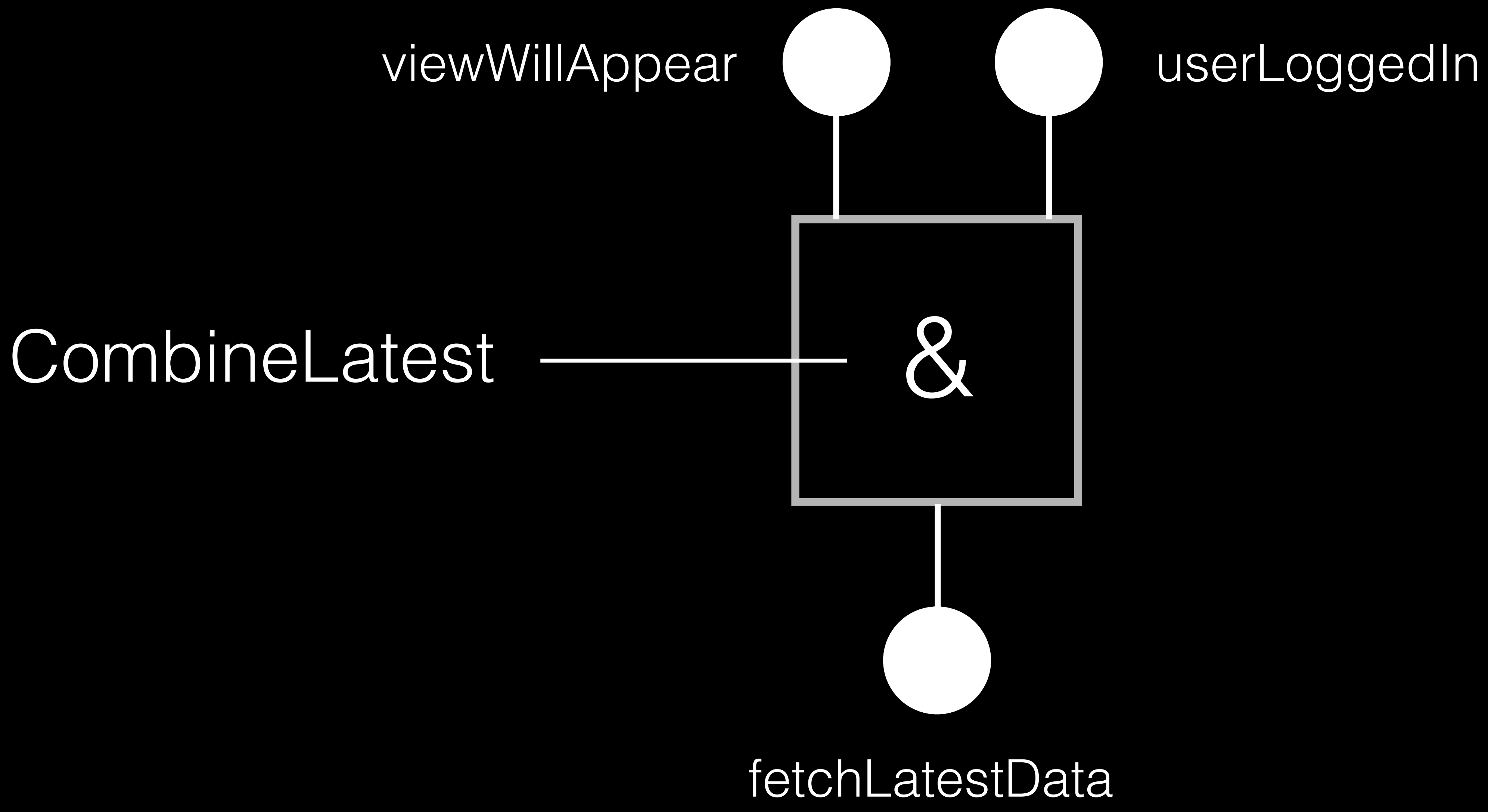
  timerWithTimeInterval:target:selector:userInfo:repeats:

- KVO

Output

- UI的变化
- Model的变化
- 网络请求
- Crash

Signal



viewWillAppear

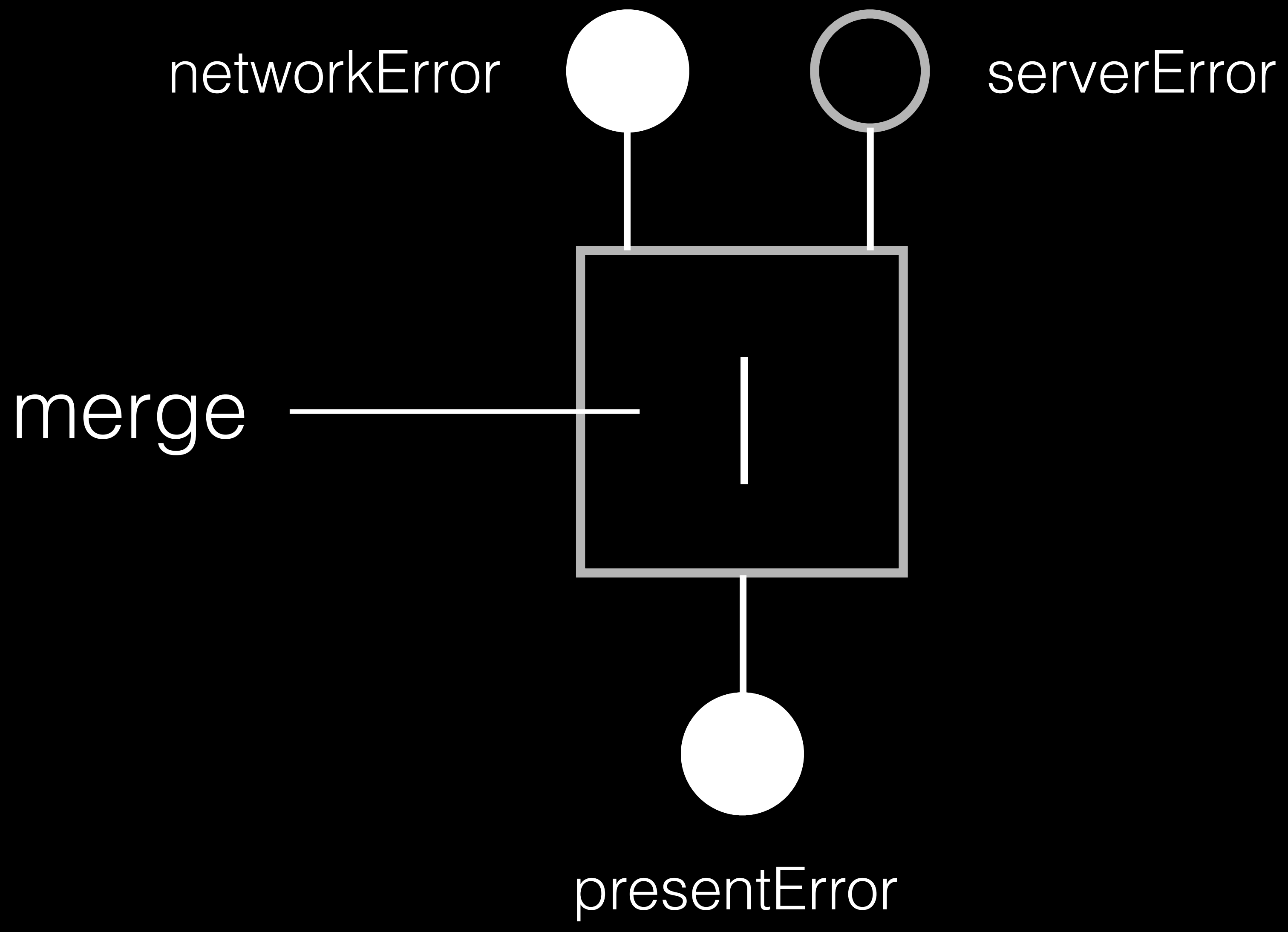
userLoggedIn

CombineLatest

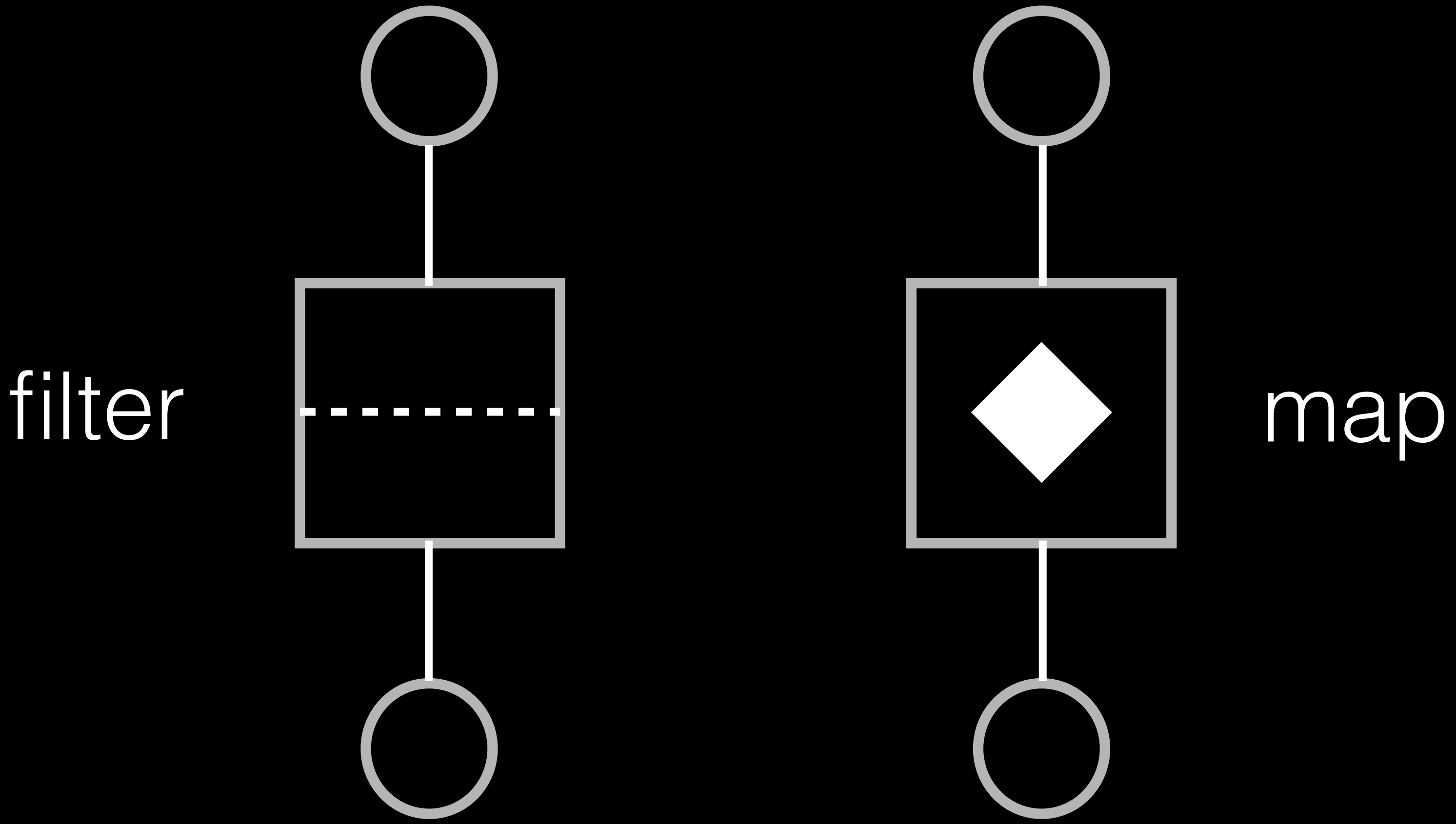
&

fetchLatestData

```
[[RACSignal
   combineLatest:@[viewWillAppearSignal, loggedInSignal]]
   subscribeNext:^(id x) {
   // fetch latest data
}];
```


networkError
serverError
merge
|
presentError

```
[[RACSignal
   merge:@[netErrorSignal, serverErrorSignal]]
   subscribeNext:^(id x) {
   // present Error
}];
```

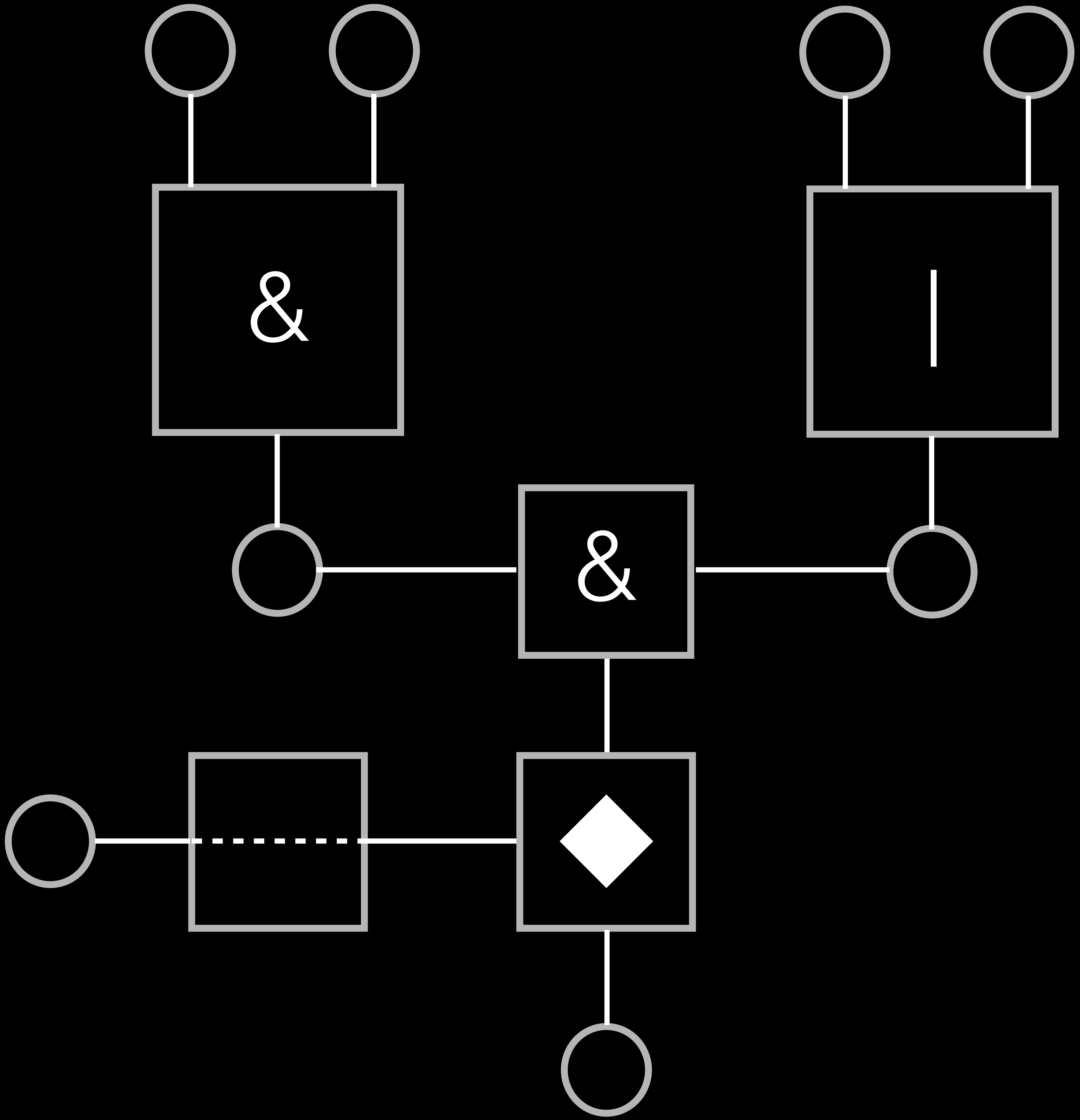


filter

map

```
[anyNumberSignal filter:^BOOL(NSNumber *number) {
    return [number intValue] % 2;
}];
```

```
[userDataSignal map:^id(NSDictionary *userData) {
    return [[HBUser alloc] initWithDictionary:userData error:NULL];
}];
```

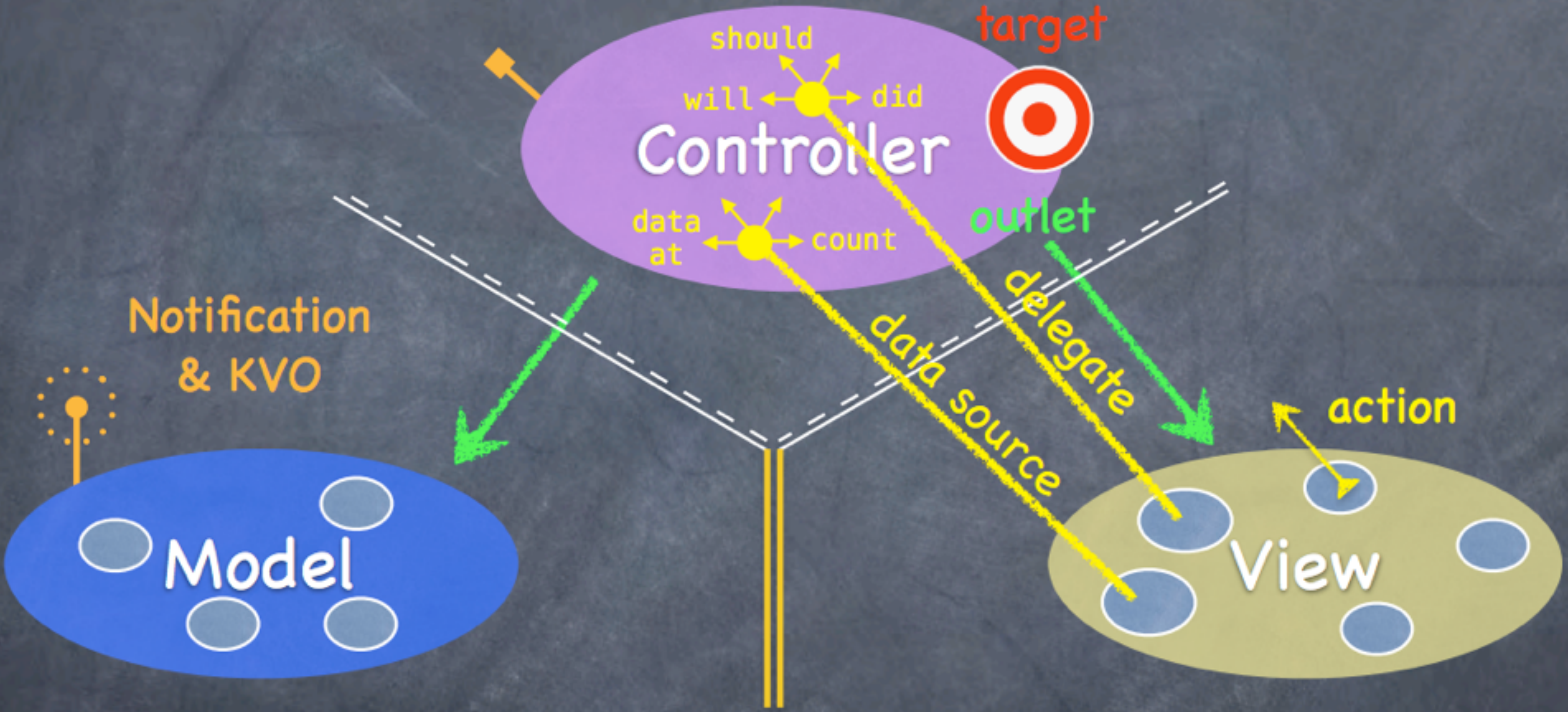


RACCommand

- 通常用来表示某个Action
- 常常跟UIControl绑定使用
- 有几个重要属性
    - executing
    - executionSignals
    - errors

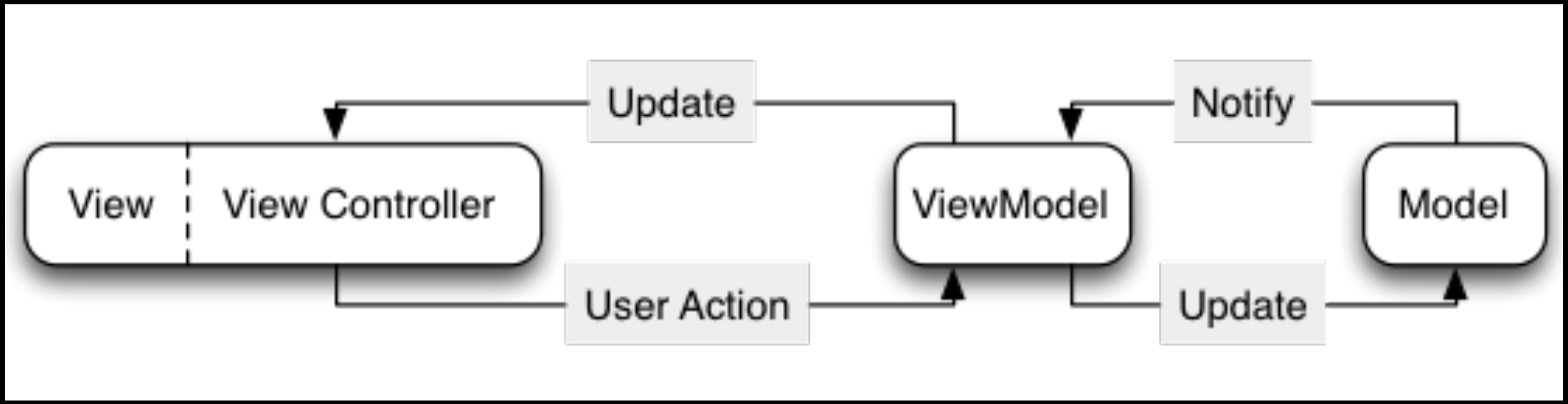
MVVM

MVC

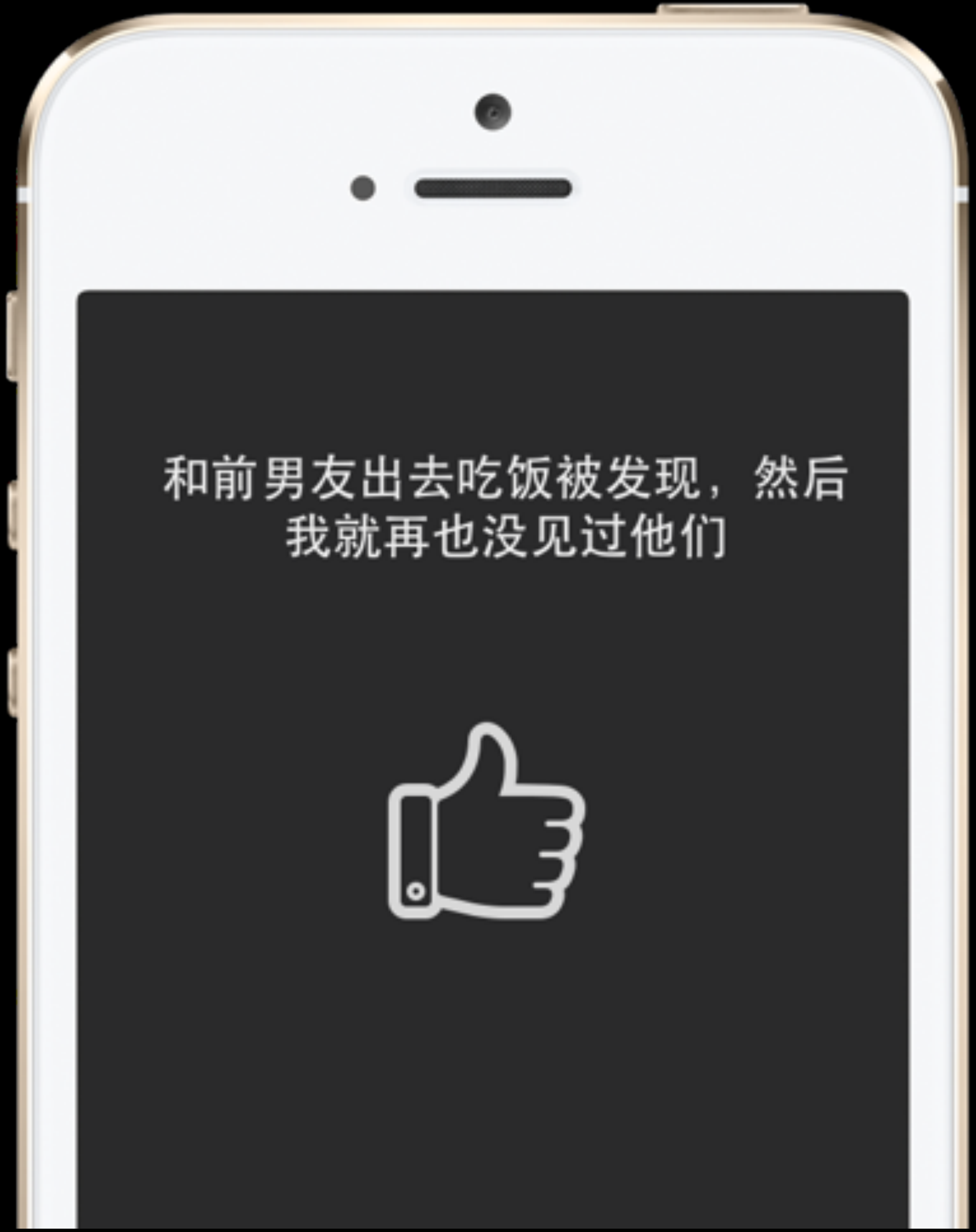


should
will
did
Controller
target
data
at
count
outlet
Notification
& KVO
delegate
data source
action
Model
View

MVVM


Update
Notify
View
View Controller
ViewModel
Model
User Action
Update

Demo



和前男友出去吃饭被发现，然后我就再也没见过他们

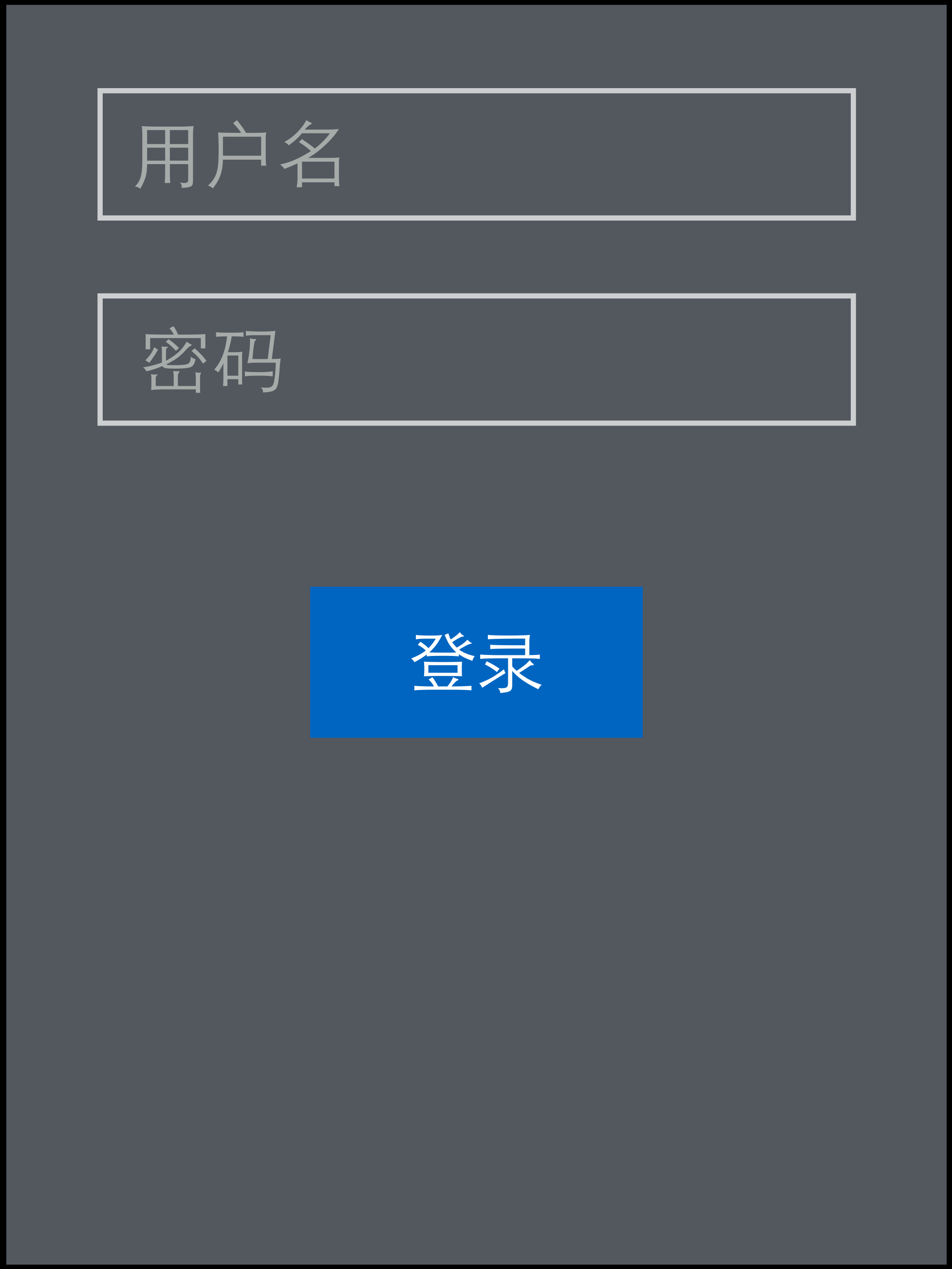
👍

```
[RACObserve(self, shouldPopLoginView) subscribeNext:^(id x) {
    // pop up login view
}];
```

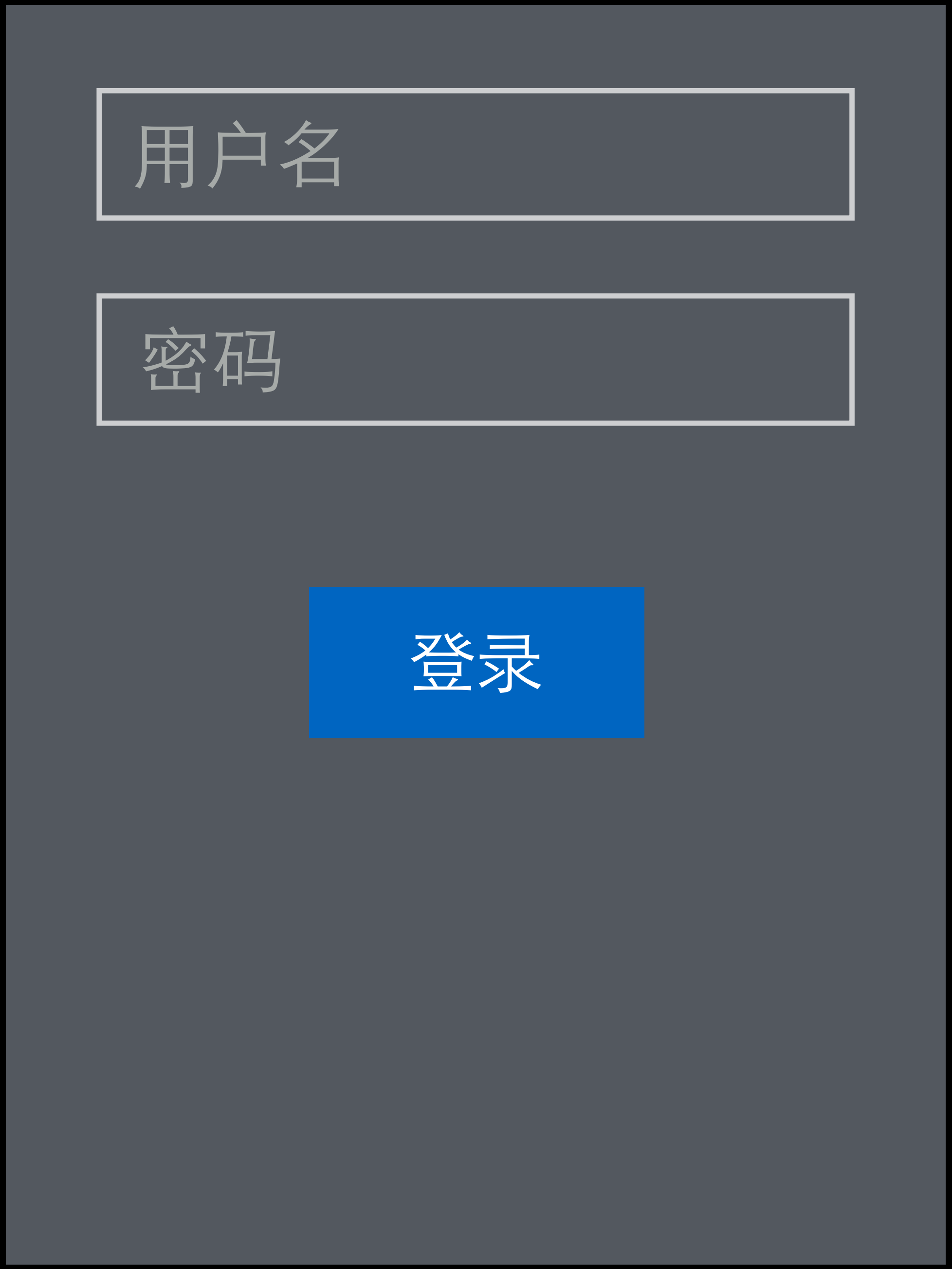
```
RAC(self, likeButton.highlighted) = [[self.likeCommand.executionSignals
                                                   switchToLatest]
                                                   map:^id(id value) {
    // handle result
    return @YES; // or @NO
}];
```

```
self.likeButton.rac_command = [[RACCommand alloc]
                               initWithEnabled:[self.likeCommand.executing not]
                               signalBlock:^RACSignal *(id input) {
    if (!self.hasLoggedIn) {
        self.shouldPopLoginView = YES;
    } else {
        [self.likeCommand execute:nil];
    }
    return nil;
}];
```

登录



登录



```
RAC(self.loginButton, enabled) = formValid;

RACSignal *formValid = [RACSignal
  combineLatest:@[
    self.usernameField.rac_textSignal,
    self.passwordField.rac_textSignal,
  ]
  reduce:^(NSString *name, NSString *password){
    return name.length && password.length;
  }
];
```

登录

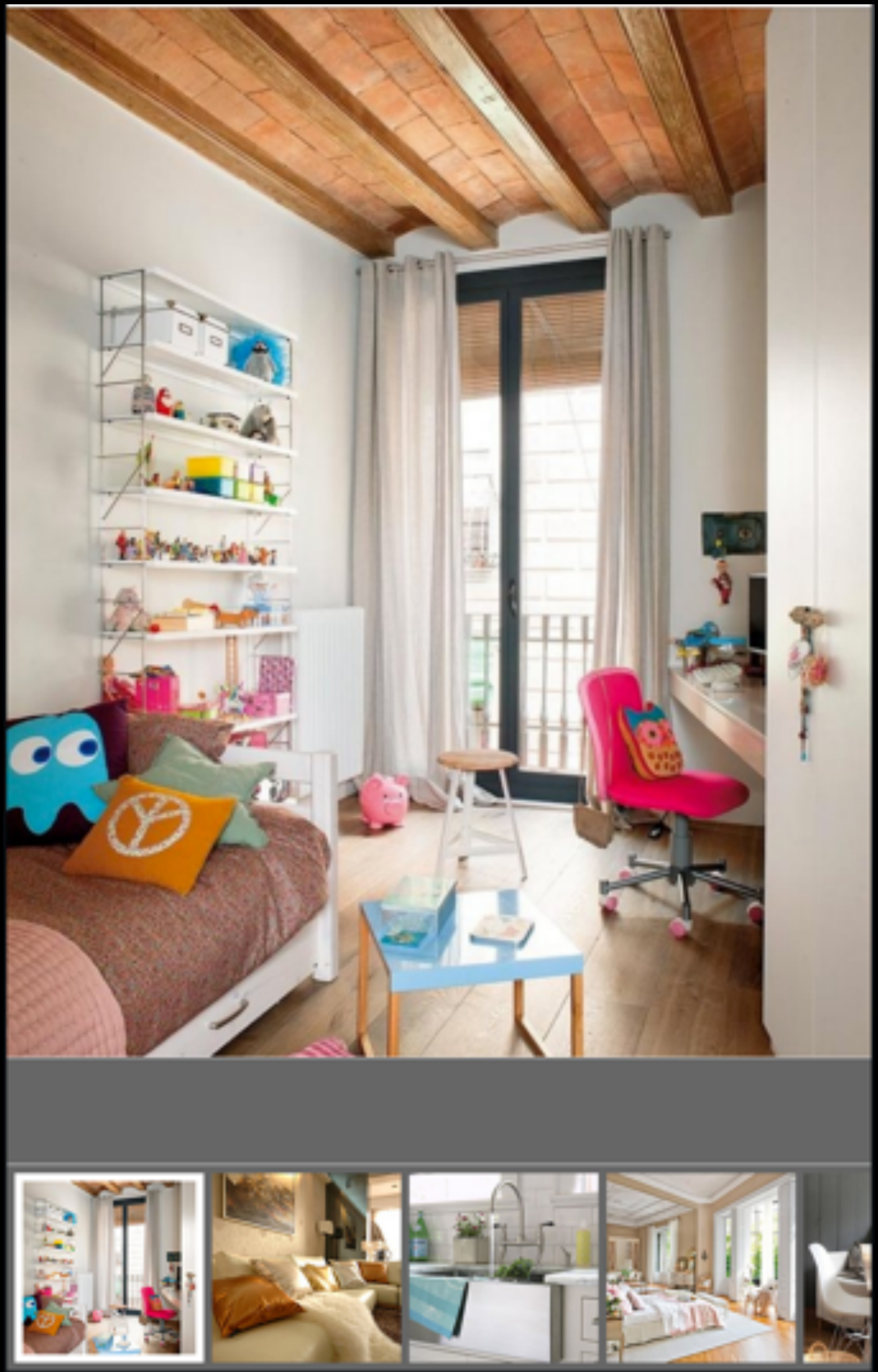
```
RACSignal *textColor = [executing map:^(NSNumber *x) {
    return x.boolValue ? gray : black;
}];
```

```
RAC(self.usernameField.textColor) = textColor;
RAC(self.passwordField.textColor) = textColor;
```

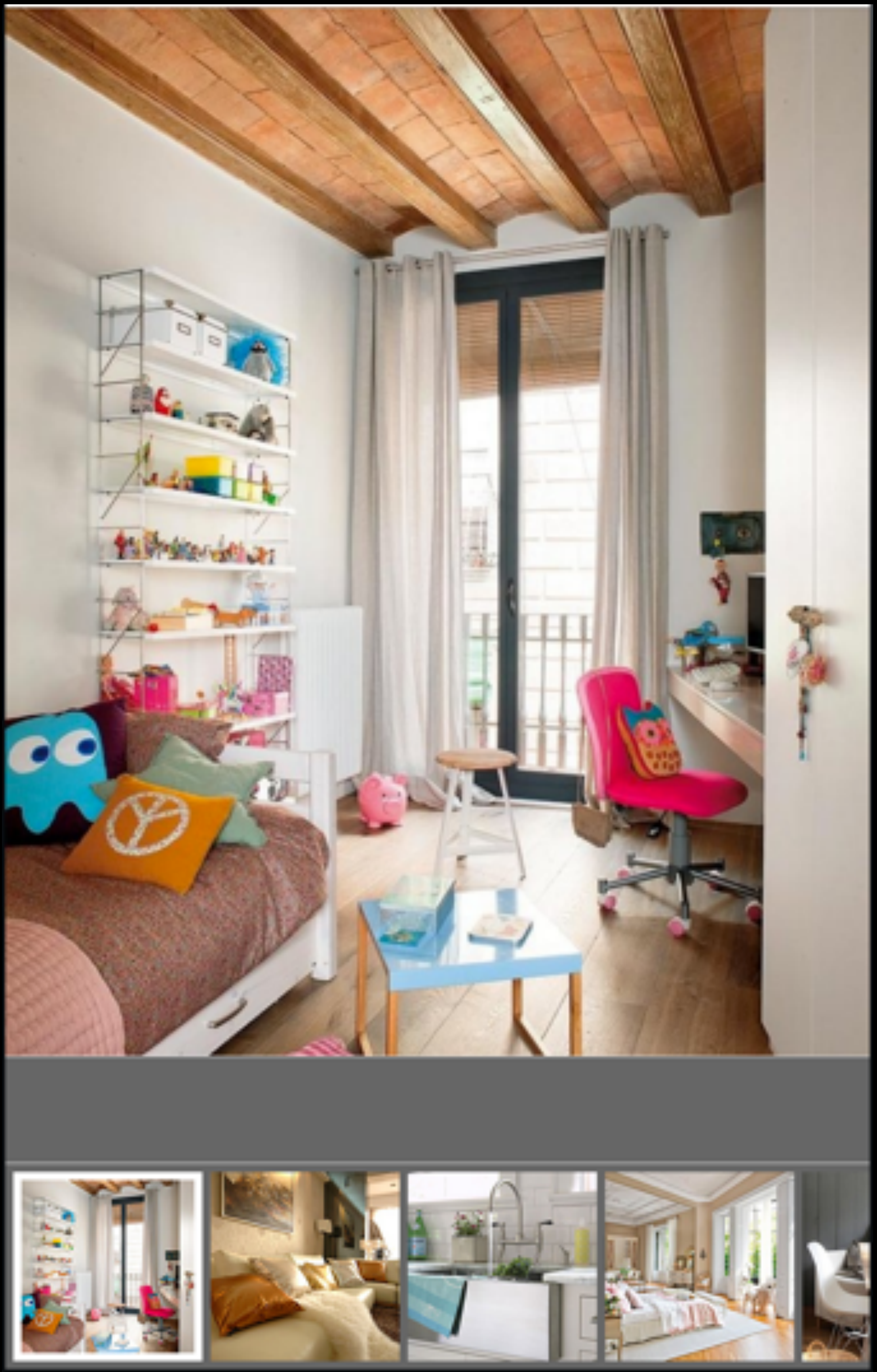
```
RACSignal *notProcessing = [executing map:^(NSNumber *x) {
    return @(!x.boolValue);
}];
```

```
RAC(self.usernameField.enabled) = notProcessing;
RAC(self.password.enabled) = notProcessing;
```

iPhoto

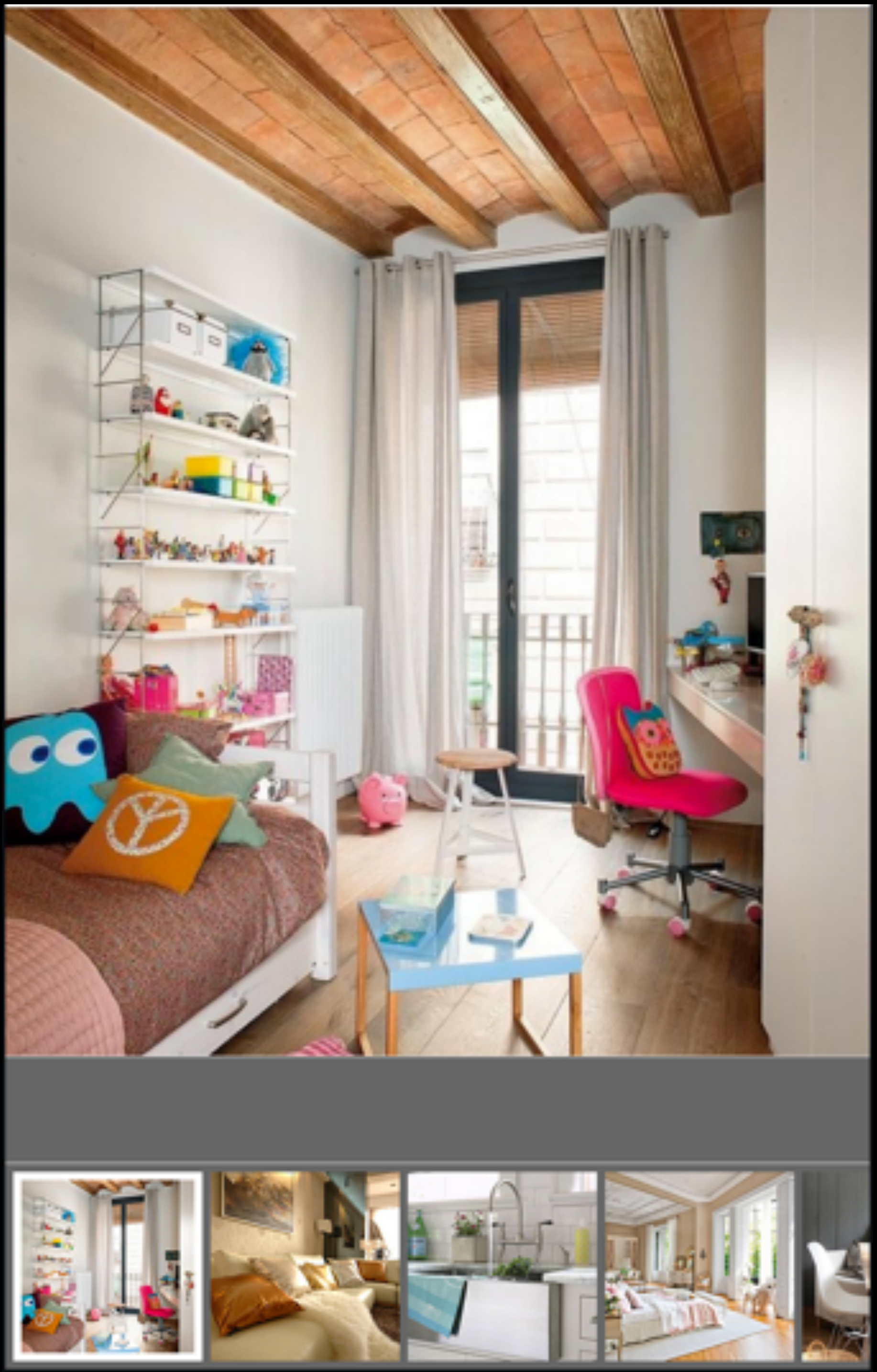


iPhoto



```
[RACObserve(viewModel, indexPath)
subscribeNext:^(NSIndexPath *indexPath) {
   [self.collectionView
   scrollToItemAtIndexPath:indexPath];

   CellViewModel *cvm =
   viewModel.cellViewModels[indexPath.row];

   cvm.active = YES;

   CellViewModel *lastActiveCVM =
   viewModel.lastActiveCellViewModel;

   lastActiveCVM.active = NO;

   viewModel.lastActiveCellViewModel = cvm;
}];
```

iPhoto



```
- (void)scrollViewDidEndDecelerating:(UIScrollView *)scrollView
{
    // 根据scrollView算出indexPath
    viewModel.indexPath = indexPath;
}
```

```
- (void)collectionView:(UICollectionView *)collectionView
    didSelectItemAtIndexPath:(NSIndexPath *)indexPath
{
    // 处理高亮
    viewModel.indexPath = indexPath;
}
```

注意事项

使用经验

- 解题思路
- 性能
- 调试
- 内存使用
- sendNext
- Scheduler
- 使用property还是signalProperty
- TableView的处理
- RAC在花瓣的使用场景

解题思路

- 百思不得解
- 想到一种方法，但好像不够RAC
- 算了，时间有限，还是回到熟悉的编程模式吧

性能

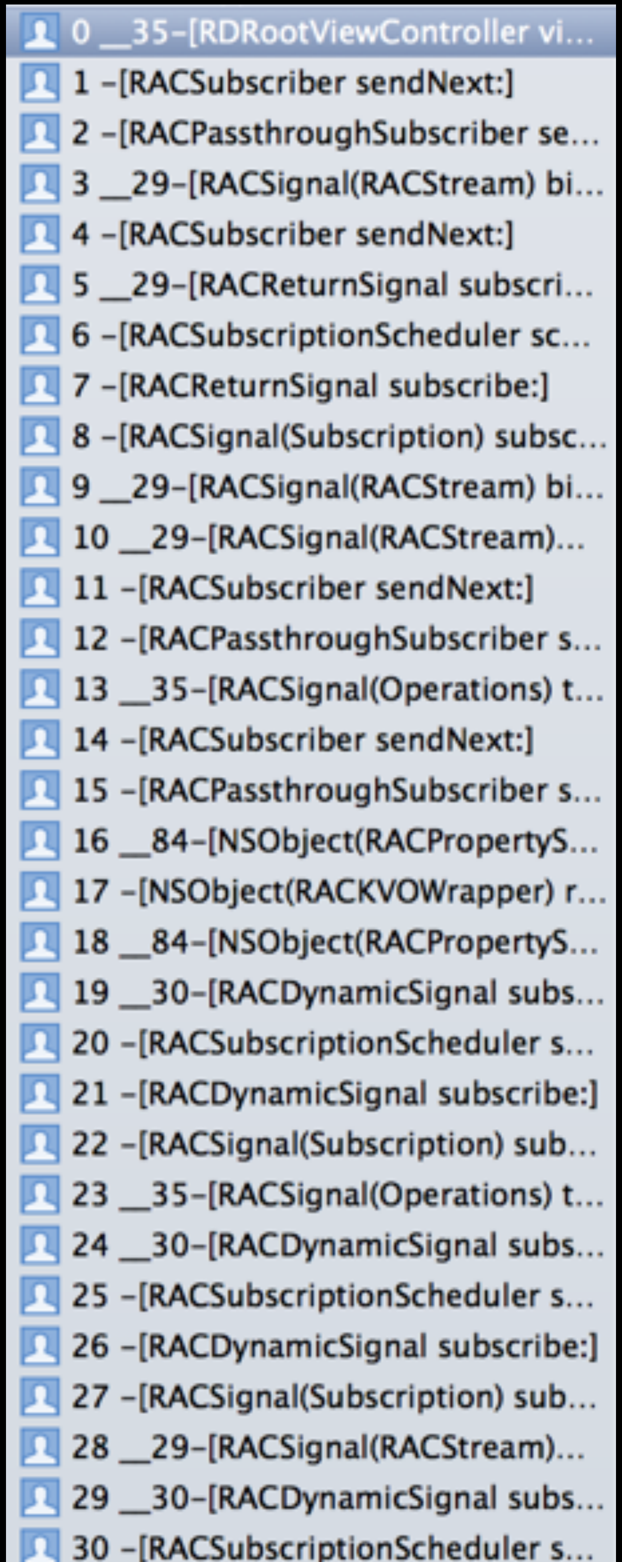
- subscribeNext 很慢，比纯KVO慢了1个数量级
- 接收到变化比较慢，比纯KVO慢了大约5倍

调试

```
[signal subscribeNext:^(id x){

}];
```

调试

```
[signal subscribeNext:^(id x){

}];
```



```
0 __35-[RDRootViewController vi...
1 -[RACSubscriber sendNext:]
2 -[RACPassthroughSubscriber se...
3 __29-[RACSignal(RACStream) bi...
4 -[RACSubscriber sendNext:]
5 __29-[RACReturnSignal subscri...
6 -[RACSubscriptionScheduler sc...
7 -[RACReturnSignal subscribe:]
8 -[RACSignal(Subscription) subsc...
9 __29-[RACSignal(RACStream) bi...
10 __29-[RACSignal(RACStream)...
11 -[RACSubscriber sendNext:]
12 -[RACPassthroughSubscriber s...
13 __35-[RACSignal(Operations) t...
14 -[RACSubscriber sendNext:]
15 -[RACPassthroughSubscriber s...
16 __84-[NSObject(RACPropertyS...
17 -[NSObject(RACKVOWrapper) r...
18 __84-[NSObject(RACPropertyS...
19 __30-[RACDynamicSignal subs...
20 -[RACSubscriptionScheduler s...
21 -[RACDynamicSignal subscribe:]
22 -[RACSignal(Subscription) sub...
23 __35-[RACSignal(Operations) t...
24 __30-[RACDynamicSignal subs...
25 -[RACSubscriptionScheduler s...
26 -[RACDynamicSignal subscribe:]
27 -[RACSignal(Subscription) sub...
28 __29-[RACSignal(RACStream)...
29 __30-[RACDynamicSignal subs...
30 -[RACSubscriptionScheduler s...
```

sendNext

- 由于ObjectiveC语言的限制，无法得知next的值到底是哪种类型，使用起来略有不便，需要看注释/代码，才能知道传的是什么值，然后手动转一下。

Scheduler

- deliverOn:
- subscribeOn:

使用property还是signalProperty

如无必要，使用正常的property即可，外部可以对它RACObserve或KVO

TableView的处理

把每个TableViewCell视为一个独立的View，也给它提供一个cellViewModel，而这个cellViewModel来自于tableViewController.viewModel.cellViewModels，由于cell跟cellViewModel是绑定的，所以在VC中只要改变cellViewModel，cell就会自动做出响应

RAC在花瓣的使用场景

- API Client
- 瀑布流和采集列表详情页
- 画板页
- 个人页
- 「一起」

建议

- 可以去了解下理念
- 可以在自己的side project中用起来
- 等RAC 3.0

谢谢