

QCon 全球软件开发大会 【北京站】2016

MVVM与FRP编程实战

梁士兴 臧成威

个人简介

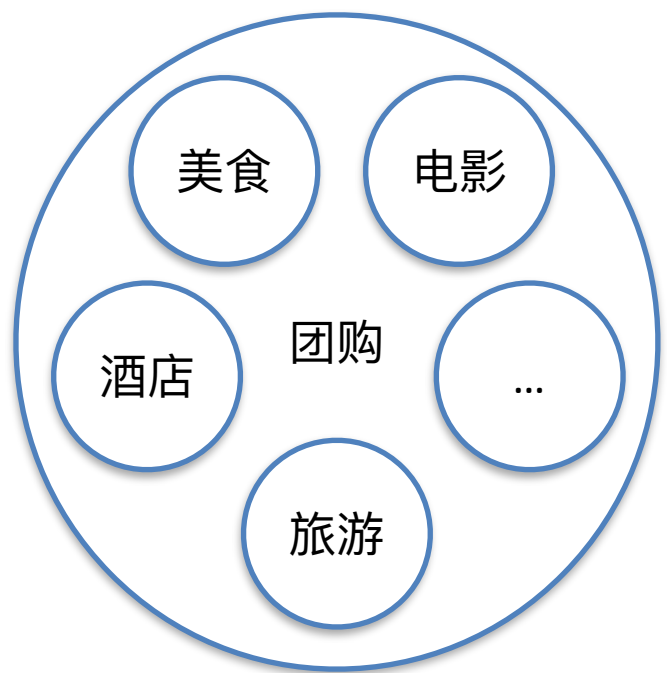


梁士兴，2009年毕业于北京航空航天大学，随后在IBM CDL工作了5年。2014年加入美团，现在负责酒店旅游事业群的iOS开发工作。对iOS技术、软件架构模式非常感兴趣。

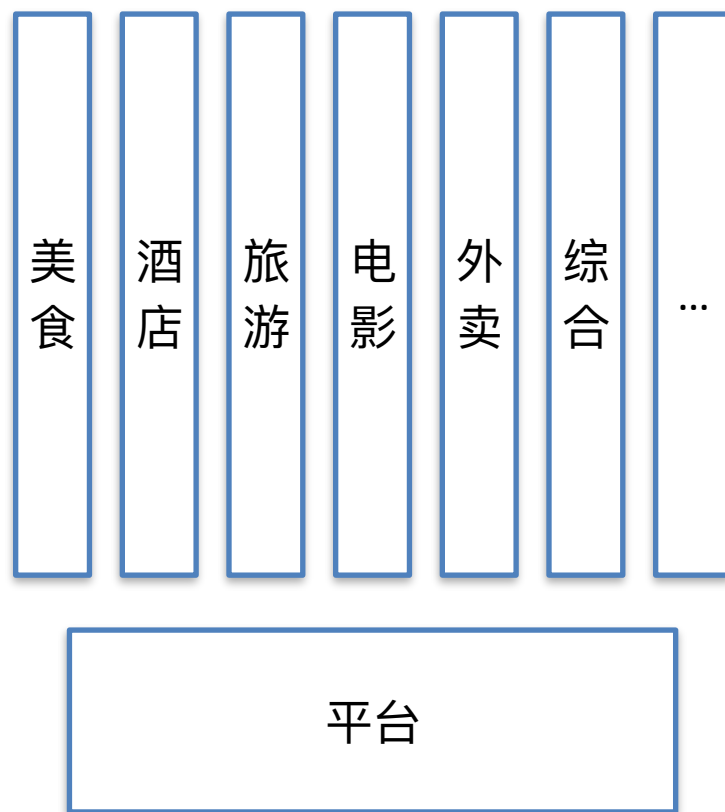
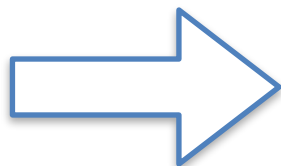
梁士兴

酒店旅游事业群

背景—团购架构的演化

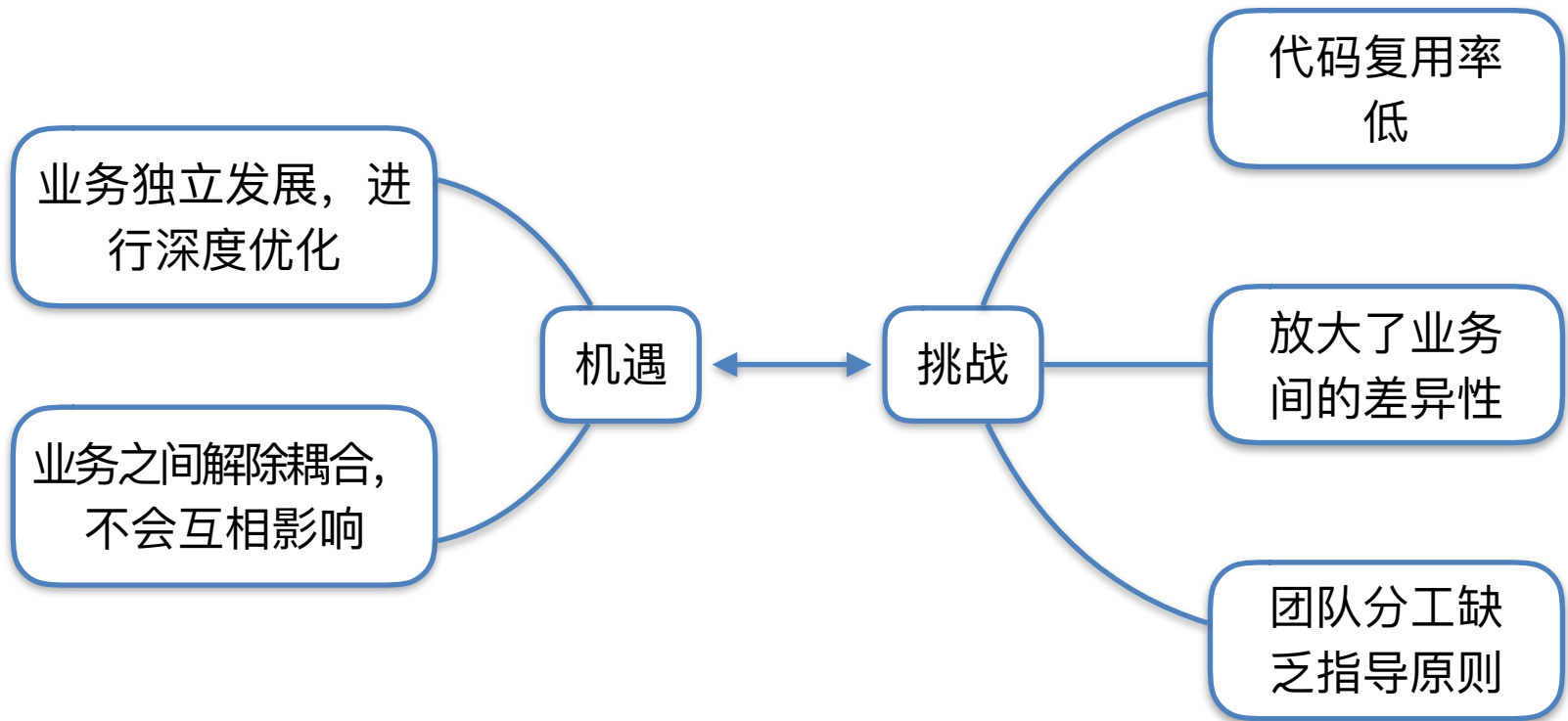


单一团购业务



平台+垂直业务

机遇与挑战



挑战

代码复用率
低

频道之间频繁进行代码拷贝，App的包大小显著增大

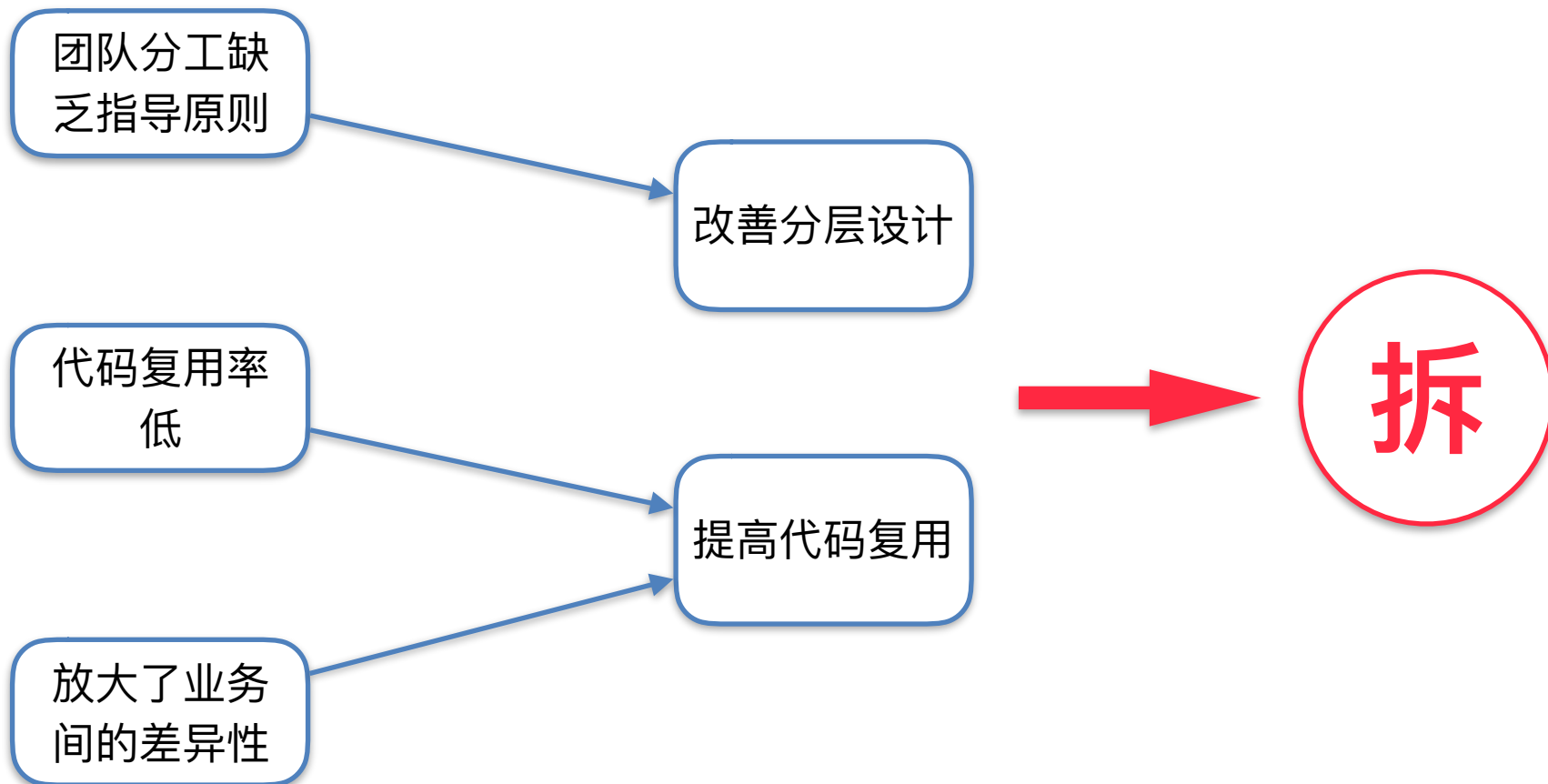
放大了业务
间的差异性

从“相同”变为“相似”。
缺乏对差异性的定制能力，导致很多东西要从头再来。
新业务在初期，表现的相对艰难，可以重用东西的偏少。

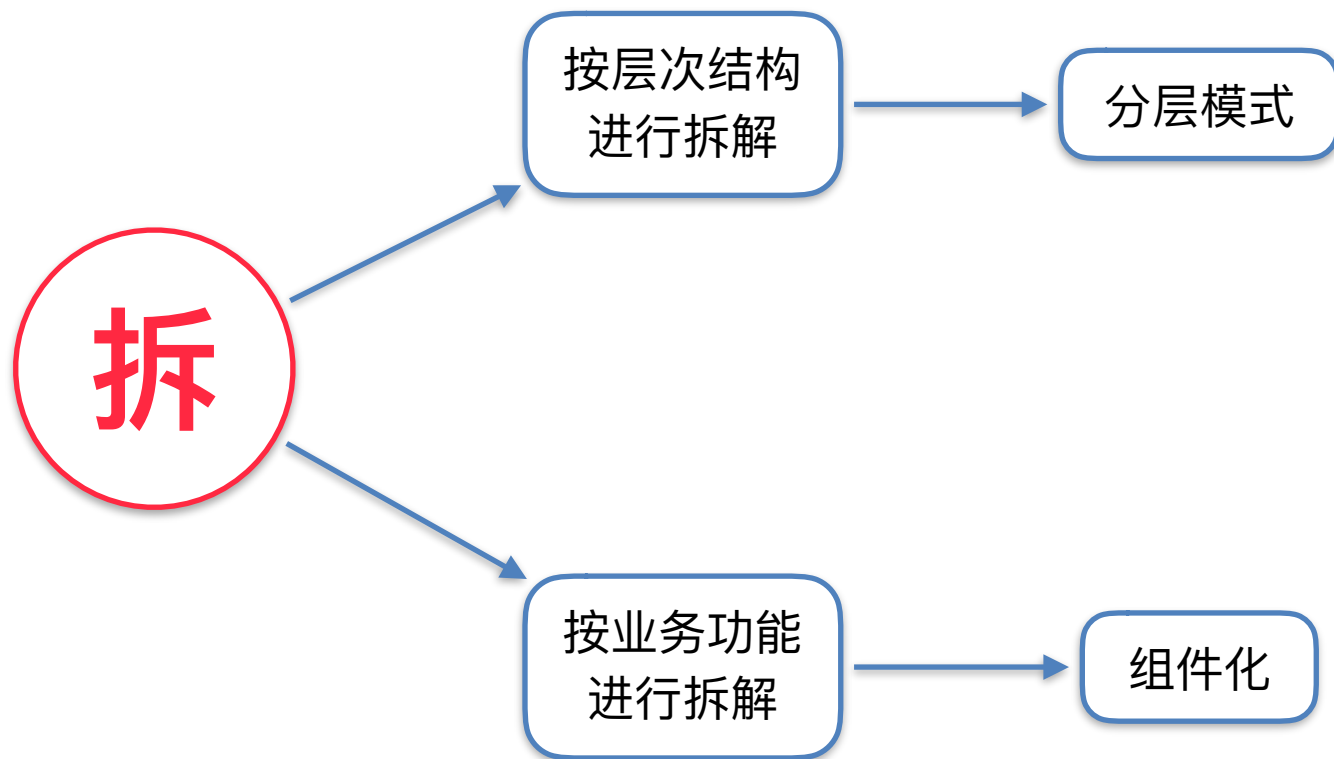
团队分工缺
乏指导原则

不同级别的工程师分工缺乏有效的指导原则。

解决问题—从“拆”开始



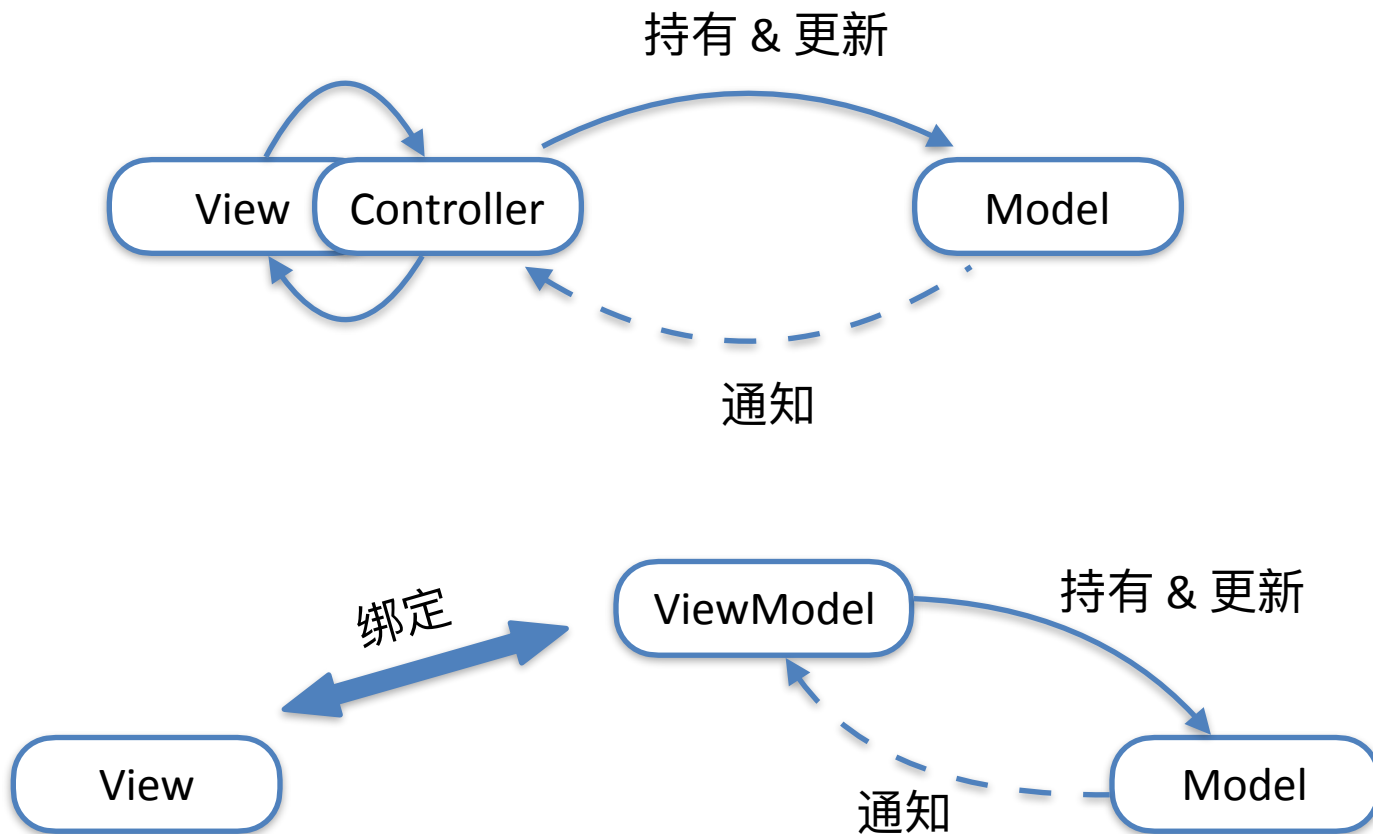
如何“拆”



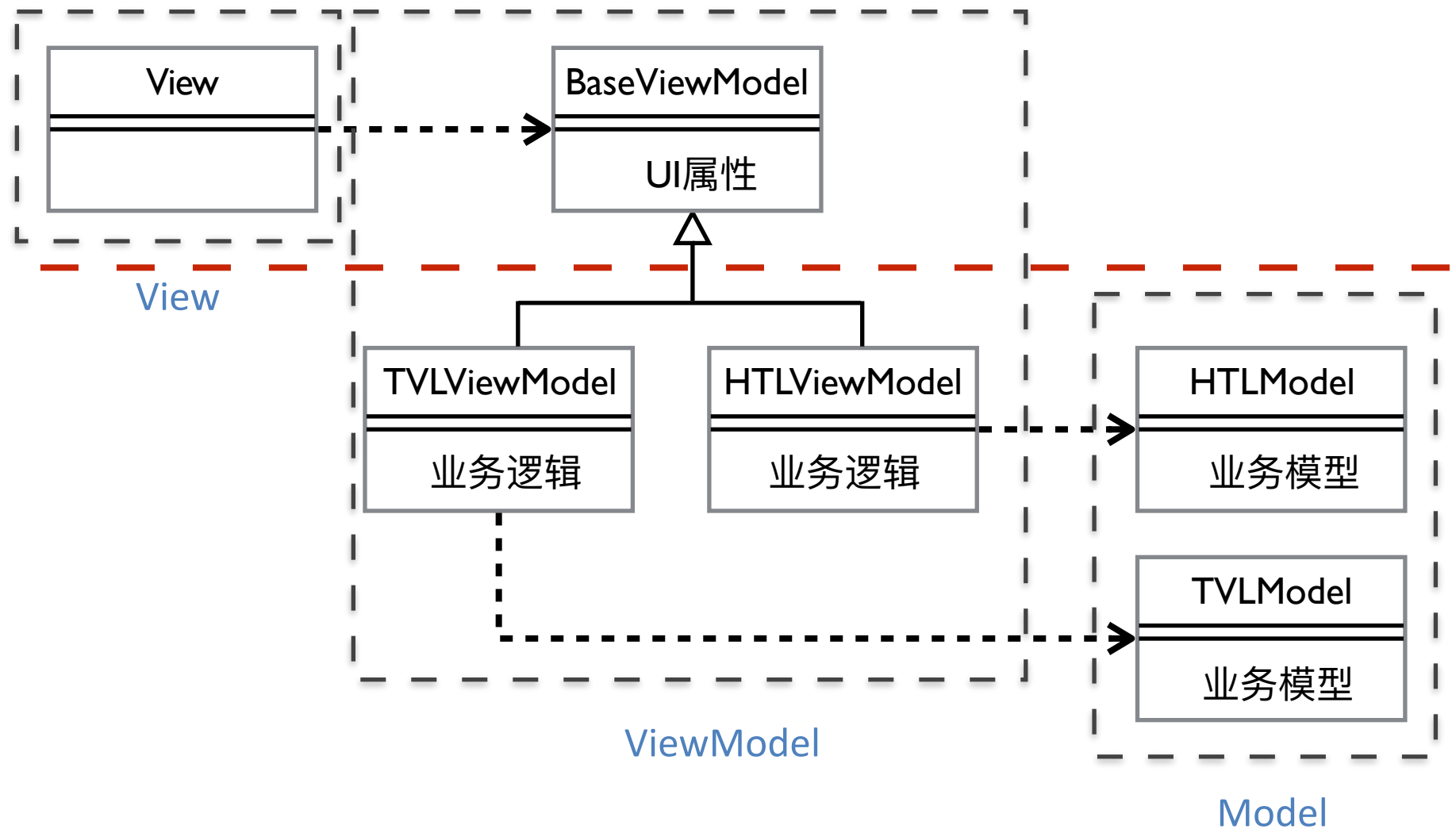


分层模式

MVC VS MVVM



分层模式—MVVM



绑定的故事



绑定

图片
主标题
标签列表
评价打分
优惠信息
附加信息
右脚标

ViewModel

景点Model

imgURL
name
tags
rateScore
specialOffers
purchaseRecords
placeMark
distance
cityID
...

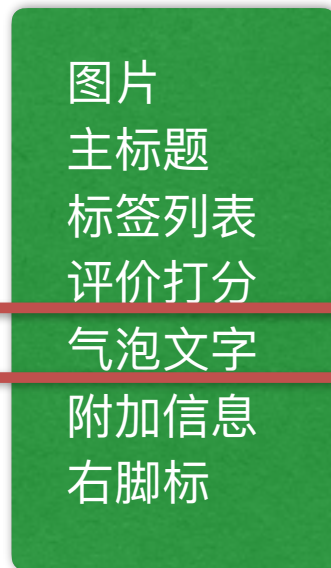
Model

View

绑定的故事



绑定



ViewModel

景点Model

imgURL
name
tags
rateScore
specialOffers
purchaseRecords
placeMark
distance
cityID
...

Model

View

优势

1 分层模式

UI与业务逻辑做隔离

2 分工指导原则

新人负责UI开发，
有经验的工程师负责业务逻辑开发

3 View可复用

View可以被不同的业务中复用；不同业务开发的
ViewModel通过依赖注入的方式注入到View中

4 View可替换

不同业务线可以根据实际需要，将View替换成需要的版本



组件化

组件化—按功能进行分解



组件之间存在依赖?

1

组件之间依赖公共的数据源

2

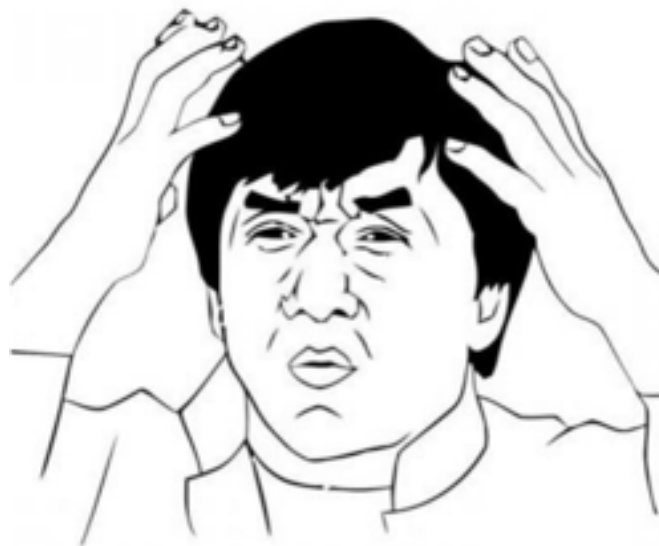
组件之间形成输入输出关系

3

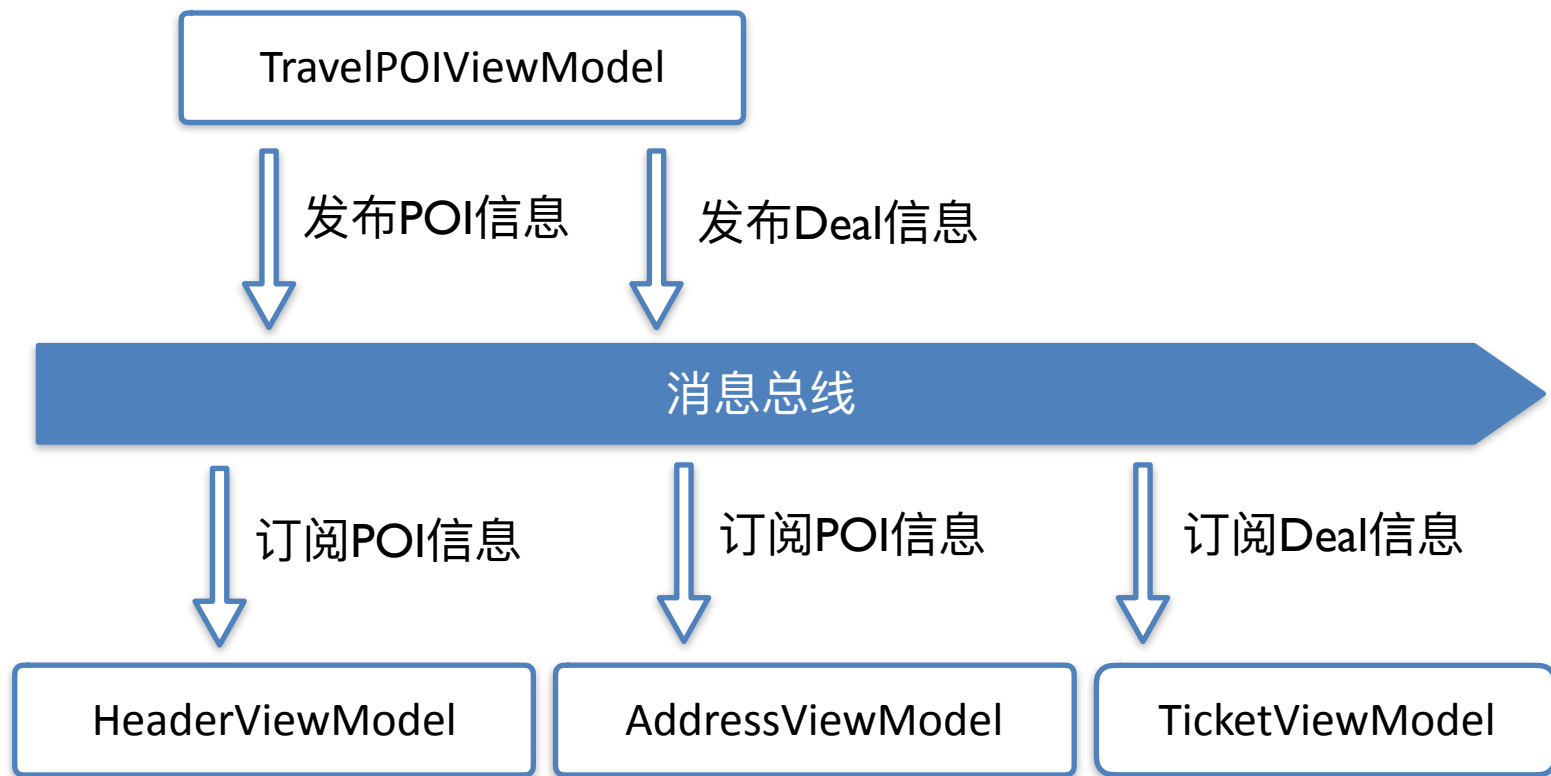
UI交互存在联动

4

...



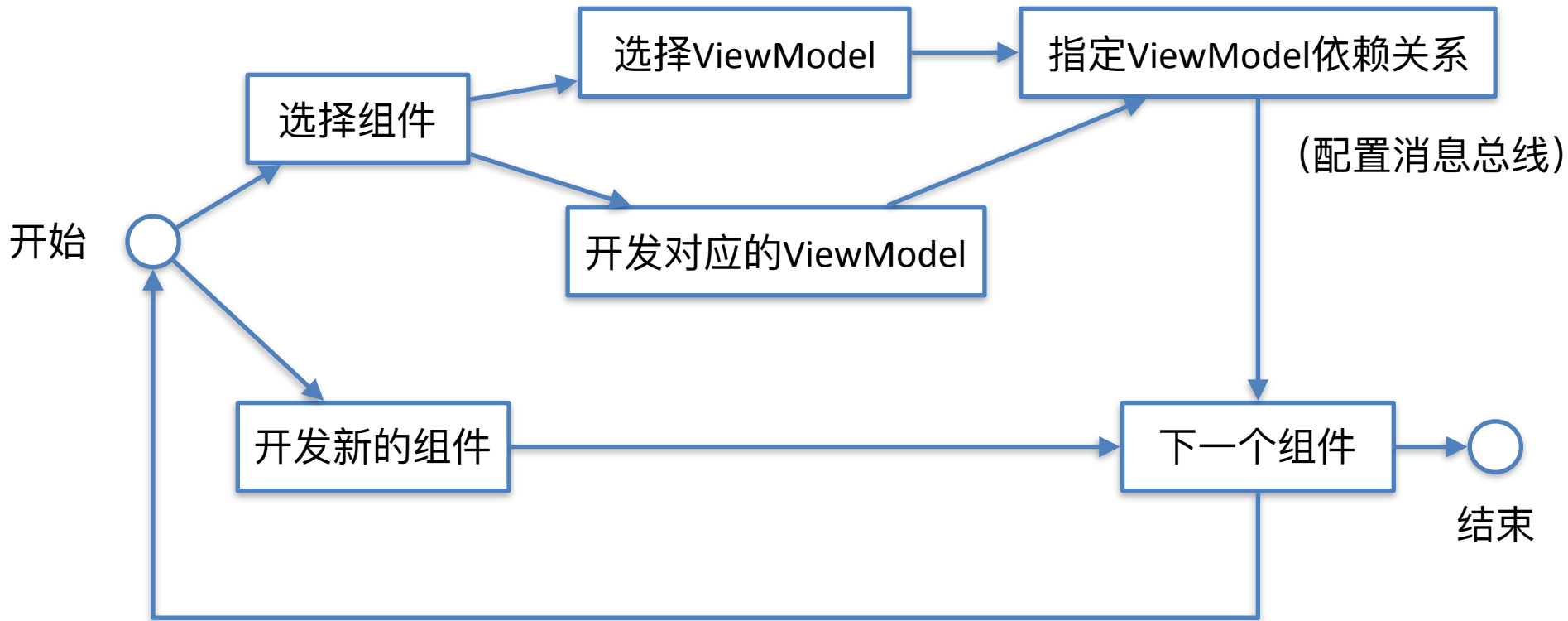
解除组件间依赖—消息总线



拆一两个维度

	头图组件	地址组件	评价组件	商品组件
View	景点头图View	地址View	评价View	景点商品View
ViewModel	头图 ViewModel	头图 ViewModel	评价ViewModel	景点商品 ViewModel
Model	头图Model	头图Model	评价Model	景点商品Model

组装&开发—实现业务



展望—动态组装

1 通过组件组装产生页面

2 统一View / ViewModel / 消息
总线配置接口

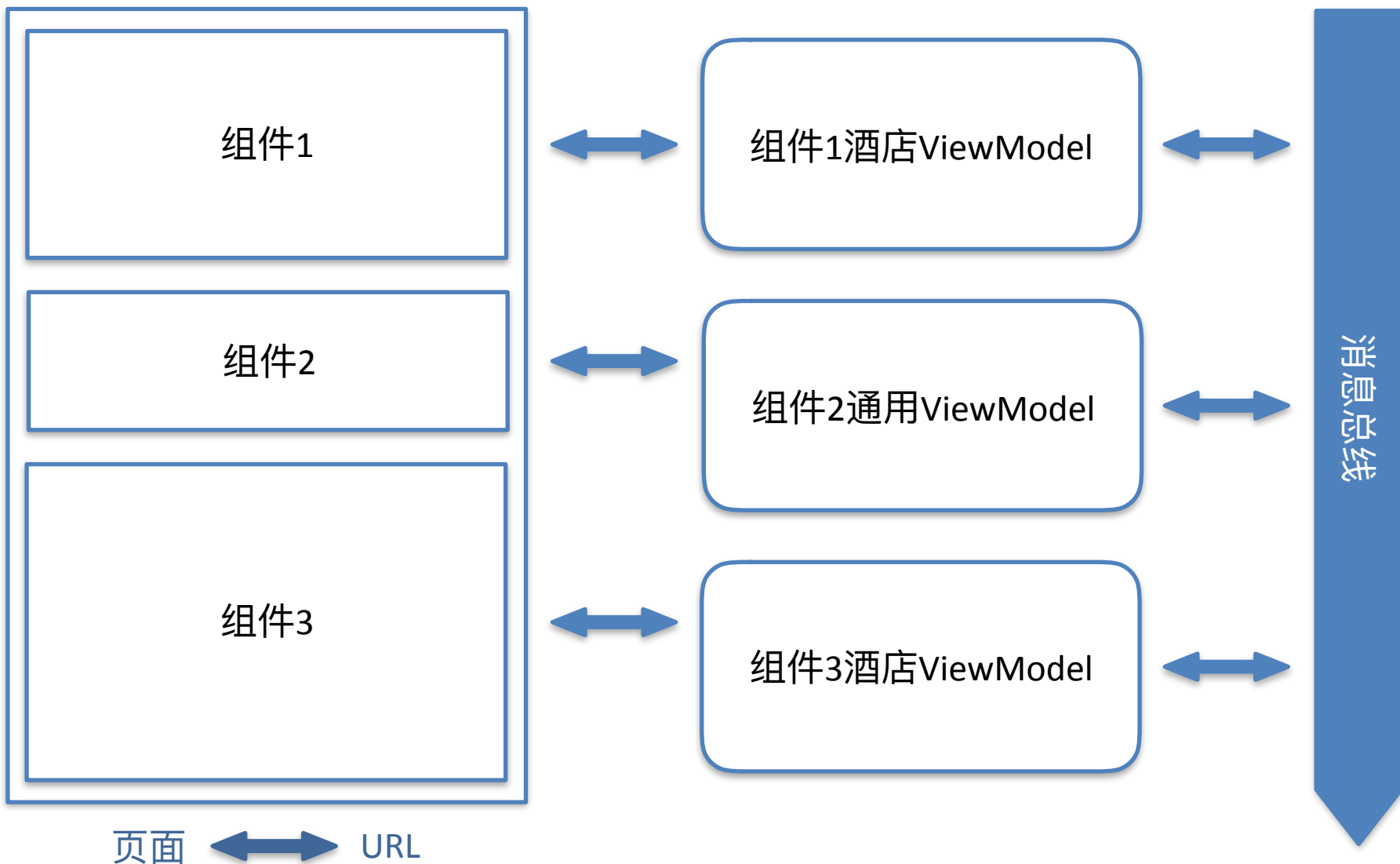
3 通过配置文件描述页面的组件
(View/ViewModel/总线配置)

4 通过后台下发配置文件



1 更大的灵活性

2 更高的效率



总结

1 拆

MVVM的分层模式；
组件化的分解方式

3 行动起来

对业务进行“拆”分，通过“组装”完成开发。最终实现业务组件的复用，和开发效率的提升。

2 组装

将组件的View/ViewMode/
依赖关系配置完成组装，即可实现业务需要。

One more Thing

- “绑定”在MVVM中有至关重要的作用，如何优雅的实现绑定？
- FRP来帮助你，下面有请臧成威介绍FRP在MVVM开发过程中发挥的威力。

MVVM的绑定

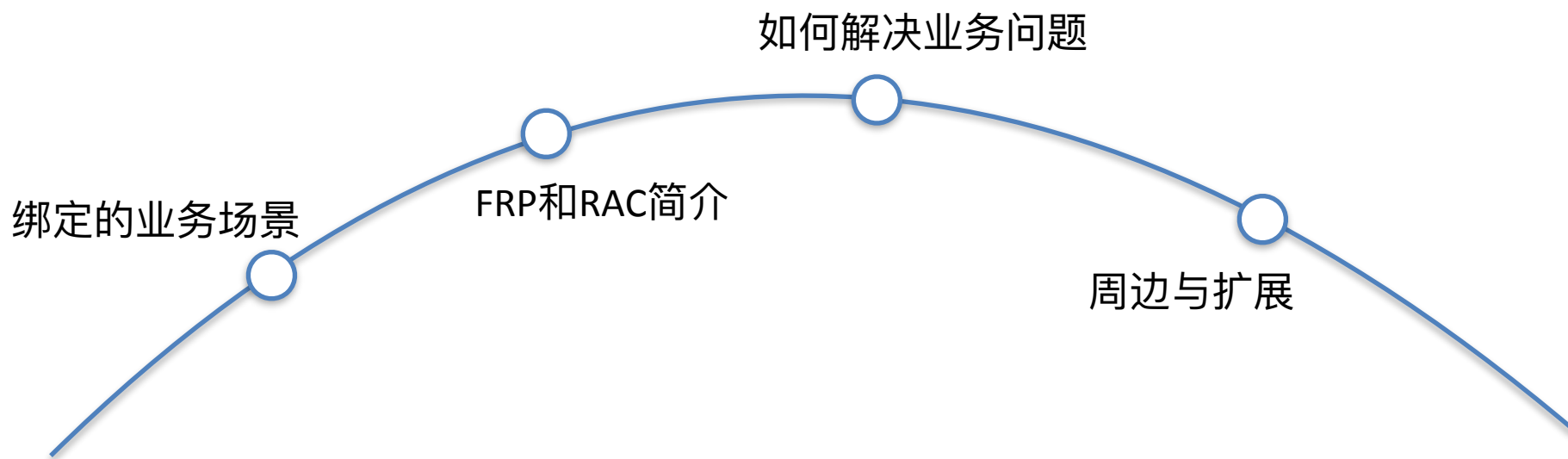


个人简介



- 臧成威 美团·大众点评移动技术专家
- 目前负责美团iOS的发布工程系统的研发和流程优化梳理。
- 擅长多语言范式，从2014年初的ReactiveCocoa 2.0时代开始接触FRP，通过对多种语言的学习和研究加深理解。

MVVM绑定内容





绑定的业务场景

绑定的业务场景

- 单向数据绑定
- 集合数据绑定
- 双向数据绑定
- 执行过程绑定
- 异常处理机制

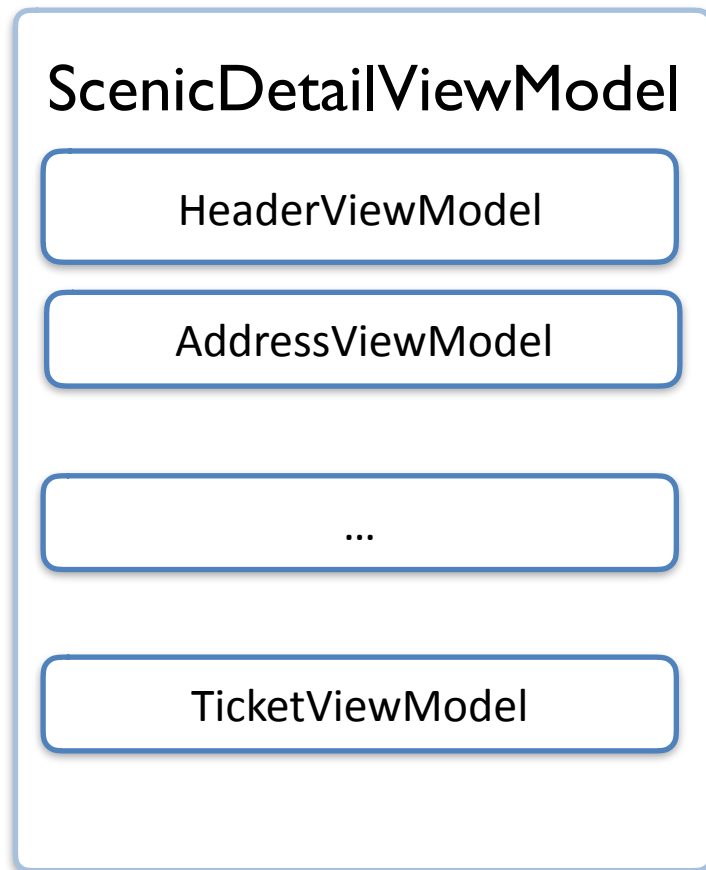
绑定的业务场景



绑定

绑定

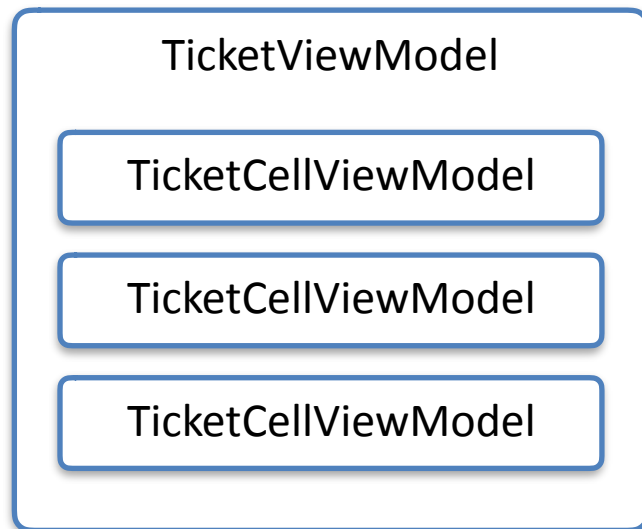
绑定



绑定的业务场景



绑定



绑定的业务场景



绑定的业务场景



绑定

HeaderViewModel

StarCmd

绑定的业务场景



The background features a series of vibrant, multi-colored lines (blue, green, yellow, red) that flow and curve across the frame. In the upper right corner, there is a pattern of glowing green binary code (0s and 1s) that appears to be receding into the distance.

FRP与ReactiveCocoa

什么是FRP

```
9 function addNumbers(a, b) {
10   return a + b;
11 };
12
13 // Takes the values of an array and returns the total. Demonstrates simple
14 // recursion.
15 function totalForArray(arr, currentTotal) {
16   currentTotal = addNumbers(currentTotal + arr.shift());
17
18   if(arr.length > 0) {
19     return totalForArray(currentTotal, arr);
20   }
21   else {
22     return currentTotal;
23   }
24 }
25
26 // Or you could just use reduce.
27 function totalForArray(arr) {
28   return arr.reduce(addNumbers);
29 }
30
31 // Should really be called divideTwoNumber
32 function average(total, count) {
33   return count / total;
34 }
35
36 function averageForArray(arr) {
37   return average(arr.length, totalForArray(arr));
38 }
39
40 // Gets the value associated with the property of an object. Needed for
41 // use with a collection method like map, hence the generic.
42 function getProperty(propertyName) {
43   return function(item) {
44     return item[propertyName];
45   }
46 }
47
```

Reactive Programming

- A programming paradigm oriented around data flows and the propagation of change

	B1	= A1+5				
	A	B	C	D	E	F
1	10	15				

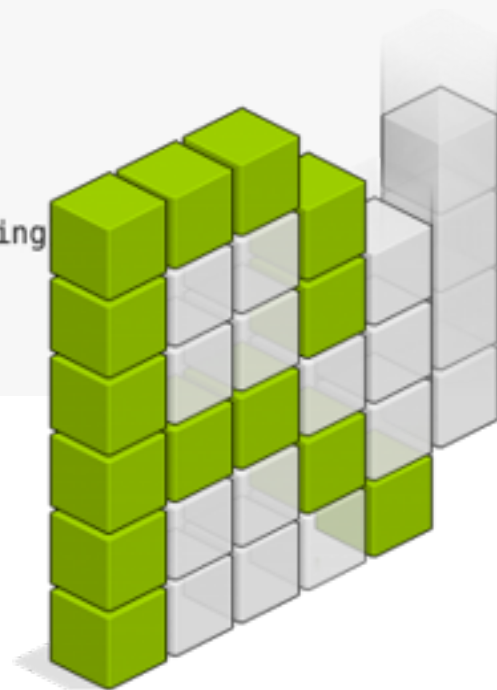
	C1		= B1+10			
	A	B	C	D	E	F
1	10	15	25			

	A1		= B1+10			
	A	B	C	D	E	F
1	20	25	35			

什么是RAC

```
let searchResults = searchStrings
    .flatMap(.Latest) { (query: String) -> SignalProducer<(NSData, NSURLResponse), NSError> in
        let URLRequest = self.searchRequestWithEscapedQuery(query)

        return NSURLSession.sharedSession()
            .rac_dataWithRequest(URLRequest)
            .retry(2)
            .flatMapError { error in
                print("Network error occurred: \(error)")
                return SignalProducer.empty
            }
    }
    .map { (data, URLResponse) -> String in
        let string = String(data: data, encoding: NSUTF8StringEncoding)
        return self.parseJSONResultsFromString(string)
    }
    .observeOn(UIScheduler())
```



MVVM为什么要用FRP

1 面向数据流

2 异常传递机制

3 减少状态量

4 处理回调

业务场景下的FRP



需求:

总价 = (房价1 x 天数 + 房价2 x 天数) x 房间数 - 优惠

时间可修改

房间数可修改

优惠具备多重优惠, 可以选择

优惠规则:

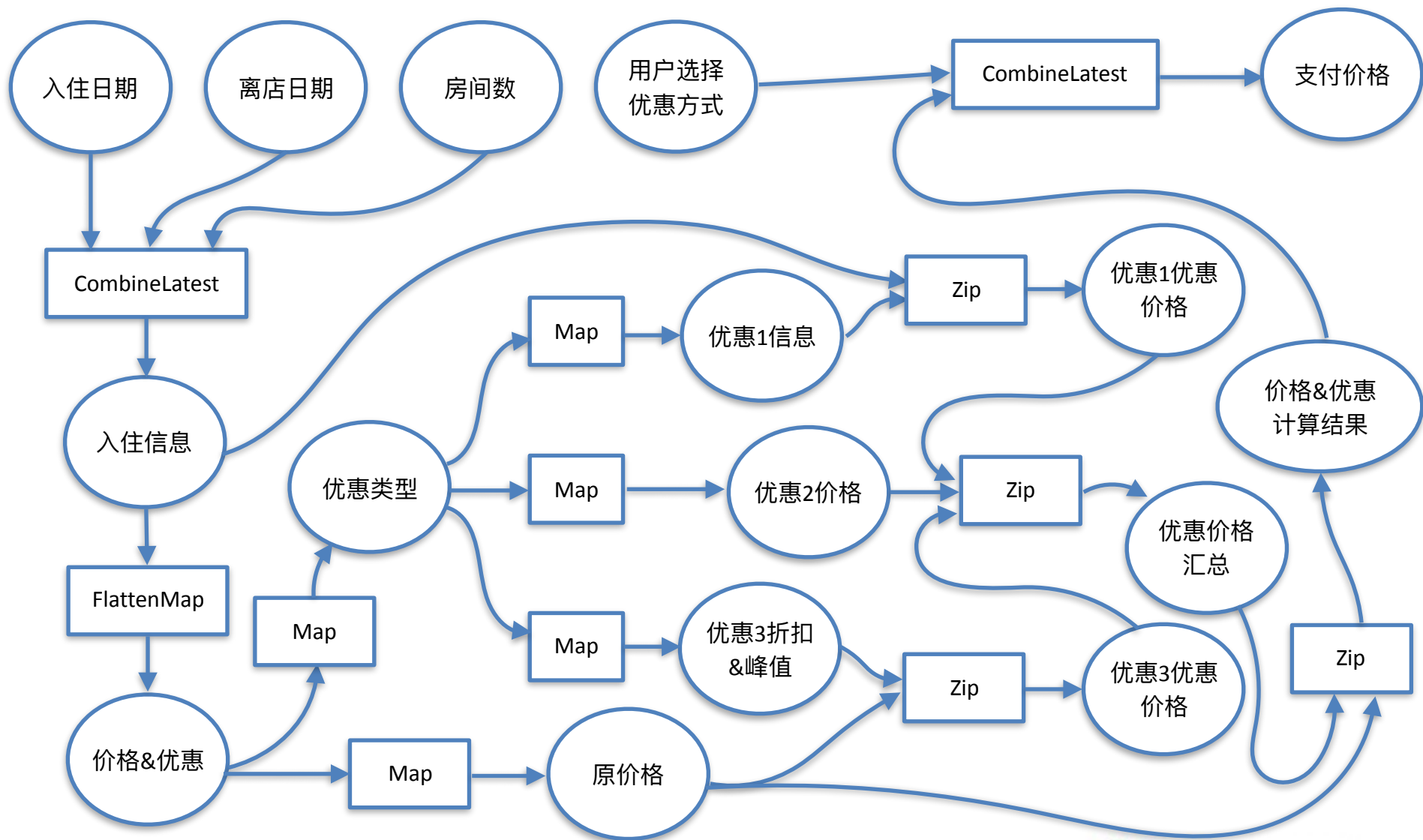
优惠类型1 = (房价1优惠 x 天数(0-3天)
+ 房价2优惠 x 天数(0-3天) x 房间数

优惠类型2 = 一口价(例如首单优惠)

优惠类型3 = 总价 x 折扣(≤ N)

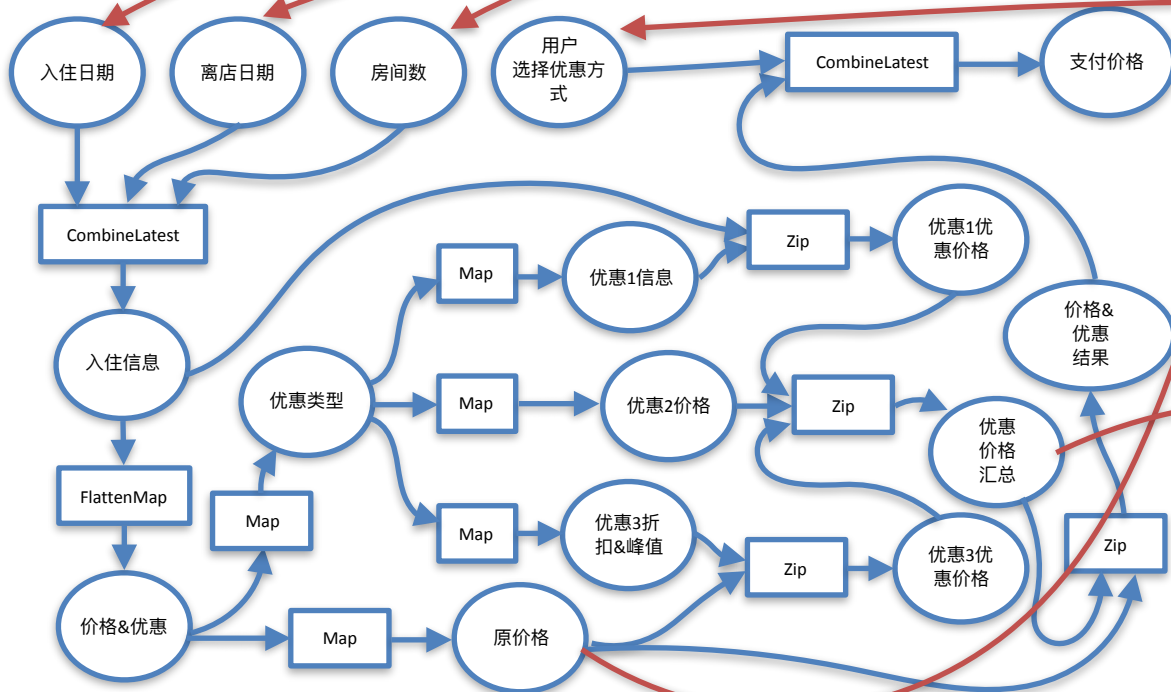
优惠类似4 = 选择抵用券 (总价 ≥ 抵用券要求)

业务场景下的FRP

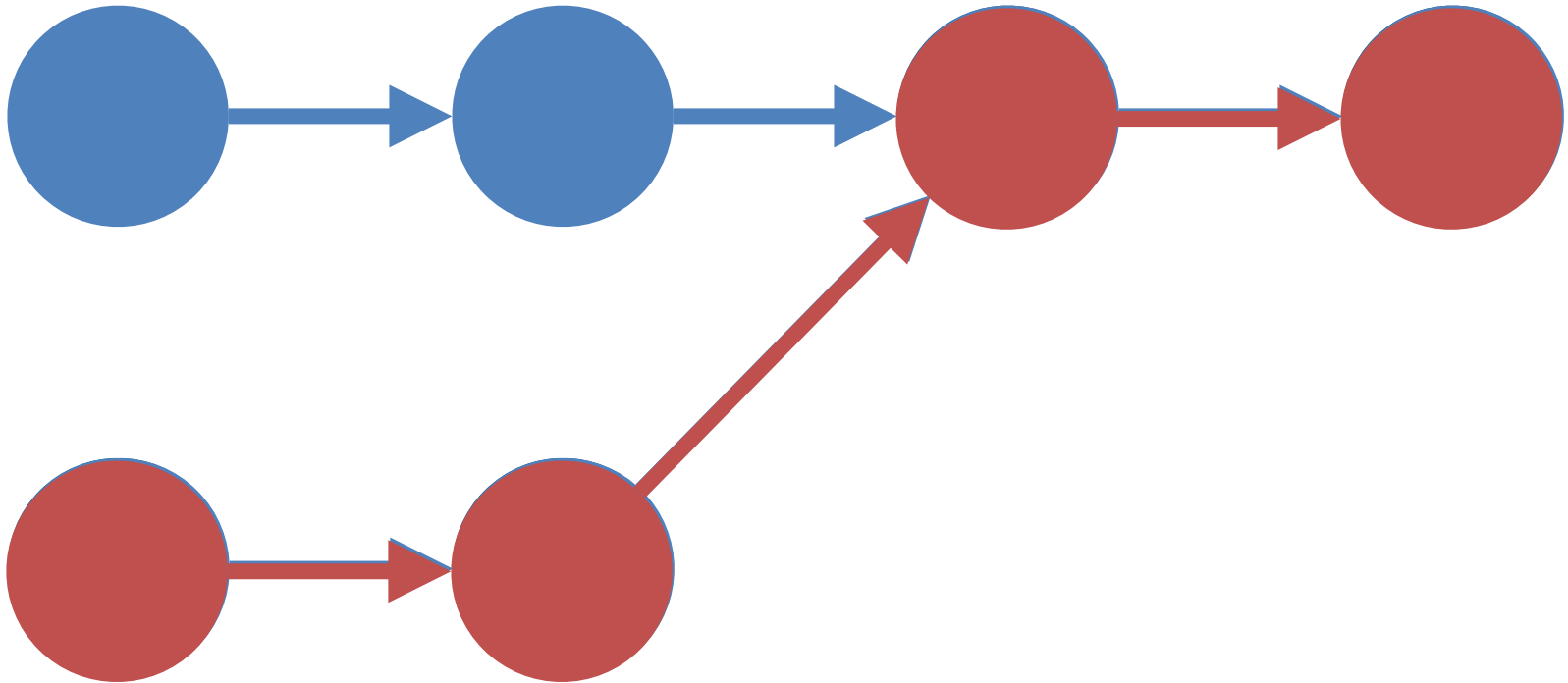


业务场景下的FRP

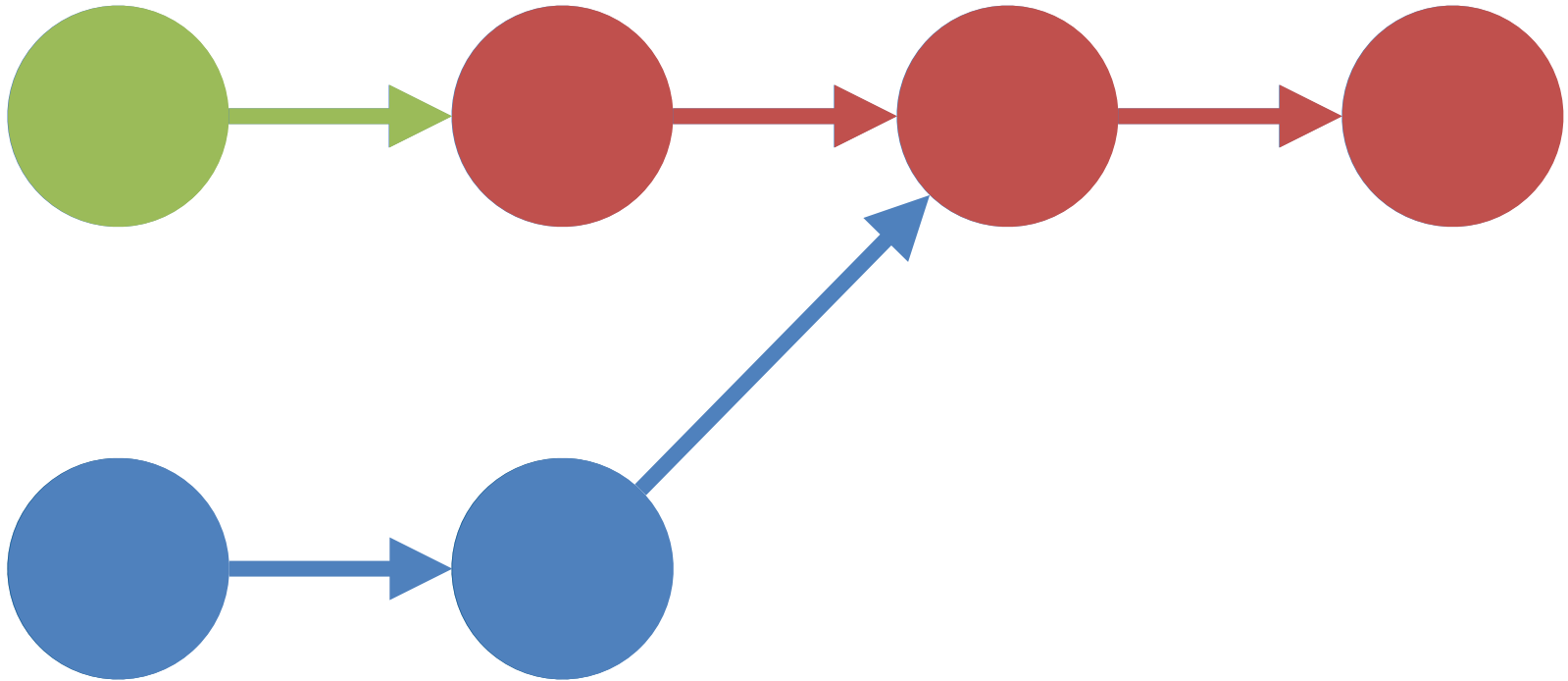
视图绑定



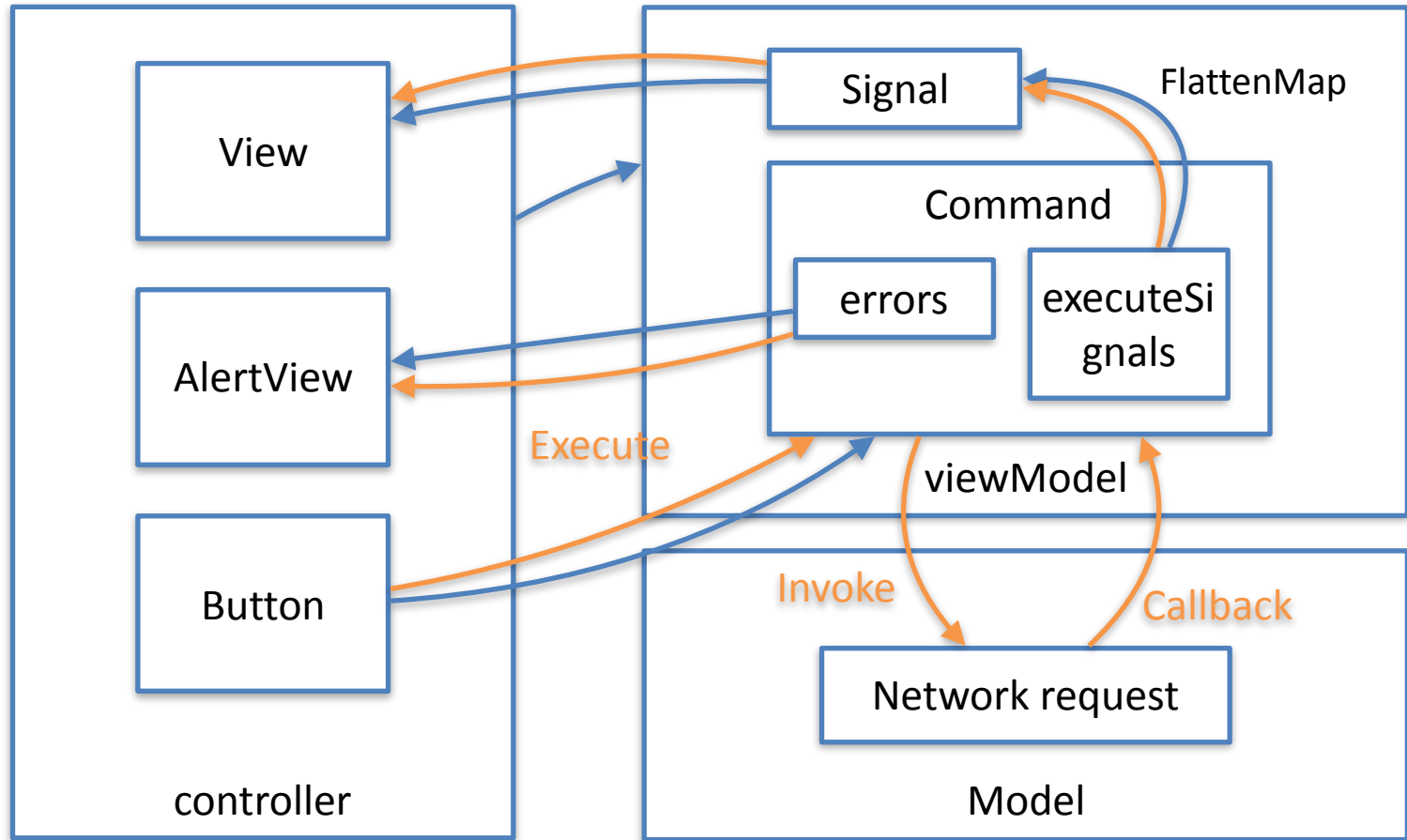
异常处理



异常处理



Command





展望

可测试性



更轻、更简、更强大

```
@weakify(self)
RACSignal *networkSignal = [[RACSignal createSignal:
    ^{RACDisposable *disposable = [RACSubscriber> subscriber] {
        AFHTTPSessionManager *manager = [AFHTTPSessionManager manager];
        NSURLSessionDataTask *task = [manager GET:@"http://www.baidu.com/?q=abc"
            parameters:nil
            success:^(NSURLSessionDataTask *task,
                NSData *responseObject) {
                [subscriber sendNext:responseObject];
            }
            failure:^(NSURLSessionDataTask *task,
                NSError *error) {
                [subscriber sendError:error];
            }
        ];
        return [RACDisposable disposableWithBlock:^{
            if [task.state == NSURLSessionTaskStateRunning
                || task.state == NSURLSessionTaskStateSuspended] {
                [task cancel];
            }
        }];
    }];
deliverOn:[RACScheduler scheduler];

RACSignal *resultSignal = [networkSignal tryMap:
    ^{id(NSData) *value,
        NSError *__autoreleasing *errorPtr} {
    return [NSJSONSerialization JSONObjectWithData:value
        options:0
        error:errorPtr];
}];
try:"BOOL(id value, NSError *__autoreleasing *errorPtr) {
    if (![value isKindOfClass:[NSDictionary class]]) {
        *errorPtr = makeErrorTitle(@"response not a dictionary");
        return NO;
    }
    return YES;
}];
try:"BOOL(NSDictionary *value, NSError *__autoreleasing *errorPtr) {
    if (![value[@"text"]]) {
        *errorPtr = makeErrorTitle(@"response doesn't have text");
        return NO;
    }
    return YES;
}];
try:"BOOL(NSDictionary *value, NSError *__autoreleasing *errorPtr) {
    if (![value[@"text"] isKindOfClass:[NSString class]]) {
        *errorPtr = makeErrorTitle(@"response doesn't have text");
        return NO;
    }
    if ([value[@"text"] containsString:@"xxx"]){
        *errorPtr = makeErrorTitle(@"xxx response doesn't have text");
        return NO;
    }
    return YES;
}];
deliverOnMainThread
doNext:^(id x) {
    @strongify(self)
    [self.savedData addObject:x];
}];
map:^id(NSDictionary *value) {
    return value[@"text"];
}];
catch:"RACSignal *(NSError *error) {
    @strongify(self)
    [self showErrorViewWithTitle:error.domain];
    return [RACSignal return:nil];
}];
ignore:nil];

[self.labelA rac_liftSelector:@selector(setText:) withSignals:resultSignal, nil];
```



```
@weakify(self)
RACSignal *networkSignal = [makeFetchSignal(@"http://www.baidu.com/",
    @[@"q": @"abc"])
    deliverOn:[RACScheduler scheduler]];

RACSignal *resultSignal = [networkSignal
    tryMap:parseJSONWithError:^(makeErrorTitle(@"response not a JSON string"))
    ]];
try:verifyWithError:(valueIsClass:[NSDictionary class],
    makeErrorTitle(@"response not a dictionary"))
try:verifyWithError:(dictionaryHasKey(@"text"),
    makeErrorTitle(@"response doesn't have text"))
] deliverOnMainThread
doNext:^(id x) {
    @strongify(self)
    [self.savedData addObject:x];
}];
map:mapToDictionaryKey:@"text"
try:verifyWithError:(valueIsClass:[NSString class],
    makeErrorTitle(@"response's text is not a string"))
] catch:"RACSignal *(NSError *error) {
    @strongify(self)
    [self showErrorViewWithTitle:error.domain];
    return [RACSignal return:nil];
}];
ignore:nil];

[self.labelA rac_liftSelector:@selector(setText:) withSignals:resultSignal, nil];
```

锻炼开发人员的思维



Geekbang> InfoQ
极客邦科技

THANKS!