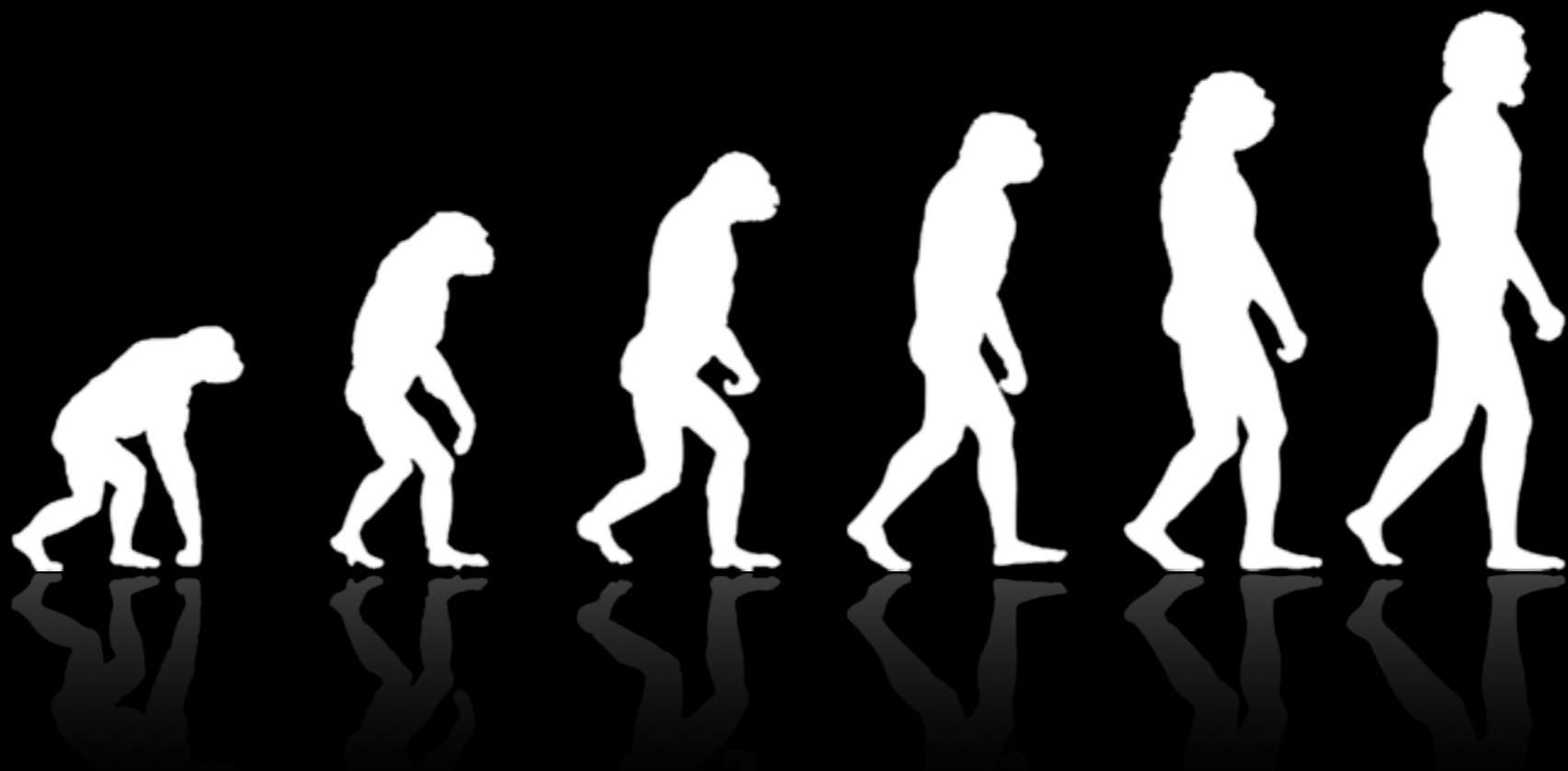


手机天猫解耦之路

天猫·伯灵 | @gaosboy

手机天猫解耦之路

天猫·伯灵 | @gaosboy



解耦之路，就是**架构进化**之路



- 什么在推动进化
- 架构怎么进化
- 开发模式在升级
- 解耦离不开工具



什么在推动进化





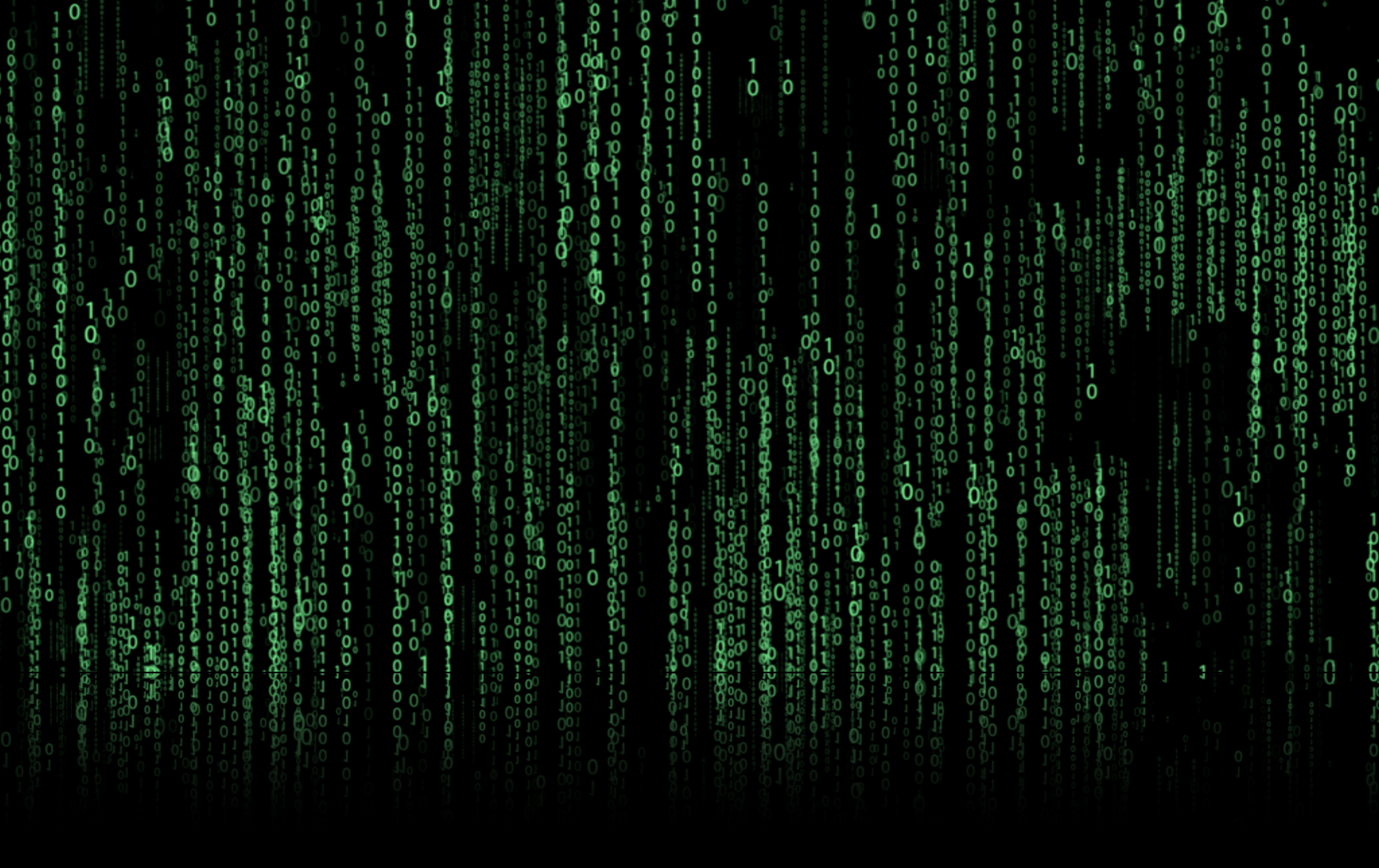
业务升级





团队规模 & 合作方式





代码规模 & 工程规模





新技术



架构怎么进化





升级开发模式





各维度解耦





完善工具集





干货预警



iOS



开发模式在升级





一个工程





一个工程 & 多个模块



出现团队分工 • 跨团队合作

- 按分工拆分**模块**
- 用伟大的**文件夹**组织代码
- 开心的在**一个**工程里玩耍





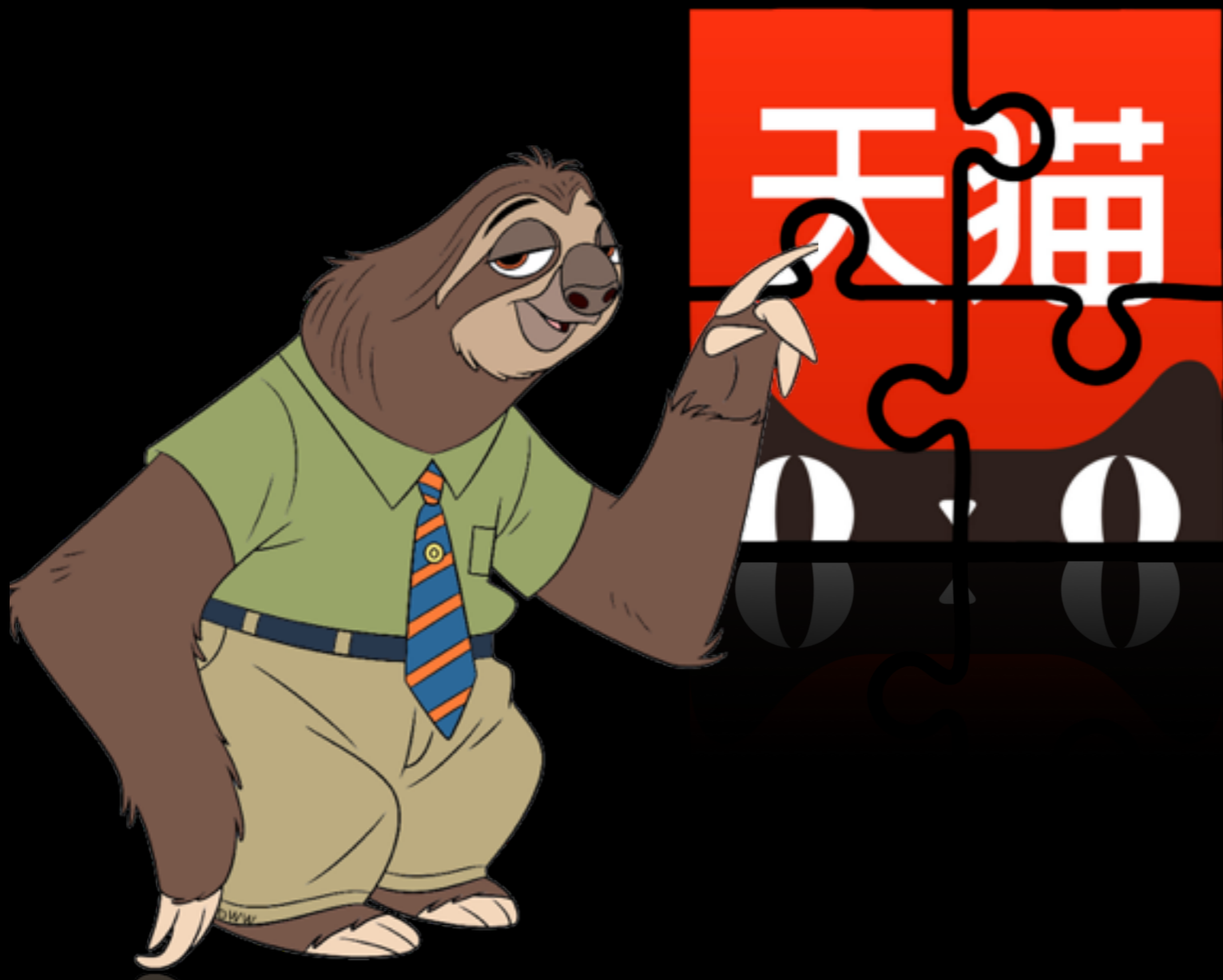
一个工程 & 多个插件



团队变多 • 工程变大

- 把模块**插件化**
- 插件代码**独立**管理
- 引入**依赖管理**设施
- 仍然在**一个**工程里玩耍





一个工程 & 多个插件



多个工程 & 集成发布



慢崩溃了 • 分而治之

- 把插件**Bundle化**
- Bundle**独立**编译&运行
- 强制**解除**依赖
- 工程**分而治之**





Bundle化解耦工程



Bundle独立!

- 按功能横向切分，按业务纵向切分
- 每一个Bundle可以独立编译&运行
- 解放开发环境的繁冗依赖
- 避免集成过程的全量编译

一步一步一个脚印

- 代码独立，先从形式上解耦
- 独立代码工程化，为独立运行打下基础
- 梳理依赖关系，独立工程可编译
- 放弃源码依赖，提速集成编译

解耦离不开工具



什么耦合令人痛苦

- 界面耦合
- 依赖耦合
- 工程耦合



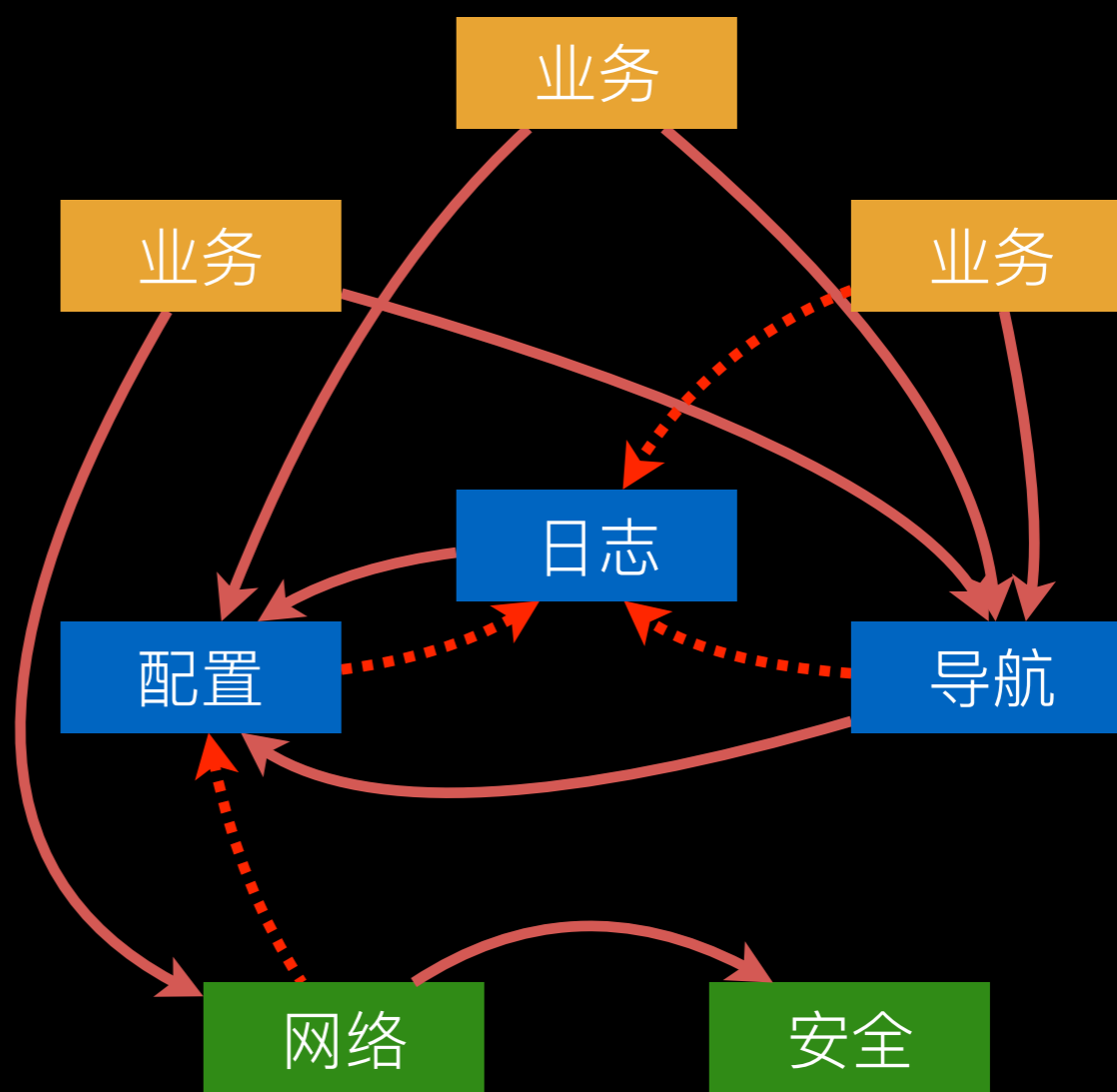


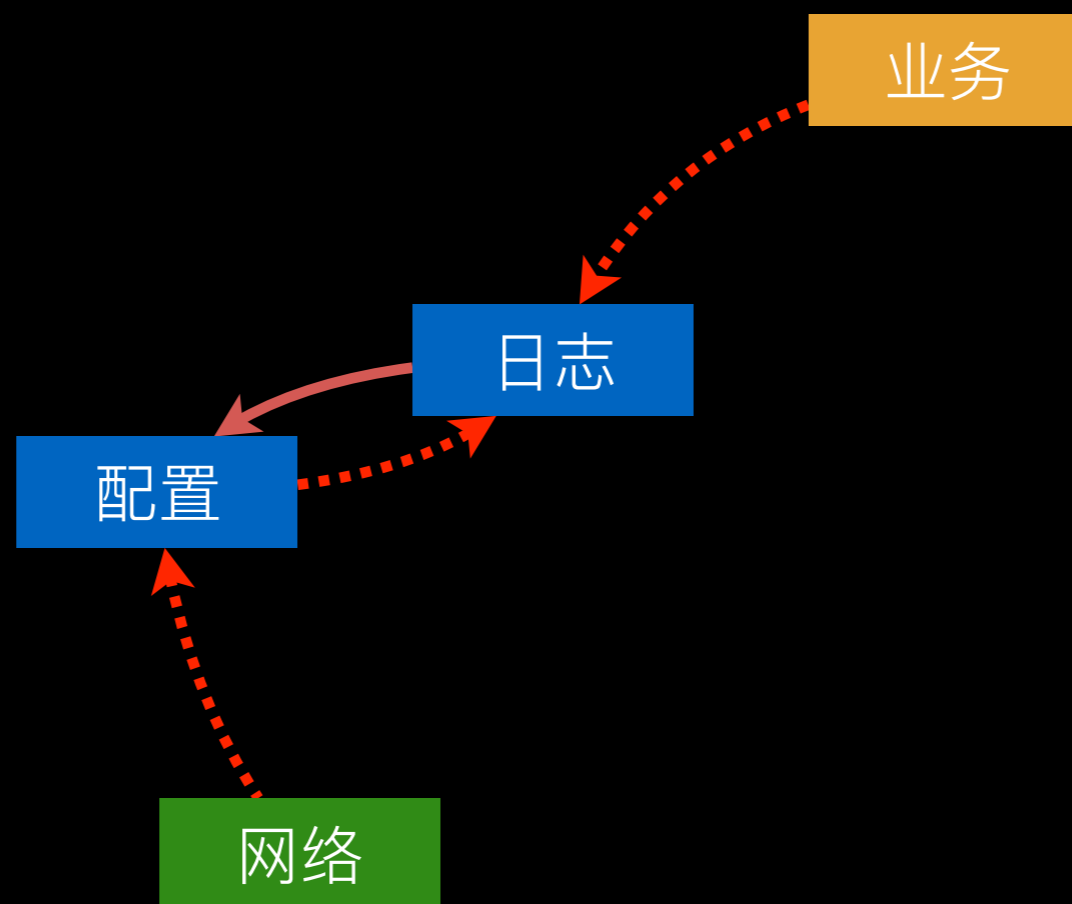
Beehive框架解耦依赖



历史上的依赖关系纷繁复杂

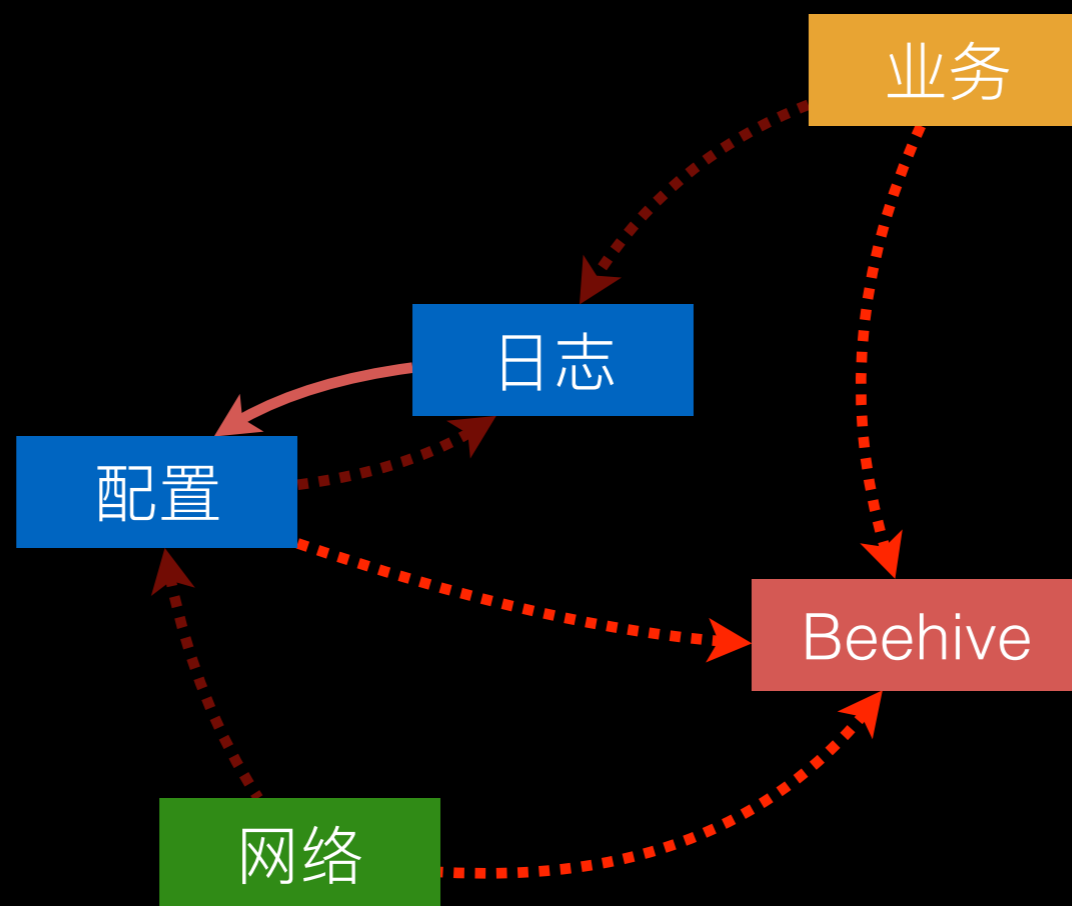
- 模块循环依赖
- 层间反向依赖
- 非强功能依赖





拿掉一些，看得清楚些



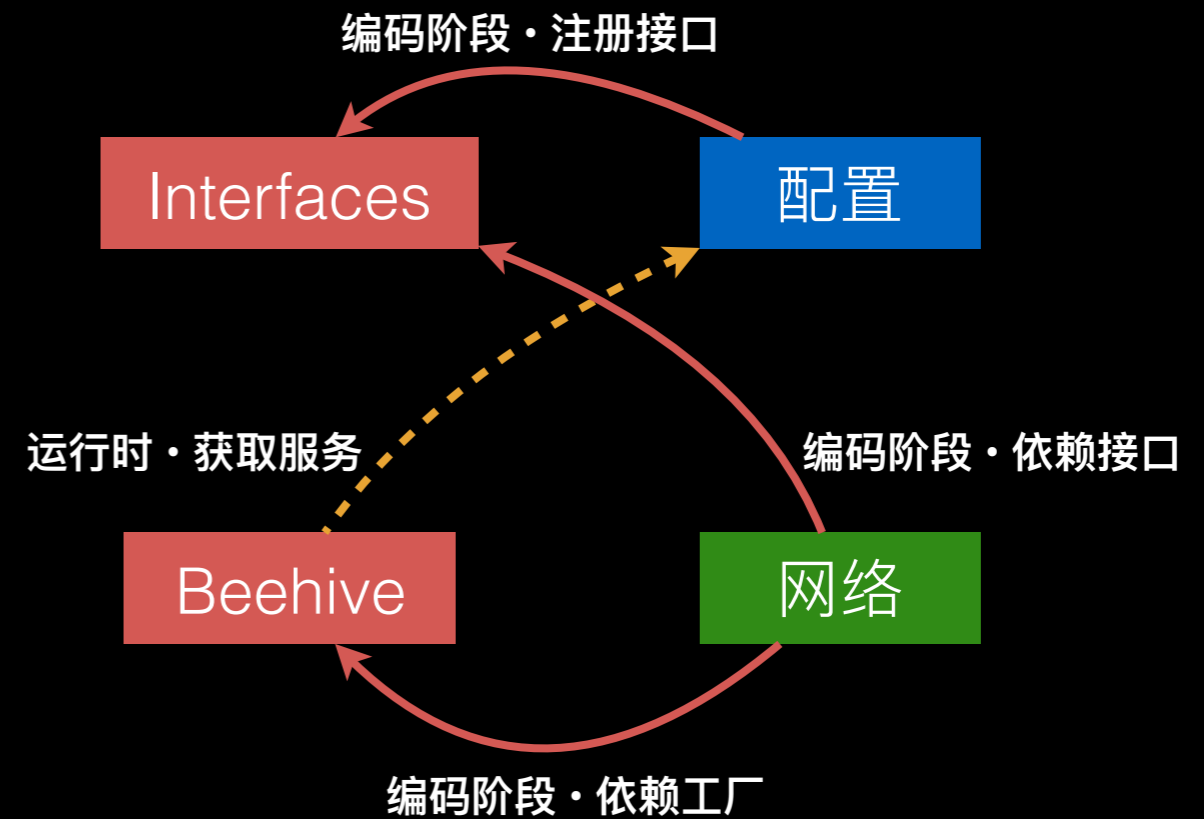


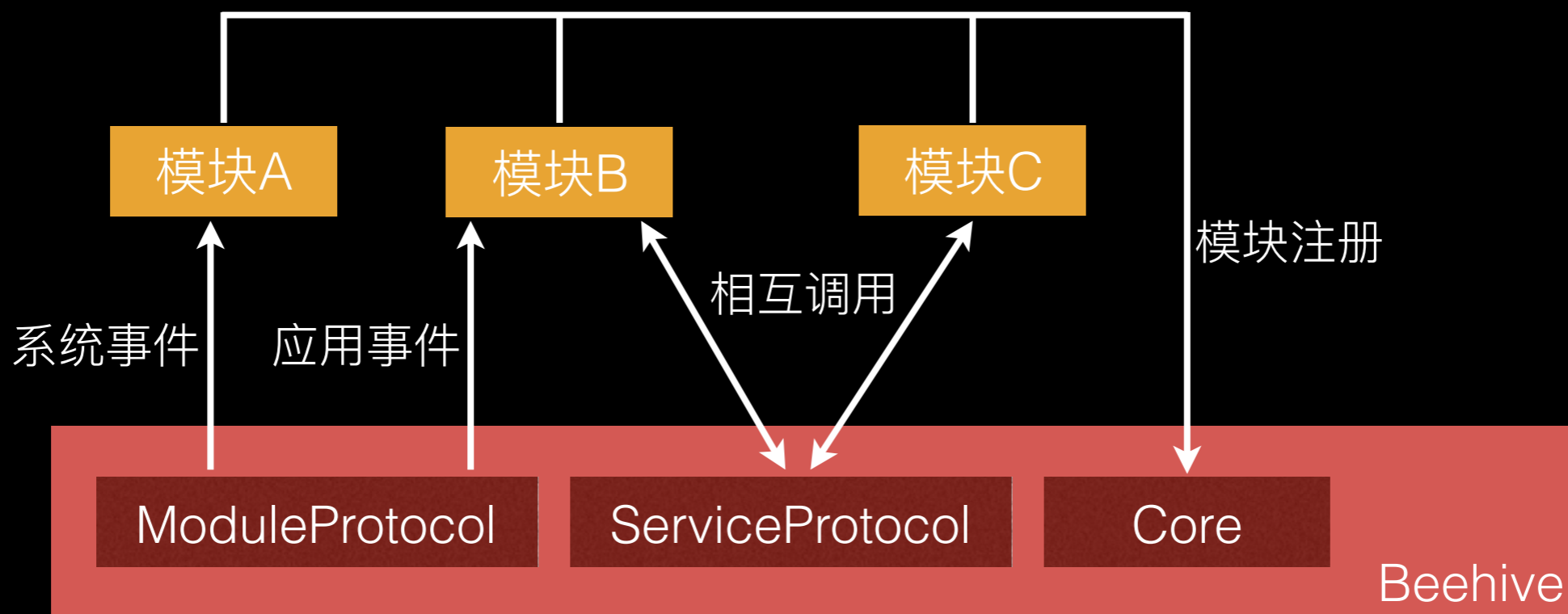
Beehive解决这些问题



Beehive

- 服务声明接口并**注册**到公共区
- 调用方依赖**公共区**
- 根据接口，使用**工厂**获取服务





Beehive架构



系统事件

- Will Launch
- Did Launch
- Will Terminate
- Did Become Active
- Will Enter Background
- Will Resign Active
- Did Enter Background



扩展事件

- Mod Setup
- Mod Init
- Mod Splash





统跳协议解耦界面



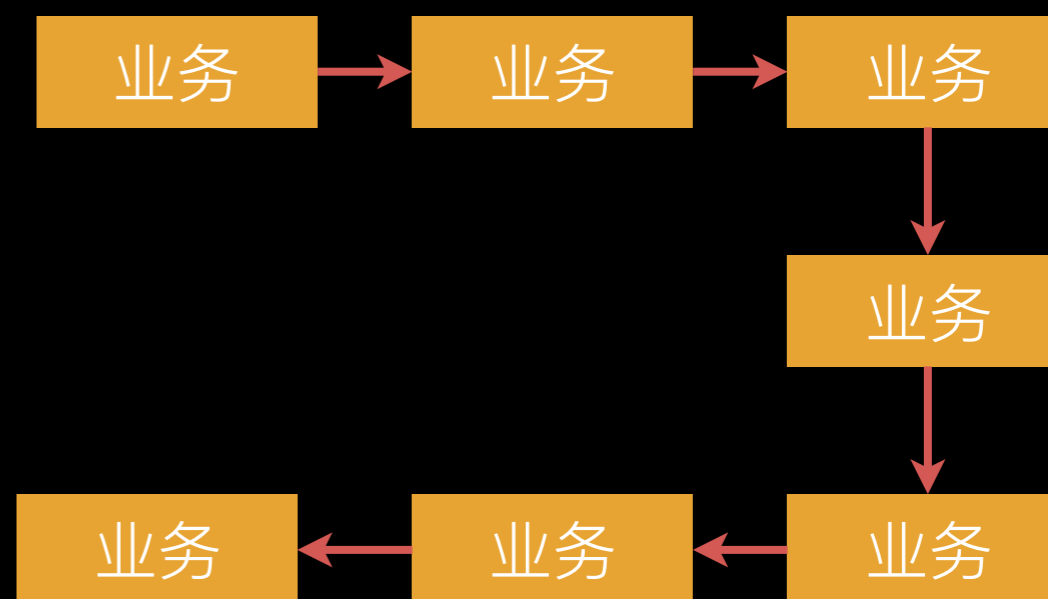
```
- (void)click
{
    UINavigationController *nextVC = [[UINavigationController alloc] init];
    [self.navigationController pushViewController:nextVC animated:nextVC];
}
}
```

硬编码，强依赖



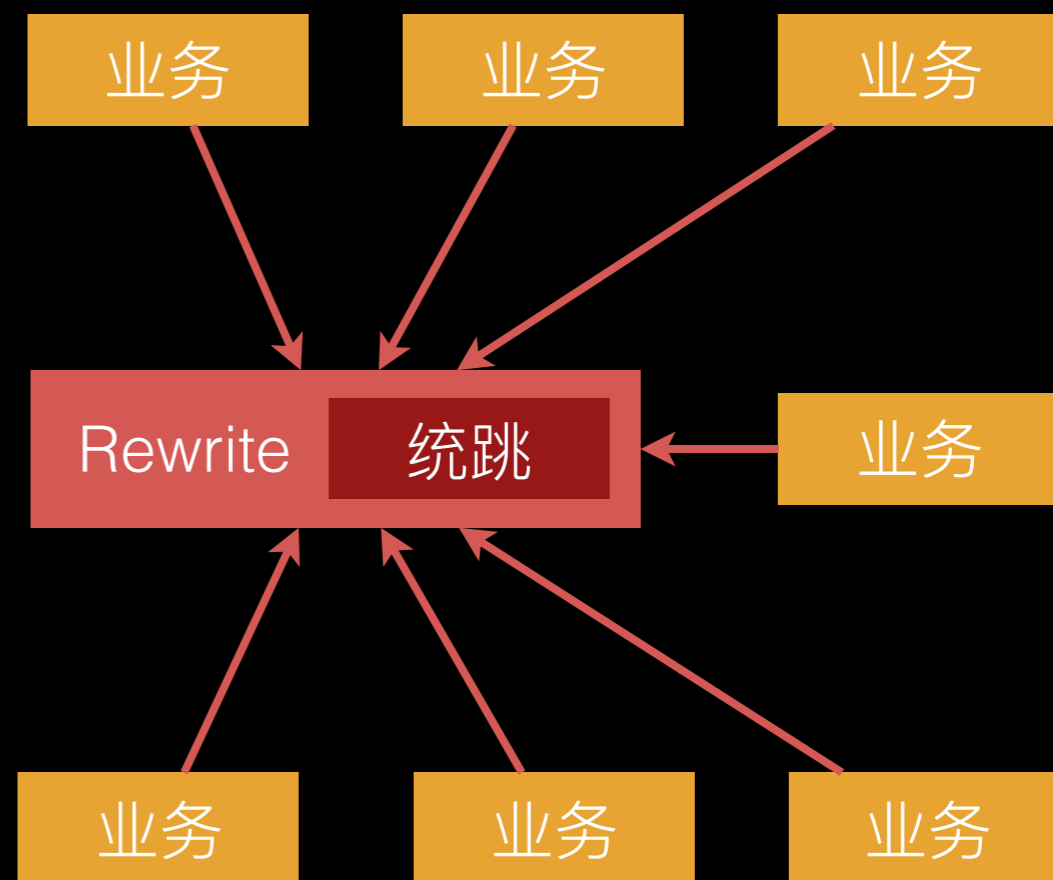
传统的界面跳转

- 业务链路上的全部模块强依赖
- 无法实现业务链路灵活调整
- 降级无能



使用统跳解耦界面

- 导航器 — **URL-Service**的映射
- 默认提供一个**页面跳转**Service
- 业务把公开的页面**注册**到统跳
- 根据**URL**，导航器统跳执行跳转



服务名称, page.tm
表示页面跳转

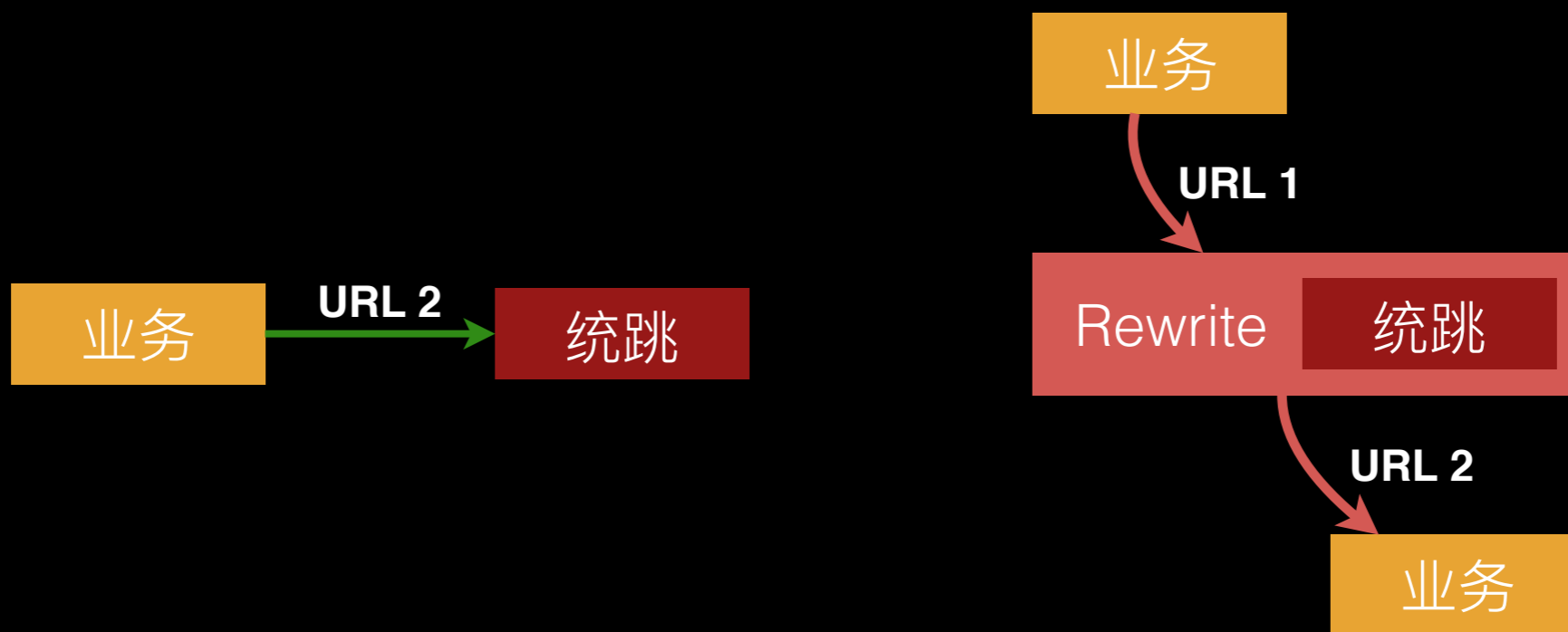
tmall://page.tm/search?q=iPad&sort=price

私有协议标示,
表明这是一个
Native服务

服务内部参数, 在条服
务里path标示页面,
query承载业务参数

URL规则





调用方使用**习惯的**URL 1 & Rewrite转换为**注册**的URL 2

模式串	转换串	标记位
<code>^(?:https?:)?\W/detail(?:.m)?.tmall.com\?item.htm\?(.*)</code>	<code>tmall://page.tm/itemDetail?\$1</code>	

把一切变成**字符串**之间的转换， 召唤万能的**正则**



```
- (void)click:(NSURL *)url
{
    [[TMURLNavigator sharedInstance] aliOpenUrl:url withTarget:self withParams:@{ }];
}
```

使用导航器解耦



- 以字符串为核心，强解耦
- URL给动态性提供最大可能性
- 字符串没有约束，不准确
- URL参数类型严重受限
- 以interface为核心，弱解耦
- 接口依赖基本没有动态可能
- 接口定义强依赖，准确
- Native Code支持全部参数类型





“Premature optimization is the root of all evil.”

– *Donald Knuth*





- 业务要求界面链路的强动态性
- 模块强依赖，开发阶段痛苦



Rewrite引擎

CocoaPods

Atlas

Maven

统一跳转器

Beehive

Rewrite Bundle化开发

独立编译&运行

业务

架构升

升级

级



<http://pingguohe.net>

手机天猫技术大赏 • 苹果核



Q & A

THANKS