

微信Android模块化架构重构实践

carlguo(郭锐)

腾讯高级工程师



CNUTCon 2017

全球运维技术大会

上海·光大会展中心大酒店 | 2017.9.10-11

智能时代的新运维

大数据运维
安全
SRE
DevOps
Kubernetes
Serverless
游戏运维
AIOps
智能化运维
基础架构
监控
互联网金融



主办方

Geekbang > InfoQ

极客邦科技



实践驱动的IT教育



<http://www.stuq.org>

斯达克学院 (StuQ)，极客邦旗下实践驱动的IT教育平台。通过线下和线上多种形式的综合学习解决方案，帮助IT从业者和研发团队提升技能水平。



10大职业技术领域课程

SPEAKER INTRODUCE

郭锐 腾讯高级工程师

- 10年本科毕业于华南理工大学并入职腾讯，早期作为合作部门参与微信国际化开发，2013年转入微信 Android 团队。
- 主要负责微信 Android 端的基础体验优化，开发了 WxSVG 和 Vending 框架来提升微信终端视觉和流畅性体验。
- 目前专注在微信 Android 客户端的架构重构和设计，推动模块化开发，追求提升开发效率和研发质量。



TABLE OF CONTENTS 大纲

- 回顾，微信Android架构历史
- 问题又来，模块化之后的“坑”
- 解决问题，重塑模块化
- 思考，取舍和选择
- 代码之外，架构之内

微信Android架构回顾



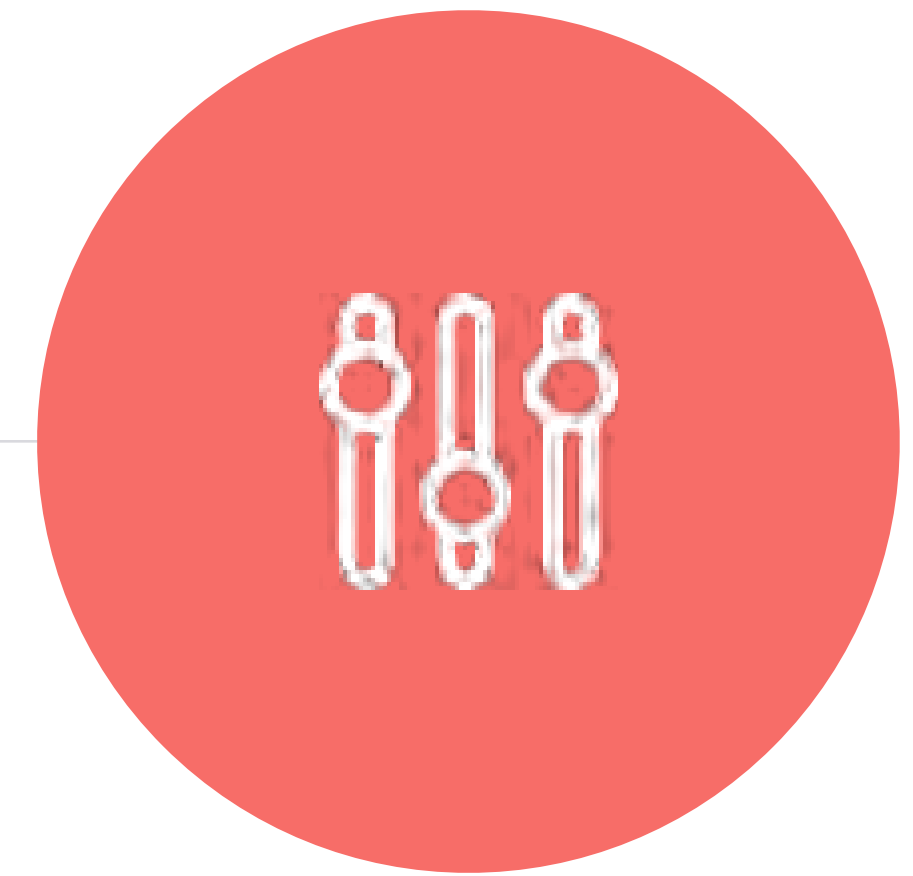
v1.x

简单分层架构



v2.x

多进程架构



v3.x

多子工程并行开发架构

微信Android架构回顾

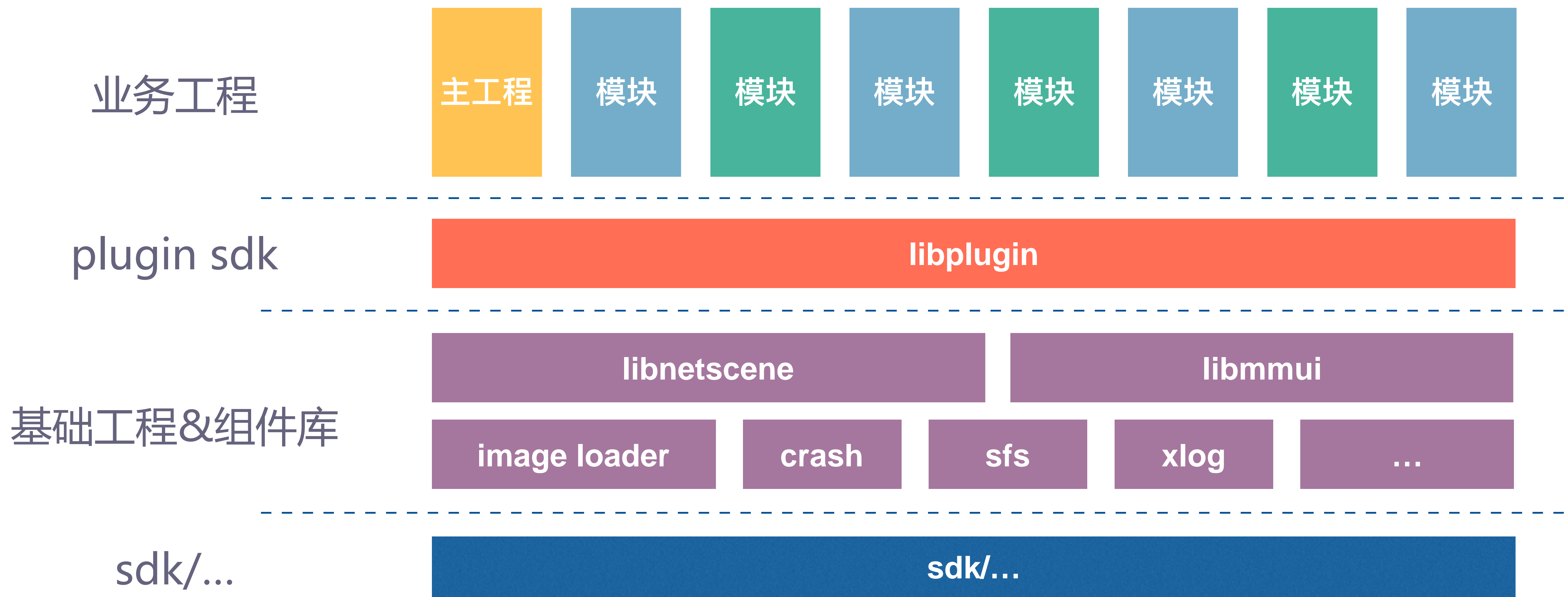
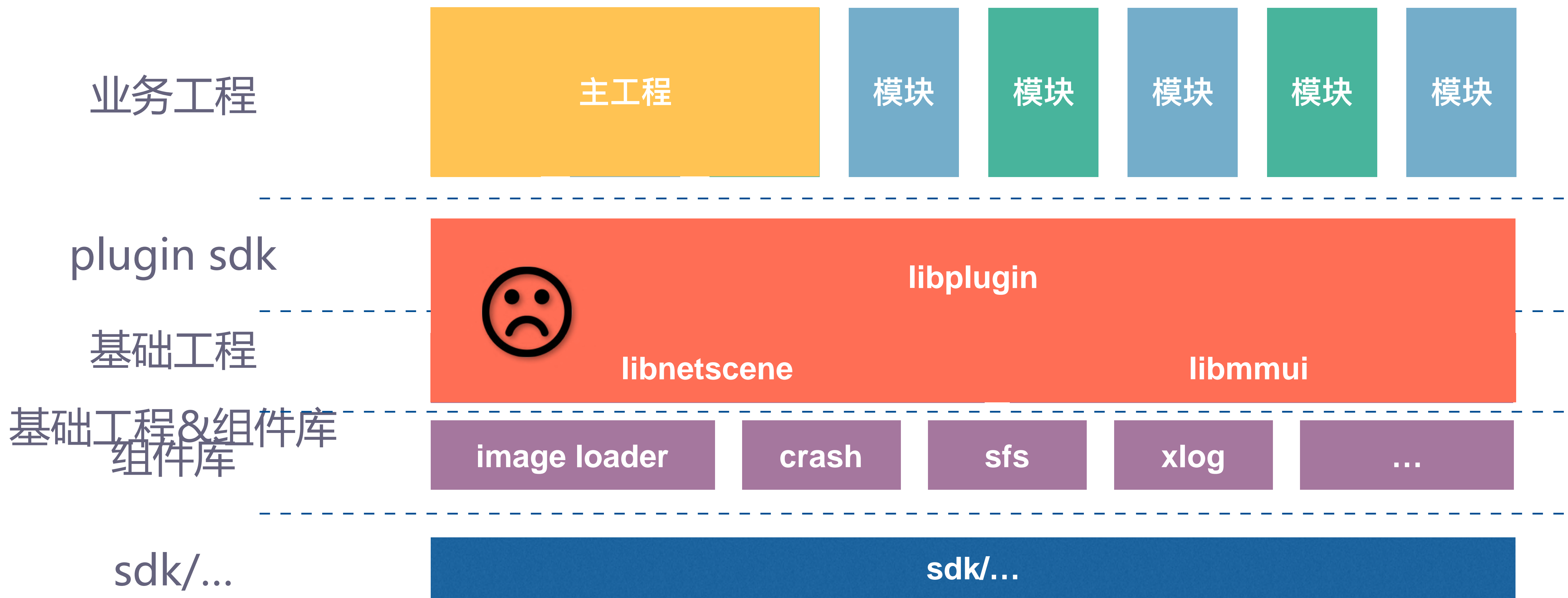


TABLE OF CONTENTS 大纲

- 回顾，微信Android架构历史
- 问题又来，模块化之后的“坑”
- 解决问题，重塑模块化
- 思考，取舍和选择
- 代码之外，架构之内

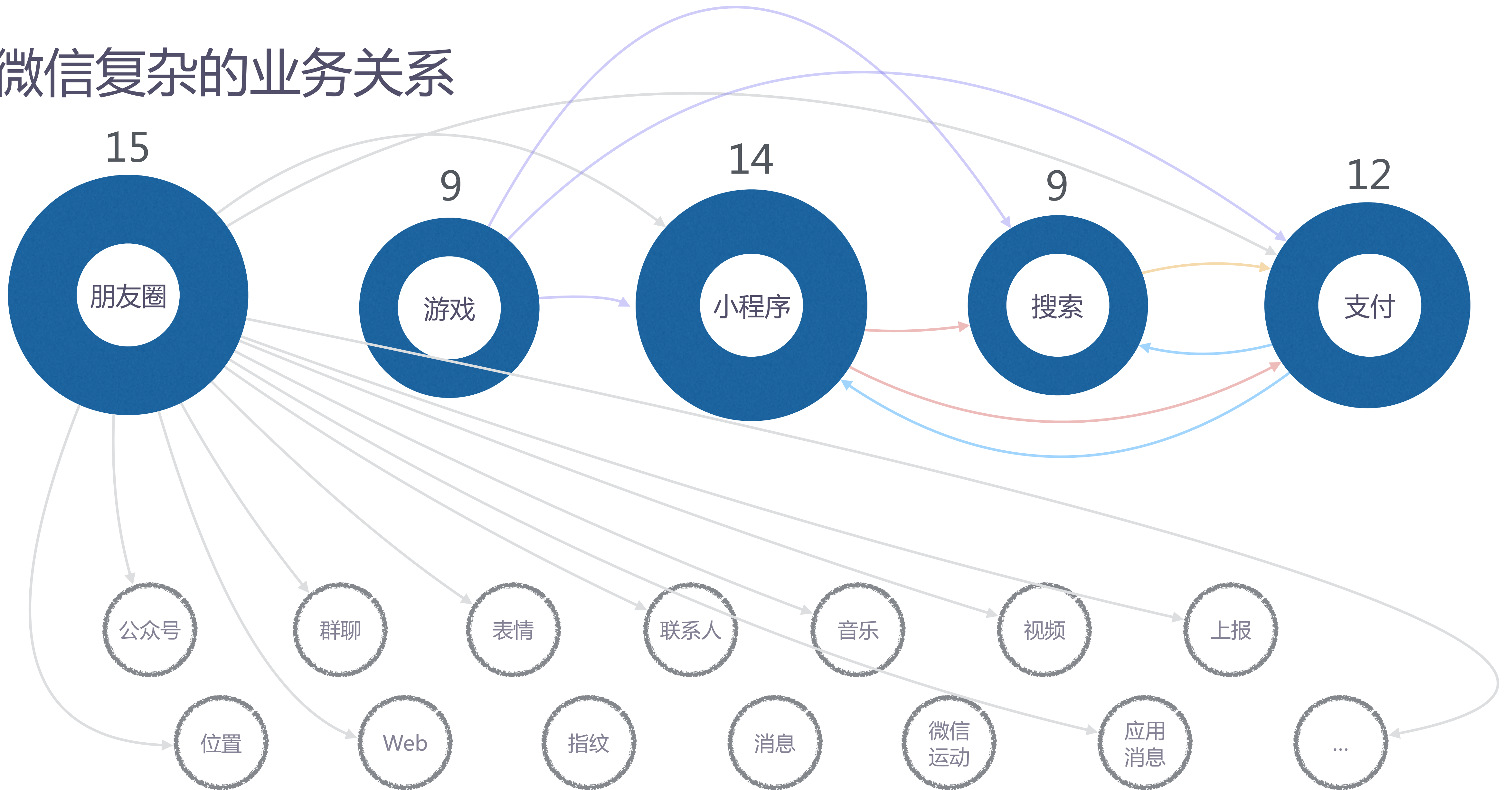
原本好好的架构出了什么问题？



问题出在哪？

- 我们在思考，模块化之后，这些问题是怎么来的？
- 先来看一张图

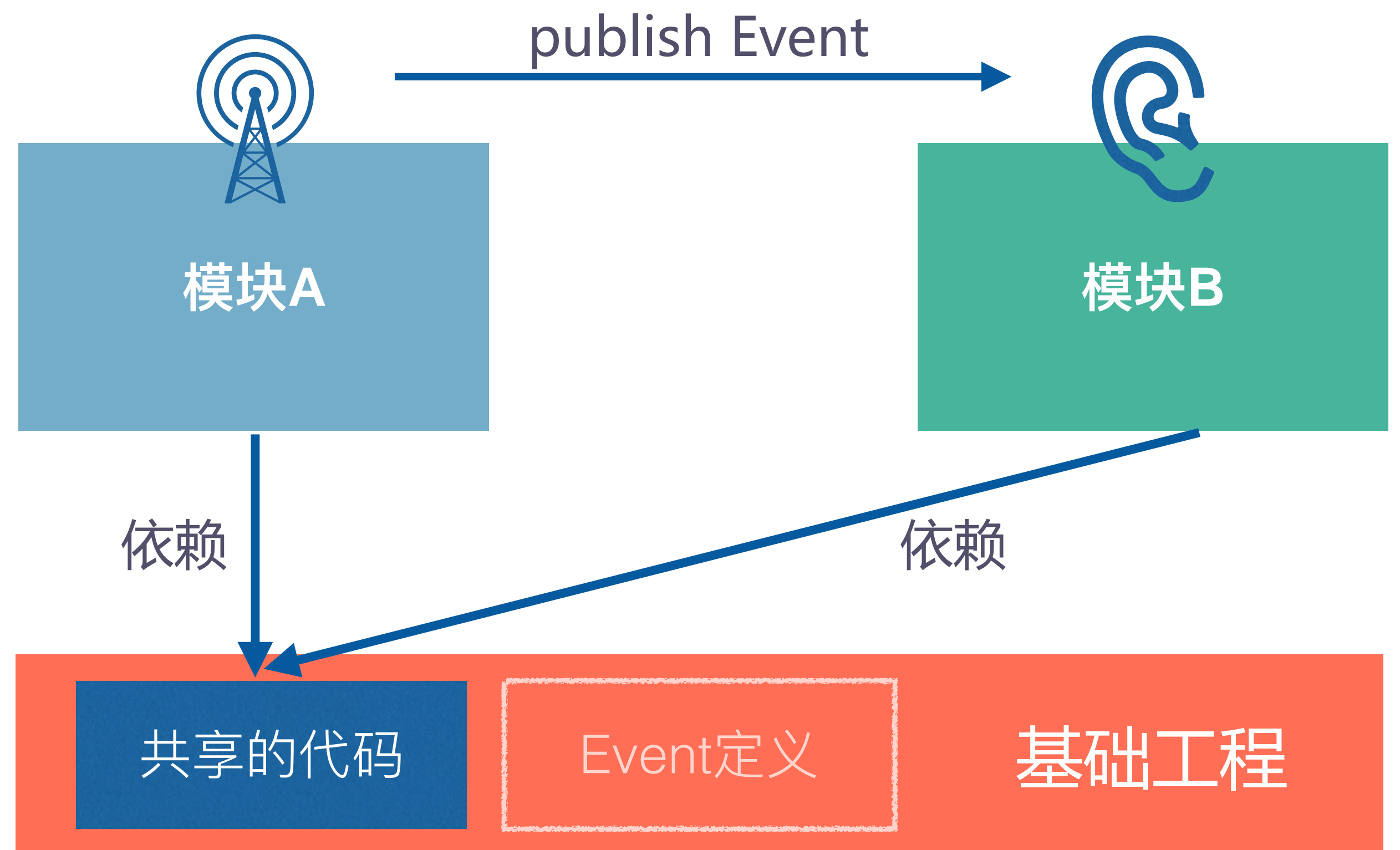
微信复杂的业务关系



问题出在哪

- 基础工程的膨胀

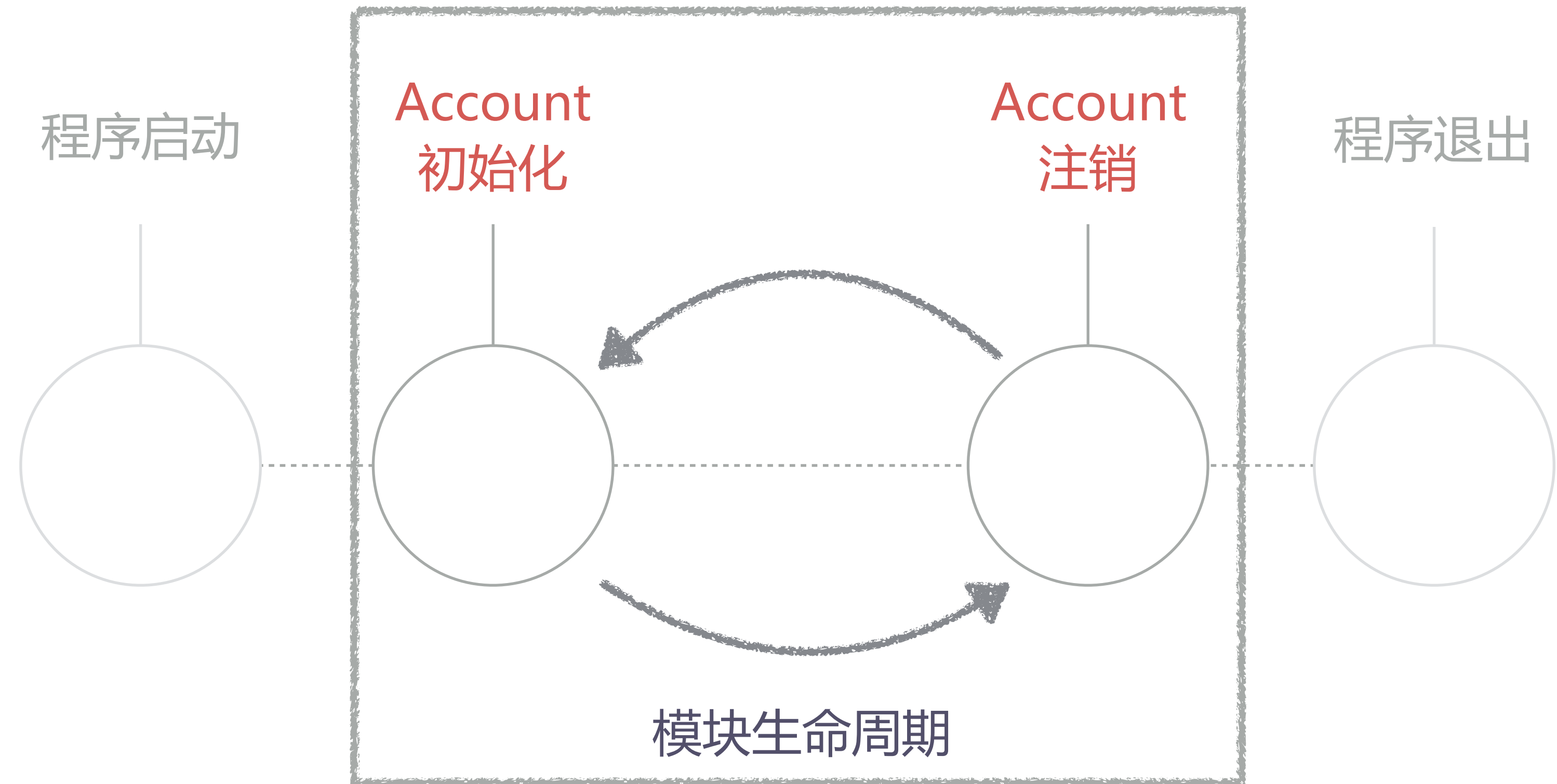
- 使用Event总线作为通信的手段，为了共享Event类型，Event通常被下沉到基础工程中
- 当需要共享一些数据结构，例如ChatroomMember，一样需要下沉代码
- 更进一步，一些业务代码需要被共享，例如ChatroomStorage会被很多业务使用，那么Storage和相关的Logic类，一起下沉到“基础工程”中
- 越来越多的代码需要被“共享”，下沉...



问题出在哪

- 主工程膨胀的原因

- 主工程业务功能的增加
- 模块的生命周期设计存在问题
- 进程启动的逻辑全都在主工程



问题出在哪

- 模块的“代码边界”有点问题
 - 由于gradle module的编译性能问题，我们将所有的工程都对接在一个module上。
 - 事实证明，任何单纯的编码约定，即使对此熟知的团队都无法保证代码不被破坏。
 - 被业务代码“逐渐污染”的基础工程，内部依赖同样错综复杂。

模块化之后的“坑”

- 面对这些问题，我们意识到整个架构的代码正在不断被腐蚀
- 这样的代码我们是否还能“忍忍”??

重构契机

- 硬件同学希望跟进微信最新组件、协议、服务
- 精简代码，移植业务模块，快速实现类似“微信相册”这样的产品功能
- 这不是“忍忍”就能解决的问题

TABLE OF CONTENTS 大纲

- 回顾，微信Android架构历史
- 问题又来，模块化之后的“坑”
- 解决问题，重塑模块化
- 思考，取舍和选择
- 代码之外，架构之内

目标

重塑模块化

拆解目标

改变通信方式

重新设计模块

约束代码边界

改变通信方式

重新设计模块

约束代码边界

通信方式

- 不只使用事件总线或协议通信，使用“SDK”的形式暴露接口
 - 事件总线并不适合所有的通信方式，更适合广播，可作为补充
 - 协议通信定义繁琐，适合需要序列化、有隔离的场景
 - 配合IDE补全，开发方便

通信方式



注册IService

```
regService(IMyService.class, myService);
```

访问IService

```
service(IMyService.class).doServiceWork();
```

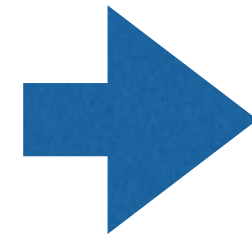
通信方式——简便的接口暴露

- 方便的接口暴露
 - 不用每个模块都手动分两个工程
 - 接口类的代码文件，和实现代码可以放在一起，但实现类并不会被外面的工程“引用”
- 我们的方式——“.api”化

通信方式——简便的接口暴露



IContactStorage.java



IContactStorage.api



table.db

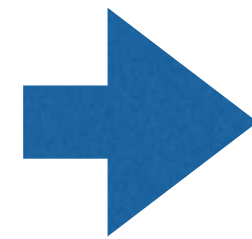
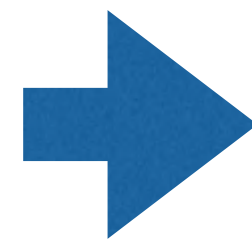


table.db.api



wechat.proto



wechat.proto.api

通信方式——简便的接口暴露

settings.gradle

```
include_with_api(":plugin-messenger")
```

build.gradle

```
dependencies {  
    compileApi(":plugin-messenger")  
}
```

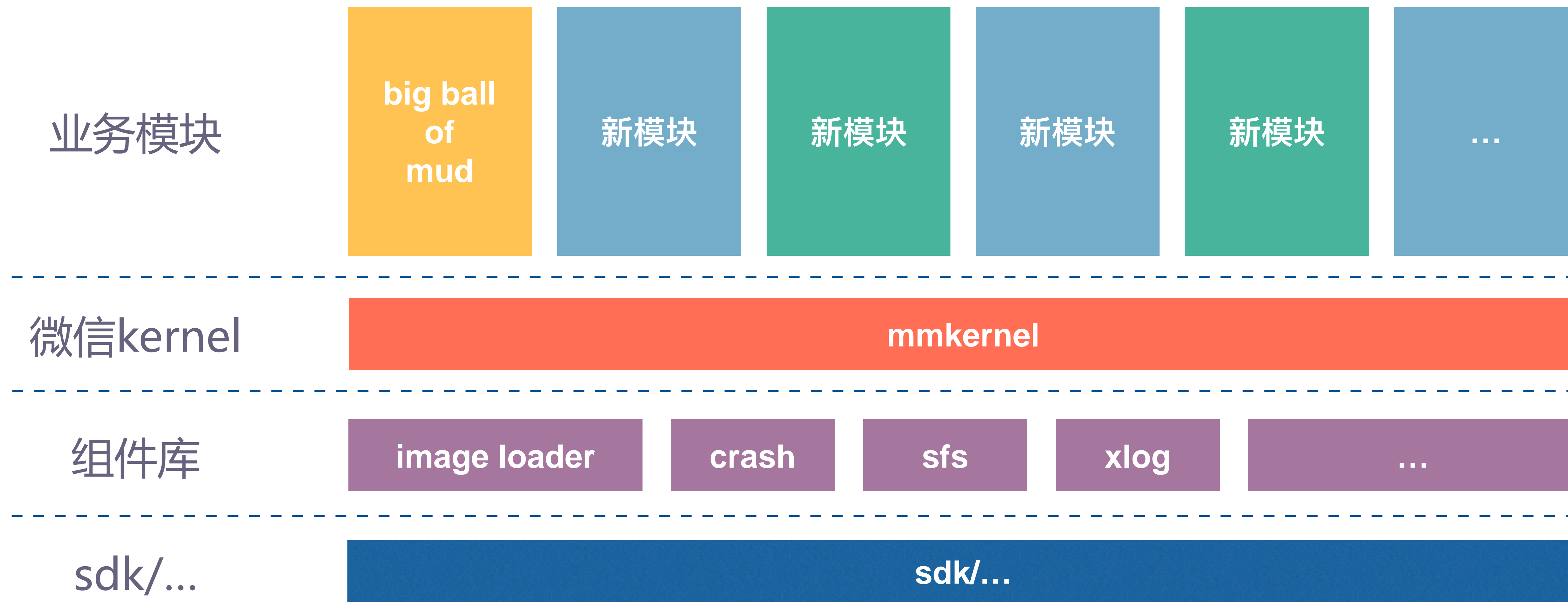
设计目标

改变通信方式

重新设计模块

约束代码边界

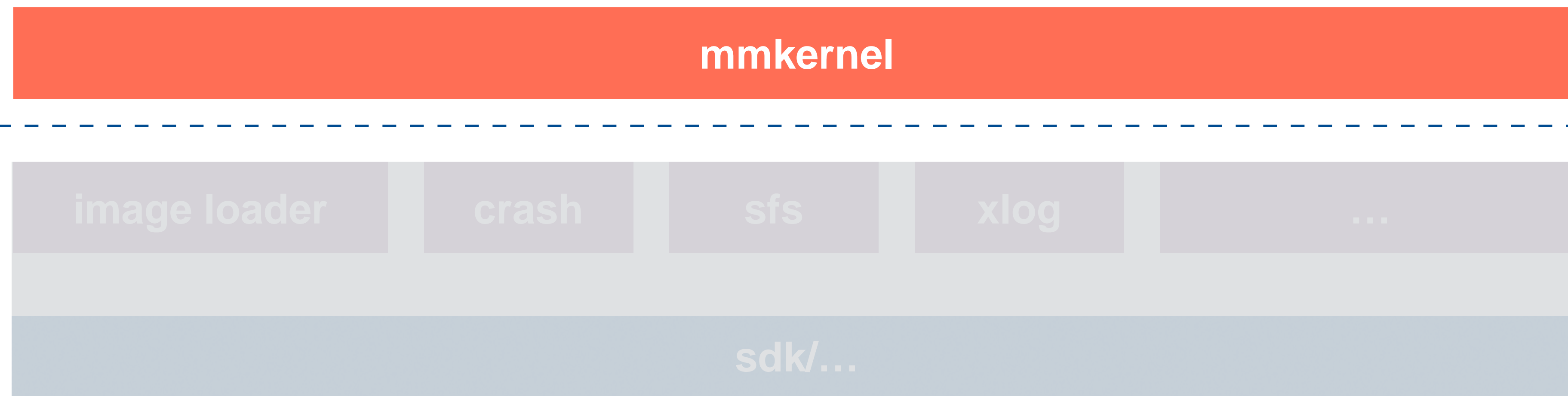
重新设计模块——分层结构改造



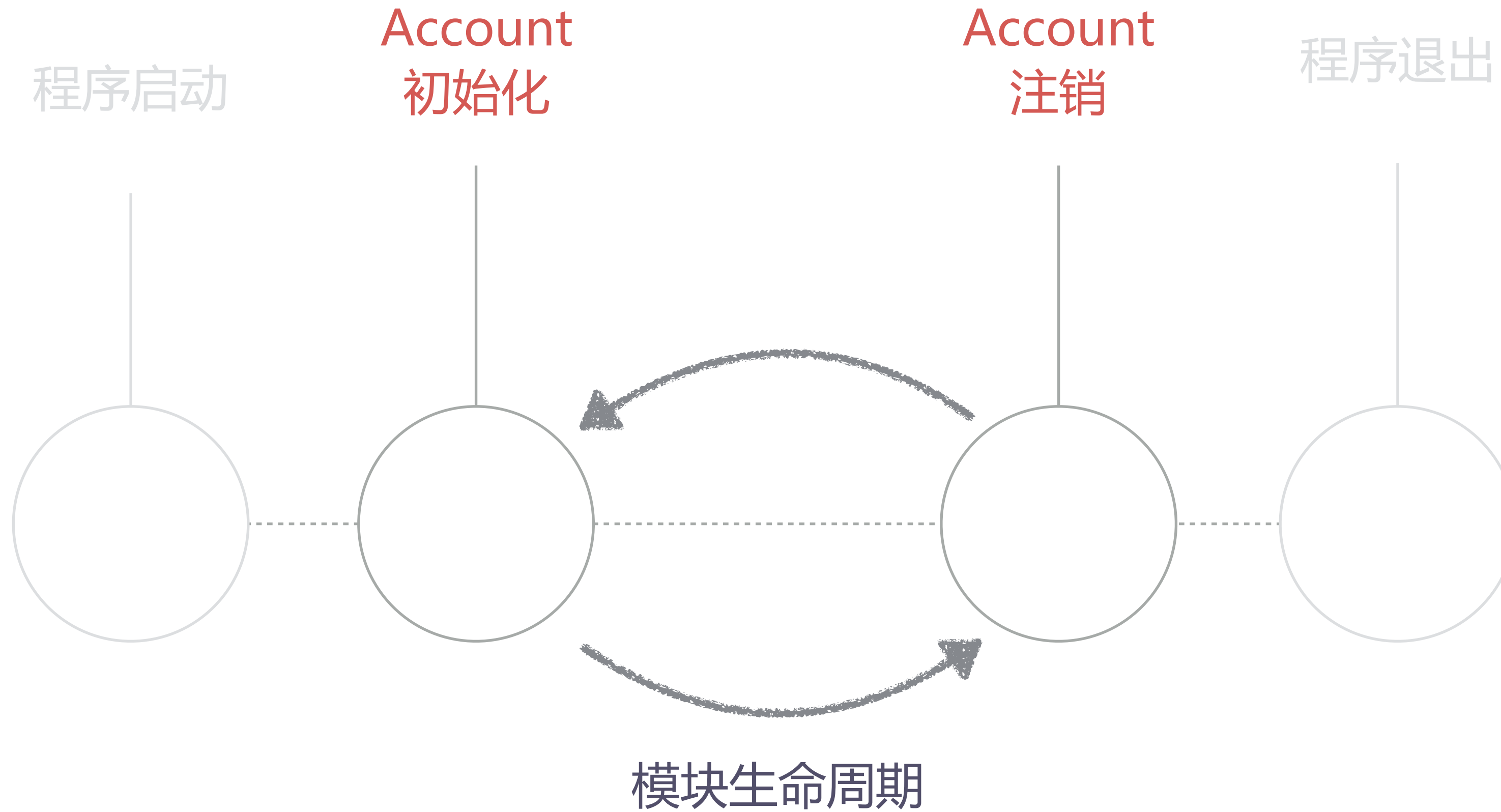
重新设计模块——最小化kernel工程的职责



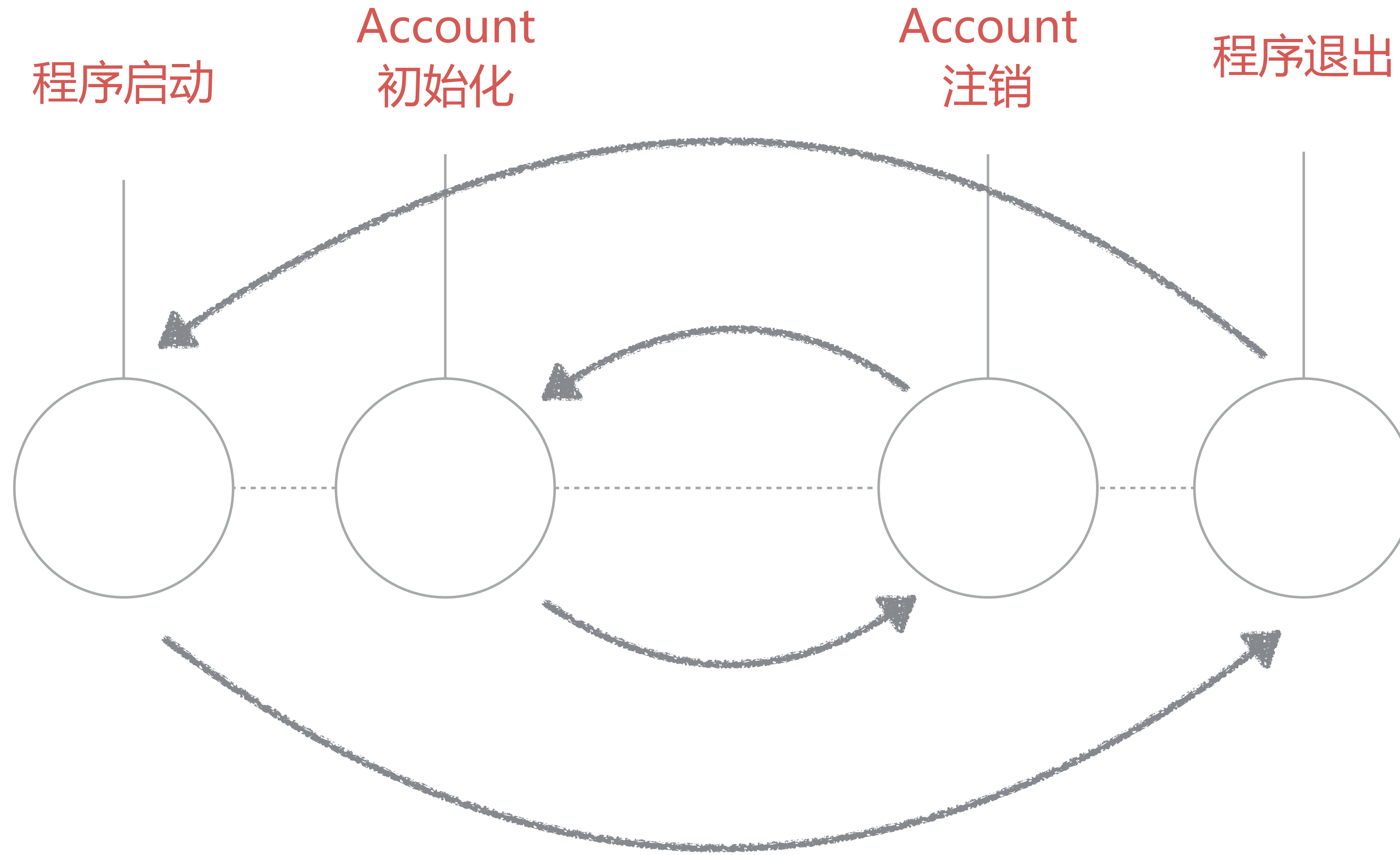
微信kernel



重新设计模块——正确的生命周期设计



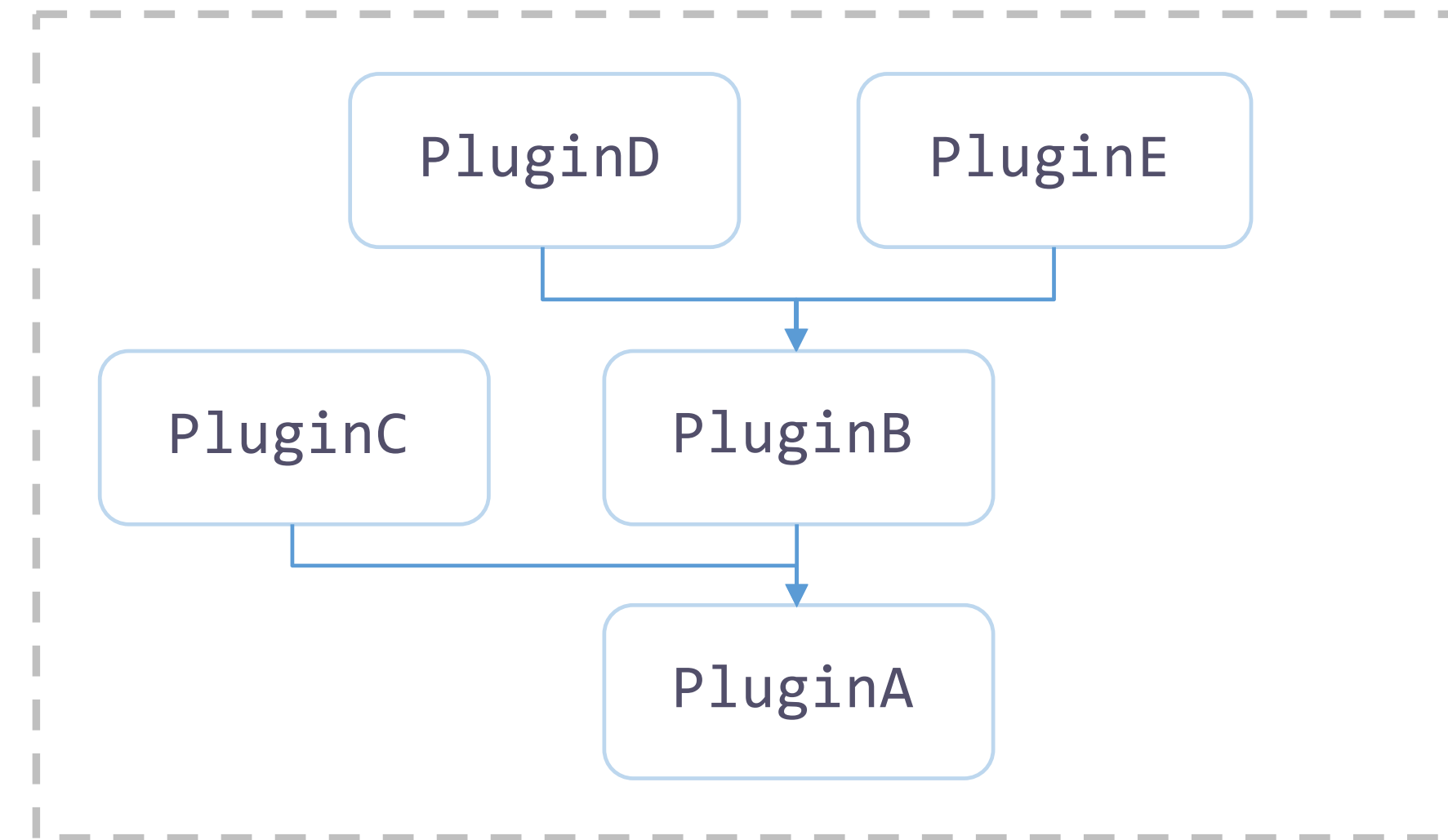
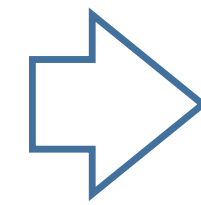
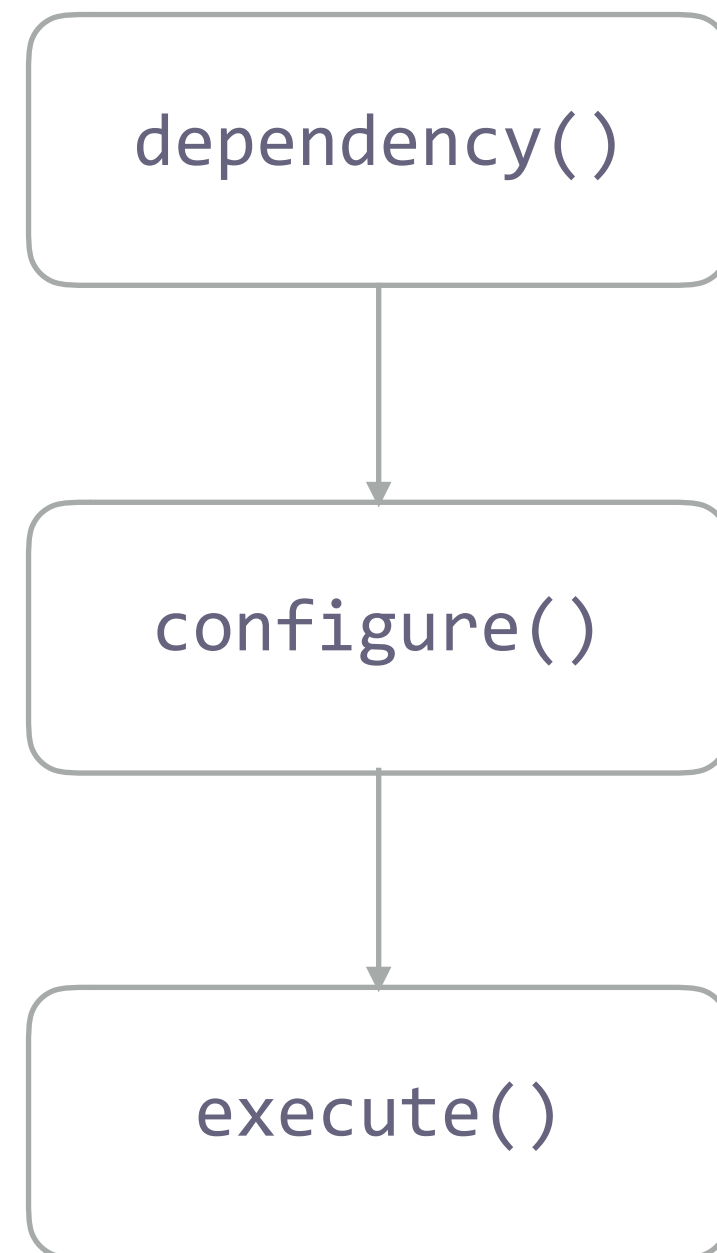
重新设计模块——正确的生命周期设计



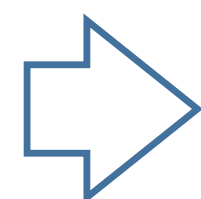
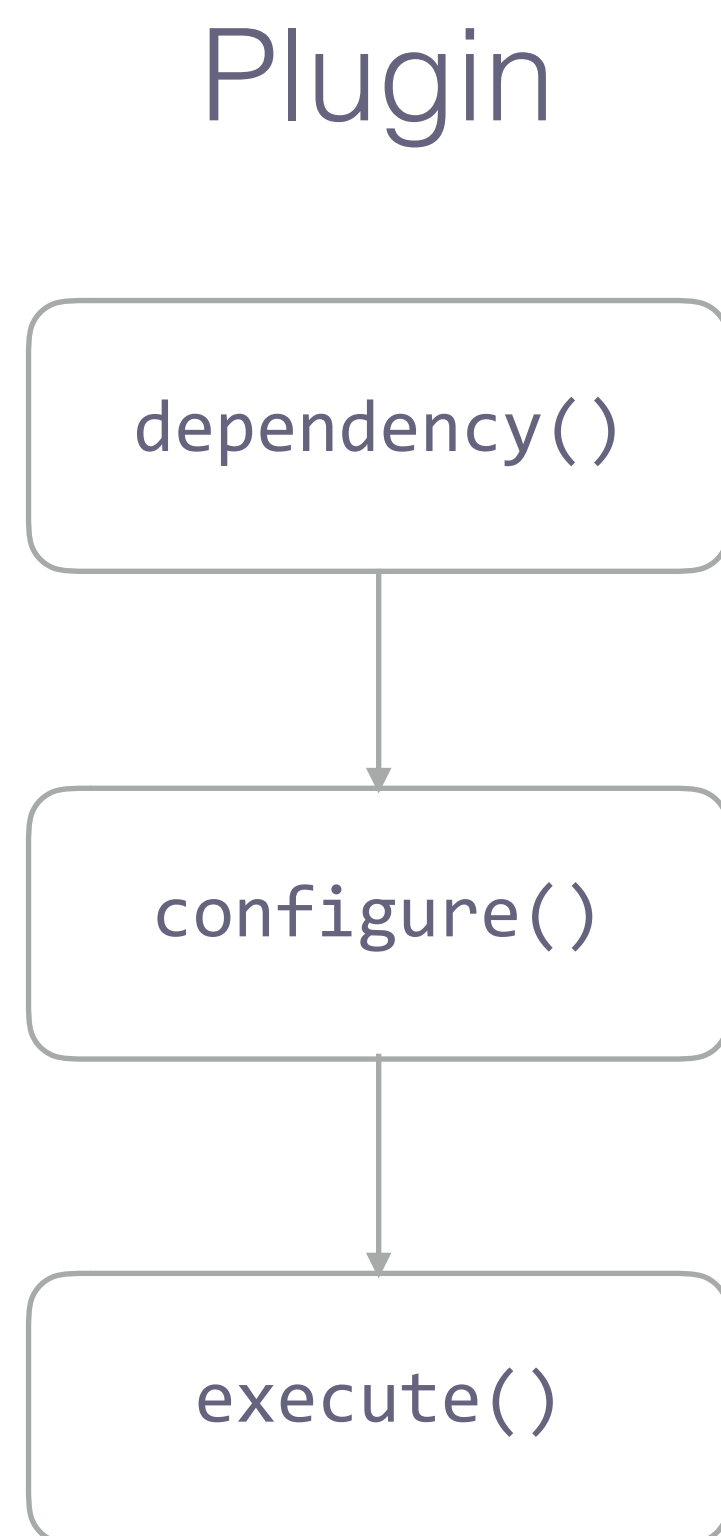
模块生命周期更完整

重新设计模块——加载阶段的变化

Plugin



重新设计模块——加载阶段的变化

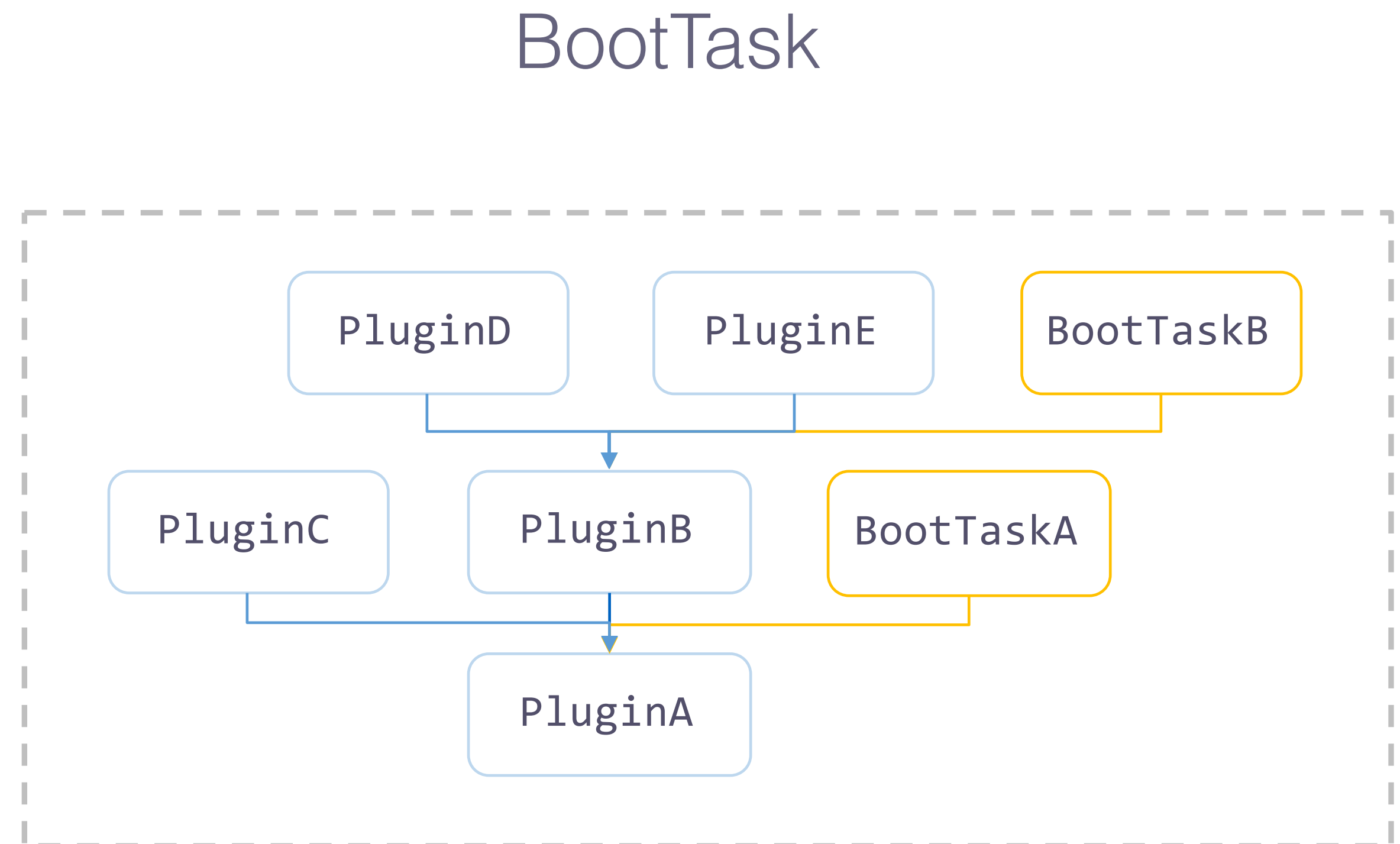
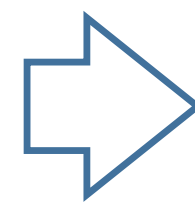
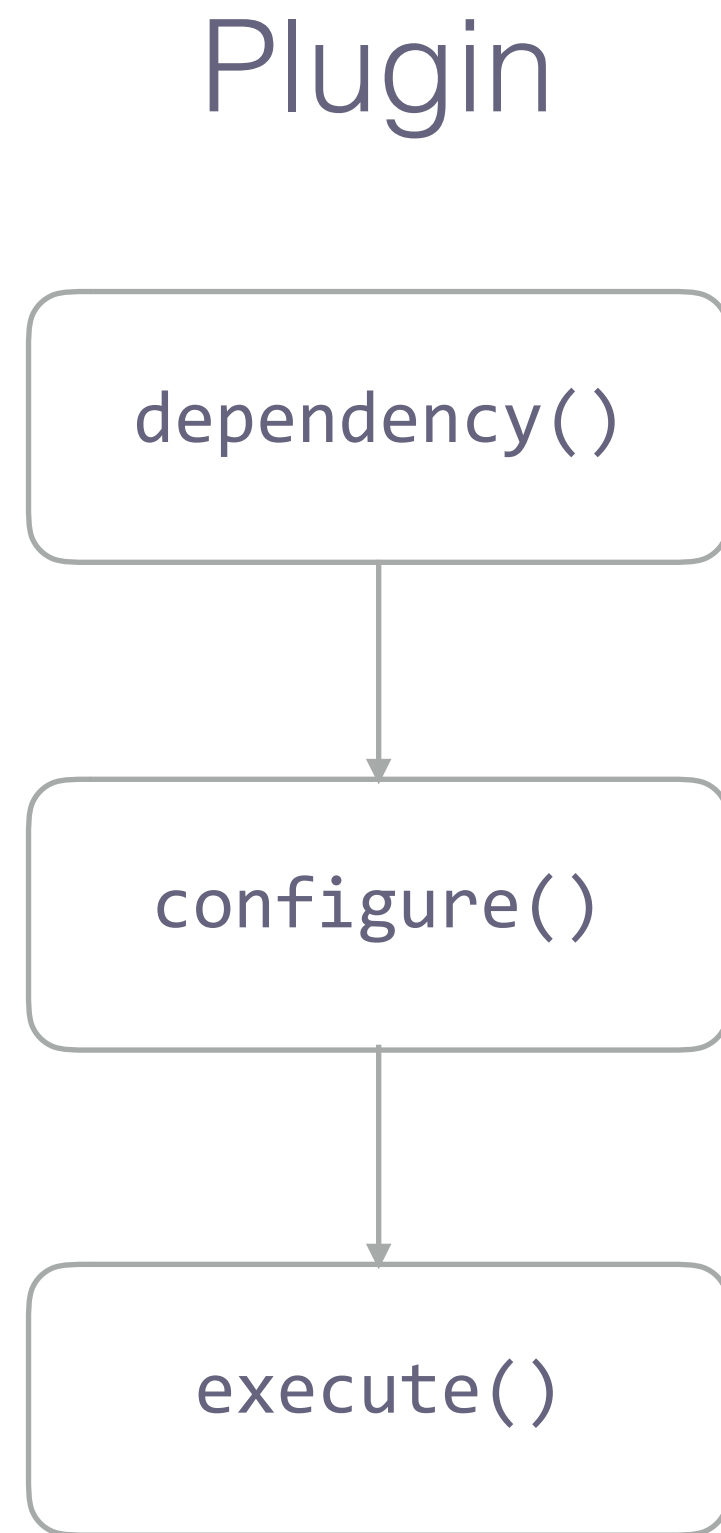


- 配置一些数据
- 注册IService
- 插入BootTask

```
regService(IMyService.class, myService);
```

```
new SFSTask()  
    .after(plugin(IPluginZero.class))  
    .before(this);
```


重新设计模块——加载阶段的变化



设计目标

改变通信方式

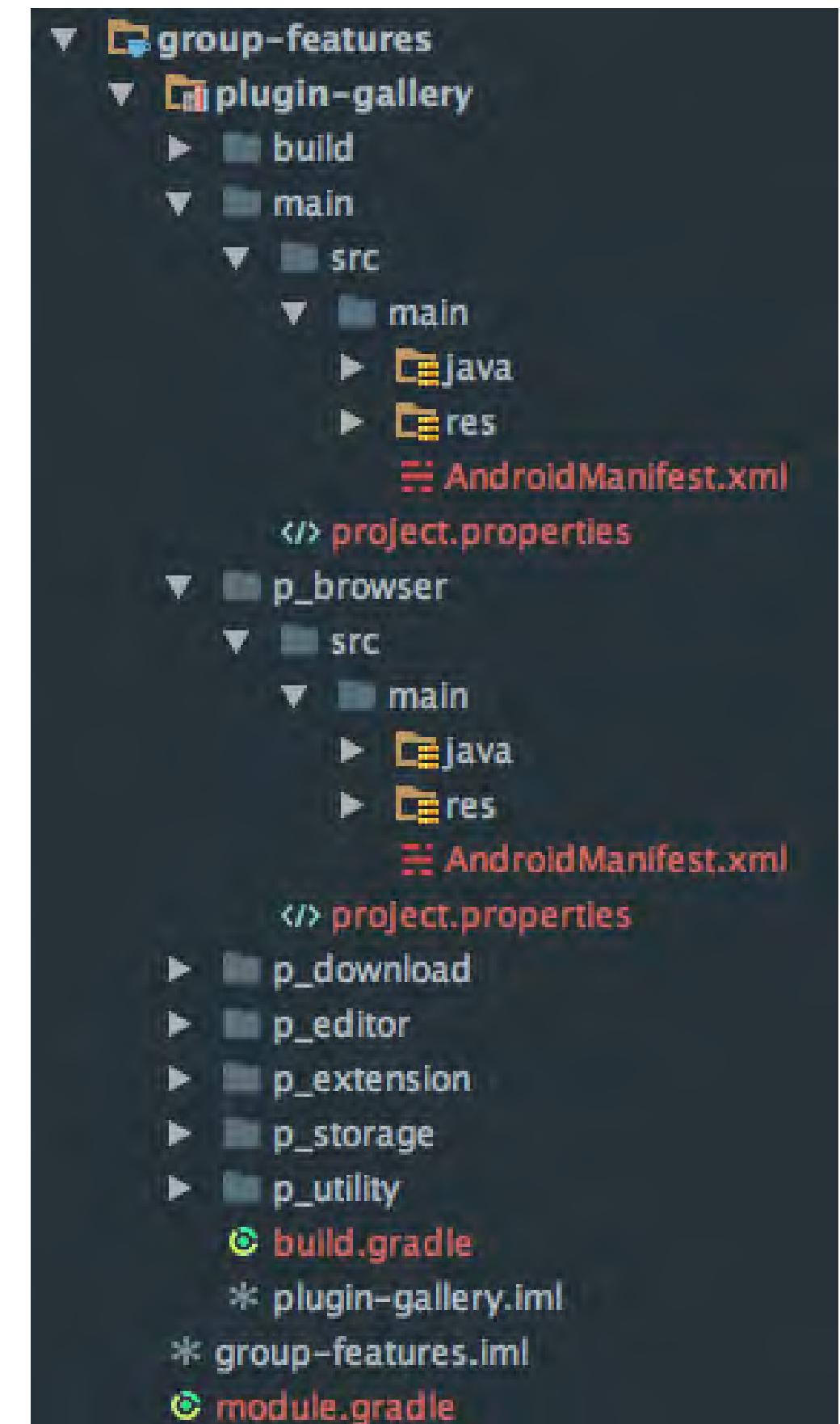
重新设计模块

约束代码

约束代码——pins工程结构

- pins工程结构
 - 约束代码粒度和小代码边界的利器
 - 避免的超量module的创建，更轻量
- code-check

```
apply plugin: 'com.tencent.mm.toolkit.codecheck'
```



设计目标

改变通信方式

重新设计模块

约束代码边界

看看重构效果

- 具体能做到的
 - 满足了硬件同学们提出的需求，存在的问题都得以解决
 - 一行代码的增量编译耗时可以减少60%
 - 架构设计不好量化效果，不过我们还是尝试做一个demo来说明

```
5:05
Welcome to WeChat nano.

Your need login first.

Please enter your account:
grtest2
Please enter your password:
*****

Login...
Login Success!
Welcome back 测grtest2eidii.
Initializing ... 100%
Initialize done.

Click what you wanna do:
All My Friends Add Friend Logout

Select your friends to chat :)

Total 2123 friend(s), page (1/213):
ashu 伯伯鸟 Carl 大怪兽1号 foreverzeus
grmobile7sbc grmobile8 旱圣的大号q 零落~
龙小康 >>>
```

```
5:06
Your need login first.

Please enter your account:
grtest2
Please enter your password:
*****

Login...
Login Success!
Welcome back 测grtest2eidii.
Initializing ... 100%
Initialize done.

Click what you wanna do:
All My Friends Add Friend Logout

Select your friends to chat :)

Total 2123 friend(s), page (1/213):
ashu 伯伯鸟 Carl 大怪兽1号 foreverzeus
grmobile7sbc grmobile8 旱圣的大号q 零落~
龙小康 >>>
```

```
5:07
Login...
Login Success!
Welcome back 测grtest2eidii.
Initializing ... 100%
Initialize done.

Click what you wanna do:
All My Friends Add Friend Logout

Select your friends to chat :)

Total 2123 friend(s), page (1/213):
ashu 伯伯鸟 Carl 大怪兽1号 foreverzeus
grmobile7sbc grmobile8 旱圣的大号q 零落~
龙小康 >>>

You are chatting with [Carl] now :
测grtest2eidii: hi
Carl: [疑问]啥
测grtest2eidii: 没事
```

WeChat Nano

基于重塑后的模块化，一天之内可开发出的极精简版本微信——WeChat Nano



安装包大小约 3.5 MB，完整版本的10%



内存占用是完整版本的25%*

注：只计算不同部分的PSS，Dalvik PSS + Native PSS + .dex mmap PSS

TABLE OF CONTENTS 大纲

- 回顾，微信Android架构历史
- 问题又来，模块化之后的“坑”
- 解决问题，重塑模块化
- 思考，取舍和选择
- 代码之外，架构之内

取舍和选择

- 热门的大前端
 - “远水不解近渴”
- 插件化/沙盒化，模块动态性
 - 模块化与动态性分离
 - 微信过于复杂的数据关系
 - 复杂度和兼容性的代价
- 纯粹的模块化
 - 问题的根本在于更彻底的模块化设计、解耦、限制代码

取舍和选择

- 热门的大前端

“远水不解近渴”

- 插件化/沙盒化，模块动态性

模块化与动态性分离

复杂度和兼容性的代价

微信过于复杂的数据关系

- 纯粹模块化

- 问题的根本在于更彻底的模块化设计、解耦、限制代码

TABLE OF CONTENTS 大纲

- 回顾，微信Android架构历史
- 问题又来，模块化之后的“坑”
- 解决问题，重塑模块化
- 思考，取舍和选择
- 代码之外，架构之内

代码之外，架构之内

- 模块负责人制度
 - 渐进式的代码审查文化，适合没有从一开始执行全员性质Code Review的项目
 - 改变修改和优化代码的动机，创造想法落地的机会
 - “代码归属感”

代码之外，架构之内

- 模块边界划分经验之谈

- 面对划定模块边界的细节问题，“讲一个符合逻辑的故事”是一种简单有效的划分标准
- 模块的一般组织方式，可以将模块分成三个部分：implementation工程、api工程、library工程

THANKS

让创新技术推动社会进步

HELP TO BUILD A BETTER SOCIETY WITH
INNOVATIVE TECHNOLOGIES

Geekbang >

极客邦科技

InfoQ ueue

专注中高端技术人员的技术媒体



EGO EXTRA GEEKS' ORGANIZATION
NETWORKS

高端技术人员学习型社交平台



StuQ ueue
斯达克学院

实践驱动的 IT 教育平台

