

Tangram

@伯灵 / 天猫

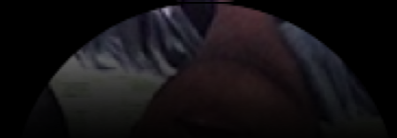
我们花了5年时间跟平台打游击
忽略了前端技术本身的进化

JS/HTML转Native技术
天生有跨平台和快速发布的属性
性能和稳定性却仅仅是接近原生Native

我想，5年前如果我们专注：
WebKit的性能优化
WebView的能力扩展
Mobile Web开发技术进化

那么，今天的局面应该会完全不一样





Tangram.pingguohe.net Team



Tangram的误会

- Tangram有开发模式，但不是**开发框架**
- Tangram具有动态性，但不是**动态化方案**
- Tangram被开发者使用，但用户不是**开发者**

“面向业务的界面解决方案”



面向业务

- 不求完整，只求够用
- 追求灵活性有节制，性能和效率可以放肆



解决方案

- 开发者是我们的合作伙伴
- 全链路建设，不留短板



iOS或Android的实现细节 如何高效接入Tangram

以上内容你都可以在 Tangram.pingguohe.net 找到

以上内容今天 **不** 讲



体系 & 生态 (30%)

Tangram的全链路思考

Tangram SDK (20%)

开源了，无论如何说两句

TAC (40%)

我们如何让跨栈成为日常开发常态

Tangram 2.0 (10%)

未来3-5个月Tangram会变成什么样

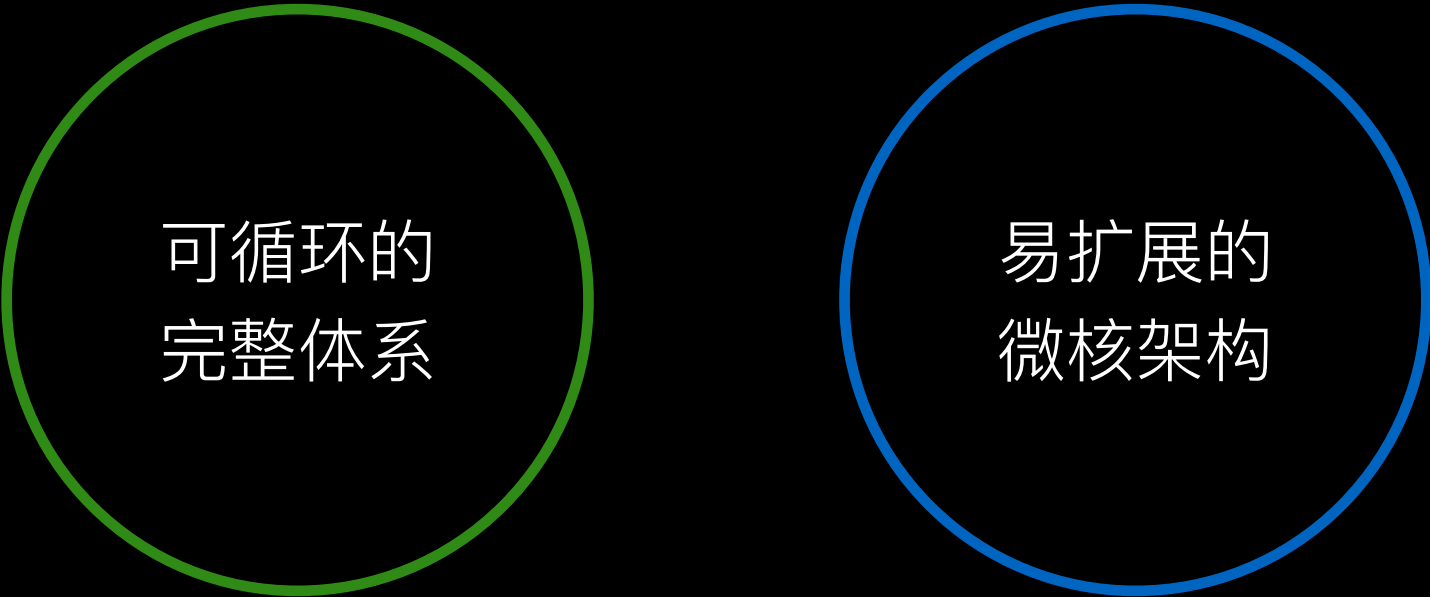


体系 & 生态



我们对生态的理解

- 所有角色是合作关系，互相促进
- 所有人都贡献，没有人单纯的消费
- 没有管理员，体系和规则成就生态



可循环的
完整体系

易扩展的
微核架构



围绕角色构建的体系



框架开发

对框架的可用性、稳定性负责。是整个产品的技术基石，决定了产品质量的底线高度

用户

最终产品的使用者。是整个产品的目标，整个产品都围绕创造用户价值而设计

业务开发

产品的直接实现。把产品设计实现的一线开发，需要快速奔跑，决定了产品的上限高度

业务产品

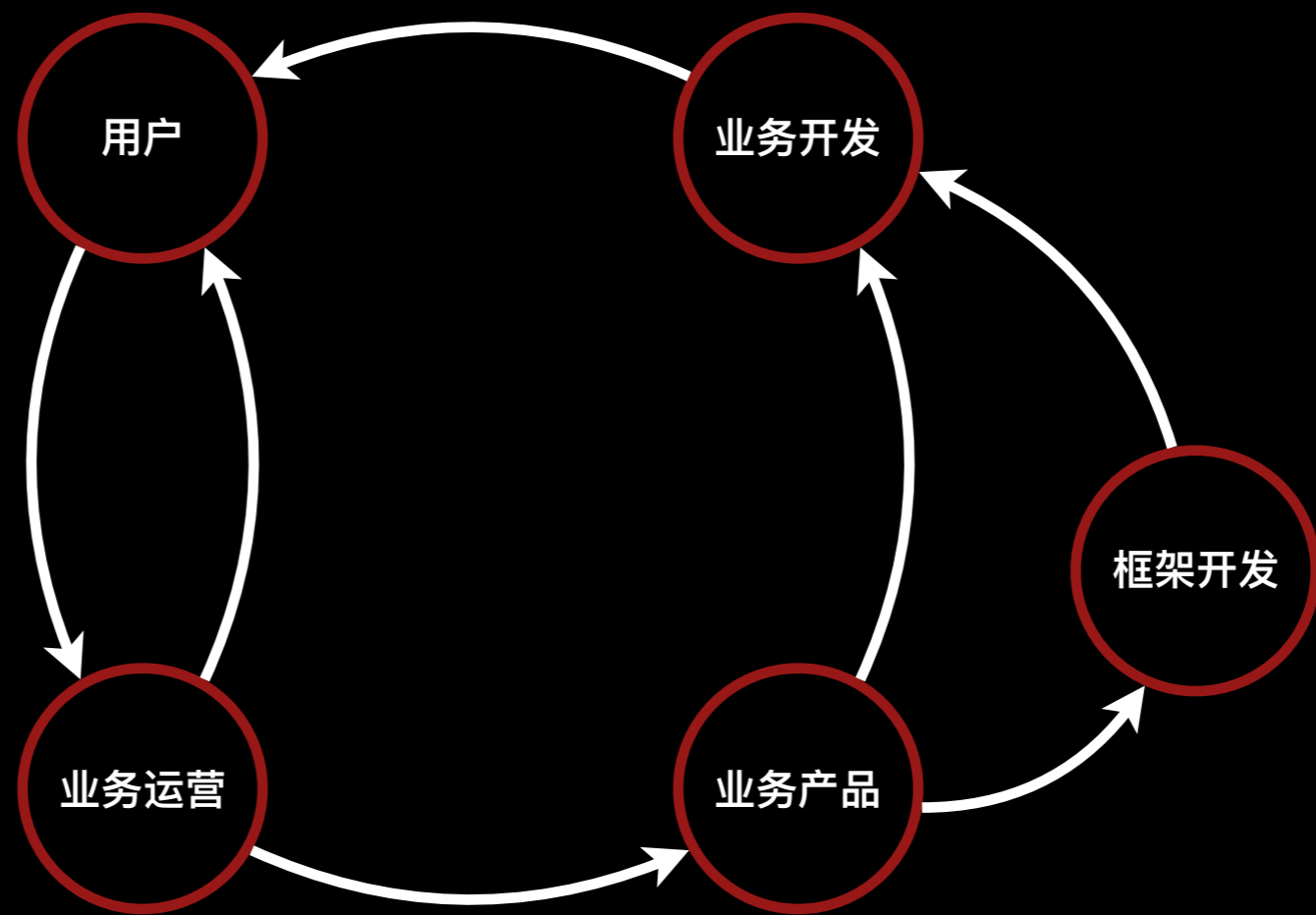
产品设计者。决定了整个产品以何种面貌示人，某种程度上是制定方向的人

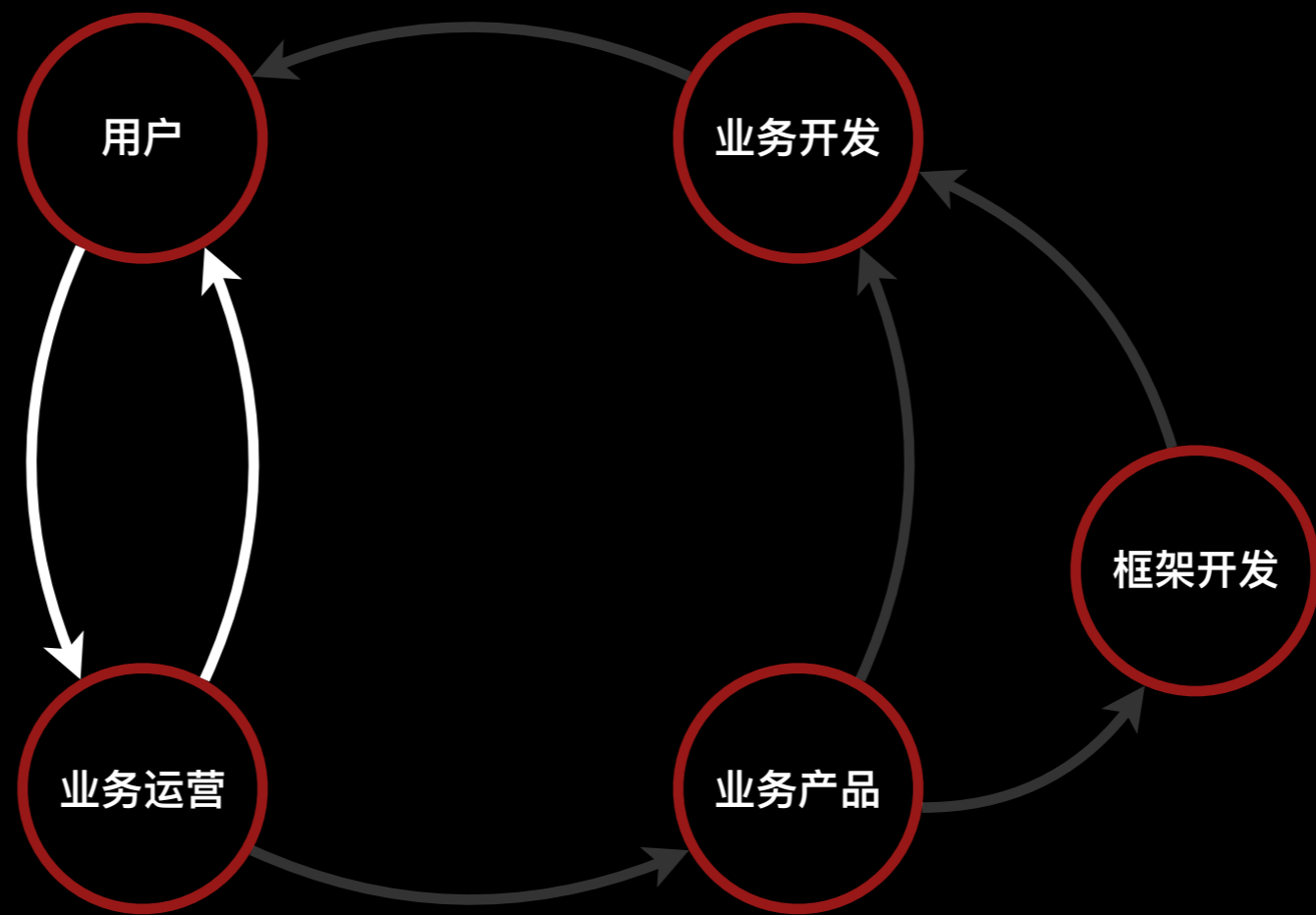
业务运营

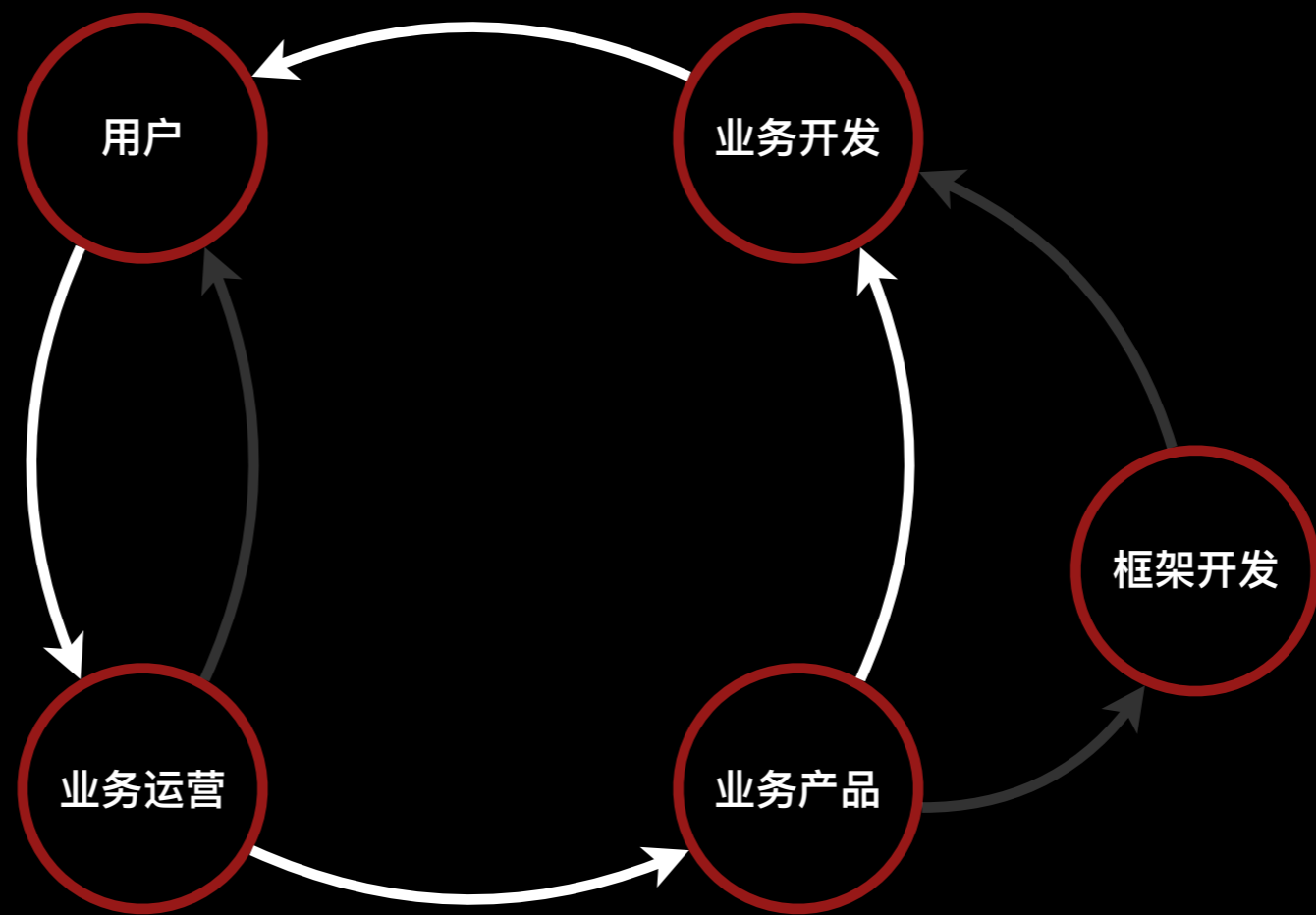
业务掌舵者。站在一线把控产品方向和管理用户的人。

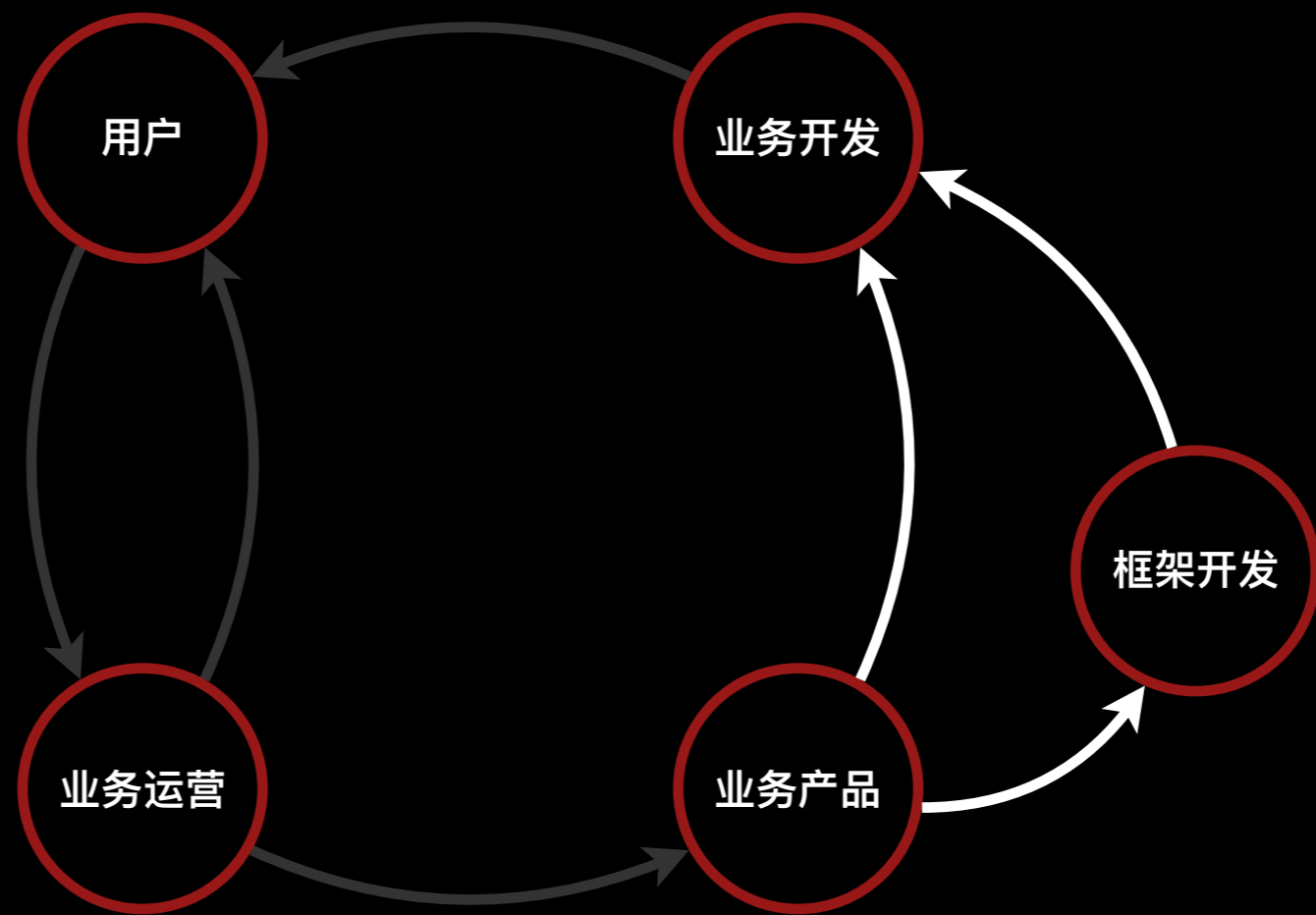
角色

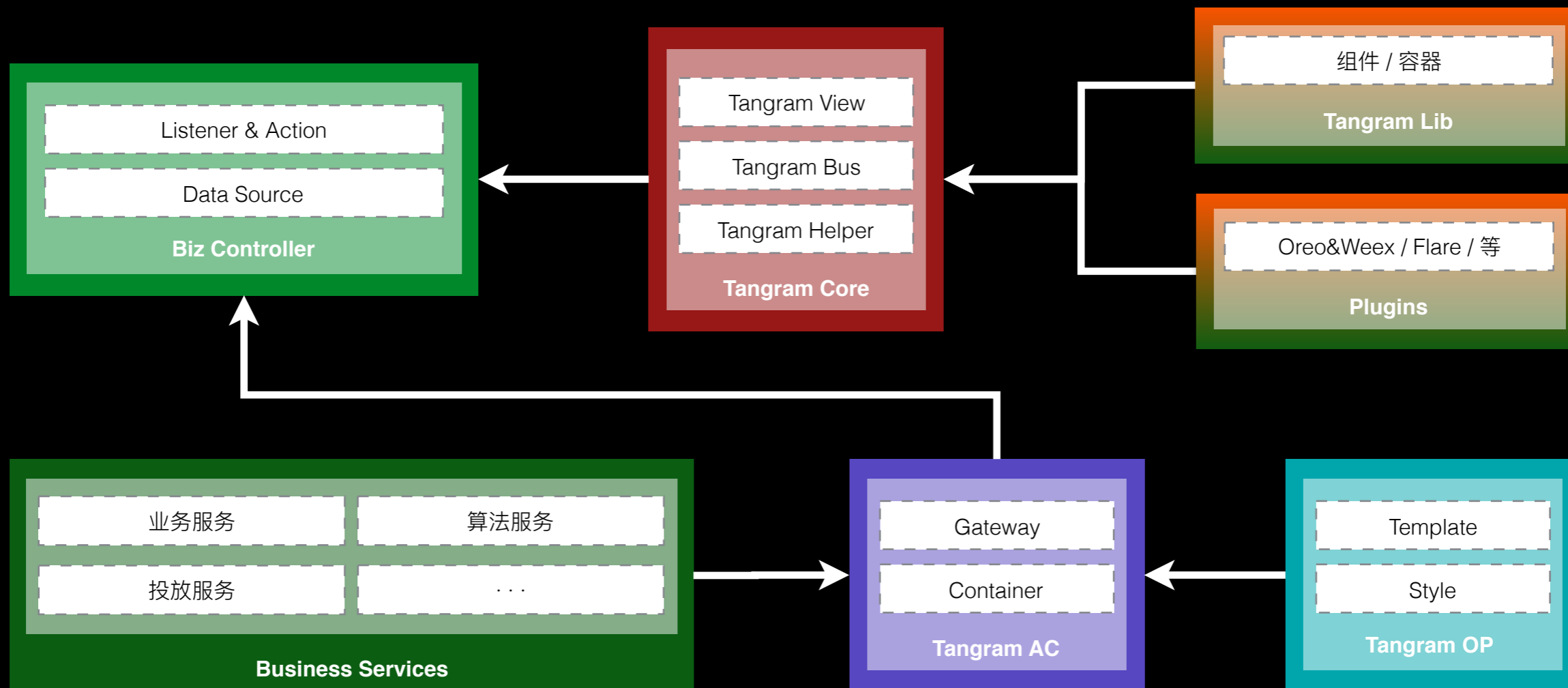






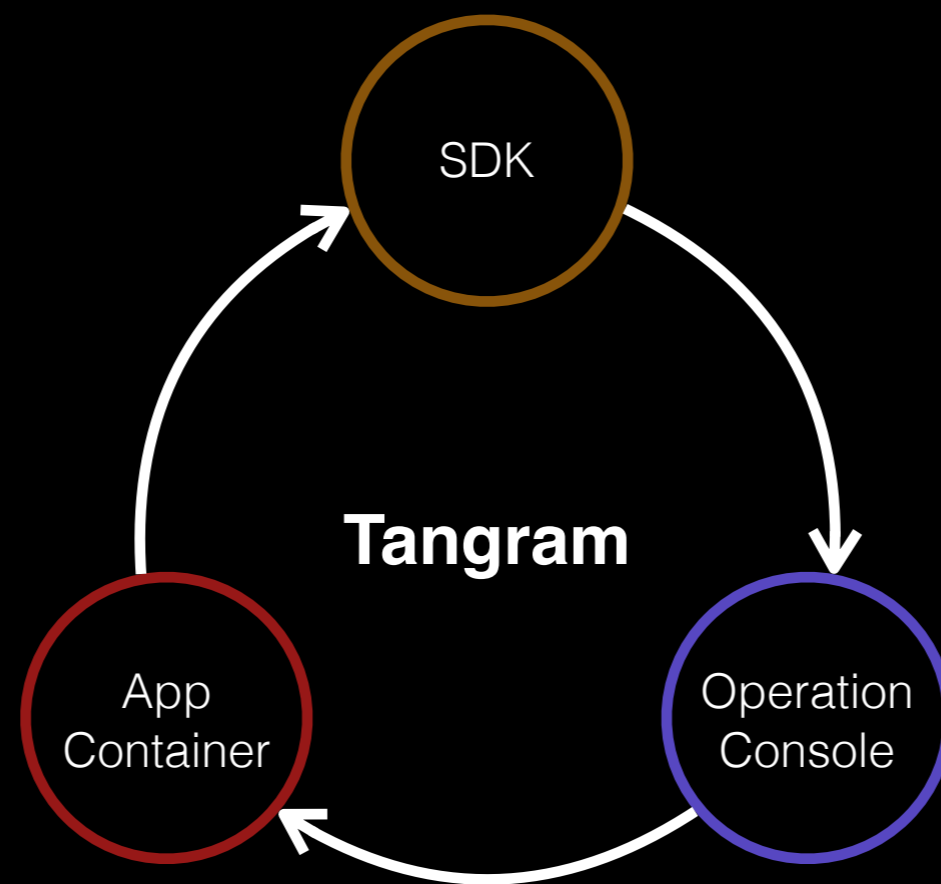






前端 + 后端 + 开发 + 运维





Tangram体系



SDK

解决渲染问题，关注性能，效果，动态性，灵活性，稳定性

App
Container

后端逻辑容器，降低后端逻辑开发成本，支持高效，稳定的业务处理

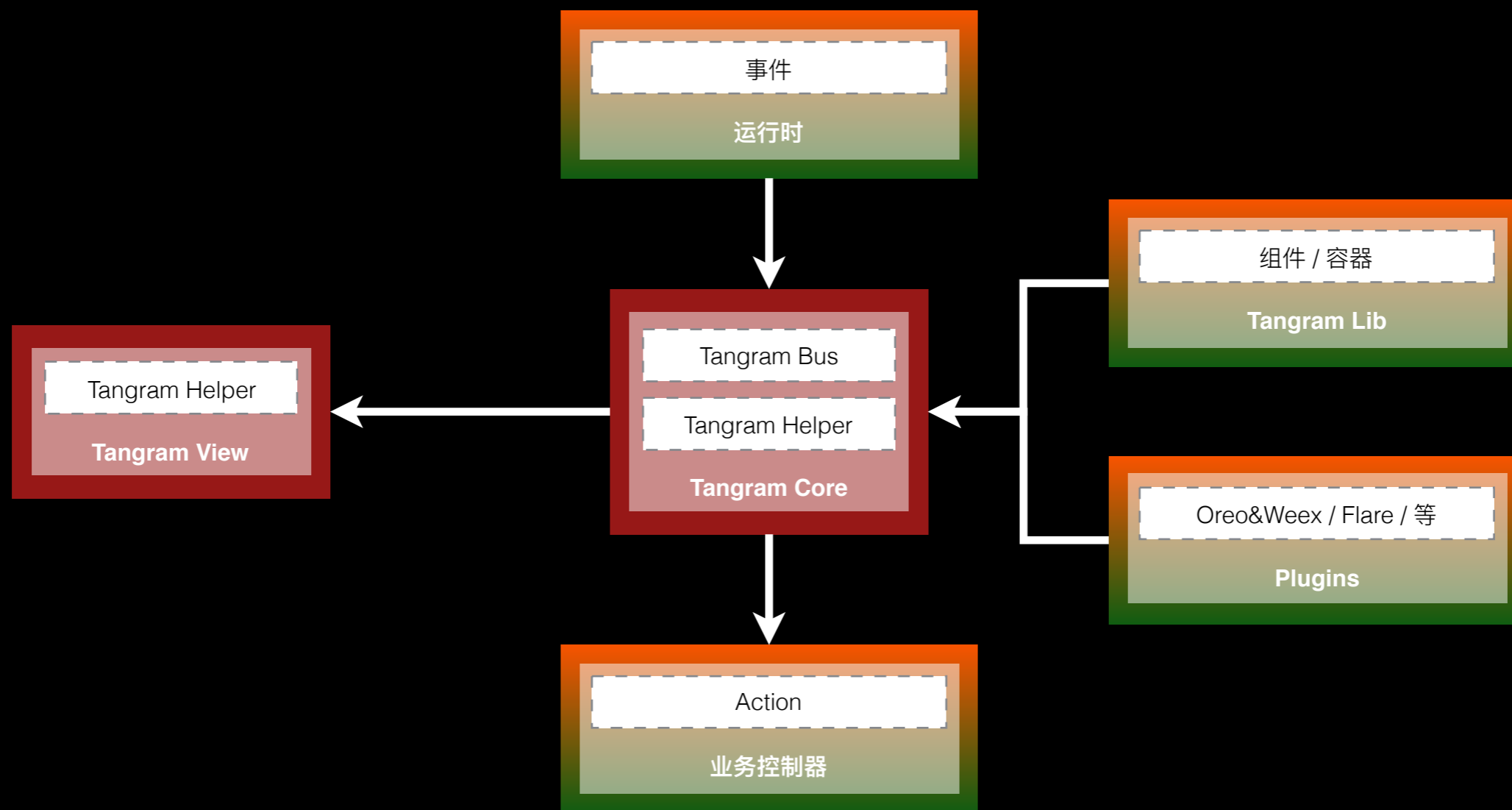
Operation
Console

界面控制台，通过工具的方式，在日常运营中释放开发资源

Tangram体系

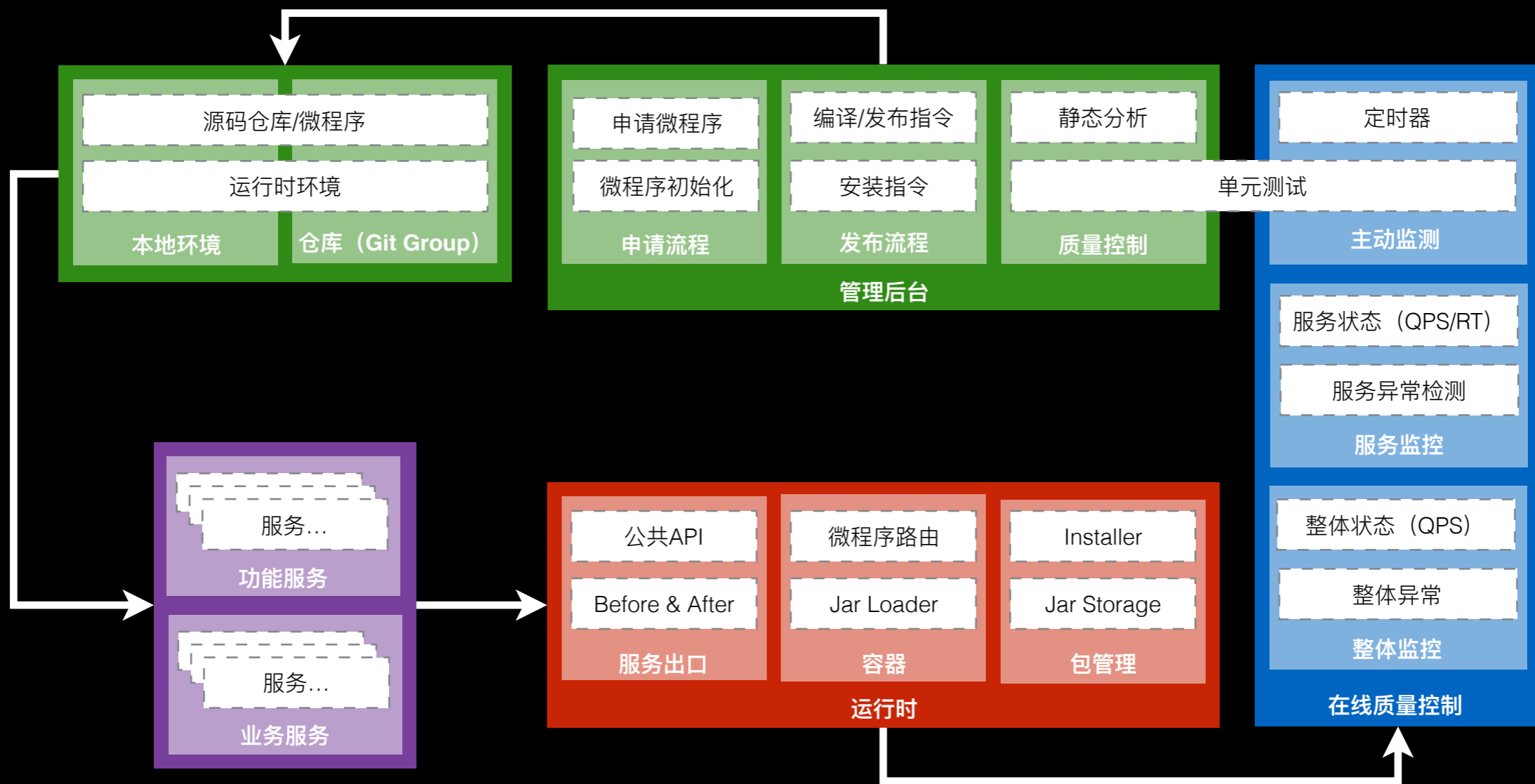
绝对公平的微核架构



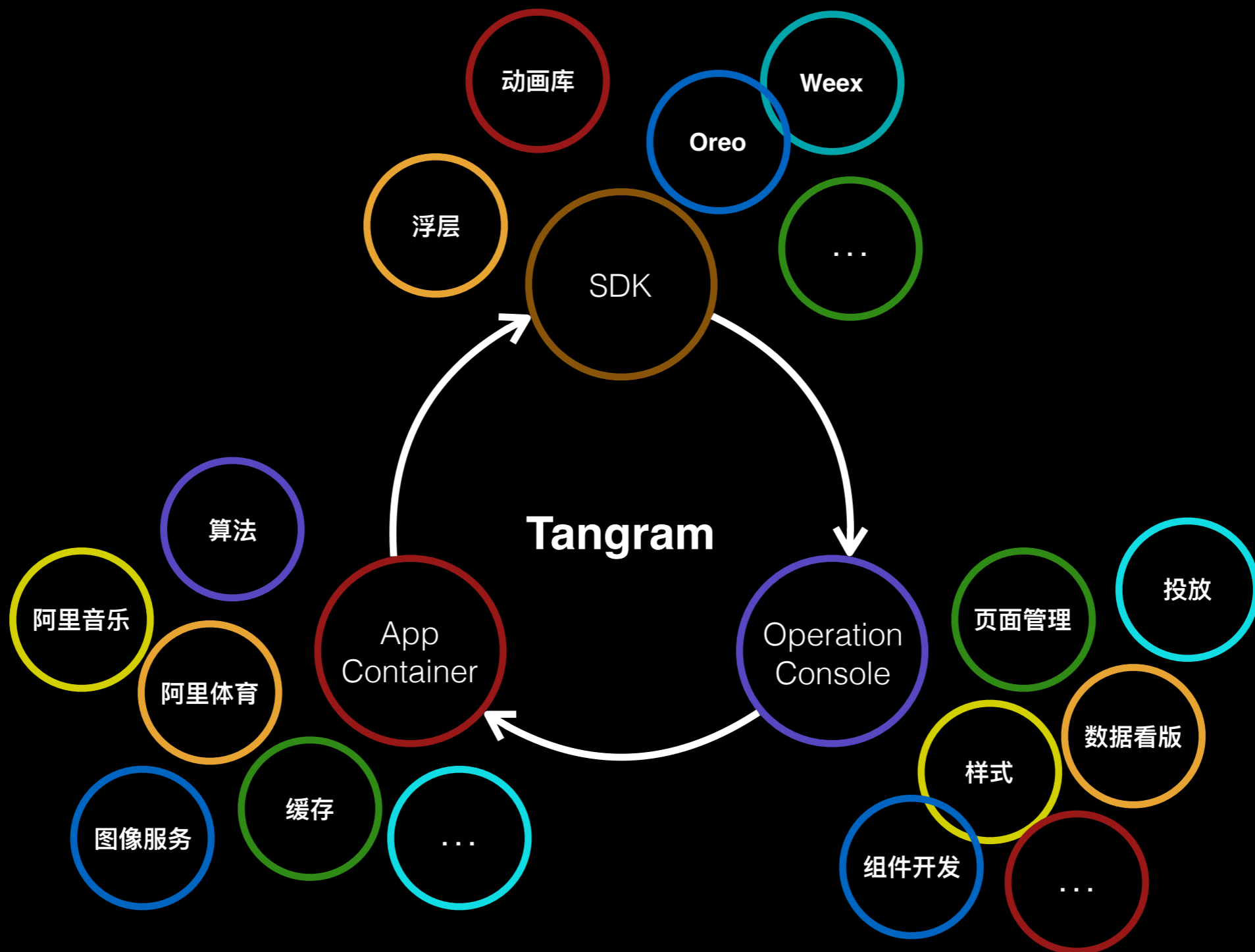


SDK





TAC



Tangram SDK





天猫APP首页

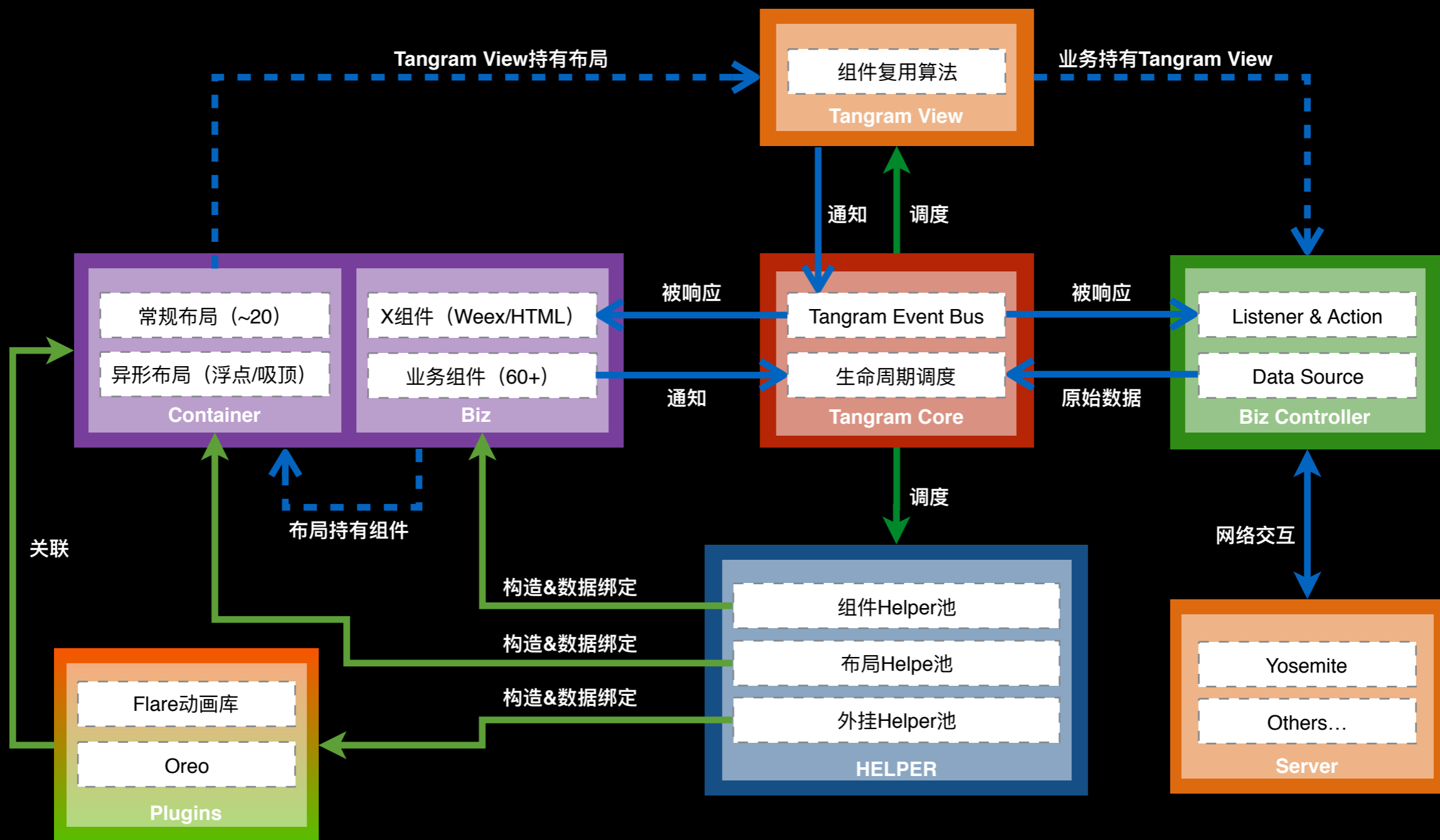


天猫热点 千万别为这四部坏片花一分钱
刷了这么多年牙！现在才发现全错了！



组件 + 模板





SDK架构



模板动态化 & 组件动态化



性能保障



能力	Tangram	UICollectionView
高效回收	√	√
跨层复用	√	×
以视图id为索引的复用逻辑	√	×
灵活选择复用逻辑	√	×

自定义回收/复用机制

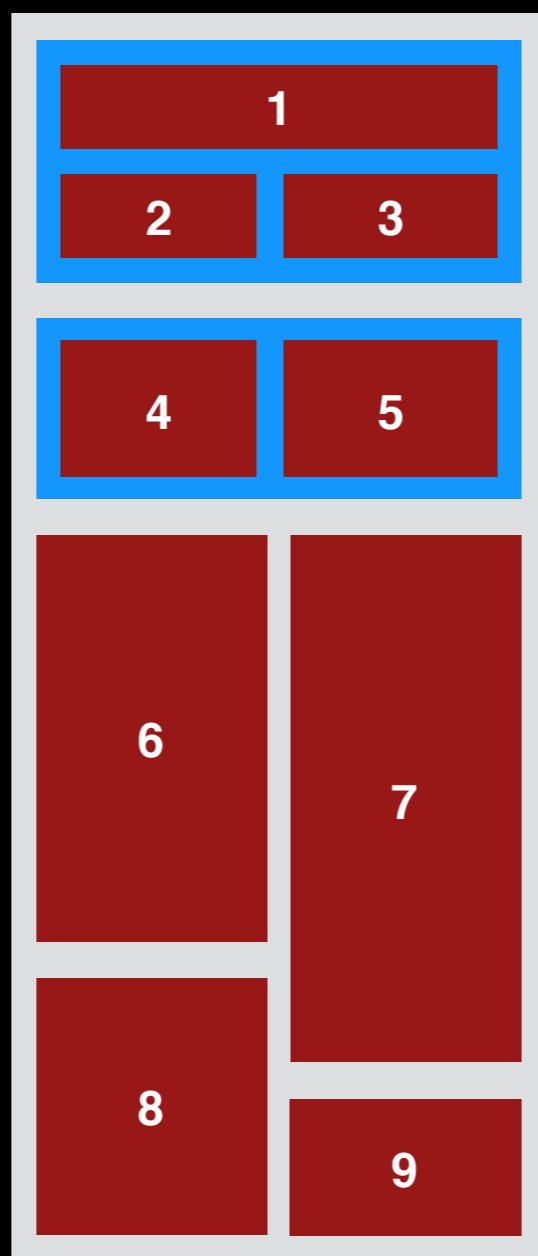


复用类型	复用策略
同类型复用	以组件类型作为复用id 同种类型组件做复用
指定复用id	同种类型组件内再细分复用池 增强复用时视图性能
不复用	创建之后，仅做一次赋值 来回滑动不变化，适应高性能需要

多级复用逻辑

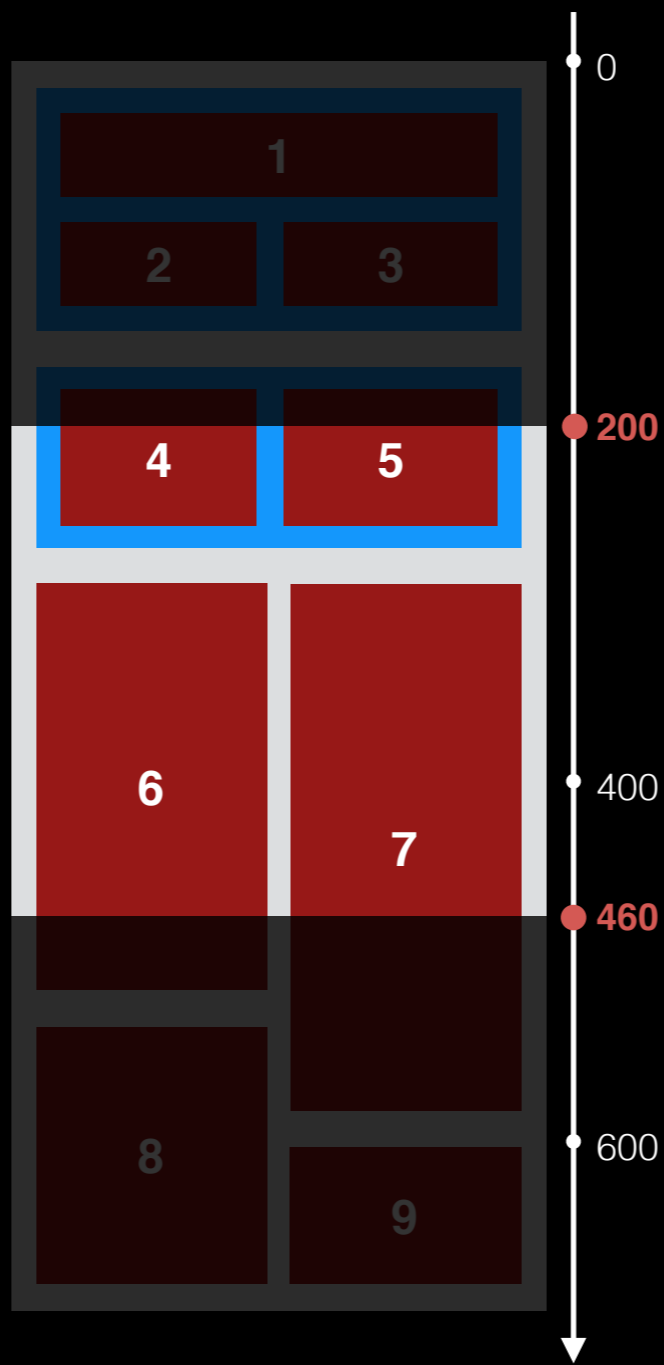
回收/复用算法





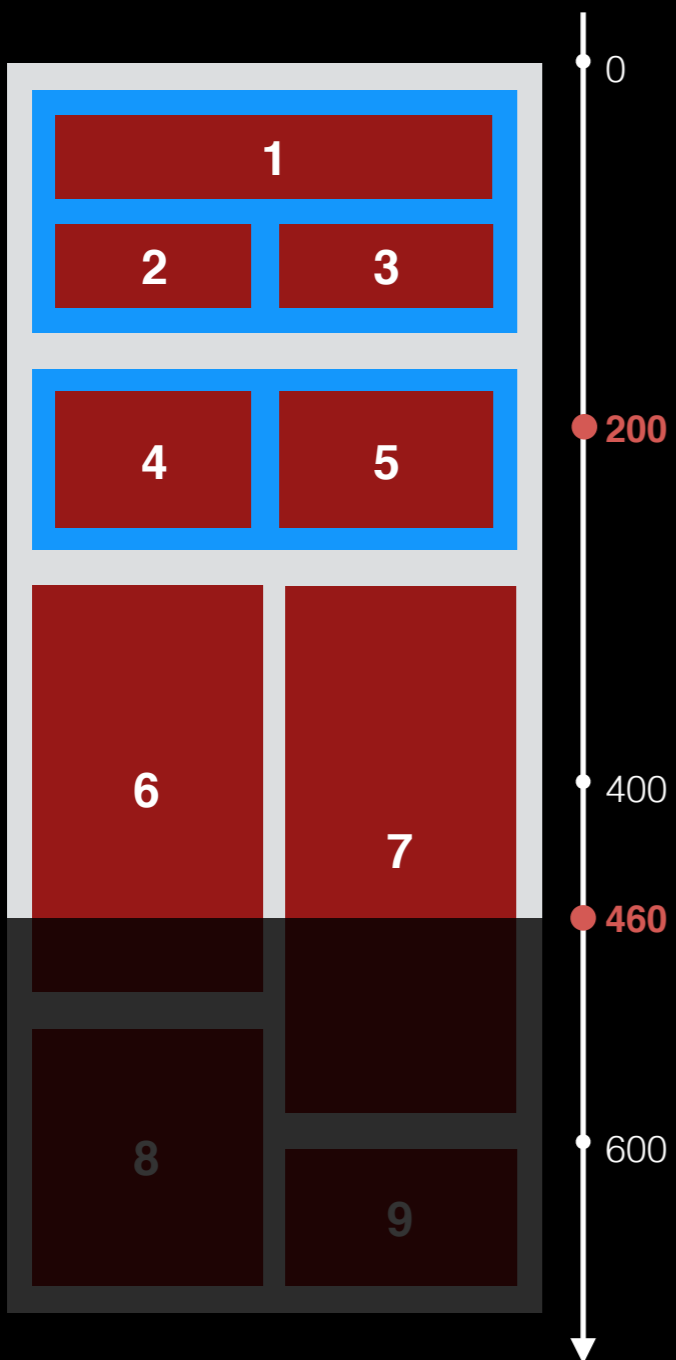
整体页面





可视区域组件发现





视图顶部坐标在
可视区域底部以上

⇒ [1, 2, 3, 4, 5, 6, 7]

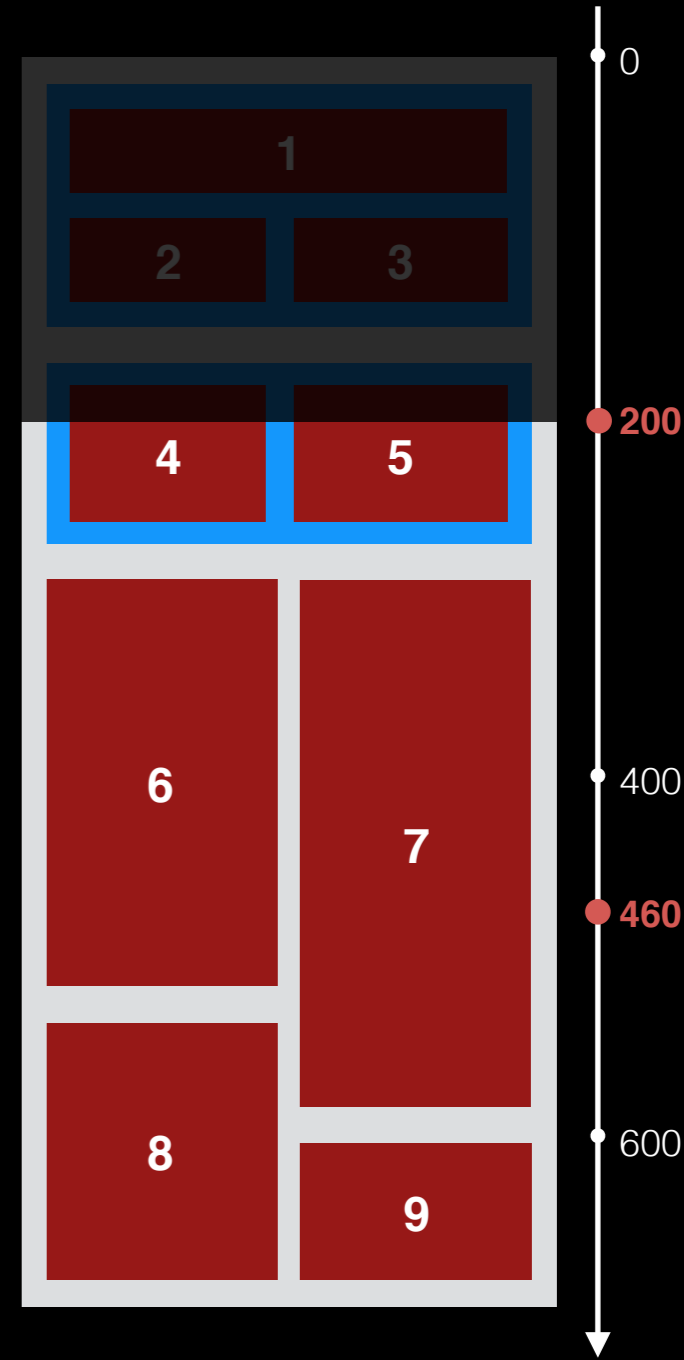
×

[4, 5, 6, 7, 8, 9]

⇐

∥

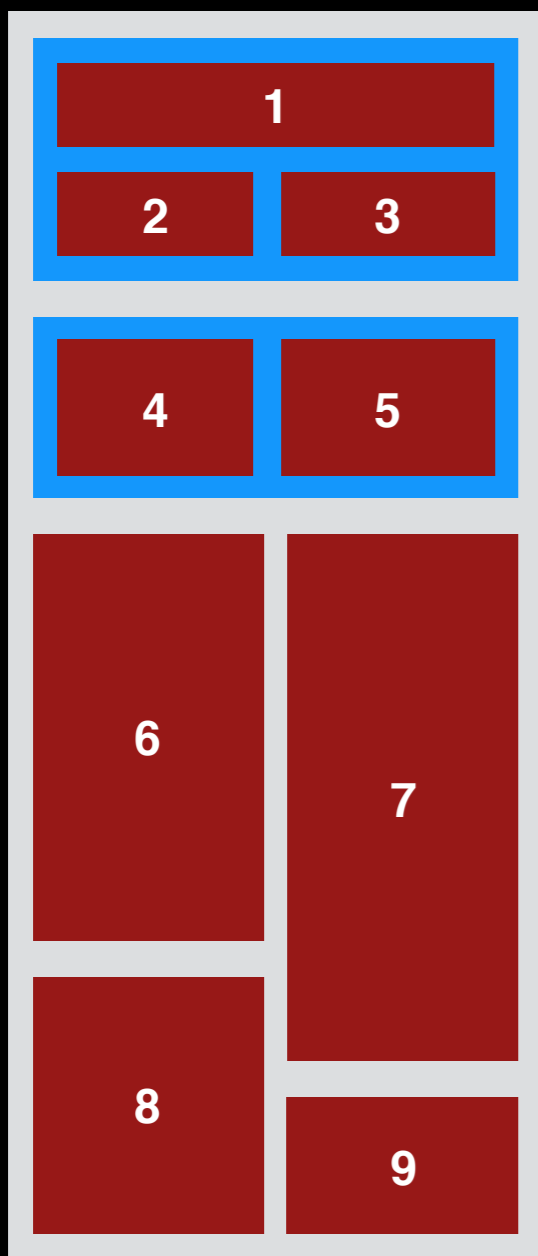
[4, 5, 6, 7]



视图底部坐标在
可视区域顶部以下

找到两个集合求交集





➡
创建索引

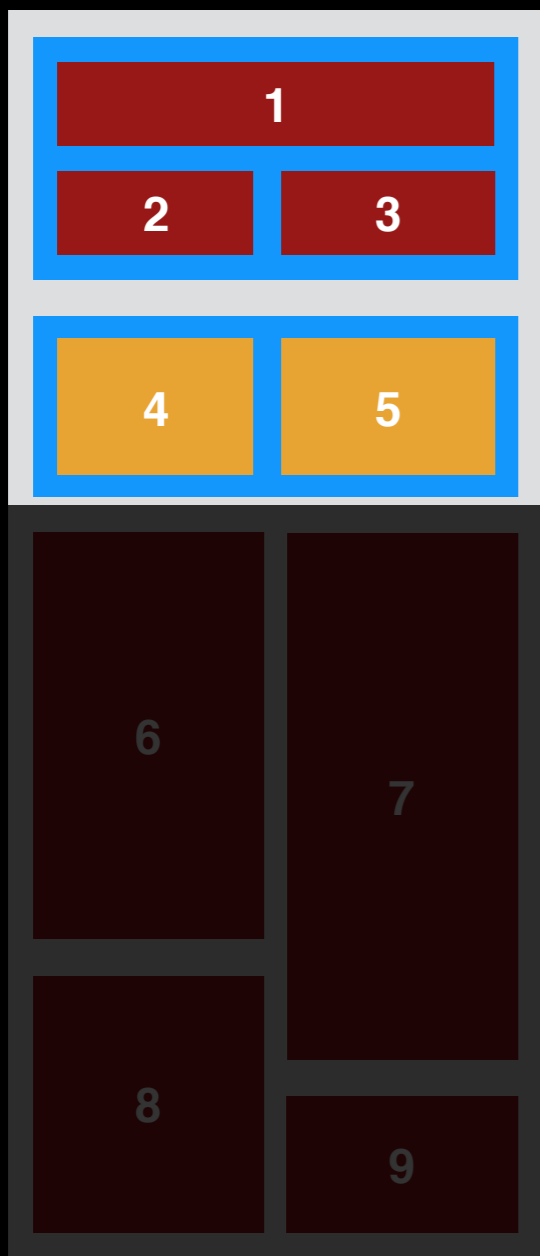
Top	Bottom
1	9
2	8
3	7
4	6
5	5
6	4
7	3
8	2
9	1

➡
两次二分查找

[1, 2, 3, 4, 5, 6, 7]
x
[4, 5, 6, 7, 8, 9]
||
[4, 5, 6, 7]

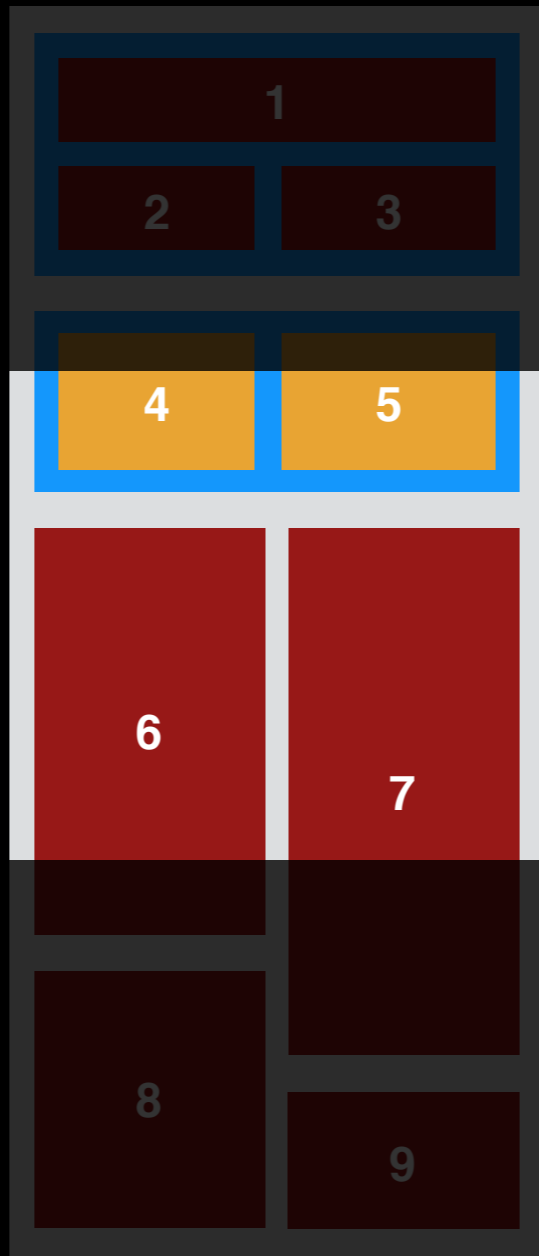
双索引模型





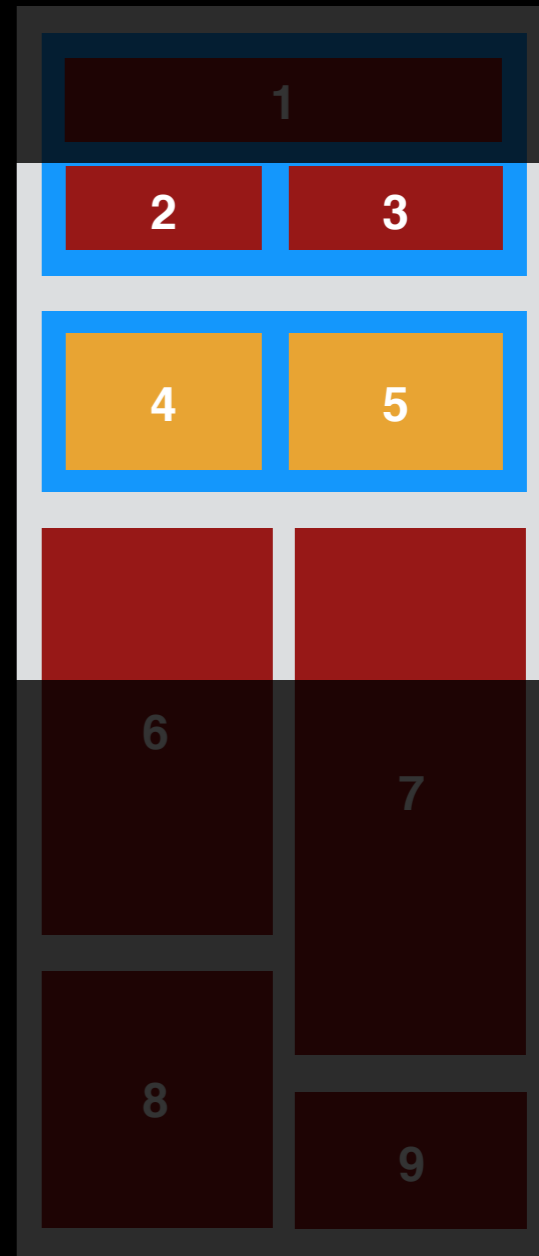
首屏

1 2 3 和 4 5是不同类型



回收

1, 2, 3被标记为可复用



复用

优先选择2, 3复用渲染

原始组件优先复用原则



TAC模式下的跨栈开发



跨 栈



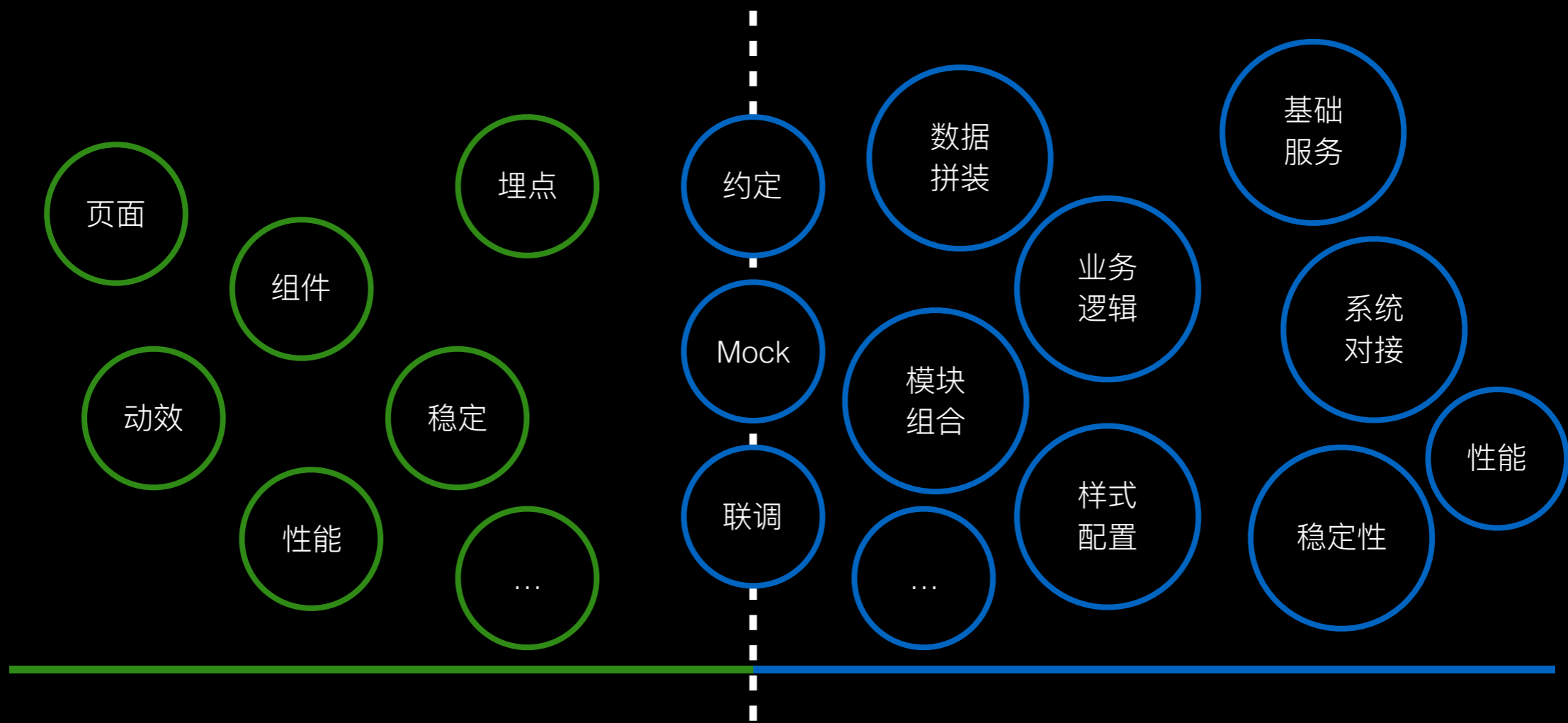
概念

- 不是 —— 一个人有承担多个角色的能力
- 而是 —— 一个人以多个角色完成工作的模式



优势

- 开发效率 —— 一个人承担多种角色，把沟通和交流成本降到最低，提升效率
- 产品效果 —— 在整个开发过程中，体现的是一个人的意志，把这个人技术的理解贯彻的更加彻底



传统模式

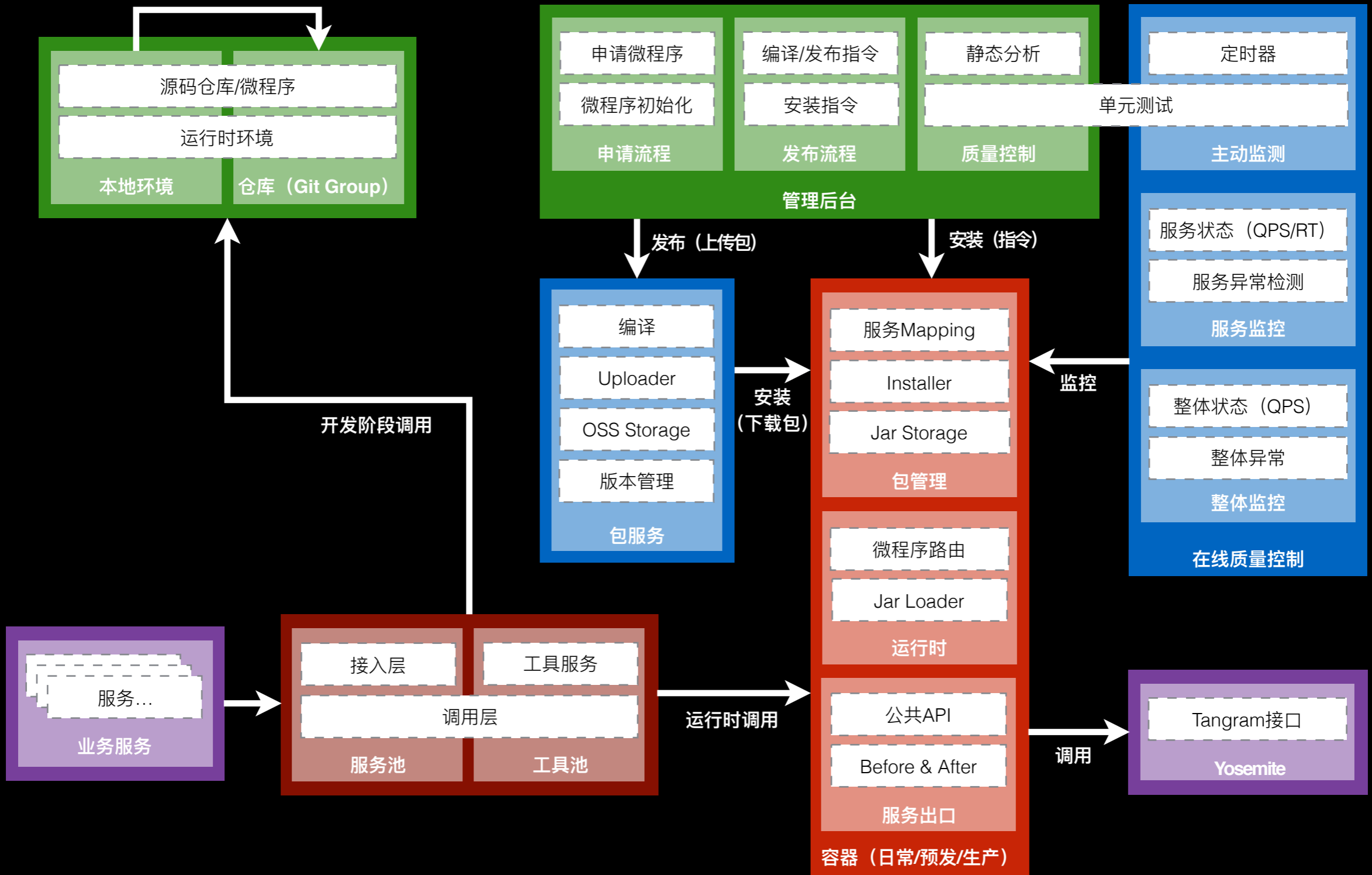




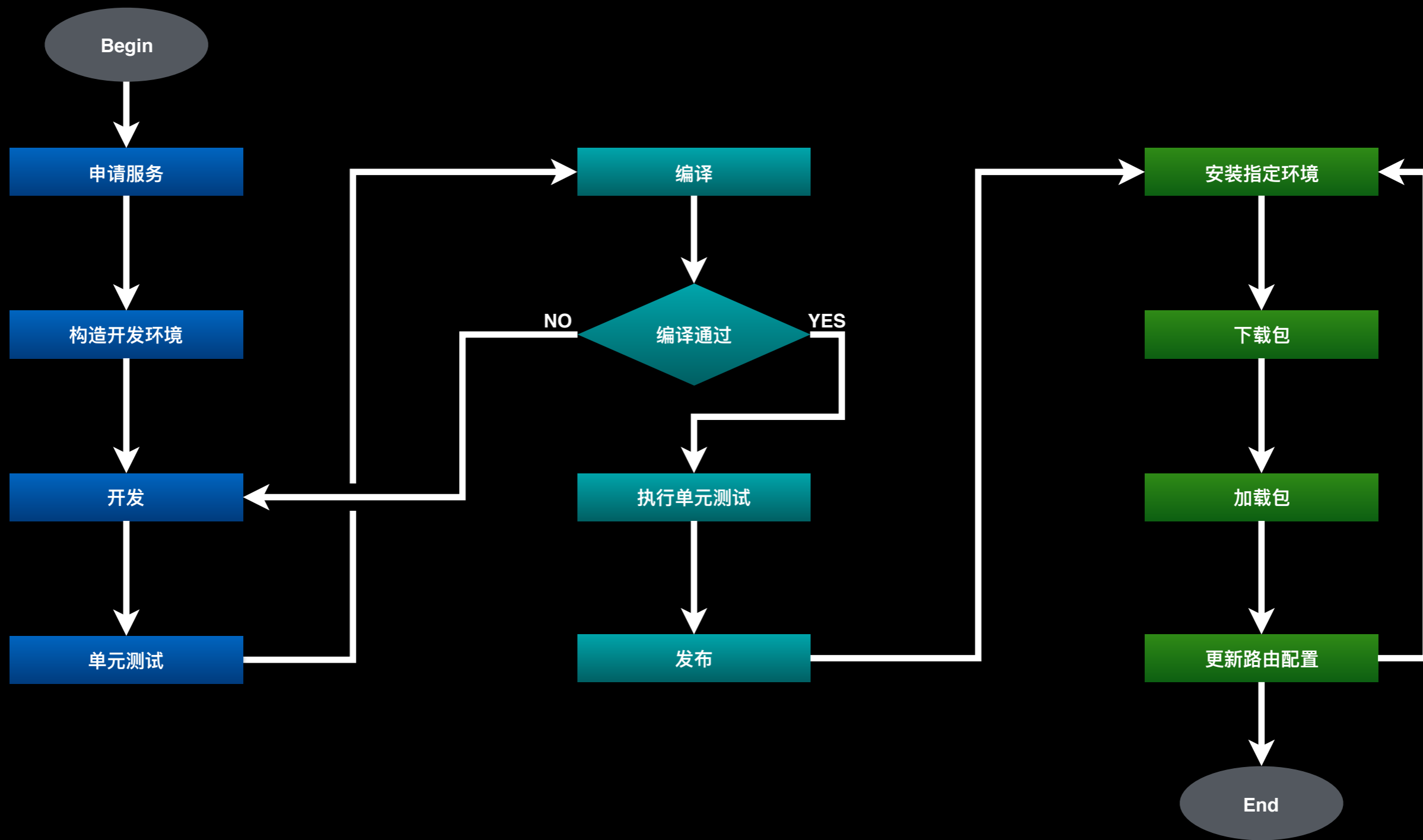
TAC模式

打破壁垒

- 定义低成本开发与发布流程，打破后端开发的**流程壁垒**
- 低成本搭建与维护开发环境，打破后端开发的**环境壁垒**
- 建设高稳定性保障的容器，绕开后端开发的**经验壁垒**



Tangram App Container



TAC流程



	前端开发	后端开发
 业务逻辑	√	√
 系统稳定性		√
 服务对接		√
 前端	√	√
 客户端性能	√	
 体验丰富度	√	

角色与重心



Example



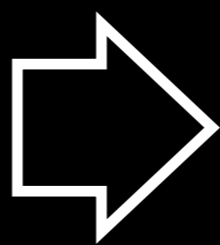


G.18

五润厨房置物架免打孔壁
挂碗架沥水架收纳架30...

¥75 

[\[找相似\]](#) 小二为你精选出了更多
相似商品，快点击这里 >>



G.18

五润厨房置物架免打孔壁
挂碗架沥水架收纳架30...

¥75 

G.18 理想生活狂欢节
24小时尖货闪购 >>

业务变更 —— 变更打底内容



...

```
if ( item.content.isEmpty() )
```

```
then
```

```
    item.content = item.similarItems;
```

...



...

```
if ( item.content.isEmpty() )
```

```
then
```

```
    if ( item.is618() )
```

```
    then
```

```
        item.content = 618;
```

```
    else
```

```
        item.content = item.similarItems;
```

...



Tangram 2.0



Tangram App Container 开源

Tangram Operation 开源

New Tangram Item Technology

Tangram Page Container



THANKS!



<http://tangram.pingguohe.net>