

与狼共舞 - 爱奇艺移动业务后台系统 架构设计与优化实践

邢常亮

爱奇艺 技术总监



QCon

全球软件开发大会

成为软件技术专家的 必经之路

[北京站] 2018

2018年4月20-22日 北京·国际会议中心

7折 购票中, 每张立减2040元
团购享受更多优惠



识别二维码了解更多



极客时间

重拾极客精神·提升技术认知

下载极客时间App

获取有声IT新闻、技术产品专栏，每日更新



扫一扫下载极客时间App

AiCon

全球人工智能与机器学习技术大会

助力人工智能落地

2018.1.13 - 1.14 北京国际会议中心



扫描关注大会官网

SPEAKER INTRODUCE



邢常亮

爱奇艺技术总监

负责爱奇艺移动业务后台系统的架构设计与研发管理，具有丰富的互联网高并发系统的架构设计经验。

14年加入爱奇艺，主导完成多次移动业务后台系统的技术架构升级与改造，使得系统经受住了多次业务洪峰的洗礼和考验，支撑爱奇艺移动业务实现从千万级用户到亿级用户的跨越。

在加入爱奇艺之前，曾在IBM工作多年，从事基础中间件方面的研发及技术架构的工作。个人对于高并发、云计算、大数据等方向都有较大的兴趣和热爱。

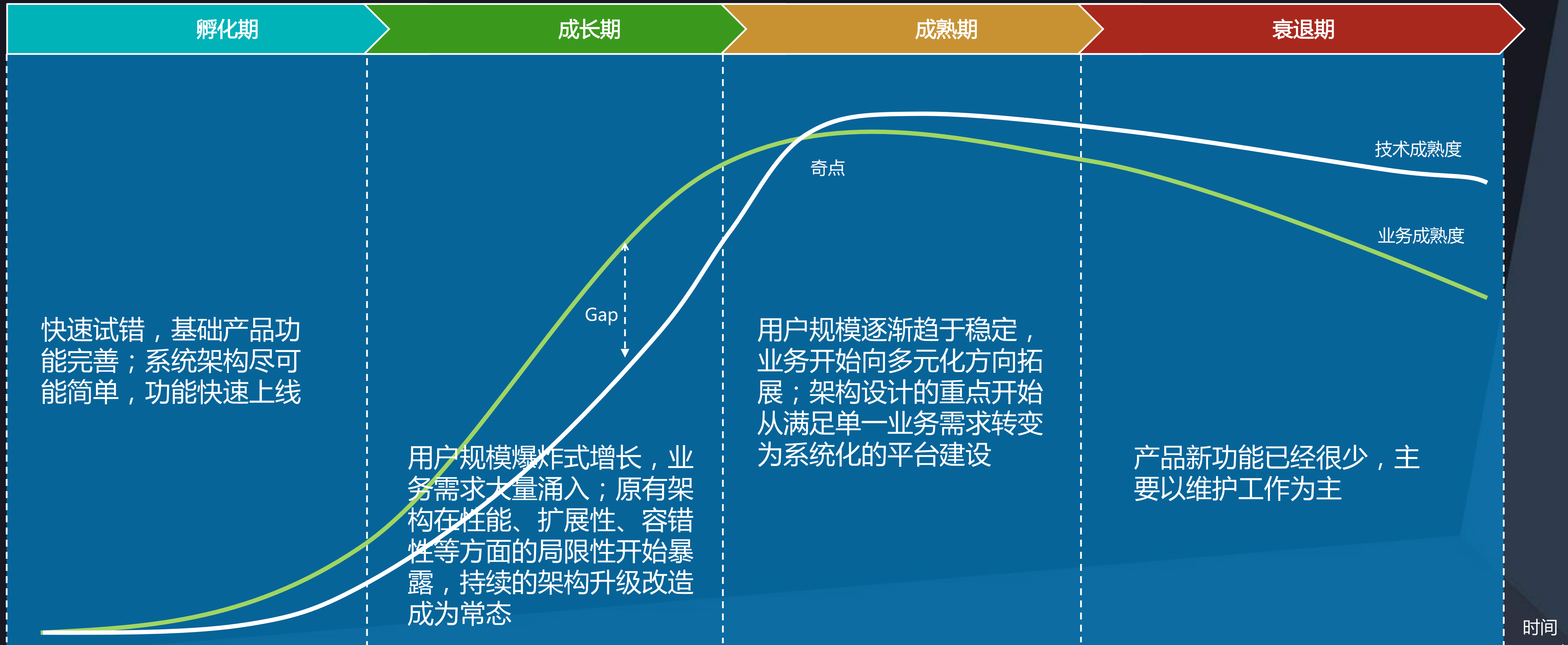
TABLE OF CONTENTS 大纲

- 后台架构演进的必经之路
- 应对流量洪峰的三板斧
- 后台架构设计的发展思考

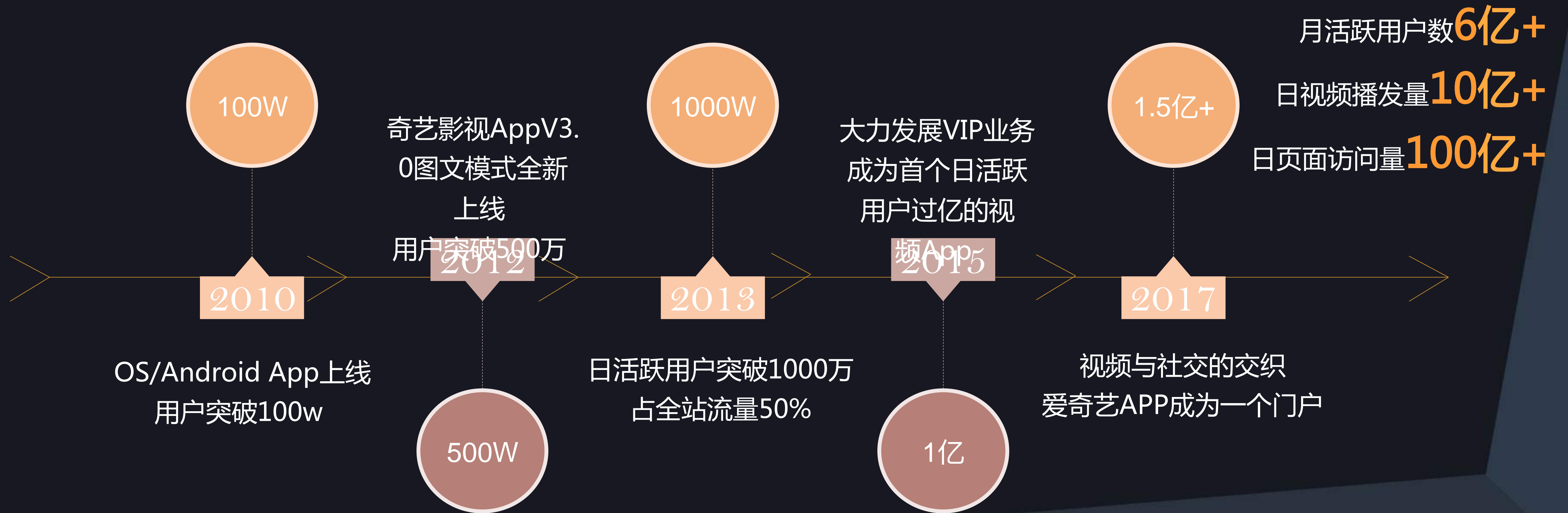
PART ONE

后台架构演进的必经之路

一个互联网产品的生命周期



爱奇艺移动APP的发展历程

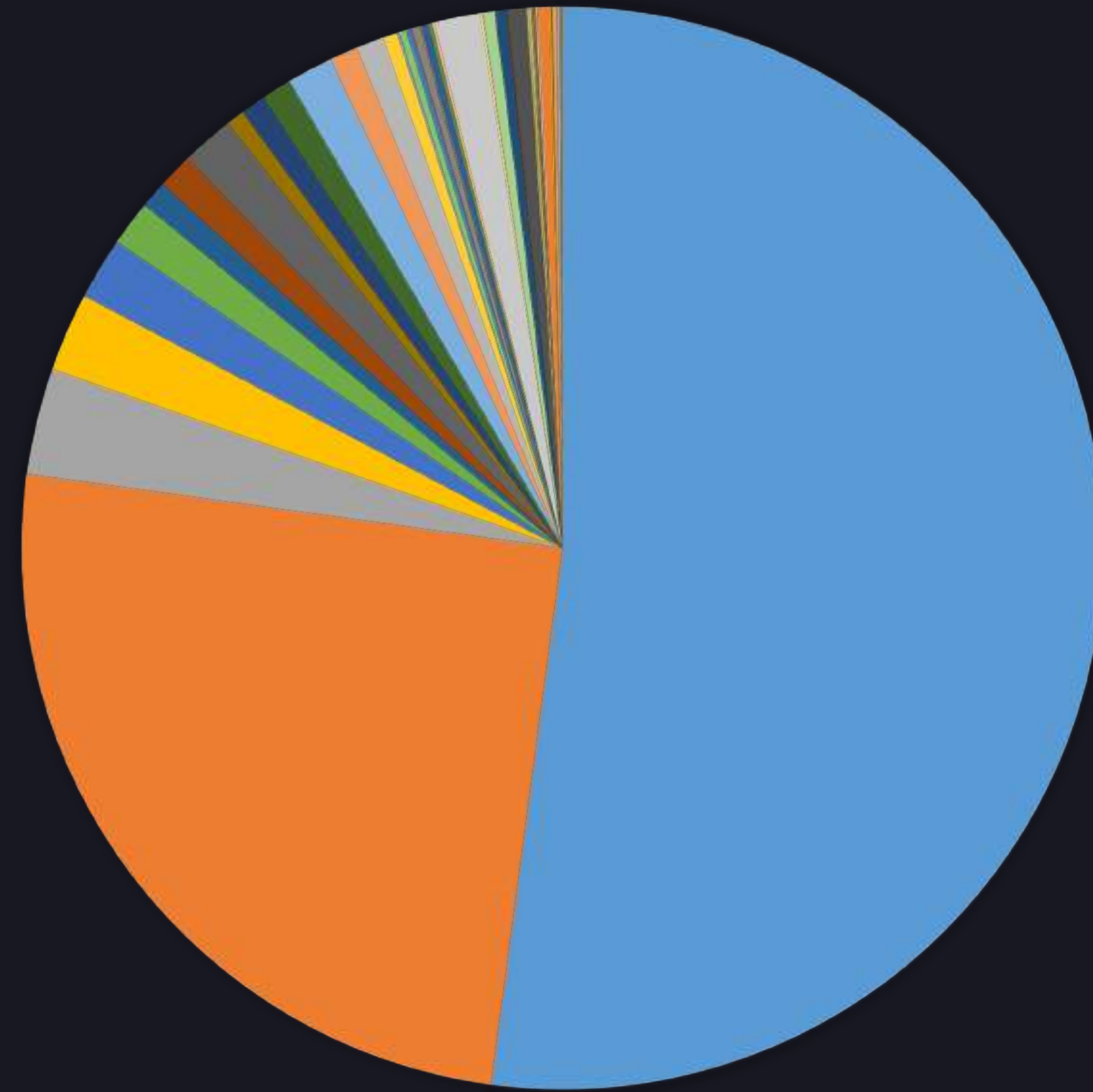


爱奇艺移动后台的架构升级



面临的挑战

APP版本快速迭代，版本碎片化问题严重，导致后台接口背负的历史包袱越来越重



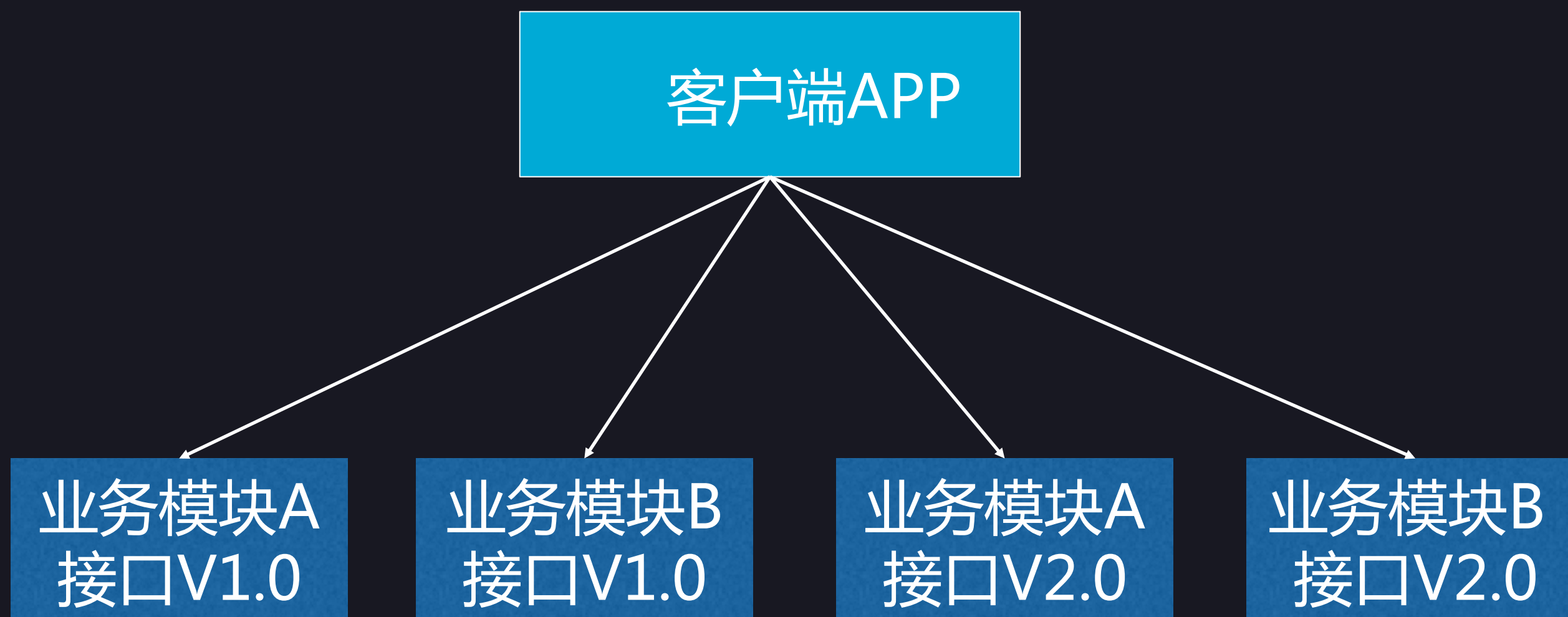
业务需求大量涌入，研发疲于应付，没有精力进行架构设计与代码重构，质量急剧下降

客户端APP版本占比

2014 - 化整为零

两个原则：

- 按照业务模块自上而下垂直拆分
- 引入接口版本概念进行持续迭代



改造成果：

- 系统从1个拆分成了首页、播放、频道页、会员、搜索等6个功能独立的子模块
- 代码工程从1个拆分为6个
- 接口个数从3个拆分为30+个
- 代码中“超级类”消失不见
- Redis缓存从1个大集群拆分为5个小集群

案例数据：

- 首页接口2.0升级改造后，接口参数从60多个精简到20+个，返回数据包大小较少了50%，接口响应时间缩短了一半

2015 - 华丽转身

基础框架重新积累

开发人员技能转型

挑战

新功能重复开发

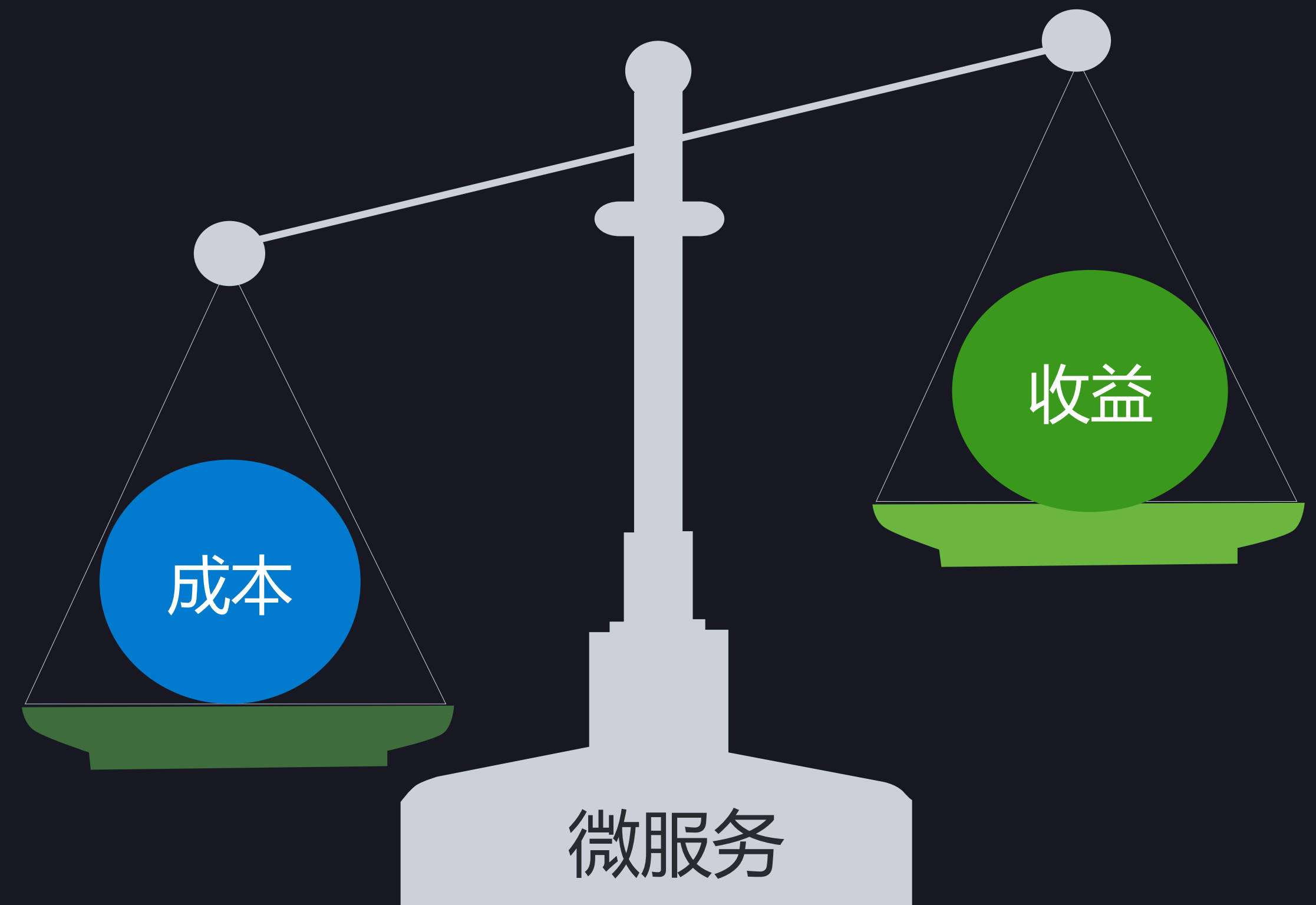


2016 - “微” 服务化

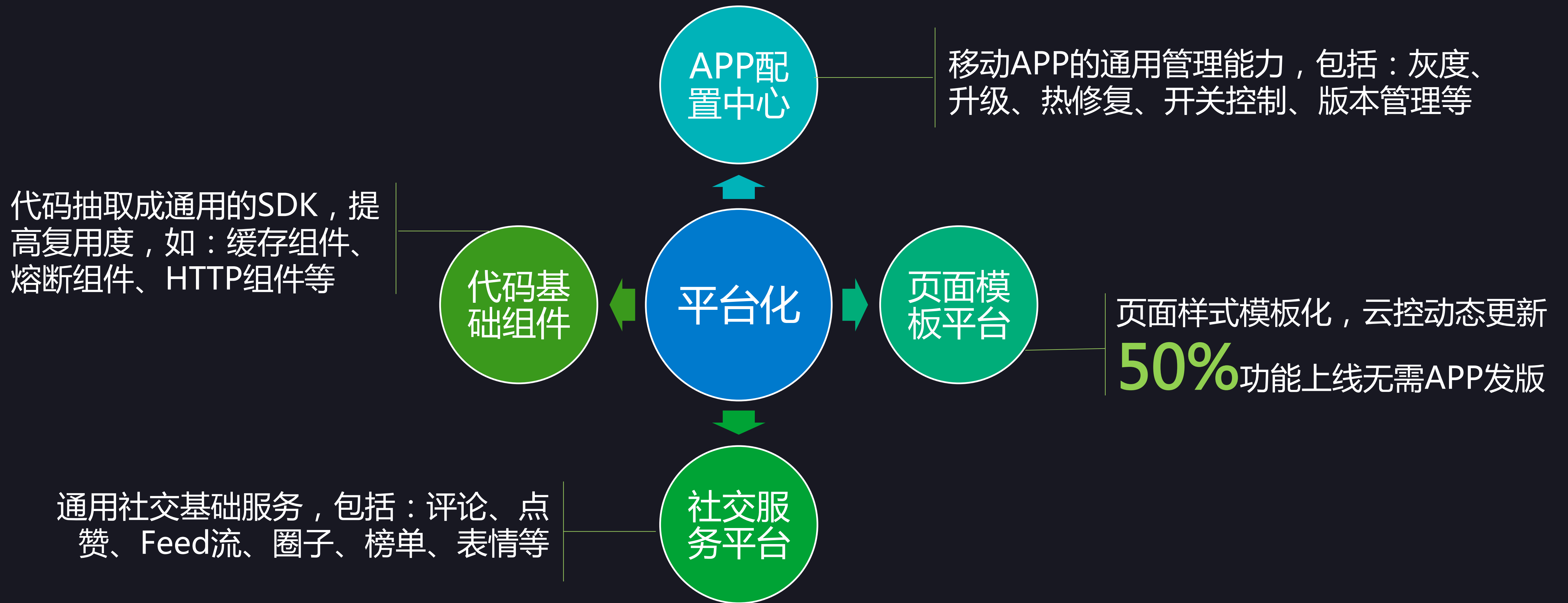
微服务 \neq RPC \neq Spring boot \neq 容器 && 有成本

采取的策略：

- 有节制的进行基础服务的抽取
- 有节制的引入新的技术与框架



2017+ - 平台之路



PART TWO

应对流量洪峰的三板斧

一部热剧引发的“血案”



sina 新闻中心 传媒

爱奇艺“盗墓悲喜剧”:会员付费引发服务器宕机

2015年07月06日09:57 娱乐资本论

刚刚过去的48小时, 爱奇艺经历了“悲喜交加”的一幕幕。

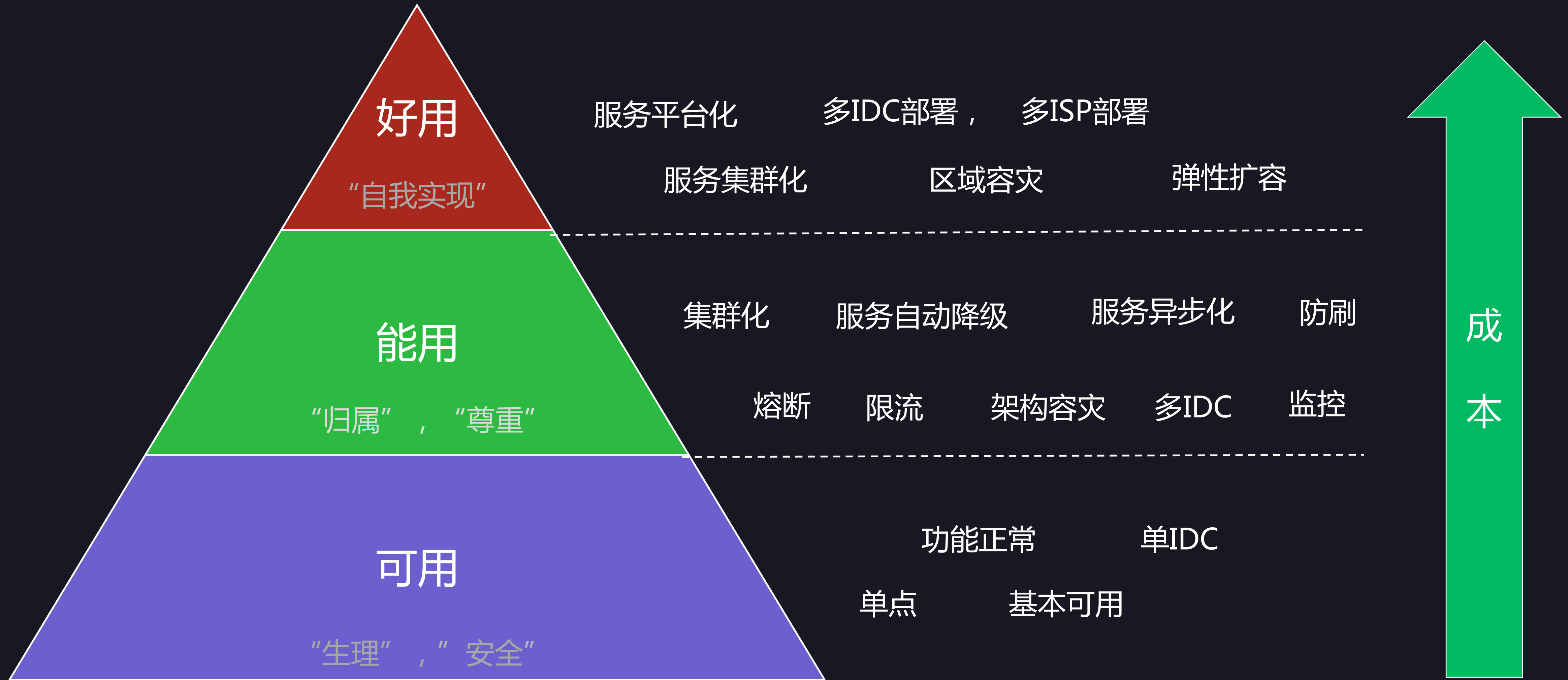
7月3日晚8点, 《盗墓笔记》在爱奇艺全集上线, 由于必须是“爱奇艺VIP会员”才能观看全集, 这掀起了“充值付费潮”。然而, 瞬间涌入的流量, 让爱奇艺服务器宕机了! 最多时有超过70%的VIP会员无法观看《盗墓》。

经过爱奇艺的攻城狮和程序猿连夜抢修, 当晚11点, 60%的VIP会员恢复了正常观看。7月5日, 爱奇艺更是宣布, 7月5日24点前充值的所有会员, 都将赠送2周的VIP会员权益。

“大宕机”逐渐风平浪静之后, 几个重要的数据引发关注——爱奇艺app冲上了App Store排行榜“免费榜”的第一名、“畅销榜”的第二名。这意味着:

- 1、爱奇艺app的下载量在App Store排名第一;
- 2、爱奇艺app在盗墓全集上线后几十个小时内产生总收入在整个苹果商店排名第二。

一个后台系统的“马斯洛模型”



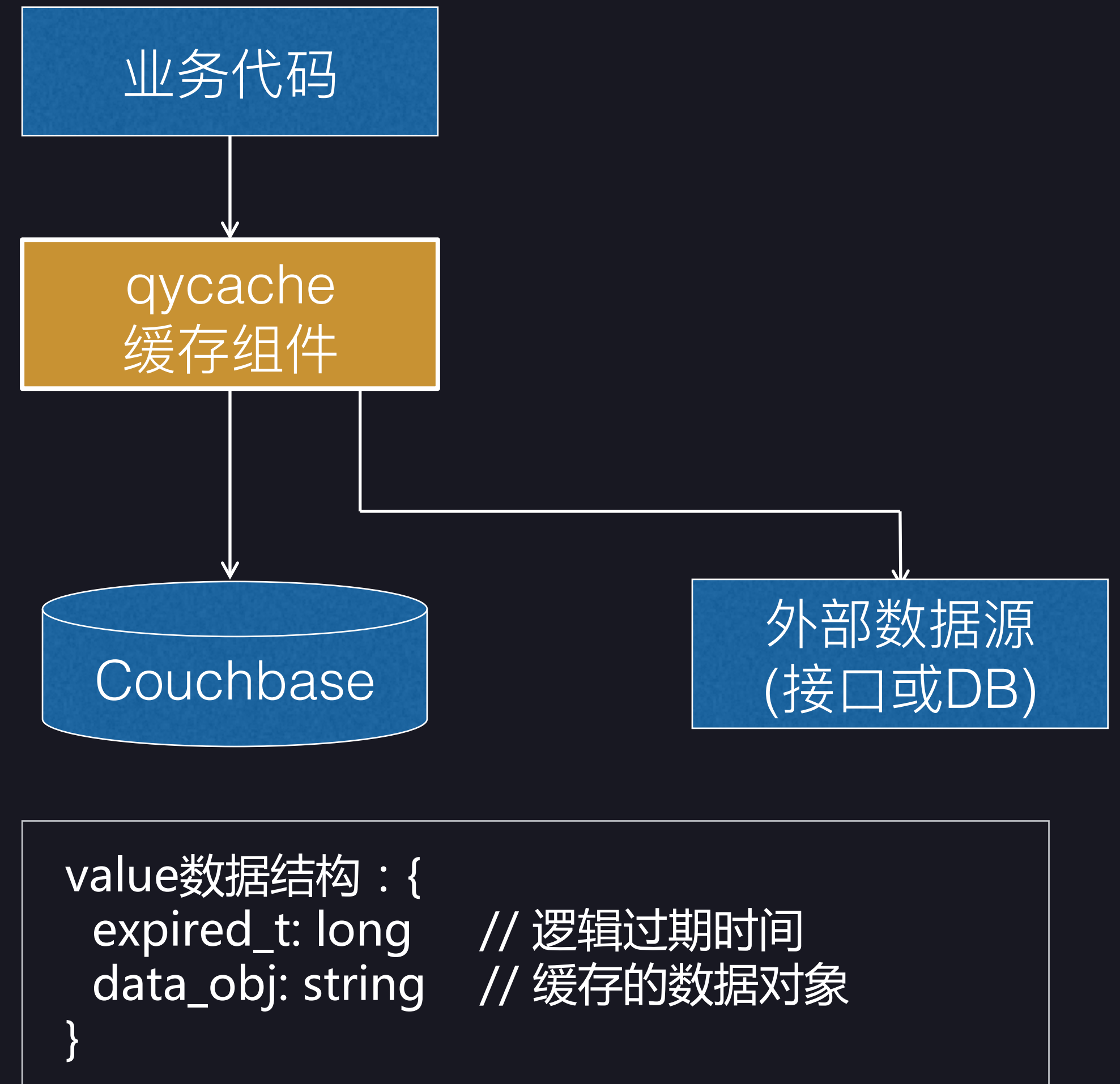
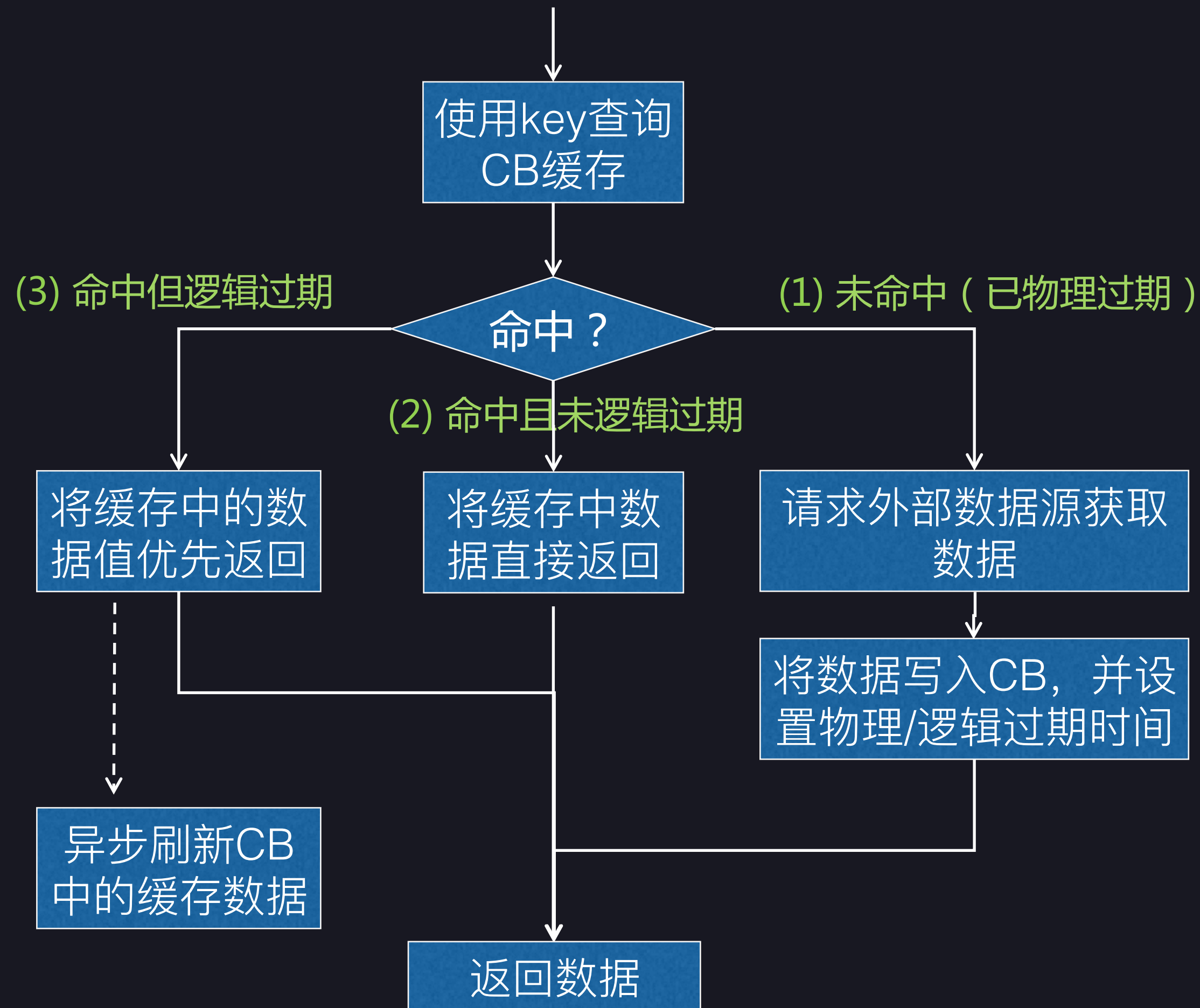
快 - 缓存

“计算机科学只存在两个难题：缓存失效和命名。” —— Phil KarItton

使用到的缓存技术：

- 视频CDN：码流文件
- 静态CDN：图片，资源文件
- 缓存中间件：热数据，中间计算结果，特定场景的持久化数据
- 接口本地缓存：热数据，响应时间敏感，实时性要求不高
- APP本地缓存：伪静态资源，减少接口访问压力，降低接口依赖

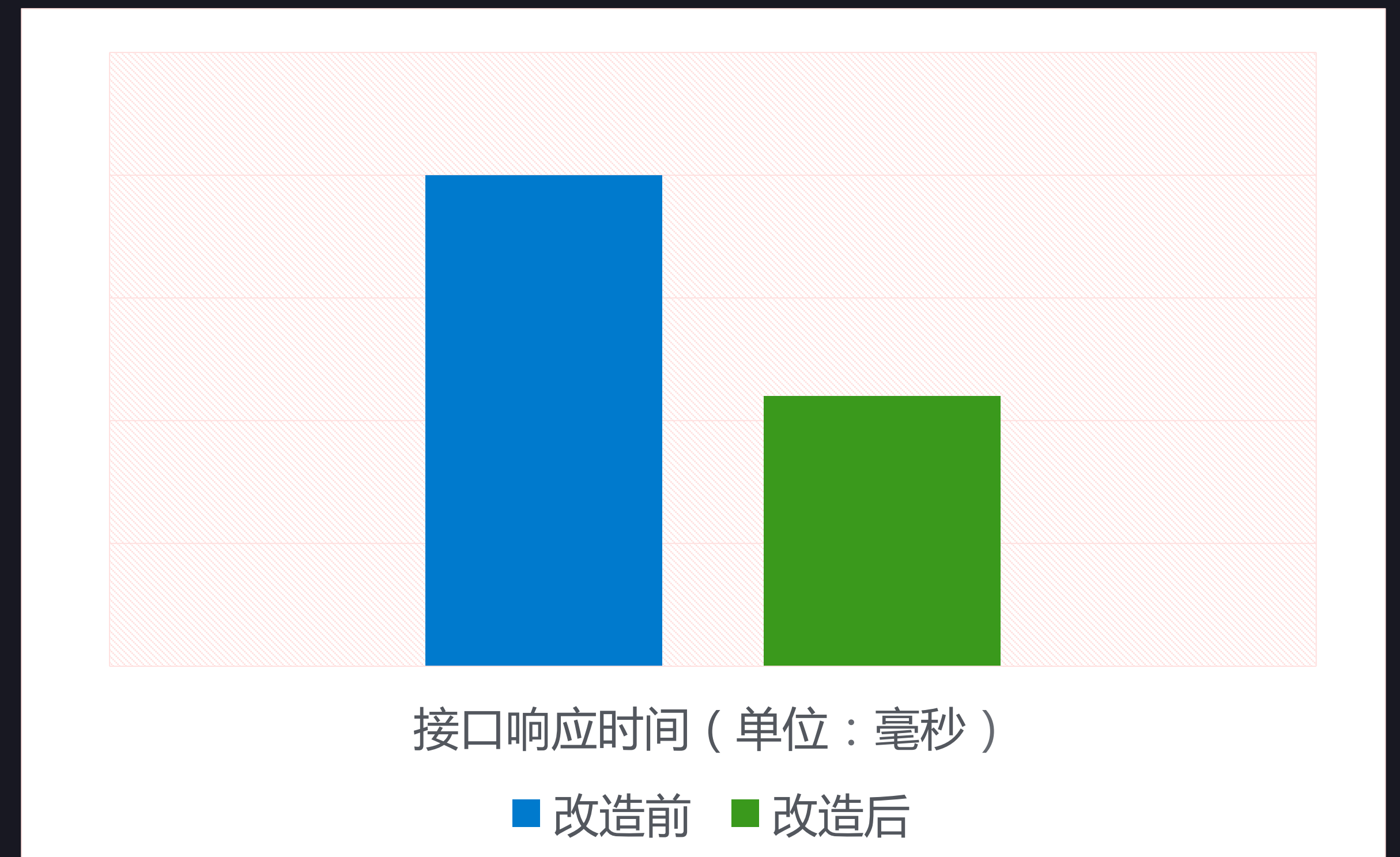
缓存组件 – qycache



快 - 异步化

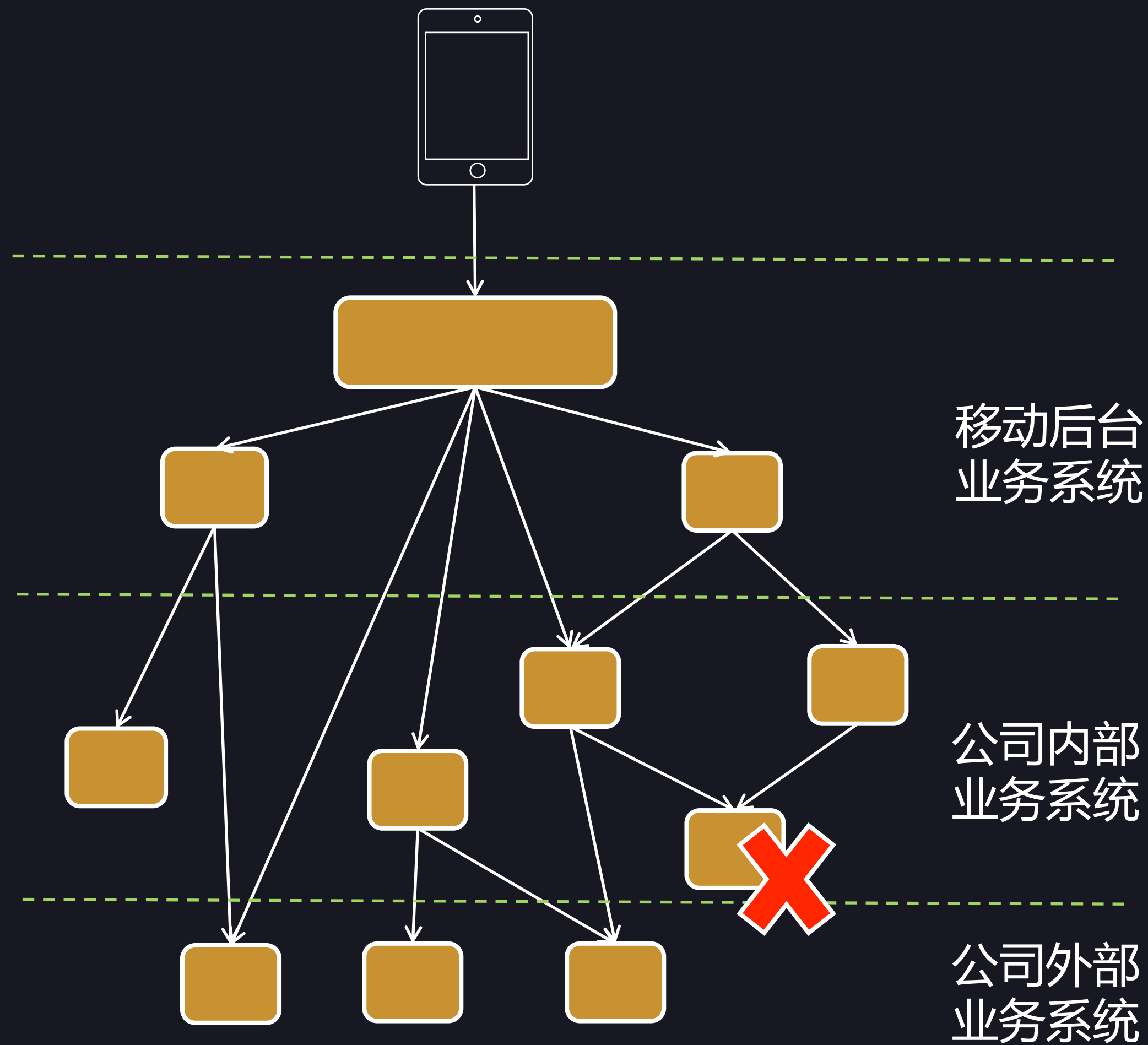
- 使用的技术
 - 队列：MQ，Kafka
 - 子线程并行处理，避免串行等待
- 收益
 - 缩短接口处理时间，提高系统吞吐率
 - 非阻塞式调用，减少资源占用
- 教训
 - 过渡的异步化会降低系统的可维护性

案例：频道页接口异步化改造后，响应时间缩短**40%**



稳 - 服务降级

在复杂交错的网络调用链路中，一个局部的故障会产生“连锁反应”，可能导致整体系统的雪崩。



服务降级策略

- 熔断
- 安全数据
- 限流

使用到的技术

- Hystrix熔断器
- Meerkat (自研组件)
- Nginx+lua

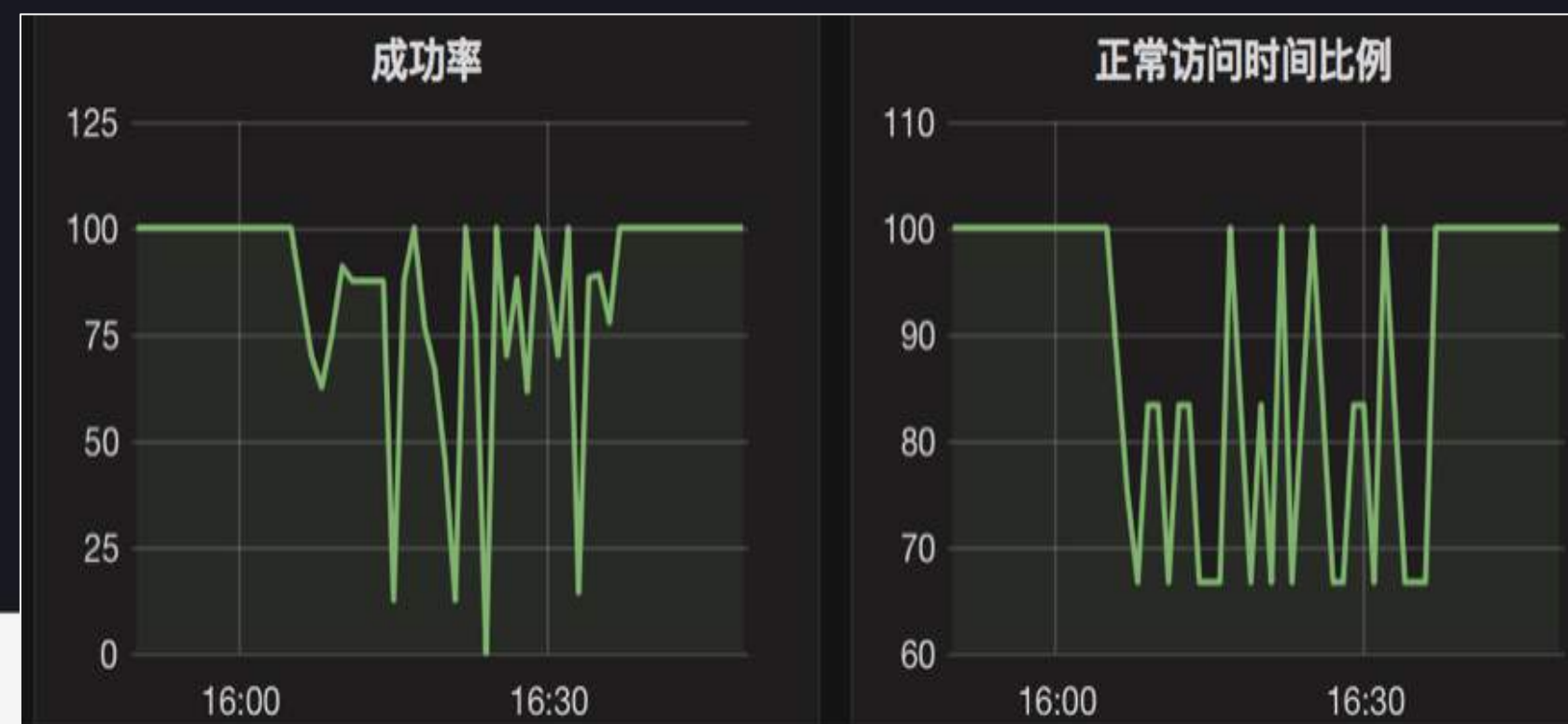
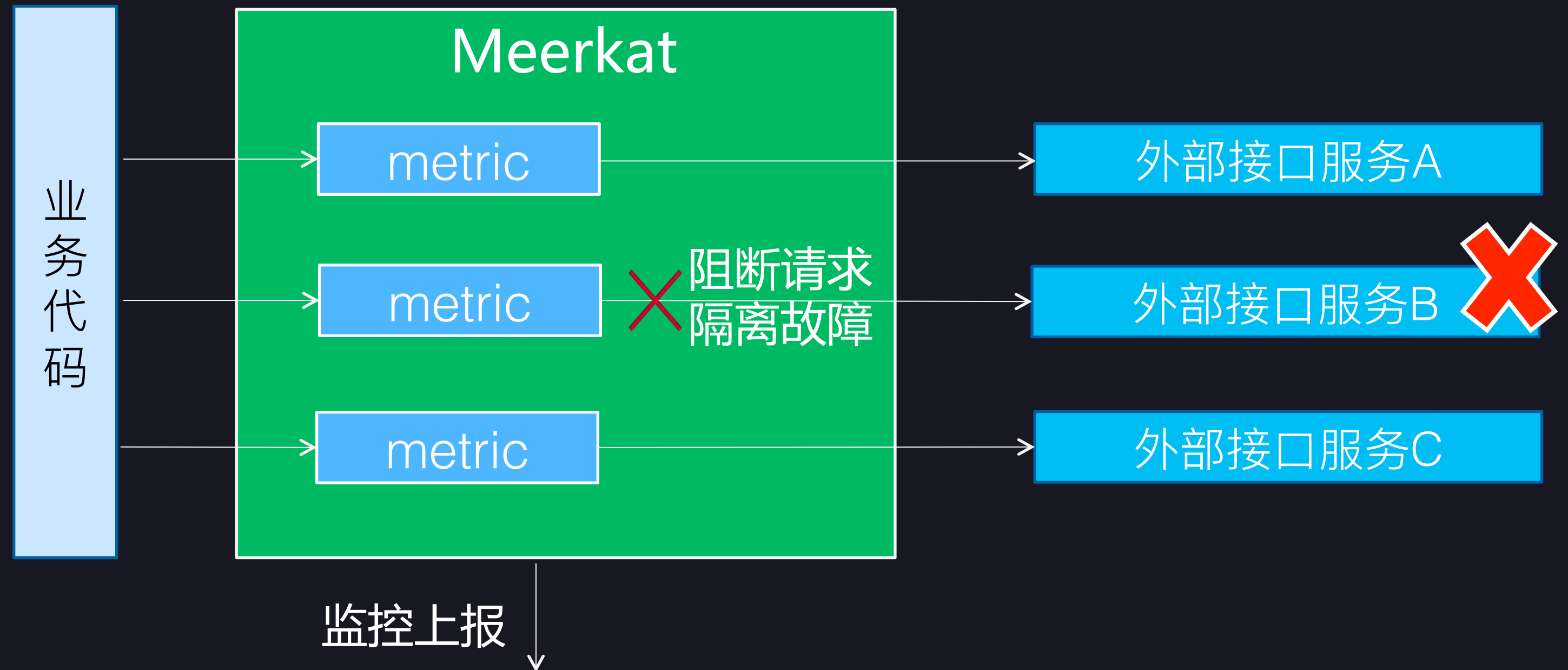
熔断组件 - Meerkat

熔断

- 故障隔离
- 自动恢复
- 配置动态加载

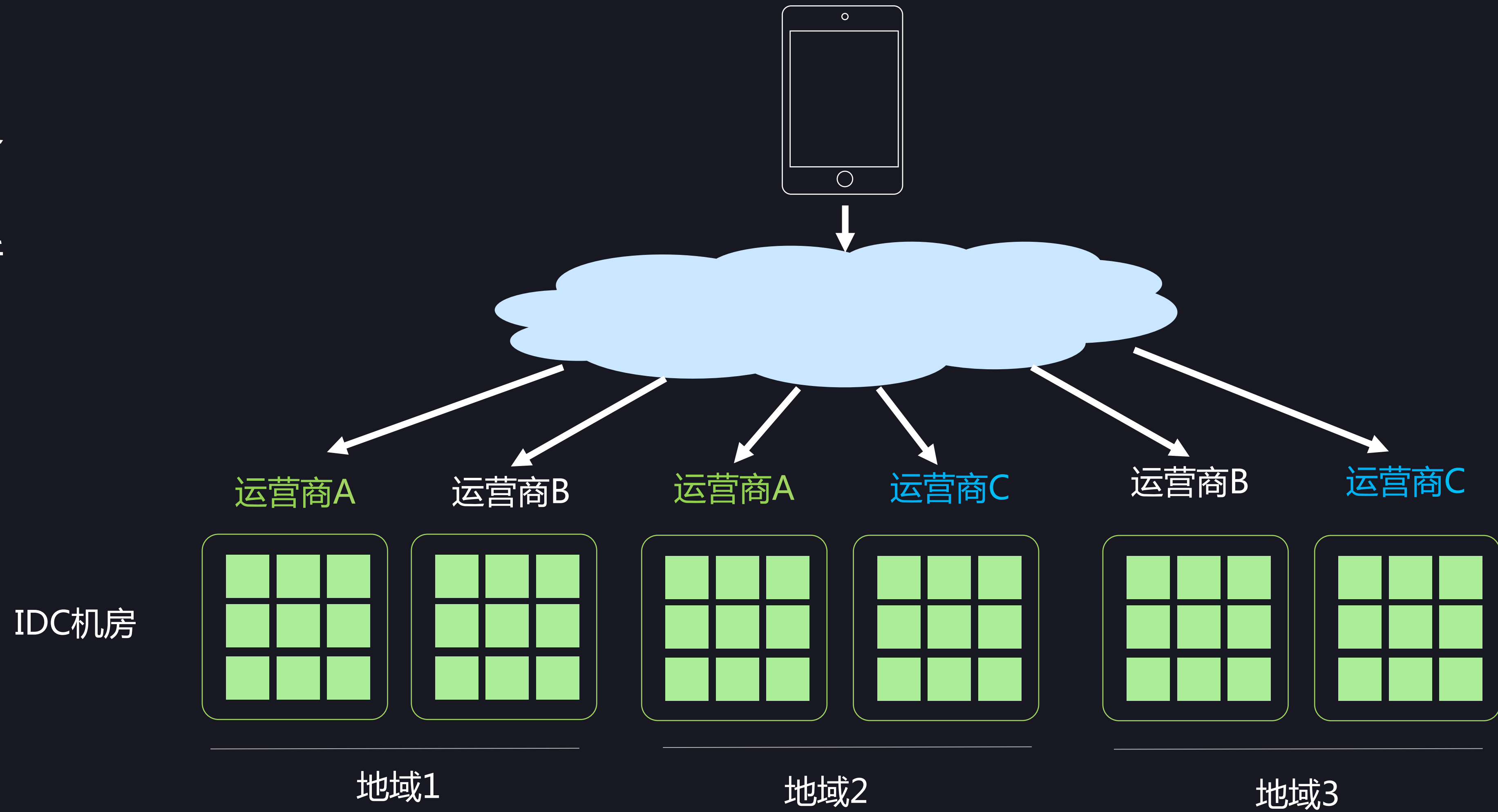
监控

- 接口调用监控
- 熔断历史记录
- 自定义扩展



稳 - 多机房HA

- 区域性灾备
- 多机房双活



准 - 监控

“缺少监控的线上系统就像是闭眼飞行”

关注的元素

- 系统层面：CPU，内存，网络
- 应用层面：QPS，时间，成功率
- 业务层面：逻辑正确性

使用的技术

- Python
- Zabbix + Grafana + 时序DB
- 邮件、短信、IM

PART THREE

后台架构设计的发展思考

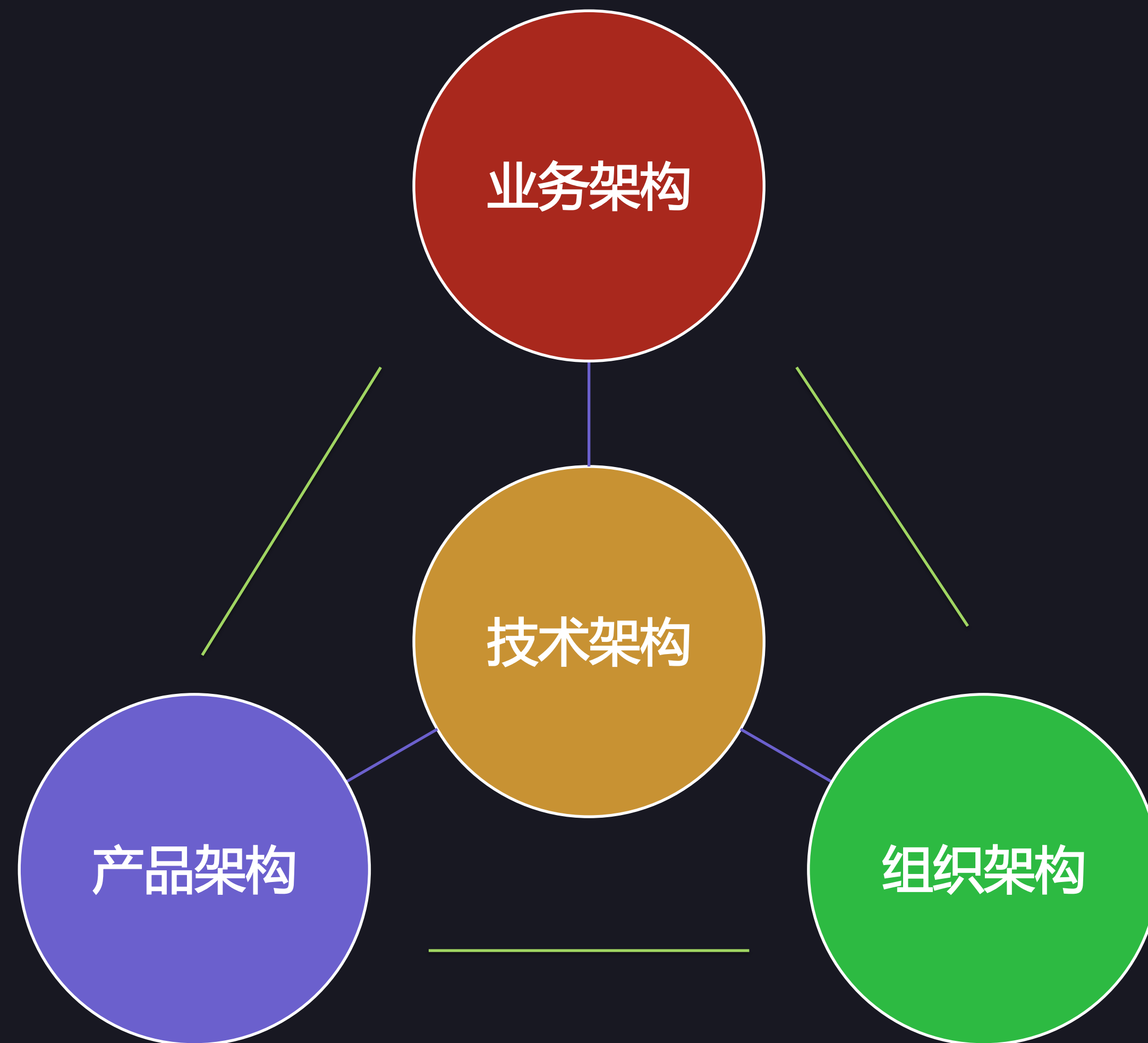
如何设计一个好的后台架构？

面向对象，分层，模块化，SOA，微服务，容器，云计算，Devops，人工智能，VR，深度学习，BI，大数据，虚拟现实，人机交互，区块链

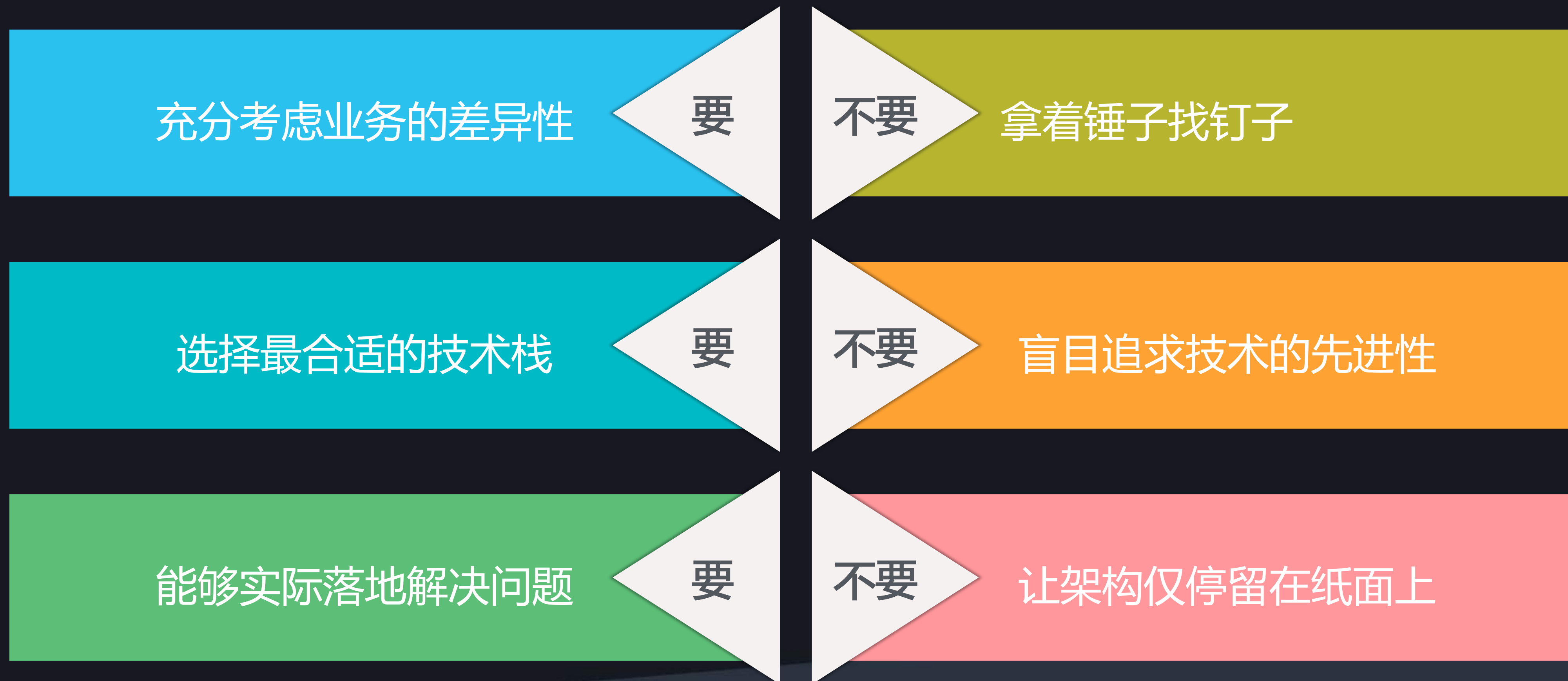


C++，Java，Go，Spring，Tomcat，Jetty，Apache，HDFS，Storm，Hadoop，ES，Thrift，Nginx，MQ，Docker，Swarm，Kubernetes，Kafka

影响技术架构的关键因素



架构设计的“三条军规”



好架构需要持续不断的打磨

$$0.99^{365} = 0.03$$

$$1.01^{365} = 37.8$$



检验架构好坏的最终标准 – 业务成功



总结 & 问答

- 后台架构演进的必经之路
- 应对流量洪峰的三板斧
- 后台架构设计的发展思考



THANK YOU

如有需求，欢迎至 [讲师交流会议室] 与我们的讲师进一步交流

