

APKBUS

The background of the slide is a blue-tinted image of a computer monitor. The monitor displays a dashboard with a line chart at the top showing data trends over time, and a pie chart below it. The text 'APKBUS' is overlaid on the left side of the monitor image.

蚂蚁金服 mPaaS 模块化开发与架构重构深度解析

王磊

APKBUS

1

支付宝在移动端的架构演进与思考

2

mPaaS 模块化架构及能力

3

基于 mPaaS 架构的重构思考

APKBUS

1

支付宝在移动端的架构演进与思考

2

mPaaS 模块化架构及能力

3

基于 mPaaS 架构的重构思考

支付宝的三个时代

独木舟



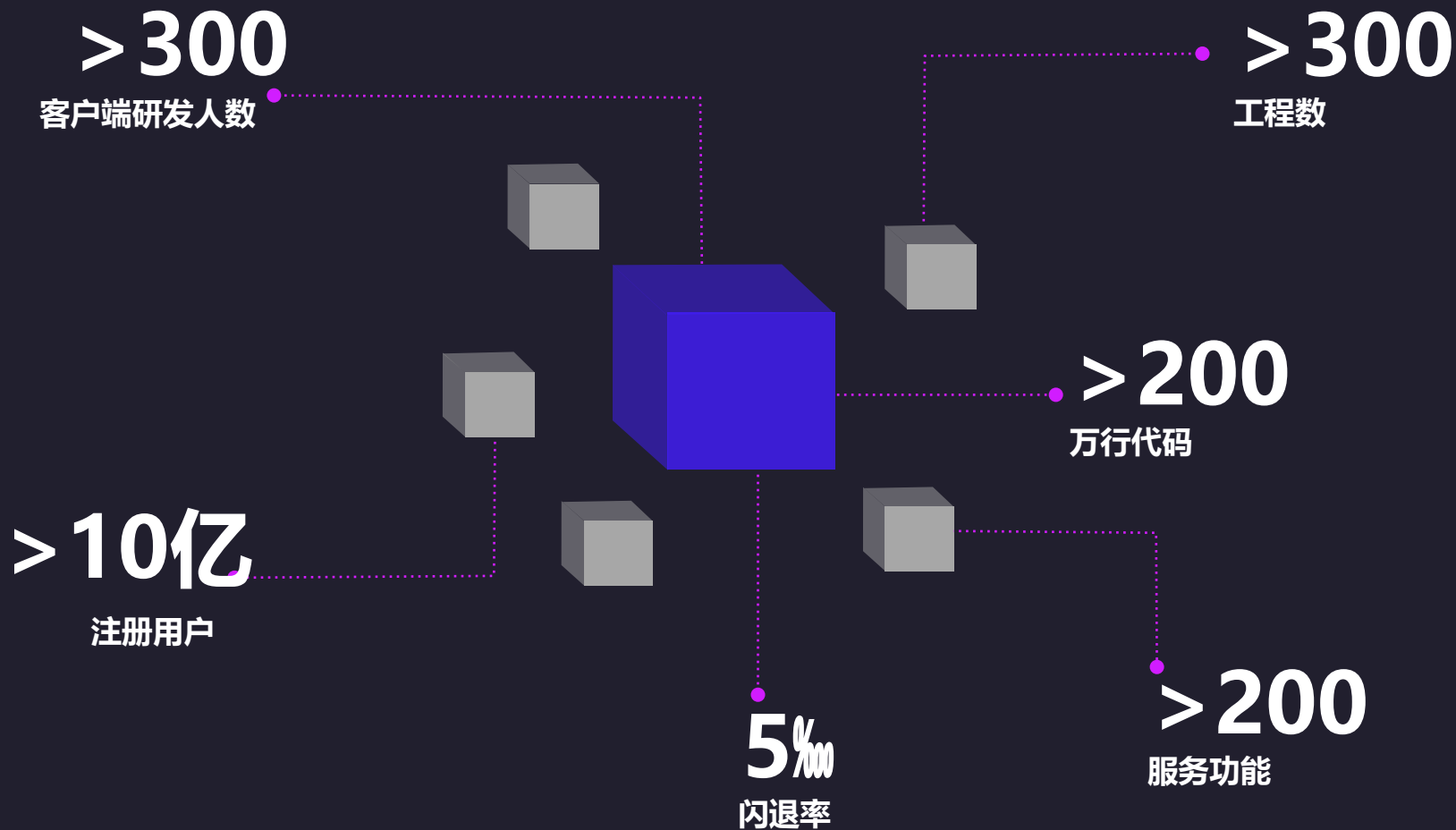
战列舰



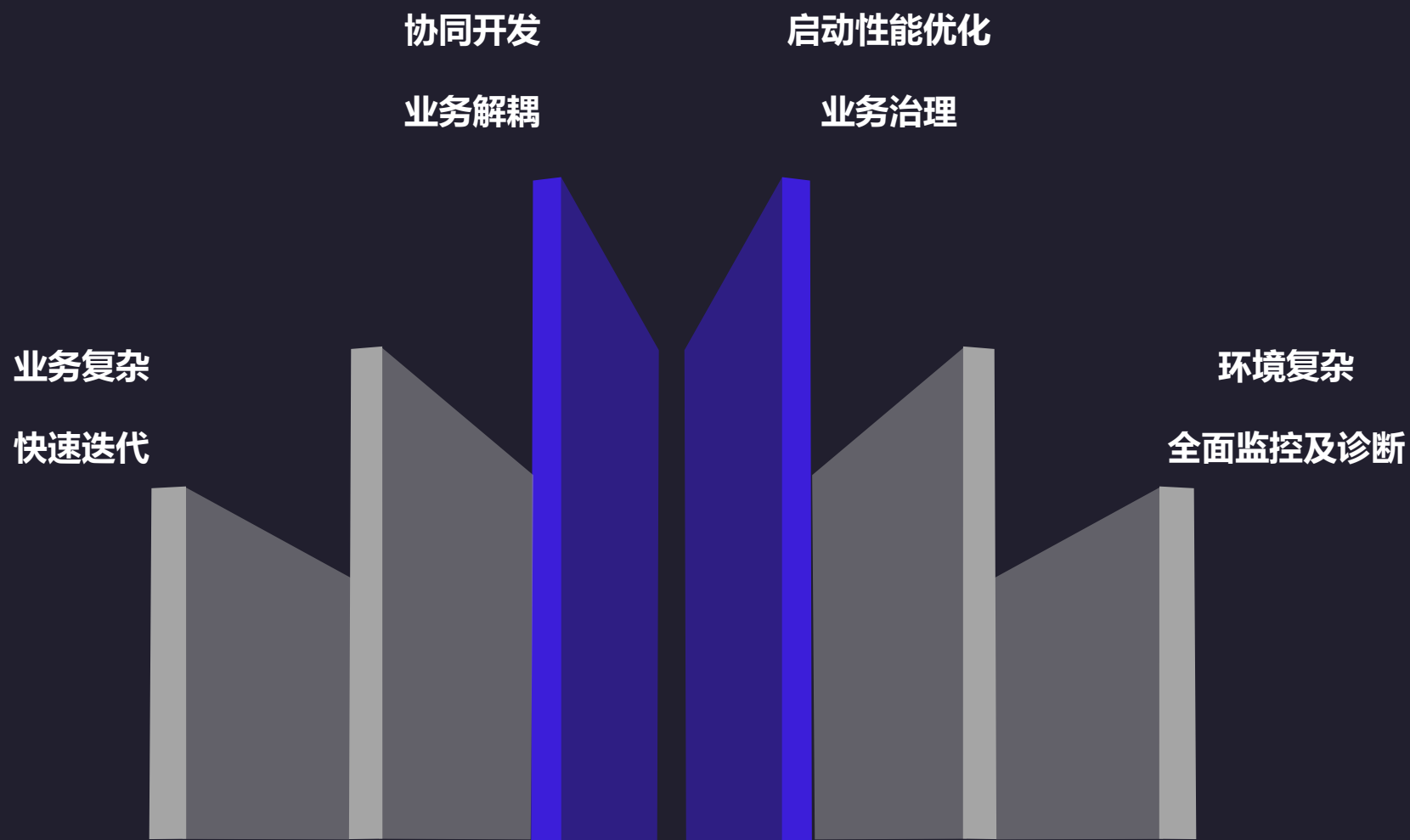
航空母舰



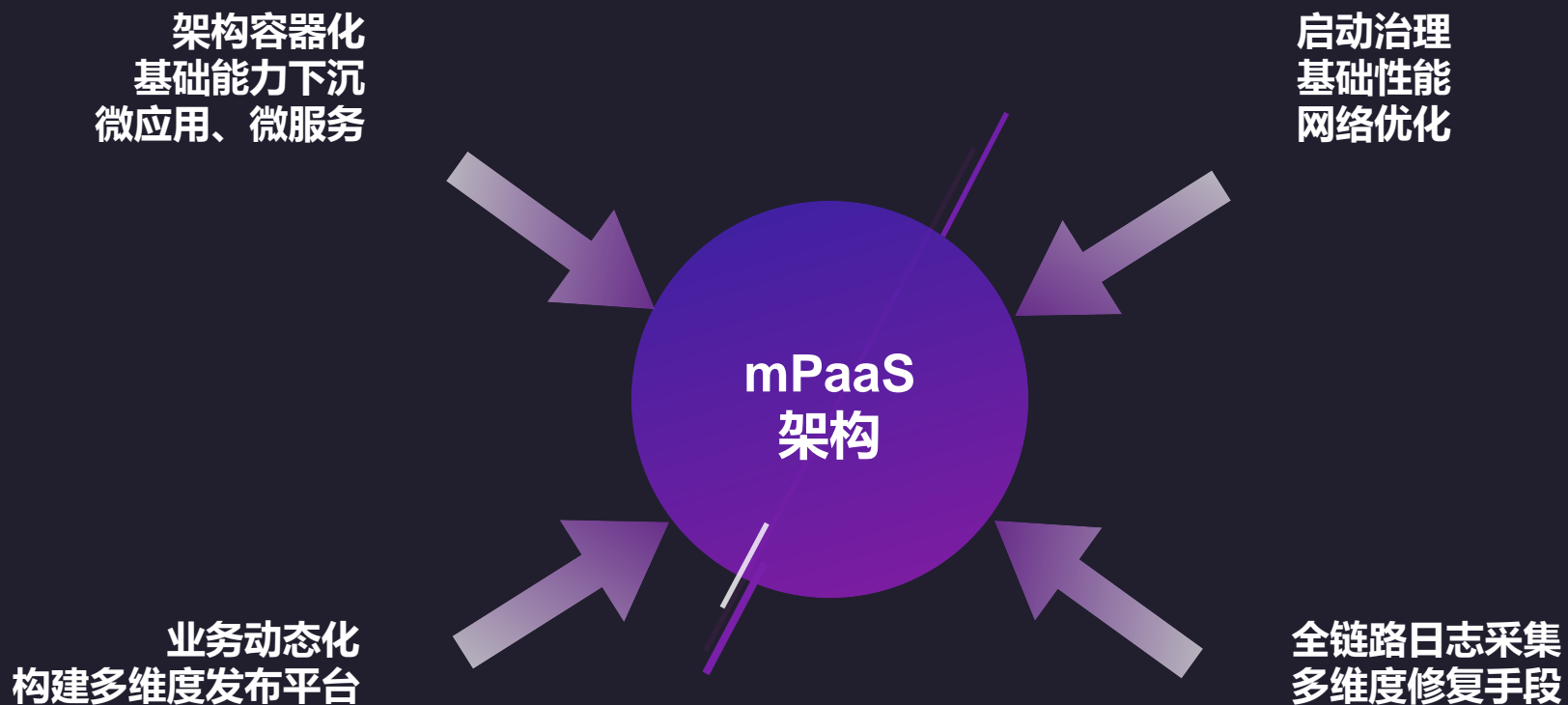
现在的支付宝



业务复杂性带来的技术挑战



解决方向



APKBUS

1

支付宝在移动端的架构演进与思考

2

mPaaS 模块化架构及能力

3

基于 mPaaS 架构的重构思考

mPaaS 模块化架构图

业务层

余额宝

口碑

花呗

蚂蚁森林

财富

服务层

人脸识别

OCR 智能

设备指纹

智能投放

定位服务

组件层

安全

网络

多媒体

H5 容器

UI 组件

发布

框架层

容器管理

资源管理

框架

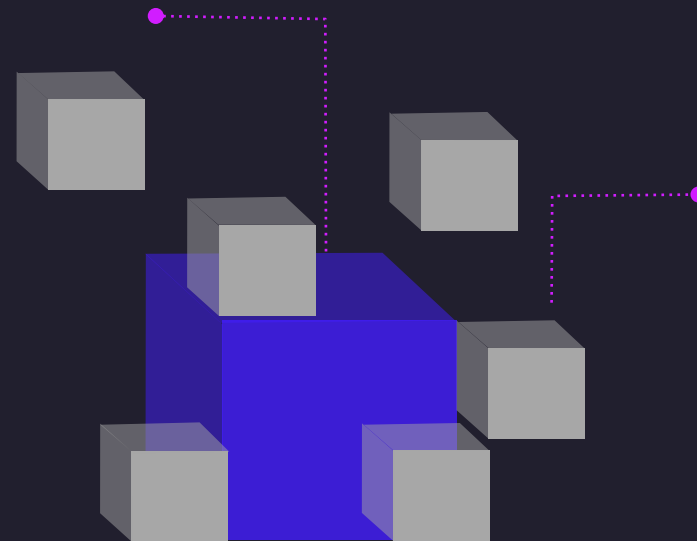
切面

日志监控

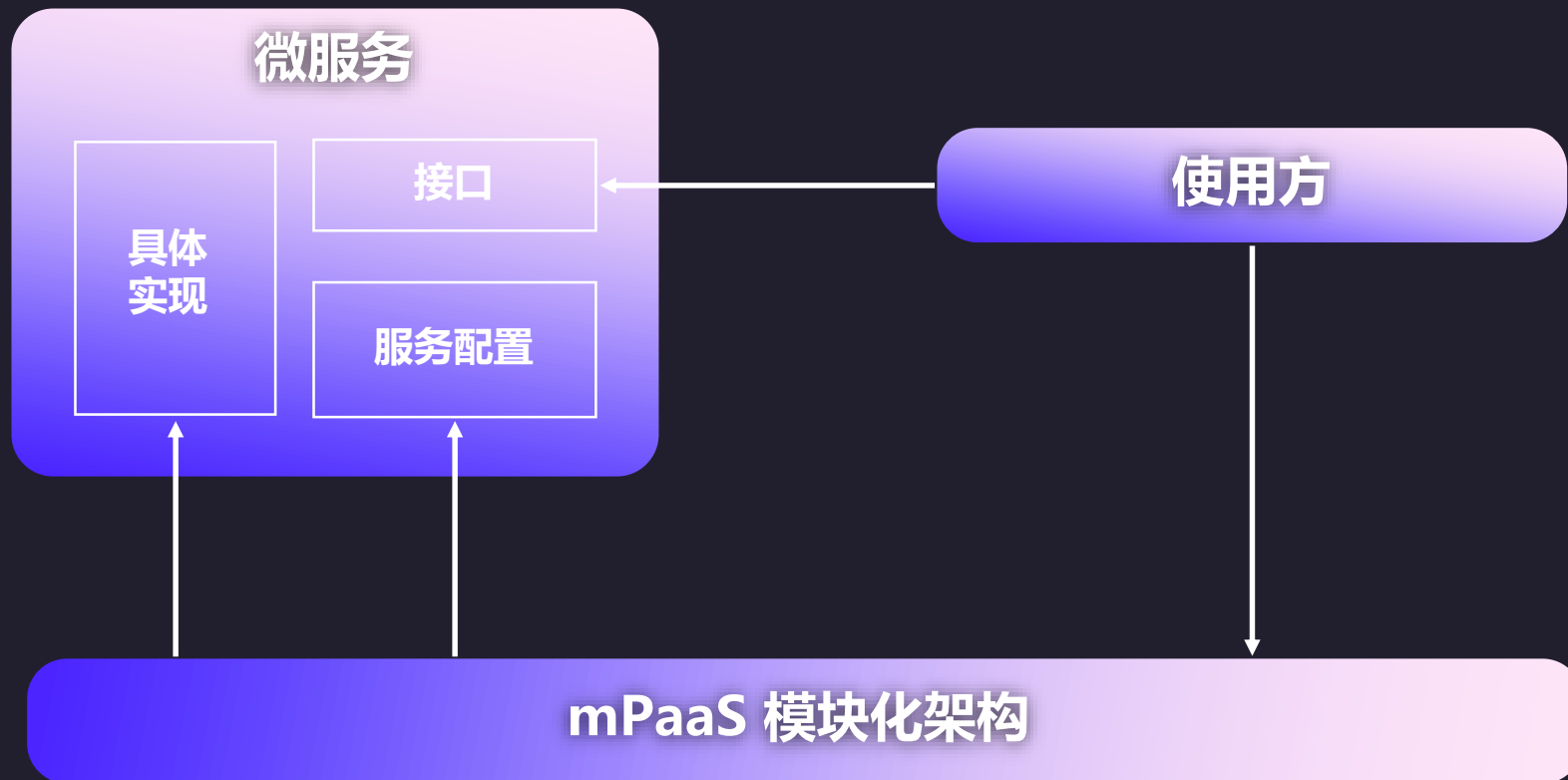
Quinox 架构

基于 mPaaS 的 App 研发，就像积木搭建（Bundle）一样轻松

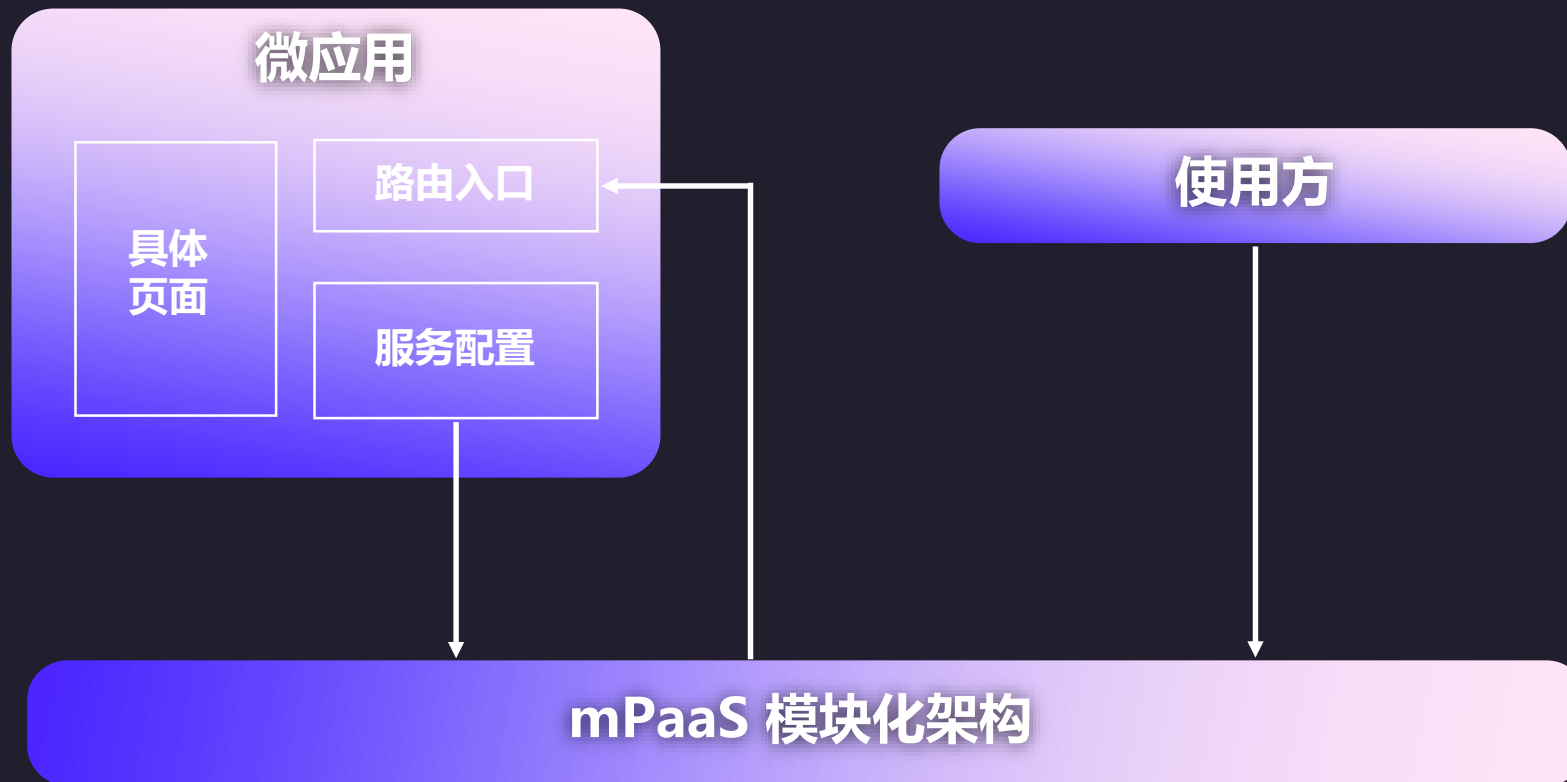
- 灵活性：模块/组件可插拔，不影响编译
- 独立性：模块可由团队或个人独立维护
- 复用性：模块可被任意宿主程序所引用
- 隔离性：接口和实现分离，微应用路由跳转



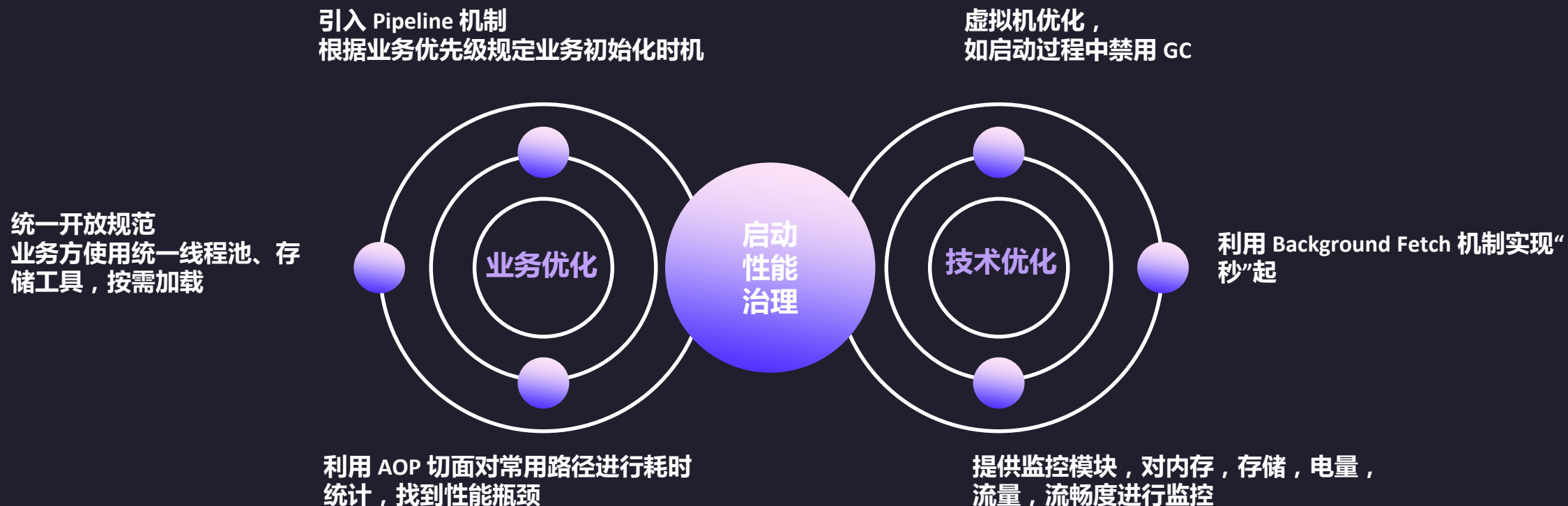
微服务



微应用



启动性能治理



动态化

代码量膨胀，安装包越来越臃肿

集中点发布，迭代慢

依赖系统框架，兼容性差

用户使用的服务数量有限



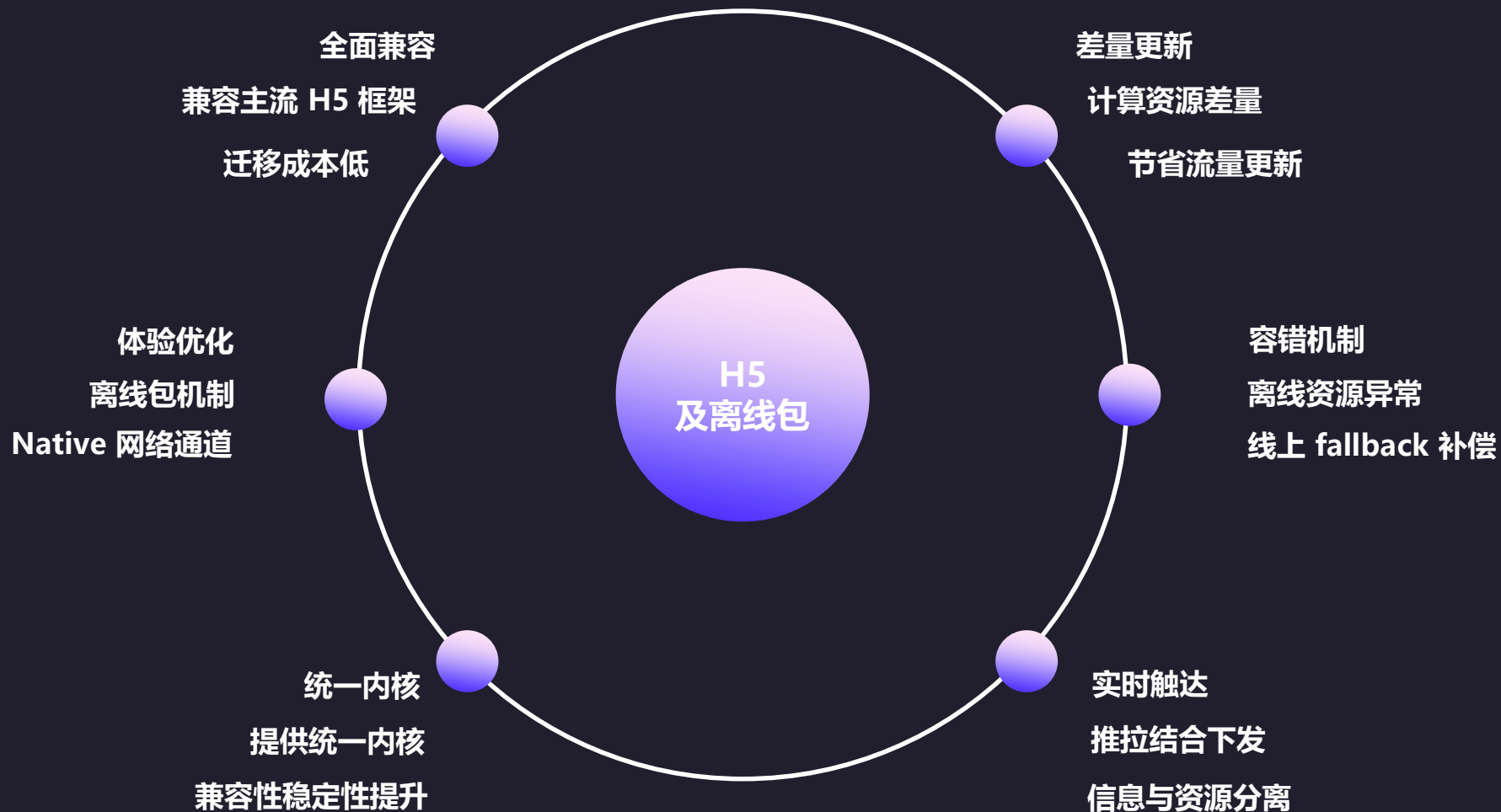
节省流量存储，安装包大小可控

快速发布，各发各的，不依赖客户端发版

一套代码多端运行，统一标准

应用变为平台，服务按需加载

H5 及离线包



小程序



- 统一的小程序内核，可互投在基于 mPaaS 架构的任何应用上
- 依托业内成熟的渲染技术/生态，更强的 Web 渲染引擎，表现力更强
- 提供丰富的组件，帮助业务方快速实现产品
- 整合 mPaaS 框架，离线包技术，复用 H5 插件
- 完善的 App/Page 生命周期，支持内嵌 Widget 模式

多维发布



稳定性保障



通过接口解耦与业务解耦
开发者只需关心自己的业务做
好业务迭代与测试
集成之后验证跑通即可



多纬度的灰度策略：
白名单、时间窗、应用版本号、
系统版本号、手机型号
通过日灰度，周灰度，月灰度，
把风险逐步放大



配置下发
进行 Native 热修复
H5 离线包动态更新业务
小程序动态升级



闪退监控
卡顿监控、卡死监控
业务异常监控、自定义数据
清洗图标
异常警报

APKBUS

1

支付宝在移动端的架构演进与思考

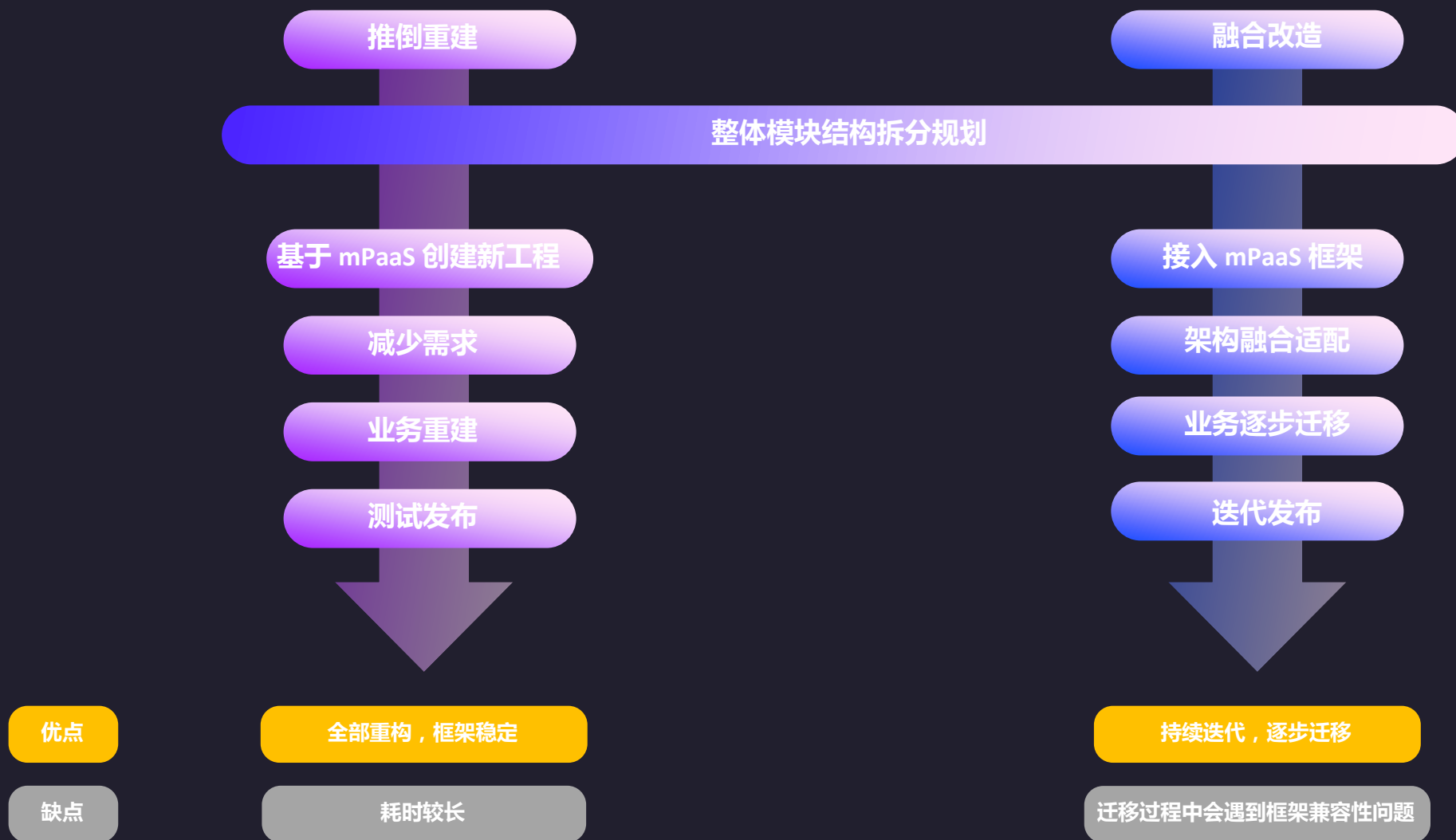
2

mPaaS 模块化架构及能力

3

基于 mPaaS 架构的重构思考

架构重构



能力重构

将应用通用的组件进行沉淀，提炼，解耦

基础
组件
下沉

通过公共 bundle 的形式，对其他 bundle 提供支持

将业务抽象成接口，并以 service 形式提供

核心能力
微服务化

将核心的业务能力进行微服务化

业务团队之间的调用以接口形式，相互无影响

业务重构

使用 mPaaS 提供的微应用的能力，
将核心 native 业务，改造成微应用

核心业务
微服务化

对第三方或业务方提供小程序标准，
将应用升级为生态平台

三方业务
小程序
生态化

复杂业务
H5 化

把二级业务，或者变化较多的业务，
转换成 H5 前端开发，以离线包形式加载

发布迭代重构

App: Native



产品 1 : H5



产品 2 : 小程序



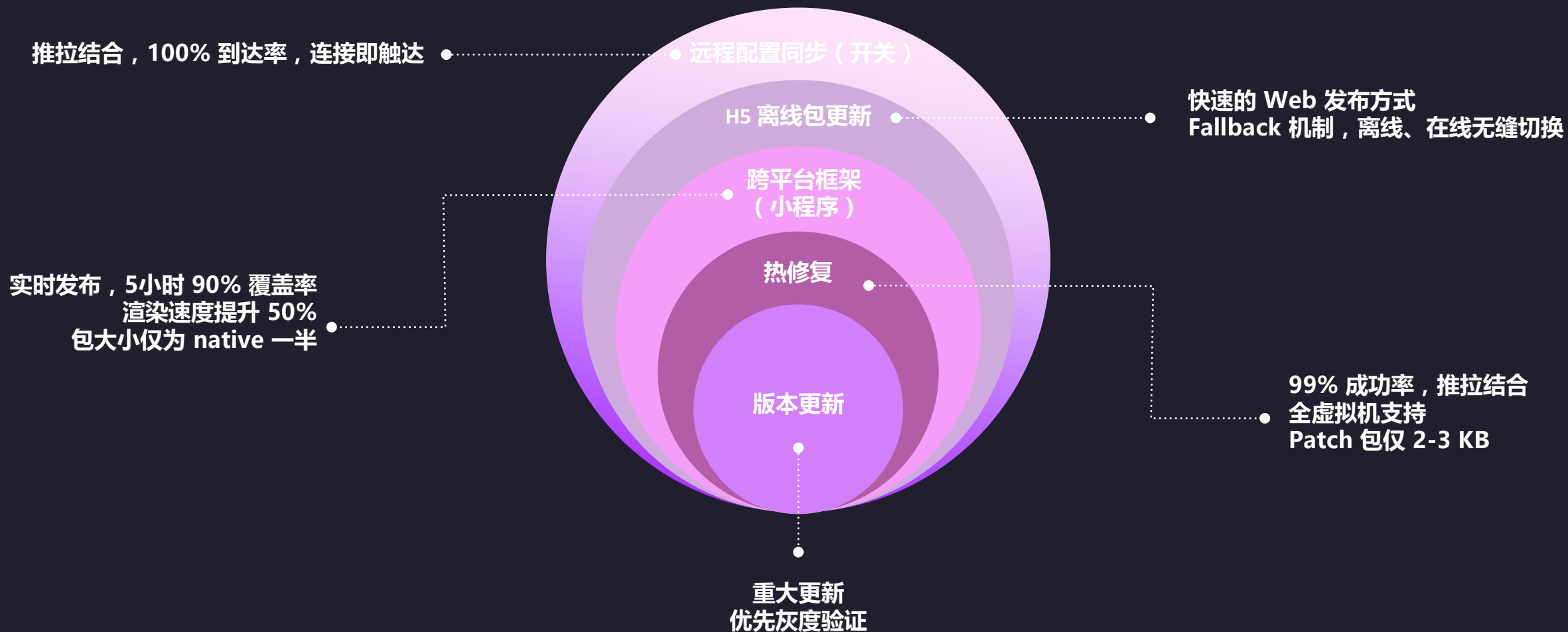
研发模式：

1. 按照业务纬度拆分
2. 各业务独立团队，独立演进，独立发布。
3. Follow 发布流程，规范审批流程 + 研发流程：

发布流程：

1. 指标先行
2. 智能灰度，内灰&外灰
3. 监控 + 修复循环
4. 上线线上质量体系兜底

运维保障体系重构



经典重构案例：12306

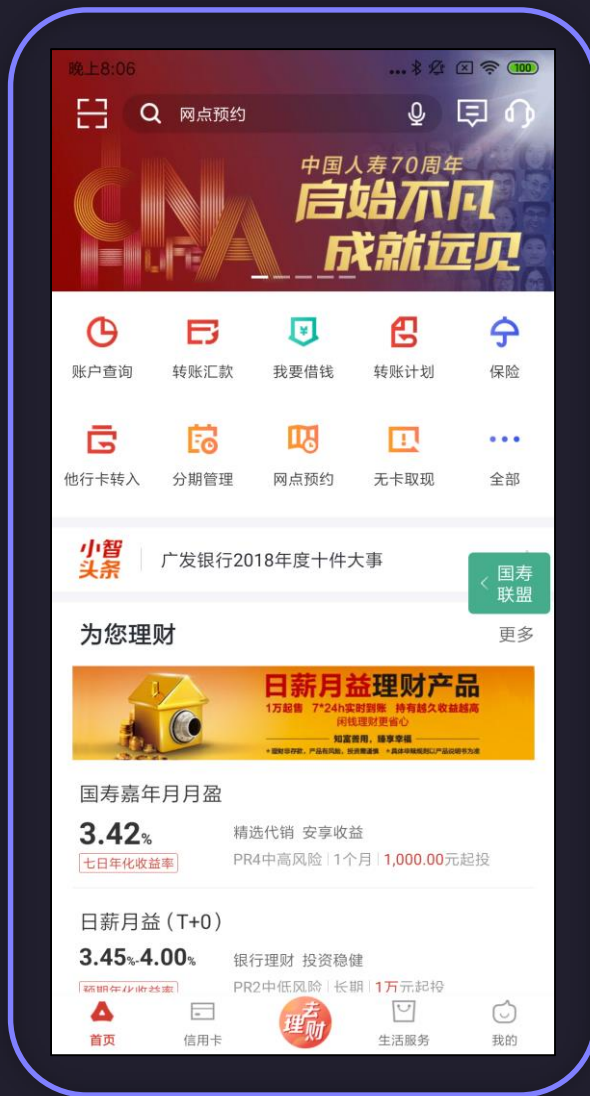


成熟稳定，全程护航春运

17 年底通过 mPaaS 进行了系统的重构，使用了 mPaaS 框架以及多项组件及服务，解决了 12306 移动端开发多年的痛点

作为国民级 App，mPaaS 全程护航
作为开发角度，客户表示当年是移动渠道最轻松的一年
作为使用者角度，多数用户也表示使用体验改善了很多

经典重构案例：广发银行“发现精彩” App



背景

之前采用其他移动端研发平台，性能一直是瓶颈，尤其启动时间，在部分机型上长达 10 秒钟，使用 mPaaS 重构后，平均启动性能稳定在 2 秒左右

研发效能

广发是两款 App 先后上线的，两个 App 均使用 mPaaS 框架进行重构。其内部信用卡模块、账户体系等存在通用的部分，通过 mPaaS 模块化架构拆除可复用的服务和微应用，使得后开发的手机银行客户端在研发工时上节省了近 50% 的时间。

欢迎交流分享



欢迎关注「mPaaS」公众号
第一时间获取 mPaaS 最新技术架构实践与产品迭代资讯。

今日分享文章将在第一时间整理完毕，通过公众号推送，
欢迎指正交流。

Thank You

</> 欢迎关注安卓巴士公众号



www.apkbus.com