

高可用、弹性动态的 金融级移动架构在蚂蚁金服的演进之路

罗其平（重岳）

蚂蚁金服-技术专家



目录

- 支付宝在移动端的架构演进与思考
- 高可用、弹性动态的移动金融架构演进
- 快速构建、复制稳定、高可用移动金融应用



现在的支付宝

8.7亿

年活跃用户

5%₀₀

日闪退率

200+

服务功能

3+

人均使用



三个时代

独木舟



战列舰



航空母舰



三个时代

大杂烩

网络

存储

H5

支付

开发

昨天的代码好好的，今天不能跑了...

Android 又 65535 了!!!

改完了在编译，我先去喝杯咖啡~

代码合并好几天，测试包还没发布

启动太慢，用起来不流畅

测试



三个方向

○ 团队协作

- 架构分层
- 基础能力下沉
- 业务分治

○ 研发效率

- 接口与实现分离
- 二进制依赖
- 微应用模式

○ 性能与稳定

- 启动治理
- 基础性能
- 网络优化



战列舰时代

业务层

余额宝

账单

花呗

蚂蚁森林

消息中心

服务层

人脸识别

支付

应用中心

智能投放

登录

组件层

安全

网络

图片

H5容器

UI组件

存储

框架层

容器

微应用框架

服务框架

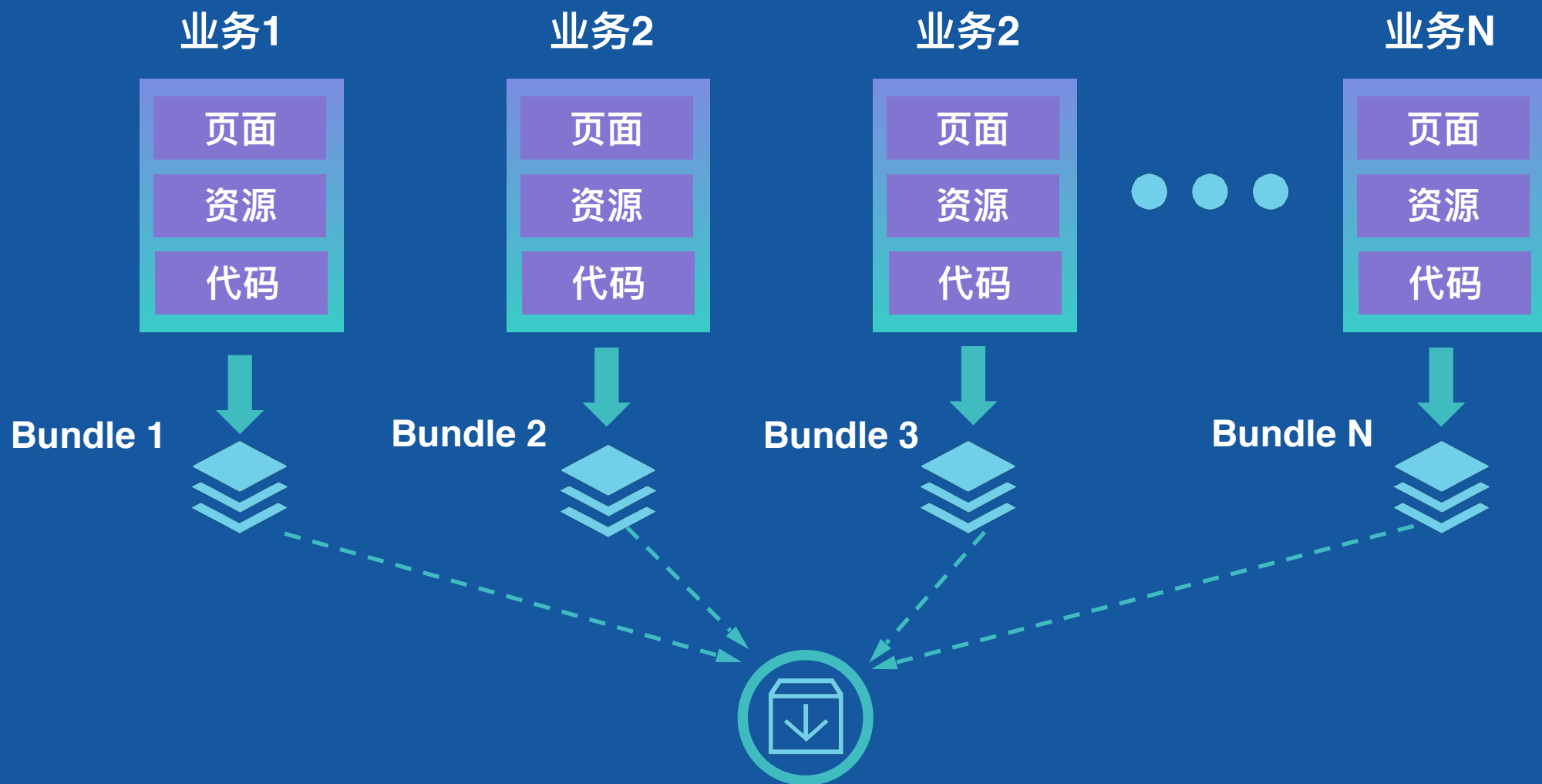
Pipeline

mPaaS

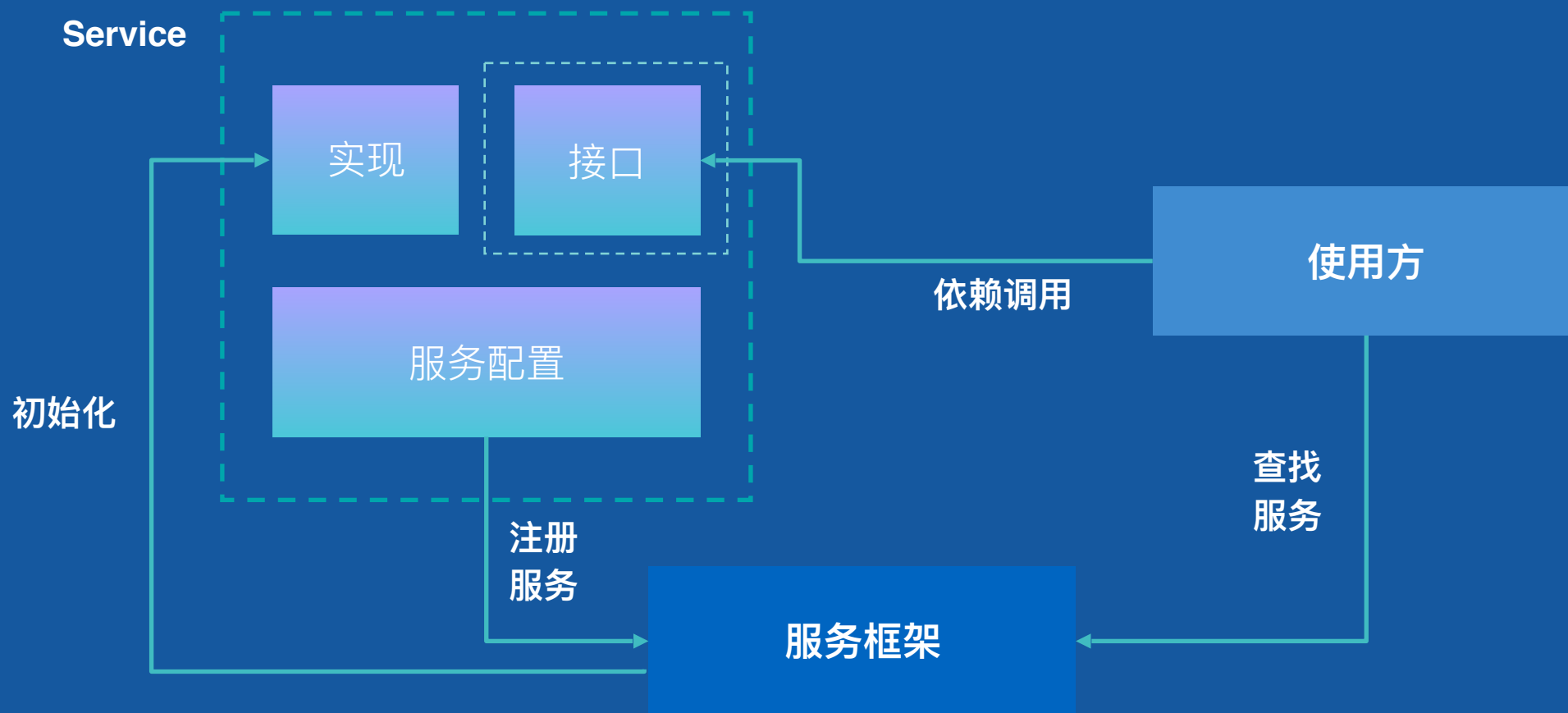
Mobile platform as a service



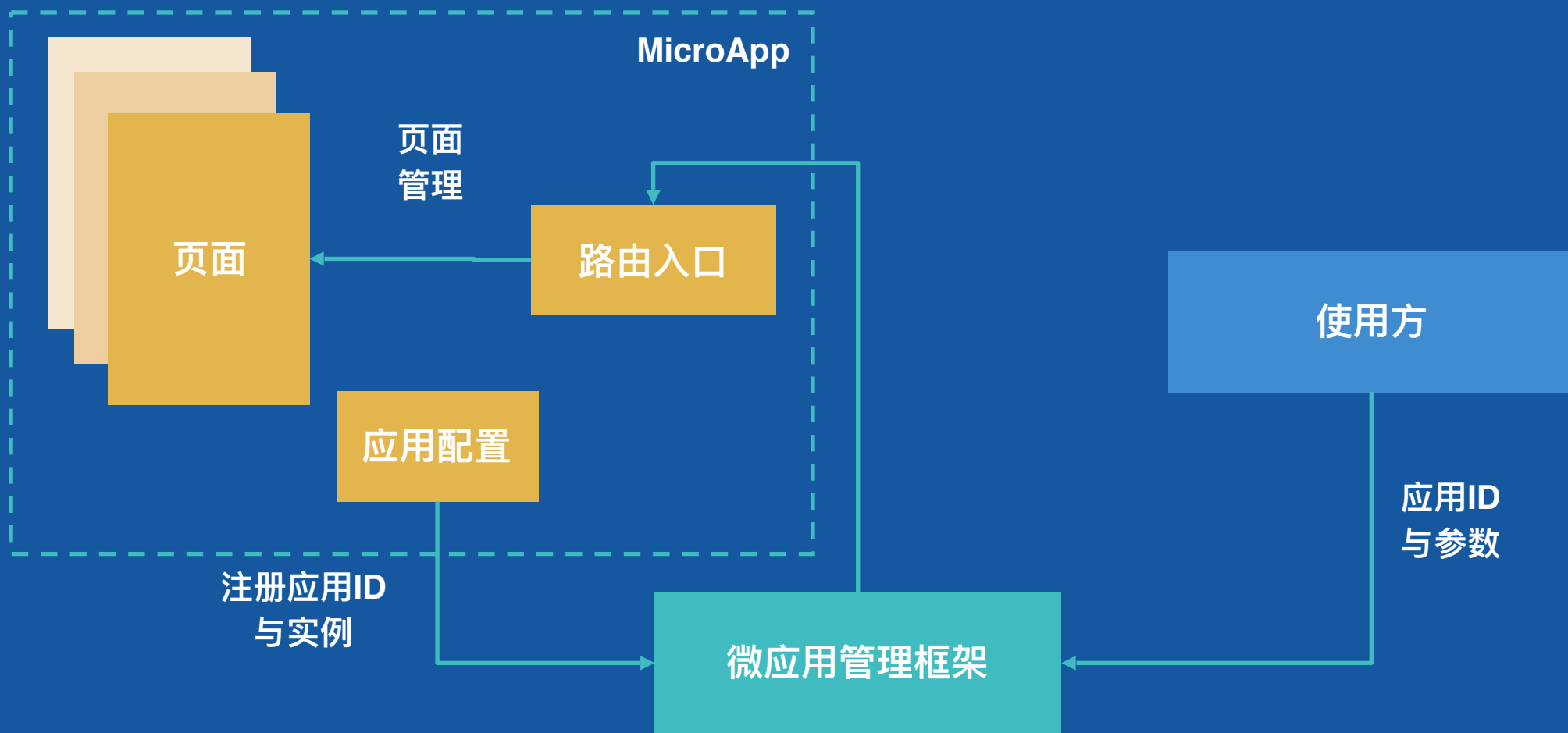
业务分治



接口与实现分离 - Service



微应用化



性能治理

统一开放规范，业务方使用统一线程池、存储工具，按需加载

引入 Pipeline 机制，根据业务优先级规定业务初始化时机

利用 AOP 切面对常用路径进行耗时统计，找到性能瓶颈

内存，存储，电量，流量，流畅度监控

● 框架层面

● 基础指标跟踪优化

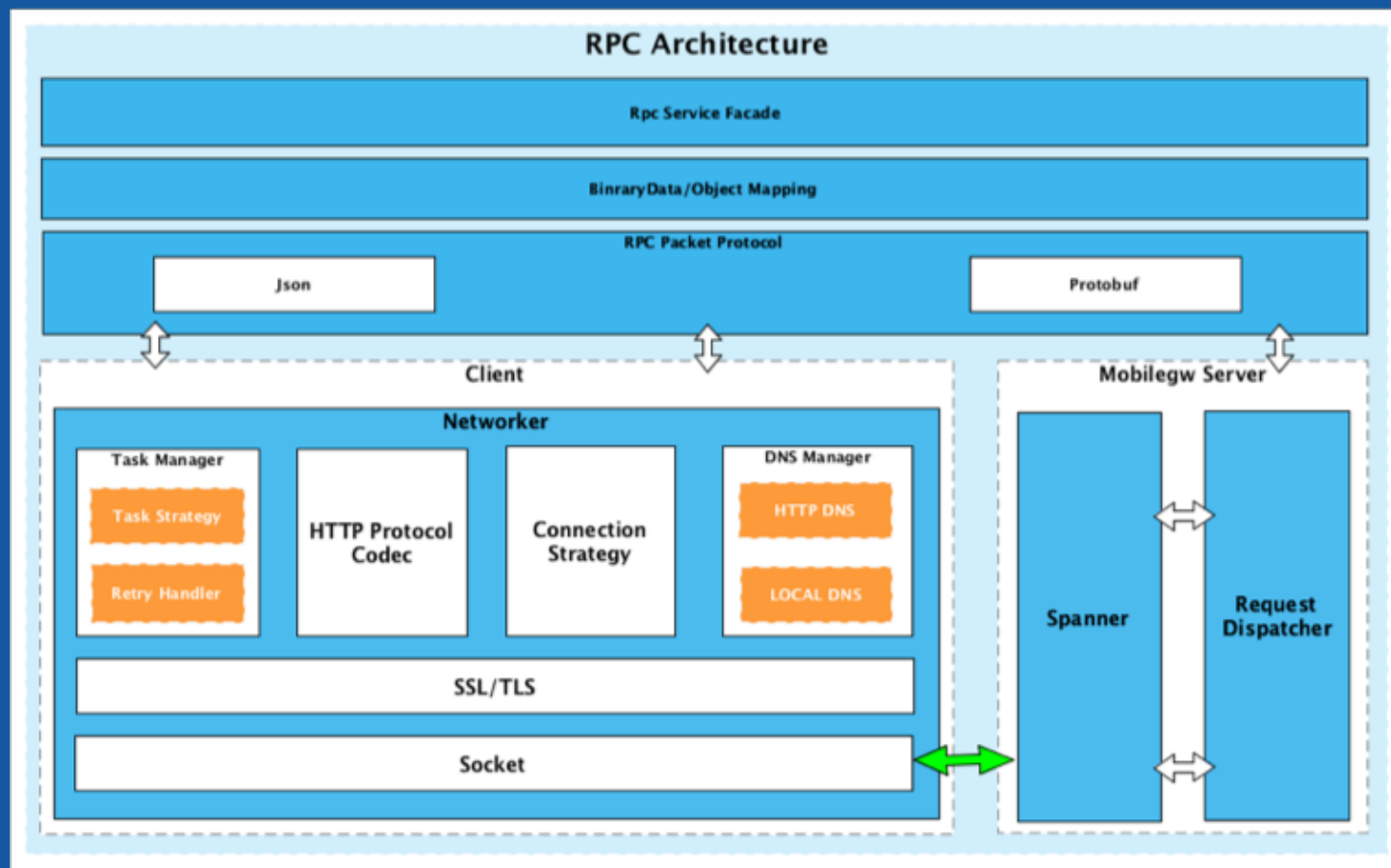
● 向下突破

虚拟机优化，如启动过程中禁用 GC

利用 Background Fetch 机制实现“秒”起



一体化移动网络服务



自研MMTP协议

- 0RTT 握手协议
- Protobuf 数据格式
- 链路复用
- Z-standard 压缩算法
- 自定义字典

HTTP-DNS

- 避免 local DNS 缺陷
- 防劫持
- 容灾调度

推拉结合

- 全量数据拉模式，增量数据推模式



更高的要求：打造航母

- 业务数量与类型急速膨胀，固定集中时间点发布效率太低
- 代码量膨胀，安装包过大

- 用户对体验的诉求形成舆论压力
- 用户投诉需要得到迅速处理
- 用户使用环境复杂度远高于内部测试环境

业务
井喷

开放
平台

- 打造生态，引入三方服务
- 第三方服务质量影响支付宝整体体验

用户
反馈

监管
要求

- 监管要求时刻在变，要积极拥抱监管

高可用
弹性动态



动态研发模式

- 代码量膨胀，安装包太大，iOS代码段大小超限不能过审
- 集中点发布，迭代慢
- 用户使用的服务数量有限



- 一套代码多端运行
- 按需加载，节省流量存储，安装包大小可控
- 统一开放标准
- 快速发布，各发各的，不依赖客户端发版



打磨Web体验

差量更新

计算资源差量
节省带宽流量

容错补偿

离线资源无法访问
自动 fallback 到线上

深度定制组件

自定义扩展标签
嵌入原生组件
体验更突出

前后端分离

页面资源离线化
网络请求走 Native 通道，
解决白屏、打开速度慢等问题

推拉结合

RPC+SYNC
保证版本更新率

Android 独立浏览器内核

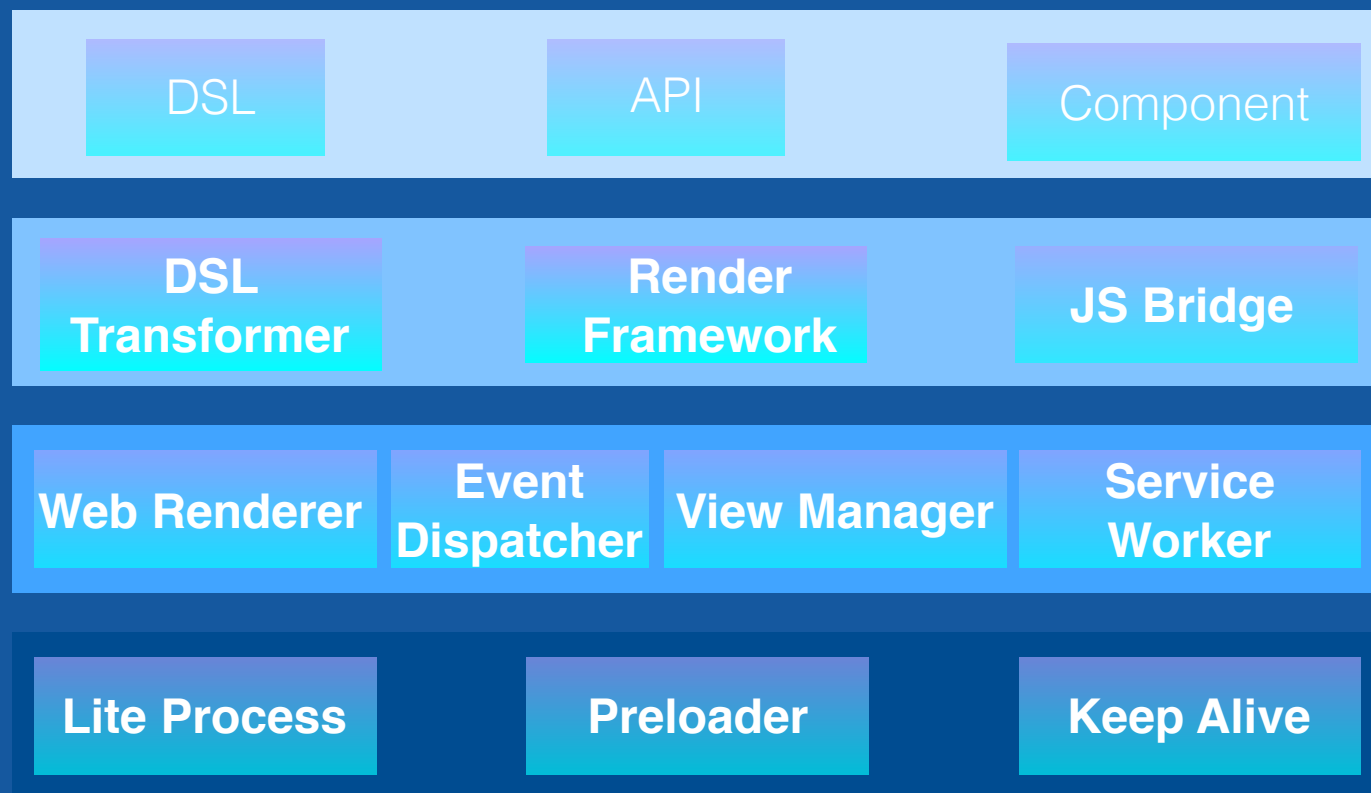
解决 Android 机型碎片化
带来的兼容性问题

全面监控

资源加载异常
JS 执行异常
页面耗时



小程序



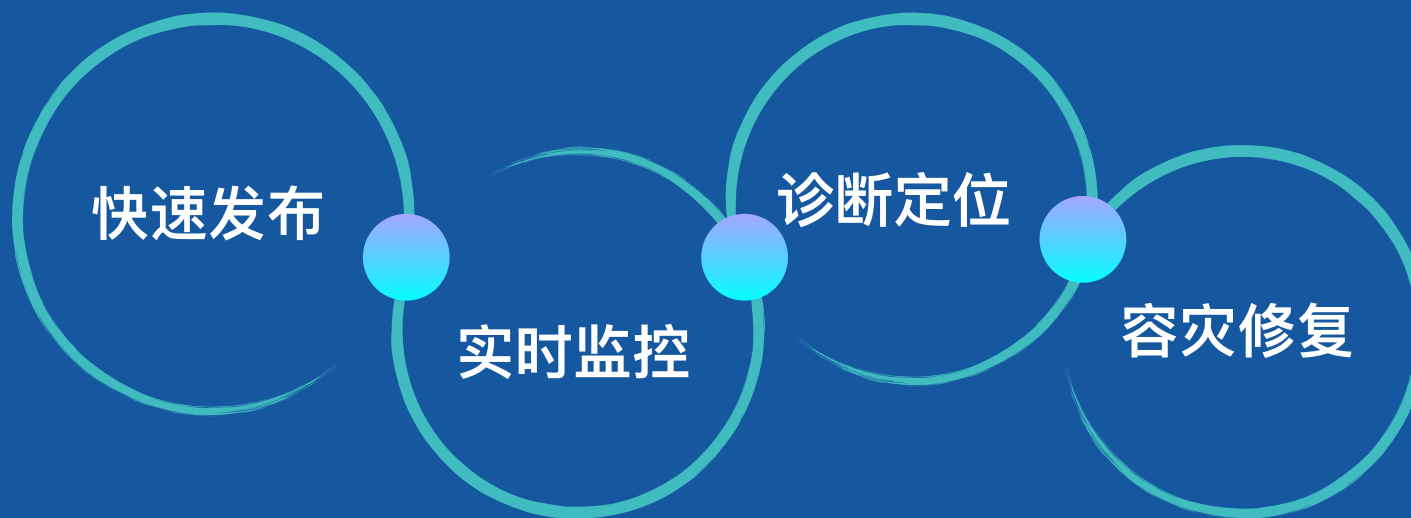
支付宝能力全面开放

简单易用，IDE 支持

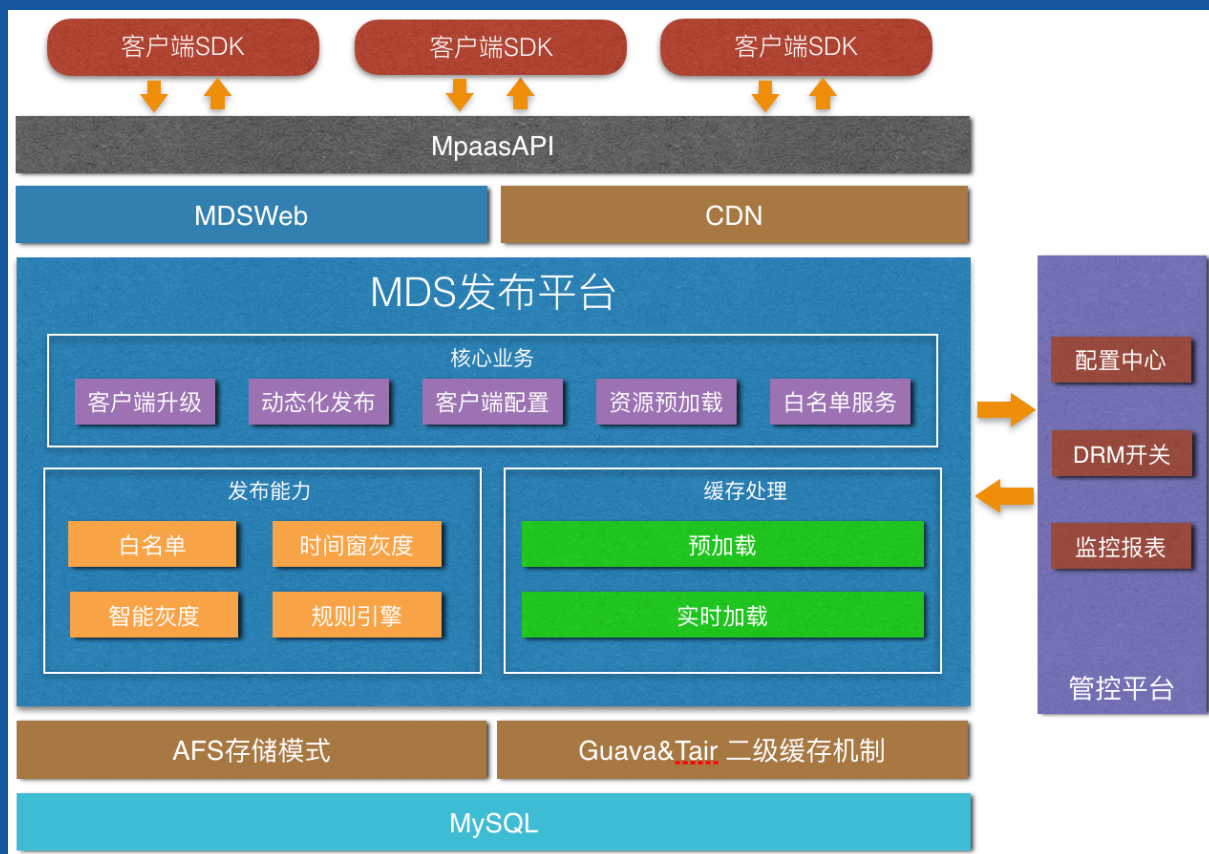
未来可平滑切换原生渲染



线上高可用保障体系



快速发布



智能灰度能力，多种升级策略

内部灰度，外部灰度，人群地域、机型网络等多种规则可供选择

增量差分离线包更新能力

减少数据冗余及设备带宽，在移动端网络条件不稳定场景下体现优势

系统高性能保障

RT < 3ms , QPS 5w/s, 触达率 99.99%



监控、诊断

监控指标

- 闪退
- 流畅度
- 电量
- 流量
- 不可用埋点

上报策略

- 区分流畅度，高优先级实时上报
- 独立轻量进程
- 定制化开关及多种采样上报策略

上报方式

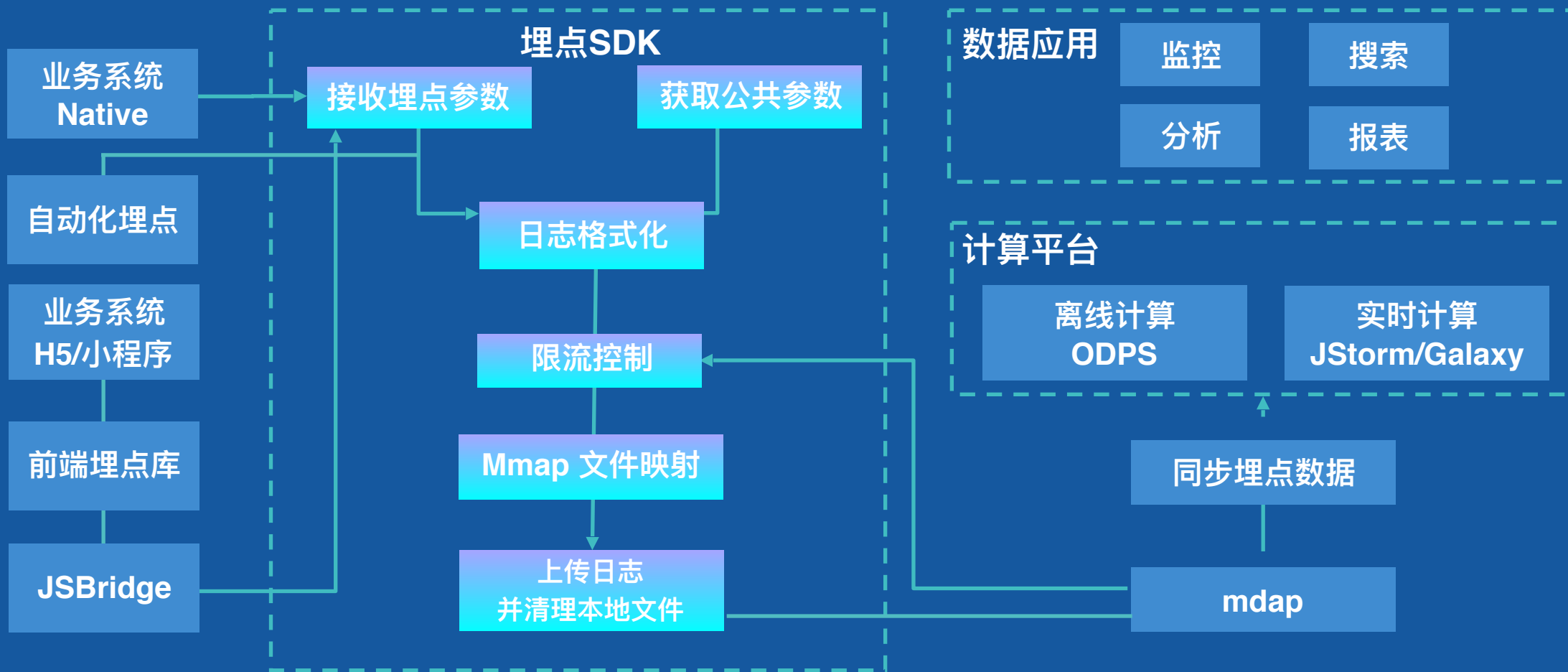
- 自动上传
- 周期性检查上传
- 诊断指令驱动上传

诊断分析

- 用户行为分析
- APP日志分析



实时监控



容灾处理

故障隔离

预设开关，业务出现故障时立即
推送配置，屏蔽止血

自动恢复

客户端启动阶段监听死锁、闪退与首屏加载异常，
连续启动失败时重置清空应用数据。

流量熔断

Hook网络调用，统计请求次数和流量消耗。
在不同策略下达到阈值时，触发异常上报或熔断

动态化能力修复

Hotpatch 修复原生代码，可监控，可回滚



传统金融机构 App 存在的问题



启动时间长，经常崩溃



研发效率低下
团队协作程度不高



监控处于空白
客户端发生问题无法定位和解决



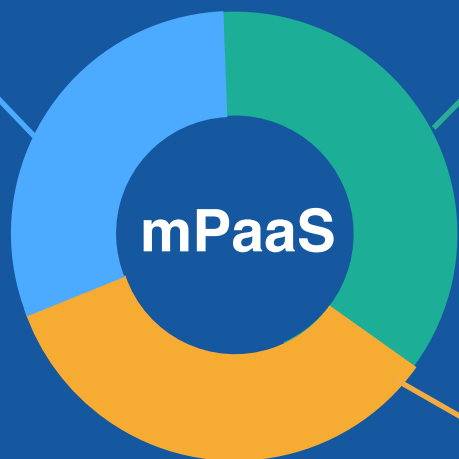
发布灰度缺失
无法小范围验证产品能力



助力金融APP

研发

客户端框架
移动网关
数据同步
消息推送
H5容器
小程序框架
日志埋点
热修复



运维

用户行为分析
闪退分析
性能分析
异常告警
自定义分析

发布

智能灰度
离线包

- 快速复制
- 高可用，高稳定
- 高度工程化，大团队协作开发快



经典案例：广发银行



5%₀₀

日闪退率，
以前统计基本处于空白

1.5s

启动时间，
以前 30s 或更长

3个月

手机银行在发现精彩后
启动改版，复用大量发现精彩组
件，开发时间大幅减少



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。



谢谢

THANK YOU

