

# 混合应用的工程化实践

美团点评 平台及酒旅事业群 李罡





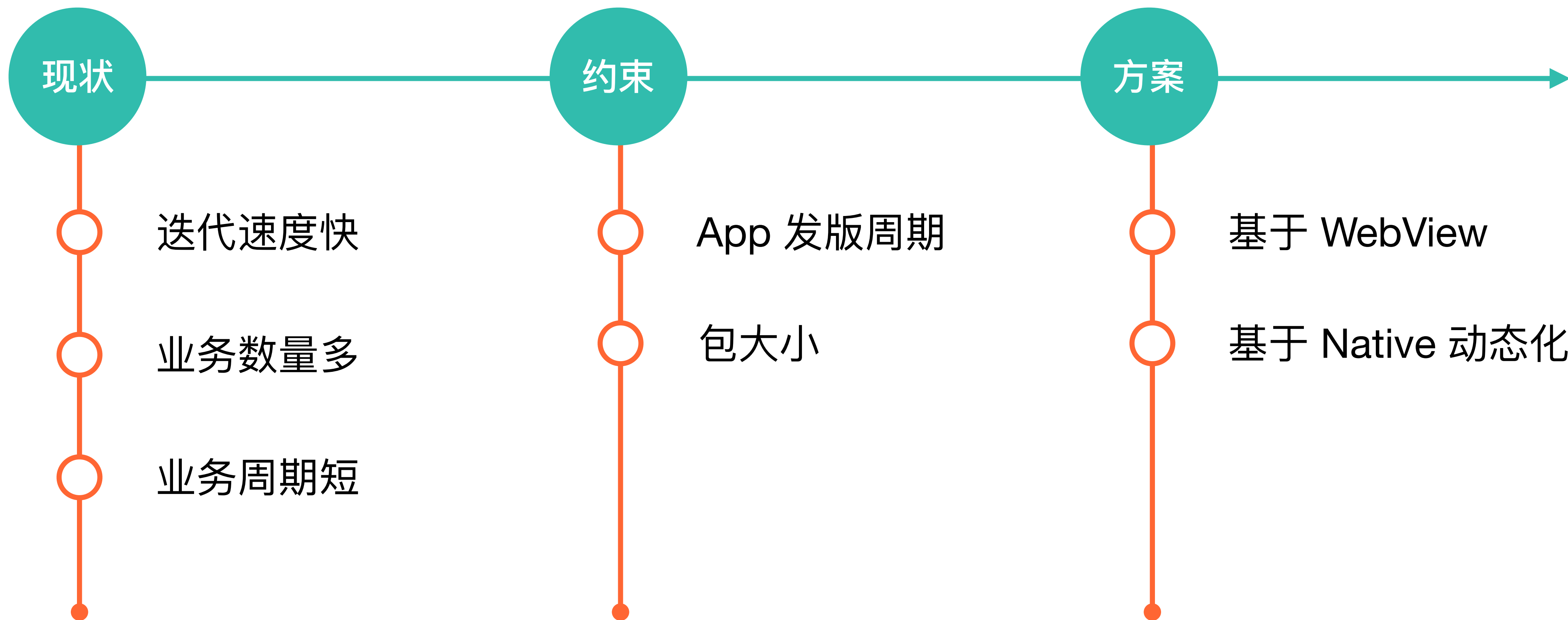
李罡

前端工程师，平台前端基础架构

曾经从事桌面软件开发

目前负责前端发布系统、混合应用框架产品

[github.com/sartrey](https://github.com/sartrey)



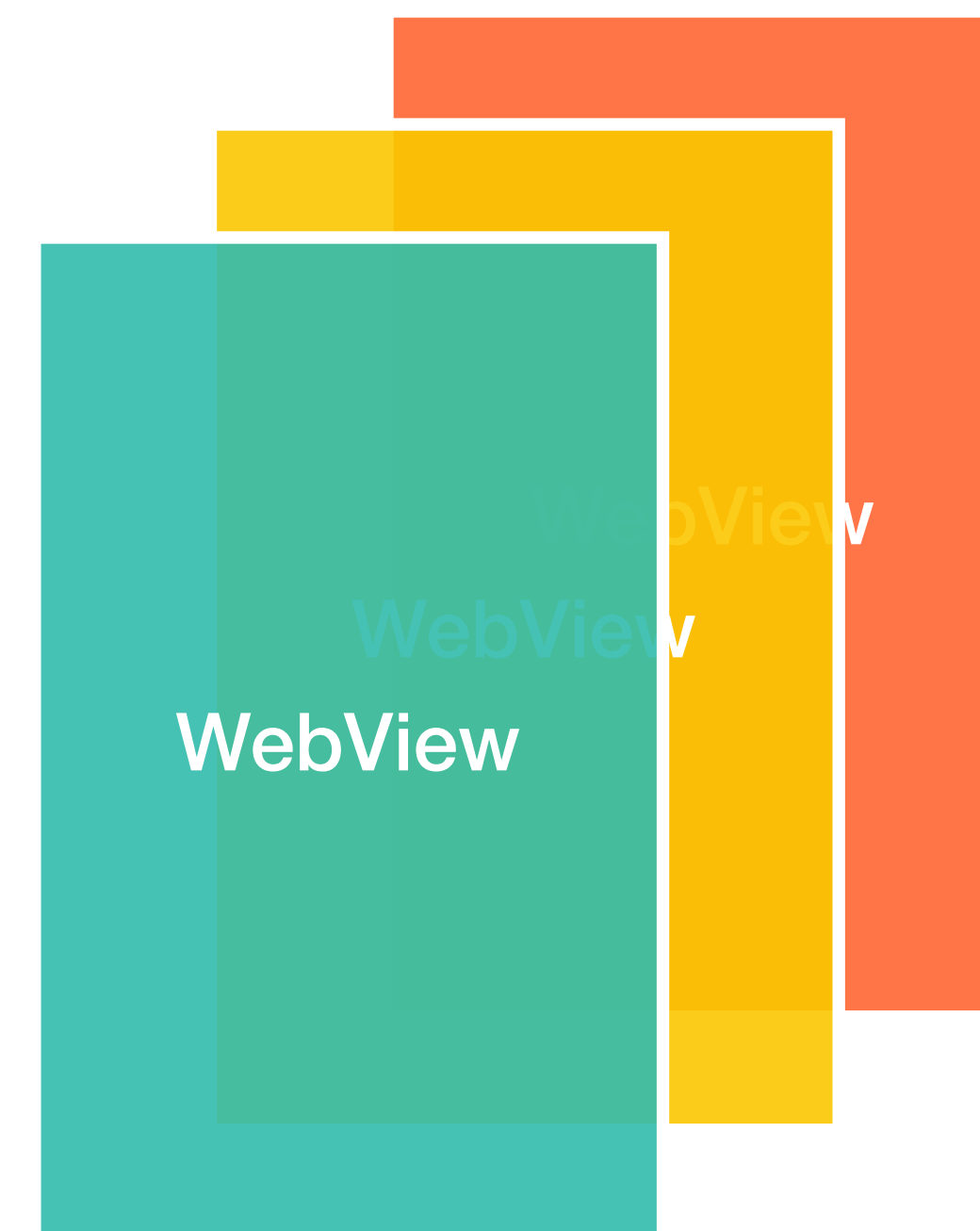
# Hybrid 应用



**设备**  
提供底层支持



**桥**  
连接设备与容器



**容器**  
承载页面

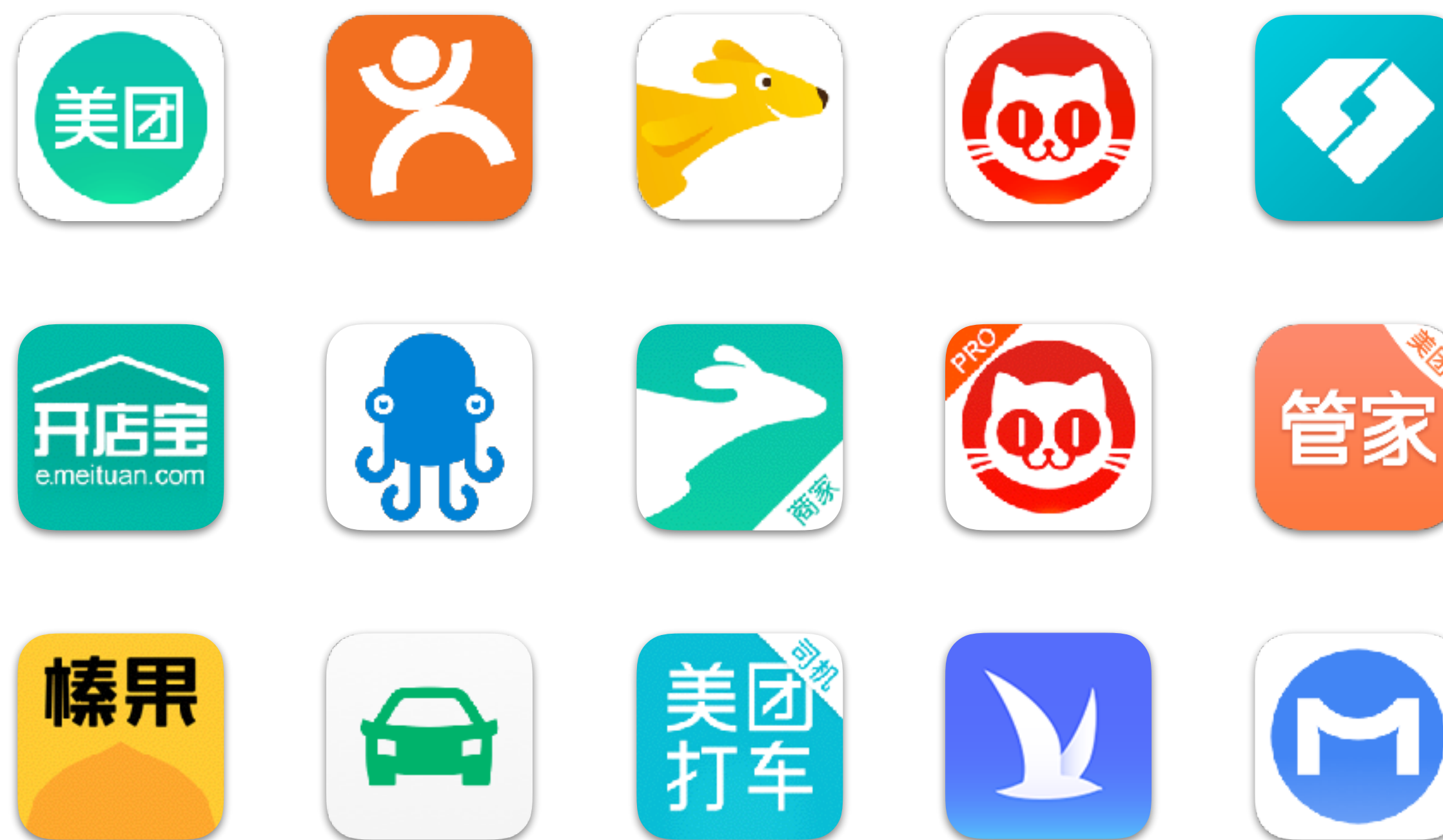
# 我们的产品与服务

## 容器

- 接入多个 App, Crash 率极低
- 支持离线化、长连接通道

## 桥

- 提供基础 API, 支持扩展业务 API
- 推进多个历史框架收敛下线
- 支持微信、支付宝等第三方 App



100 个业务桥需要 100 次发版吗?  
如何屏蔽复杂网络环境的体验差异?  
怎样收集数据更有利于发掘数据内涵?

## 桥鉴权

简化改善效率

## 网络优化

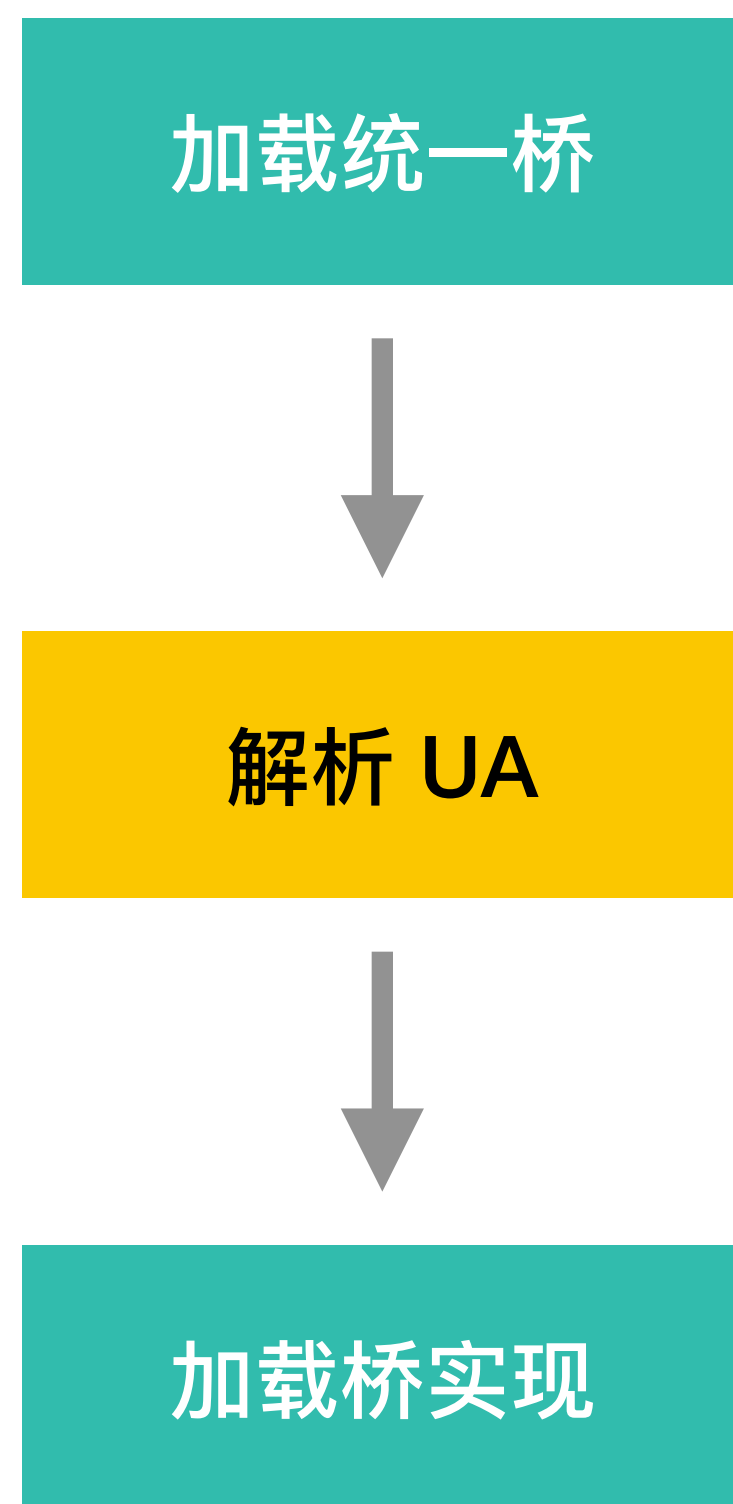
分段各个击破

## 数据收集

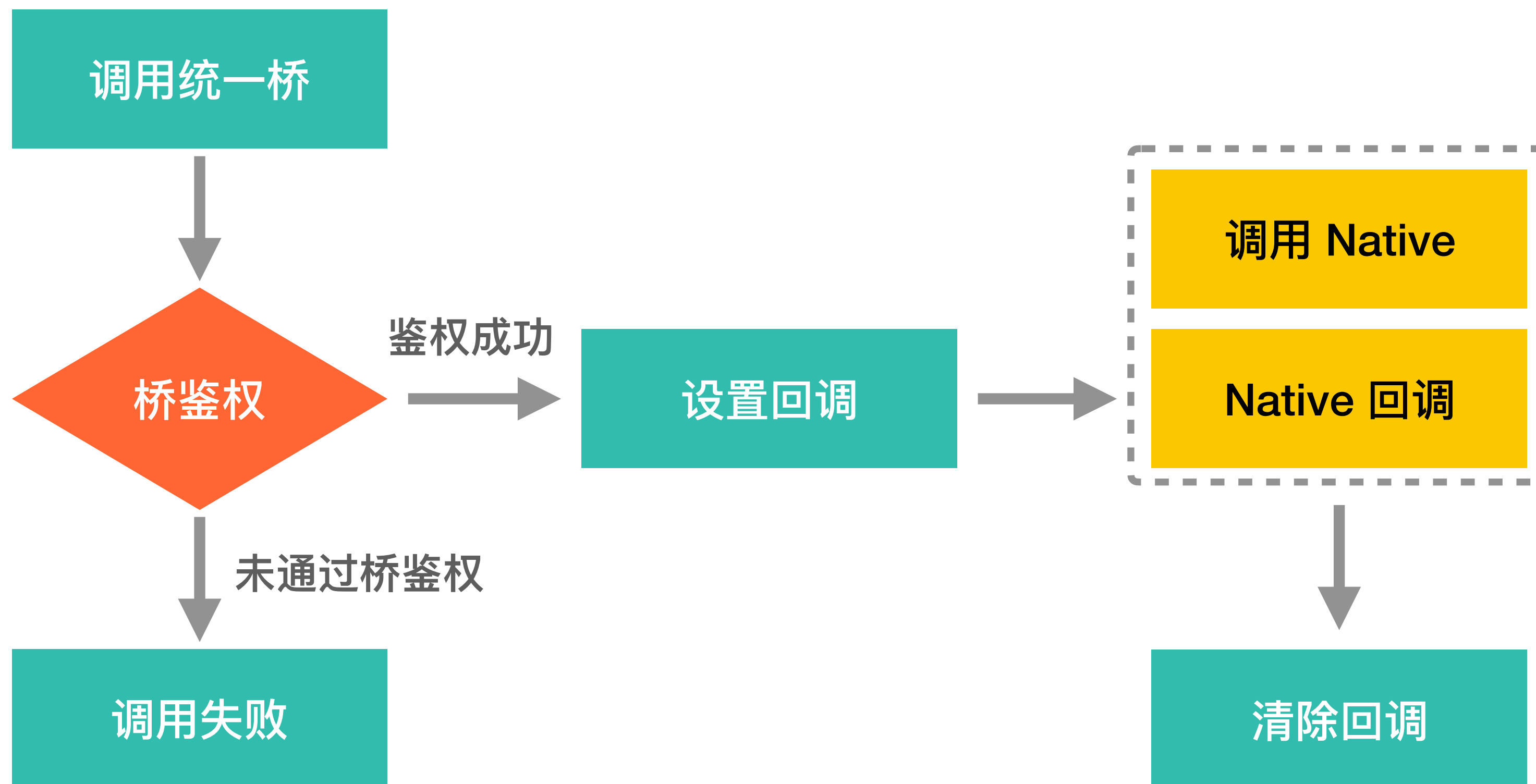
克制避免风险

## 案例一 · 桥鉴权

# 桥的加载与调用



加载



调用



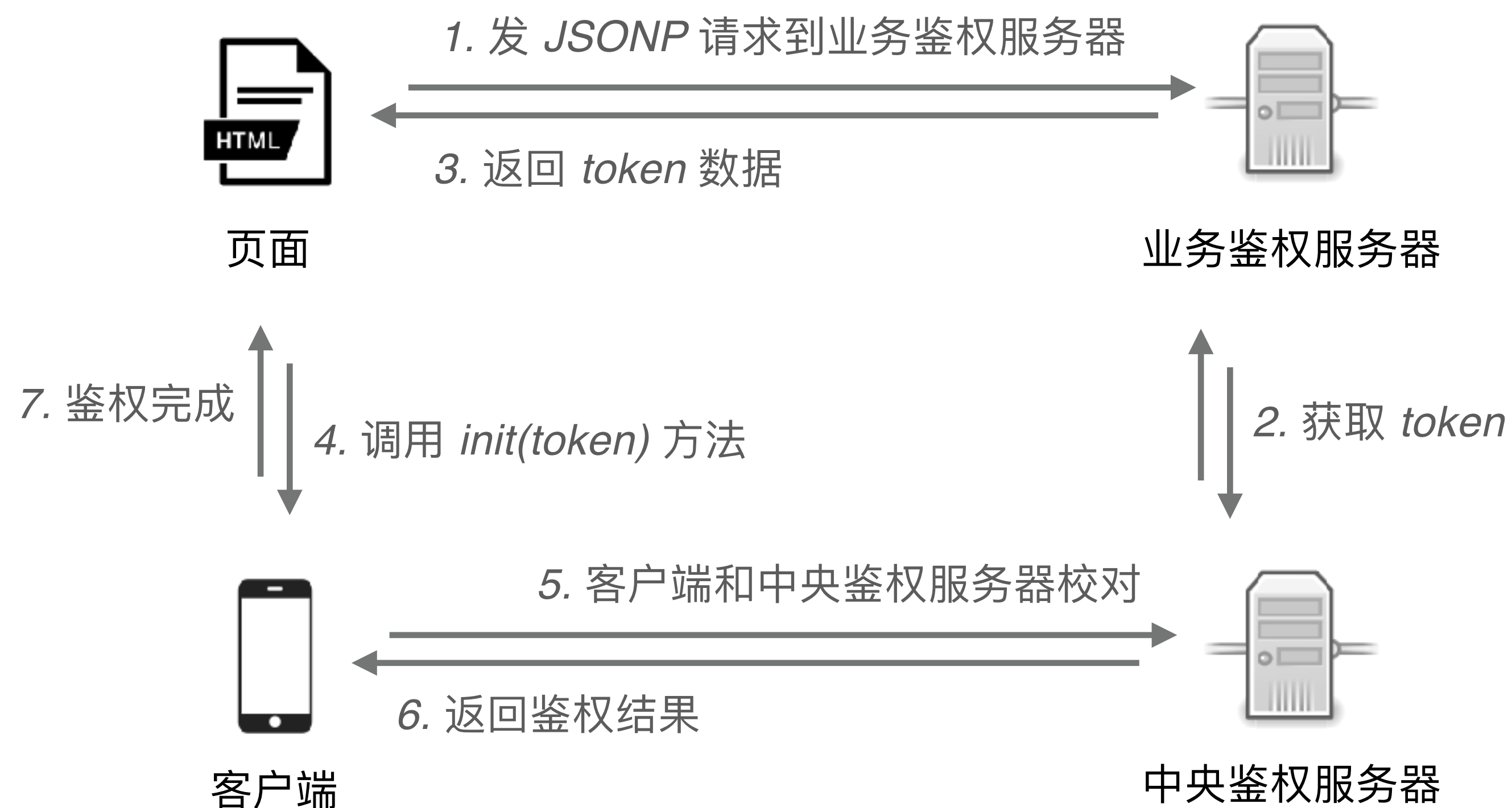
# 业界方案 OAuth2

## 优点与缺点

- 完整、安全、标准
- 复杂、缓慢、风险传递

## 典型应用

- 微信第三方应用登录
- 美团桥鉴权 (旧)



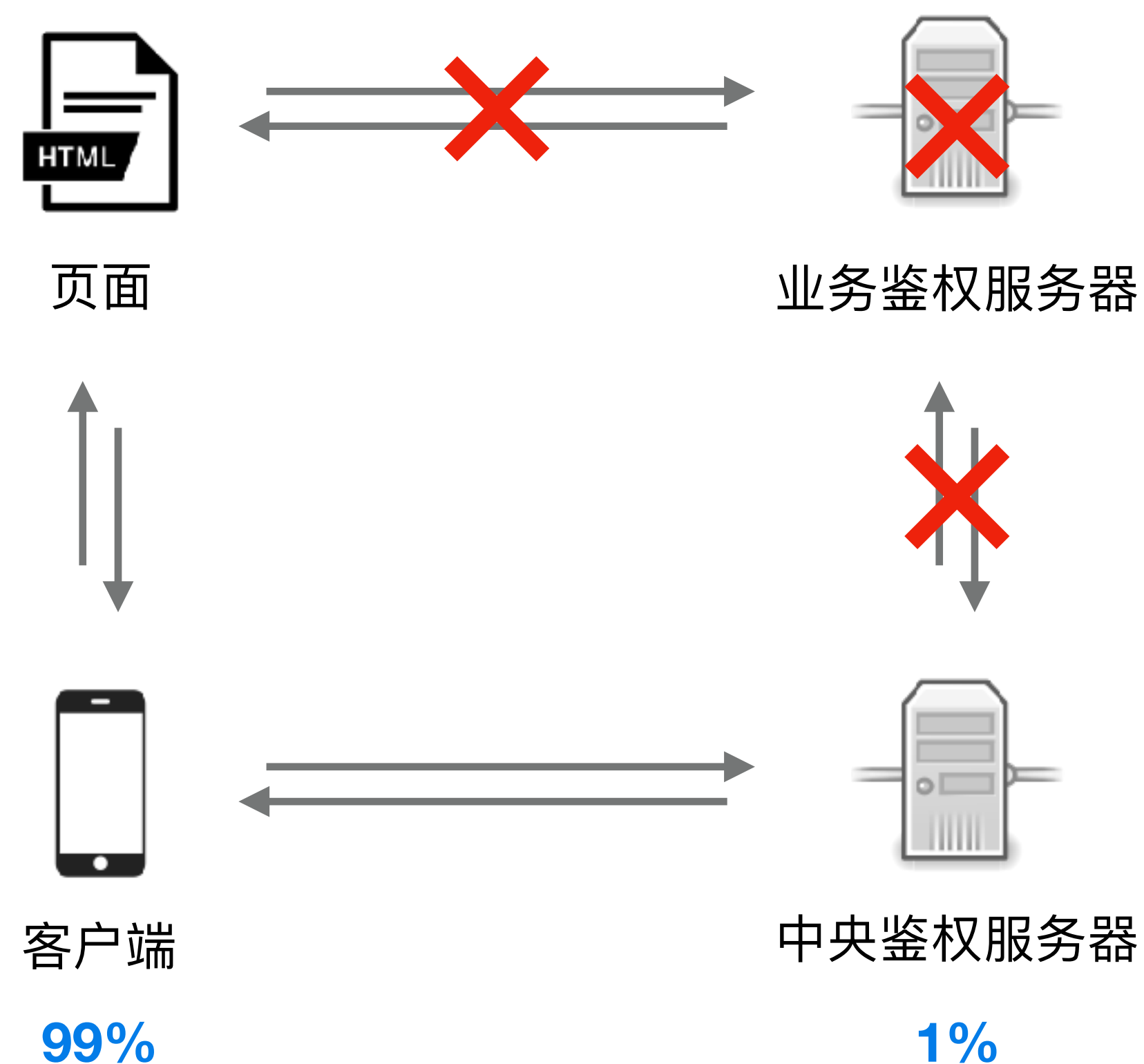
# 用户侧鉴权

## 业务实际情况

- 页面偶尔出现功能失效
  - 弱网、鉴权服务宕机
- **99%** 的页面是公司，**1%** 的页面是第三方

## 改进方案

- 本地验证内部业务
- 网络验证第三方业务
- 验证只做白名单匹配



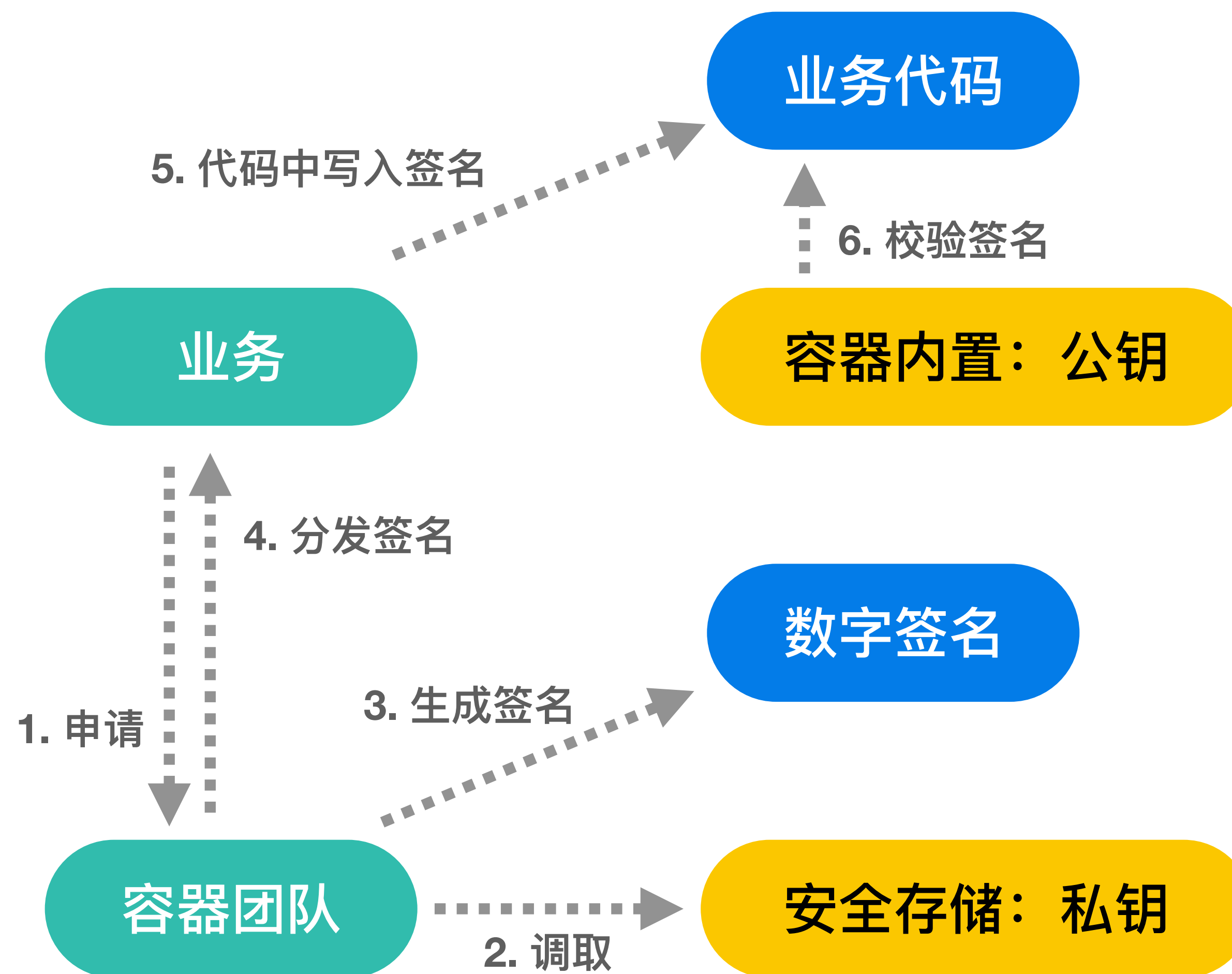
# 开发者鉴权

## 业务实际情况

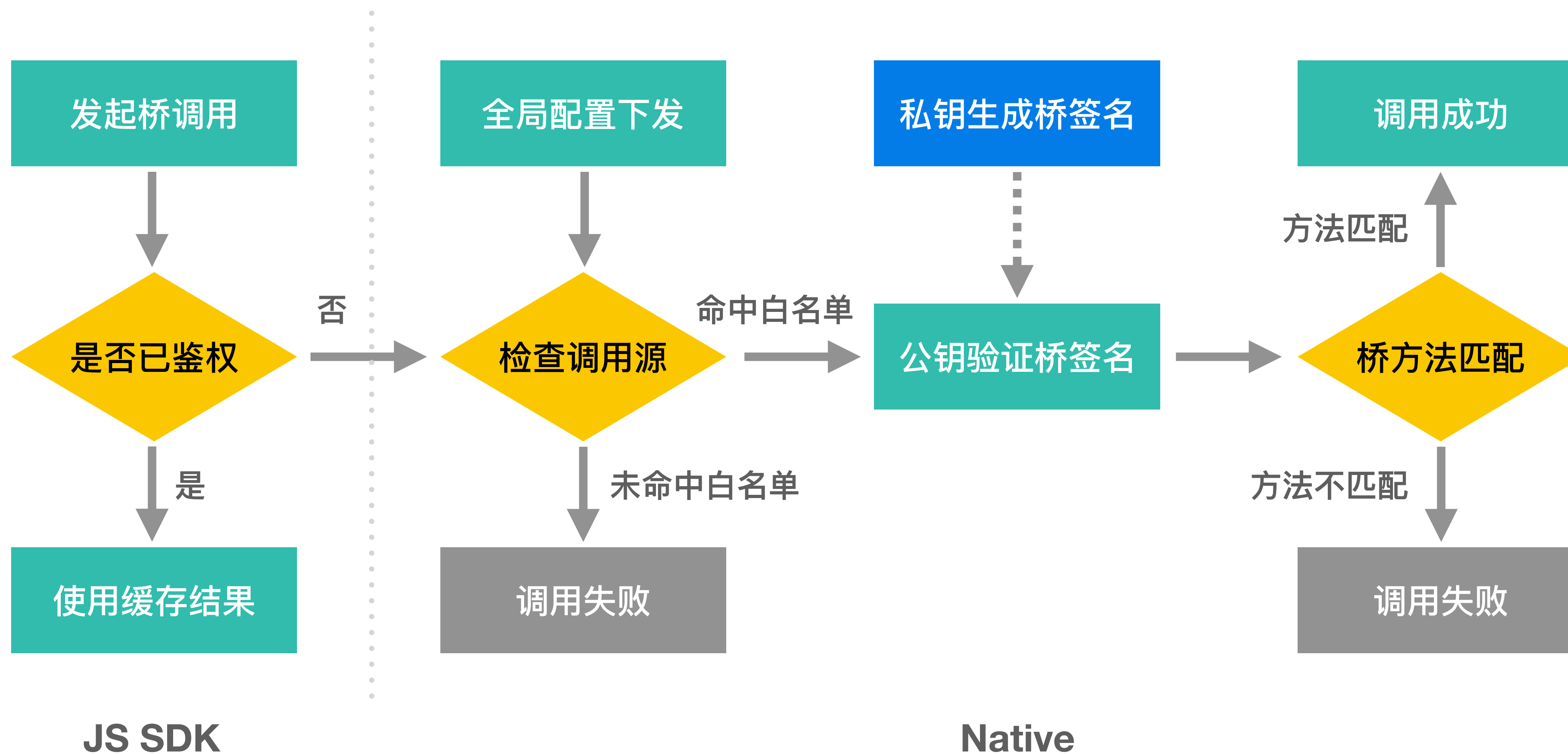
- 申请业务桥需要等待容器组件发版
  - 发版节奏不一致、升级引入新风险
- **平均每周都有业务桥申请**

## 改进方案

- 使用数字签名验证替代白名单
- 授权方掌握私钥提供签名
- 容器携带公钥和验证逻辑



# 简化桥鉴权

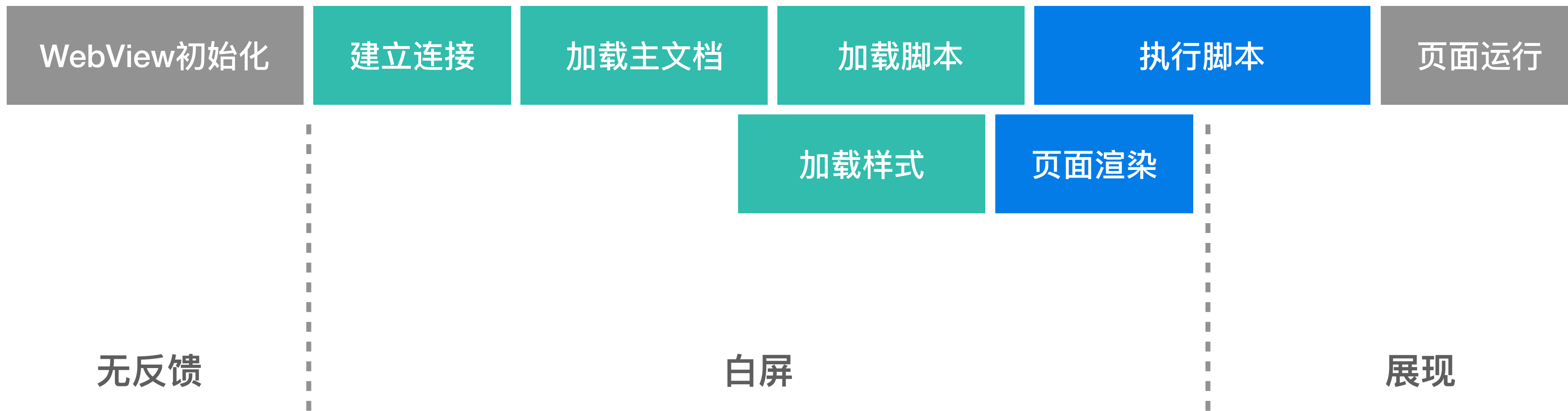


## 案例二 · 网络优化

# 页面的加载渲染

移动端页面首屏时间消耗 = 资源加载 + 首屏渲染 + 脚本执行

- \* 受限于设备性能，页面渲染和脚本执行时间需要重视
- \* 参考 <https://tech.meituan.com/WebViewPerf.html>



# 渐进式网络优化



# 离线化 / 所谓 PWA

## PWA

- 渐进式 Web 应用
  - Service Workers、App Manifest、Push
- 承载业务无关逻辑

## LocalStorage 缓存

- 定制 Loader 强耦合发版流程
- 重度侵扰业务开发、容量和能力受限
- 兼容性良好

## Service Workers

- 本质是网络代理 + 独立 JS 进程
- Android 5.0+, 支持占比约 **84%**
- iOS 11.3 已支持

## 统一容器离线化

- 类似 Service Workers
- 基于请求拦截实现
- iOS 部分支持 / WKWebView 暂不支持



# 离线化 / 不是银弹

不要迷信离线化

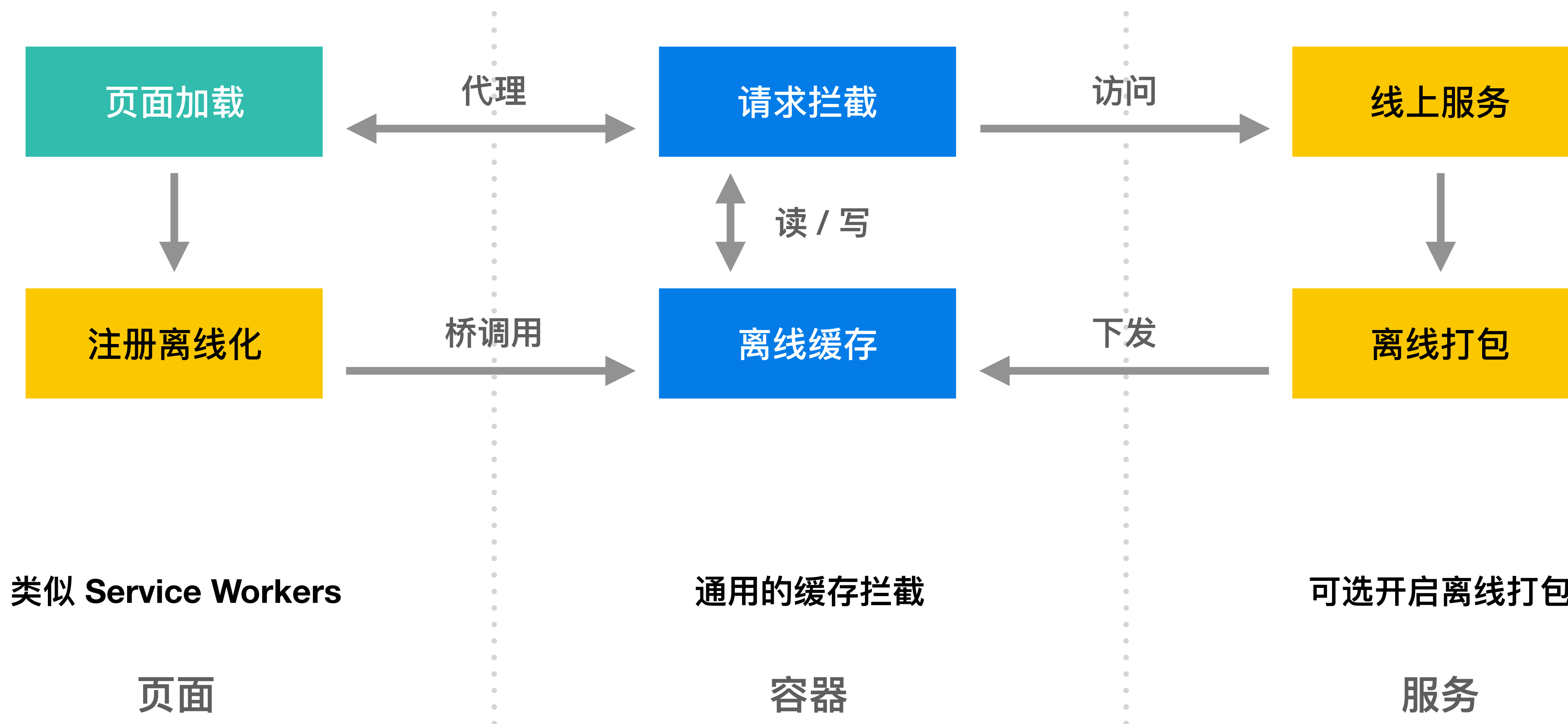
- 我们不可能比浏览器内核更懂缓存
  - 较 HTTP 缓存的网络收益甚微
- 弱网成功率提升、首次性能提升

主文档配合

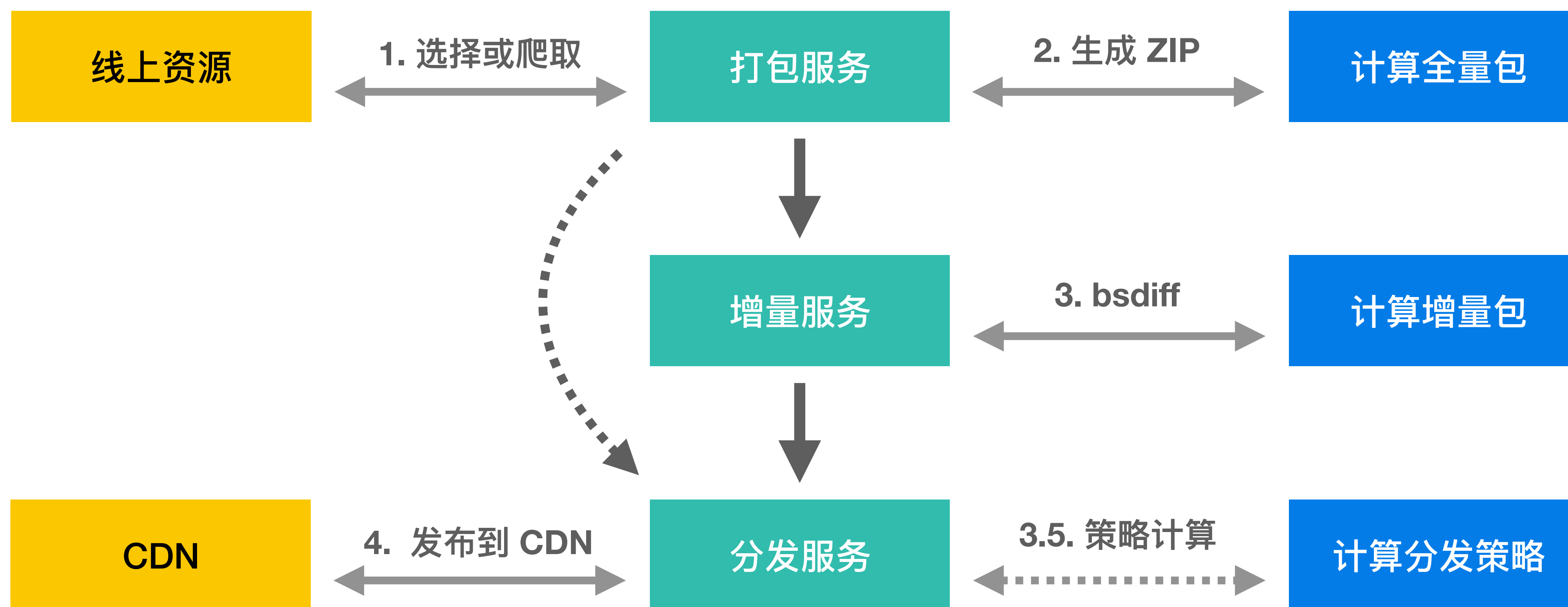
- 避免完全 SSR
  - AppShell + 客户端渲染
- 动态生成主文档何苦离线化



# 离线化 / 主逻辑



# 离线化 / 离线包预下发



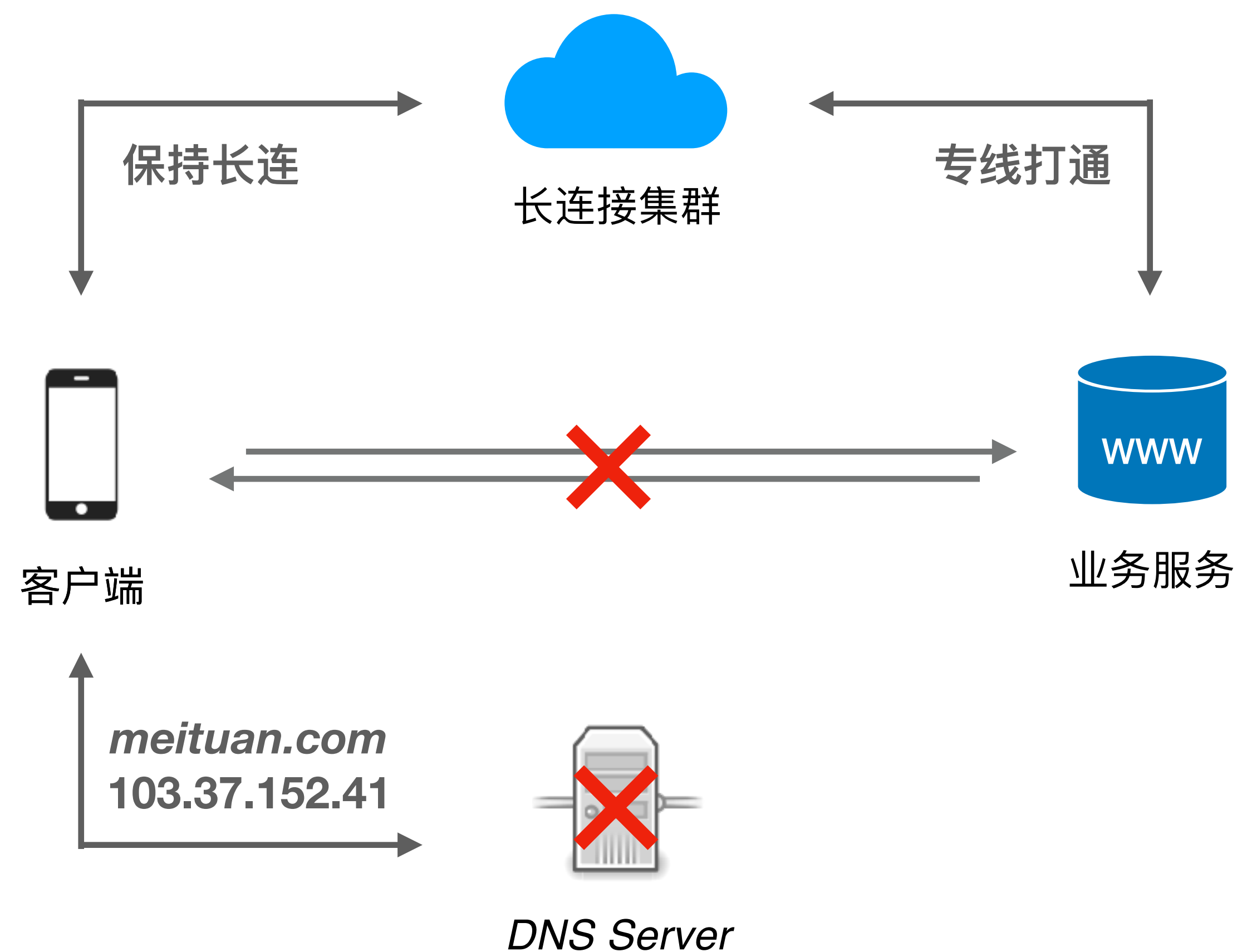
# 链路优化 / 长连接

## 基本假设

- 非 HTTP 请求不会被劫持
- 传输数据短小高频

## 业务相性

- AJAX 请求最适合
- 主文档需要权衡
- 静态资源最不适合



## 链路优化 / 容器对接长连接的踩坑过程

第一坑 / 桥性能有瓶颈



第二坑 / 拦截处理 POST



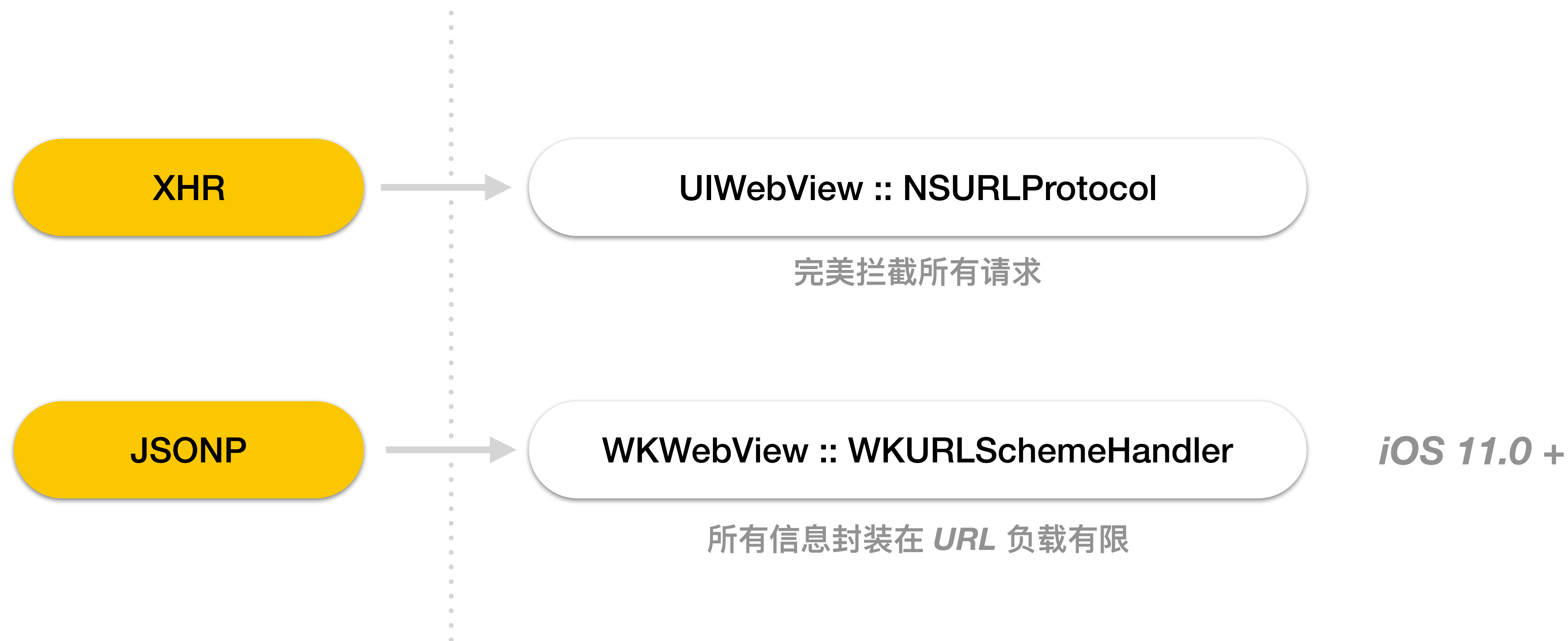
第三坑 / POST 跨域协商

- **evaluateJavascript 固有消耗**
  - 执行速度正相关响应体积
- **用拦截器替代桥**
  - 桥性能瓶颈无从优化

- **shouldInterceptRequest 缺陷**
  - 安卓不支持拦截 POST
- **用自定义 header 模拟 body**
  - 页面请求添加 header 承载 body
  - 拦截后 header 换成 POST body

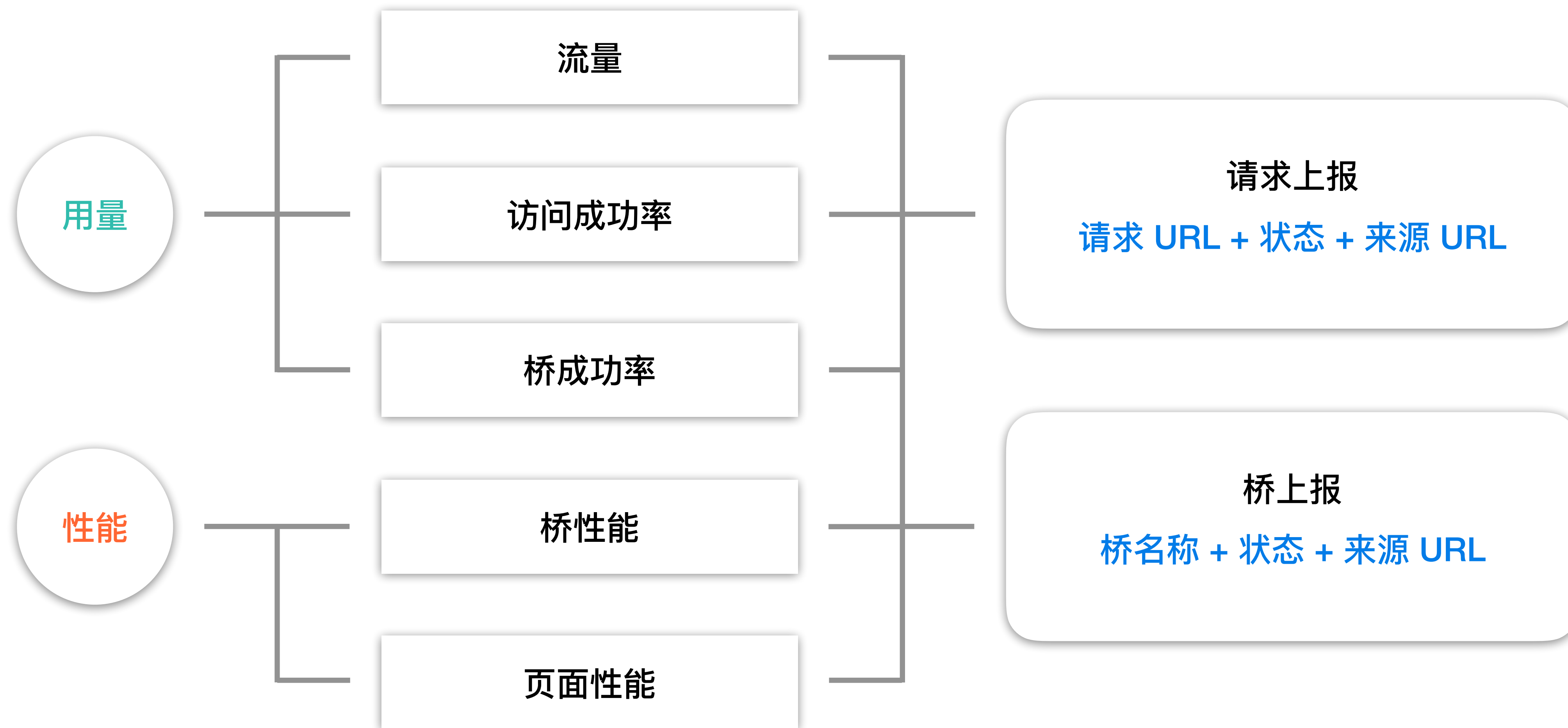
- **自动发起 OPTIONS**
  - 浏览器自动进行跨域协商
  - 自定义 header 参与协商内容
- **篡改 OPTIONS 协商内容**
  - 移除自定义 header
  - 避免服务端感知到这一层

## 链路优化 / WKWebView 对接长连接



## 案例三 · 数据收集

# 容器与桥的关键指标

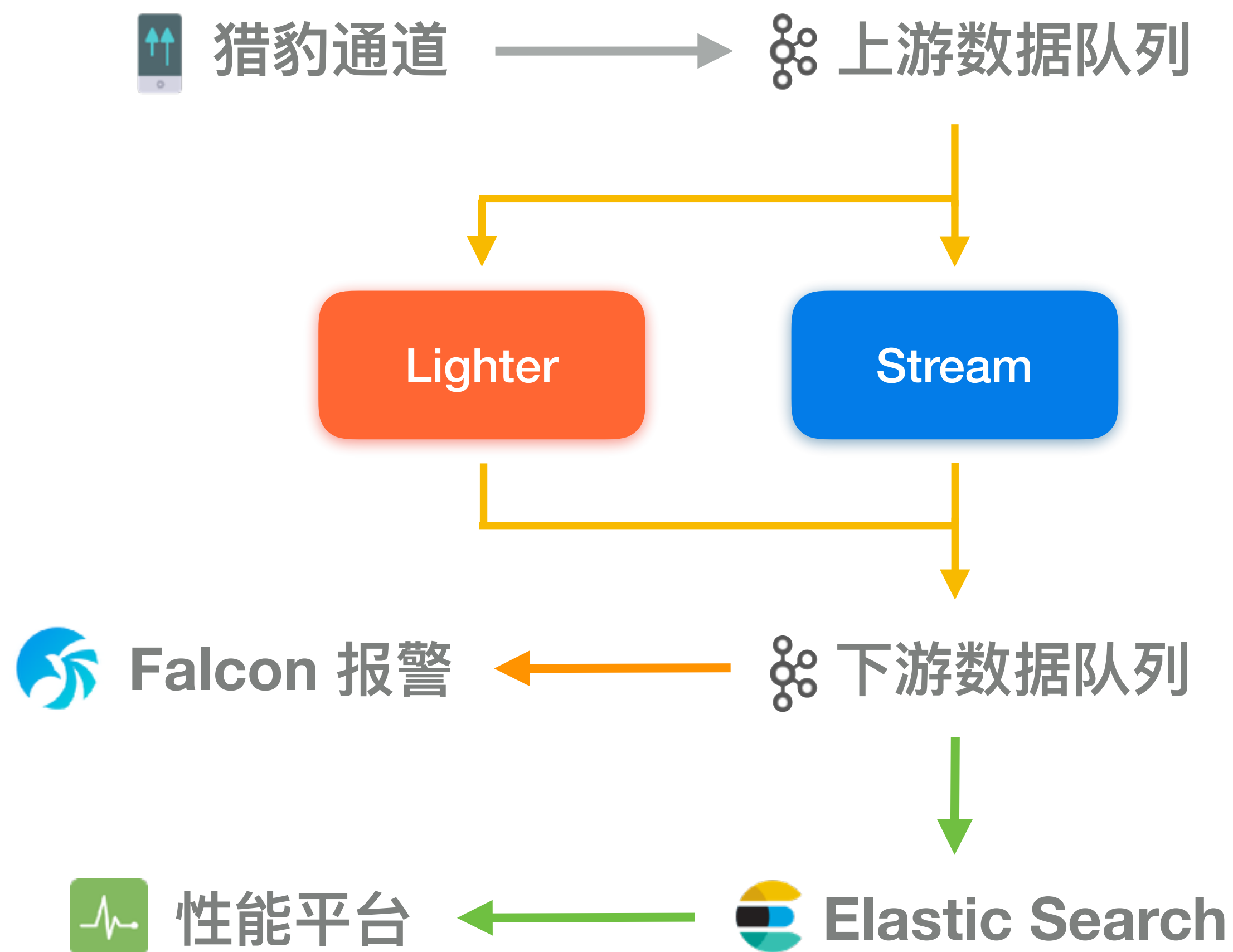




# 计算设施

## 算力成本

- 维度展开的乘法规模
  - 采样丢弃 + 减少维度
- 重 CPU 计算任务
  - 调度算法



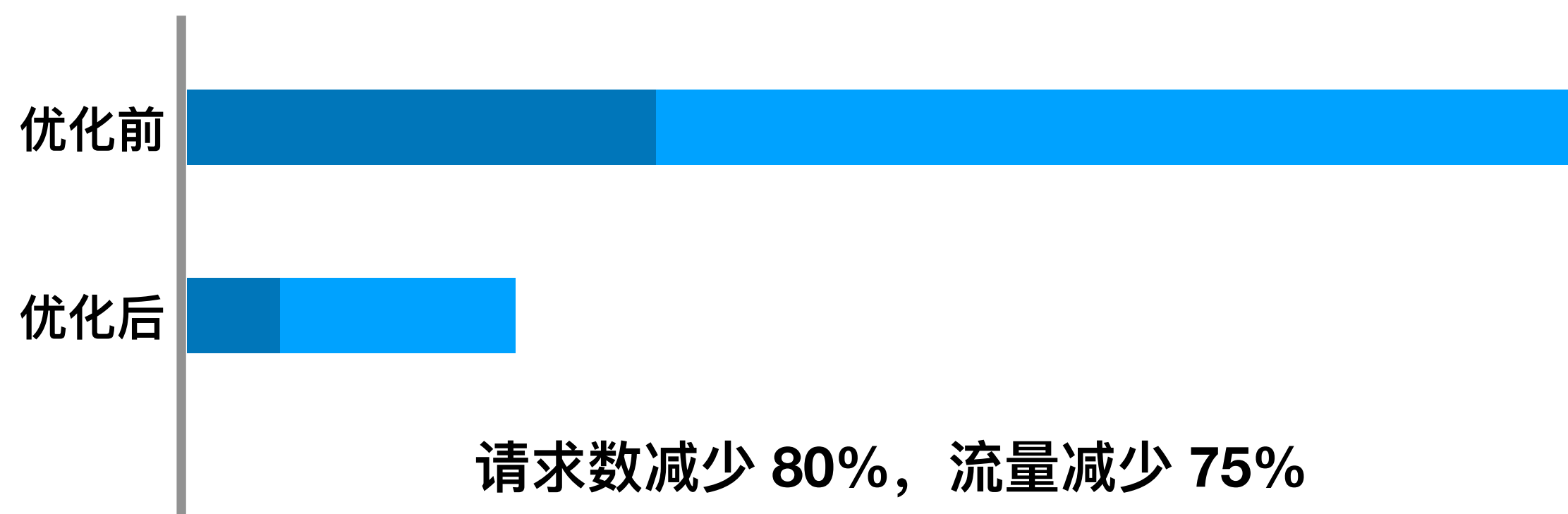
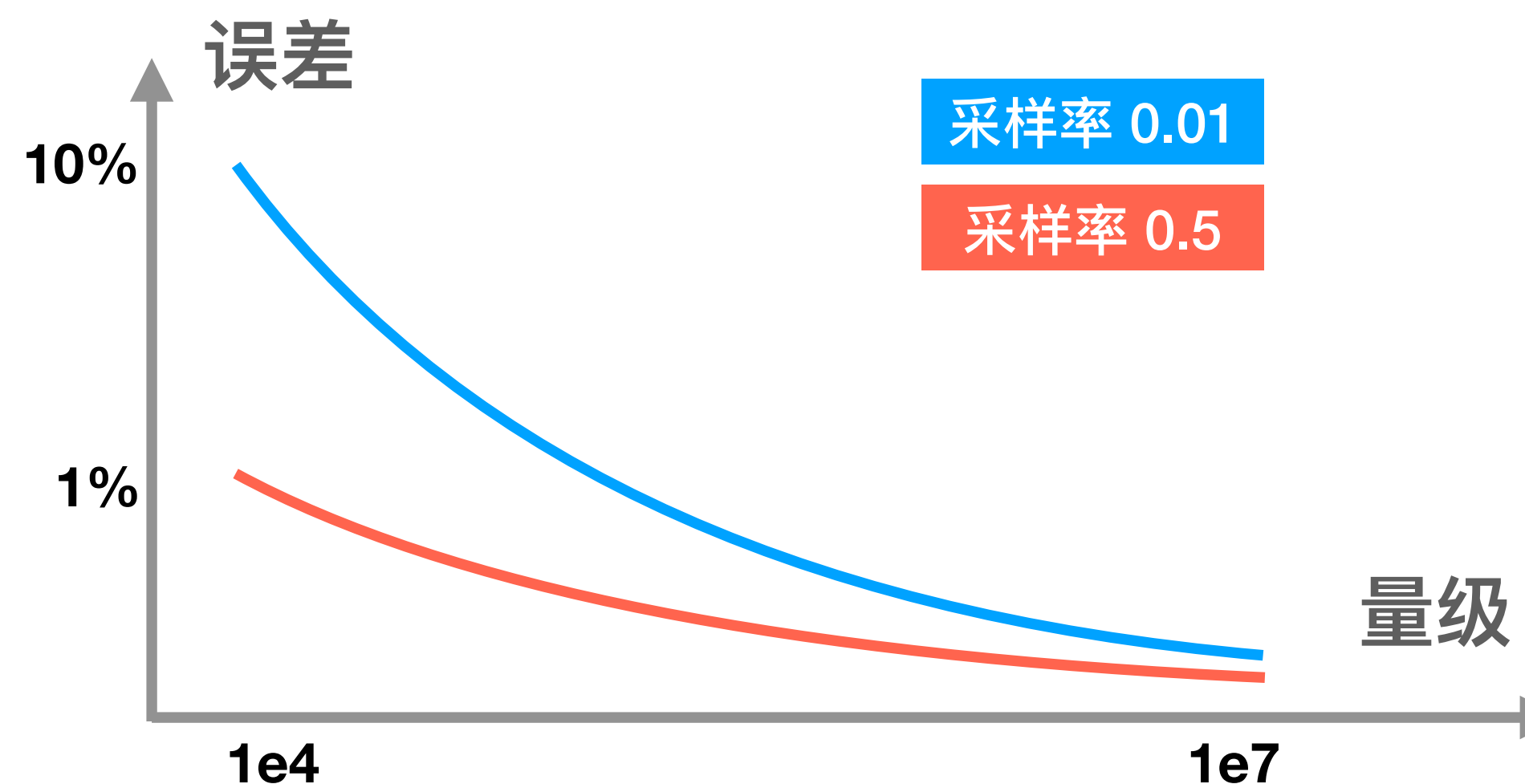
# 优化数据上报

## 总量控制

- 基于设备 ID 采样
- 白名单强制上报
- 容器关闭时一定上报

## 数据压缩

- 一次请求聚合更多记录
- 移除 query / cookie
- Gzip



# 优化数据构成

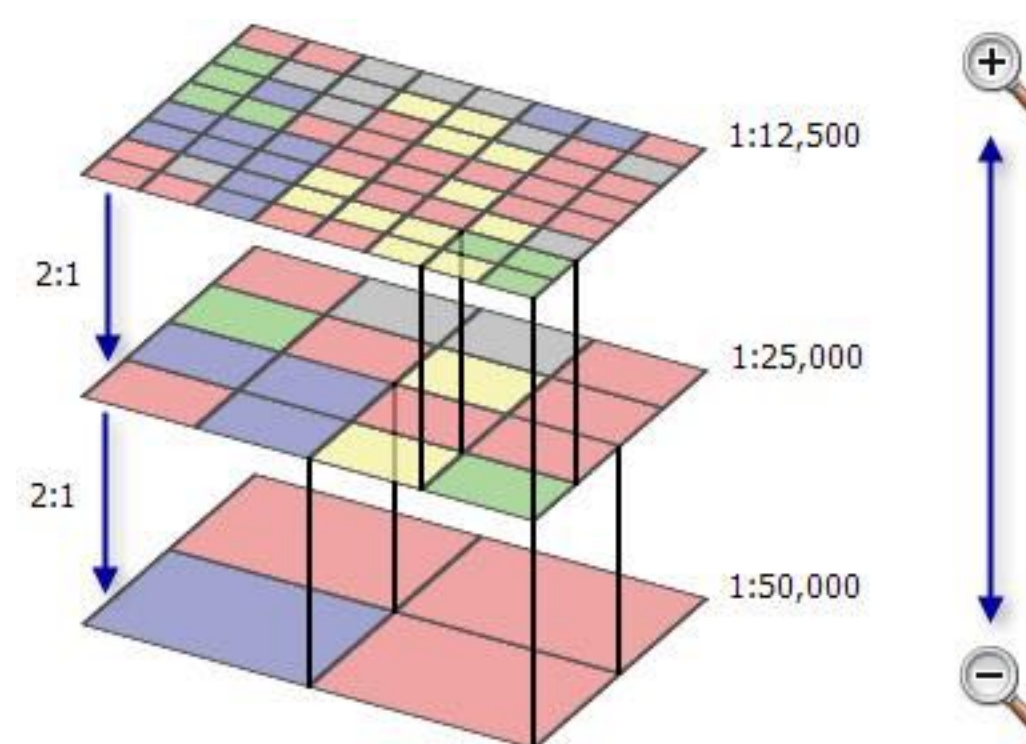
上报那么多数据，你真的看过吗

- 每天 **100W URL × 100W URL** 的日志
- 每天上报 **全中国所有的** 地理坐标

**https://i.meituan.com/trip /123/456**  
**https://i.meituan.com/trip /abc/def**  
 ↓  
**https://i.meituan.com/trip /\*/\***

简单的策略减省大量的冗余

- URL 泛化
  - 优化后每个维度减少 **99% 冗余数据**
- 地理网格
  - 从无穷多的坐标到有限的地理网格



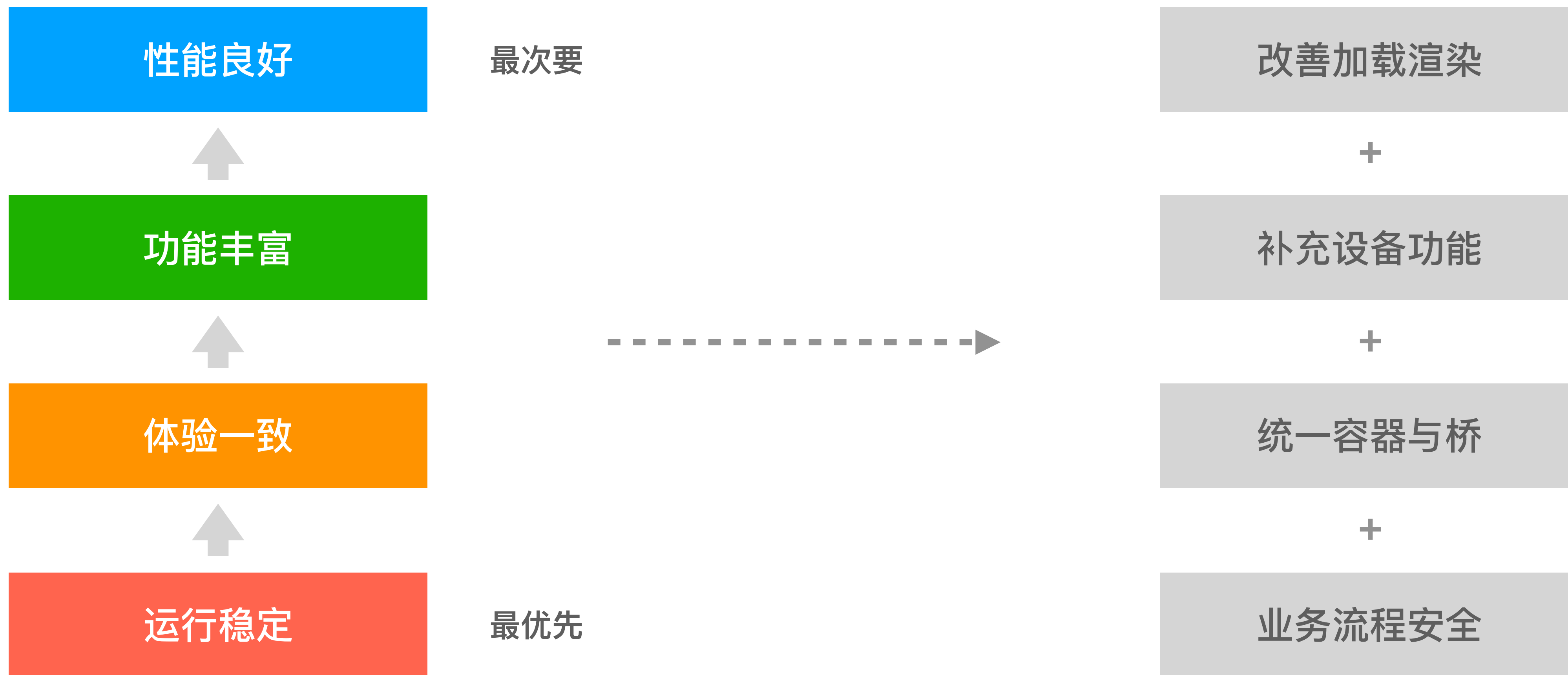
**116.487353, 40.008305**

**数据规模 e17**



**116.5, 40.0**  
**数据规模 e6**

# 容器与桥的原则



# Q&A

更多技术资料  
欢迎关注“美团点评技术团队”



招聘：前端工程师

邮箱：[yanghao04@meituan.com](mailto:yanghao04@meituan.com)

# 桥的通信原理



# 容器的内核选型 / Android

## WebView

- 基于系统 WebView 增强
- 独立于系统升级
  - 从 Android 4.4 开始

## 桥通信

- JsPrompt
  - Cordova
- JavascriptInterface
  - 存在安全问题 / 4.2+ 注解显式启用



*BABEL*





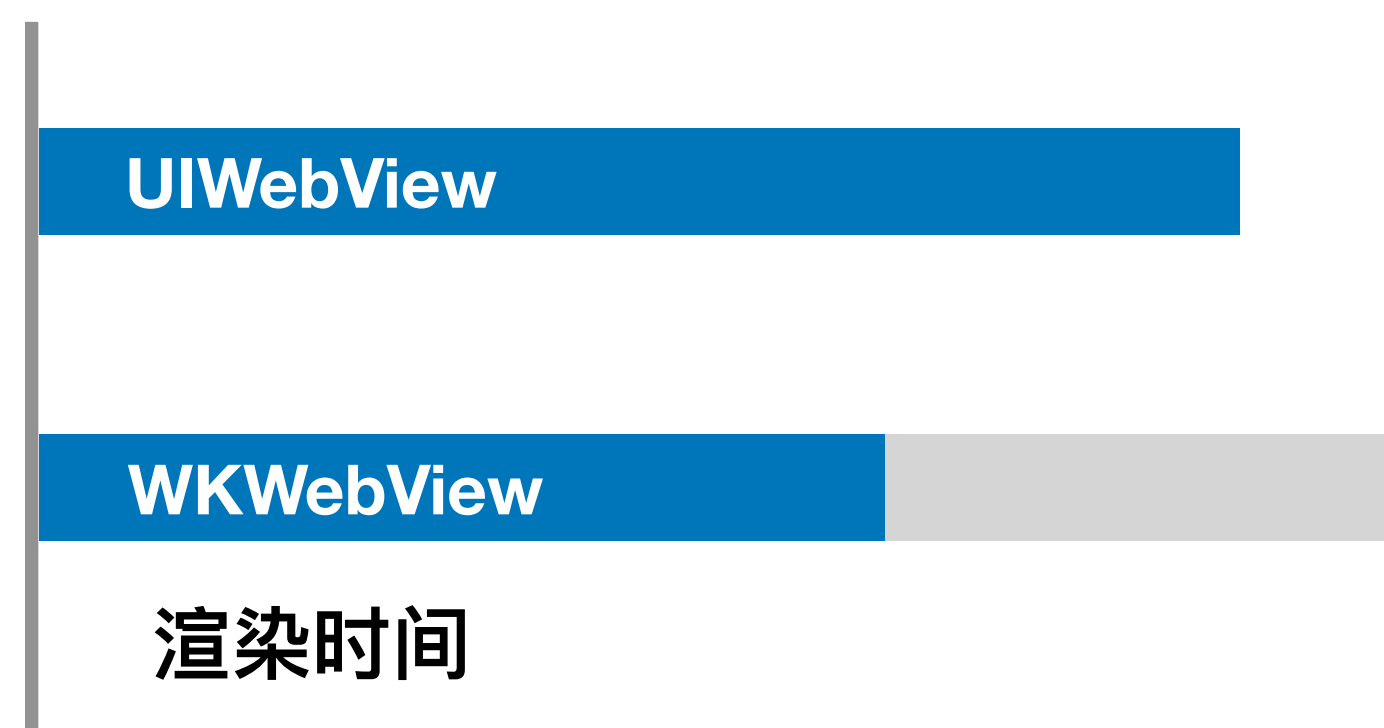
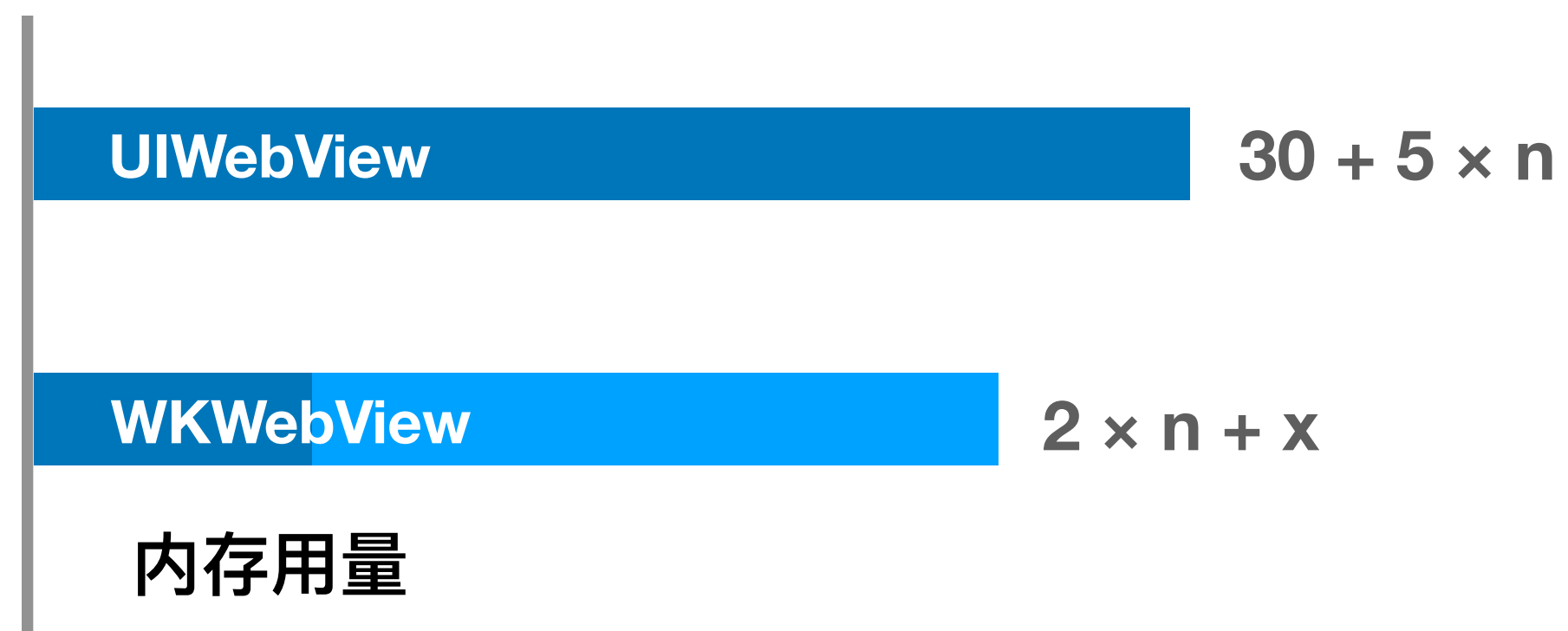
# 容器的内核选型 / iOS

## WebView

- 基于 UIWebView 增强
- 支持 WKWebView
  - 进程隔离 / 不支持任意拦截

## 桥通信

- shouldStartLoadWithRequest
  - 构造 iframe 发起请求
  - Cordova
- WKScriptMessageHandler



# WebView / Native 动态化

## WebView

- 基于 Web 技术
- 跨平台、多 App 复用
- 性能一般 / 功能依赖桥

## Native 动态化

- 基于自定义 DSL
- 用户体验好
- 缺乏规范 / 维护成本高

