

# 手机淘宝的客户端架构探索之路

冯森林 / 无锋

资深技术专家

Alibaba · 无线事业部

# 关于

- 冯森林 ( 无锋 / Oasis Feng )
  - 2005 : 华为 · 核心网 · 平台技术部
  - 2008 : 阿里巴巴 · 淘宝网
    - 2008 : 服务端 SOA 架构、大规模分布式系统中间件
    - 2010 : SNS 项目 , 从后端、前端 到 客户端技术
    - 2012 : 无线事业部 · 客户端架构
- 阿里巴巴无线事业部
  - 负责手机淘宝并为阿里巴巴各条无线产品线提供基础技术和设施

## 产品挑战

- 承载并整合多样化的业务生态



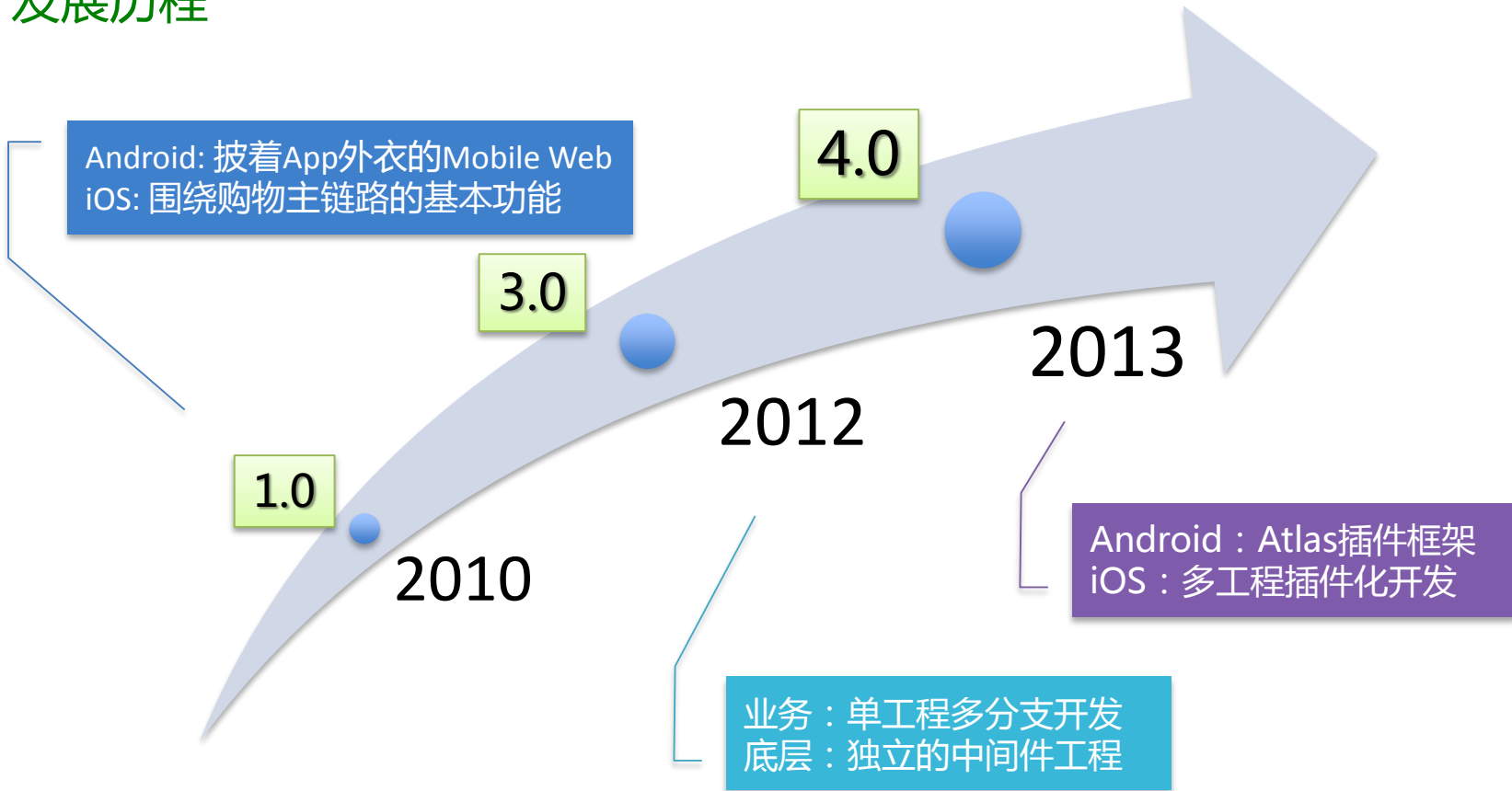
## 研发挑战

- 去年『All-In』的时候
  - 大量业务的同时涌入
  - 火车模型的悬崖效应
  - 10余个团队的代码整合
  - Android 2.x 方法数天花板

『量变』呼唤『质变』！



# 发展历程



## 不同的探索 - Hybrid

- WindVane ( 2012-2014 )
  - 衔接 Android/iOS 的导航交互及动画
  - URL总线 ( 对接Native UI总线 )
  - JS Bridge : JS <-> Native ( + 性能及安全性增强 )
  - 缓存 & 预缓存 ( 兼容Cache-Control + AppCache )
  - 数据采集、本地/网络性能监控
  - 安全加固、审核、隔离，人机识别
  - 增强的网络层 ( SPDY、DNS旁路解析等 )

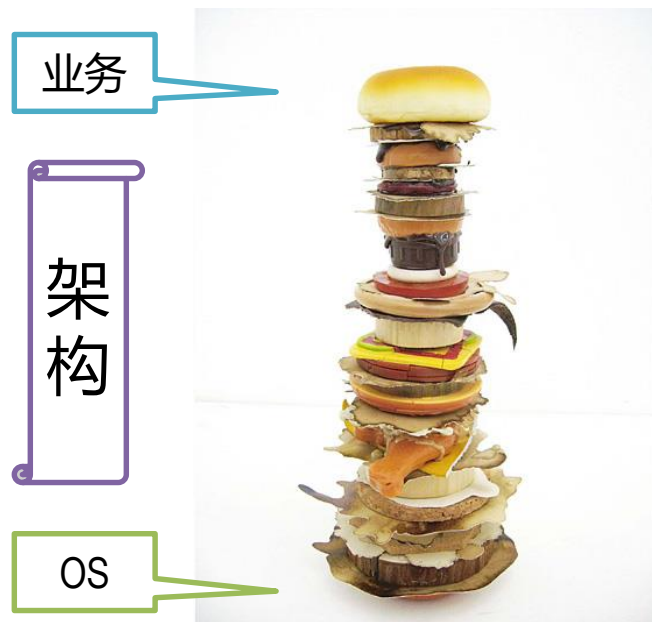


# 2014

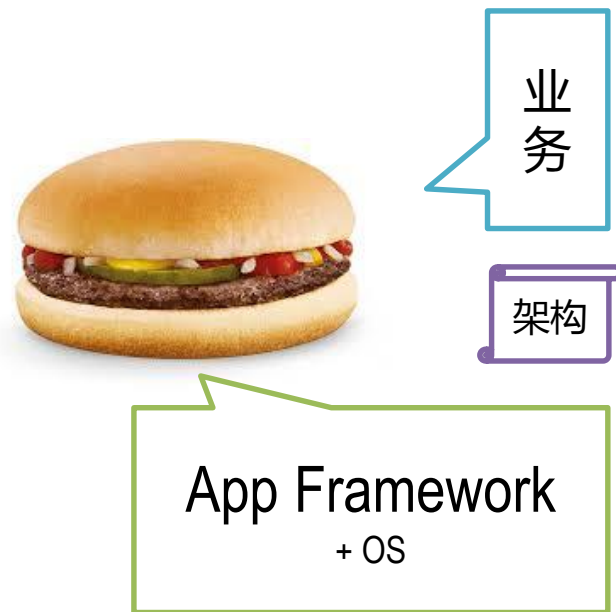
手机淘宝自诞生以来，最大规模的底层重构

归一、轻量、透明、延展、敏捷

## 反思



传统的服务端架构

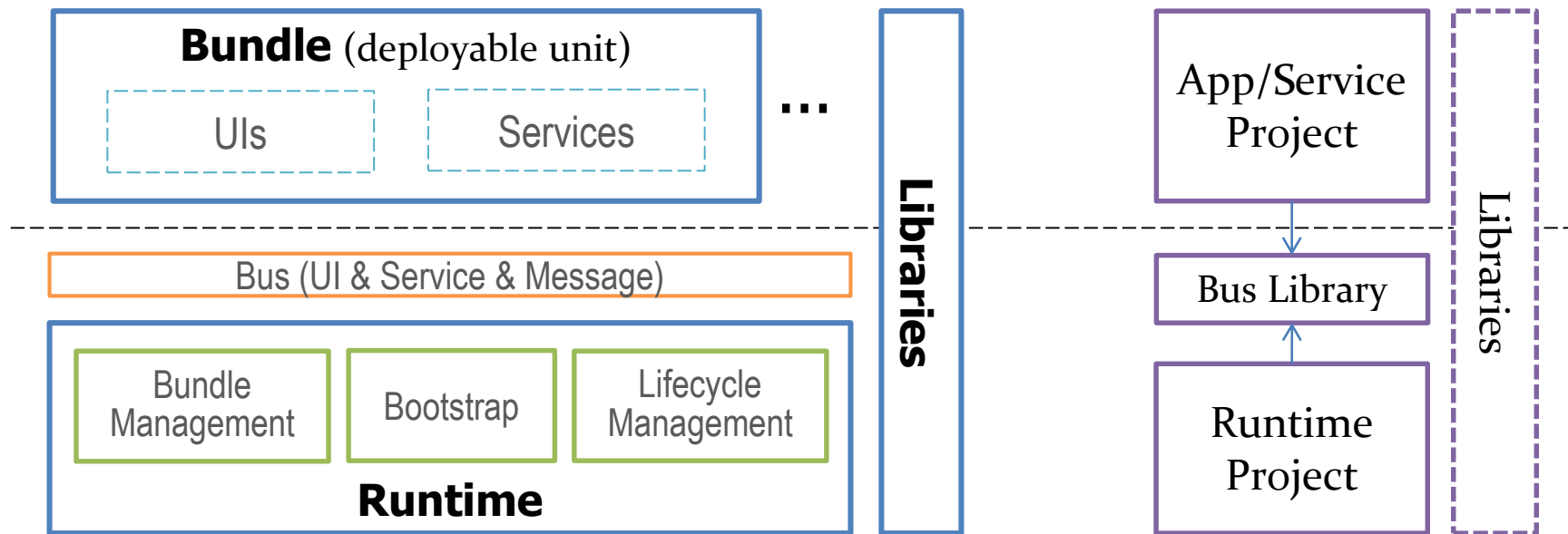


客户端App架构



归一

- 一切皆组件：告别插件，拥抱组件（bundle）

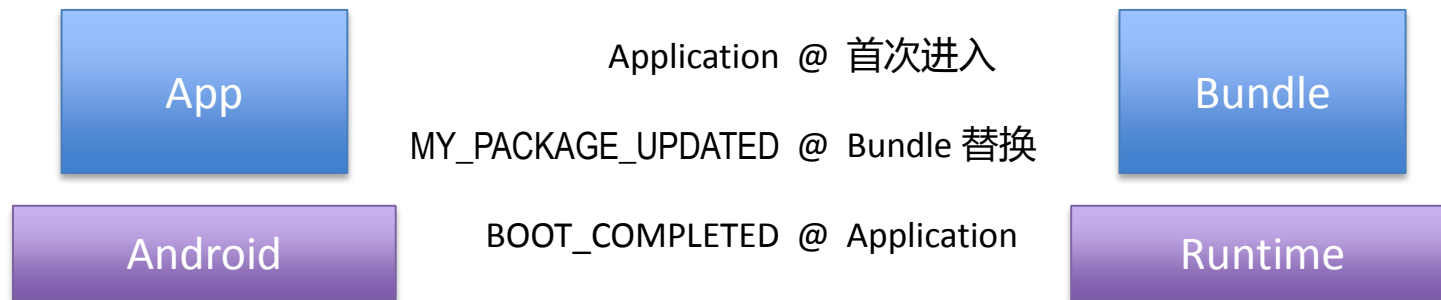


## 轻量

- 微内核：只负责最基础的管理职能（~150K）
  - 启动流程
    - 容器初始化
    - 类加载（首次启动还需dexopt）
    - 核心中间件初始化（可异步）
    - 启动入口Bundle
  - 组件管理：只负责『添加』、『删除』、『替换』（支持批量事务化）
    - 与上层的动态部署组件配合完成组件的在线升级

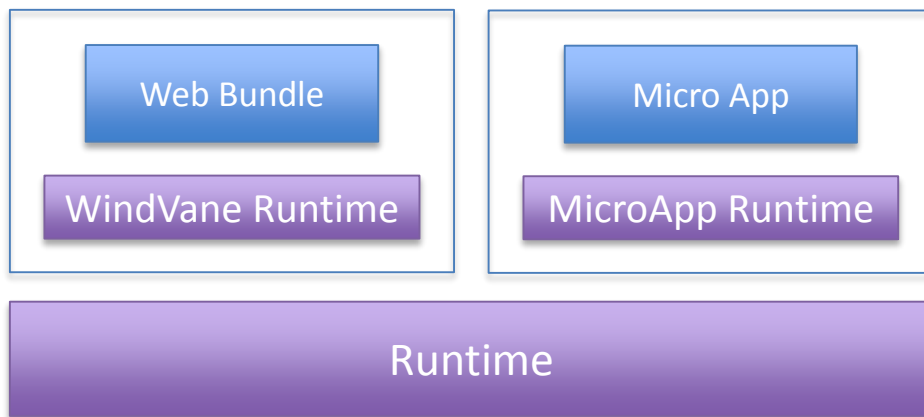
## 透明

- 透明的容器（Android）：Bundle 即 App，容器亦 OS
  - 生命周期管理：Bundle 开发与 App 开发的无差异化



# 透明

- 透明的容器：Bundle 即 App，容器亦 OS
  - 二级容器



# 透明

- 总线
  - UI总线：以跨平台统一的URL作为寻址方式（ Web、 Android、 iOS ）
    - item.taobao.com/... → DetailActivity / TBTradeDetailViewController
    - 自动降级：没有 Bundle 承载的 URL，将自动以 Web 容器加载
    - Android：去中心化 + URL Hook / iOS：中心分发（支持Hook）
  - 服务总线：基于接口的服务调用（ Android：兼容原生AIDL ）
    - [Android] 轻量化的服务启动（向下兼容原生的Service Binding）
    - [Android] 支持跨进程、跨App的服务部署
  - 消息总线：基于OS原生消息机制（ Broadcast / NSNotification ）
    - [Android] 进程内使用轻量级消息通道

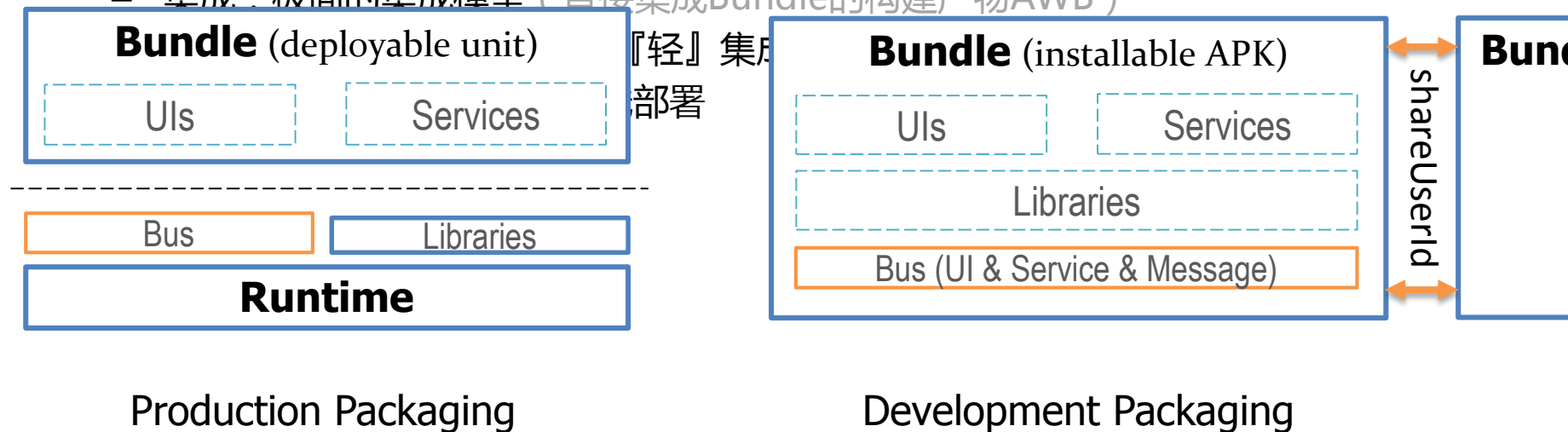
## 延展

- 标准化 ( Android )
  - Bundle的交付产物：AAR，构建产物：AWB ( 相当于APK剔除了共享依赖 )
  - Bundle间可跨进程，甚至跨App对接
- 灵活性
  - Bundle的自由组装 ( 手机淘宝、淘宝生活、码上淘 )
  - Bundle对容器无依赖
- 适应性：从微型App到巨型App，按需采用的容器能力
  - 容器完全独立的三大职能：启动加载、生命周期管理、组件管理
  - [Android] 三大总线均为OS能力的封装，具有极佳的互操作性



## 延展

- 从巨型App时代的臃肿，回归田园App时代的轻盈
  - 开发：像微型App一样便捷的开发调试（不依赖容器）
    - 开发环境的Bundle构建速度：3~10秒
  - 集成：极简的集成模型（直接集成Bundle的构建产物AWB）



# 敏捷

- 『Move fast and break things』

via Hot Patch

- 线上严重问题的快速修复（小时级的响应时间）
- 可对 Android Framework 代码打补丁
- AOP 编码形式
  - Before / After / Replace 某个方法
- 支撑了『Project Nish』（目前尚未公开的神秘项目）
- 暂时只支持 Dalvik，即将适配 ART 和 阿里云OS



即将开源，敬请期待！



## 耗时2个月完成

- 6月初上线以上
  - 集成 Bundle : 30+
  - 改造为服务 : 10+ ( 登录、缓存、网络访问层..... )
  - Hot Patch 修复线上严重故障 10+ 起
  - Patch 平均大小 : 5KB
  - 最大的阵痛 : 生产构建打包系统的改造 ( 因涉及改写aapt实现资源分区 )

## 思考：从『壶中界』走向『云和山的彼端』

- Native App 能否像 Web 网站一样随时部署，即用即取？
  - 资源（图片）的『云端 + Cache』化
  - 中间件的全面Bundle化，Bundle的细粒度化
  - 基于依赖结构的代码（Class）按需加载机制（类似 RequireJS）

像 Web 一样轻盈的 Native App ！



# Thanks!

冯森林 / 无锋 / Oasis Feng  
资深技术专家 @ Alibaba 无线事业部  
Twitter / G+ / LinkedIn / anywhere: oasisfeng

