

# 混合应用平台构建实战

小米 | MIUI | 董红光

# 项目背景

# MIUI直达服务平台

- 应用和网页服务形态的问题
- 新的服务形态特点
  - 用户体验提升
  - 开发者获益
  - 平台提升效率

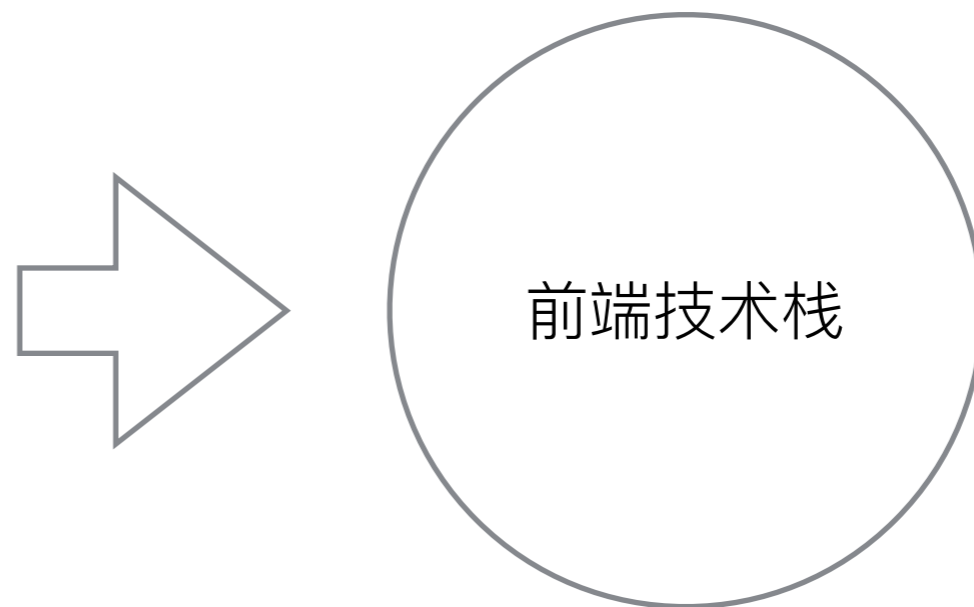
# 平台目标

- 无需下载和安装
- 接近原生的能力
- 接近原生的性能和体验
- 学习门槛低，开发效率高
- 支持降级处理

# 技术选型

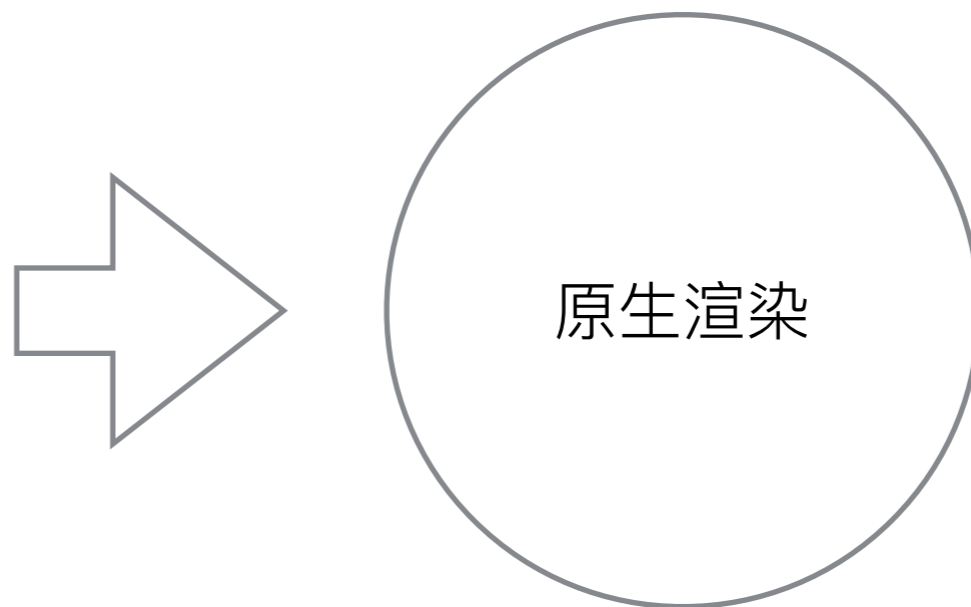
# 技术栈

- 运行效率
- 资源复用
- 学习门槛
- 开发效率



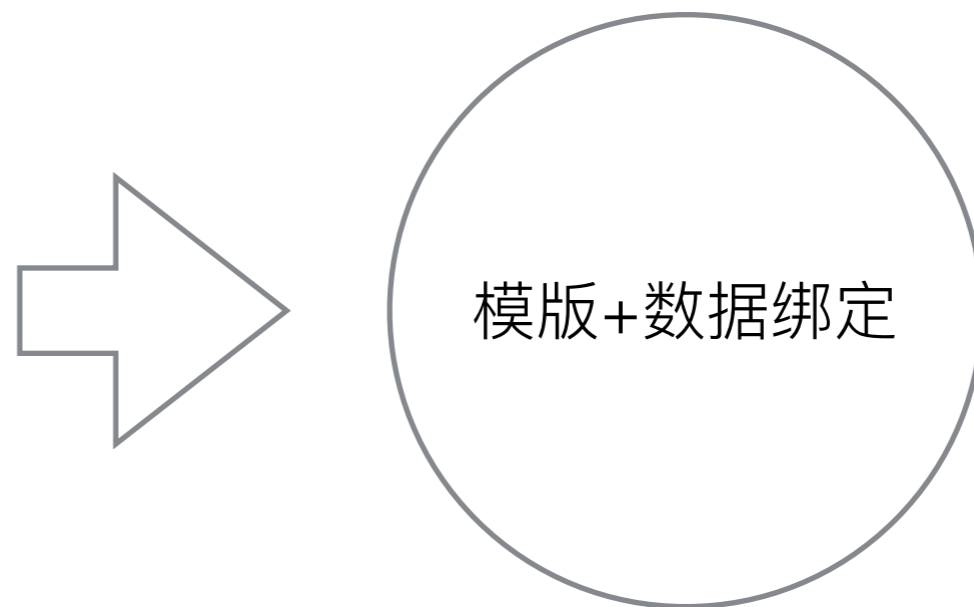
# 渲染引擎

- 体验
- 性能
- 内存
- 可行性



# 应用框架

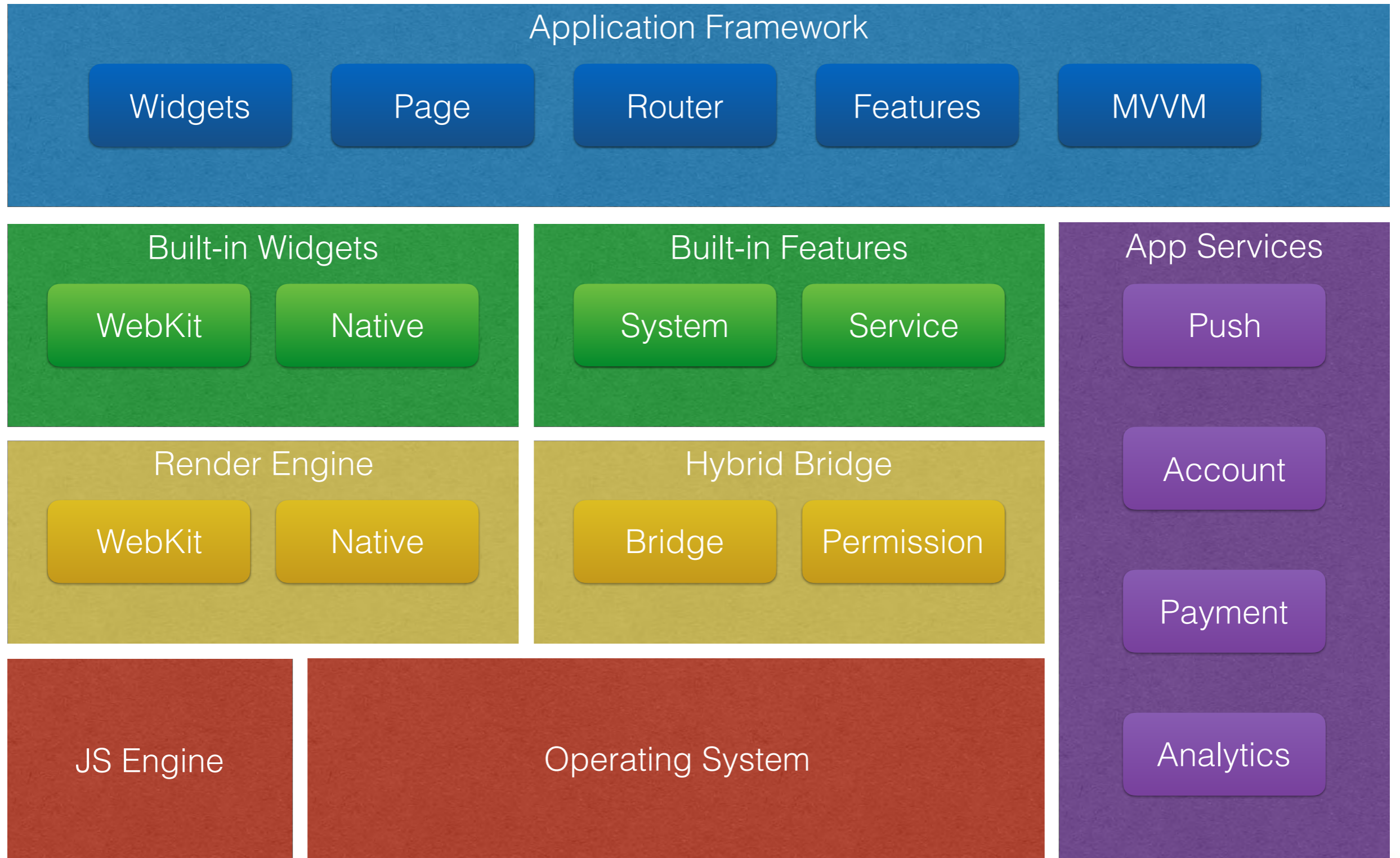
- 可行性
- 学习门槛
- 项目复杂性
- 易维护性





运行时

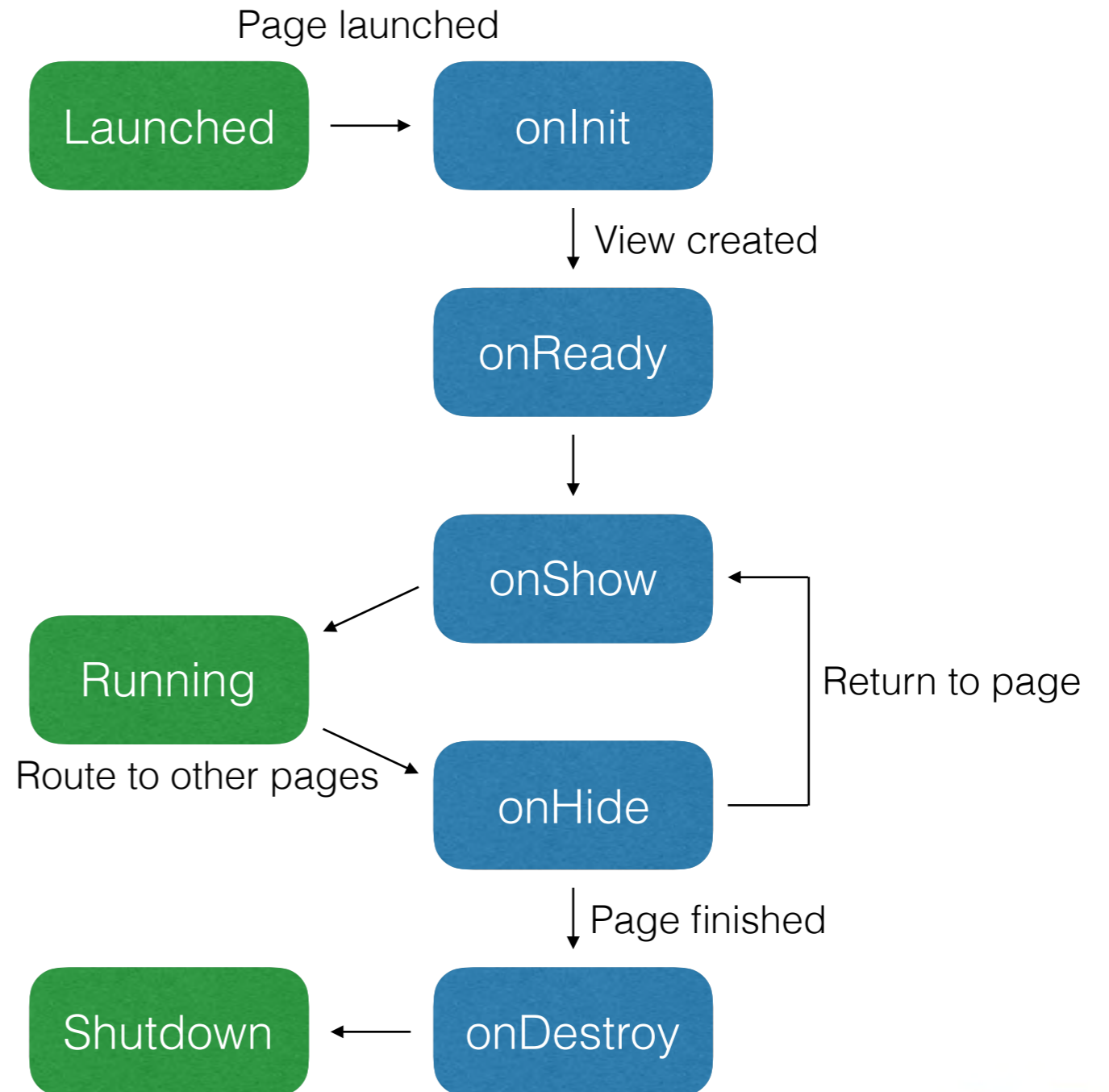
# 架构 - Runtime



运行时：应用框架

# 页面

- 基本形式
  - 模版+数据绑定
  - template/style/script
- 生命周期管理
  - 视图销毁
  - 页面销毁



# 组件化

- Web Components
- 基本形式：同页面
- 可用性
  - 参数传递
  - 事件传递
  - 占位符

# 布局与样式

- 设计依据
  - 使用频率与复杂性
  - 遵循标准
- 布局：盒模型 + Flex布局
- 选择器

运行时：渲染引擎

# 性能瓶颈

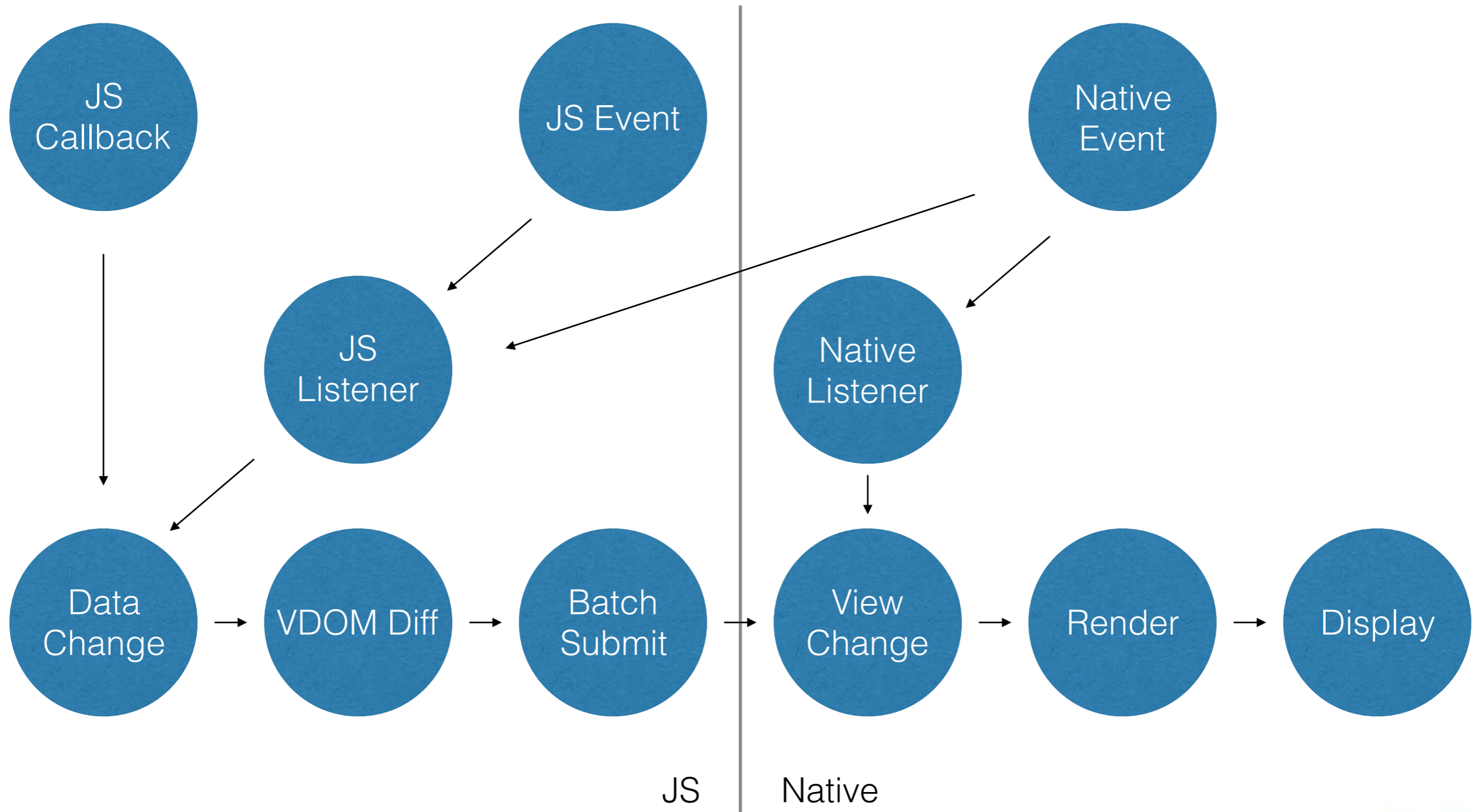
- 传统前端瓶颈
  - DOM操作
  - 动画
  - 单线程
- 原生渲染瓶颈
  - JavaScript <-> Java



# 虚拟DOM

- 操作代价低
- 可批量提交
- 易于帧同步
- 可多端输出

# 渲染流程



# 视图复用

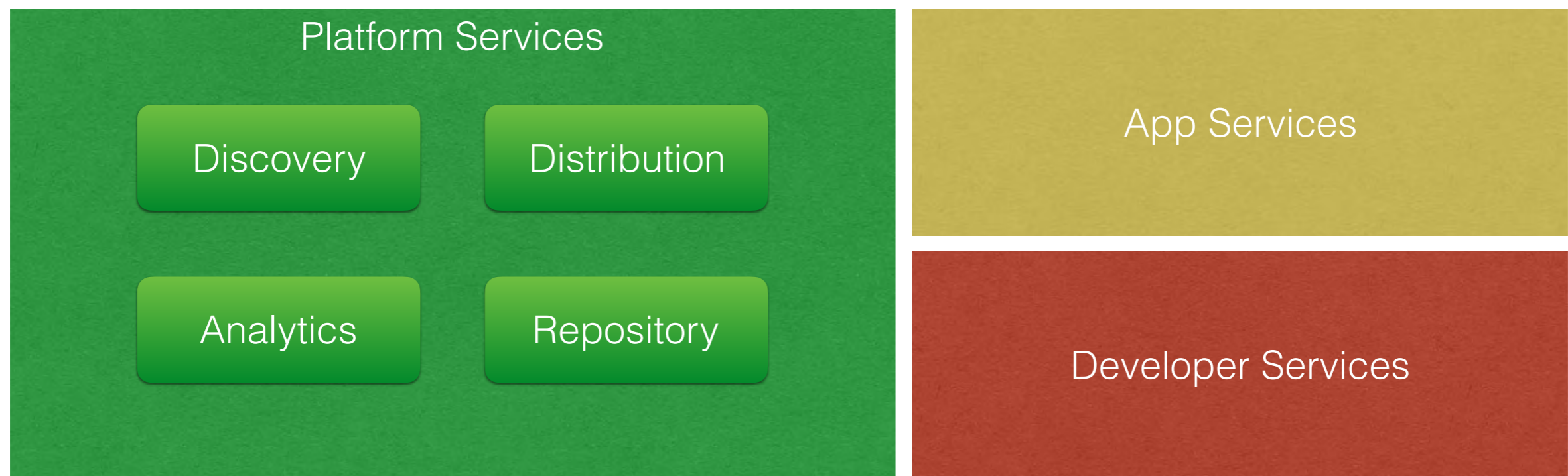
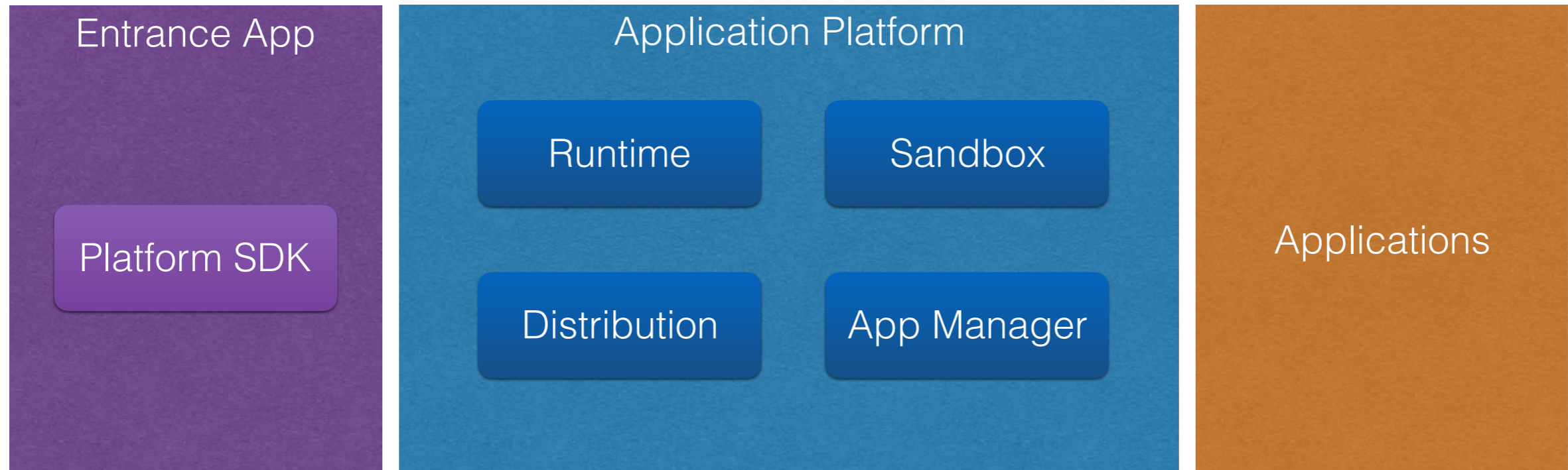
- 相同节点的识别与移动
  - unique id
- 相同类型的视图复用
  - view type
- 窗口区域
  - 滚动策略

# 动画

- CPU -> GPU
  - 降低reflow/repaint
  - transform
- 动画声明
  - animation + @keyframe

# 平台化

# 架构 - Platform



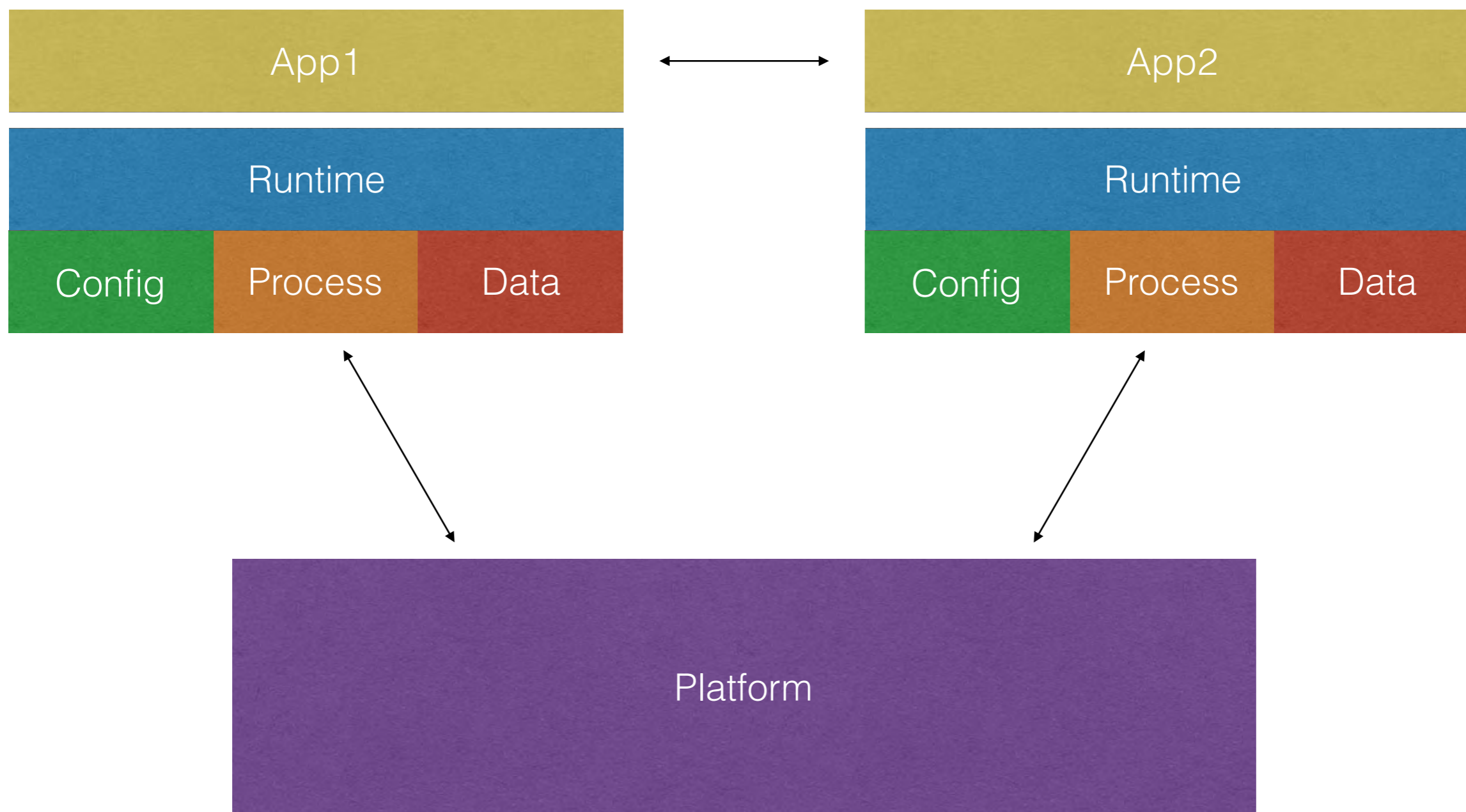
# 平台化：隐私与安全

# 权限与签名

- 应用权限
  - 权限声明
  - 用户二次确认
- 应用签名
  - 自签名
  - 证书



# 沙盒机制



平台化：秒开

# 秒开

- 首次使用
  - 初始大小
  - 应用拆分
- 后续使用
  - 离线缓存
  - 应用更新

THANKS!

