



主办方: **msup**[®] | **ARCHNOTES**
架构

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE

在线教育业务场景下的 跨平台应用框架设计与实践

VIPKID 曹斌 & 袁弋洋



01

VIPKID 在线教室跨平台技术演进

02

VIPKID 应用层跨平台技术演进



| 什么是跨平台？

可以在多种操作系统不同硬件架构的平台上工作



| 如何□ □ 跨平台□?

跨平台语言

"一次编写, 到处运行" — Java

"一次编写, 到处??" — C++

跨平台中间件

Socket、OpenGL

Asio、CEF

跨平台开发套件

Qt、Electron

React Native、Flutter



跨平台技术型思路



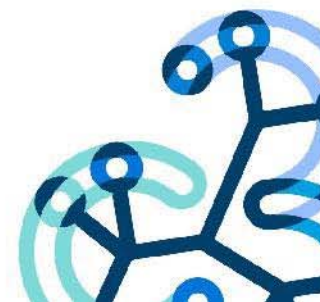
业务场景推演
业务场景和目标平台



人员结构评估
现状、市场行情、影响力



分层抽象
自顶向下



VIPKID 在教育场景概览

学习中心

在线教室、辅助学习
Windows、macOS
iPad、APad



Dino Home

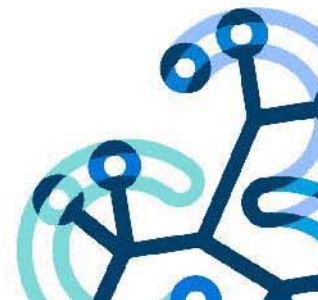
监课、回放
Android、iOS

VIPKID Teach

在线教室、课程管理、备课、教师招募
Windows、macOS
iPad
Android、iOS

VIPKID 英语

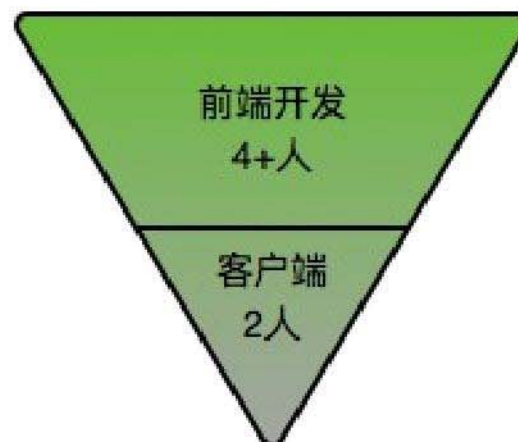
约课、监课、回放、学习报告
Android、iOS



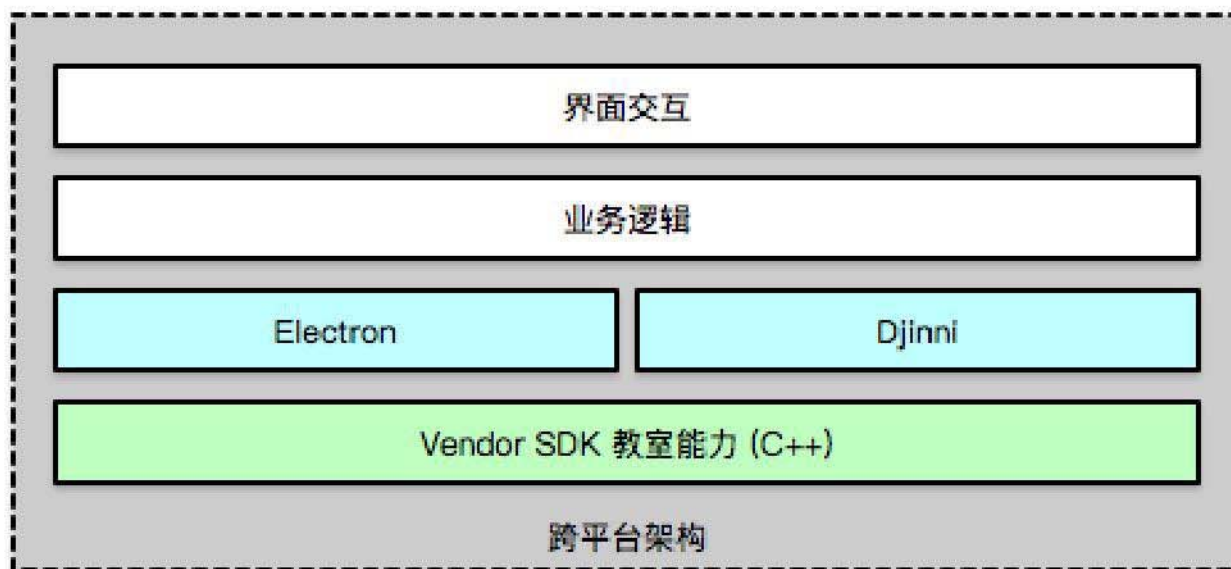
人口结构



在线教室初期
快速上线



跨平台方案




 + Djinni
Electron



| Djinni /genie/

自动生成跨语言接口

Java proxy class & JNI marshaling
Obj-C interface and Obj-C++ marshaling
C++ abstract base class

使用 IDL 描述接口和数据
结构



跨平台方案改□



Qt +  + Djinni



如何设计跨平台内核？

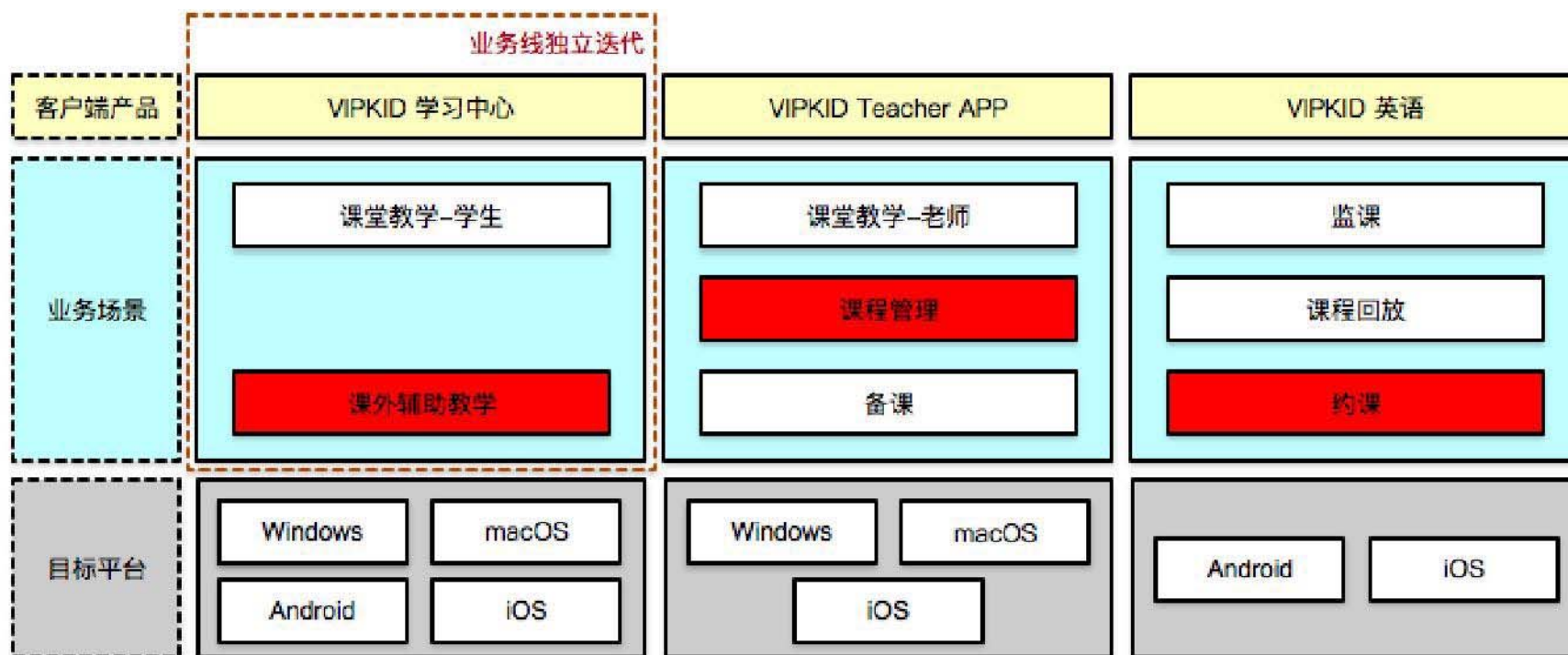


| 如何 □ □ 跨平台内核

适度选取、自顶向下
分层和抽象



□ □ □ 景分抽象 □



Web 技术实现



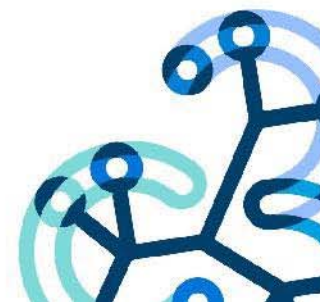
□ □ □ 场景抽象 □

□ □ □ 件 □

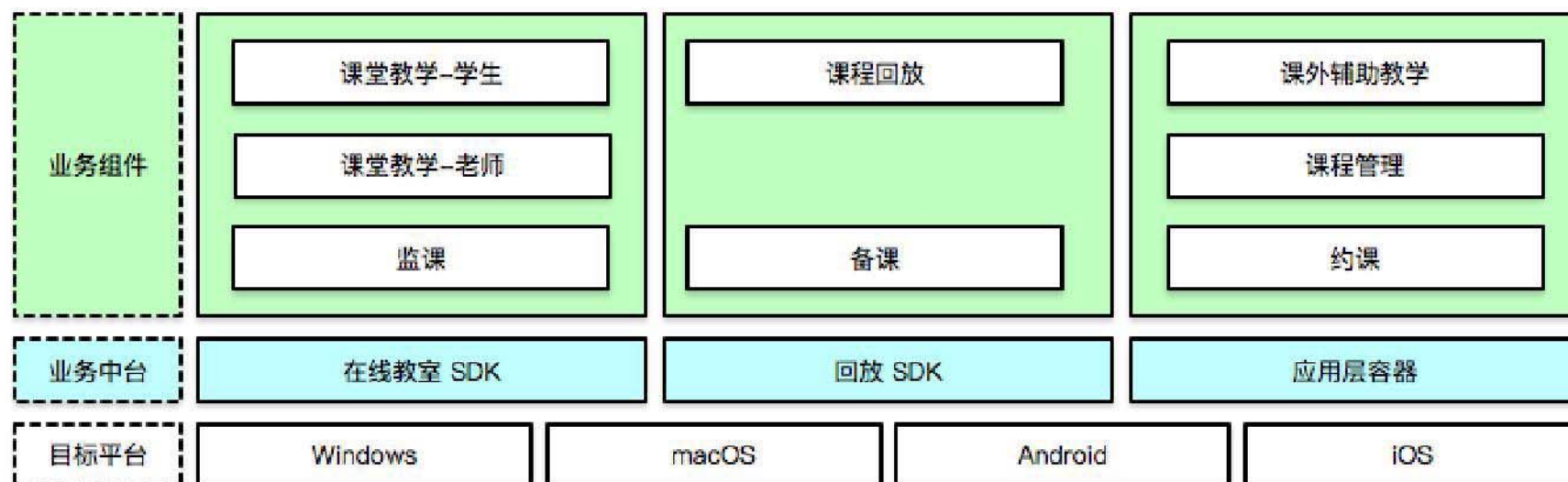
客户端产品功能快速迭代开发

□ □ 中台 □

围绕在线教室相关场景进行能力抽象
提供平台友好的接口供业务组件层进行调用



□ □ □ 景分抽象 □



中台分抽象

在「教室基」

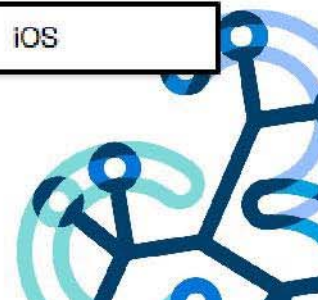
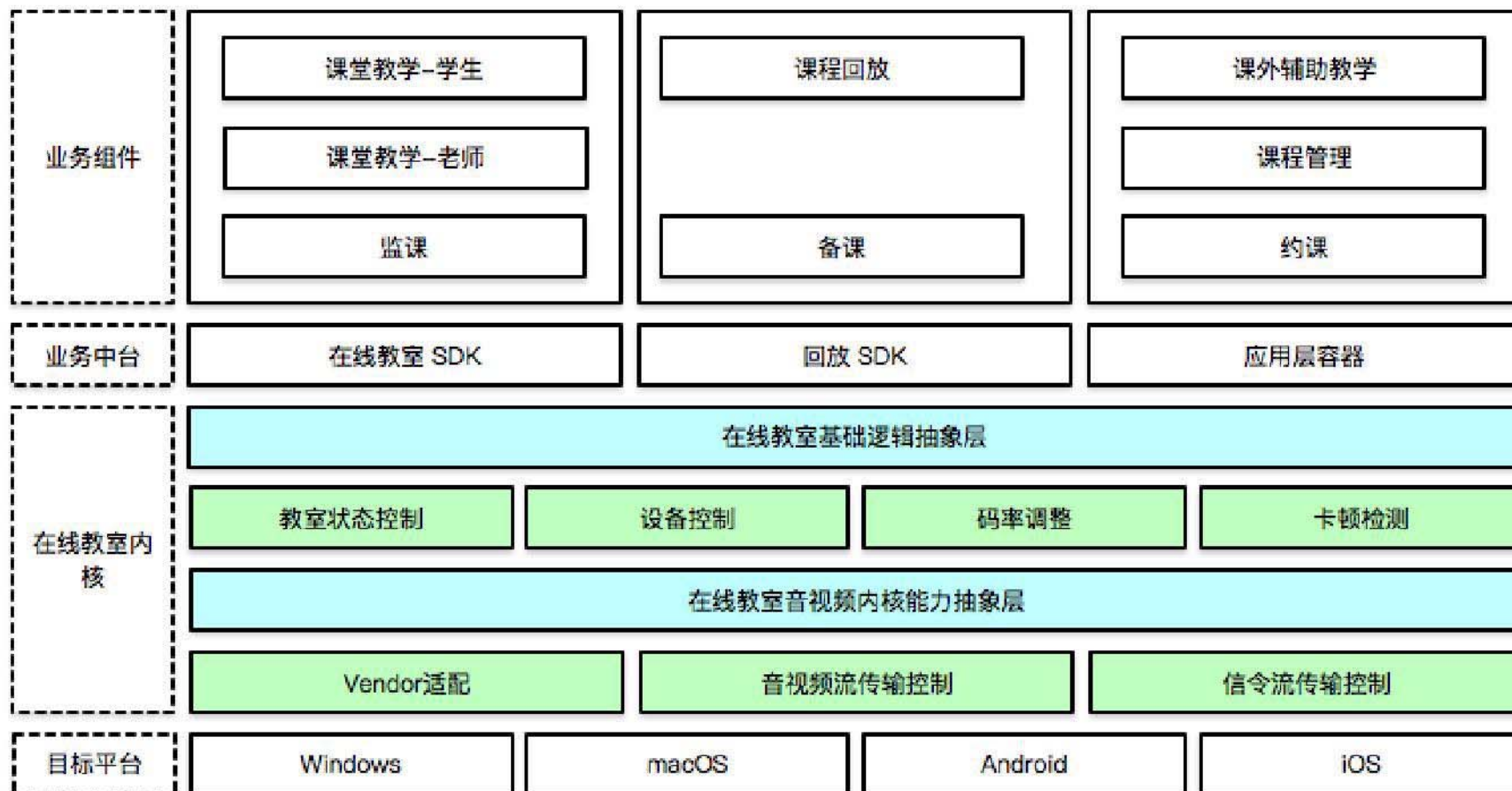
不同平台和网络环境下构建一个稳定、高质量的在线教室最小单元

在「教室」内核

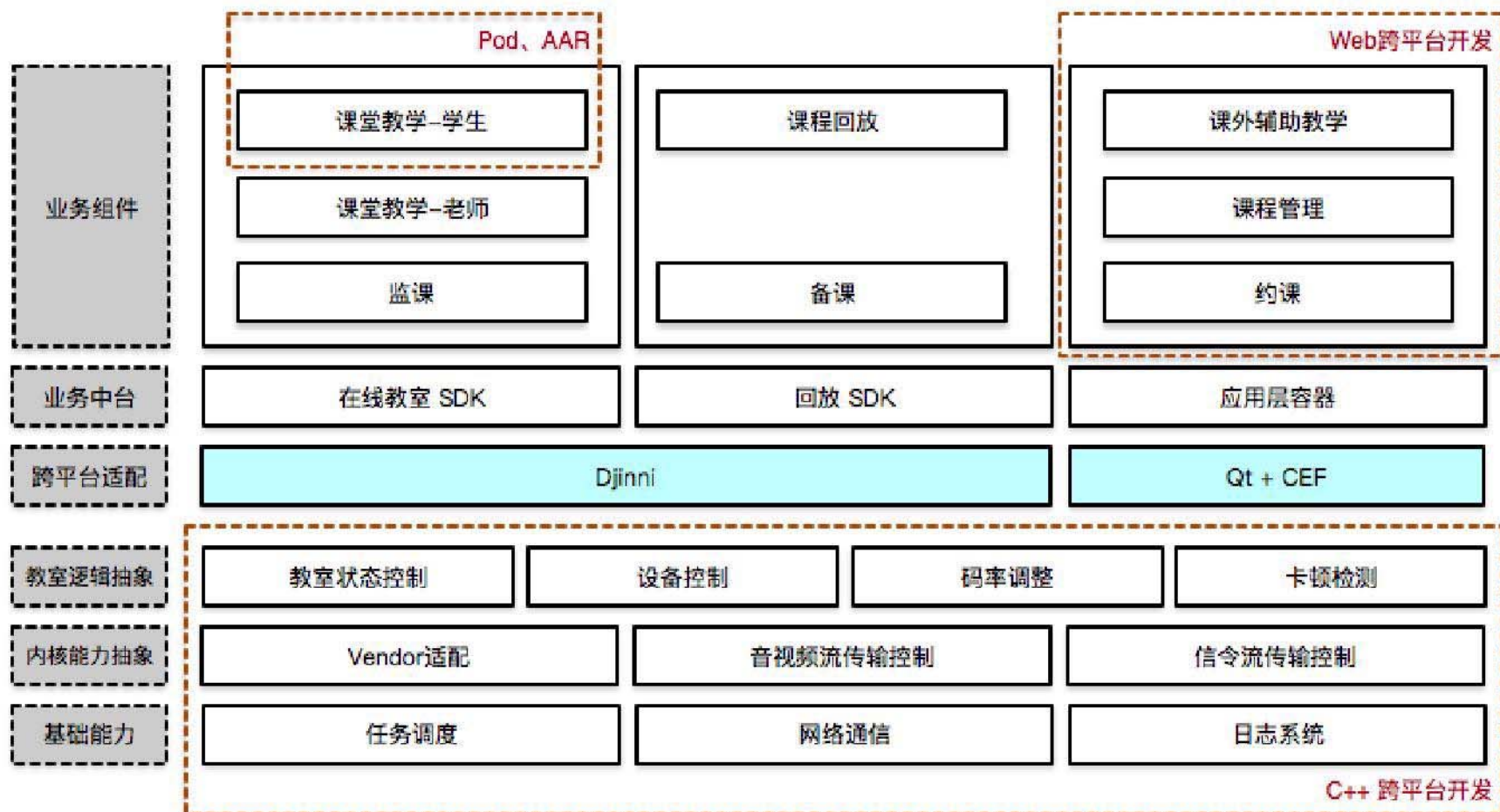
抽象音视频、信令传输能力



□ □ 中台分抽象 □

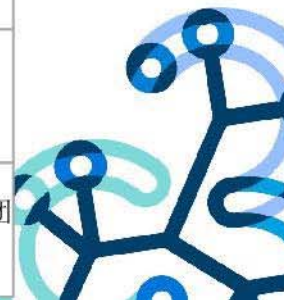


VIPKID 在 □ 教室跨平台整体架构



□ 用跨平台的方案□

方案	Web+WebView	RN/Weex	Flutter
开发效率	高	一般（需要特定的开发和调试工具）	一般（需要特定的开发和调试工具）
开发语言	HTML+CSS+JS	JS(React/Vue)	Dart
社区生态	很成熟	发展了一段时间 (RN>Weex)	起步，但是发展迅速
渲染性能	低	高于 WebView(取决于 JSC 的效率)	高 (自建的平台渲染引擎)
上手成本	低	中	高
维护成本	低	中	高
适合场景	快速试错和迭代，人力紧张的团队	在满足业务快速迭代的前提下，追求更好的性能和体验	追求更高的性能，开发资源充裕的团队



用跨平台的方案 定义架构

终端渠道



业务需求

周期短

持续交付

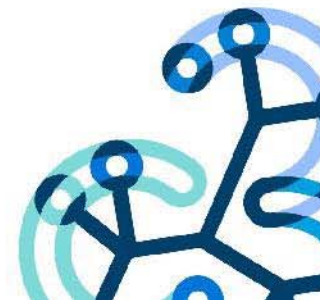
云化

技术栈

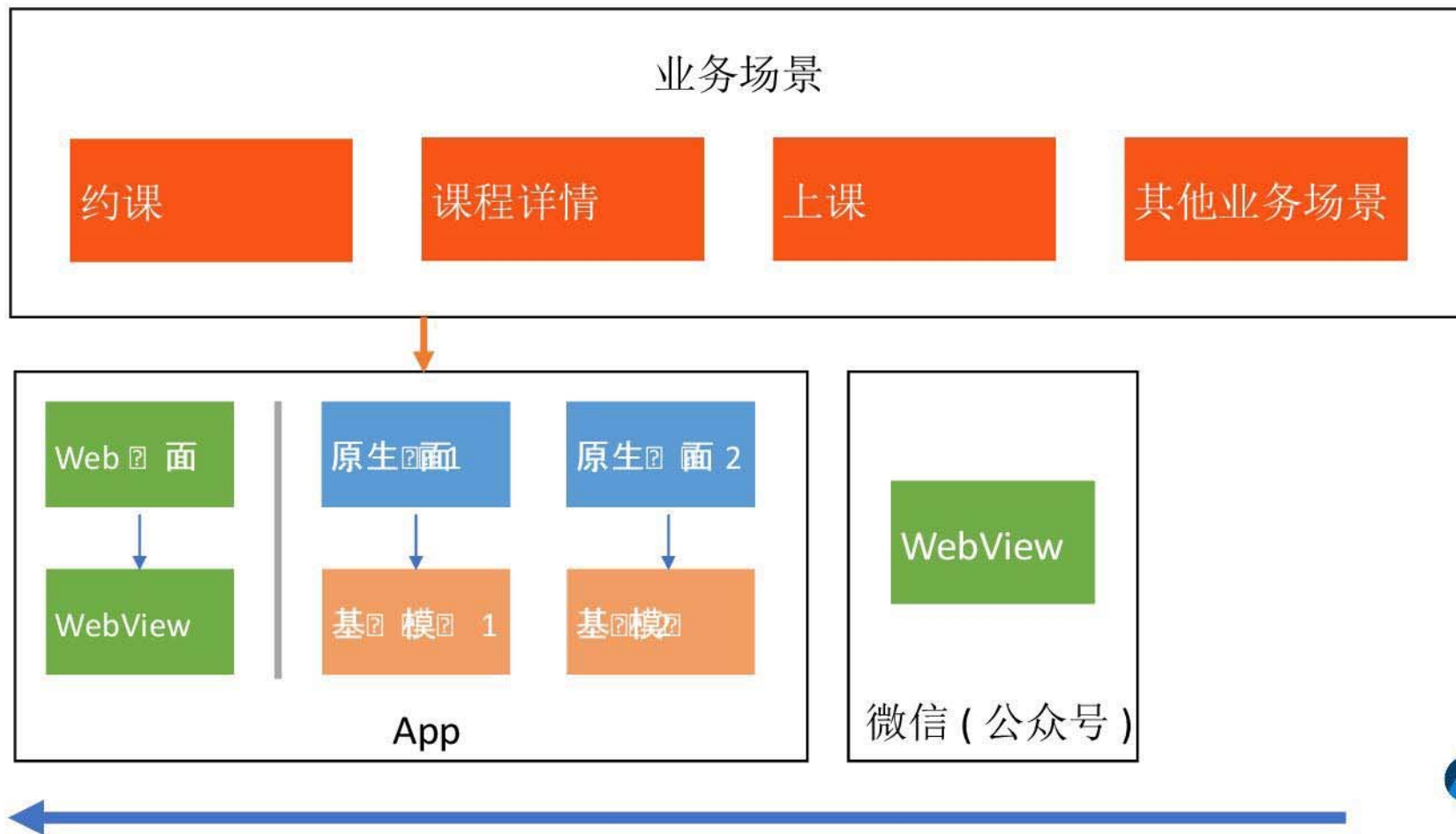
Web 前端

X

Native 开发



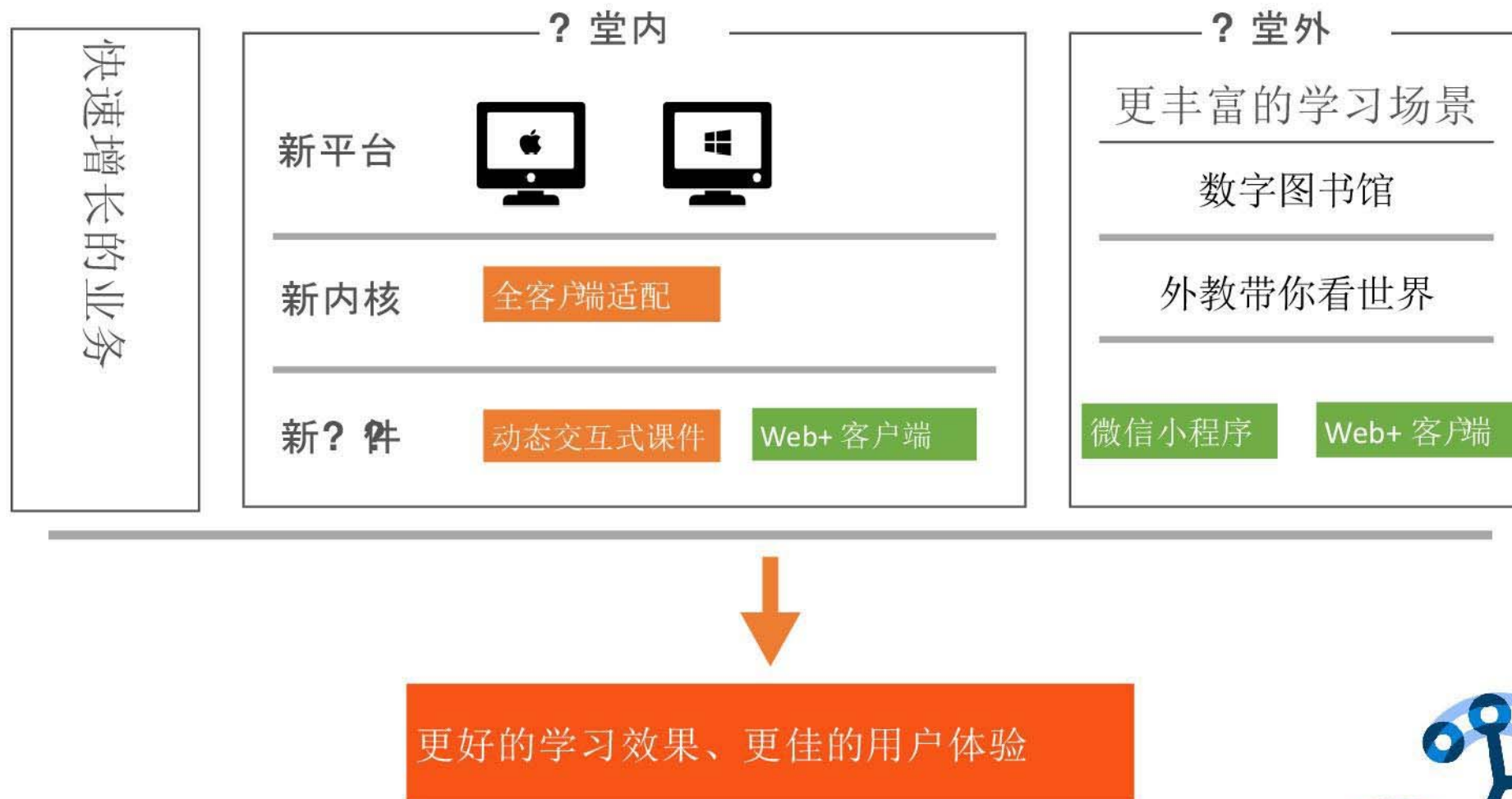
VIPKID 跨平台方案—□ 用容器化



Before 2017



VIPKID 容器化方案——来的挑



VIPKID 跨平台方案演进

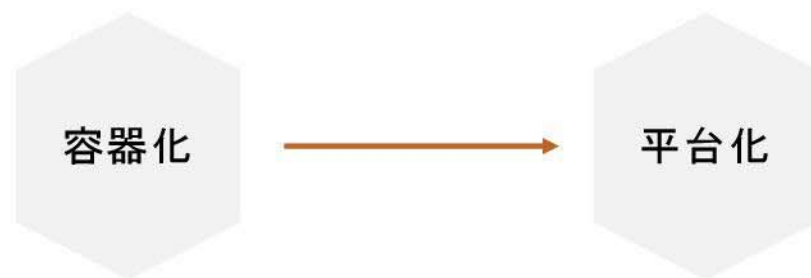
CASE 1: 如何将? 生在? ? ? 件中的操作同步? ? 方

构建容器统一通信框架

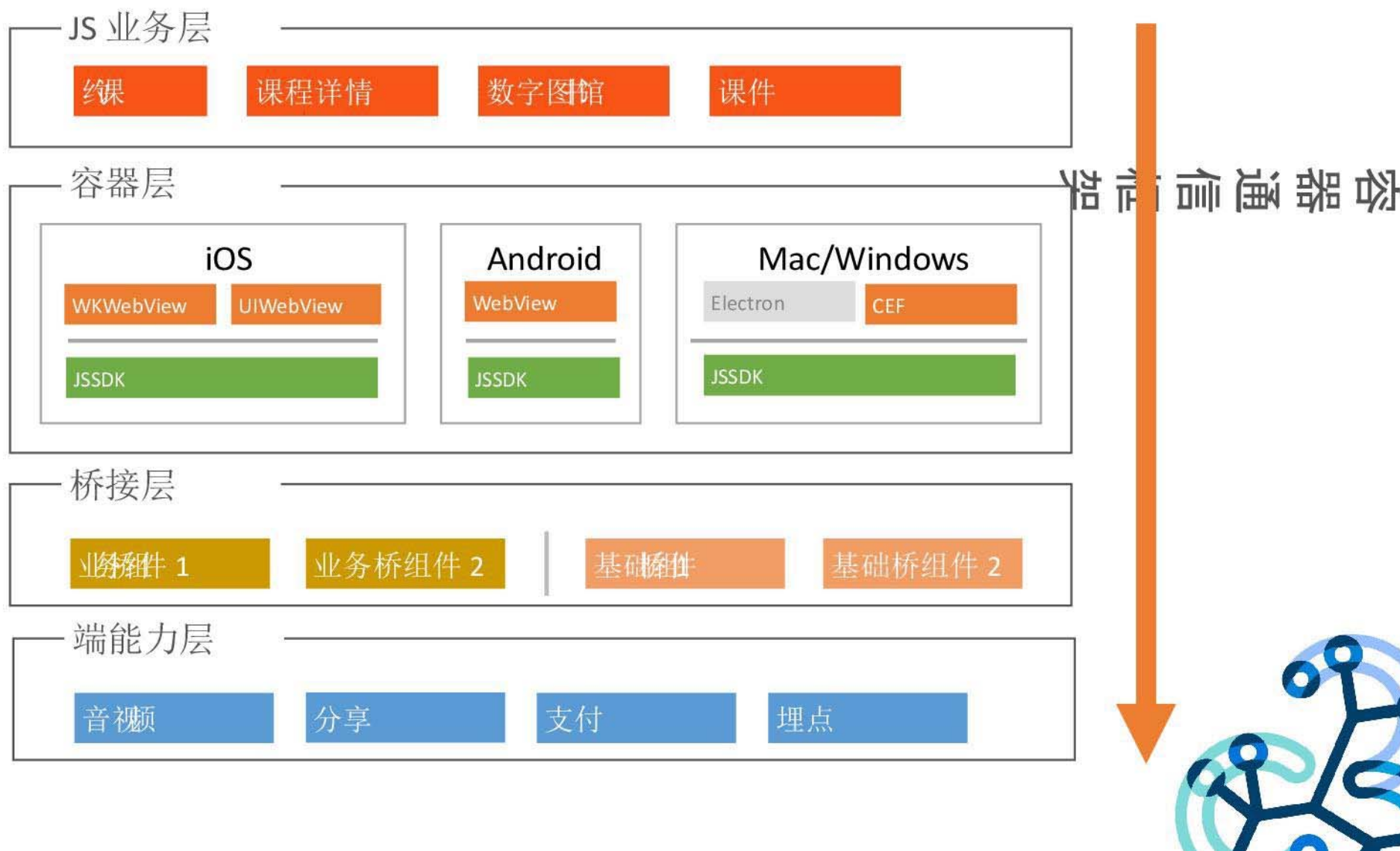
CASE 2: ? 后作加慢? 流程易中断? 如何北? ? ?

从 UI 和性能两方面打造接近于原生页面的体验

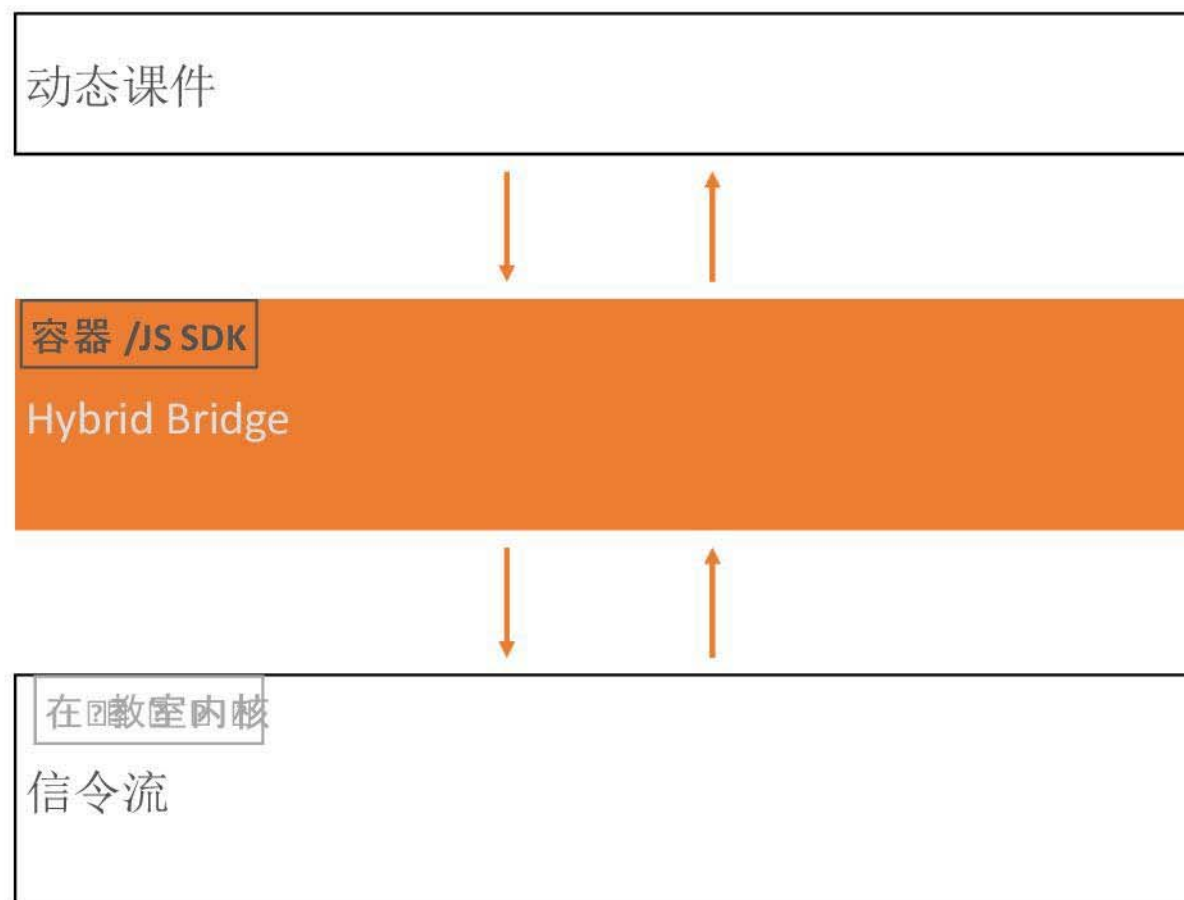
o o o



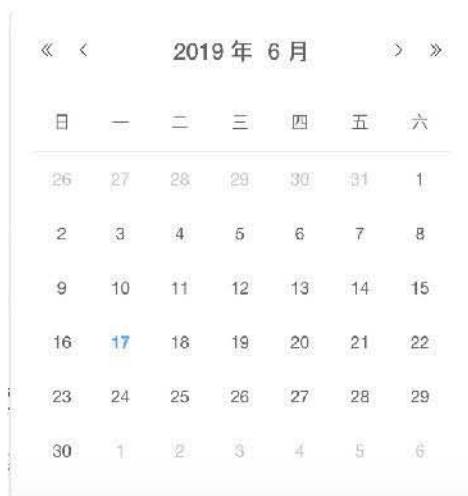
VIPKID 平台化方案—构建统一通信框架



VIPKID 平台化方案—构建统一通信框架

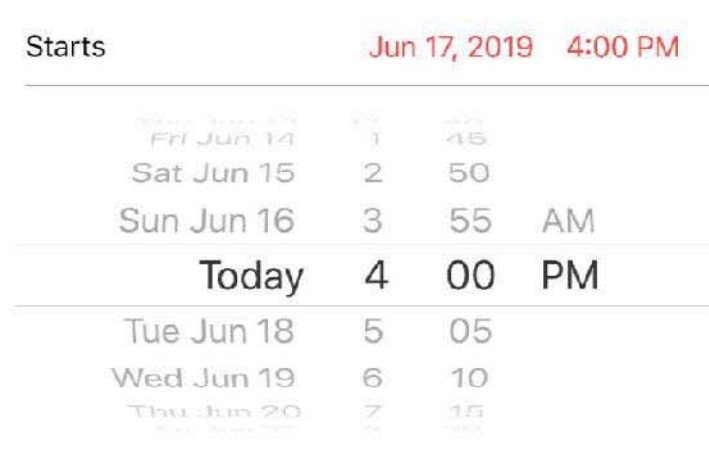


VIPKID 平台化方案—UI 体验



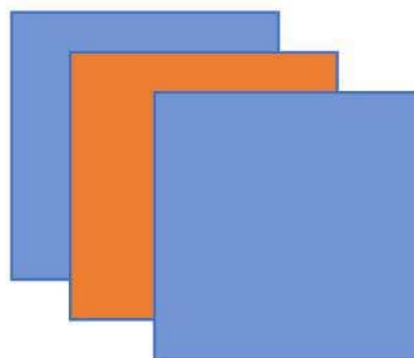
H5

图式不一致



iOS Native

UI 层带来交互问题



Native UI



WebView UI



VIPKID 平台化方案— UI 体验

构建与原生页面一致的 UI 风格

UI 桥组件

NavigationBar

TabBar

Dialog

Loading

DatePicker

Toast

Others...



VIPKID 平台化方案—加载提速

系统资源预热

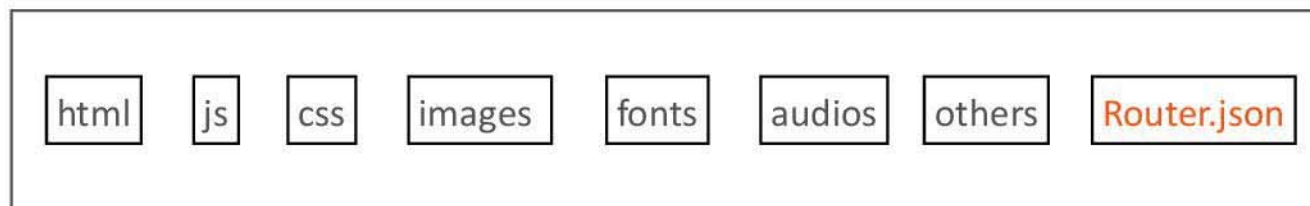


重要页面提前加载

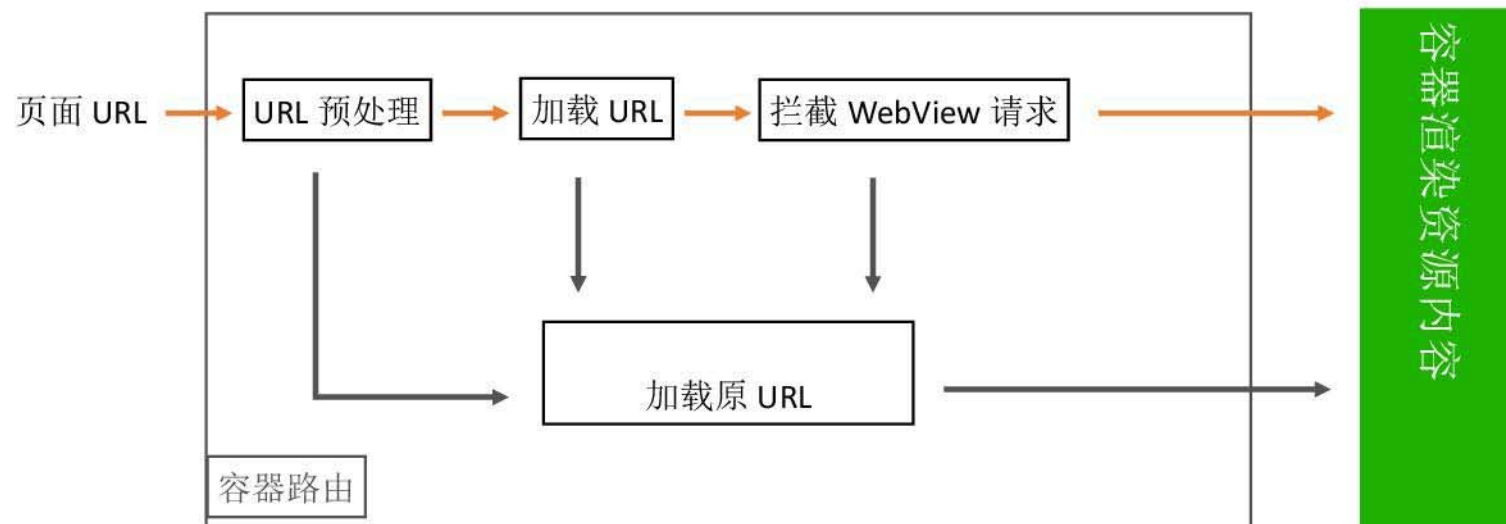


VIPKID 平台化方案-页面离线化

离线包



页面加载



VIPKID 平台化方案— WKWebView Tips

面及源加方式??

file:/// 协议加载页面

1. 无请求拦截
2. 会存在很多跨域问题

NSURLProtocol

1. 配合私有 API 拦截
2. POST 请求 Body 丢失

WKURLSchemeHandler

1. 自定义 scheme 请求拦截
2. POST 请求 Body 丢失

Cookie

问题

1. 同步有延迟
2. WKHTTPCookieStore iOS11 才能用

解决方案

HookCookie.js

AJAX ? 求跨域
POST Body ? 失

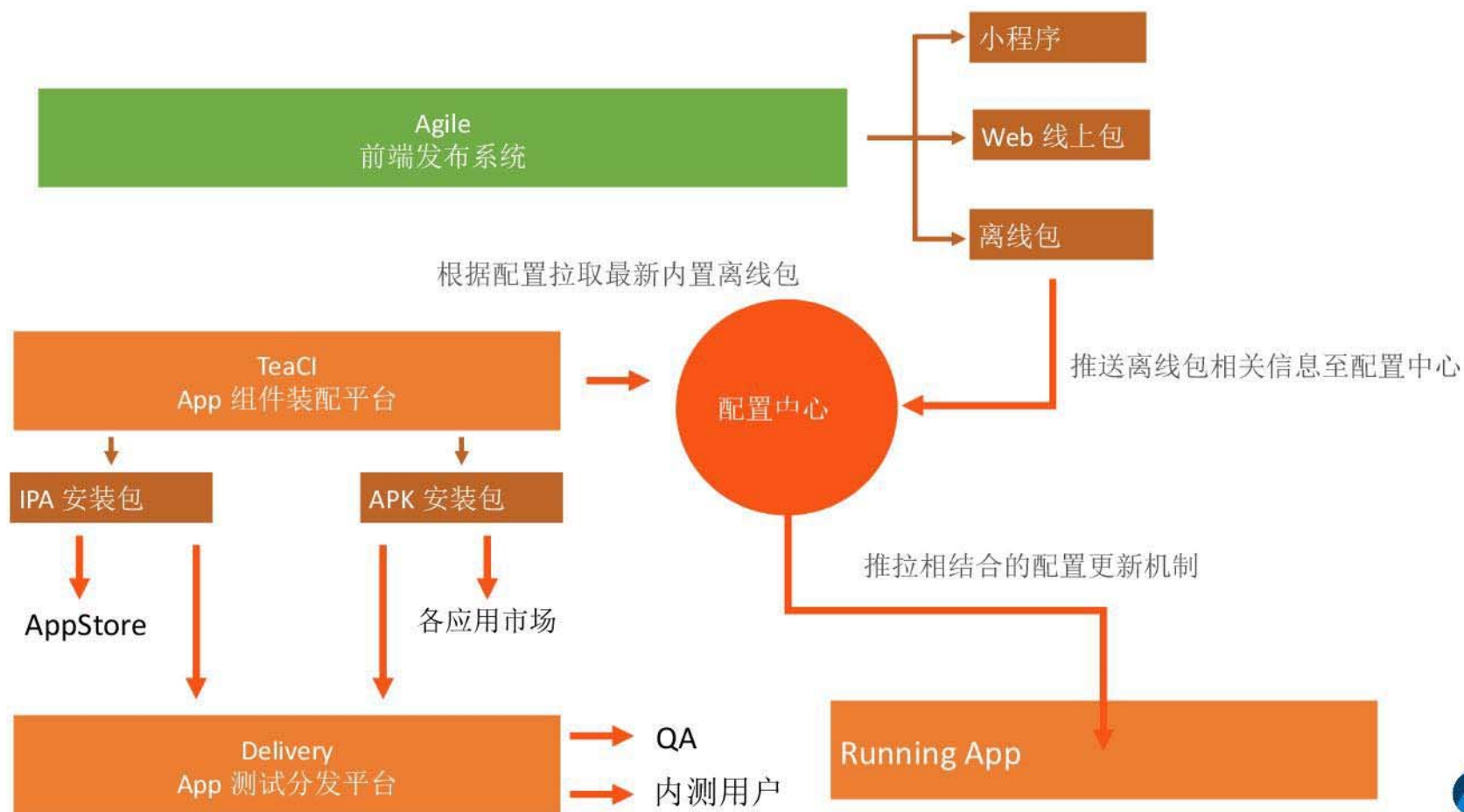
HookXHR.js

Hybrid 桥接口

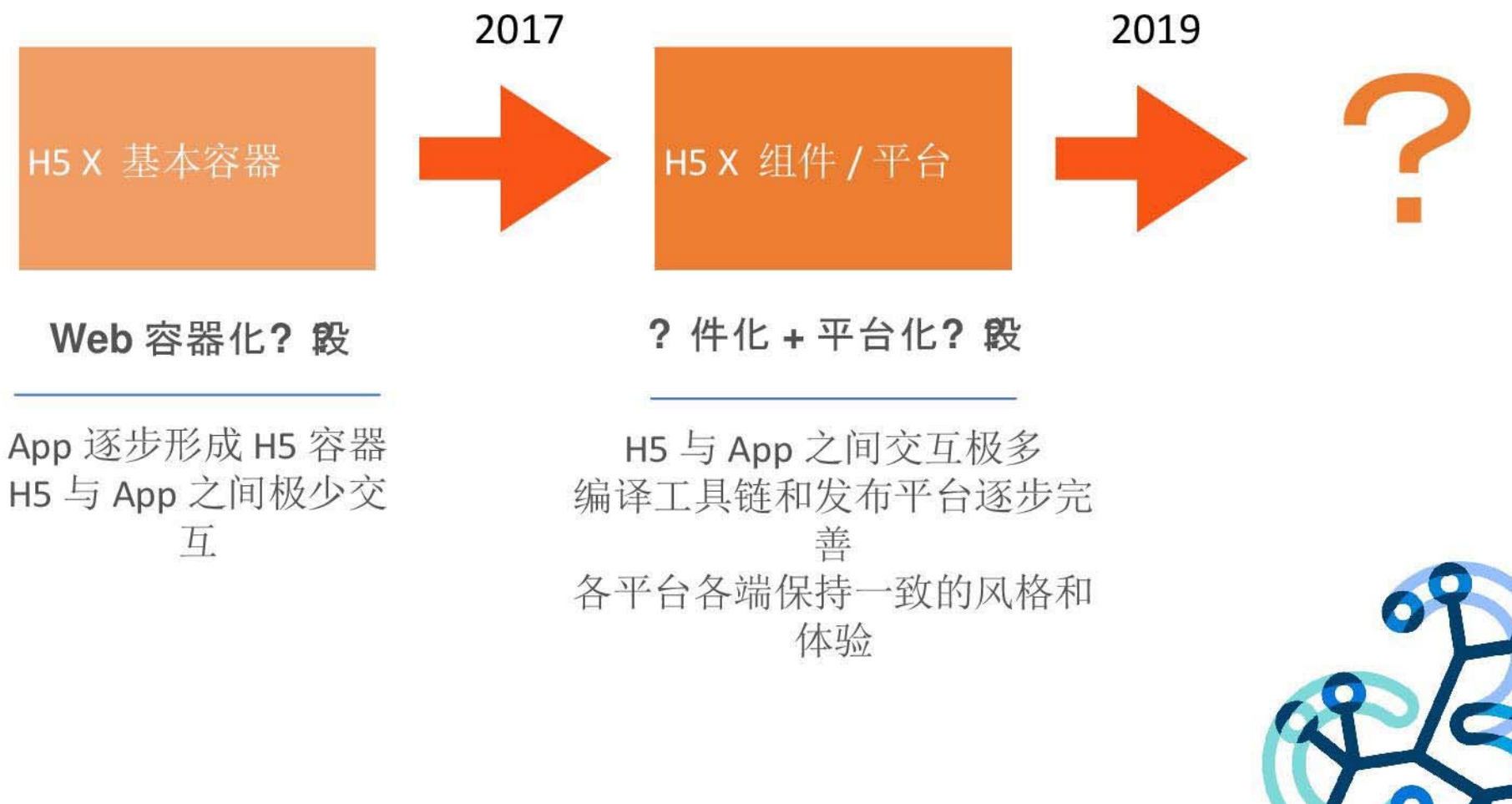
Native Network Module



VIPKID 平台化方案-工程化建设



VIPKID 应用层跨平台方案—总结



| VIPKID 应用层跨平台方案—展望

更好的性能

复杂页面频繁的 JS 通信依然存在性能瓶颈，在低端设备上表现尤其明显

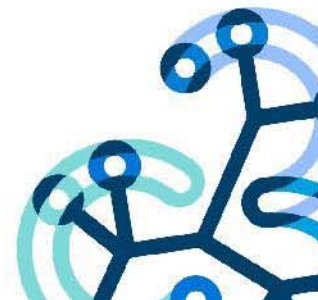
更强的生产力

多端共用一份 UI 设计稿，能否让多端也共用一份 UI 代码呢？
让原生工程师能更专注于客户端的业务逻辑以及性能表现

更好的开发体验

配套开发、调试工具
热加载

Try Flutter!



| Flutter— 机遇与挑战并存

机遇

iOS、Android 及 Web 使用同一套 codebase
声明式 UI 带来新的开发体验
新的平台，建立技术影响力

挑战

与原生代码的混编，工程化建设
包大小
可能面临的未知问题

