

# 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



**TOP100Summit**

全球软件案例研究峰会

时间：11月15~17日

地点：北京国际会议中心

100个年度最值得学习案例

**MPD工作坊（深圳站）**

时间：9月21~22日

地点：深圳博林圣海伦酒店

20个3小时大时段沙盘课程

Make Professional  
mpd

100

Make Professional  
mpdMake Professional  
mpd**MPD工作坊（北京站）**

时间：7月06~07日

地点：北京国家会议中心

20个3小时大时段沙盘课程

**MPD工作坊（上海站）**

时间：10月26~27日

地点：上海

20个3小时大时段沙盘课程

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**或不同**硬件架构**的电脑上工作



## | 如何实现跨平台?

### 跨平台语言

“一次编写, 到处运行” — Java

“一次编写, 到处编译” — C++

### 跨平台中间件

Socket、OpenGL

Asio、CEF

### 跨平台开发套件

Qt、Electron

React Native、Flutter



实际工作中该怎么选择



## | 跨平台技术选型思路



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在线教育场景概览

### 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

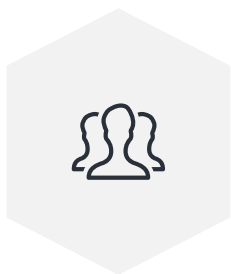
### VIPKID 英语

约课、监课、回放、学习报告  
Android、iOS

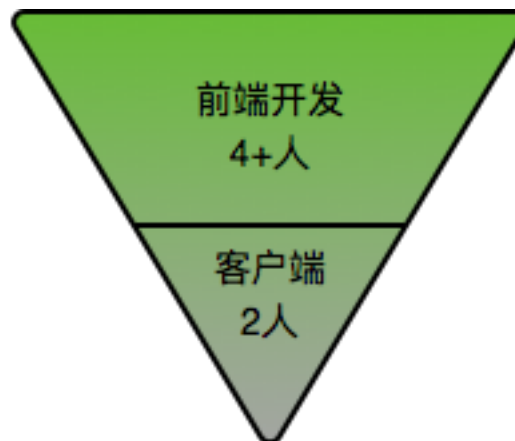




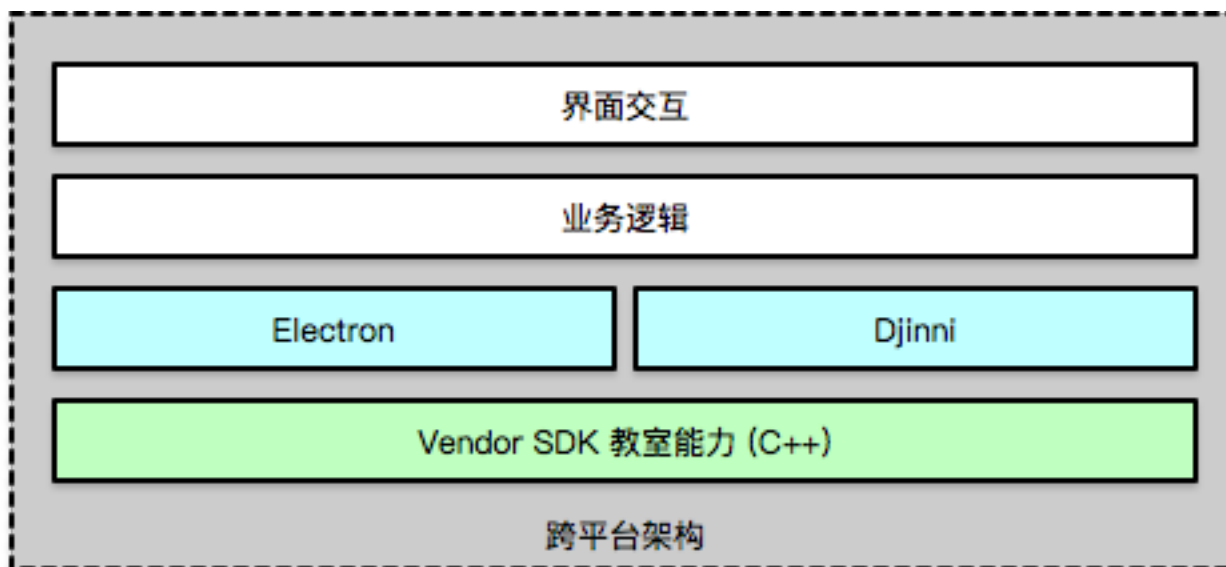
## | 人员结构评估



在线教室初期  
快速上线



## | 跨平台方案




 + Djinni  
**Electron**



## | Djinni /genie/

### 自动生成跨语言桥接代码

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



## | Djinni /genie/

```
photo = record {
    # server id of the photo
    id: i64;

    filename: string;
}

photo_model = interface +c +j +o {
    # returns the currently sync'd photos
    get_photos(): list<photo>;
}
```



## | 上线后遇到的问题

Crash率飙升

Electron贡献了大部分崩溃

多Vendor接入

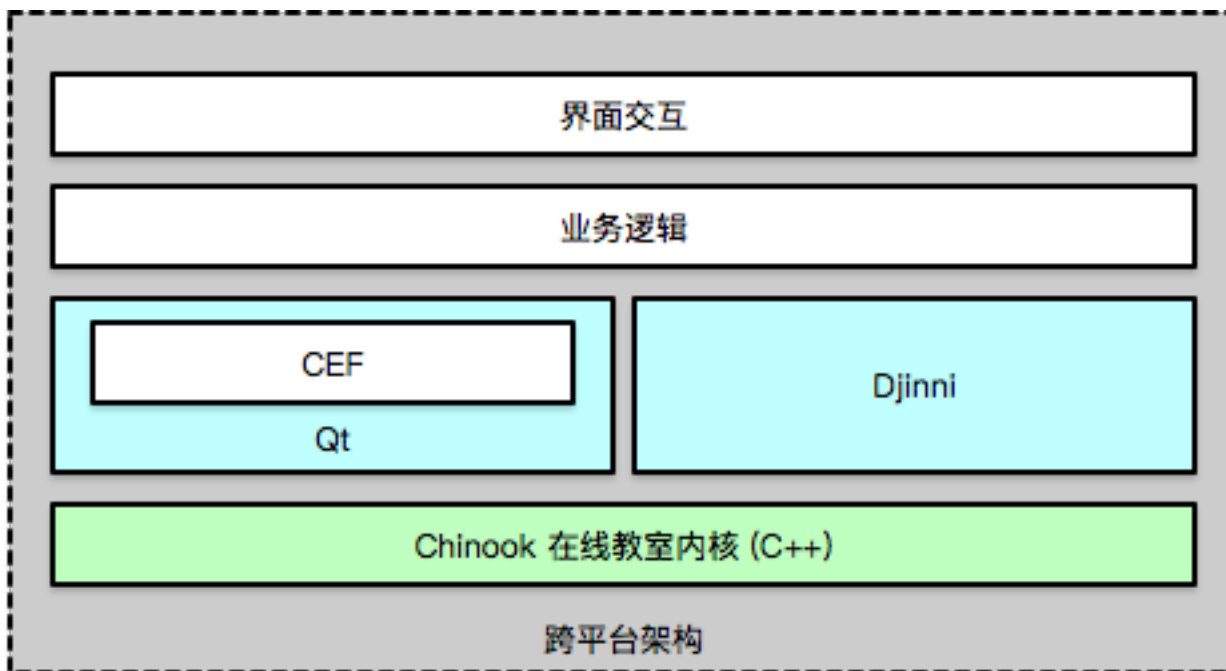
解除音视频能力单点依赖


教室场景丰富

更多的平台能力依赖接入



## 跨平台方案改进



Qt +  + Djinni



## 如何设计跨平台内核?



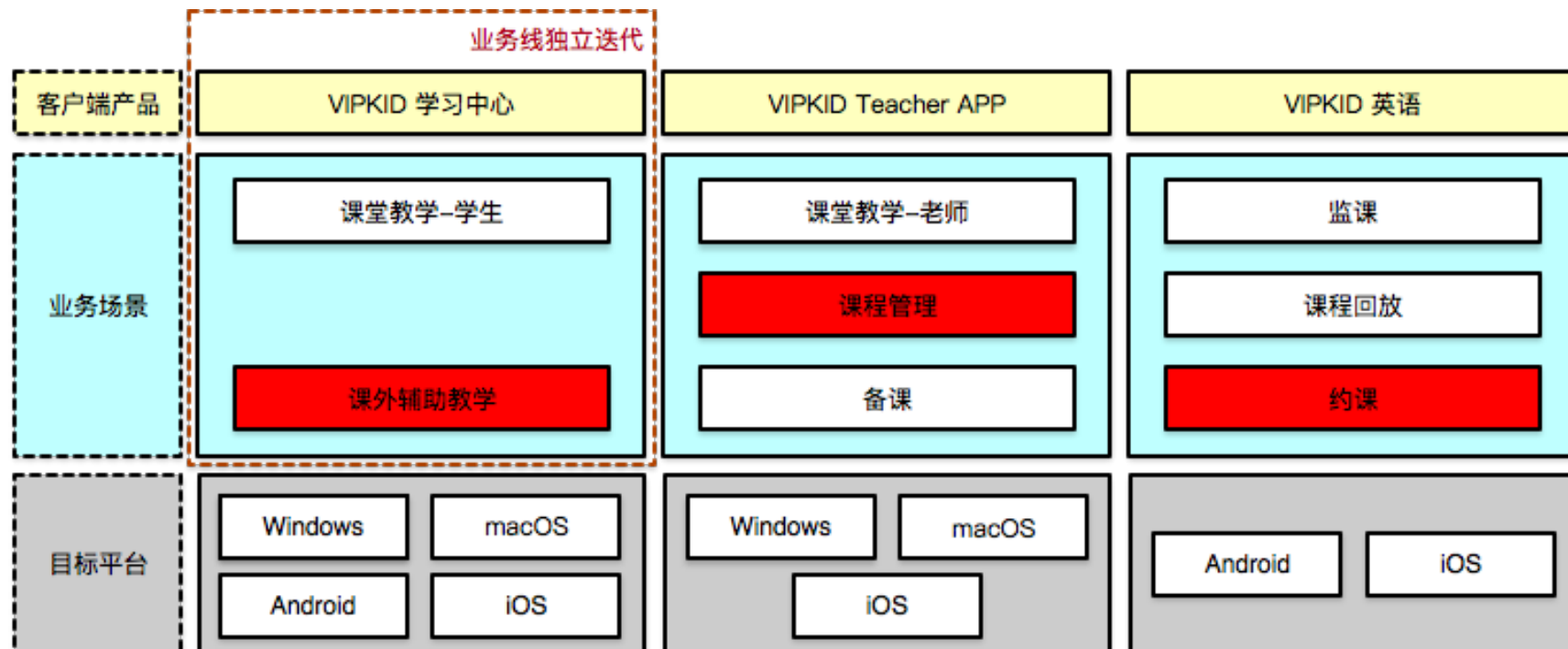
## | 如何设计跨平台内核

适度选取、自顶向下  
分层和抽象





## 业务场景分层抽象



 Web 技术实现



## | 业务场景分层抽象

### 业务组件层

客户端产品功能快速迭代开发

### 业务中台层

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



## 业务场景分层抽象



## | 业务中台分层抽象

### 在线教室基础逻辑

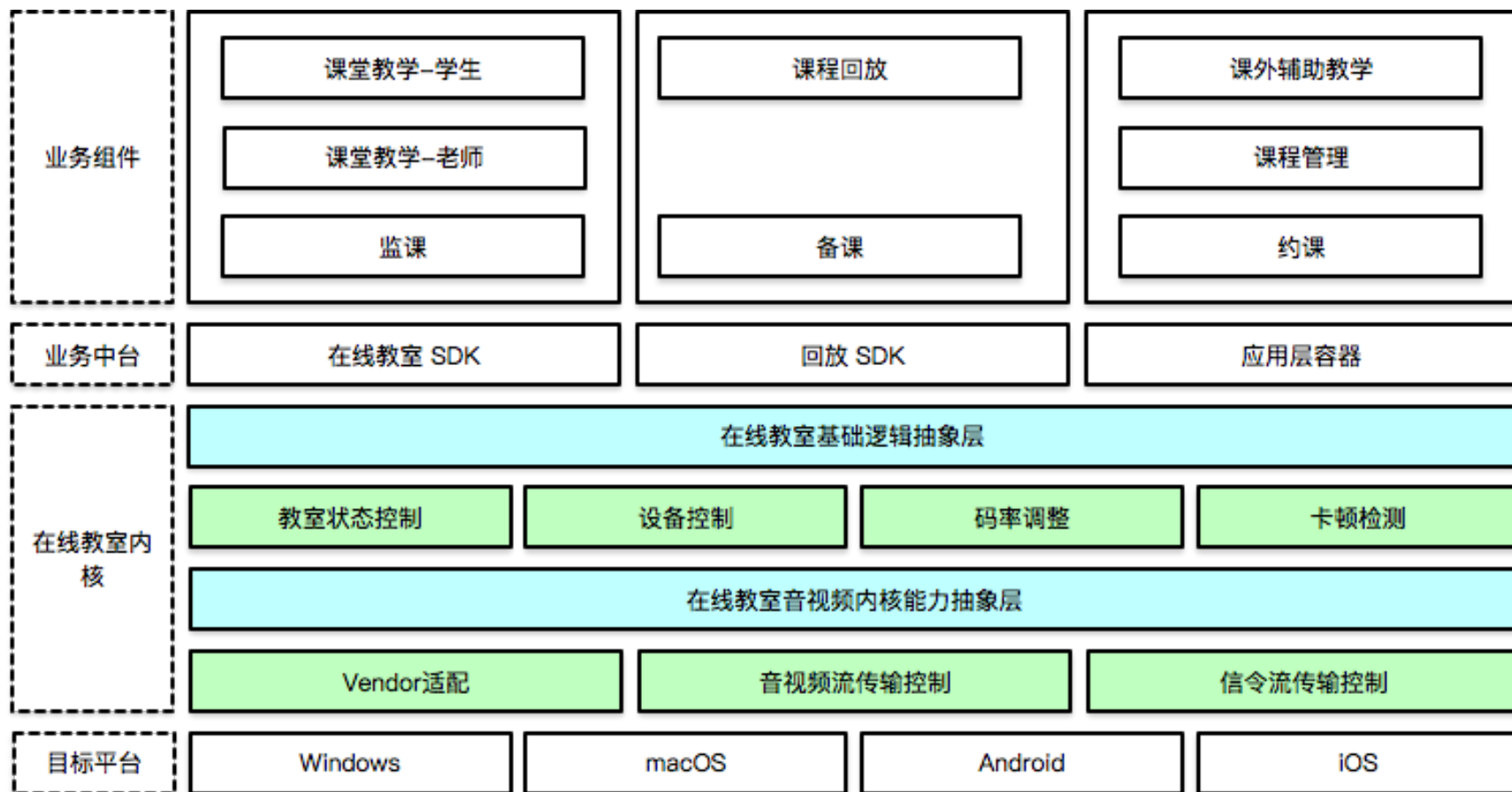
不同平台和网络环境下构建一个稳定、高质量的在线教室最小单元

### 在线教室音视频内核

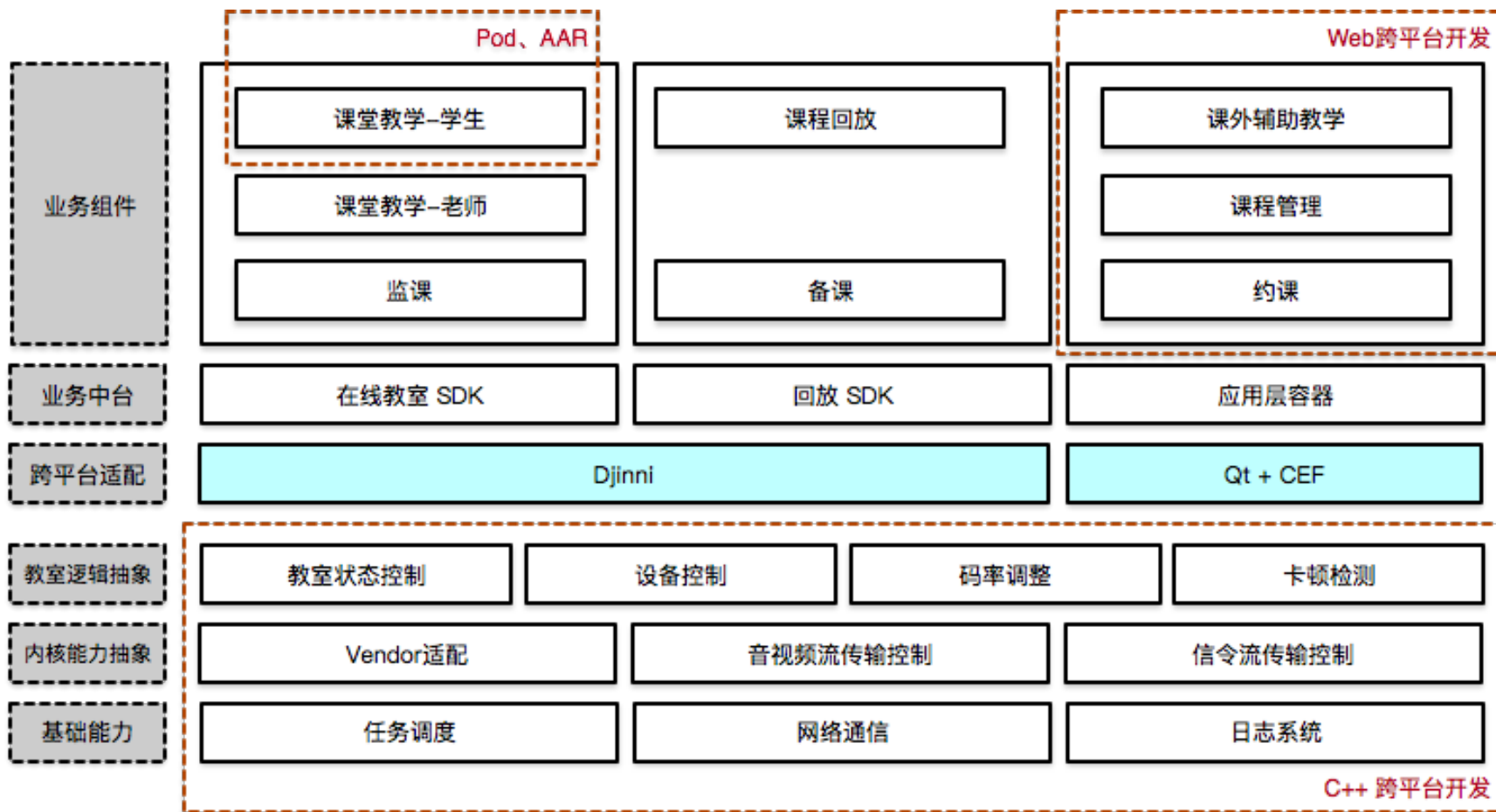
抽象音视频、信令传输能力



## 业务中台分层抽象



## VIPKID在线教室跨平台整体架构



## VIPKID应用层跨平台方案



## 应用层跨平台的方案

方案	传统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	稳定发展了一段时间(RN>Weex)	刚起步，但是发展迅速
渲染性能	低	高于WebView(取决于JSC的效率)	高(自建的平台统一渲染引擎)
上手成本	低	中	高
发版成本	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

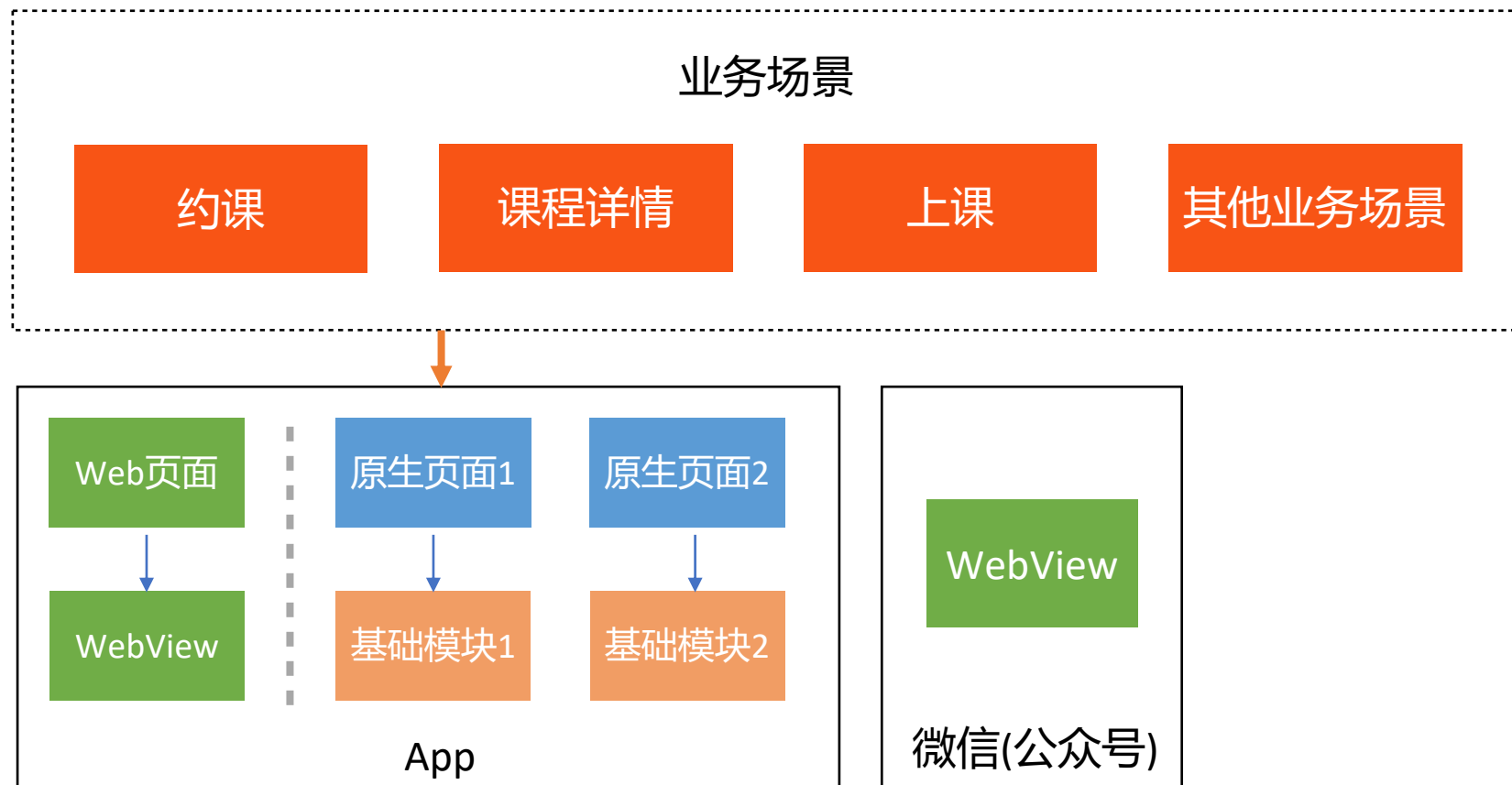




## 应用层跨平台的方案—业务决定架构



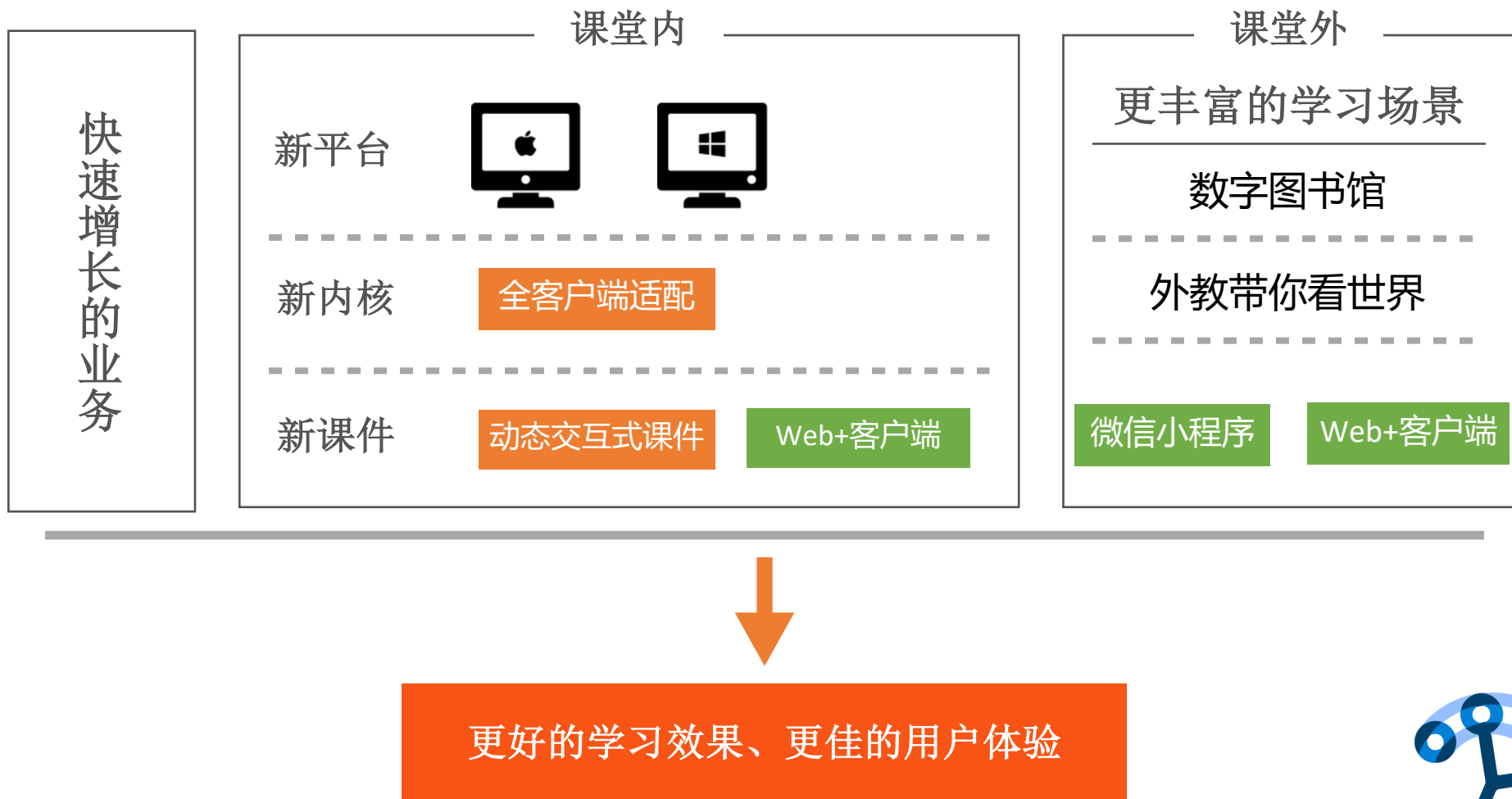
## VIPKID跨平台方案—应用容器化



Before 2017



## VIPKID容器化方案—业务带来的挑战



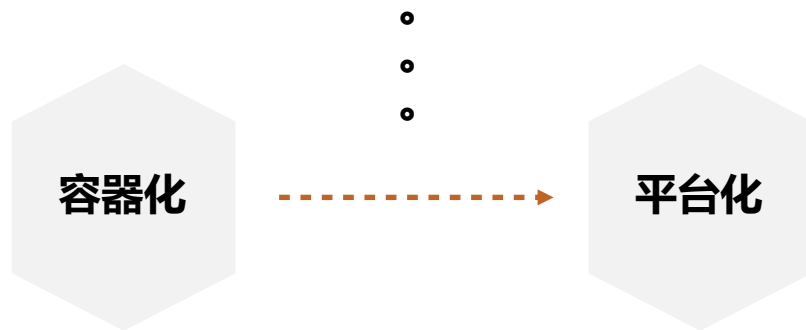
## VIPKID跨平台方案演进

### CASE 1: 如何将师生在动态课件中的操作同步给对方

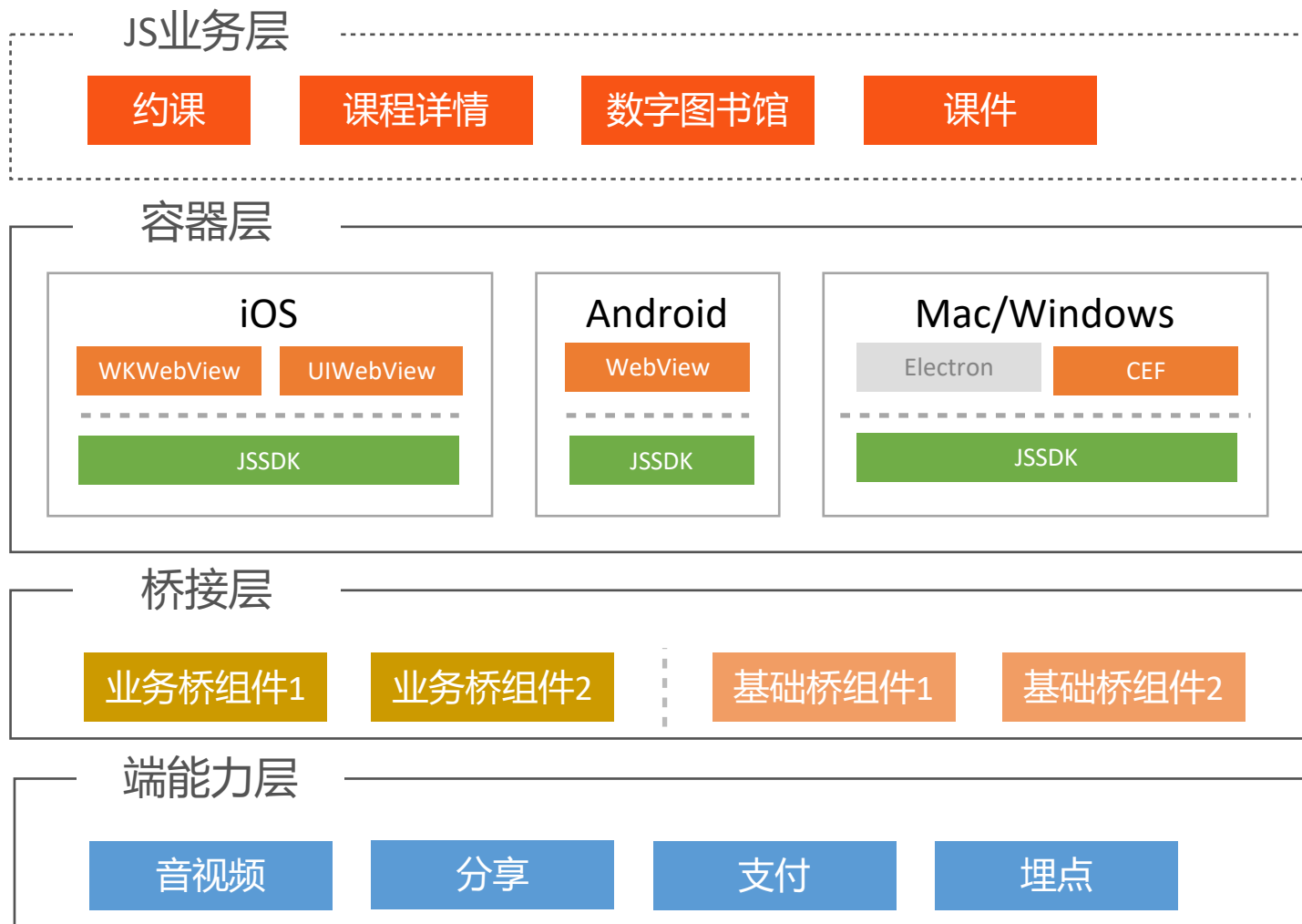
构建容器统一通信框架

### CASE 2: 课后作业加载慢、流程易中断，如何优化

从UI和性能两方面打造接近于原生页面的体验



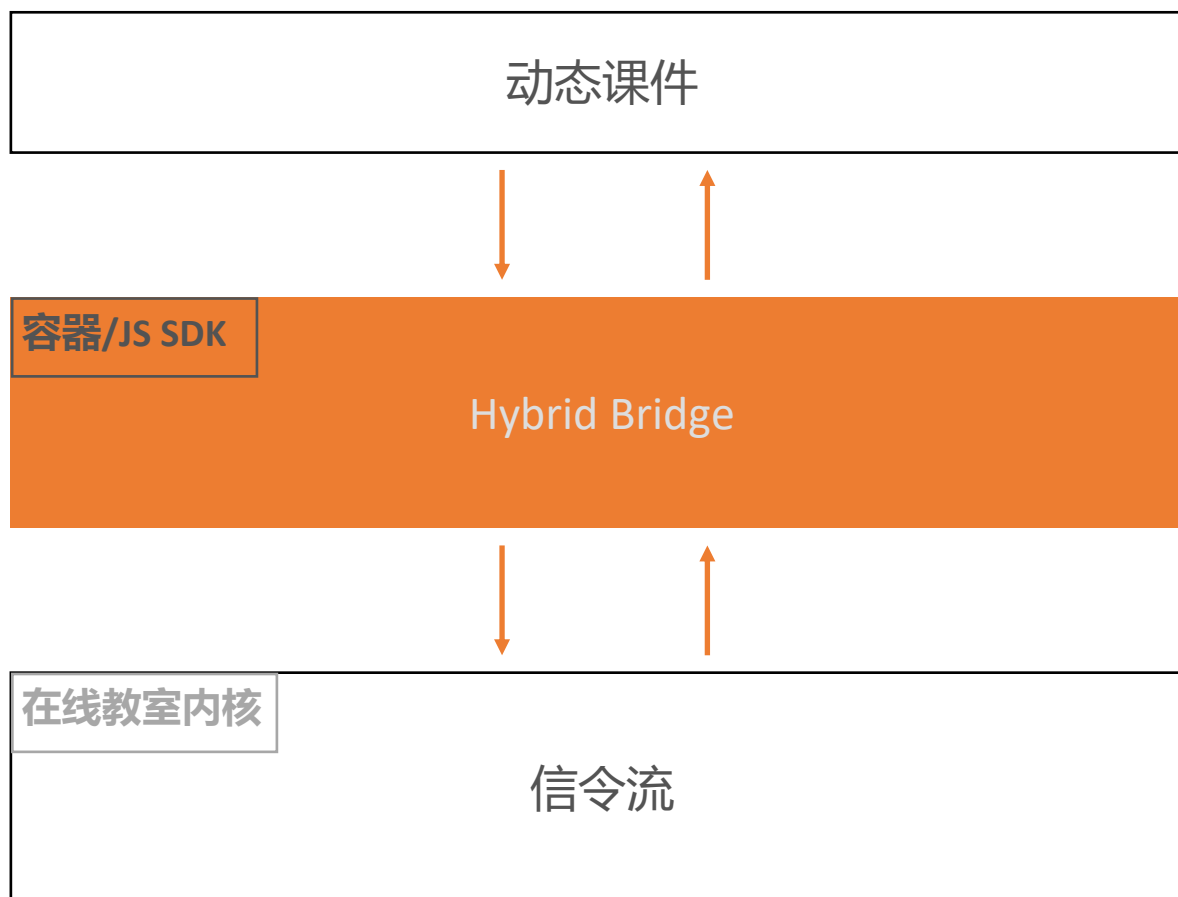
## VIPKID平台化方案—构建统一通信框架



容器通信框架



## VIPKID平台化方案—构建统一通信框架

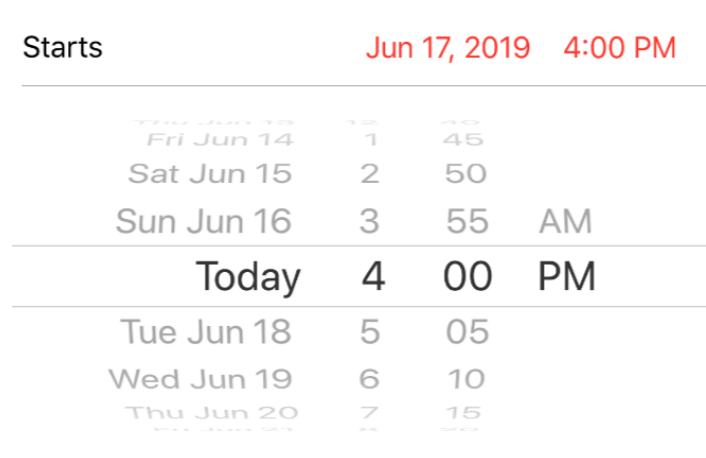


## VIPKID平台化方案—UI体验



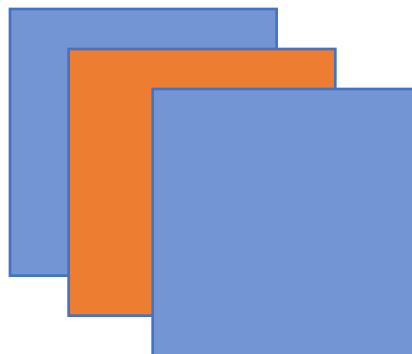
H5

样式不一致



iOS Native

UI层级带来交互问题



Native UI



WebView UI



## | VIPKID平台化方案—UI体验





## VIPKID平台化方案—加载提速

### 系统资源预热



### 重要页面提前加载

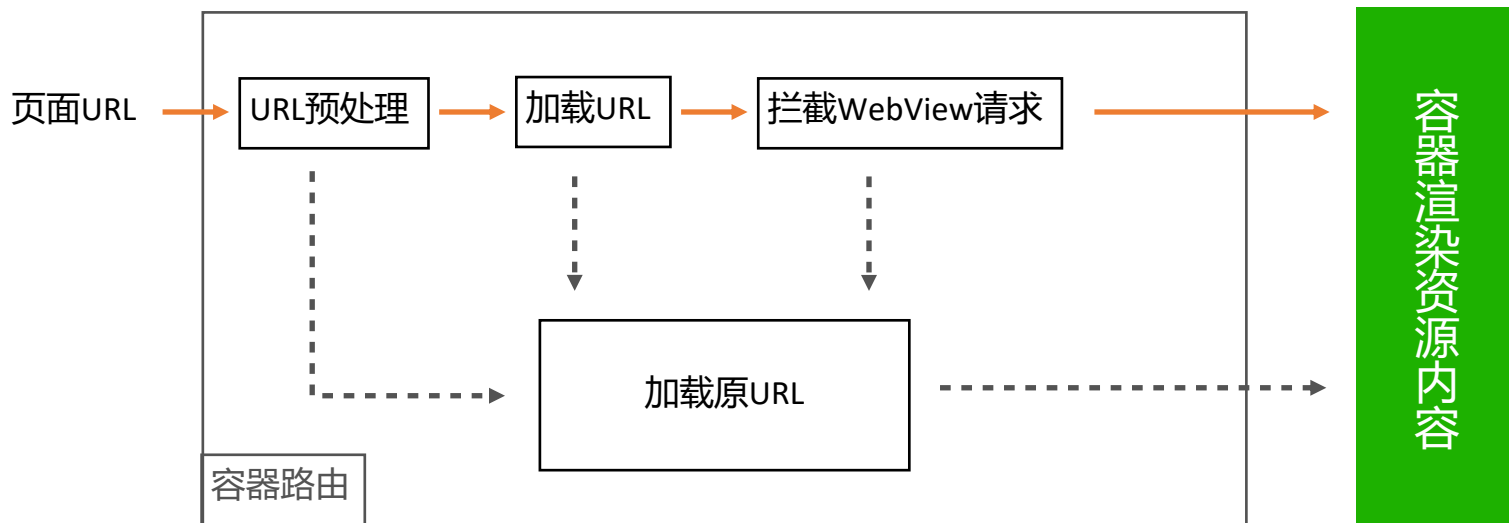


## VIPKID平台化方案—页面离线化

### 离线包



### 页面加载



## VIPKID平台化方案—WKWebView Tips

### 页面及资源加载方式

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请求跨域 POST Body丢失

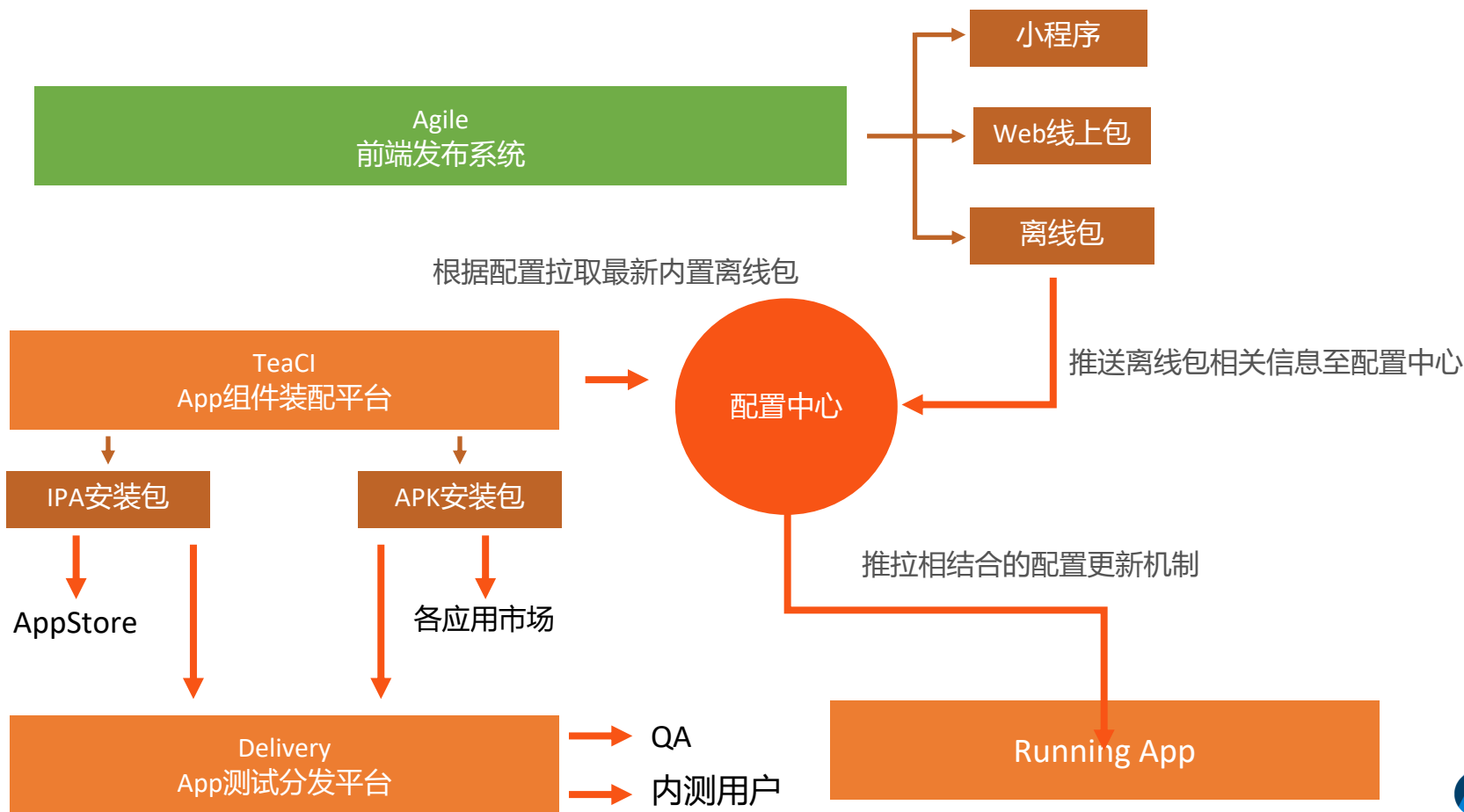
HookXHR.js

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





欢迎关注msup微信公众账号

关注大会微信公共账号，及时了解大会动态、

日程及每日更新的案例！

