



微信小程序云端解决方案

探索之路

探索之路



小程序特性思考



挑战与尝试



整合



前端工程师如何理解小程序？

index.wxml x

```
1 <view class="root">
2   <view class="line">
3     <text>综合示例</text>
4     <view bindtap="openChat" class="demo">
5       三木聊天室
6       <view class="link">Demo体验 </view>
7     </view>
8   </view>
```

index.wxss x

```
1 .root{
2   background: #efeff4;
3   position: absolute;
4   top: 0;
5   left: 0;
6   right: 0;
7   bottom: 0;
8 }
```

chat.js x

```
36 // 声明聊天室页面
37 Page({
38
39   /**
40    * 聊天室使用到的数据，主要是消息集合以及当前输入框的文本
41    */
42   data: {
43     messages: [],
44     inputContent: '大家好啊',
45     lastMessageId: 'none',
46   },
```

CSS



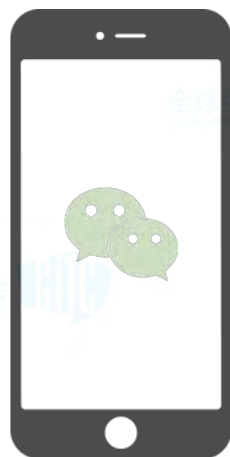
小程序是 H5 应用吗？

不是



小程序不是 H5 应用

- 无法使用 DOM 接口
 - HTML5 生态中基于 DOM 的库全部无法使用（如 jQuery）
- 并非 URL 访问，没有域名概念
 - 不存在跨域问题
 - 无 Cookie 存储
- 小程序更偏向 CS 架构



小程序和传统 CS 架构的
区别在哪儿？



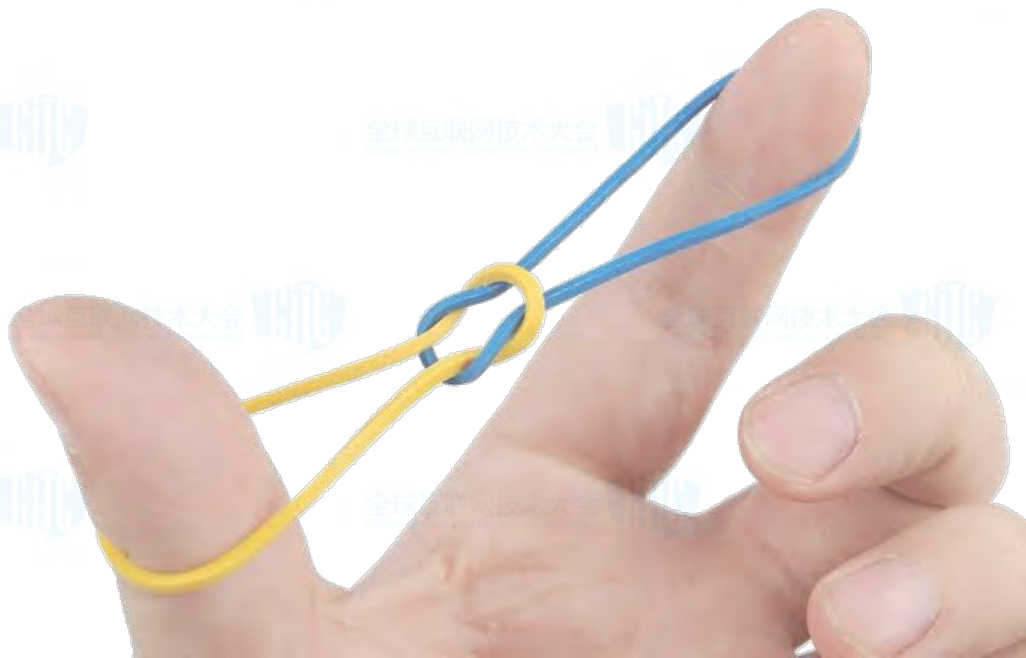
网络和续航

- 网络环境复杂
- 过于消耗资源会导致续航下降
- 小程序对网络优化和资源优化提出要求



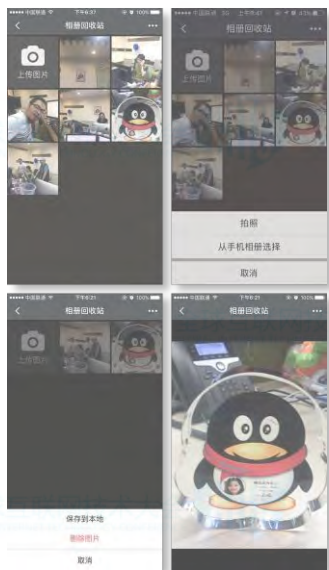
伸缩能力

- 传播和社交特性可能会带来业务的突然增长
- 小程序对后台架构的伸缩能力有高要求





我们进行的尝试



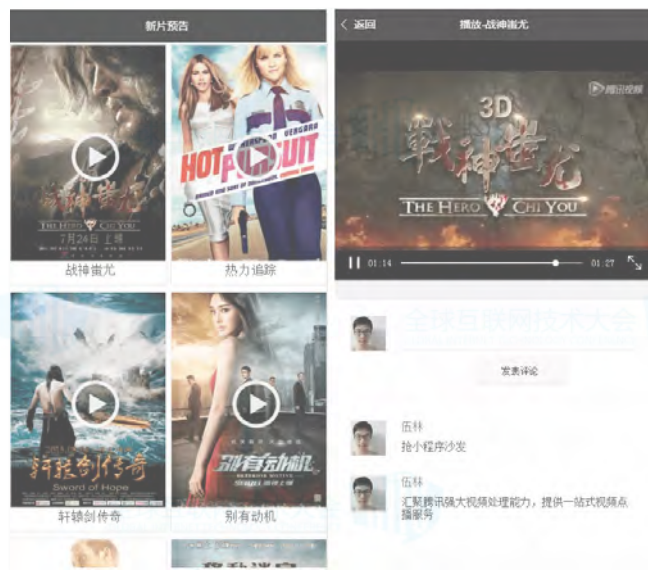
上传下载场景



会话管理服务



WebSocket 服务



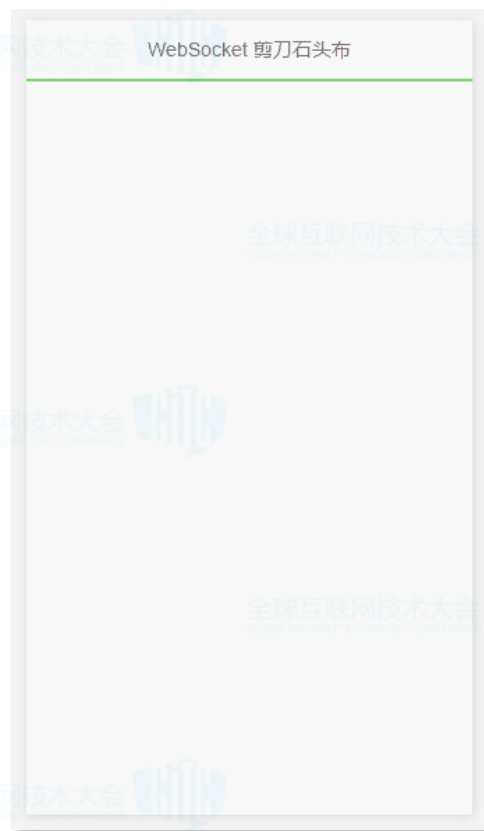
视频点播场景



挑战在哪儿



会话服务



WebSocket 服务



会话管理的挑战



- 小程序不支持 Cookie 传输，需要自己实现会话服务



会话管理目标

- 完成微信要求的鉴权流程，生成用户会话
- 利用会话确定每个请求对应哪个微信用户
- 安全性和扩展性要求



会话建立过程

会话管理客户端 SDK

会话管理服务器 SDK

小程序



1. 请求登录

(code, rawData, signature)

业务服务器



2. 换取会话

(code, appid, appsecret)

微信服务器



3. 返回会话

(session_key)

6. 返回会话

(session_id)

5. 建立并
保存会话

4. 验证签名



Redis



会话服务效果分析

- 以 Node SDK 的形式完成了功能性目标
- 其它后端语言不通用
- 小程序 appId 和 appSecret 安全性问题
- 缺少横向扩展能力

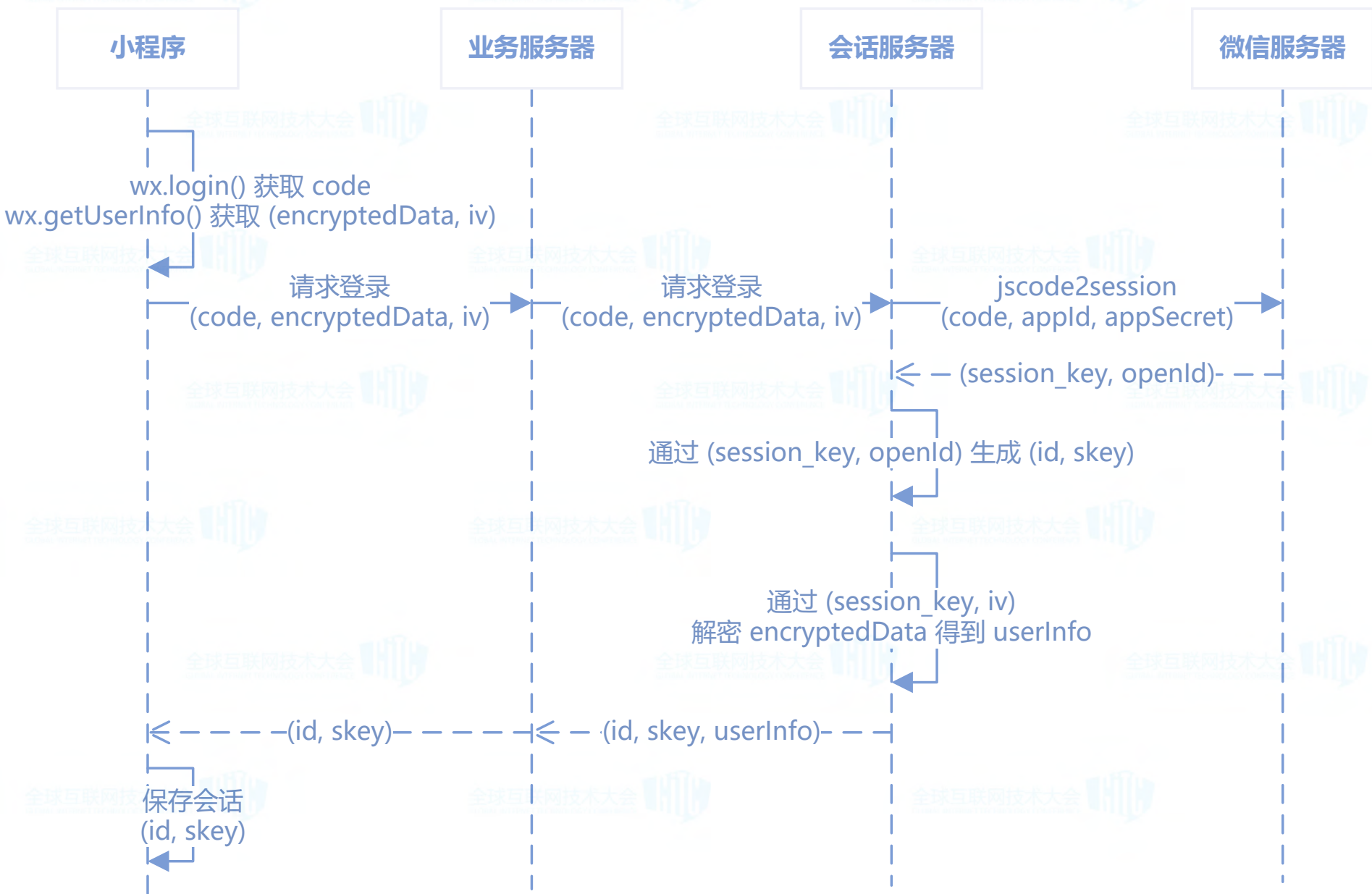


会话服务改进

- 独立会话管理服务器
- 提供多语言 SDK
- appId 和 appSecret 存放数据库

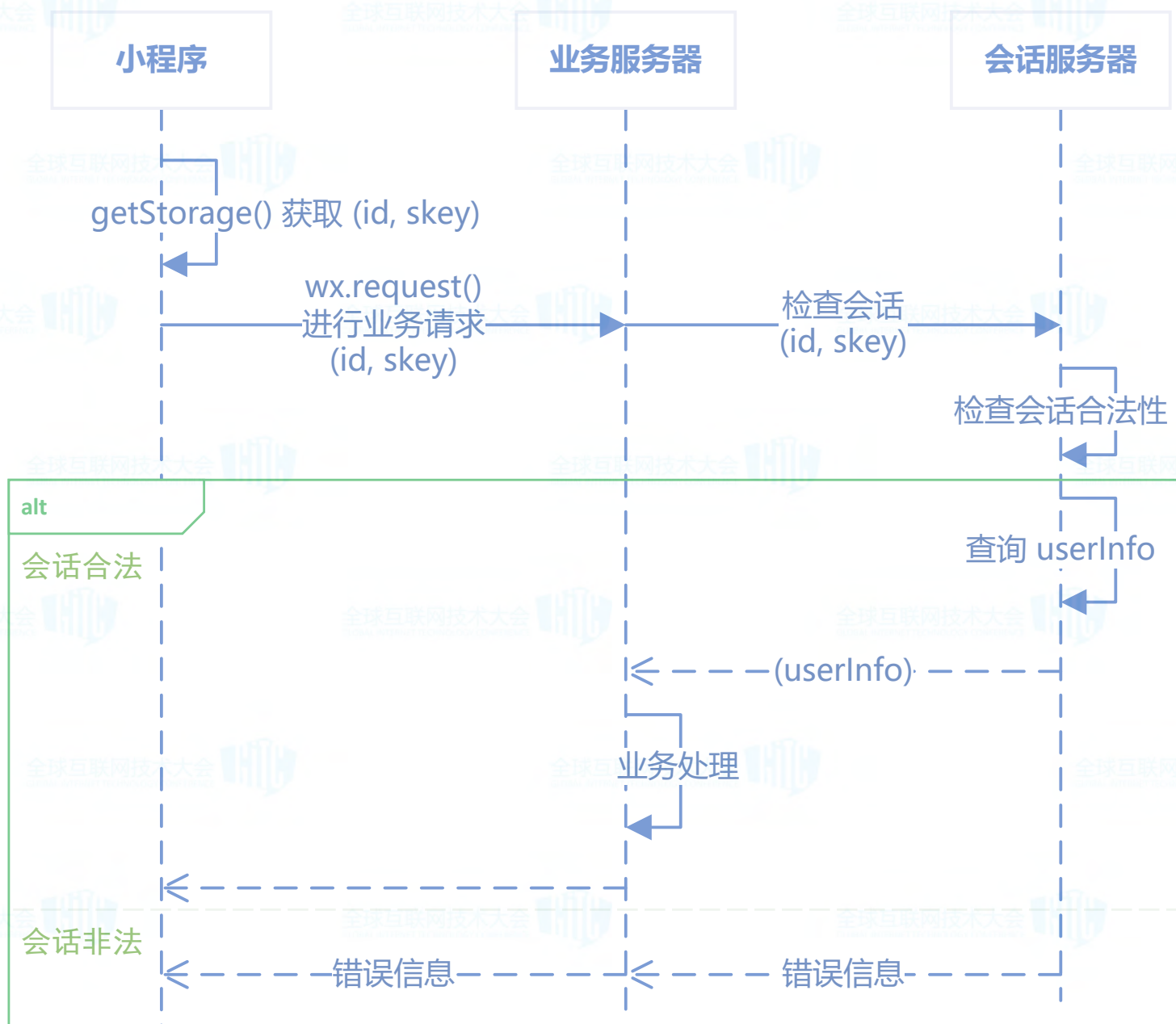


改进的会话建立流程





改进的会话检查流程





会话服务改进分析

- 流程和安全性完全符合微信鉴权要求
- 独立会话服务器的优势
 - 方便单独升级、扩展
 - 业务服务器 SDK 开发成本低，方便支持多语言



WebSocket 的挑战

WebSocket 剪刀石头布

- 小程序为何要支持 WebSocket ?



通信方式对比

传统连接

- 建立连接耗时长
- 需要建立多个连接
- 长连接方案只能串行传输
- 耗电体验差

WebSocket

- 仅需要建立一个连接
- 双向实时通信
- 省电体验好



一张图解释 WebSocket





WebSocket 的尝试

WebSocket 剪刀石头布

- 使用 Socket.IO 实现后端
- Socket.IO 客户端在小程序上无法运行
- 学习 Socket.IO 协议后实现了一个简版的客户端
- 实现剪刀石头布游戏逻辑



WebSocket 效果分析

- 验证了 WebSocket 的可用性
- 阉割版的 Socket.IO 客户端不可控
- 后端实现门槛高，且不好横向扩展

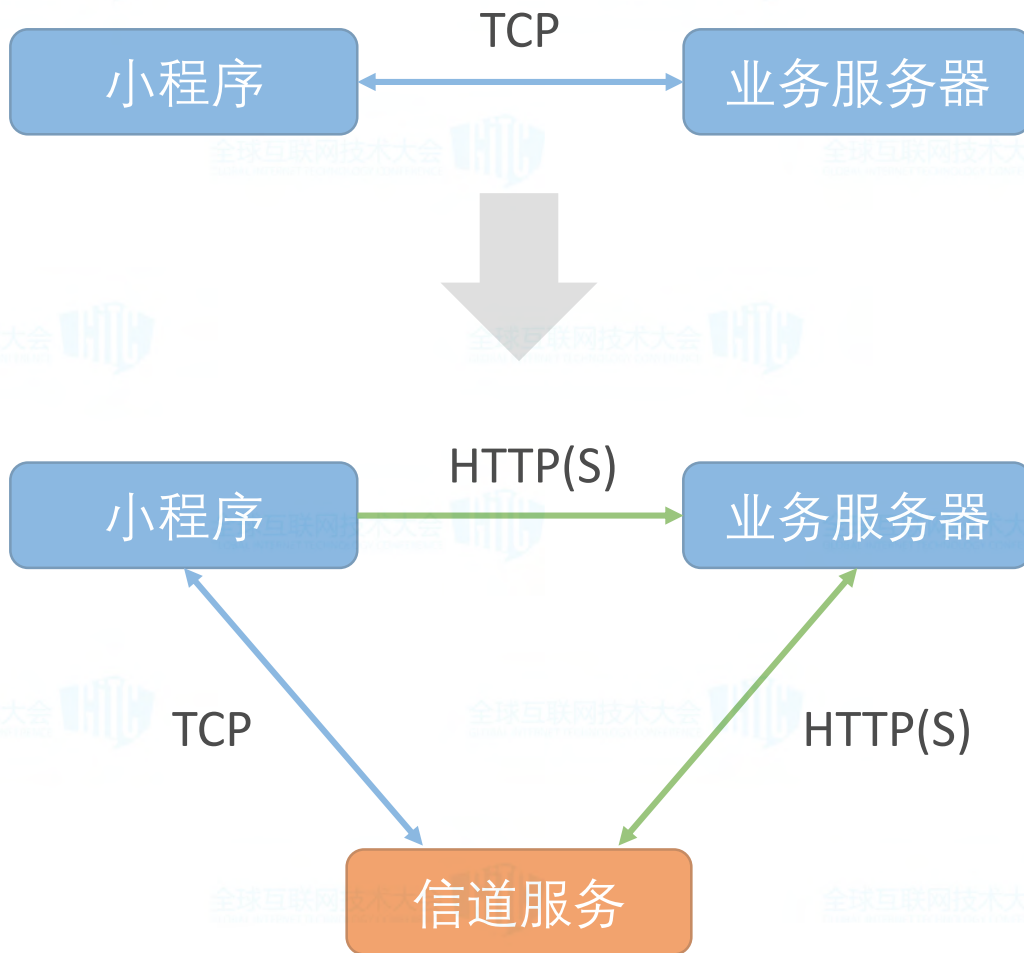


WebSocket 方案改进

- 使用 PaaS 服务支持 WebSocket 连接

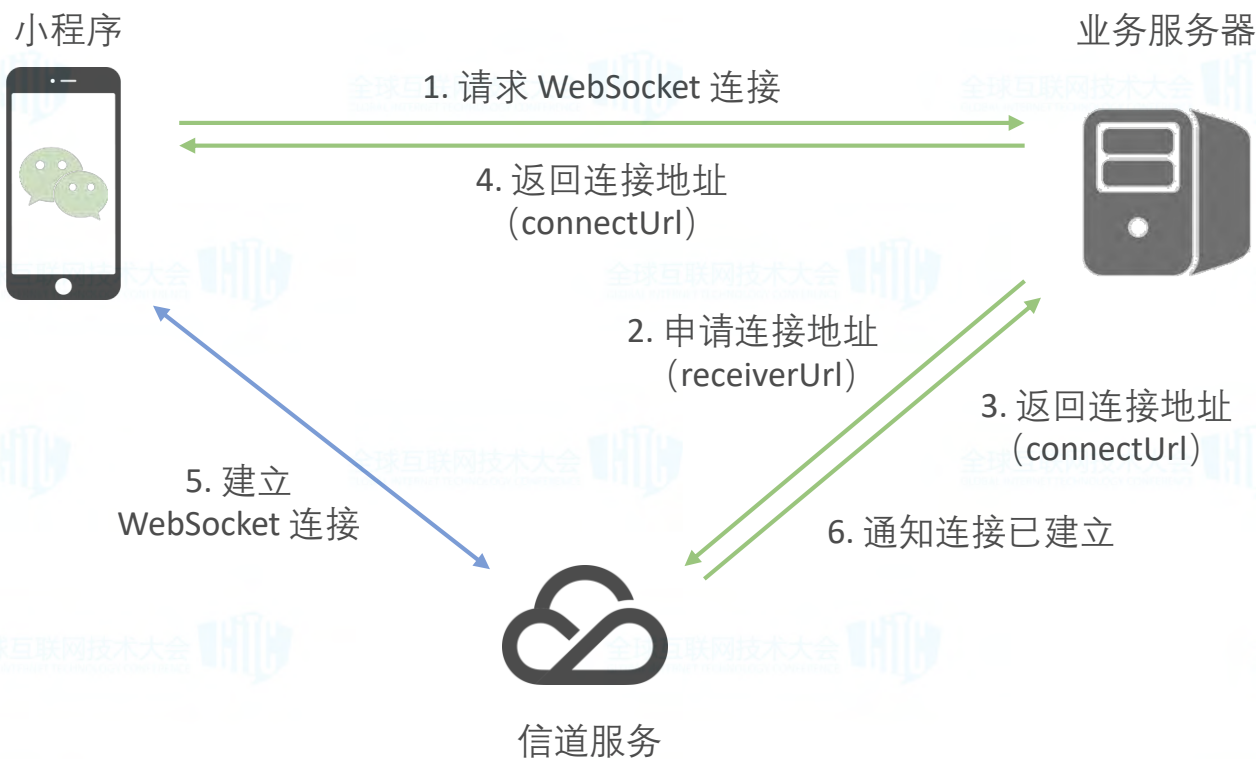


WebSocket 方案改进





使用 PaaS 服务建立 WS 连接





使用 PaaS 服务进行 WS 通信

小程序



业务服务器



a. 发送消息

d. 推送到客户端

c. 推送消息

b. 推送到业务服务器



信道服务



PaaS 信道服务的优势和局限

- 优势

- 平台提供扩展能力
- 业务服务器无需支持 WebSocket 协议
- 减轻服务器连接资源
- 稳定性和性能保障

- 局限

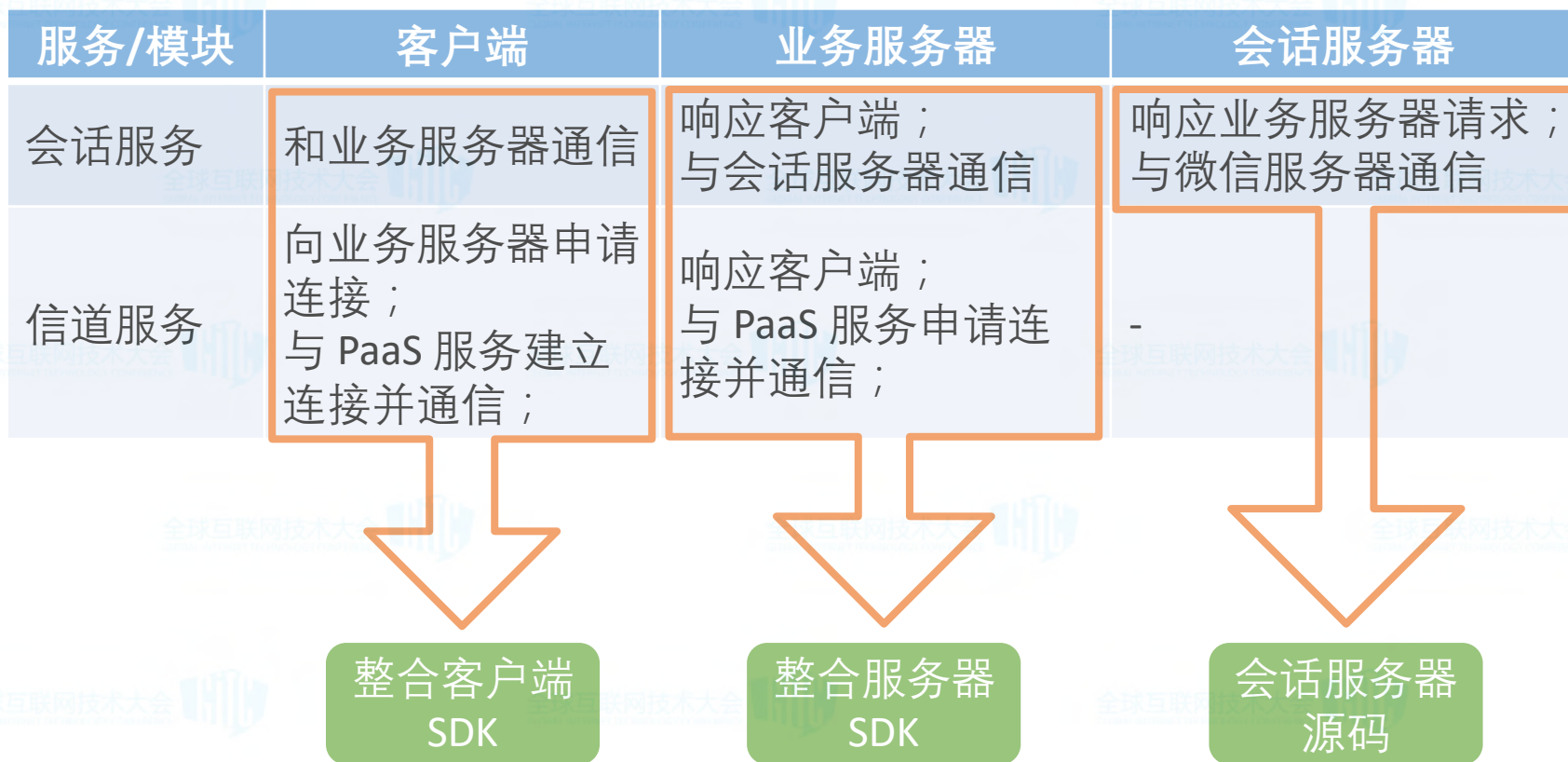
- 业务服务器和信道服务器间公网通信延迟
- 调试便利性不如传统连接方式



信道服务如何支持会话？



服务与模块关系



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

整合客户端
SDK

+

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

整合服务器
SDK

+

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

会话服务器
源码

Wafer

微信小程序全栈开发资源

Wafer

微信小程序全栈开发资源



<https://github.com/tencentyun/wafer>



Wafer 包含什么？

Wafer

信道服务

会话服务

两个服务



丰富文档

Client SDK

PHP SDK

C# SDK

Java SDK

Node SDK

Session Server

全栈资源

One More Thing



小程序开发者有多累？



小程序开发者



Wafer 做了什么？



小程序开发者

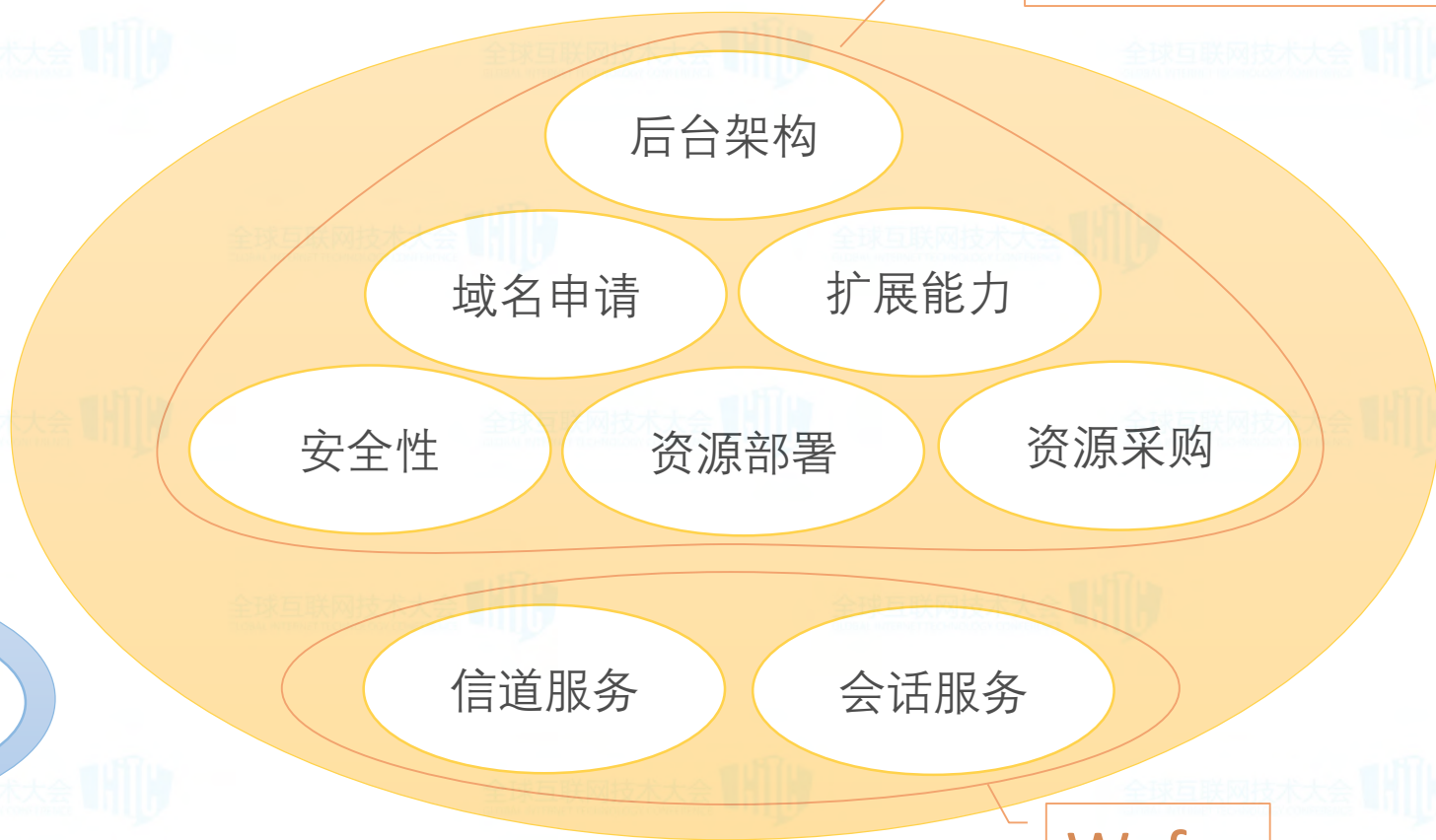


Wafer



Wafer + 一站式部署

一站式部署



Wafer

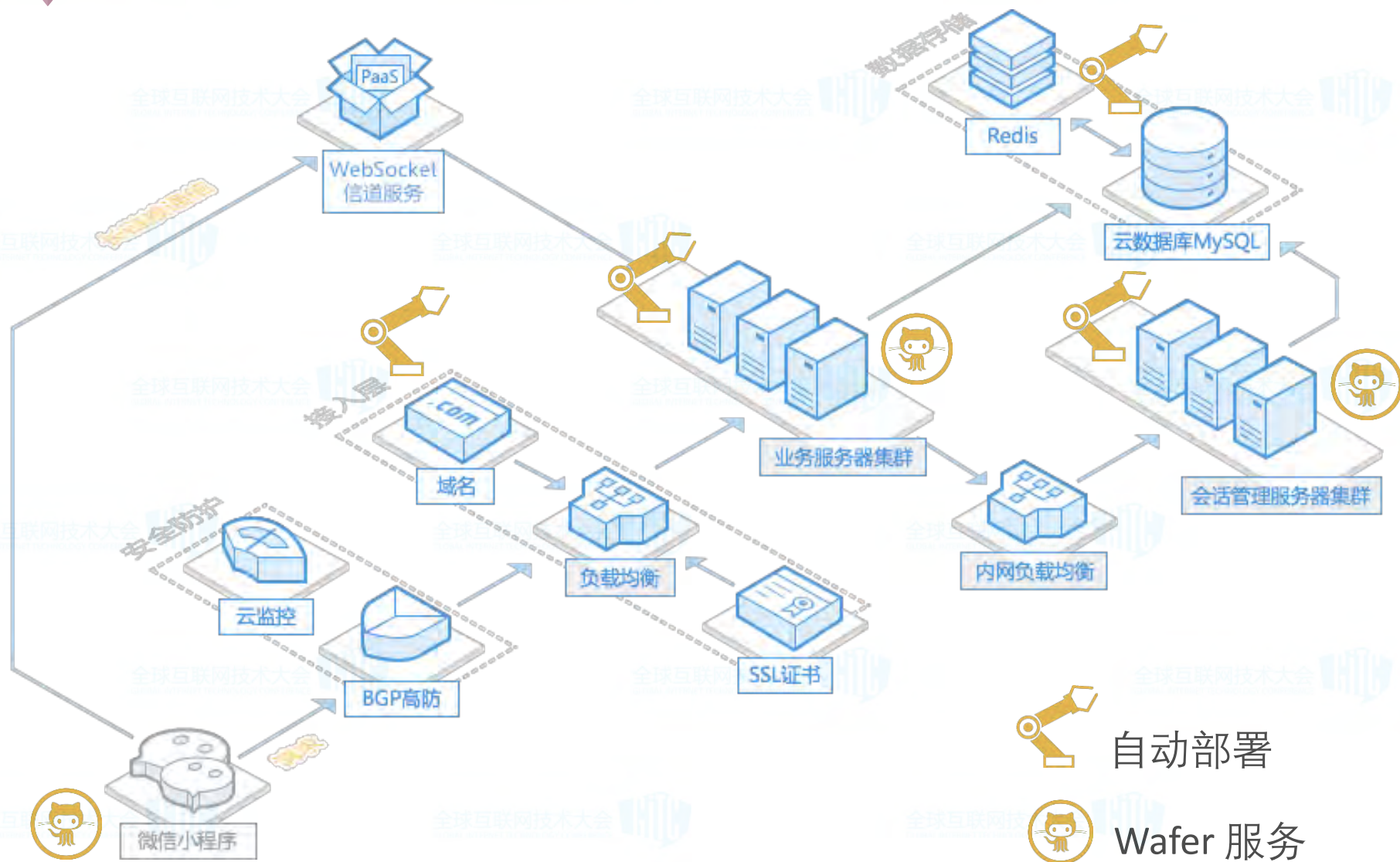
解决方案

业务开发

小程序开发者

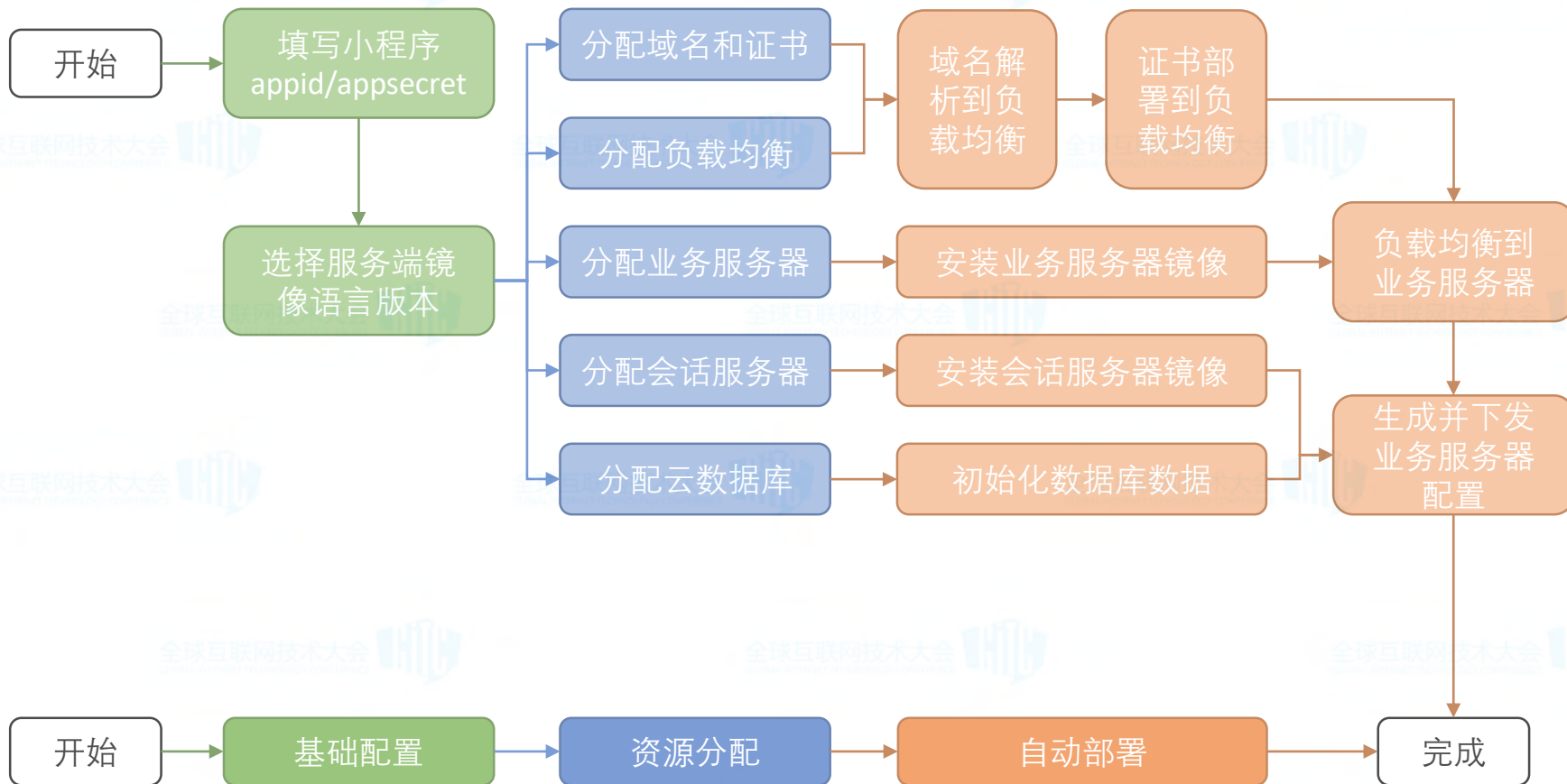


Wafer 产品化实践方案





自动部署过程





腾讯云小程序解决方案



基础设施
自动部署
软件与平台
Wafer - 开源

小程序业务开发

专业 & 专注

Walfer 产品化实践

腾讯云小程序解决方案



That's all

Question Time