



微信小程序云端解决方案

探索之路



黄荣奎 (liamhuang)
腾讯web前端高级工程师

2011年毕业加入腾讯
参与QQ秀, QQ农场, QQ牧场项目

2013年 负责QQ会员, 腾讯手机QQ游戏平台, 在
H5 (Hybird App) 方面有丰富的经验

2016-至今 负责腾讯云网站服务, 网络安全业务

微信小程序是一种全新的连接用户与服务的方式，它可以在微信内被便捷地获取和传播，同时具有出色的使用体验。



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？



小程序是 H5 应用吗？

不是



小程序不是 H5 应用

- 无法使用window对象
- 无法使用DOM元素
 - HTML5 生态中基于 DOM 的库全部无法使用（如 jQuery）
 - 小程序有自己的模板语言WXML（mvvm模型）
- 并非 URL 访问，没有域名概念。无法跳转到url上
 - 不存在跨域问题
 - 无 Cookie 存储
- 小程序更偏向 CS 架构
- 性能更加好,体验更加接近原生



小程序如何开发？



iPhone 6

wifi

quickstart

app.json

app.wxss

app.js

index.js

...



编辑



调试



项目

WeChat



黄荣奎

Hello World

- pages
 - index
 - index.js
 - index.wxml
 - index.wxss
 - logs
 - logs.js
 - logs.json
 - logs.wxml
 - logs.wxss

- utils
 - app.js
 - app.json
 - app.wxss

```
1 //index.js
2 //获取应用实例
3 var app = getApp()
4 Page({
5   data: {
6     motto: 'Hello World',
7     userInfo: {}
8   },
9   //事件处理函数
10  bindViewTap: function() {
11    wx.navigateTo({
12      url: '../logs/logs'
13    })
14  },
15  onLoad: function () {
16    console.log('onLoad')
17    var that = this
18    //调用应用实例的方法获取全局数据
19    app.getUserInfo(function(userInfo){
20      //更新数据
21      that.setData({
22        userInfo:userInfo
23      })
24    })
25  }
26 })
27
```

pages/index/index.js

:



编译



关闭



第1步全局配置 app.json

```
{
  "pages": [
    "pages/index/index",
    "pages/logs/index"
  ],
  "window": {
    "navigationBarTitleText": "Demo"
  },
  "tabBar": {
    "list": [{
      "pagePath": "pages/index/index",
      "text": "首页"
    }, {
      "pagePath": "pages/logs/logs",
      "text": "日志"
    }]
  },
  "networkTimeout": {
    "request": 10000,
    "downloadFile": 10000
  },
  "debug": true
}
```



第2步注册程序 app.js

```
App({
  onLaunch: function() {
    // Do something initial when launch.
  },
  onShow: function() {
    // Do something when show.
  },
  onHide: function() {
    // Do something when hide.
  },
  globalData: 'I am global data'
})
```



第3步开发应用 index.js

▼ pages

▼ index


 index.js

 index.wxml

 index.wxss

▼ logs

 logs.js

 logs.json

 logs.wxml

 logs.wxss

```
var app = getApp()
```

```
Page({
```

```
  data: {
```

```
    motto: 'Hello World',
```

```
    userInfo: {}
```

```
  },
```

```
  //事件处理函数
```

```
  bindViewTap: function() {
```

```
    wx.navigateTo({
```

```
      url: '../logs/logs'
```

```
    })
```

```
  },
```

```
  onLoad: function () {
```

```
    console.log('onLoad')
```

```
    var that = this
```

```
    //调用应用实例的方法获取全局数据
```

```
    app.getUserInfo(function(userInfo){
```

```
      //更新数据
```

```
      that.setData({
```

```
        userInfo:userInfo
```

```
      })
```

```
    })
```

```
  }
```



模板渲染WXML(WeiXin Markup Language)

```
<!--wxml-->
```

```
<view> {{message}} </view>
```

```
// page.js
```

```
Page({  
  data: {  
    message: 'Hello MINA!'  
  }  
})
```

```
app.getUserInfo(function(userInfo){  
  //更新数据  
  that.setData({  
    message:"got userInfo"  
  })  
})
```

index.wxxml



```
1 <!--index.wxml-->
2 <view class="container">
3   <view bindtap="bindViewTap" class="userinfo">
4     <image class="userinfo-avatar" src="{{userInfo.avatarUrl}}" background-size=
5     <text class="userinfo-nickname">{{userInfo.nickName}}</text>
6   </view>
7   <view class="usermotto">
8     <text class="user-motto">{{motto}}</text>
9   </view>
10 </view>
```

index.js



```
1 //index.js
2 //获取应用实例
3 var app = getApp()
4 Page({
5   data: {
6     motto: 'Hello World',
7     userInfo: {}
8   },
9   //事件处理函数
10  bindViewTap: function() {
11    wx.navigateTo({
12      url: '../logs/logs'
13    })
14  }
```

index.wxss



```
1 /**index.wxss**/
2 .userinfo {
3   display: flex;
4   flex-direction: column;
5   align-items: center;
6 }
7
8 .userinfo-avatar {
9   width: 128rpx;
10  height: 128rpx;
11  margin: 20rpx;
12  border-radius: 50%;
13 }
```



黄荣奎

Hello World



丰富的客户端能力

网络：网络请求，上传，下载，websocket

媒体：图片，视频，录音，文件

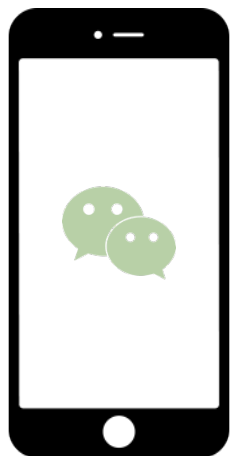
数据：本地存储

位置：位置信息，地图操作

设备：网络状态，系统信息，重力，罗盘

界面：交互反馈，导航，动画，绘图

开放接口：鉴权体系



小程序存在的挑战在哪儿？



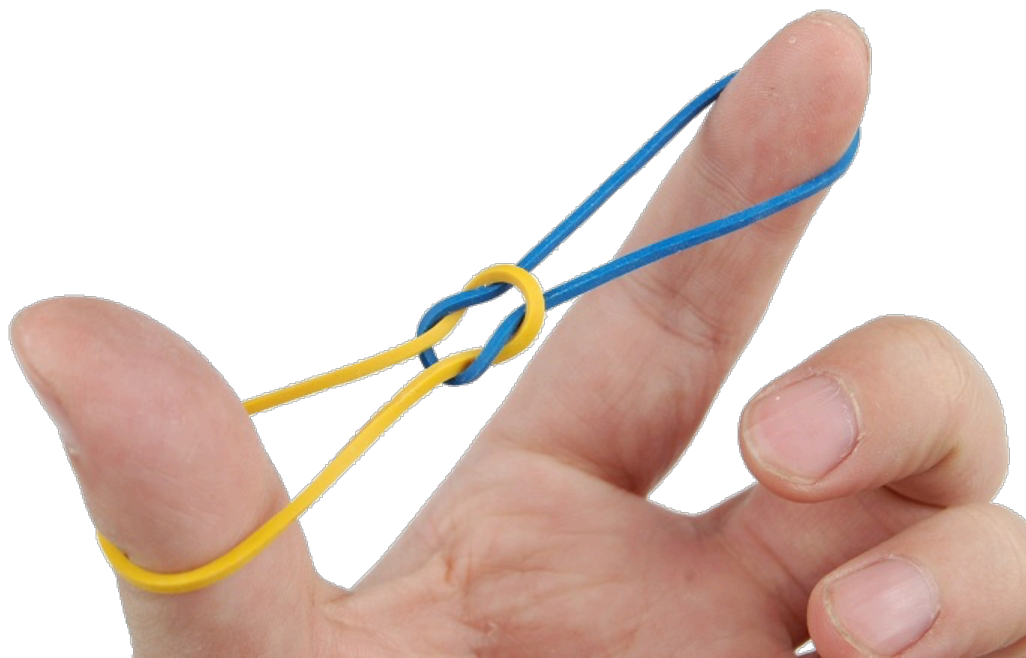
网络和续航

- 在移动网络，环境更加复杂
- 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要求更高



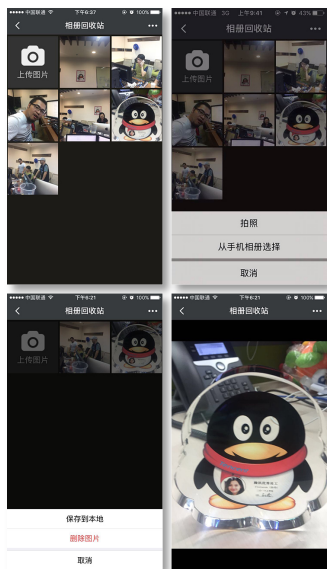
伸缩能力

- 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





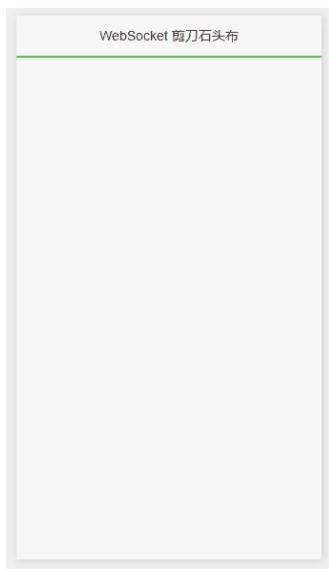
我们进行的尝试



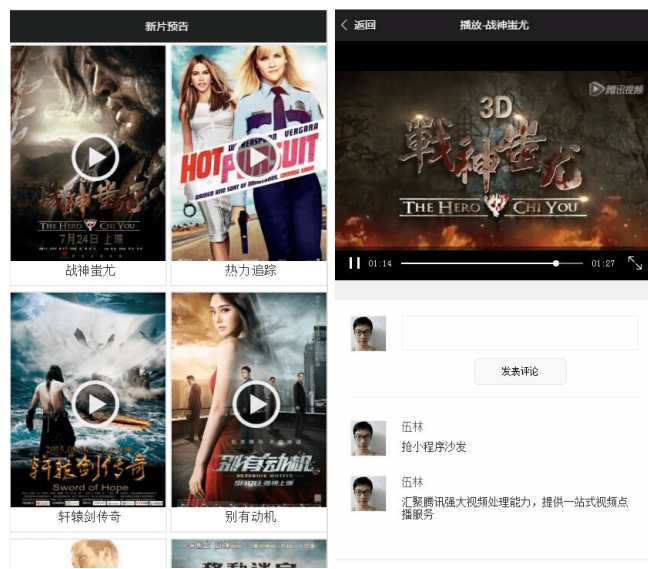
上传下载场景



会话管理服务



WebSocket 服务



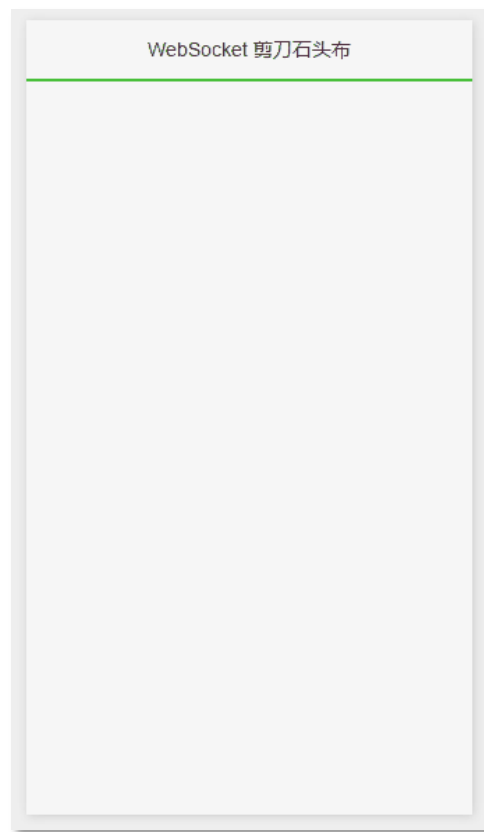
视频点播场景



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

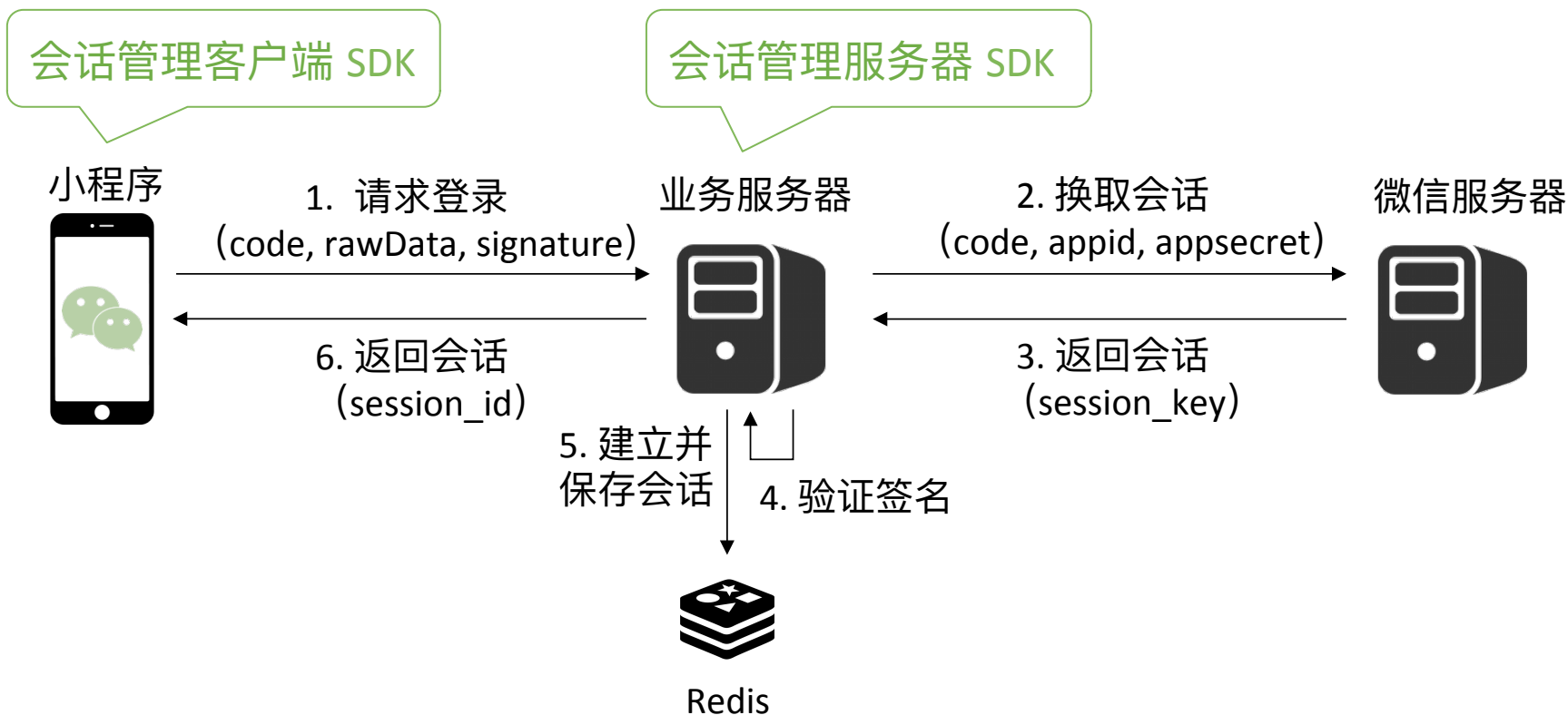


会话管理目标

- 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





会话服务效果分析

- 小程序 appId 和 appSecret 安全性问题
- 缺少横向扩展能力

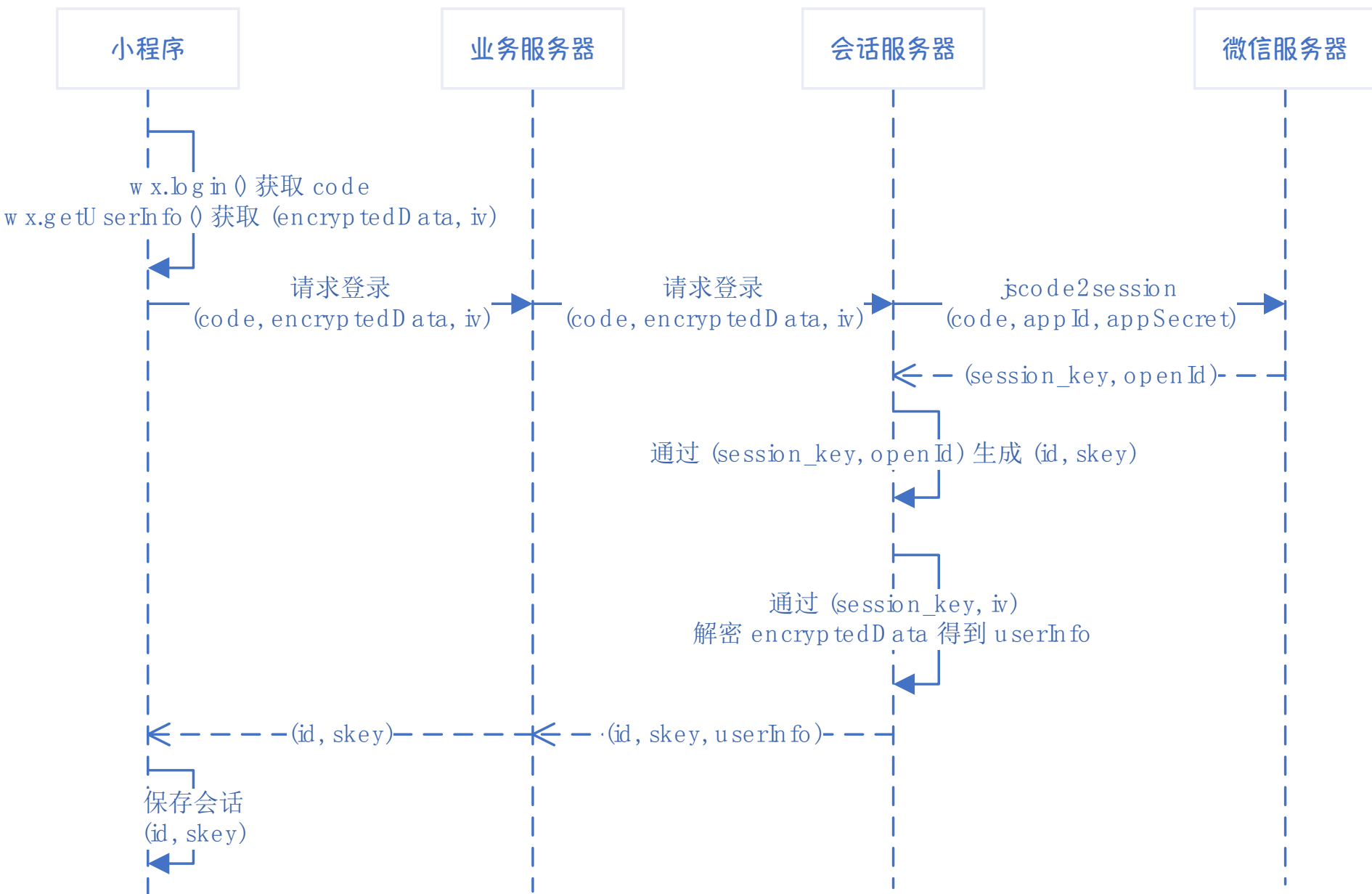


会话服务改进

- 独立会话管理服务器，放置在内网
- 业务服务器通过负载均衡连接，方便扩展
- appId 和 appSecret 存放内网数据库

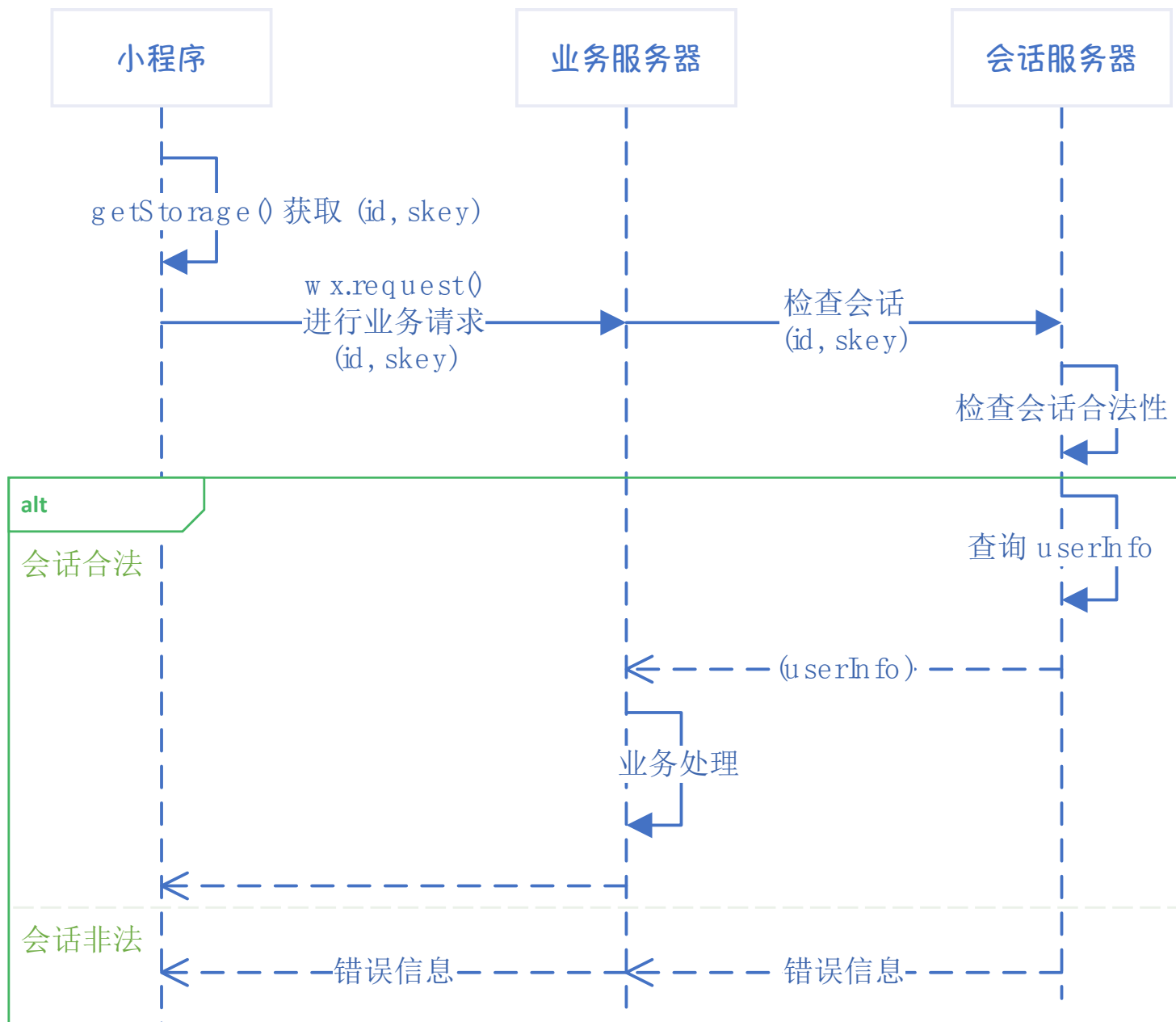


改进的会话建立流程





改进的会话检查流程



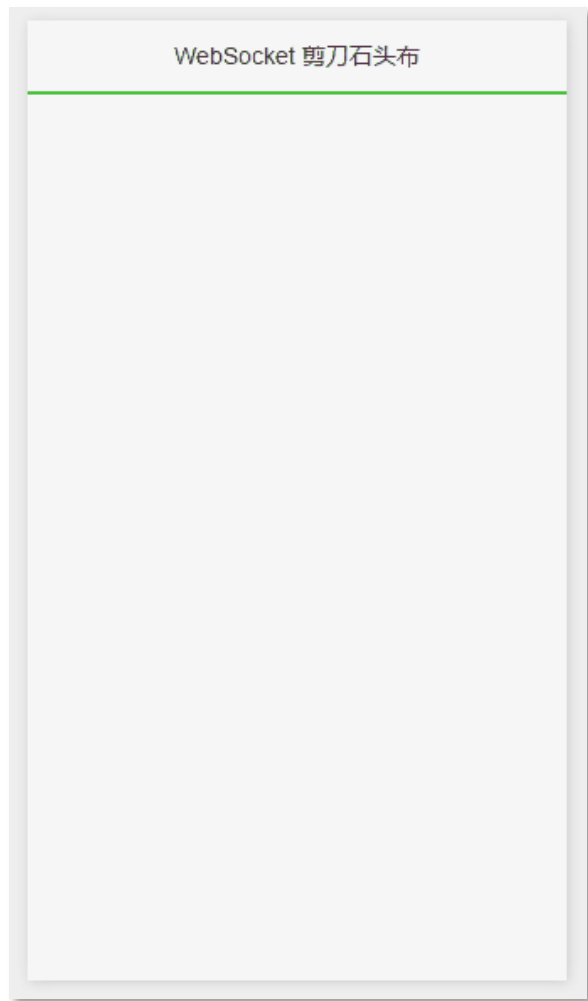


会话服务改进分析

- 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



- 小程序为何要支持 WebSocket?



通信方式对比

传统连接

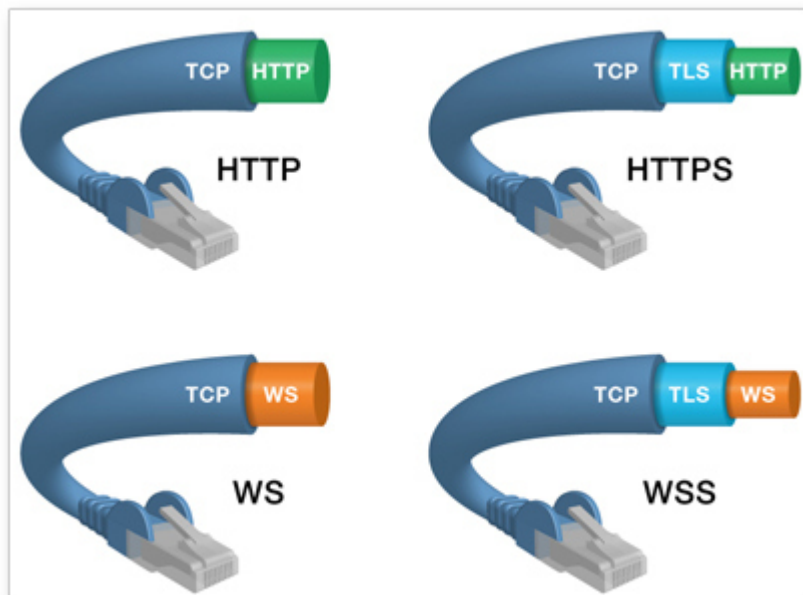
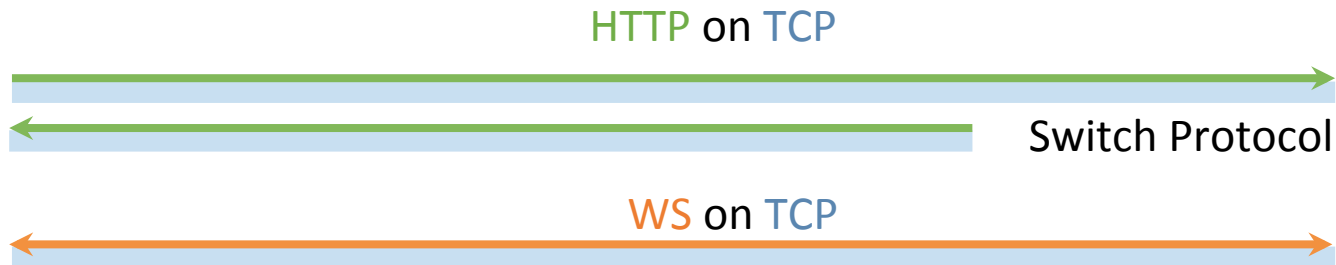
- 建立连接耗时长
- 需要建立多个连接
- 长连接方案只能串行传输
- 耗电体验差

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

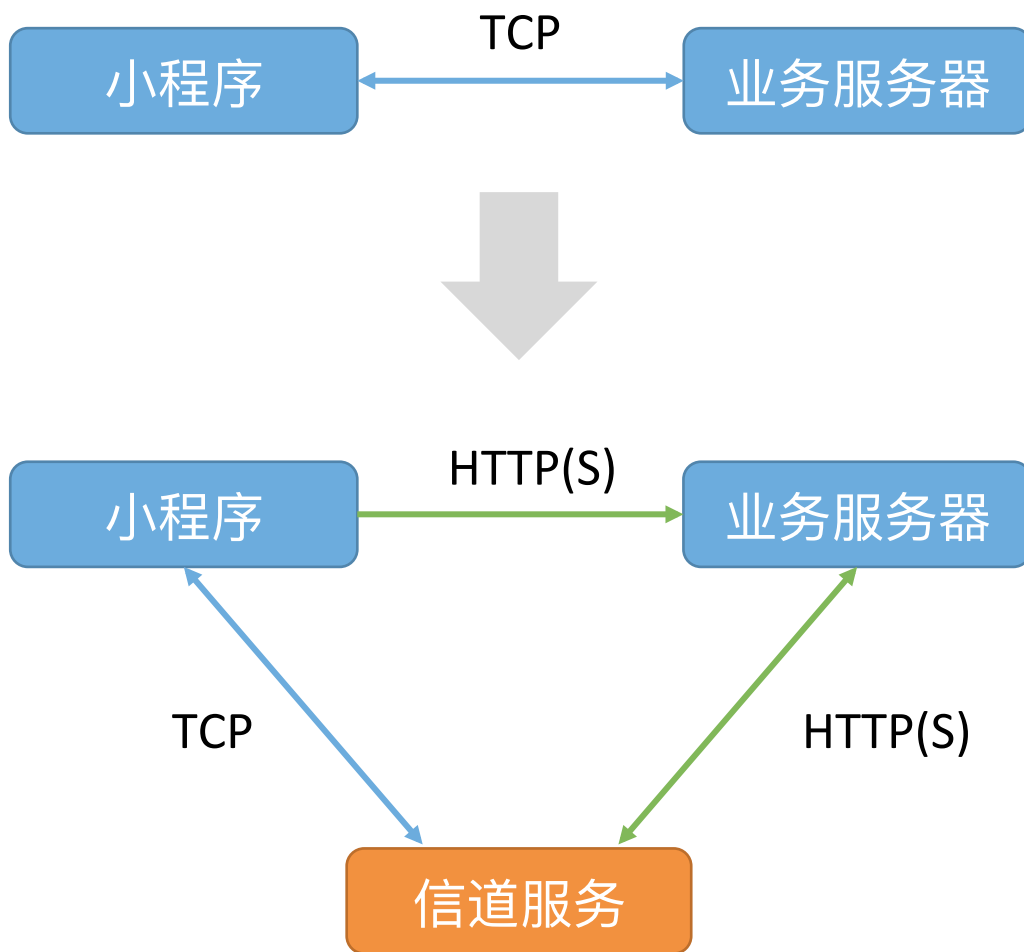


WebSocket 方案改进

- 使用 PaaS 服务支持 WebSocket 连接

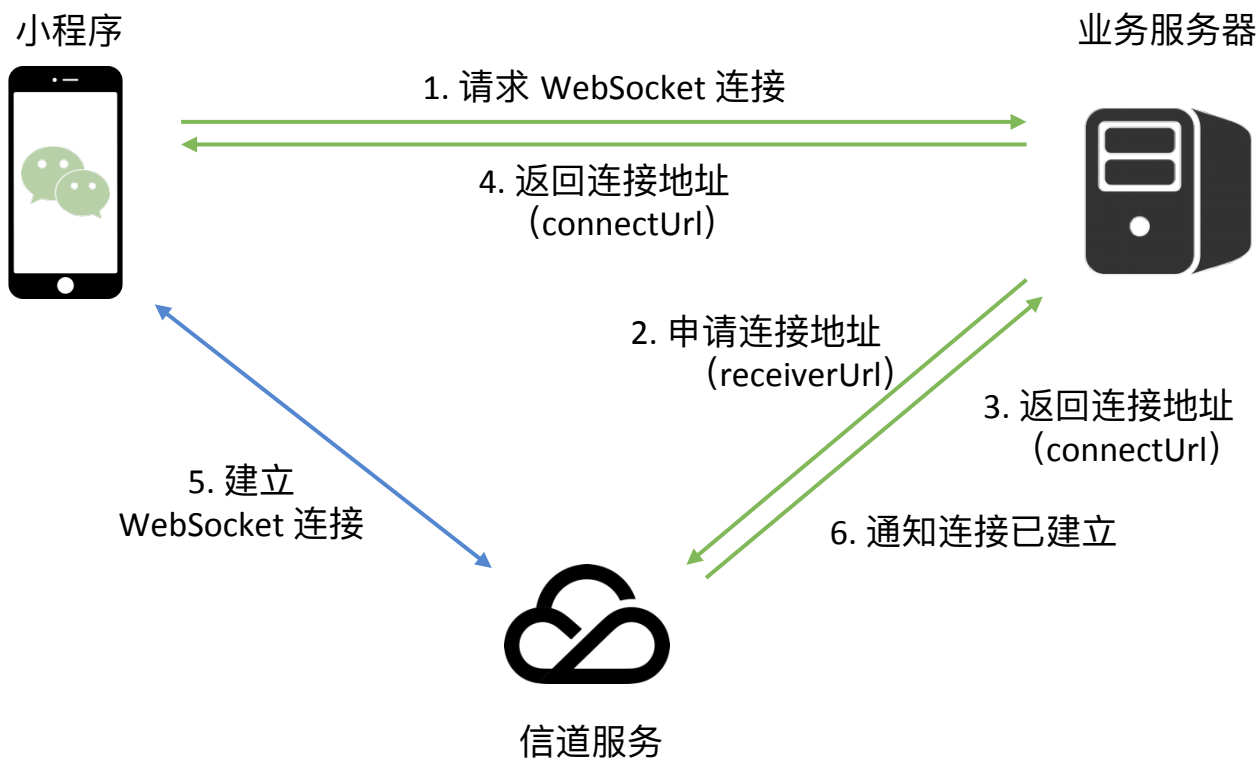


WebSocket 方案改进



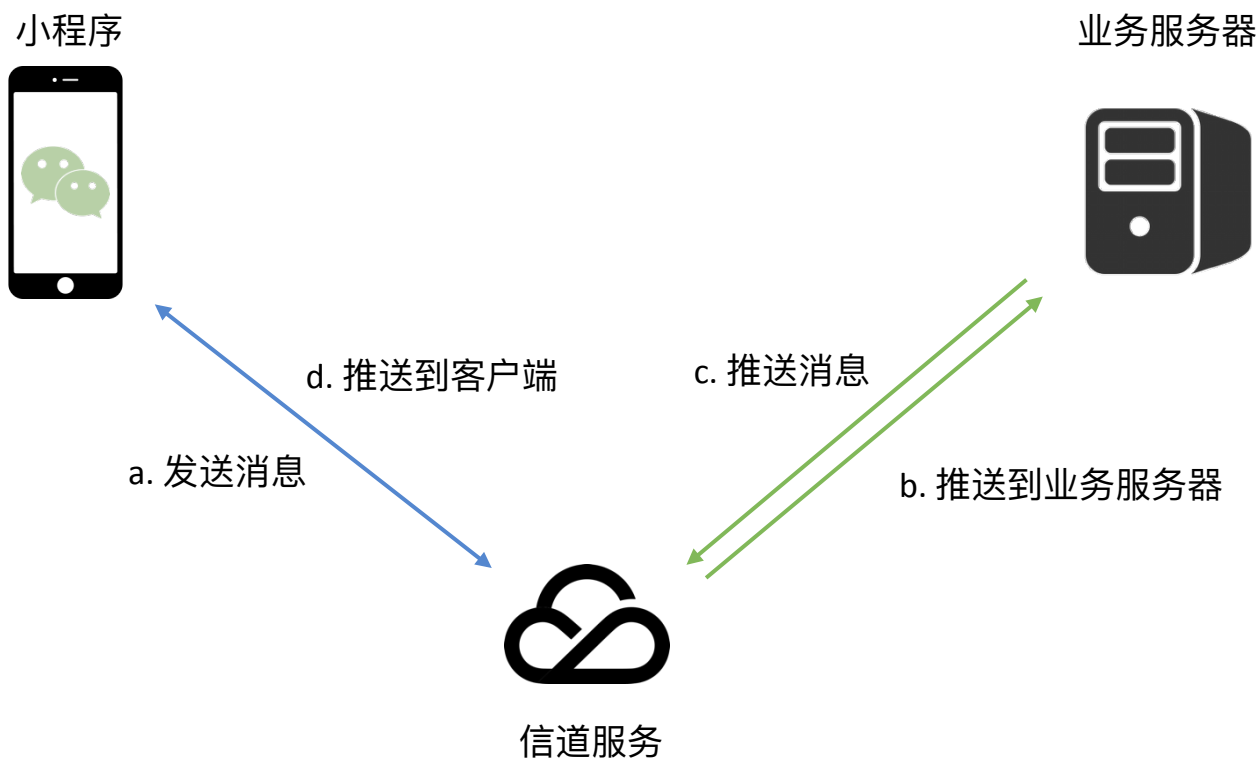


使用 PaaS 服务建立 WS 连接





使用 PaaS 服务进行 WS 通信





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

Wafer

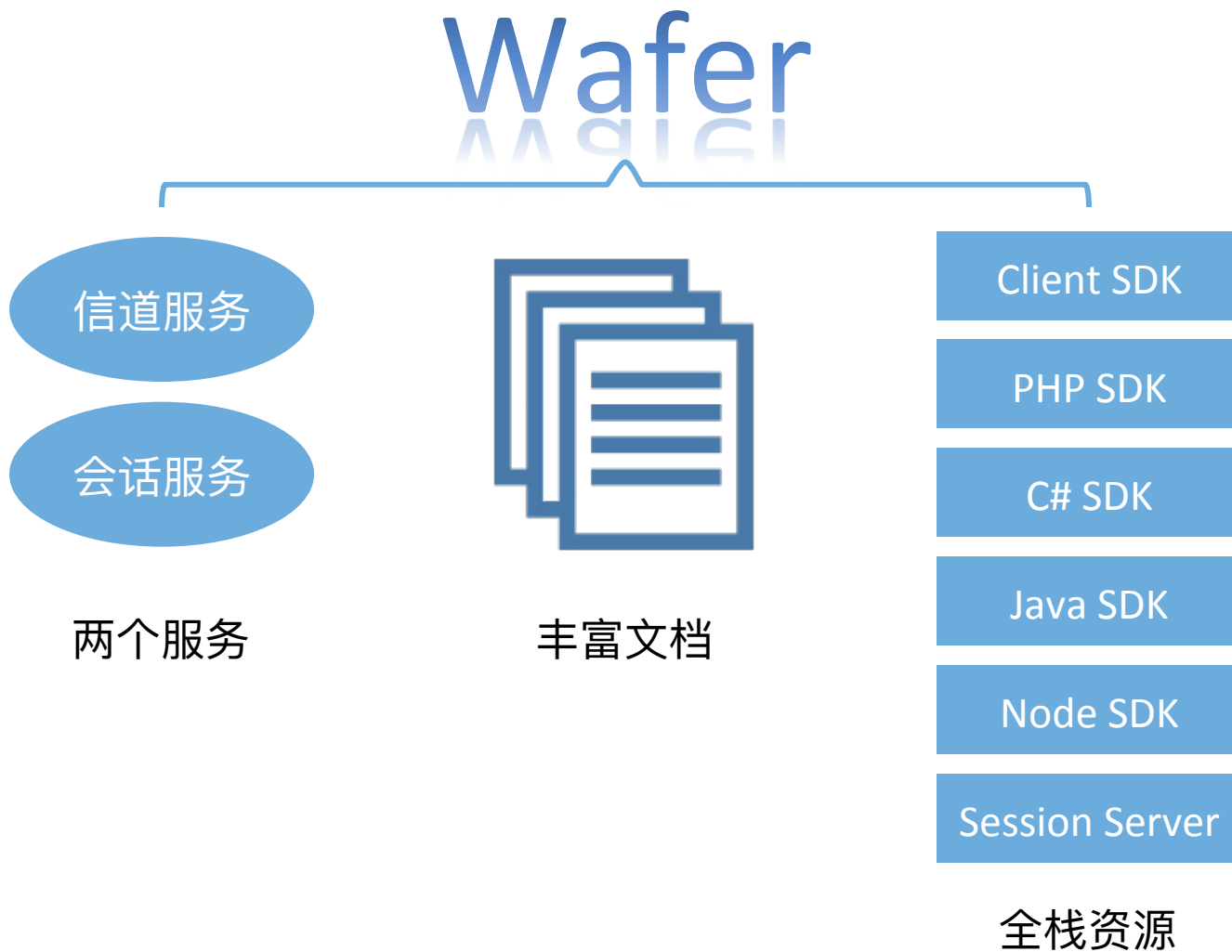
微信小程序全栈开发资源



[https://github.com/tencentyun/
wafer](https://github.com/tencentyun/wafer)



Wafer 包含什么?



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

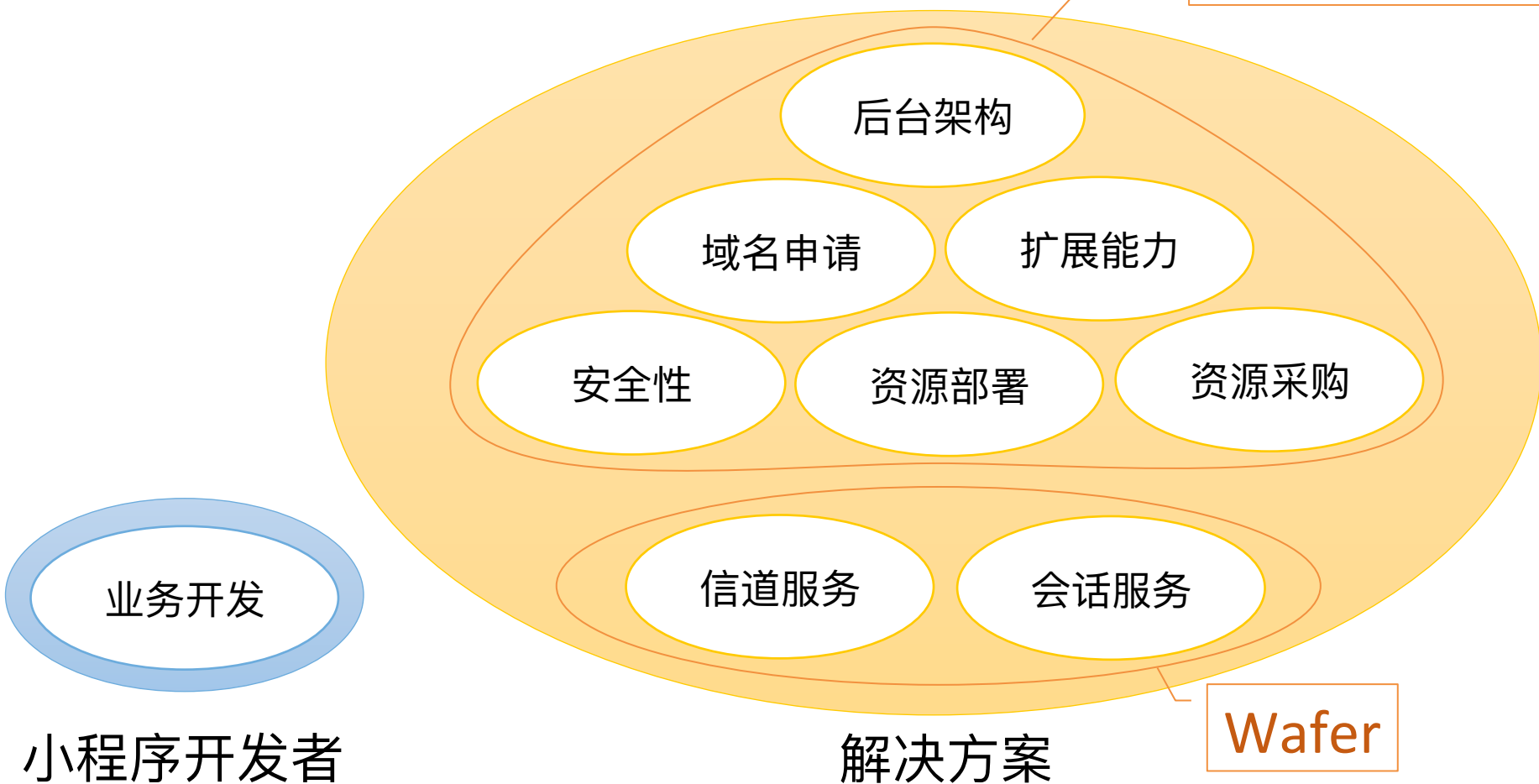


Wafer



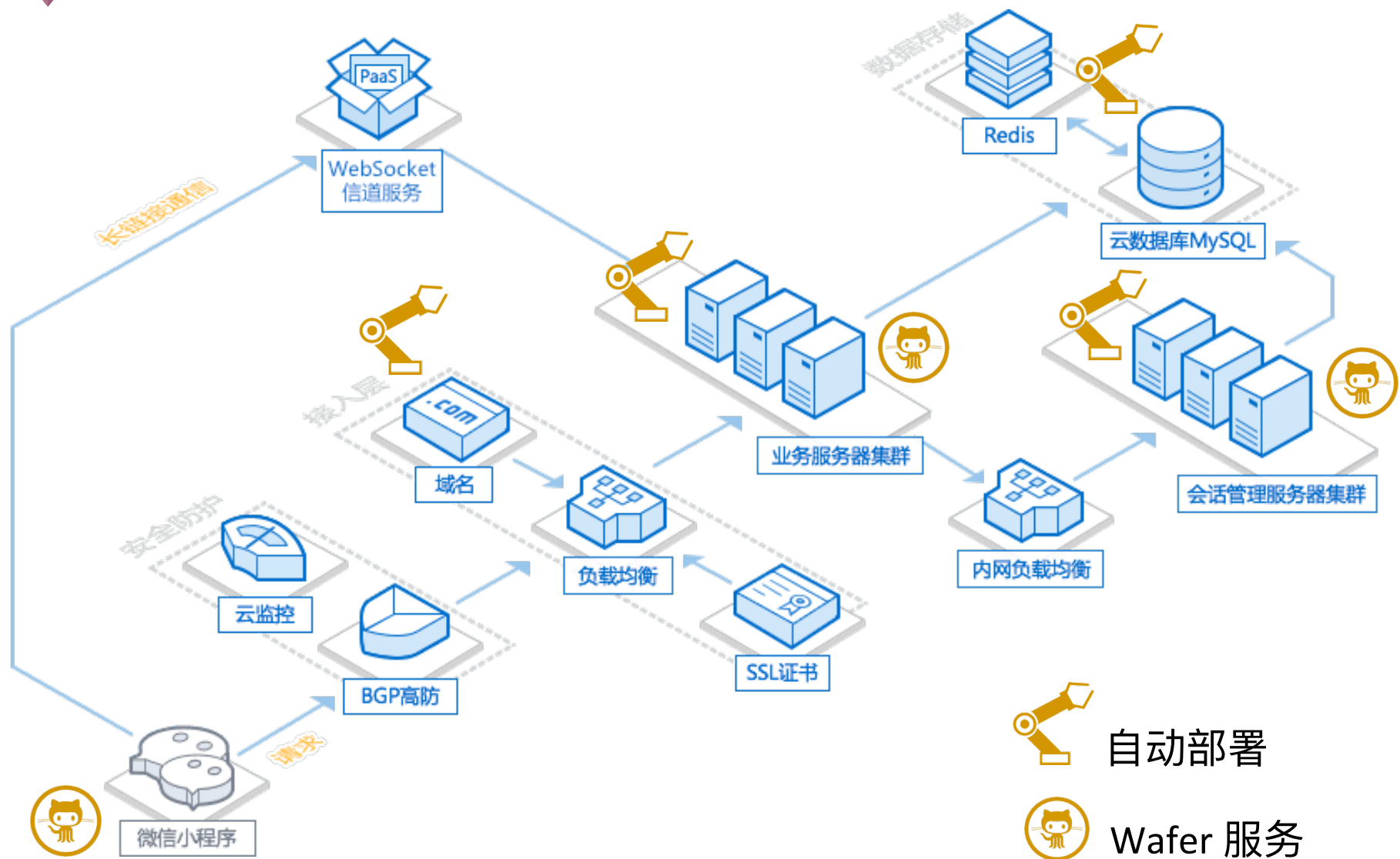
Wafer + 一站式部署

一站式部署



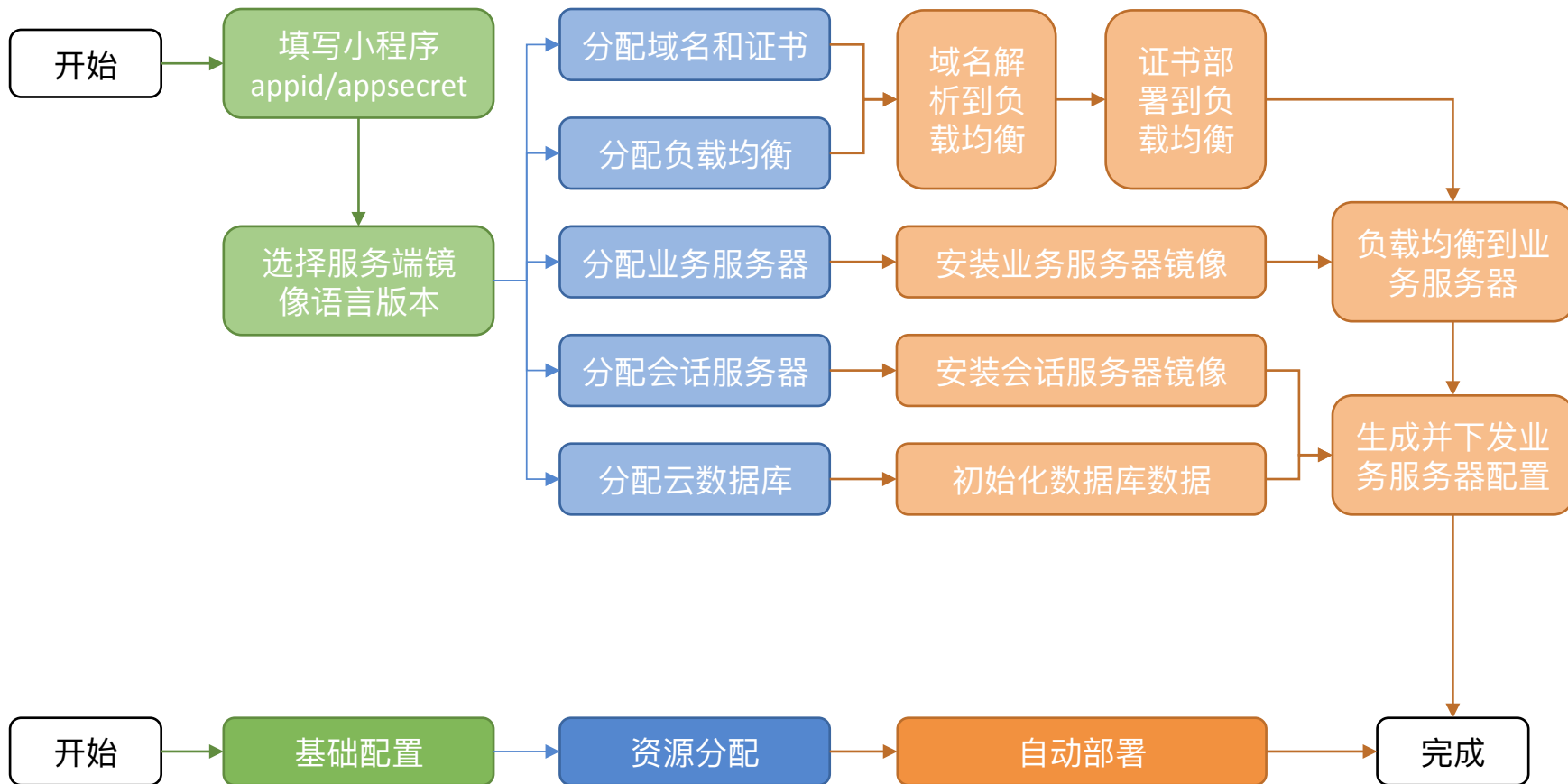


Wafer 产品化实践方案





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

Wafer 产品化实践

腾讯云小程序解决方案



That's all

Question Time