



架构迎接未来变化
IAS 2018



苏宁易购小程序架构实践

苏宁 禹立彬



小程序群与割裂的互联网

- 截止今日，互联网头部公司都推出了自己的小程序产品，腾讯系的微信小程序，应用在微信和QQ，百度的百度小程序，应用在手机百度和百度地图，头条小程序，应用在今日头条和抖音，还有各大厂家的快应用等项目、
- 同时，WAP/APP等应用场景也需要同步开发，成本巨大。





小程序生态

- 围绕着小程序，各厂家也推出了一系列针对小程序的生态标准，联合登录，授权获取用户信息，支付，通知消息，客服等等
- 对于这些生态功能来说，也需要一系列基础服务去支持





小程序的不足

- 组件化能力不足，代码复用能力弱
- API不统一，代码无法兼容
- 这些呼唤着快速解决方案





苏宁的小程序群

- 苏宁易购小程序，苏宁拼购，苏宁特卖，苏宁母婴。。。。
- 苏宁优惠等



苏宁小程序平台

小程序平台架构

	苏宁易购微信	苏宁特卖微信	苏宁拼购微信	苏宁母婴微信	苏宁优惠微信	苏宁易购百度
应用	登录	购物车	订单	商品详情	Webview	频道促销
框架工具	购物流程同步器	微信插件	百度转换器	易购转换器	异常监控	运营埋点
小程序服务	联合登录	二维码	服务通知	百度推荐	图片合成	客服消息



CONTENTS

1

小程序基础服务





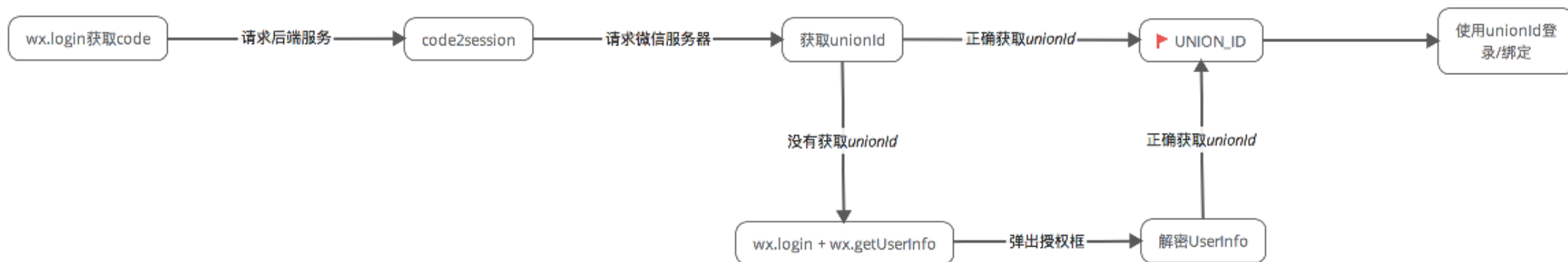
域名收敛服务

1

通过统一的网关，request域名收敛



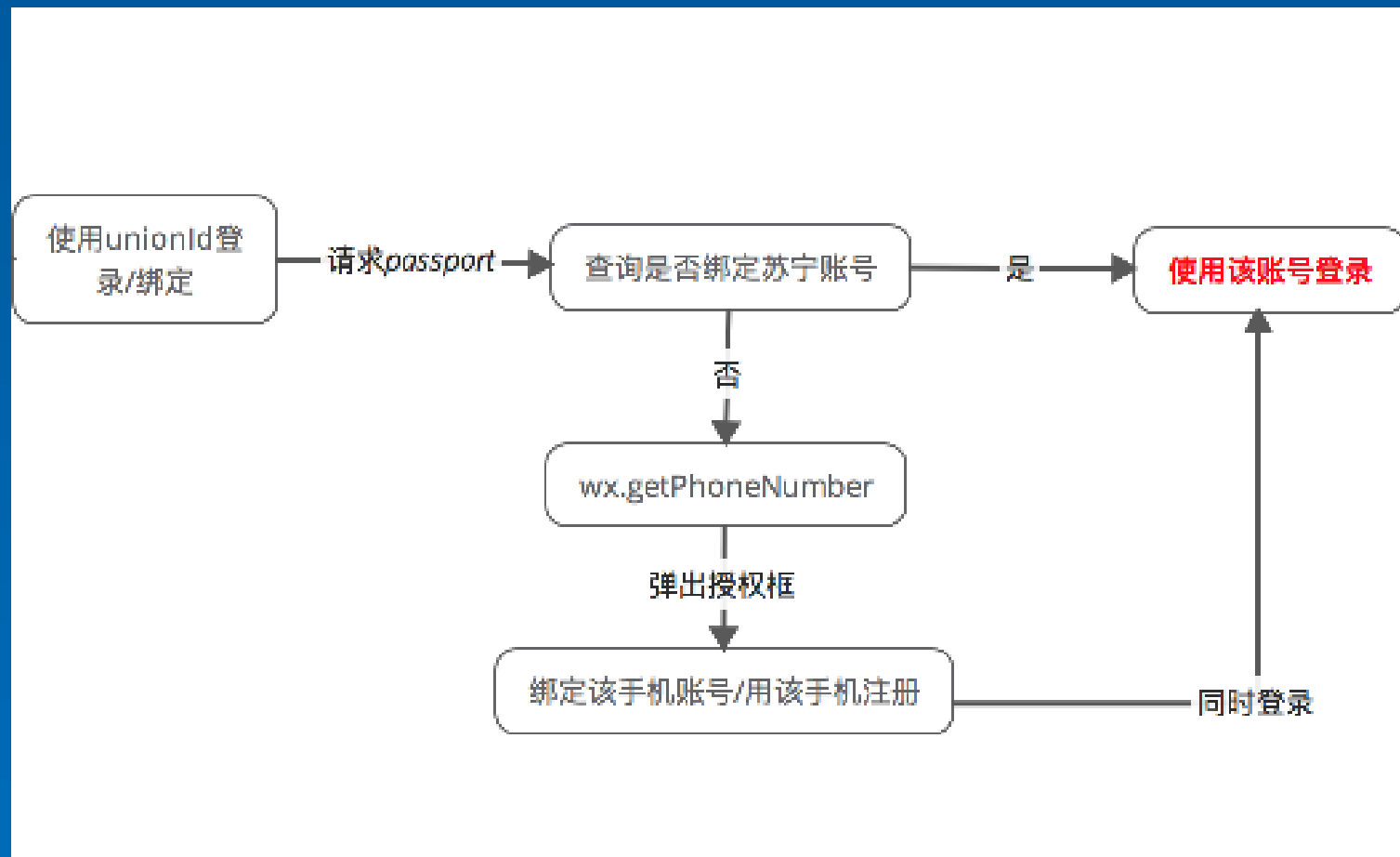
联合登录策略 —— 尝试无授权登录+授权登录



微信/百度小程序联合登录—快速绑定

使用媒体提供的手机号接口，快速注册/绑定

微信/百度共用





小程序服务-二维码&图片

- 二维码服务—— 二维码预生成服务
- 图片合成—— 素材+二维码朋友圈分享





小程序服务-服务通知

- 微信——收集formid
- 业务系统消耗formid发送模板消息





小程序服务-数据推荐

- 百度推荐——用户冷启动服务
- 基于商品库推荐





CONTENTS

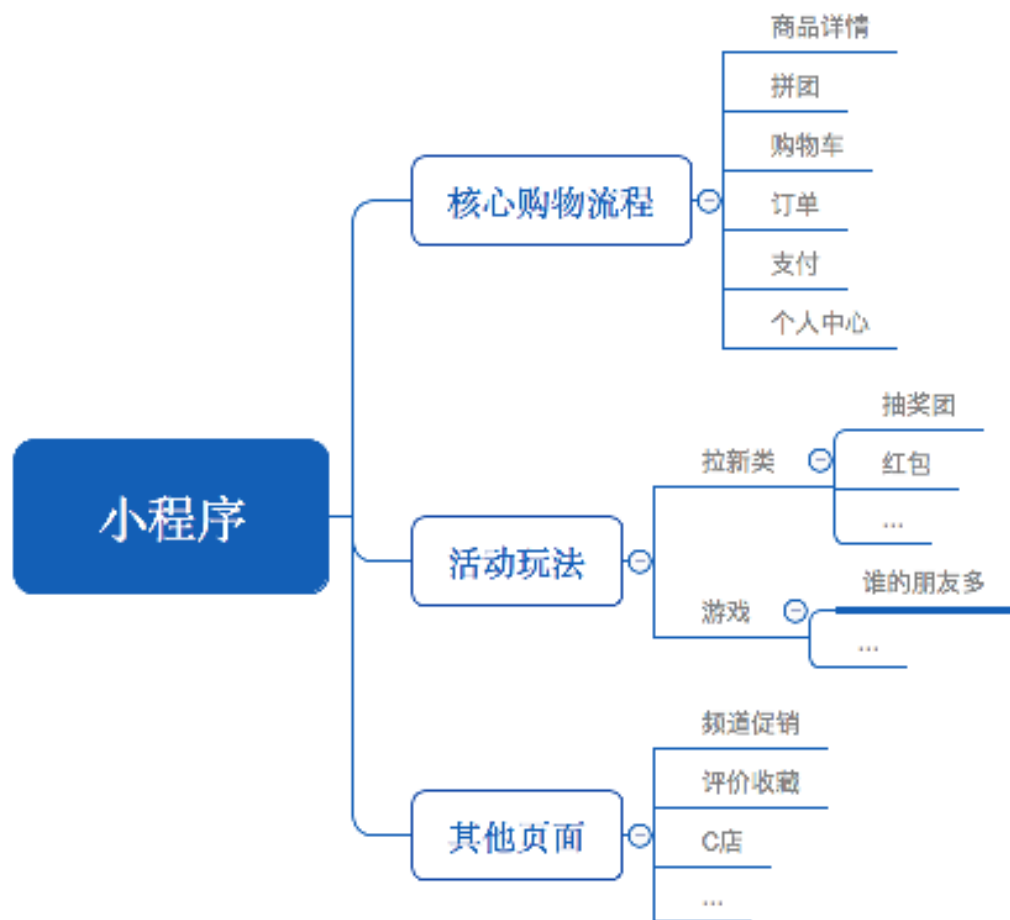
2

小程序前端框架



小程序的前端构成

小程序的业务主要分为，核心购物流程，活动玩法，辅助页面三类





复用购物流程——同步器

- 主购物流程由于登录，支付，TAB原因，无法插件化
- 微信小程序间跳转策略的变化，使得无法集中在一个小程序上实现购物流程，只能每个小程序都具备。
- 同步器中包含了最新的购物流程文件包——订单，购物车，商品详情，个人中心等
- 同步器同步代码，启动新小程序时，在购物主流程上的情况下，同步生成标准项目。





代码同步器

```
module.exports = {  
  '购物车': [{  
    form: '/pages/cart',  
  }],  
  '促销': [{  
    form: '/packageA/pages/quan',  
  }],  
  '订单': [{  
    form: '/packageA/pages/order',  
  }],  
  '查看物流': [{  
    form: '/packageA/pages/logistics',  
  }],  
  '券搜索': [{  
    form: '/packageA/pages/quanSearch',  
  }],  
  '四级页': [{  
    form: '/pages/fourth',  
  }]  
}
```

新的垂直的小程序，使用同步器生成项目脚手架。





微信小程序插件

- 面向活动玩法类
- 小游戏
- 微信官方提供
- 降级到Webview





百度小程序转换器

- 基于百度官方的wx2swan
- 将同等逻辑的微信小程序页面转换为百度
- 无法精确匹配每一个组件
- 少量修改，减少工作量





易购系APP转换器

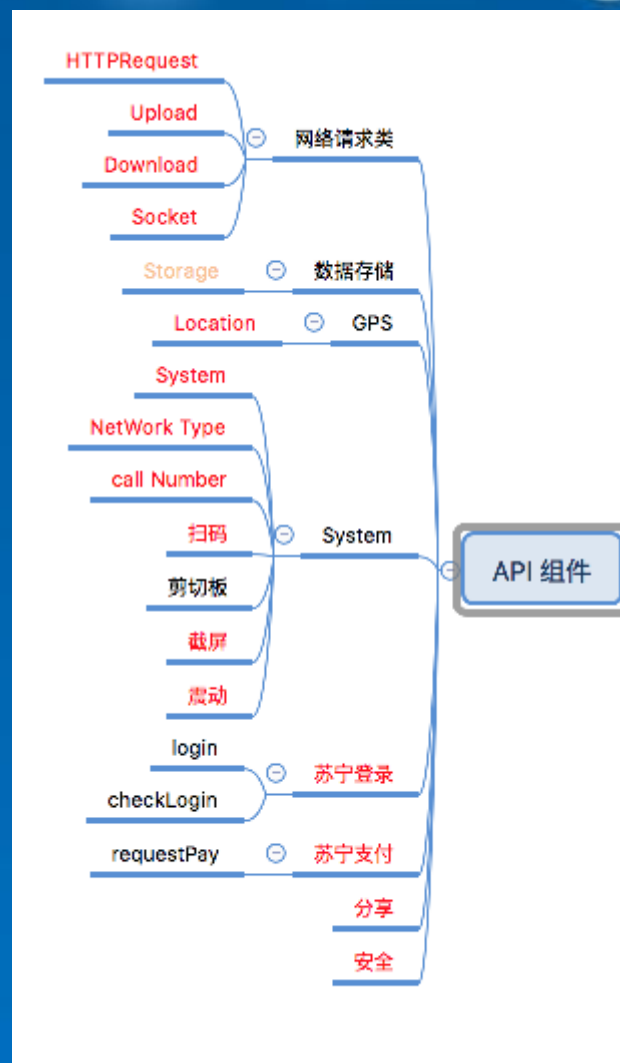
- 将小程序代码转换为基于ReactNative框架的易购版小程序
- 降低易购APP大小
- 减少开发成本





同步组件

- 使用ReactNative同步实现微信小程序组件
- 100+ API
- 30+ Components





Babel转换代码

- data转换为state
- 样式重新解析
- Element转换

```
$ sminip npm run build

> @ build /Users/lihao/Example/sminip
> wx2sminip build --source=/Users/lihao/Work/MINIP/temai0906 --output=temai/dist --watch

> 需要转换的源码目录路径: /Users/lihao/Work/MINIP/temai0906.
> 代码转换后生成的目标路径: temai/dist.

[Success] transform the source file "/Users/lihao/Work/MINIP/temai0906/app.js" to the out file "/Users/lihao/Example/sminip/temai/dist/app.js" is successd!
[Success] transform the source file "/Users/lihao/Work/MINIP/temai0906/app.wxss" to the out file "/Users/lihao/Example/sminip/temai/dist/app.stylesheet.js" is successd!
[Success] transform the source file "/Users/lihao/Work/MINIP/temai0906/ddress/address.js" to the out file "/Users/lihao/Example/sminip/temai/dist/ddress/address.js" is successd!
[Success] transform the source file "/Users/lihao/Work/MINIP/temai0906/ddress/address.wxss" to the out file "/Users/lihao/Example/sminip/temai/dist/ddress/address.stylesheet.js" is successd!
[Success] transform the source file "/Users/lihao/Work/MINIP/temai0906/pages/auth/auth.js" to the out file "/Users/lihao/Example/sminip/temai/dist/pages/auth/auth.js" is successd!
[Success] transform the source file "/Users/lihao/Work/MINIP/temai0906/pages/auth/auth.wxss" to the out file "/Users/lihao/Example/sminip/temai/dist/pages/auth/auth.stylesheet.js" is successd!
[Success] transform the source file "/Users/lihao/Work/MINIP/temai0906/pages/auth/auth.wxml" to the out file "/Users/lihao/Example/sminip/temai/dist/pages/auth/auth.jsx.js" is successd!
[Success] transform the source file "/Users/lihao/Work/MINIP/temai0906/pages/couponList/couponList.js" to the out file "/Users/lihao/Example/sminip/temai/dist/pages/couponList/couponList.js" is successd!
[Success] transform the source file "/Users/lihao/Work/MINIP/temai0906/pages/couponList/couponList.wxss" to the out file "/Users/lihao/Example/sminip/temai/dist/pages/couponList/couponList.stylesheet.js" is successd!
[Success] transform the source file "/Users/lihao/Work/MINIP/temai0906/pages/logistics/logistics.js" to the out file "/Users/lihao/Example/sminip/temai/dist/pages/logistics/logistics.js" is successd!
[Success] transform the source file "/Users/lihao/Work/MINIP/temai0906/pages/logistics/logistics.wxss" to the out file "/Users/lihao/Example/sminip/temai/dist/pages/logistics/logistics.stylesheet.js" is successd!
```





- 业务异常监控 - 自研
- 系统异常监控 - 微信官方后台





运营埋点

- PV UV
- 访问链路
- 转化率





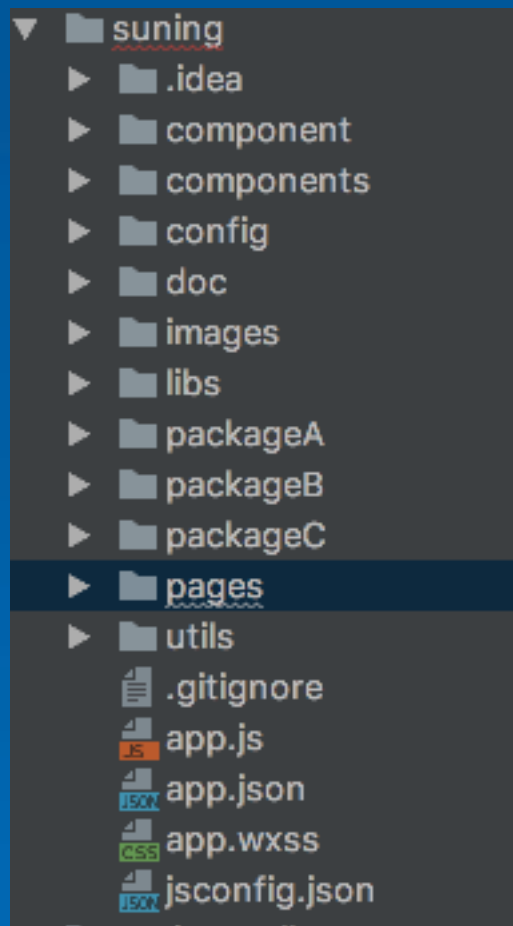
CONTENTS

3

小程序业务组件



- 使用原生的方式开发，没有使用wepy等框架
- 分包加载，主包原则只放购物主流程
- 尽量压缩业务代码，减少文件大小
- 控制基础库版本，减少对用户的影响

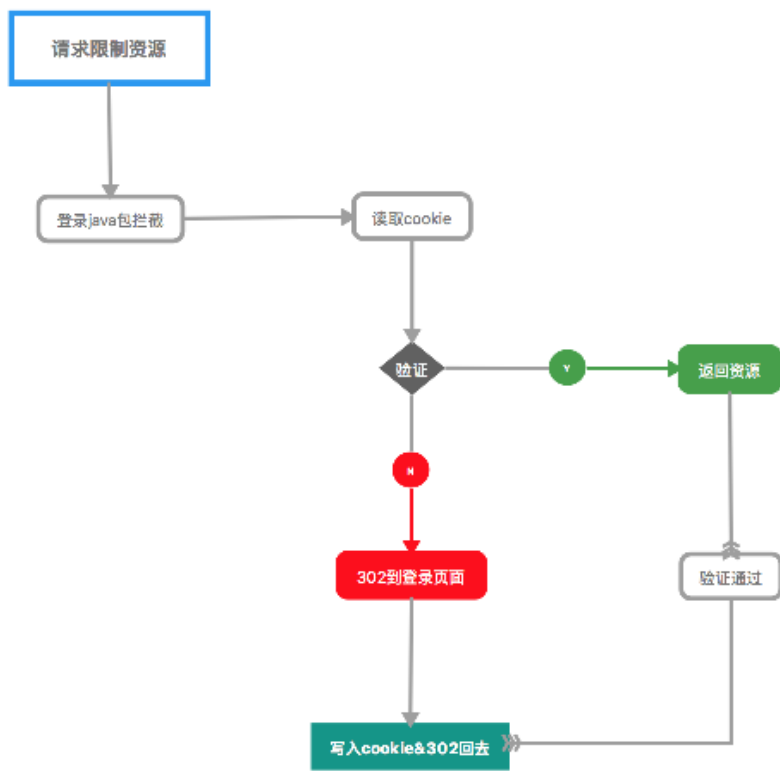


request, 登录与302





request, 登录与302

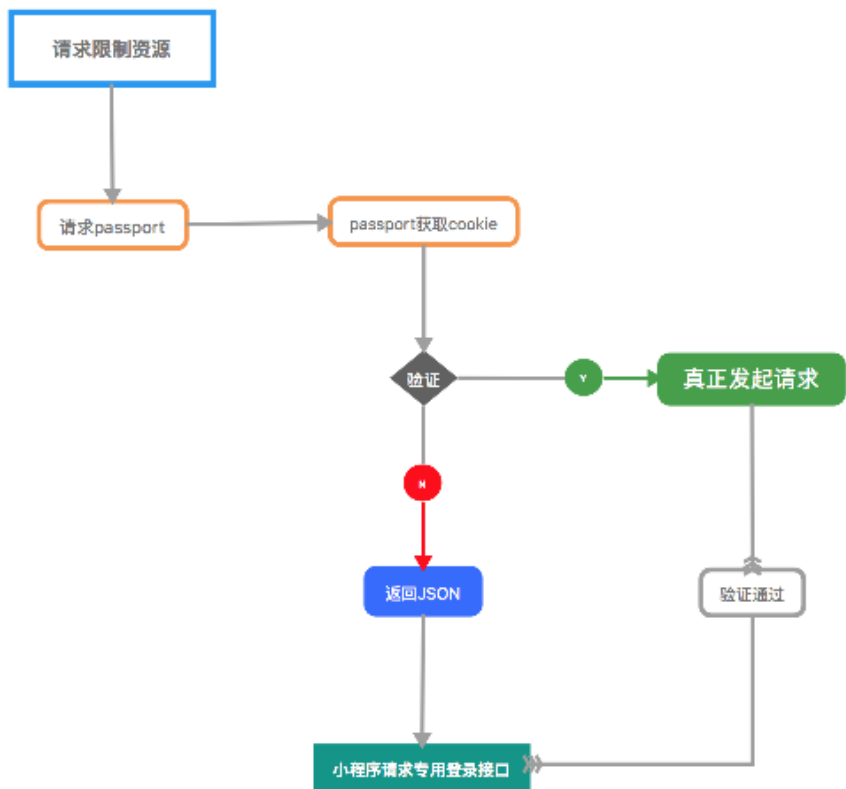


在网页状态下运行良好的登录机制，在小程序下，无法兼容。





loginRequest组件



请求专用接口，并保存登录状态

同时禁止小程序接口中用302传递cookie





小程序Webview

- Webview无法传递cookie
- 大量页面使用Webview嵌入
- 官方都没有提供方便的传递方法





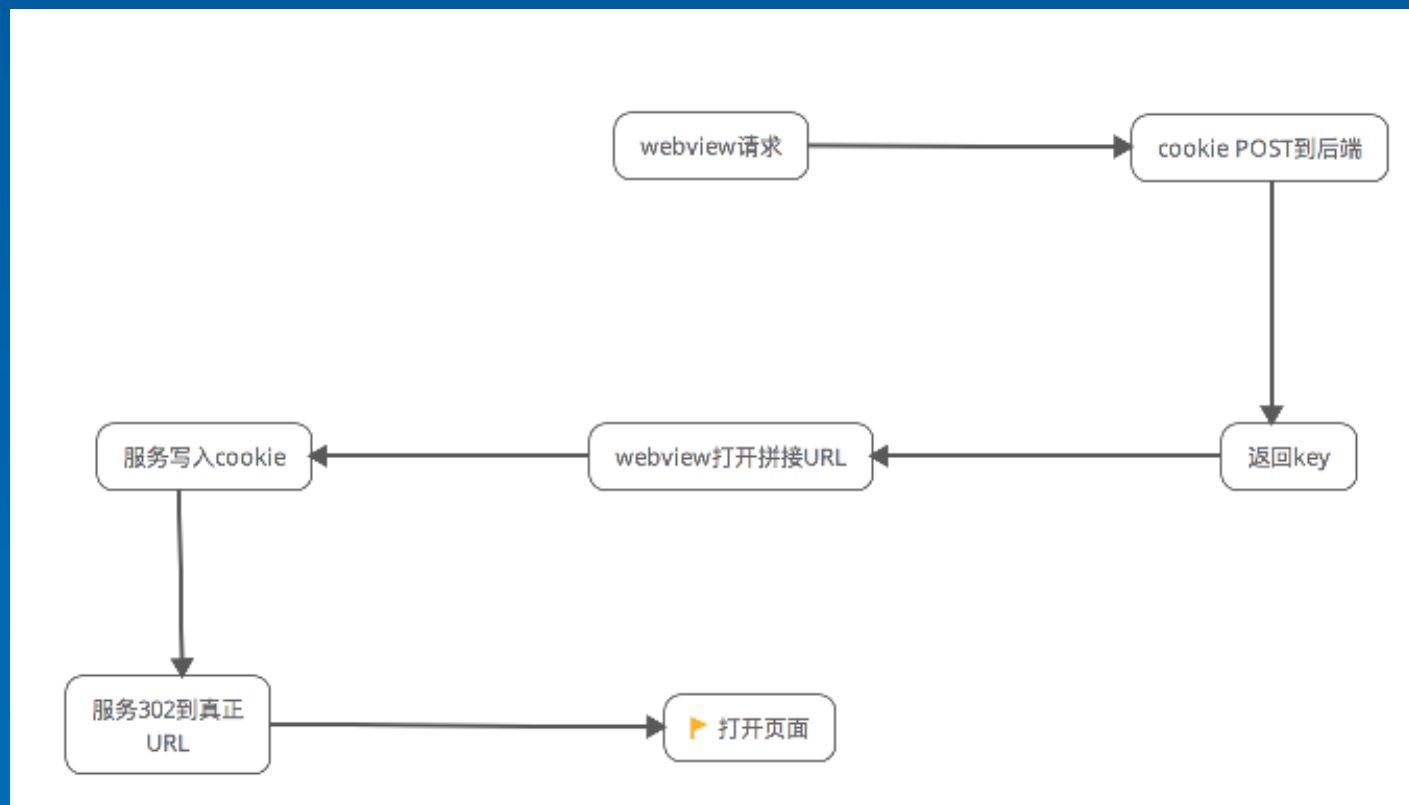
Webview + hash传值

- 将cookie作为hash拼接到URL中
- 内嵌的web页面用JS取hash里的cookie参数，document.cookie写入
- cookie写入后，登录状态存续
- 页面本身不需要登录，限制资源请求在cookie写入后
- cookie不能是http only



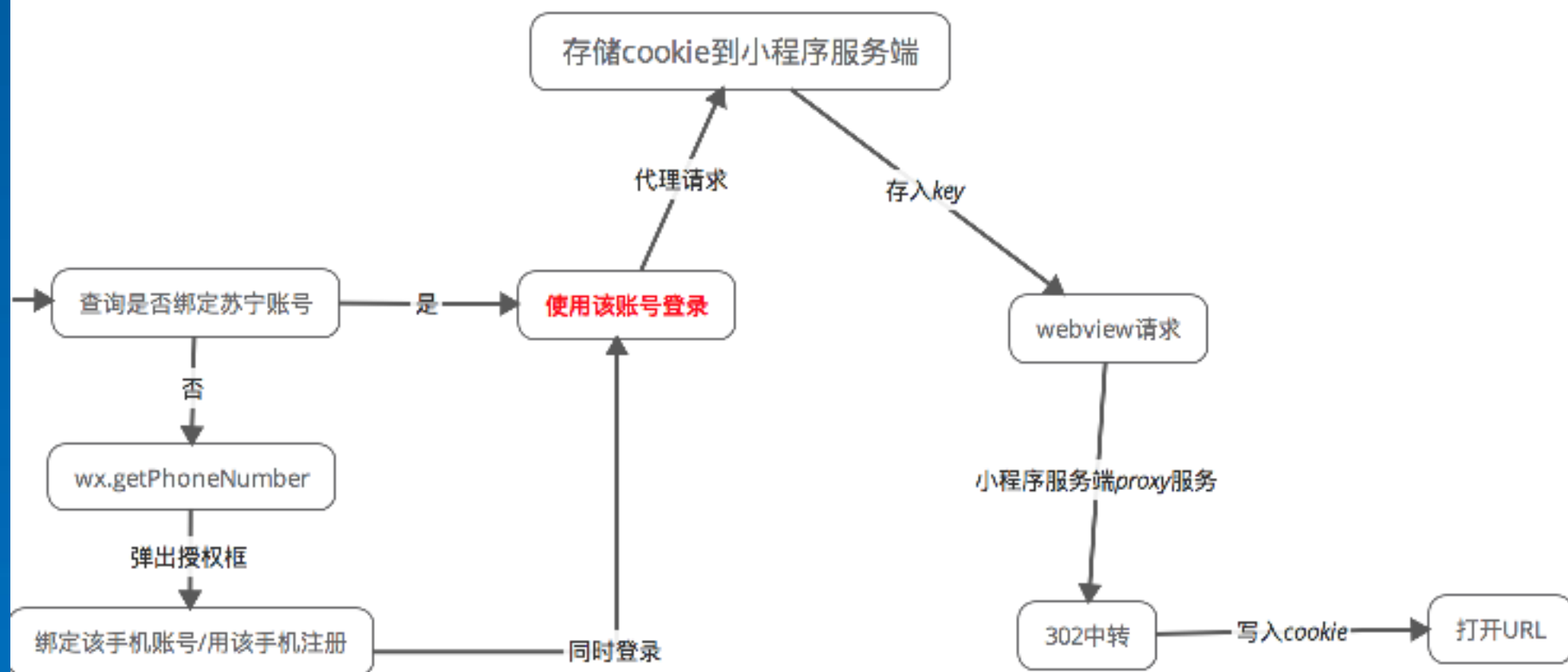
Webview + 后端中转

发起真正的请求前将cookie，POST到后端服务，获得cookie对应的key，然后请求
`proxy?key={key}&url={trueURL}`，后端服务写入cookie后，302跳转到真正需要打开的页面，于是cookie传递





Webview + 后端中转





其他业务组件

- 支付（百度和微信）
- 广告投放组件
- 推客支撑组件
-



苏宁 小程序平台

- 自上线以来，支撑了十余个微信小程序，和百度小程序
- 支撑了双十一期间每天几百万的UV的访问量
- 支持业务拉新，提高分享转化率
- 随着业务的发展，未来将支持头条系及快应用。



谢谢 & QA

