

小程序 webview 应用实践

张所勇 | 开放业务前端负责人 @ 转转





提纲



- 1 小程序技术演进
- 2 webview VS 小程序
- 3 h5多端兼容方案
- 4 小程序sdk设计
- 5 webview常见问题











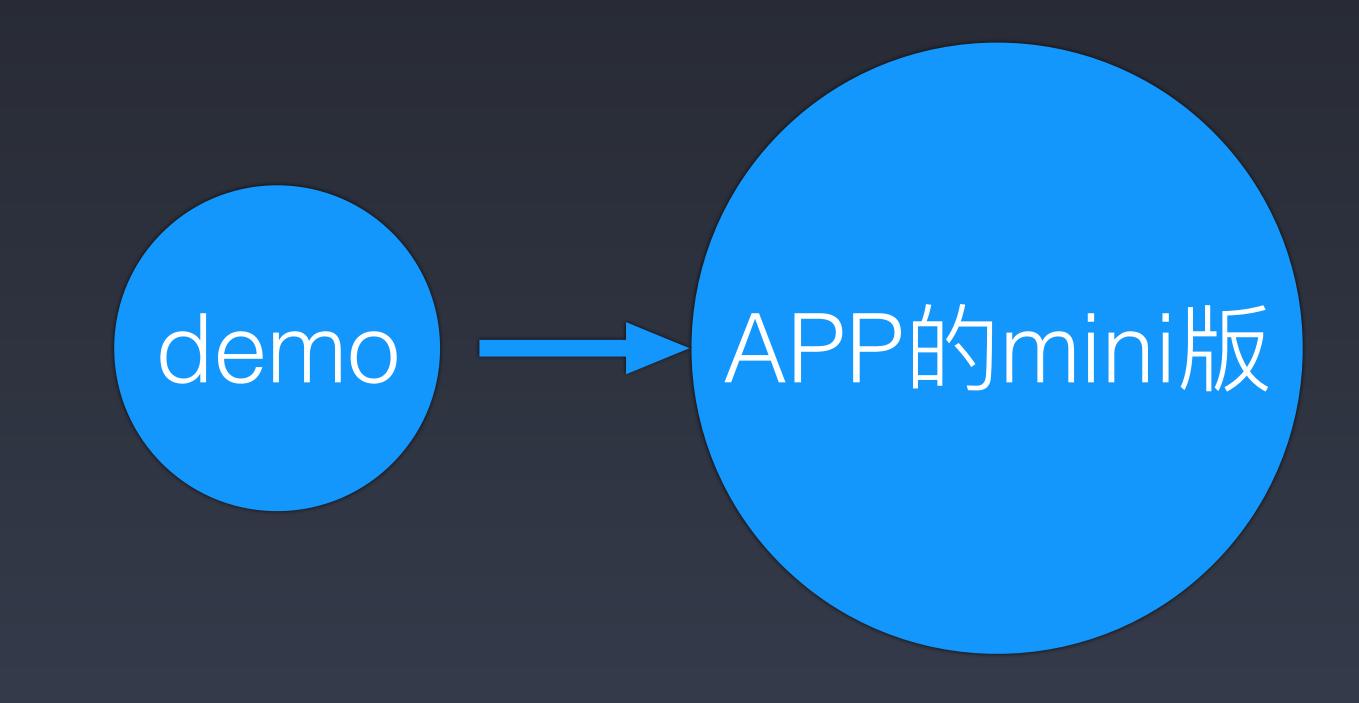






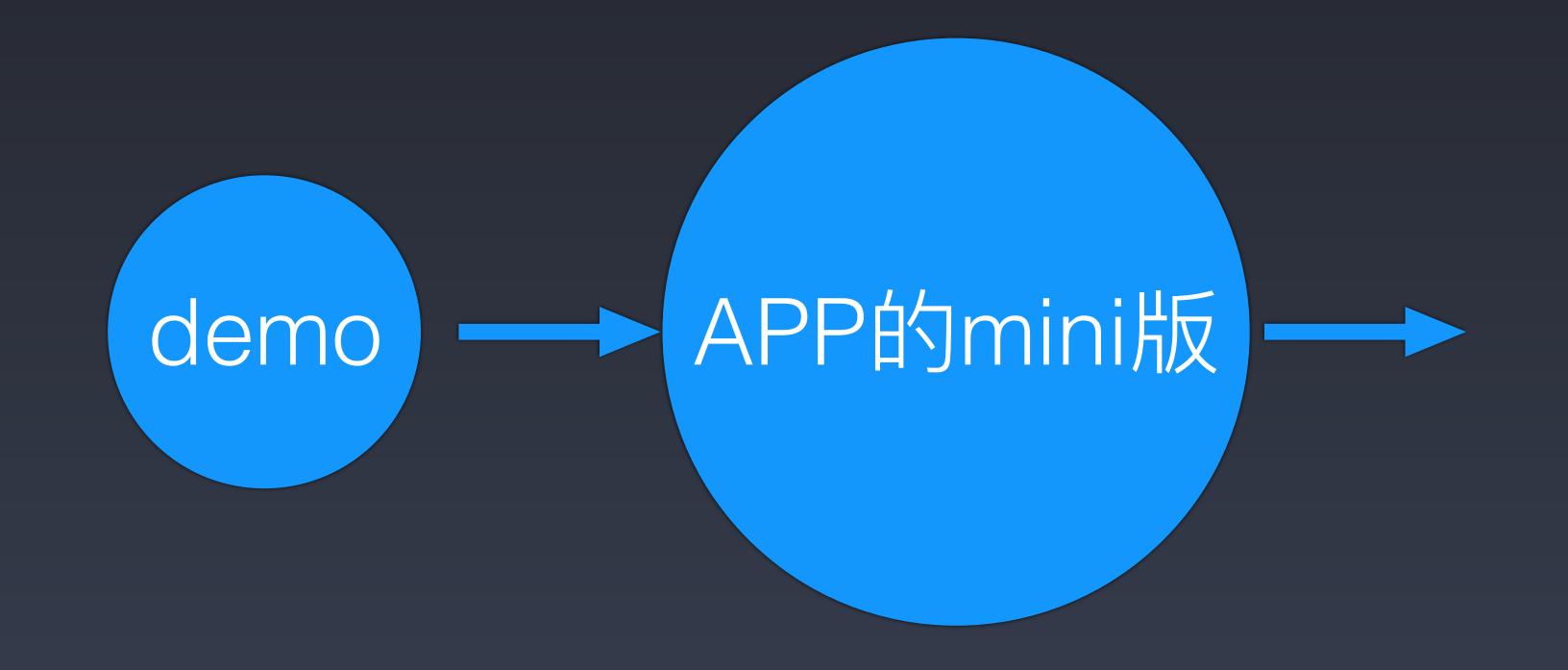






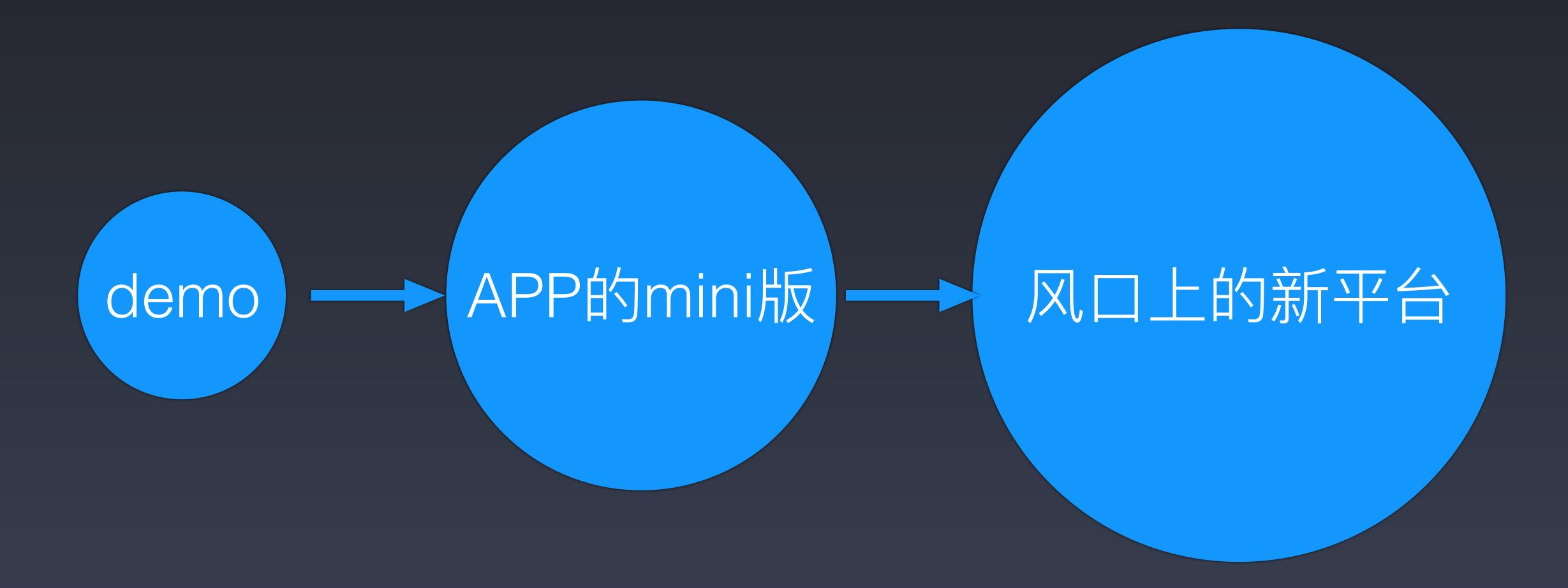










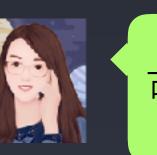




存量H5怎么办







可以之间使用我们的H5吧?

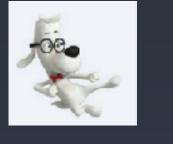




不行啊,现在小程序不支持, 只能用小程序再写一遍



可是我们不会写怎么 我们写吧?Mua~ 可是我们不会写怎么办,你帮







三种开发方式比较



| 开发方案 | 优点 | 缺点 | 存量H5 |
|--------|----------|------------------|------------|
| 纯小程序 | 体验好,速度快 | 不灵活,需要发版审核 增加包体积 | 改造成小程序工程量大 |
| 编译多端 | 一套代码多端运行 | api兼容成本高 | 与APP功能耦合严重 |
| 小程序+h5 | 灵活,工作量小 | 交互、通信成本高 | 与APP功能耦合严重 |



提纲



1 小程序技术演进

2 webview VS 小程序

- h5多端兼容方案
- 4 小程序sdk设计
- 5 webview常见问题



webview与小程序对比



| | webview | 小程序 |
|--------|----------------|-----------|
| 离线能力 | 低 | 高 |
| 页面切换体验 | - - | 高 |
| 首屏速度 | SSR等技术不逊于小程序原生 | 离线能力是天然优势 |
| 二次渲染速度 | | 低 |
| 列表性能 | 高 | 很低 |
| 操作响应 | | 低 |
| 开发灵活性 | | 低 |
| api能力 | | |



提纲



- 1 小程序技术演进
- webview VS 小程序
- 3 h5多端兼容方案
 - 1 小程序sdk设计
 - 5 webview常见问题



一套代码,多端运行







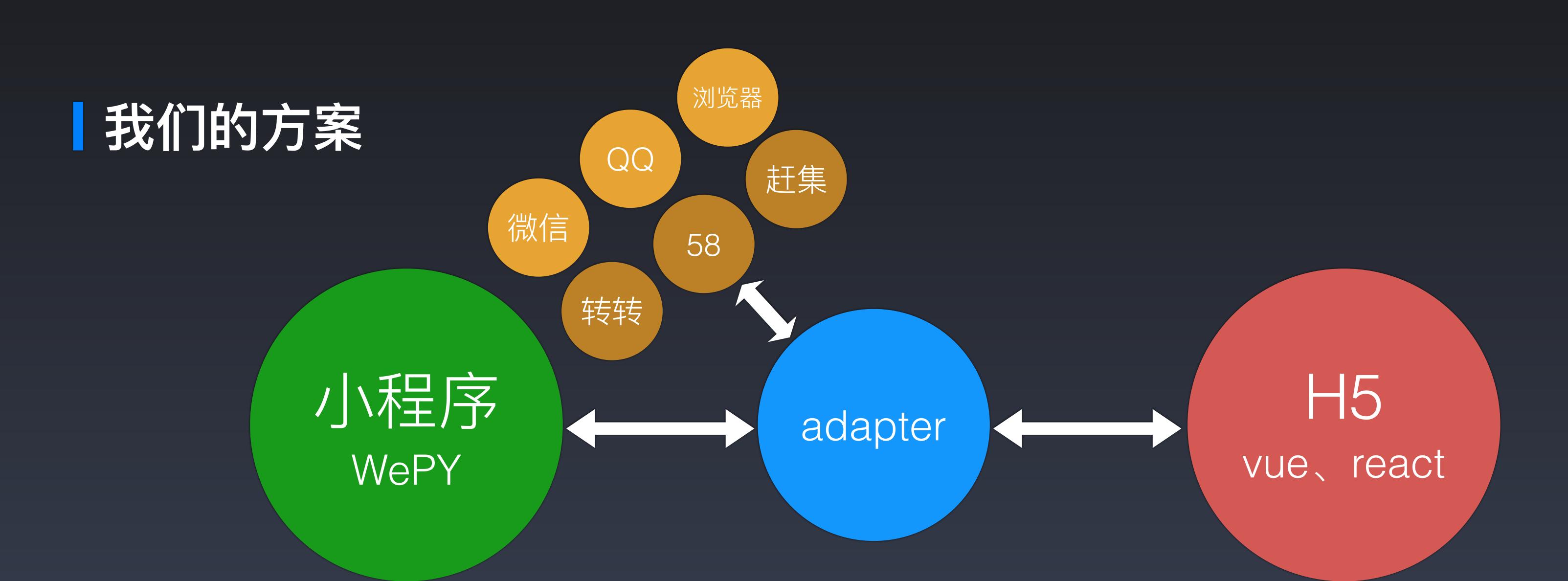
bad case

```
ジ掘金
```

```
if(app){
 //some sdk
}else if(wechat){
  //some sdk
}else if(gg){
  //some sdk
}else if(browser){
  //some sdk
}else if(miniProgram){
  //some sdk
```

- ●逻辑复杂
- 功能耦合
- 复用性差



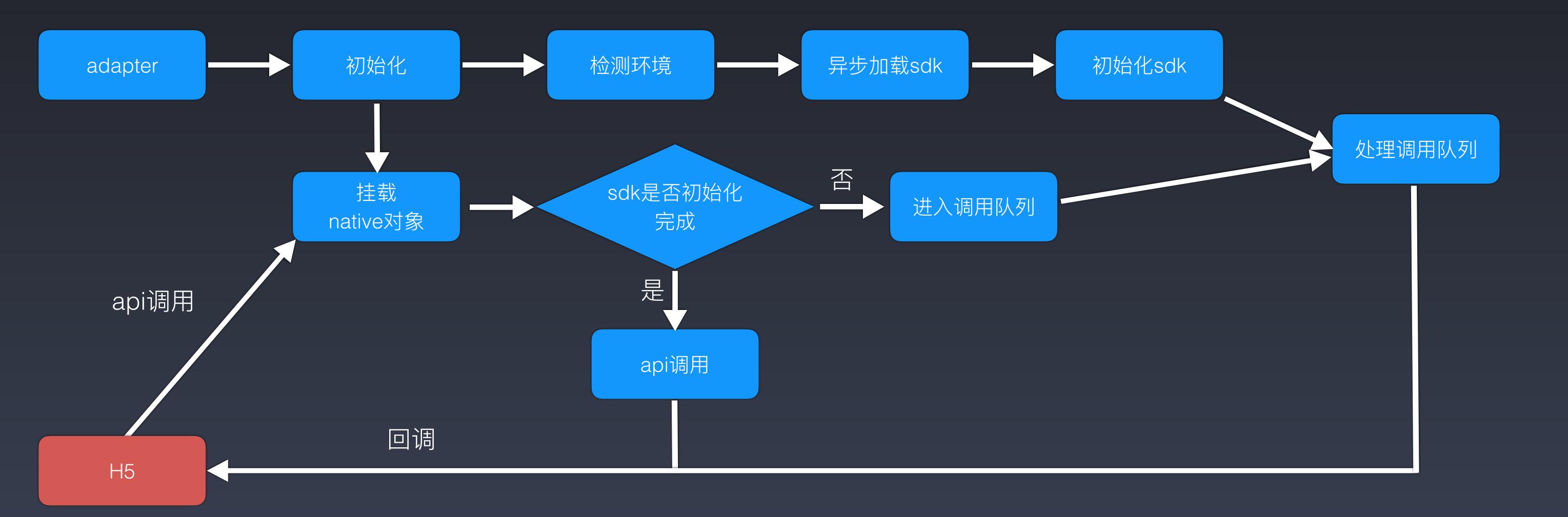


ジ掘金



adapter原理



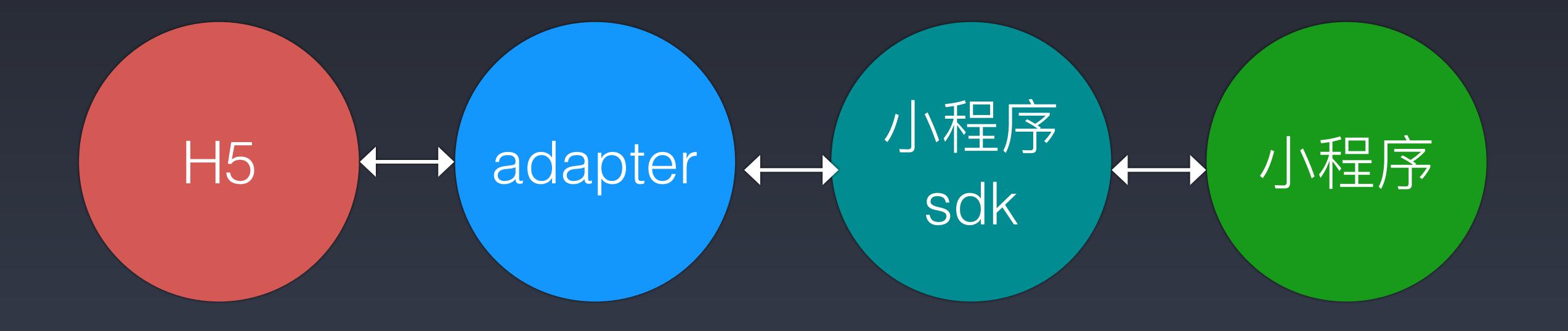




adapter

ジ掘金

- ●每个项目接入adapter
- ●根据环境加载小程序sdk
- ●对每个api进行统一封装





提纲

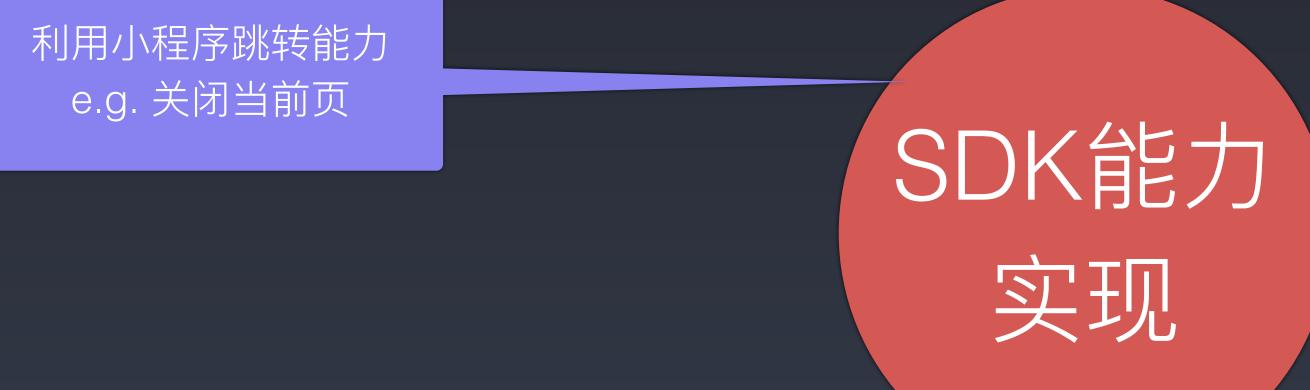


- 1 小程序技术演进
- 2 webview VS 小程序
- h5多端兼容方案
- 4 小程序sdk设计
 - webview常见问题



SDK的四种实现







小程序sdk实现·跳转能力



- wx.miniProgram.navigateTo
- wx.miniProgram.navigateBack
- wx.miniProgram.switchTab
- wx.miniProgram.reLaunch
- wx.miniProgram.redirectTo

```
例子:
const close = ()=>{
  wx.miniProgram.navigateBack({delta:1})
}
```



SDK四种实现



利用小程序跳转能力 e.g. 关闭当前页

SDK能力 实现 e.g. 扫码

利用微信sdk



打开小程序某个页面 e.g. 个人中心

对应小程序页面



native.skipToProfile()





小程序sdk实现·进入页面

- wx.miniProgram.navigateTo
- wx.miniProgram.navigateBack
- wx.miniProgram.switchTab
- wx.miniProgram.reLaunch
- wx.miniProgram.redirectTo

```
例子:

const skipToProfile = ()=>{
    wx.miniProgram.navigateTo({
        url: '/pages/profile'
    })
}
```





SDK四种实现



利用小程序跳转能力 e.g. 关闭当前页

SDK能力 实现 利用微信sdk e.g. 扫码



打开小程序某个页面 e.g. 个人中心 h5与小程序通信 e.g. 设置分享





设置分享



设置分享: H5





native.share(shareData)

api功能实现:设置分享



•方案一: postMessage

•方案二:设置和检测 webview 的 url 变化



设置分享: sdk

```
シ掘金
```

```
const share = (shareData)=>{
  let isSupport = checkSDKVersion() //检查是否支持postMessage
 if(isSupport){
   //支持postMessage
   wx.miniProgram.postMessage({
     data:{shareData:shareData}
 }else {
   //否则,拼接到url中
   let urlWithShareData = combine(location.href,shareData)
   location.replace(urlWidthShareData)
```



设置分享:小程序

```
シ掘金
```

```
<web-view src="{{url}}" bindmessage="onPostMessage"></web-view>
<script>
 export default {
    //....
   onPostMessage(e){
     this.shareData = e.detail.data
   onShareAppMessage(options){
     if (this.shareData) {
       //使用postMessage数据
      } else if (options.webViewUrl) {
       //使用url携带的数据
     } else {
       //使用默认数据
</script>
```



简化分享













简化分享











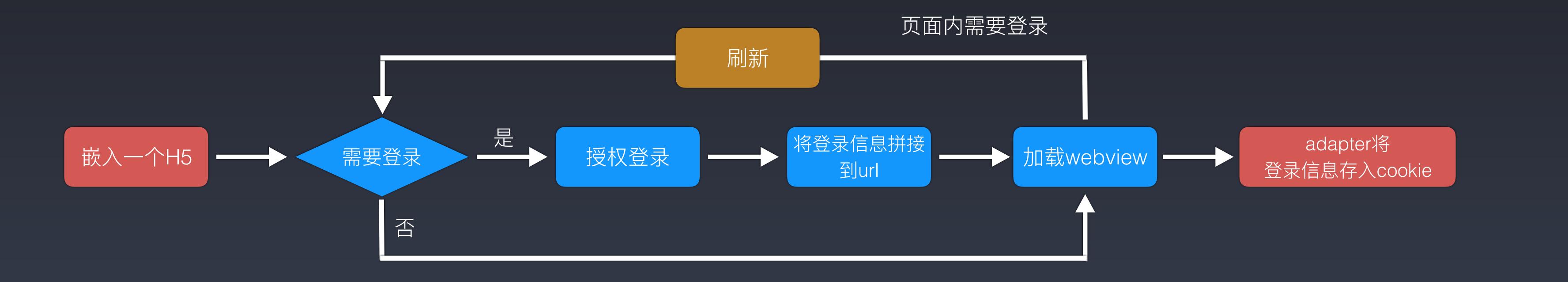






登录







提纲



- 1 小程序技术演进
- 2 webview VS 小程序
- h5多端兼容方案
- 4 小程序sdk设计

webview常见问题







区分环境



区分webview与微信浏览器





window.__wxjs_environment === 'miniprogram'不准确





wx.ready() 或 WeixinJSBridgeReady 事件中判断







支付: H5





native.skipToOrder(orderData)

支付: SDK

```
シ掘金
```

```
const skipToOrder = (options)=>{
 if(options.payOnly){
   //进入精简支付页
   wx.miniProgram.navigateTo({
     url: '/pages/pay'+combine(options)
  }else {
   //进入下单支付页
   wx.miniProgram.navigateTo({
     url:'/pages/buy'+combine(options)
```



支付:小程序

```
シ掘金
```

```
onLoad(options) {
  let payConfig = options;
  new Promise((resolve, reject)=>{
   wx.requestPayment({"timeStamp": payConfig.timeStamp...});
  .then((res)=>{
   if(util.isWaPage(url)){
     //跳转到小程序的支付成功页
     wxPromise.redirectTo({
       url
     });
    }else{
     // 刷新上一个webview页面地址
     routeParams.setBackFromData({
       webviewRedirectUrl: url
     wxPromise.navigateBack();
```









formld收集





落地页做成小程序原生页面



中途进入小程序页面(支付、IM等)





















左上角返回率70%+







左上角返回率70%+

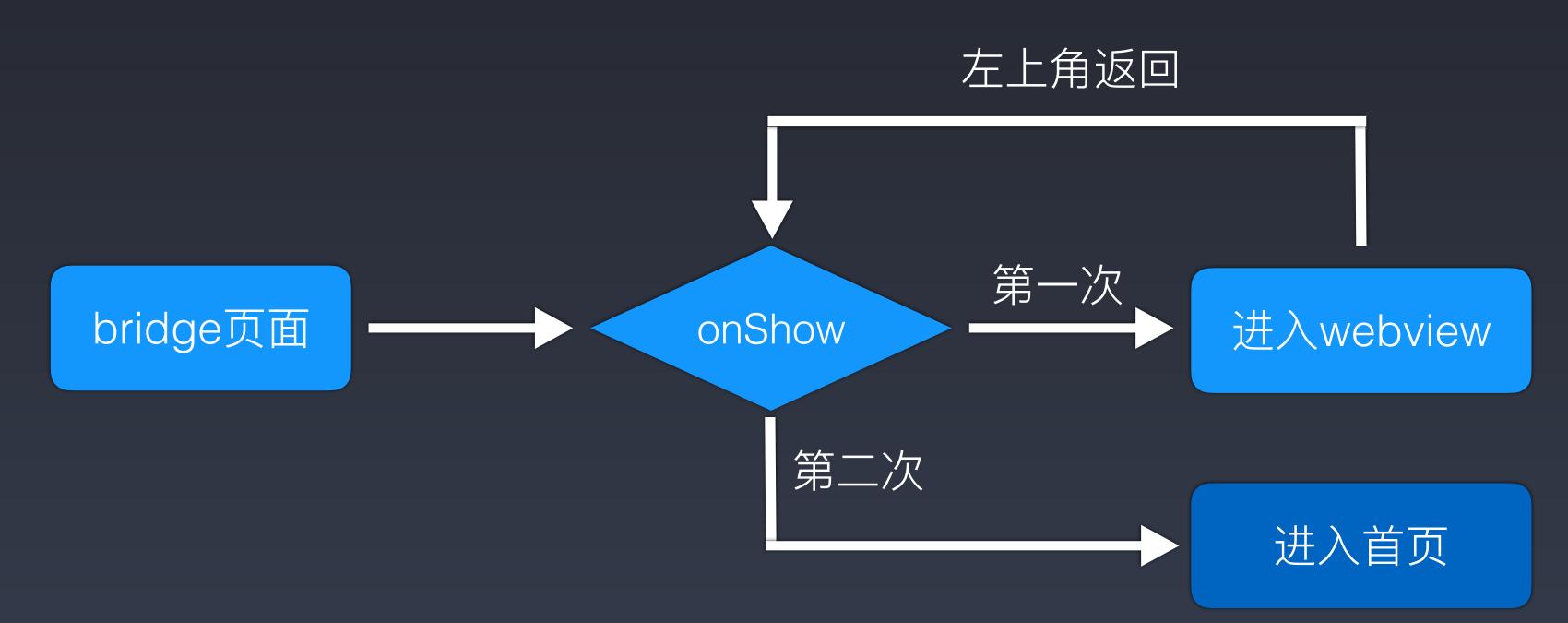






一 后续访问率10%+







要点回顾

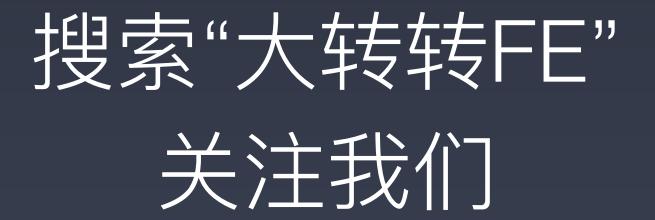


- 1 小程序技术演进
- webview VS 小程序
- h5多端兼容方案
- 4 小程序sdk设计
- 5 webview常见问题



沙掘金







シ掘金

Thankyou!





