

美团扫码付H5迁移小程序实践

主讲人：陈瑶

2018.4.21





陈瑶

2015.07 加入美团平台团购i版

2016.05 加入智能支付业务部

目前负责扫码付小程序相关业务

目标

对于目前已有的产品二维码扫码付（每日**百万订单、千万流量**），实现**从H5到微信小程序**的平滑迁移，为商户提供更好、更快的收银解决方案。

挑战

信任感

业务稳定性要求高

扫码付业务承载了百万的订单量，对服务的稳定性要求高，小程序是否能支撑扫码付业务尚待验证

成熟度

小程序技术栈推出时间短

小程序技术栈自成一体，底层实现黑盒，尚待验证

01 建立信任感



扫码付小程序第一版

做了一些粗糙的基础建设...

粗糙的基础建设

小程序JS异常监控



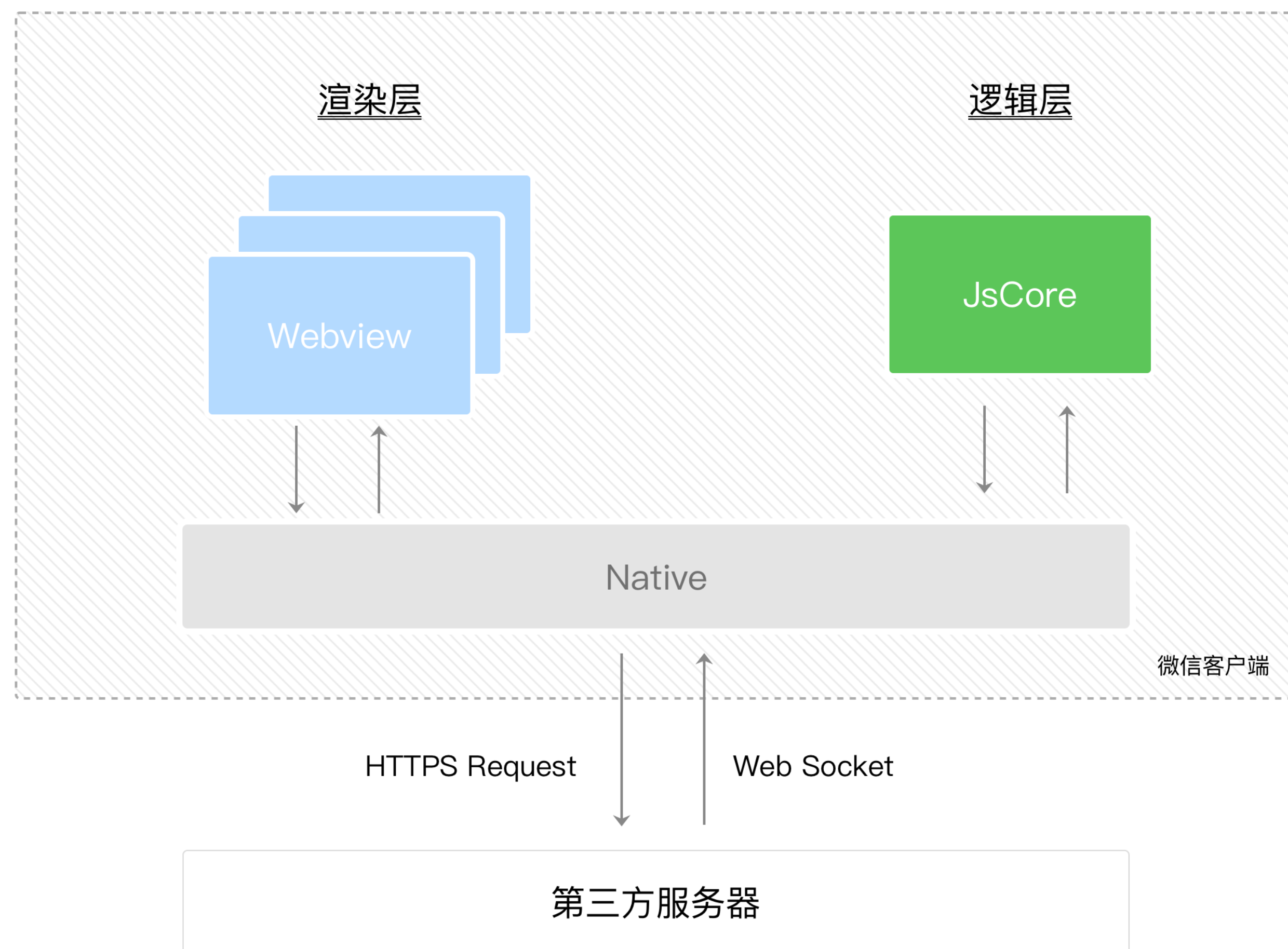
粗糙的基础建设

Webview中的js异常统计方法

1. window.onerror
2. try-catch

粗糙的基础建设

小程序的通信模型及异常上报



- 渲染层和逻辑层由2个线程管理
- 逻辑层采用JSCore做管控和安全管理
- Webview侧使用window.onerror
- 逻辑侧使用try-catch

粗糙的基础建设

小程序异常收集方式

```
App({
  onLaunch: function(options) {
    // Do something initial when launch.
  },
  onShow: function(options) {
    // Do something when show.
  },
  onError: function(msg) {
    this.pushErrorToServer({
      content: e,
      category: t,
      sec_category: n,
      level: "error"
    })
  },
  pushErrorToServer: function(errMsg) {
    //异常信息上传到服务器
    wx.request({...})
  },
})
```

逻辑层对各个生命周期方法包裹了try-catch，并同时在捕获到异常后，触发了App中的onError

粗糙的基础建设

问题：微信API（基础库）兼容性

原因

1. 基础库兼容性
2. 微信API兼容性

解决方案：

1. 增加兼容性判断
2. 减少易错API使用
3. 异常收集，易错API尽量try catch

粗糙的基础建设

问题： 微信框架兼容性

现象

1. 框架传参缺失
2. 框架传参错误

解决方案

1. 报错信息收集
2. 传参严格性校验
3. 用户日志记录
4. 及时上传日志到微信（添加好友处输入:up），并及时push微信解决

粗糙的基础建设

问题：接口可用性

原因

- 1.业务异常
- 2.网络失败（网络抖动或者弱网失败）

解决方案

- 1.保证前端页面友好
- 2.客户端报警反馈
- 3.重试机制
- 4.及时止损
- 5.第三方服务降级

粗糙的基础建设

小程序 VS H5页面性能

	H5	小程序
渲染能力	Top 50: 795ms Top 90: 1587ms Top 95: 2112ms Top 99: 5883ms	Top 50: 359ms Top 90: 979ms Top 95: 1263ms Top 99: 3159ms

页面DomReady

Onload > onReady

02 合作共赢

问题
发现了一些问题...

小程序唤起转化率(启动/扫码): 仅有93%

问题
发现了一些问题...

小程序ios端的首屏时间： 2.3s

小程序Android端的首屏时间： 2.9s

当我们扫小程序码时，发生了什么？

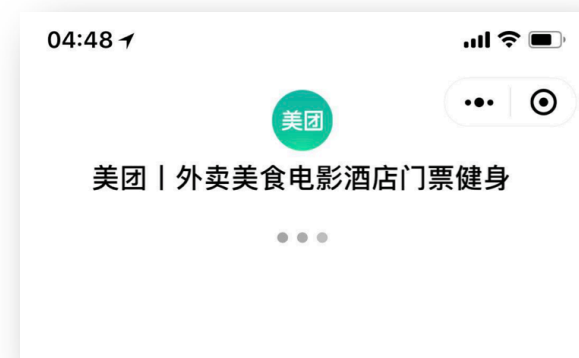
小程序：

扫码

小程序 url
校验

基本信息
获取

展示启动页面



小程序是否
已缓存

否

下载

是

更新

否

是否为最新
版本

是

初始化页面

小程序启动

加载小程序代码包

结论： 下载或者更新耗时长

验证黑盒机制

包代码大小对比

业务	扫码唤起成功率	包代码大小
点评点餐小程序	97.5%	1.8M
扫码付小程序	93%	3M

提升扫码调起率

减小代码包大小

A

分包加载

推动微信上线分包加载

B

减小主包和分包

主包尽量做到空包

C

图片CDN化

占用代码包约50%的图片走CDN

拆离Base64 的图片

提升扫码调起率

优化后效果:

业务	总包大小	扫码调起成功率	分包大小
优化前	3122kb	93%	2040kb
优化后	2951kb	96%	361kb

提升扫码调起率

问题

每一次的代码发布，调起转化率当天会下降？

提升扫码调起率

更新机制



实际实现

- 6小时内未使用过，有新版本则同步更新使用；
- 5个版本以内做增量更新；

问题

更新机制

期望：老版本不强制更新，异步更新新版本



微信侧：性能优先策略和版本优先



业务侧方案：降低发版频率，有一定的发版周期

提升扫码调起率

如何从96%到100%？

去掉7天不使用清理缓存机制

提升扫码调起率

如何从96%到100%？

纯分包模式（无主包）

提升扫码调起率

如何从96%到100%？

分包发布弱依赖

提升扫码调起率

如何从96%到100%？

客户端白屏增加动效

还需要些时间...

问题

企业级开发

- 自动化发布（各个业务线发版影响其他业务线）
- 单一环境（开发者工具、体验版）
- 项目成员不可拓展（开发人员+体验人员 < 90）
- request合法域名数量、业务域名不支持泛域名
- 上线强依赖（先审核再发布）
- 自动化测试

回顾

小程序还有一段很长的时间路要走...

Q&A