



携程下一代无线App架构设计

陈浩然

Agenda

- App工程解耦和组件化架构
- 移动网络服务通道治理和优化
- 无线研发支持平台

携程无线现状



在线



呼叫中心



无线端

2015

从OTA转变为MTA
(Mobile Travel Agency)



如何实现传统互联网业务无线化?

Conway's Law

“Any organization that designs a system will produce whose structure is a copy of the organization's communication structure.”

组织解耦

多个业务团队

- 自行决定设计、开发和发布周期

基础框架团队

- 提供基础框架（开发框架、功能SDK等）和服务（系统管理、网络、存储等）

技术解耦

解耦传统n-层架构方式

- SOA化，提供Microservice

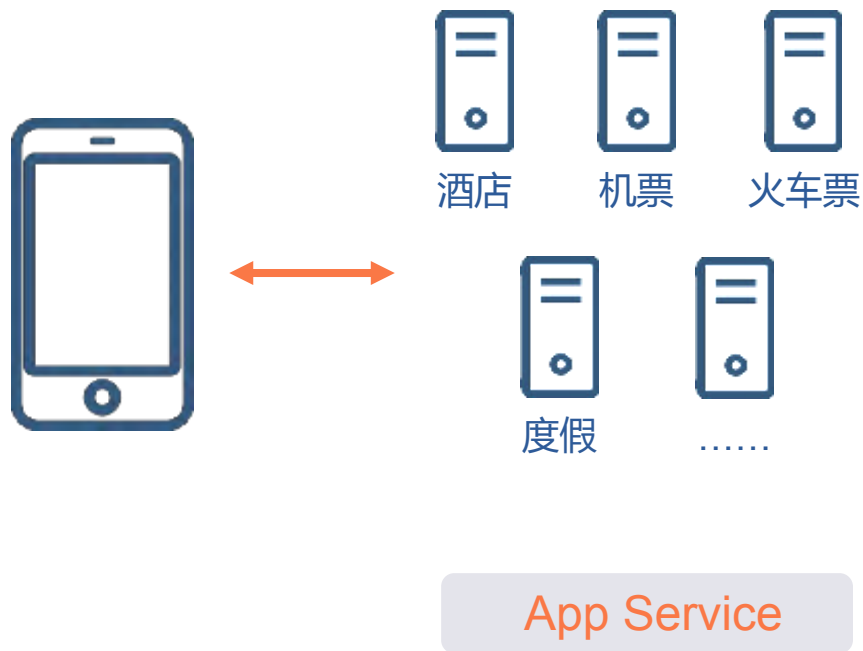
解耦数据

- 提供数据服务，实现数据模型的封装、管理和访问

解耦App端开发模式

- 支持组件化/插件化开发
- 升级组件化集成打包

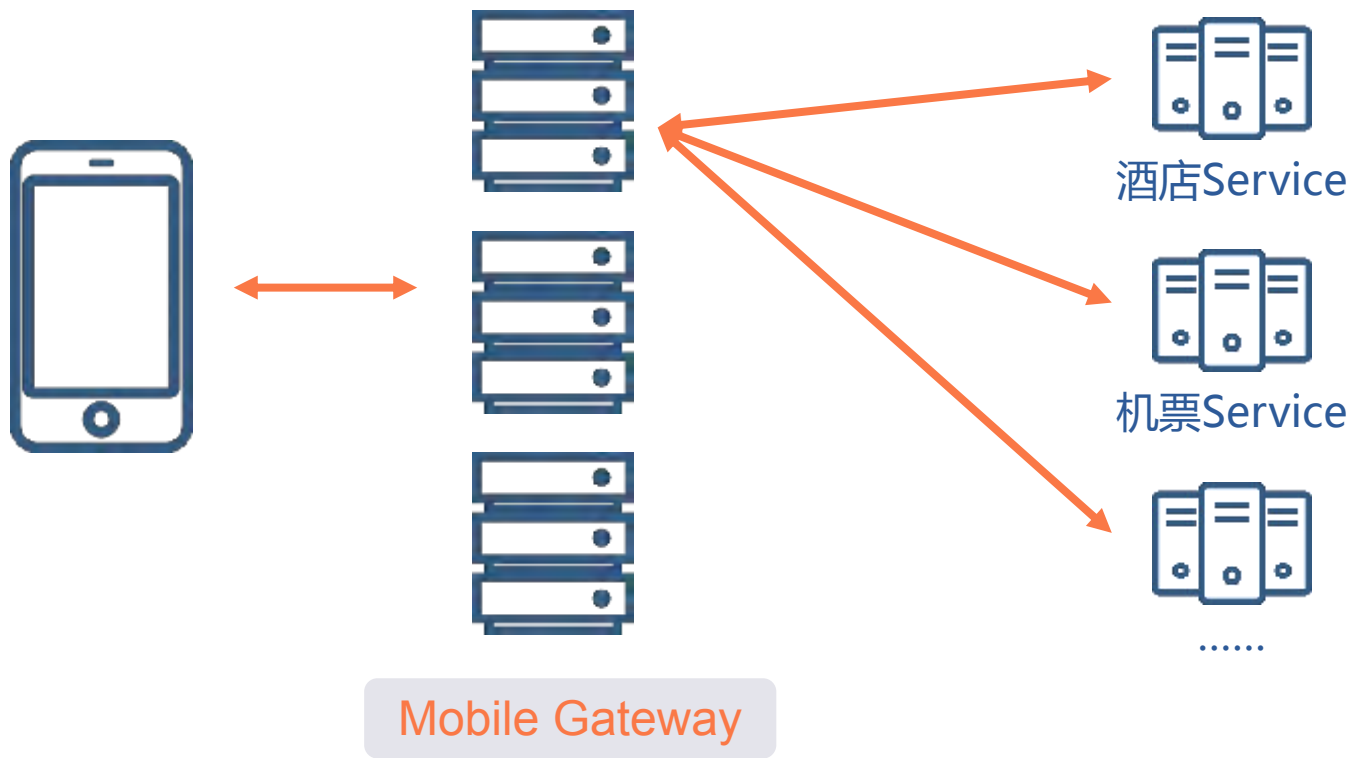
服务端



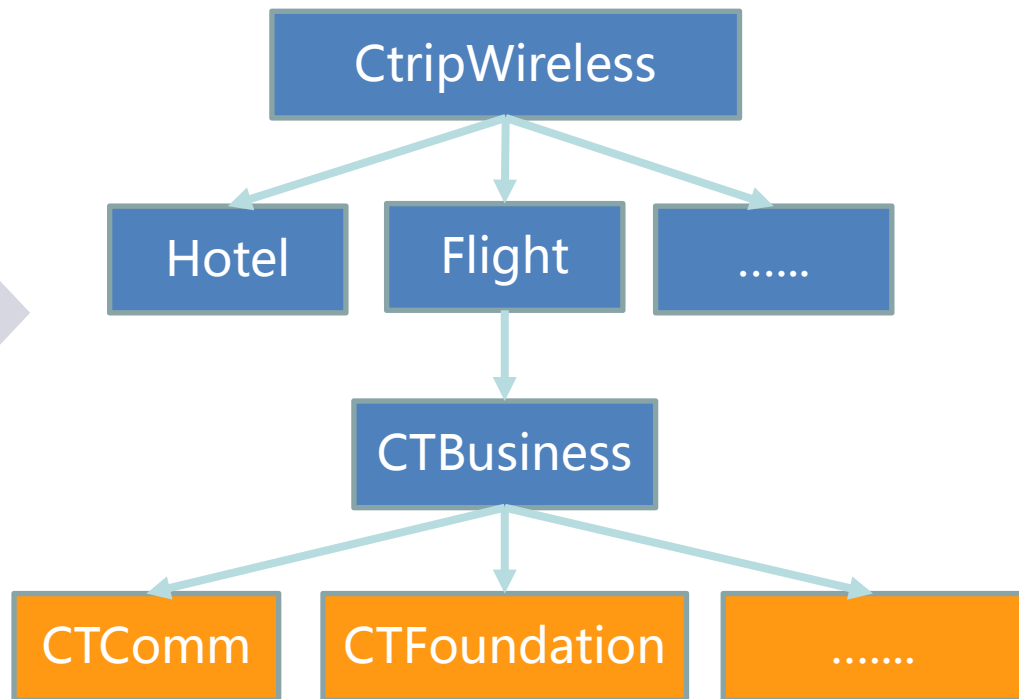
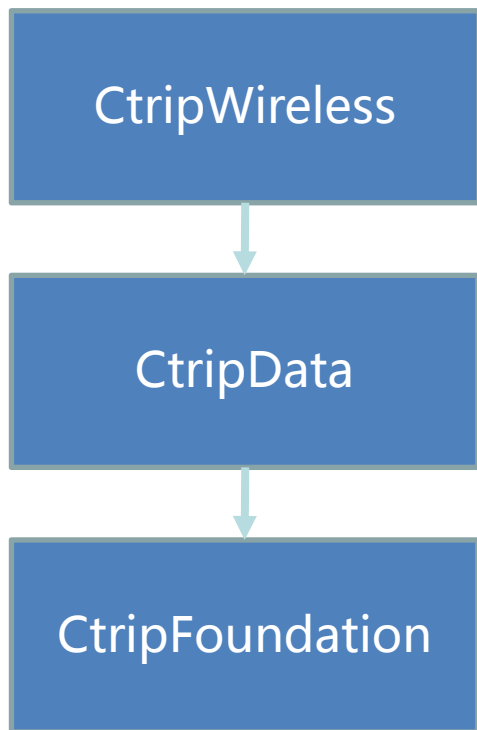
曾经的问题:

- 进程耦合
- 缺少负载均衡
- 越少监控
- 缺少熔断
- 安全性风险
-

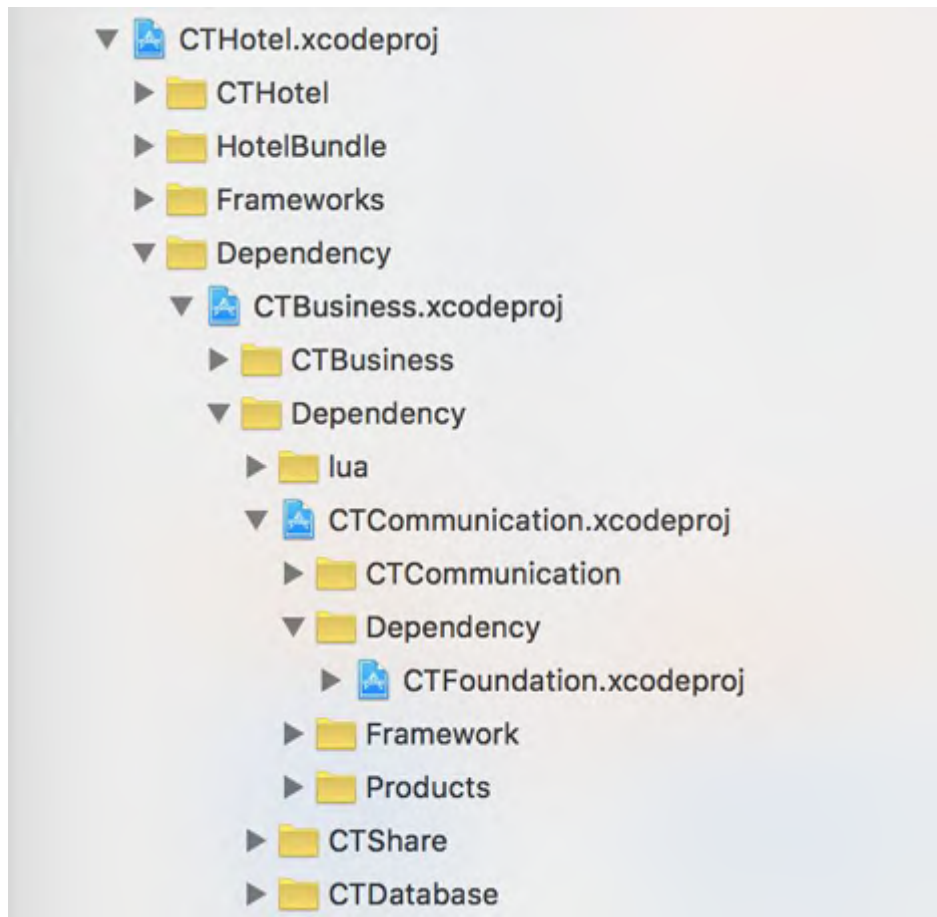
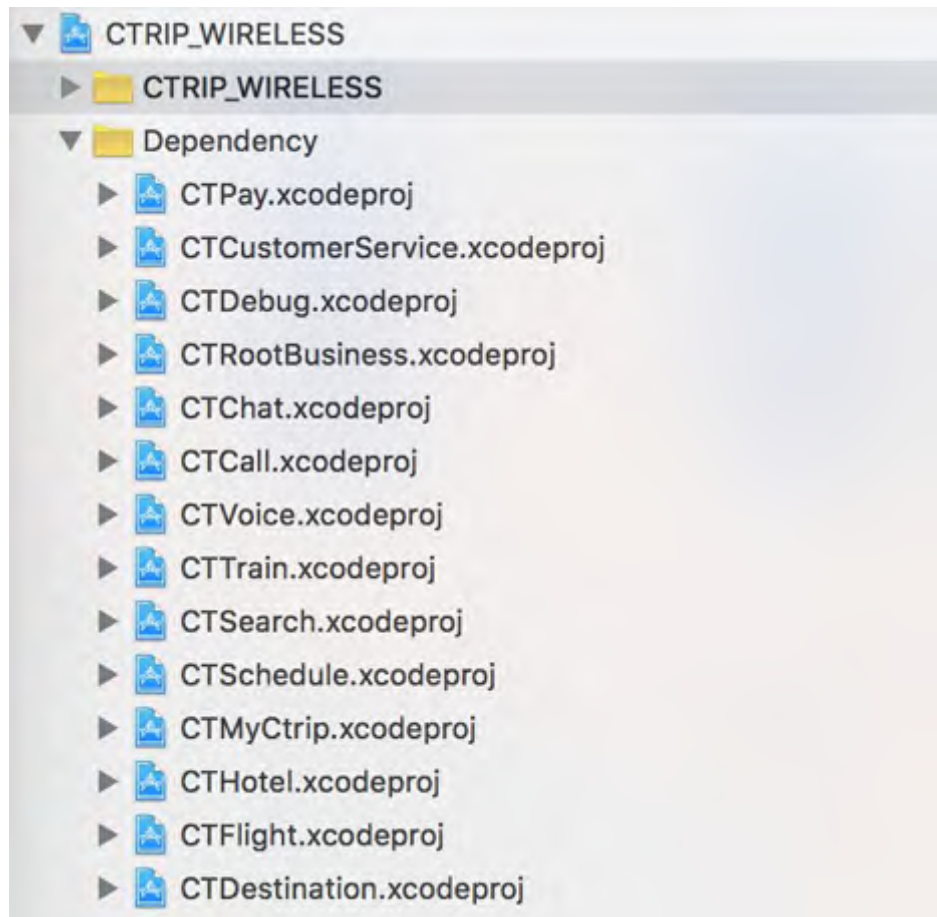
服务端



App端



App工程配置解耦



App框架组件化

核心基础功能SDK化

- 通讯、定位、Hybrid、数据库、登录、分享、基础库等SDK
- 可以直接提供给其他BU独立App使用

公用业务功能组件化

- 地图、日历、城市、图片、通讯录等公共组件
- 减少各BU重复开发工作量

App业务容器化

隔离性和稳定性：

- Bundle间相互独立，单个模块的故障影响范围在内部，不会导致整体App的crash

扩展性和伸缩性：

- 具备一定程度的运行时和动态部署能力，支持Bundle的动态加载和Hotfix

性能：

- 独立统计Bundle的性能指标

插件化和动态加载框架 - DynamicAPK

更少的迁移成本

- 无需做任何activity/fragment/resource的proxy实现

提升启动速度

- 启动时仅加载必需的模块，Android 5.0以下优于MultiDex

按需下载和加载任意功能模块

- 包含代码和资源模块动态加载
- 支持Hotfix

CtripMobile / DynamicAPK

Watch 163

Unstar 1,349

Fork 454

Code

Issues 30

Pull requests 2

Wiki

Pulse

Graphs

Settings

Solution to implement multi apk dynamic loading and hot fixing for Android App. (实现Android App多apk插件化和动态加载，支持资源分包和热修复) — Edit

Hybrid框架

Hybrid与Hybrid功能互通

- 尽可能共用Native组件：地图、图片浏览等

离线包模式

- 设计时即支持离线包模式，降低资源加载时间
- 因App Size问题部分低频业务使用直连模式

查分增量更新

- 离线包更新时仅更新发生变化的资源文件
- 使用7z压缩降低更新包大小，比zip压缩小30%

Agenda

- App工程解耦和组件化架构
- 移动网络服务通道治理和优化
- 无线研发支持平台

移动网络通道

Native服务网络请求

Hybrid服务网络请求

Push服务网络请求

IM服务网络请求

用户行为和性能日志上传网络请求

网络通道治理思路

减少连接次数

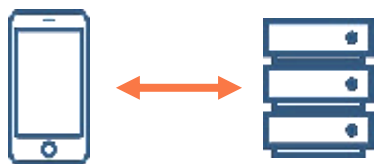
避免DNS劫持和内容劫持

通过减少发送次数和压缩来减少流量

更安全和稳定

支持多数据中心多活的网络通道灵活调度

Native服务网络请求



- 使用TCP长连接+短连接实现网络服务
- 根据网络状况2G/3G/4G/WIFI进行调优参数
- 根据连接/读/写不同阶段使用重试机制



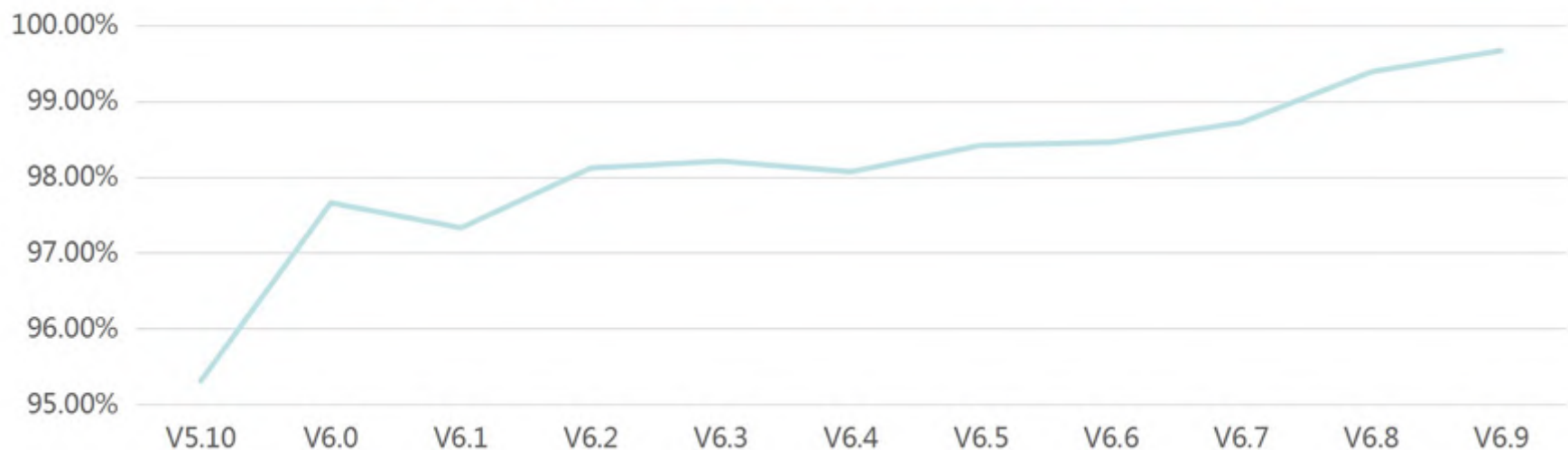
- 使用IP列表避免DNS解析失败或者劫持
- 根据网络延迟选择服务端IP (Ping值估算RTT)



- 使用ProtocolBuffer+Gzip减少Payload

Native服务网络请求

App端到端网络服务成功率达到 **99.87%**



Hybrid服务网络请求

常见问题

- DNS劫持
- 内容劫持

解决方案

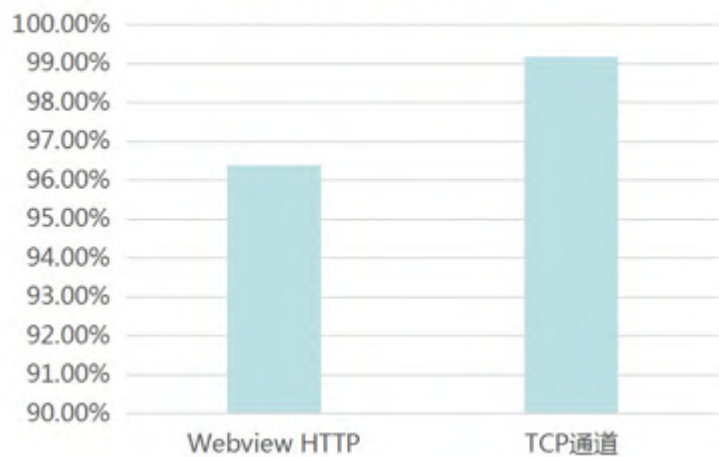
- 拦截Webview的请求实现转发
- Hybrid框架层实现转发

Hybrid服务网络请求

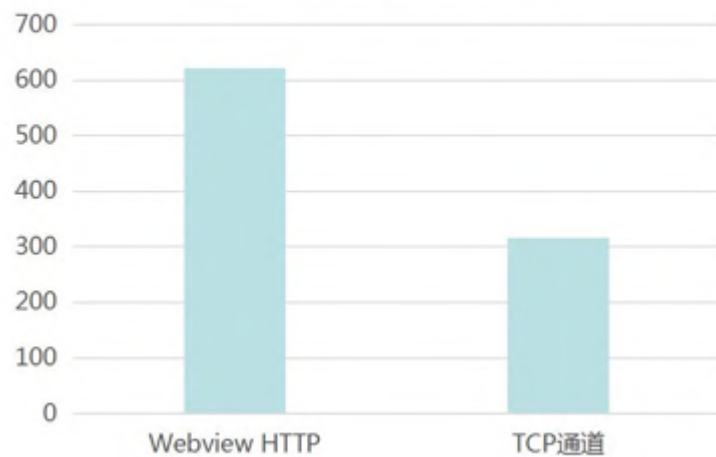
最终方案

- Hybrid框架层通过TCP通道进行业务服务转发
- 页面等静态资源通过MAA等加速产品实现防劫持和加速

网络请求成功率



网络请求平均耗时 (ms)



其他通道治理

尽量基于TCP长连接实现网络服务

- Push
- IM
- 用户行为和性能日志上传

优先使用成熟服务

- 国内：MAA等防劫持和加速产品
- 海外：Akaima TCP/HTTP专有加速通道

Agenda

- App工程解耦和组件化架构
- 移动网络服务通道治理和优化
- 无线研发支持平台

无线研发支持平台

为什么需要支持平台？

- App端到端性能管理
- 用户行为统计
- 持续集成
- 配置中心
-

集成平台

- 支持Bundle独立打包，实现持续集成，提高集成效率

自动化测试平台

- 实现兼容性、性能和稳定性自动化测试
- Record-Replay方式的功能自动化测试

发布平台

- 支持Hybrid和Hotfix发布

运营平台

- 管理App配置

监控平台

- App性能数据监控

持续集成

主页 / Mobile CD

主页

我的任务


历史包

PD打包

插件列表


技术支持

Android Build




Create Browse

IOS Build



Create Browse

Hybrid Build



Create Browse

渠道打包




Create Browse

Native Hotfix



Browse

Market Build



Create Browse

持续集成

主页 / PD打包

主页 我的任务 历史包 PD打包 插件列表 技术支持 vchr陈浩然

Q筛选

rel/6.14(6.14)

Config Build New Package

Build ID	SIT	Native	Hybrid	Status	Automation	Create Time	Owner	Log	Download	Comment	Detail	Action
2766653	SITSUPER	生产	生产	Success		03-14 13:48	gqxue					
2766524	SIT2	生产	生产	Success		03-13 18:22	gqxue					
2766311	SIT1.0	测试	UAT	Success		03-11 19:53	gqxue					
2765957	12345	生产	生产	Success		03-09 14:39	eqzhu					
2765955	2345	测试	UAT	Success		03-09 14:36	eqzhu					
2765954	1234	测试	UAT	编译失败		03-09 14:35	eqzhu					
2765953	1234	测试	UAT	编译失败		03-09 14:34	eqzhu					

左侧列表:

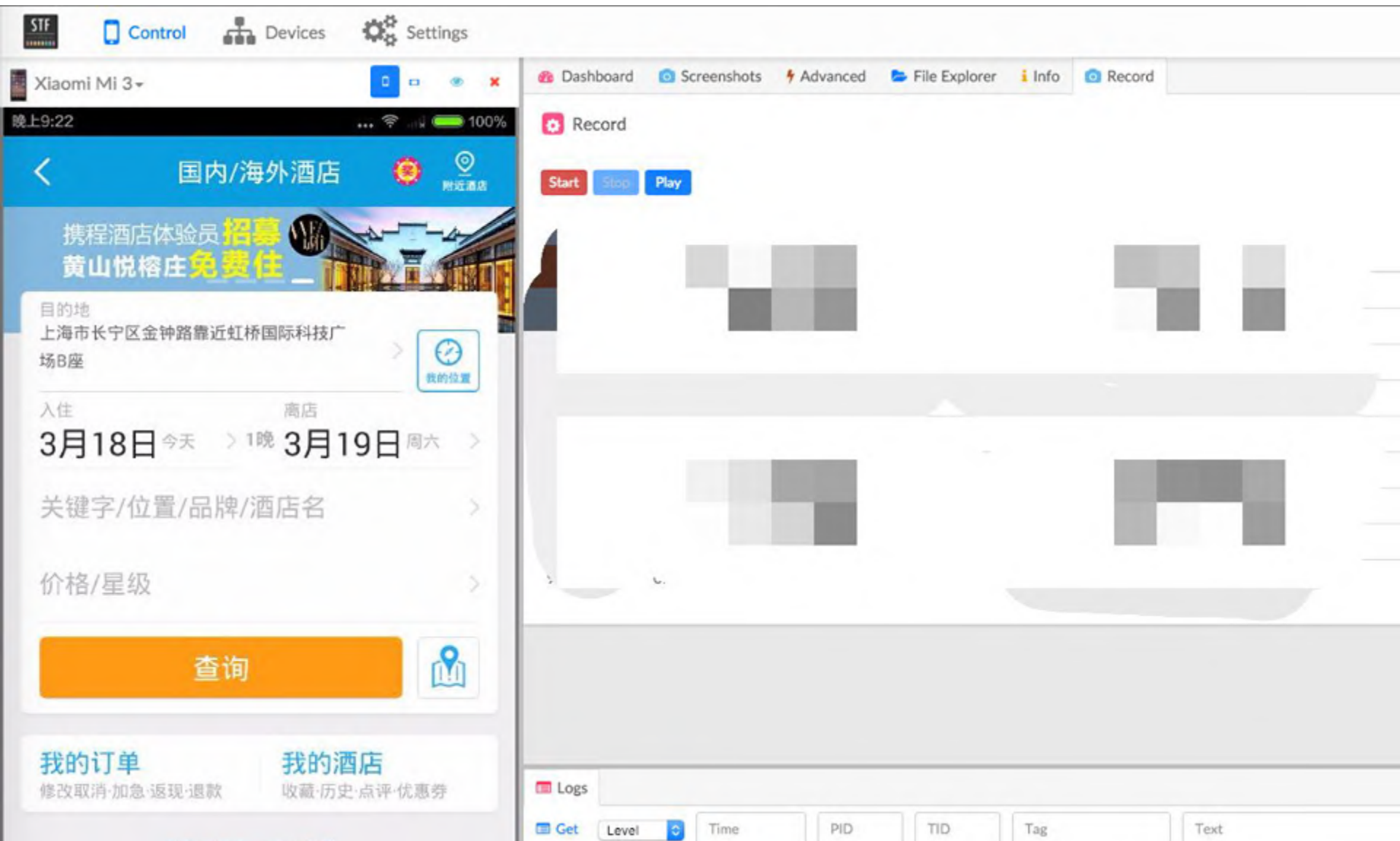
- V:6.15 rel/6.15 03-18 顾艳娟
- V:6.14.2 rel/6.14 03-17 汤伟明
- V:6.15 rel/6.15 03-10 张昌霖
- V:6.12 rel/6.12_an 03-10 汤伟明
- V:6.14 rel/6.14 03-09 朱显庆
- V:6.14 rel/6.14 03-09 顾艳娟
- V:6.14 rel/6.14 03-04 刘李丰

自动化测试平台

基本思路

- 基于STF (Smartphone Test Farm)管理机器
- Android定制STF实现Record-Replay功能
- iOS基于Instruments实现

自动化测试平台



监控平台

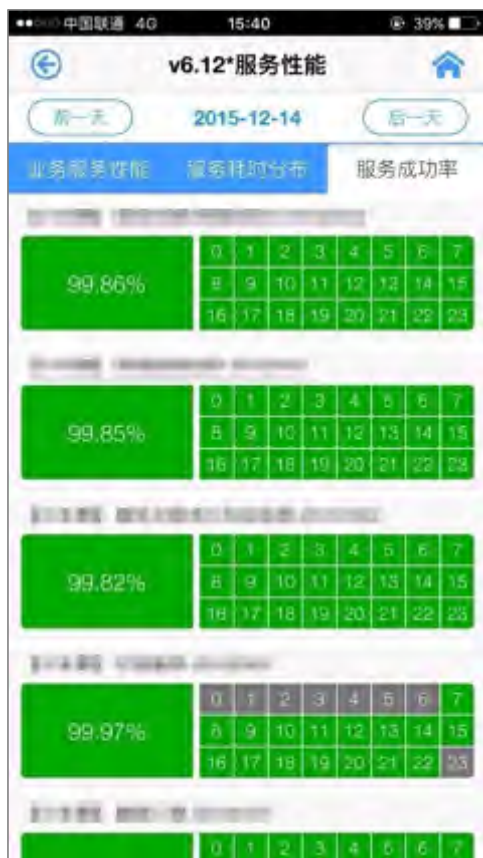
采集性能数据

- 自建：日志SDK采集日志，上传至服务端，日志消息经Kafka存入HDFS（RCFile格式），Hive用于查询
- 第三方：OneAPM、听云等工具

制定性能指标

- 网络性能：网络服务成功率、平均耗时、耗时分布
- 定位：获取经纬度成功率、城市定位成功率
- 启动时间、内存、流量等指标
- 多种纬度：系统、App版本、网络状况、位置等

如何展示端到端数据



Thanks

技术公众号：CtripMobile

