

携程无线持续交付平台

赵辛贵



赵辛贵

- 08年开始移动开发
- 13年入职携程
- 目前负责移动 App(Native/RN/Hybrid)框架相关工作
- 关注性能质量和开发效率

目录

CONTENTS

- 1 背景介绍
- 2 集成平台
- 3 测试平台
- 4 发布平台
- 5 运营平台

背景介绍—数据

组织架构

- 300+ 开发/测试
- 37个业务团队
- 开发人员分散

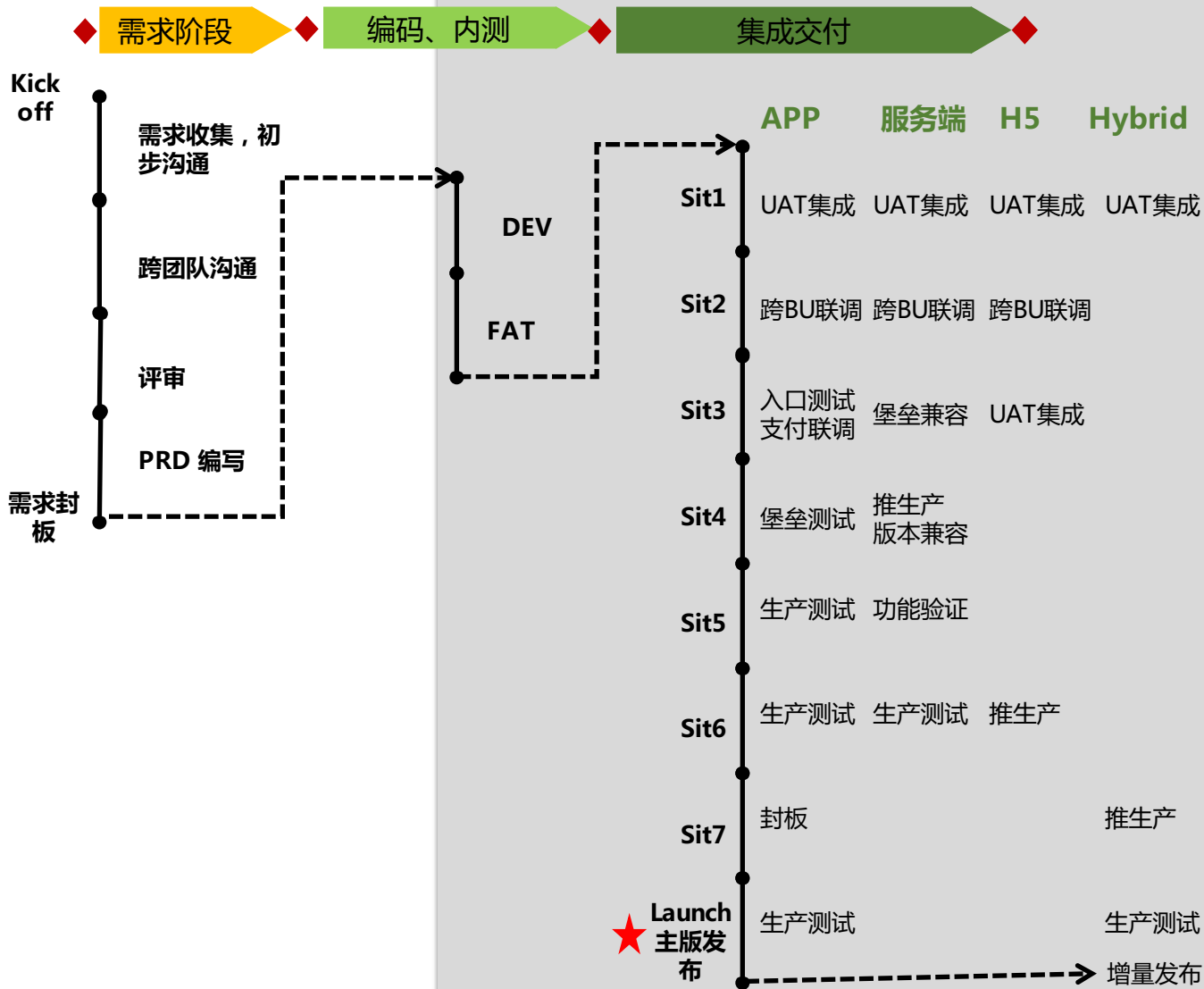
工程规模

- 30+ Native模块
- 30+ React Native模块
- 60+ H5 Hybrid模块
- iOS:代码110W行
- Android:代码272W行

集成发布

- 4000+次App打包/每月
- 3000+次Hybrid&RN发布/每月
- 1.5月一个版本
- 3天集成测试时间

背景介绍—开发流程



背景介绍—集成平台演化

1. 原始形态



2. 1.0版本-减少人工干预

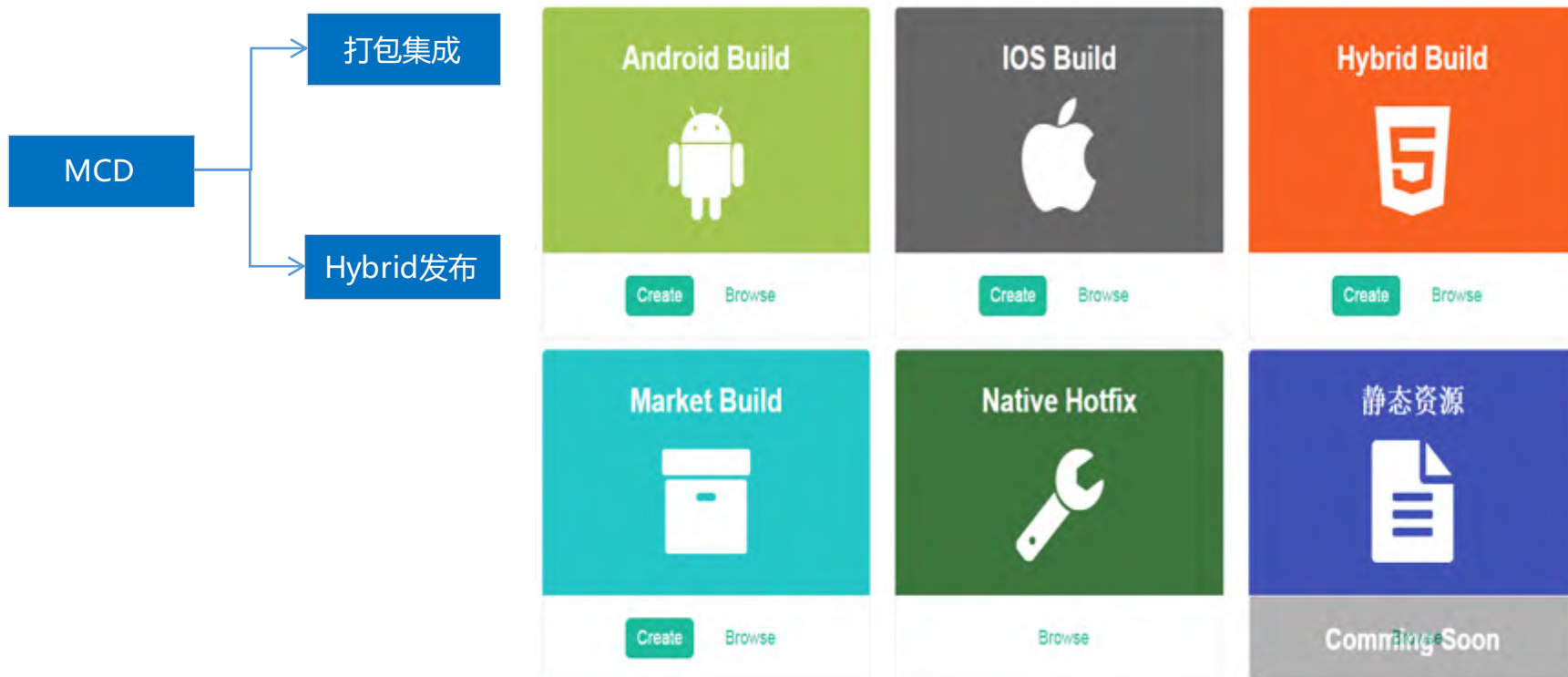


- MCD, Mobile Continuous Delivery的简称
- 基于Jenkins的任务和调度管理



背景介绍—集成平台演化

3. 老的发布系统下线， MCD整合Hybrid发布功能



背景介绍—集成平台演化

4. 移动开发生命周期支撑系统(2.0)



集成平台—功能与定位

用户

- 开发、测试、项目经理(PMO)

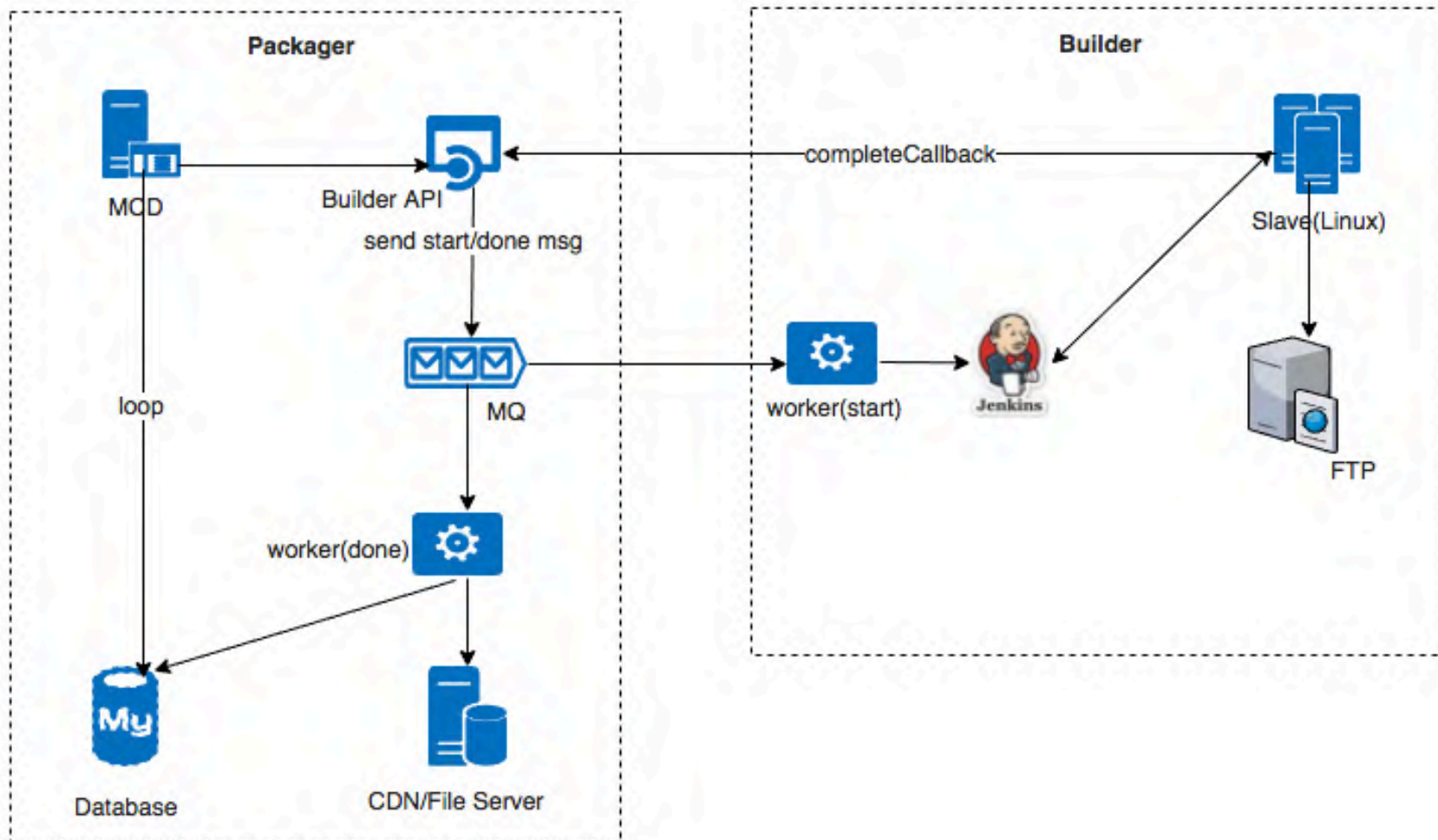
目标

- 确保能够快速及时的打出测试、生产包
- 提高产品交付效率，降低沟通成本

功能

- 打包、持续集成、测试包管理、模块管理

集成平台—整体架构



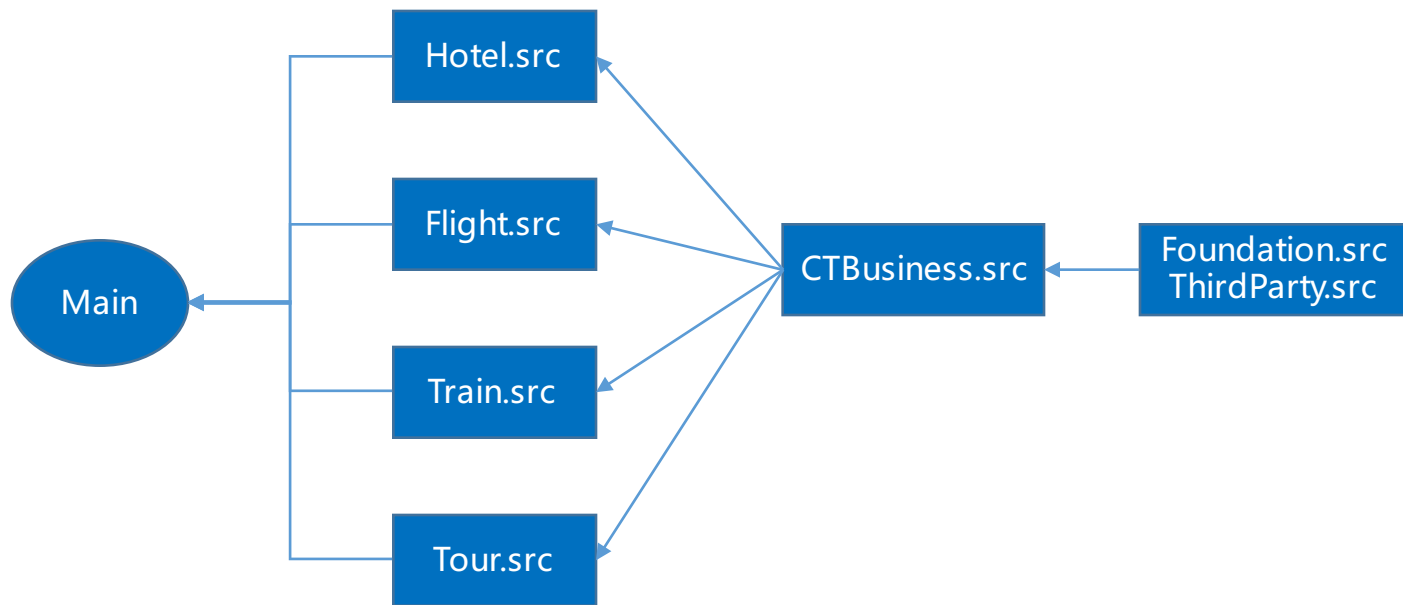
集成平台—早期开发流程



说明：

1. 所有代码在一个Git仓库
2. 各个业务之间基于源代码依赖
3. SIT集成时间大约需要1周时间

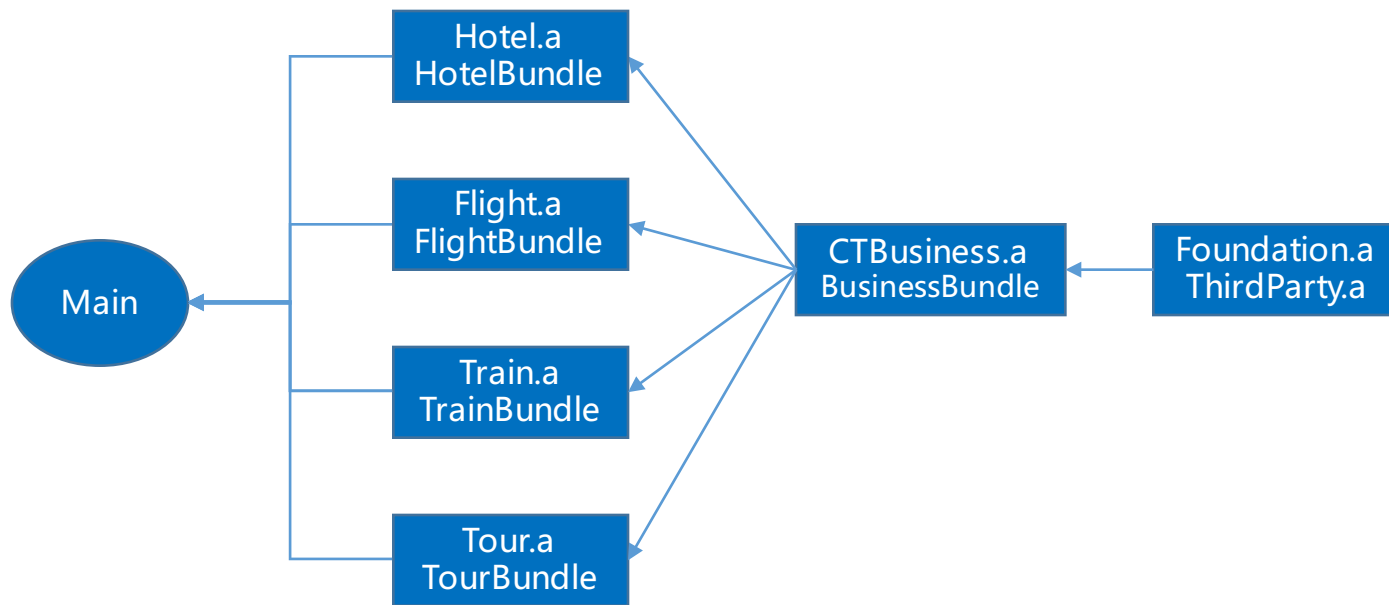
集成平台—源码依赖的工程结构



痛点：

1. 每次都要从代码仓库下载全量代码
2. 开发/打包编译时间长，MacPro需要8-10min编译，普通iMac需要20-30min编译
3. 相互影响，编译容易失败

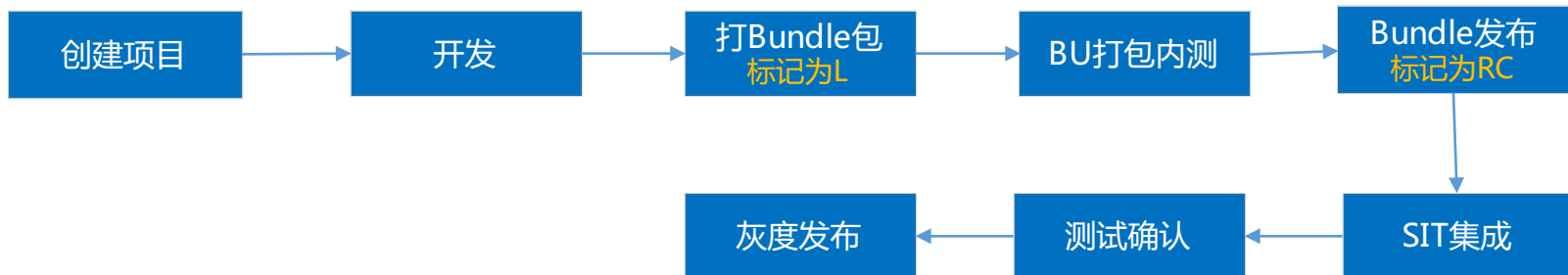
集成平台—Bundle依赖的工程结构



说明：

1. 引入中间产物Bundle(静态库/AAR)依赖
2. 发布流程发生变化

集成平台—集成发布流程优化



1. 功能开发完成，打Bundle包，默认发布标记为L(Latest)
2. 开发阶段，业务之间依赖彼此的Latest包
3. 测试通过，Bundle标记为RC(Release Candidate)
4. SIT集成阶段打包依赖RC包

集成平台—安装包Bundle列表

安装包由各个业务模块Bundle组合而成，每个包都可查看当前Bundle列表

Ctrip_V7.3.0_17042809_test_2990224_zcyfb.apk Bundle 详情

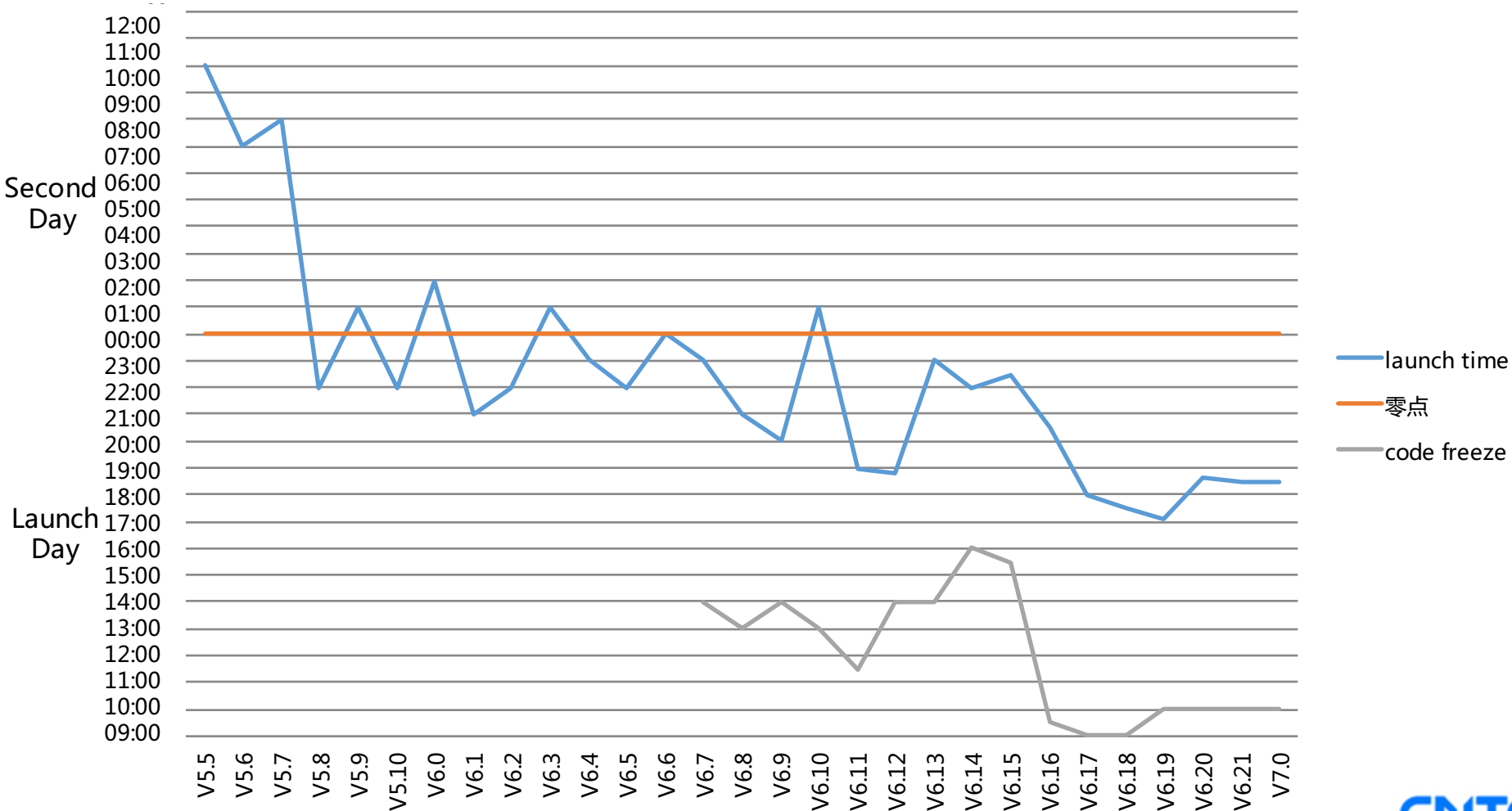
#	模块	版本	是否最新	是否RC	发布人	发布时间	查看	发布
1	CTTour	7.3.0_2017.04.24.123810	L		王智勇	04/24 12:38:12	查看	发布
2	CTHotelComment	7.3.0_2017.04.21.144343	L		王智勇	04/21 14:43:51	查看	发布
3	CTReactNative	7.3.0_2017.04.17.151525	L	RC	王智勇	04/17 15:15:26	查看	
4	CtripMain	7.3.0_2017.04.17.151519	L	RC	王智勇	04/17 15:15:20	查看	
5	CTDebug	7.3.0_2017.04.13.093757	L		王智勇	04/13 09:38:02	查看	发布
6	CTScheduleBusiness	7.3.0_2017.04.13.060222	L	RC	王智勇	04/13 06:02:25	查看	
7	CTSchedule	7.3.0_2017.04.13.055727	L	RC	王智勇	04/13 05:57:33	查看	
8	CTDestinationMain	7.3.0_2017.04.13.022325	L	RC	王智勇	04/13 02:23:38	查看	
9	CTCall	7.3.0_2017.04.13.005118	L	RC	王智勇	04/13 00:51:19	查看	
10	CTTrainBusiness	7.3.0_2017.04.13.002014	L	RC	王智勇	04/13 00:20:15	查看	

集成平台—改造完成之后

1. 集成Build时间从10min下降到3min(MacPro环境)
2. 版本集成时间从1周降低到3天
3. 开发Build时间缩短，开发效率提升

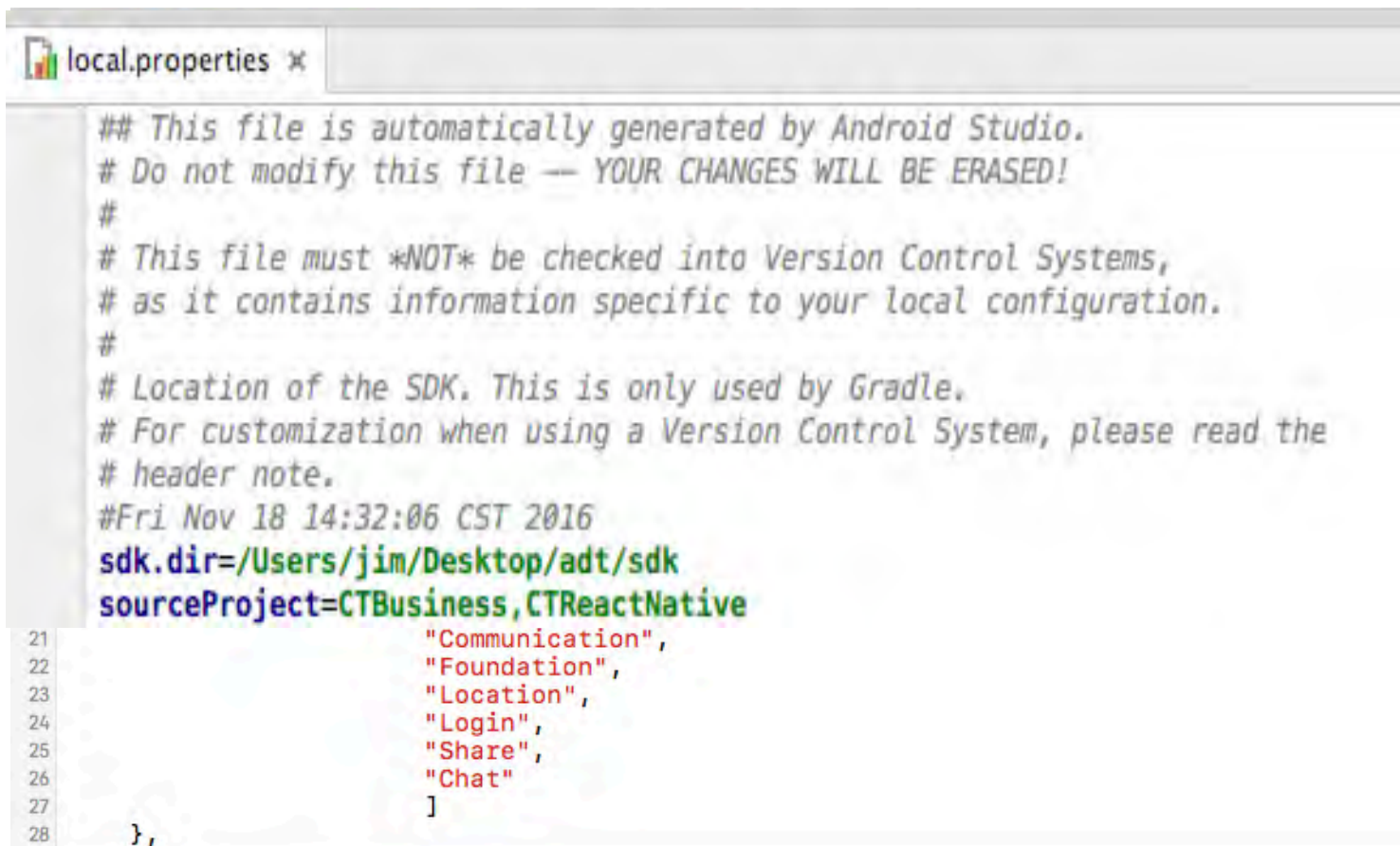
step	时间	opt	action	content	message
1	2017-04-28T15:27:17		send2MQ	success	
2	2017-04-28T15:27:17		post_to_jenkins		
3	2017-04-28T15:27:17		pull_code	start	拉取代码
4	2017-04-28T15:28:12		pull_code	end	
5	2017-04-28T15:28:12		package	start	构建集成包
6	2017-04-28T15:29:10		package	end	bundle依赖打包
7	2017-04-28T15:29:10		archive	start	压缩构建产物
8	2017-04-28T15:29:23		archive	end	
9	2017-04-28T15:29:23		upload	start	上传构建产物至存储服务器
10	2017-04-28T15:29:29		upload	end	
11	2017-04-28T15:29:30		send2MQ	finished	{'buildid': '2990648', 'app_type': 'ANDROID_INTEGRATE', 'notifymessage': 'build completed'}
12	2016-03-03T15:14:05		buildcount	value	1
13	2016-03-03T15:14:05		upload	info	1 file(s), size is 135480746

集成平台一版本发布日Launch时间



集成平台—Bundle调试问题

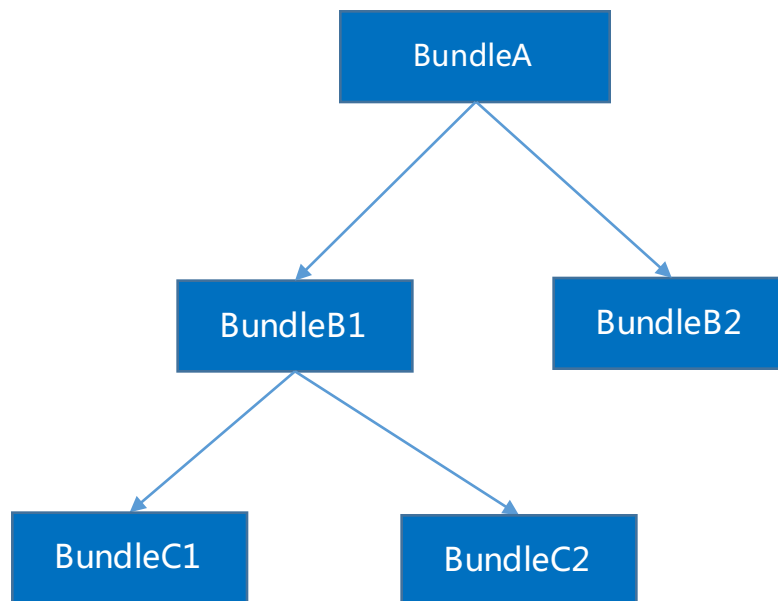
开发脚本，提供简单配置，一键切换功能



```
local.properties x
## This file is automatically generated by Android Studio.
# Do not modify this file — YOUR CHANGES WILL BE ERASED!
#
# This file must *NOT* be checked into Version Control Systems,
# as it contains information specific to your local configuration.
#
# Location of the SDK. This is only used by Gradle.
# For customization when using a Version Control System, please read the
# header note.
#Fri Nov 18 14:32:06 CST 2016
sdk.dir=/Users/jim/Desktop/adt/sdk
sourceProject=CTBusiness,CTReactNative
21         "Communication",
22         "Foundation",
23         "Location",
24         "Login",
25         "Share",
26         "Chat"
27     ]
28 },
```

集成平台—Bundle依赖

1. 脚本中配置当前Bundle的依赖关系
2. Bundle build完成，发布之前，需要检测是否正确打出包
3. 被依赖的Bundle build完成之后，自动递归build依赖它的Bundle



集成平台—Bundle发布控制

1. Size控制

- Bundle build完成，自动计算Size
- 准确计算方法：资源文件zip大小+可执行程序新增zip大小*AppStore加壳系数
- 超过Bundle设定的大小，无法发布RC

2. Build支持Bundle灵活选择配置

- 某个Bundle功能有问题时候，可以选择使用该bundle的稳定版本

3. 发布权限控制

- 发布日固定时间，禁止当前发布版本的所有Bundle RC发布权限

集成平台—其它问题

1. 打包慢

- 缓存一切能缓存的内容
- iOS业务bundle比较大(包含4个CPU架构), 定时从文件服务器同步
- 打包依赖的离线包下载之后, 本地cache

2. 定制包--Bundle的继承

- 发布完成之后, 创建新版本, 或者创建新的定制包分支, 可以从已发布版本继承其Bundle

测试平台—功能与定位

用户

- 开发、测试

目标

- 给业务测试人员提供自动化测试平台
- 自动化运行性能测试
- 提升产品的质量和稳定性

功能

- 代码扫描、单元测试、自动化测试、性能测试等

测试平台—静态代码扫描与单元测试

1. 代码行数和重复率：Sonar
2. 静态代码分析：Infer
3. Unit Test：XCTest/JUnit /Robolectric/Mockito

模块	所属BU	Sonar		Infer		Unit Test			详情	状态	Infer	操作
		代码行数	重复率	问题数	NPE数	Case数	成功率	覆盖率				
携程APP	酒店	108328(-)	3.80%(-)	10(-)	7(-)	-(-)	-(-)	-(-)	i	✓	✓	重置 ▶ 重跑
携程APP	酒店	2659(2659)	11.70%(11.70%)	0(0)	0(0)	-(-)	-(-)	-(-)	i	✓	✓	重置 ▶ 重跑
携程APP	酒店	65207(59710) ↑	2.40%(2.40%)	4(0) ↑	4(0) ↑	-(-)	-(-)	-(-)	i	✓	✓	重置 ▶ 重跑
携程APP	酒店	9519(-)	2.30%(-)	1(-)	1(-)	-(-)	-(-)	-(-)	i	✓	✓	重置 ▶ 重跑
















测试平台—自动化测试

1. 基于Appium二次开发
2. 支持业务部门自动化测试人员运行脚本与报表导出

#	项目类型	项目名称	Tags	创建人	部门	平台	操作	结果	编辑
9	instruments	IOSCaseDriver	ios	陈明	基础业务研发部	ios	运行	查看	≡ ×
10	uiautomator	CaseDriver	android	陈明	基础业务研发部	android	运行	查看	≡ ×
11	uiautomator	hotel_team1_autoCases	android	陈明	酒店研发部	android	运行	查看	≡ ×
12	uiautomator	TestMonkeyRunner	android	陈明	基础业务研发部	android	运行	查看	≡ ×

测试平台—性能测试

1. 对于发布前的版本进行monkey测试
2. 统计内存、CPU、电量、流量、启动时间、crash等信息，并和历史版本数据对比
3. 整合到集成平台，可配置成打包完成自动运行

Build ID	SIT	Env	Status	Automation	Build Info	Log	Download
2762590	SIT2.11	生产 生产	Success		02-17 16:00 mcd	 log	
2762589	SIT2.11	测试 生产	Success		02-17 16:00 mcd	 log	
2762588	SIT2.11	测试 UAT	Success		02-17 16:00 mcd	 log	
2762571	SIT2.10	生产 生产	Success		02-17 15:00 mcd	 log	
2762570	SIT2.10	测试 生产	Success		02-17 15:00 mcd	 log	

测试平台—设备租赁管理

1. 基于STF二次开发，管理设备
2. 最大化利用机房已有设备

设备	操作系统	分辨率	状态
 OPPO X9007	Android4.3	1080x1920	租用
 三星 SM-G9280	Android6.0.1	1440x2560	租用
 魅族 M355	Android4.4.4	1080x1800	租用
 华为 MT7-TL10	Android4.4.2	1080x1920	luonj租用中
 三星 SM-N9009	Android5.0	1080x1920	租用

发布平台—功能与定位

用户

- 开发、测试

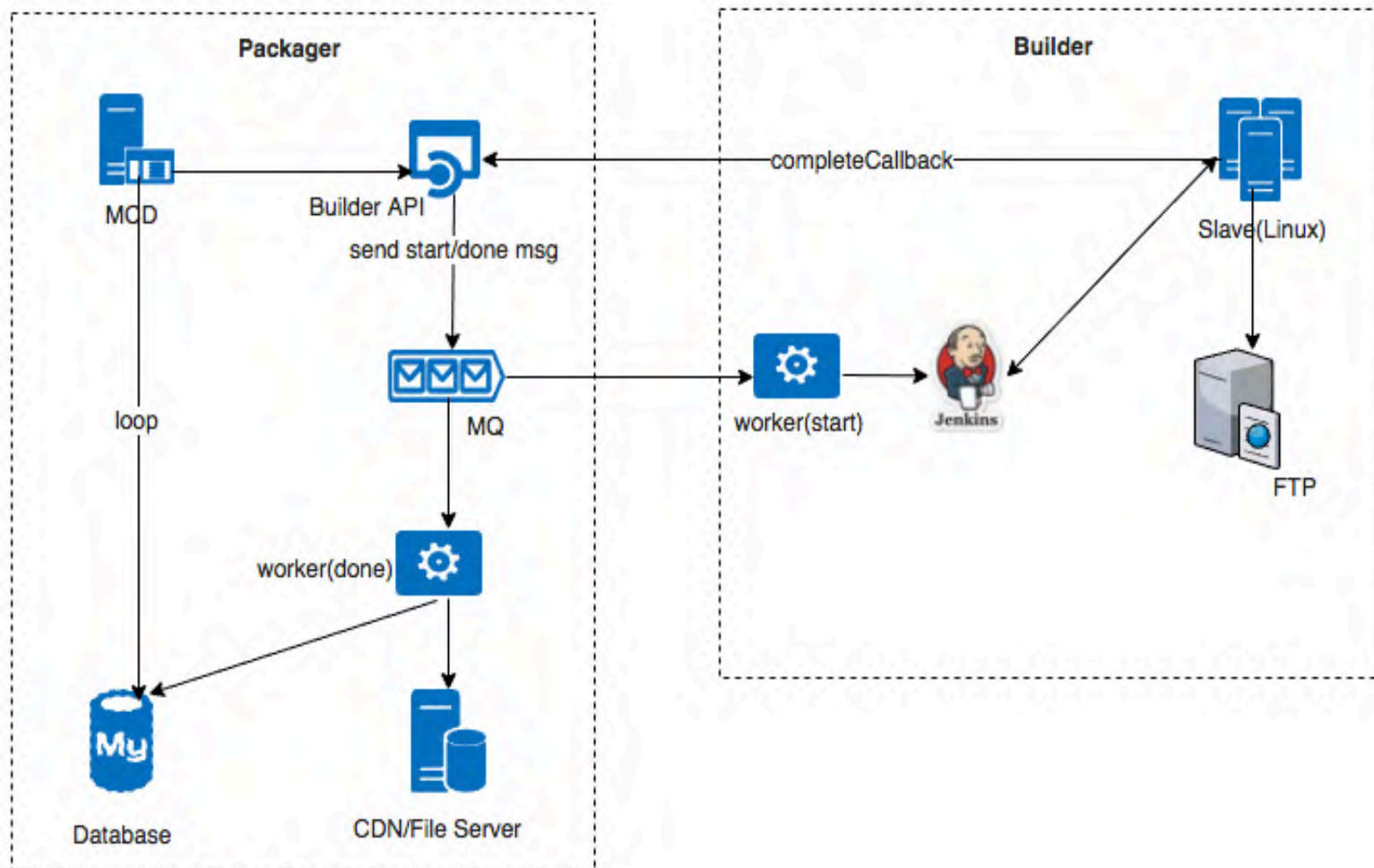
目标

- 提高产品交付效率
- 确保发布内容及时正确的更新到用户App中

功能

- H5 Hybrid—离线包动态更新
- React Native—离线包动态更新
- Hotfix—动态发布
- Android Bundle—动态发布

发布平台—整体架构



发布平台—发布流程设计



发布单：确认代码分支和发布App版本

版本：对发布单做一次发布

版本名称	版本号	版本创建人	FAT	UAT	PRD	UAT测试通过	发布
1.0.8	20170427200728	[Redacted]	✓发布成功 100%下载中	✓发布成功 100%下载中	✓发布成功 100%下载中	Yes UAT	发布

Commit Id(默认最新)	e56b157edddaec1595143c337bc40364f4687d7c
committer	wu_xx
提交描述	更新crm依赖 7.3Change-Id: I2b6fbab949a769c956f8e45273bbe764f0adcde6
版本名称	您可输入想要的版本名称(48字符以内)
版本备注	description

通知邮箱	message_dev ×
------	---------------

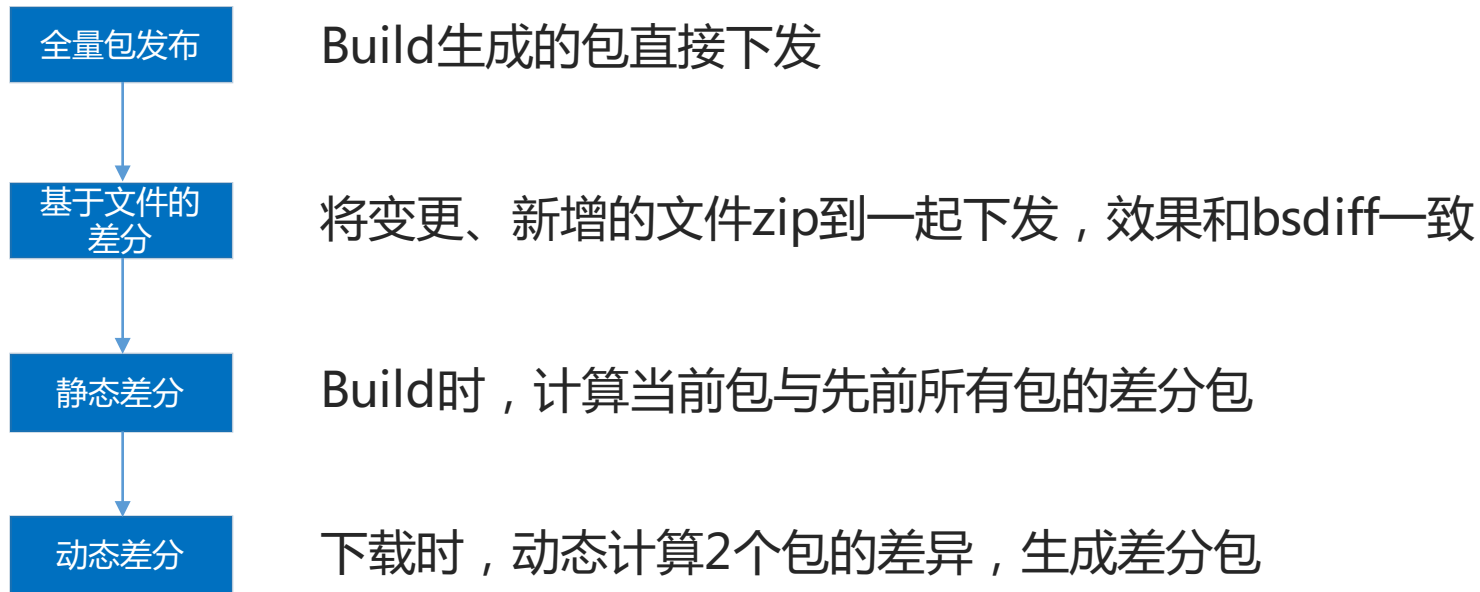
发布平台—数据监控



#	版本	平台	是否差分	Mobile Id	下载数	状态
1	7.3.2	android	是	158572	394318	正常
2	7.3.2	android	是	158572	949	下载增量失败
3	7.3.2	android	是	158572	723	写下载文件错误
4	7.3.2	android	是	158572	1	解压下载增量文件失败
5	7.3.2	android	是	158572	80	签名错误
6	7.3.2	android	是	158572	4	从Asset复制原始文件失败

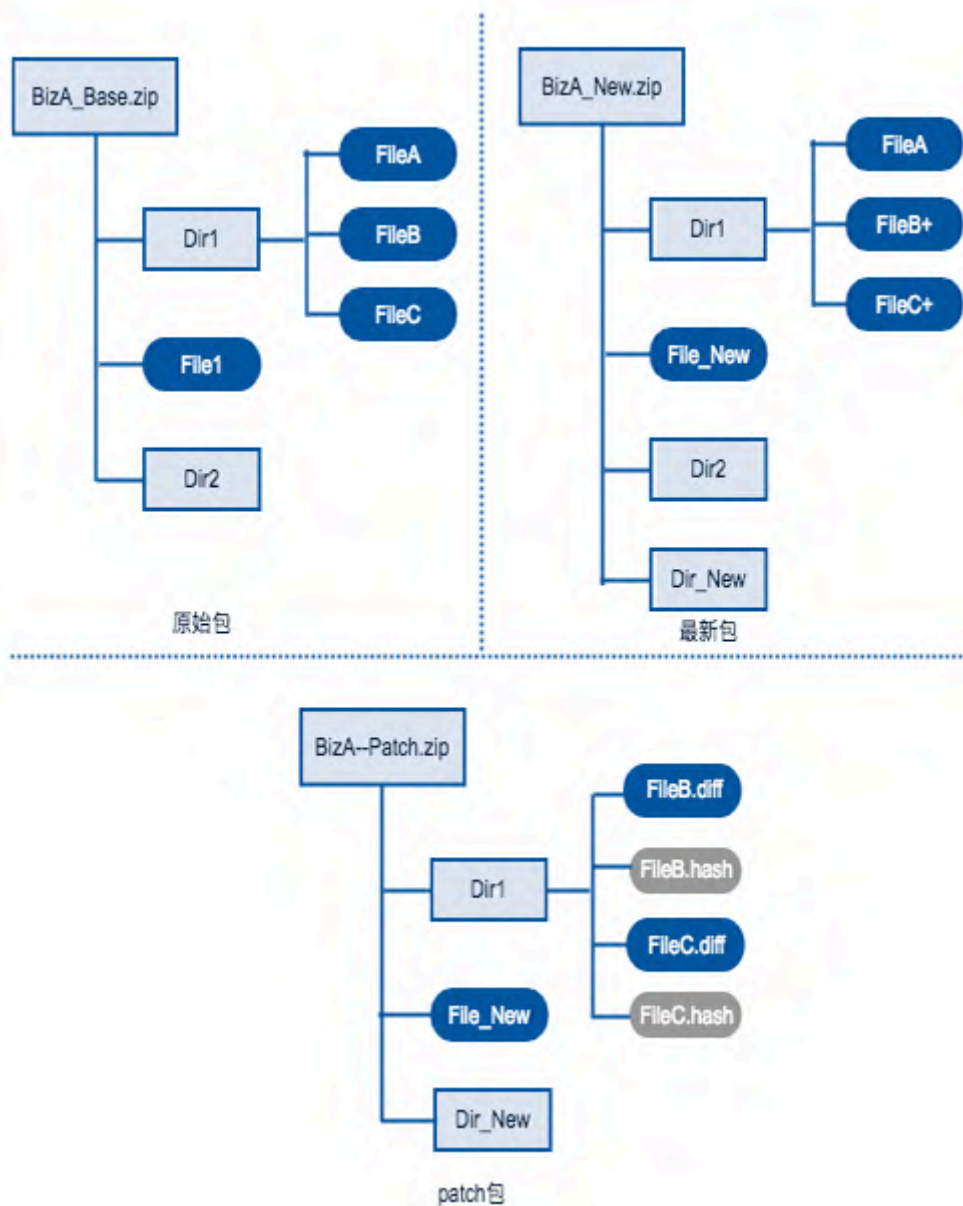
下载&安装整体成功率99.5%+

发布平台—资源包大小持续优化



发布平台—改进的差分算法

1. 先解压
2. 文件夹比对，diff其中的变化文件
3. zip文件夹中的diff文件



发布平台——组实测数据

App中初始包(KB)	下发离线包(KB)	bsdiff结果(KB)	优化效果(KB)
87.2	88.3	57.1	14.4
188.2	184.7	140.1	9.41
375.3	369.9	276.4	2.9
1021.5	970.57	611.8	10.08

发布1周之后的版本，12个离线包可以下载，总共Size 148 KB

发布平台—依赖管理

1. 数字表示优先级，数字越小，优先级越高
2. 框架类模块优先级为0，业务模块优先级为100
3. 相同优先级，根据发布时间下载
4. 整体作为列表，串行下载

发布平台—安全问题

1. 下载列表中返回文件RSA签名
2. 离线包下载完成后，验证文件签名
3. Hotfix文件加密之后下发，本地加密存储，使用时候临时解密
4. bsdiff生成的patch文件，需要返回最新文件hash值，供App端merge完成之后校验正确性

发布平台—灰度与回滚

1. 灰度发布

- 发布完成之后，逐步选择一定比例用户生效

2. 回滚

- React Native&Hybrid重新发布
- Hotfix回滚删除本地的版本

3. 暂停发布

- 对于某些功能/修复没有生效，但是对用户没有负面影响，直接暂停发布

运营平台一简介

用户

- 开发、产品经理

目标

- 方便线上App的运营配置

功能

- MobileConfig配置、活动配置、服务器IP流量配置、发布查询等

THANKS!

