QCon[上海站]
全球软件开发大会2016

# React Native项目实战之最后一公里
# 增量升级解决方案

SPEAKER

宝宝树 babytree 胡继伟、曹楠

International Software
Development Conference

主办方 Geekbang 极客邦科技 InfoQ

# 背景介绍

APP中有部分核心功能使用H5开发，性能比较差。在大用户量的情况下，我们需要保证产品的使用体验 —— 技术解决需求。

在保证开发的效率和产品的节奏的要求下，混合开发已经是常规的开发模式。

# Native部分

开发体验差，用户体验好

优势：
用户体验极佳，接口完备，什么都可以做。

劣势：
不能跨平台，一个逻辑两套代码，开发、测试成本都比较高；
开发效率一般，coding -> 编译 -> 打包 -> 调试 -> coding；
版本发布不灵活，受应用市场限制，频繁发版用户体验也不好。

# H5部分

开发体验好，用户体验差

优势：
代码跨平台，开发效率高，发版灵活

劣势：
用户体验不如原生，优化成本较高

# 宝宝树的尝试

15年的一次闲谈一致认为React Native是比较合适的选择；

16年搭个班子工作之余开始做；

在核心产品宝宝树孕育中，集成了RN模块，目前已经上线了专家答模块。

FaceBook带来的礼物，完全满足了我们的预期，甚至高于预期，开发体验、用户体验都极佳，希望有更多的团队能去尝试。

# 我们做了哪些事

1. App瘦身；
2. Android首屏白屏优化；
3. 封装基础框架层，提供丰富基础组件；
4. 自定义原生组件；
5. 提供rnm原生接口；
6. 安全机制，bundle文件校验机制；
7. 增量升级；
8. App RN设置选项；

# 有哪些收益

Web的开发体验，Native的用户体验；

代码保存即生效，不用等待漫长的编译打包发布过程；

组件化复用率高

测试效率高，业务逻辑一套代码，两平台表现基本一致；

版本发布灵活，使用增量升级流量超小，用户无感知；

跨平台，代码一致性95%以上。

# 有哪些不足

Web需要单独开发，适用无Web或弱Web的项目；

用户体验仍没有达到原生效果，单线程js控制的动效，有时会有轻微掉帧；

样式无选择器概念，写起来较为臃肿；

React Native尚未发布1.0版本。

# 言归正传

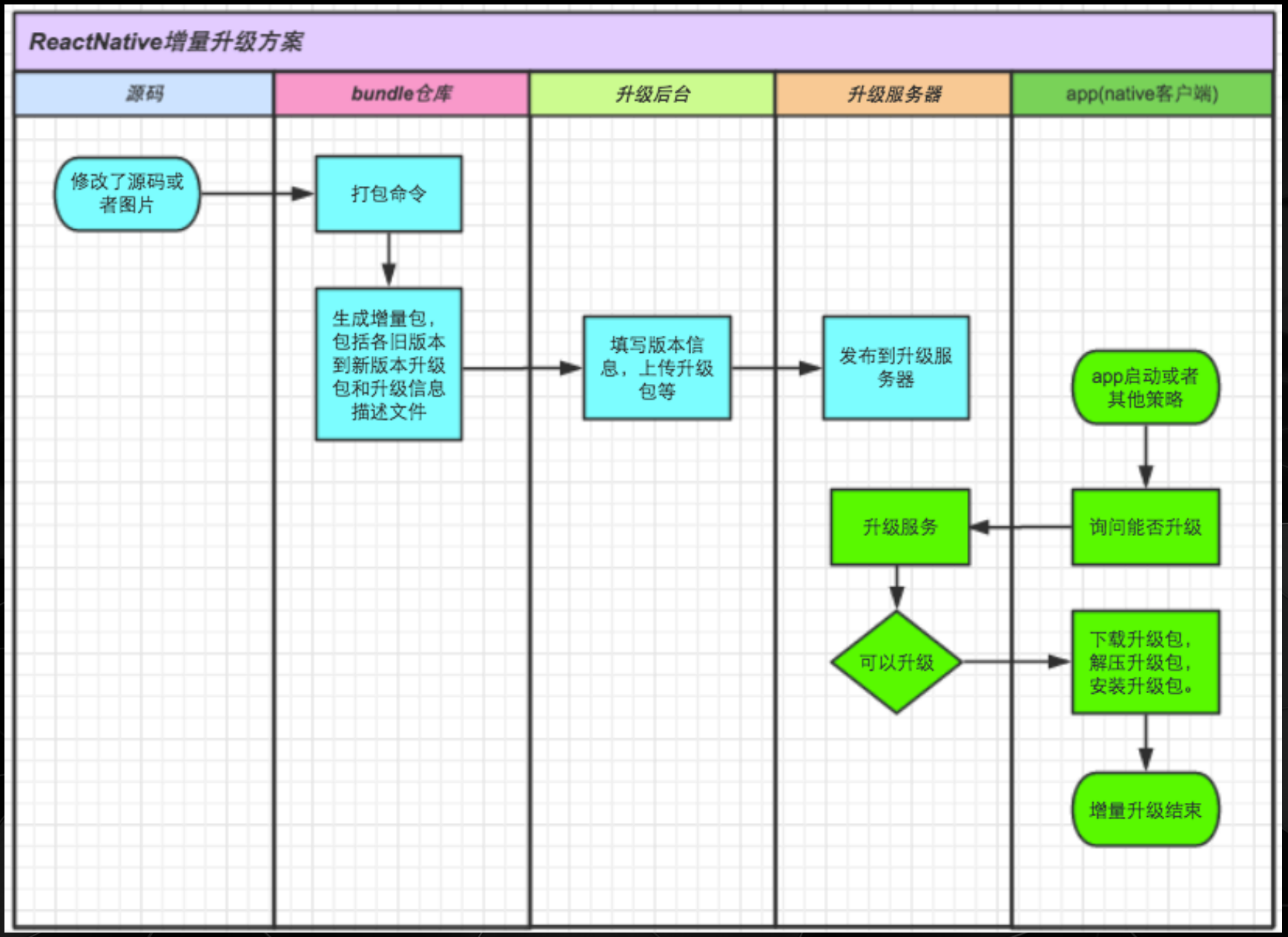
## 宝宝树RN增量升级解决方案

# 目标

高稳定性：任何情况下app功能都不受影响

高性能：升级过程用户无感知

# 体系结构


ReactNative增量升级方案
源码
bundle仓库
升级后台
升级服务器
app(native客户端)
修改了源码或者图片
打包命令
生成增量包，包括各旧版本到新版本升级包和升级信息描述文件
填写版本信息，上传升级包等
发布到升级服务器
app启动或者其他策略
升级服务
询问能否升级
可以升级
下载升级包，解压升级包，安装升级包。
增量升级结束

# 增量是指?

全量包

原生代码（oc，java）

jsbundle文件

图片资源

增量包

jsbundle补丁

图片资源补丁

jsbundle文件增量指的是，代码的改动有多少，增量patch的补丁就有多少，那些没有改动的代码部分是不在补丁的范围内的。

图片资源的增量指的是，升级补丁包中只包含新增的图片和有改动的图片。

# 算法概述

首先，计算增量包：
新版本(v10) - 旧版本(v1到v9) = 增量包
（会有9个包，v1~v10.zip,v2~v10.zip,,,,,v9-v10.zip）

然后，app根据自己的当前版本(比如V6)，下载对应的增量包（V6-V10.zip）。

最后，app中通过
旧版本(v6) + 增量包(v6~v10.zip) = 新版本(v10)，
计算出了新版本的全量包。

# 增量包算法：算法公式

补丁公式：新版本 – 原始版本 = patch

升级公式：新版本 = patch + 原始版本

# 增量包算法：RN完整包内容

原生代码部分，oc或java代码

jsbundle文件

图片资源

# 增量包算法：RN增量包内容

jsbundle文件补丁

图片资源补丁

# 增量包算法：jsbundle补丁生成

新版本与原始版本进行diff，生成补丁。

满足公式：新版本 - 原始版本 = patch

# 增量包算法：diff库介绍

google-diff-match-patch
google实现的针对字符串的diff库，有java，oc，js的实现，但在部分android机上执行会占满cpu资源，导致app卡顿。

bsdiff（推荐）
性能更佳的针对二进制文件的diff库，有c和node的实现，性能佳，手机上运行无感知。

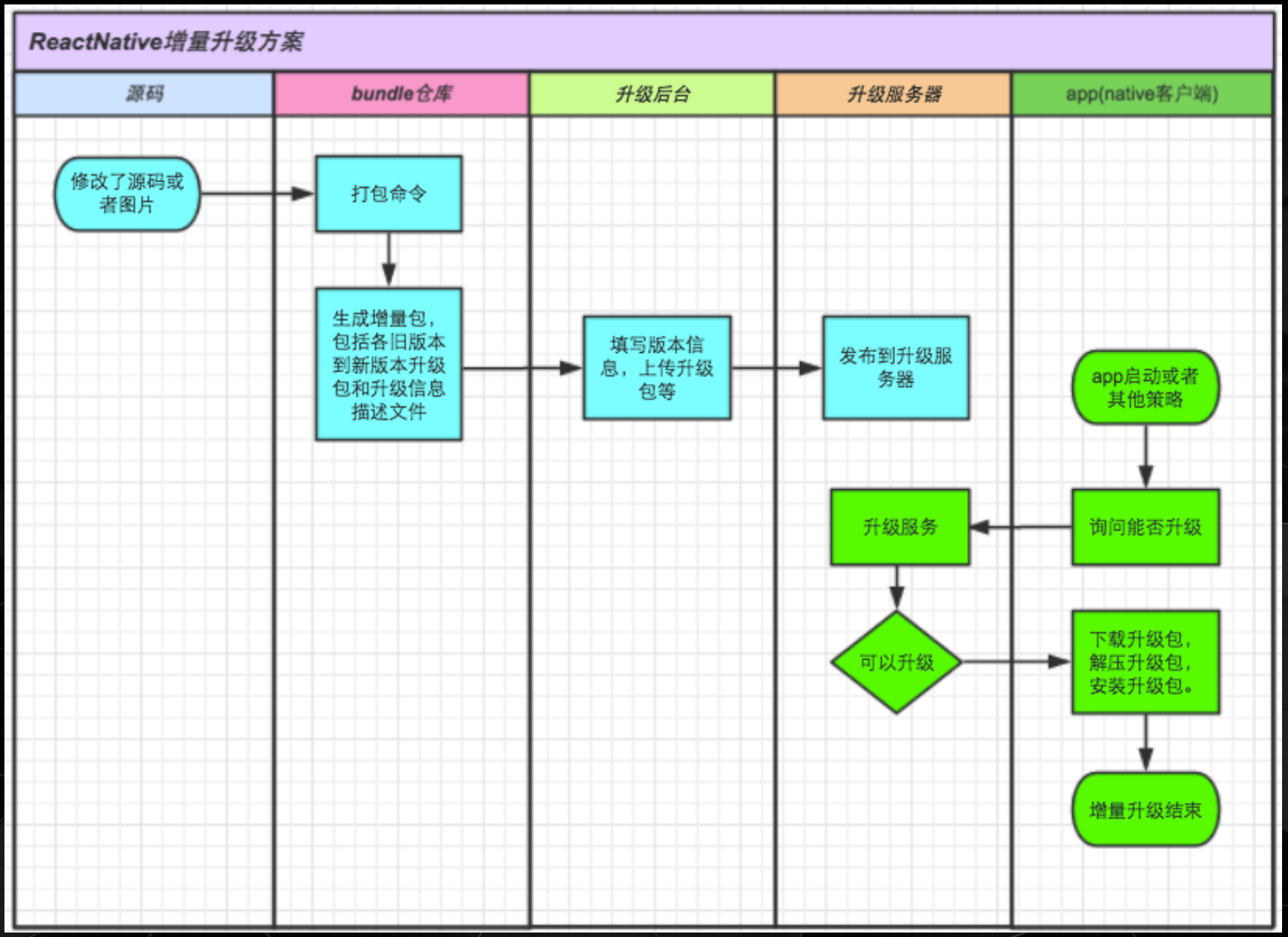
# 增量包算法：图片补丁生成

新图片：新版本有，原始版本无

有改动的图片：新版本与原始版本md5不同

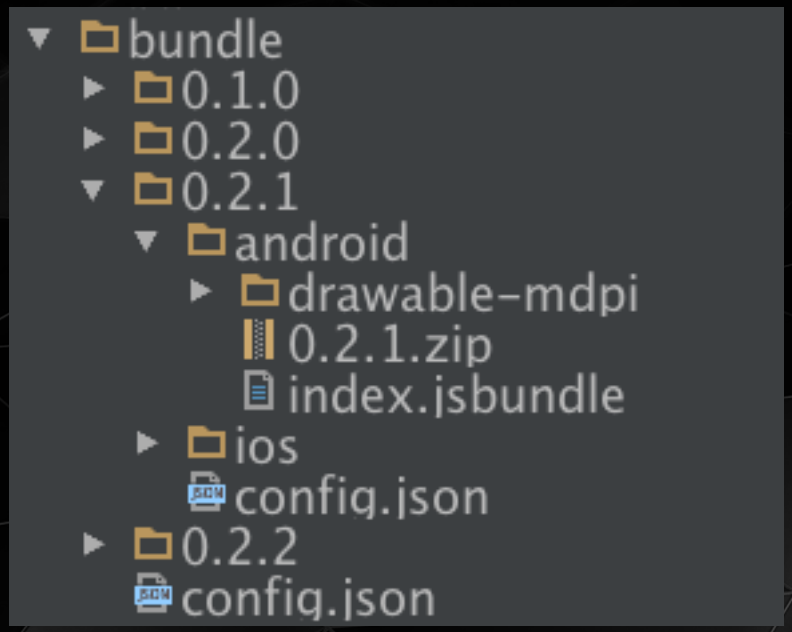
删除的图片：不处理(app升级后会清理)

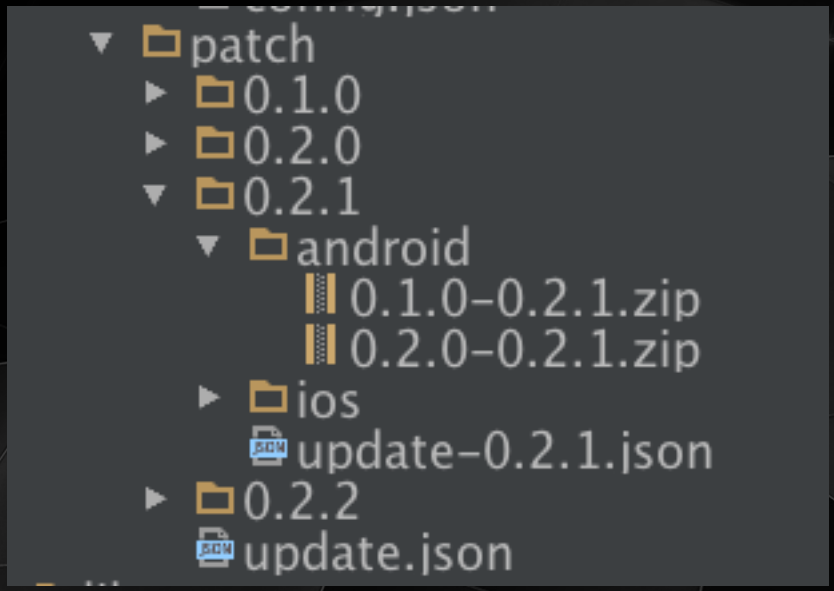
# 体系结构


ReactNative增量升级方案
源码
bundle仓库
升级后台
升级服务器
app(native客户端)
修改了源码或者图片
打包命令
生成增量包，包括各旧版本到新版本升级包和升级信息描述文件
填写版本信息，上传升级包等
发布到升级服务器
app启动或者其他策略
升级服务
询问能否升级
可以升级
下载升级包，解压升级包，安装升级包。
增量升级结束

# bundle仓库

用于存储所有版本的全量和增量包；



# bundle仓库

提供打包工具

node bundle x.x.x 和 node patch x.x.x

会调用RN的打包命令，同时根据上述算法自动生成全量包和增量包。

# 升级后台

提交版本信息和增量包到服务器

校验增量包文件正确性

校验增量包有效性

提供上线、下线开关,可供线下测试用

# 测试流程

将升级包提交到升级后台，设置为线下；

App中使用RN设置菜单，设置为开发模式；

开发模式的App可以升级线下状态的增量包；

测试通过，在升级后台中设置为线上，此时所有用户都能升级了。

# App升级流程

App启动时，访问API询问是否有新包；

有，下载增量包，校验增量包md5；

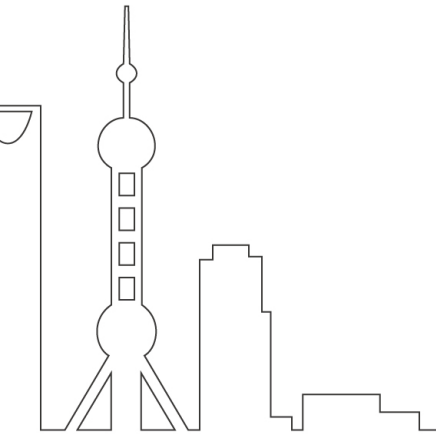
解压，升级，校验新版本jsbundle的md5；

完成升级，在下一次打开RNView的时候会使用新版本

# 服务端升级API算法

根据App提交的参数（App版本，Bundle版本），计算是否有新版本包。

增量包要求的最低App版本
保证JS和Native之间的兼容性，否则JS会调用到Native未提供的接口，导致App崩溃



# Thanks!

International Software Development Conference

宝宝树 babytree

主办方 Geekbang 极客邦科技 InfoQ