

大搜车 ReactNative 开发集成流程体系演变

芋头@大搜车无线团队

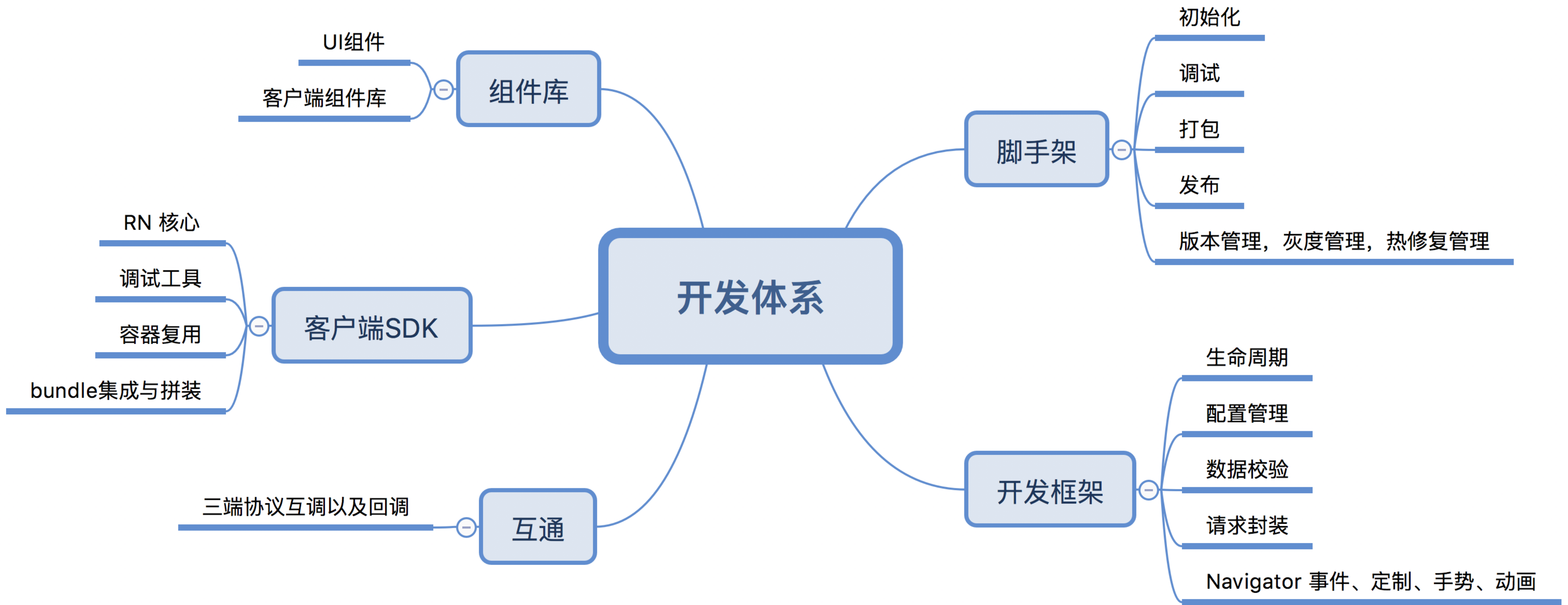
一、场景

场景

- 重业务逻辑，轻营销。更重要的是可控性而不是灵活。
- 核心解决客户端人力膨胀问题，主要由 Native 开发，需尽量遵从其习惯。
- 公司内多个 app，以及合作方的 app，业务繁多。
- 完全耦合进 Native，模糊三端概念，互通互调。

战略级， 没有退路

二、体系简介



三、开发与集成

最初版

当时核心需要解决的问题

- 实现热更新
- 版本锁定
- 三端互通
- 业务解耦，业务之间互相独立
- 发版时带上最新版的 bundle

版本锁定

- 大量的跨端依赖，端与RN的版本可能不兼容。
- RN 官方包 需要可以随时升级，因为不稳定因素很多。
- SDK 可以不断迭代功能，而不影响线上业务。

SRN-HUB 版本管理服务

最初思路

- 在服务端判断请求来自哪个版本的app
- 服务端维护一个关系：哪个app的哪个版本可以更新哪个版本的bundle
- 根据这个关系，返回是否需要下载某个 bundle
- 太复杂的架构不是好架构，要对使用者尽量透明

新型 React Native 依赖模型和代码更新逻辑

客户端	依赖管理服务（服务端）	React Native 远程代码	React Native 项目
<div style="border: 1px solid black; border-radius: 10px; background-color: #4CAF50; color: white; padding: 5px; text-align: center;">启动</div> <div style="border: 1px solid black; background-color: #ADD8E6; padding: 10px; margin-top: 10px;"> 本地依赖 配置文件 {"xx": "1.2.0"} </div>	<div style="border: 1px solid black; border-radius: 10px; background-color: #4CAF50; color: white; padding: 5px; text-align: center;">记录版本 1.2.9</div> <div style="border: 1px solid black; border-radius: 10px; background-color: #4CAF50; color: white; padding: 5px; text-align: center; margin-top: 10px;">记录版本 1.3.0</div> <div style="border: 1px solid black; border-radius: 10px; background-color: #4CAF50; color: white; padding: 5px; text-align: center; margin-top: 10px;">根据规则获取 最新可信赖的 版本</div> <div style="border: 1px solid black; border-radius: 10px; background-color: #4CAF50; color: white; padding: 5px; text-align: center; margin-top: 10px;">规则：取中间版本号相同的 版本的最新版本的包</div> <div style="border: 1px solid black; background-color: #ADD8E6; padding: 10px; text-align: center; margin-top: 10px;">1.2.9</div>	<div style="border: 1px solid black; background-color: #ADD8E6; padding: 10px; text-align: center;">线上包 1.2.9</div> <div style="border: 1px solid black; background-color: #ADD8E6; padding: 10px; text-align: center; margin-top: 10px;">线上包 1.3.0</div> <div style="border: 1px solid black; background-color: #ADD8E6; padding: 10px; text-align: center; margin-top: 10px;">线上包 1.2.9</div>	<div style="border: 1px solid black; border-radius: 10px; background-color: #4CAF50; color: white; padding: 5px; text-align: center;">打包发布 1.2.8</div>
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="width: 30%;"> <div style="border: 1px solid black; border-radius: 10px; background-color: #4CAF50; color: white; padding: 5px; text-align: center;">运行本地代码</div> </div> <div style="width: 40%; text-align: center;"> <div style="border: 2px solid black; width: 40px; height: 40px; margin: 0 auto; transform: rotate(45deg); background-color: #FFA500;"></div> <div style="border: 1px solid black; border-radius: 10px; background-color: #4CAF50; color: white; padding: 5px; text-align: center;">与本地代码 版本号对比</div> <div style="margin-top: 5px;">否</div> <div style="border: 1px solid black; border-radius: 10px; background-color: #4CAF50; color: white; padding: 5px; text-align: center;">下载更新</div> </div> <div style="width: 30%;"> <div style="border: 1px solid black; border-radius: 10px; background-color: #4CAF50; color: white; padding: 5px; text-align: center;">运行 React Native 代码</div> </div> </div>			

```

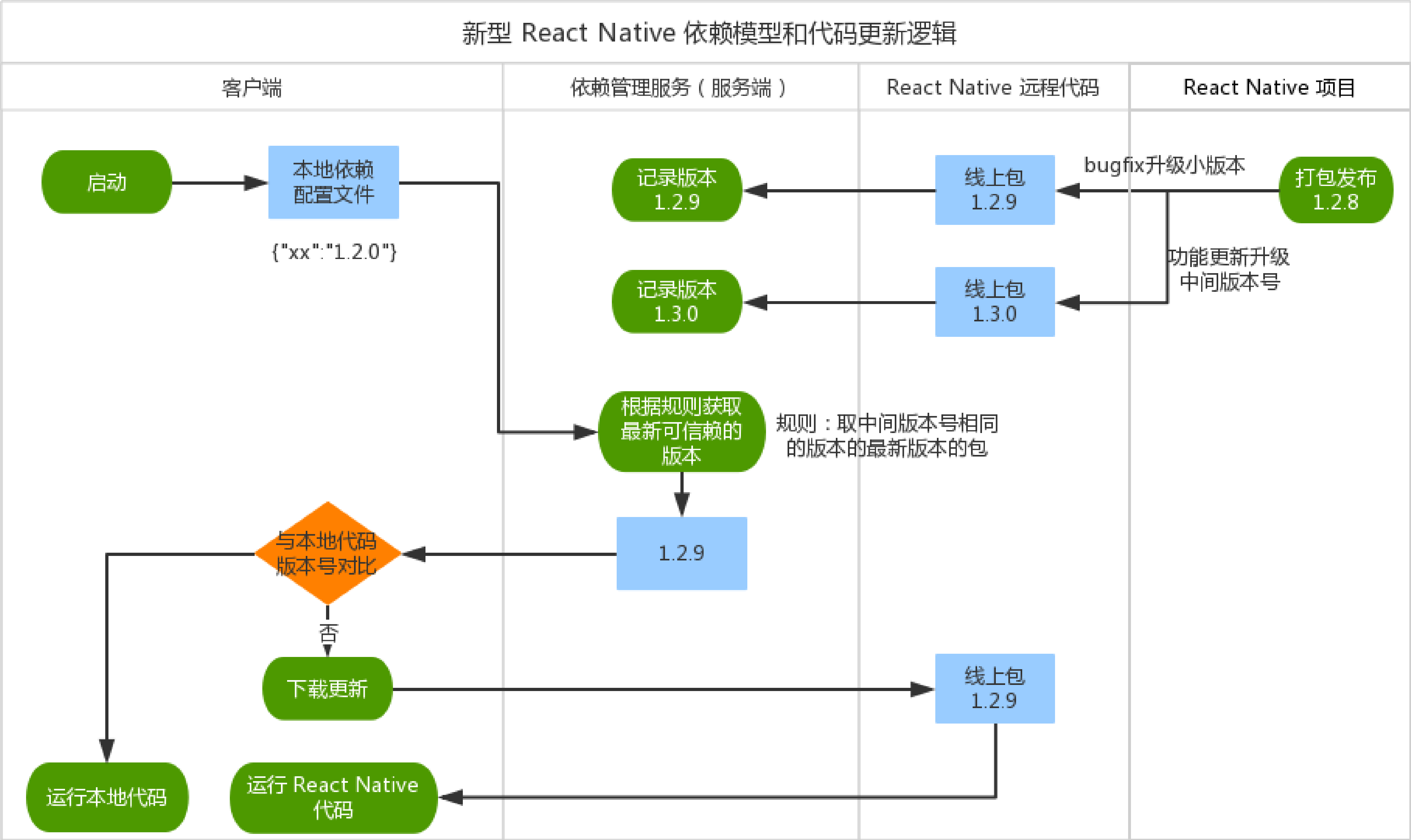
graph TD
    subgraph Client
        Start([启动]) --> LocalConfig["本地依赖配置文件  
{\"xx\": \"1.2.0\"}"]
        LocalConfig --> Compare{与本地代码版本号对比}
        Compare -- 否 --> Download([下载更新])
        Compare -- 是 --> RunLocal([运行本地代码])
        Download --> RunRN([运行 React Native 代码])
    end

    subgraph Service
        Record1([记录版本 1.2.9])
        Record2([记录版本 1.3.0])
        Rule([根据规则获取最新可信赖的版本])
        RuleText["规则：取中间版本号相同的  
版本的最新版本的包"]
        Result["1.2.9"]
    end

    subgraph RemoteCode
        Online1["线上包 1.2.9"]
        Online2["线上包 1.3.0"]
        Online3["线上包 1.2.9"]
    end

    subgraph Project
        Release([打包发布 1.2.8])
    end

    Release -- "bugfix升级小版本" --> Online1
    Release -- "功能更新升级中间版本号" --> Online2
    Online1 --> Record1
    Online2 --> Record2
    Record1 --> Rule
    Rule --> Result
    Result --> Compare
    Result --> Online3
    Online3 --> RunRN
  
```



统一 Router

- 没有复杂的存储规则，所有规则都在版本号里。
- $\$ \{major\} . \$ \{feature\} . \$ \{patch\}$
- 版本号升级根据发布类型自动管理。
- 依赖声明在 app 工程内，由 app 自己决定依赖，而不是三方管理依赖。

问题

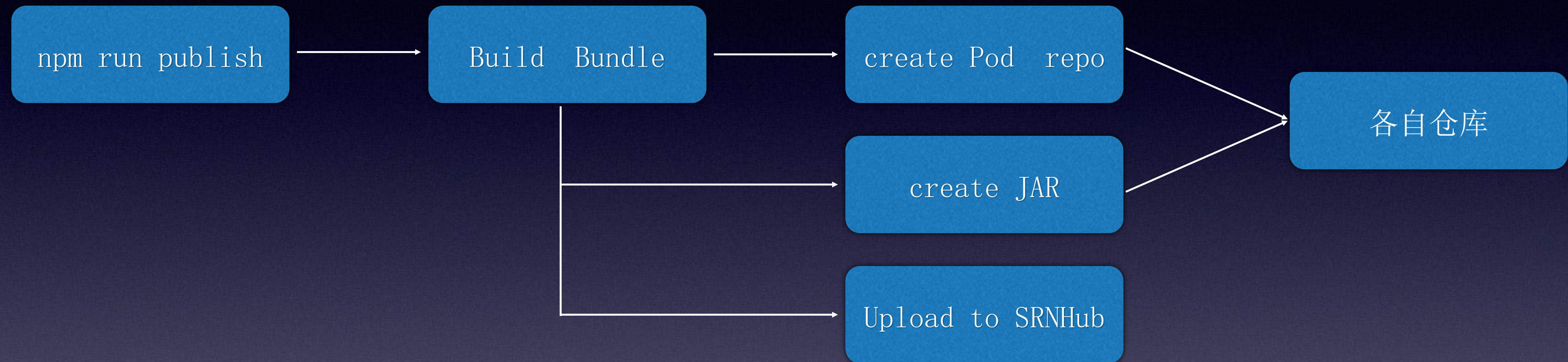
- 发布前需要RN开发修改APP内的依赖文件，这个文件和 podfile ，以及 maven的依赖不在一起，客户端难以适应。
- bundle 包有环境的概念，在开发流程里不断切换环境的时候需要不断改依赖文件，很麻烦。
- 测试环境一律走覆盖式热更新，但是上线前会忘记修改依赖。
- 热更新被滥用，规则成为摆设。
- 打包时需要拉取依赖的bundle集成到本地，拖慢打包过程。

2.0 版

核心解决

- 将 **bundle** 编译成 **Native** 模块
- **bundle** 抽离公用模块
- 扫码调试等辅助开发工具

app 发布流程



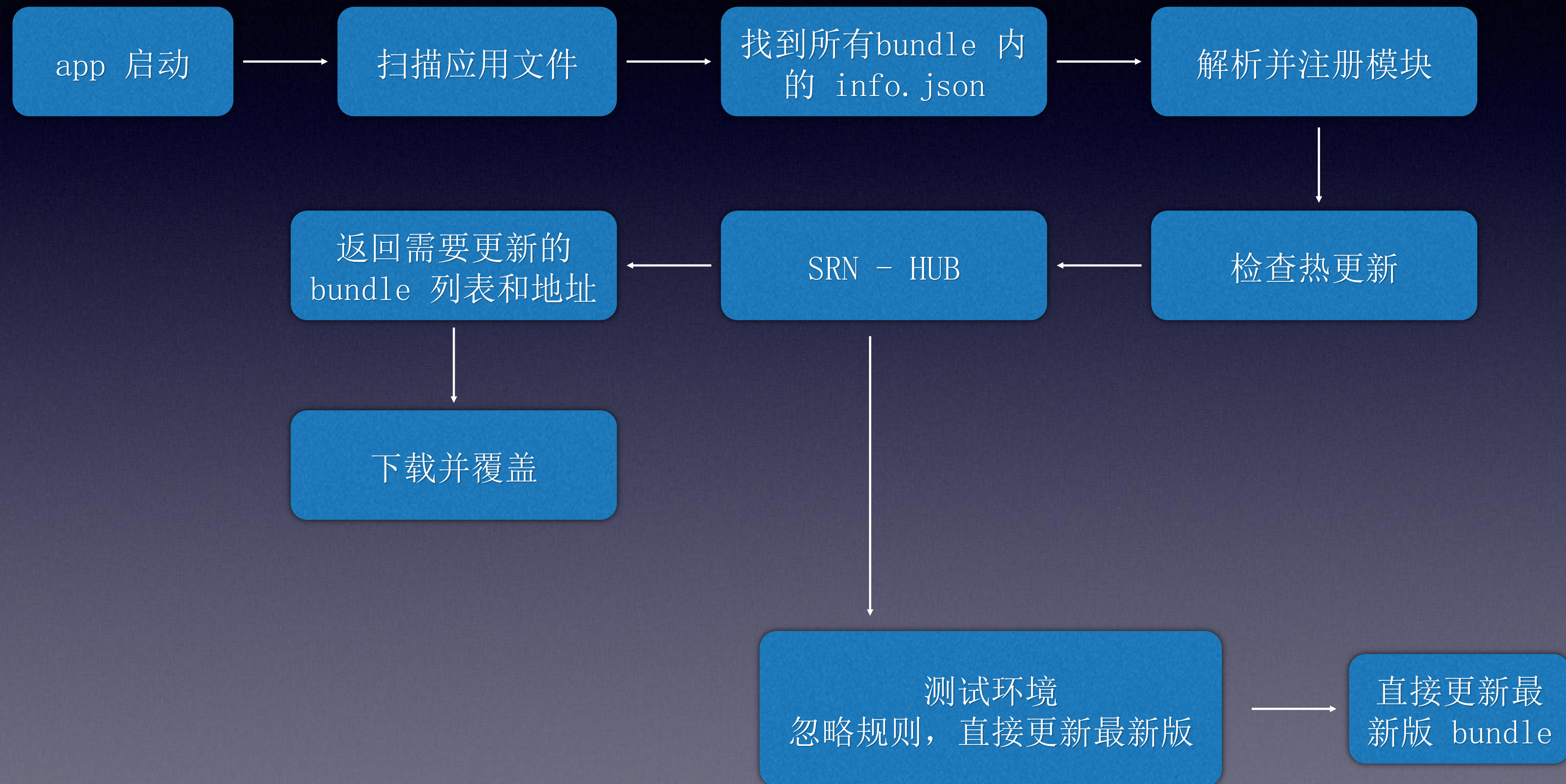
然后在原生工程里声明依赖及版本


```
def static androidRnDependency() {  
    def androidDependency = [  
        'com.souche.rn:DFCDirectRent:0.5.0-SNAPSHOT',  
        'com.souche.rn:DFCWXAccountManager:0.2.1-SNAPSHOT',  
        'com.souche.rn:dfc_groupingstaff:0.2.8-SNAPSHOT',  
        'com.souche.rn:dfc_report:0.4.0-SNAPSHOT',  
        'com.souche.rn:dfc_weibao:0.3.1-SNAPSHOT',  
        'com.souche.rn:dfc_weizhang:0.2.1-SNAPSHOT',  
        'com.souche.rn:RNClueSetting:0.4.0-SNAPSHOT',  
        'com.souche.rn:xfd_platform:0.3.6-SNAPSHOT',  
        'com.souche.rn:dfc_coupons:0.6.2-SNAPSHOT',  
        'com.souche.rn:SCCFastAuction:0.2.0-SNAPSHOT',  
        'com.souche.rn:dfc_wallet_rn:0.4.2-SNAPSHOT',  
        'com.souche.rn:SCCCarSourceFilter:0.2.0-SNAPSHOT',  
        'com.souche.rn:RNCarBrandSelect:0.1.6-SNAPSHOT',  
        'com.souche.rn:RNCitySelect:0.1.3-SNAPSHOT',  
        'com.souche.rn:grab_car:0.1.13-SNAPSHOT',  
    ]  
    androidDependency  
}
```

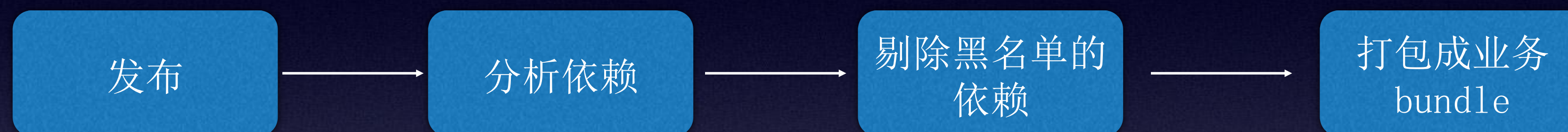


```
139  #RN
140
141  pod 'SCCReactNative', '2.0.5'
142  pod 'DFCDirectRent', '0.3.2'
143  pod 'DFCWXAccountManager', '0.2.1'
144  pod 'dfc_groupingstaff', '0.2.8'
145  pod 'dfc_report', '0.3.0'
146  pod 'dfc_weibao', '0.3.1'
147  pod 'dfc_weizhang', '0.2.1'
148  pod 'RNClueSetting', '0.4.0'
149  pod 'xfd_platform', '0.3.6'
150  pod 'dfc_coupons', '0.6.2'
151  pod 'SCCFastAuction', '0.2.0'
152  pod 'dfc_wallet_rn', '0.4.2'
153  pod 'SCCCarSourceFilter', '0.2.0'
154  pod 'RNCarBrandSelect', '0.1.6'
155  pod 'RNCitySelect', '0.1.3'
156  pod 'grab_car', '0.1.13'
157  end
```

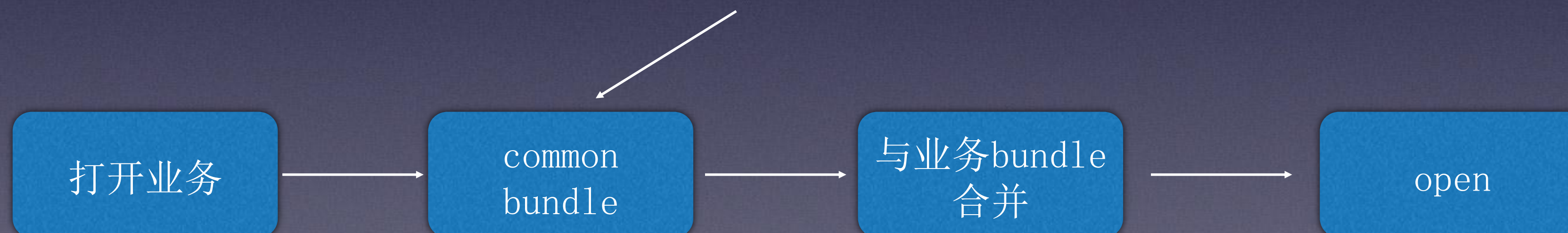

app 启动后流程



bundle 拆分



内置在工程 SDK 中



扫码调试



```
{  
  "bundleName": "workspace",  
  "bundleProps": {},  
  "bundleAddress": "http://192.168.105.16:8080"  
}
```

Carrier 4:54 PM RN模块

Back

请输入module

请输入Host,默认localhost

请输入完整的url

参数

添加

Key: 请输入

Value: 请输入

删除

历史记录

清理

srn_ui :: localhost :: {}

framework :: localhost :: {}

NavHelper :: localhost :: {}

srn :: localhost :: {}

srn_template :: localhost :: {}

跳转调试RN页面

扫一扫进入RN

遗留问题

- 发布脚本时仍然会让你选择发到哪个环境，其实对于原生包来说，没有环境的概念，而是用版本号格式表达。
- 热更新导致发布不可控，产品一直想要热更新，业务太多，app内容频繁发生变化。
- 测试不方便，不知道当前是什么版本的业务，也控制不了要测试的是哪个版本的业务，测试通过后需要修改依赖。

3.0 版

- 遵照客户端理念，去掉包环境的概念
- 关闭热更新，去掉版本号依赖规则，增加热修复
- 新的测试工具，可以回朔任何业务的任何版本
- 分离测试工具与业务 SDK 的耦合
- 容器复用机制

热修复

- 不按照版本号规则，因为不需要严格的版本锁定。
- 直接在后台定义需要把哪个版本的业务 bundle 更新成哪个版本的 bundle 的规则。
- 热修复需要抄送副总裁和各个大佬才能发布。

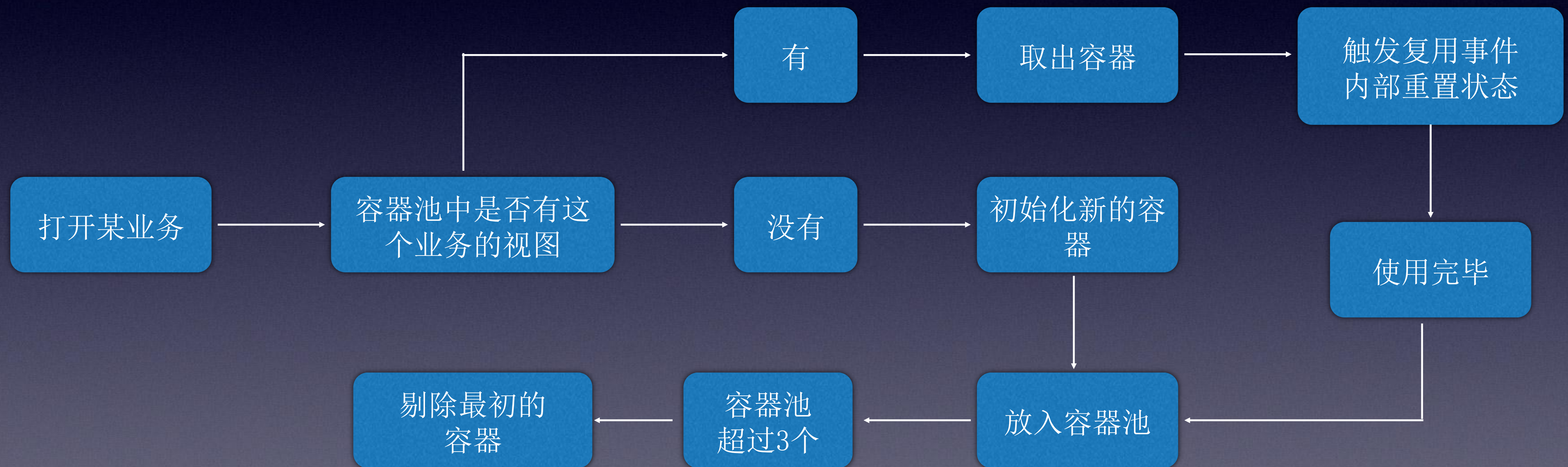
测试工具

不依赖更新规则
手动选择测试包

iPhone 6 - iOS 11.0	
Carrier	5:42 PM
Back	<div>RN模块清除全部</div>
业务名	更新时间
当前版本	最新版本
rndemo	2017-08-15 16:29:19
0.5.9	0.6.4
dfc_groupingstaff	2017-06-20 18:27:51
0.2.1-beta.22	0.2.1-beta.22
SCCFastAuction	2017-07-15 11:56:15
0.1.1-beta.10	0.1.1-beta.10
xfd_platform	2017-08-15 15:00:41
0.3.2-beta.33	0.3.6-beta.2
dfc_report	2017-08-17 11:17:12
0.2.1-beta.11	0.2.2-beta.2
RN	

iPhone 6 - iOS 11.0	
Carrier	5:43 PM
RN模块	<div>RN模块历史版本清理</div>
业务名 : SCCFastAuction	
当前版本 : 0.1.1-beta.10	
0.1.1-beta.10	开发提测版
master	2017-07-15 11:56:15
0.1.1-beta.9	开发提测版
master	2017-07-14 16:22:52
下载	
0.1.1-beta.8	开发提测版
master	2017-07-14 11:22:50
下载	
0.1.1-beta.7	开发提测版
master	2017-07-11 18:59:18
下载	
0.1.1-beta.6	开发提测版
master	2017-07-07 14:31:54
下载	
0.1.1-beta.5	开发提测版
master	2017-07-03 16:33:49
下载	
0.1.1-beta.4	开发提测版
master	2017-07-03 13:00:25
下载	
0.1.1-beta.3	开发提测版
master	2017-06-30 15:24:05
下载	

容器复用



“脱离场景谈架构都是耍流氓。”

— 路人甲

Thanks

芋头

大搜车无线架构团队

Nodejs 中间件、客户端、前端、RN

个人开发者

《颜文字输入法》《二十一点睡前故事》

《前端乱炖》《设计搞》



芋头 

浙江 杭州



扫一扫上面的二维码图案，加我微信