

基于React Native 的三端融合实践



About me

- 陈达孚
- 沪江Web前端架构部
- Web、Native端融合



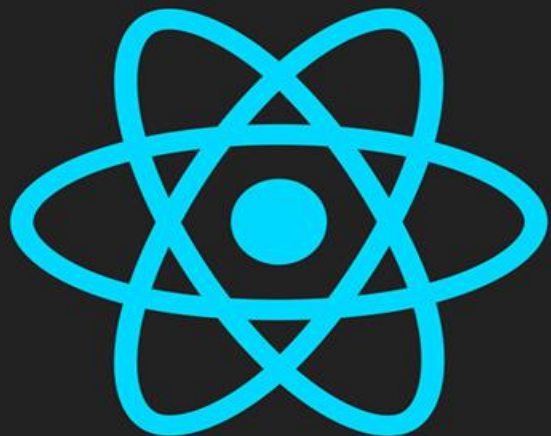
摘要

- 为什么选择React Native
- 三端融合
- React Native的上线保证



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术



React Native for Cross-platform
Apps on iOS & Android



















为什么选择React Native

React Native优势

- 两端共享一套业务代码
- 接近原生的性能
- 社区活跃

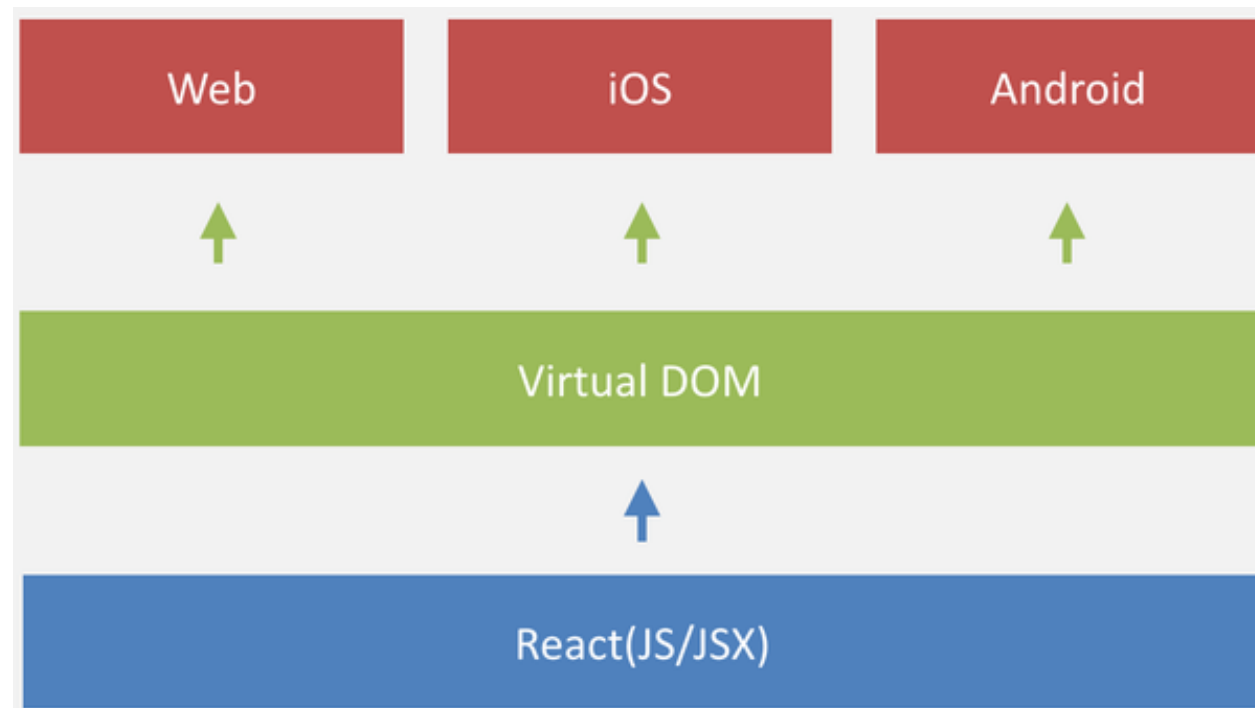
Who's using React Native? 中生代技术 FRESHMAN TECHNOLOGY

Thousands of apps are using React Native, from established Fortune 500 companies to hot new startups. If you're curious to see what can be accomplished with React Native, check out these apps!

 Facebook iOS · Android Using React Native in the Facebook App	 Facebook Ads Manager iOS · Android How We Built the First Cross-Platform React Native App	 Facebook Groups iOS React Native: Bringing Modern Web Techniques to Mobile	 Instagram iOS · Android React Native at Instagram
 Airbnb iOS · Android Hybrid React Native Apps at Airbnb	 Baidu (手机百度) iOS · Android Baidu is a search engine that has 600 million users.	 Discord iOS Using React Native: One Year Later	 Gyroscope iOS Building a visualization experience with React Native
 li.st Android Building li.st for Android with React Native	 QQ Android QQ is a Chinese messaging service with 829 million active accounts	 Townске iOS "I would recommend React Native in a heartbeat."	 Vogue iOS
 Walmart iOS · Android	 Artsy iOS	 Bdsdiet Android	 Bitt Wallet iOS

React Native劣势

- Learn once write anywhere
- 部分组件性能
- 早期开发成本和融合开发成本



沪江应用现状

- 三端独立完成，复用率低
- 通过Web容器接入在线页面



学习，成为更好的自己

问题-原生

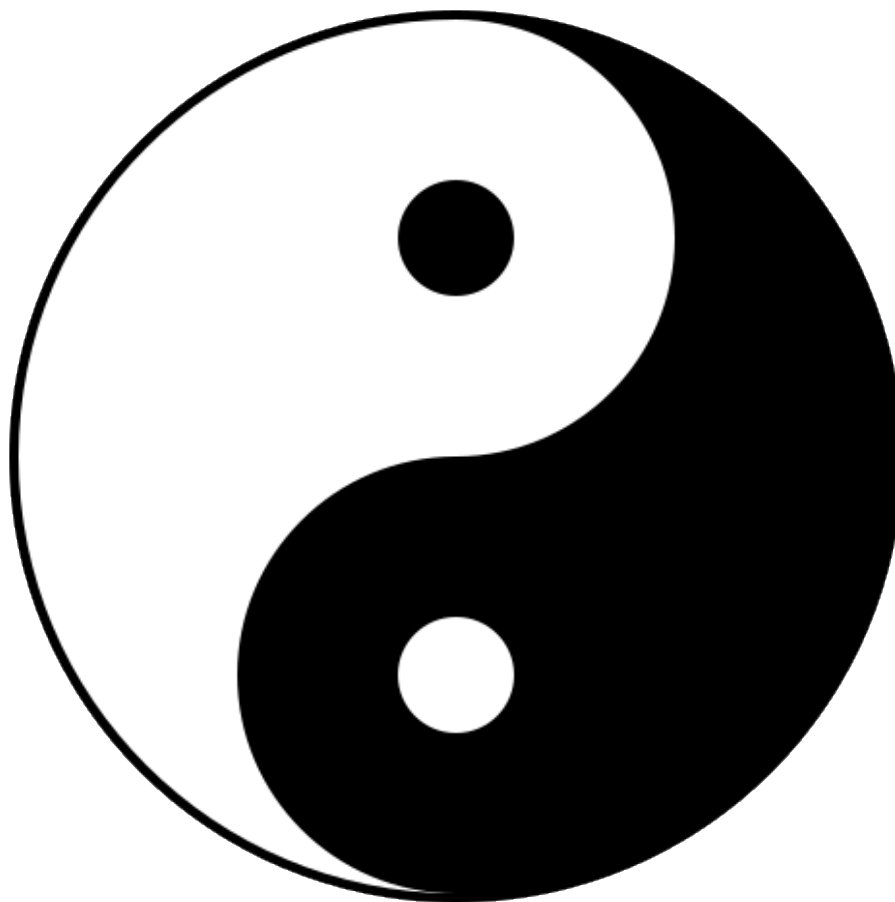
- 体验好
- 频繁发版
- 复用率低
- 活动大促前的上线准备





问题-在线页面

- 业务方需要写大量兼容代码
- 体验较差
- 遇到网络问题展示不出



三端融合



三端融合？

- 对立和统一
- 一套代码三端复用
- 限定来控制复杂度

思路与方案

- React Web组件 + React
- Babel编译
- 完整Web框架

React Web组件 + React

- 初次开发成本低
- 组件代码冗余
- 不确定API
- 底层优化复杂

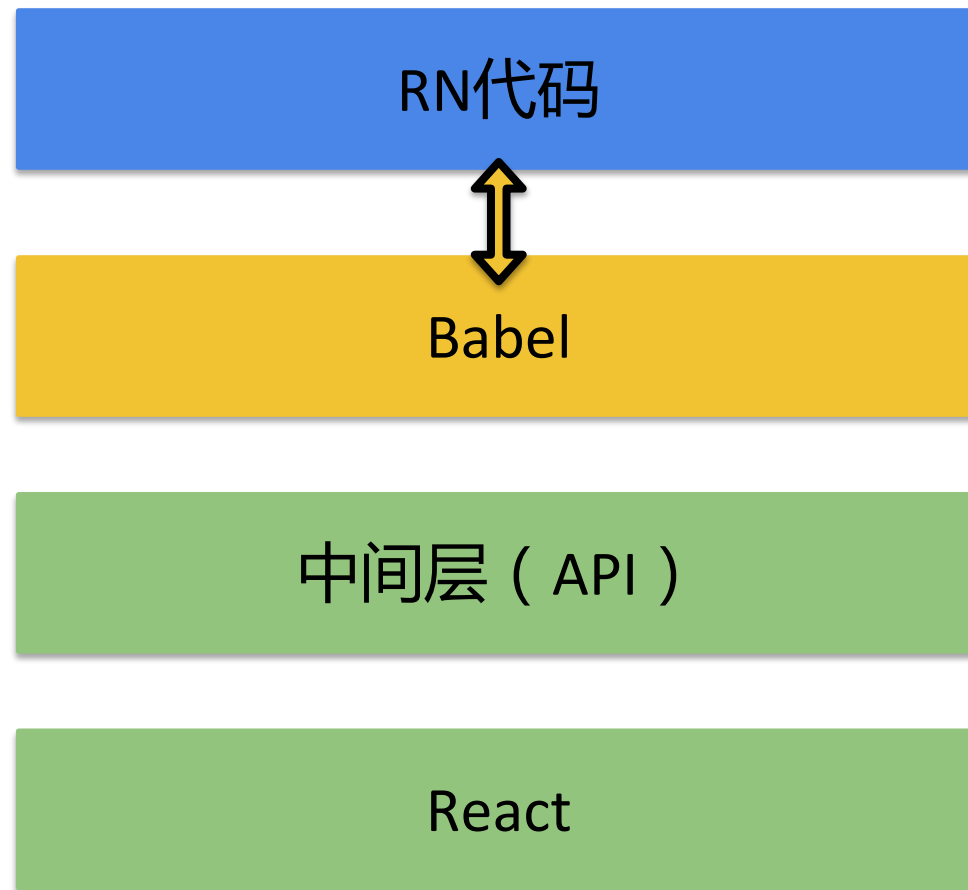
RN代码

RN WEB

React

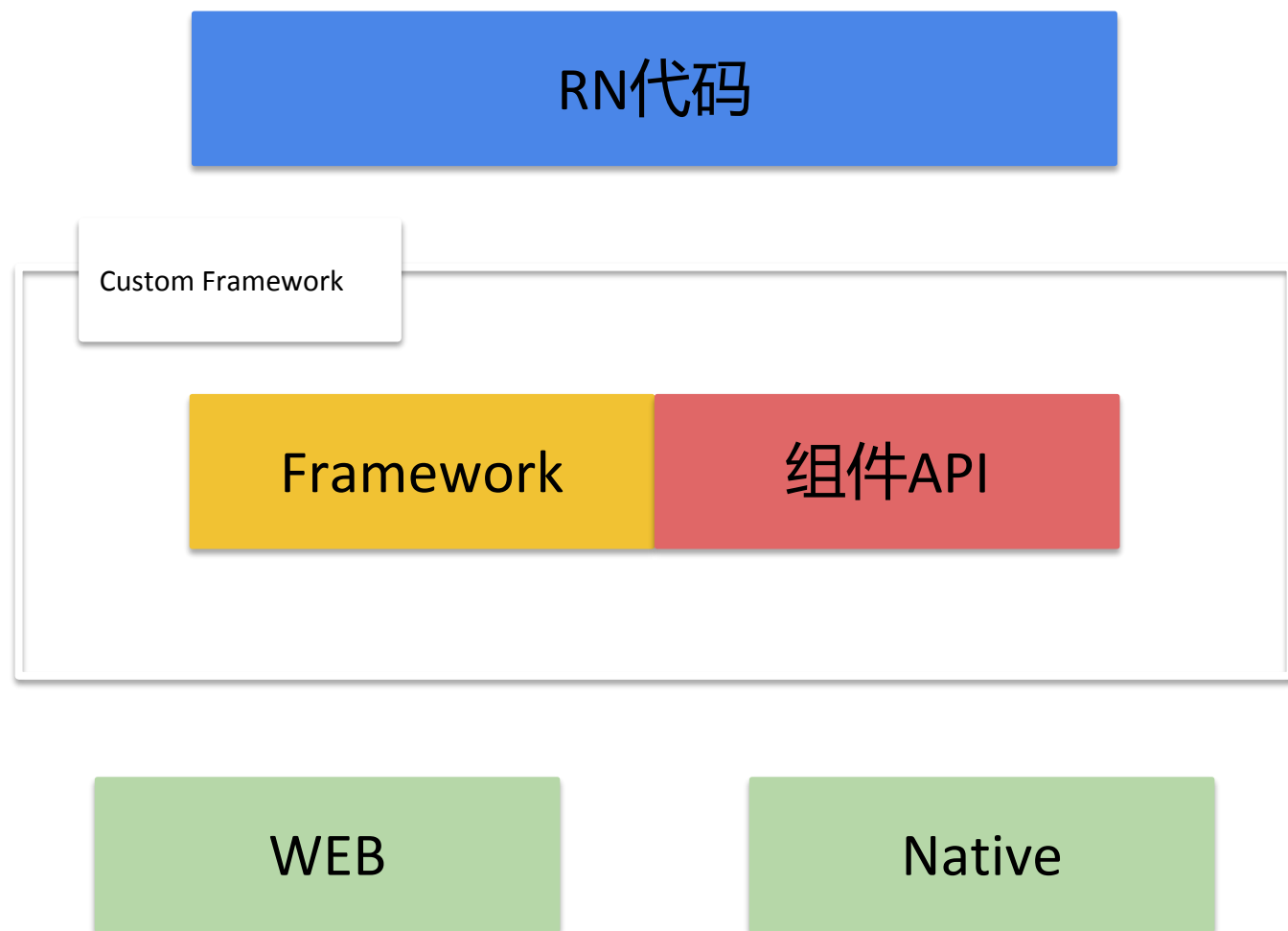
Babel编译 + React

- 复杂度过高
- 业务方可以后期调整



Web组件 + 定制化框架

- 将React Native代码视作一套DSL语言
- 部分组件三端兼容性
- React-navigation



Web组件 + 定制化框架

- 摆脱组件和框架依赖关系中的不确定性 (React到React-DOM , deprecated API)
- 减少组件中的冗余代码
- providesModule

Web组件 + 定制化框架



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

- 兼容性
- API支持能力
- 实际开发中的情况

常用组件及API

- View , ScrollView , ListView , Text, Image, TouchableHighlight, StyleSheet
- Web思路下的开发大量组件和API并不会使用
- 空方法并提示

入口文件修改

```
1 import { AppRegistry } from 'react-native';  
2 import ClassCenter from './src/containers/ClassCenter';  
3  
4 AppRegistry.registerComponent('classCenter', () => ClassCenter);
```



```
1 import { render } from 'cassia';  
2 import ClassCenter from './src/containers/ClassCenter';  
3  
4 render(<ClassCenter />);
```

打包工具设置别名



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

```
resolve: {  
  alias: {  
    'react' : 'cassia',  
    'react-native': 'cassia'  
  }  
}
```



组件框架

- 什么需要在组件中做
- 什么需要在框架中做

Web组件

- 处理React Native组件中的props，转换到相应的Web实现，或者初步处理交给框架二次处理
- 处理React Native的API，调用Web API，降级方案中调用Hybrid API



框架

- 组件生命周期
- setState与更新队列
- 事件系统
- Vitrual DOM

框架精简

- 不稳定的API
- React提供，React Native不提供的API
- PropTypes
- 合成事件
- 旧语法和旧浏览器的支持
- 简化了部分事务系统和调度系统

框架添加功能

- Props(Style)处理
- 部分公共API
- React Native框架API:
 - findNodeHandle
 - setNativeProps
 - NativeMethods....



整合的优势

- 减少API的频繁引入
- 优化组件的编写成本
- 更小的包体积
- 减少重复的props检查
- Webpack2



整合的优势



- 性能？



中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

React Native的上线保证

学习，成为更好的自己

开发上线RN

- 横向团队在RN开发中的职责
- 完整的技术架构方案
- 技术保证

开发环境

- 测试容器
- Web开发通过发测试包迅速查看在APP中的运行情况
- 热更新应用代码

保障

- 前端监控
- 崩溃监控
- 数据上报
- 业务方通过配置中心选择

TYPE		
Term	Count	Action
and_lan_d ict	6604	🔍 🗑️

ERROR CODE		
Term	Count	Action
无网络, 安 装包词库未 收录, 离线 词库未下载	3284	🔍 🗑️
有网络, 本 地和后台词 库未收录	1103	🔍 🗑️
连接超时	805	🔍 🗑️
连接超时, 本地词库	760	🔍 🗑️

ERROR INFO		
Term	Count	Action
	6604	🔍 🗑️

IMEI		
Term	Count	Action
小D查词	5356	🔍 🗑️
小D通用	996	🔍 🗑️
1	252	🔍 🗑️

更新

- 拆包
- 打包工具
- 自动生成配置信息
- MMP2.0
- 针对版本的管理更新机制

Search:

序号	图标	应用名	平台	包名	正式应用	操作
2		听力酷-Android	Android	com.hj.dictation	是	编辑 版本管理 渠道部署 删除
5		大沪江-Android	Android	com.hujiang.normandy	是	编辑 版本管理 渠道部署 删除
14		Doraemon Demo	IOS	com.hujiang.DoraemonExample	否	编辑 版本管理 渠道部署 删除
15		TestDemo	IOS	com.hujiang.hjm.normandy	否	编辑 版本管理 渠道部署 删除
16		小D	Android	com.hujiang.dict	否	编辑 版本管理 渠道部署 删除
18		学习工具测试	Android	com.hujiang.studytool	否	编辑 版本管理 渠道部署 删除
27		AndroidSDKPackage	Android	com.hujiang.test	是	编辑 版本管理 渠道部署 删除
39		开心词场	IOS	com.hujiang.wordplus.trial	是	编辑 版本管理 渠道部署 删除

功能性组件开发



- Login
- BI
- Share

性能优化

- 最好的方法是扬长避短
- 预加载
- 针对性的优化 (ListView)
- RN自身的优化

Before:

```
Animated.timing(this.state.animatedValue, {  
  toValue: 1,  
  duration: 500,  
}).start();
```

After:

```
Animated.timing(this.state.animatedValue, {  
  toValue: 1,  
  duration: 500,  
  useNativeDriver: true, // <-- Add this  
}).start();
```

性能优化-ListView



- Web端优化
- Native端优化

性能优化-ListView

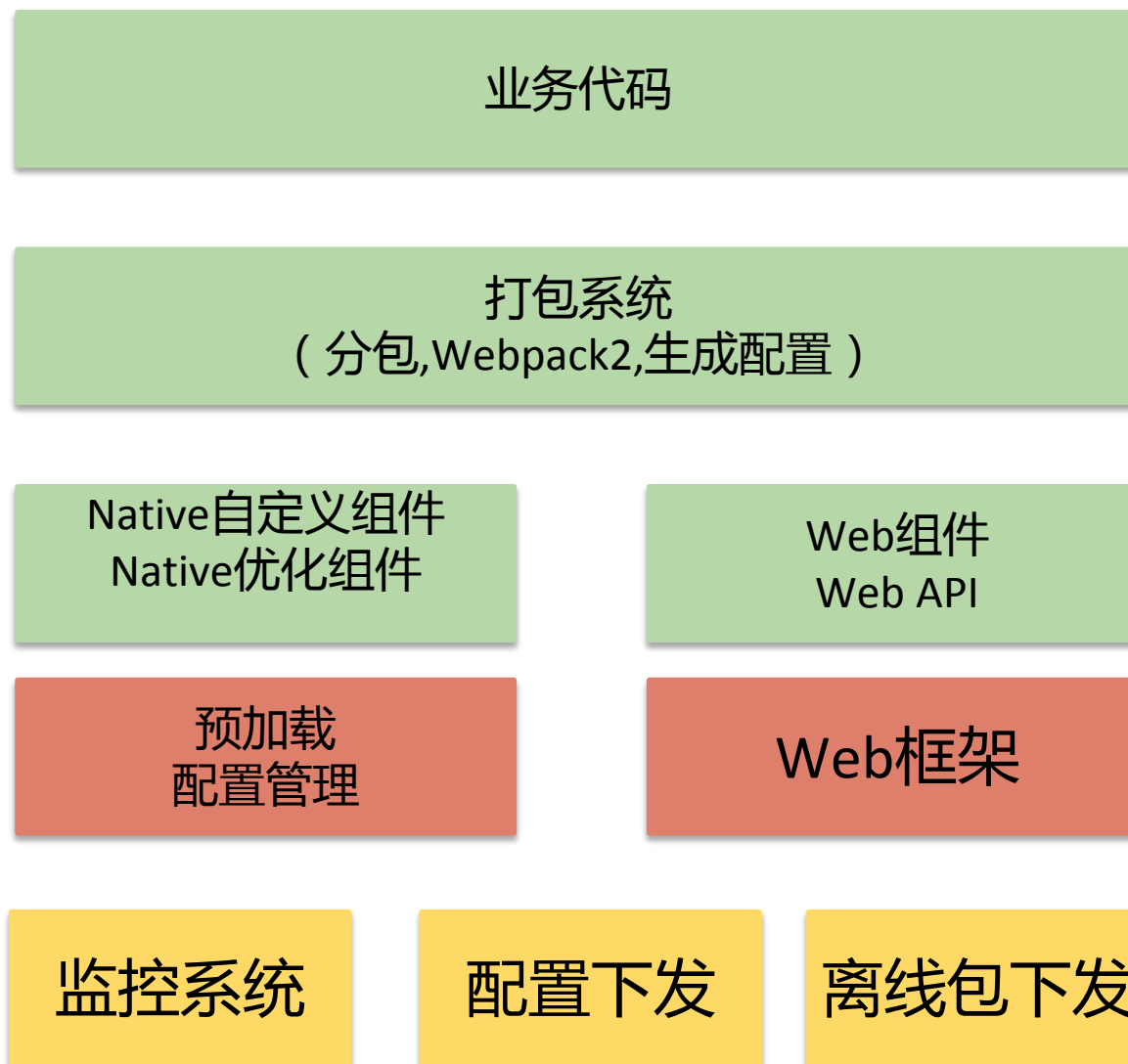


中生代技术
FRESHMAN TECHNOLOGY

IT大咖说
不止于技术

- 自动移除不显示的 View , 但是实际上还是创建了所有的子 View
- 缓存而非复用
- FlastList
- recyclerView , UITableView(不同形态的View表现并不好)

整体架构



未来

- 开源
- 更多的优化组件（三端统一基础）
- 替代React-Native中的React部分
- 业务结合中的调整

大前端时代



- Web端和Native端的差异与统一
- Native , React Native , Hybrid , PWA

沪江

学习，成为更好的自己