中国移动开发者大会
Mobile Developer Conference China 2016

从 React 到 ReactNative
渐进强化应用体验

张臻 @ NHN Tehcorus
2016

mdcc.csdn.net

自我介绍

- 张臻
- Web 前端开发部门经理 @ NHN Techorus
  - NHN comico 集团内各 B2C 项目（以游戏，娱乐，社交为主）的 Web 开发
  - 从 2015 年上半年开始使用 React 进行开发
- Github
  - https://github.com/toruta39

MDCC 2016
中国移动开发者大会
Mobile Developer Conference China 2016

ReactNative 技术选型的背景

mdcc.csdn.net

- 一次编写，几乎可以在所有平台运行
- 搜索引擎可爬性
- 版本控制简单
- 可分享性

- 页面的载入时间，离线体验
- 无法利用主屏幕图标，推送通知等功能增强应用画面外的互动
- 受限于浏览器的实现，用户体验难以匹敌 Native 应用
  - 相机，音频，视频，后台，振动，剪贴板，应用内支付……
- Android 的老旧机种上的性能问题

- 利用 Native 平台的优势来增强用户体验
  - 尽可能地将现有的 Web 应用快速包装成 Native 应用上线
  - 尽可能让现有的开发者轻松上手 Native 的开发
  - 尽可能实现 iOS/Android/Web 间跨平台的代码复用，减少未来多平台对应可能带来的负担

- 迅速将 Web 应用打包成 Native 应用

  ReactNative as Corvado

- 将 Web 应用移植到 Native 平台

  ReactDOM to ReactNative

- 实现 iOS/Android/Web 间跨平台代码复用

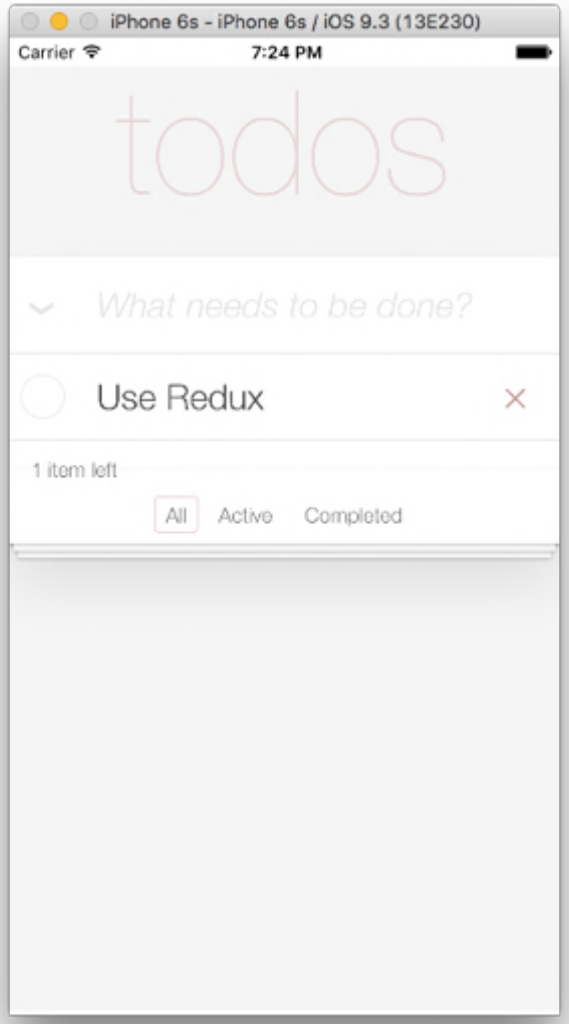
  ReactNative for Web

- 用 TodoMVC 的 React + Redux 的用例代码为基础，实际尝试了 3 种方案的实现

中国移动开发者大会
Mobile Developer Conference China 2016

ReactNative as Corvado

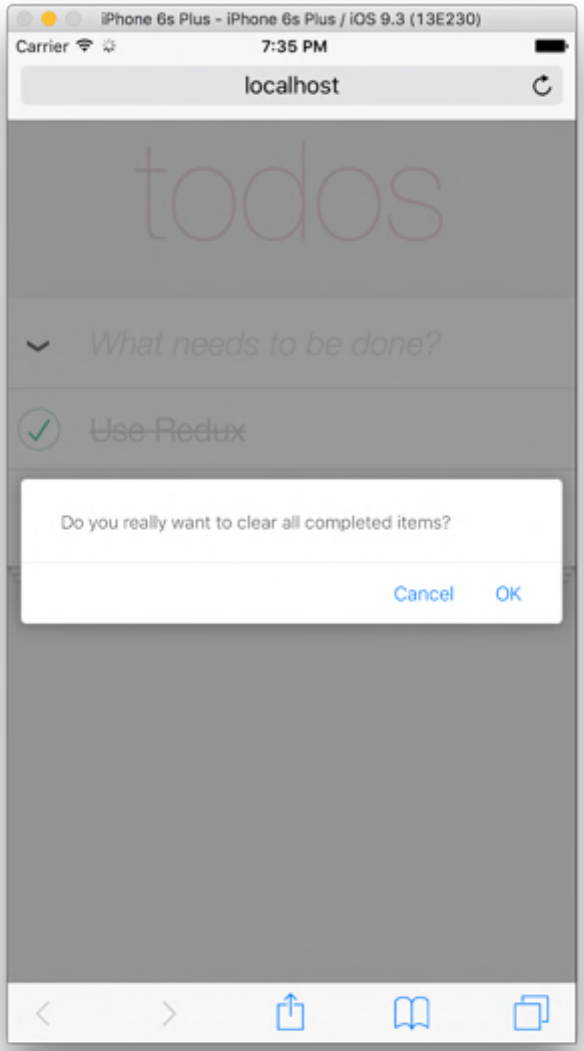
mdcc.csdn.net


iPhone 6s - iPhone 6s / iOS 9.3 (13E230)
Carrier
7:24 PM
todos
What needs to be done?
Use Redux
1 item left
All
Active
Completed

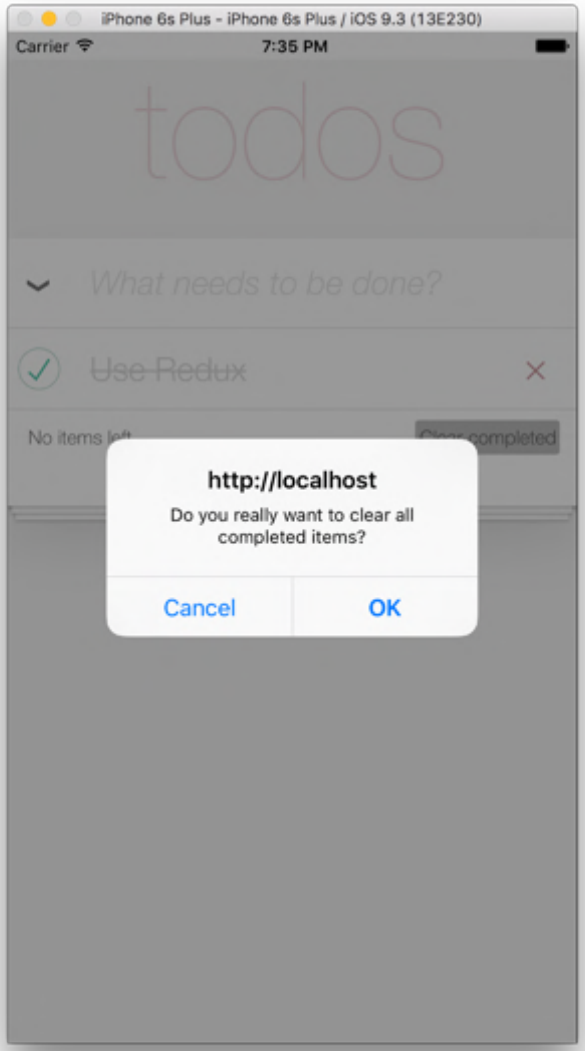
- 将现有的 Web 应用封装在一个 WebView 里面
- 对需要增强体验的部分，局部性地利用 WebView 和 Native 的双向通信，从 WebView 调用 Native 功能

- 将 React Web 应用封装为 Native iOS 应用
- 利用 Native 的 API 来改善用户体验
- 项目代码：
  - https://github.com/toruta39/ReactNativeTodoMVC/tree/feature/webview-bridge

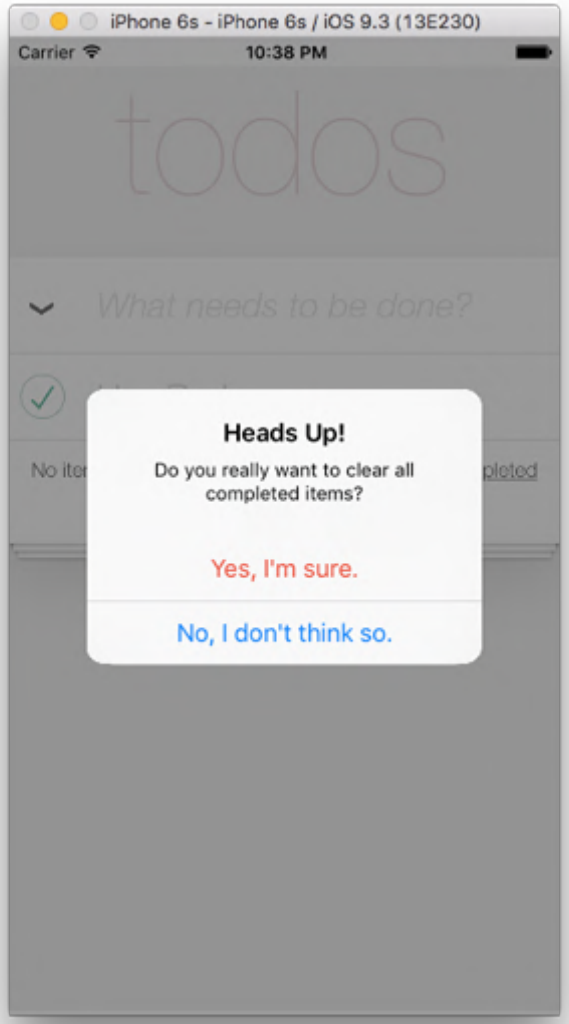
利用 Native 的 API 来改善用户体验



Browser



WebView

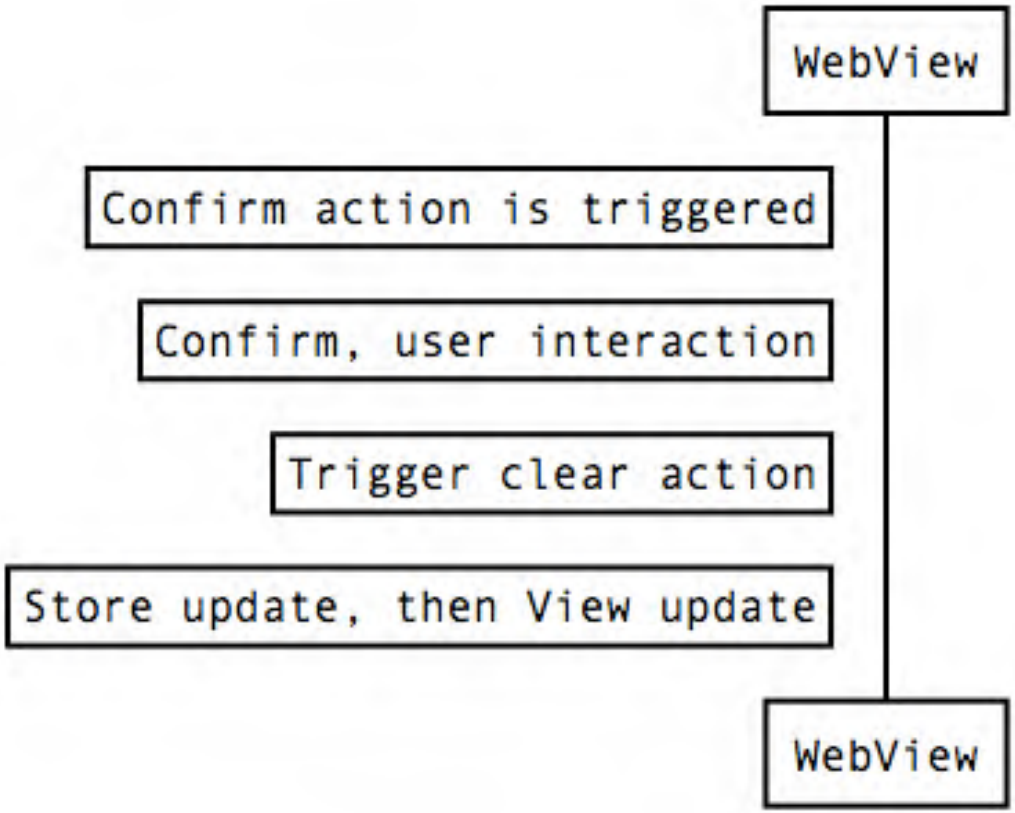

iPhone 6s - iPhone 6s / iOS 9.3 (13E230)
Carrier
10:38 PM
todos
What needs to be done?
Heads Up!
Do you really want to clear all completed items?
Yes, I'm sure.
No, I don't think so.

Native

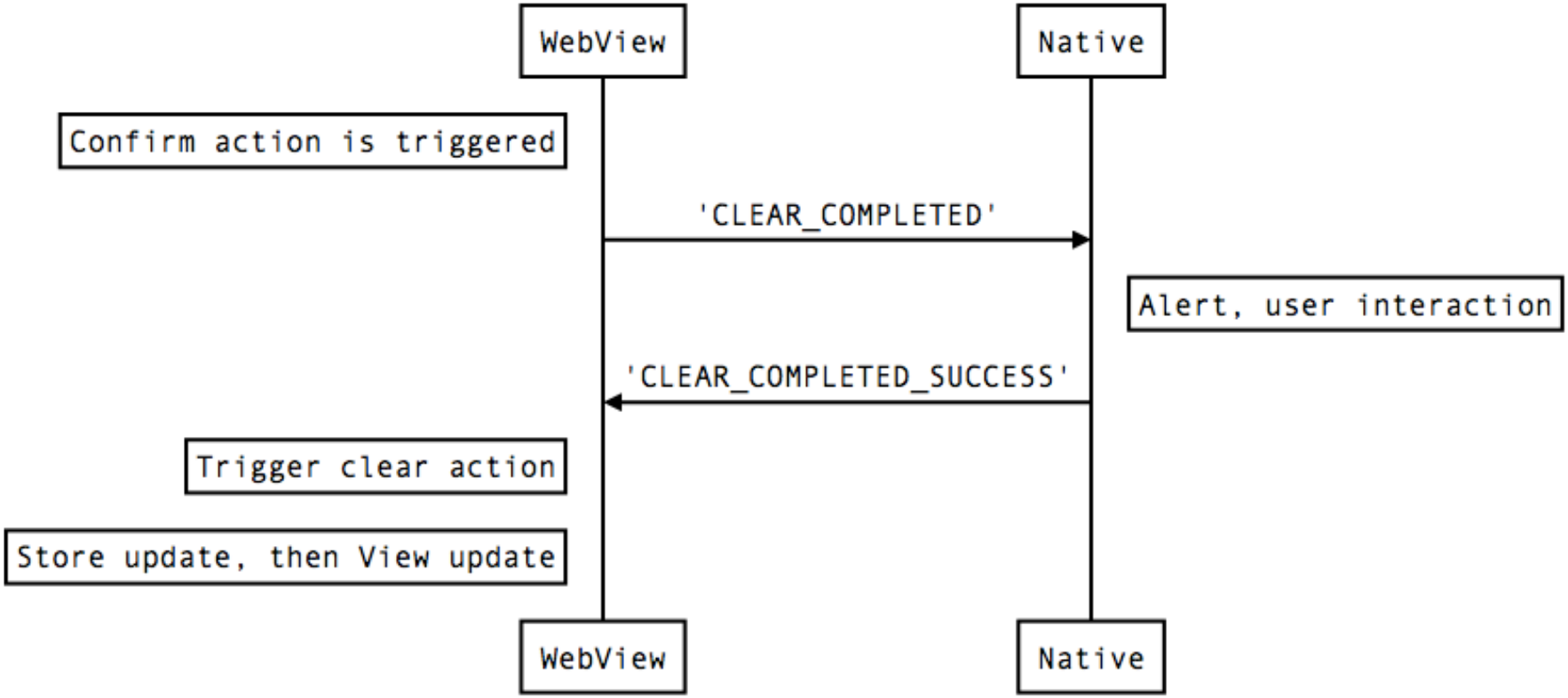
- WebView 和 Native 双向通信部分可以利用开源项目
  - https://github.com/alinz/react-native-webview-bridge

```
<WebViewBridge ref="webviewbridge"
               onBridgeMessage={this.onBridgeMessage.bind(this)}
               source={{uri: "http://localhost:3000"}} />
```

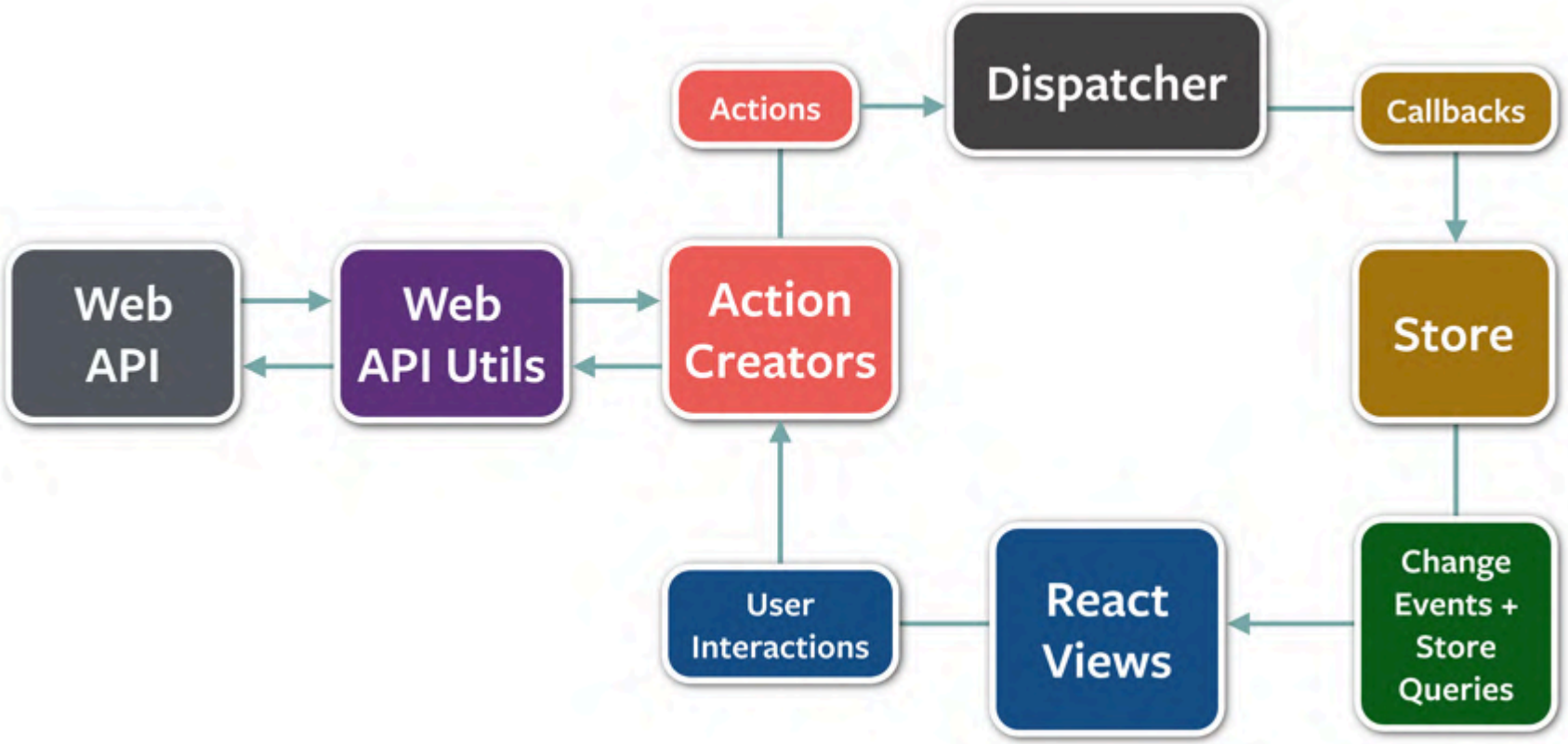
Sequence: Before



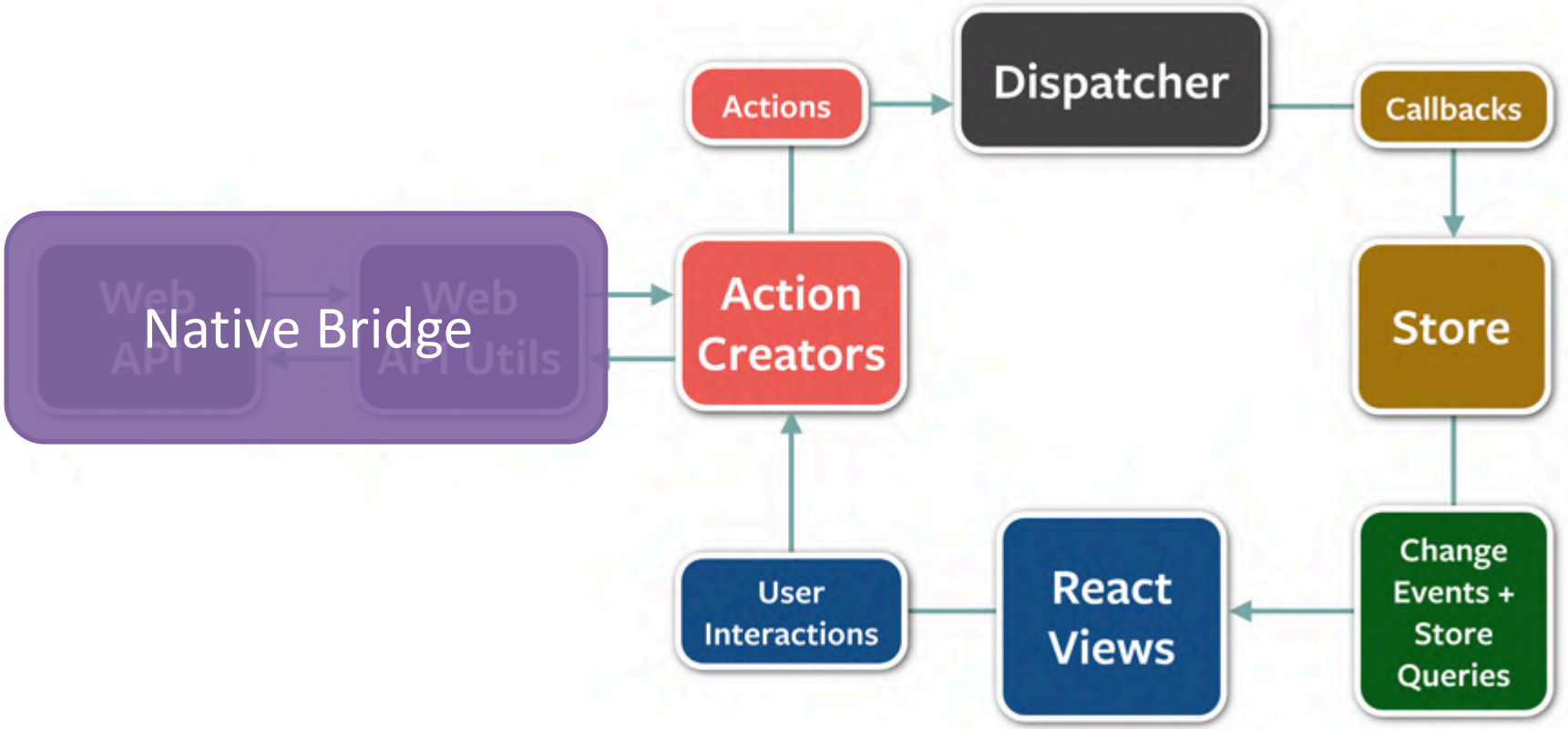
Sequence: After



WebView: Action Creator



WebView: Action Creator



WebView: Action Creator

```
export function clearCompleted() {
  return (dispatch) => {
    if (global.WebViewBridge) {
      WebViewBridge.send('CLEAR_COMPLETED')
      WebViewBridge.onMessage = (message) => {
        if (message === 'CLEAR_COMPLETED_SUCCESS') {
          dispatch({ type: types.CLEAR_COMPLETED })
        }

        WebViewBridge.onMessage = null
      }
    } else {
      if (confirm('Do you really want to clear all completed items?')) {
        dispatch({ type: types.CLEAR_COMPLETED })
      }
    }
  }
}
```

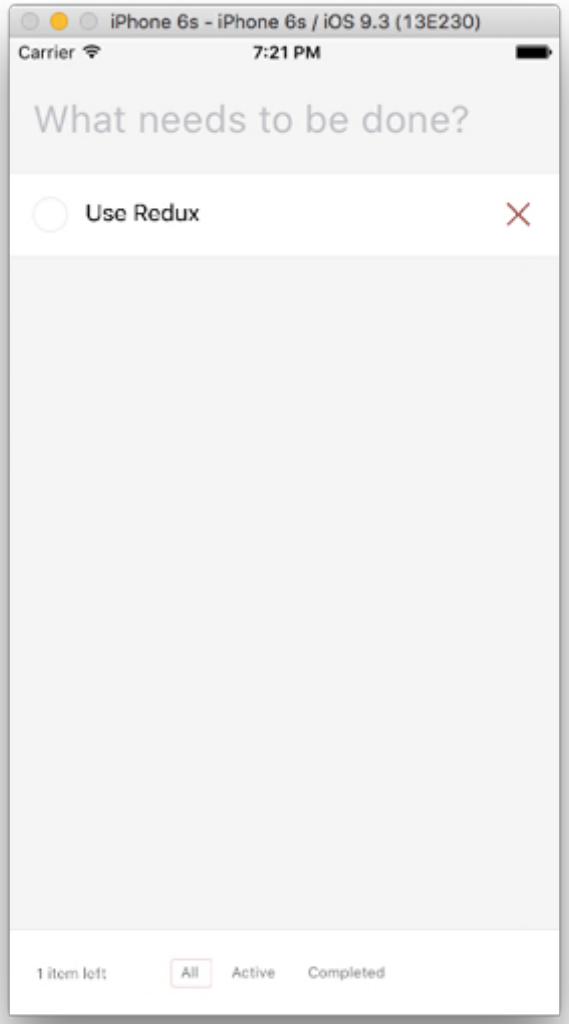
```
onBridgeMessage(message) {
  if (message === 'CLEAR_COMPLETED') {
    Alert.alert(
      'Heads Up!',
      'Do you really want to clear all completed items?', [
        {
          text: 'Yes, I\'m sure.',
          onPress: () => this.refs.webviewbridge.sendToBridge('CLEAR_COMPLETED_SUCCESS'),
          style: 'destructive'
        },
        {
          text: 'No, I don\'t think so.'
        }
      ])
  }
}
```

- Pros
  - UI 部分的代码可以保证在所有平台之间通用
  - 最大程度上使用现成的Web代码，重构工作量最少
  - 学习曲线最平缓
- Cons
  - WebView, Native, Server 三边的双向通信容易变得十分复杂
  - 画面渲染，导航，动画等方面的体验仍然难以和 Native 应用抗衡
  - WebView 的调试问题

中国移动开发者大会
Mobile Developer Conference China 2016

ReactDOM to ReactNative

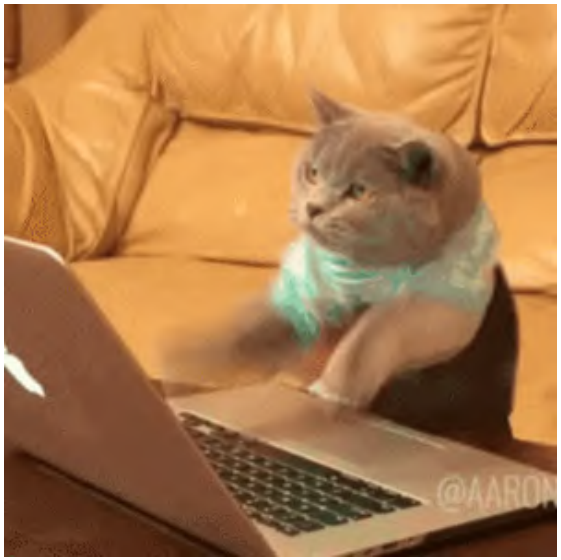
mdcc.csdn.net


iPhone 6s - iPhone 6s / iOS 9.3 (13E230)
Carrier
7:21 PM
What needs to be done?
Use Redux
1 item left
All
Active
Completed

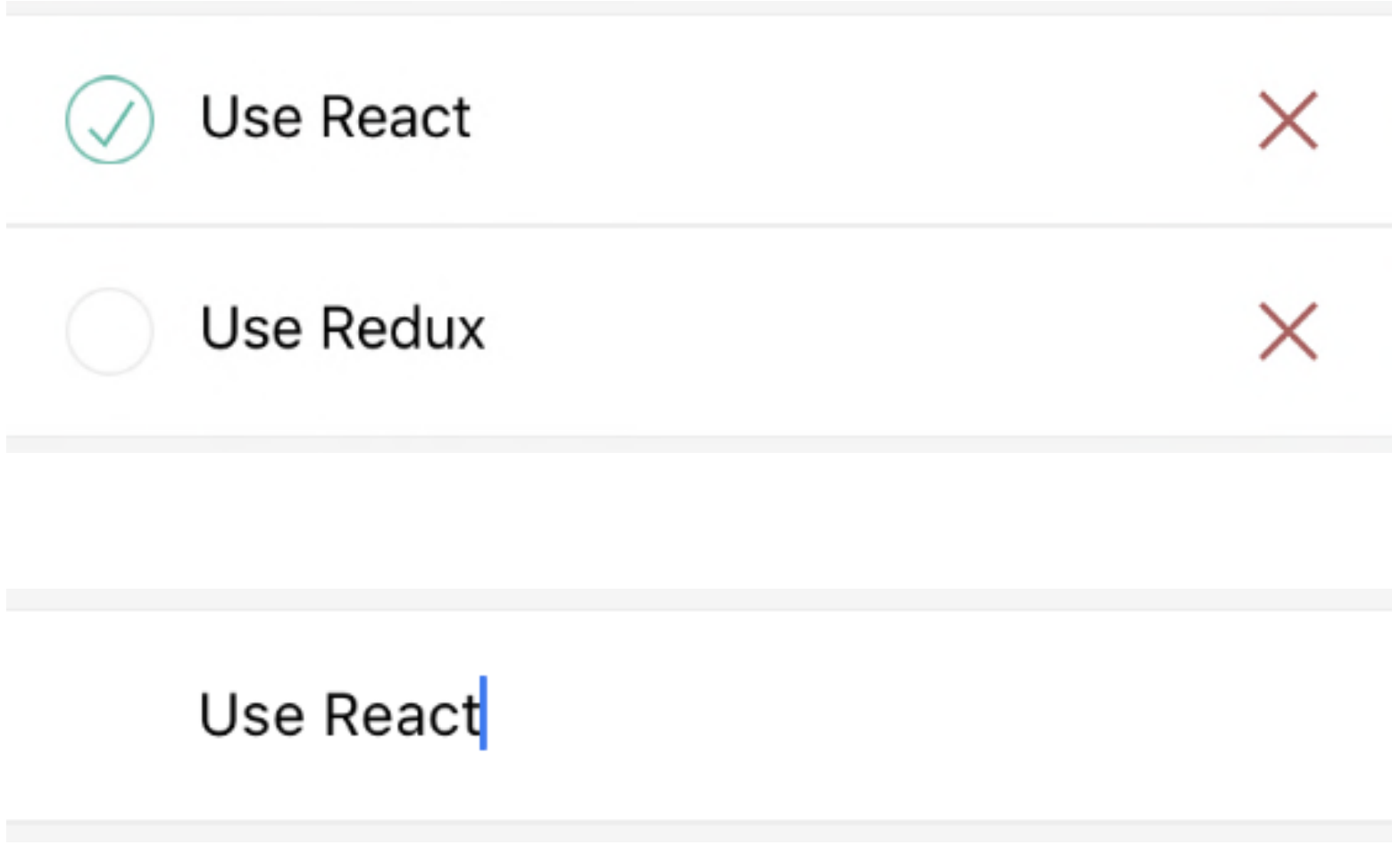
- 将 ReactDOM 代码移植到 ReactNative
- 移植需注意 ReactNative 是在 Native 环境里面执行 JavaScript ，需要排除对 BOM, DOM 等针对浏览器的 API 的依赖，以及样式上对 CSS 的依赖

- 弃用 DOM，置换 HTML 元素成 ReactNative 组件
  - 画面排版活用 View, Text, Image, ScrollView 组件
  - 表单元素活用 TextInput, Slider, Switch, Picker 组件
  - 注意弃用 DOM 的同时也意味着将不能使用面向浏览器的库（比如 jQuery）
- 弃用 CSS，转用 Inline Style 属性
  - 利用 ReactNative 组件的 style 属性来指定样式
  - 活用 JavaScript 特性以及 StyleSheet API 进行样式管理

- ReactNative提供了3种和服务器通信的手段
  - Fetch API
  - XMLHttpRequest API
  - WebSocket API
- 通信层可以根据各自情况进行重构


@AARON

- 用 ReactNative 组件把 TodoMVC 重构成 Native iOS 应用
- 项目代码：
  - https://github.com/toruta39/ReactNativeTodoMVC/tree/feature/port-to-native


Use React
Use Redux
Use React

TodoItem.js: Before

```
<div className="view">
  <input className="toggle"
         type="checkbox"
         checked={todo.completed}
         onChange={() => completeTodo(todo.id)} />
  <label onDoubleClick={this.handleDoubleClick.bind(this)}>
    {todo.text}
  </label>
  <button className="destroy"
          onClick={() => deleteTodo(todo.id)} />
</div>
```

TodoItem.js: After

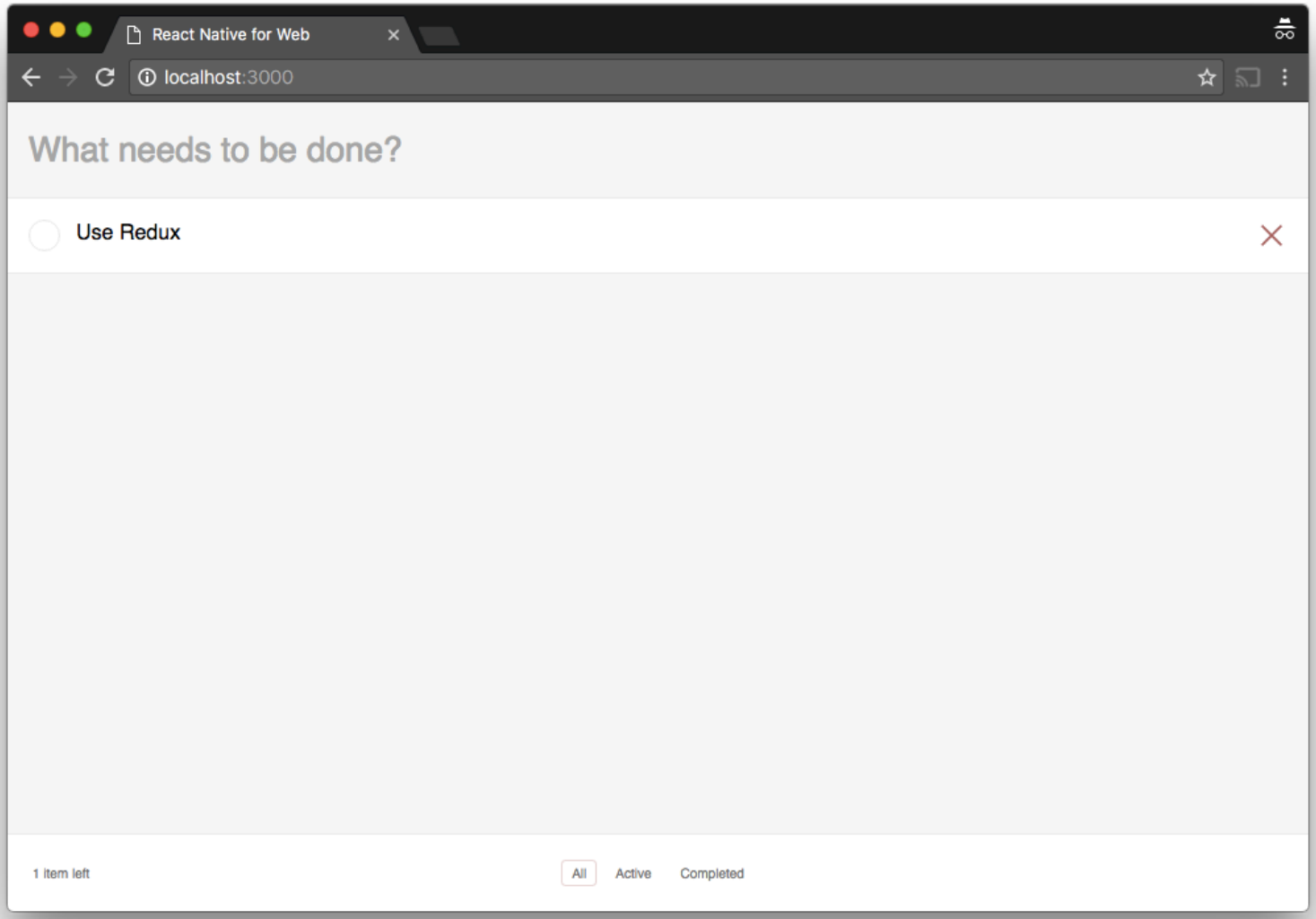
```
<View style={style.innerContainer}>
  <TouchableOpacity style={style.left} onPress={() => completeTodo(todo.id)} >
    <View style={[style.status, todo.completed && style.completedStatus]}>
      {todo.completed && <Image source={require('./assets/tick.png')} style={style.tickImage}/>}
    </View>
  </TouchableOpacity>

  <TouchableOpacity style={style.center} onPress={this.handleDoubleClick.bind(this)}>
    <Text numberOfLines={1} style={[style.text]}>{todo.text}</Text>
  </TouchableOpacity>

  <TouchableOpacity style={style.right} onPress={() => deleteTodo(todo.id)} >
    <Image source={require('./assets/cross.png')} style={style.crossImage}/>
  </TouchableOpacity>
</View>
```

- Pros
  - 可以只用 JavaScript 来开发 Native UI，从用户角度来看大部分情况下 UX 和 Native 应用无异
  - 学习曲线自由，根据需求选择组件学习，或是学习 Native 开发
  - 业务逻辑部分的代码仍然可以通用
- Cons
  - 组件，样式和测试需要重构，Native 组件行为也有差别，工作量庞大
  - 调试以及性能分析时需要 Native 开发的知识背景
  - 移植后的组件代码因为和 DOM 完全分离，无法直接用于 Web 平台

中国移动开发者大会
Mobile Developer Conference China 2016

ReactNative for Web

mdcc.csdn.net


React Native for Web
localhost:3000
What needs to be done?
Use Redux
1 item left
All
Active
Completed

- ReactNative 通过 Bridge 抽象了各个 Native 平台的 UI 实现，让开发者可以只用 React 和 JavaScript 开发各平台的 Native 应用

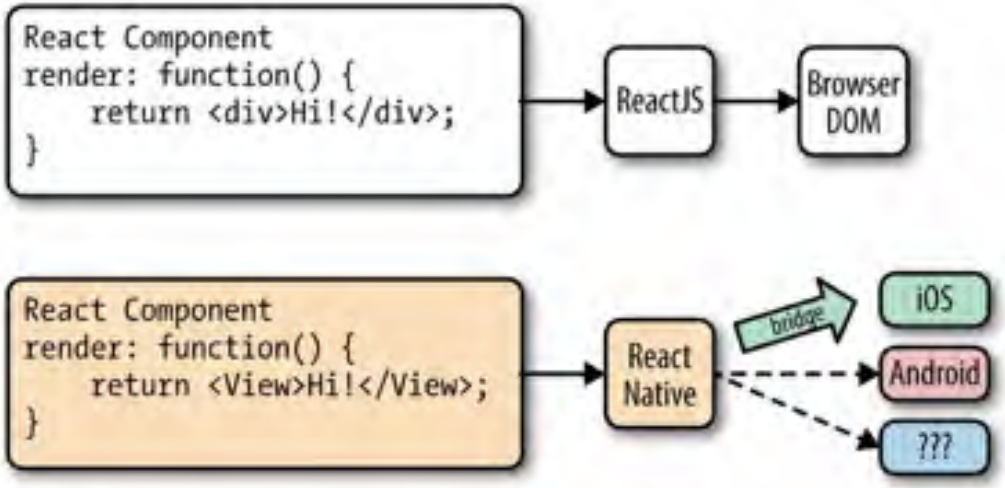


Figure 2-2. React can render to different targets

From *Learning React Native: Building Native Mobile Apps with JavaScript*
http://bit.ly/learning-react-native

- 现在开源社区开发的一部分 Bridge
  - ptmt/react-native-macos
  - ReactWindows/react-native-windows
  - CanonicalLtd/react-native
  - ProjectSeptemberInc/gl-react

- 如果为 ReactNative 的组件提供 Web 平台的 Bridge 的话，就可以兼容 Web 平台，实现 iOS/Android/Web 跨三平台的代码复用
- ReactNative for Web
  - https://github.com/necolas/react-native-web

- 让刚才重构的 Native iOS 应用兼容 Web 平台
- 项目代码：
  - https://github.com/toruta39/ReactNativeTodoMVC/tree/feature/use-react-native-web

- Pros
  - 可以实现 iOS/Android/Web 三平台的代码复用
- Cons
  - 项目还不够成熟，有许多尚未完成的 API
  - Native 组件和 Web 组件行为上有相当的区别
  - 现状还不足以在正式环境上利用，但是作为一个发展方向值得考虑

Takeaways

- ReactNative 让 JavaScript 开发者可以不用学习新的语言，直接开发 Native 应用
- Native 应用的开发风格可以自由选择，可以保守地只用 WebView 封装，也可以激进地完全用 Native 组件重构，提供最佳的用户体验
- ReactNative 内部的 Bridge 的概念，让 ReactNative 拥有无限的可能性，只要有人开发 Bridge 的话就可以兼容新平台

- ReactNative 将会巩固其 JavaScript 跨平台方案的决定性低位，今后也会有越来越多的使用案例
- React 团队在 React.js Conf 上提到了 ReactNative代码在 Web 的复用问题，或许可以期待官方或者社区的动作
- 通过 ReactNative 开发跨平台应用， iOS/Android/Web 开发者可以在 High Level 使用相同的语言进行开发，增加各平台团队之间的交集，减少跨平台开发的支出

MDCC 2016

中国移动开发者大会

Mobile Developer Conference China 2016

Q&A

mdcc.csdn.net

MDCC 2016

中国移动开发者大会

Mobile Developer Conference China 2016

Thanks for coming

mdcc.csdn.net