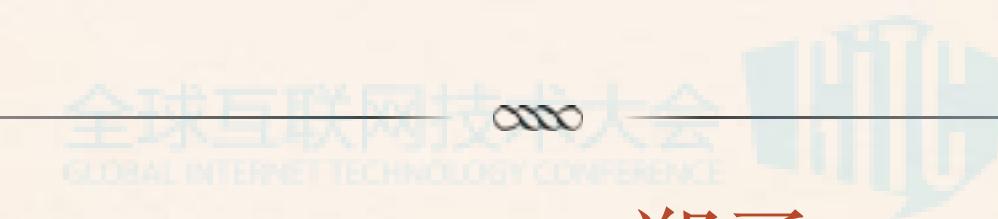




RN-WEB



AUTHOR: 郑勇

携程基础部





现为携程计算机技术（上海）有限公司前端高级架构师，主要从事前端技术探索，前端性能优化，前端框架设计与编写，目前主要工作是

开发 $RN\text{-}web$, 在 RN 基础上实现最终的

iOS , $Android$, $h5$ 三端打通。

郑勇



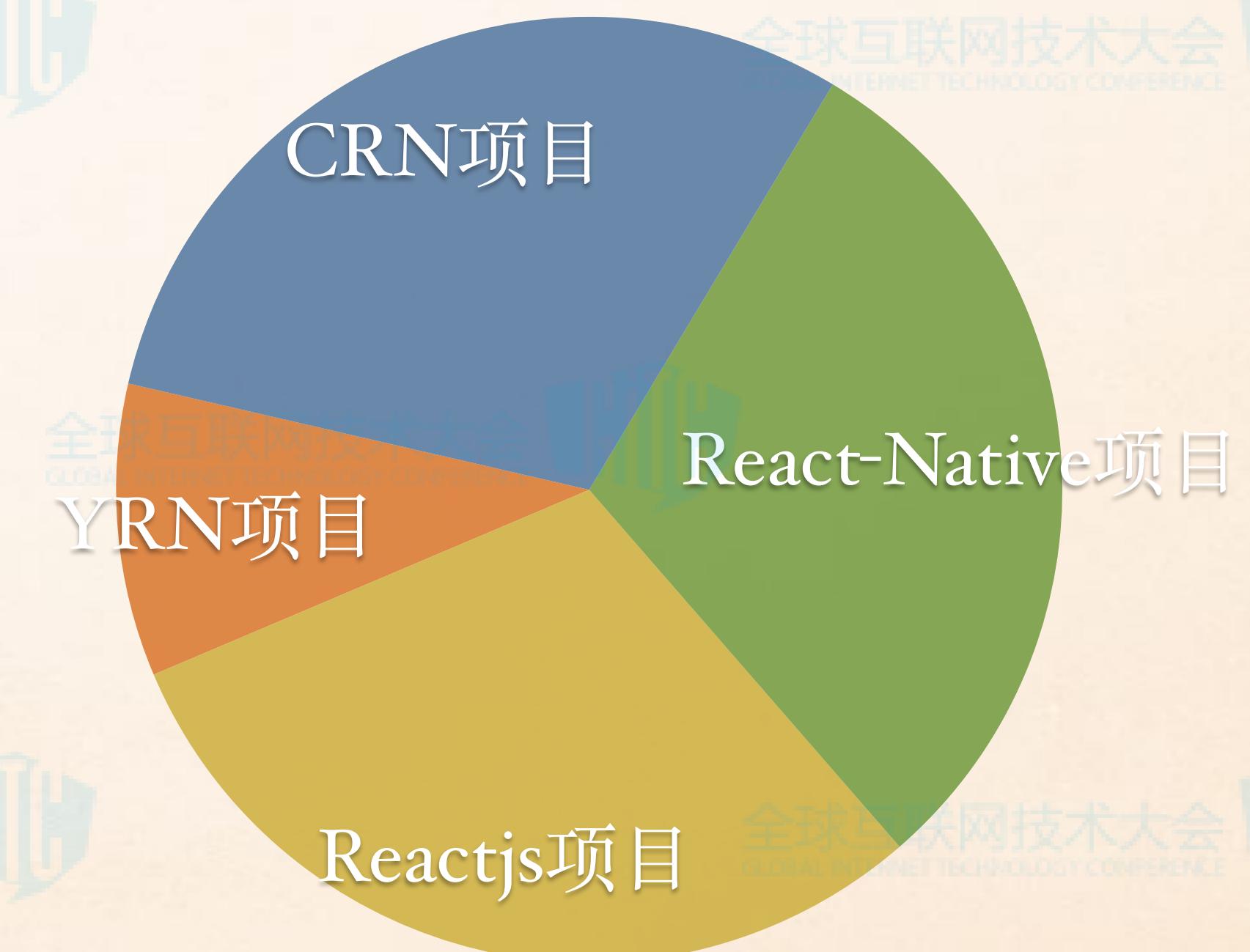
前端需求



WHAT'S CRN-WEB

❖ 在 *React, React-Native* 基础之上，结合 *CRN* 最终打通项目在 *iOS, Android, H5* 三端运行效果的框架。

❖ 支持以下项目类型：





从HELLO WORLD开始



```
import {Component} from 'react';
import {View,Text,AppRegistry} from 'react-native';

class HelloWorld extends Component{
  render(){
    return(
      <View>
        <Text>HelloWorld</Text>
      </View>
    )
  }
}

AppRegistry.registerComponent('HelloWorld', ()=> HelloWorld);
```

Package Tool:

```
'react': path.join(__dirname,
  '../..../node_modules/@ctrip/react'),
```

'react-native':

```
path.join(__dirname, '../..../src',
  '/react-native'),
```



线上项目效果展示



全球互联网技术大会
携 程 特 价 机 票



全球互联网技术大会
携 程 特 价 好 礼



全球互联网技术大会

全球互联网技术大会

全球互联网技术大会

全球互联网技术大会

效果对比



效果对比2

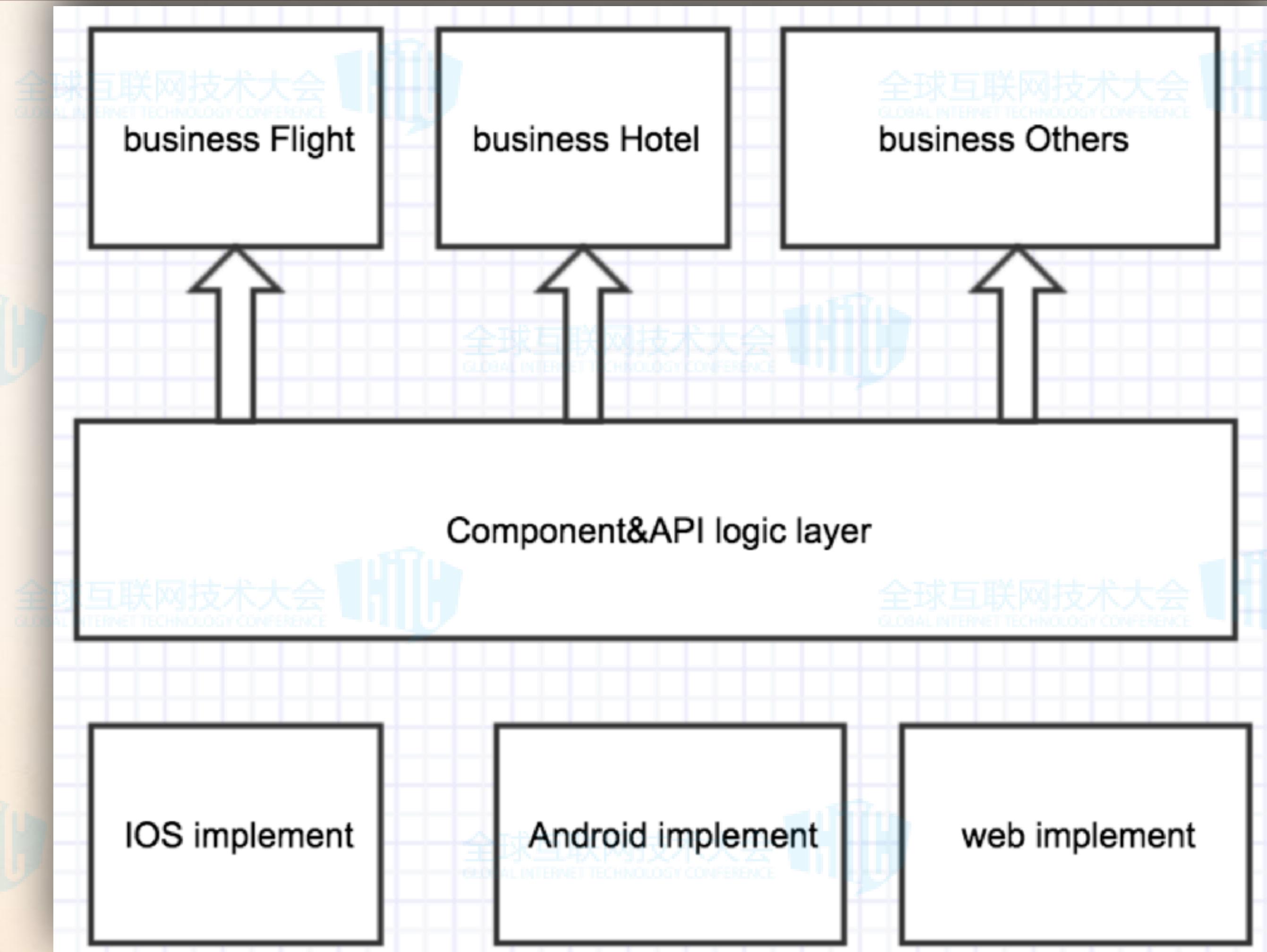


效果对比3



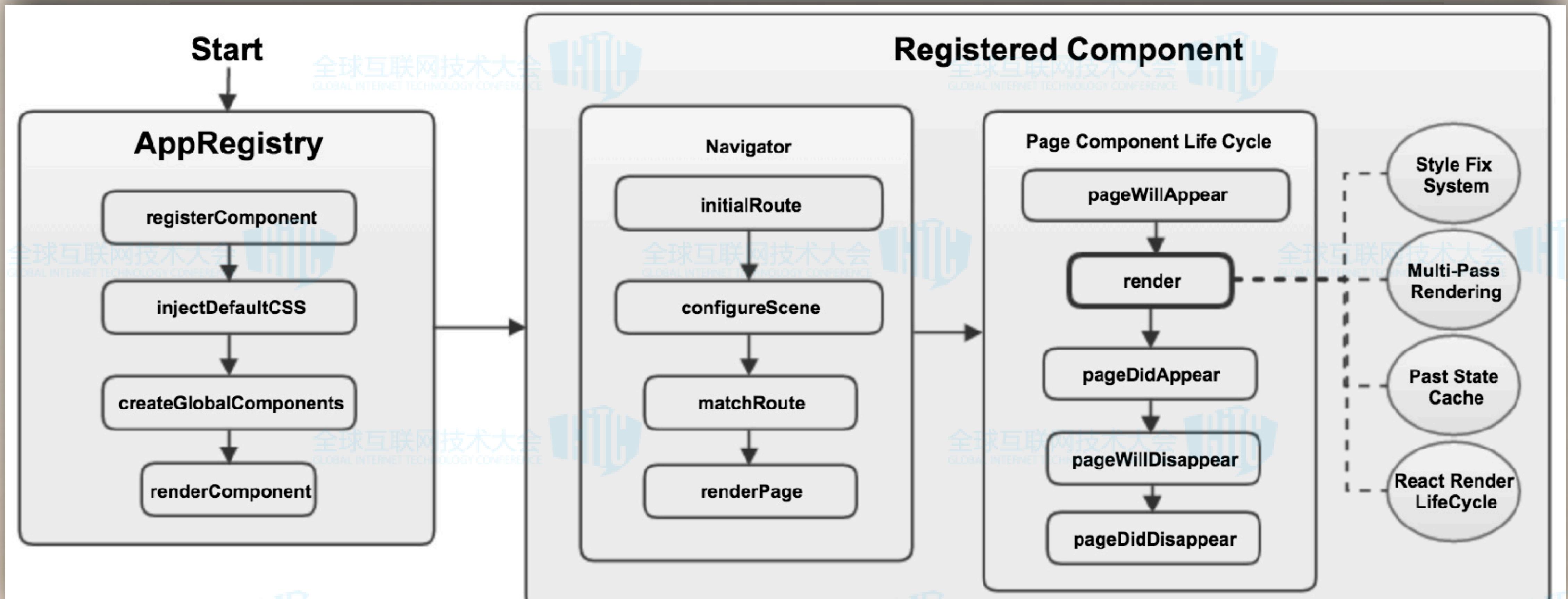


CRN-WEB总体设计



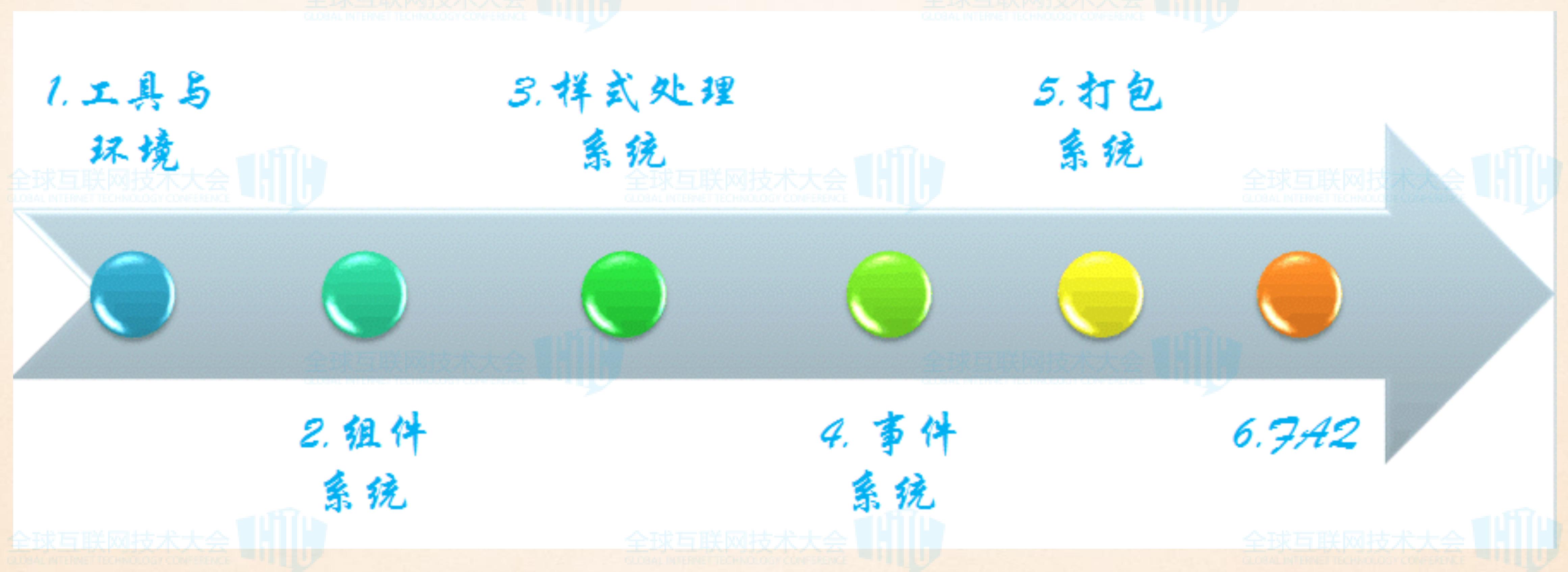


CRN-WEB运行流程





CRN-WEB具体实现





1. 工具与环境



Node环境与node插件



html5, css, js, ES6, ES7



Webpack及其插件, AST, uglifyJS



Babel及其插件



React, React-Native

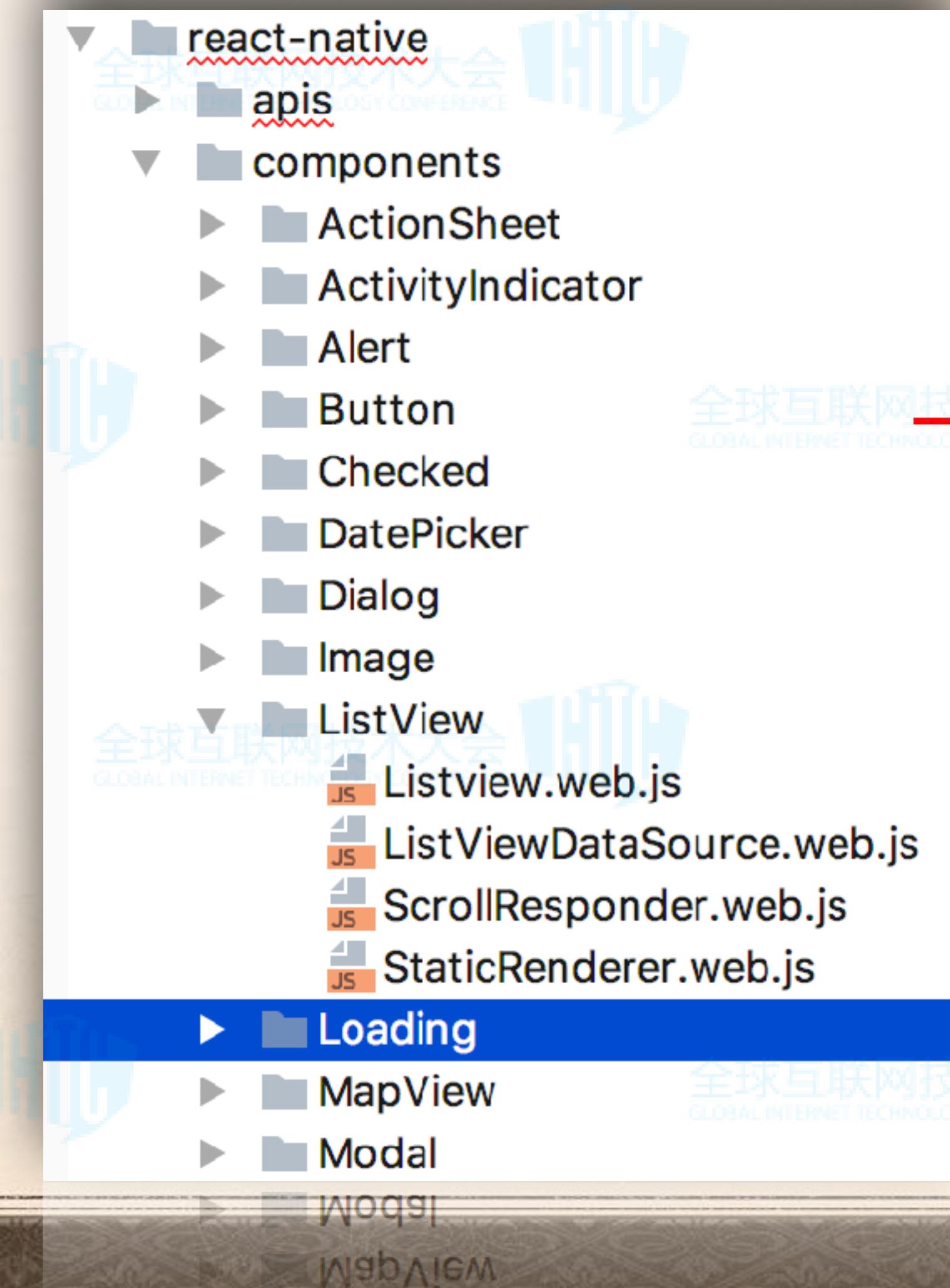
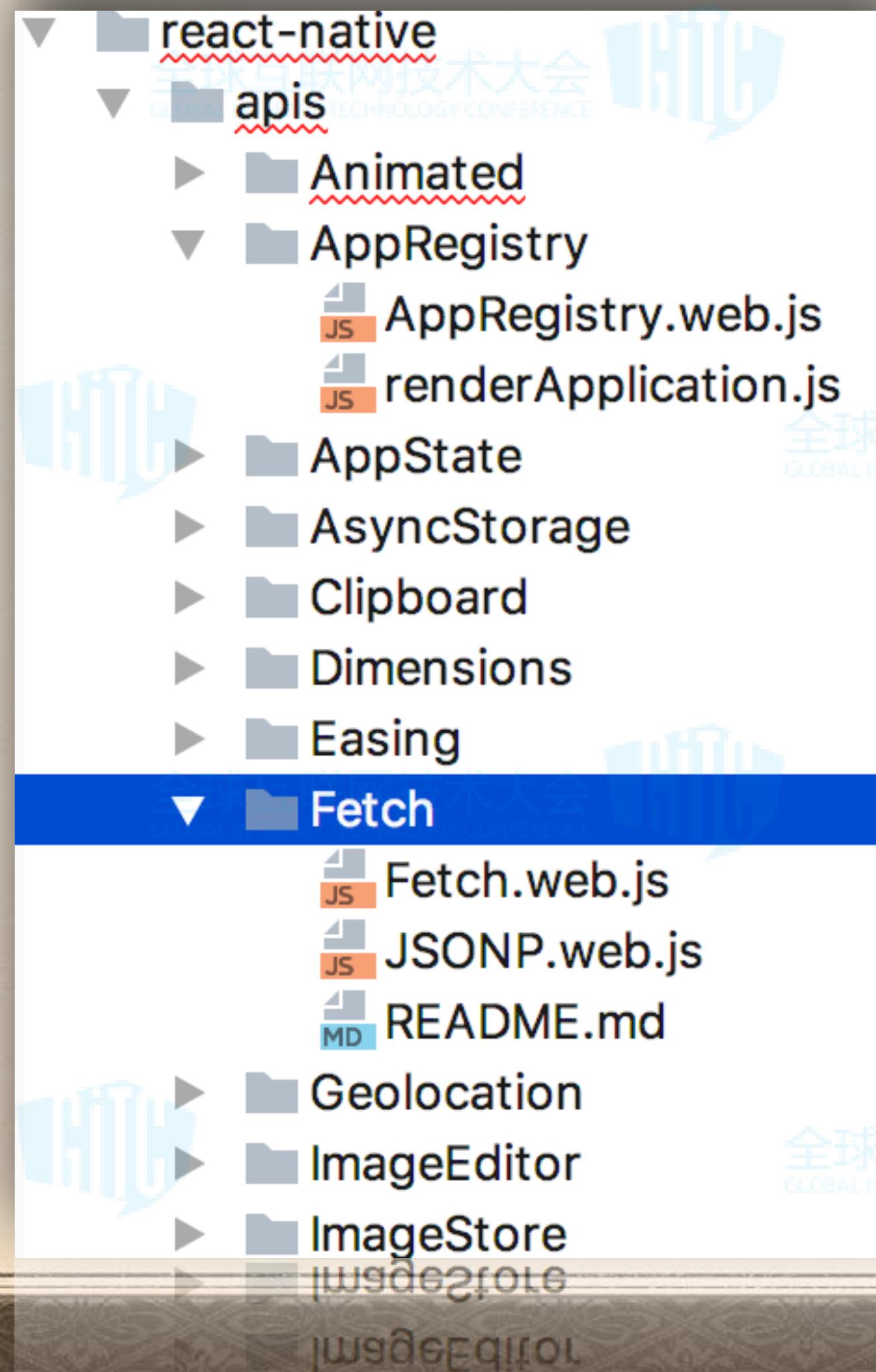


第三方插件



2.1 组件系统

REACT-NATIVE组件系统：30+40+30(19+23)(44)



2.2 组件系统

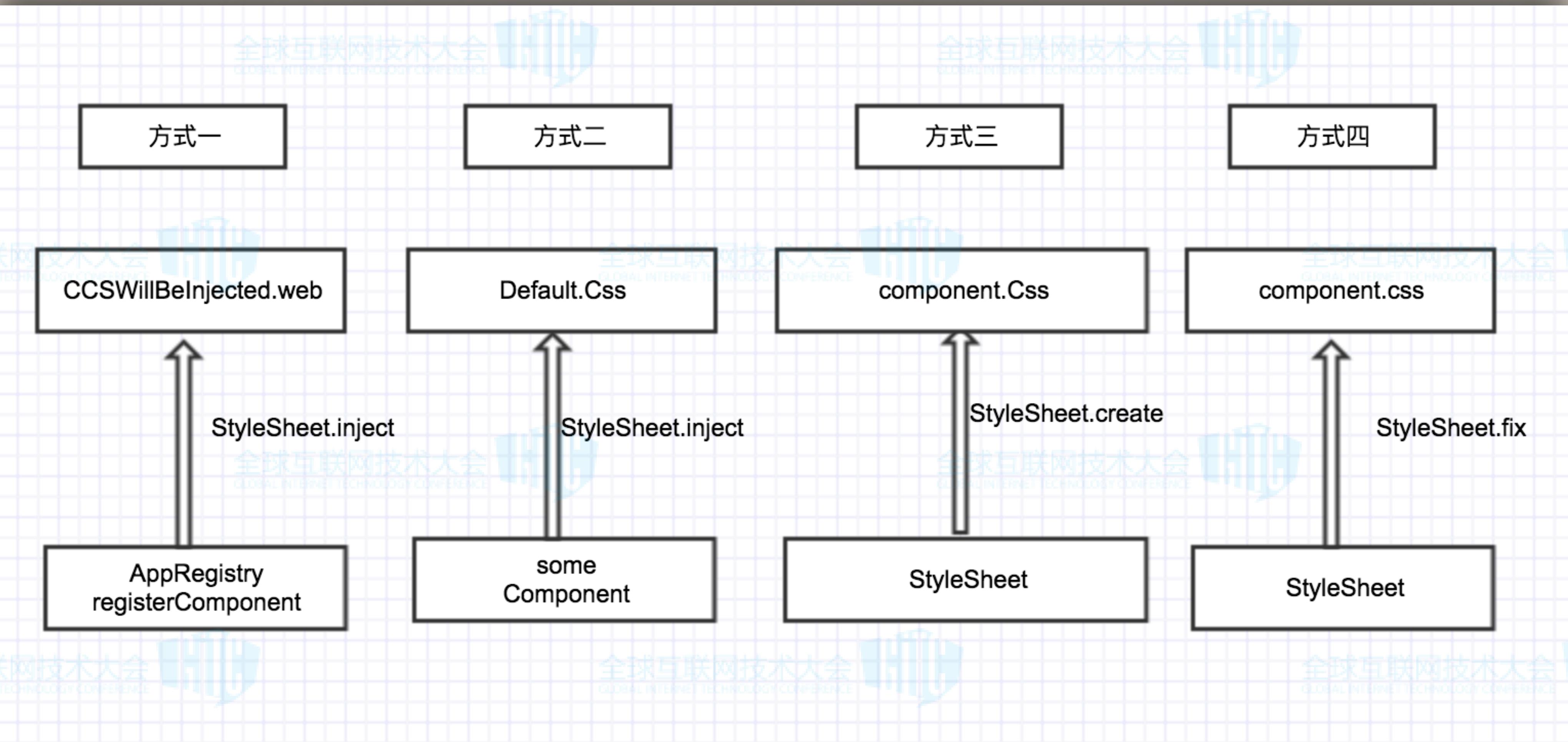
CRN组件系统：70

The screenshot shows a file tree on the left and a code editor on the right. The file tree under 'src' contains a 'crn' folder with various components like ActionSheet, Ajax, Fetch, HeaderView, HtmlText, Loading, LocalStorage, Location, Map, MessageCenter, Models, skin, Stores, SwipeListItem, User, Util, ABTesting.js, AddressBook.js, and App.js. The code editor displays the 'App.js' file, which defines the 'App' component.

```
29     }
30
31  ↘ ↗ export default class App extends Component {
32      constructor(props) {
33          super(props);
34          this.url = this.props.url;
35          this.urlQuery = this.props.urlQuery||{};
36      }
37
38      init({pages, navigationBarConfig}) {
39          this.pages = pages;
40
41          this.navigationBarHidden = (navigationBarConfig
42          this.navigationBarColor = (navigationBarConfig
43      }
44
45      componentWillMount() {
46          this.navigation = new Navigation(this);
47          global.currentApp = this;
48      }
49
50      render () {
51          let navigationBarConfig = new NavigationBarConfig();
52
53          return (
54              <Navigator
55                  ...
56              >
57          );
58      }
59  }
```



3.1 样式处理系统





3.2 样式处理系统



- ❖ fixUnit,

基本样式单位是rem，转化样式单位，

- ❖ fixBorder

对borderStyle进行一般化处理

- ❖ fixFlexBox

处理flexBox样式集，自动判断选用2009, 2011, 2012等进行兼容性样式处理

- ❖ fixTransform

将rn数组形式的fixed成普通的web格式，处理transformMatrix等

- ❖ fixBoxShadow

处理shadowOffset, shadowRadius, shadowColor, shadowOpacity等等

- ❖ fixPaddingMargin

处理paddingHorizontal, marginVertical这些样式

- ❖ fixCssName

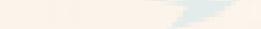
处理样式名称，前缀等



4.1 事件处理系统

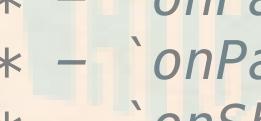


使用了PanResponder，它提供一个对触摸响应系统的
Responder的可预测的包装，和React-Native保持一致的事件
处理流程

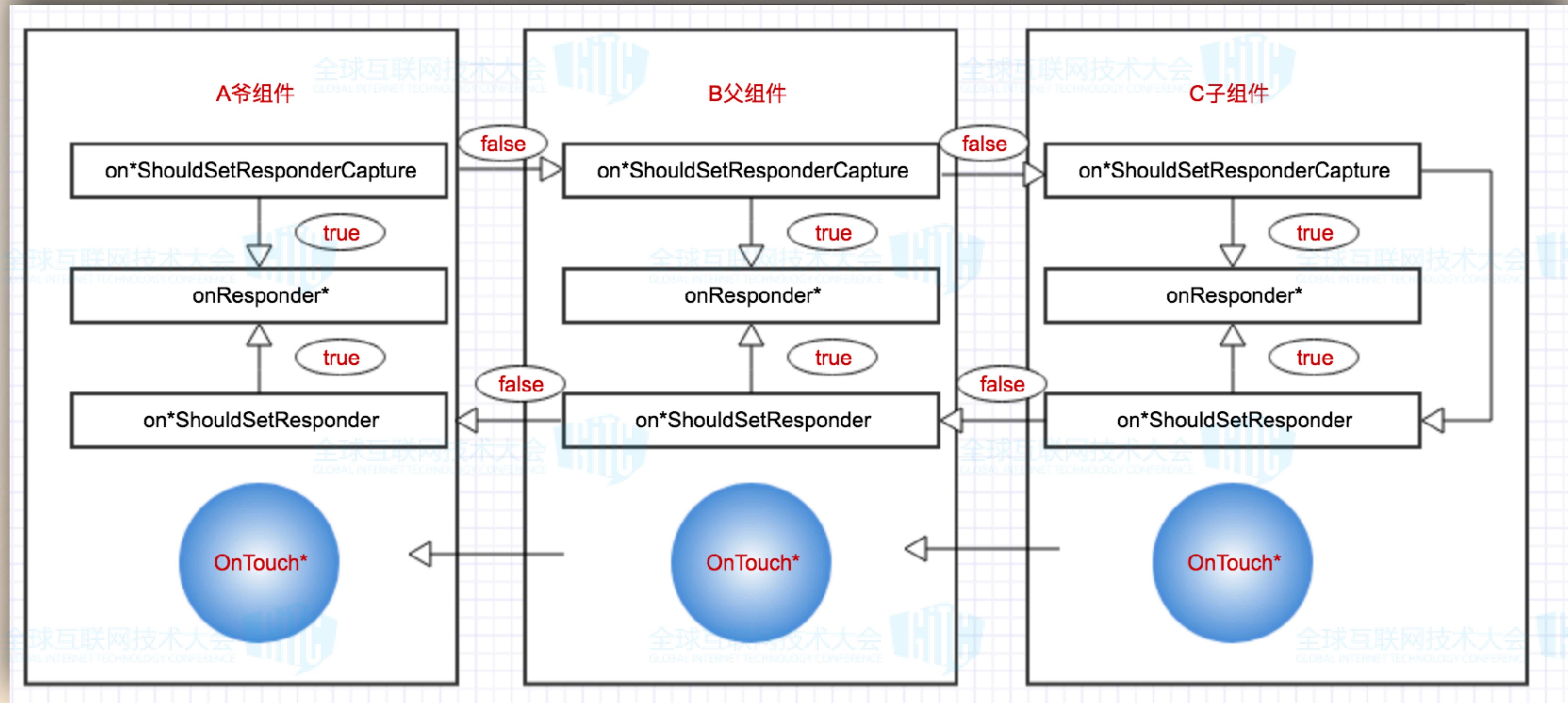


❖ onStartShouldSetResponder(i3)

- * - `onMoveShouldSetPanResponder: (e, gestureState) => {...}`
- * - `onMoveShouldSetPanResponderCapture: (e, gestureState) => {...}`
- * - `onStartShouldSetPanResponder: (e, gestureState) => {...}`
- * - `onStartShouldSetPanResponderCapture: (e, gestureState) => {...}`
- * - `onPanResponderReject: (e, gestureState) => {...}`
- * - `onPanResponderGrant: (e, gestureState) => {...}`
- * - `onPanResponderStart: (e, gestureState) => {...}`
- * - `onPanResponderEnd: (e, gestureState) => {...}`
- * - `onPanResponderRelease: (e, gestureState) => {...}`
- * - `onPanResponderMove: (e, gestureState) => {...}`
- * - `onPanResponderTerminate: (e, gestureState) => {...}`
- * - `onPanResponderTerminationRequest: (e, gestureState) => {...}`
- * - `onShouldBlockNativeResponder: (e, gestureState) => {...}`

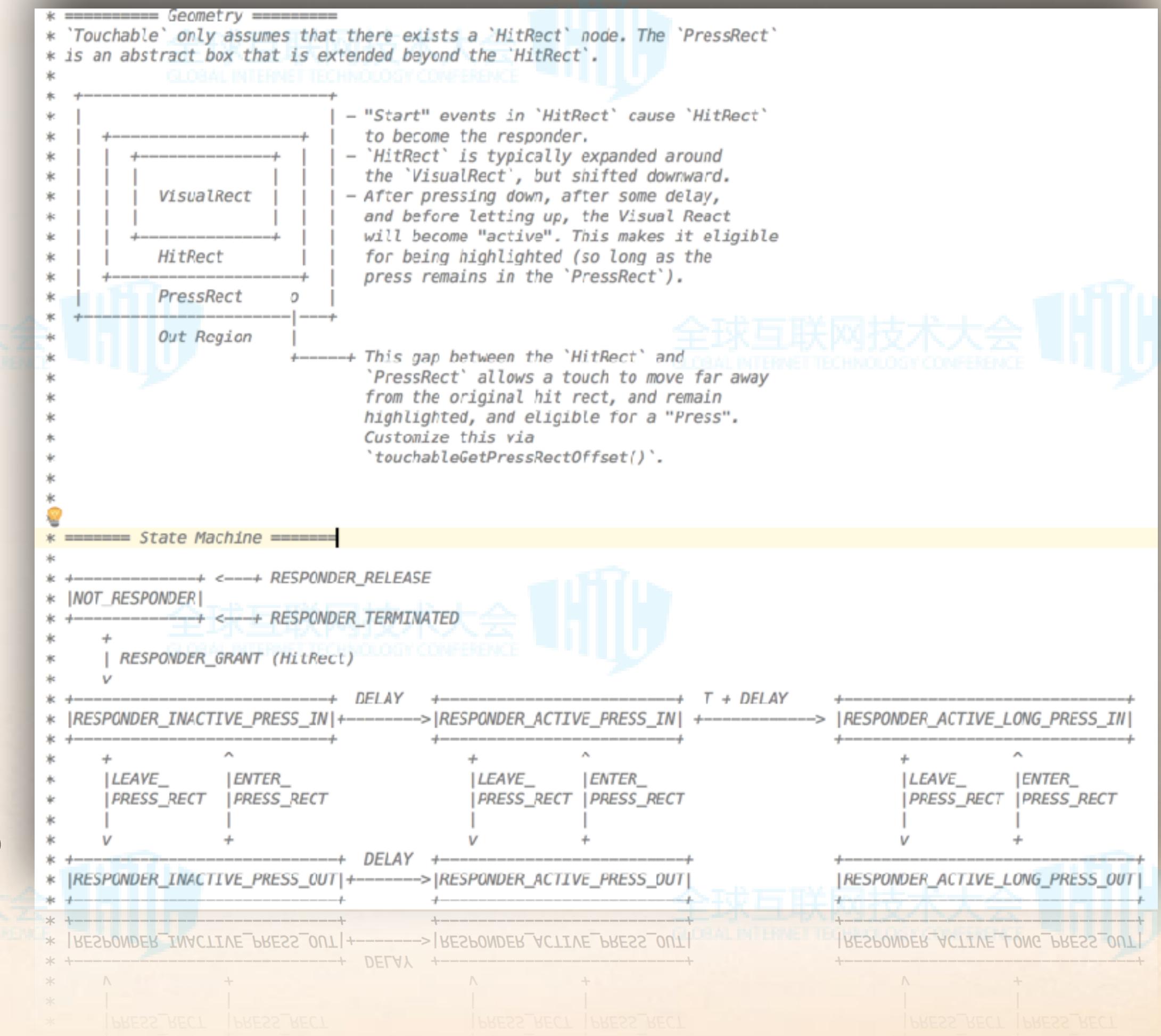


4.2 父子组件事件传递



4.3 事件处理系统

- ◆ Touchable.web
 - ◆ TouchableHighlight.web
 - ◆ TouchableNativeFeedback.web
 - ◆ TouchableOpacity.web
 - ◆ TouchableWithoutFeedback.web





5 CRN-WEB打包工具



Prepare (Entry, Version, Environment, Third Component Check)

Webpack(presets, loaders, plugins)

Babel(Syntax process, Tree Shaking, Transform<sync,async>)

Create (Java project, DotNet project, static project)

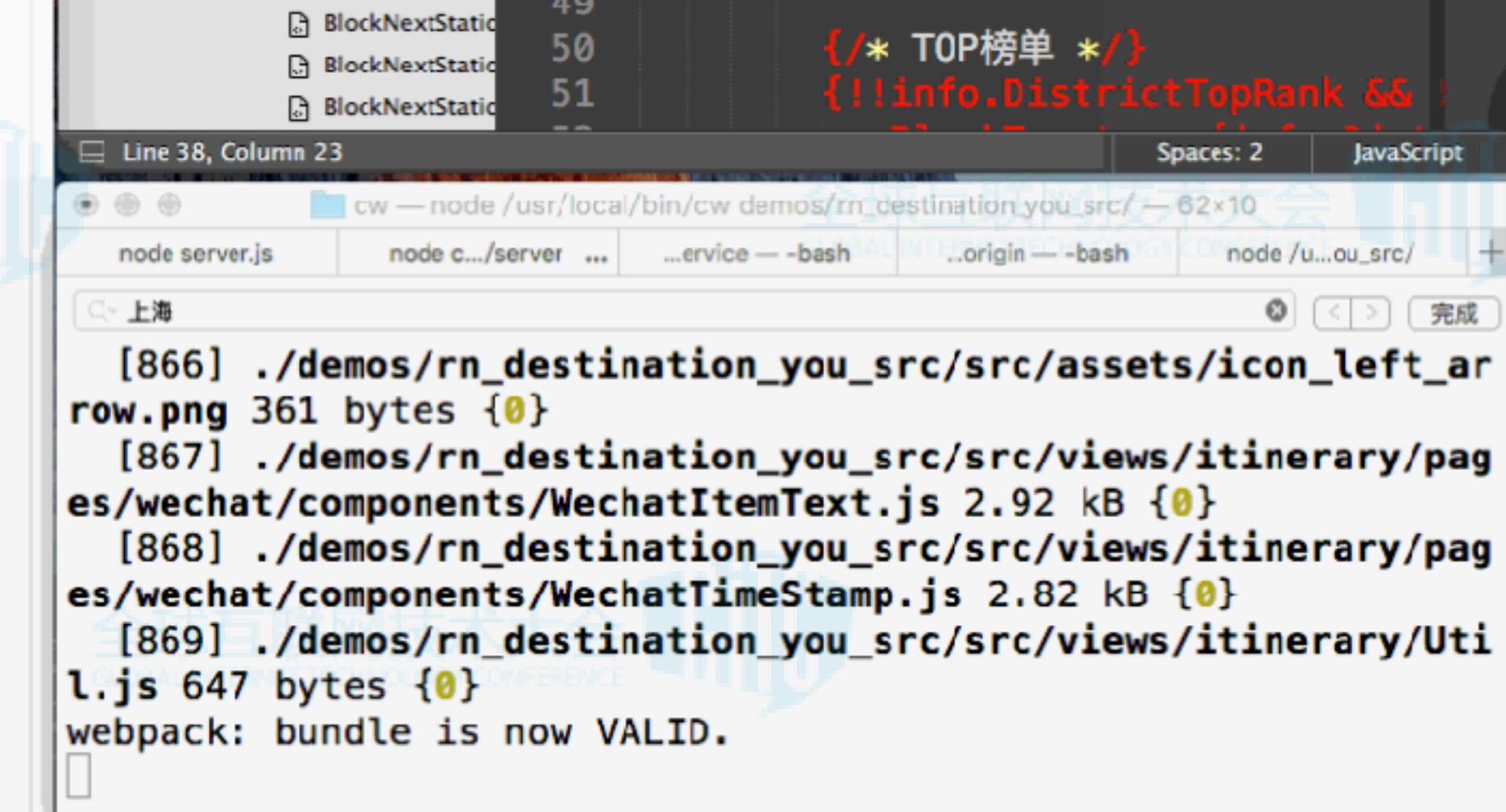
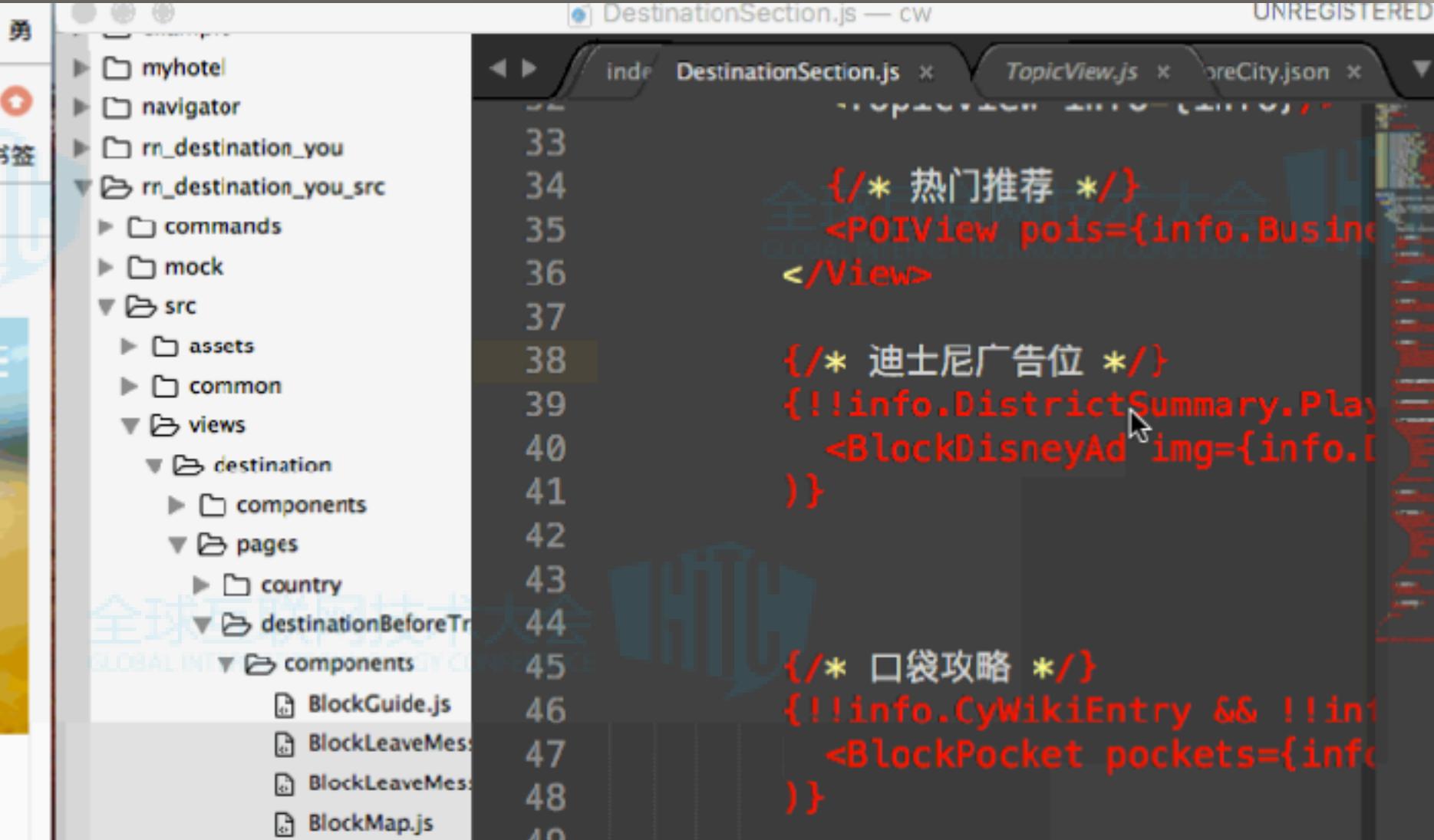
Setup (Dev, Release, Live Model)

Statistic (size, dependency...)





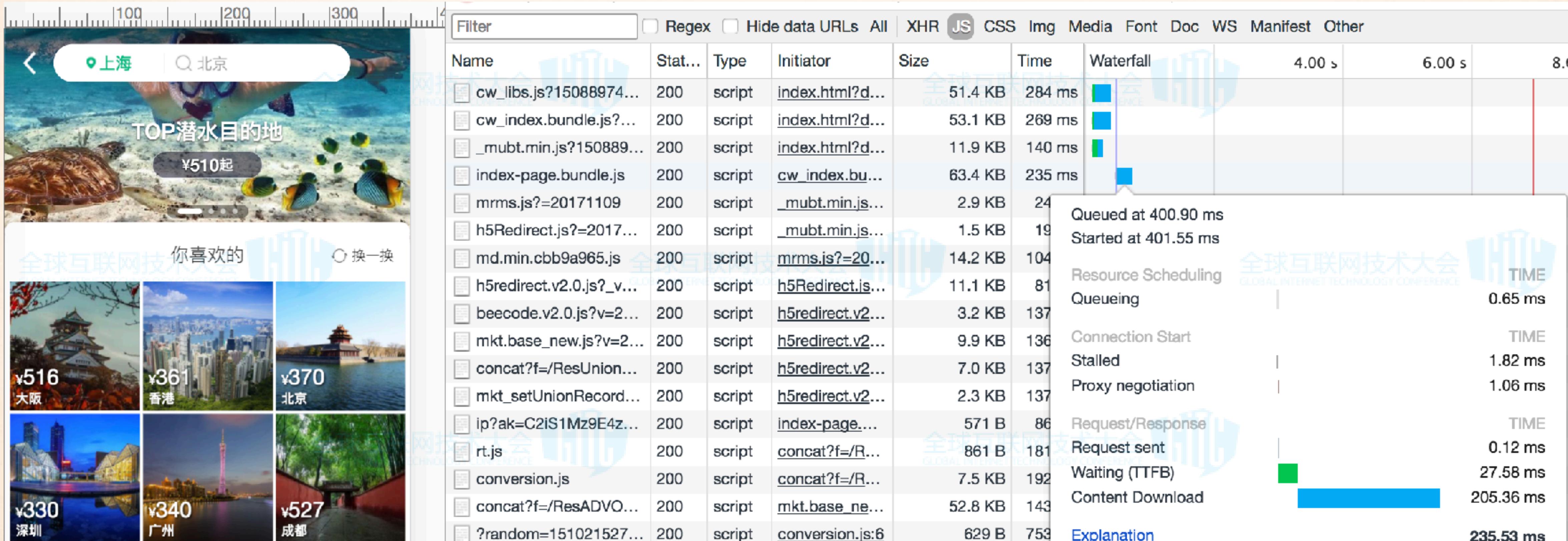
5.1 CRN-WEB打包工具



The image displays a composite screenshot illustrating the CRN-WEB packaging process. On the left, a mobile browser window shows a travel-related application interface for Shanghai, featuring a search bar, weather info (26-34°C), and various travel categories like 微领队, 美食, 购物, 玩乐, and 住宿. A prominent Disney advertisement for "玩转上海迪士尼" is visible. On the right, a code editor (Visual Studio Code) shows the file `DestinationSection.js`, which contains JavaScript code for rendering POI views and Disney advertisements. Below the code editor, a terminal window shows the command `node server.js` being run, followed by the output of a webpack build command for the `rn_destination_you_src` directory, indicating a successful bundle creation.



5.2 CRN-WEB打包工具



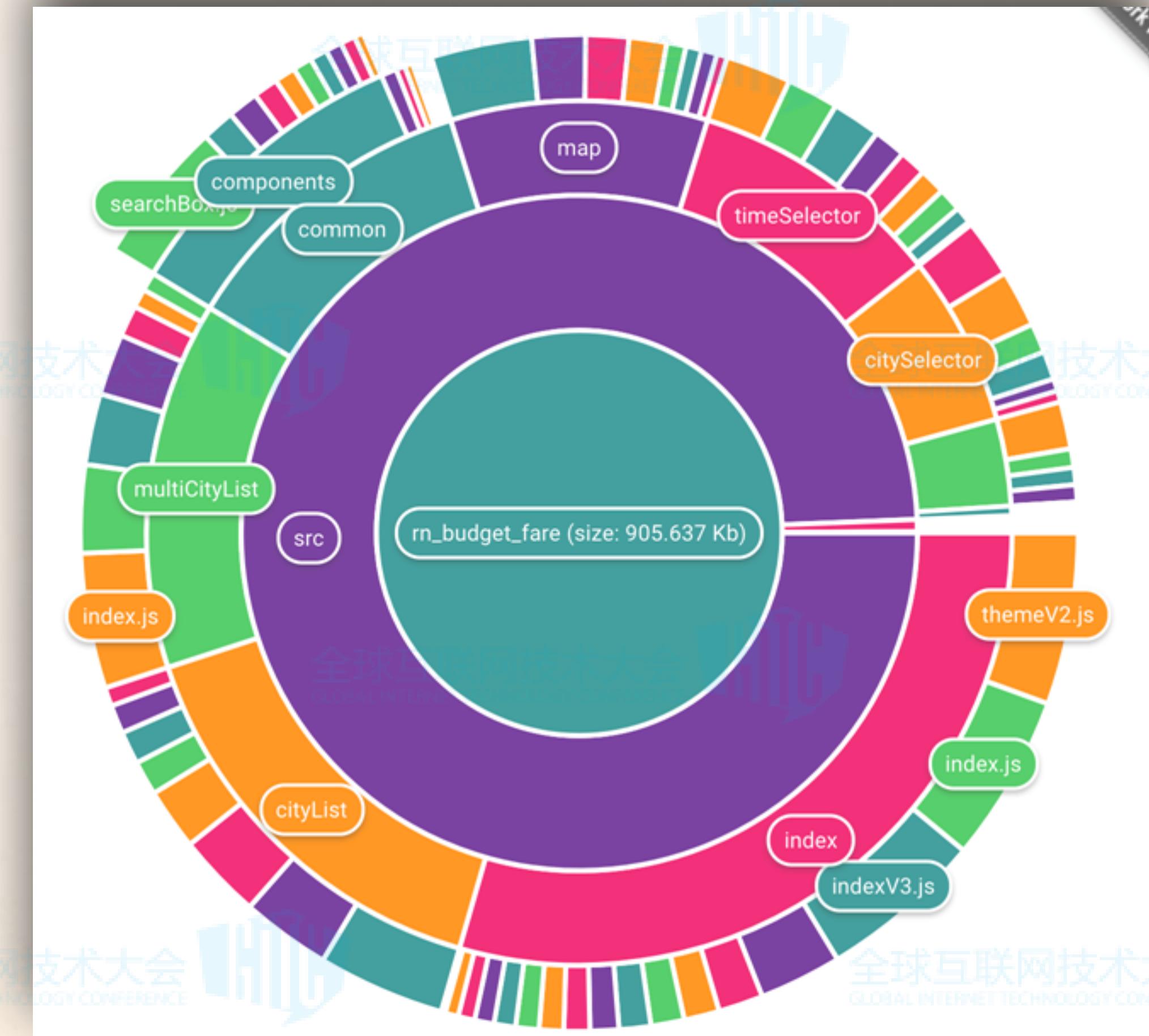
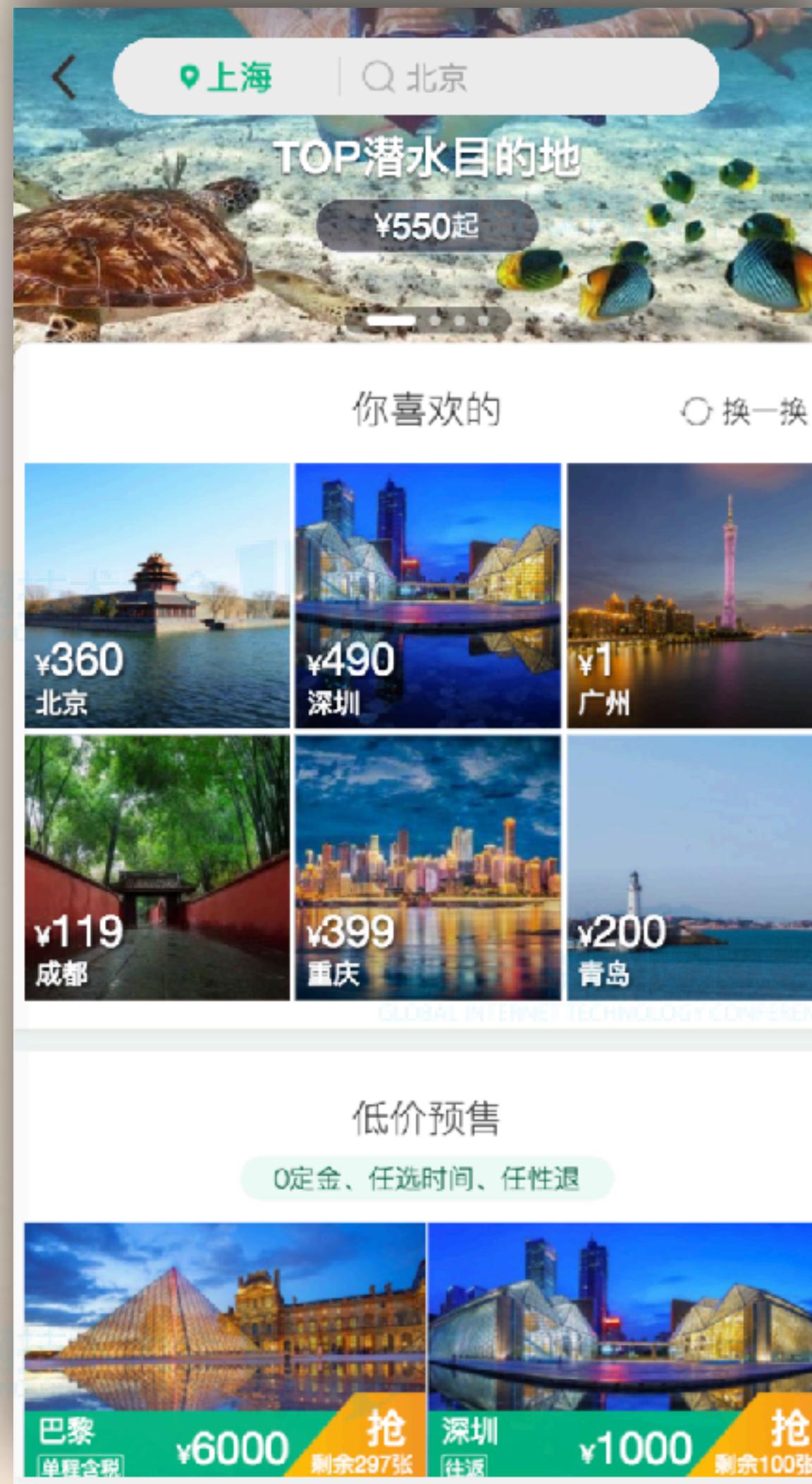


5.3 CRN-WEB打包工具



reasons	name	module	location		
index-page	109	.../bu/rn_budget_fare/main.js	21:45-168		
modules					
modules	id	name	size	chunks	flags
167	.../bu/rn_budget_fare/src/index/index.js	49 KiB	3		built
178	.../bu/rn_budget_fare/src/common/FStyleSheet.js	3 KiB	0 1 2 3 4 5 6 8 9		built
179	.../bu/rn_budget_fare/src/common/Log.js	1049 bytes	0 1 2 3 4 5 6 7 8		built
180	.../bu/rn_budget_fare/src/common/utils.js	2 KiB	0 1 2 3 4 5 6 7		built
181	./src/crn/LinearGradient/LinearGradient.web.js	4 KiB	0 1 2 3 4 5 6 9		built
182	./src/react-native/components/Touchable/TouchableHighlight.web.js	5 KiB	0 1 2 3 4 5 7 8		built
185	.../src/react-native/components/Touchable/TouchableHighlight.web.js	2 KiB	0 1 5 3 4 2 1 8		built

5.4 CRN-WEB打包工具





6 关于许可协议



We use a standard BSD license paired with an additional patent grant for most of our open source projects. For brevity, we call this combination the Facebook BSD+Patents license. We've compiled some answers to common questions about the additional patent grant:

Does the additional patent grant in the Facebook BSD+Patents license terminate if I create a competing product?

No. 开发和Facebook竞争的产品，是否授权终止？—— NO

Does the additional patent grant in the Facebook BSD+Patents license terminate if I sue Facebook for something other than patent infringement?

No. 如果因非专利侵权起诉Facebook，是否授权终止？—— NO

Does the additional patent grant in the Facebook BSD+Patents license terminate if Facebook sues me for patent infringement first, and then I respond with a patent counterclaim against Facebook? 如果Facebook先起訴我专利侵权，我使用专利反诉，是否授权终止？
—— NO，除非你的反诉使用的专利和Facebook的BSD+专利软件有关。

No, unless your patent counterclaim is related to Facebook's software licensed under the Facebook BSD+Patents license.

Does termination of the additional patent grant in the Facebook BSD+Patents license cause the copyright license to also terminate?

No. 是否Facebook BSD+专利授权中，专利授权终止导致版权也终止？—— NO

React许可协议解惑



Facebook官方



难点：

1.对浏览器的兼容性问题的处理，

答：对浏览器的兼容性问题的处理，对于一般性的样式支持差异问题在Stylesheet.fix中做统一处理，特殊的兼容性问题做特殊处理。

2.Reactjs本身太大，

答：长期来看和兄弟公司去哪儿合作，提供Reactjs的替代品；短期内使用打包工具尽可能优化打包size，使用如Treeshake, dead-code-remove，等等技术

3.部分低端机型在滚动中，特别卡顿，

答：部分低端机型在滚动中，特别卡顿，对React进行优化，批量更新，减少嵌套层级等。

4.依赖管理,版本问题等等

答：依赖管理,版本问题等等，先固定版本，后定升级目标等等。

5.工程量巨大，React-native在变化

答：固定阶段性成果，按计划升级

6.细节打磨

如：*chrome 12px, web上1px太粗*