

React Native 实践

张自萌

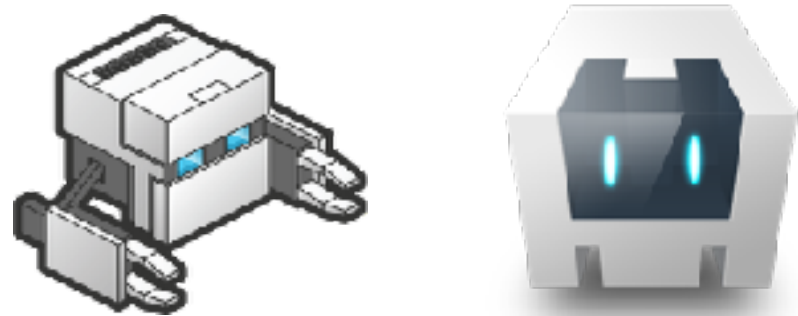
大纲

- 常见APP开发技术以及对比
- React Native 两种开发模式对比
- React Native 的原理介绍
- ListView 性能优化
- 动画优化
- 白屏优化
- 拆包按需加载

常见的移动端开发技术



HTML5



Hybrid

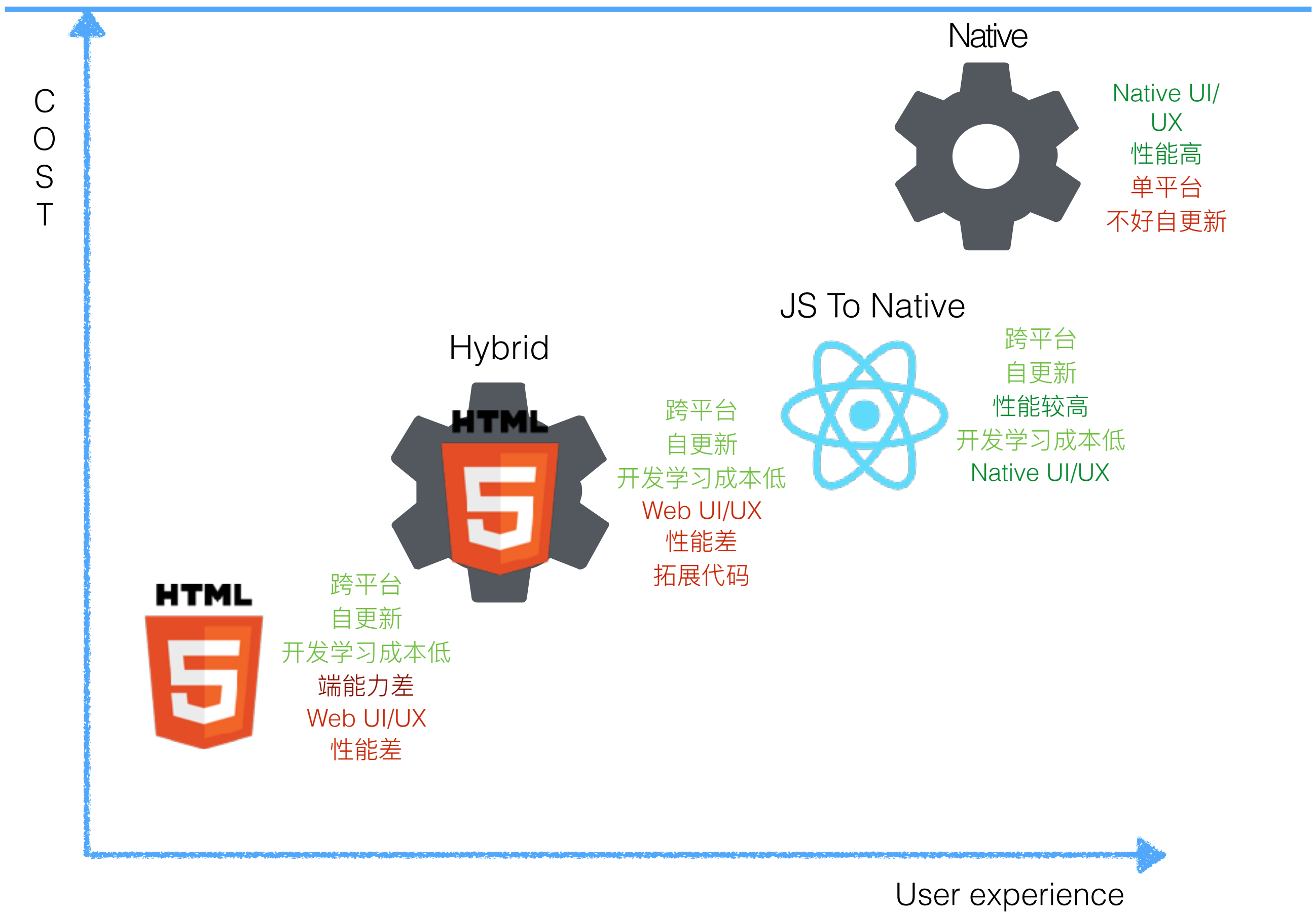


JS to Native



Native

技术对比



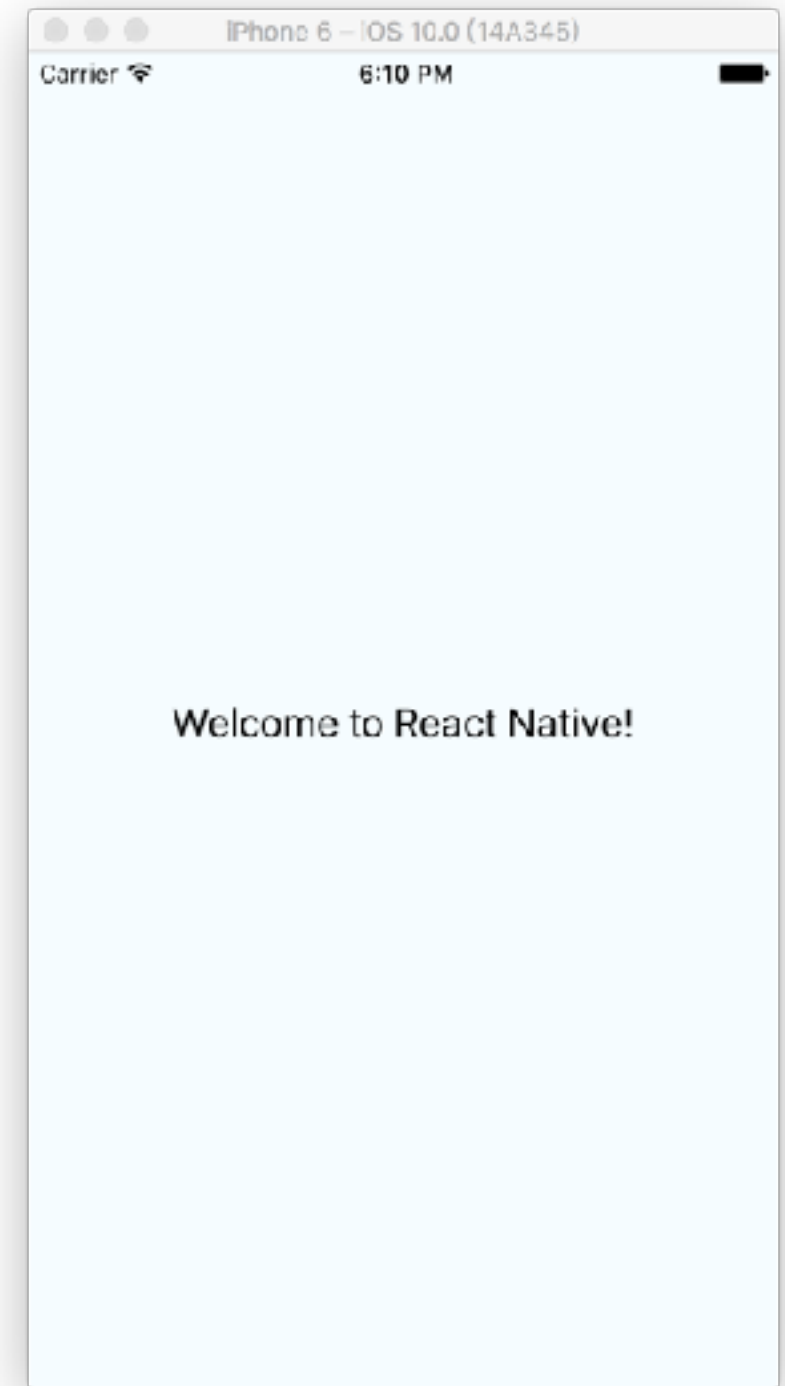
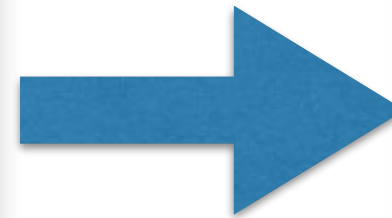
Hello World

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

export default class HelloWorld extends Component {
  render() {
    return (
      <View style={styles.container}>
        <Text style={styles.welcome}>
          Welcome to React Native!
        </Text>
      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#F5ECFF',
  },
  welcome: {
    fontSize: 20,
    textAlign: 'center',
    margin: 10,
  }
});

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



与web前端开发差异

- JSX vs HTML
- CSS-layout vs css
- ECMAScript 6 vs ECMAScript 5
- React native View vs DOM

单RN开发模式



微粉

Native + RN开发模式



Native



RN



RN



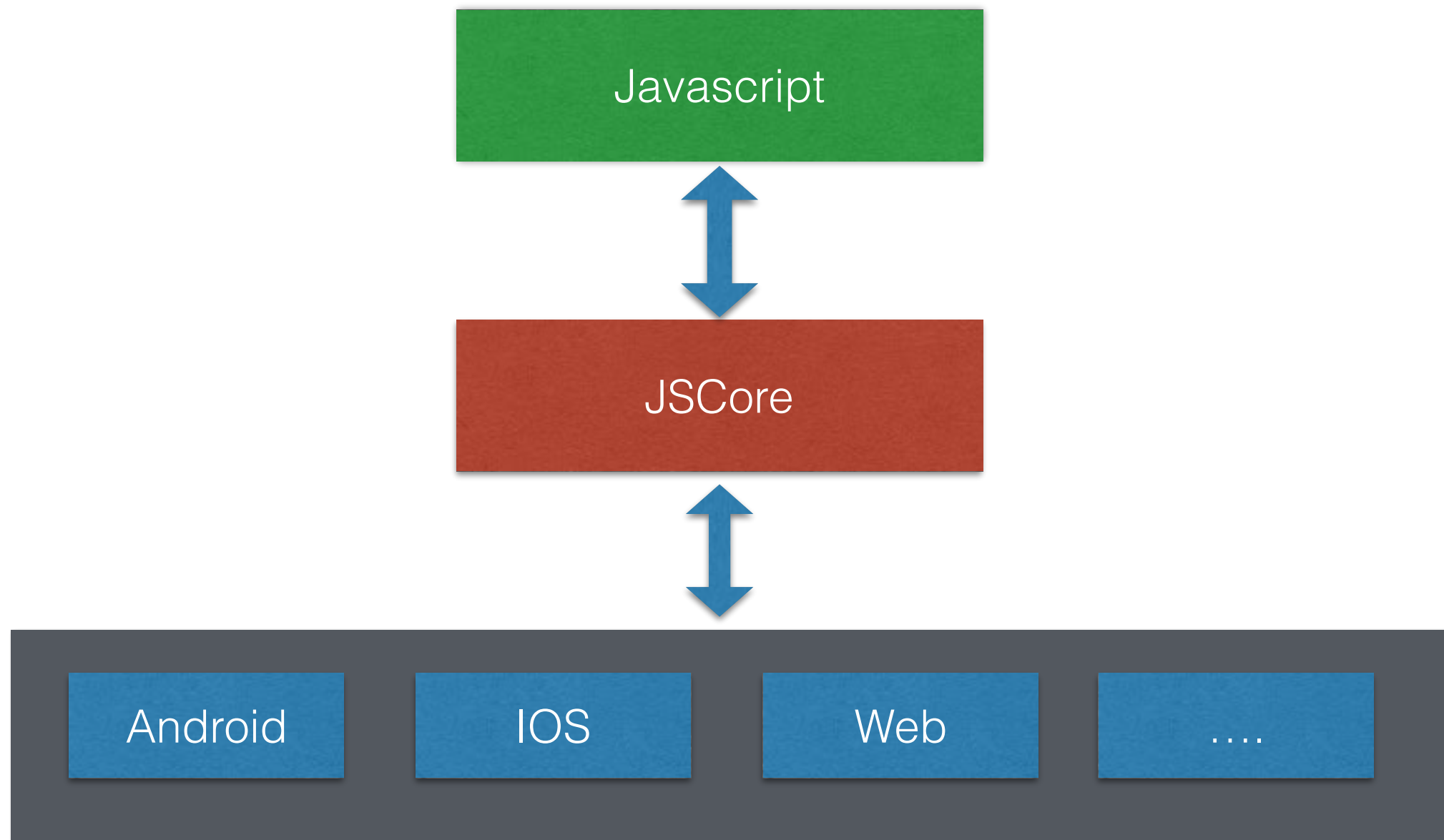
RN



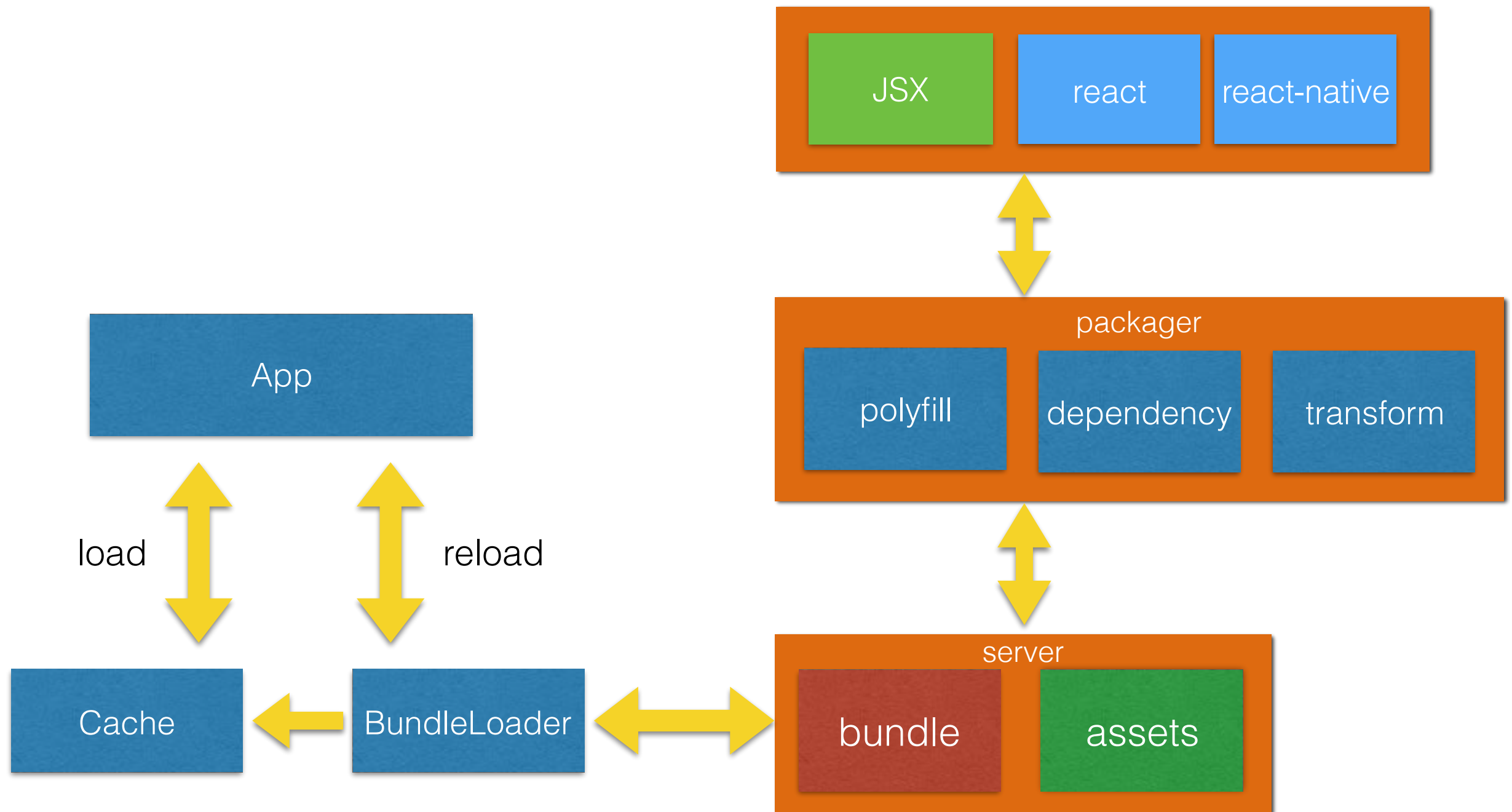
单RN VS Native + RN

- 单RN
 - 适合新项目
 - 风险较大，回滚成本高
 - 复杂应用在低端机上有性能问题
- Native + RN
 - 适合老项目
 - 风险较低，回滚成本低
 - 需要改造native代码

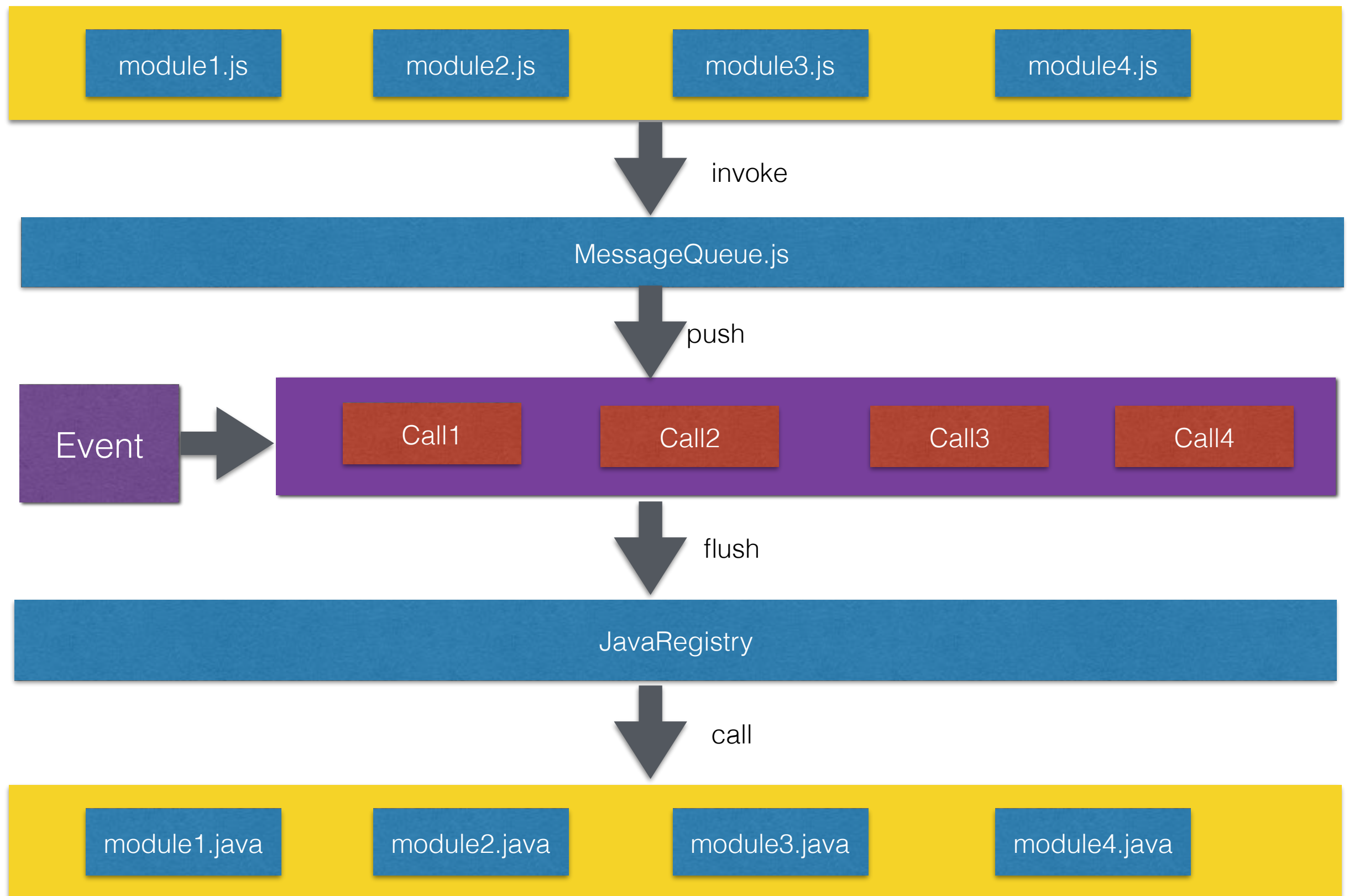
认识React Native



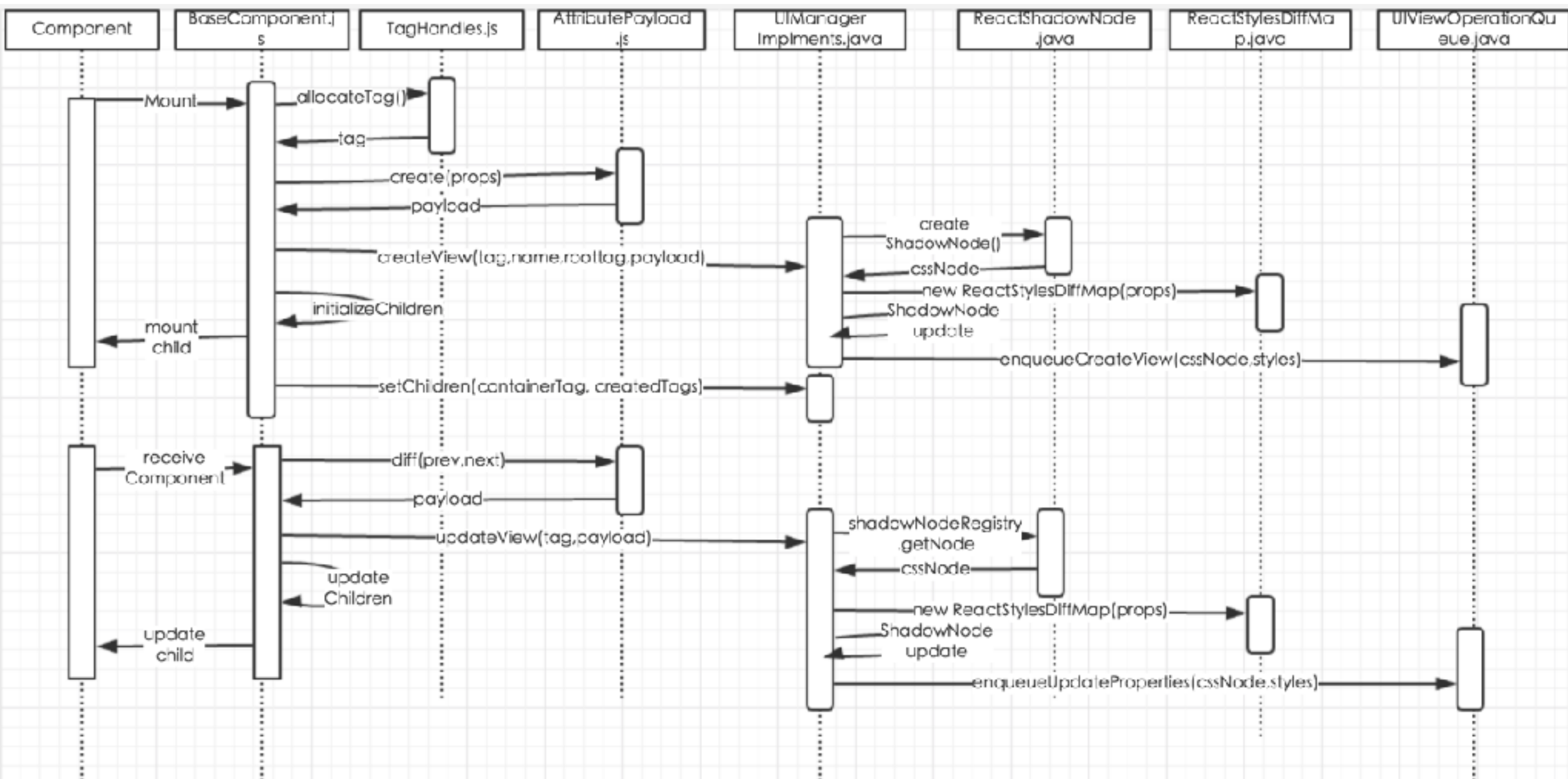
React Native加载机制



React Native 通信机制



React Native UI原理



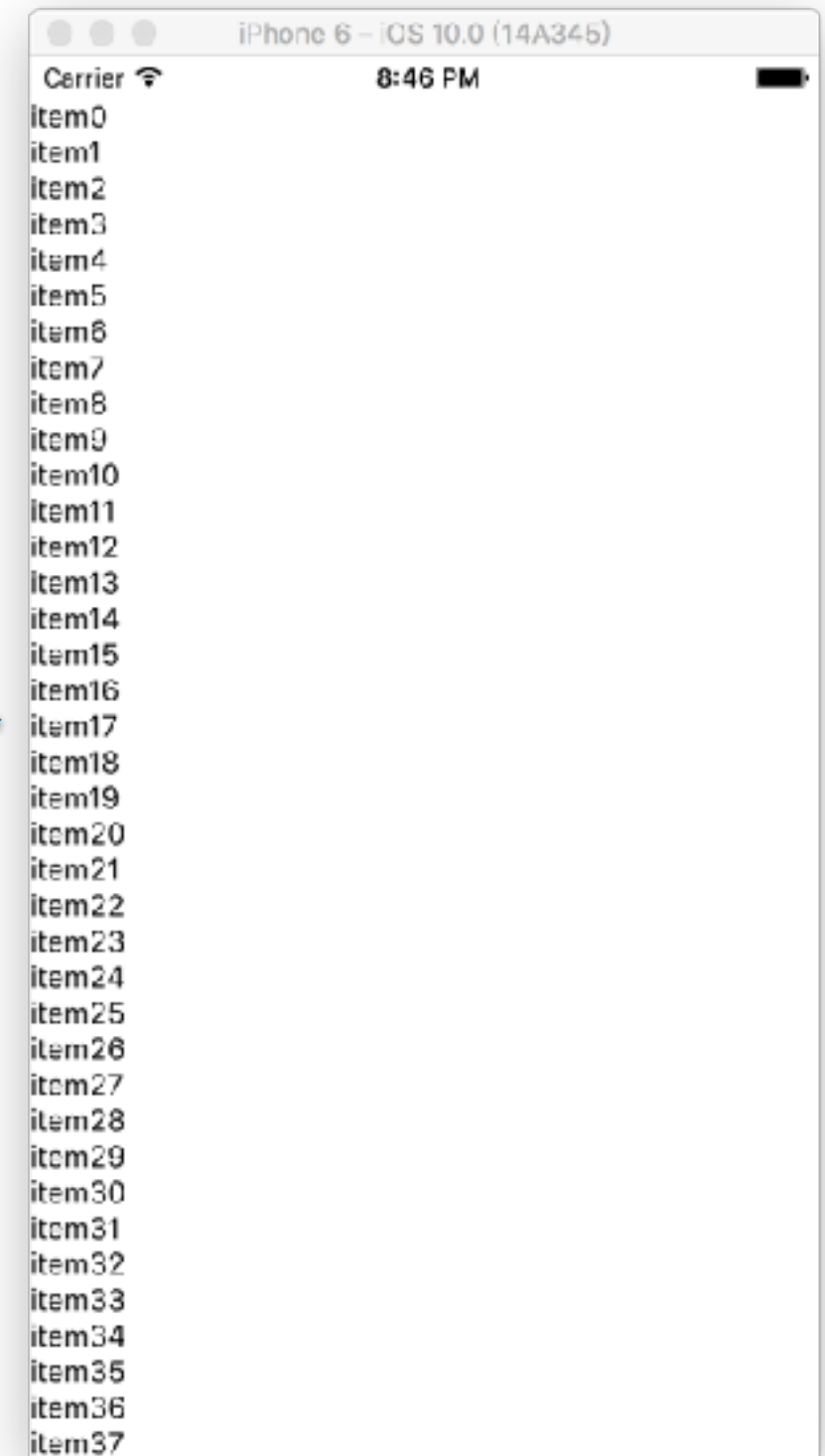
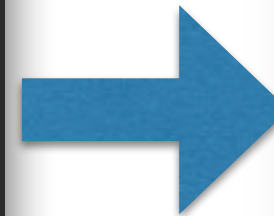
ListView

```
import React, {Component} from 'react';
import {
  AppRegistry,
  StyleSheet,
  Text,
  ListView
} from 'react-native';

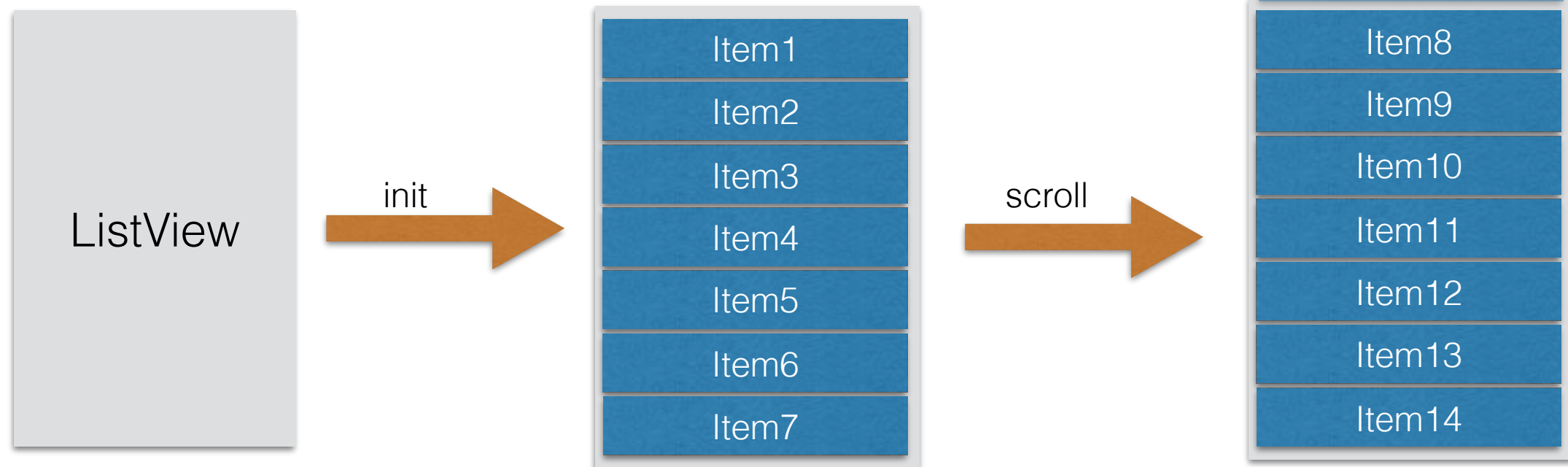
class MyComponent extends Component {
  constructor() {
    super();
    const ds = new ListView.DataSource({rowHasChanged: (r1, r2) => r1 !== r2});
    const arr = [];
    for(let i = 0; i < 1000; i++){
      arr.push("item"+i)
    }
    this.state = {
      dataSource: ds.cloneWithRows(arr),
    };
  }

  render() {
    return (
      <ListView
        initialListSize={40}
        pageSize={40}
        dataSource={this.state.dataSource}
        renderRow={(rowData) => <Text>{rowData}</Text>}
      />
    );
  }
}

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



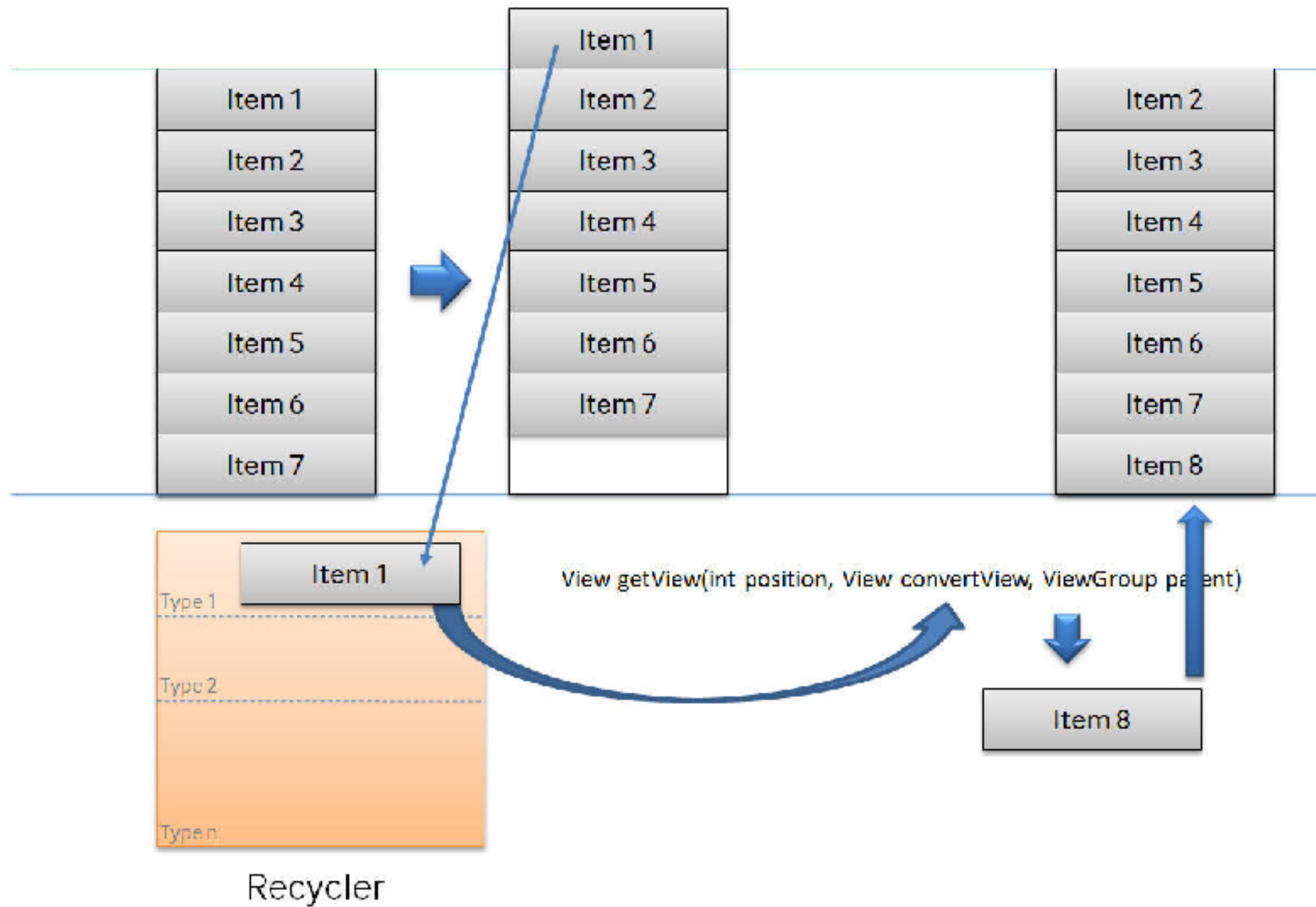
ListView加载原理



ListView加载采用顺序懒加载方式

- 合理设置初始化个数可以降低白屏时间
- 使用renderfooter渲染loading,当快速滑动渲染跟不上可以让用户先看到loading, 提升用户体验

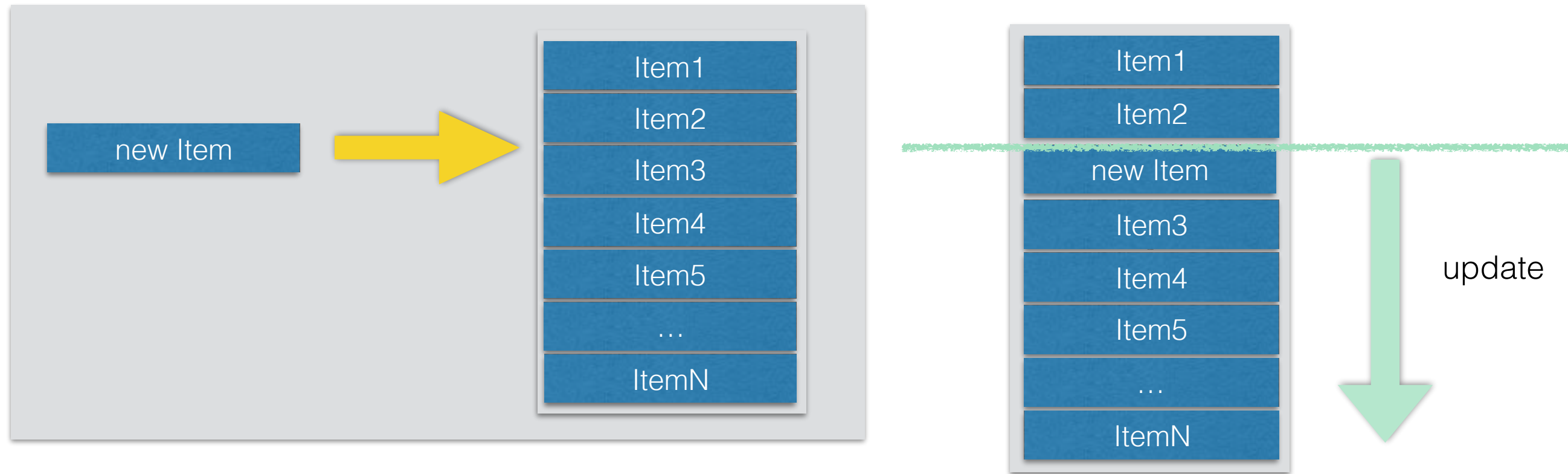
RecyclerView



ListView vs RecyclerView

- View 没有复用
- 插入删除性能差
- 无法做锚点功能
- 存在内存泄露问题

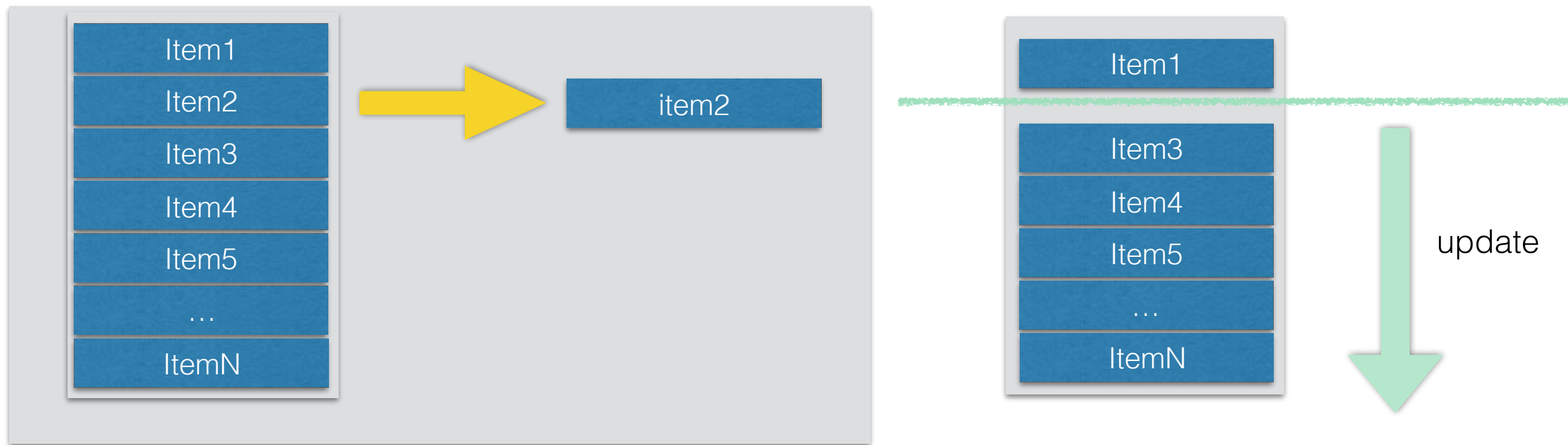
Listview插入



慎用listview的非底部插入功能，它会从插入点到列尾全部更新view，导致页面卡顿

如果使用顶部刷新，建议刷新完成后让listview key++，加快渲染速度

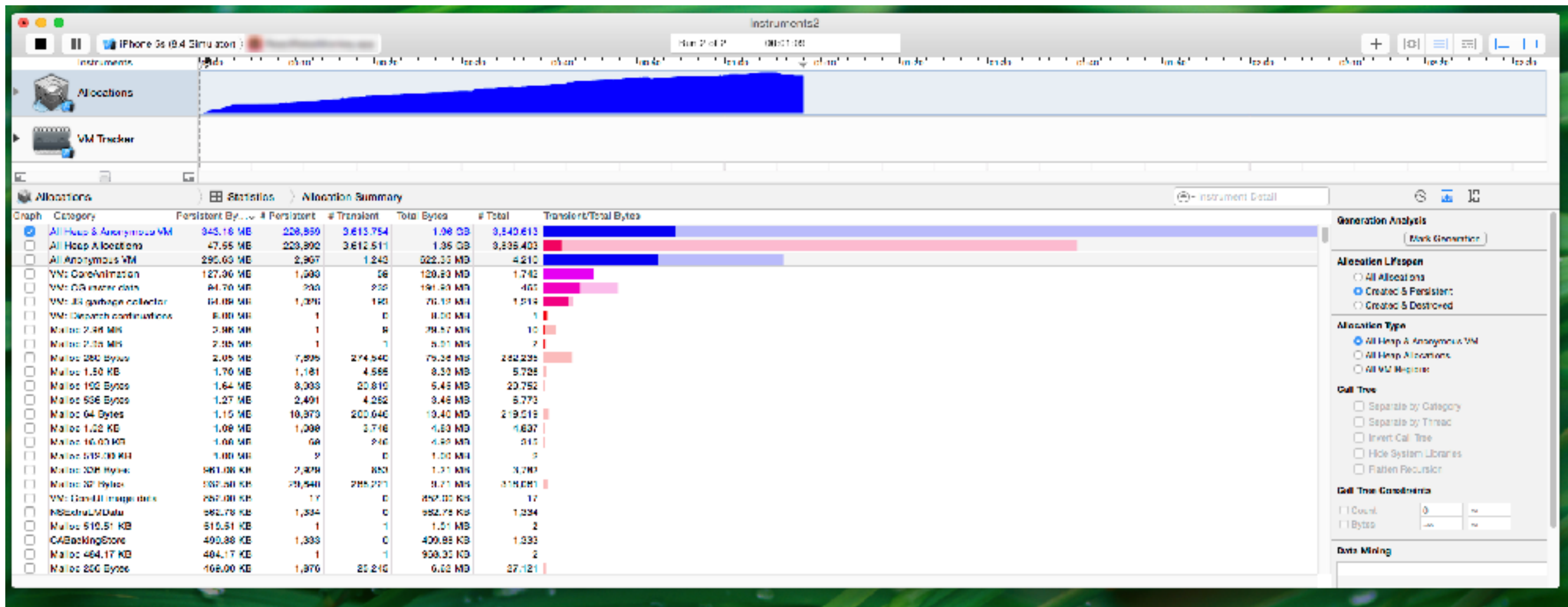
Listview删除



慎用listview删除功能，它会从插入点到列尾全部更新view，导致页面卡顿

建议删除item的时候，直接将item高度设置为0，而不要重新渲染 listview

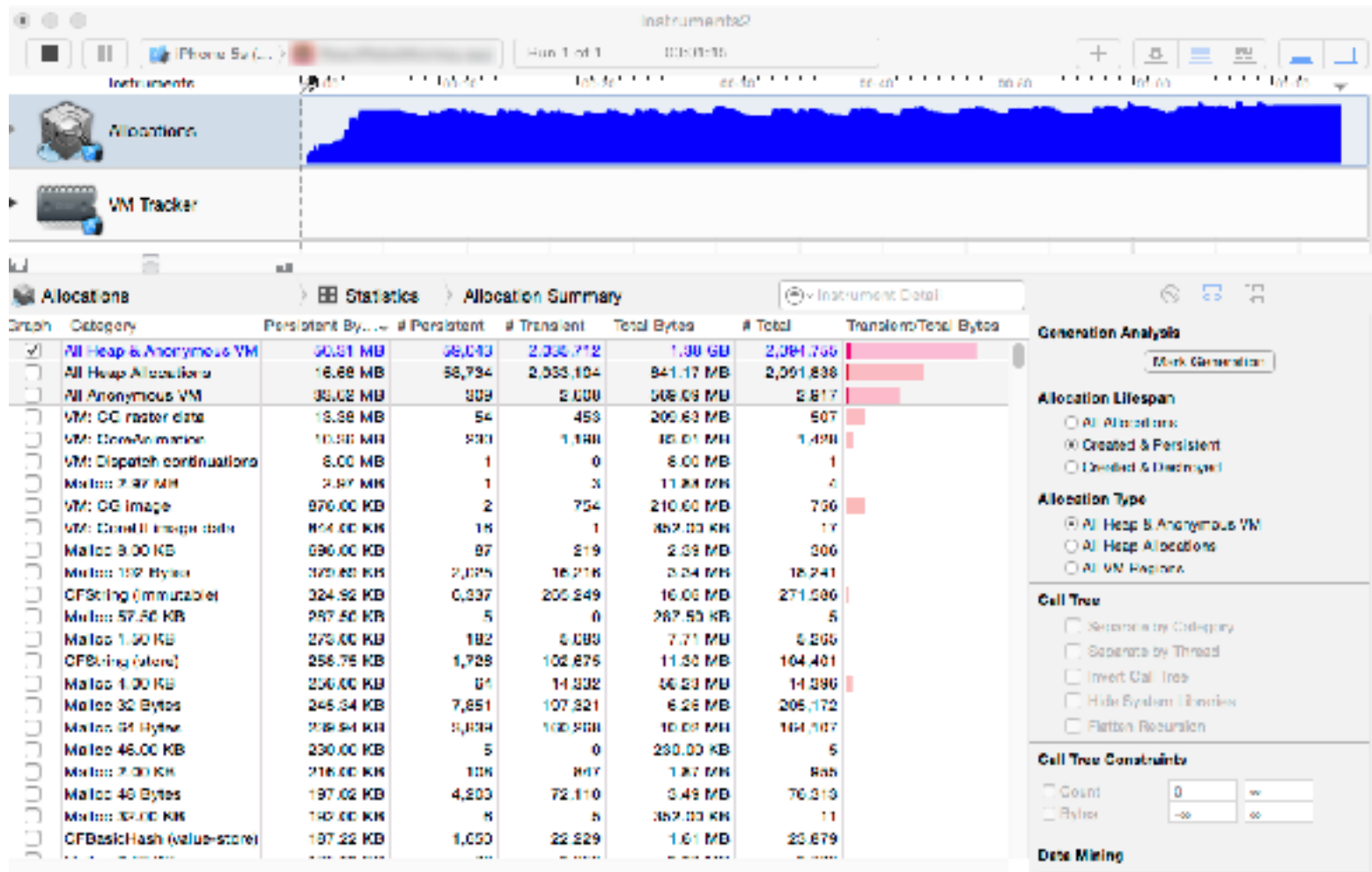
Listview内存分析



随着listview 滑动，item不断加载，内存呈线性增长趋势，当内存警告时，系统会主动回收，内存降低，有OOM的风险

SGListView优化内存

<https://github.com/sghiassy/react-native-sglistview>



- 实现方案
将不在列表可视区的item置换成空view
- 优势
能有效降低内存占用，不会导致OOM
- 劣势
滑动过程过快重新渲染会导致白屏

React Native动画

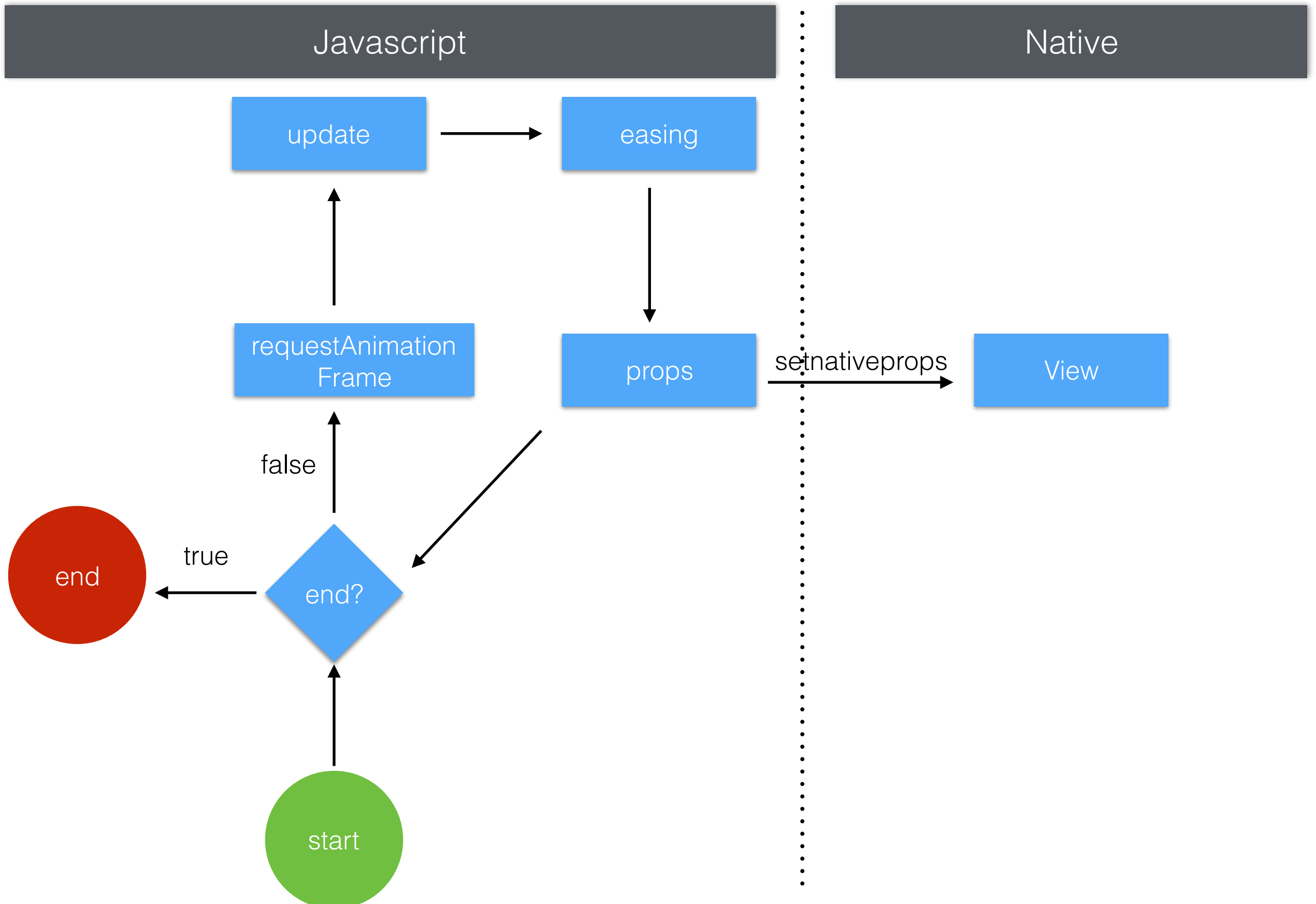
```
class RandomView extends React.Component {
  state: any;

  constructor(props) {
    super(props);
    this.state = {
      firstValue: new Animated.Value(0),
      secondValue: new Animated.Value(0),
    };
  }

  componentDidMount() {
    Animated.timing(this.state.firstValue, {toValue: 1, duration: 2000}).start();
    Animated.spring(this.state.secondValue, { toValue: 100 , tension: -10 }).start();
  }

  render() {
    return (
      <Animated.View // Special animatable View
        style={{
          opacity: this.state.firstValue, // Binds
          translateX: this.state.secondValue,
        }}>
        <Text>Hello</Text>
      </Animated.View>
    );
  }
}
```

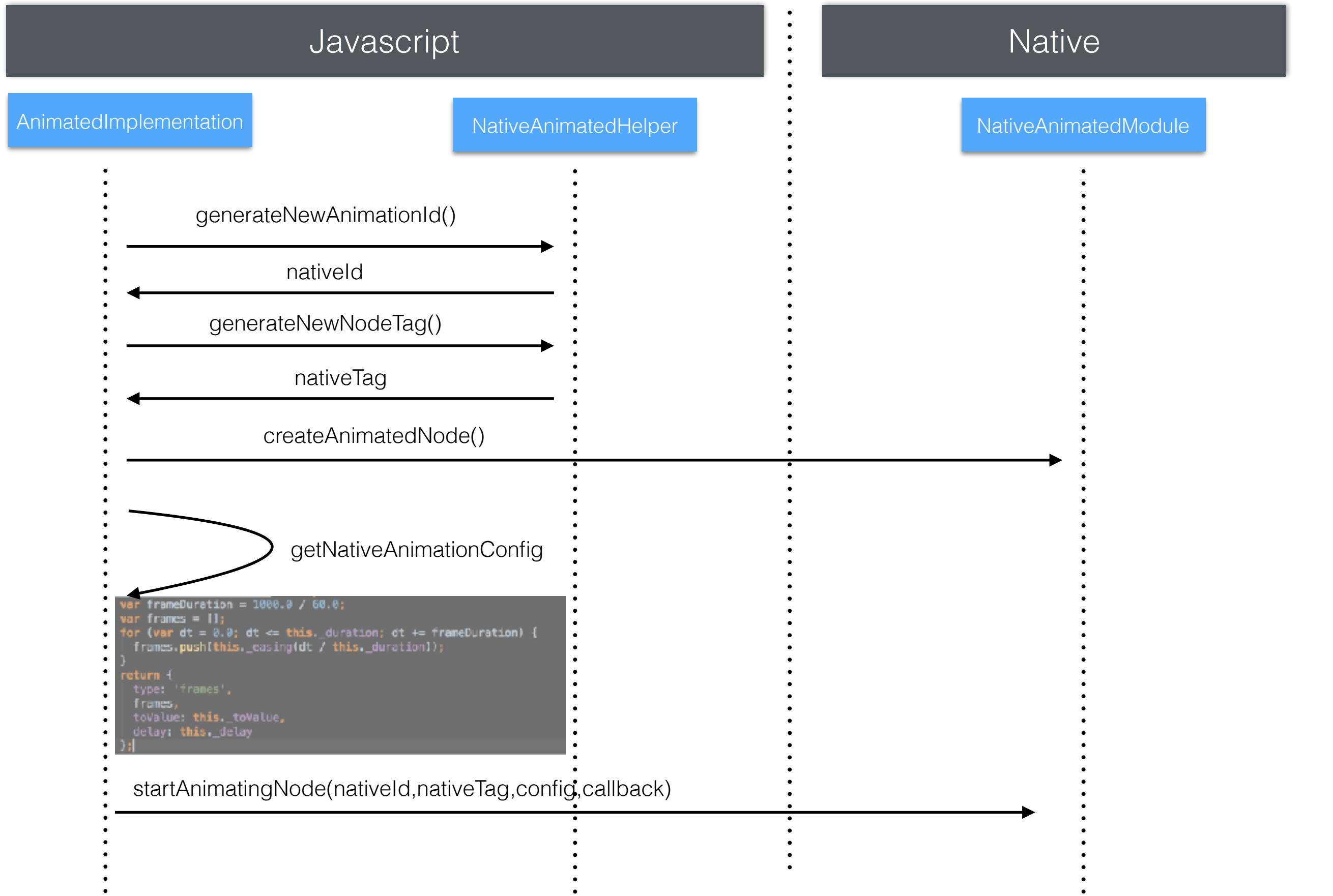
React Native动画-AnimatedView



React Native动画-AnimatedView

- js利用requestAnimationFrame进行60fps循环
- 采用setNativeProps直接更新状态
- js是单线程，每帧状态更新需要跟native通讯，容易掉帧

React Native动画-useNativeDriver



React Native动画-useNativeDriver

- useNativeDriver需要手动打开
- 去掉js端的timer，改用native来做timer
- js提前算好所有frames，一次性传给native
- duration越长，frames数组越大
- 开启后有一定的性能提升

React Native动画优化-预定义动画

```
/*component*/
UIManager.dispatchViewManagerCommand(
  findNodeHandle(this.refs.AnimationView),
  UIManager.AnimationView.Commands.start, [{
    animId: this._animId,
    rotate: this._rotate,
    translate: this._translate,
    scale: this._scale,
    opacity: this._opacity,
    duration: this._duration || 200,
    easing: ""+JSON.stringify(this._interpolator),
    delay: this._delay || 0
  }]
)
/*调用*/
<AnimationView ref="view"/>

this.refs.view.clearAnimation().setTranslate({
  from: {x: 0, y: 1000},
  to: {x: 0, y: 0}
}).setOpacity({
  to: 1
}).setScale({
  from: {x: 0, y: 0},
  to: {x: 1, y: 1}
}).onStart(() => {
  this.setState({
    opacity: 1
  })
}).setEasing({
  type: "bounce"
}).setDuration(10000).start();
```

js

```
public void startAnimation() {
  //初始化from
  setTranslationX(fromX);
  setTranslationY(fromY);
  //.....
  //执行动画
  animate().translationX(toX).translationY(toY).alpha(alpha).rotation(rotate)
    .scaleX(scaleX).scaleY(scaleY).setDuration(duration)
    .setInterpolator(new BounceInterpolator())
    .setStartDelay(delay).setListener(new Animator.AnimatorListener() {
      @Override
      public void onAnimationStart(Animator animation) {
        dispatchEvent(animId, "start");
      }

      @Override
      public void onAnimationEnd(Animator animation) {
        dispatchEvent(animId, "end");
      }

      @Override
      public void onAnimationCancel(Animator animation) {
      }

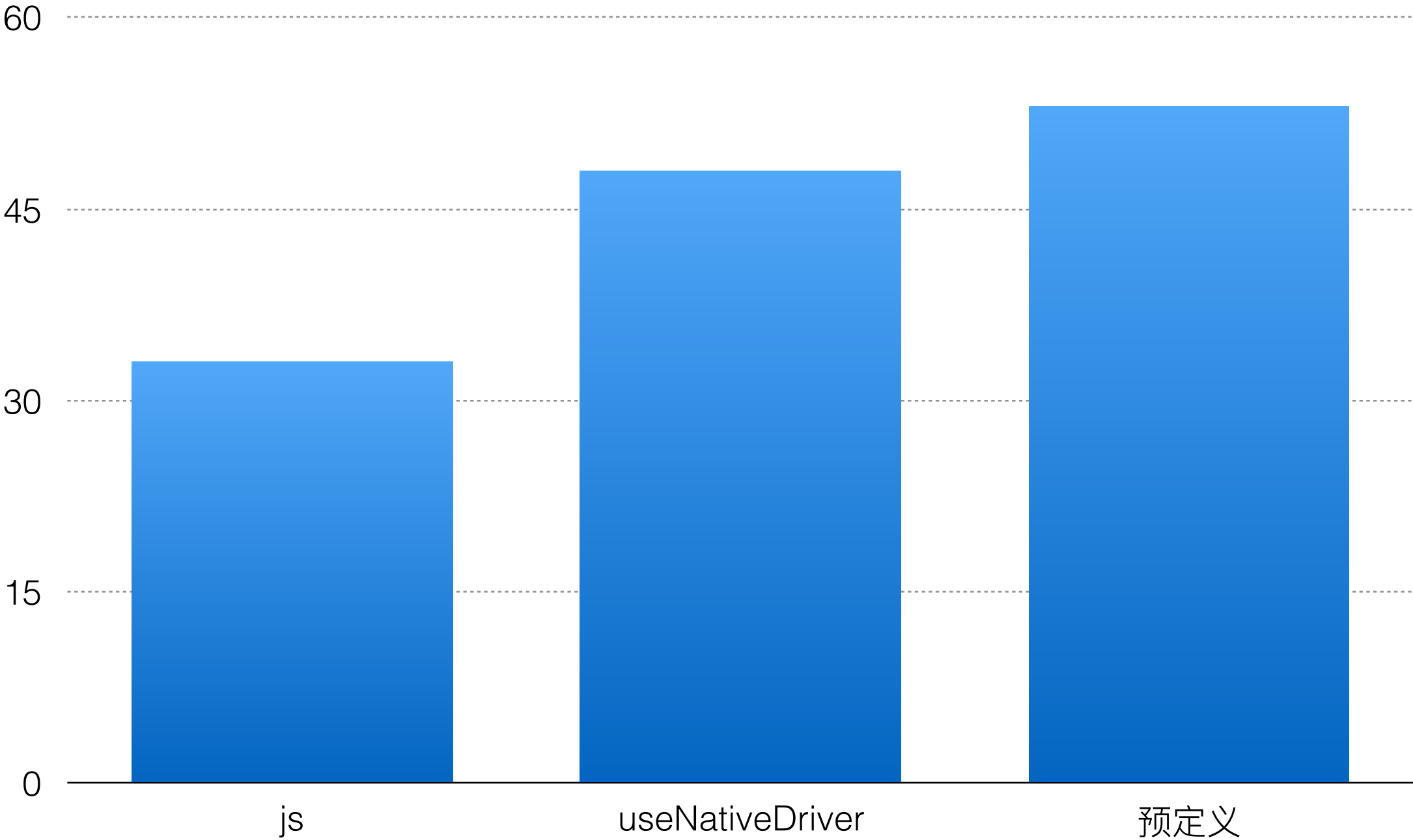
      @Override
      public void onAnimationRepeat(Animator animation) {
      }
    })
  }.start();
}
```

native

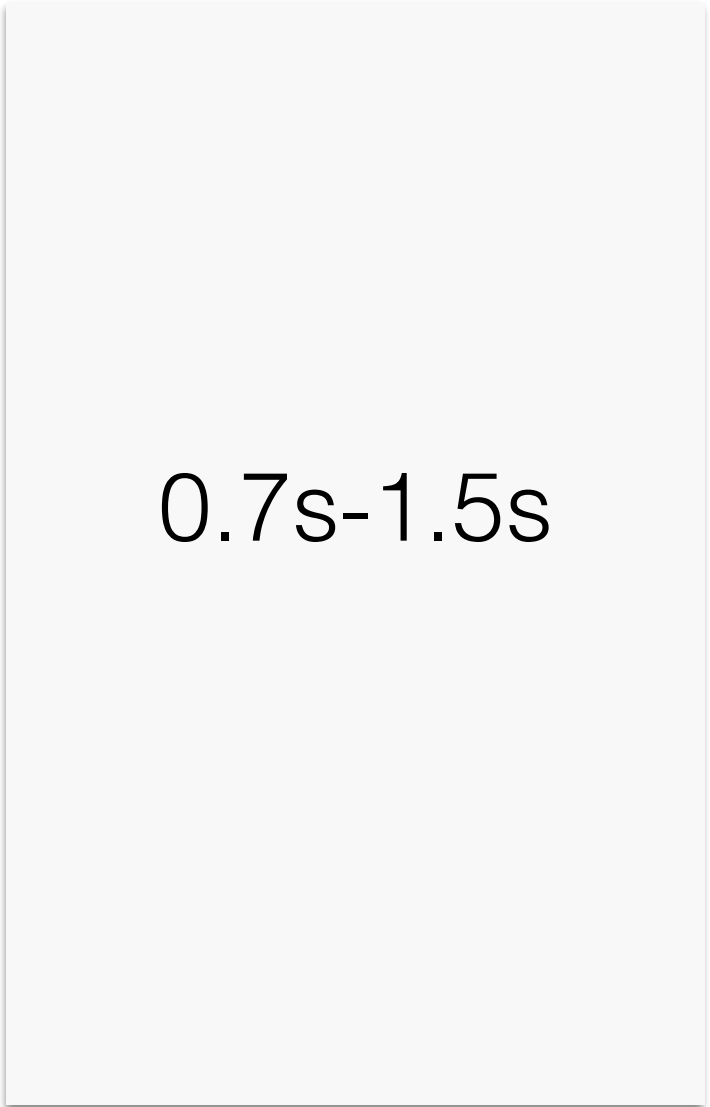
React Native动画优化-预定义动画

- Native预定义好常见的动画类型和easing算法
- js直接将配置发送到native执行
- 不在预定义动画类型外建议使用useNativeDriver

React Native动画帧率

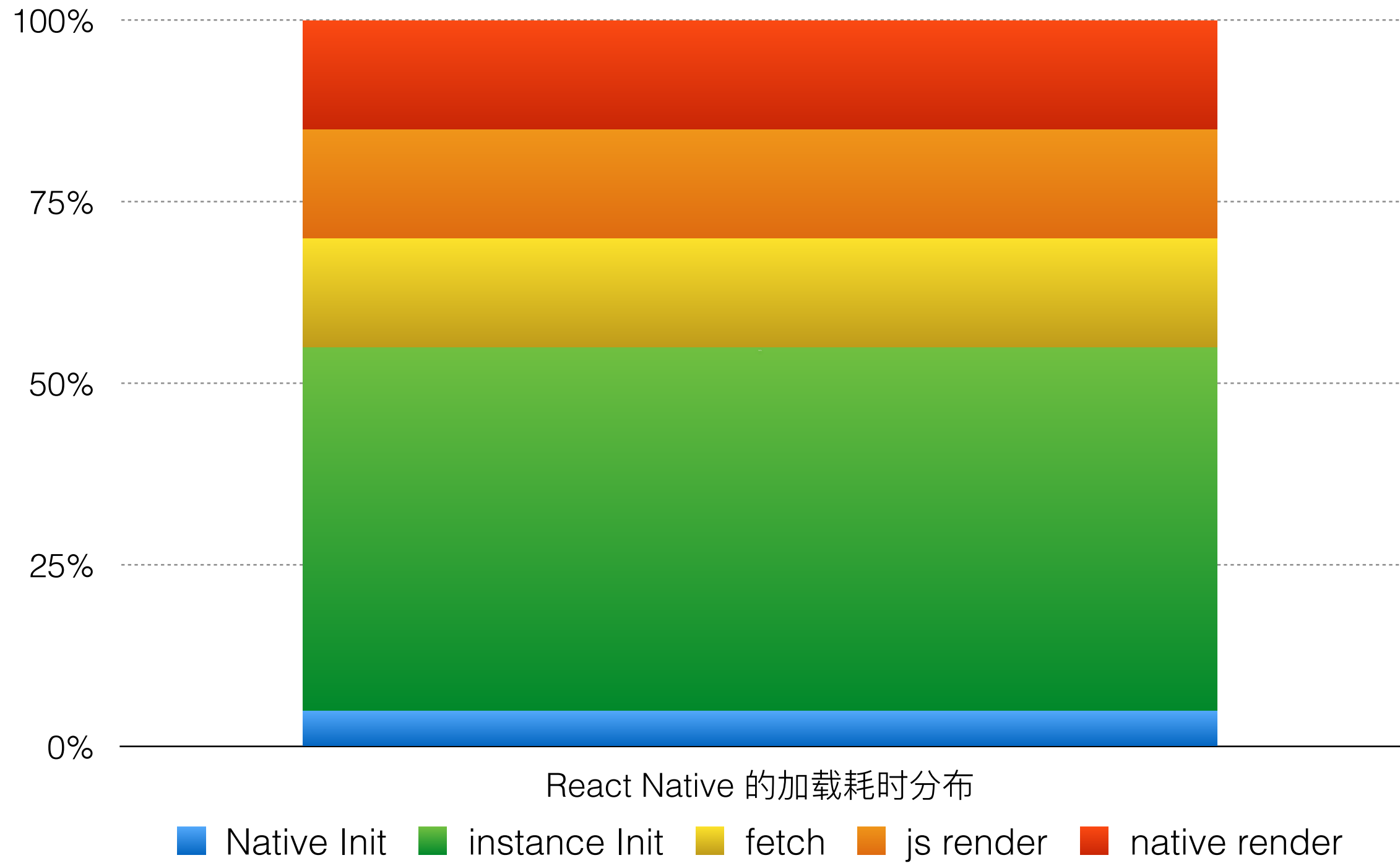


Android白屏



0.7s-1.5s

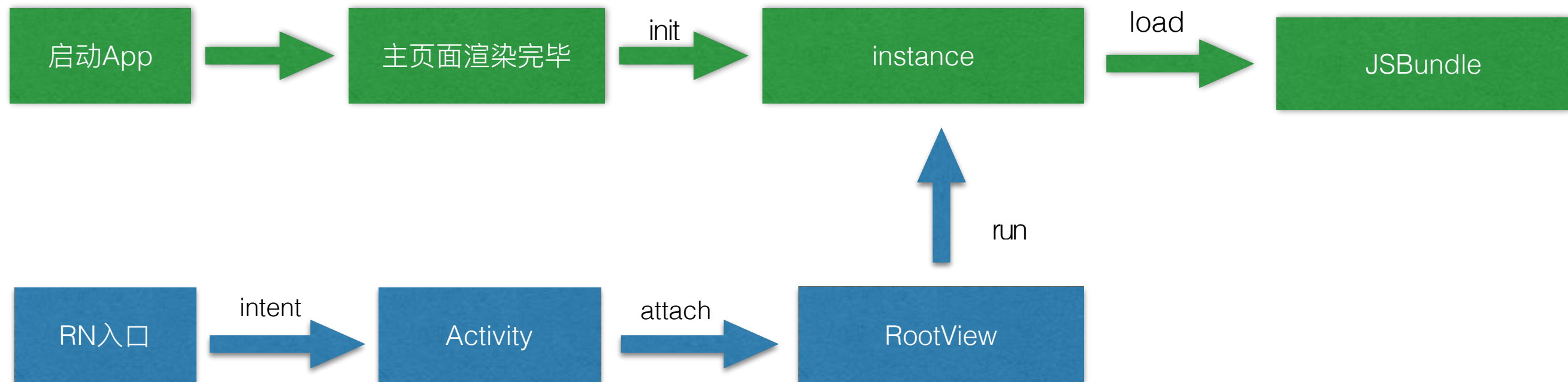
Android加载耗时分布



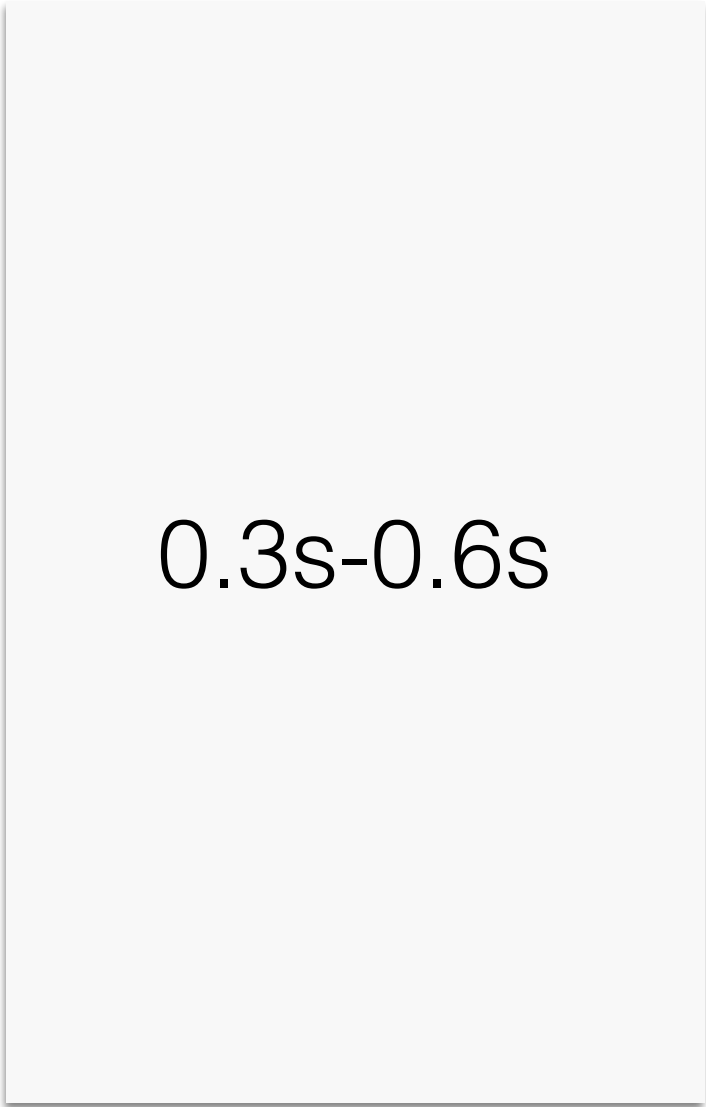
instance初始化耗时最严重

Android白屏优化

优化采用内存换时间，提前初始化JSCore

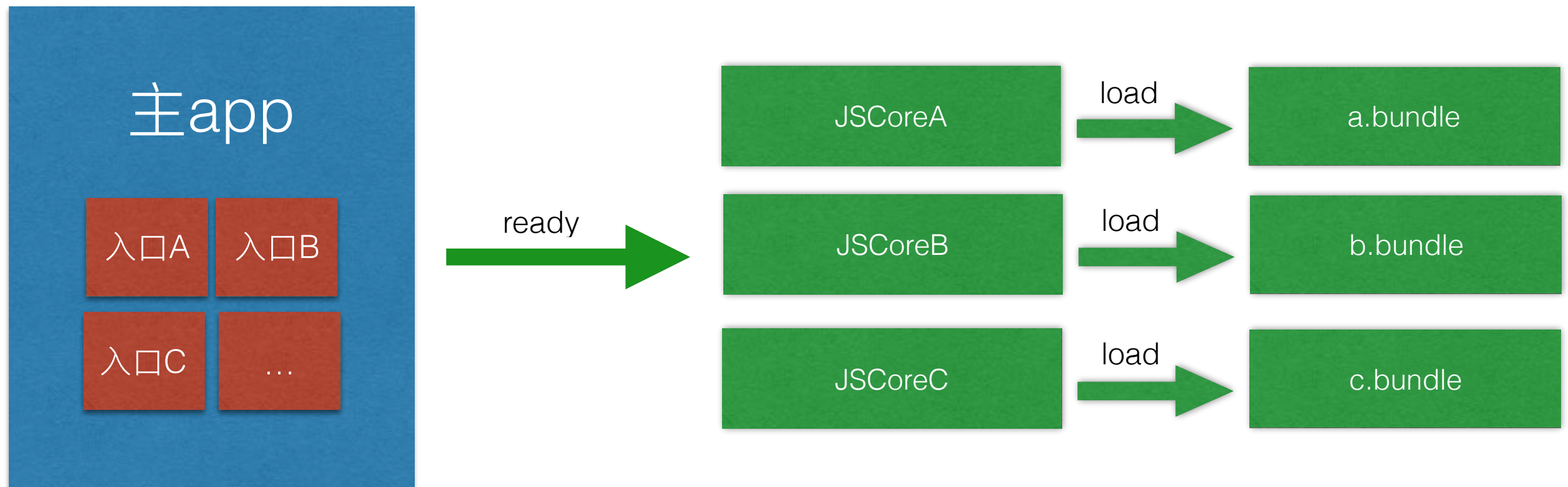


Android白屏优化后



0.3s-0.6s

预加载的问题

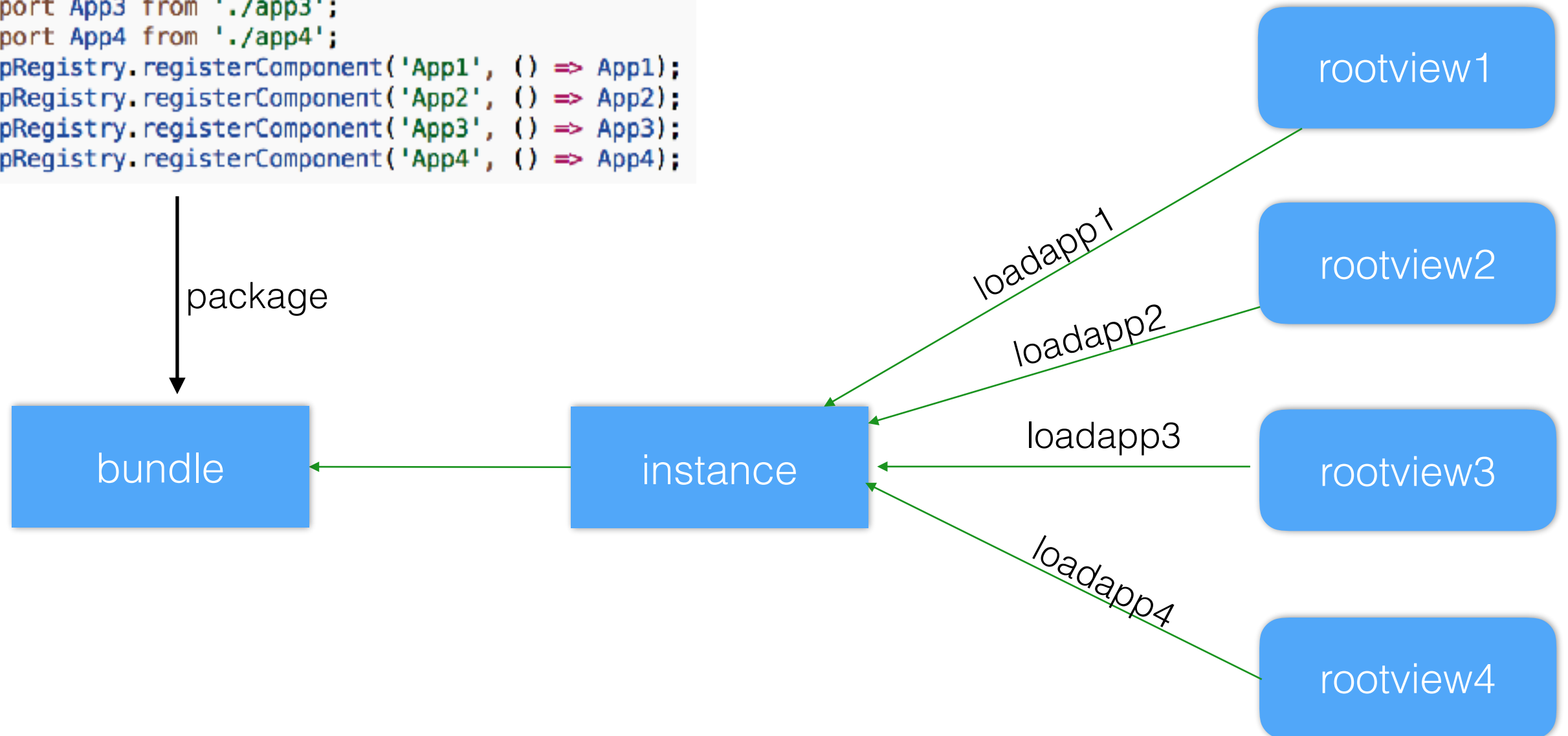


入口越多预加载的个数越多，内存开销越大

multi-component

```
import {
  AppRegistry
} from 'react-native';

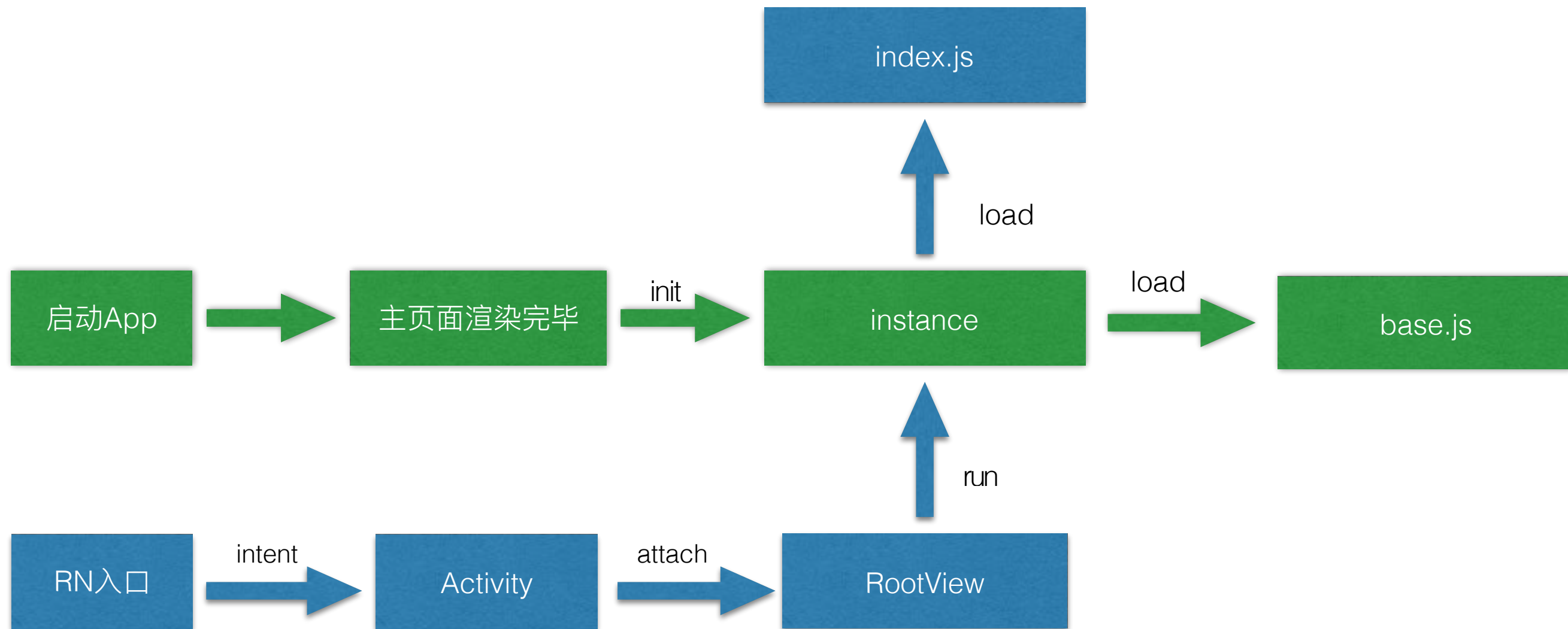
import App1 from './app1';
import App2 from './app2';
import App3 from './app3';
import App4 from './app4';
AppRegistry.registerComponent('App1', () => App1);
AppRegistry.registerComponent('App2', () => App2);
AppRegistry.registerComponent('App3', () => App3);
AppRegistry.registerComponent('App4', () => App4);
```



multi-component

- instance只存在一份
- 将所有业务打在一个bundle中
- 不同的业务加载不同的component
- 提前初始化所有业务成本较高

按需加载



将bundle拆分成base+index，启动只缓存一份JSCore并加载base
加载业务的时候分别加载各自的业务代码

一个完整的bundle.js

```
/*系统模块, 全局定义, polyfill, babel*/
/*prelude.js*/!function(_){_.__DEV__=!1,_.__BUNDLE_START_TIME__=Date.now()}("undefined"!==typeof global?global:"undefined"!==typeof self?se
/*require.js*/!function(r){"use strict";function t(r,t){r in u||(u[r]={factory:t,hasError:!1,isInitialized:!1,exports:void 0})}function i
/*polyfills.js*/!function(n){Object.assign=function(n,e){for(var f=1,f<arguments.length;f++){var l=arguments[f];if(null!=l)for(var i in l
/*console.js*/!function(n){function e(n){function e(e){return function(){var t;t=1===arguments.length&&"string"===typeof arguments[0]?argu
/*error-guard.js*/!function(r){function n(){var n=function(r){throw r};r.ErrorUtils.setGlobalHandler(n)}var l={_inGuard:0,_globalHandler:
/*Number.es6.js*/!function(e){void 0===Number.EPSILON&&Object.defineProperty(Number,"EPSILON",{value:Math.pow(2,-52)}),void 0===Number.MA
/*String.prototype.es6.js*/!function(t){String.prototype.startsWith||(String.prototype.startsWith=function(t){"use strict";if(null==this)
/*Array.prototype.es6.js*/!function(e){function r(e,r){if(null==this)throw new TypeError("Array.prototype.findIndex called on null or und
/*Array.es6.js*/!function(n){Array.from||(Array.from=function(n){if(null==n)throw new TypeError("Object is null or undefined");var e,o,t=
/*Object.es7.js*/!function(e){!function(){var e=Object.prototype.hasOwnProperty;"function"!==typeof Object.entries&&(Object.entries=functi
/*babelHelpers.js*/!function(e){var r=e.babelHelpers={};r.createRawReactElement=function(){var e="function"===typeof Symbol&&("function"==
/*入口模块*/
__d(0/*index.js*/,function(e,t,r,n){var i=t(12),u=(babelHelpers.interopRequireDefault(i),t(41)),l=t(377),o=babelHelpers.interopRequireDef
/*react-native基础模块*/
__d(12 /*./node_modules/react/react.js*/,function(t,s,c,e){"use strict";c.exports=s(13)});
__d(13 /*./node_modules/react/lib/React.js*/,function(e,t,r,n){"use strict";var a=t(14),o=t(15),c=t(27),l=t(30),s=t(31),i=t(36),m=t(19),p
__d(14 /*./node_modules/object-assign/index.js*/,function(r,e,t,n){"use strict";function o(r){if(null===r||void 0===r)throw new TypeError
__d(15 /*./node_modules/react/lib/ReactChildren.js*/,function(t,n,u,e){"use strict";function r(t){return(""+t).replace(k,"$&/"})}function
__d(16 /*./node_modules/react/lib/PooledClass.js*/,function(n,o,e,t){"use strict";var r=o(17),i=o(18),function(n){var o=this;if(o instan
...
__d(177 /*./node_modules/react-native/Libraries/JavaScriptAppEngine/Initialization/InitializeJavaScriptAppEngine.js*/,function(e,r,n,t){f
...
__d(375,function(r,t,e,n){"use strict";var i=t(17),a=t(15),o=t(19),u=t(22),f=(t(18),t(21)),{create:function(r){if("object"!==typeof r||!r||
__d(376,function(r,o,n,i){"use strict";function t(r){return Array.isArray(r)?r.concat():r&&"object"===typeof r?e(new r.constructor,r):r}fu
/*业务模块*/
__d(377,function(e,t,r,l){Object.defineProperty(l,"__esModule",{value:!0});var a=t(12),n=babelHelpers.interopRequireDefault(a),u=(t(41),t
__d(378,function(e,t,d,n){"use strict";function o(e){return e&&e.__esModule?e:{default:e}}n.__esModule=!0,n.connect=n.Provider=void 0;var
__d(379,function(e,t,r,o){"use strict";function n(e){return e&&e.__esModule?e:{default:e}}function i(e,t){if(!(e instanceof t))throw new
__d(380,function(e,s,u,i){"use strict";i.__esModule=!0;var p=s(12);i.default=p.PropTypes.shape({subscribe:p.PropTypes.func.isRequired,dis
__d(381,function(o,e,n,r){"use strict";function t(o){"undefined"!==typeof console&&"function"===typeof console.error&&console.error(o);try{
__d(382,function(t,e,r,o){"use strict";function s(t){return t&&t.__esModule?t:{default:t}}function n(t,e){if(!(t instanceof e))throw new
__d(383,function(e,t,r,n){"use strict";function u(e,t){if(e===t)return!0;var r=Object.keys(e),n=Object.keys(t);if(r.length!==n.length)ret
__d(384,function(n,t,r,u){"use strict";function e(n){return function(t){return(0,i.bindActionCreators)(n,t)}}u.__esModule=!0,u.default=e;
__d(385,function(e,t,d,o){"use strict";function r(e){return e&&e.__esModule?e:{default:e}}o.__esModule=!0,o.compose=o.applyMiddleware=o.b
__d(386,function(e,t,n,r){"use strict";function o(e){return e&&e.__esModule?e:{default:e}}function i(e,t,n){function r(){b===w&&(b=w.slic
__d(387,function(t,r,n,o){function c(t){if(!a(t)||e(t)===i)return!1;var r=u(t);if(null===r)return!0;var n=s.call(r,"constructor")&&r.const
__d(388,function(n,t,o,i){function e(n){return null===n?void 0===n?f:d:(n=Object(n),g&&g in n?r(n):u(n))}var c=t(389),r=t(392),u=t(393),d=
/*执行系统模块和运行系统模块*/
;require(177);/*InitializeJavaScriptAppEngine*/
;require(0);/*入口模块*/
```

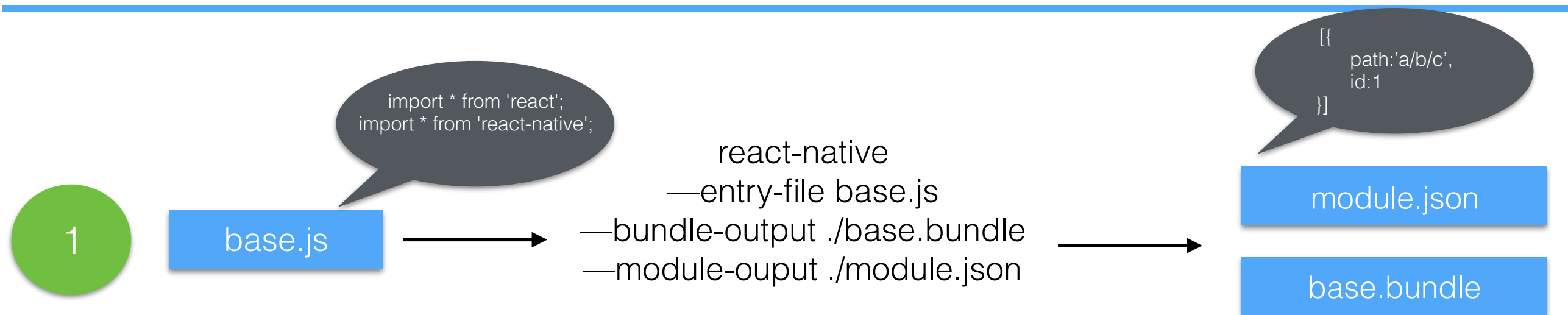
一个完整的bundle.js

- 每个文件是一个独立的module
- module加载遵循commonjs规范
- Define的标识符采用数字自增
- 每一行代表一个module文件

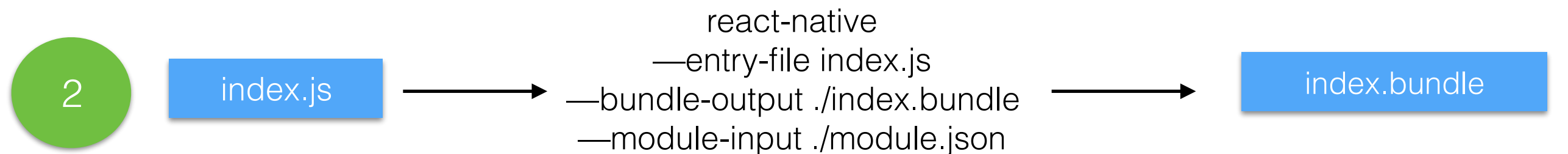
packager工作流程



拆包实现



1. onResolutionResponse 过滤掉index入口文件
 2. bundle.finalize 取消生成require mainModuleId
 3. finalizeBundle 输出完整的module.json
-



1. onResolutionResponse 过滤掉base module 和 system module
2. createModuleIdFactory 将module path返回id,业务path返回md5(path)

拆包结果

```
!function(_){_.__DEV__=!1,_.__BUNDLE_START_TIME__=Date.now()}("undefined"!==typeof global?global:"undefined"!==typeof self?self:tl  
!function(r){"use strict";function t(r,t){r in u||{u[r]={factory:t,hasError:!1,isInitialized:!1,exports:void 0}}}function i(r){  
!function(n){Object.assign=function(n,e){for(var f=1;f<arguments.length;f++){var l=arguments[f];if(null!=l)for(var i in l)n[i]=  
!function(n){function e(n){function e(e){return function(){var t;t=l===arguments.length&&"string"==typeof arguments[0]?argument:  
!function(r){function n(){}var n=function(r){throw r};r.ErrorUtils.setGlobalHandler(n)}var l={_inGuard:0,_globalHandler:null,set:  
!function(e){void 0===Number.EPSILON&&Object.defineProperty(Number,"EPSILON",{value:Math.pow(2,-52)}),void 0===Number.MAX_SAFE_  
!function(t){String.prototype.startsWith||(String.prototype.startsWith=function(t){"use strict";if(null==this)throw TypeError()  
!function(e){function r(e,r){if(null==this)throw new TypeError("Array.prototype.findIndex called on null or undefined");if("fun  
!function(n){Array.from||(Array.from=function(n){if(null==n)throw new TypeError("Object is null or undefined");var e,o,t=argumen  
!function(e){!function(){var e=Object.prototype.hasOwnProperty;"function"!==typeof Object.entries&&(Object.entries=function(n){i  
!function(e){var r=e.babelHelpers={};r.createRawReactElement=function(){var e="function"==typeof Symbol&&("function"==typeof Sy  
__d(12,function(t,s,c,e){"use strict";c.exports=s(13)});  
__d(13,function(e,t,r,n){"use strict";var a=t(14),o=t(15),c=t(27),l=t(30),s=t(31),i=t(36),m=t(19),p=t(37),E=t(39),u=t(40),y=(t(  
__d(14,function(r,e,t,n){"use strict";function o(r){if(null===r||void 0===r)throw new TypeError("Object.assign cannot be called  
__d(15,function(t,n,u,e){"use strict";function r(t){return(""+t).replace(k,"$&/")}function o(t,n){this.func=t,this.context=n,th  
...  
__d(373,function(e,r,s,i){"use strict";function u(e,r){this.value=e,this.requestChange=r}function n(e){var r={value:void 0===e?  
__d(374,function(t,e,r,n){"use strict";function a(t,e){var r={};return function(n){r[e]=n,t.setState(r)}}var c={createStateSette  
__d(375,function(r,t,e,n){"use strict";var i=t(17),a=t(15),o=t(19),u=t(22),f=(t(18),t(21)),{create:function(r){if("object"!==type  
__d(376,function(r,o,n,i){"use strict";function t(r){return Array.isArray(r)?r.concat():r&&"object"==typeof r?e(new r.construct  
;require(177);  
//# sourceMappingURL=base.sourcemap.json
```

```
base.android.js
```

```
__d("78cd4f3625d26d20a5295707741c4ed4",function(e,t,r,n){var i=t(12),a=(babelHelpers.interopRequireDefault(i),t(41)),b=t("98b78b78b1430c9fd11e494bc16802be9ca",function(e,t,r,l){Object.defineProperty(l,"__esModule",{value:!0});var a=t(12),b=babelHel__d("1ddda6b11e09d6159482e468df53aaea",function(d,e,c,a){"use strict";function t(d){return d&&d.__esModule?d:{default:d}}a.__es__d("9ccbf827b329350587fe67c8e6028d21",function(e,t,r,o){"use strict";function n(e){return e&&e.__esModule?e:{default:e}}functio__d("8d330e9ffcb767ce25d398ccab68f9cf",function(e,s,c,u){"use strict";u.__esModule=!0;var i=s(12);u.default=i.PropTypes.shape({s__d("d2f18cfd8054b076d46fb28160786da",function(d,f,a,e){var b=f("7948e156aa76dea9e854e16c7831674f"),c=b.Symbol;a.exports=c});__d("7948e156aa76dea9e854e16c7831674f",function(e,t,c,f){var a=t("cc5b533a59521368a1845d924b276d61"),s="object"==typeof self&&se__d("cc5b533a59521368a1845d924b276d61",function(c,t,b,e){var o="object"==typeof c&&c&&c.Object===Object&&c;b.exports=o});...__d("0dfecf61dcc82a5a8fd8524e91a0f9d6",function(e,a,f,c){function d(e){var a=[(0,r.applyMiddleware)(t.default)],f=r.compose.appl__d("97480f1dc1bc794c30a033f2068ccbe7",function(t,n,u,c){"use strict";function e(t){return function(n){var u=n.dispatch,c=n.getS__d("bbc49c5b3413283ce9856c8640052990",function(e,t,a,s){function n(){var e=arguments.length>0&&void 0!==arguments[0]?arguments[__d("6aceb87a0113b57af3067e40cb648014",function(n,e,t,a){Object.defineProperty(a,"__esModule",{value:!0});var l=function n(e){ba;require("78cd4f3625d26d20a5295707741c4ed4");  
//# sourceMappingURL=myattention.sourcemap.json
```

index.android.js

总结

- RN解决了webapp体验不流畅的问题
- RN解决了Native跨平台和更新困难的问题
- RN通过jscore引擎连接js和native
- Listview没有复用，有性能问题，一些技巧能提升渲染性能
- 使用动画的时候开启useNativeDriver能提升动画性能
- 提前初始化instancemanager有助于减少白屏时间
- bundle拆分按需加载能减少不必要的内存浪费

THANKS

QA