

OSCC源创会

年终盛典 2016

React-Native

▶ Pintec 吴慕贤



自我介绍

Pintec 移动端基础架构

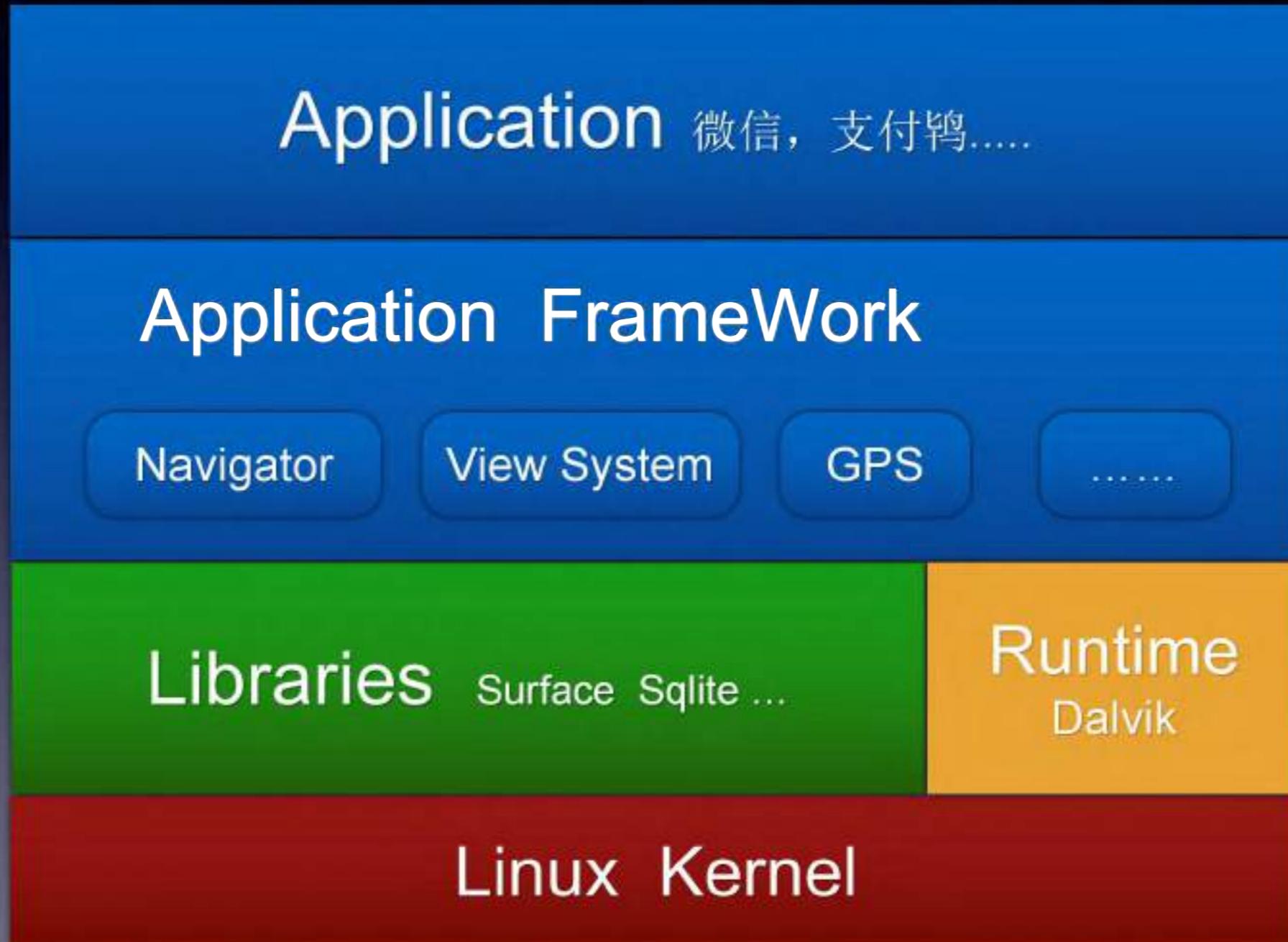
- 移动端的基础框架
- 跨平台开发框架
- 性能优化(网络,UI,计算)
- 移动端安全

为什么React-Native, Weex, 小程序会这么火

- 两(三)端代码公用，更低的开发、测试成本
- 更低的开发门槛
- 客户端动态发布，业务迭代更快
- 缩小安装包体积(iOS非常明显)
- 更方便彻底的业务解耦能力
- 相比h5，更好的用户体验

React-Native, Weex 底层原理

系统架构图



Application 微信, 支付码.....

Application Framework — RN

Application Framework

Navigator

View System

GPS

.....

Libraries Surface Sqlite ...

Runtime
Dalvik

Linux Kernel

Application 微信, 支付码.....

Application Framework — RN

Application Framework

Navigator

View System

GPS

.....

Runtime

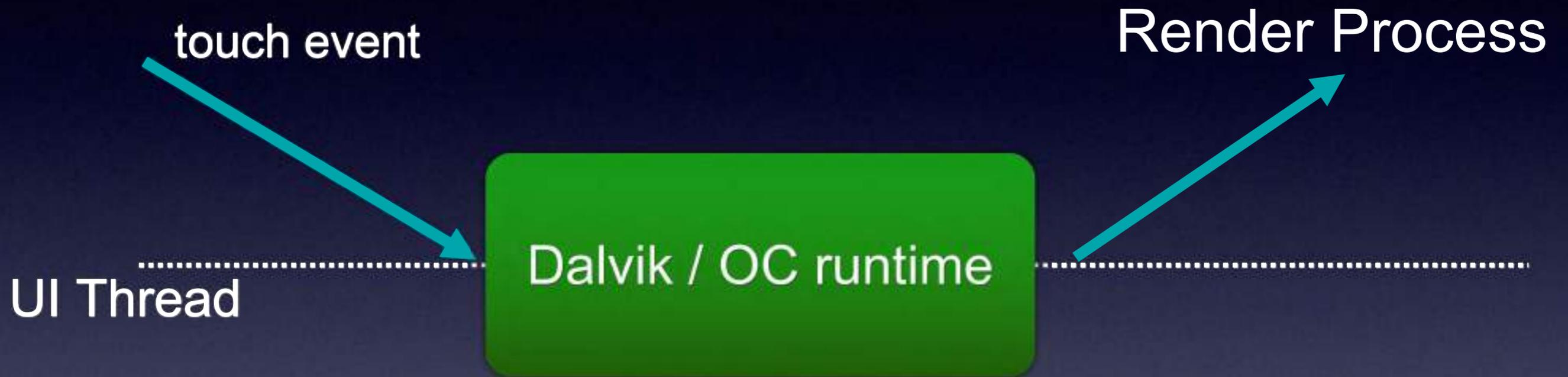
JS Engine

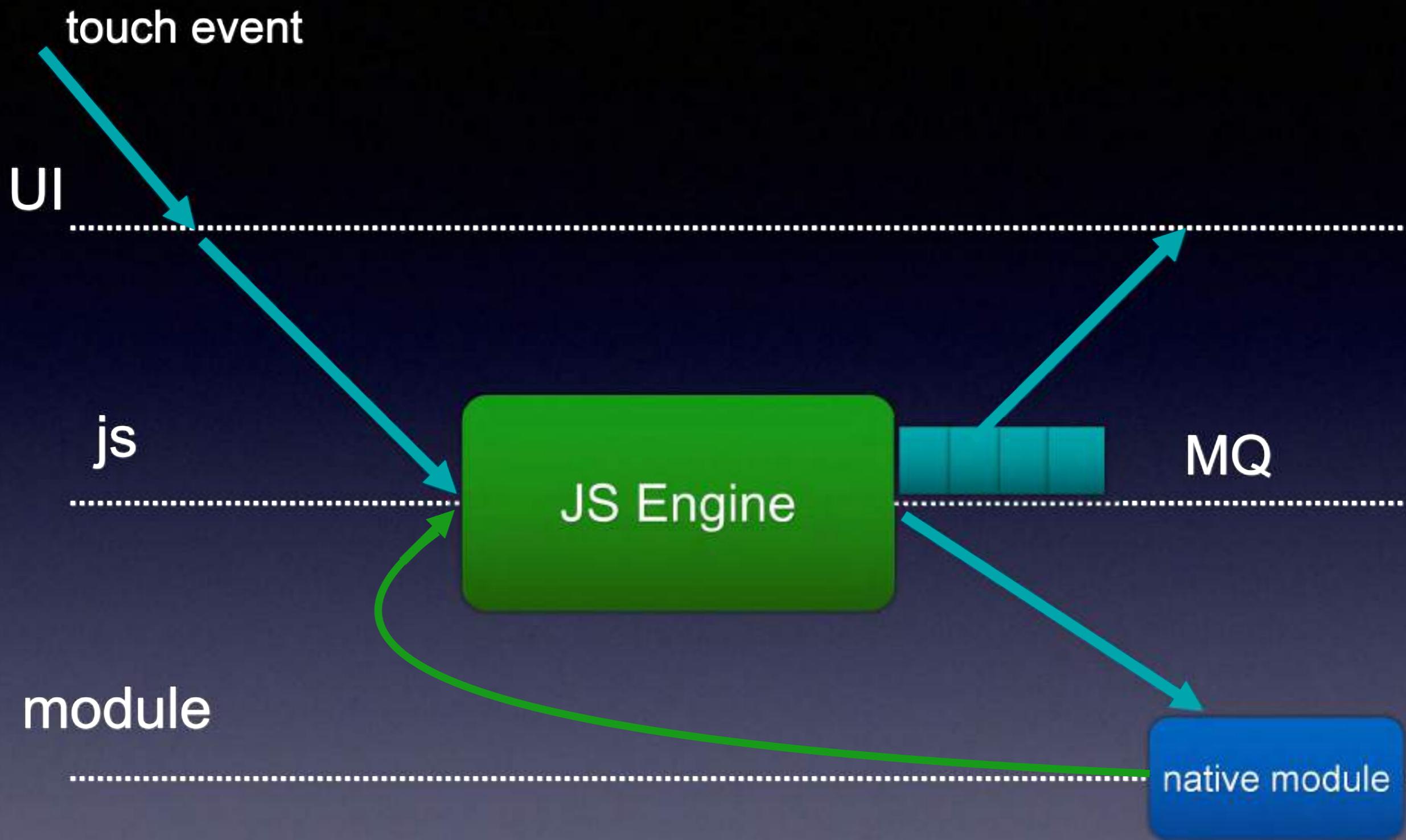
Libraries Surface Sqlite ...

Runtime
Dalvik

Linux Kernel

传统的桌面应用程序运行方式





多线程运行带来的问题

所有涉及到**native**和**js**交互的地方都是异步的

记录仪 “卖疯了” 这个价只敢卖1天

金指码 恭喜 直降300元



天猫



聚划算



天猫国际



外卖



天猫超市



充值中心



飞猪旅行



领金币



到家



分类



热议 诺基亚概念机亮相：侧滑键盘美哭了！
热议 手机这个功能或将取消，你不能不知道！

淘抢购

01:15:52



有好货

高颜值美物



爱逛街

时髦流行款



必买清单

帮您整理好



AutoLayout

React

Vue

acquaintanceship

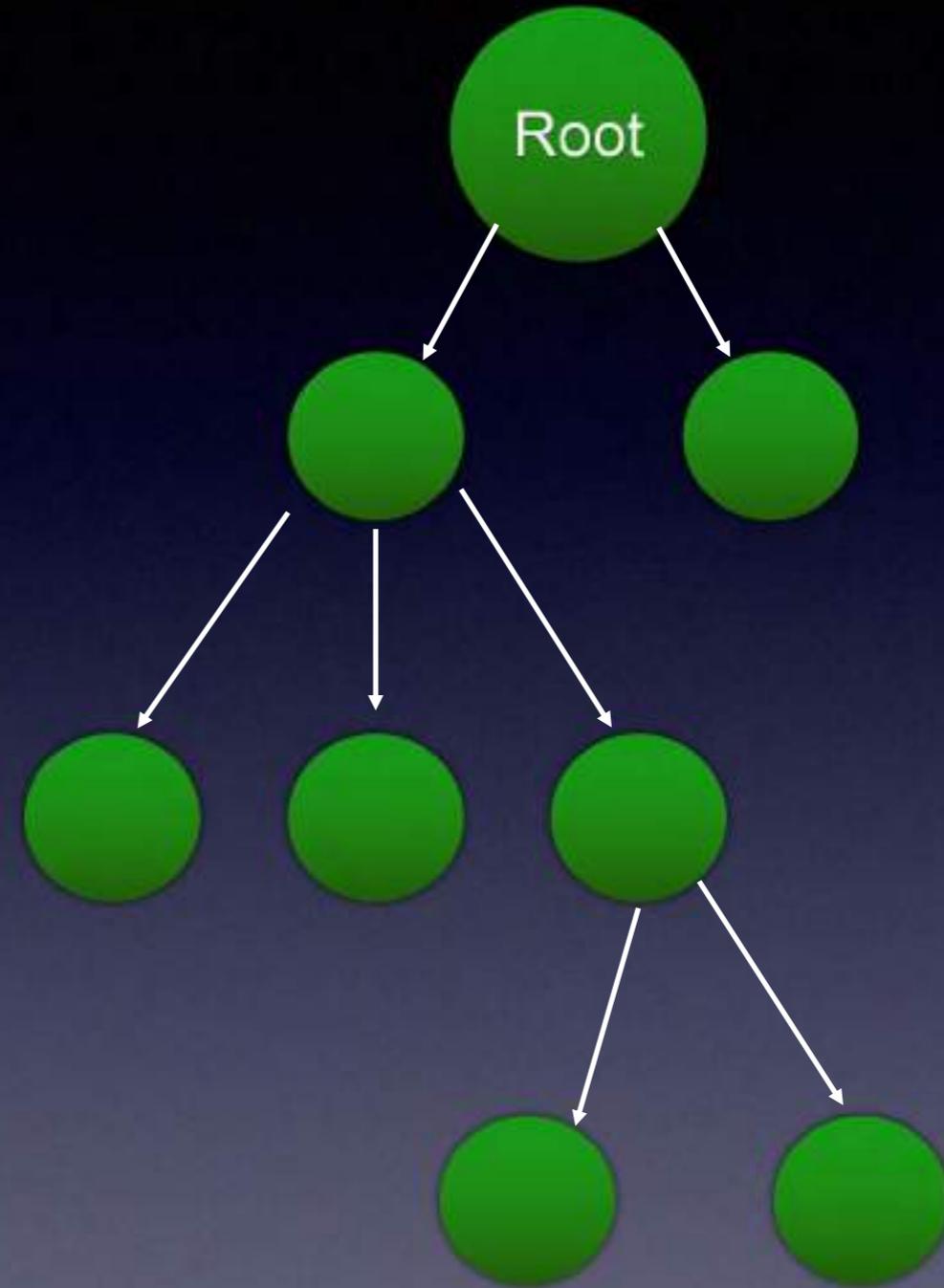
思维江化

支付鸨

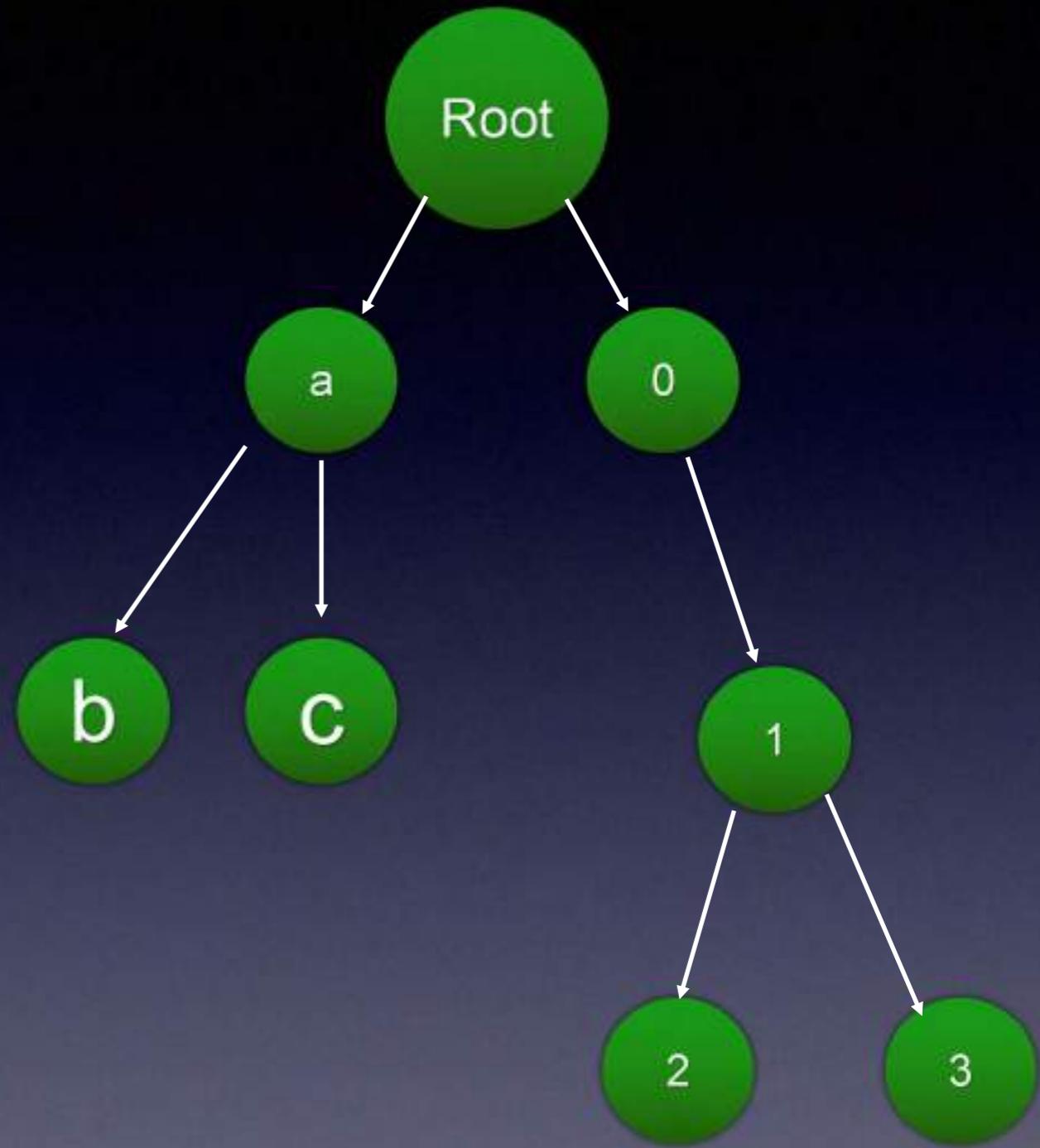
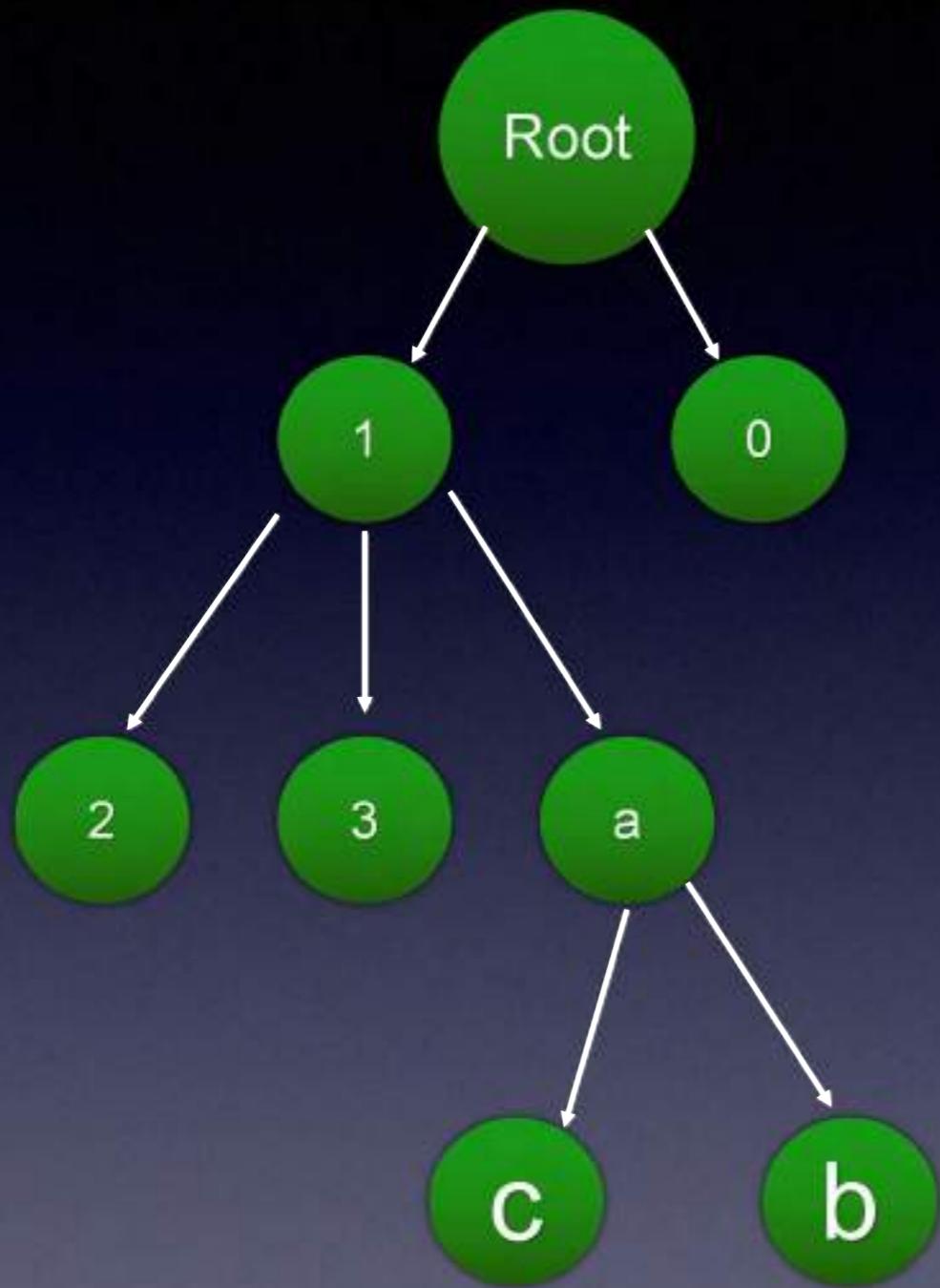
布局引擎消耗 < 1ms

View 映射

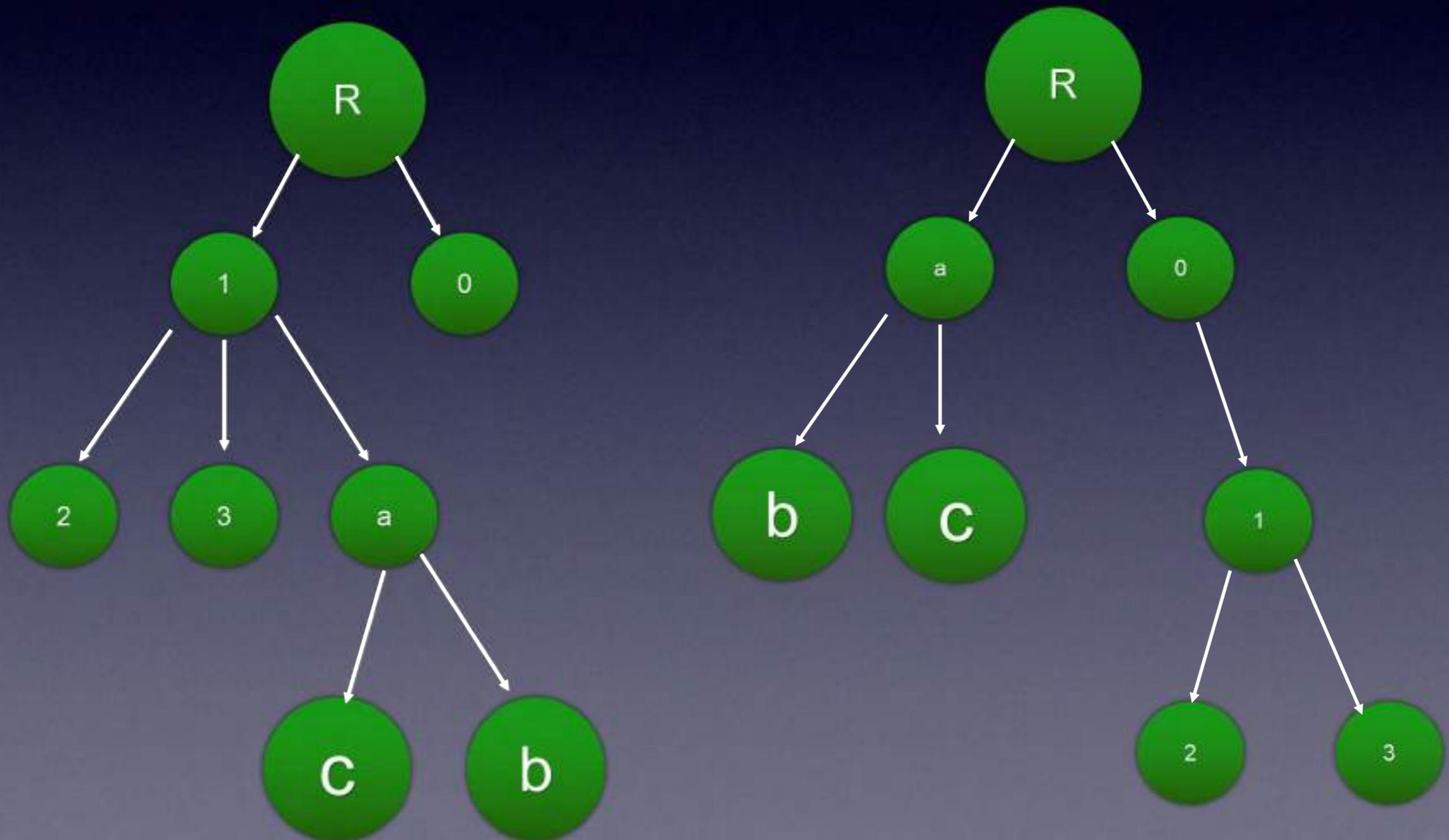
VC / Activity



Virtual Dom Diff



- **view**不会跨层级移动
- 不同**z-index**的**view**不可复用
- 同级**view** **Id**相同优先考虑复用



Virtual Dom Diff 好处与坏处

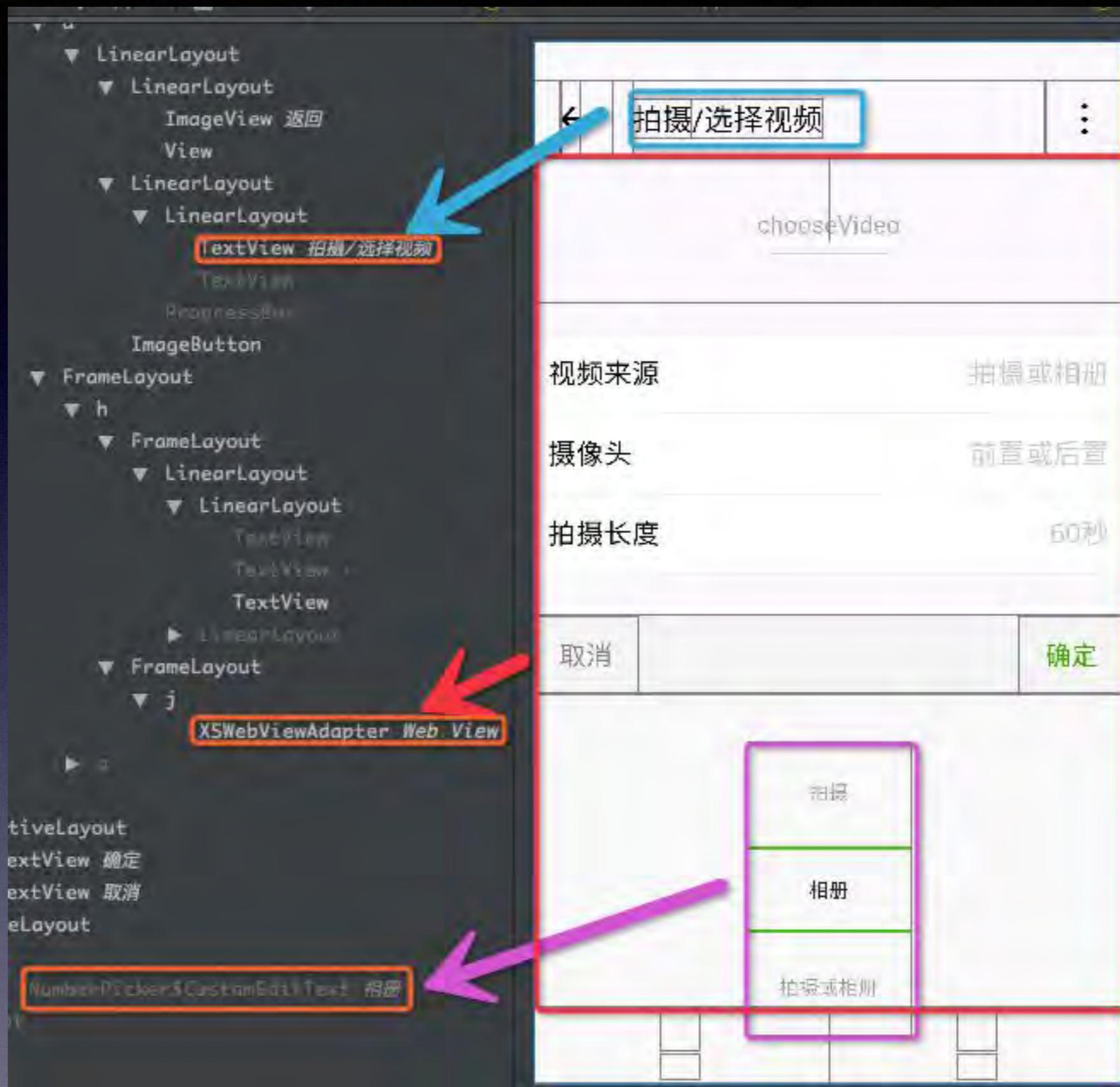
验证码 1

验证码 2

验证码 1

验证码 2

微信小程序



- ▼ LinearLayout
 - ImageView 返回
 - View
- ▼ LinearLayout
 - ▼ LinearLayout
 - TextView 创建画布
 - TextView
 - TextView
 - ImageButton
- RelativeLayout
- 1
- ▼ FrameLayout
 - ▼ LinearLayout
 - ▼ LinearLayout
 - TextView
 - TextView
 - TextView
 - ▶ LinearLayout
 - ▼ FrameLayout
 - ▼ j
 - ▼ X5WebViewAdapter Web View
 - AppBrandDrawableView**

← 创建画布 ⋮

asfdqwerqwerwqerwqerwqerwerweqrqwrewqer
asfdqwerqwerwqerwqerwqerwerweqrqwrewqer
asfdqwerqwerwqerwqerwqerwerweqrqwrewqer
asfdqwerqwerwqerwqerwqerwerweqrqwrewqer
asfdqwerqwerwqerwqerwqerwerweqrqwrewqer
asfdqwerqwerwqerwqerwqerwerweqrqwrewqer
asfdqwerqwerwqerwqerwqerwerweqrqwrewqer

drawImage	fillText
fill	stroke
clearRect	beginPath
closePath	moveTo

小程序 VS RN / Weex

- UI上完全跨平台
- 前端学习成本低

- ❖ 开发能力局限性较大
- ❖ 性能不够

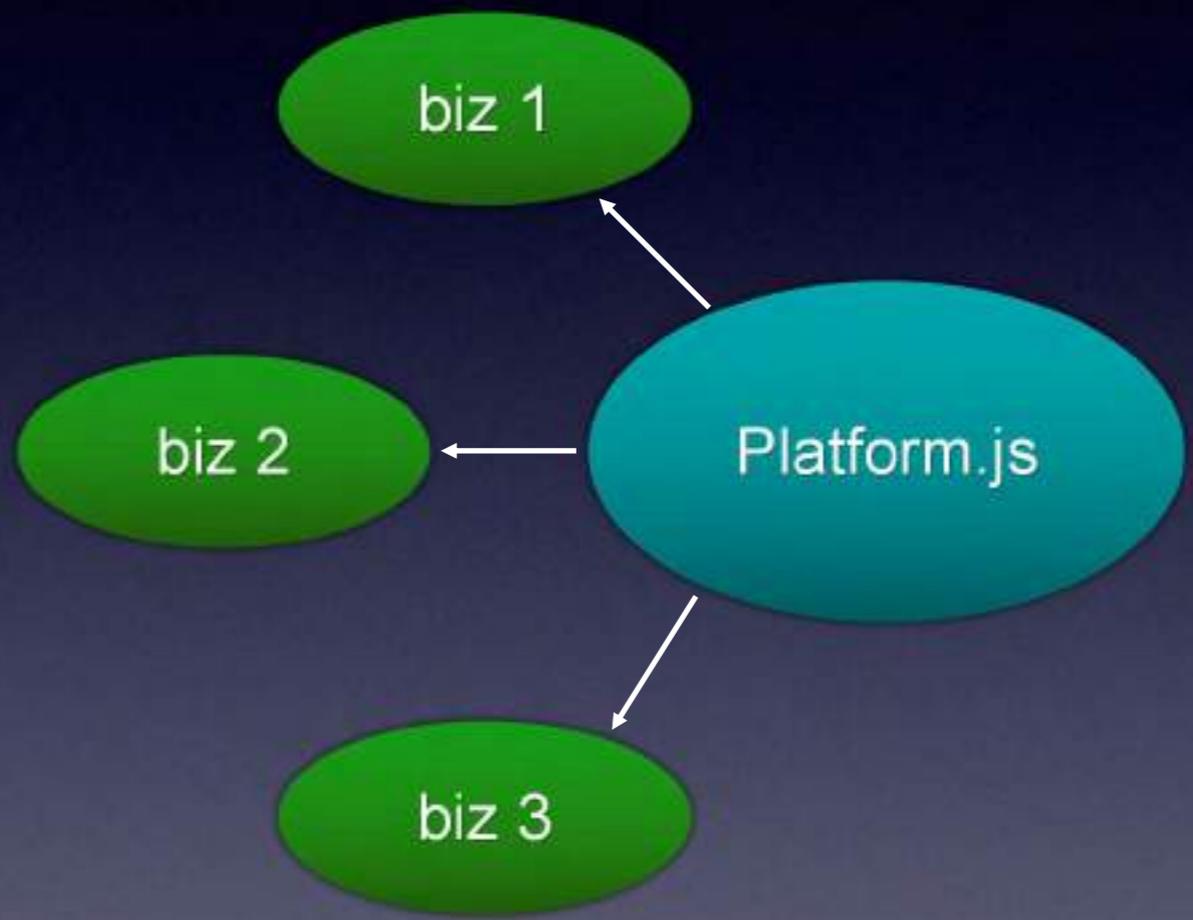
- 几乎可以完全实现桌面应用的开发需求
- 性能媲美native

- ❖ UI样式上受制于系统特性

我们对RN的一些改造

跨平台代码复用

Js bundle 打包拆分



页面加载速度优化

创建RootView



创建ReactContext

NativeModule

ViewManager

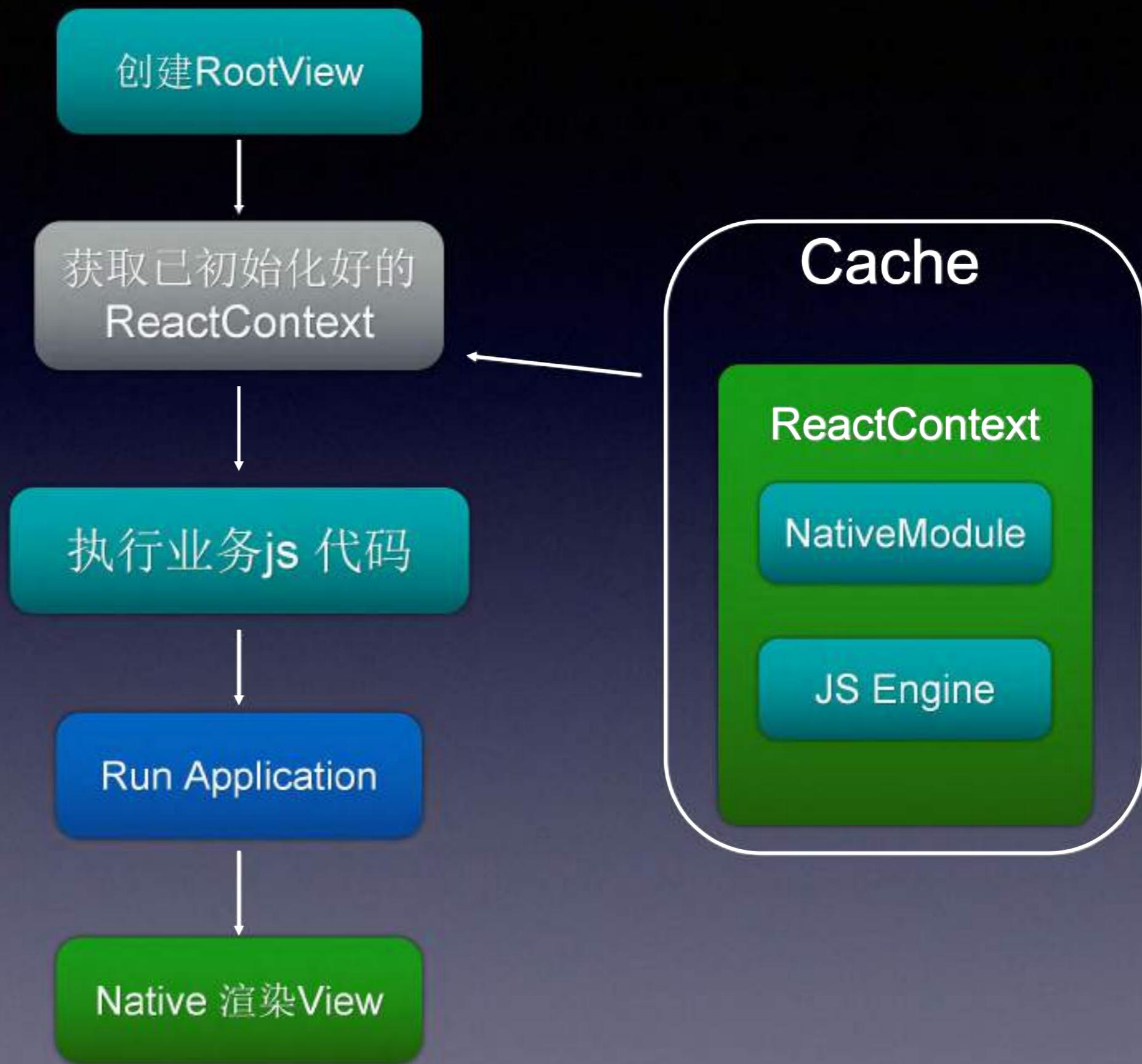
JS Engine

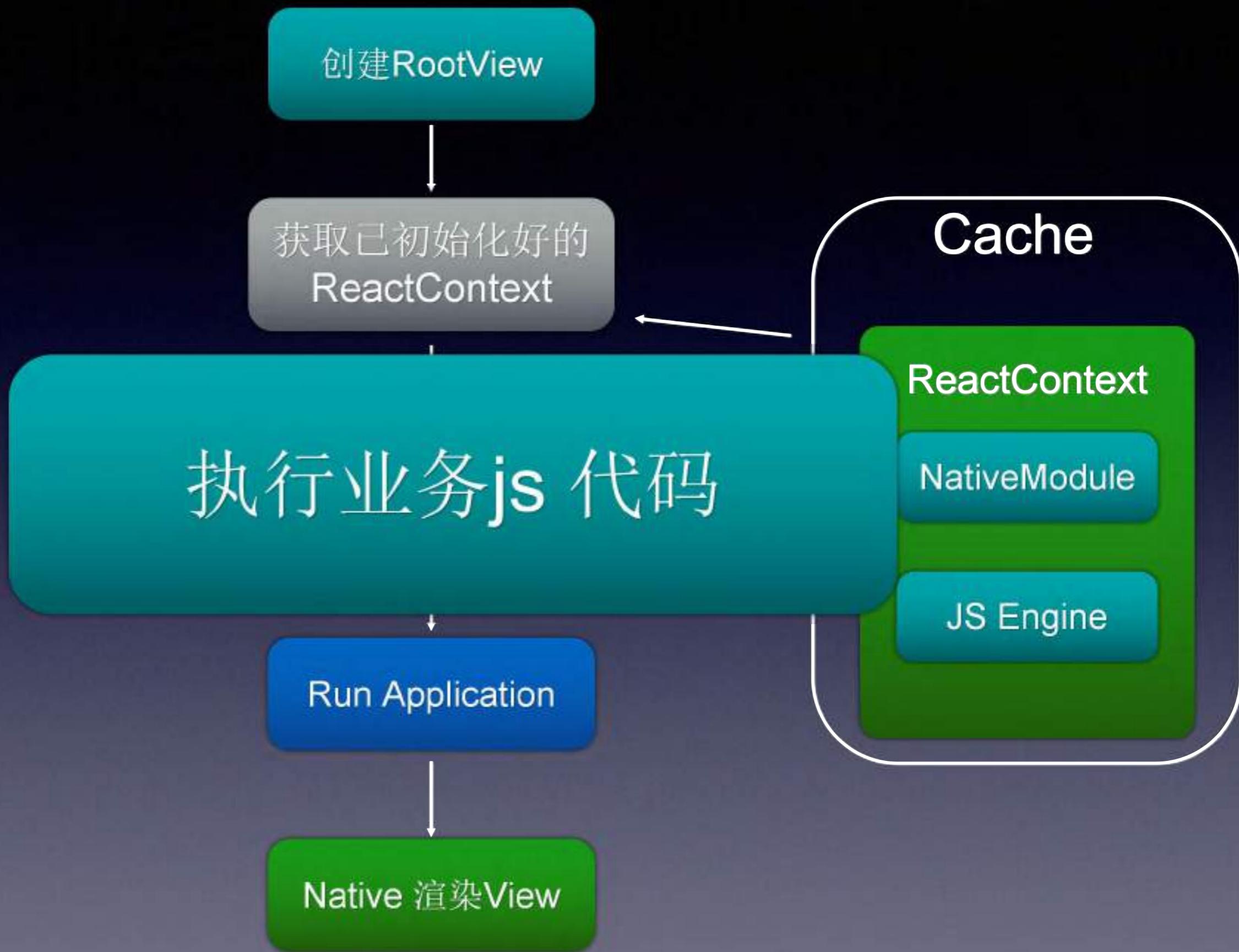


Run Application



Native 渲染View





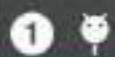
Unbundle + NativeRequire

```
__d("react/lib/ReactChildReconciler.js",  
function(global, require, module, exports) {  
  var ReactReconciler=require("react/lib/ReactReconciler.js");  
  
  var instantiateReactComponent=require("react/ReactComponent.js")  
  module.exports=ReactChildReconciler;  
},  
"react/lib/ReactChildReconciler.js");  
  
__d("react/lib/instantiateReactComponent.js",  
function(global, require, module, exports) {  
  'use strict';  
  module.exports=instantiateReactComponent;  
},  
"react/lib/instantiateReactComponent.js");
```

JS 后台执行代码的能力

JS 后台执行代码的能力

- 业务代码可以在后台执行任务或者接受广播
- 第二次页面打开速度飞快
- 页面之间共享内存，交互更方便，节省内存
- 业务之间js context互相隔离，监控各业务消耗内存



3:15



RN 配置

HybridId:

ModuleName:

ViewName:

打开RN页面

配置HYBRIDID调试信息

LOG日志查看

与Native页面混搭

A

B

C

home

list

首页

List页

RootView

order

填单页

order confirm

order info

订单
确认页

订单
详情页

RootView

ListView

1. 如何处理View的复用
2. 如何处理快速滚动

View 复用

item 1

```
@Override
```

```
public View getView(int position, View convertView,
```

```
    Fruit fruit = getItem(position);
```

```
    View view;
```

```
    if (convertView == null) {
```

```
        view = LayoutInflater.from(getContext()).in
```

```
    } else {
```

```
        view = convertView;
```

```
    }
```

item 1

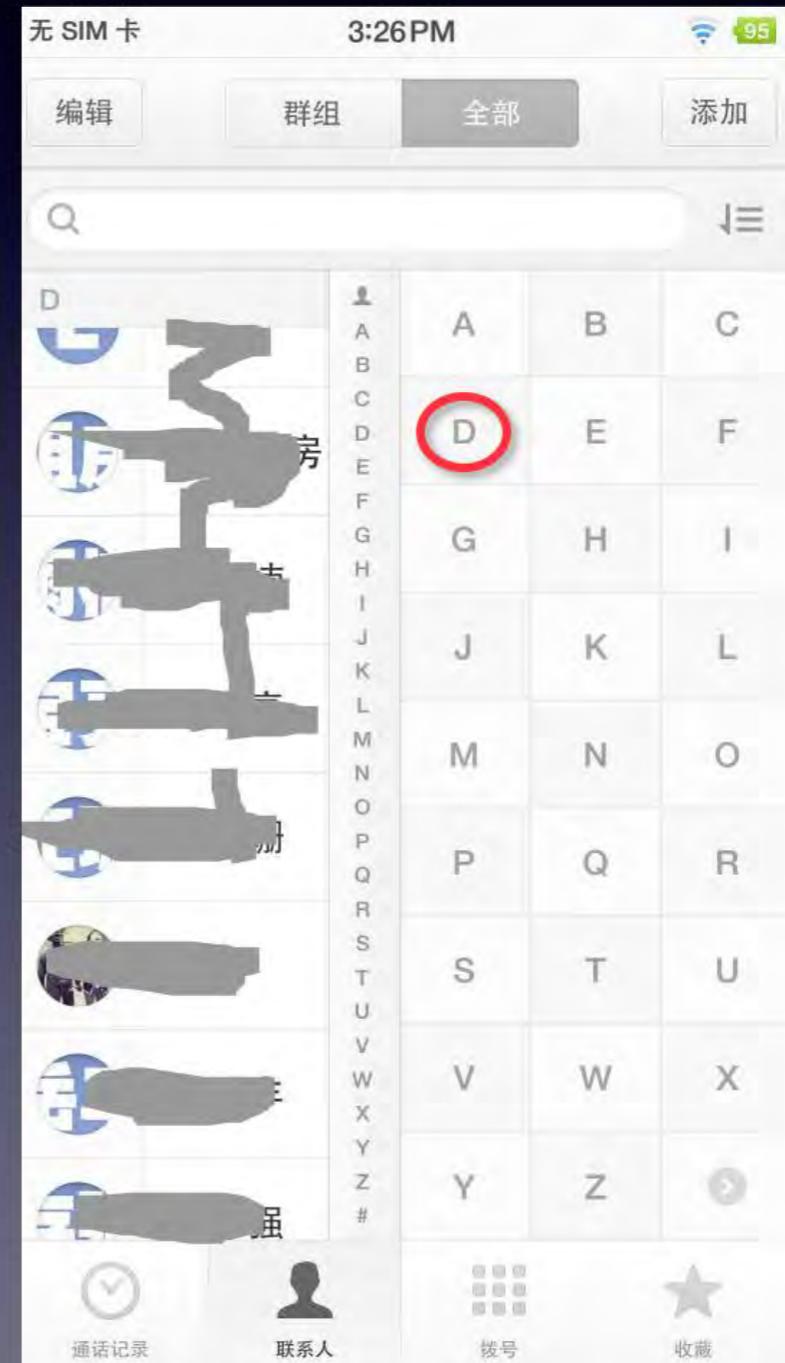
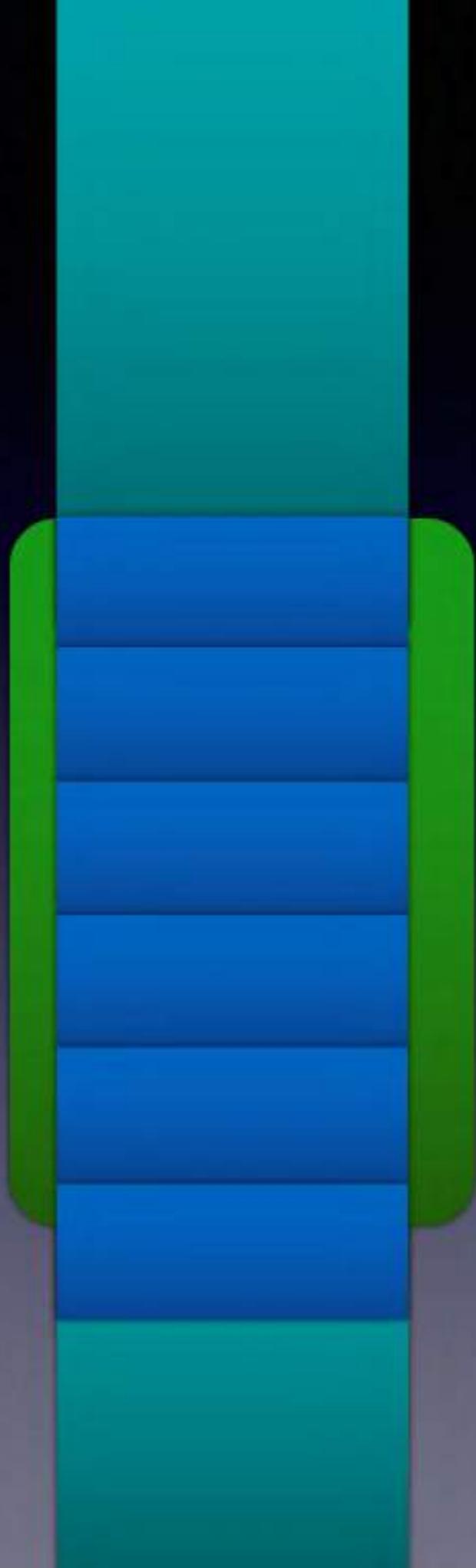
View 复用

能不能做一个View的数据到属性的映射

验证码 1

验证码 2

ListView 快速滚动



RN资源包更新和版本控制

还不能做哪些事

1. Canvas 支持
2. 异步带来的各种问题
 1. 布局结果
 2. 同步获取数据
 3. ...

JS 写代码会不会很容易被破解

反编译看到代码

1. 代码必须经过minify
2. 通过修改JS Engine来运行加密后的代码
3. iOS通过修改系统jscore来获取代码
4. 核心代码模块应该放在动态连接库

篡改代码后运行

1. 必须通过一定的安全校验算法来保证源代码的完整性和安全性

跨平台开发框架该往哪个方向发展？



Brent Vatne REP · Exponent

A big performance bottleneck with animations is that we need to send over the animated value on each frame from the JavaScript thread, and so when we are doing things like pushing new scenes with the Navigator, we need to defer rendering the scene using InteractionManager until the transition is completed.

Another example is loading spinners - if we make them a pure JS component powered by Animated, then they will freeze whenever we block for some frames on the JS thread to do a bunch of work -- spinners exist to prevent these kinds of janky experiences! So the only good solution currently is to use a native one or an image.



寒冬winter

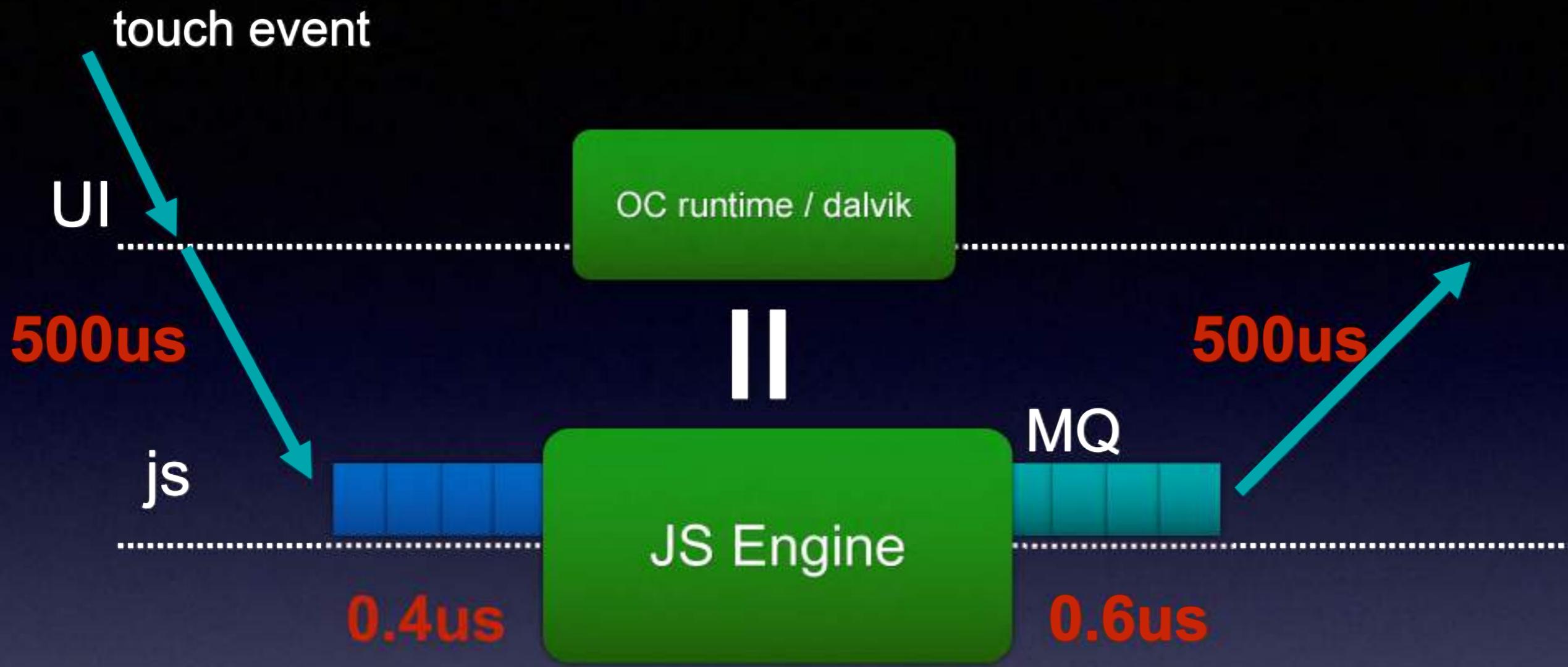
11月25日 20:13 来自 微博 weibo.com

那我说说，我给weex提了一个proposal，在这里 [网页链接](#)

因为长期以来，weex的js-native通讯高消耗，造成了js响应高频事件会很卡，比如pan-move，如果你想做一个拖动效果，这将会非常非常蛋疼，所以我构思了这样一种表达式绑定机制，在一次手势的start中，设定绑定关系，move时由native去执行，最后由native在手势结束时销毁。

这样可以有效解决weex的手势，做一些炫酷的效果，现在等着weex的owner们响应啦~ [收起全文](#) ^

1. listview
2. canvas
3. 所有的动画



render

render

16000us

VR

For Mac

For
Winphone

**RN, Weex 是一种完全有能力替代
现有Native开发方式的框架**

Q&A

