REACT NATIVE大型项目通用化架构

莫卓颖

腾讯高级前端工程师

跨平台

支持IOS、Android、浏览器

高性能

首屏、首次交互时间

高效率开发

组件生态、开发体系

动态更新

支持版本动态更新

➢ 方案比较

	Flutter	React Native	Weex
组件生态	26	11856	458
动态更新	Android支持	均支持	均支持
同构	两端同构	两端同构	三端同构
开发框架	Dart	React	Vue
安装包体积	较大	较小	较小
性能	较好	一般	一般

维护多套代码

介于Native与H5之间的性能

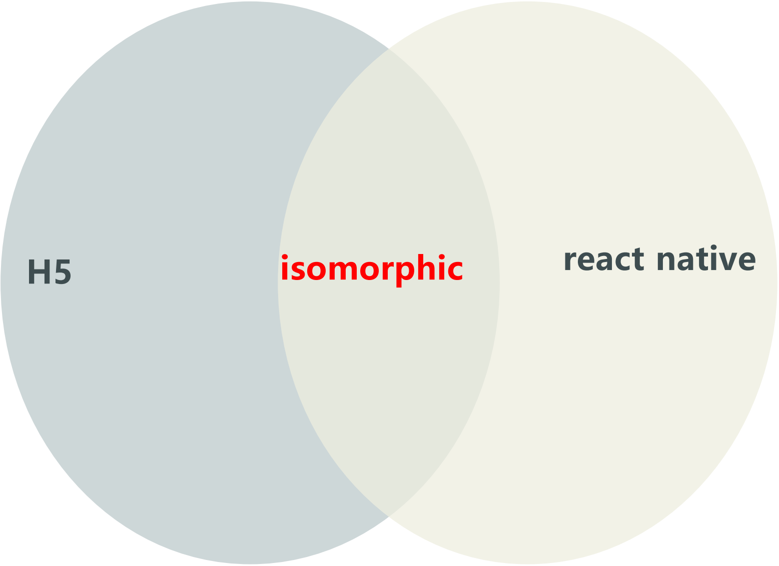
异常概率更高

章节
Part 01

强同构体系

➢ 优势

- 效率以及维护成本
- 产品多平台推广
- 容灾互备

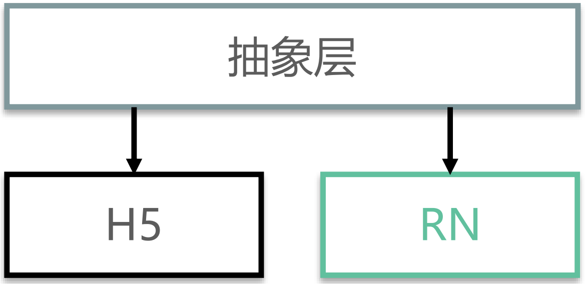


● 方案1：构建跨两个平台的抽象层

抽象一套统一的开发方式

已有方案：Microsoft/reactxp

X 业务需要完全重构，再增加额外学习成本

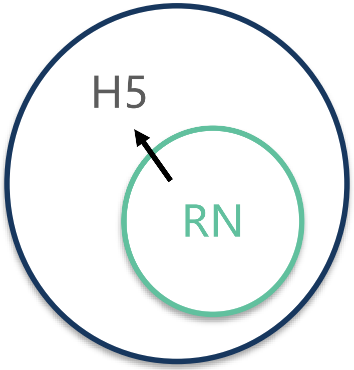


● 方案2：由React Native向H5转换

一套RN代码，部署两个场景

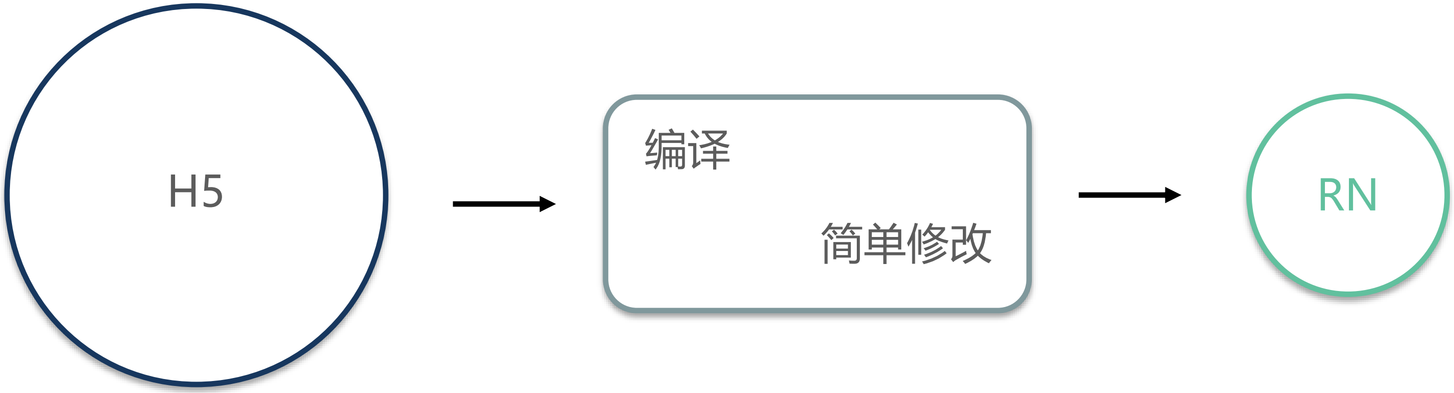
已有方案：taobaofed/react-web，
necolas/react-native-web

X 对新业务适用，不适用已有业务

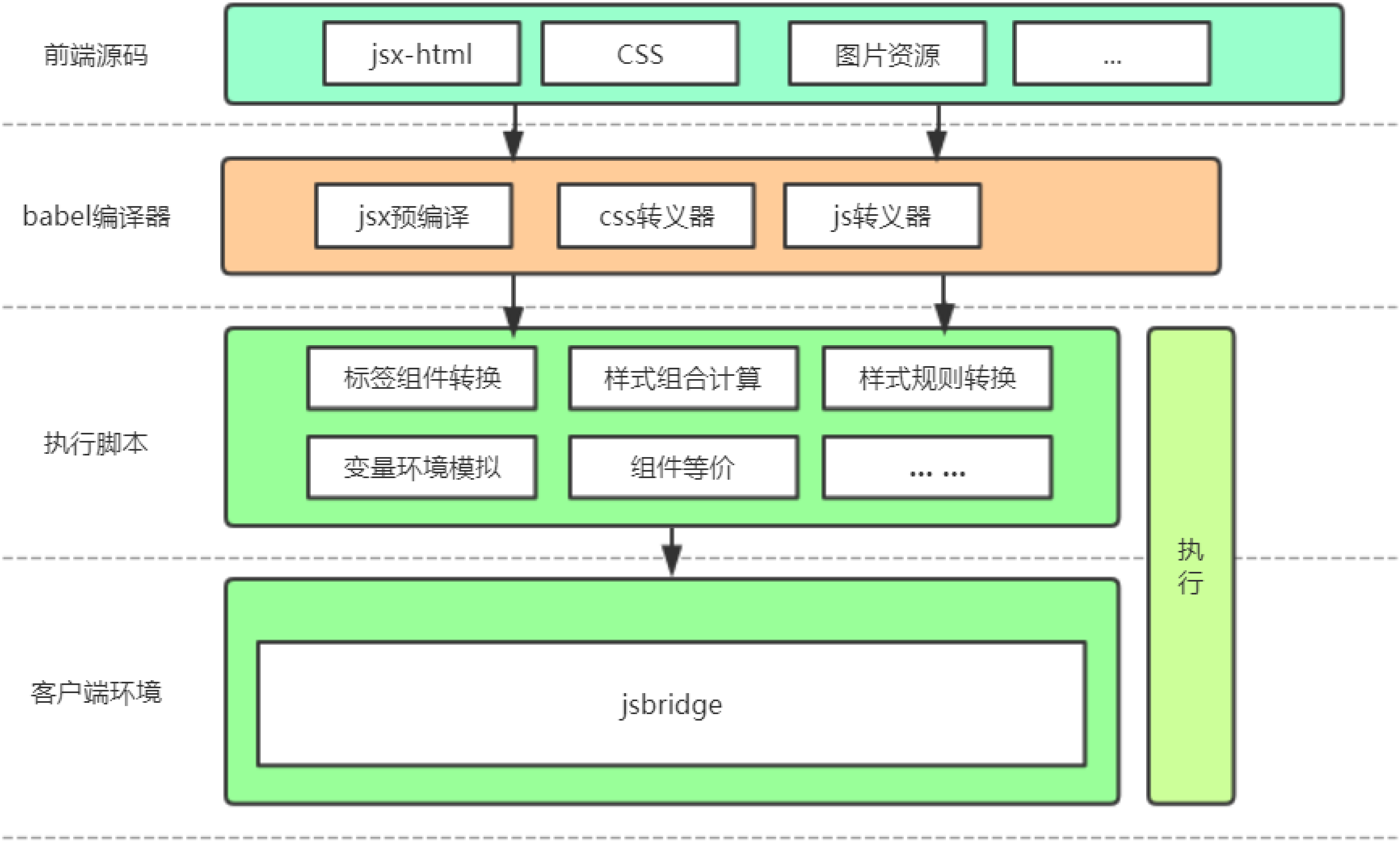


● 方案：由H5向React Native转换

一套H5代码，经过自动化构建和简单修改，实现到RN的强同构转换



优势	1. 基于现有H5业务改造，符合团队业务情况 2. 无需太多RN相关知识储备，H5即可开发 3. 旧页面迁移成本小



前端源码

jsx-html

CSS

图片资源

...

babel编译器

jsx预编译

css转义器

js转义器

执行脚本

标签组件转换

样式组合计算

样式规则转换

变量环境模拟

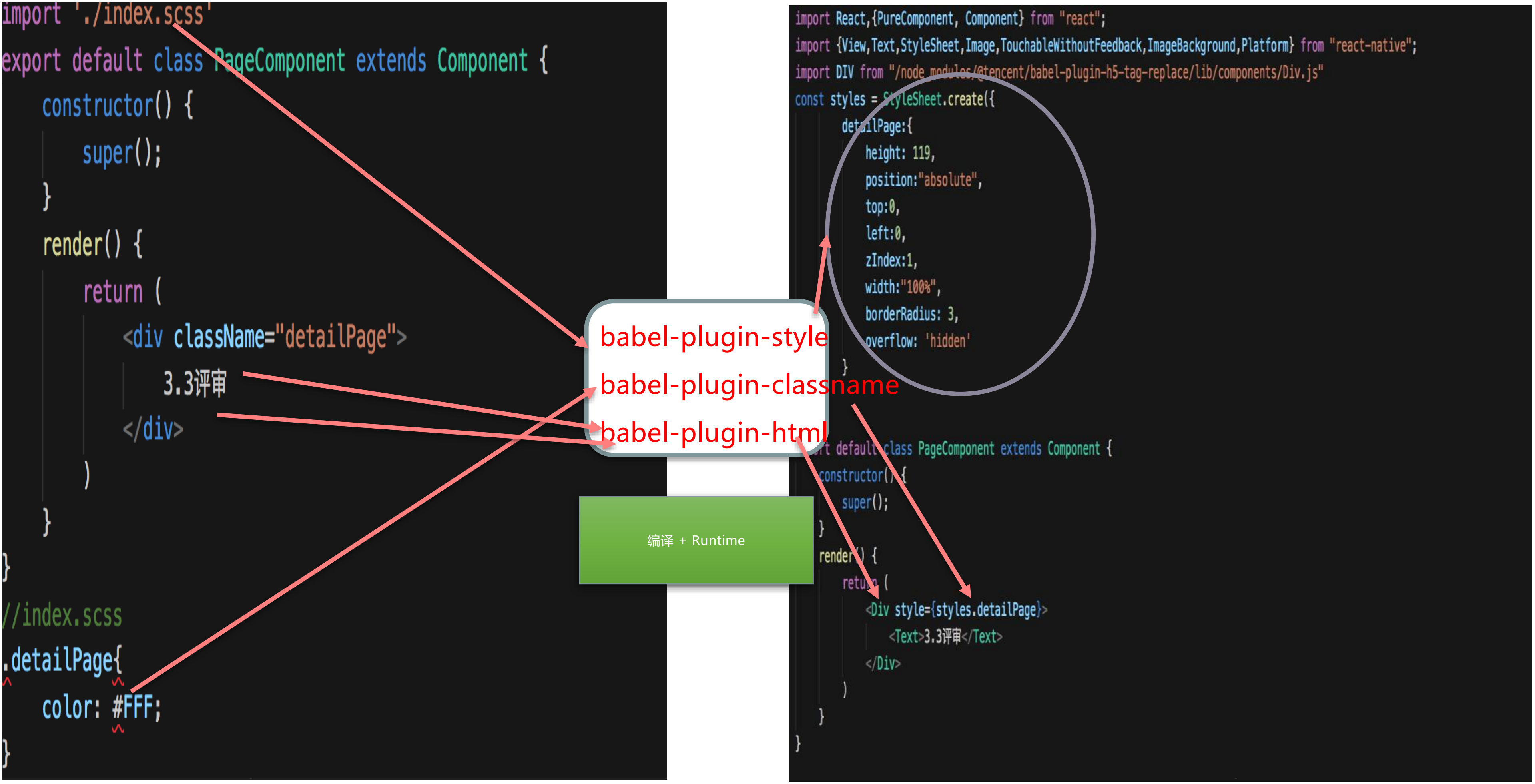
组件等价

... ...

客户端环境

jsbridge

执行



```
import './index.scss'
export default class PageComponent extends Component {
  constructor() {
    super();
  }
  render() {
    return (
      <div className="detailPage">
        3.3评审
      </div>
    )
  }
}
//index.scss
.detailPage{
  color: #FFF;
}
```

babel-plugin-style

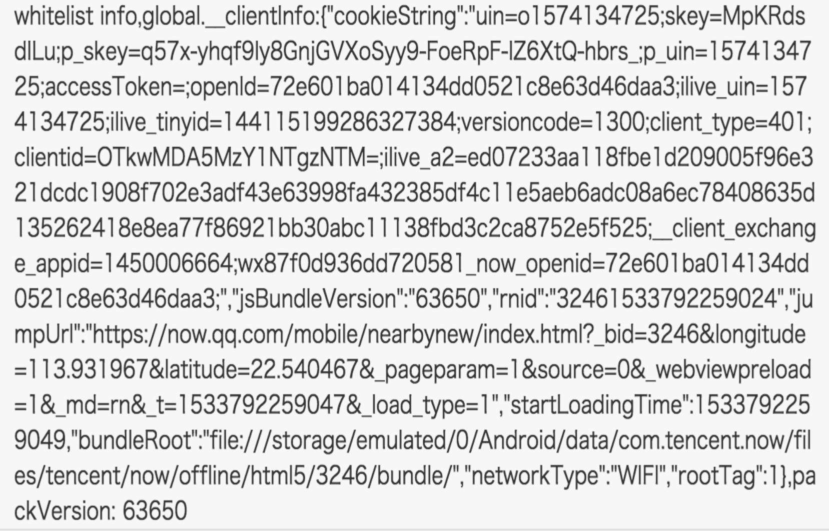
babel-plugin-classname

babel-plugin-html

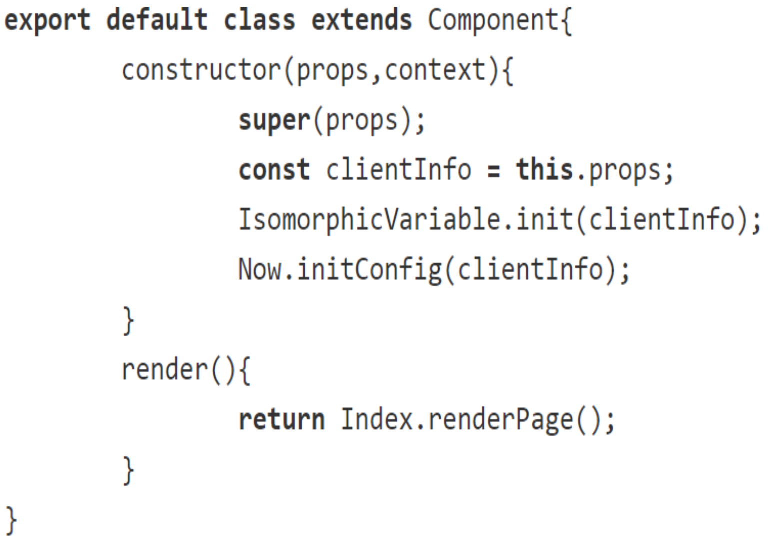
编译 + Runtime

```
import React,{PureComponent, Component} from "react";
import {View,Text,StyleSheet,Image,TouchableWithoutFeedback,ImageBackground,Platform} from "react-native";
import DIV from "/node_modules/@tencent/babel-plugin-h5-tag-replace/lib/components/Div.js"
const styles = StyleSheet.create({
    detailPage:{
        height: 119,
        position:"absolute",
        top:0,
        left:0,
        zIndex:1,
        width:"100%",
        borderRadius: 3,
        overflow: 'hidden'
    }

rt default class PageComponent extends Component {
  constructor() {
    super();
  }
  render() {
    return (
      <Div style={styles.detailPage}>
        <Text>3.3评审</Text>
      </Div>
    )
  }
}
```

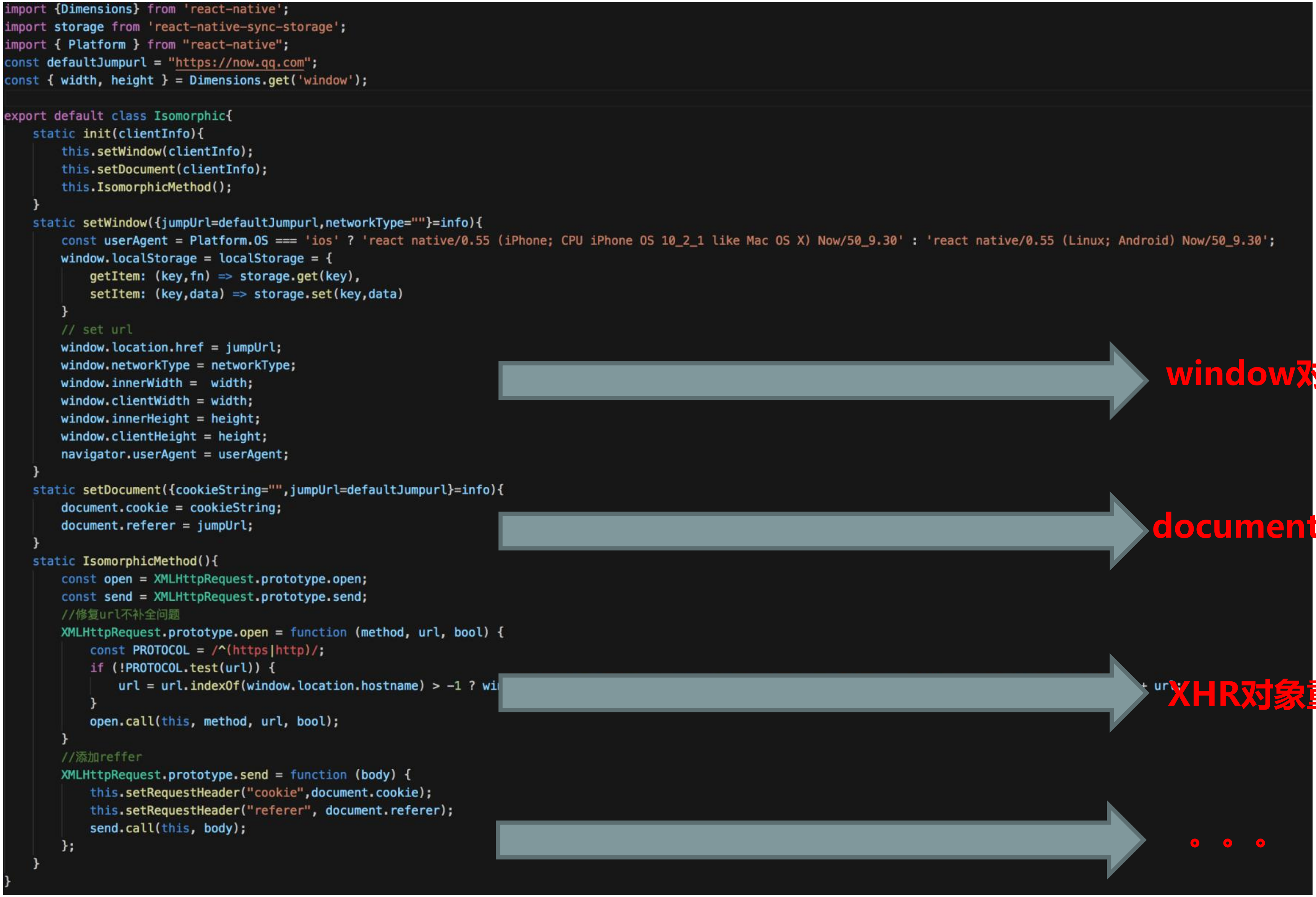
➢ 浏览器环境模拟



```
whitelist info,global.__clientInfo:{"cookieString":"uin=o1574134725;skey=MpKRds
dlLu;p_skey=q57x-yhqf9ly8GnjGVXoSyy9-FoeRpF-lZ6XtQ-hbrs_;p_uin=15741347
25;accessToken=;openId=72e601ba014134dd0521c8e63d46daa3;ilive_uin=157
4134725;ilive_tinyid=144115199286327384;versioncode=1300;client_type=401;
clientid=OTkwMDA5MzY1NTgzNTM=;ilive_a2=ed07233aa118fbe1d209005f96e3
21dcdc1908f702e3adf43e63998fa432385df4c11e5aeb6adc08a6ec78408635d
135262418e8ea77f86921bb30abc11138fbd3c2ca8752e5f525;__client_exchang
e_appid=1450006664;wx87f0d936dd720581_now_openid=72e601ba014134dd
0521c8e63d46daa3;","jsBundleVersion":"63650","rnid":"32461533792259024","ju
mpUrl":"https://now.qq.com/mobile/nearbynew/index.html?_bid=3246&longitude
=113.931967&latitude=22.540467&_pageparam=1&source=0&_webviewpreload
=1&_md=rn&_t=1533792259047&_load_type=1","startLoadingTime":153379225
9049,"bundleRoot":"file:///storage/emulated/0/Android/data/com.tencent.now/fil
es/tencent/now/offline/html5/3246/bundle/","networkType":"WIFI","rootTag":1},pa
ckVersion: 63650
```



```
export default class extends Component{
        constructor(props,context){
                super(props);
                const clientInfo = this.props;
                IsomorphicVariable.init(clientInfo);
                Now.initConfig(clientInfo);
        }
        render(){
                return Index.renderPage();
        }
}
```



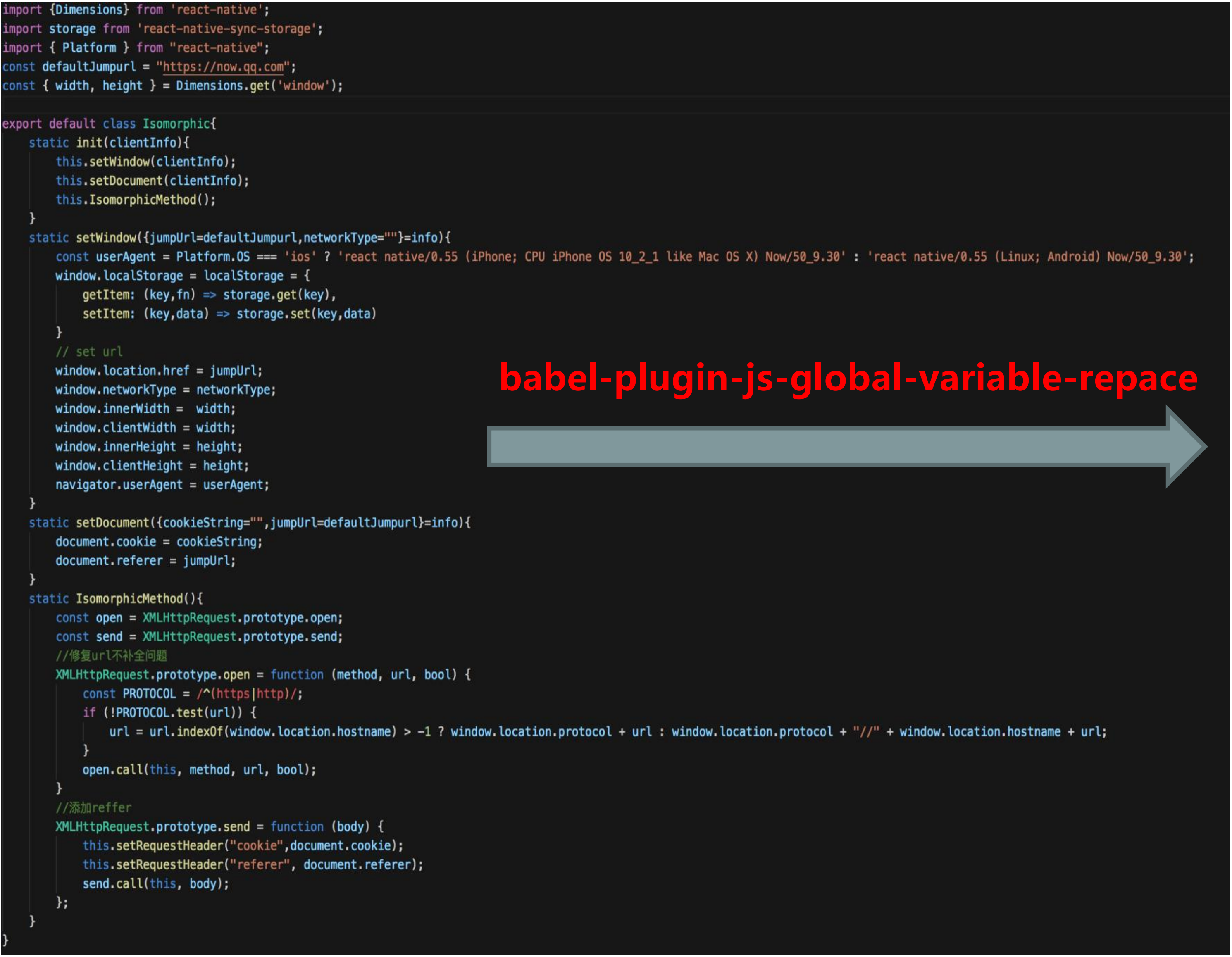
```
import {Dimensions} from 'react-native';
import storage from 'react-native-sync-storage';
import { Platform } from "react-native";
const defaultJumpurl = "https://now.qq.com";
const { width, height } = Dimensions.get('window');

export default class Isomorphic{
    static init(clientInfo){
        this.setWindow(clientInfo);
        this.setDocument(clientInfo);
        this.IsomorphicMethod();
    }
    static setWindow({jumpUrl=defaultJumpurl,networkType=""}=info){
        const userAgent = Platform.OS === 'ios' ? 'react native/0.55 (iPhone; CPU iPhone OS 10_2_1 like Mac OS X) Now/50_9.30' : 'react native/0.55 (Linux; Android) Now/50_9.30';
        window.localStorage = localStorage = {
            getItem: (key,fn) => storage.get(key),
            setItem: (key,data) => storage.set(key,data)
        }
        // set url
        window.location.href = jumpUrl;
        window.networkType = networkType;
        window.innerWidth =  width;
        window.clientWidth = width;
        window.innerHeight = height;
        window.clientHeight = height;
        navigator.userAgent = userAgent;
    }
    static setDocument({cookieString="",jumpUrl=defaultJumpurl}=info){
        document.cookie = cookieString;
        document.referer = jumpUrl;
    }
    static IsomorphicMethod(){
        const open = XMLHttpRequest.prototype.open;
        const send = XMLHttpRequest.prototype.send;
        //修复url不补全问题
        XMLHttpRequest.prototype.open = function (method, url, bool) {
            const PROTOCOL = /^(https|http)/;
            if (!PROTOCOL.test(url)) {
                url = url.indexOf(window.location.hostname) > -1 ? wi                + ur
            }
            open.call(this, method, url, bool);
        }
        //添加reffer
        XMLHttpRequest.prototype.send = function (body) {
            this.setRequestHeader("cookie",document.cookie);
            this.setRequestHeader("referer", document.referer);
            send.call(this, body);
        };
    }
}
```

window对象重置

document对象重置

XHR对象重置

。。。

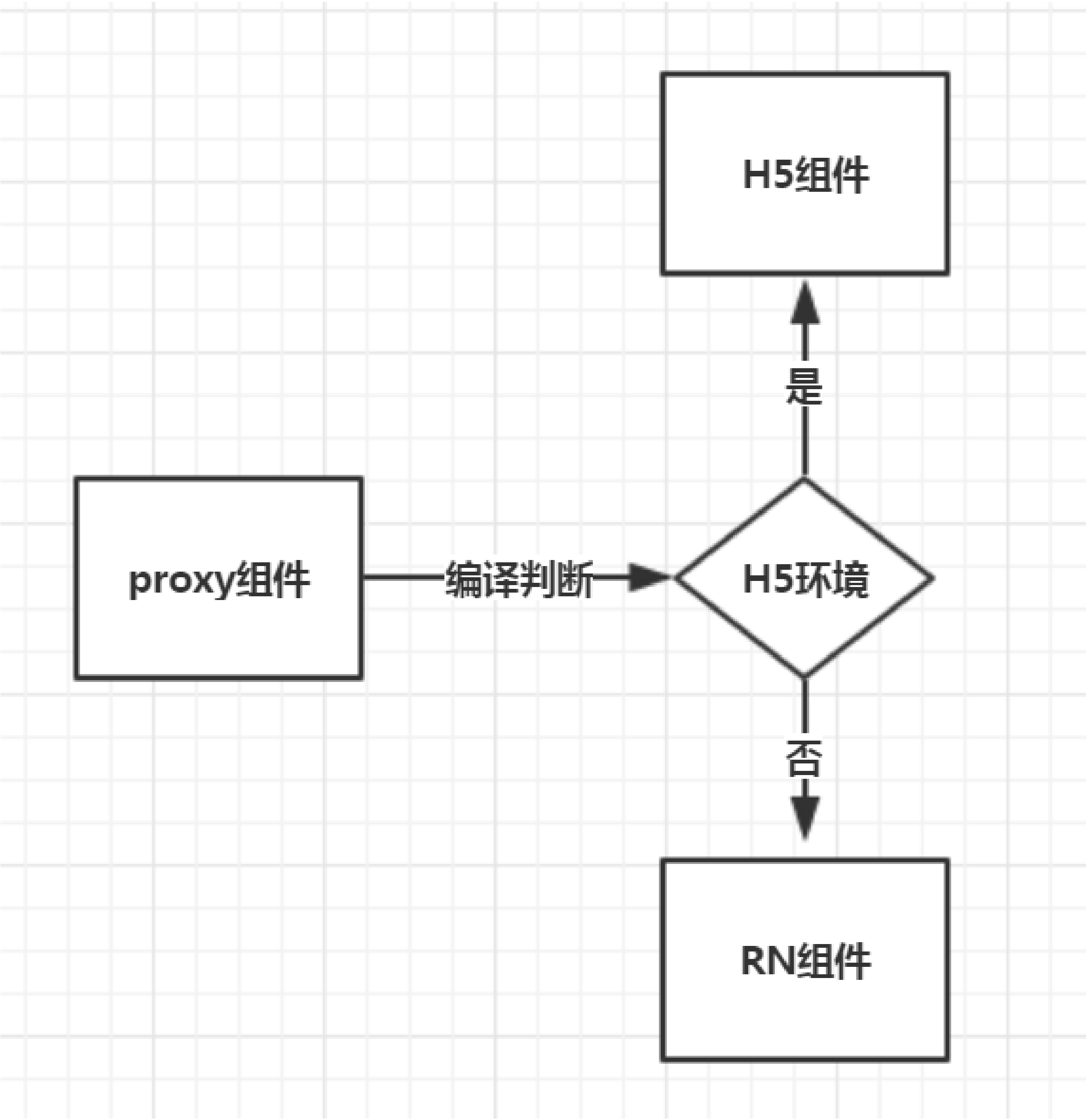


```
import {Dimensions} from 'react-native';
import storage from 'react-native-sync-storage';
import { Platform } from "react-native";
const defaultJumpurl = "https://now.qq.com";
const { width, height } = Dimensions.get('window');

export default class Isomorphic{
    static init(clientInfo){
        this.setWindow(clientInfo);
        this.setDocument(clientInfo);
        this.IsomorphicMethod();
    }
    static setWindow({jumpUrl=defaultJumpurl,networkType=""}=info){
        const userAgent = Platform.OS === 'ios' ? 'react native/0.55 (iPhone; CPU iPhone OS 10_2_1 like Mac OS X) Now/50_9.30' : 'react native/0.55 (Linux; Android) Now/50_9.30';
        window.localStorage = localStorage = {
            getItem: (key,fn) => storage.get(key),
            setItem: (key,data) => storage.set(key,data)
        }
        // set url
        window.location.href = jumpUrl;
        window.networkType = networkType;
        window.innerWidth =  width;
        window.clientWidth = width;
        window.innerHeight = height;
        window.clientHeight = height;
        navigator.userAgent = userAgent;
    }
    static setDocument({cookieString="",jumpUrl=defaultJumpurl}=info){
        document.cookie = cookieString;
        document.referer = jumpUrl;
    }
    static IsomorphicMethod(){
        const open = XMLHttpRequest.prototype.open;
        const send = XMLHttpRequest.prototype.send;
        //修复url不补全问题
        XMLHttpRequest.prototype.open = function (method, url, bool) {
            const PROTOCOL = /^(https|http)/;
            if (!PROTOCOL.test(url)) {
                url = url.indexOf(window.location.hostname) > -1 ? window.location.protocol + url : window.location.protocol + "//" + window.location.hostname + url;
            }
            open.call(this, method, url, bool);
        }
        //添加reffer
        XMLHttpRequest.prototype.send = function (body) {
            this.setRequestHeader("cookie",document.cookie);
            this.setRequestHeader("referer", document.referer);
            send.call(this, body);
        };
    }
}
```

babel-plugin-js-global-variable-repace

require(“window”).location

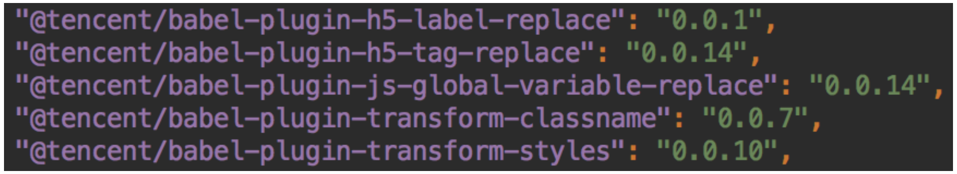
…


H5组件
是
proxy组件
编译判断
H5环境
否
RN组件

➢ 同构接入

- 同构组件 →

1) 基础同构



```
"@tencent/babel-plugin-h5-label-replace": "0.0.1",
"@tencent/babel-plugin-h5-tag-replace": "0.0.14",
"@tencent/babel-plugin-js-global-variable-replace": "0.0.14",
"@tencent/babel-plugin-transform-classname": "0.0.7",
"@tencent/babel-plugin-transform-styles": "0.0.10",
```

- 浏览器环境初始化

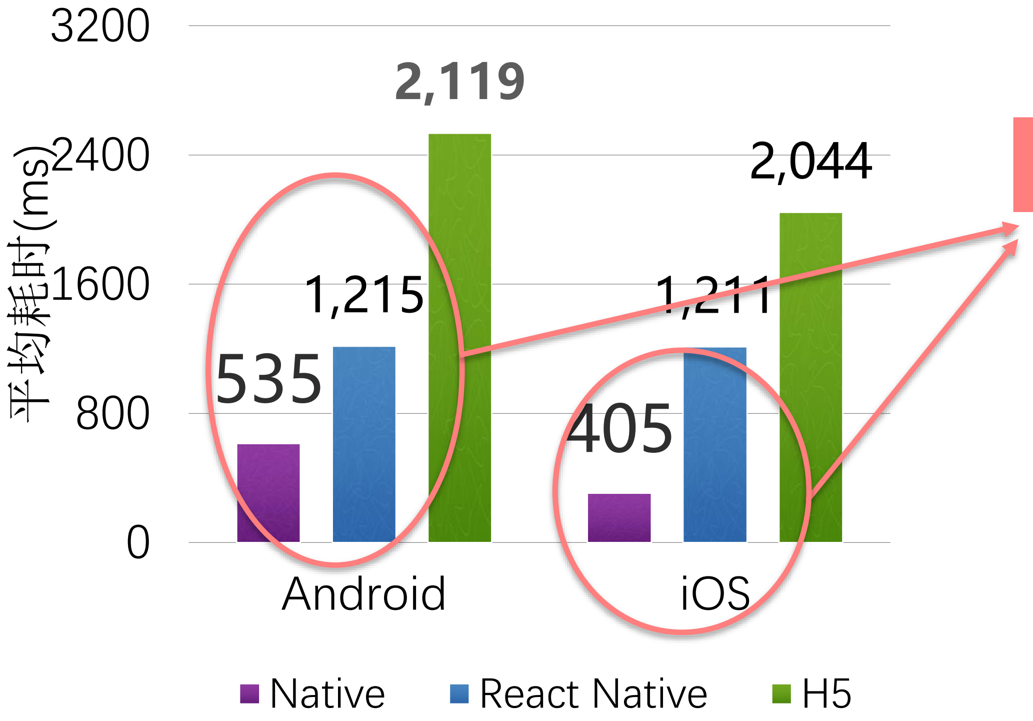
1) cookie，useragent等传递

- 同构组件

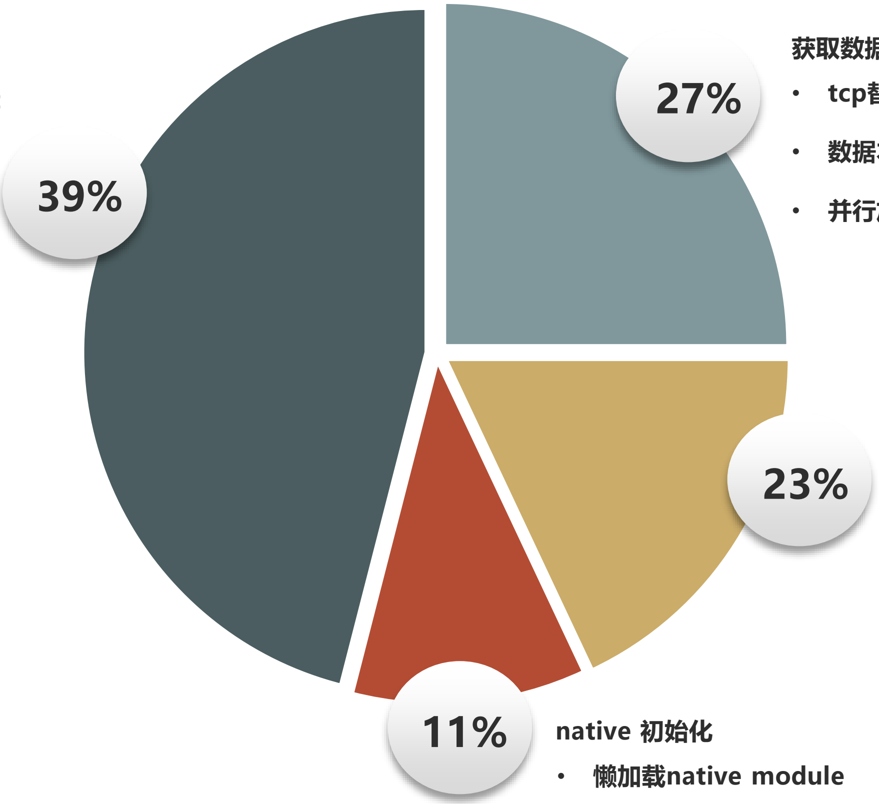
1) flatlist,scrollview等

章节
Part 02

性能优化



问题 处于Native与h5之间的性能



jsBudle加载+require init

- 减少jsBundle体积
- lazyload js module
- 离线包

获取数据

- tcp替换http
- 数据本地缓存
- 并行加载

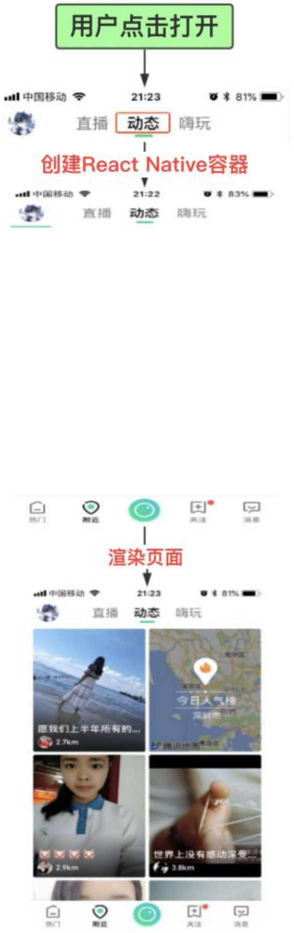
渲染上屏幕

- 渲染区分优先级

native 初始化

- 懒加载native module

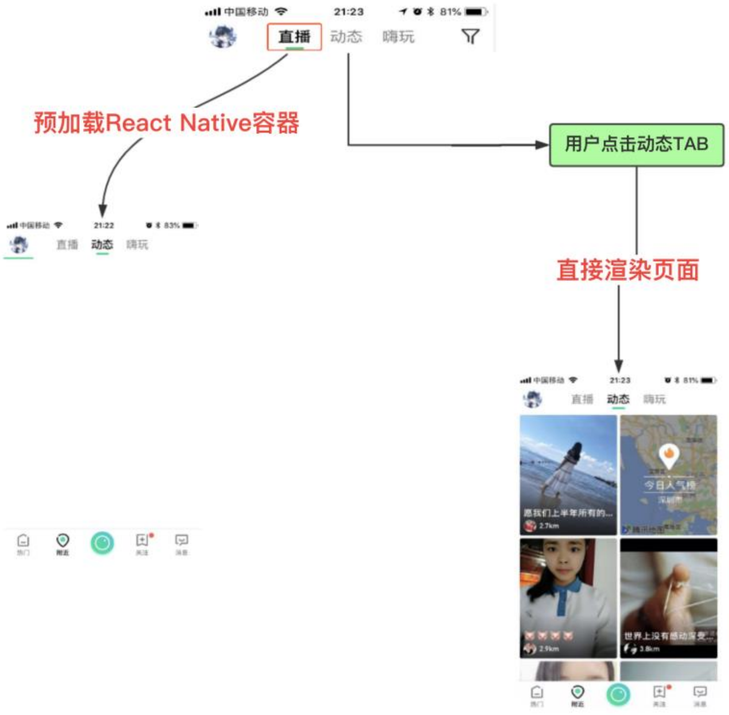
传统方式



用户点击打开

创建React Native容器

渲染页面

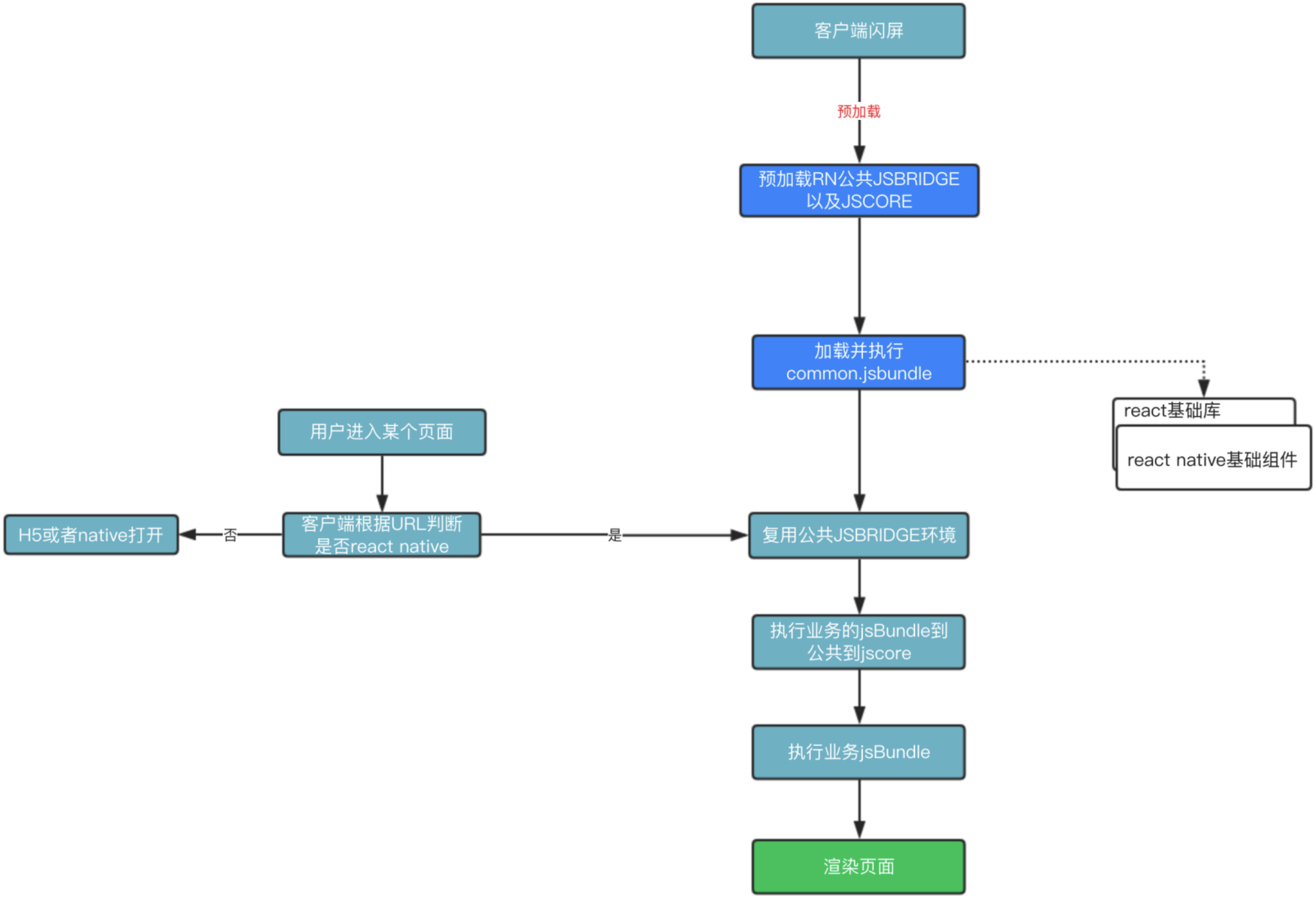


预加载React Native容器

用户点击动态TAB

直接渲染页面

单例方式



客户端闪屏

预加载

预加载RN公共JSBRIDGE
以及JSCORE

加载并执行
common.jsbundle

react基础库

react native基础组件

用户进入某个页面

客户端根据URL判断
是否react native

否

H5或者native打开

是

复用公共JSBRIDGE环境

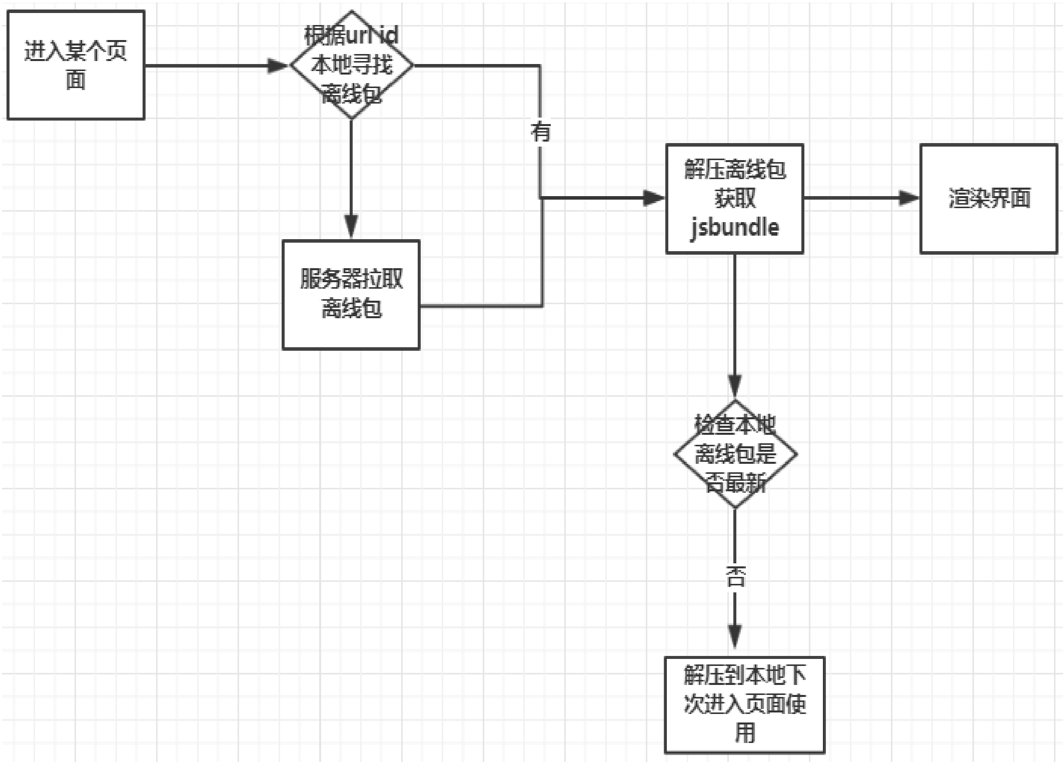
执行业务的jsBundle到
公共到jscore

执行业务jsBundle

渲染页面

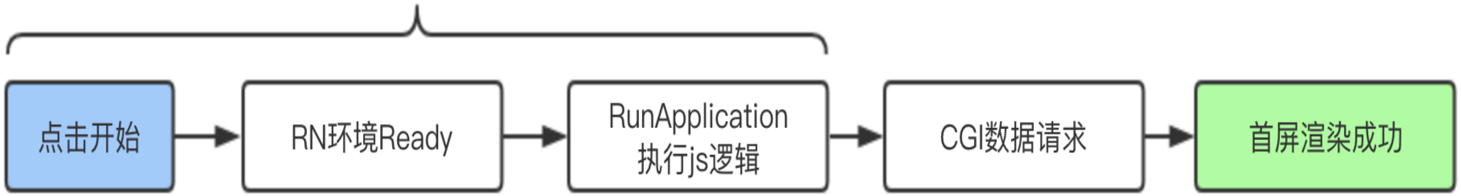
➢ 离线包

- 作资源离线化，解决断网或者网络条件不好情况界面无法展现或者慢
- 三种不同功能的离线包，分别为安装离线包、线上离线包、推送离线包



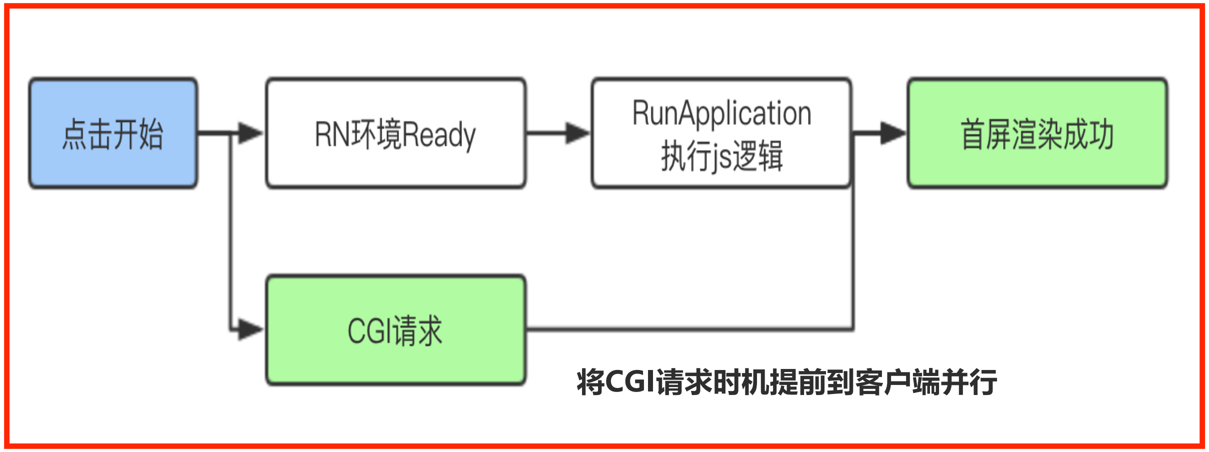
CGI串行等待时间

传统方式：



点击开始 → RN环境Ready → RunApplication 执行js逻辑 → CGI数据请求 → 首屏渲染成功

CGI预加载方式：



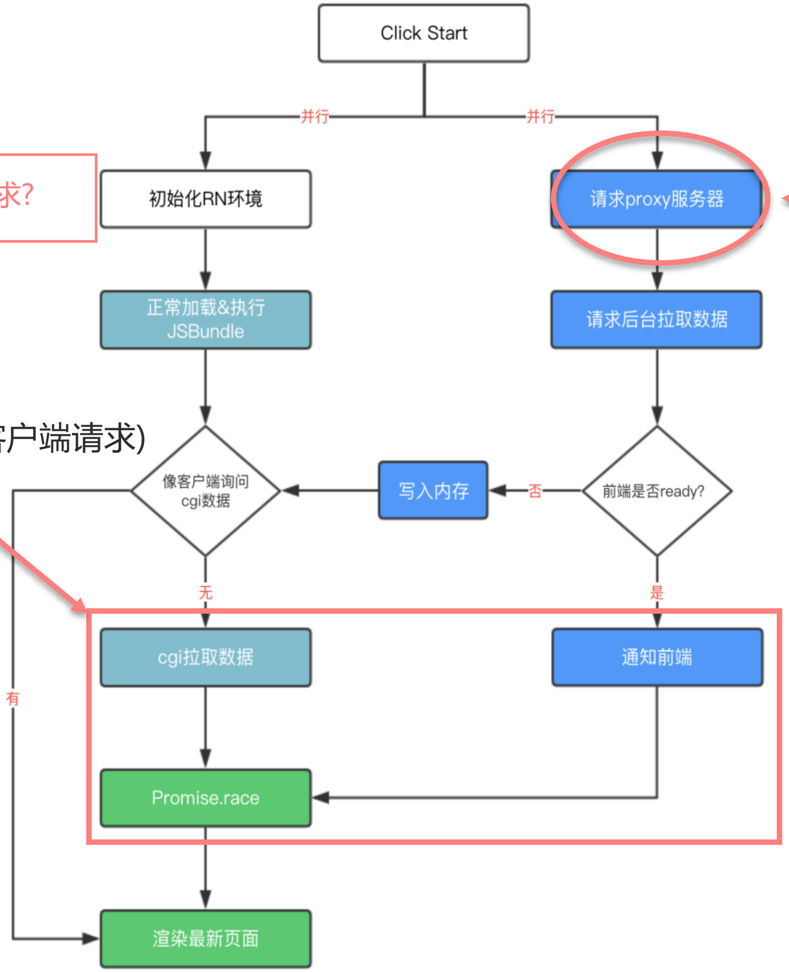
点击开始 → RN环境Ready → RunApplication 执行js逻辑 → 首屏渲染成功

点击开始 → CGI请求 → 首屏渲染成功

将CGI请求时机提前到客户端并行

问题2 JS Ready早于客户端数据请求?

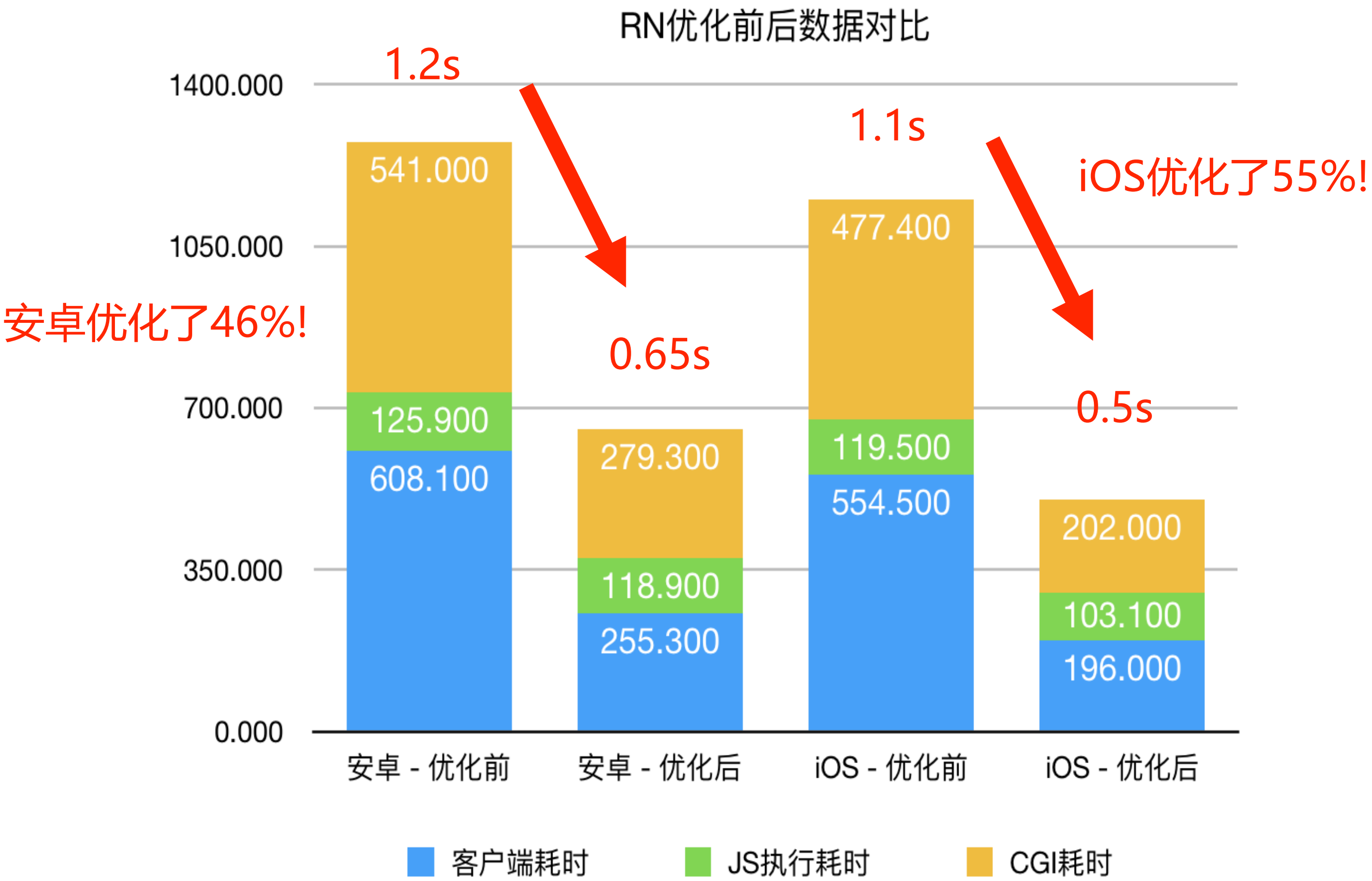
- 非等待，继续发送cgi请求
- Data = Promise.race(cgi, 客户端请求)



问题1 客户端如何知道请求什么CGI？

- 统一请求代理服务器
- 服务端配置各业务cgi
- 从url或cookie中解析参数值

RN优化前后数据对比



1.2s

0.65s

安卓优化了46%!

1.1s

0.5s

iOS优化了55%!

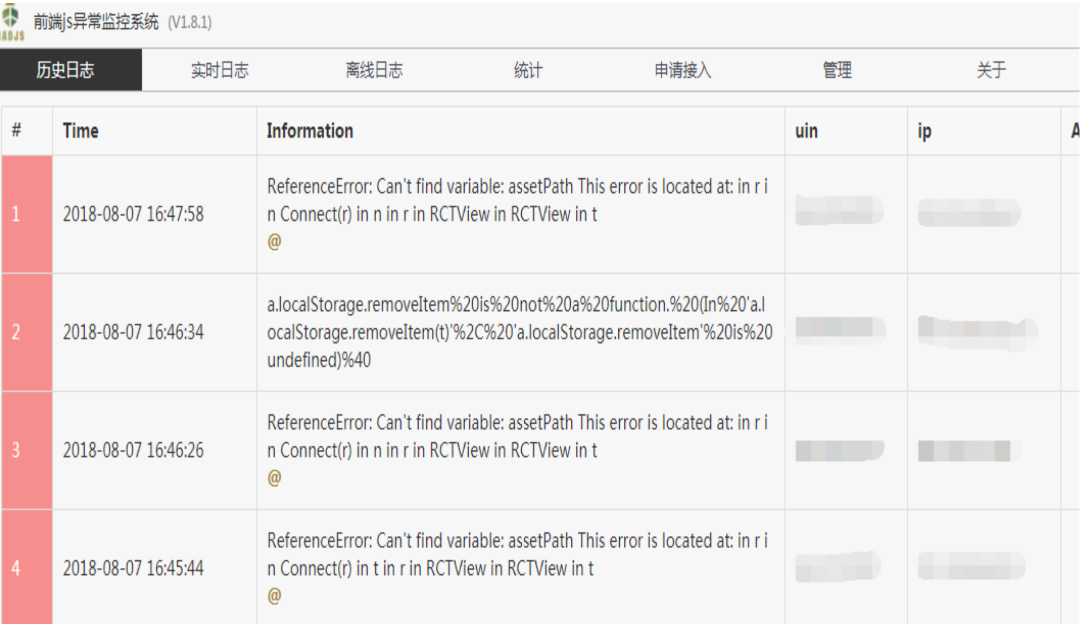
章节

Part 03

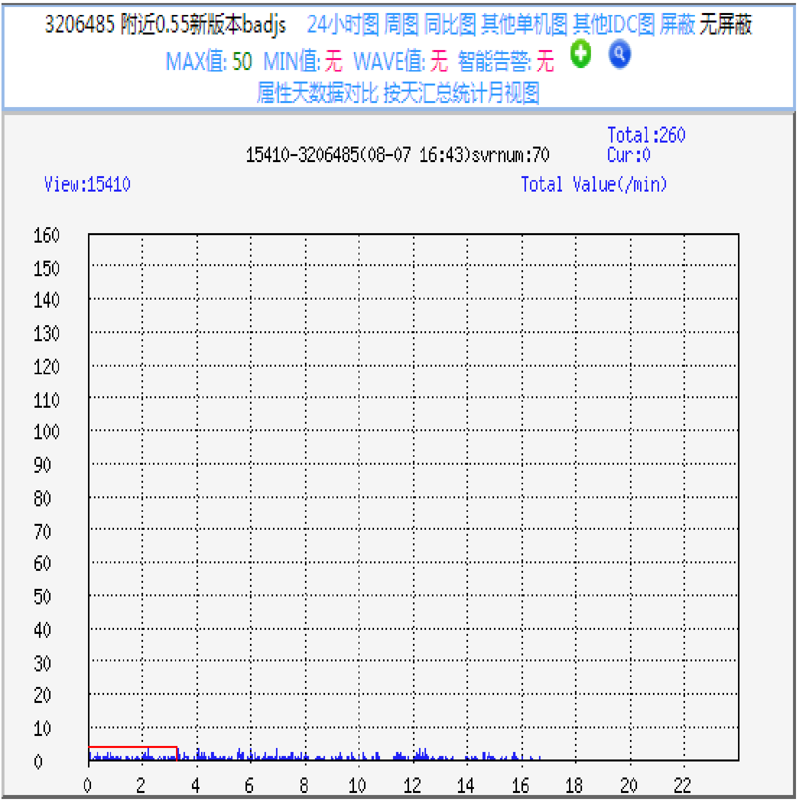
监控体系

前端错误日志上报系统

- 实时上报错误脚本异常日志
- 错误堆栈展示
- 错误脚本量异常智能告警



#	Time	Information	uin	ip	A
1	2018-08-07 16:47:58	ReferenceError: Can't find variable: assetPath This error is located at: in r in Connect(r) in n in r in RCTView in RCTView in t @	[redacted]	[redacted]	
2	2018-08-07 16:46:34	a.localStorage.removeItem%20is%20not%20a%20function.%20(In%20'a.localStorage.removeItem(t)'%2C%20'a.localStorage.removeItem'%20is%20undefined)%40	[redacted]	[redacted]	
3	2018-08-07 16:46:26	ReferenceError: Can't find variable: assetPath This error is located at: in r in Connect(r) in n in r in RCTView in RCTView in t @	[redacted]	[redacted]	
4	2018-08-07 16:45:44	ReferenceError: Can't find variable: assetPath This error is located at: in r in Connect(r) in t in r in RCTView in RCTView in t @	[redacted]	[redacted]	



3206485 附近0.55新版本badjs

MAX值: 50 MIN值: 无 WAVE值: 无 智能告警: 无

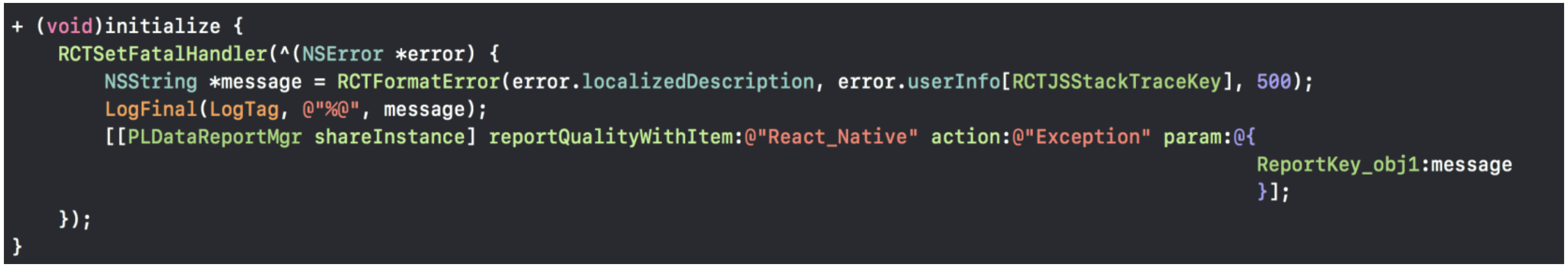
15410-3206485(08-07 16:43)svrnum:70

Total:260
Cur:0

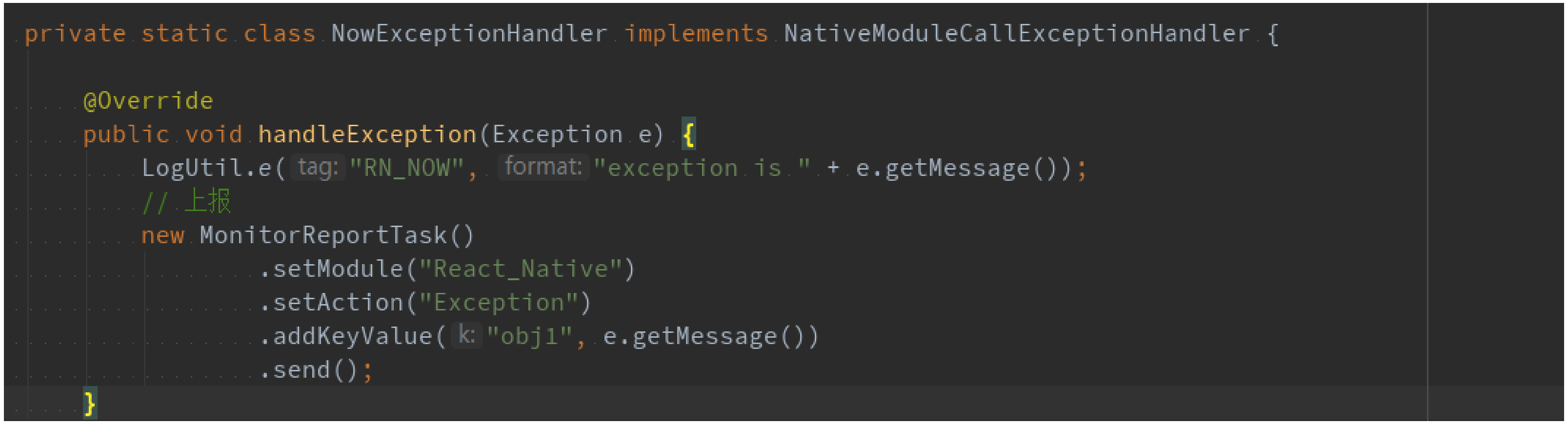
View:15410

➢ 错误日志上报系统

- IOS错误日志上报
- Android错误上报
- 错误脚本量常智能告警



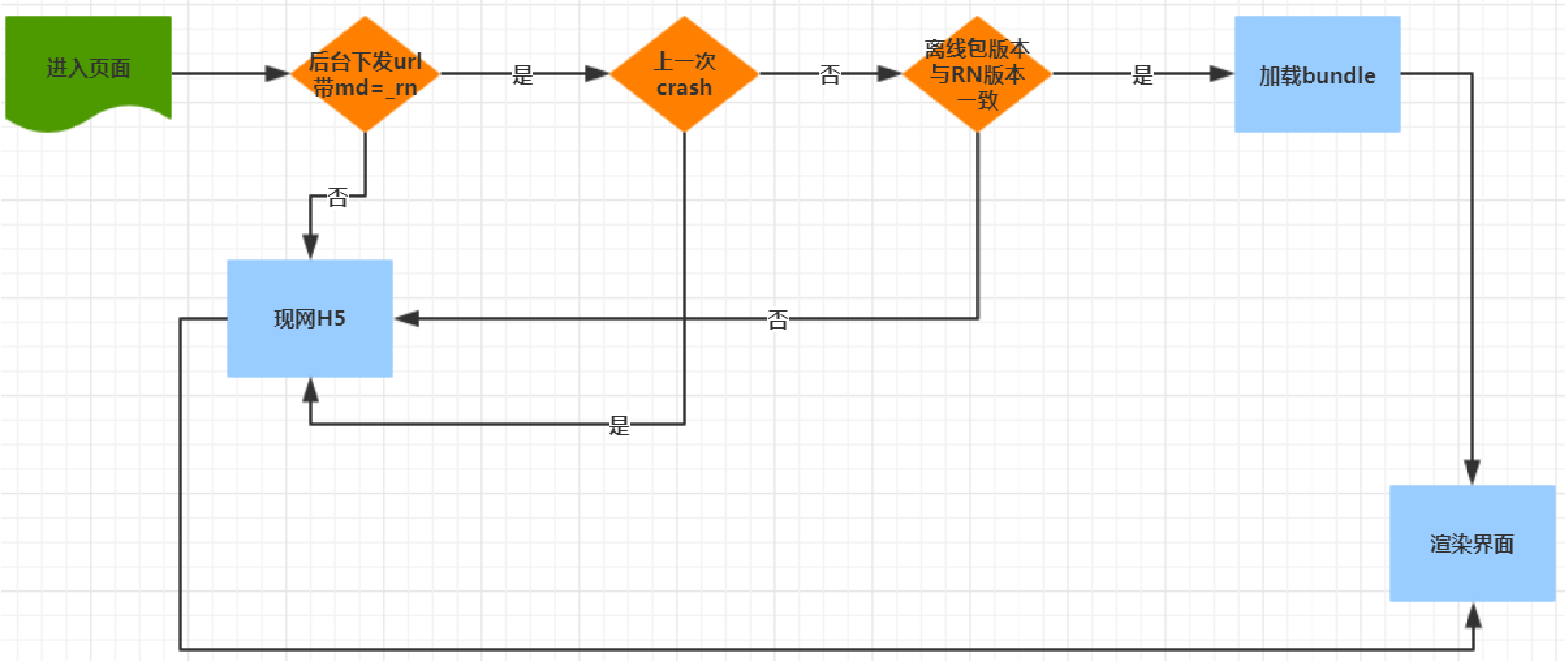
```
+ (void)initialize {
    RCTSetFatalHandler(^(NSError *error) {
        NSString *message = RCTFormatError(error.localizedDescription, error.userInfo[RCTJSStackTraceKey], 500);
        LogFinal(LogTag, @"%@", message);
        [[PLDataReportMgr shareInstance] reportQualityWithItem:@"React_Native" action:@"Exception" param:@{
                                                                                                          ReportKey_obj1:message
                                                                                                          }];
    });
}
```



```
private static class NowExceptionHandler implements NativeModuleCallExceptionHandler {

    @Override
    public void handleException(Exception e) {
        LogUtil.e( tag: "RN_NOW", format: "exception is " + e.getMessage());
        // 上报
        new MonitorReportTask()
                .setModule("React_Native")
                .setAction("Exception")
                .addKeyValue( k: "obj1", e.getMessage())
                .send();
    }
```

➢ 容灾策略

- 不兼容版本自动切换H5
- crash自动切换H5
- url指定切换



END

感谢您的聆听.

-莫卓颖