

Hybrid app开发与前端架构改造实践

——基于Android pad平台

@dm.yang

2015.04.25

关于



- 杨德模 (dm.yang)
 - 唯品会/高级开发工程师
 - 负责特卖会移动平台的前端架构优化
&基础设施建设
- ⌚ <https://github.com/chemdemo>
- ⌚ <http://weibo.com/chemdemo>



结算

小星星 18534276538
广东省广州市荔湾区花海街20号

其他地址 >

送货时间不限



支付方式

-  货到付款(刷卡)
-  货到付款(现金)
-  支付宝网页版

费用详情

商品金额	¥876
唯品会发货，运费	¥0
待支付总额	¥876

开具发票

总金额 **¥876** (已省 ¥2879)

提交订单 19:32

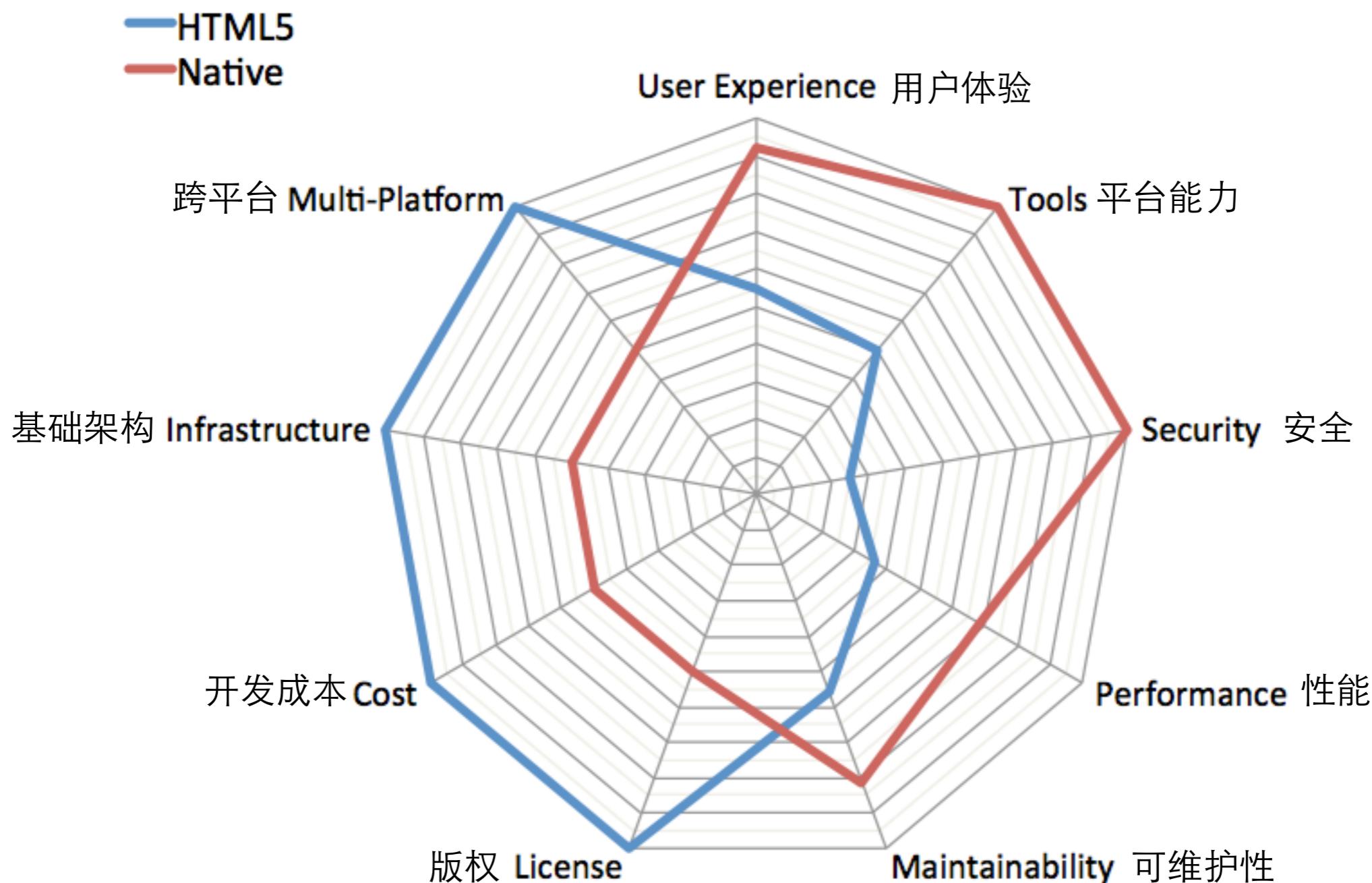
主要内容

► Hybrid实践

► 前端工程体系改造

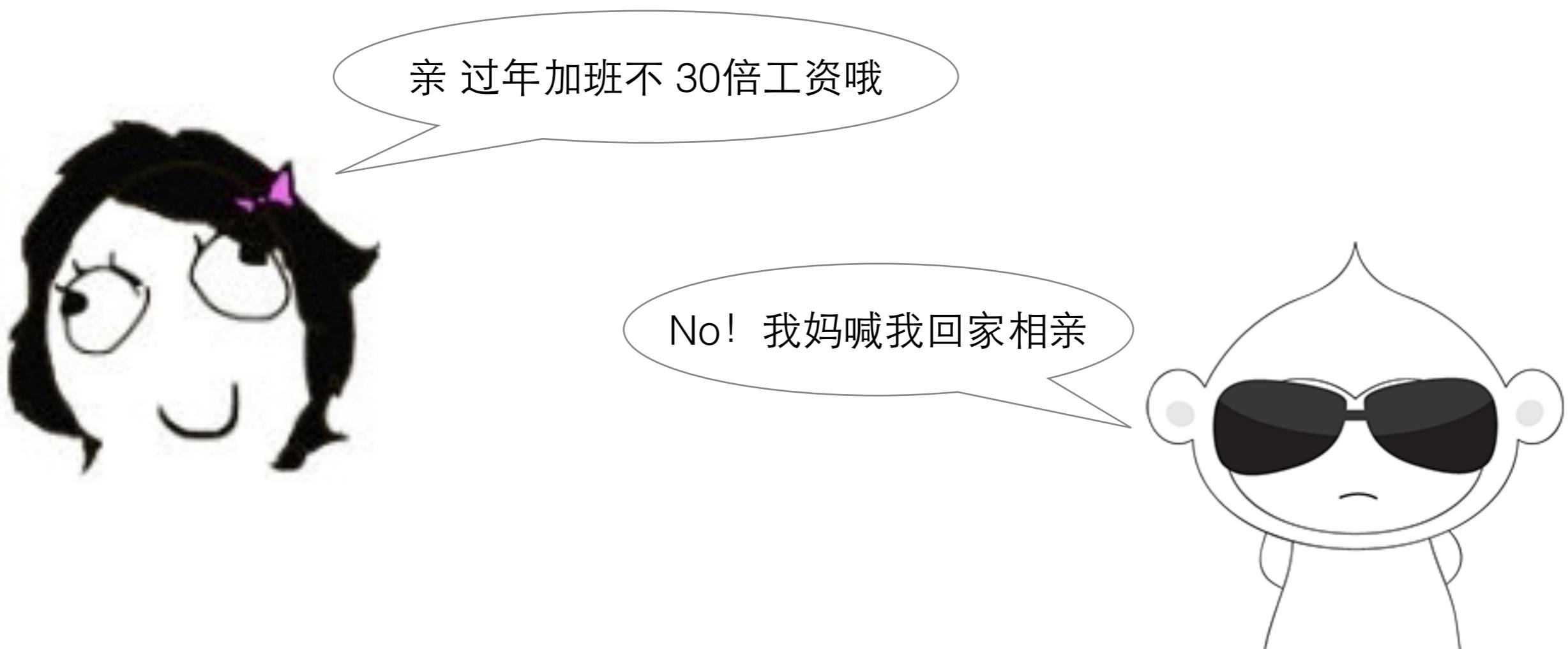
► Hybrid番外篇

Native + HTML5

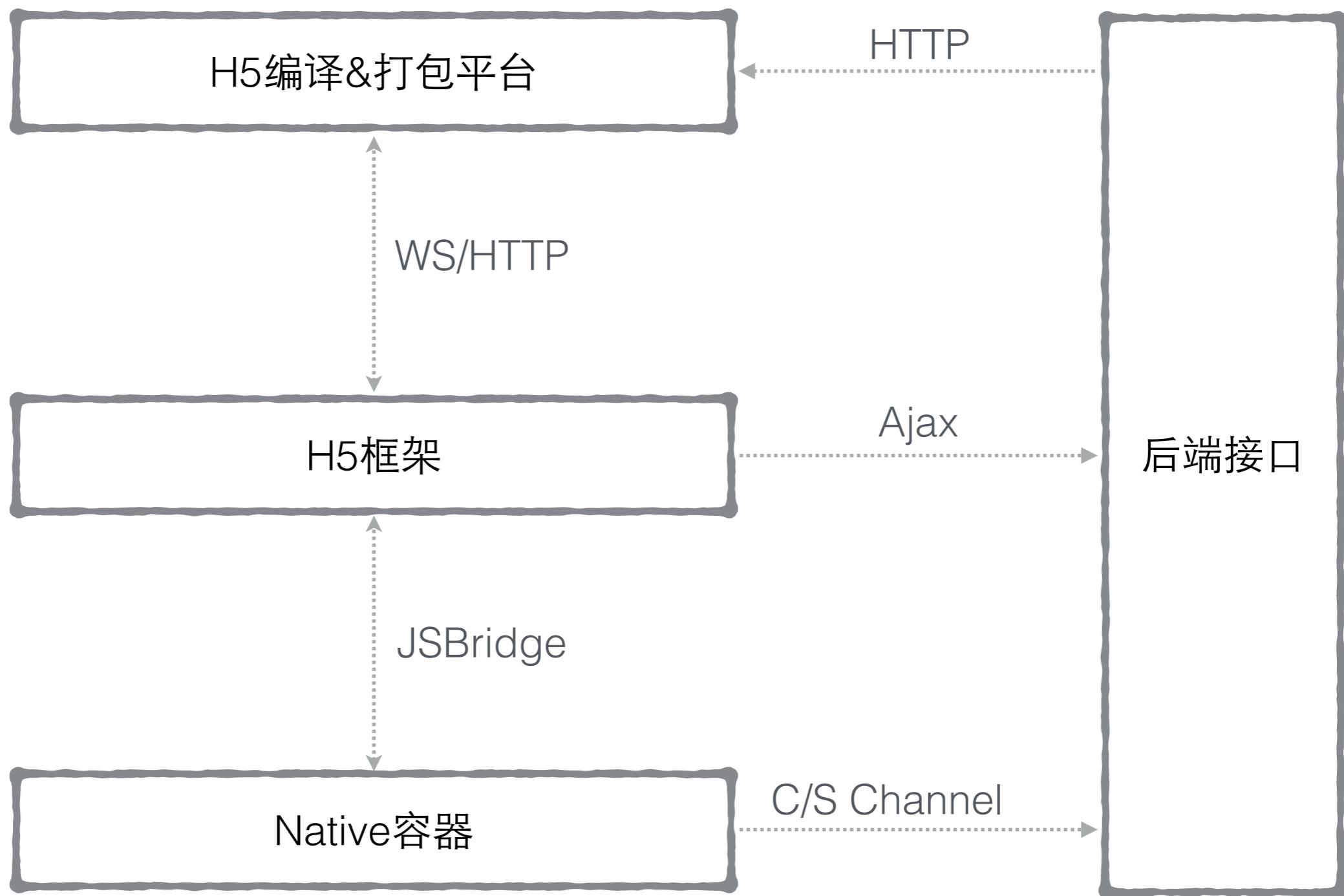


背景

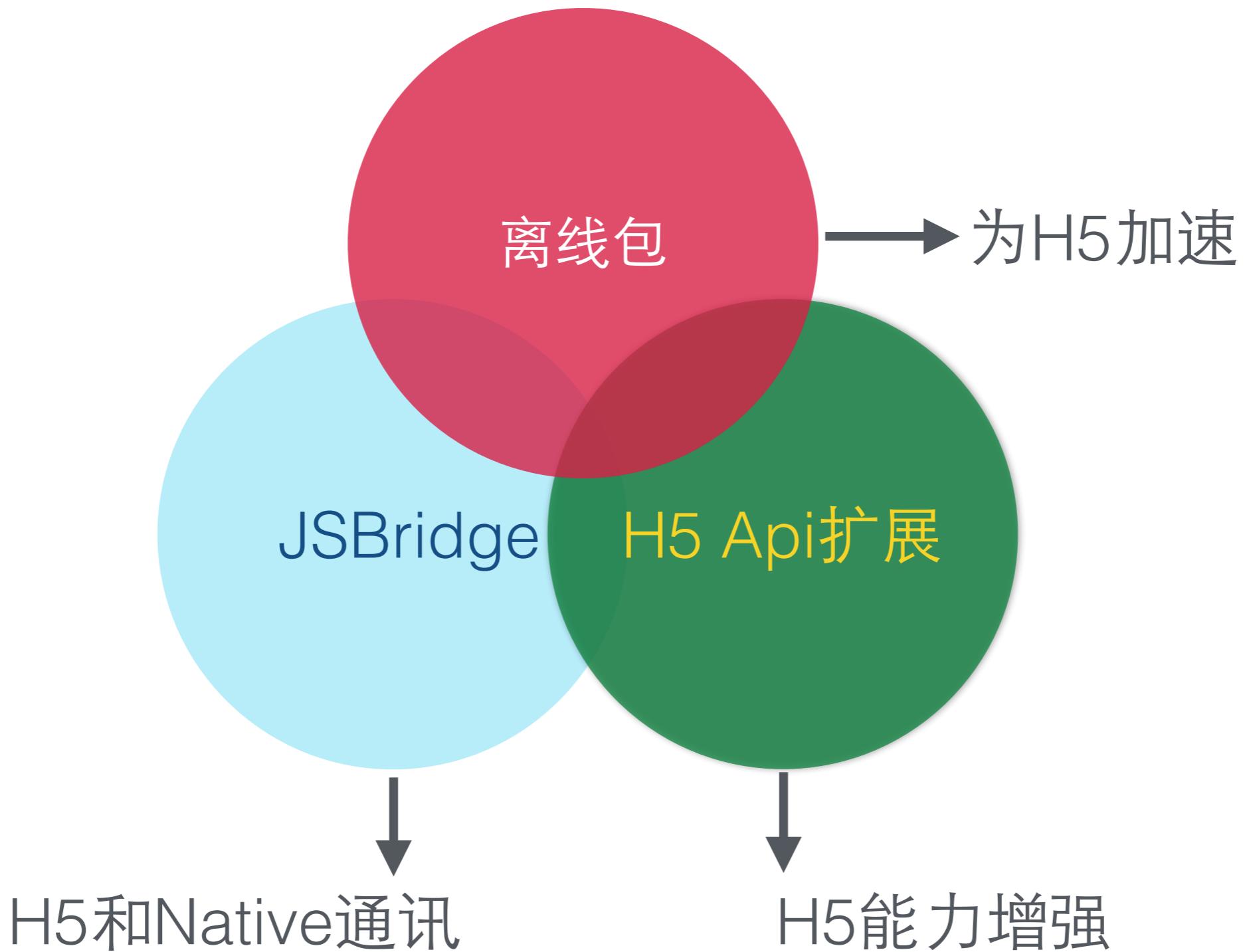
- Android Pad平台需要上App，项目很赶



V1.0 架构



Hybrid设计要点



JSBridge



WebView辅助类

- android.webkit.WebSettings: Web容器的一些设置，如允许js执行、开启LocalStorage、开启web缓存等
- android.webkit.WebViewClient: 定制WebView Page 级别的UI和行为
- android.webkit.WebChromeClient: 定制WebView 组件级别的UI、js事件拦截

JavaScript调用Android

Object Inject	addJavascriptInterface 拦截H5发起的url请求
URL Intercept	拦截prompt弹窗

addJavascriptInterface

```
// Java code  
webView.addJavascriptInterface(JavaObj, "JSObj");
```

Java反射机制漏洞(Android < V4.2)

```
// js code  
function execute(cmd) {  
    return JSObj.getClass().forName("java.lang.Runtime")  
        .getMethod("getRuntime",null).invoke(null,null).exec(cmd);  
}  
execute('some private os command');
```

webview拦截url请求

```
// Java code
webview.setWebViewClient(new WebViewClient() {
    @Override
    public boolean shouldOverrideUrlLoading(WebView w, String url) {
        // JSBRIDGE_SCHEMA即自定义的私有协议
        if(url.startsWith(JSBRIDGE_SCHEMA)) {
            webview.loadUrl("javascript:invokeWeb(" + url + ");");
            return true;
        }
        return false;
    }
});

// js code
iframe.src = 'jsbridge://foo.bar/method?p1=1&p2=2';
```

webview拦截prompt

```
// Java code
webview.setWebChromeClient(new WebChromeClient() {
    @Override
    public boolean onJsPrompt(WebView webview, String url, String message,
        String defaultValue, JsPromptResult result) {
        if(msg.startsWith(JSBRIDGE_SCHEMA)) {
            result.confirm("{\"msg\": \"Native result!\"}");
            return true;
        }
        return super.onJsPrompt(webview, url, message, defaultValue, result);
    }
});

// js code
var result = window.prompt('jsbridge://foo.bar/method?p1=1&p2=2', ''');
```

似乎很简单呀



小明

问题

- prompt是阻塞js进程的
- 异步调用怎么实现



JSBridge的实现

同步

Native: webview.loadUrl("javascript:invokeWeb(result)");

H5: iframe.src="jsbridge://namespace/method?params"

异步

调用方：在同步接口的基础上，带上请求签名（req_sn）

回调方：在返回的结果带上相应签名（res_sn），值和req_sn相同

异步调用示意



jsbridge://x.y/z?req_sn=1&...

jsbridge://a.b/c?req_sn=2&...

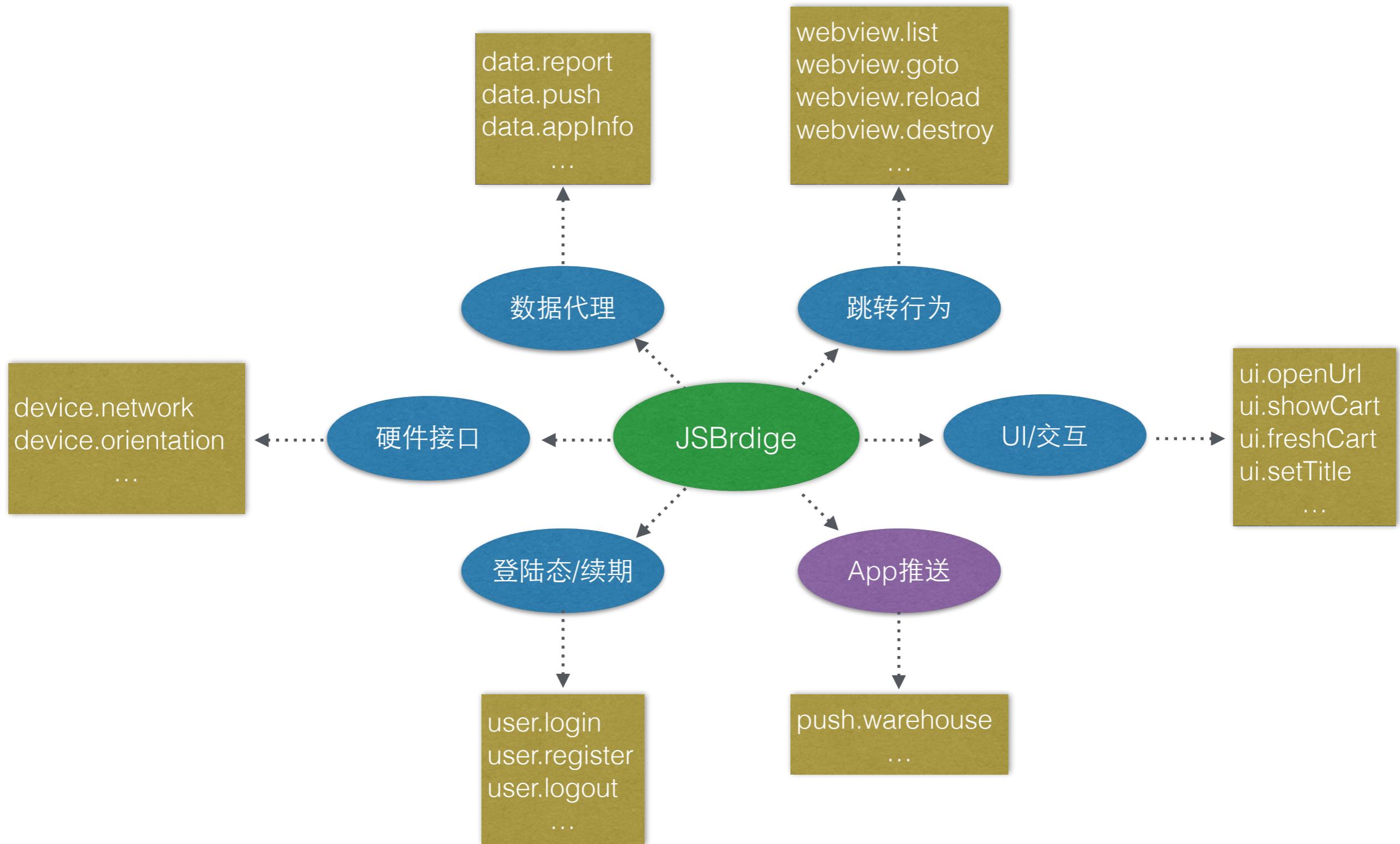
loadUrl("javascript:callbackWeb({'res_sn': 1})");

loadUrl("javascript:callbackWeb({'res_sn': 2})");

loadUrl("javascript:invokeWeb('h5.fn',{ 'req_sn': 3 })");

jsbridge://web.callback?res_sn=3&result=...

扩展H5



webview的自然回退

```
// Java code (test code)
webview.setOnKeyListener(new View.OnKeyListener() {
    @Override
    public boolean onKey(View view, int keyCode, KeyEvent keyEvent) {
        if(keyEvent.getAction() == KeyEvent.ACTION_DOWN) {
            if(keyCode == KeyEvent.KEYCODE_BACK && webview.canGoBack()) {
                // webview自己的history，这种回退不会关闭activity
                webview.goBack();
                return true;
            }
        }
        return false;
    }
});
```



支付结果

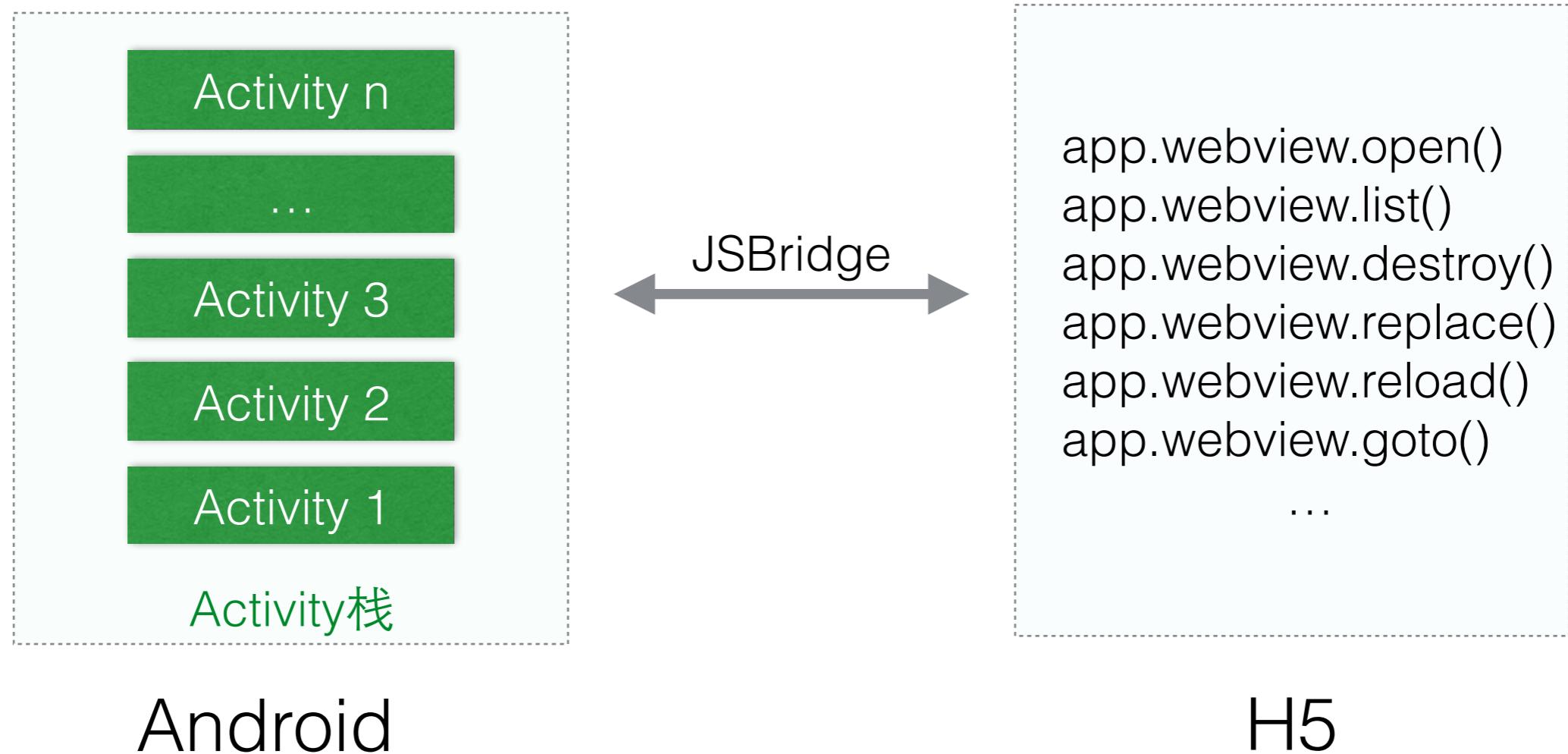


支付失败

订单将为你保留**29:38**

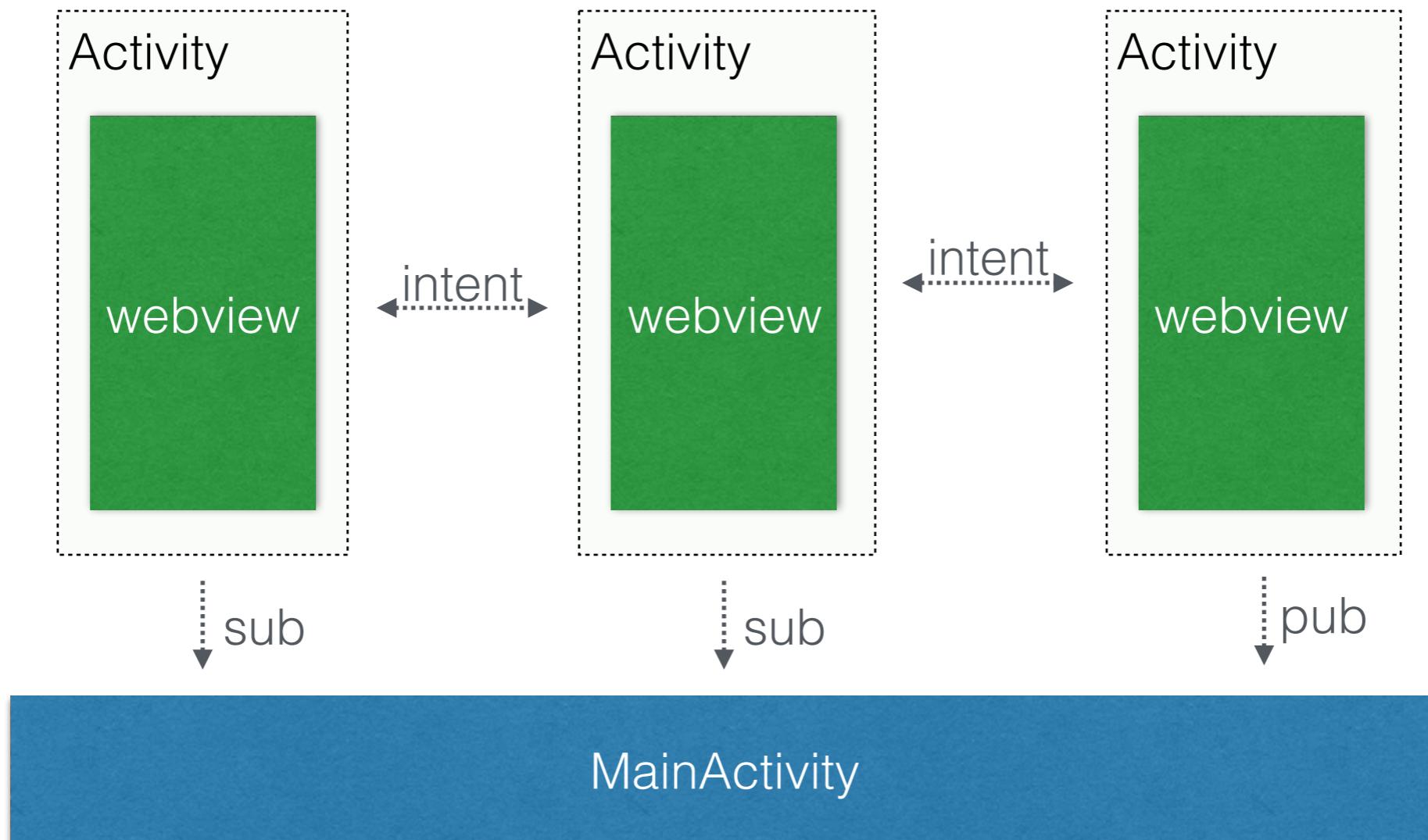
重新支付

定制history



More...

跨webview通讯？



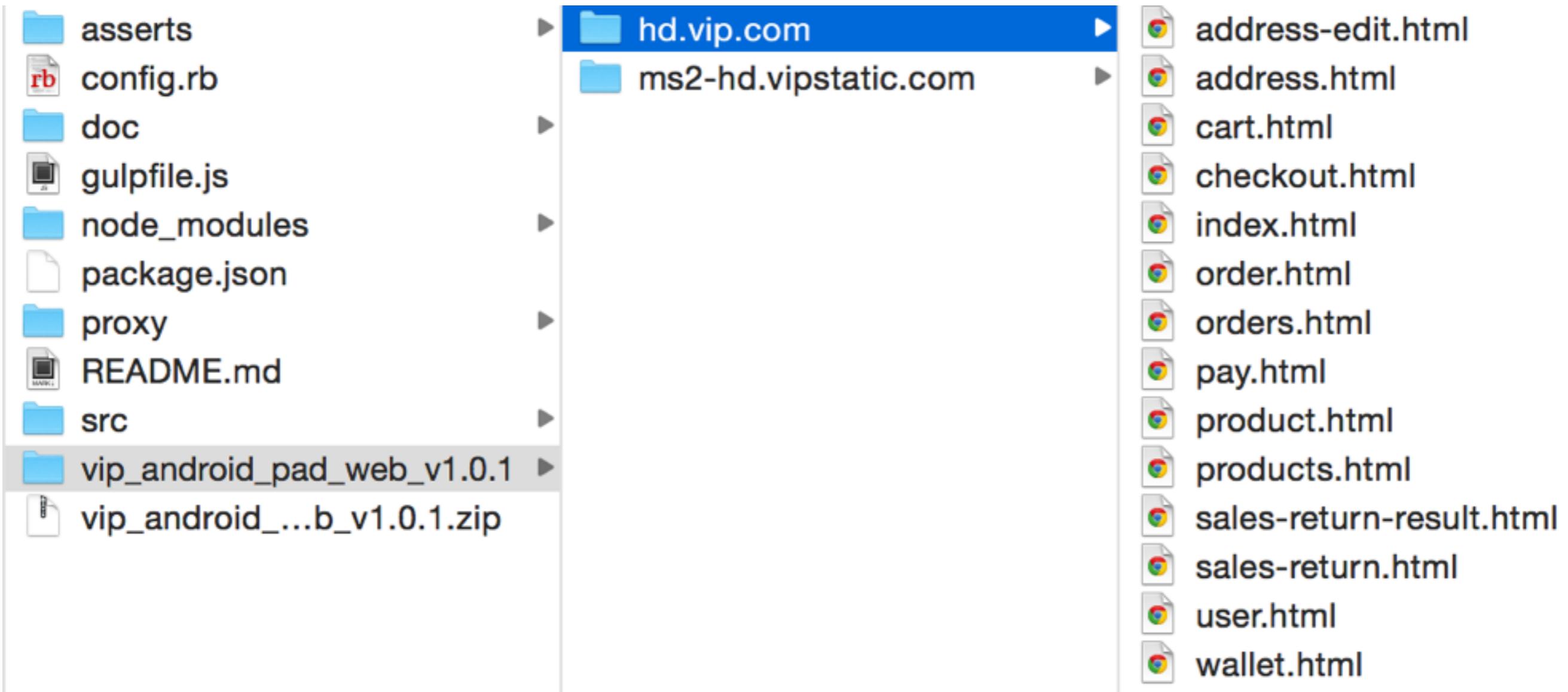
离线方案

- ✓ 把H5运行所需的静态资源“内置”到app
- ✓ 把App当成一个个local server跑在用户侧
- ✓ 按一定规则拦截H5请求，进行再分发

开发上线流程



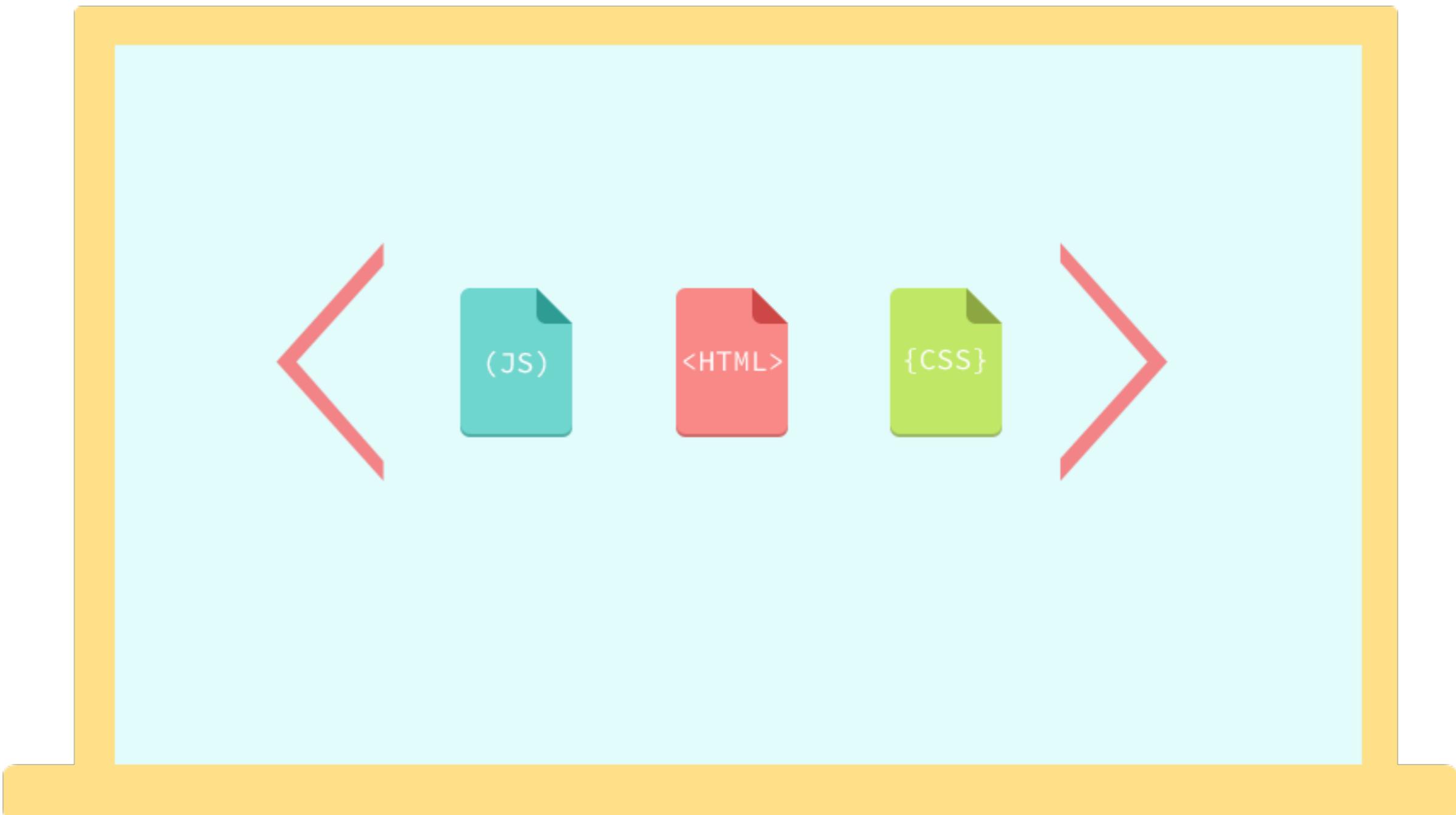
分域名打包



包更新机制

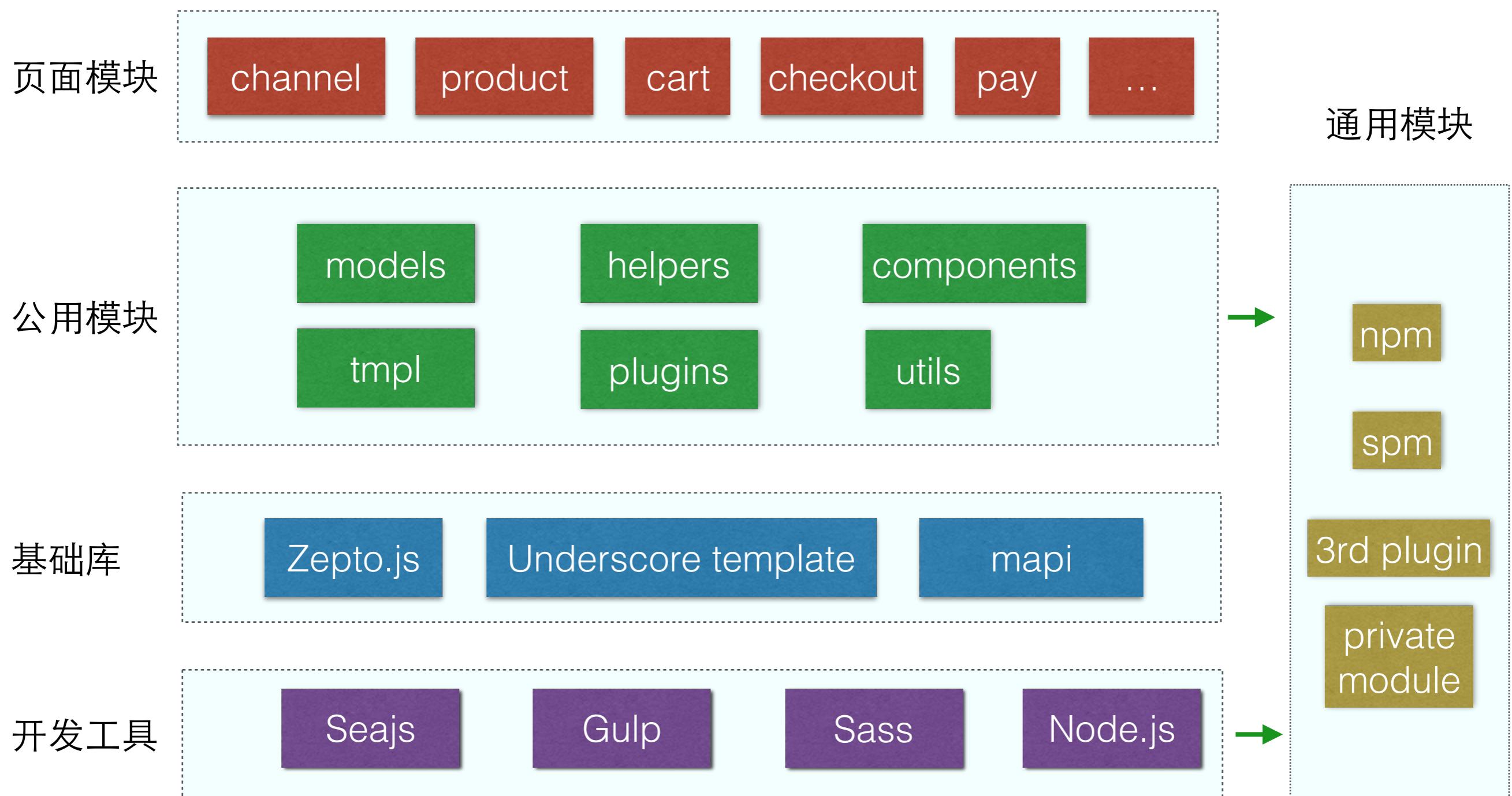
- Server push: 后台压力
- Native poll: App service
- 首开check+跨启动生效

工程化



```
./
|-- README.md                                // 项目说明
|-- asserts                                    // 编译目录
|-- vip_android_pad_web_v1.0.1.zip           // 离线包
|-- config.rb                                  // SASS配置
|-- doc                                         // 文档
|-- gulpfile.js                               // Gulp配置
|-- conf.js                                     // 开发环境配置
|-- package.json                             // Node.js, spm依赖模块
|-- proxy                                       // server proxy router
|   |-- brand_list.json                      // mock数据
|   `-- router.js                            // proxy路由中间件
`-- src                                         // 源码目录&开发目录
    |-- css                                       // SASS生成的css
    |-- html                                      // 没有套模板的静态文件，用于本地调试css
    |-- img                                       // 图片
    |-- js                                         // js模块
    |-- scss                                      // scss工程
    |-- tmpl                                      // 根据后端接口套模板，最终会编译成js
    |-- index.html                                // FE调试文件&页面真正入口
    `-- [others].html                           // 其他page
```

模块分层



js模块化写法

```
'use strict';

var urlUtil = require('./utils/url');
var domUtil = require('./utils/dom');
var strUtil = require('./utils/string');
var dateUtil = require('./utils/date');
var model = require('./models/brand');
var $ajax = require('./helpers/ajax');
var tmpl = require('./tmpl/brand');
var go2top = require('./components/go2top');
var slider = require('./plugins/swipe-slider');
var wvOpen = require('./helpers/open-url');
var report = require('./helpers/report');

require('./utils/underscore-simplify');

// code...
```

前端端共享



开发&mock

```
// task: launch a local http server
gulp.task('server', ['jst'], function(done) {
  var st = require('st');
  var app = express();
  var proxy = require('./proxy/router');
  var mount = st({
    path: path.resolve(__dirname, srcDir),
    index: 'index.html',
    passthrough: true,
    cache: false
  });
  // static files
  app.use(mount);
  // proxy & mock data
  proxy(app);

  app.listen(80, done);
});
```

```
module.exports = function(app) {
  app.get('/api/ad/getAdList', function(req, res, next) {
    res.status = 200;
    res.json({code: 0, data: ad});
  });

  app.get('/api/brand/getList', function(req, res, next) {
    var query = req.query || {};
    var limit = query.limit || 10;
    var brandData = data.brand.list;
    var list = brandData.data.list;
    var i = 0;

    while(list.length < limit) {
      list.push(list[i++ % 2]);
    }

    brandData.data.list = list;
    res.json(brandData);
  });

  app.get('/api', proxy('wapi.vip.vip.com'));
}
```

```
<!-- build:css css/common.min.css -->
<link rel="stylesheet" href="css/common.css" />
<!-- endbuild -->
<link rel="stylesheet" href="css/channel.css" inline />

<!-- build:js js/lib.min.js -->
<script src="js/lib/zepto.js"></script>
<script src="js/lib/underscore-tmpl.js"></script>
<!-- endbuild -->
```

src

```
<!-- build:js js/mapi.min.js -->
<script src="js/lib/mapi.js"></script>
<!-- endbuild -->
|
<!-- build:js js/sea-debug.min.js -->
<script src="js/spm_modules/seajs/2.3.0/dist/sea-debug.js"></script>
<script src="js/spm_modules/seajs-wrap/1.0.2/src/seajs-wrap.js"></script>
<!-- endbuild -->

<script>seajs.config({base: './js'});seajs.use('channel');</script>
```



asserts

```
<link rel="stylesheet" href="//ms2-hd.vipstatic.com/css/common.min.36f39f9a.css">
<style type="text/css">{...}</style>
<script src="//ms2-hd.vipstatic.com/js/lib.min.859a4249.js"></script>
<script src="//ms2-hd.vipstatic.com/js/mapi.min.63f9b0fd.js"></script>
<script src="//ms2-hd.vipstatic.com/js/channel.498b3237.js"></script>
```

zepto.js /js/lib	GET	200 OK
underscore-tmpl.js /js/lib	GET	200 OK
sea-debug.js /js/spm_modules/se...	GET	200 OK
mapi.js /js/lib	GET	200 OK
seajs-wrap.js /js/spm_modules/se...	GET	200 OK
seajs-text.js /js/spm_modules/se...	GET	200 OK
channel.js /js	GET	200 OK
underscore-simplify.js /js/utils	GET	200 OK
url.js /js/utils	GET	200 OK
dom.js /js/utils	GET	200 OK
string.js /js/utils	GET	200 OK
date.js /js/utils	GET	200 OK
brand.js /js/models	GET	200 OK
ajax.js /js/helpers	GET	200 OK
brand.js /js/tmpl	GET	200 OK
go2top.js /js/components	GET	200 OK
swipe-slider.js /js/plugins	GET	200 OK
open-url.js /js/helpers	GET	200 OK
report.js /js/helpers	GET	200 OK
store.js /js/helpers	GET	200 OK

spm build -I src/js -o ../../asserts/js
gulp [rev|cdn]

无缓存

mapi.min.63f9b0fd.js ms2-hd.vipstatic.com/js
lib.min.859a4249.js ms2-hd.vipstatic.com/js
channel.2064ddf1.js ms2-hd.vipstatic.com/js

永久缓存

模板收敛



局部刷新

```
# dmyang at dmyangdeMacBook-Pro.local in ~/code/work/wap/trunk/view-src/touch
$ ls -l | wc -l
142
```

控制模板数量

```
# dmyang at dmyangdeMacBook-Pro.local in ~/code/work/waphd/src
$ ls -l | grep .html | wc -l
17
```

关于Hybrid

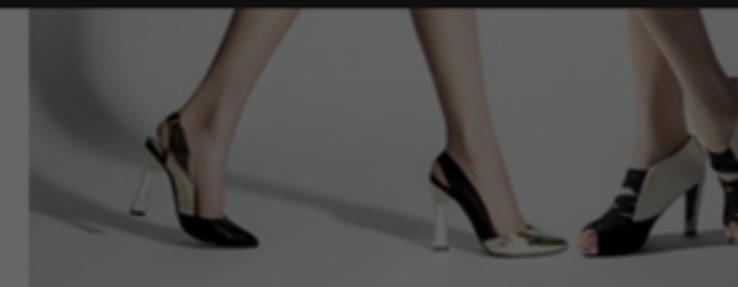
- 本质上还是web，无法完全绕开web存在的缺陷
 - 列表滚动性能、内存泄露、API兼容性、响应式布局
- 技术挑战
 - Native性能优化、H5 debug、同时拥有多端能力的人才太少
- 如何分工
 - Native专注底层架构、关键页面和数据，H5专注UI和内容



最新上线

限时上新

最后疯抢



2.9折起 UNITED NUDE女鞋专场



1.1折起 德菲蒂奥D.F.DEAR女装专场



选择收货地区



当前收货地区

广东省



A

安徽省

B

北京市

F

福建省

G

甘肃省

广东省

广西壮族自治区

贵州省

H

海南省

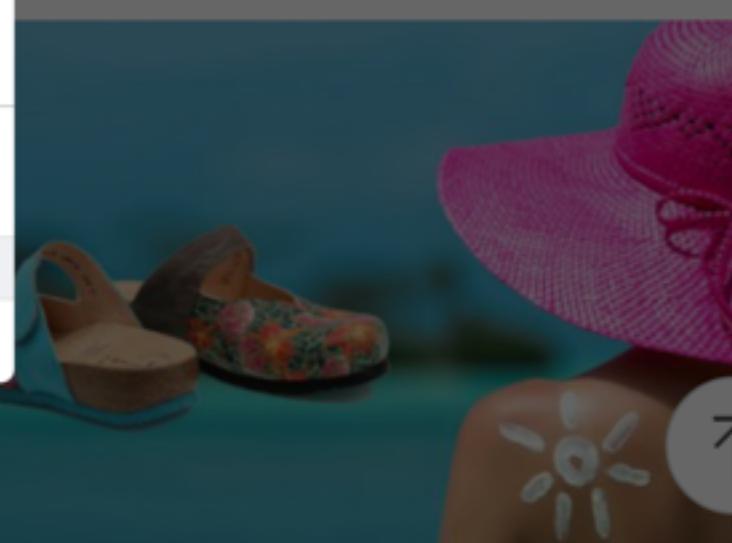


剩2天



男装专场

剩2天



Hybrid技术栈

- PhoneGap、AppCan、Ionic
- React-Native

<https://github.com/tmallfe/tmallfe.github.io/issues/18>

<http://blog.cnbang.net/tech/2698/>

<http://www.stuq.org/ppt/show/95/5c28db3ecf661208387127be8878aed3#/2>

计划

- Android Pad V1.0
- App手机版部分页面H5化
- 内部项目：React-Native

谢谢！

特别鸣谢：coding、暴走漫画

Q&A
