

# 极致的 Hybrid

——从阿里旅行「去啊」的架构演变看 Web 与 Native 的融合

@拔赤 @坦丁

阿里旅行·去啊

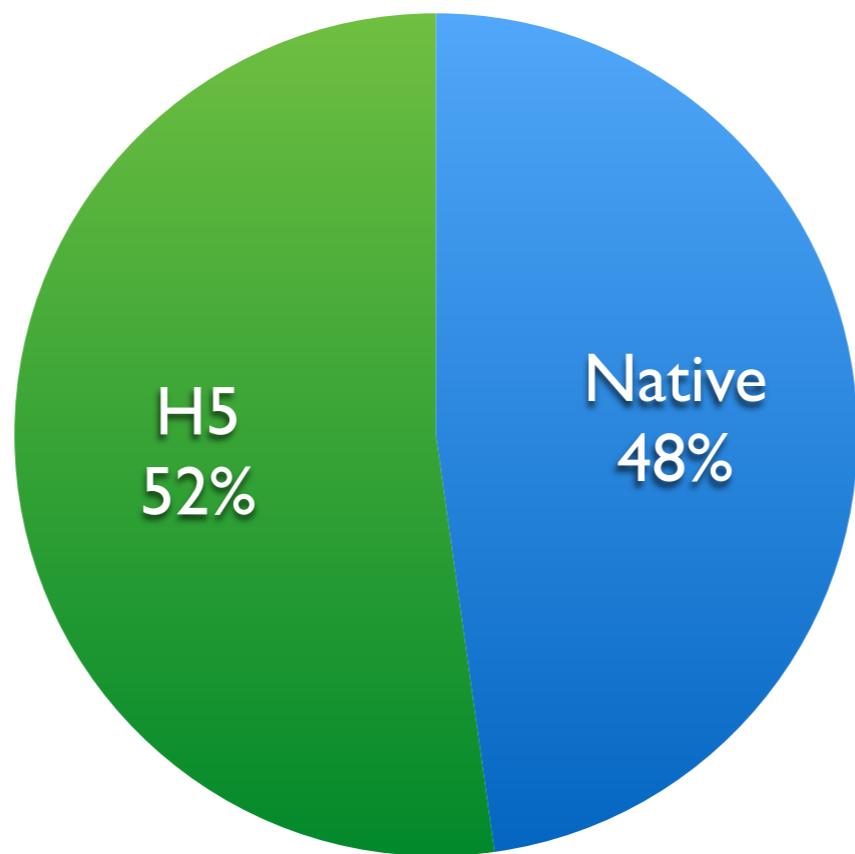
# 阿里旅行·去啊



阿里旅行 · 去啊 App  
是一款小而美的旅行类客户端

# Native & Web 强混

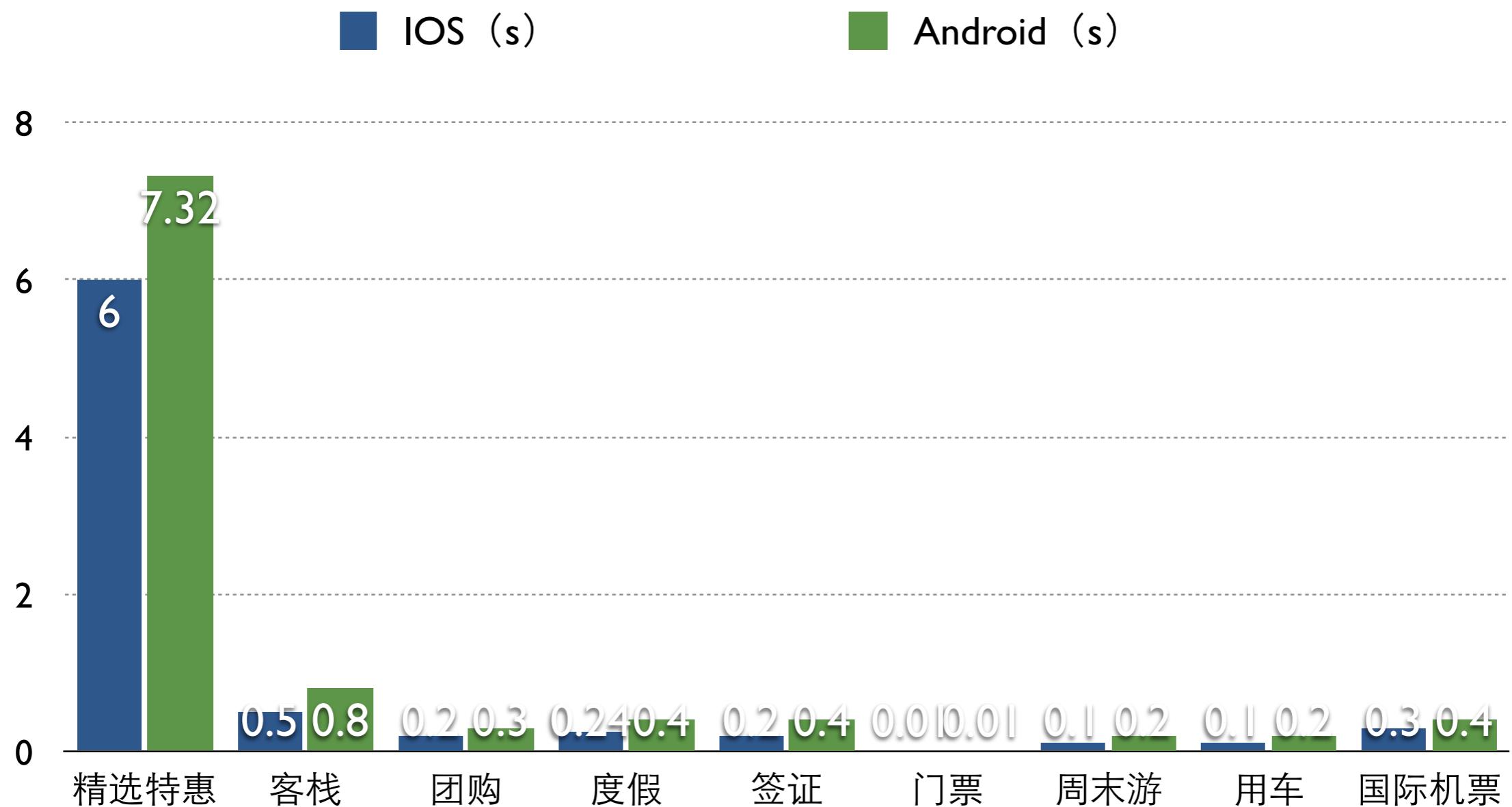
H5 和 Native 页面数量比例



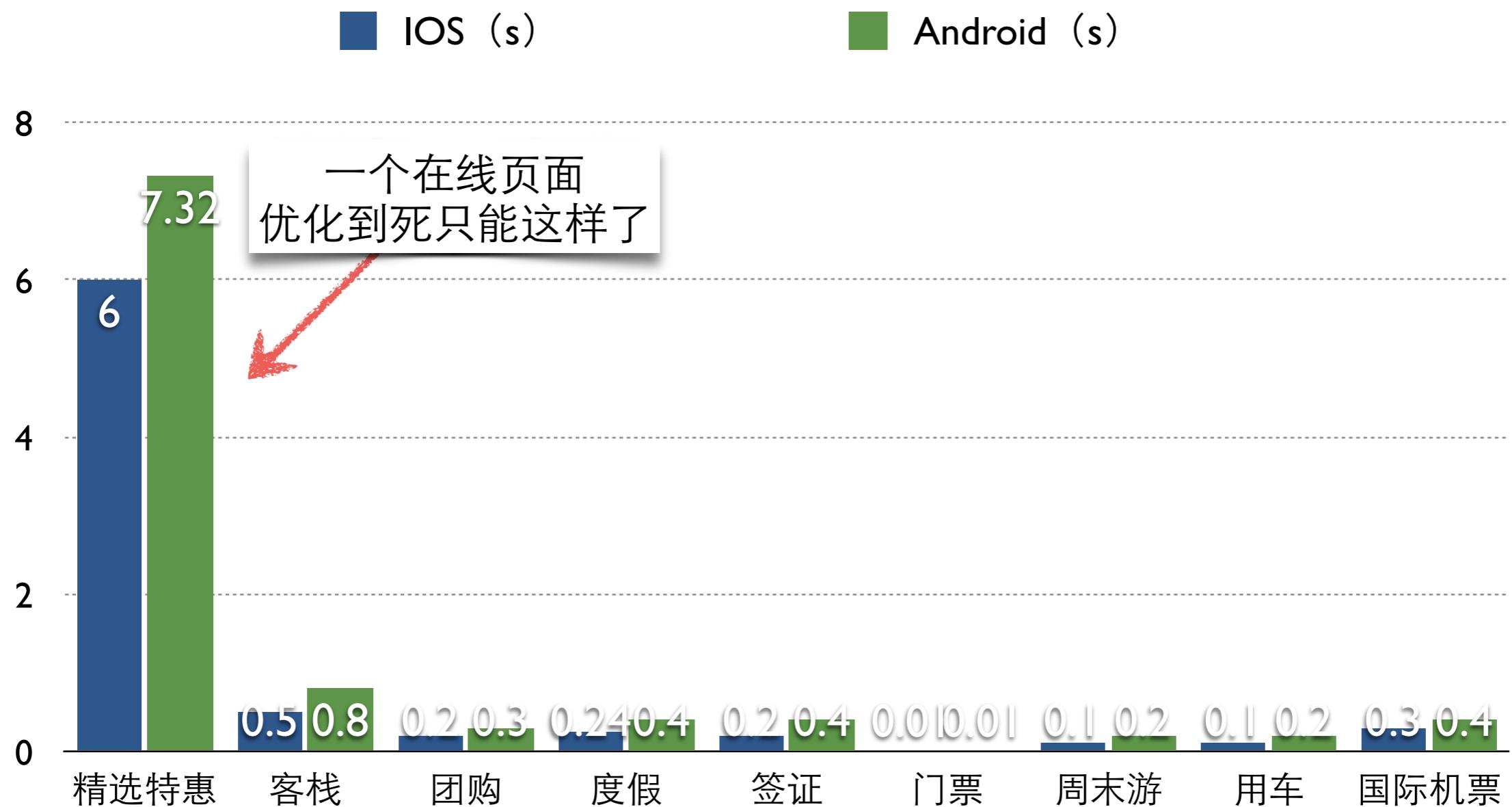
● Native

● H5

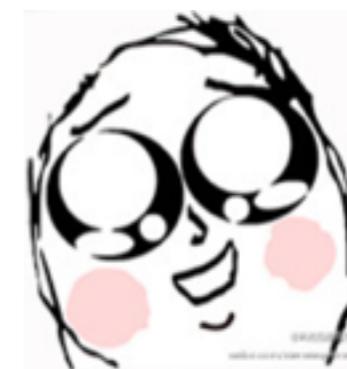
# Native & Web 强混



# Native & Web 强混



“So， 我们把 HTML/CSS/JS 预装到 App 中！ ”



Thank you.

Questions ?



# Questions?

JS/CSS/IMG 资源怎么动态更新?

什么时候更新资源?

活动页面里的动态数据怎么和HTML解耦?

虚拟域的实现以及在线离线页面怎么切换?

埋点和数据统计怎么做?

H5 与 Native 之间的互调如何实现?

WebView 的 Crash 率的难题怎么解?

Native 中 Page 资源的定位怎么规范?

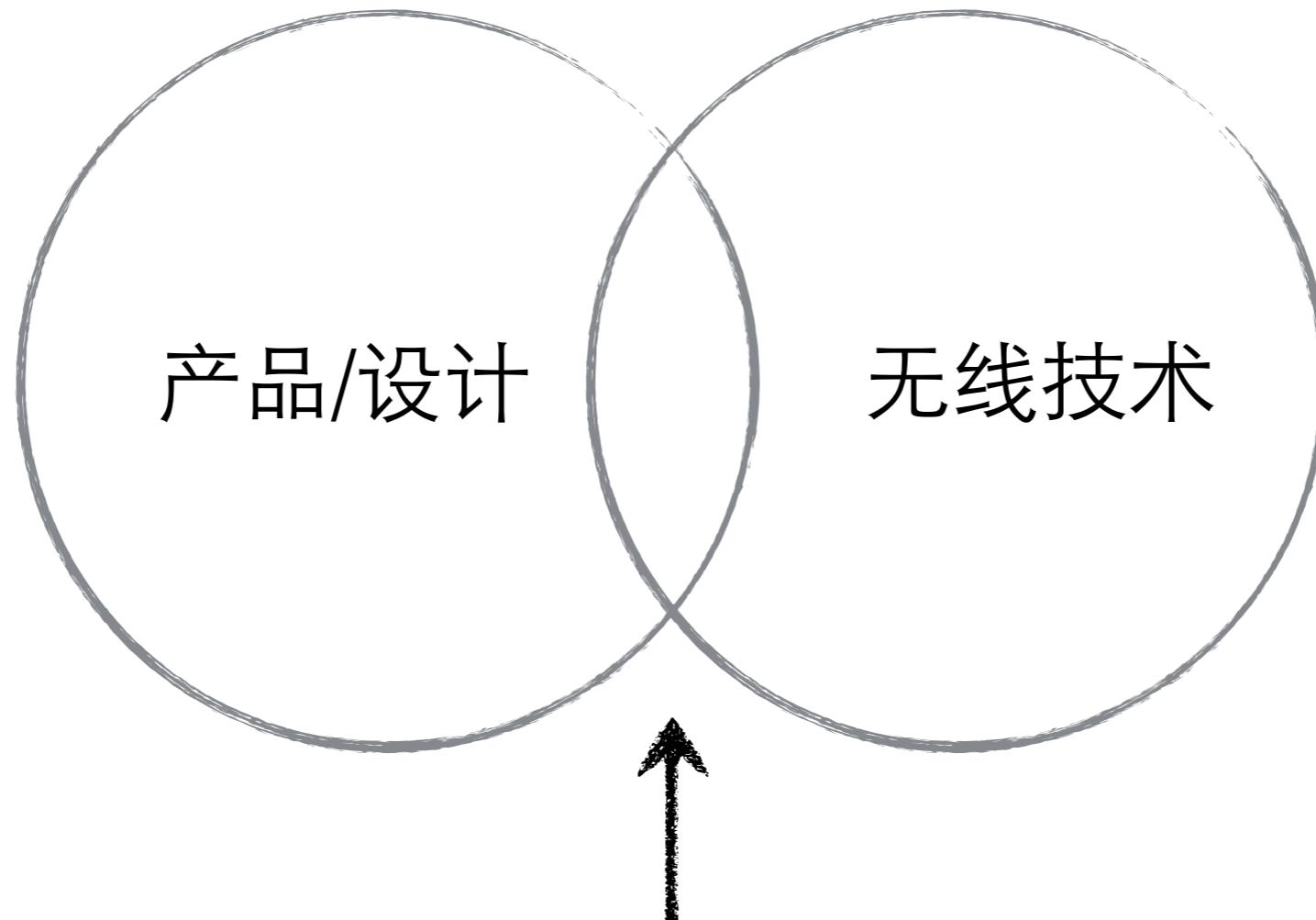
HTML 渲染如何加速?

...

# 原始诉求

- 用户：“快、快、快”——界面快点出来
- PD：“快、快、快”——功能快点上线
- 老板：“快、快、快”——快点回滚问题代码

# Hybrid



Web 的 Native 化 vs Native 的 Web 化

# Why Not Native?

1. App 发版周期远远跟不上产品节奏! (Native 的硬伤)
2. 灵活性差，模块化不起来，很难做千人千面 (也是Native的硬伤)
3. 外投引流只能靠H5 (还是Native的硬伤)
4. App 之间的互联规范必须通过 H5 的桥接 (更是Native的硬伤)

# Why not Web?

1. 弱网体验极差 (天朝的网络你懂得)
2. Wap 上的交易转化率奇低 (本质上是体验差)

# Hybrid

- 融合 · 灵活的 H5 vs 可靠的 Native  
桥、H5 容器
- 极速 · H5 的秒出  
离线包体系
- 闭环 · 交互/交易流程的多端闭环  
购票→Push/短信提醒→行程卡片→自动值机→专车服务→累积淘里程...
- 解耦 · 端与端、M 和 V  
Router 链接多端、公用代码与业务代码、数据与视图、无序的需求与有序的研发，快速迭代...

# 摆脱思维定势

- **HTML 4 到 HTML 5**
  - 基于 <a> 标签的资源互联无法自然延伸到移动终端领域
  - 专注打造 BOM 的超集： Hybrid Bridge
- **HTTP 1.1 到 HTTP 2.0<sub>(升级spdy?)</sub>**
  - 应用层协议+短链接：是否继续适用于移动互联网？
- **从 Browser 到 App**
  - App 之间天然存在技术壁垒，阻碍信息互通
  - URL 不见了，移动Web里用什么来描述“资源”

# Questions?

JS/CSS/IMG 资源怎么动态更新?

什么时候更新资源?

活动页面里的动态数据怎么和HTML解耦?

虚拟域的实现以及在线离线页面怎么切换?

埋点和数据统计怎么做?

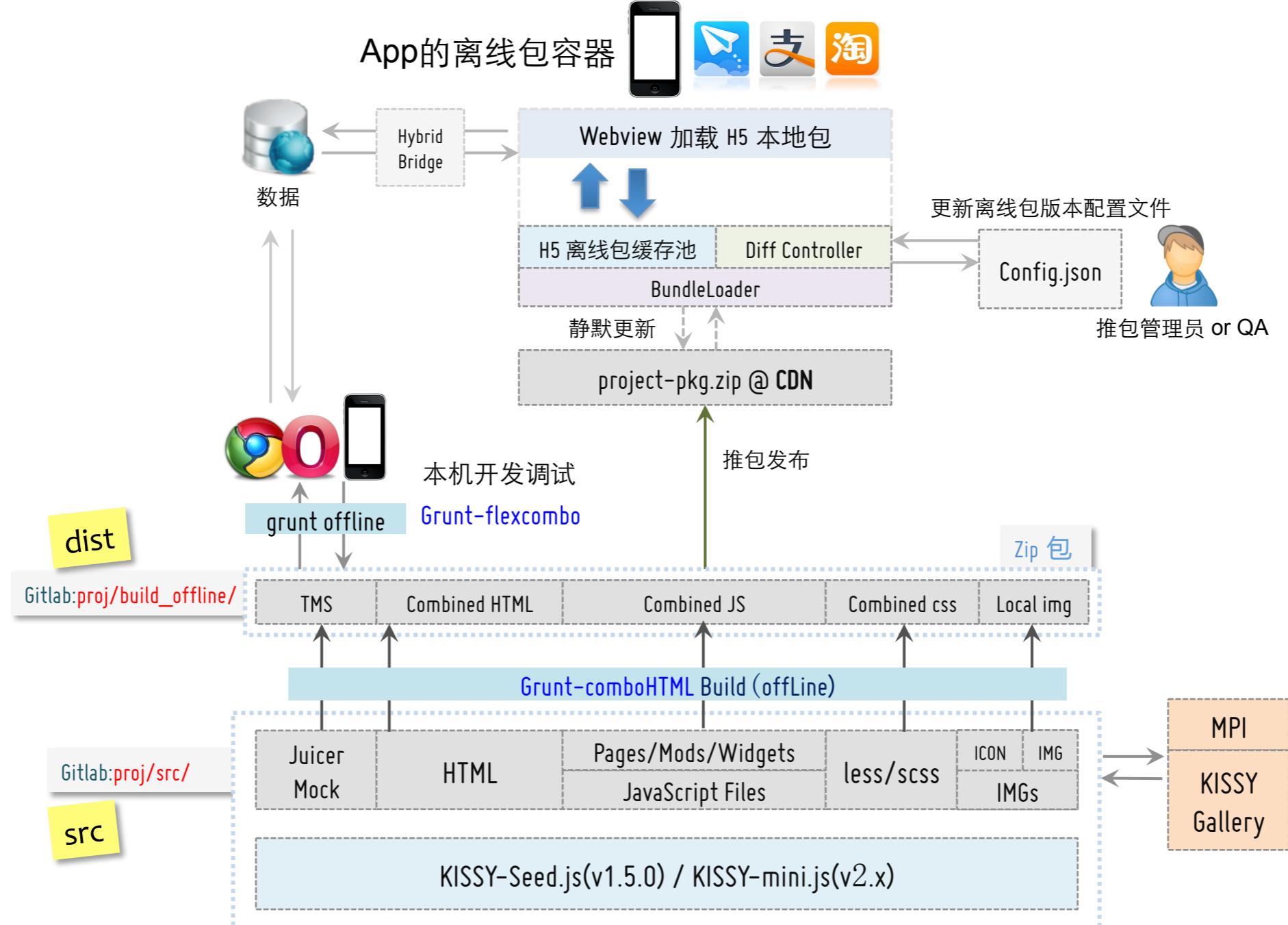
H5 与 Native 之间的互调如何实现?

WebView 的 Crash 的难题怎么解?

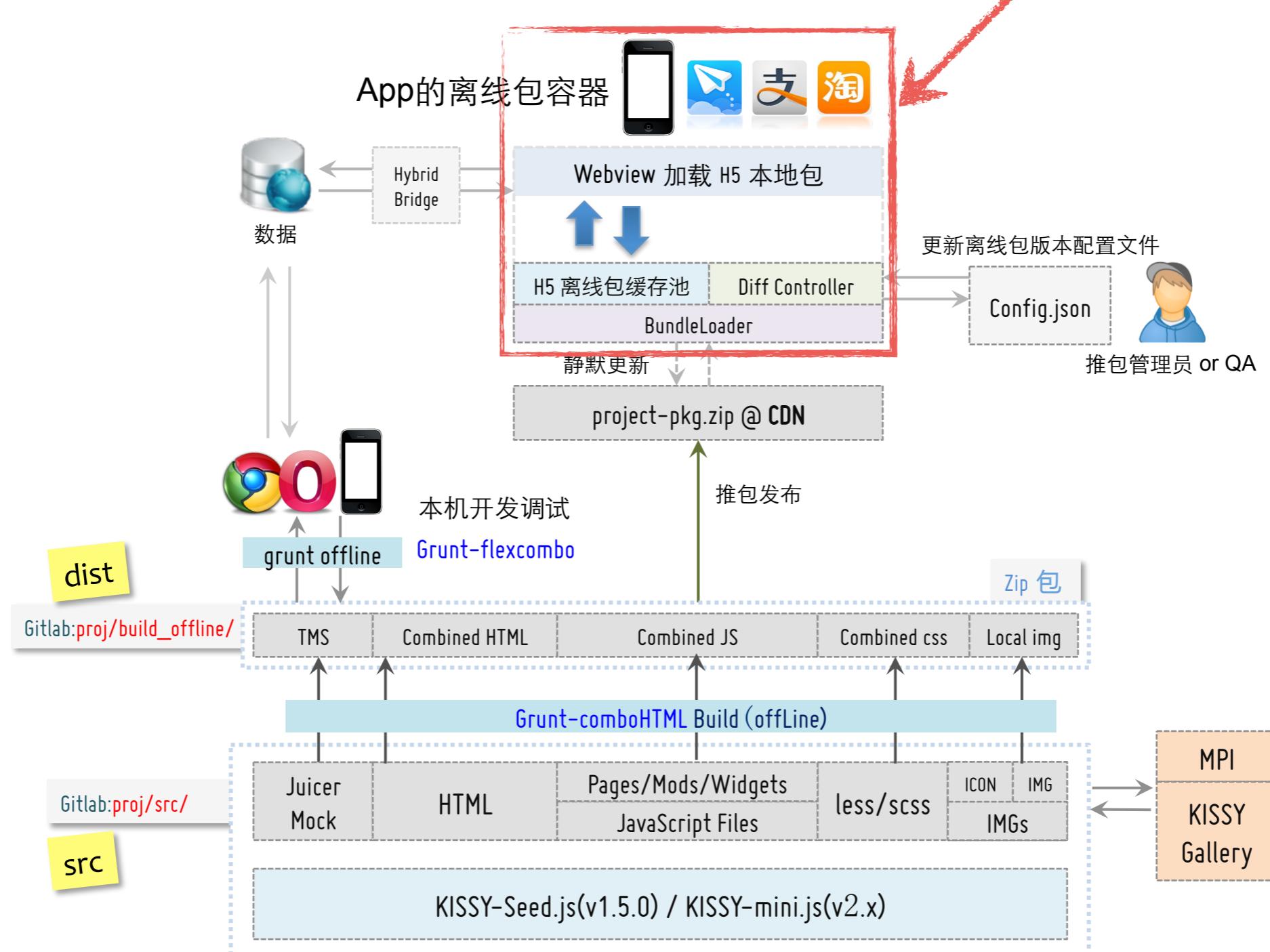
Native 中 Page 资源的定位怎么规范?

**HTML** 渲染如何加速?

...

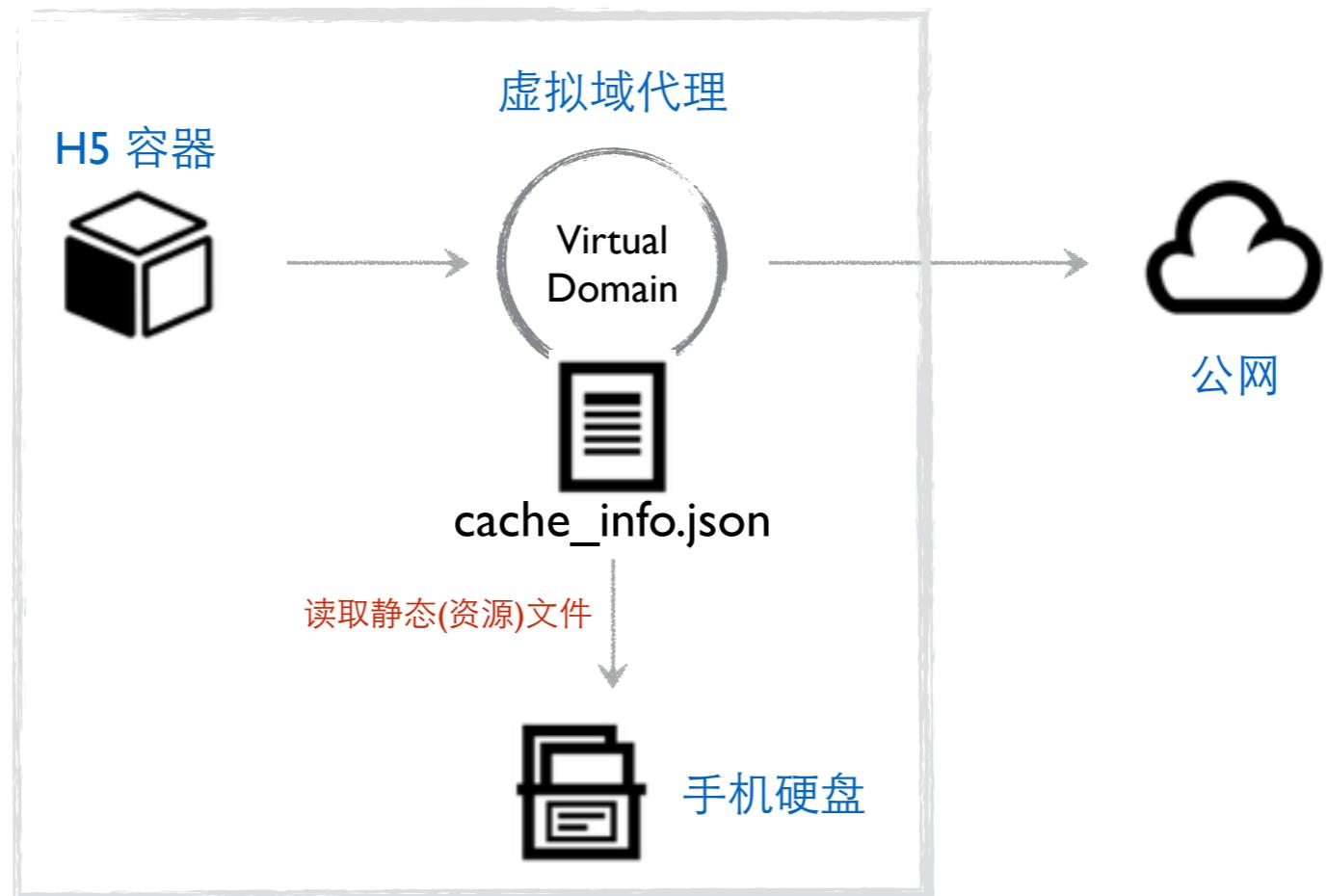


## H5 适配多端(的容器)



航旅 H5 离线环境体系

App 离线容器边界



Native 端取资源的提速：去啊 App 加载离线资源逻辑

Safari:在线页面



去啊App:离线页面



<http://h5.m.taobao.com/trip/tuan/list/index.html>

虚拟域：同样 URL 在容器中打开会走到离线页面

# Questions?

JS/CSS/IMG 资源怎么动态更新?

什么时候更新资源?

活动页面里的动态数据怎么和**HTML**解耦?

在线离线页面怎么切换?

埋点和数据统计怎么做?

H5 与 Native 之间的互调如何实现?

WebView 的 Crush 率的难题怎么解?

Native 中 Page 资源的定位怎么规范?

HTML 渲染如何加速?

...



动态区域 (TMS) 中的 M 和 V 的解耦



内嵌模块  
运营同学随时更新

异步第三方区块

动态区域 (TMS) 中的 M 和 V 的解耦

```
handle_tms_fragment(function() {/*
  <section class="tms">
    <style>
      #banner {
        width:100%;
      }
    </style>
    <div>
      
    </div>
  </section>
}, 1);
```

通过注释来异步加载一段 HTML 代码

```
<!-- #include file="extra.php" -->
```

```
<!-- #include file="extra.php" -->
```



替换成抓取函数

```
<script id="tms_fragment_1">  
get_tms_fragment("http://proxy.php?src=extra.php", "gbk", "1");  
</script>
```

```
<!-- #include file="extra.php" -->
```



```
<script id="tms_fragment_1">  
get_tms_fragment("http://proxy.php?src=extra.php", "gbk", "1");  
</script>
```



异步抓取 TMS 正文

```
handle_tms_fragment(function() {  
    /*<div>TMS内容</div>*/  
}, 1);
```

```
<!-- #include file="extra.php" -->
```



```
<script id="tms_fragment_1">  
get_tms_fragment("http://proxy.php?src=extra.php", "gbk", "1");  
</script>
```



```
handle_tms_fragment(function() {  
    /*<div>TMS内容</div>*/  
}, 1);
```



将正文塞入页面

```
function handle_tms_fragment(fn, id) {  
    var content = getFromComment(fn.toString()); //<div>TMS内容</div>  
    $('#tms_fragment_' + id).replaceWith(content);  
}
```

src



clean

qa-seed-tmsparser.js

qa-seed.js

less

sass

cssmin

grunt build

kmb

uglify

replace

combohtml

domman

replace

cacheinfo

format zip

dist



src



clean

qa-seed-tmsparser.js

qa-seed.js

gru

```
h5-home — grunt build — grunt — node — 79x15
File build/widgets/autocomplete/demo/index-min.css created.
File build/widgets/autocomplete/index-min.css created.
File build/widgets/libs/notification-min.css created.
File build/widgets/libs/reset-min.css created.
File build/widgets/libs/transition-min.css created.
File build/widgets/notification/index-min.css created.
File build/widgets/scroll-list/index-min.css created.
File build/widgets/search-input/index-min.css created.
File build/widgets/search-suggest/index-min.css created.

Running "prompt:main" (prompt) task
? 需要构建到哪些环境的离线包 (Press <space> to select)
❯○去啊
●手机淘宝
●支付宝钱包
```

cacheinfo      format zip

dist



Clam 构建工具对这些逻辑的封装

src



clean

cache\_info.json — build\_offline UNREGISTERED

cache\_info.json

```
1 {  
2   "Vtrip\flight\delayinsure\index.html": "pages\delayinsure\index.html",  
3   "Vtrip\pages\delayinsure\index.html": "pages\delayinsure\index.html",  
4   "Vtrip\flight\delivery\area.html": "pages\delivery\area.html",  
5   "Vtrip\pages\delivery\area.html": "pages\delivery\area.html",  
6   "Vtrip\flight\delivery\detail.html": "pages\delivery\detail.html",  
7   "Vtrip\pages\delivery\detail.html": "pages\delivery\detail.html",  
8   "Vtrip\flight\delivery\list.html": "pages\delivery\list.html",  
9   "Vtrip\pages\delivery\list.html": "pages\delivery\list.html",  
10  "Vtrip\flight\delivery\post.html": "pages\delivery\post.html",  
11  "Vtrip\pages\delivery\post.html": "pages\delivery\post.html",  
12  "Vtrip\flight\myorder\detail.html": "pages\myorder\detail.html",  
13  "Vtrip\pages\myorder\detail.html": "pages\myorder\detail.html",  
14  "Vtrip\flight\myorder\list.html": "pages\myorder\list.html",  
15  "Vtrip\pages\myorder\list.html": "pages\myorder\list.html",  
16  "Vtrip\flight\onsale\index.html": "pages\onsale\index.html",  
17  "Vtrip\pages\onsale\index.html": "pages\onsale\index.html",  
18  "Vtrip\flight\onsale\more.html": "pages\onsale\more.html",  
19  "Vtrip\pages\onsale\more.html": "pages\onsale\more.html",  
20  "Vtrip\flight\order\activity.html": "pages\order\activity.html",  
21  "Vtrip\pages\order\activity.html": "pages\order\activity.html",  
22  "Vtrip\flight\order\apply-success.html": "pages\order\apply-success.html",  
23  "Vtrip\pages\order\apply-success.html": "pages\order\apply-success.html",  
24  "Vtrip\flight\order\confirm.html": "pages\order\confirm.html",  
25  "Vtrip\pages\order\confirm.html": "pages\order\confirm.html",  
26  "Vtrip\flight\order\insure.html": "pages\order\insure.html",  
27  "Vtrip\pages\order\insure.html": "pages\order\insure.html",  
28  "Vtrip\flight\otalist\index.html": "pages\otalist\index.html",  
29  "Vtrip\pages\otalist\index.html": "pages\otalist\index.html",  
30  "Vtrip\flight\passenger\cert.html": "pages\passenger\cert.html",  
31  "Vtrip\pages\passenger\cert.html": "pages\passenger\cert.html",  
32  "Vtrip\flight\passenger\detail.html": "pages\passenger\detail.html",  
33  "Vtrip\pages\passenger\detail.html": "pages\passenger\detail.html",  
34  "Vtrip\flight\passenger\list.html": "pages\passenger\list.html",  
35  "Vtrip\pages\passenger\list.html": "pages\passenger\list.html",  
36  "Vtrip\flight\refund\index.html": "pages\refund\index.html",  
37  "Vtrip\pages\refund\index.html": "pages\refund\index.html"}  
Line 6, Column 13 Spaces: 2 JSON
```

dist



Cacheinfo.json 本地资源映射表

# 去啊前端基础工具

Clam: <http://clam.alitrip.net>



# Questions?

**JS/CSS/IMG** 资源怎么动态更新?

什么时候更新资源?

活动页面里的动态数据怎么和HTML解耦?

在线离线页面怎么切换?

埋点和数据统计怎么做?

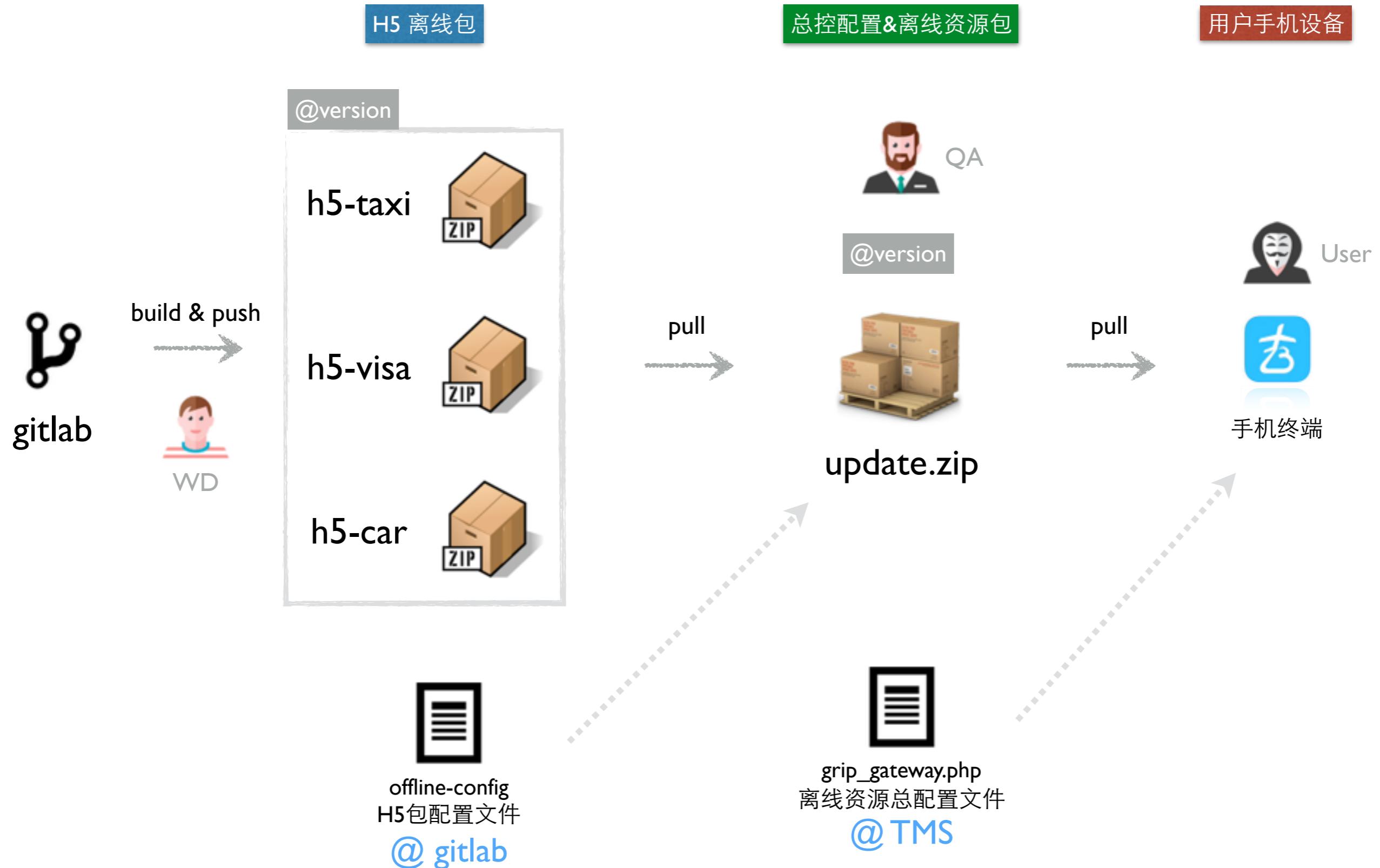
H5 与 Native 之间的互调如何实现?

WebView 的 Crash 率的难题怎么解?

Native 中 Page 资源的定位怎么规范?

HTML 渲染如何加速?

...





去啊 H5 离线包发布流程

# Questions?

JS/CSS/IMG 资源怎么动态更新?

什么时候更新资源?

活动页面里的动态数据怎么和HTML解耦?

在线离线页面怎么切换?

埋点和数据统计怎么做?

H5 与 Native 之间的互调如何实现?

WebView 的 Crash 率的难题怎么解?

Native 中 Page 资源的定位怎么规范?

HTML 渲染如何加速?

...

Native



H5



H5



Native



Native



# 构建 App 内的 Hybrid PageFlow 的传参规范

Native



H5



唤起



Native



Native



构建 App 之间的互调和传参规范

The screenshot shows a log analysis interface with two main sections. At the top, a detailed log table displays various HTTP requests with columns for RC, Mthd, Host, and Path. Several entries from 'usertrack.test.taobao.net' and 'log.m.taobao.com' are highlighted with a red box. Below this is a summary section titled '最近访问的报表' (Recently Visited Reports) which includes links like '今日实时' (Real-time Today), '基本统计' (Basic Statistics), and '用户行为' (User Behavior). The '用户行为' section is expanded, showing sub-links such as '启动次数' (Number of Starts), '使用时长' (Usage Duration), '页面路径分析' (Page Path Analysis), '关键漏斗' (Key Funnel), '控件点击' (Control Clicks), and '自定义事件' (Custom Events). A sub-section under '控件点击' is titled '控件点击统计' (Control Click Statistics) with instructions: '1. 控件点击统计某个页面上的控件被点击次数及点击控件的用户数。能够帮助你优化功能控件的位置和设计。' and '2. 提供页面中文备注名编辑功能，有中文备注名的页面，在报表中将会呈现更易读的中文名称，请通知管理员点击编辑页面及控件名称。'. The main summary table lists pages with their visit counts, active users, and active user percentages, with an '展开' (Expand) link for each row.

# 1. 日志表的双份存储

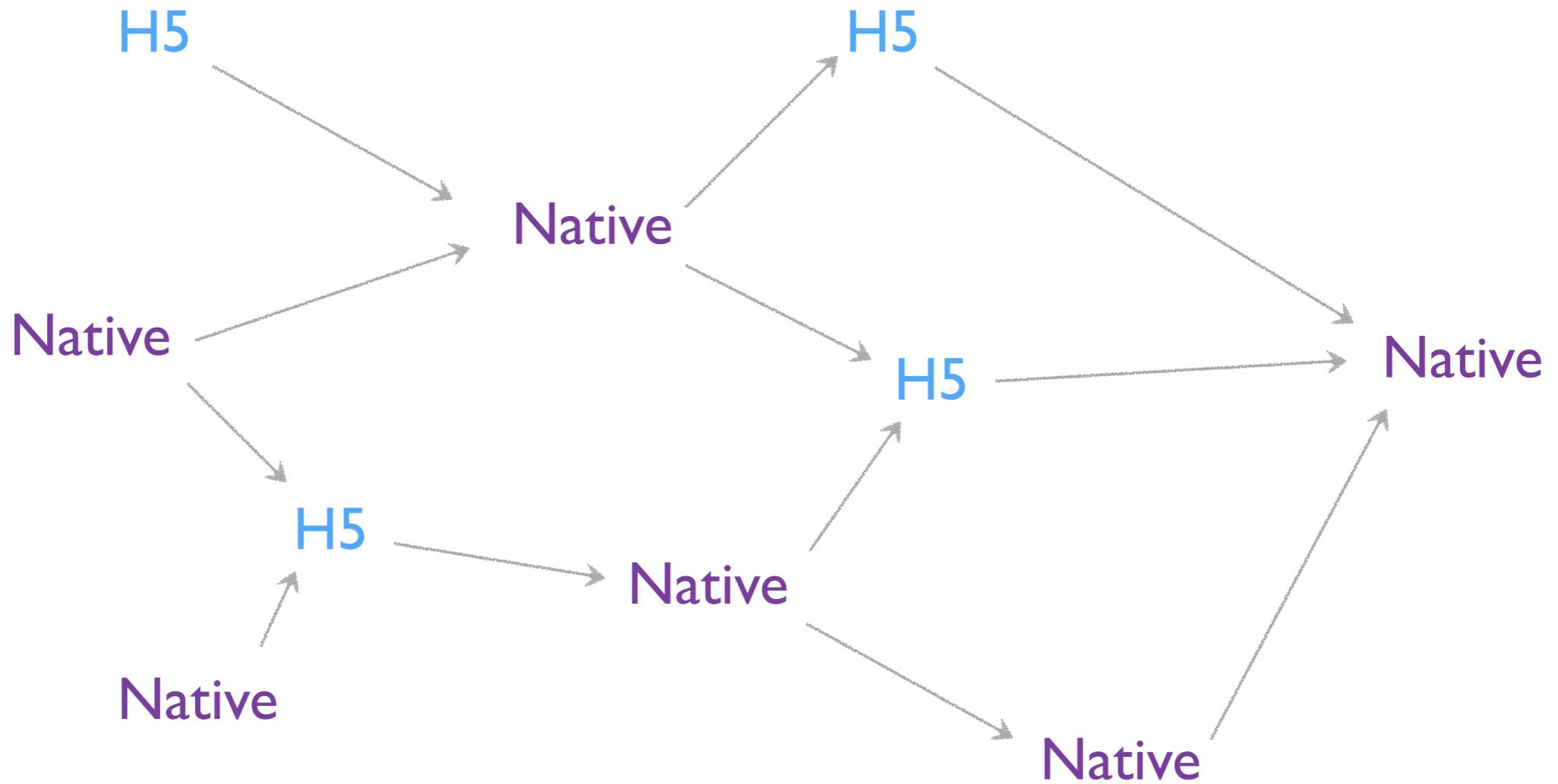
# 2. H5 Page 的别名

The screenshot displays a dashboard with two main sections. At the top, a network log table shows requests from 'usertrack.test.taobao.net' and 'log.m.taobao.com' with various HTTP methods like POST and GET. Below this is a modal window titled 'H5 Page 和 Native Page 在命名上的差异' (Difference in naming between H5 Page and Native Page) with a red arrow pointing to a specific row in the user behavior table. The user behavior table lists pages such as 'Page\_index.html' and 'Page\_Journey\_MsgBox', both highlighted with red boxes. The left sidebar includes links for '最近访问的报表', '今日实时', '基本统计', '用户行为' (selected), '启动次数', '使用时长', '页面路径分析', '关键漏斗', '控件点击' (highlighted in orange), '自定义事件', '活跃及留存', '渠道分析', '终端与网络', '社会化分享', '系统质量', and 'iOS SDK'.

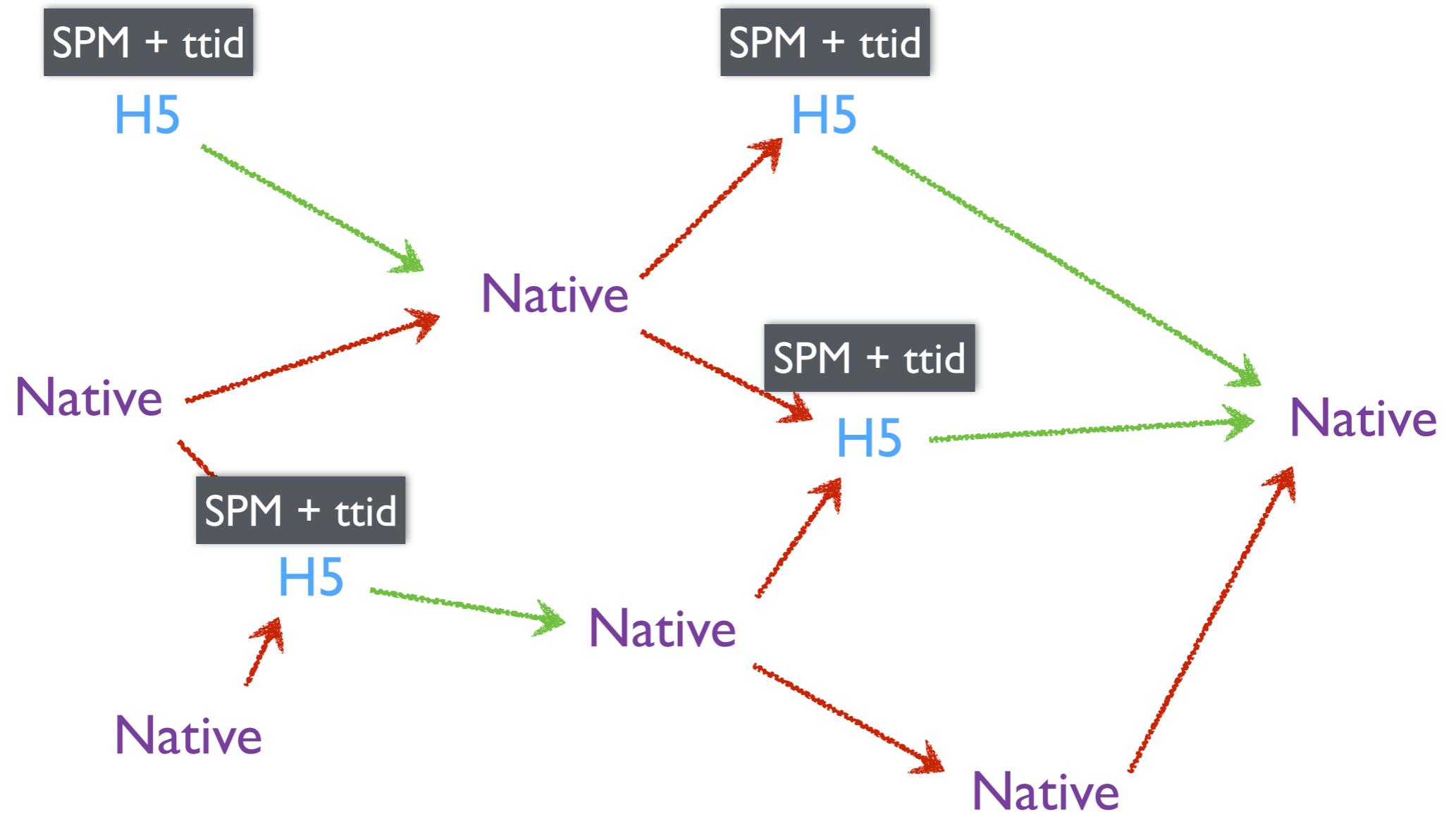
1. 日志表的双份存储
2. H5 Page 的别名

```
1 <!DOCTYPE html>
2 <html>
3 <head><meta name="data-spm" content="181.7626012"><script fixed="true">window.trackerPerformanceTime = {}</script>
4 <meta charset="utf-8">
5 <title>酒店团购</title>
6 <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0,
7 <meta name="apple-mobile-web-app-capable" content="yes">
8 <meta name="apple-mobile-web-app-status-bar-style" content="black">
9 <meta name="format-detection" content="telephone=no">
10 <meta name="format-detection" content="email=no">
11
12
13
14 <meta name="page-name" content="Tuan_List">
15 <link rel="apple-touch-icon-precomposed" href="http://gw.alicdn.com/tps/j1/TlWKqOFmFgXXa.JOZB-114-114.png">
16 <!--Page StyleSheets-->
17 <link tag="combocss" href="http://g.tbcdn.cn/??trip/h5-tuan/0.1.11/pages/list/index-min.css?v=4191194175_46805"
18 <script tag="combojs" src="http://g.tbcdn.cn/??trip/h5-tuan/0.1.11/widgets/base/qa-seed-min.js,trip/h5-tuan/0.1
19 tuan/0.1.11/pages/list/list-min.js?v=1947562776_154944"></script>
20 <meta name="pageid" content="181on.h5-tuan/pages/list/index.html"><script fixed="true">window.trackerPe
21
22
23
```

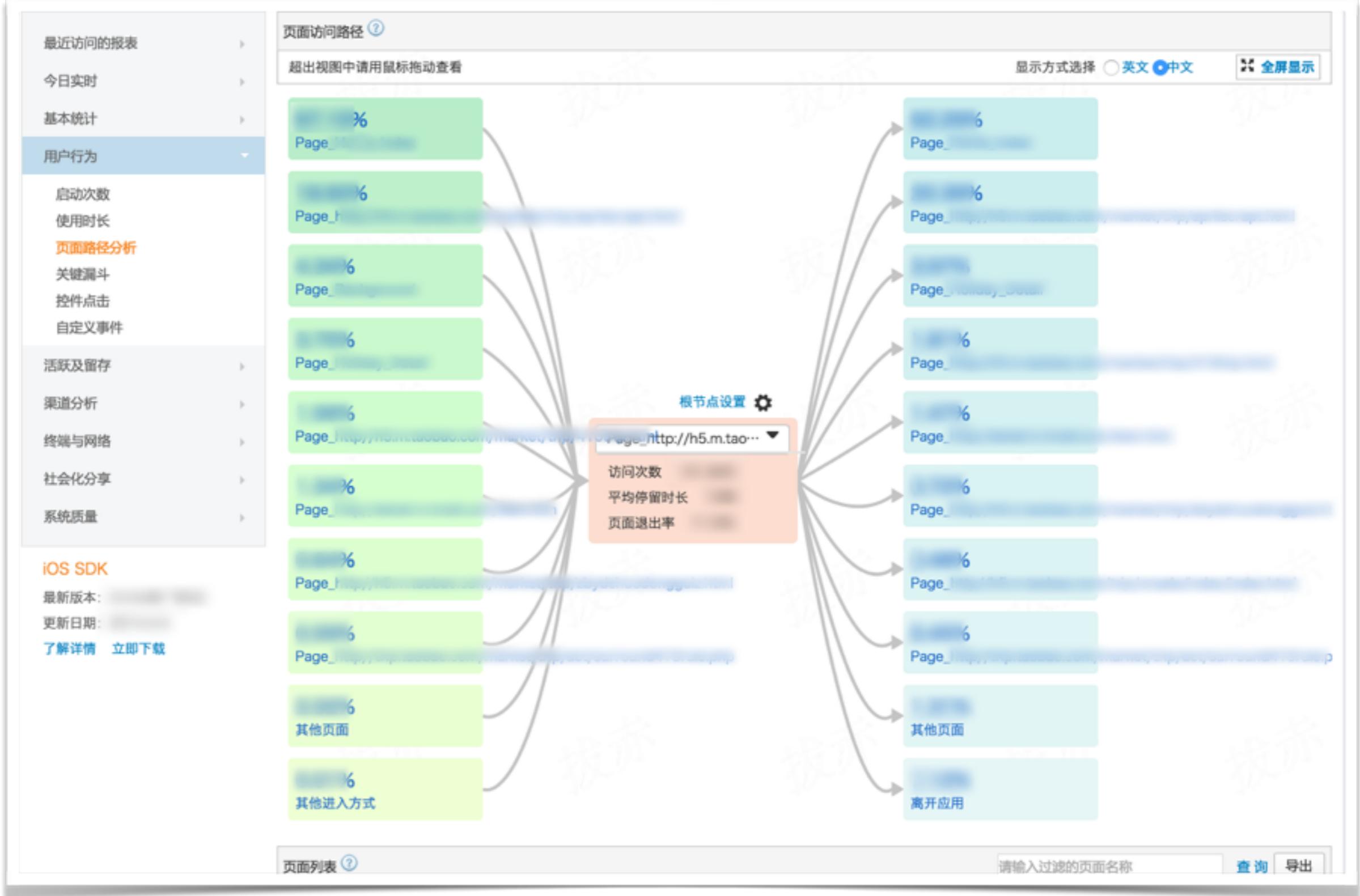
H5 Page 在容器中的别名通过<meta>标签来标示



构建 App 内的 Hybrid PageFlow 的传参规范



SPM 透传 + ttid 透传



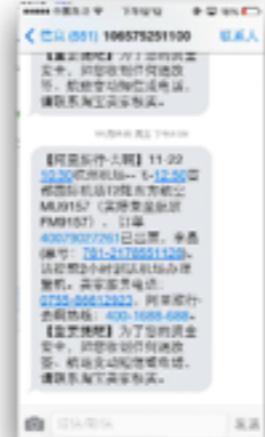
# 挖取 H5 和 Native Page 混用场景的漏斗



钱包机票H5入口



完成购票



短信提醒 or Push

未安装用户则下载



唤起去啊App



(自动)添加行程



积累淘里程



乘机提醒/专车服务



(自动)值机

新用户转化交易、交互闭环

# Questions?

JS/CSS/IMG 资源怎么动态更新?

什么时候更新资源?

活动页面里的动态数据怎么和HTML解耦?

在线离线页面怎么切换?

埋点和数据统计怎么做?

**H5** 与 **Native** 之间的互调如何实现?

WebView 的 Crash 率的难题怎么解?

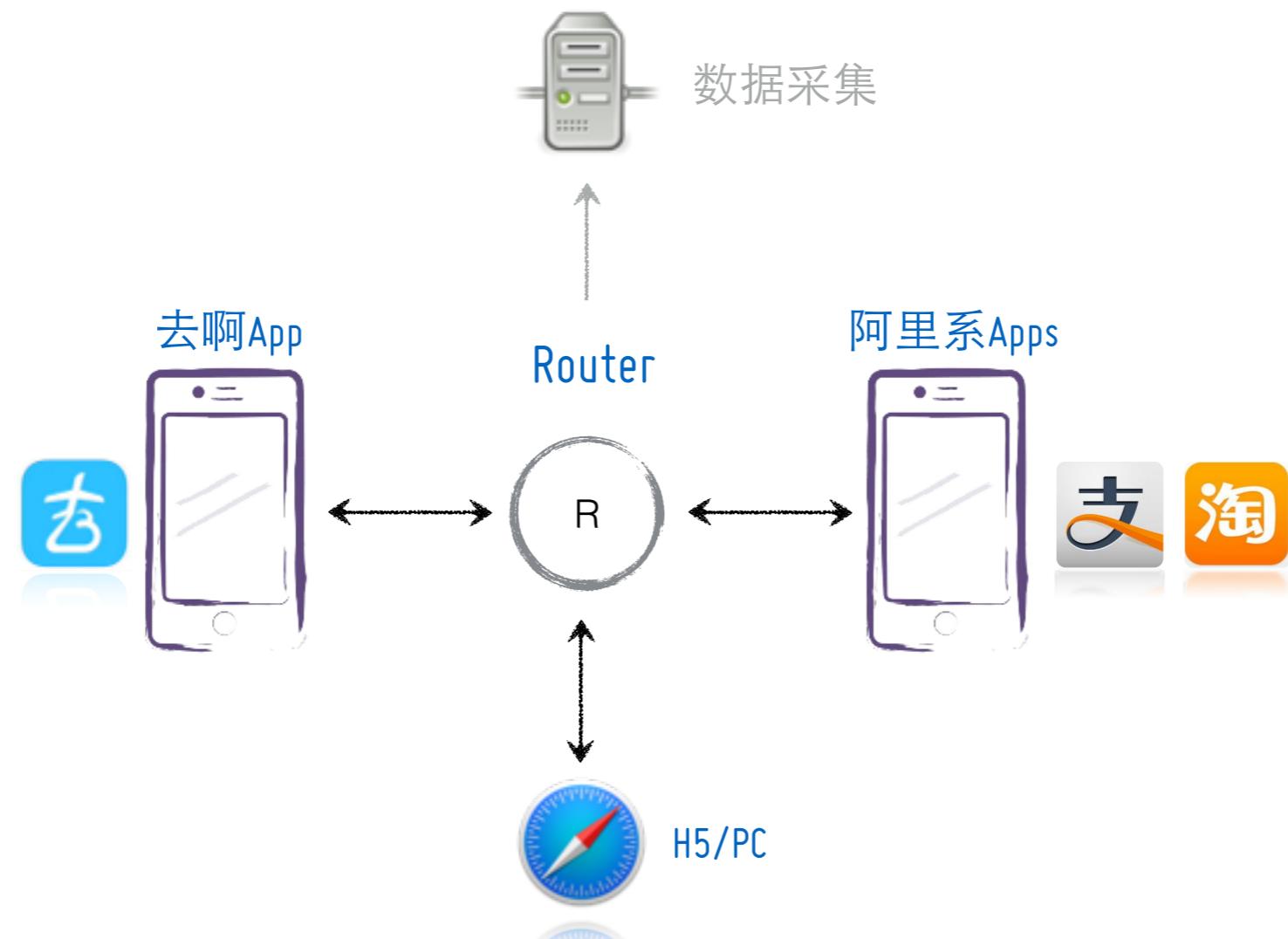
Native 中 Page 资源的定位怎么规范?

HTML 渲染如何加速?

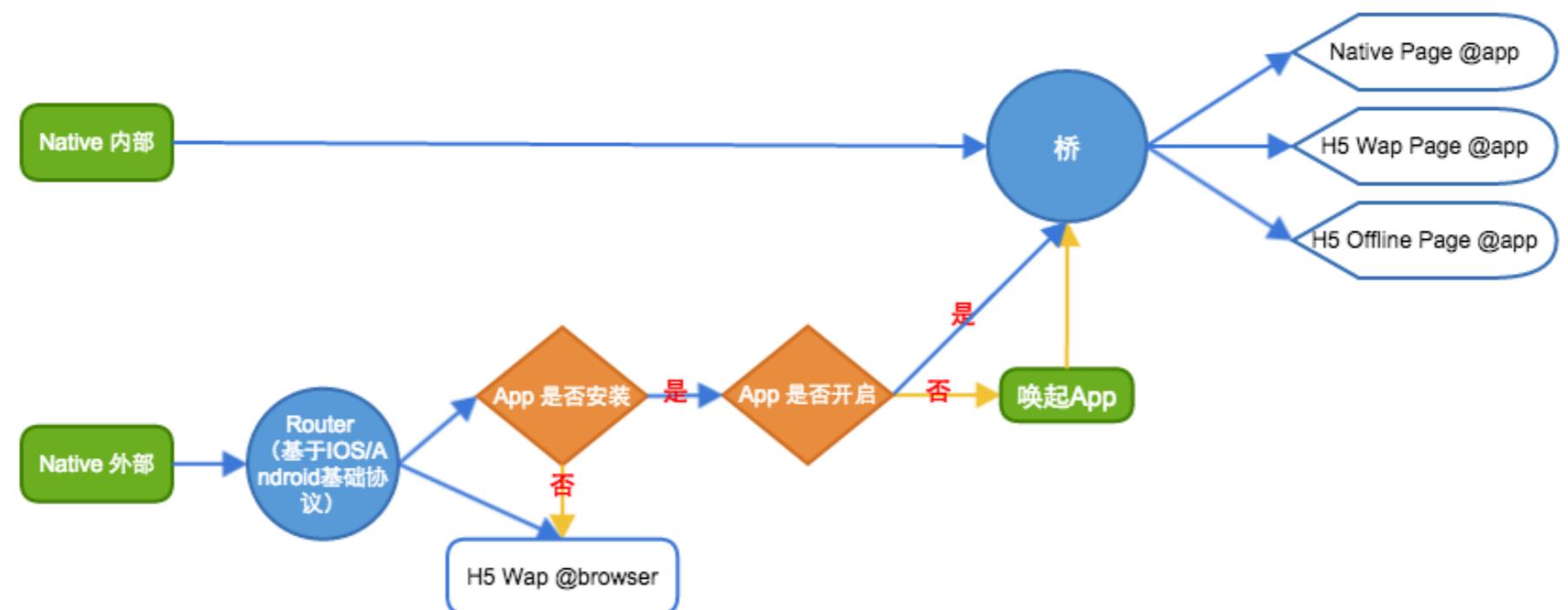
...



Bridge.js



Router: 导流/引导成交/多端Push/唤起和互调



航旅 App 桥接外部调用的路由

	<b>Web</b>	<b>Native</b>	<b>ReactNative</b>	<b>Hybrid</b>
UI 开发的灵活性	高	低	中	高
推包频度	高	低	高	高
加载/渲染性能	低	高	高	比较高
多端兼容	高	低	中	高
开发者门槛	低	高	中	低
更新时效	高	低	中	中
代码回滚效率	高	低	低	高
Page 资源管理 灵活性	高	低	低	高
Crush 率	低	低	低	高

# 重构容器

# 扩展容器

	Web	Native	ReactNative	Hybrid
UI 开发的灵活性	高	低	中	高
推包频度	高	低	高	高
加载/渲染性能	低	高	高	比较高
多端兼容	高	低	中	高
开发者门槛	低	高	中	低
更新时效	高	低	中	中
代码回滚效率	高	低	低	高
Page 资源管理 灵活性	高	低	低	高
Crush 率	低	低	低	高

# So

## I. 我们重定义了无线领域中 FrontEnd 的角色

向上多端兼容、向下重构 Web 容器

## 2. 我们重构了 Hybrid 的整个基础设施

构建工具、资源发布系统、资源离线、数据统计规范

## 3. 打破 H5 和 Native 之间的墙

90% 的场景下，H5 和 Native 互为备份，取长补短

H5 资源离线化就万事大吉了？



# Questions?

JS/CSS/IMG 资源怎么动态更新?

什么时候更新资源?

活动页面里的动态数据怎么和HTML解耦?

虚拟域的实现以及在线离线页面怎么切换?

埋点和数据统计怎么做?

H5 与 Native 之间的互调如何实现?

**WebView** 的 **Crush** 率的难题怎么解?

**Native** 中 **Page** 资源的定位怎么规范?

**HTML** 渲染如何加速?

...

	<b>Web</b>	<b>Native</b>	<b>ReactNative</b>	<b>Hybrid</b>
UI 开发的灵活性	高	低	中	高
推包频度	高	低	高	高
加载/渲染性能	低	高	高	比较高
多端兼容	高	低	中	高
开发者门槛	低	高	中	低
更新时效	高	低	中	中
代码回滚效率	高	低	低	高
Page 资源管理 灵活性	高	低	低	高
Crush 率	低	低	低	高

	Web	Native	ReactNative	Hybrid
UI 开发的灵活性	高	低	中	高
加载/渲染性能	低	高	高	比较高
多端兼容	高	低	中	高
更新时效	高	低	中	中
代码回滚效率	高	低	低	高
Page 资源管理 灵活性	高	低	低	高
Crush 率	低	低	低	高

## 基于 JSVM 重构容器：HTML/CSS 本地渲染

## 基于 VirtualDNS 的路由：Native 中的“URL”

	Web	Native	ReactNative	Hybrid
UI 开发的灵活性	高	低	中	高
加载/渲染性能	低	高	中	比较高
多端兼容	高	低	中	高

## 基于 JSVM 重构容器：HTML/CSS 本地渲染

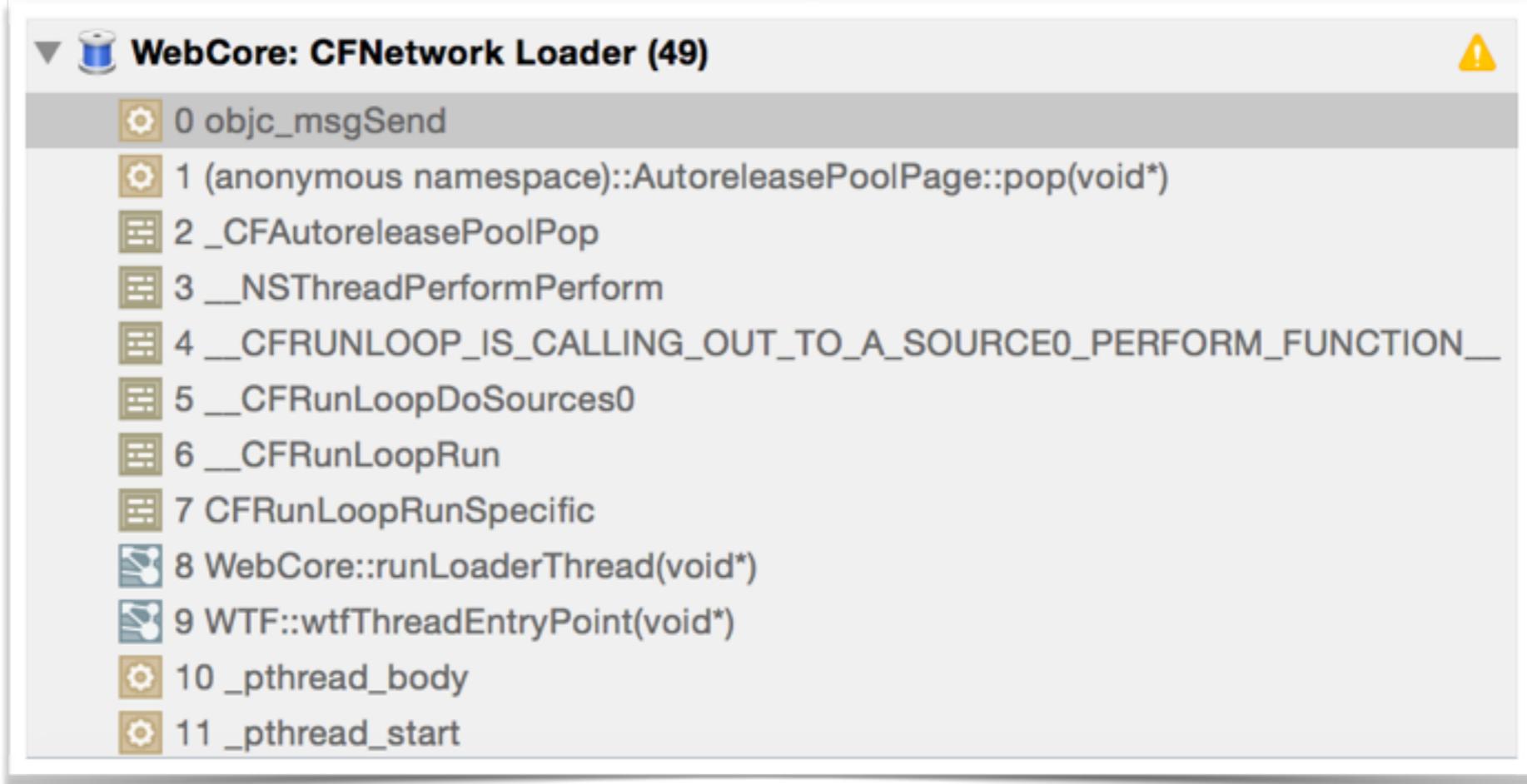
	低	高	中	比较高
多端兼容	高	低	中	高
更新时效	高	低	中	中
代码回滚效率	高	低	低	高

## 基于 VirtualDNS 的路由：Native 中的“URL”

	高	低	中	中
代码回滚效率	高	低	低	高
Page 资源管理 灵活性	高	低	低	对资源的高效管理
Crush 率	低	低	低	高

```
0 JavaScriptCore 0x3087852a JSC::LLInt::setUpCall(JSC::ExecState*, JSC::Instruction*, JSC::CodeSpecializationKind, JSC::JSValue, JSC::LLIntCallLinkInfo*) + 30
1 JavaScriptCore 0x3086f5b7 llint_slow_path_call + 123
2 JavaScriptCore 0x30873ef9 llint_op_call + 201
3 JavaScriptCore 0x30858b63 JSC::evaluate(JSC::ExecState*, JSC::SourceCode const&, JSC::JSValue, JSC::JSValue*) + 427
4 WebCore 0x37a480f7 WebCore::ScriptController::evaluateInWorld(WebCore::ScriptSourceCode const&, WebCore::DOMWrapperWorld*) + 255
5 WebCore 0x37a47fb7 WebCore::ScriptController::evaluate(WebCore::ScriptSourceCode const&) + 27
6 WebCore 0x37a47f2d WebCore::ScriptElement::executeScript(WebCore::ScriptSourceCode const&) + 325
7 WebCore 0x37afde33 WebCore::HTMLScriptRunner::executePendingScriptAndDispatchEvent(WebCore::PendingScript&) + 159
8 WebCore 0x37af709d WebCore::HTMLScriptRunner::executeParsingBlockingScripts() + 245
9 WebCore 0x37a532ff WebCore::HTMLDocumentParser::executeScriptsWaitingForStylesheets() + 27
10 WebCore 0x37a50585 WebCore::Document::didRemoveAllPendingStylesheet() + 41
11 WebCore 0x37b72005 WebCore::HTMLLinkElement::sheetLoaded() + 77
12 WebCore 0x37a5043b WebCore::StyleSheetContents::checkLoaded() + 99
13 WebCore 0x37b71949 WebCore::HTMLLinkElement::setCSSStyleSheet(WTF::String const&, WebCore::KURL const&, WTF::String const&, WebCore::CachedCSSStyleSheet const*) + 857
14 WebCore 0x37afeeb5 WebCore::CachedCSSStyleSheet::checkNotify() + 257
15 WebCore 0x37b71407 WebCore::CachedCSSStyleSheet::finishLoading(WebCore::ResourceBuffer*) + 219
16 WebCore 0x37af9d71 WebCore::SubresourceLoader::didFinishLoading(double) + 109
17 WebCore 0x37af9cd5 WebCore::didFinishLoading(_CFURLConnection*, void const*) + 125
18 CFNetwork 0x2f52d0e7 __ZN27URLConnectionClient_Classic26_delegate_didFinishLoadingEU13block_pointerFvvE_block_invoke + 75
19 CFNetwork 0x2f52bcf7
__ZN27URLConnectionClient_Classic18_withDelegateAsyncEPKcU13block_pointerFvP16_CFURLConnectionPK33CFURLConnectionClientCurrent_VMaxE_block_invoke_2 + 55
20 CoreFoundation 0x2f7fa8f1 CFArrayApplyFunction + 37
21 CFNetwork 0x2f4c36bb RunloopBlockContext::perform() + 183
22 CFNetwork 0x2f4c3579 MultiplexerSource::perform() + 221
23 CFNetwork 0x2f4c340d MultiplexerSource::_perform(void*) + 49
24 CoreFoundation 0x2f89220b __CFRUNLOOP_IS_CALLING_OUT_TO_A_SOURCE0_PERFORM_FUNCTION__ + 15
25 CoreFoundation 0x2f8916db __CFRunLoopDoSources0 + 207
26 CoreFoundation 0x2f88fecf __CFRunLoopRun + 623
27 CoreFoundation 0x2f7faebf CFRunLoopRunSpecific + 523
28 CoreFoundation 0x2f7faca3 CFRunLoopRunInMode + 107
29 WebCore 0x37a921ab RunWebThread(void*) + 419
30 libsystem_pthread.dylib 0x3a6a3919 _pthread_body + 141
31 libsystem_pthread.dylib 0x3a6a388b _pthread_start + 103
32 libsystem_pthread.dylib 0x3a6a1aa4 thread_start + 8
```

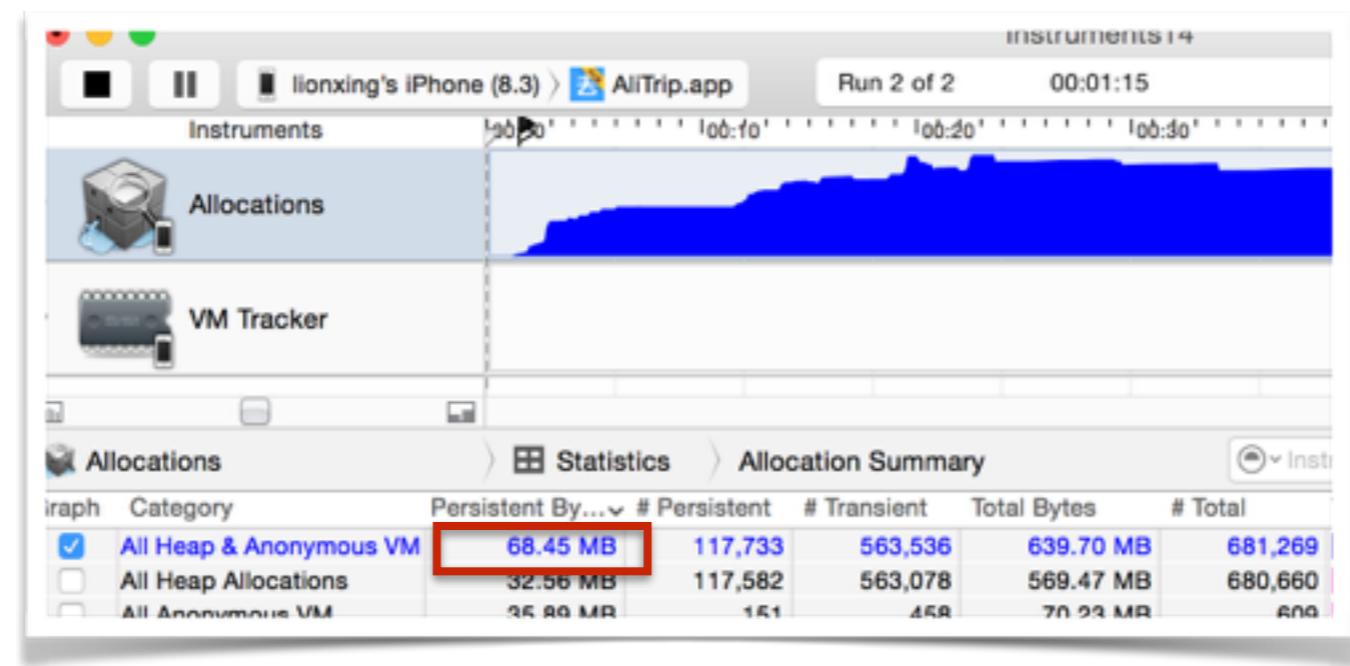
```
0 WebCore 0x00000001928a860c WebCore::CachedResource::unregisterHandle(WebCore::CachedResourceHandleBase*)
+ 144
1 WebCore 0x00000001928a856c WebCore::CachedResourceHandleBase::~CachedResourceHandleBase() + 32
2 WebCore 0x0000000192a0d6c4 WebCore::MemoryCache::pruneDeadResourcesToSize(unsigned int) + 352
3 WebCore 0x0000000192922de0 WebCore::MemoryCache::prune() + 128
4 WebCore 0x0000000192959aec WebCore::DocumentLoader::detachFromFrame() + 400
5 WebKitLegacy 0x00000001937c2b08 WebDocumentLoaderMac::detachFromFrame() + 20
6 WebCore 0x0000000192d5b914 WebCore::FrameLoader::detachFromParent() + 224
7 WebKitLegacy 0x0000000193829430 __29-[WebView(WebPrivate) _close]_block_invoke + 372
8 WebCore 0x00000001935f78b0 HandleRunSource(void*) + 368
9 CoreFoundation 0x0000000183f94240 __CFRUNLOOP_IS_CALLING_OUT_TO_A_SOURCE0_PERFORM_FUNCTION__ + 24
10 CoreFoundation 0x0000000183f9359c __CFRunLoopDoSources0 + 448
11 CoreFoundation 0x0000000183f91594 __CFRunLoopRun + 712
12 CoreFoundation 0x0000000183ebd2d4 CFRunLoopRunSpecific + 396
13 WebCore 0x0000000192920894 RunWebThread(void*) + 468
14 libsystem_pthread.dylib 0x0000000195fe3dc8 _pthread_body + 164
15 libsystem_pthread.dylib 0x0000000195fe3d24 _pthread_body + 0
16 libsystem_pthread.dylib 0x0000000195fe0ef8 thread_start + 4
```



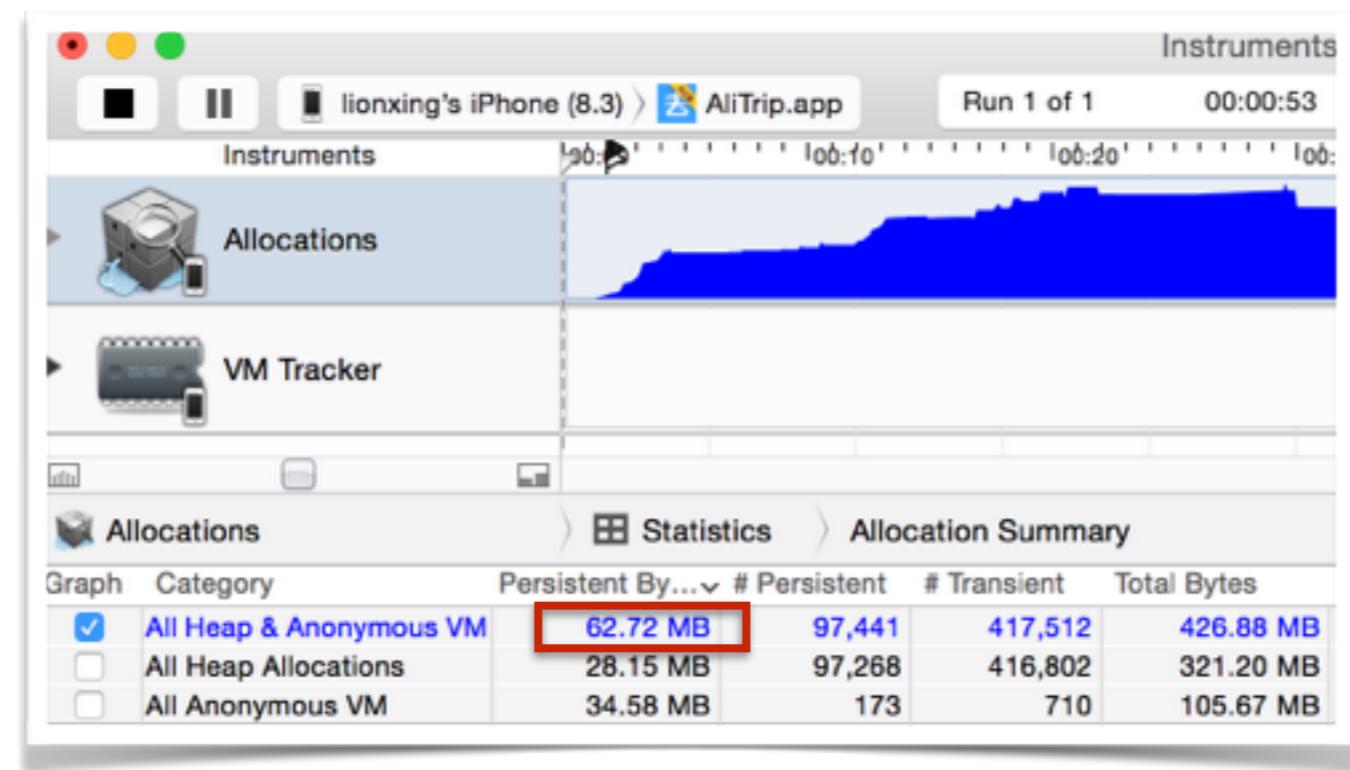
The call stack visualization shows a tree structure of function calls. The root node is "WebCore: CFNetwork Loader (49)". Below it is a list of 16 numbered entries, each with a small icon indicating its source library. The first entry is highlighted with a grey background.

- ▼ **WebCore: CFNetwork Loader (49)** ⚠
- 0 objc\_msgSend
- 1 (anonymous namespace)::AutoreleasePoolPage::pop(void\*)
- 2 \_CFAutoreleasePoolPop
- 3 \_\_NSThreadPerformPerform
- 4 \_\_CFRUNLOOP\_IS\_CALLING\_OUT\_TO\_A\_SOURCE0\_PERFORM\_FUNCTION\_\_
- 5 \_\_CFRunLoopDoSources0
- 6 \_\_CFRunLoopRun
- 7 CFRunLoopRunSpecific
- 8 WebCore::runLoaderThread(void\*)
- 9 WTF::wtfThreadEntryPoint(void\*)
- 10 \_pthread\_body
- 11 \_pthread\_start

# 内存啊，内存！



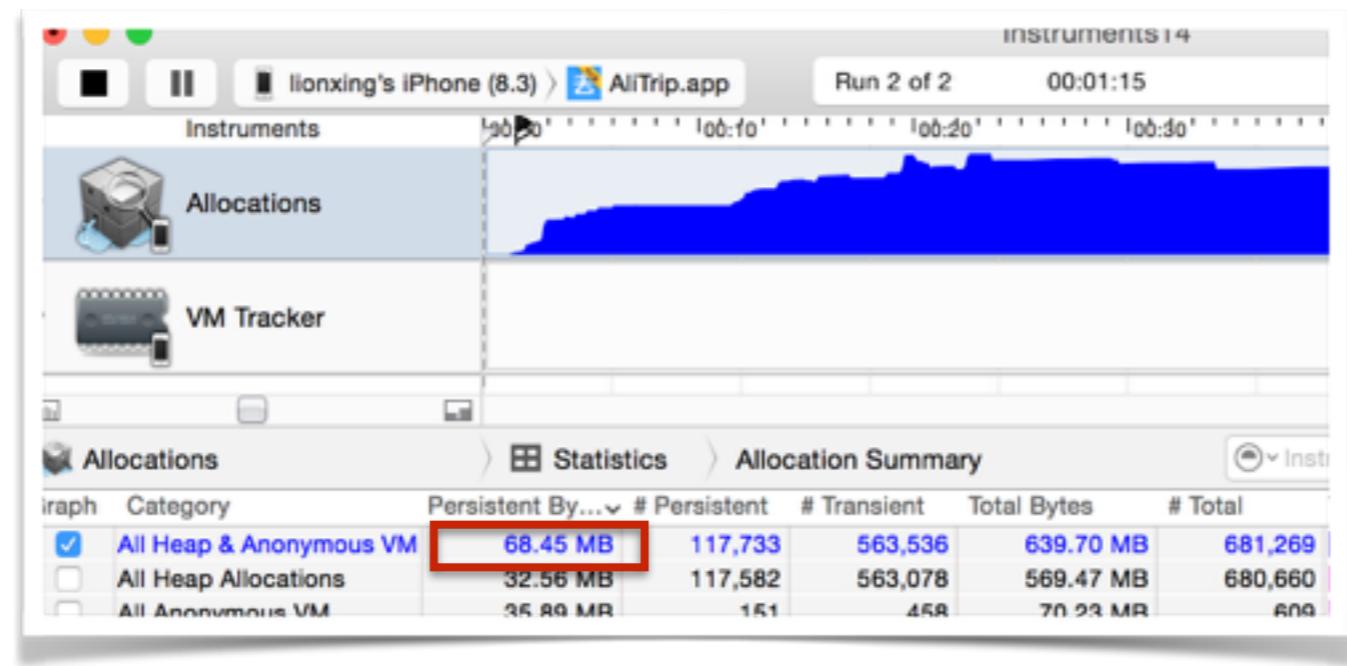
优化前



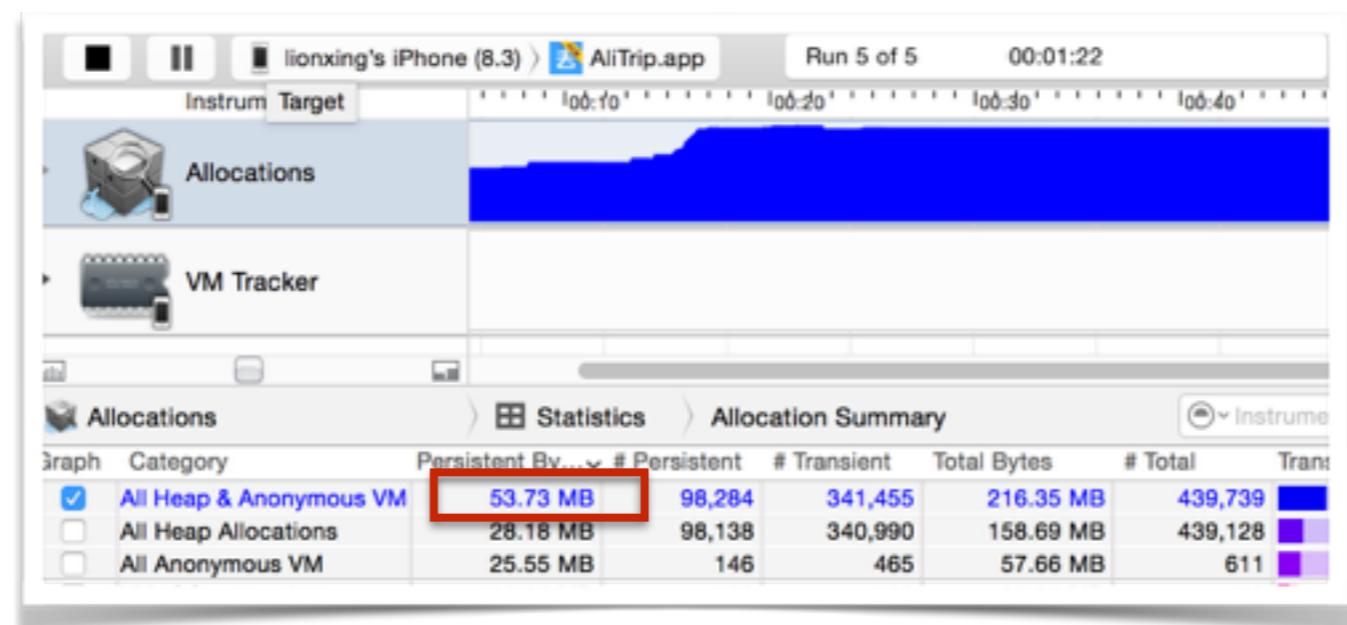
优化后

JS 执行频繁对内存的影响！

# 内存啊，内存！



优化前



优化后

图片质量对内存的影响！



低质量图片



高质量图片

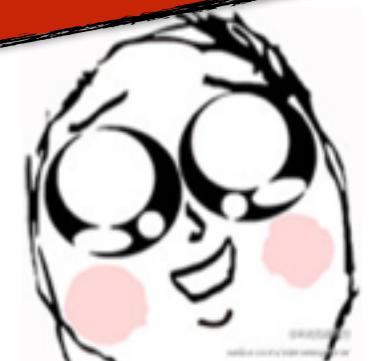


低质量图片



高质量图片

无差别

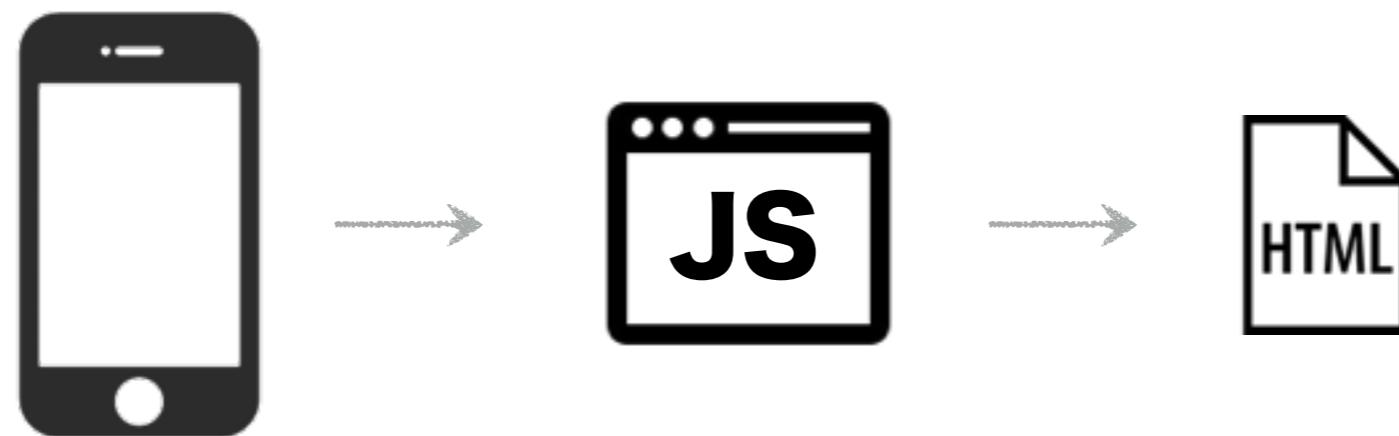


# 运行时 Hybrid 三个阶段



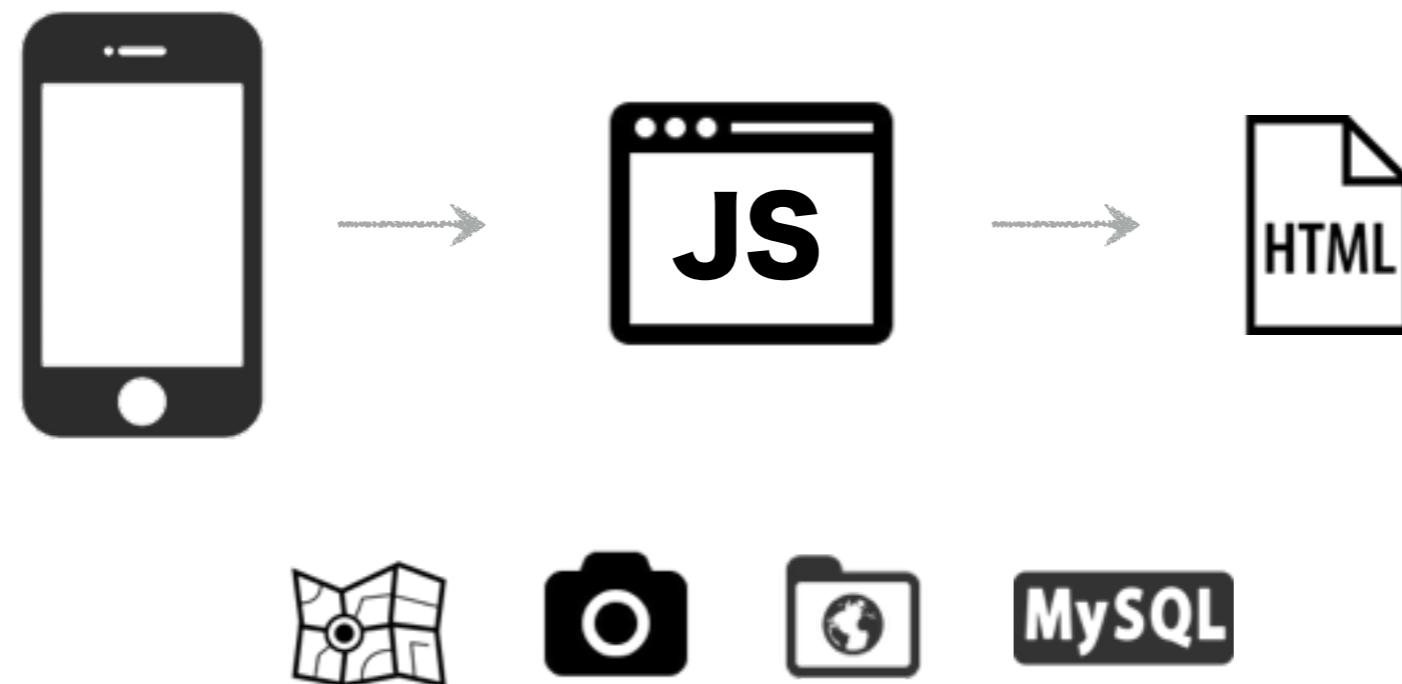
第一阶段：App 内嵌 WebView 页面

# 运行时 Hybrid 三个阶段



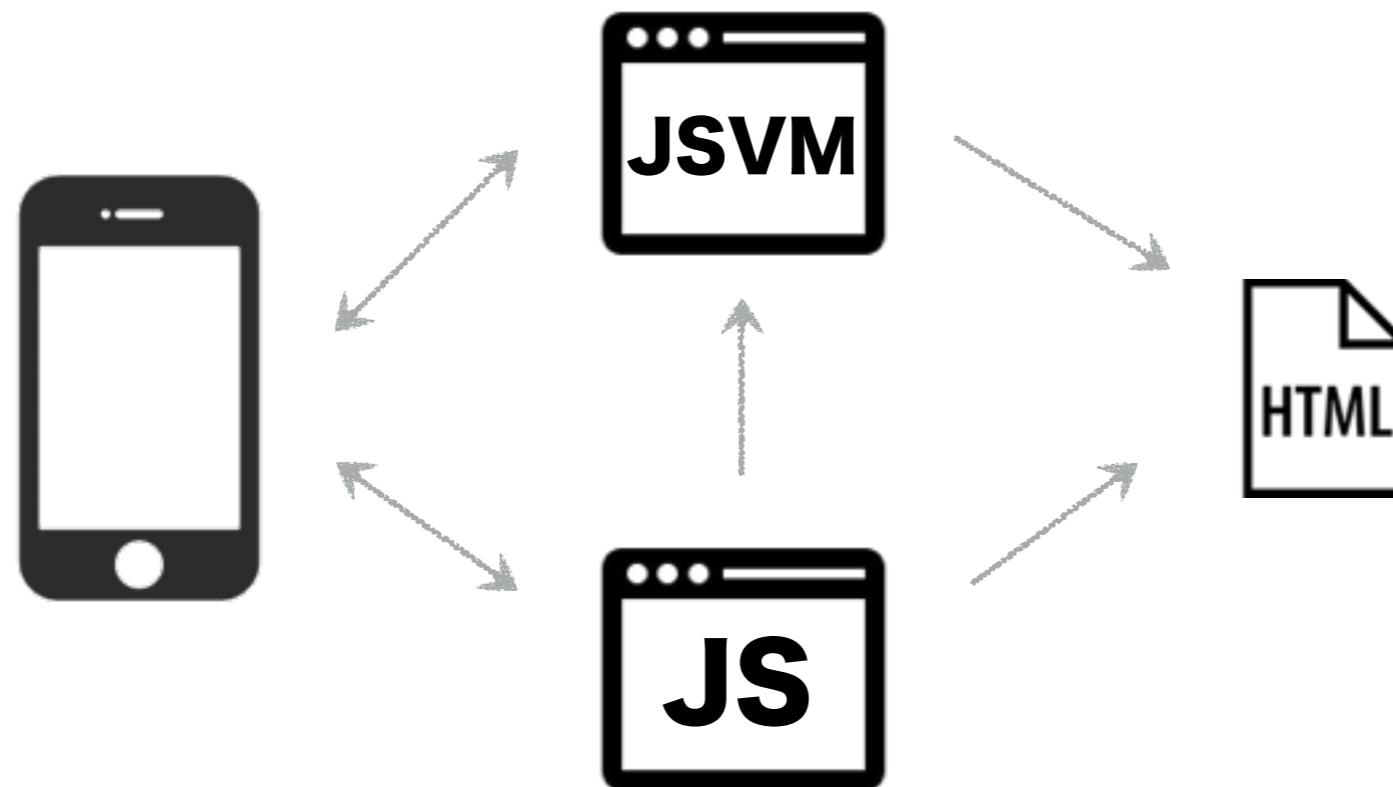
第二阶段：WebView 容器 + JSBridge

# 运行时 Hybrid 三个阶段



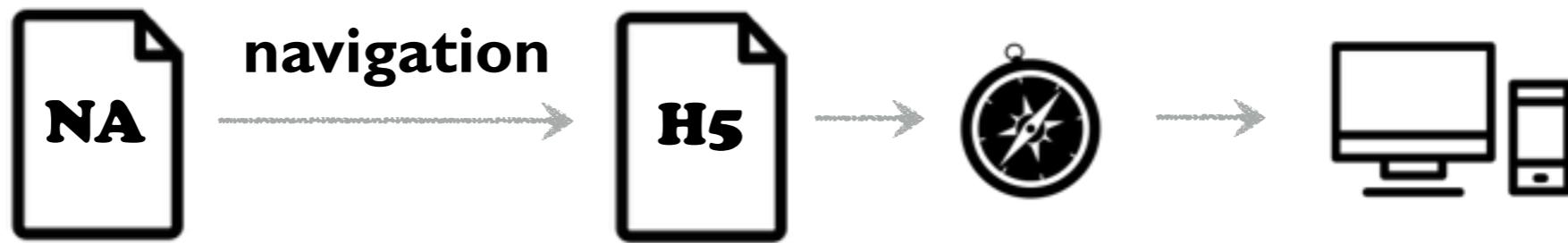
第二阶段：WebView 容器 + JSBridge

# 运行时 Hybrid 三个阶段

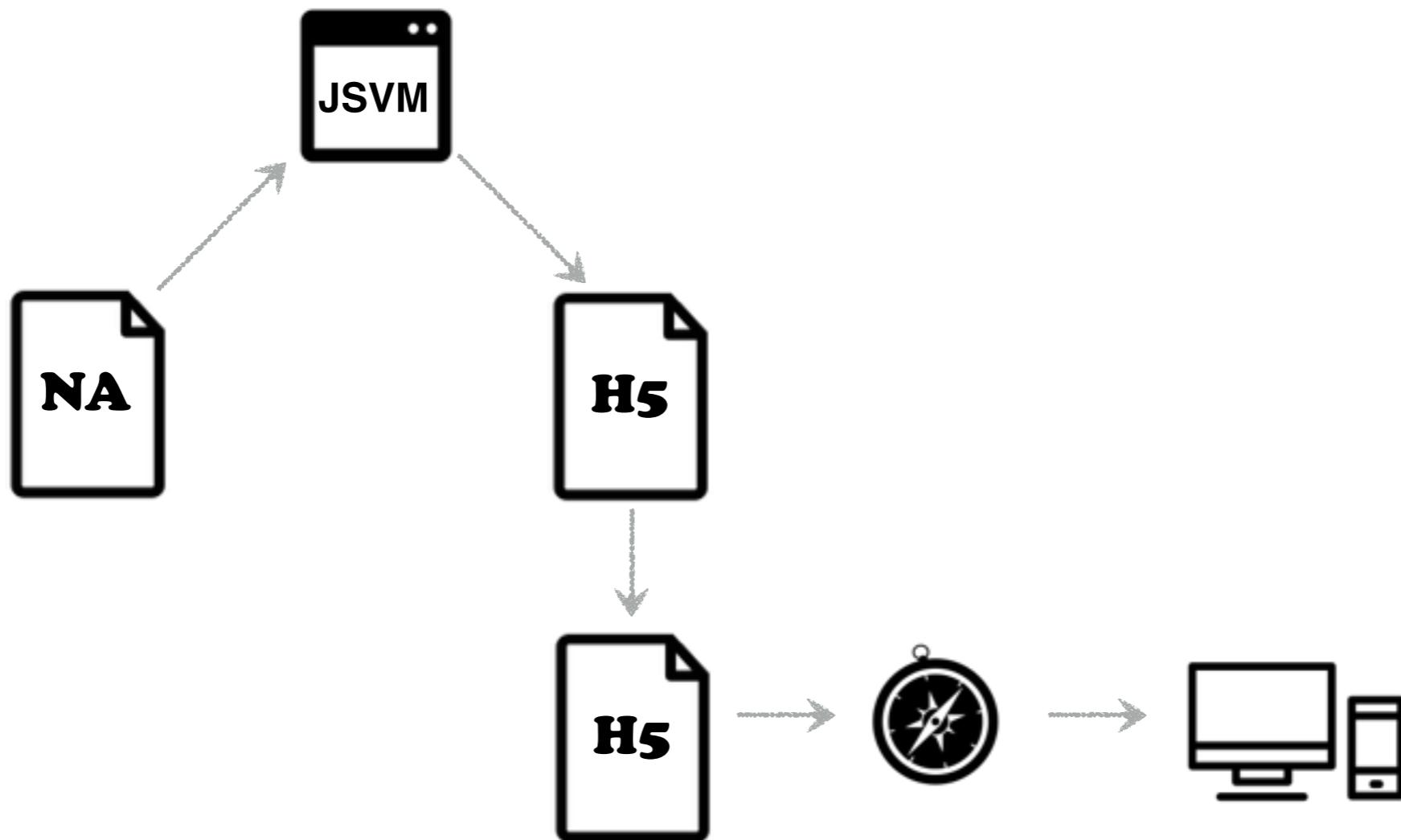


第三阶段：WebView 容器 + JSVM

# WebView 容器 + JSVM



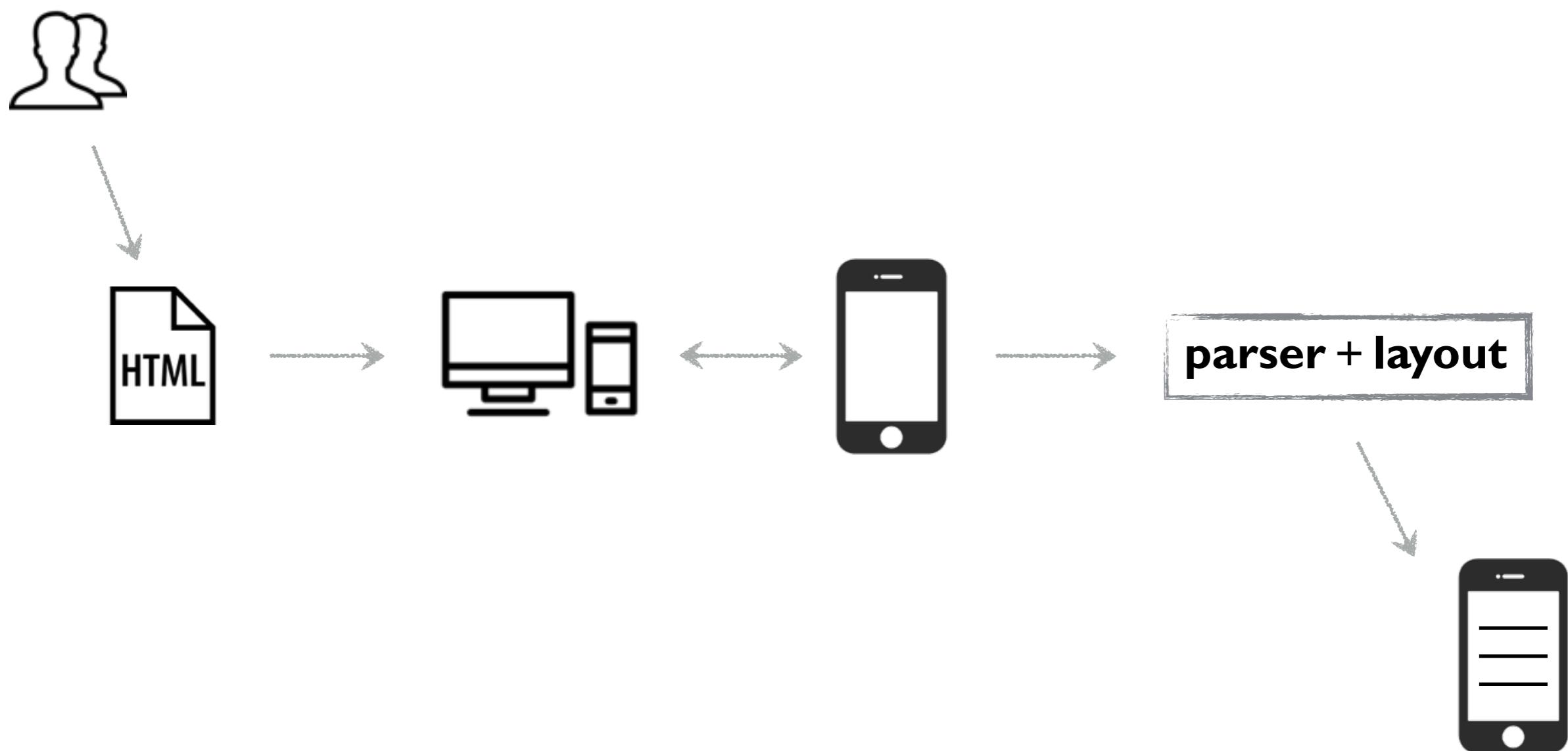
# WebView 容器 + JSVM



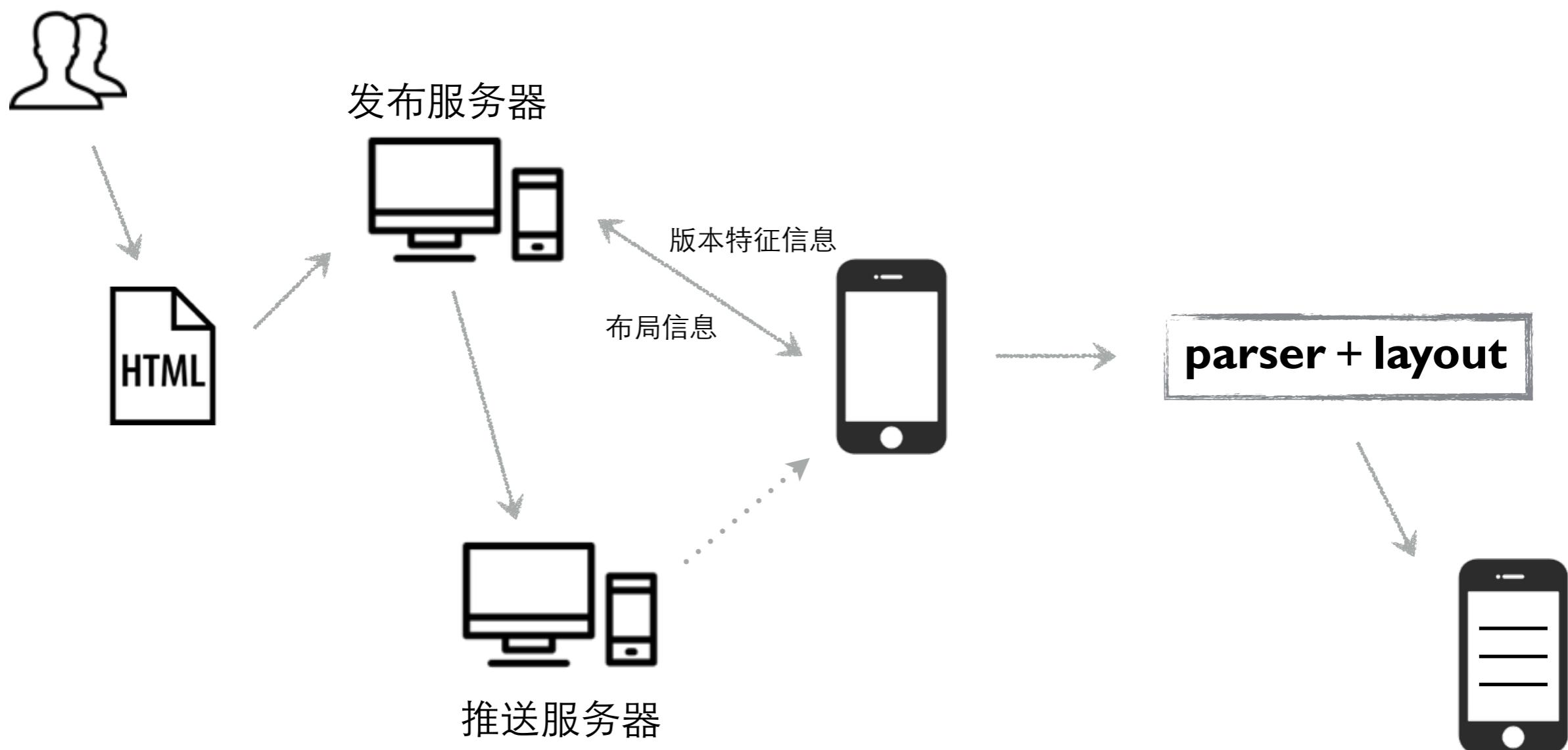
# HTML + CSS 本地渲染



# HTML + CSS 本地渲染

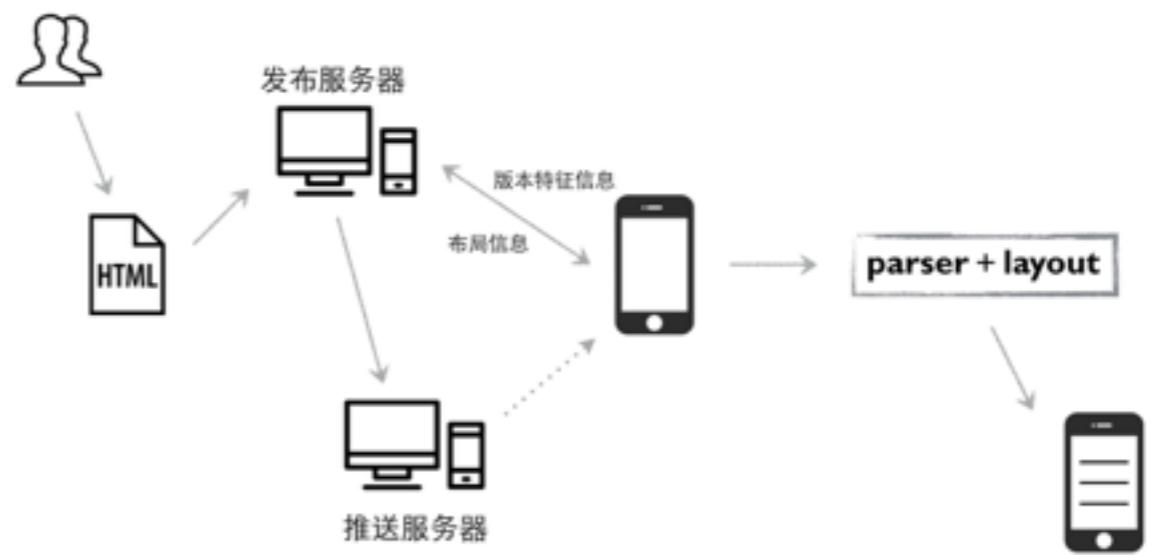


# HTML + CSS 本地渲染



# HTML + CSS 本地渲染

1. 可动态更新的页面 + 布局
2. 更少，更可控的内存
3. H5 开发受到一定局限性



# Questions?

JS/CSS/IMG 资源怎么动态更新?

什么时候更新资源?

活动页面里的动态数据怎么和HTML解耦?

虚拟域的实现以及在线离线页面怎么切换?

埋点和数据统计怎么做?

H5 与 Native 之间的互调如何实现?

WebView 的 Crash 率的难题怎么解?

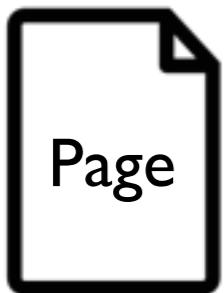
**Native 中 Page 资源的定位怎么规范?**

HTML 渲染如何加速?

...

# Native 中 Page 资源定位规范

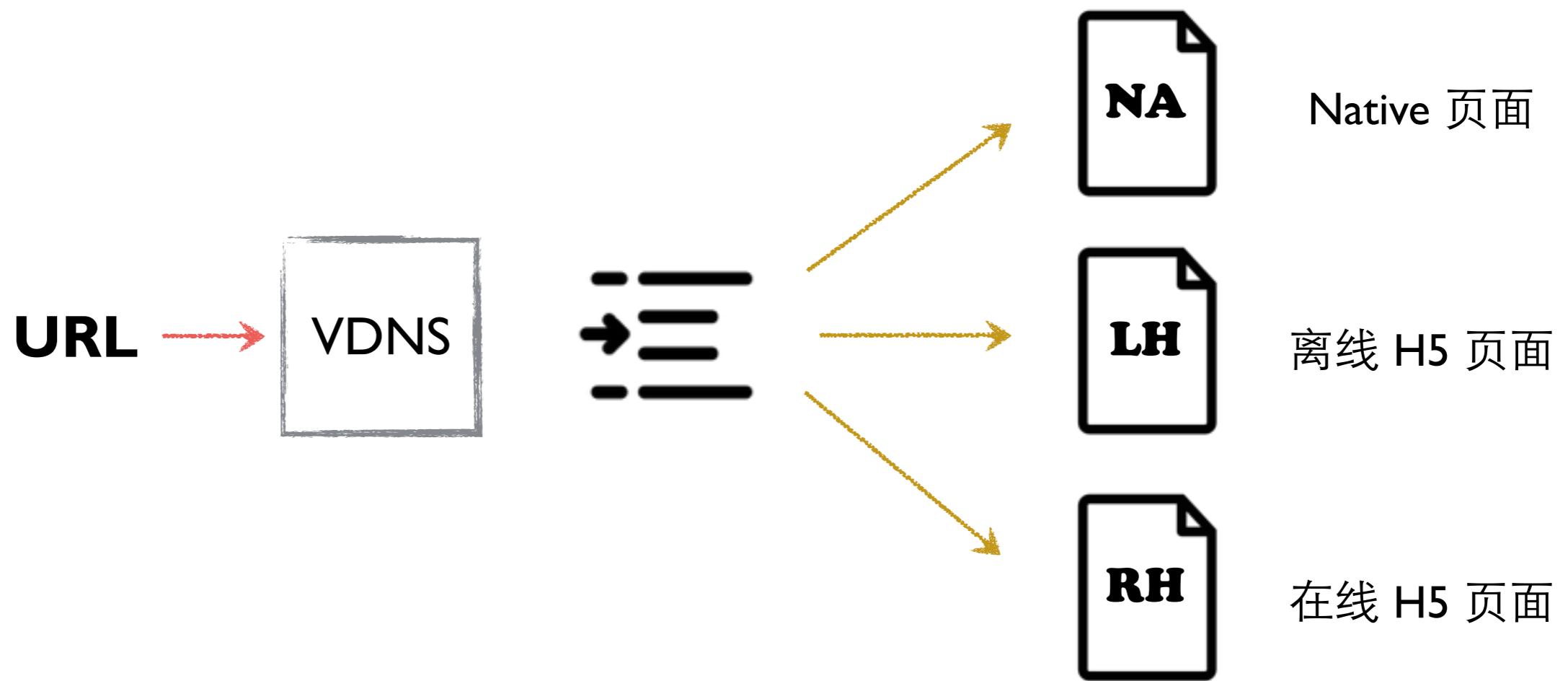
## I. 用 URL 规范给所有 Page 设定唯一标示



PageName —→ <http://xxx.xxx.xxx/xxx/xxx.html>

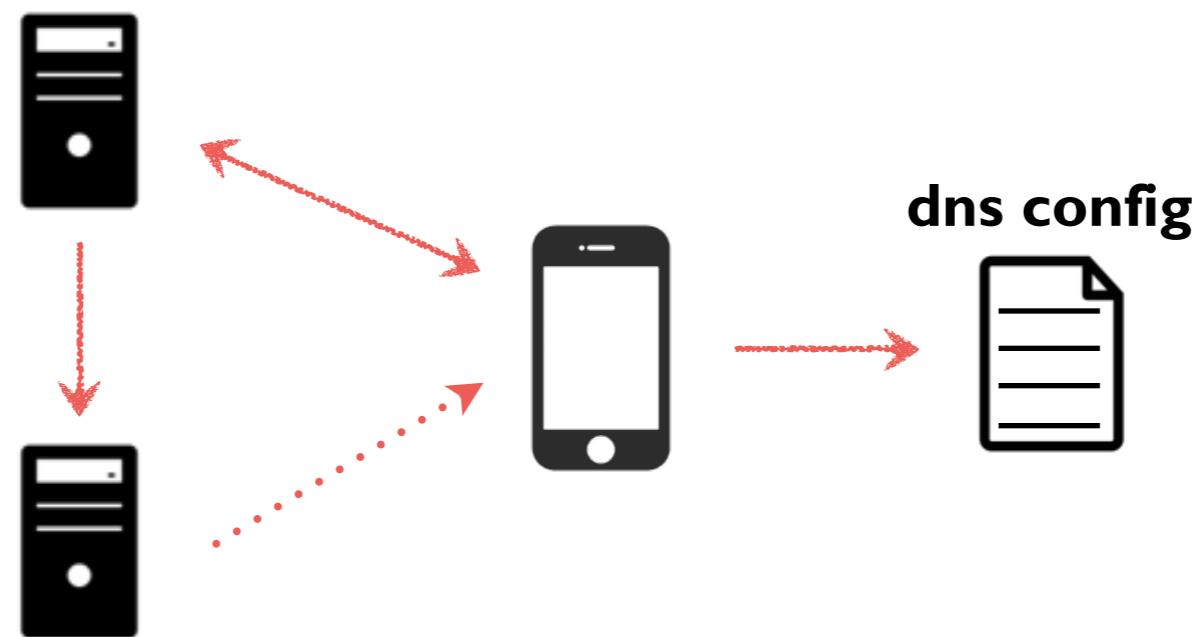
# Native 中 Page 资源定位规范

1. 用 URL 规范给所有 Page 设定唯一标示
2. 虚拟域：设置本地虚拟 DNS

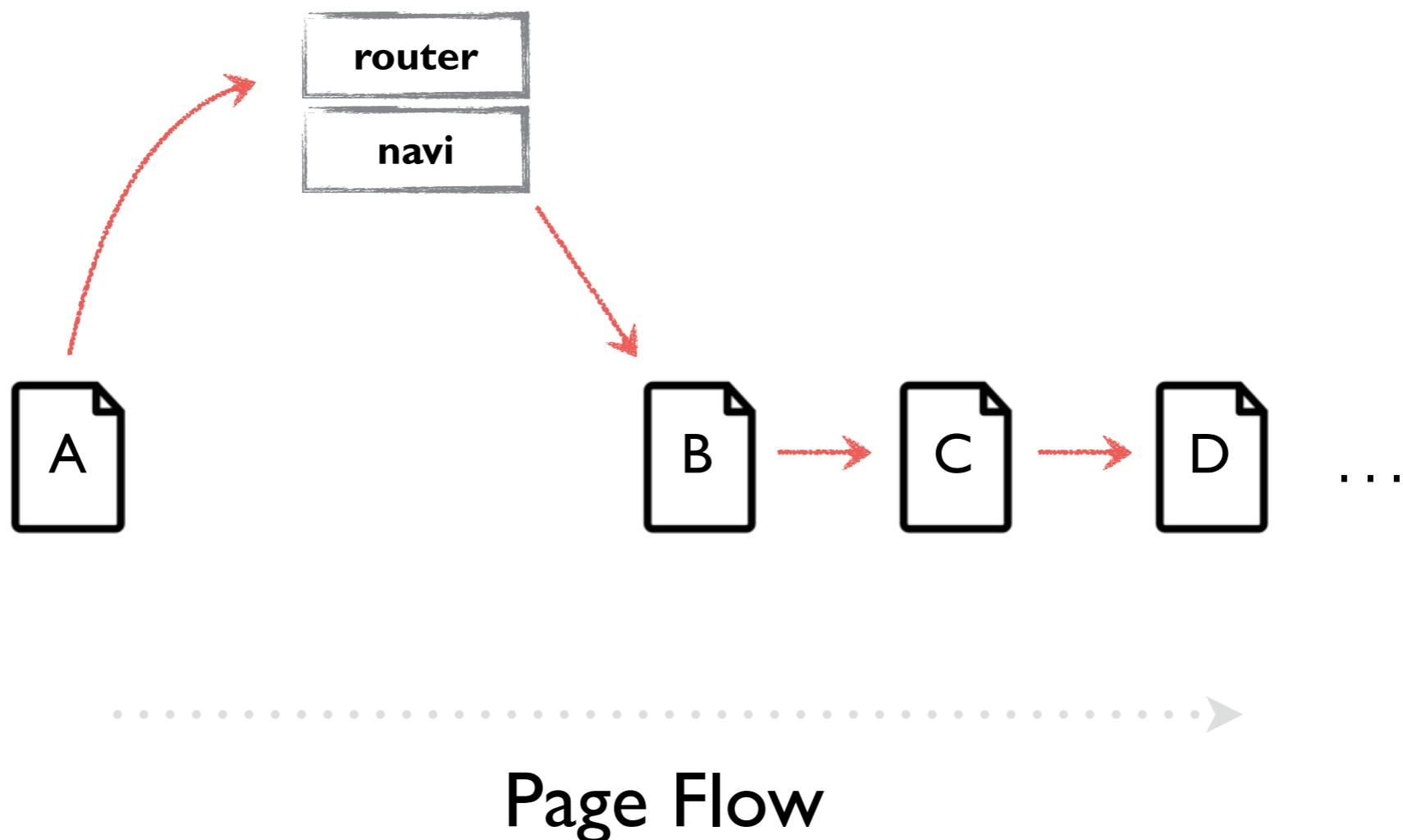


# Native 中 Page 资源定位规范

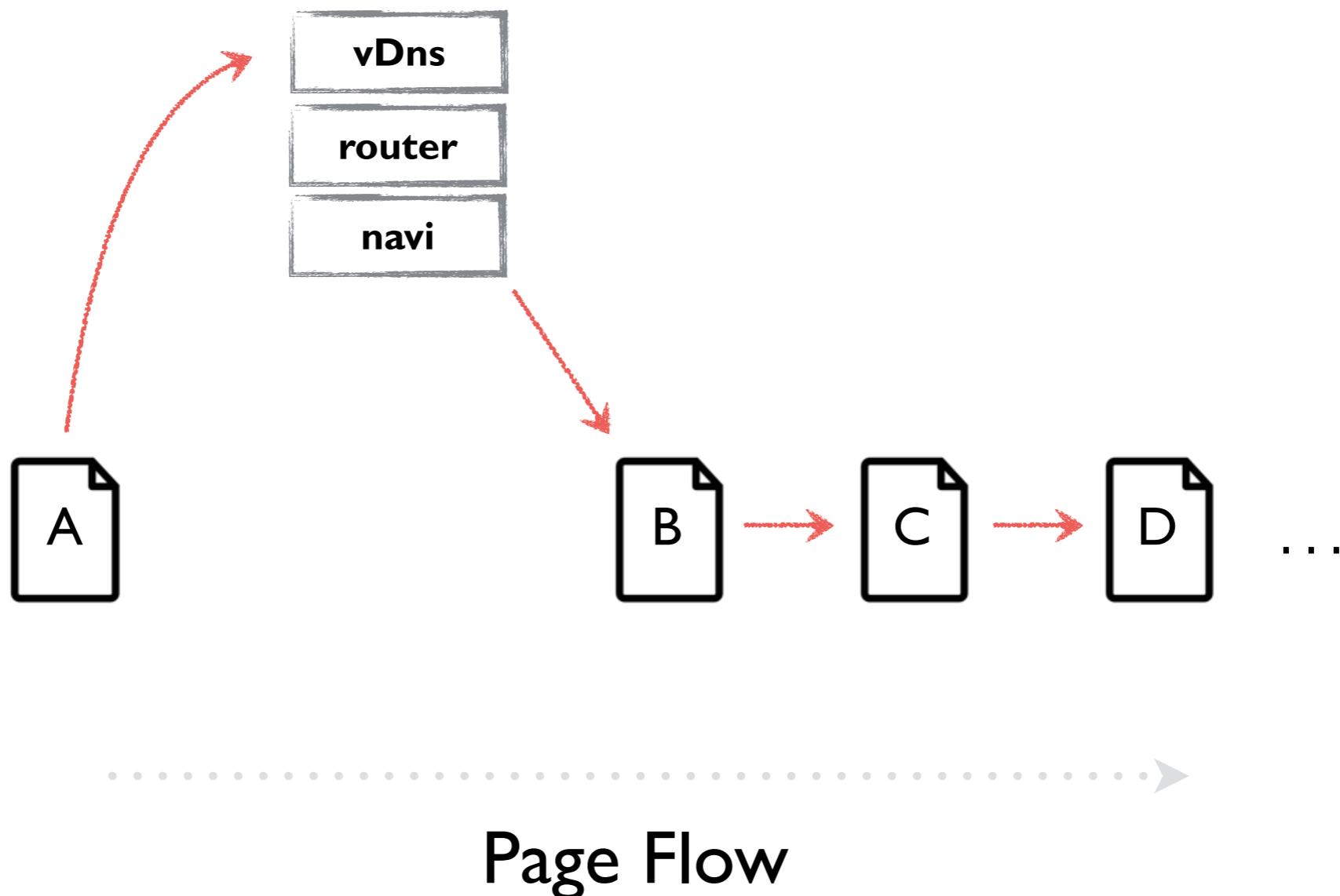
1. 用 URL 规范给所有 Page 设定唯一标示
2. 虚拟域：设置本地虚拟 DNS
3. 通过配置服务器和推送服务器来更新虚拟 DNS



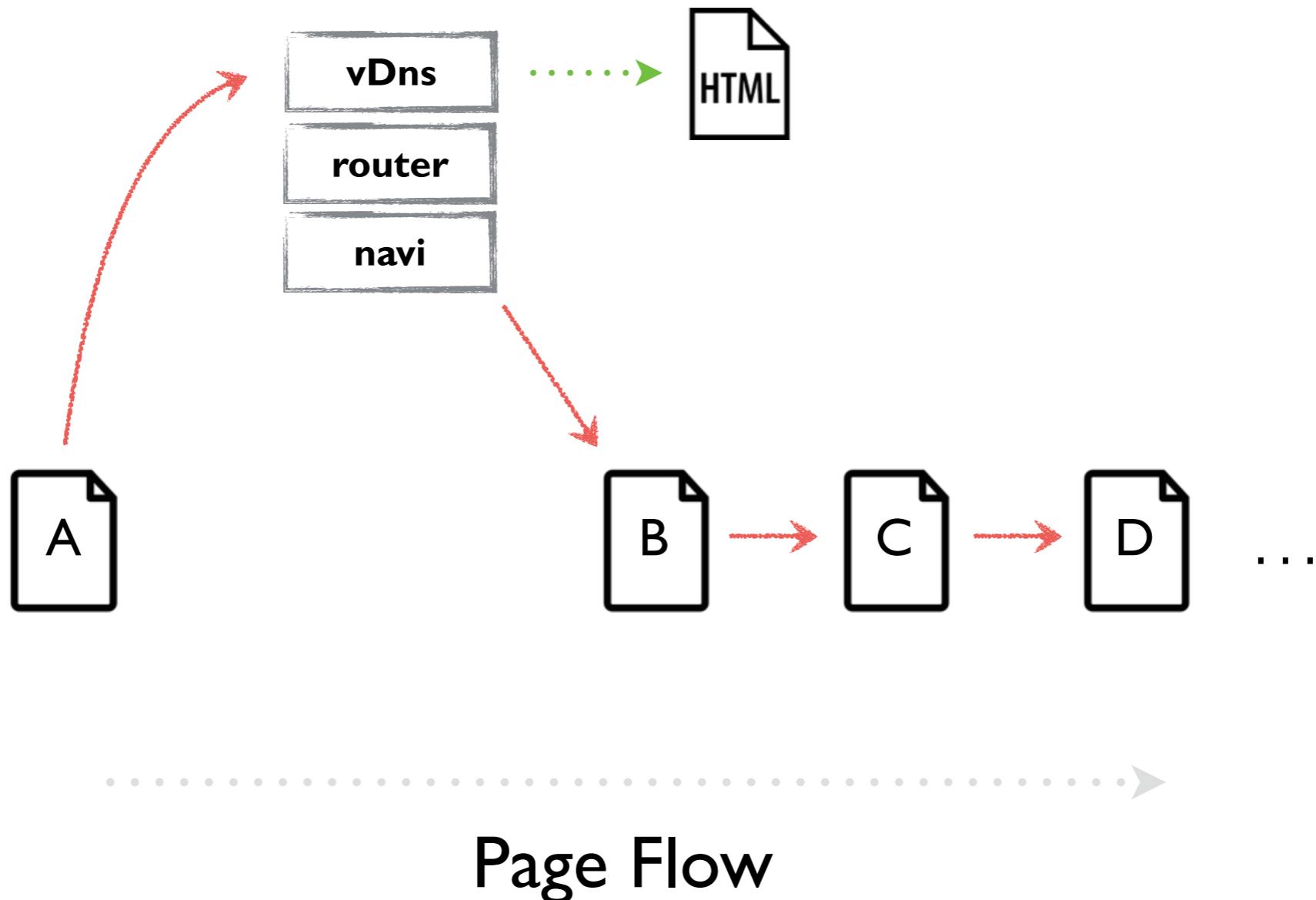
# Native 中 Page 资源定位规范



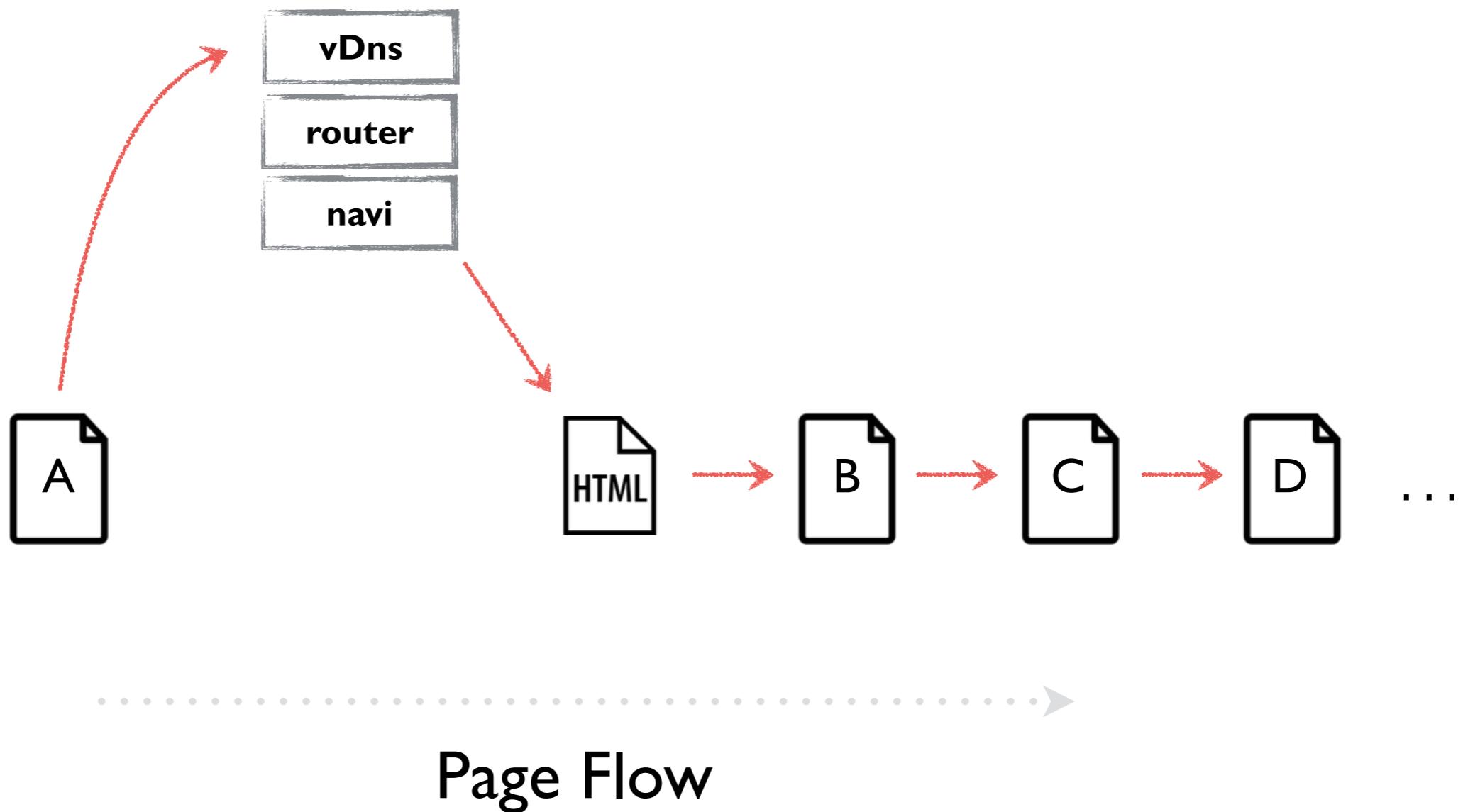
# Native 中 Page 资源定位规范



# Native 中 Page 资源定位规范



# Native 中 Page 资源定位规范

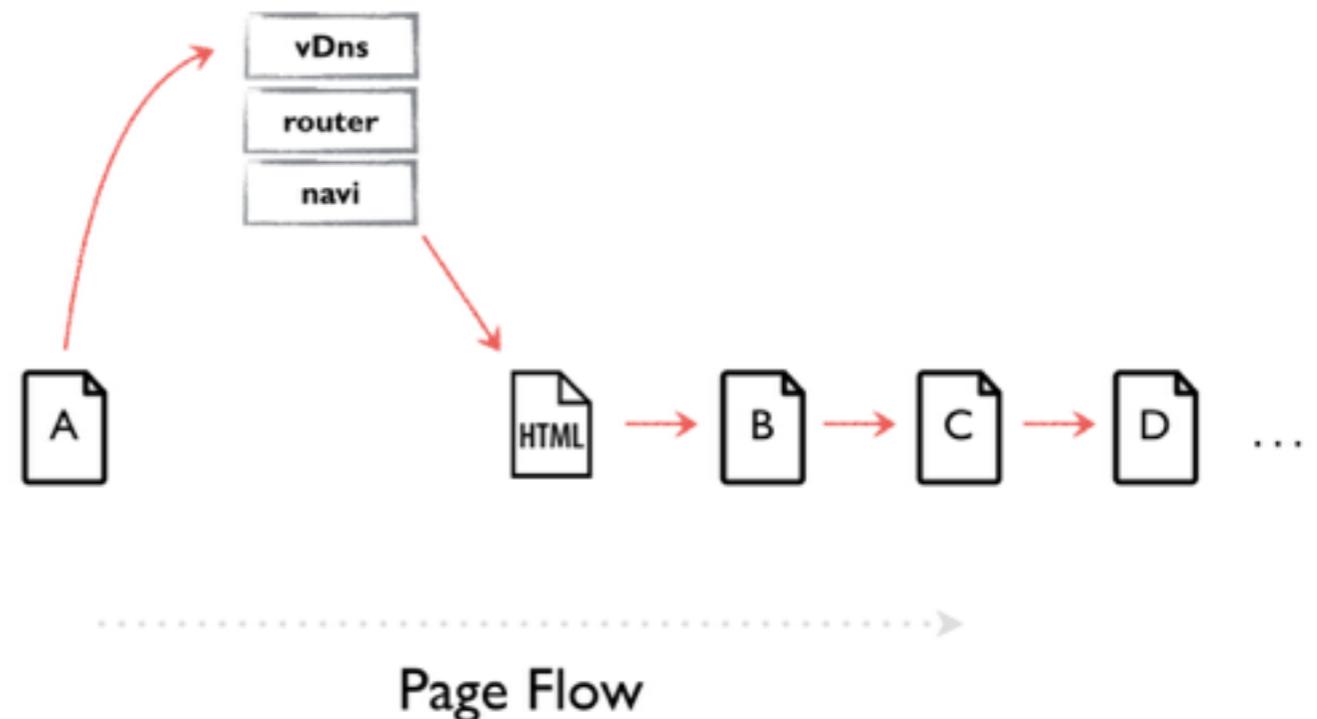


# Native 中 Page 资源定位规范

I. 更加灵活的运营

2. ABTest

3. HotPatch





好的设计驱动技术创新  
好的技术给设计无限想象

# 永不止步的前端

## I. 网页重构、JavaScript 类库、初级性能优化

1. 2006~2010: w3c、YUI、jQuery、prototype、前端代码仓库...
2. 2009~2012: 在 HTTP 和 DOM 渲染层面重新组织资源的加载

## 2. 团队协作、组件化

1. 2010~2013: 页面搭建工具、活动页面模板、店铺装修
2. 2012~2014: commonJS、requireJS、SeaJS、AMD、CMD

## 3. 效率工具、前后端分离

1. 2013~2014: NodeJS、Restful API、理清 BackEnd 和 FrontEnd 的职能边界

## 4. 移动终端、Hybrid

1. 2015: HTML5、CSS3、H5 和 Native 桥接、资源离线化、NativeScript、ReactNative
2. 2015: JSVM、Virtual DNS

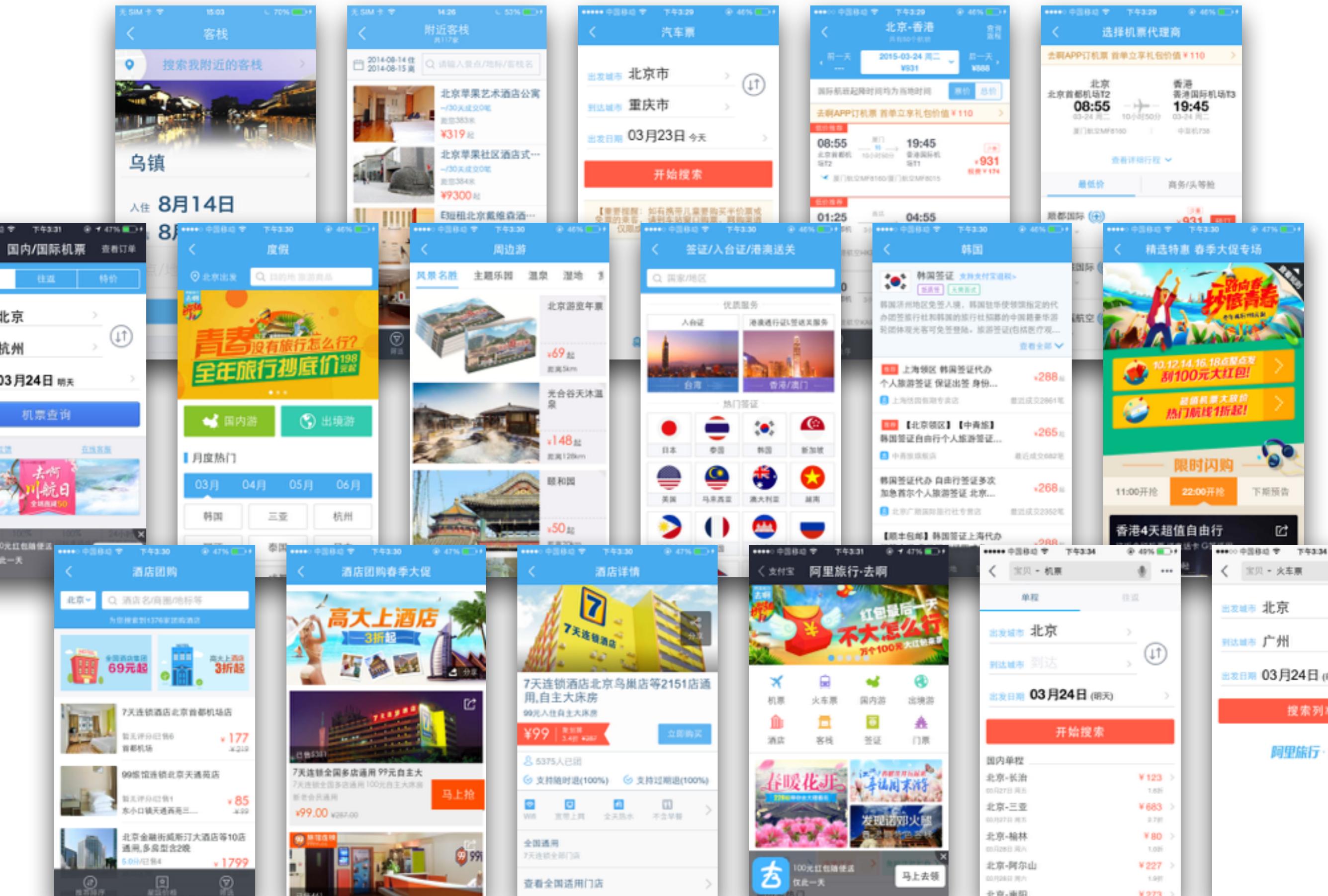
# 原始诉求

- 用户：“快、快、快”——界面快点出来
- PD：“快、快、快”——功能快点上线
- 老板：“快、快、快”——快点回滚问题代码

# 原始诉求

- 用
- PI
- 老

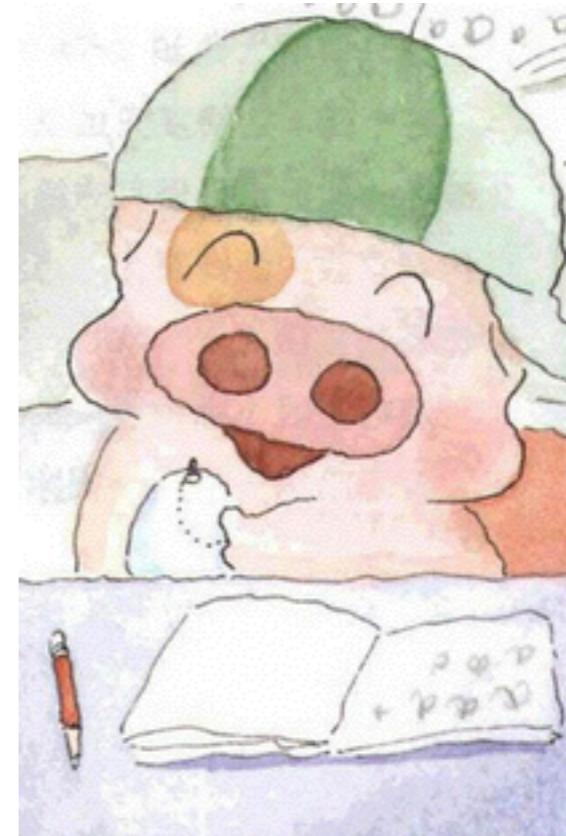




“小”而美的去啊 App

# 阿里旅行·吉啊

我们需要你的加入！



F2E 简历发送至 [bachi@taobao.com](mailto:bachi@taobao.com)



A photograph of a long, straight asphalt road stretching into the distance, framed by two large, dark, snow-capped mountains. The sky is overcast with heavy clouds. The road is marked with white dashed lines and yellow vertical markers. In the foreground, the road surface is visible with some texture and a small puddle on the left side.

Thank You~