QQ会员H5架构优化实践

赖文辉

概要

1、个人简介

2、业务架构要求

3、架构优化实践

4、QA

个人简介

2009年

2010年

Baidu百度

2012年

築龍网
zhulong.com

2013年

QQ会员
VIP.QQ.COM

QQ会员

宇宙第一大包月服务

年轻、个性、酷炫

运营为主

业务架构要求

迭代快➡高效率　体验好➡高性能　人民币➡高质量

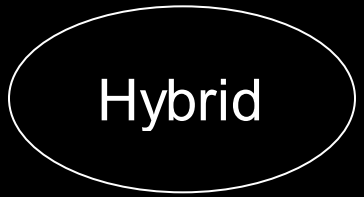
高性能

高性能

Hybrid模式

native app

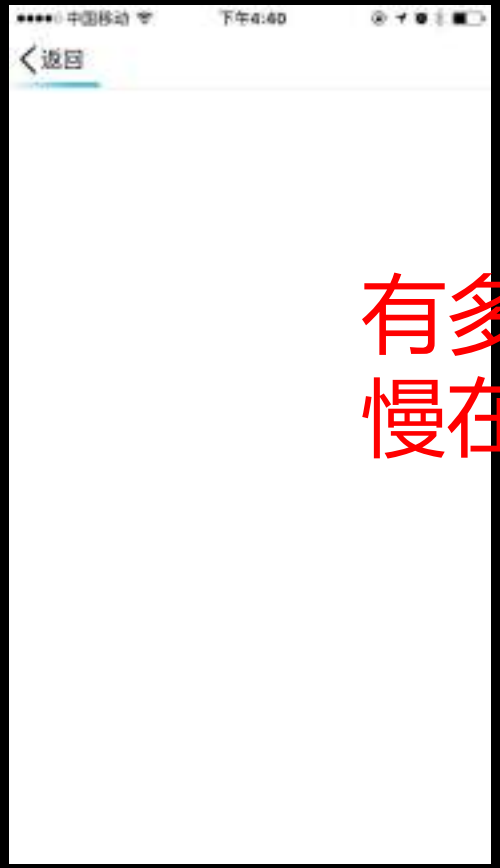
1、接口丰富
2、性能体验好
3、开发成本高
4、迭代周期长



web app

1、系统接口少
2、性能体验差
3、开发成本低
4、快速迭代

Hybrid模式最大的挑战

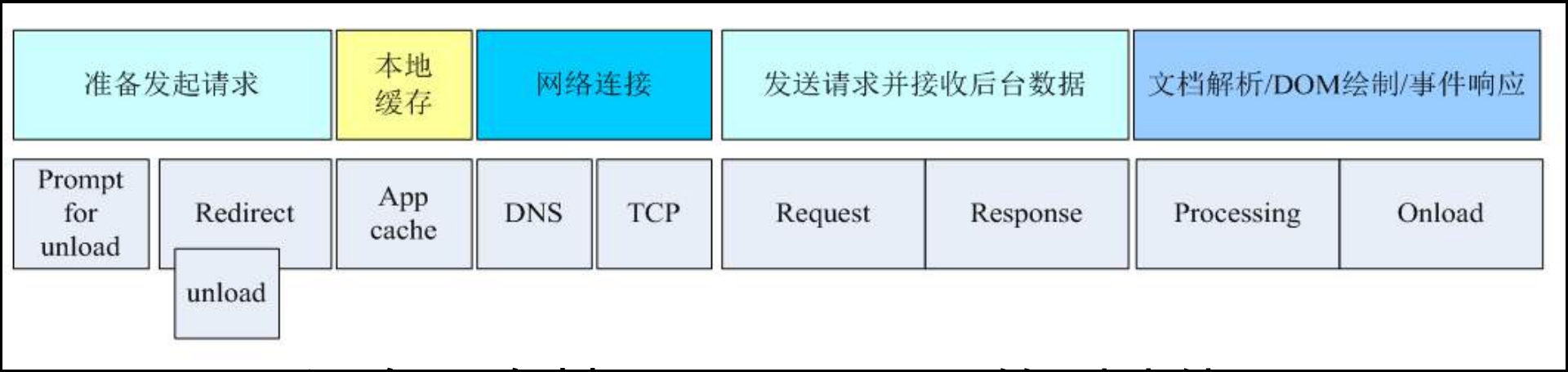


有多慢？
慢在哪里？

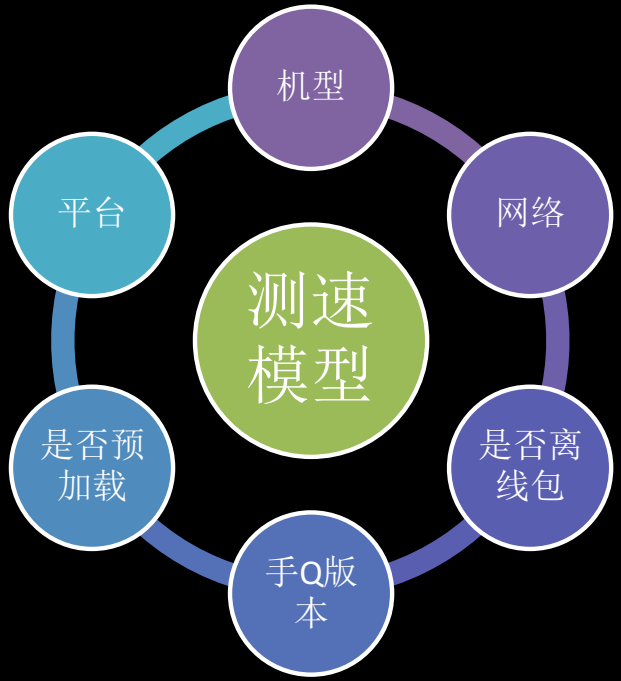
缺乏统一的测速体系

测速点	指标	权重	上报量	延时(s)
loadHtml	是	10	74519	0.234716
loadCss	否	0	74308	0.173813
首屏	否	0	74565	2
loadJS	否	0	74562	0.64021
页面加载完成	否	0	74561	0.876145

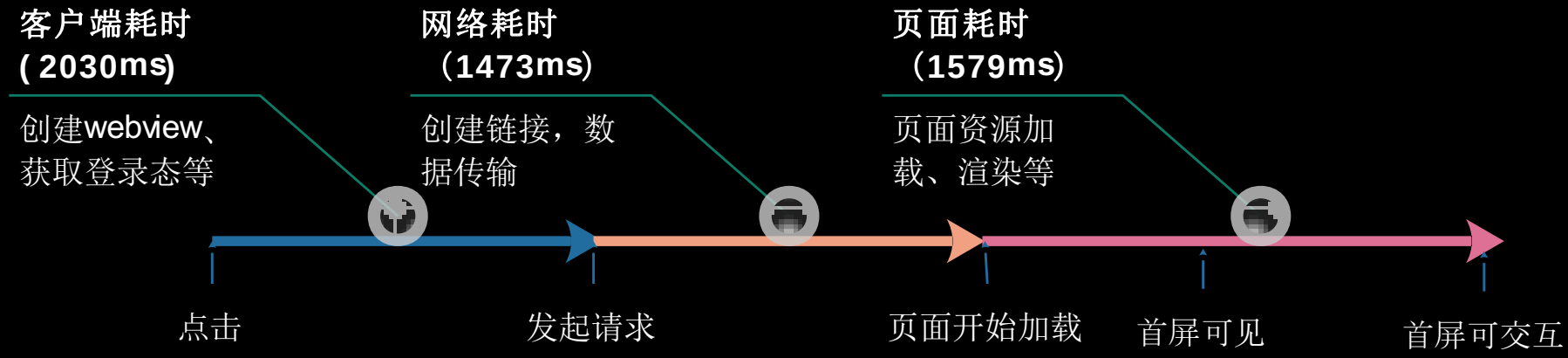
缺乏统一的测速体系


准备发起请求
本地缓存
网络连接
发送请求并接收后台数据
文档解析/DOM绘制/事件响应
Prompt for unload
Redirect
unload
App cache
DNS
TCP
Request
Response
Processing
Onload

Hybird测速体系建设



Hybrid页面耗时(android)

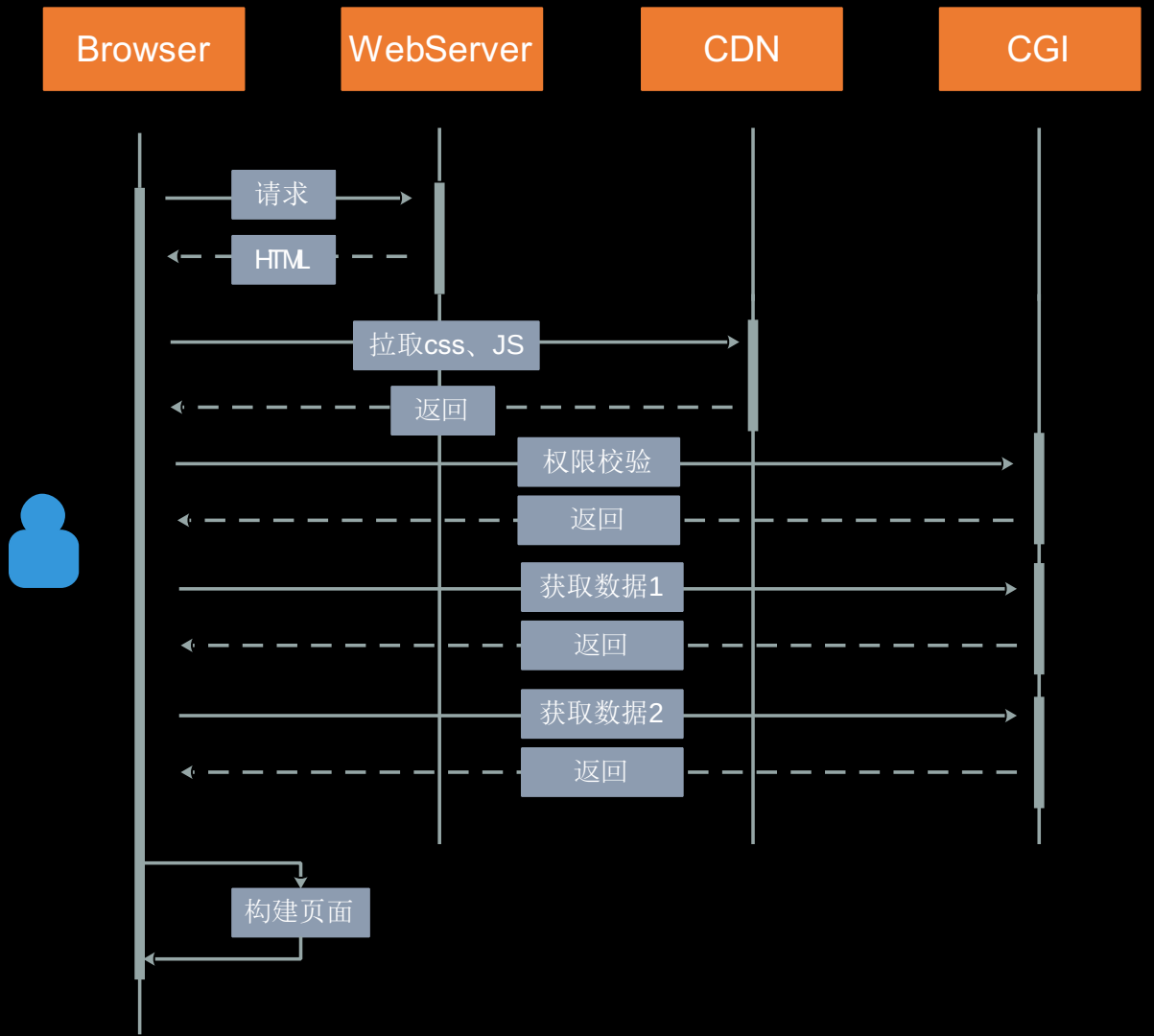


性能优化1.0

前端渲染

JS逻辑重

CGI请求多

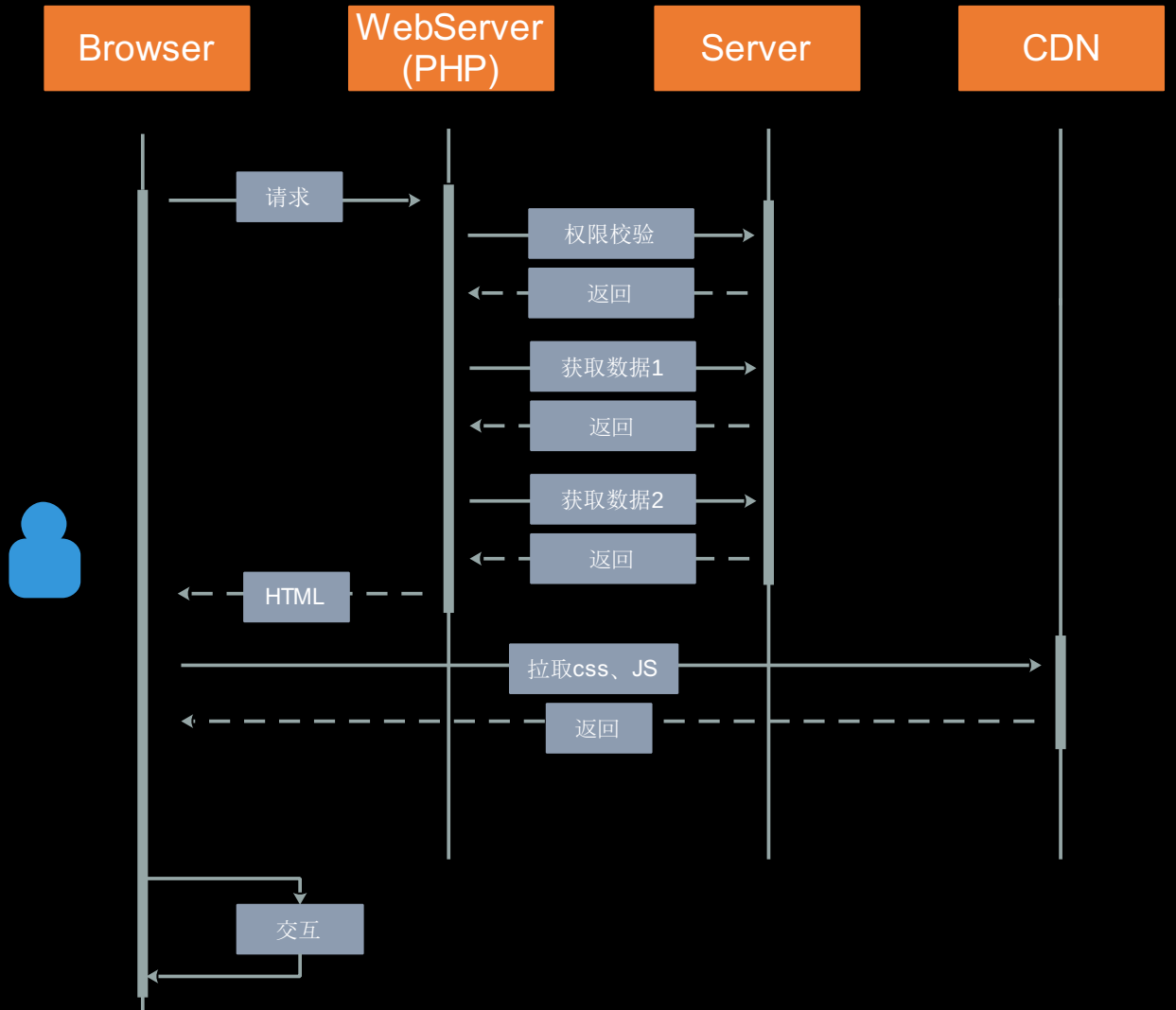


性能优化2.0

服务端渲染

JS代码减少

数据服务端获取

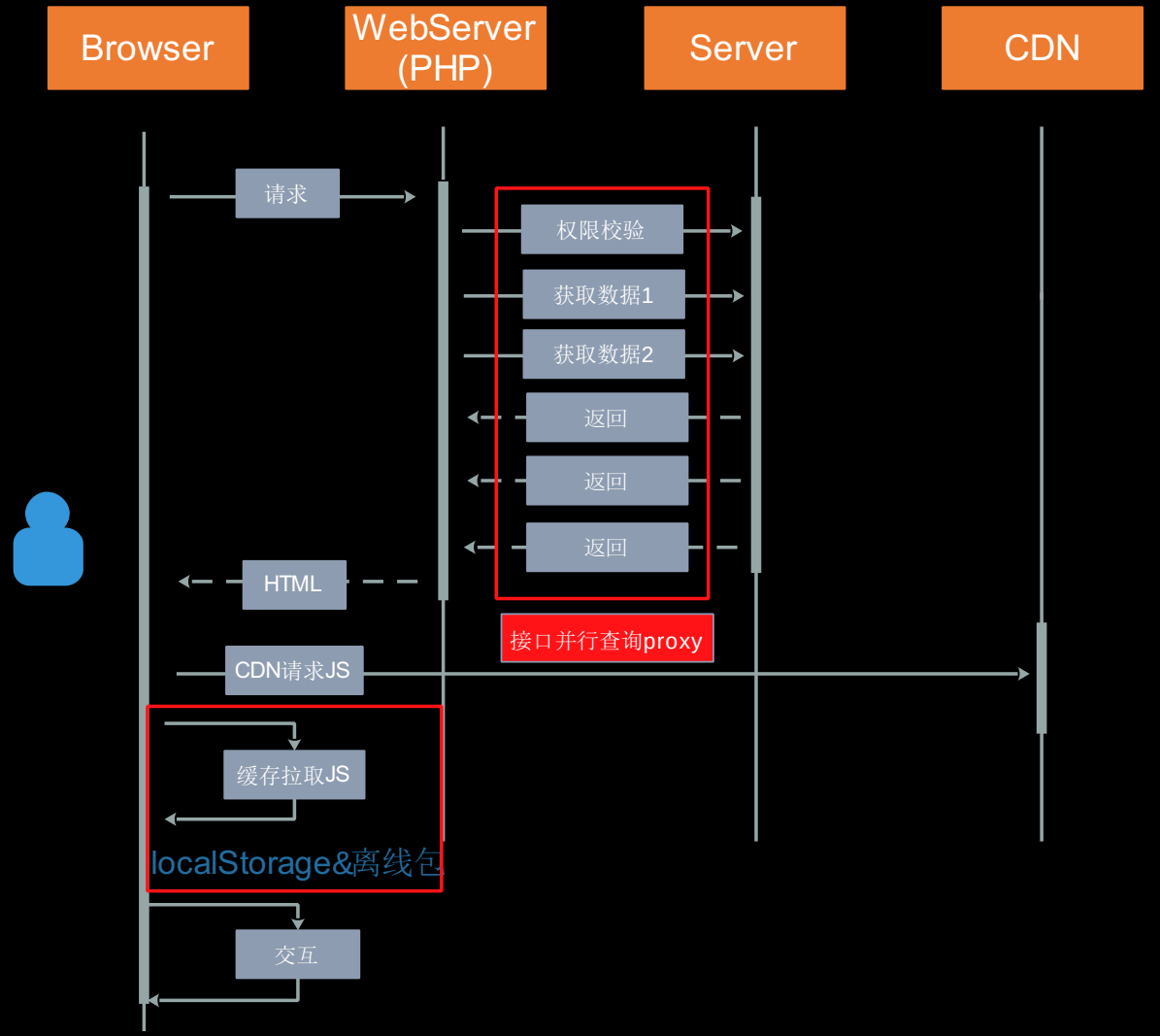


性能优化3.0

数据查询并行化

JS localStorage缓存

公共类库复用&离线

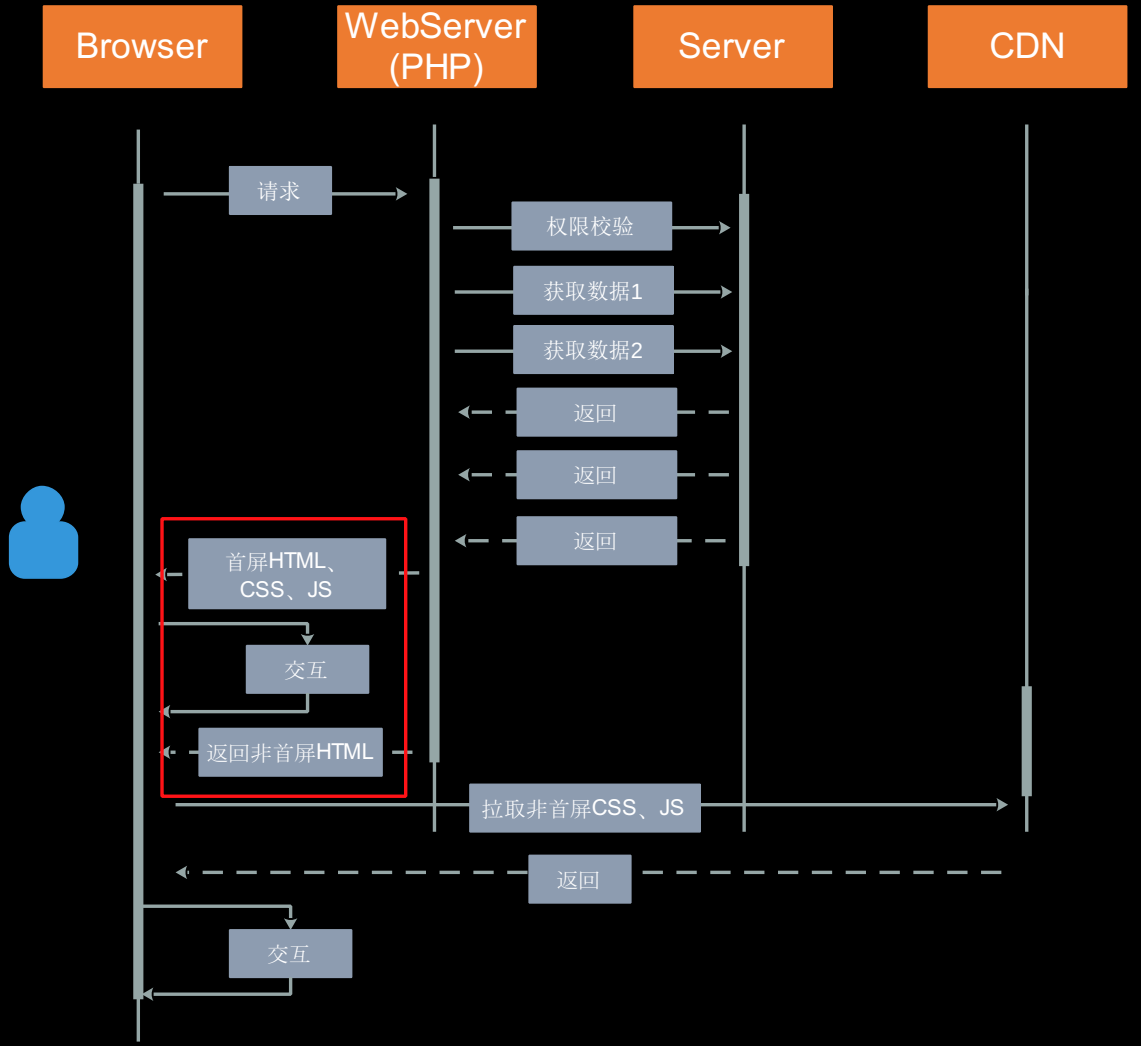


性能优化4.0

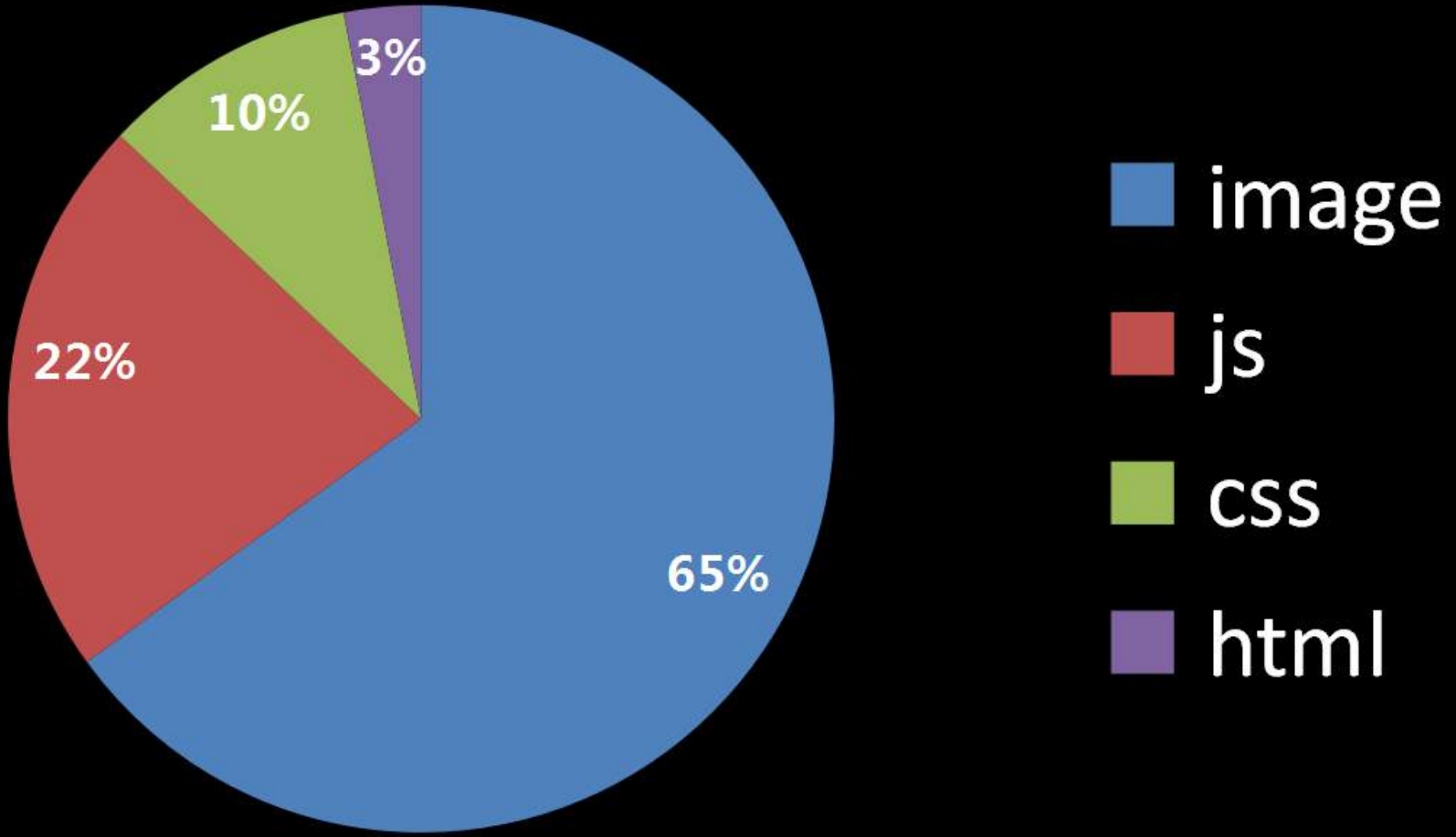
首屏分离，先加载首屏

首屏去除类库依赖

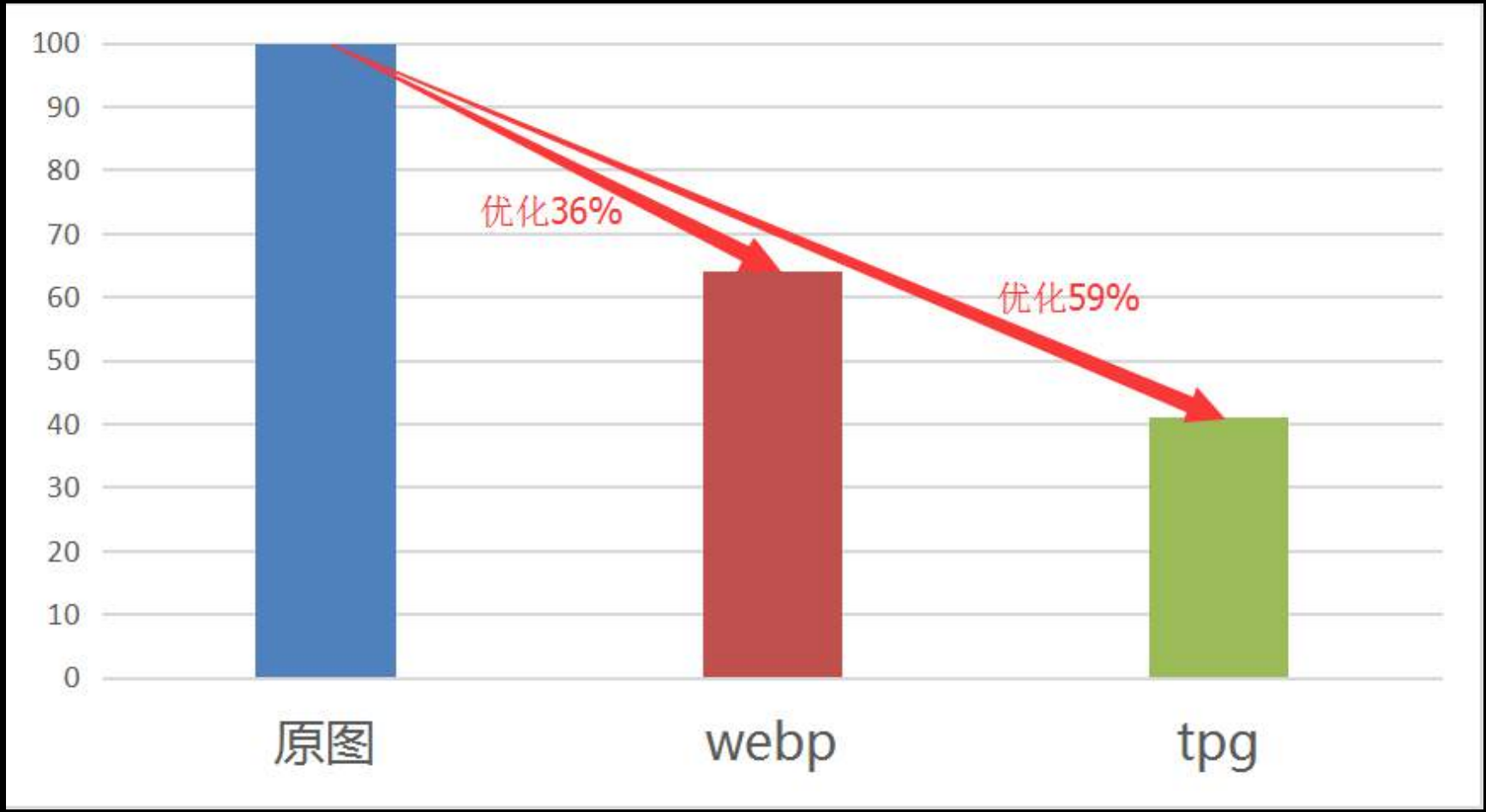
首屏依赖js\css打入页面



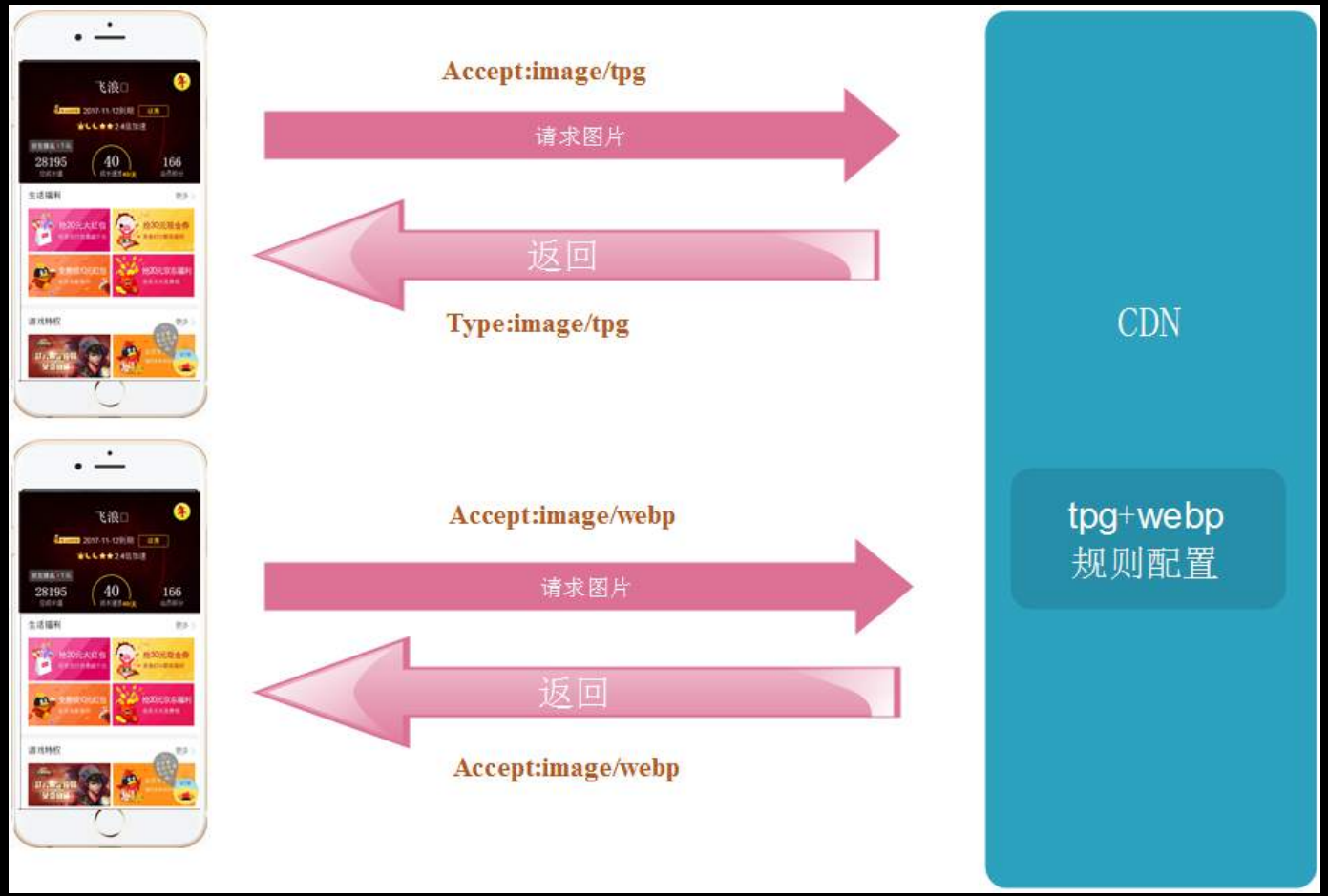
图片优化



图片格式优化--接入成本高



新图片格式--页面透明接入方案



更进一步-图片自适应


750px*200px
QQ会员
京东购物特权
震撼上线
750px*200px
640px*170px
480px*128px
750px
640px
480px

图片自适应--页面透明接入方案

Accept:image/tpg
User-Agent:Pixel/750

请求图片

返回

Size:100px*100px
Type:image/tpg

CDN
（100px*100px）

tpg+自适应
规则配置

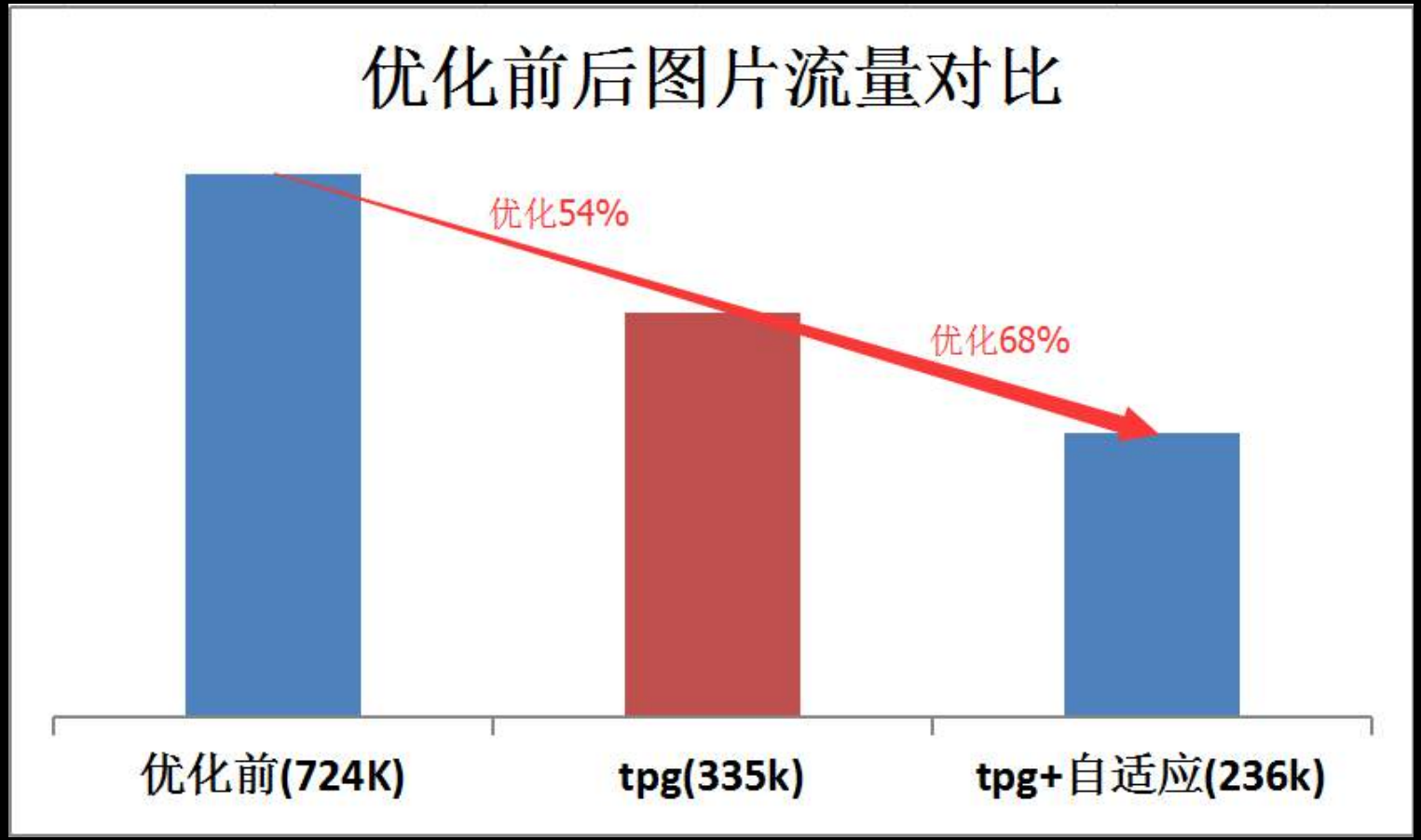
Accept:image/tpg
User-Agent:Pixel/640

请求图片

返回

Size:85px*85px
Type:image/tpg

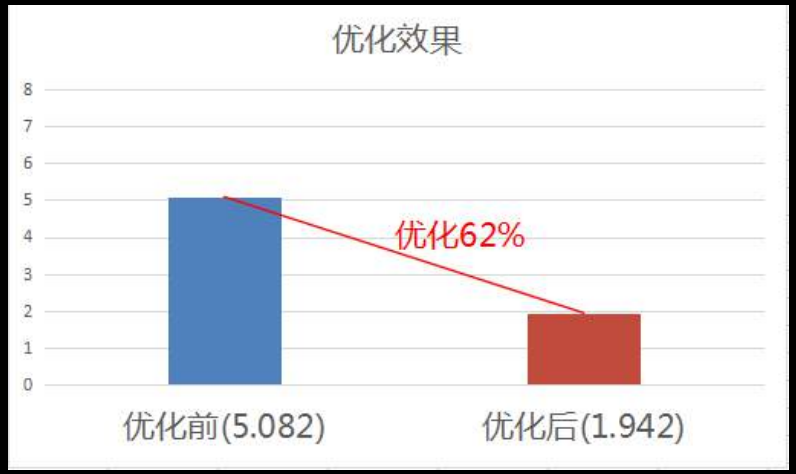
优化效果



性能优化总结


性能优化
客户端优化
webview并行初始化
离线包优化
webview预加载
网络加载优化
减少请求数、请求内容
数据查询并行化
网络加速
页面优化
雅虎前端优化35条
首屏分离
减少类库依赖

性能后效果



客户端耗时 (691ms)

创建webview、获取登录态等

网络耗时 （859ms）

创建链接，数据传输

页面耗时 （392ms）

页面资源加载、渲染等

点击

发起请求

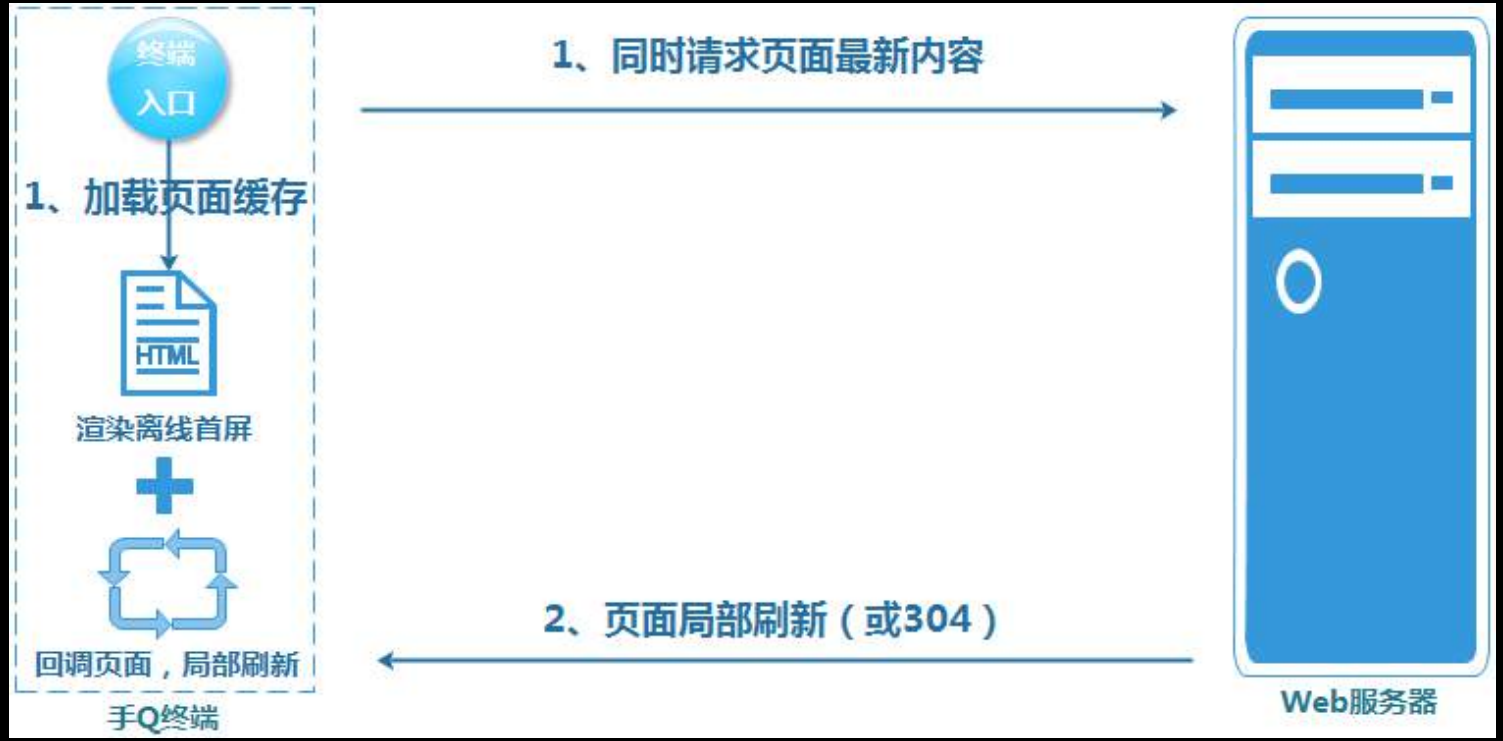
页面开始加载

首屏可见

首屏可交互

还能怎么优化：sonic

sonic是一种动态页面缓存方案



数据模版分离

HTML

```
<html>
<head><style></style></head>
<body>
    <!--wnsdiff-key1--><div id="key1"><div>XXX1</div></div><!--wnsdiff-key1-end-->
    <!--wnsdiff-key2--><div id="key2"><div>XXX2</div></div><!--wnsdiff-key2-end-->
    <!--wnsdiff-key3--><div id="key3"><div>XXX3</div></div><!--wnsdiff-key3-end-->
</body><script></script>
</html>
```

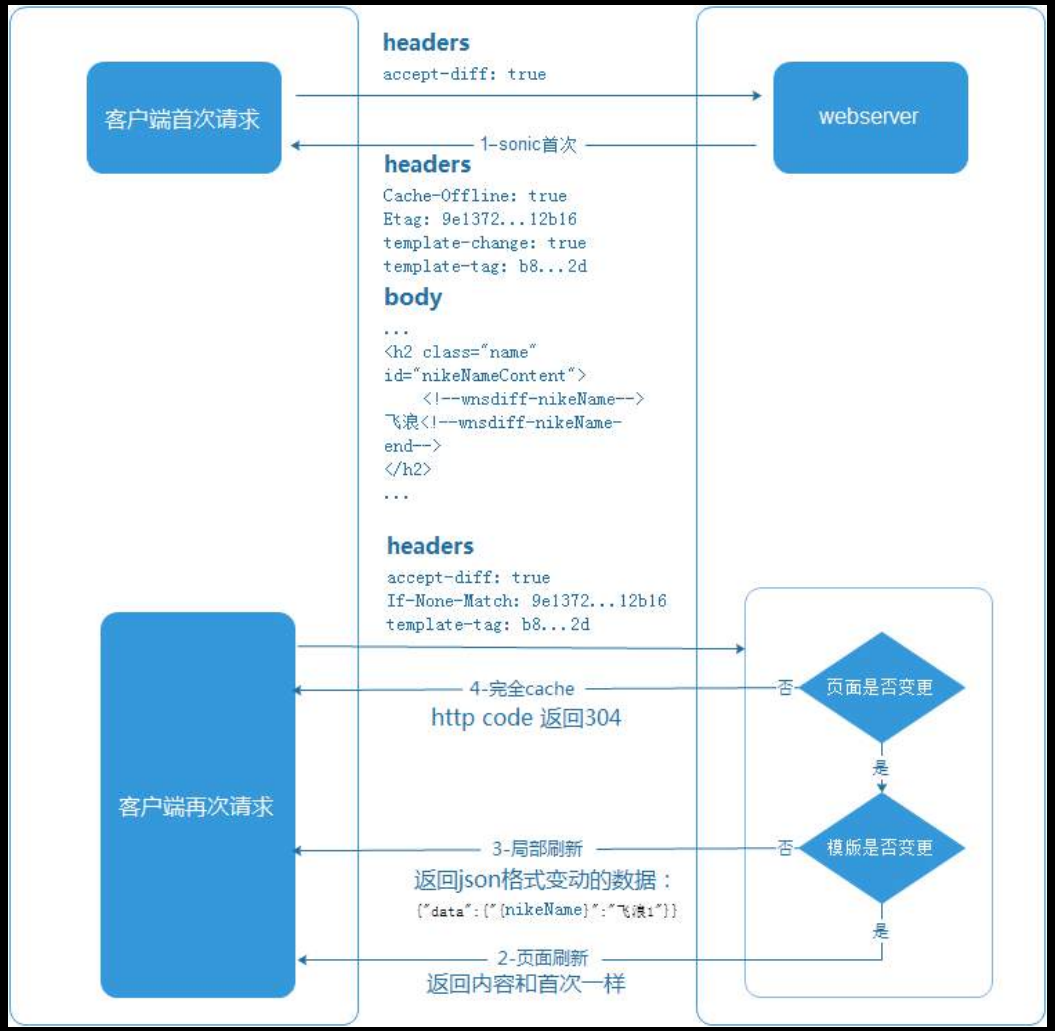
Template

```
<html>
<head><style></style></head>
<body>
    <!--wnsdiff-key1--><div id="key1"> </div><!--wnsdiff-key1-end-->
    <!--wnsdiff-key2--><div id="key2"> </div><!--wnsdiff-key2-end-->
    <!--wnsdiff-key3--><div id="key3"> </div><!--wnsdiff-key3-end-->
</body><script></script>
</html>
```

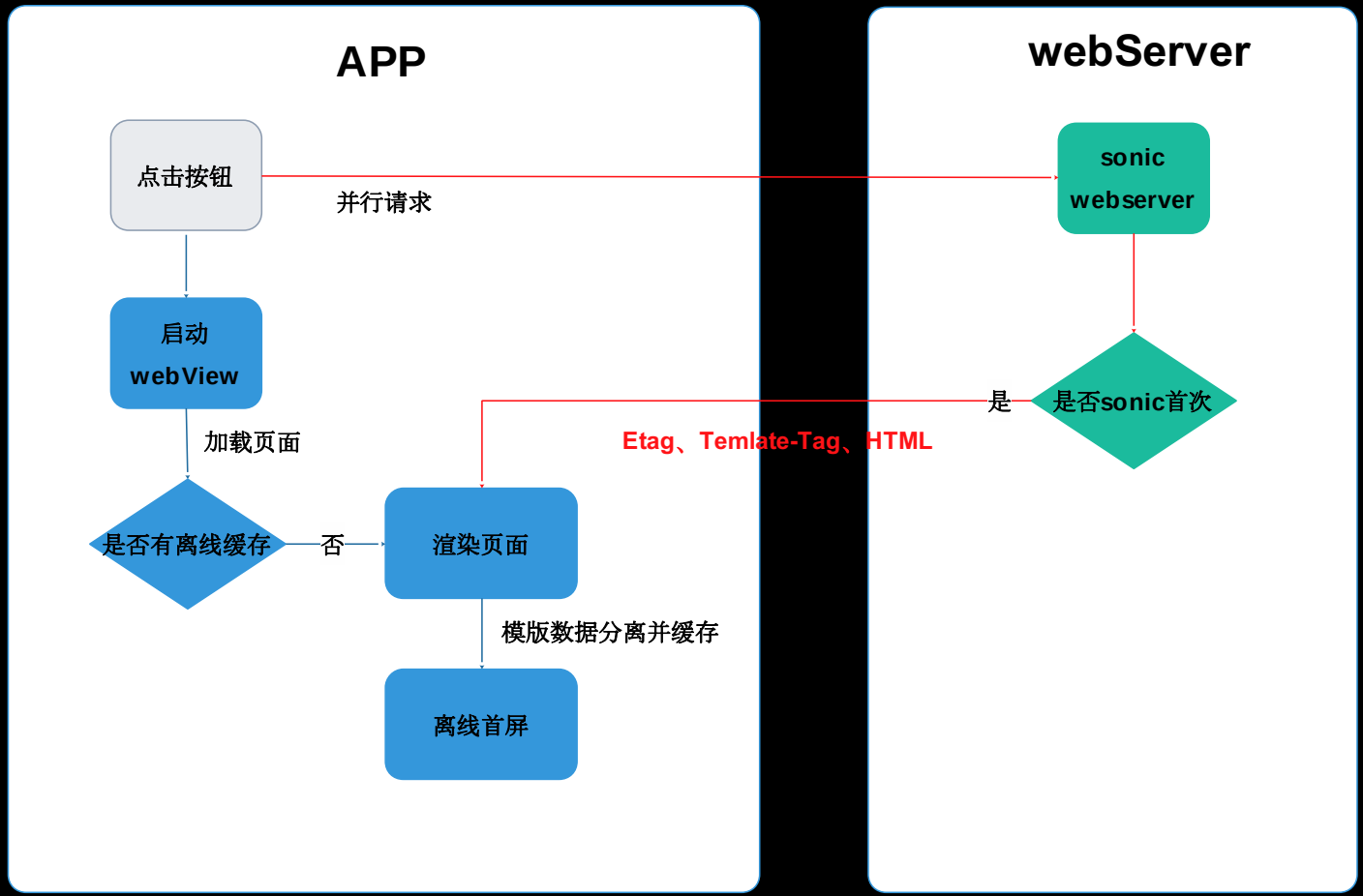
Data

```
{
  "{key1}":"<div>XXX1</div>",
  "{key2}":"<div>XXX2</div>",
  "{key3}":"<div>XXX3</div>"
}
```

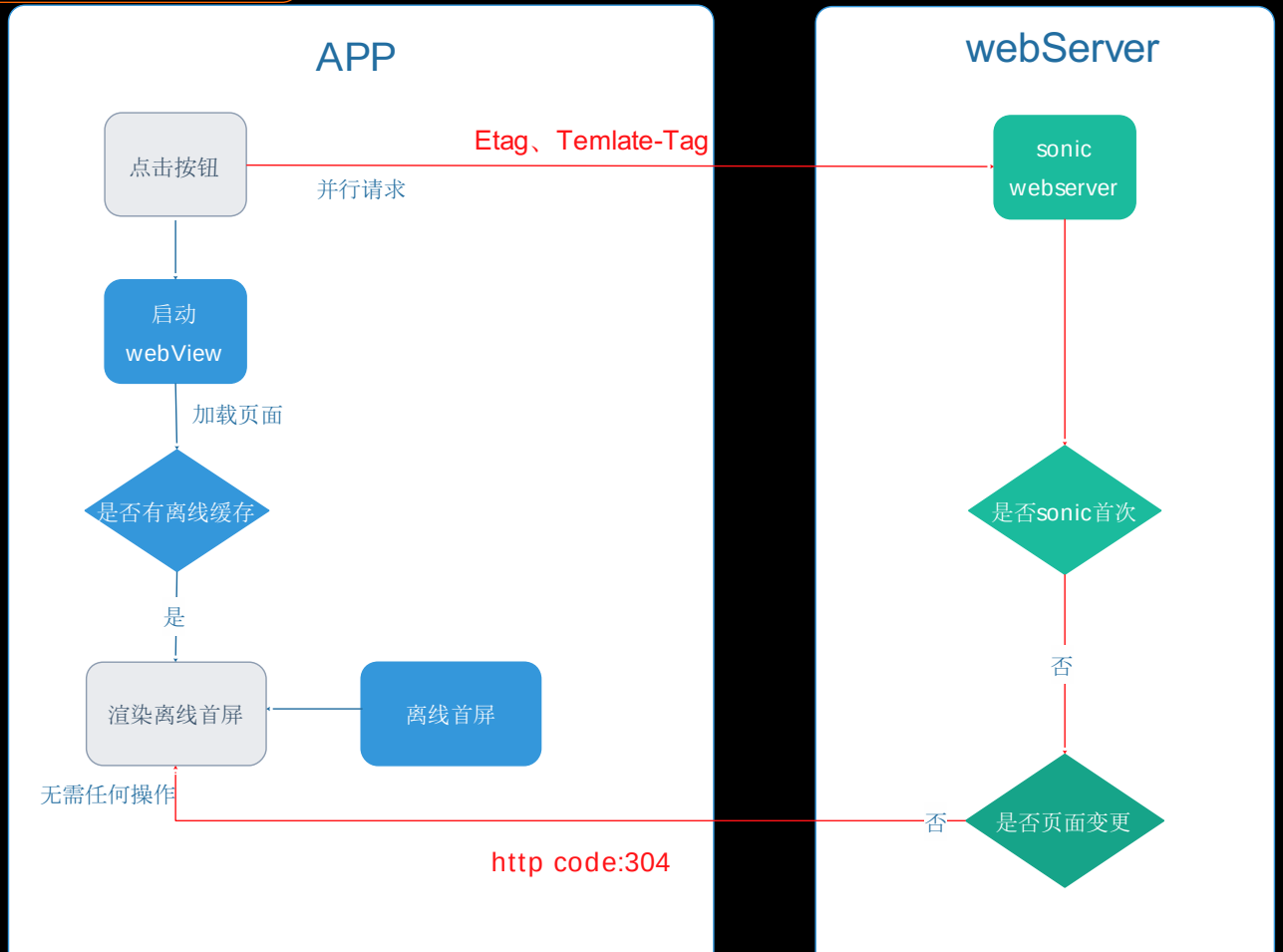
前后端交互协议


客户端首次请求
headers
accept-diff: true
webserver
1-sonic首次
headers
Cache-Offline: true
Etag: 9e1372...12b16
template-change: true
template-tag: b8...2d
body
...
<h2 class="name"
id="nikeNameContent">
<!--wnsdiff-nikeName-->
飞浪<!--wnsdiff-nikeName-
end-->
</h2>
...
headers
accept-diff: true
If-None-Match: 9e1372...12b16
template-tag: b8...2d
客户端再次请求
4-完全cache
http code 返回304
否
页面是否变更
是
否
模版是否变更
是
3-局部刷新
返回json格式变动的数据：
{"data":{"{nikeName}":"飞浪1"}}
2-页面刷新
返回内容和首次一样

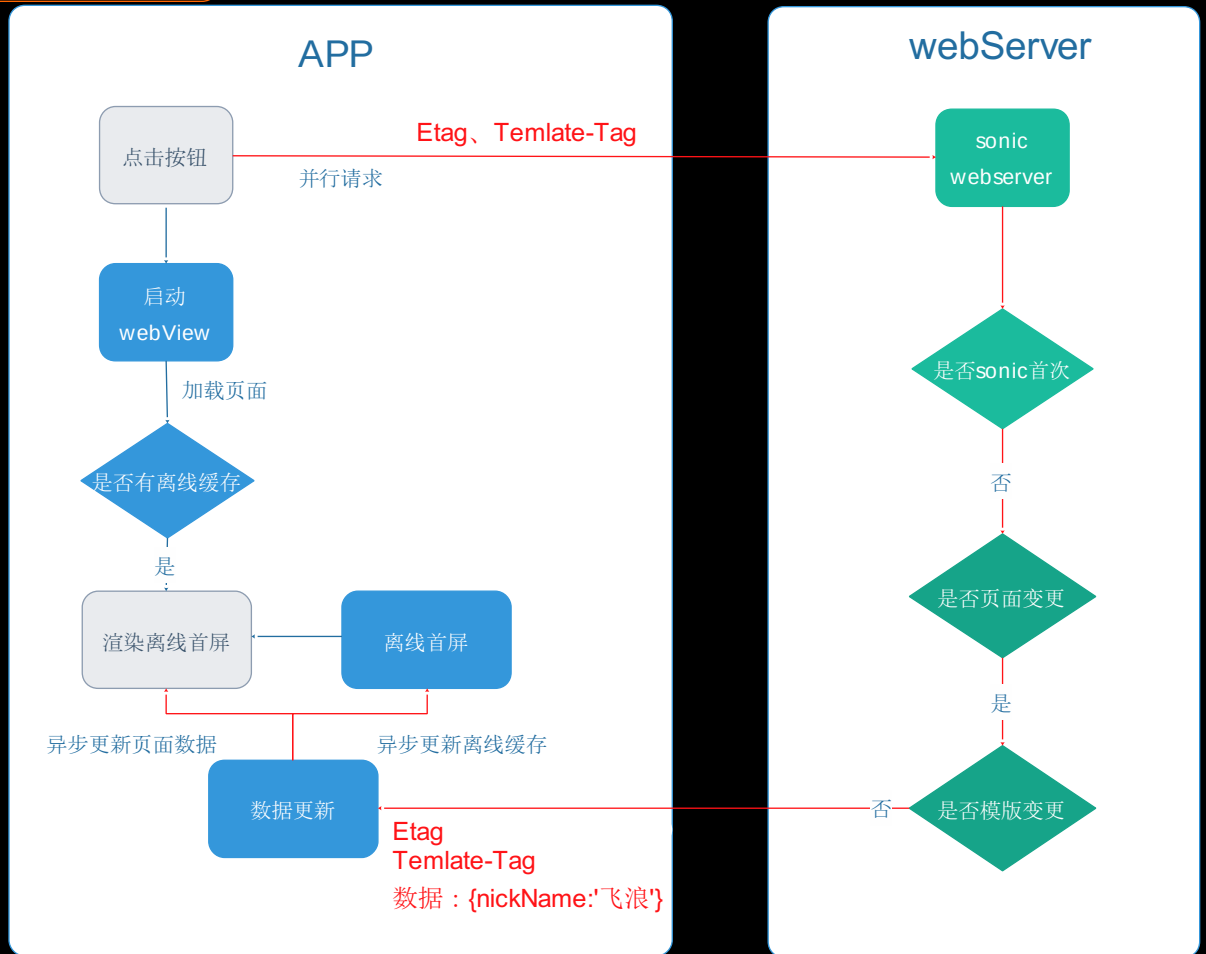
sonic首次


APP
webServer
点击按钮
并行请求
sonic webserver
启动 webView
加载页面
是否sonic首次
是
Etag、Temlate-Tag、HTML
是否有离线缓存
否
渲染页面
模版数据分离并缓存
离线首屏

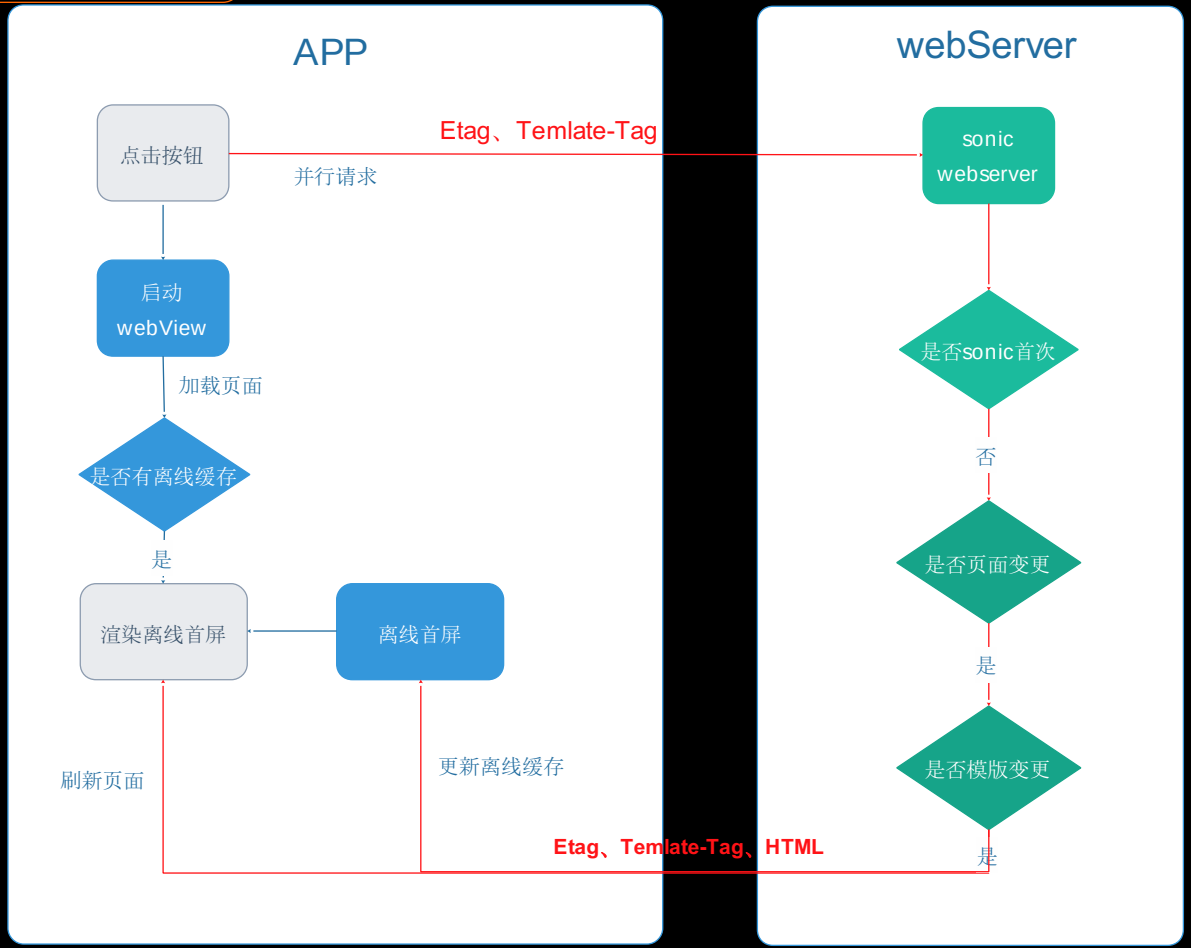
sonic页面无变更


APP
webServer
点击按钮
Etag、Temlate-Tag
并行请求
sonic webserver
启动 webView
加载页面
是否有离线缓存
是否sonic首次
是
否
渲染离线首屏
离线首屏
无需任何操作
否
是否页面变更
http code:304

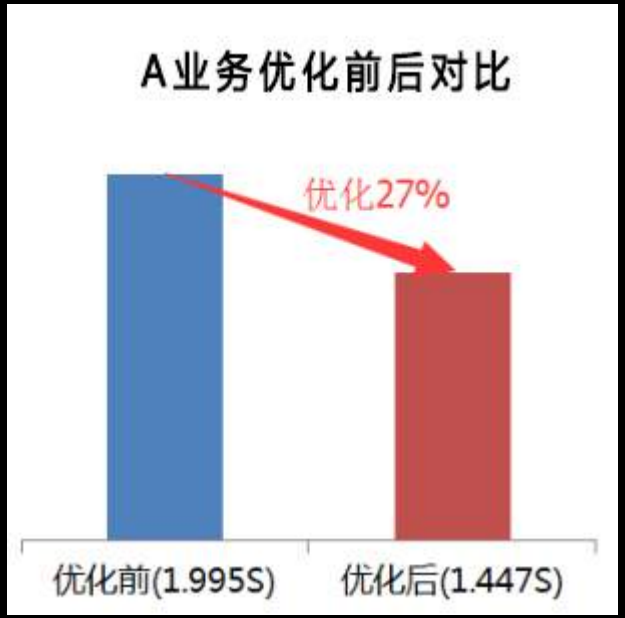
sonic数据更新

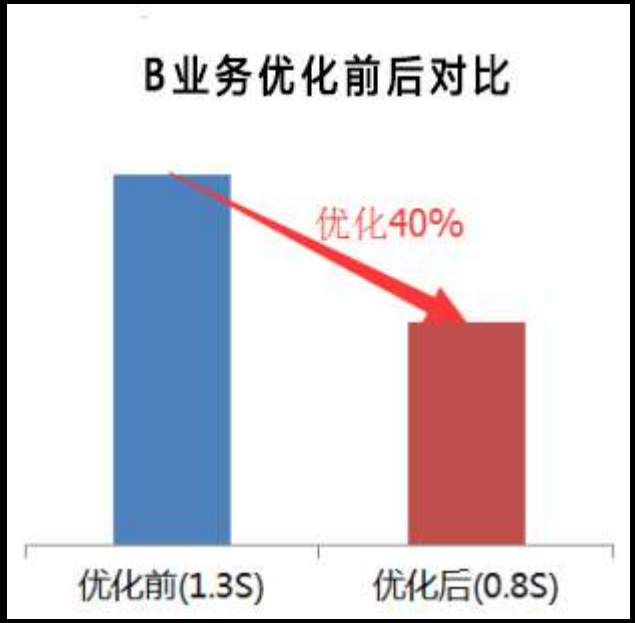

APP
webServer
点击按钮
Etag、Temlate-Tag
并行请求
sonic webserver
启动 webView
加载页面
是否sonic首次
是否有离线缓存
否
是
是否页面变更
渲染离线首屏
离线首屏
是
异步更新页面数据
异步更新离线缓存
数据更新
否
是否模版变更
Etag
Temlate-Tag
数据：{nickName:'飞浪'}

sonic模版更新


APP
webServer
点击按钮
Etag、Temlate-Tag
并行请求
sonic webserver
启动 webView
加载页面
是否sonic首次
否
是否有离线缓存
是
是否页面变更
渲染离线首屏
离线首屏
是
是否模版变更
刷新页面
更新离线缓存
Etag、Temlate-Tag、HTML
是

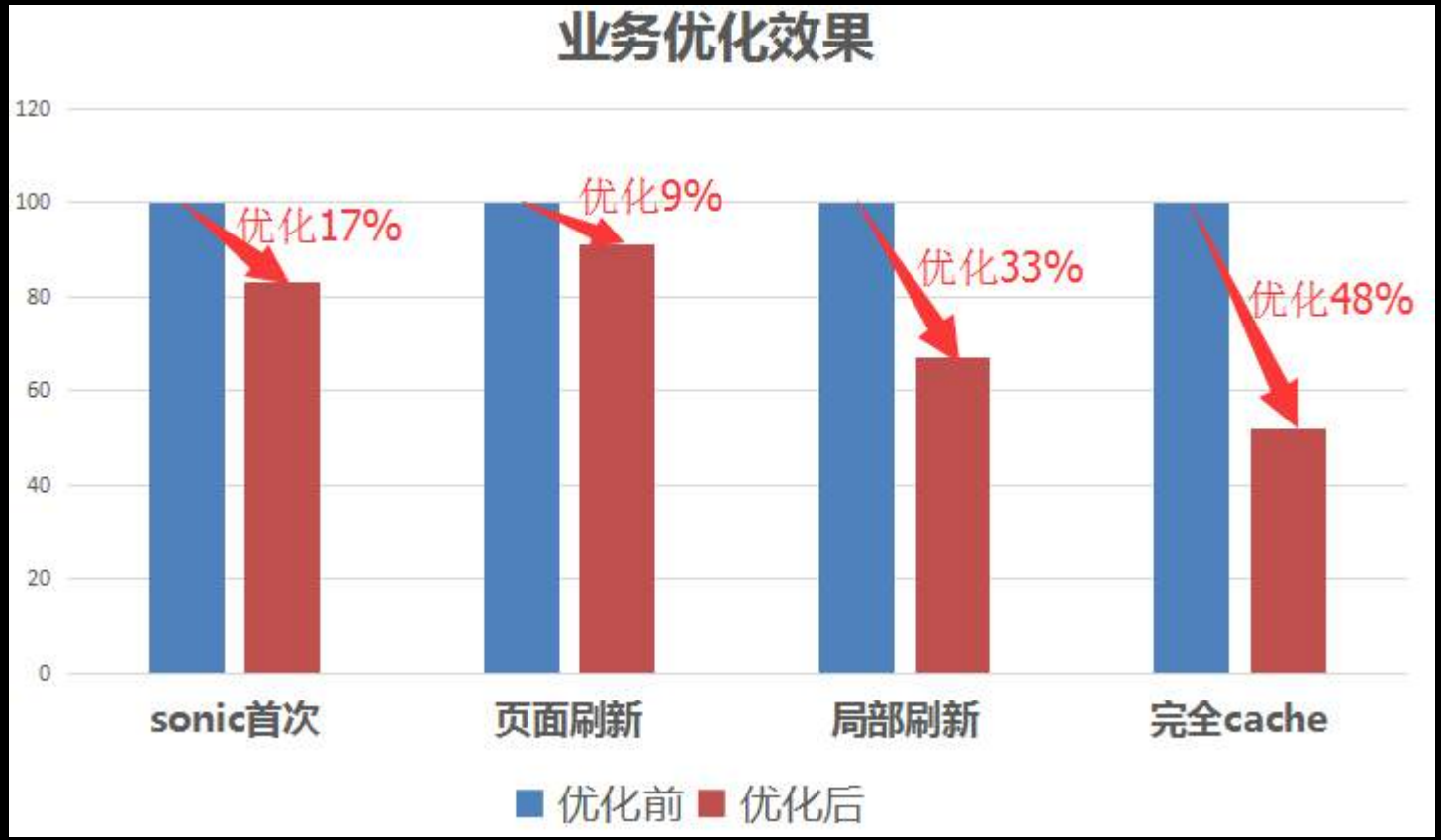
sonic优化效果





页面变更频率越低，优化越明显

sonic各状态优化效果



提升sonic缓存率，减少刷新场景

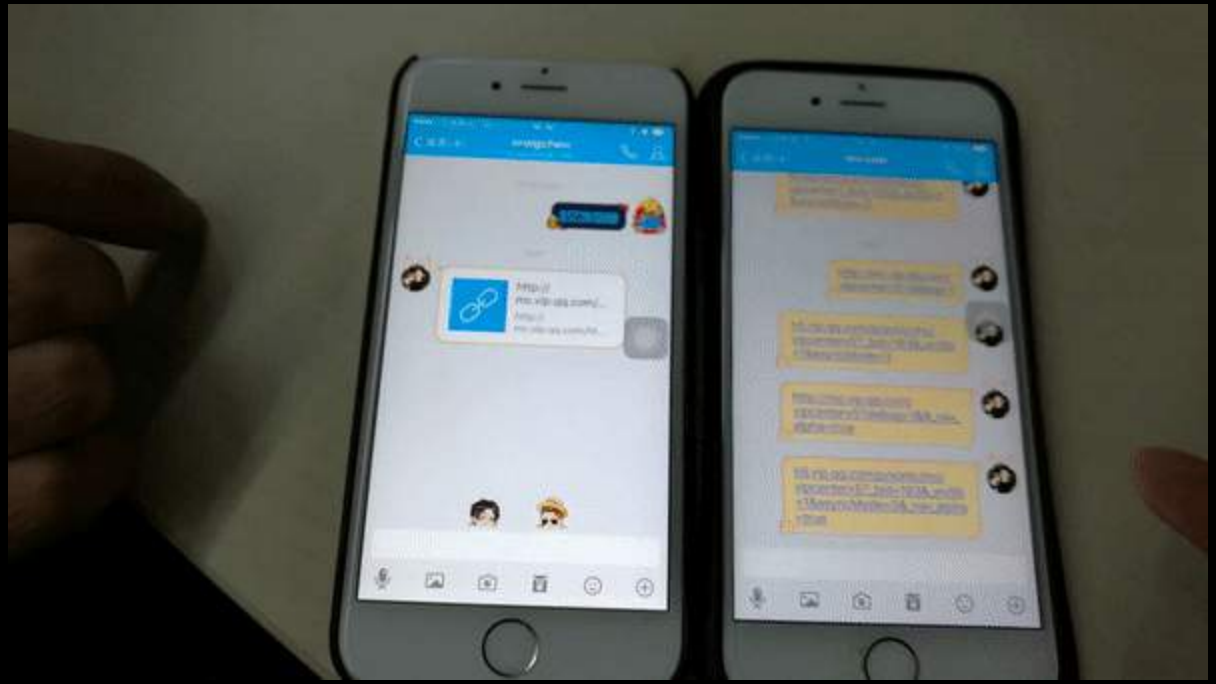
1、模版清除功能

2、sonic预加载

3、合理拆分模版和数据，尽量减少模版变更

sonic优化后效果

在有cache的情况下，实现和native接近的秒开体验



高效率

高效率

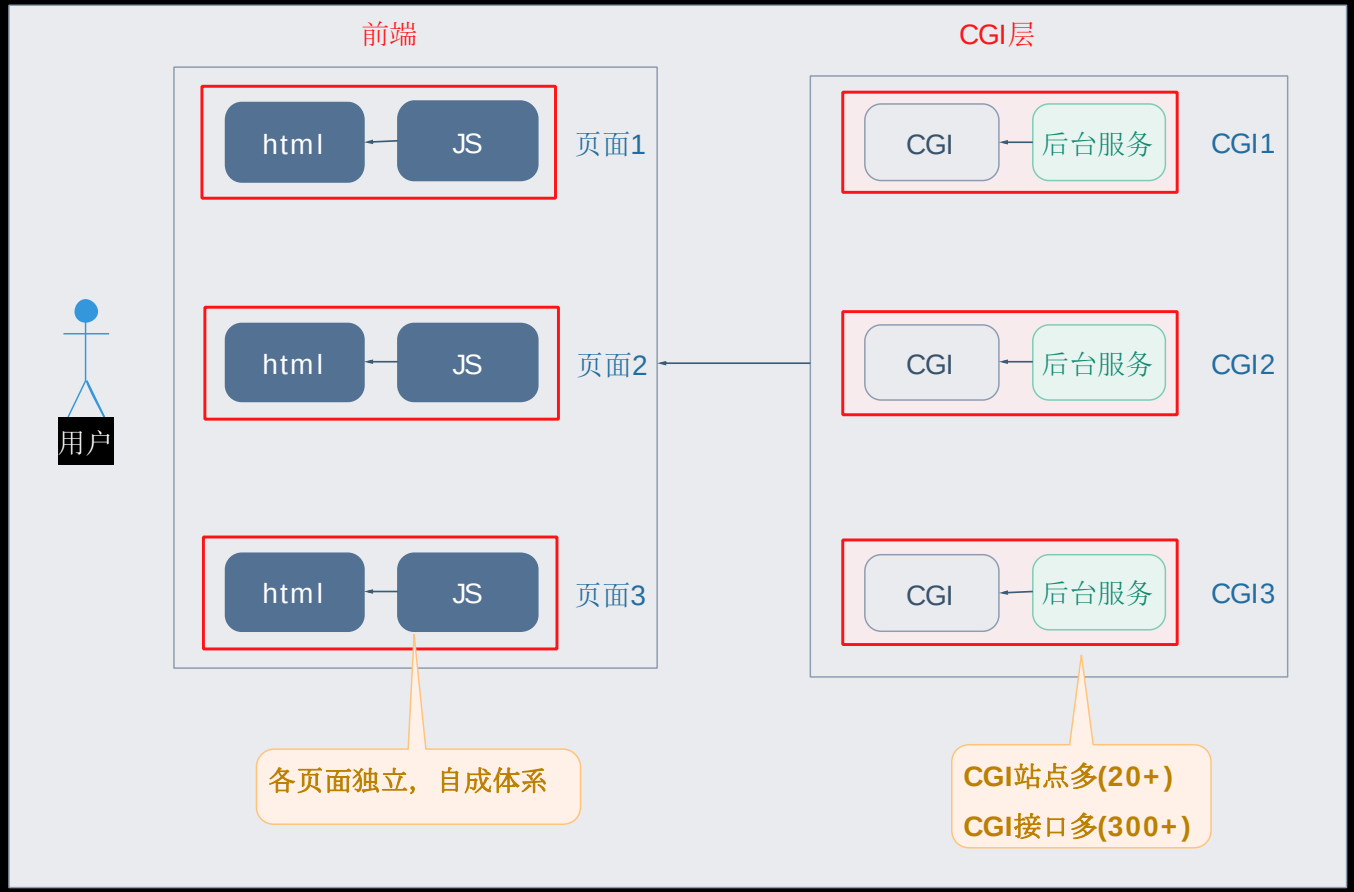
降低开发成本，提升系统可维护性

单页面应用

页面独立

开发周期长

维护困难

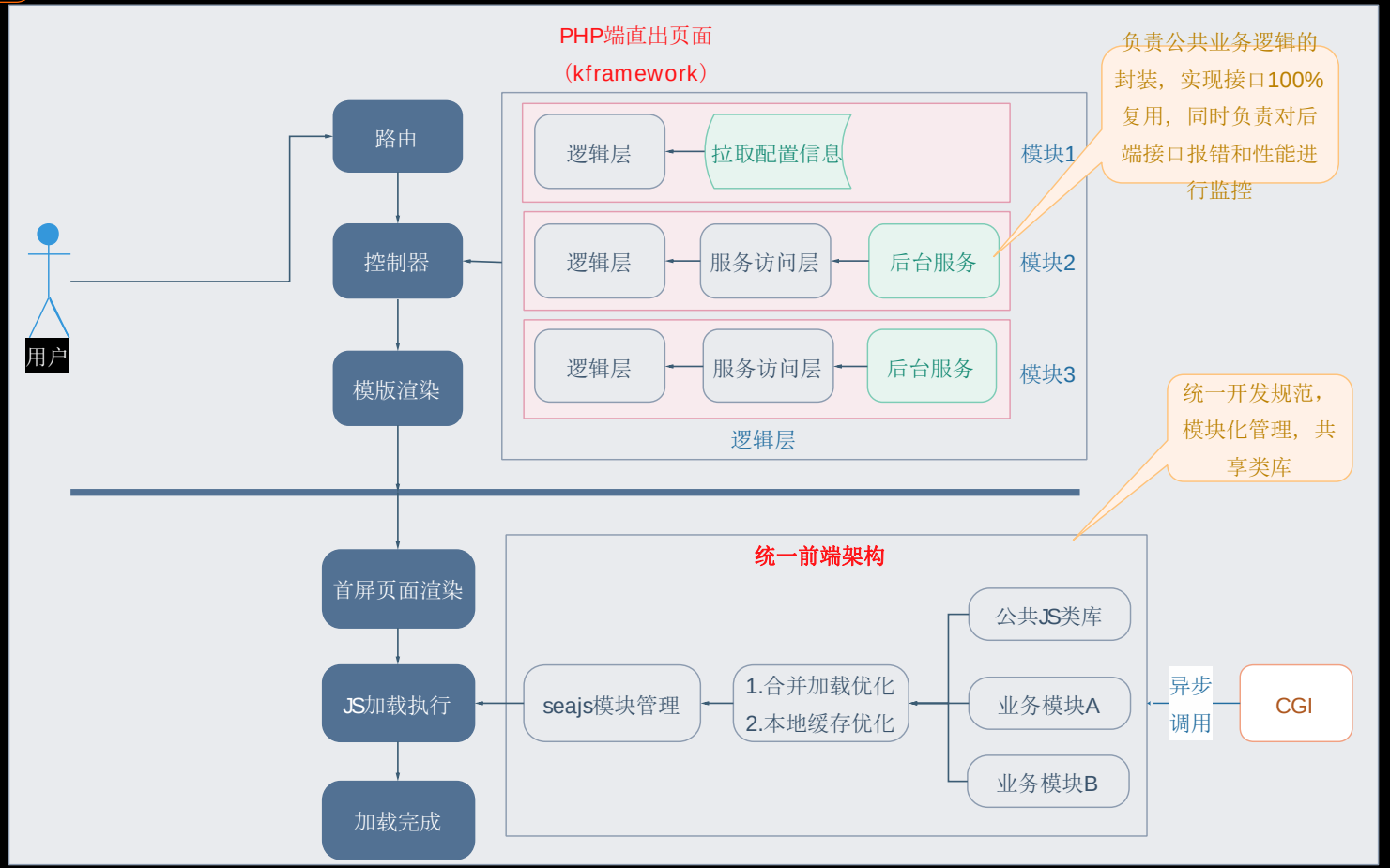


框架规范开发

KISS原则

模块化解耦

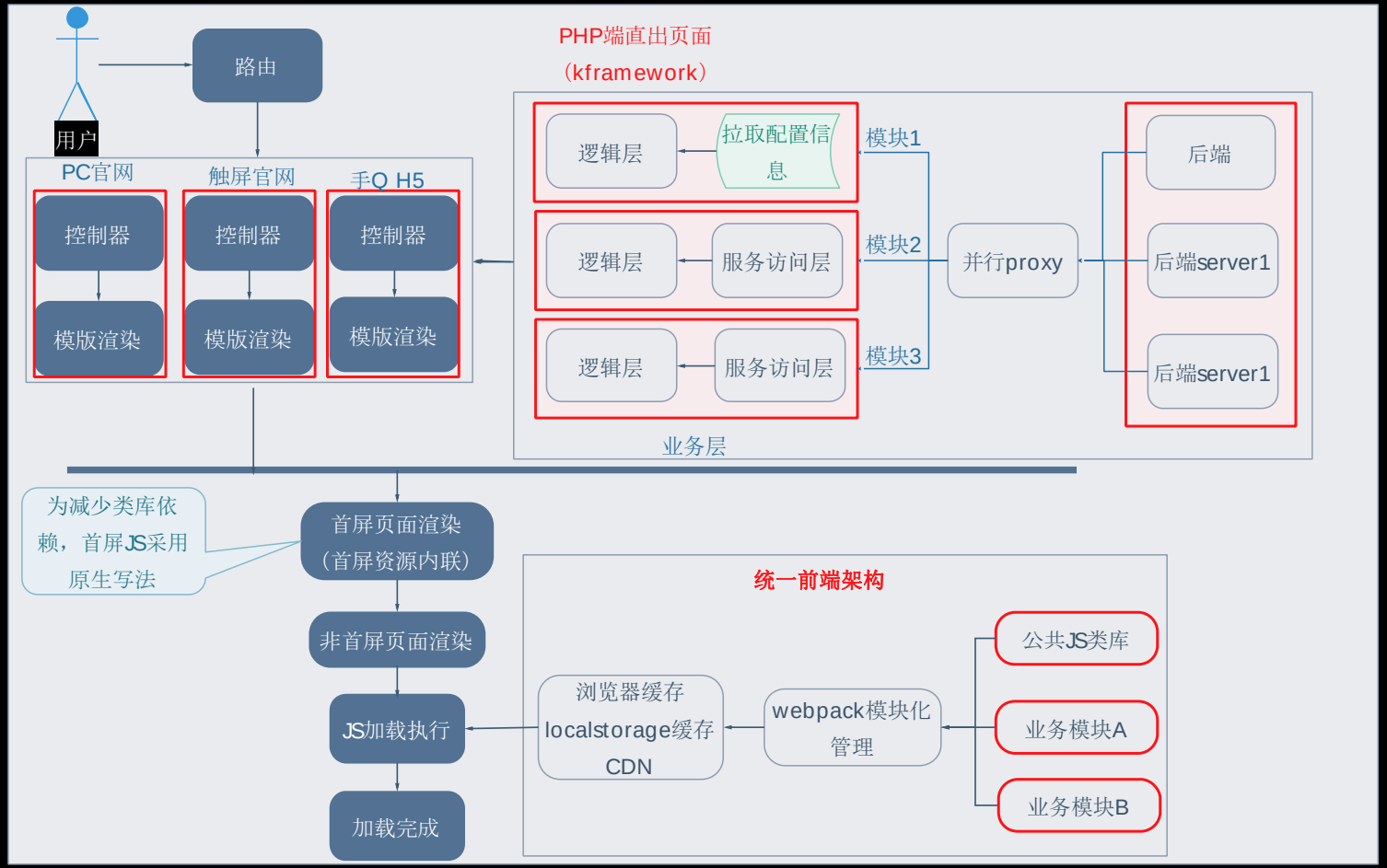
逻辑分层



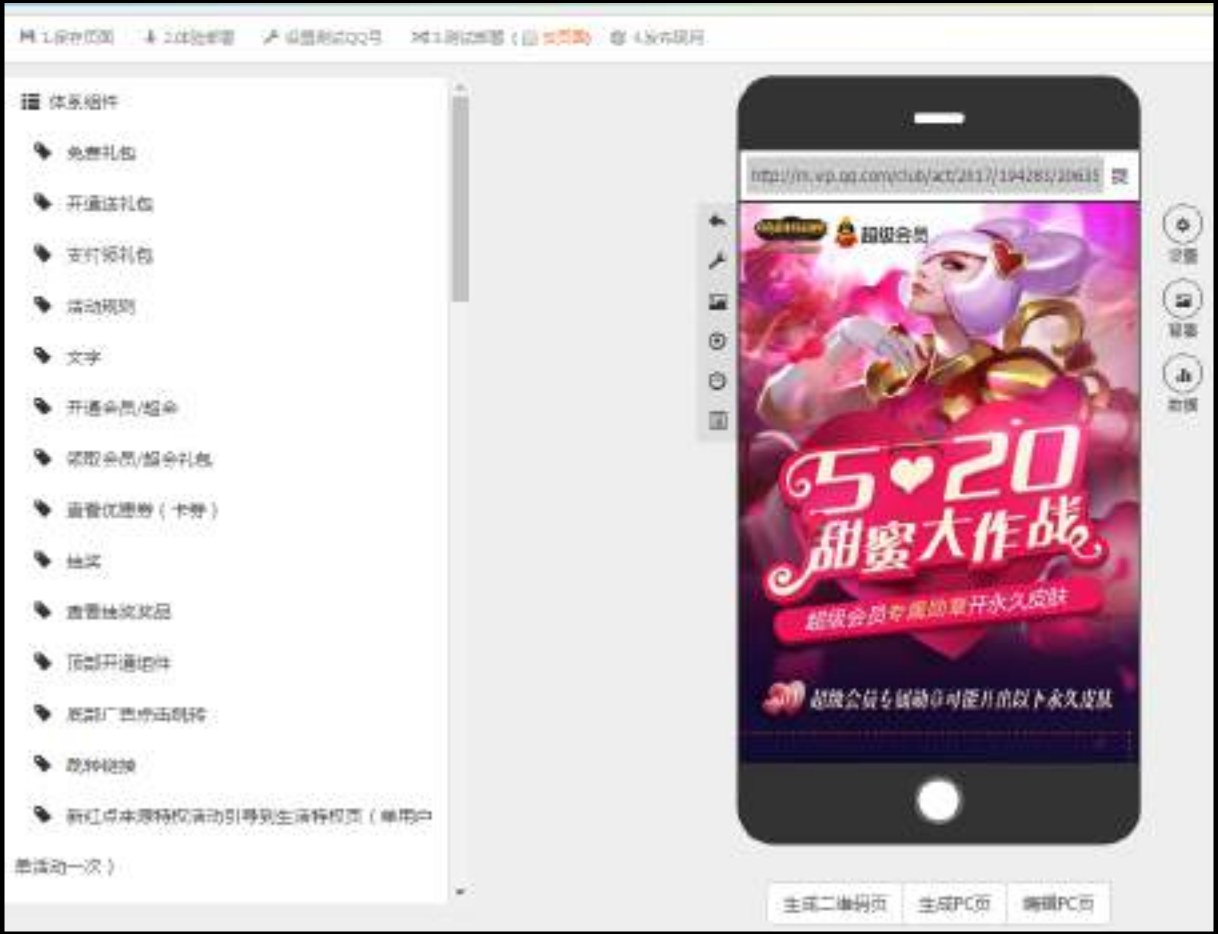
多站点应用

多站点复用

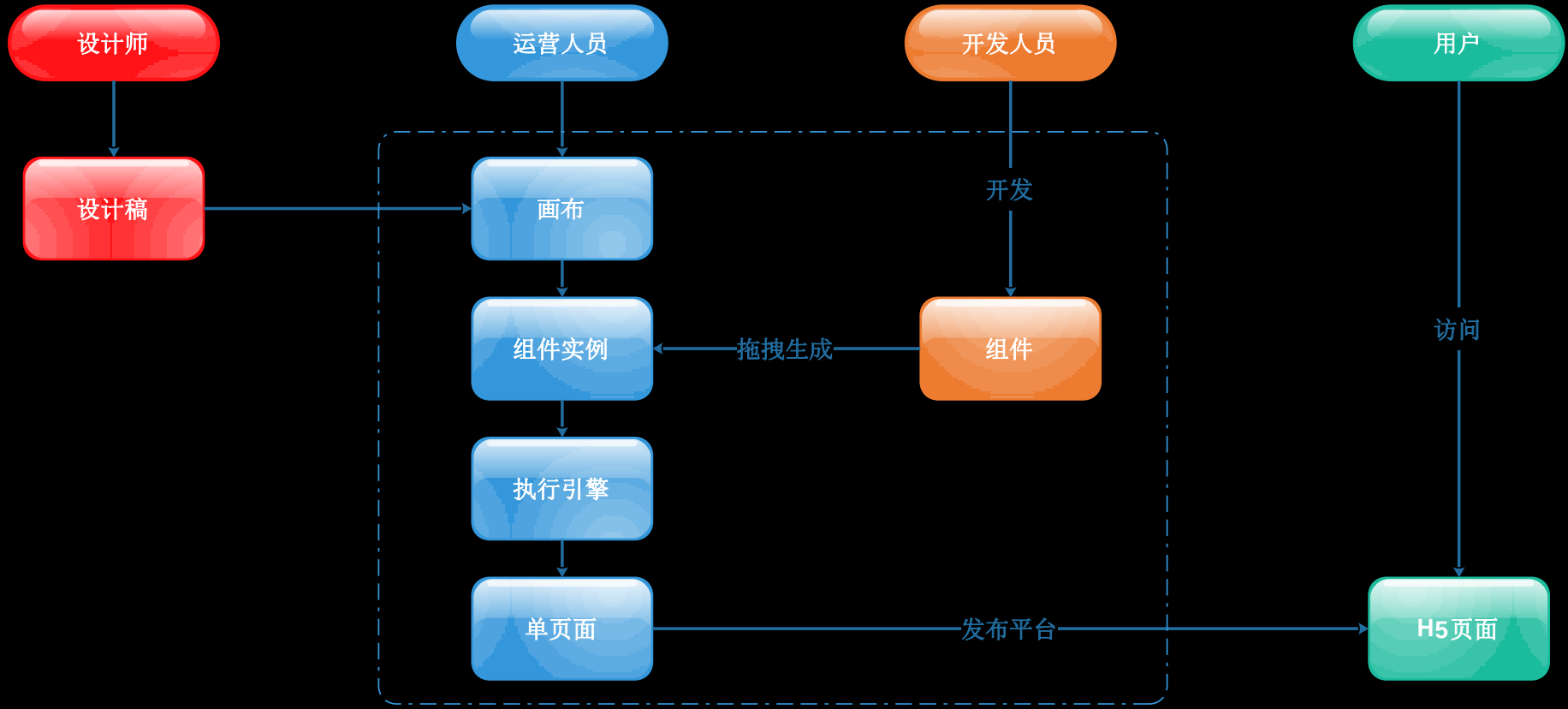
工程化



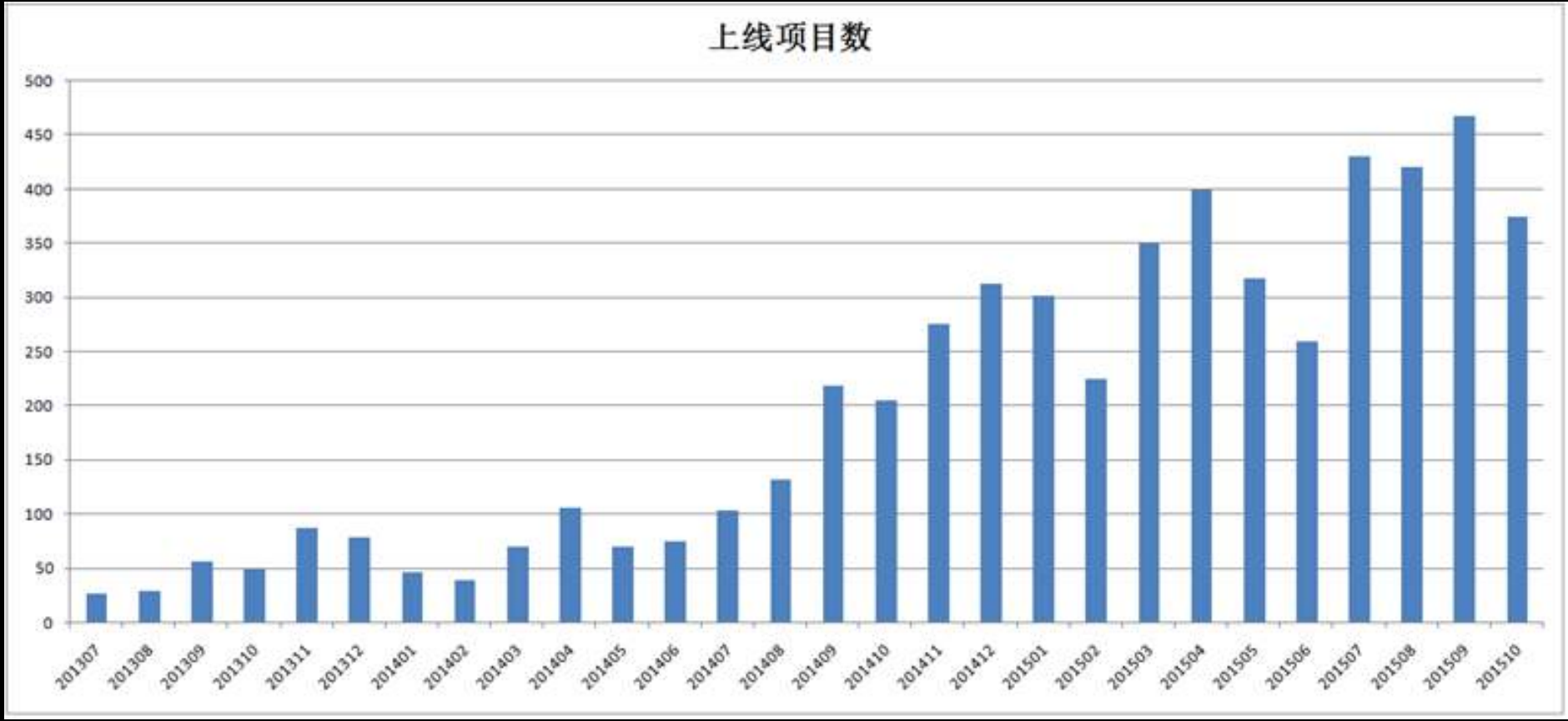
活动开发自动化


体系组件
免费礼包
开通送礼包
支付领礼包
活动规则
文字
开通会员/超会
领取会员/超会礼包
查看优惠券（卡券）
抽奖
查看抽奖奖品
顶部开通组件
底部广告点击跳转
超级会员
5·20
甜蜜大作战
生成二维码页
生成PC页
编辑PC页

活动开发自动化



活动开发自动化效果

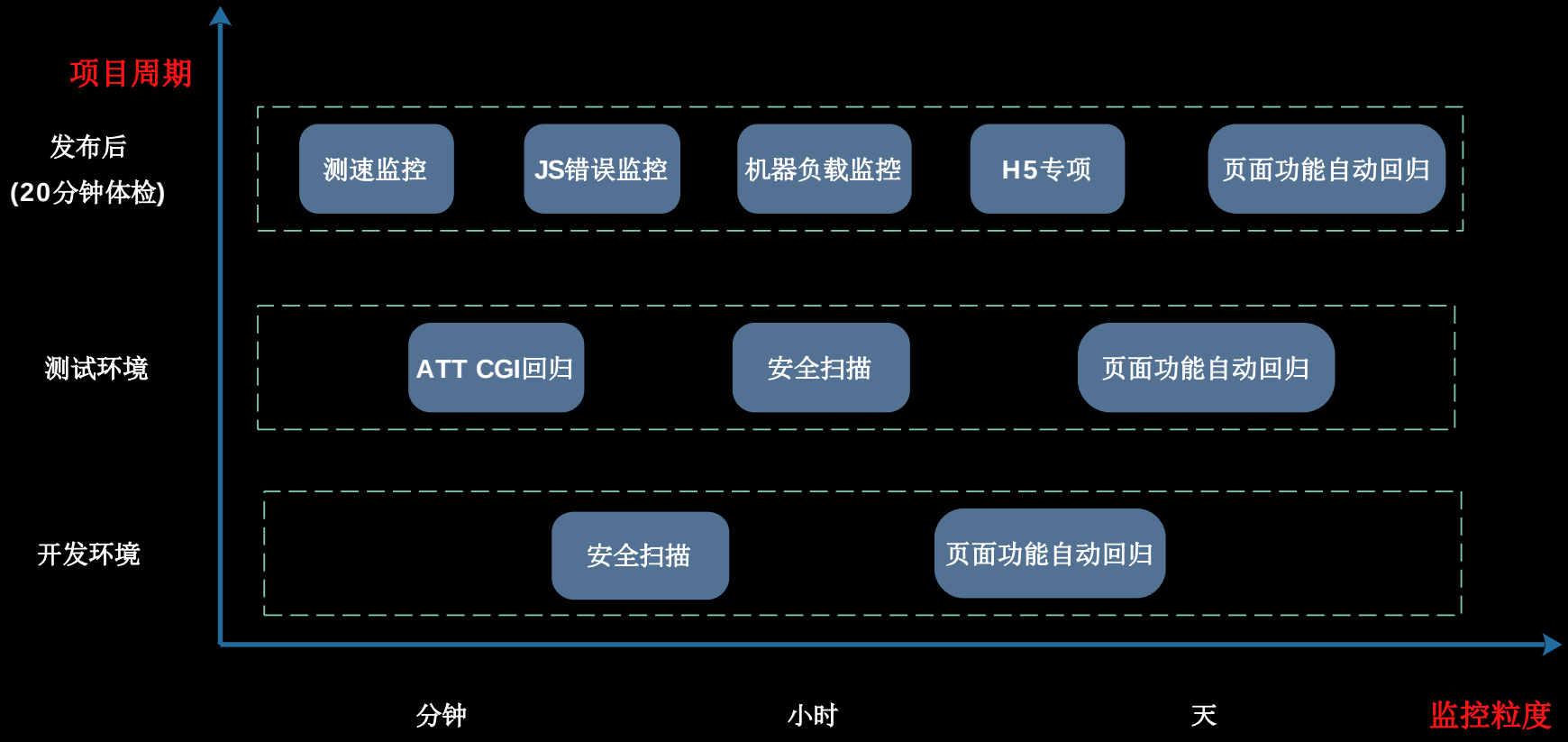

上线项目数
500
450
400
350
300
250
200
150
100
50
0
201307
201308
201309
201310
201311
201312
201401
201402
201403
201404
201405
201406
201407
201408
201409
201410
201411
201412
201501
201502
201503
201504
201505
201506
201507
201508
201509
201510

高可用

高性能 | 高效率 | 高质量

高质量

持续集成

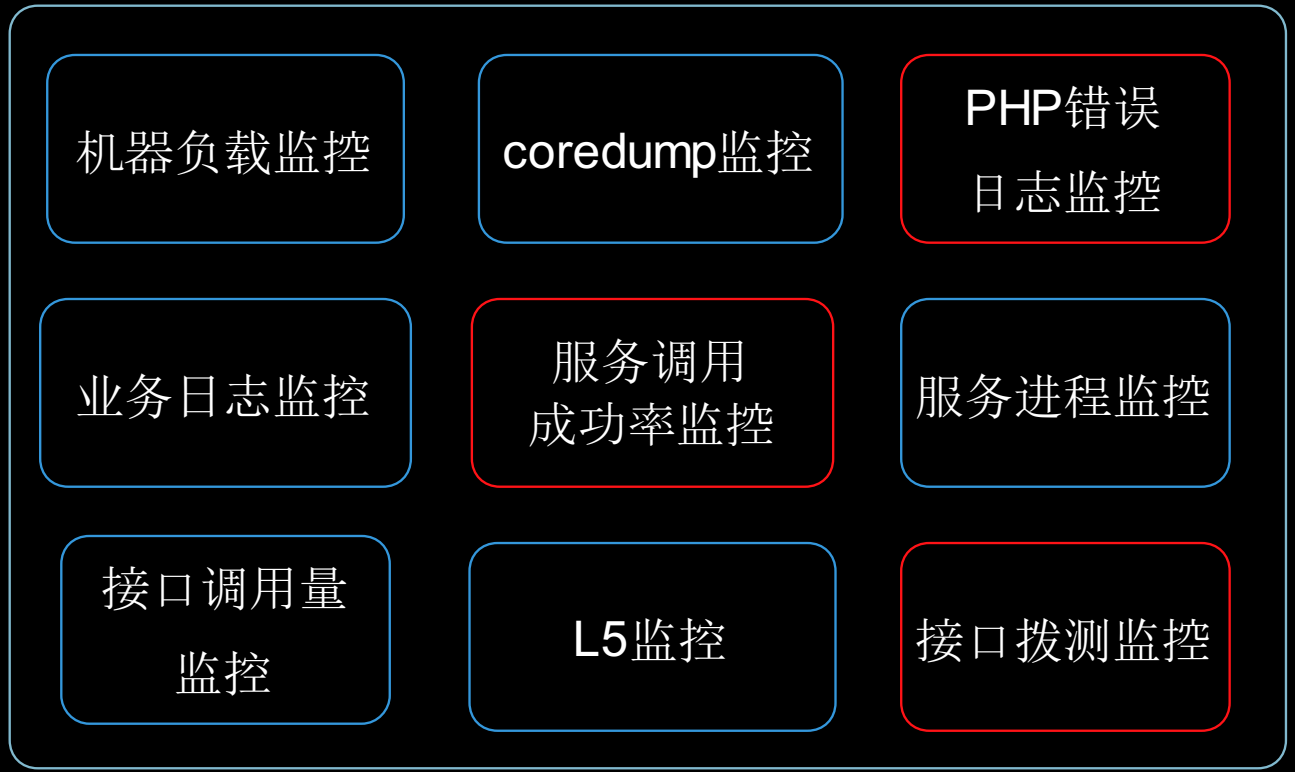

项目周期
发布后
(20分钟体检)
测速监控
JS错误监控
机器负载监控
H5专项
页面功能自动回归
测试环境
ATT CGI回归
安全扫描
页面功能自动回归
开发环境
安全扫描
页面功能自动回归
分钟
小时
天
监控粒度

完善的监控体系

客户端监控

完善的监控体系

服务端监控




QA