

# 构建时预渲染

尚寒阳





尚寒阳  
美团资深研发工程师

多年前端研发经历，负责美团闪付前端系统建设、负责美团支付前  
端基础技术

- 01 渲染方式对比
- 02 预渲染实践
- 03 预渲染效果
- 04 预渲染优化

首帧渲染方式对比

## 前端发展



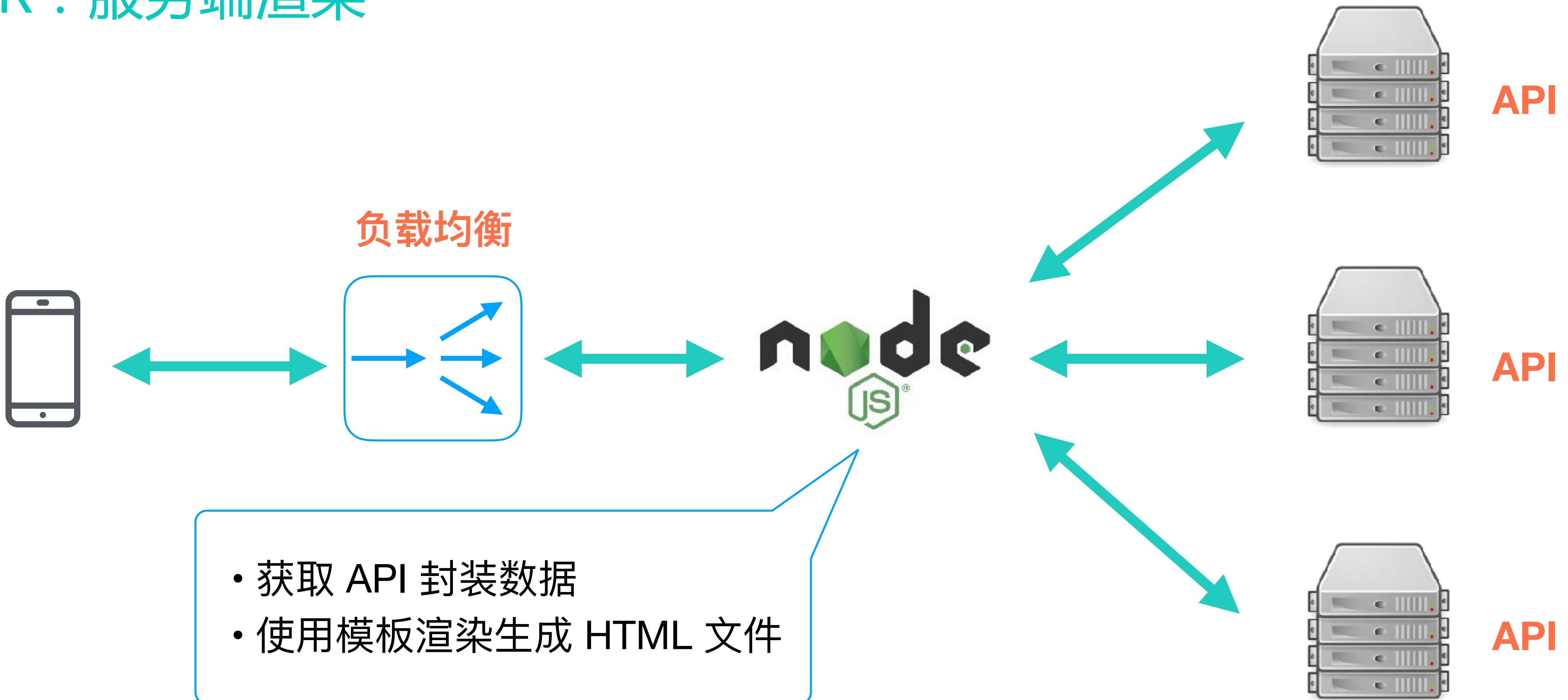
静态 → 动态

后端 → 前端

前端 → 全栈

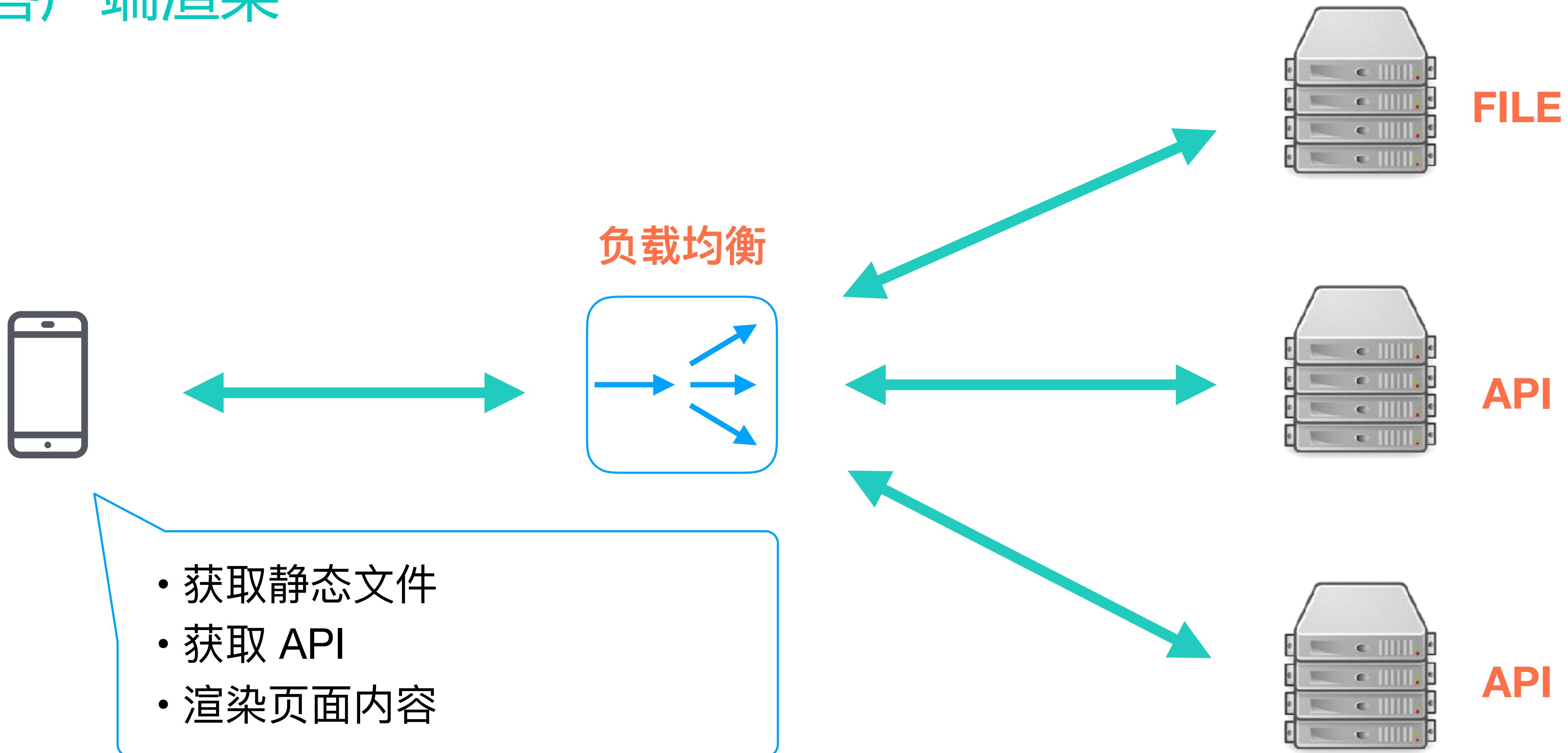
首帧渲染方式对比

## SSR：服务端渲染



首帧渲染方式对比

## CSR : 客户端渲染



## 首帧渲染方式对比

### SSR V.S. CSR



服务端渲染

- 首屏渲染快
- 利于 SEO



客户端渲染

- 内存数据共享
- 网络传输数据少，减少服务器压力
- 可维护性强

- 01 渲染方式对比
- 02 预渲染实践
- 03 预渲染效果
- 04 预渲染优化

预渲染实践

# CSR 首屏白屏问题



用户点击



白屏产生

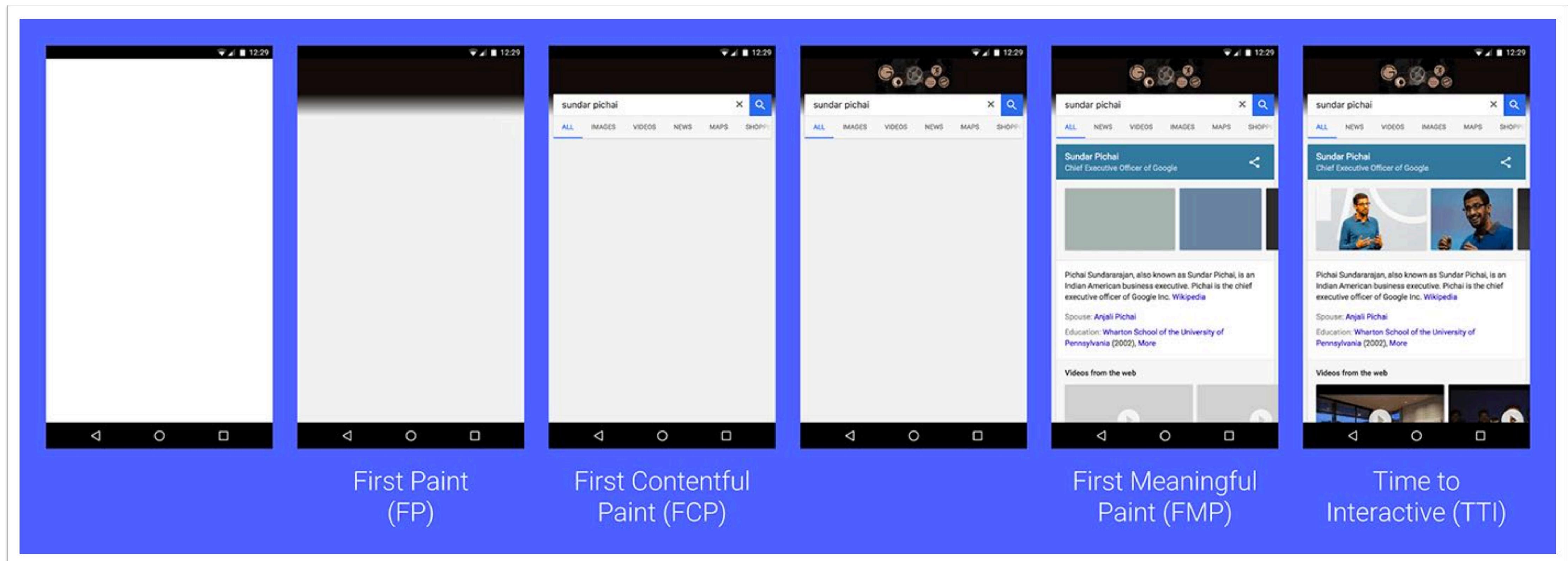


呈现内容

预渲染实践

## 首屏性能指标

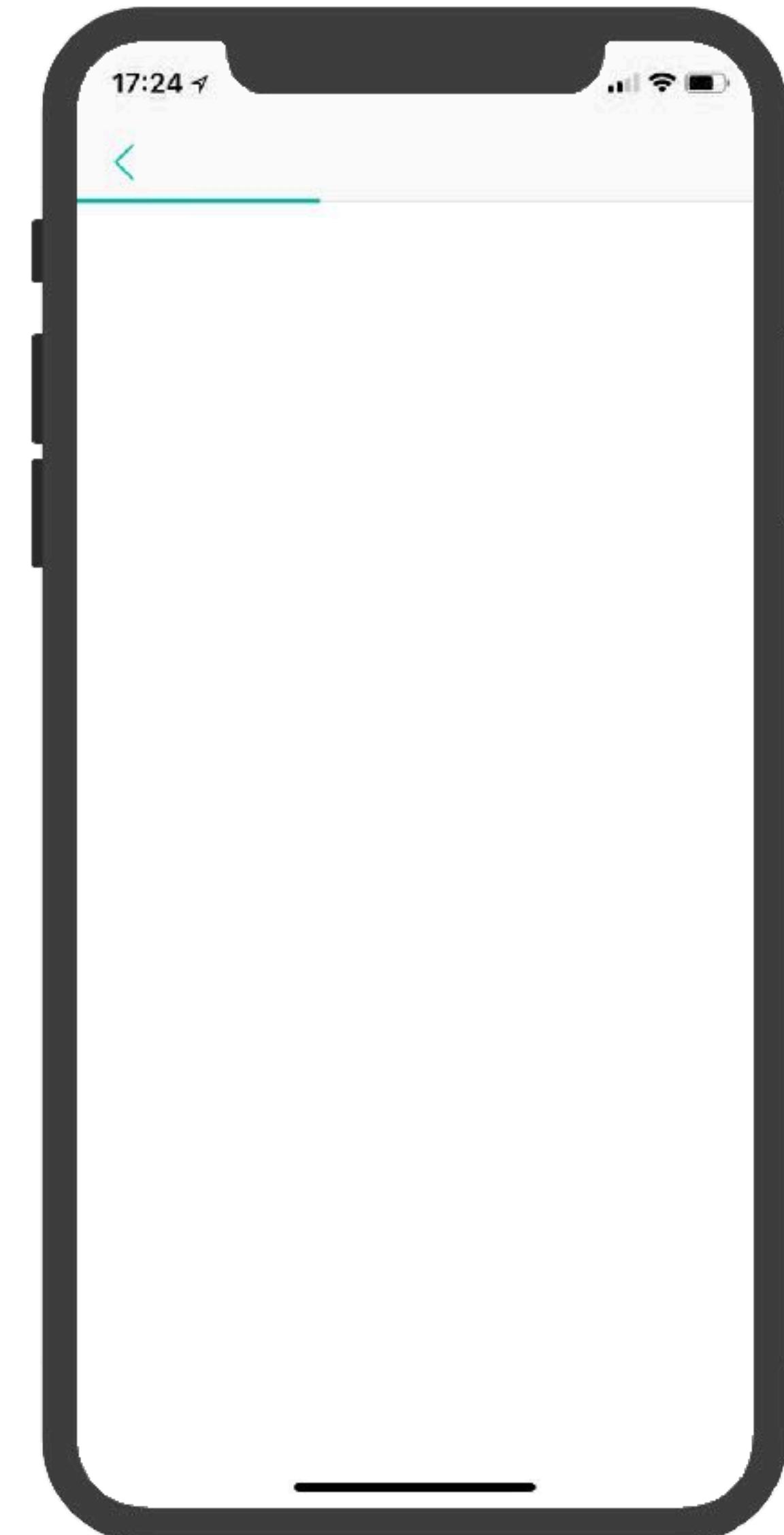
2017-05 Google: User-centric Performance Metrics



预渲染实践

## 首屏性能指标

- TTFB: Time To First Byte, 首字节时间



预渲染实践

## 首屏性能指标

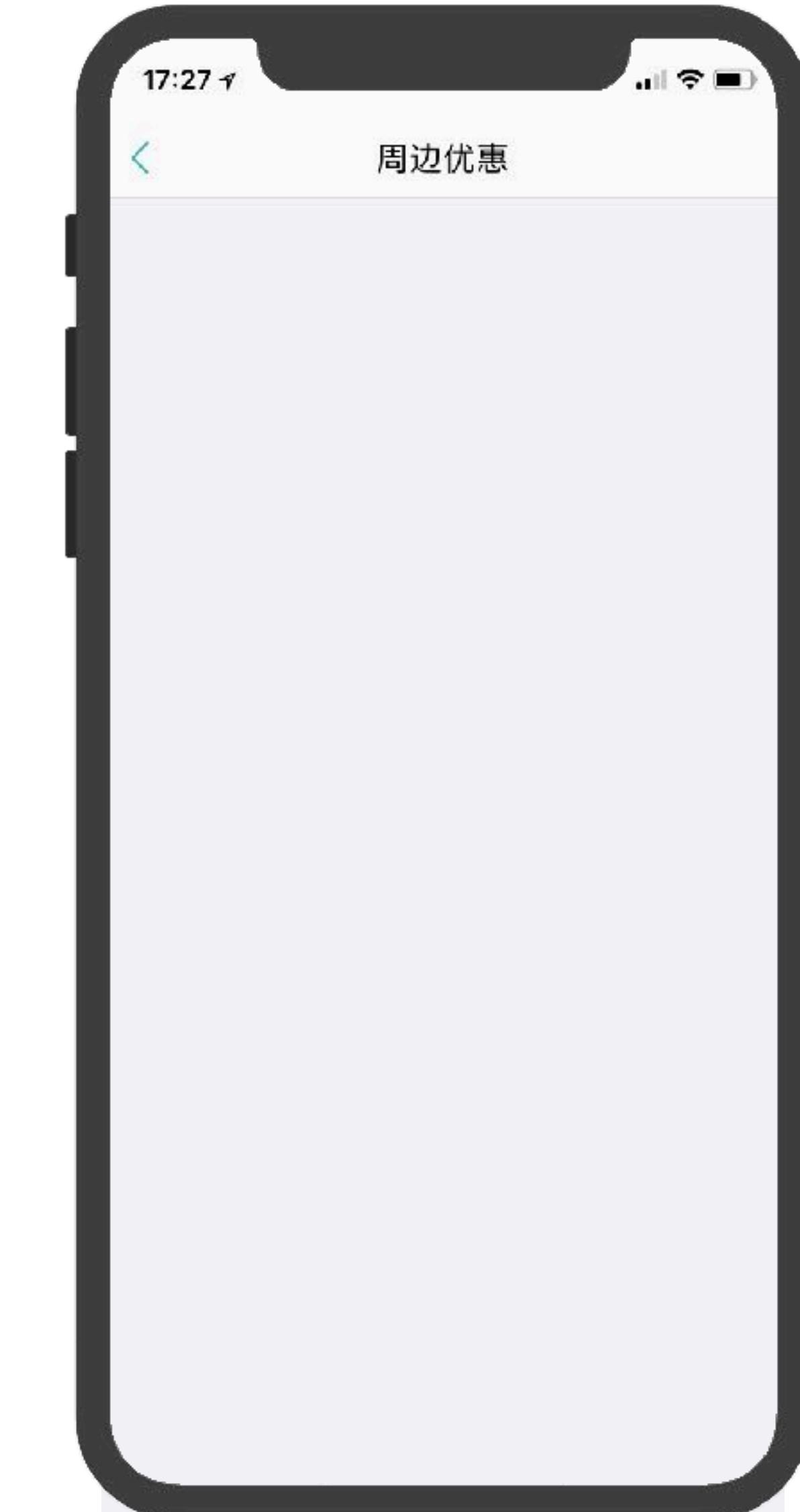
- TTFB: Time To First Byte, 首字节时间
- FP: First Paint, 首次绘制



是不是页面挂了?

等半天就给我看个白板?

....



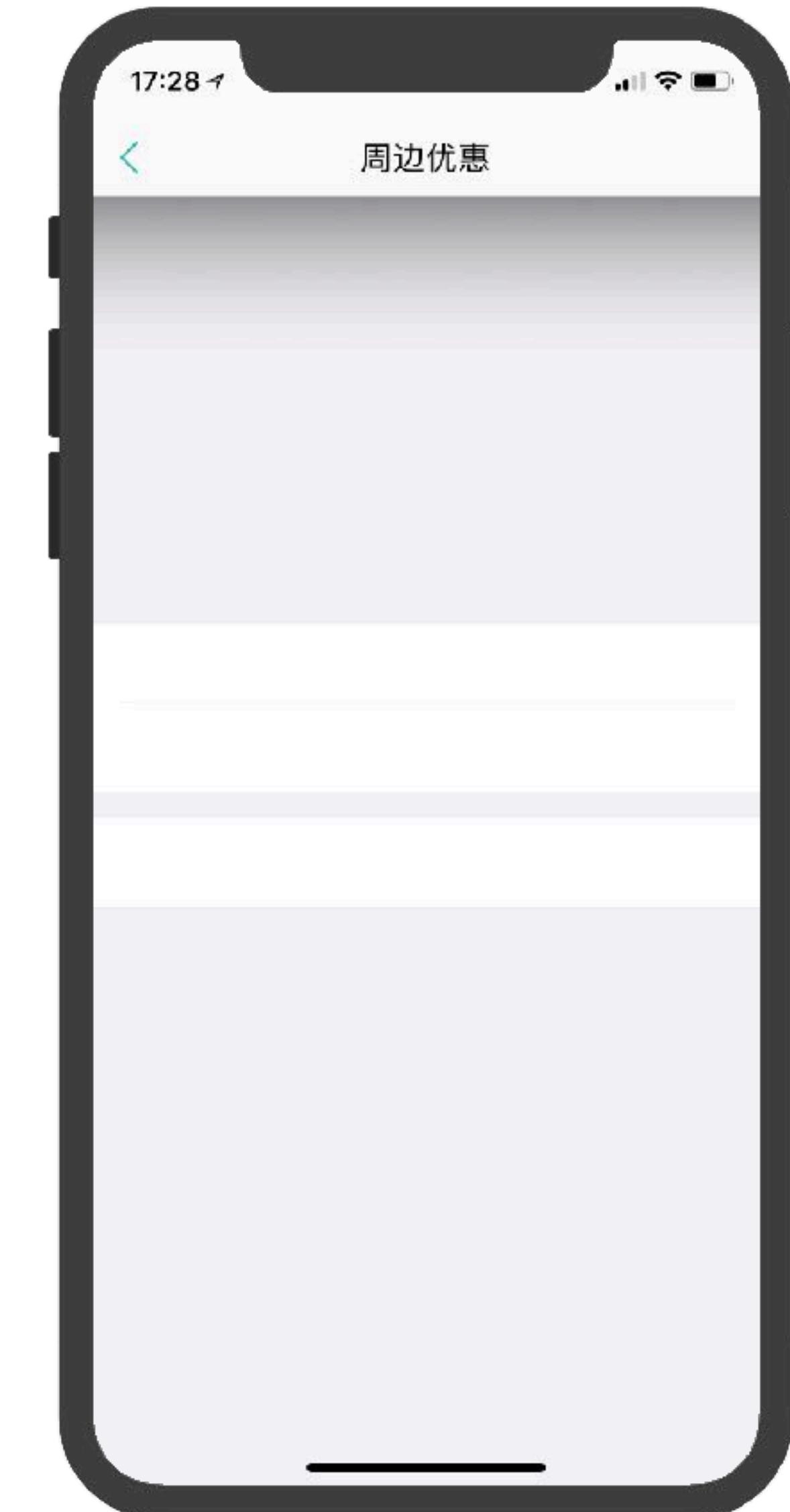
预渲染实践

## 首屏性能指标

- TTFB: Time To First Byte, 首字节时间
- FP: First Paint, 首次绘制
- FCP: First Contentful Paint, 首次有内容的绘制



快出来了，再等等



预渲染实践

## 首屏性能指标

- TTFB: Time To First Byte, 首字节时间
- FP: First Paint, 首次绘制
- FCP: First Contentful Paint, 首次有内容的绘制
- FMP: First Meaningful Paint, 首次有意义的绘制



预渲染实践

## 首屏性能指标

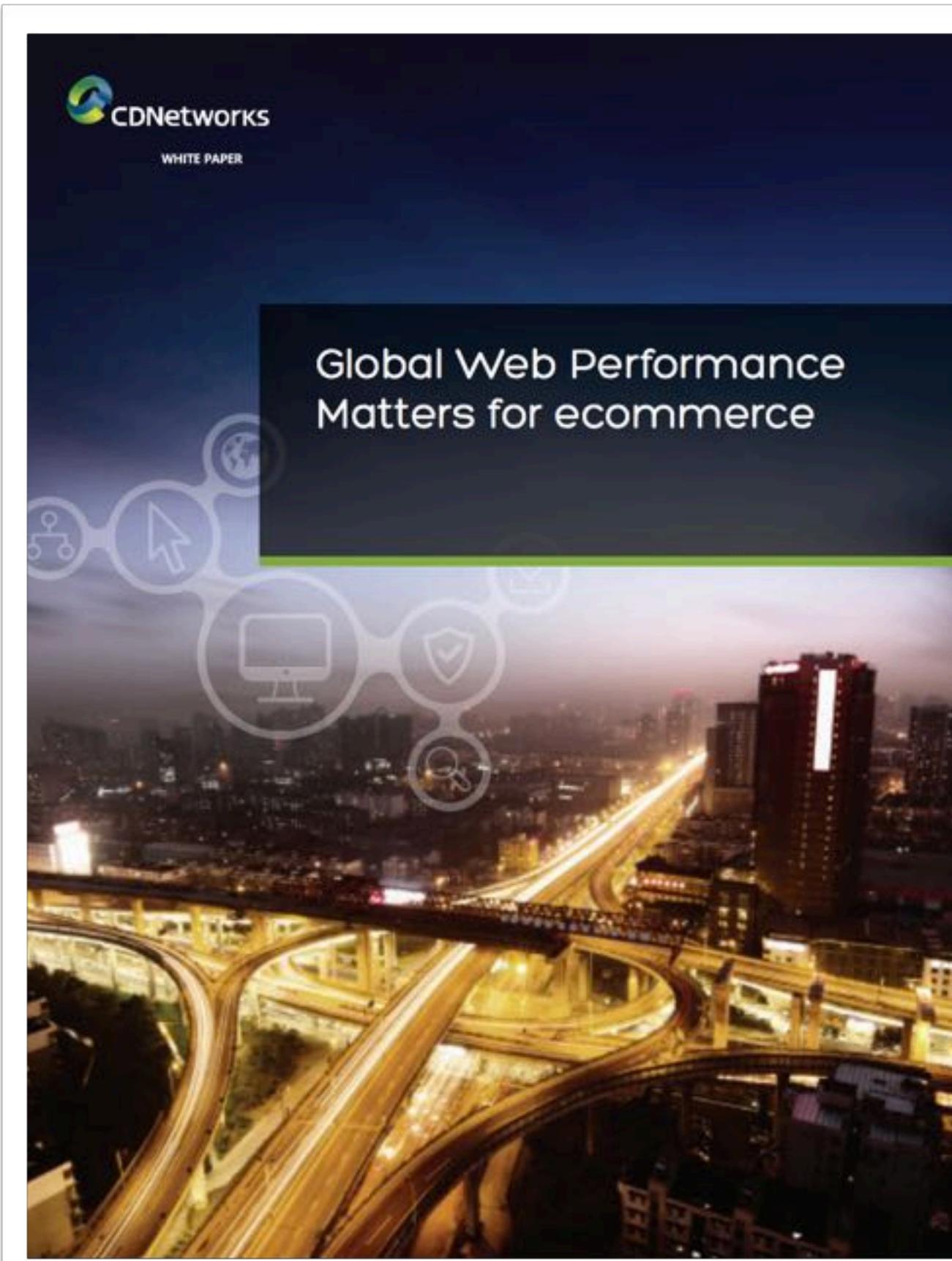
- TTFB: Time To First Byte, 首字节时间
- FP: First Paint, 首次绘制
- FCP: First Contentful Paint, 首次有内容的绘制
- FMP: First Meaningful Paint, 首次有意义的绘制
- TTI: Time To Interactive, 可交互时间

网站太慢会怎样？？



## 预渲染实践

# “慢”的影响



A growing body of evidence demonstrates the importance of attracting and retaining consumers with a high-performing site. Consider some of the most recent published data regarding their threshold for site abandonment:

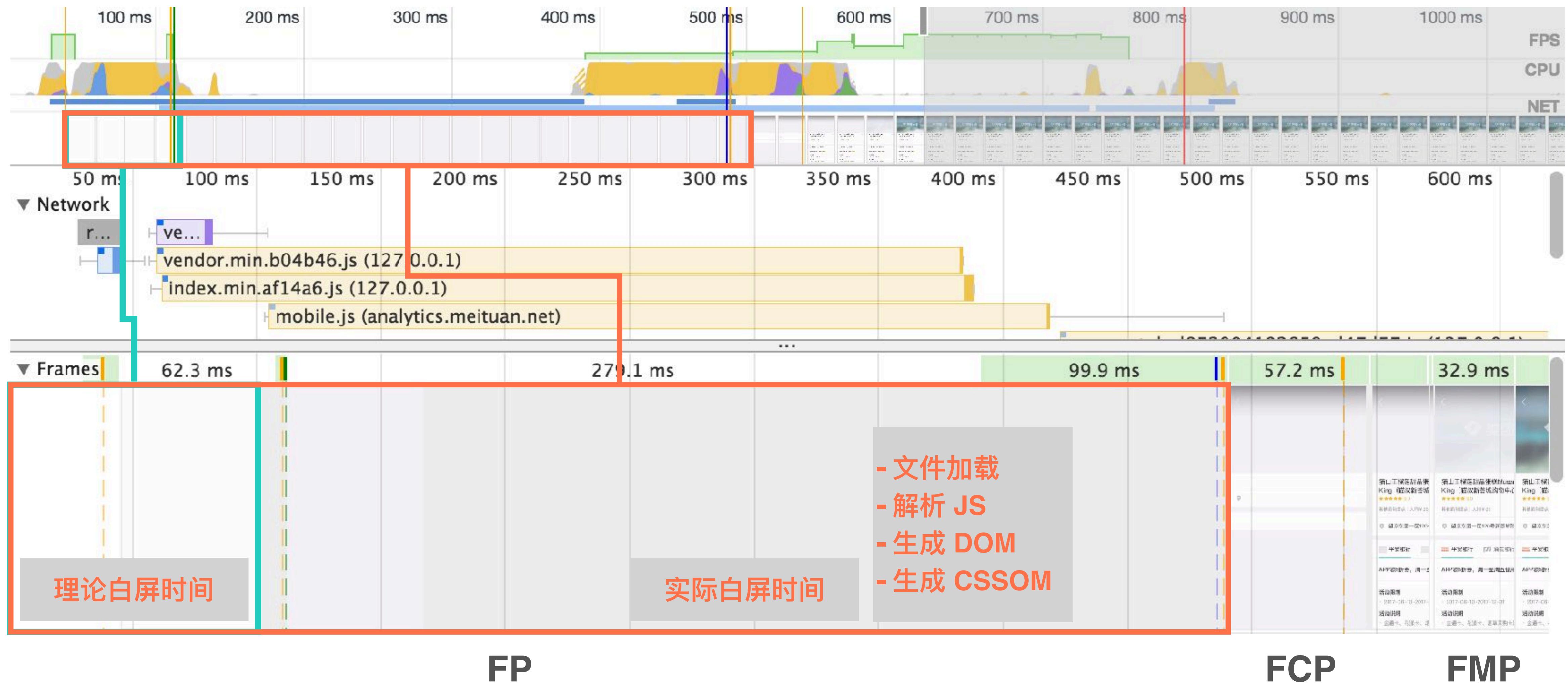
- ▶ 57% of consumers expect a web page to load in 3 seconds or less.
- ▶ 52% of online shoppers state that quick page loading is important to their site loyalty.
- ▶ A 1-second delay decreases page views by 11% and customer satisfaction by about 16%.
- ▶ Nearly half of mobile users abandon a site if it doesn't finish loading within 10 seconds.<sup>2</sup>

*57% of consumers expect a web page to load in 3 seconds or less*

<sup>2</sup> Kissmetrics, How Loading Time Affects Your Bottom Line. <http://blog.kissmetrics.com/loading-time/>

## 预渲染实践

# 为什么有白屏

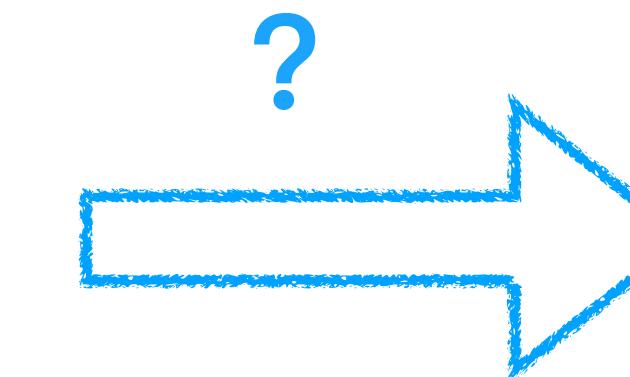


预渲染实践

## 白屏优化思路

页面白屏  
原因

1. CSS & JS 文件获取
2. JS 文件解析
3. DOM 生成
4. CSSOM 生成

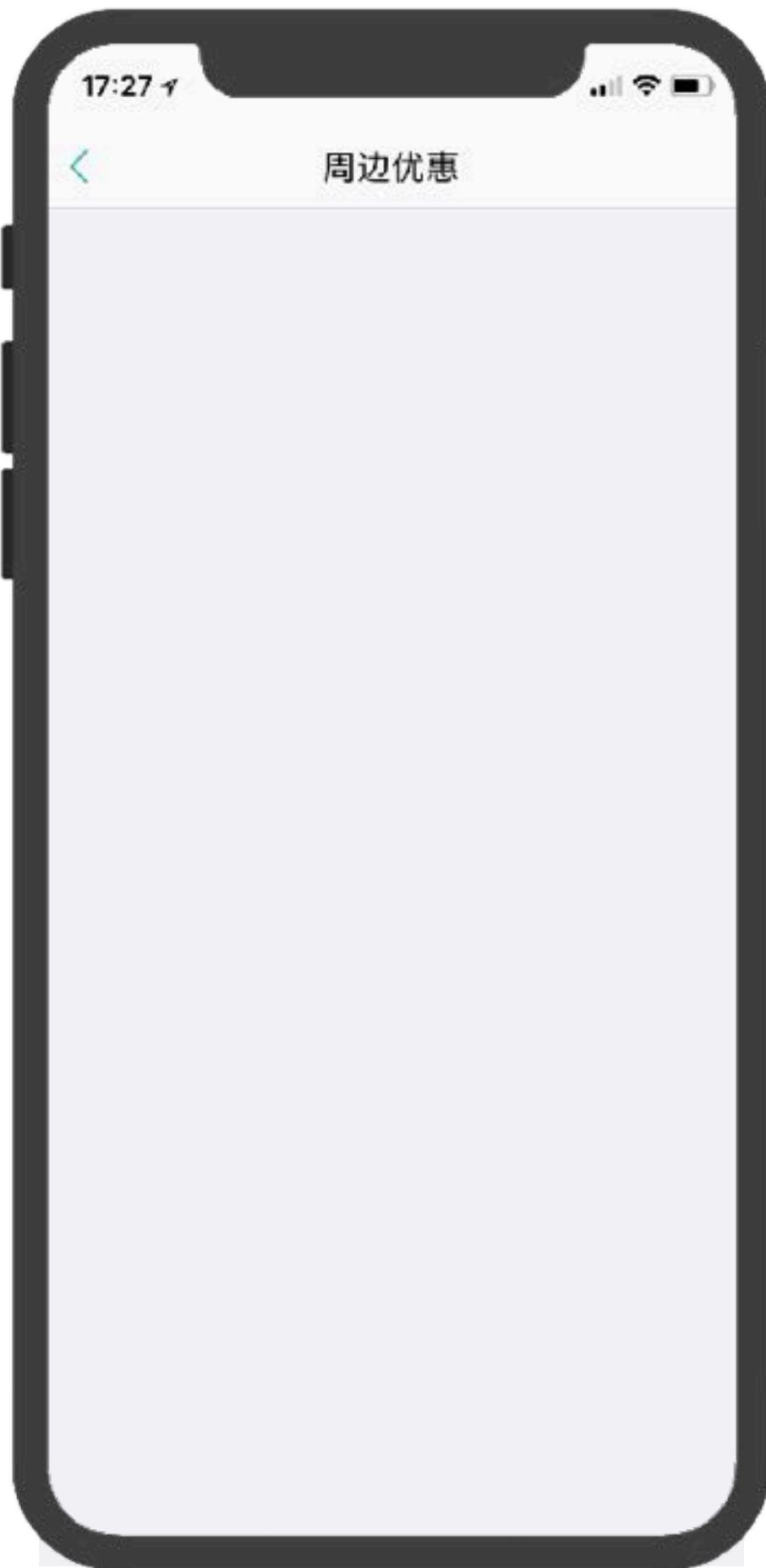


首帧 html  
包含内容

1. 基本的 DOM
2. 基本的 CSS

预渲染实践

## 首帧 html 内容 - FP 时



created

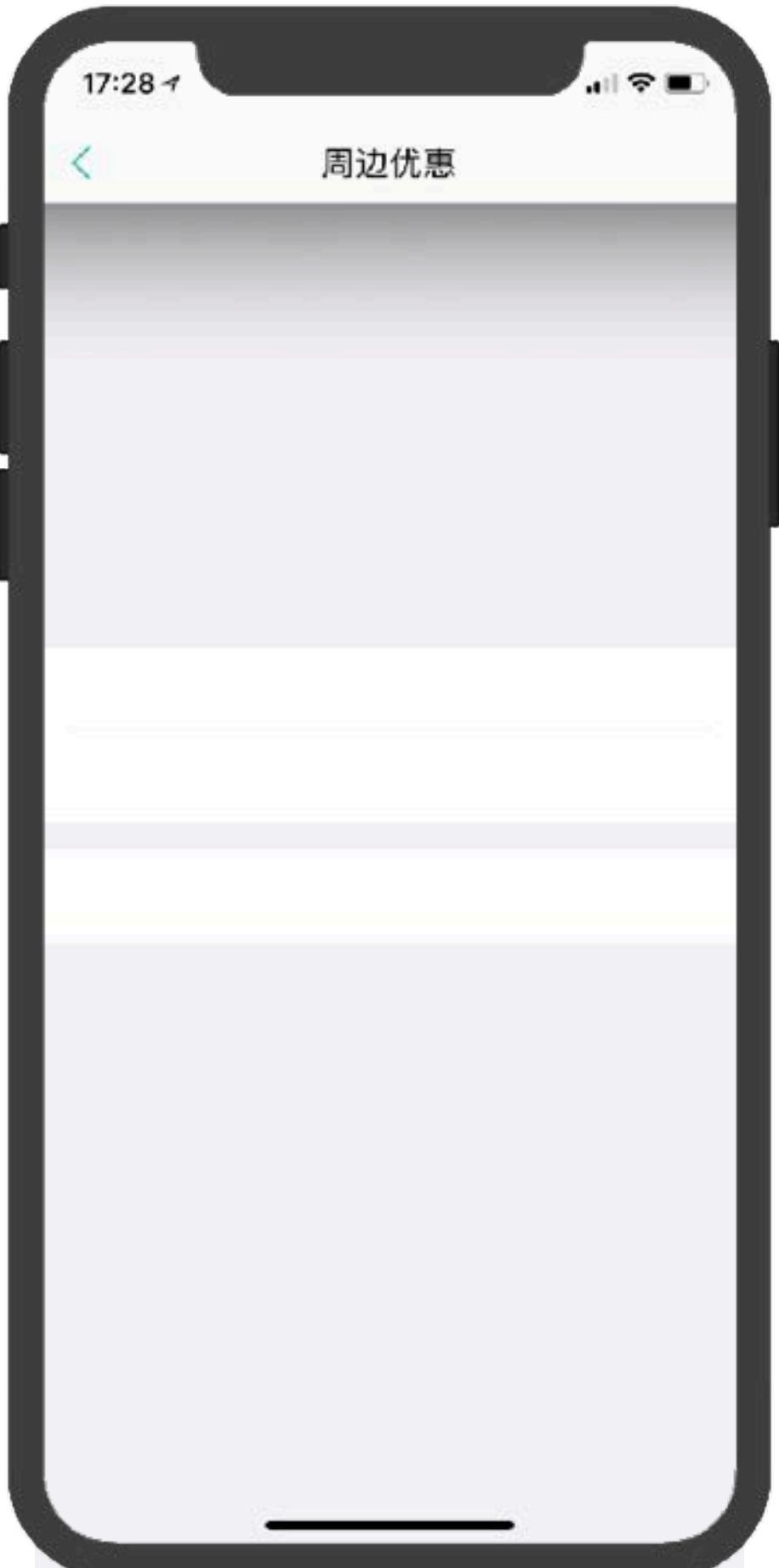


The screenshot shows the browser's developer tools with the DOM tree open. An orange box highlights the opening tag of a `<div id="app">` element. To the right of this highlighted area, the text "空节点" (empty node) is displayed in orange. The DOM tree also includes other elements like `<body ontouchstart="">`, `<script>...`, and `<div class="toast" style="display: none;"><!--><!-->`.

```
<body ontouchstart="">
  <div id="app">
  </div>
  <script crossorigin="anonymous" src="//www.dpfil...>
  <script>...
  </script>
  <script src="https://paystatic.meituan.net/resou...>
  <script src="https://paystatic.meituan.net/resou...>
  <script src="//analytics.meituan.net/analytics.j...>
  <div class="toast" style="display: none;">
    <!-->
    <!-->
  </div>
```

预渲染实践

# 首帧 html 内容 - FCP 时



mounted



页面  
基本框架

```
<body ontouchstart="">
  <div id="app">
    <div class="base-bg bg-gary has-share-card" style="overflow: auto; background-color: #f0f0f0; position: relative; height: 100%; width: 100%;">
      <!--头部-->
      <div class="head-mask"></div>
      <div class="banner">...</div>
      <div class="info-box box">
        <div class="inner">
          <div class="title">...</div>
          <!--中间内容-->
          <div class="info">...</div>
          <div class="contact">...</div>
        </div>
      </div>
      <div class="tab-box box">...</div>
    </div>
    <div>...</div>
  </div>
<script crossorigin="anonymous" src="//www.dpfile.com/app/owl/static/owl.js">...</script>
<script>...</script>
<script src="https://paystatic.meituan.net/resource/wa/poi/page/vendor.min.js">...</script>
<script src="https://paystatic.meituan.net/resource/wa/poi/page/index.min.js">...</script>
<script src="//analytics.meituan.net/analytics.js" type="text/javascript">...</script>
<div class="toast" style="display: none;">
  <!--尾部-->
</div>
```

# 预渲染实践

# 首帧 html 内容 - FMP 时



# updated

# 完整 页面内容

```
<body ontouchstart="" class="hairline" style="">
<div id="app">
  <div class="base-bg bg-gary nav-ios has-share-card" style="overflow
    <!-->
    <!-->
    <div class="head-mask"></div>
    <div class="banner" style="transform-origin: center top 0px;">
      <ul class="banner-track">
        <li class="banner-item loaded" style="background-image: url(&
      </ul>
    <!!-->
  </div>
  <div class="info-box box">
    <div class="inner">
      <div class="title">
        <h3>星巴克 (望京东路店) </h3>
        <div class="distance">13907.3km</div>
      </div>
      <div class="rate">...
      </div>
      <div class="info">...
      </div>
      <div class="contact">...
      </div>
    </div>
  </div>
  <div class="tab-box box">
    <div class="tab">
      <ul class="tab-menu">
        <li class="active">
          中国银行
            
          
  <title>支持商家</title>
  <meta name="lx:category" content="fd">
  <meta name="lx:appnm" content="Meituan UnionPay QRcode">
  <link rel="dns-prefetch" href="//analytics.meituan.net" />
  <link rel="dns-prefetch" href="//portal-portm.meituan.com" />
  <meta name="wap-font-scale" content="no">
  <meta name="apple-touch-fullscreen" content="yes">
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="format-detection" content="telephone=no,email=no">
  <meta name="viewport" content="width=device-width,initial-scale=1,>
<script>
  window.Promise || document.write(
    '<script src="//awp-assets.meituan.net/hfe/hfe-assets/polyfill/2.js">
</script> ...
<script>...
</script>
</head>

<body>
  <div id="app">
    ...
</div>
<script src="page/vendor.min.36fbbe.js"></script>
<script src="page/index.min.9adcaf.js"></script>
<script src="//analytics.meituan.net/analytics.js" type="text/javascript"></script>
</body>
</html>
```

```
.cell-arrow::after {
  right: 0;
  top: 50%;
  content: '';
  width: 8px;
  height: 8px;
  position: absolute;
  -webkit-transform: rotate(45deg);
  transform: rotate(45deg);
  border: 1px solid #C6C6C7;
  border-width: 1px 1px 0 0;
  margin-top: -4px;
}
```

```
<div id="app">
  <div class="index-home-category"></div>
  <div class="sticky-tab">
    <div class="sticky-tab-menu">...
    </div>
    <div class="fix-padding" style="display: none;"></div>
    <div class="sticky-tab-wrap" style="display: none;">...
    </div>
    <div class="sticky-tab-content">...
    </div>
  </div>
</div>
```

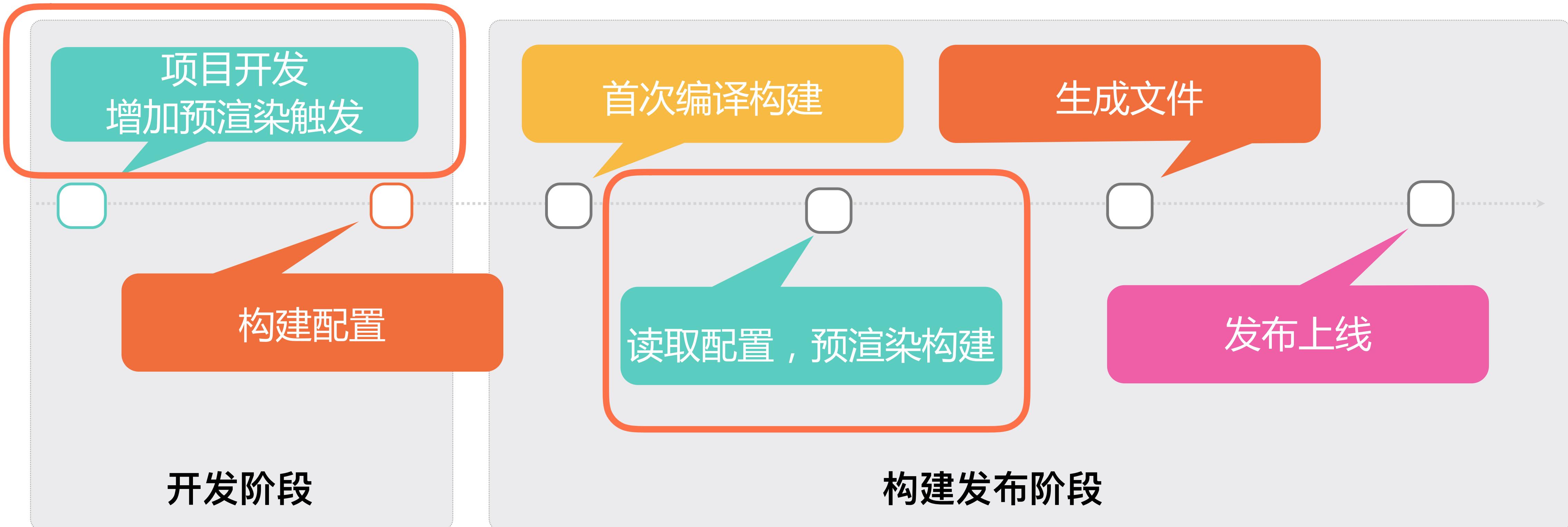
预渲染实践

## 常见的项目开发



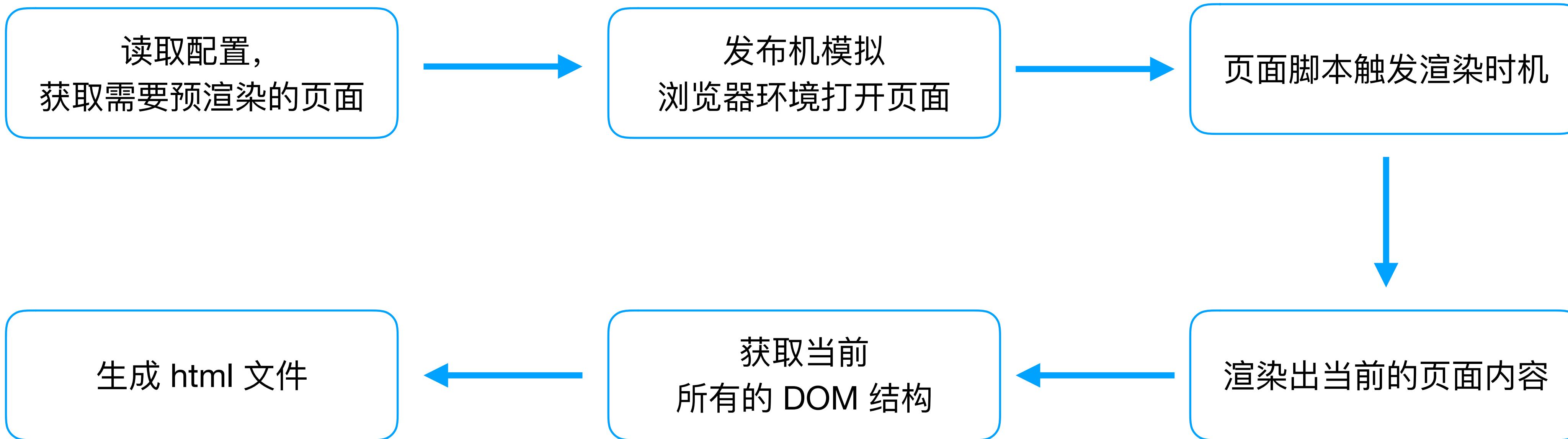
预渲染实践

## 构建时预渲染方案



预渲染实践

## 如何获取 mounted 时机的 DOM



预渲染实践

## 预渲染难点 - 静态资源可访问

构建前

/page/index.js

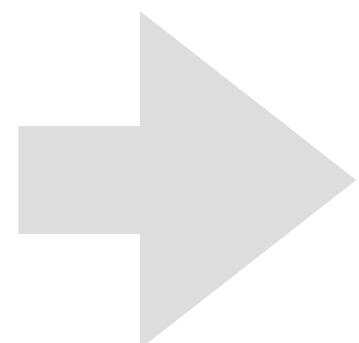
构建后

https://paystatic.meituan.net/page/index.js

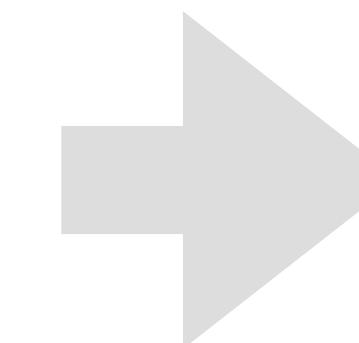
ERROR 404!



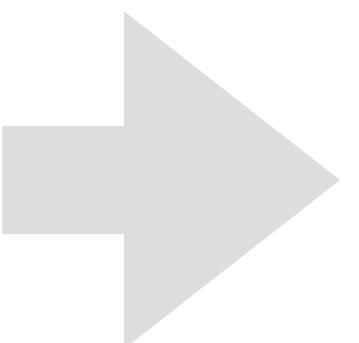
拦截模拟环境  
资源请求



判断发出请求  
URI



替换 CDN 域  
请求到预渲染  
Web 服务域



发出资源请求  
进行

预渲染实践

## 预渲染实践 - 开发阶段



触发构建时机

JS 脚本在合适的时机  
触发预渲染构建事件

配置需要预渲染的页面

让发布机明确需要编译的  
目标页面或路由

预渲染实践

## 预渲染构建阶段

### 预渲染构建 问题

#### 1. 如何模拟浏览器环境渲染

需要在发布机上进行页面渲染



**使用 PhantomJS**

#### 2. 如何提升编译效率

多页面串行效率低



**多子进程并行编译**

#### 3. 如何拦截请求

解决静态文件可访问



**node server 进行请求拦截**

## 预渲染实践

# PhantomJS 简介

## PhantomJS - Scriptable Headless Browser

**Important:** PhantomJS development is suspended until further notice ([more details](#)).

PhantomJS is a headless web browser scriptable with JavaScript. It runs on Windows, macOS, Linux, and FreeBSD.

Using QtWebKit as the back-end, it offers fast and native support for various web standards: DOM handling, CSS selector, JSON, Canvas, and SVG.

### Page automation

Access webpages and extract information using the standard DOM API, or with usual libraries like jQuery.

### Screen capture

Programmatically capture web contents, including SVG and Canvas. Create web site screenshots with thumbnail preview.

### Headless website testing

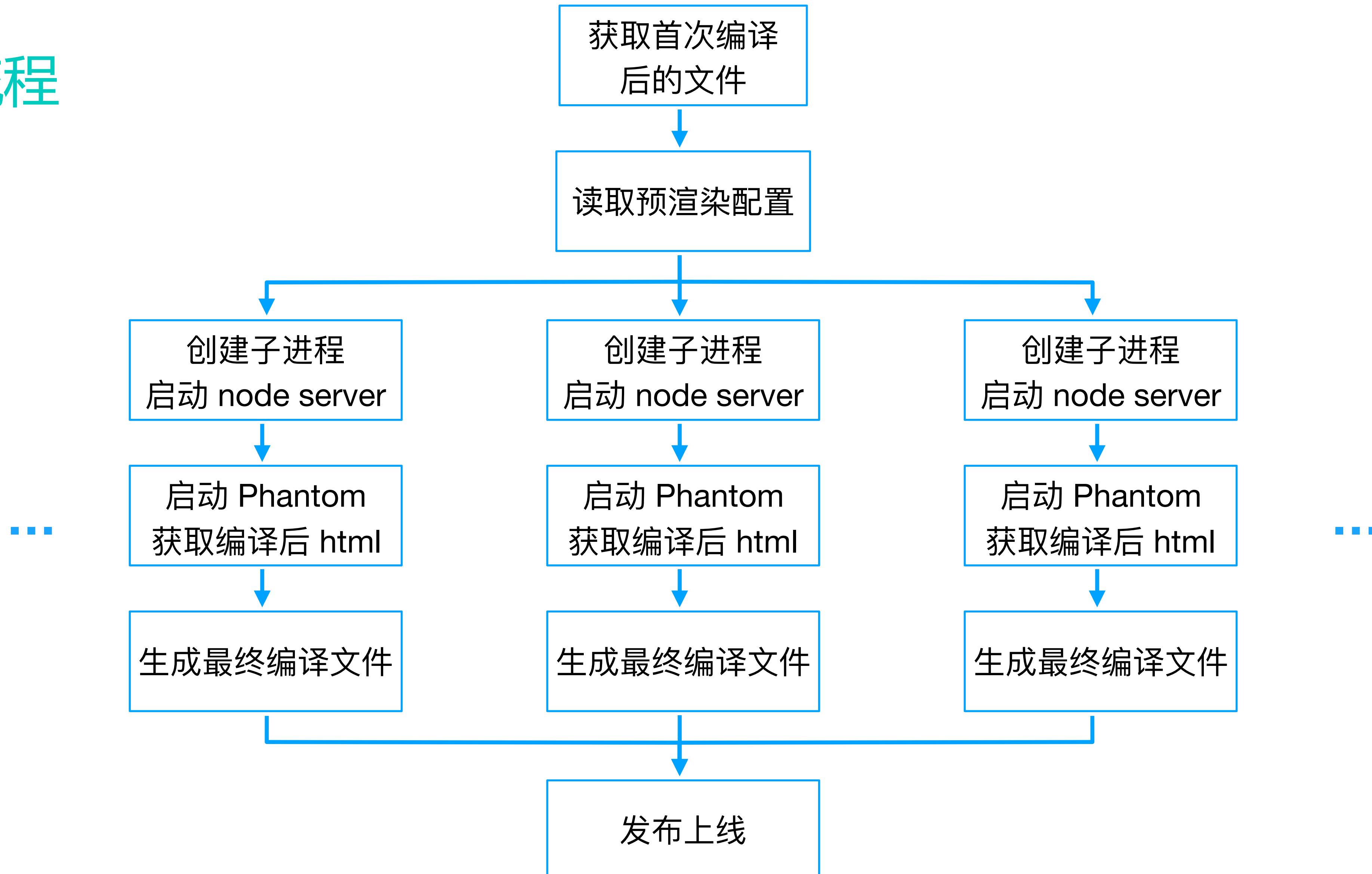
Run functional tests with frameworks such as Jasmine, QUnit, Mocha, WebDriver, etc.

### Network monitoring

Monitor page loading and export as standard HAR files. Automate performance analysis using YSlow and Jenkins.

## 预渲染实践

# 编译流程



- 01 渲染方式对比
- 02 预渲染实践
- 03 预渲染效果
- 04 预渲染优化

预渲染效果

## 体验效果对比

CSR

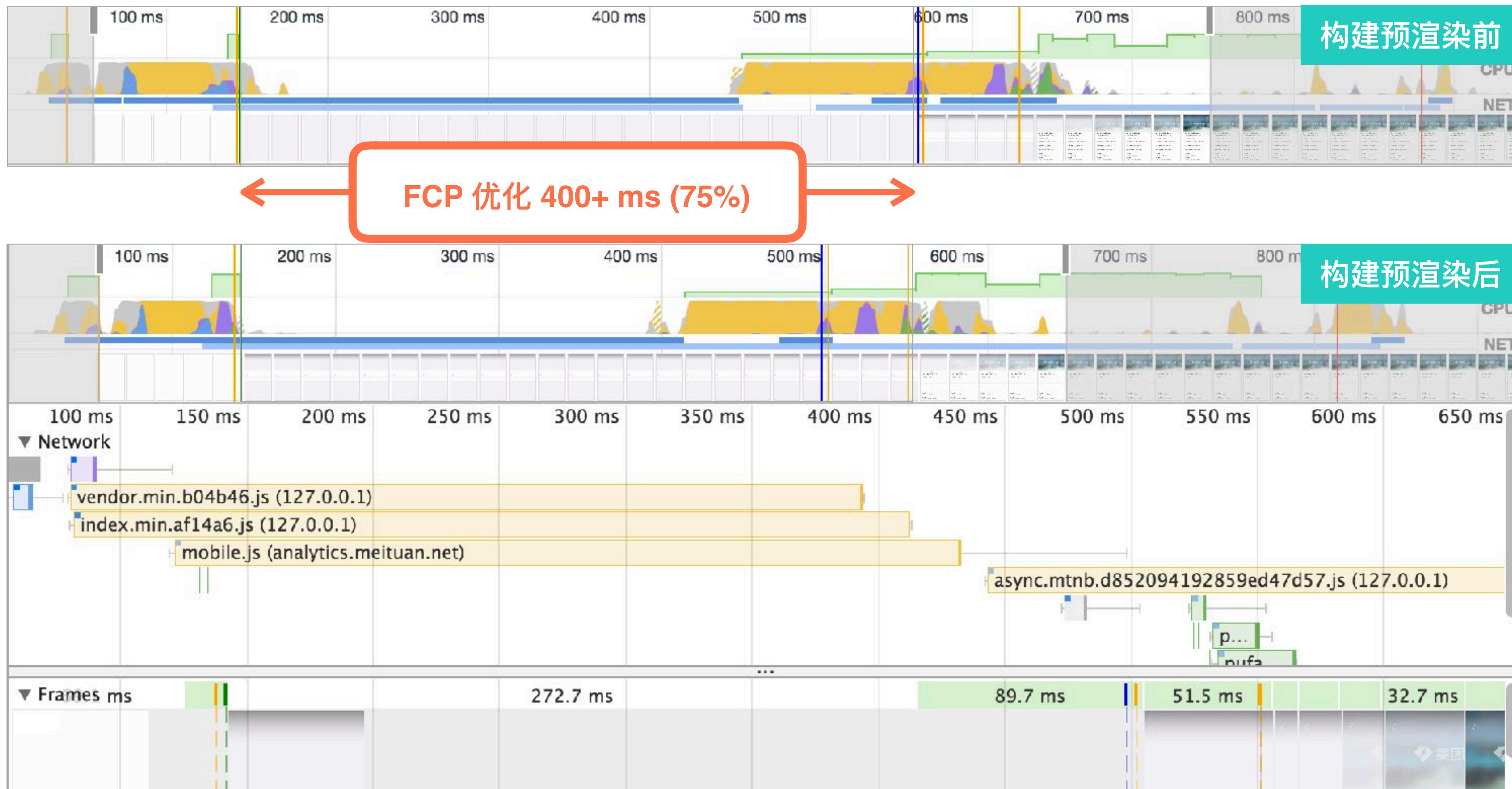


Pre-render



预渲染效果

## 数据指标对比



预渲染效果

接入简单

一行代码完成接入

```
1 import 'style/detail.less';
2 import { env, request, bridge } from 'common';
3 import { lx, LXConfig } from 'common/lx';
4 import { Vue, Component, Watch, Prerender } from 'vue-class';
5 import render from './index.html';
6 import RedPacket from 'common/red-packet';
7 import BottomButton from 'common/bottom-button';
8 import { callUpMeituan } from '@mx/util/call-up-meituan';
9
10 @Prerender
11 <component>
12   render,
13   components: { RedPacket, BottomButton }
14 </component>
15
16 export default class extends Vue {
17   created() {
18     bridge.configEH({
19       hideNavigationBar: true
20     });
21     bridge.ehShow();
22     this.loadData(this.jumpToMeituan);
23   }
24 }
```

- 
- 
- 01 渲染方式对比
  - 02 预渲染实践
  - 03 预渲染效果
  - 04 预渲染优化

# 预渲染优化

A

## 工具优化

使用 Headless Chrome &  
puppeteer 替代  
phantomjs-prerender

B

## 编译优化

自定义构建时机；  
MutationObserver

C

## 使用插件

使用或借鉴 prerender-spa-  
plugin 进行编译

# Q&A

招聘前端专家、资深前端工程师  
简历投放至：[chenyulin02@meituan.com](mailto:chenyulin02@meituan.com)



# CODE A BETTER LIFE

一 行 代 码      亿 万 生 活



更多技术干货  
欢迎关注“美团技术团队”

THANK YOU !

