

在 PWA 中使用 App Shell 模型提升性能和用户感知体验

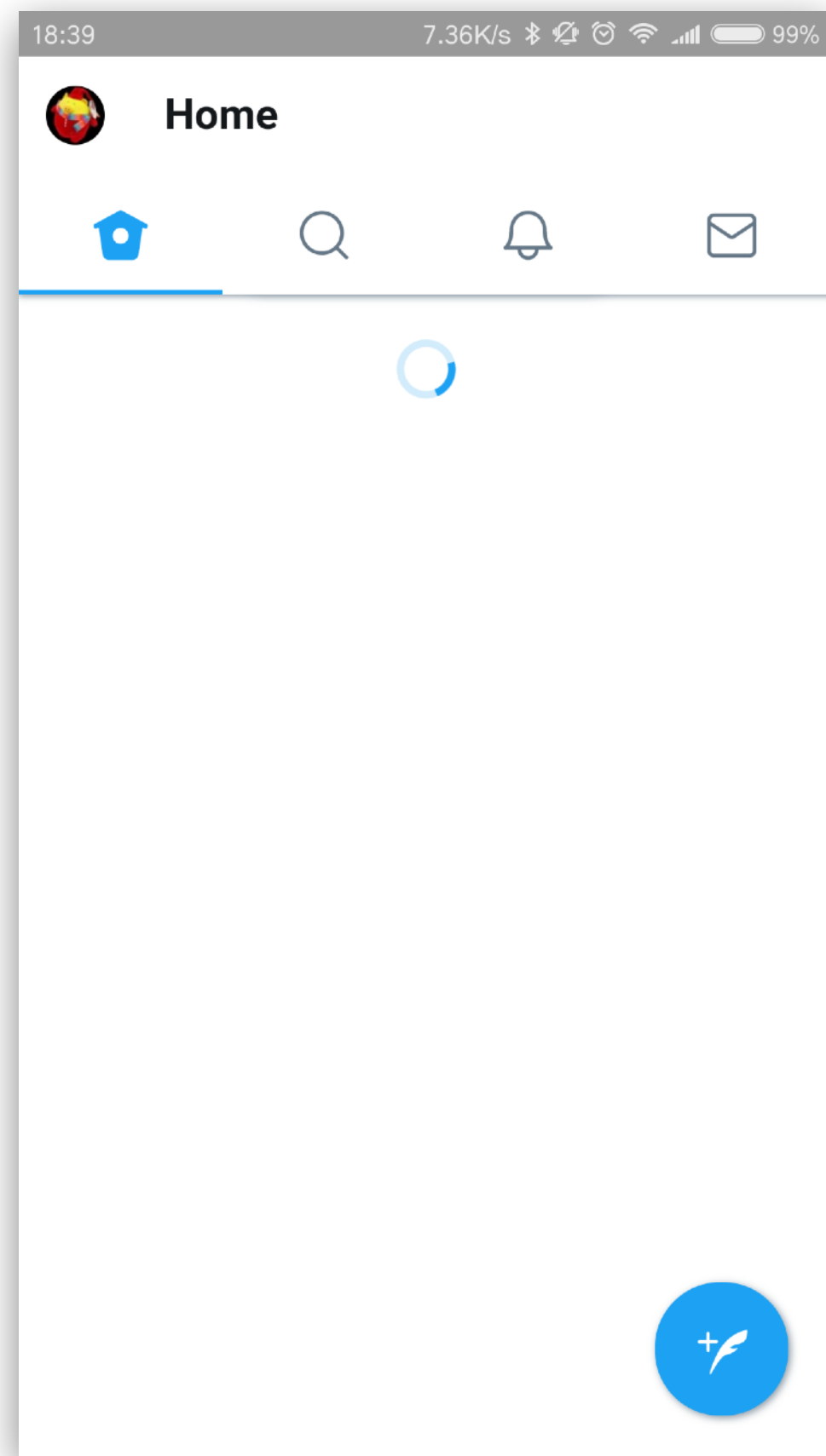
潘宇琪

百度前端高级工程师

TABLE OF CONTENTS 大纲

- App Shell 模型
- SPA 中的应用
- SSR 中的应用
- App Shell 性能
- Skeleton 方案

Native App



基础 UI



快速加载运行



离线可用

App Shell 模型



HTML + CSS + JS



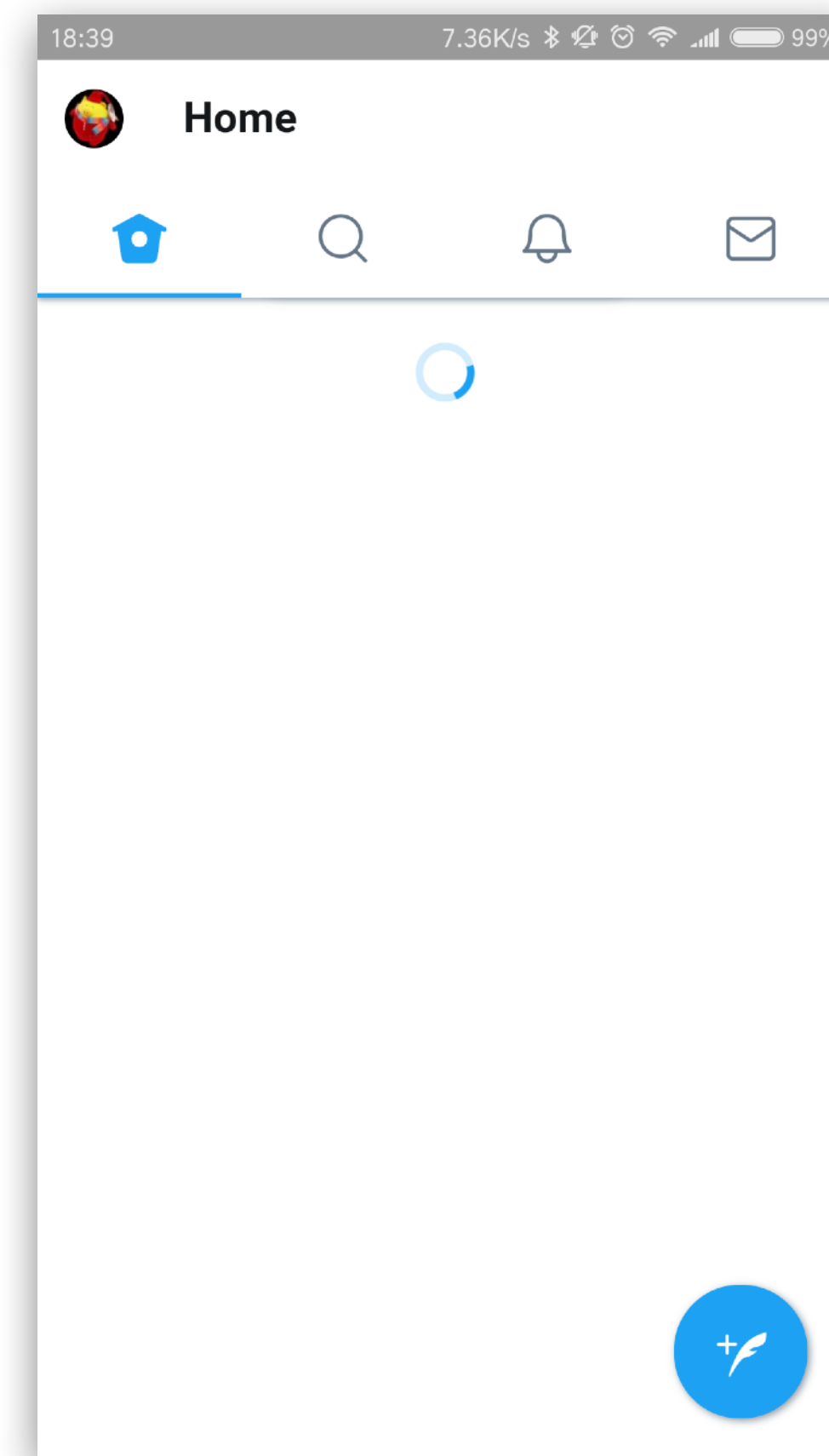
快速加载运行



动态加载后续路由



可缓存



不同架构下如何应用这一模型？

Server-side & Client-side



首屏加载速度快

同构应用 JS



后续页面跳转迅速

TABLE OF CONTENTS 大纲

- App Shell 模型
- **SPA 中的应用**
- SSR 中的应用
- App Shell 性能
- Skeleton 方案



SPA

&

PRPL 模式

PRPL 模式

Push

推送关键资源

Render

渲染初始路由

Precache

预缓存剩余路由

Lazyload

延迟加载并按需创建剩余路由

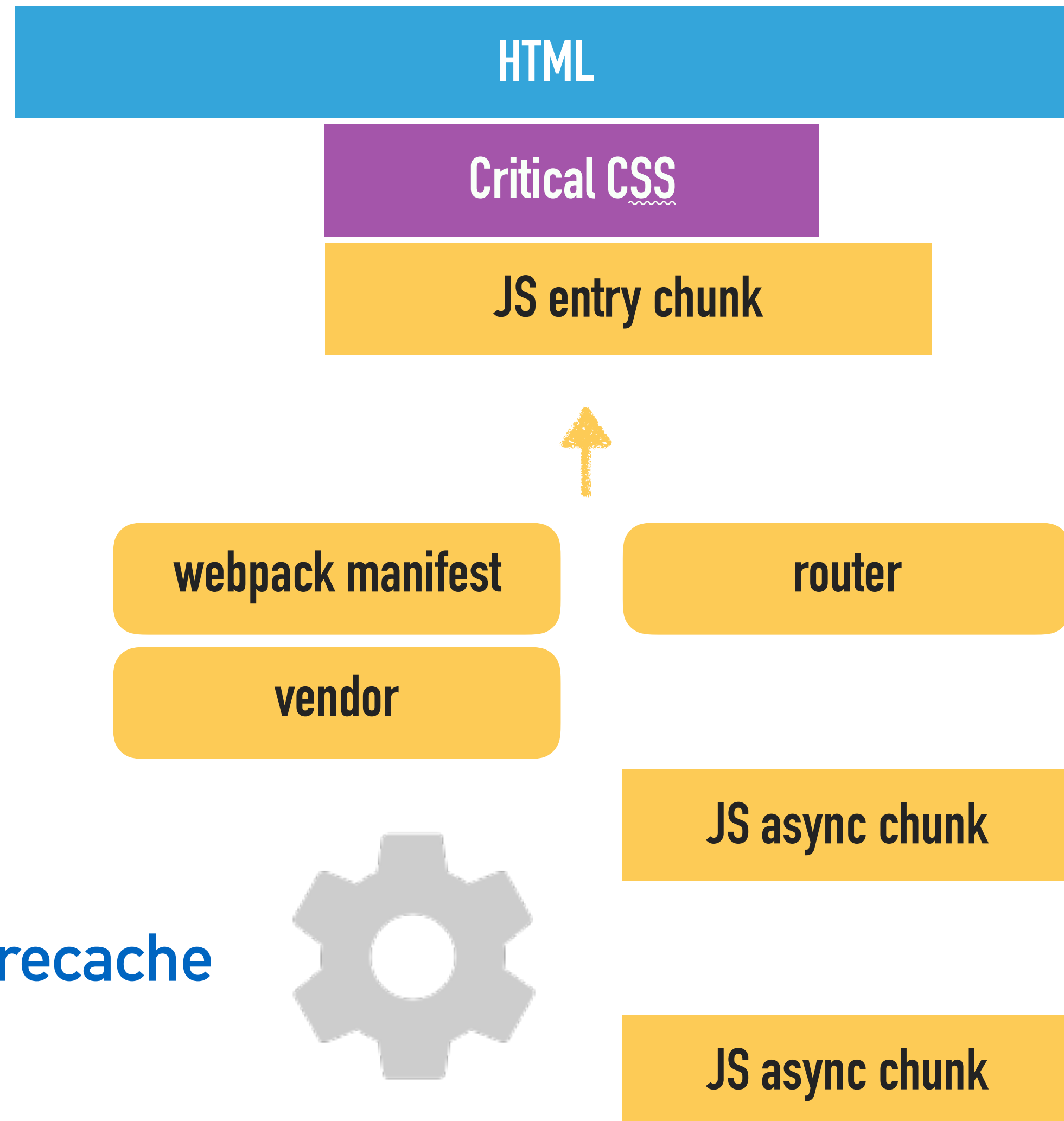
<https://developers.google.com/web/fundamentals/performance/prpl-pattern/>

Push

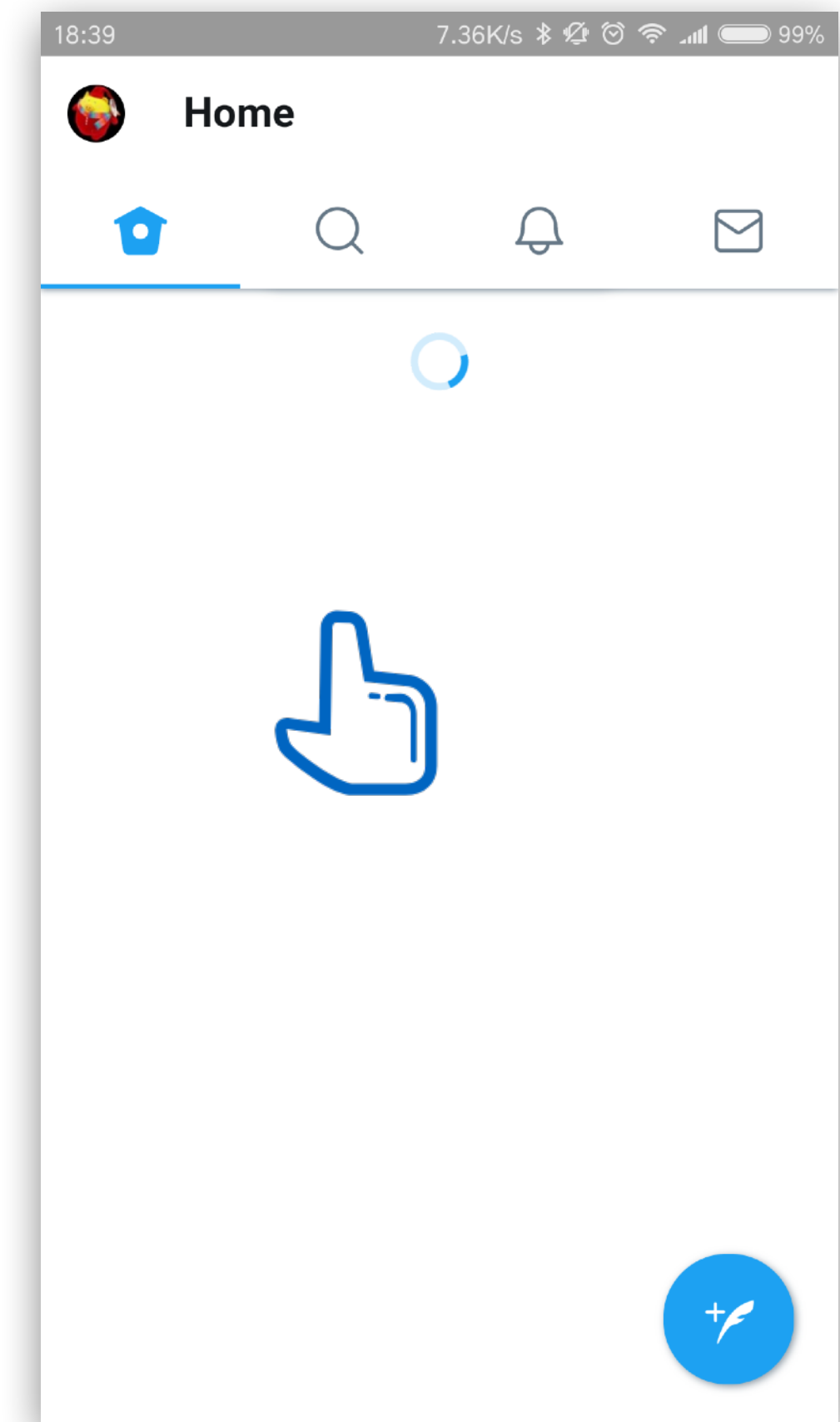
首屏

后续路由按需加载

Render



Lazyload



PRPL 相关技术



构建工具支持

HTTP

HTTP/2



前端路由



Service Worker 缓存

大量成熟框架 & 构建工具

PRPL 相关技术



构建工具支持

HTTP

HTTP/2

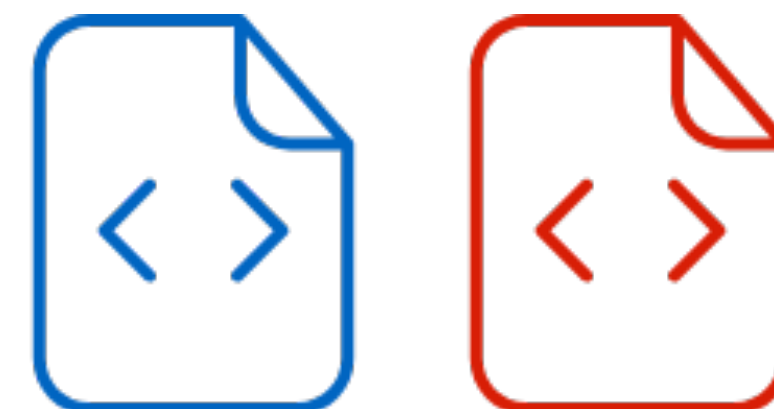
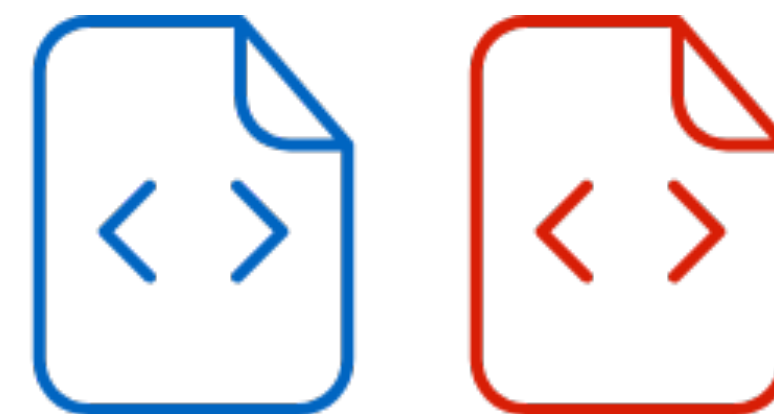


前端路由



Service Worker 缓存

代码分割



{



Dynamic import

```
import(/* webpackChunkName: 'my-view1' */  
  './my-view1')  
  .then((myView1) => {  
    // ...  
  });
```

+

```
js  
├── app.js  
└── my-view1.js
```

CommonsChunkPlugin

PRPL 相关技术



构建工具支持

HTTP

HTTP/2

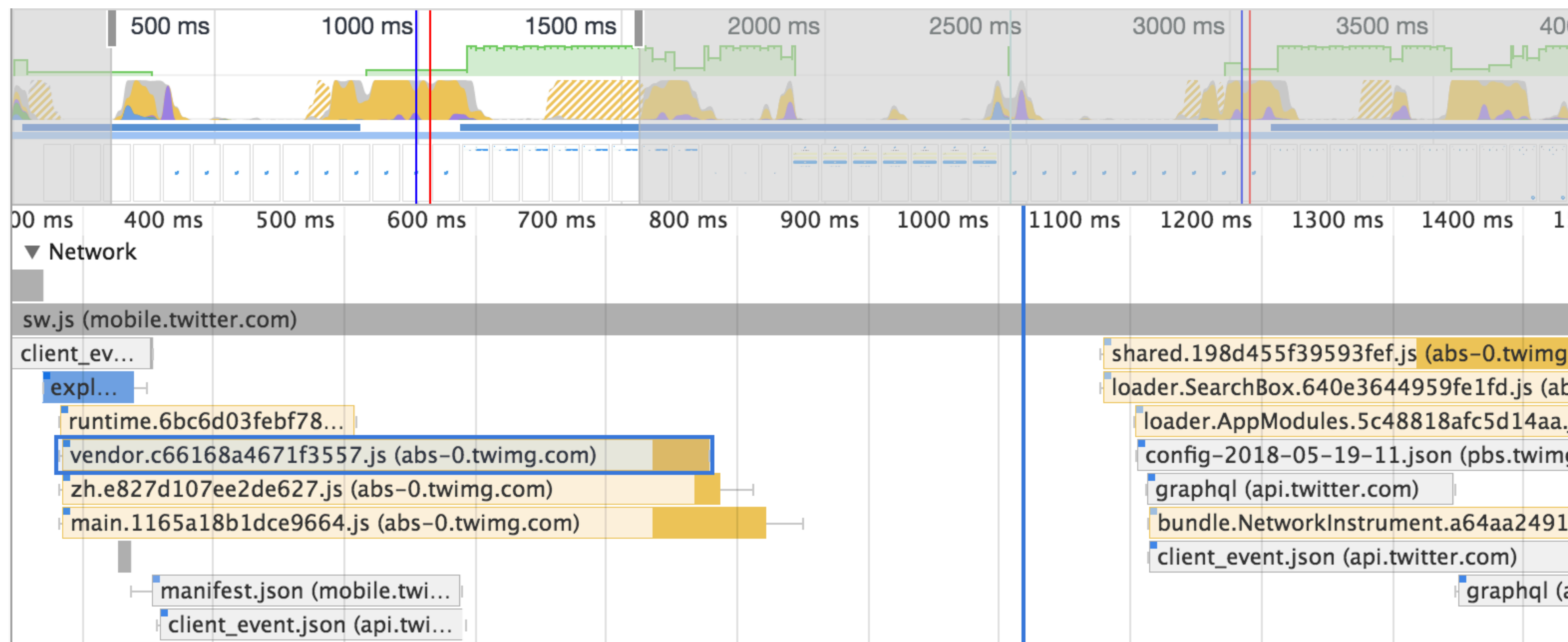


前端路由



Service Worker 缓存

推送静态资源



<link rel=preload>

```
<link rel="preconnect" href="//abs-0.twimg.com">
<link rel="preconnect" href="//api.twitter.com">
<link rel="preconnect" href="//o.twimg.com">
<link rel="preconnect" href="//pbs.twimg.com">
<link rel="preconnect" href="//t.co">
<link rel="preconnect" href="//video.twimg.com">
<link rel="dns-prefetch" href="//abs-0.twimg.com">
<link rel="dns-prefetch" href="//api.twitter.com">
<link rel="dns-prefetch" href="//o.twimg.com">
<link rel="dns-prefetch" href="//pbs.twimg.com">
<link rel="dns-prefetch" href="//t.co">
<link rel="dns-prefetch" href="//video.twimg.com">
<link rel="preload" as="script" crossorigin="anonymous" href="https://abs-0.twimg.com/responsive-web/web/tr/runtime.6bc6d03febf7874f.js">
<link rel="preload" as="script" crossorigin="anonymous" href="https://abs-0.twimg.com/responsive-web/web/tr/vendor.c66168a4671f3557.js">
<link rel="preload" as="script" crossorigin="anonymous" href="https://abs-0.twimg.com/responsive-web/web/tr/i18n/zh.e827d107ee2de627.js">
<link rel="preload" as="script" crossorigin="anonymous" href="https://abs-0.twimg.com/responsive-web/web/tr/main.1165a18b1dce9664.js">
```

PRPL 相关技术



构建工具支持

HTTP

HTTP/2



前端路由



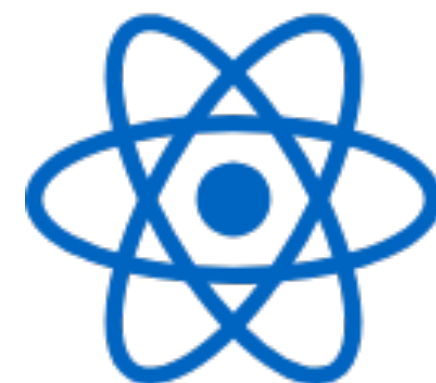
Service Worker 缓存

路由支持



API

```
var resolvedPageUrl =  
  this.resolveUrl('my-view1.html');  
  
this.importHref(resolvedPageUrl,  
  null,  
  this._importFailedCallback,  
  true  
);
```



异步组件

```
const MyView1 = () => import('./MyView1.vue');  
const router = new VueRouter({  
  routes: [  
    {  
      path: '/my-view1',  
      component: MyView1  
    }  
  ]  
});
```

PRPL 相关技术



构建工具支持

HTTP

HTTP/2



前端路由



Service Worker 缓存

资源预取

```
<link rel="prefetch" href="image.png">
```

<https://www.w3.org/TR/resource-hints/#prefetch>

Service Worker 预缓存

```
var cacheName = 'app-shell';
var filesToCache = [
  '/index.html',
  '/js/main.js',
  '/css/main.css',
  '/js/my-view1.js'
];

self.addEventListener('install', function(e) {
  e.waitUntil(
    caches.open(cacheName).then(function(cache) {
      return cache.addAll(filesToCache);
    })
  );
});
```

← App Shell

← 后续重要路由

构建时生成预缓存列表

Workbox

```
importScripts(' ./workbox-sw.prod.js ');  
importScripts(' ./precache-manifest.js ');
```

```
workbox.skipWaiting();  
workbox.clientsClaim();
```

```
workbox.precaching.precacheAndRoute(  
  self.__precacheManifest);
```

✓ 构建时生成预缓存列表

✓ 缓存更新

✓ 动态缓存...

SPA PRPL 模式

Push

HTTP/2 preload

Render

Router

Precache

Service Worker + Workbox

Lazyload

构建工具 + Router

<https://developers.google.com/web/fundamentals/performance/prpl-pattern/>

TABLE OF CONTENTS 大纲

- App Shell 模型
- SPA 中的应用
- **SSR 中的应用**
- App Shell 性能
- Skeleton 方案



SSR

&

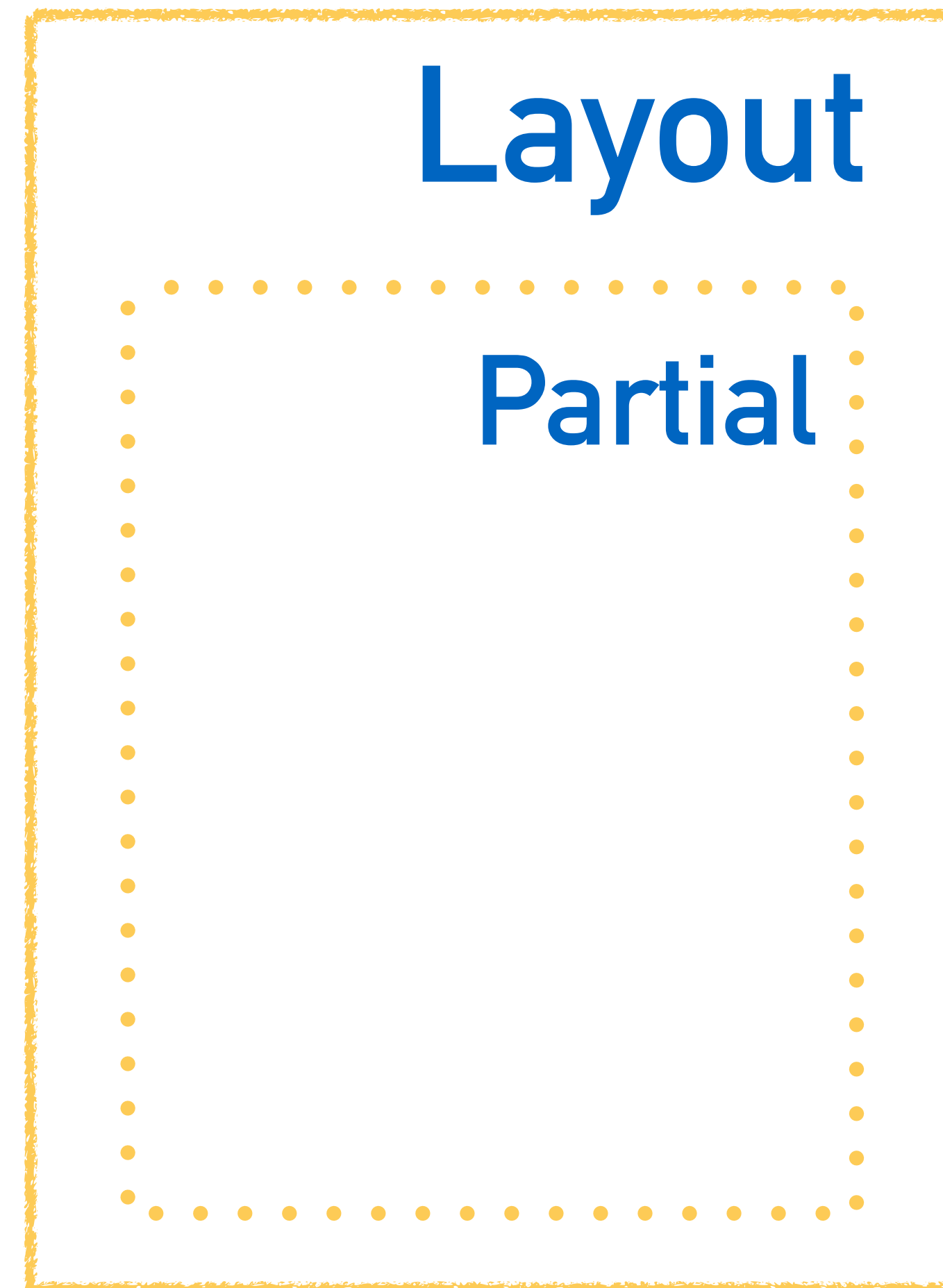
同构思路

模版 Layout + Partial



/home ←

/about ←



home

about

模版改造



/partial-home



home

/partial-about



about

/shell



Layout
Partial

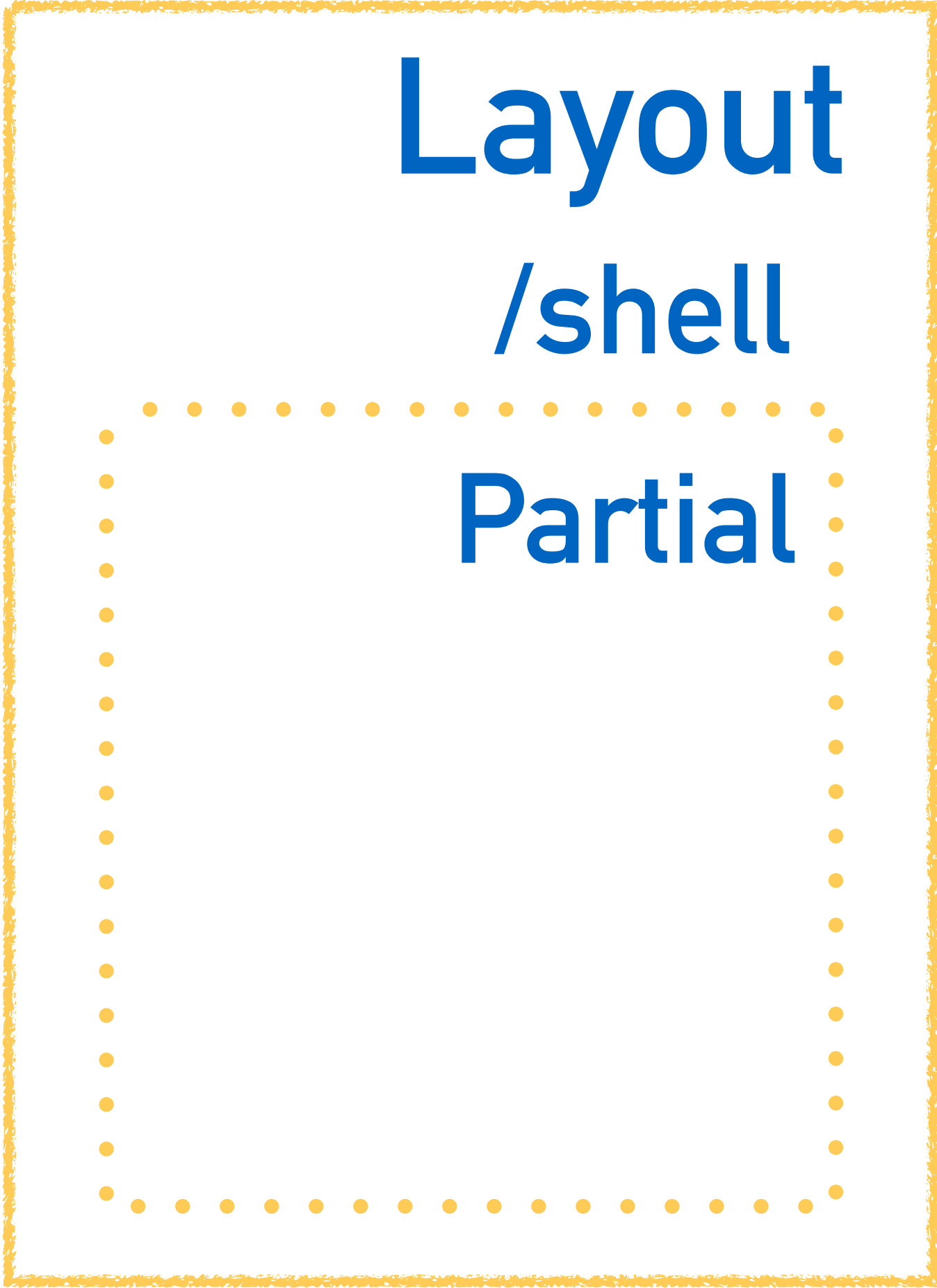
模版改造



/home



/about



/partial-home

/partial-about

同构思路

在 Service Worker 中执行

服务端路由和模版渲染



```
import templates from './templates';  
import routes from './routes';  
import shellLayout from './layout/shell.html';
```

/home → `app.get(routes.get('home'), async (req, res) => {`
渲染 → `res.write(shellLayout);`
layout → `const data = await requestData();`
`res.write(templates.partial('home', data));`
`res.end();`
`});`

请求数据
渲染 partial-home

<https://github.com/GoogleChromeLabs/so-pwa>
https://www.youtube.com/watch?v=X6yof_vIQnk

```
import templates from './templates';
import routes from './routes';
```



```
workbox.routing.registerRoute(
  routes.get('home'),
  workbox.streams.strategy([
    () => cacheStrategy.makeRequest({request: '/shell'}),
    async ({event, url}) => {
      const response = await apiStrategy.makeRequest({
        event,
        request: '/api'
      });
      const data = await response.json();
      return templates.partial('home', data);
    }
  ])
);
```

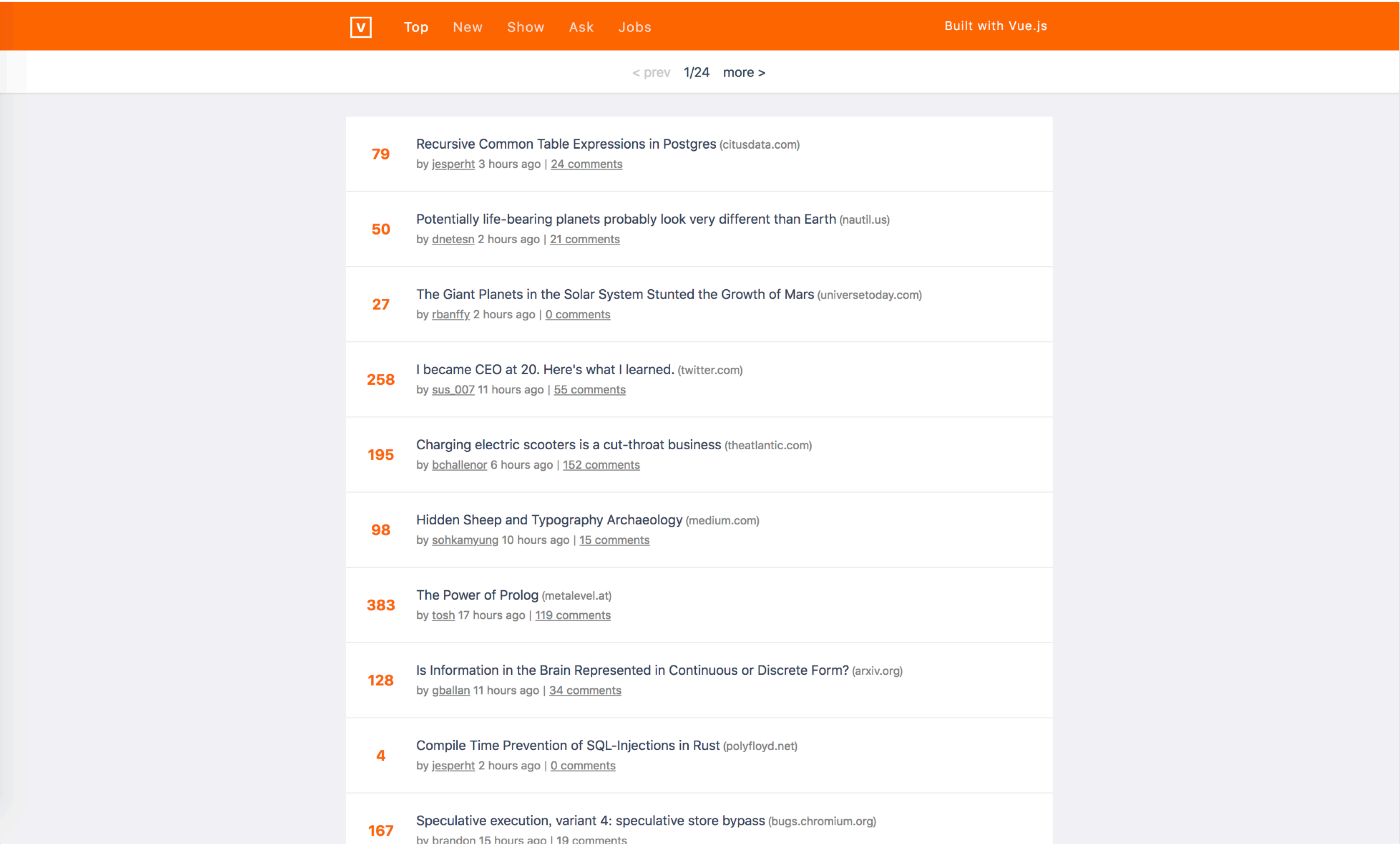
← /home

从 cache 中取出 layout

请求数据
渲染 partial-home

TABLE OF CONTENTS 大纲

- App Shell 模型
- SPA 中的应用
- SSR 中的应用
- App Shell 性能
- Skeleton 方案



Vue Hackernews

同构应用

https://github.com/vuejs/vue-hackernews-2.0



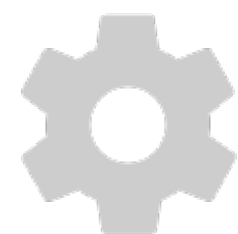
Fast 3G



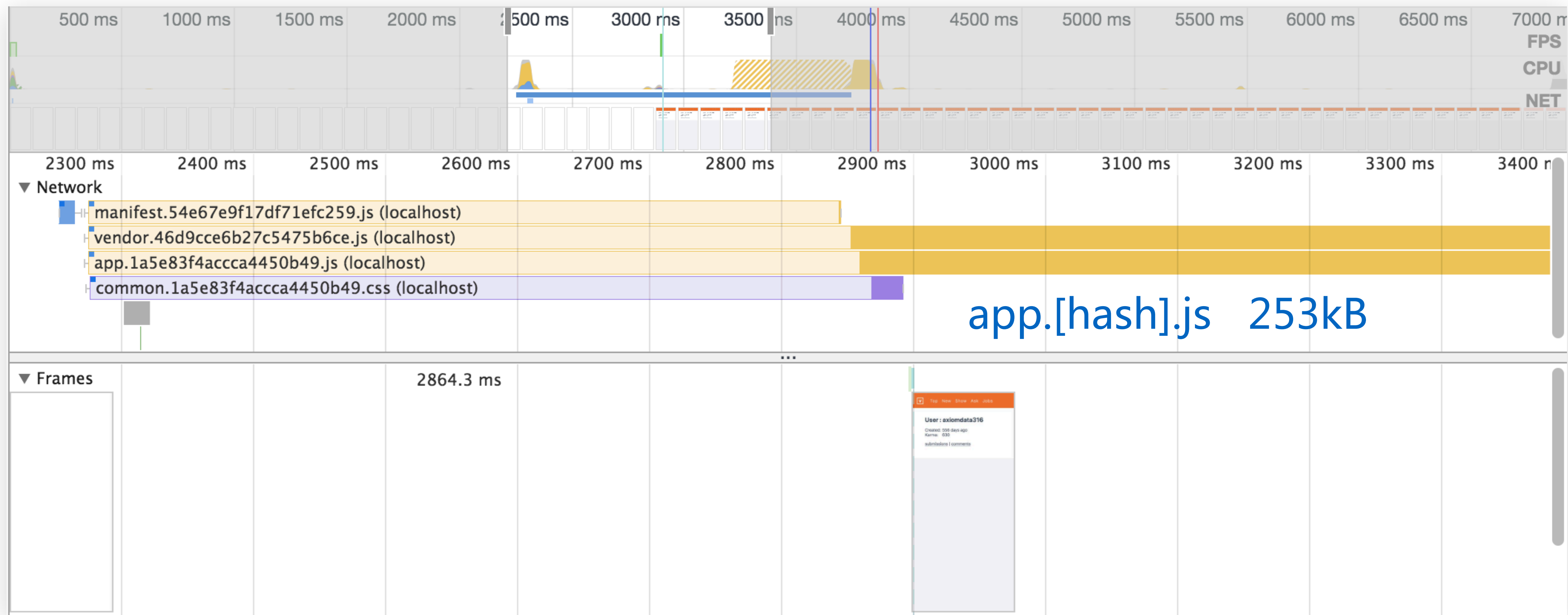
No route-based code splitting



FMP 2.9s



No service worker





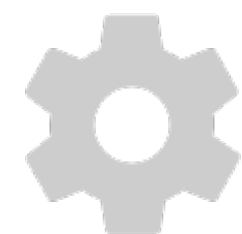
Fast 3G



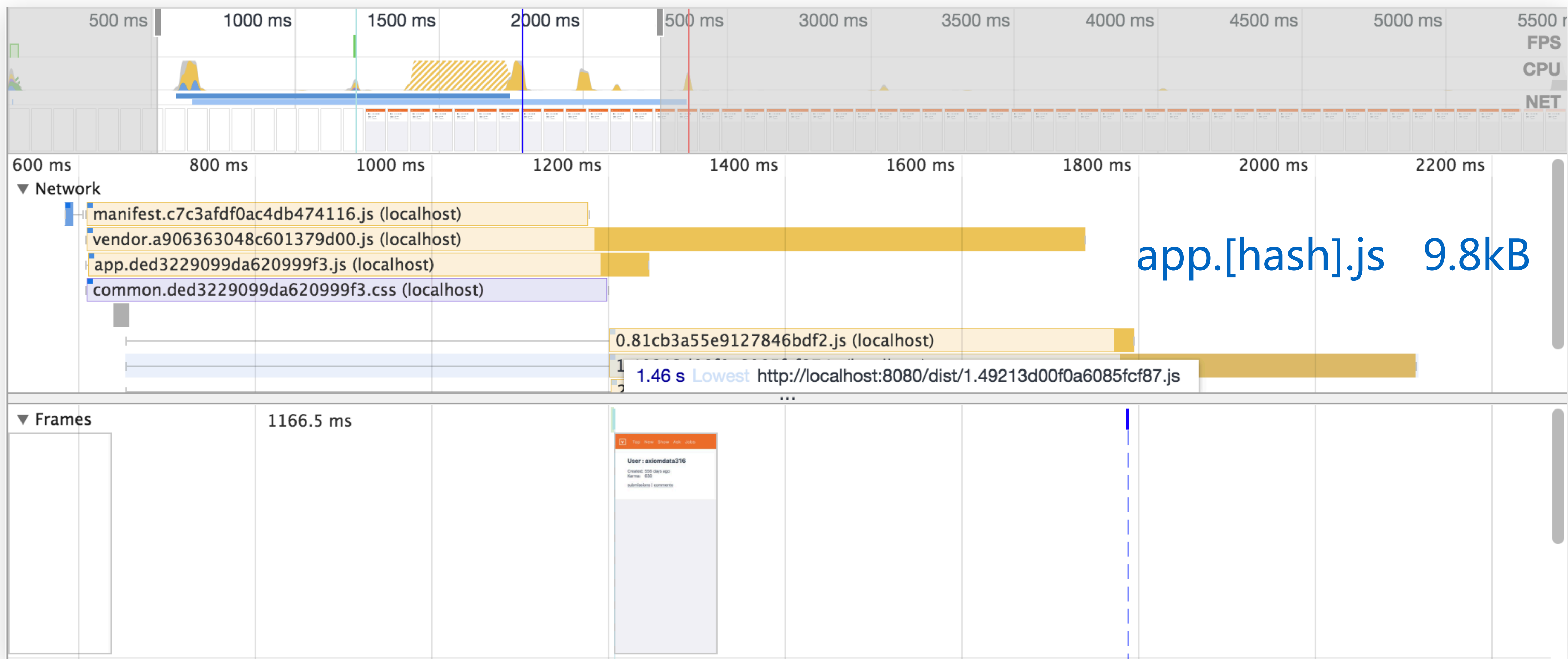
Route-based code splitting



FMP 1.2s



No service worker





Fast 3G



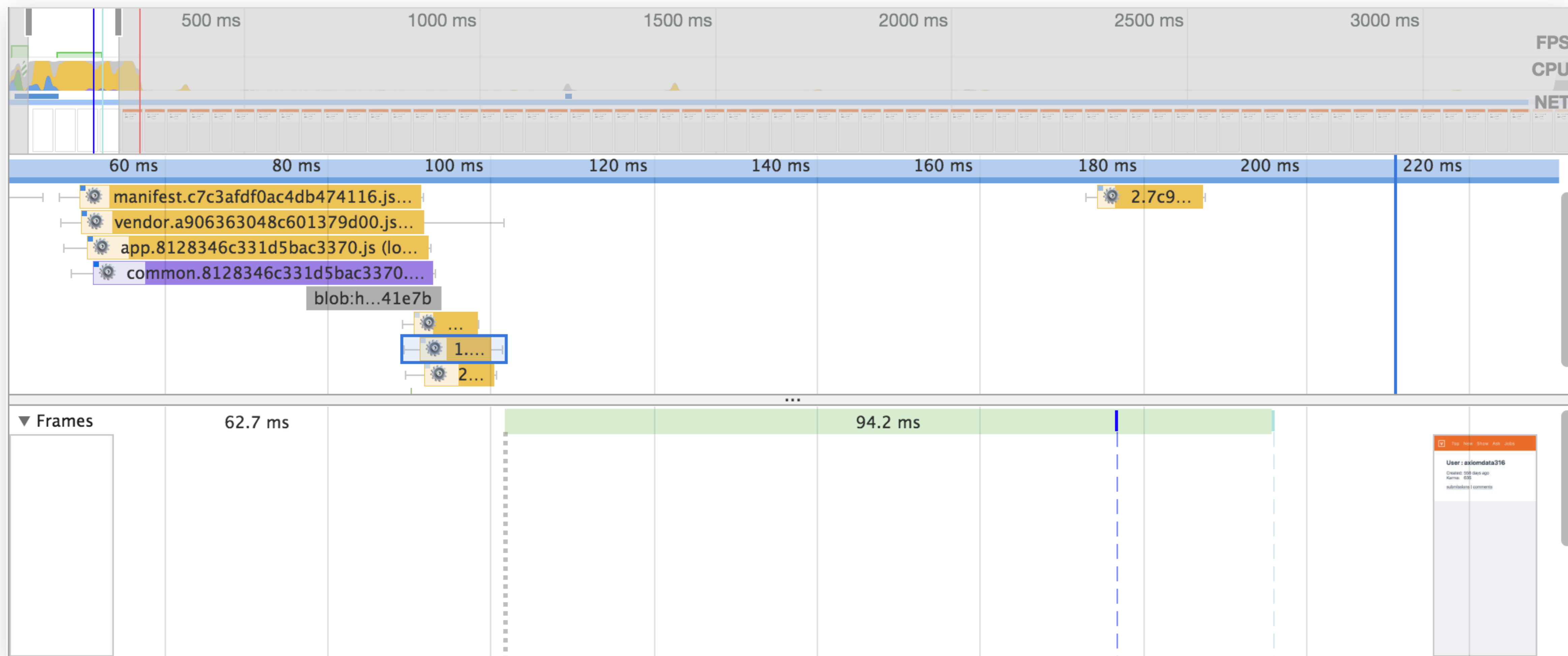
Route-based code splitting



FMP 0.22s



Service worker



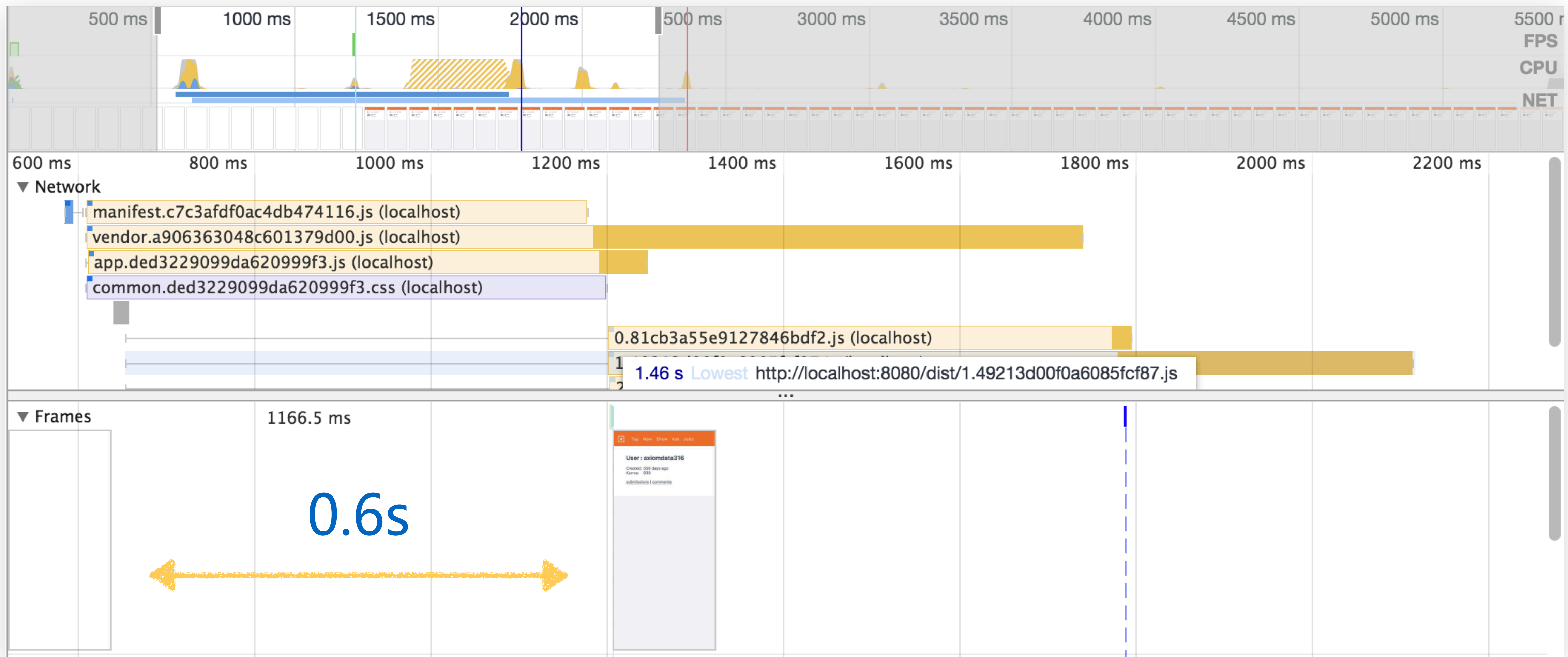


TABLE OF CONTENTS 大纲

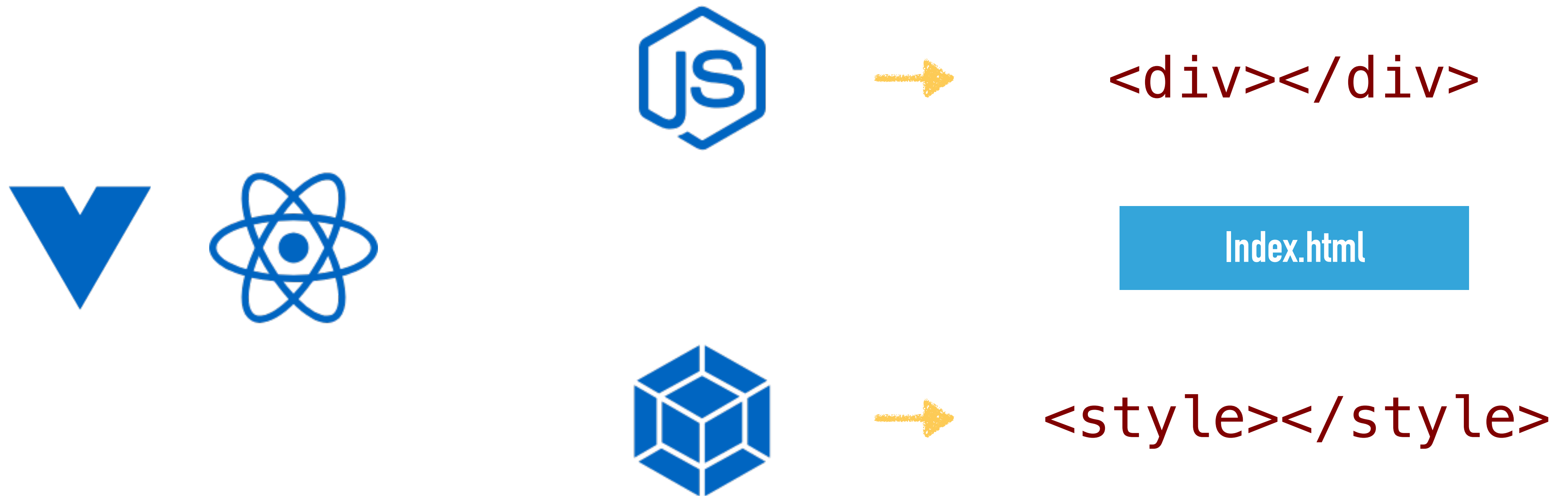
- App Shell 模型
- SPA 中的应用
- SSR 中的应用
- App Shell 性能
- **Skeleton 方案**

感知体验



构建时生成 CSS + HTML 片段 内联到 App Shell 中

SSR



<https://github.com/lavas-project/vue-skeleton-webpack-plugin>

<https://github.com/lavas-project/react-skeleton-webpack-plugin>



手动编写 Skeleton 组件 与框架 SSR 方案相关

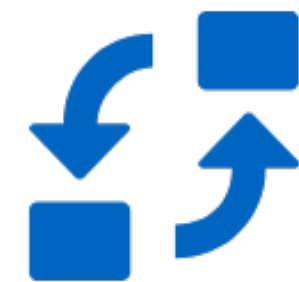
headless Chrome



在 Node.js 中打开页面



注入 JS



替换元素为占位符



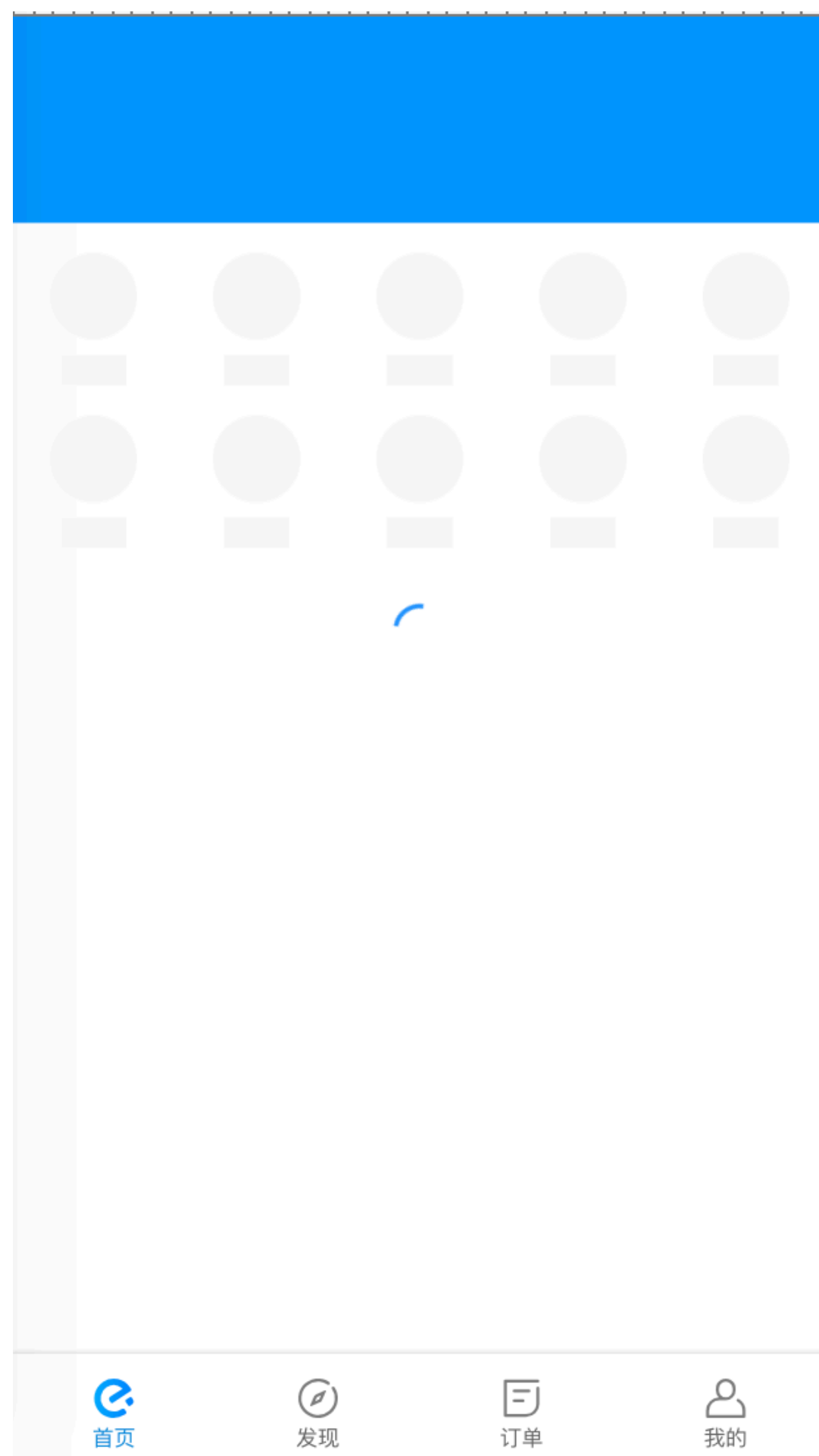
提取 HTML + CSS

<https://github.com/ElementFE/page-skeleton-webpack-plugin>

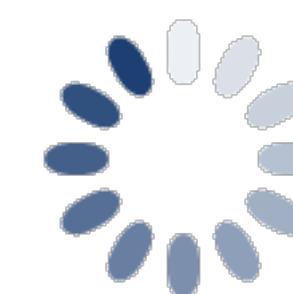
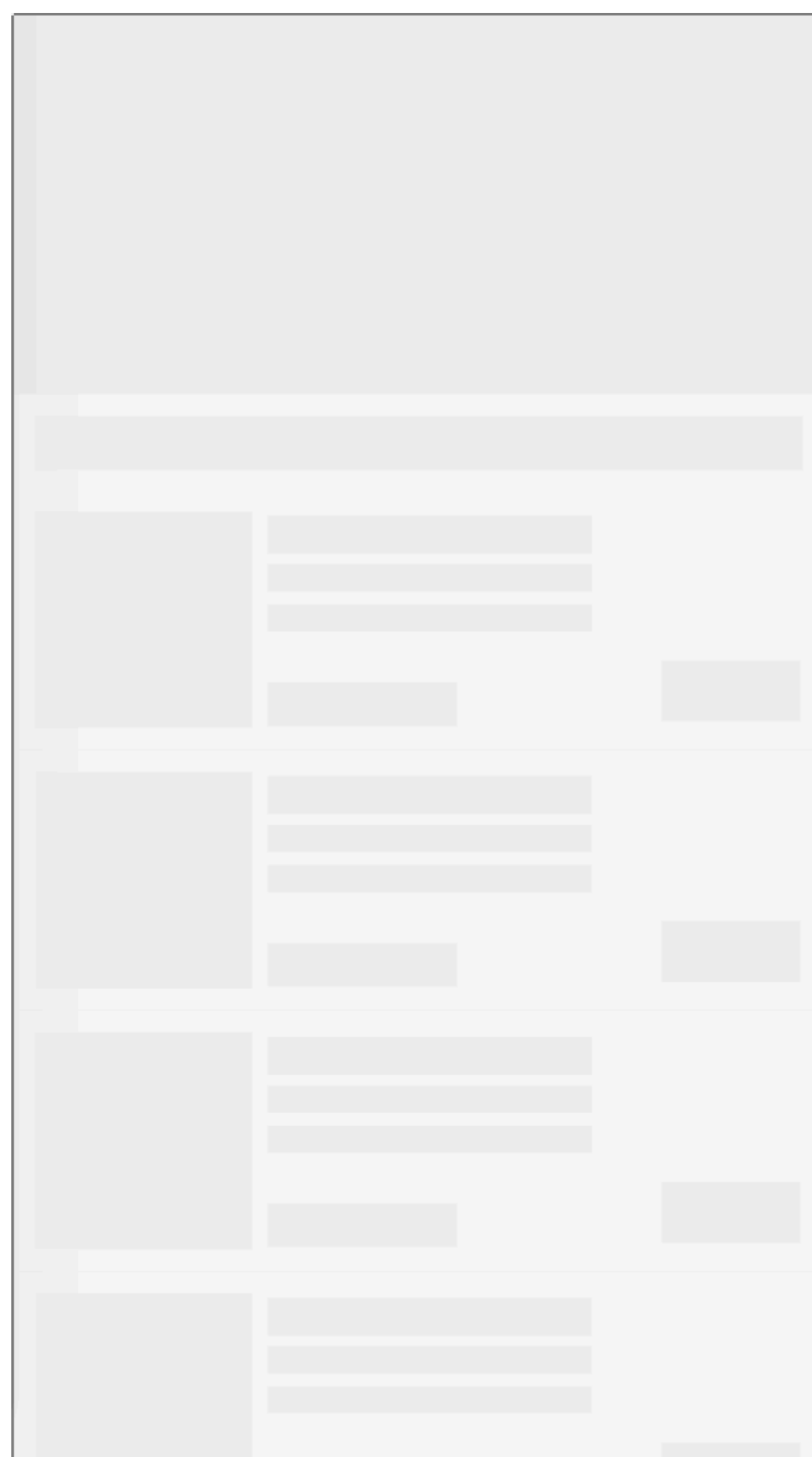


SPA 下多个页面差异很大

/home

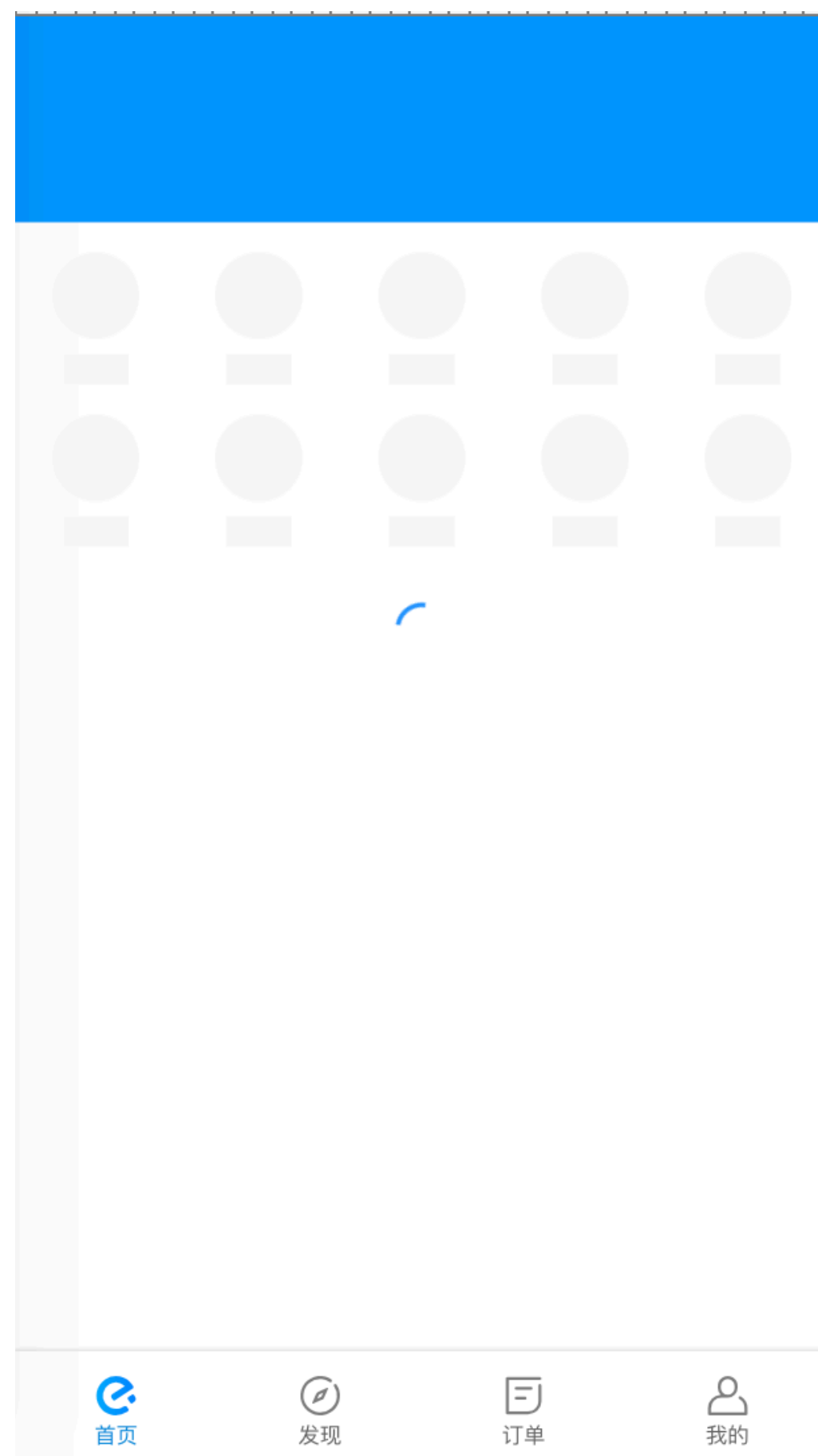


/detail



Index.html

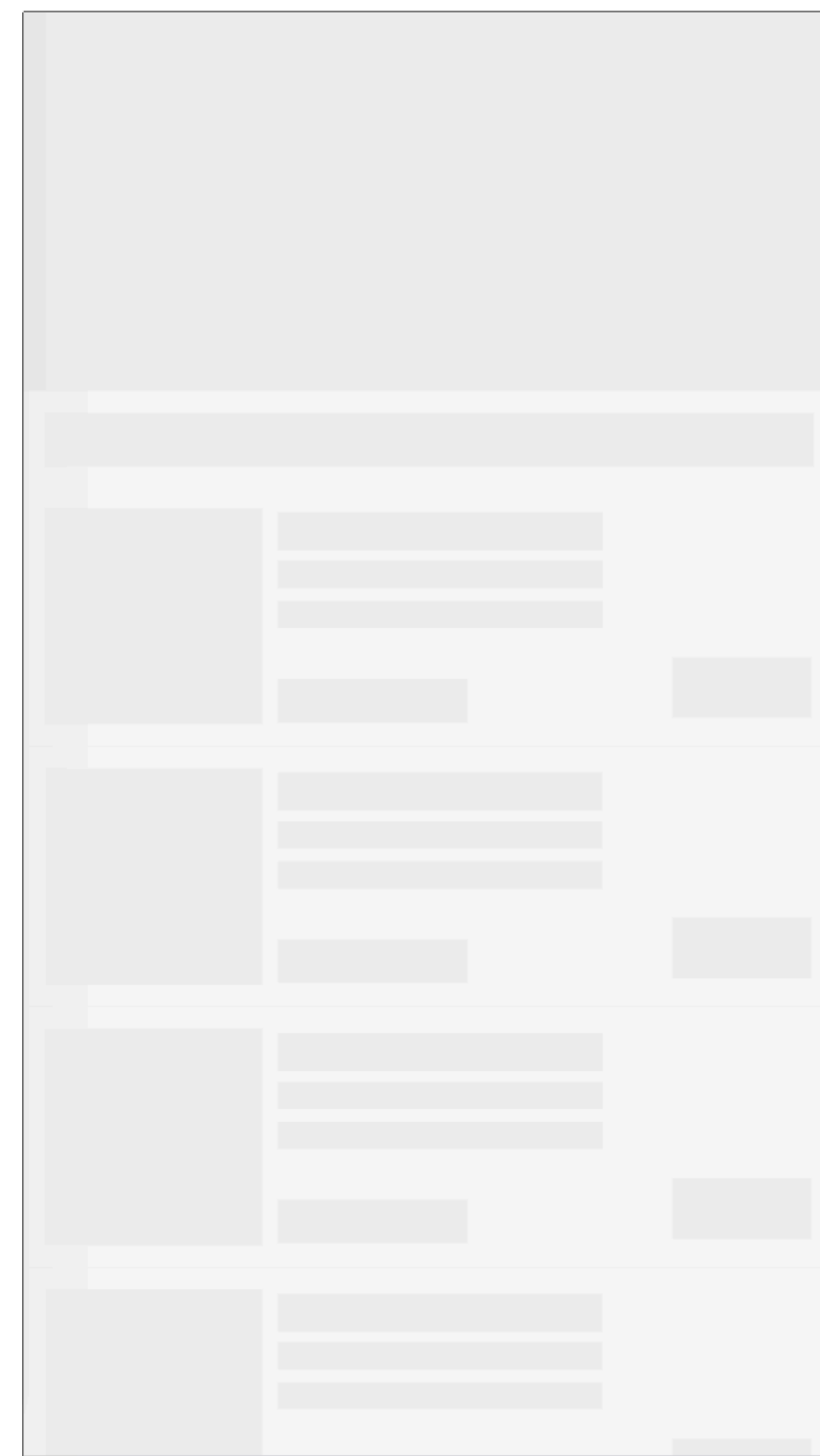
/home



Index.html



/detail





和 Loading 一样，
并不能减少 FMP
(first meaningful paint)

<https://developers.google.com/web/tools/lighthouse/audits/first-meaningful-paint>

THANKS