

**Я**ндекс

Яндекс

# SSR: DIY

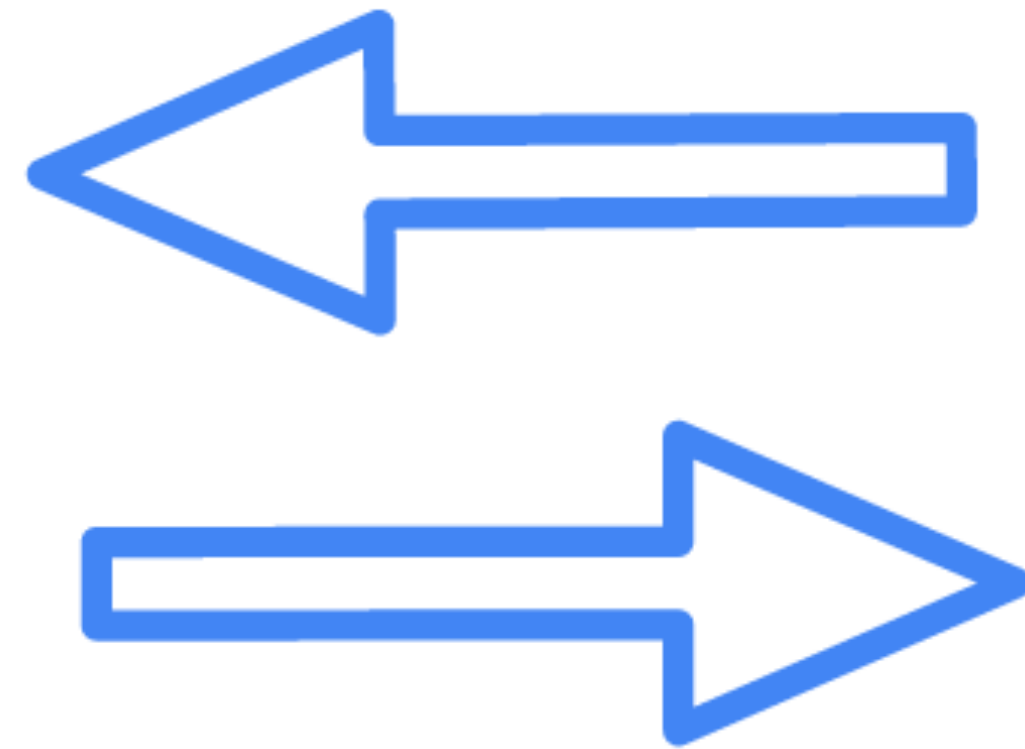
Джеймс Аквух, Software Engineer

# spinner.com

TUNING

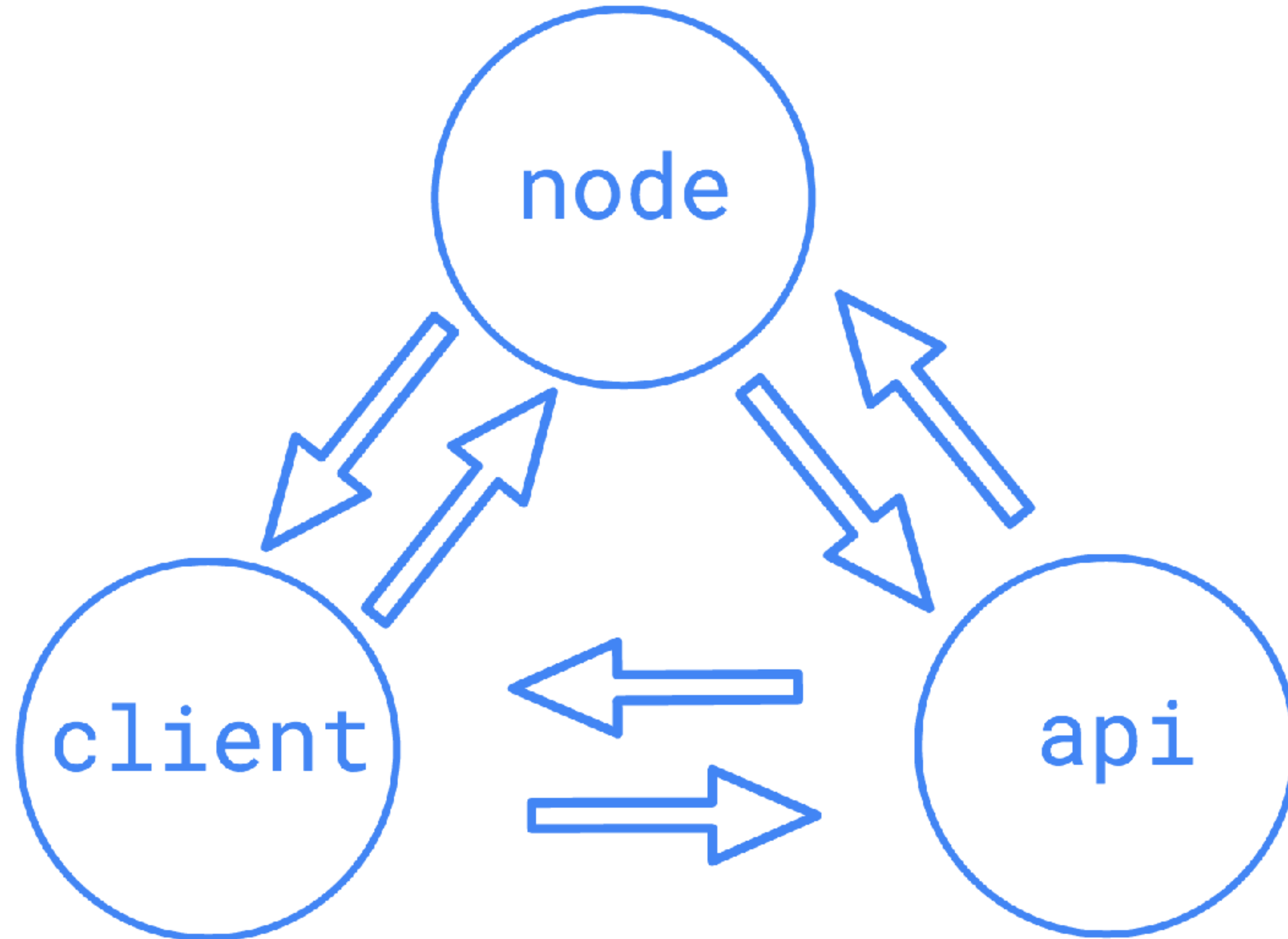
Waiting for aolradio

# SPA





# SSR



+

- › SEO
- › Никакого SPA-синдрома
- › Плавная деградация
- › Улучшение FMP, TTI
- › Кэширование HTML

```
{  
  "express": "^4.15.2",  
  "handlebars": "^4.0.7",  
  "webpack": "^2.4.1",  
  "babel-loader": "^7.0.0",  
}
```



**FMP**

**TTI**

# FMP



# TTI

# 0. SPA

```
{{!-- Document.hbs --}}
```

```
...
```

```
<link rel="stylesheet" href="dist/styles.css">
```

```
...
```

```
<div class="bicycle">  
  {{content}}  
</div>
```

```
...
```

```
<script src="dist/bundle.js"></script>
```

```
// entries/server.js
```

```
...
```

```
app.use(async (req, res, next) => {  
  const content = new SafeString(  
    `

</div>`  
  );  
  
  res.end(template({content}));  
  next()  
});


```

```
...
```

```
// entries/client.js
```

```
...
```

```
function start() {  
  const $container =  
    document.querySelector('div.bicycle');  
  const app = new Bicycle({fetch});  
  
  app.render().then(html => {  
    // XSS ! don't do that ;)  
    $container.innerHTML = html;  
  });  
}
```

```
...
```

```
// fetch.js
```

```
...
```

```
const delay = IS_SERVER ? 100 : 500;
```

```
export default function({offset = 0, count = 20} = {}) {  
  return new Promise(resolve => {  
    setTimeout(() =>  
      resolve(data.slice(offset, offset + count)),  
      delay  
    )  
  });  
}
```

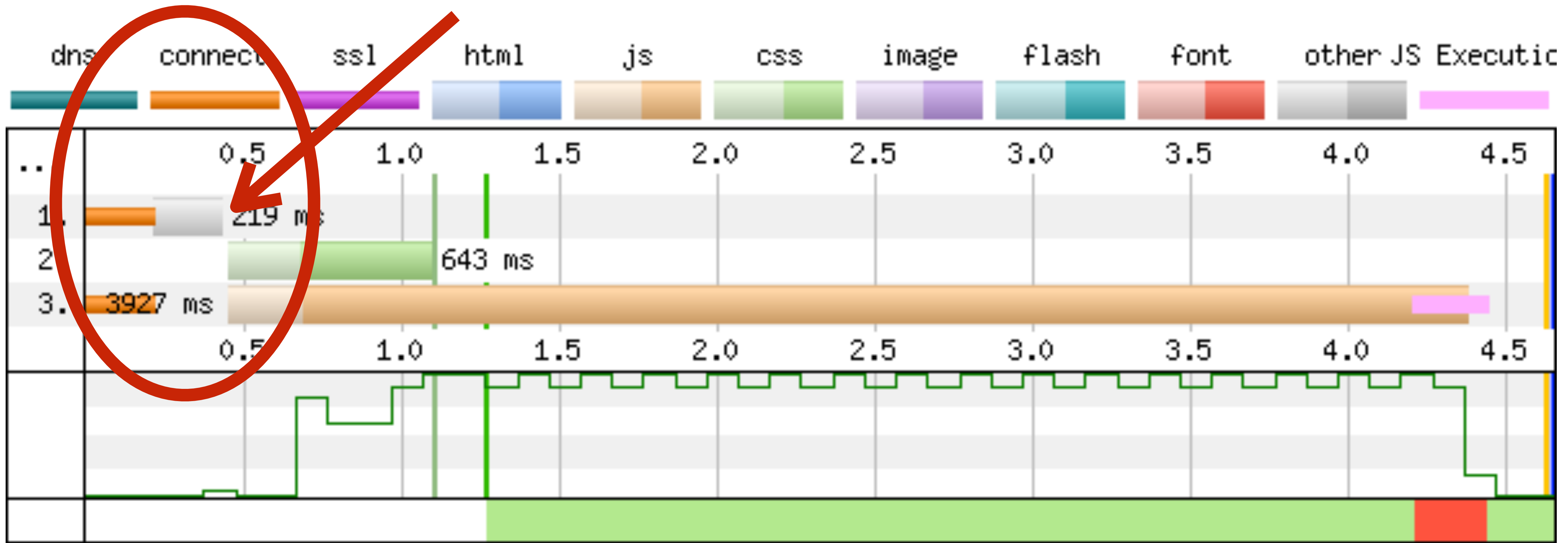
```
...
```

0-spa



0.0





**FMP: ~6300 MC**

**TTI: ~6300 MC**

# 1. SSR

```
// entries/server.js
```

```
...
```

```
app.use(async (req, res, next) => {  
  const app = new Bicycle({fetch});  
  const content = await app.render();
```

```
  res.end(template({content}));  
  next();
```

```
});
```

```
...
```

0-spa

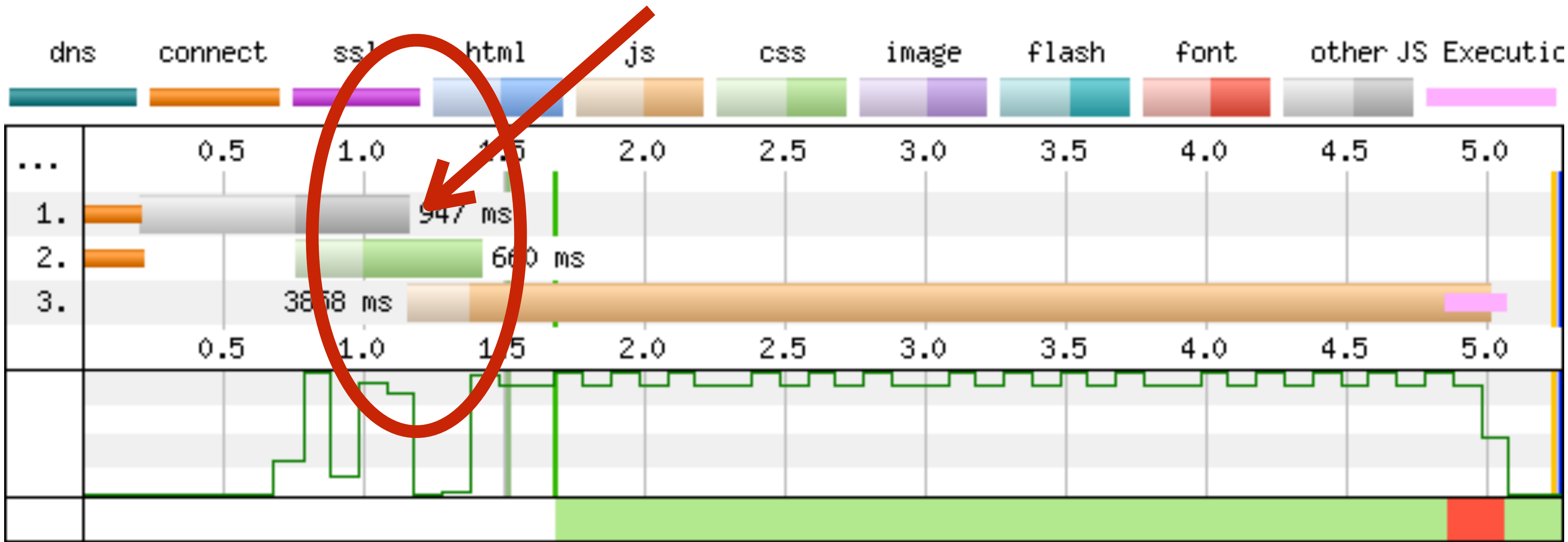


0.0

1-basic



0.0



**FMP: ~1700 MC**

**TTI: ~6900 MC**

\*

- > ~~Прямой доступ к DOM~~
- > ~~Унифицированный API~~
- > ~~Глобальные переменные~~
- > ~~Синглтоны~~
- > ~~Реактивность данных~~



# 2. Хэш-сумма

```
<div
```

```
  data-reactid=".157rq30hudc"
```

```
  data-react-checksum="556954499"
```

```
>
```

```
{{!-- Bicycle.hbs --}}
```

```
<div {{SSR_HASH_ATTR}}="{{uid}}" class="bicycle">  
  {{content}}  
</div>
```

```
// Bicycle.js
```

```
export class Bicycle {
```

```
...
```

```
  getUID () {  
    return ` ${this.count} - ${this.chunks} ` ;  
  }
```

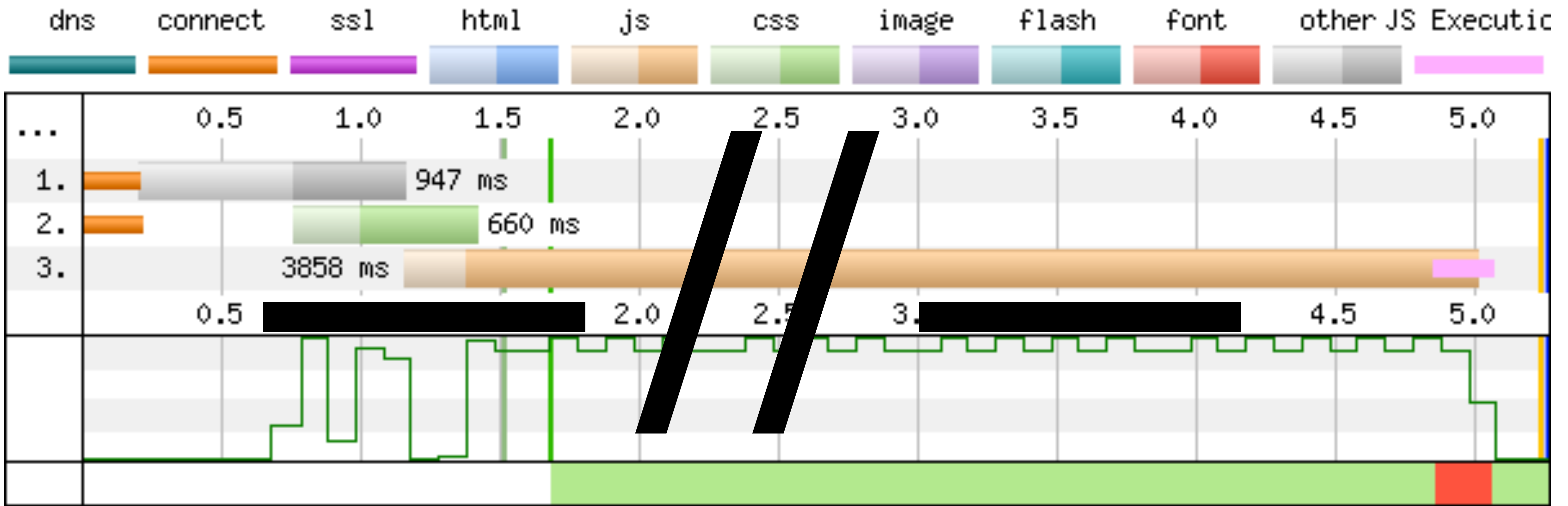
```
...
```

```
}
```

```
<div  
  class="bicycle"  
  data-ssr-hash="100-3"  
>
```

```
// entries/client.js
```

```
function start() {  
  ...  
  const app = new Bicycle({fetch});  
  
  const expectedUID = app.getUID();  
  const actualUID = $container.children[0]  
    .getAttribute(SSR_HASH_ATTR);  
  
  if (expectedUID !== actualUID) {  
    app.render().then(html => {  
      ...  
    });  
  }  
}
```



**FMP: ~1700 MC**

**TTI: ~5100 MC**



\*

uid — плохо

hash(VDOM) — хорошо

# 3. Кэширование

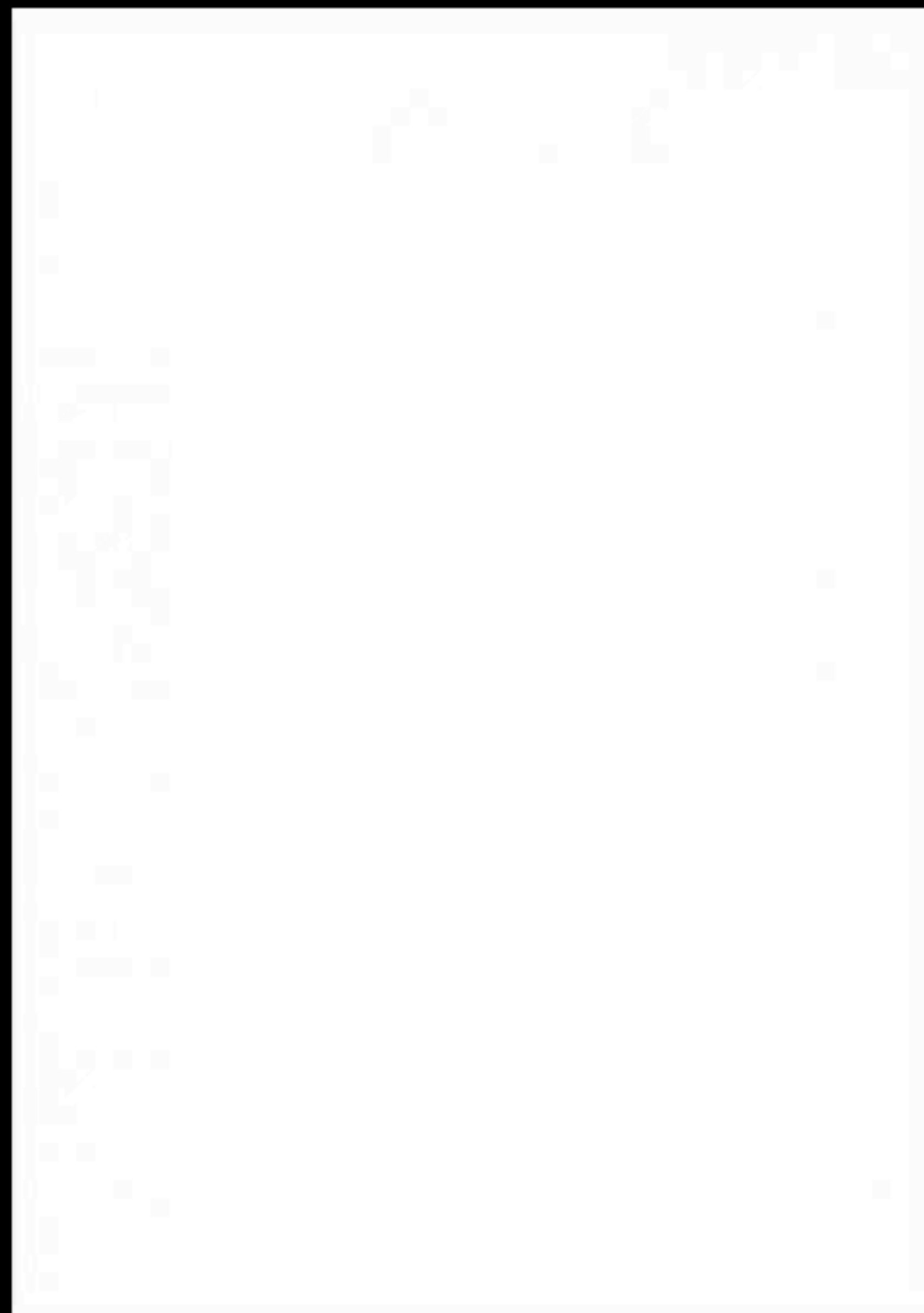
```
// Bicycle.js
import Cache from 'lru-cache';
const cache = new Cache(100);
...
  async render() {
    const uid = this.getUID();

    if (cache.has(uid)) {
      console.info('Cache hit. ');
      return cache.get(uid);
    }

    cache.set(uid, html);
    return html;
  }
...

```

1-basic



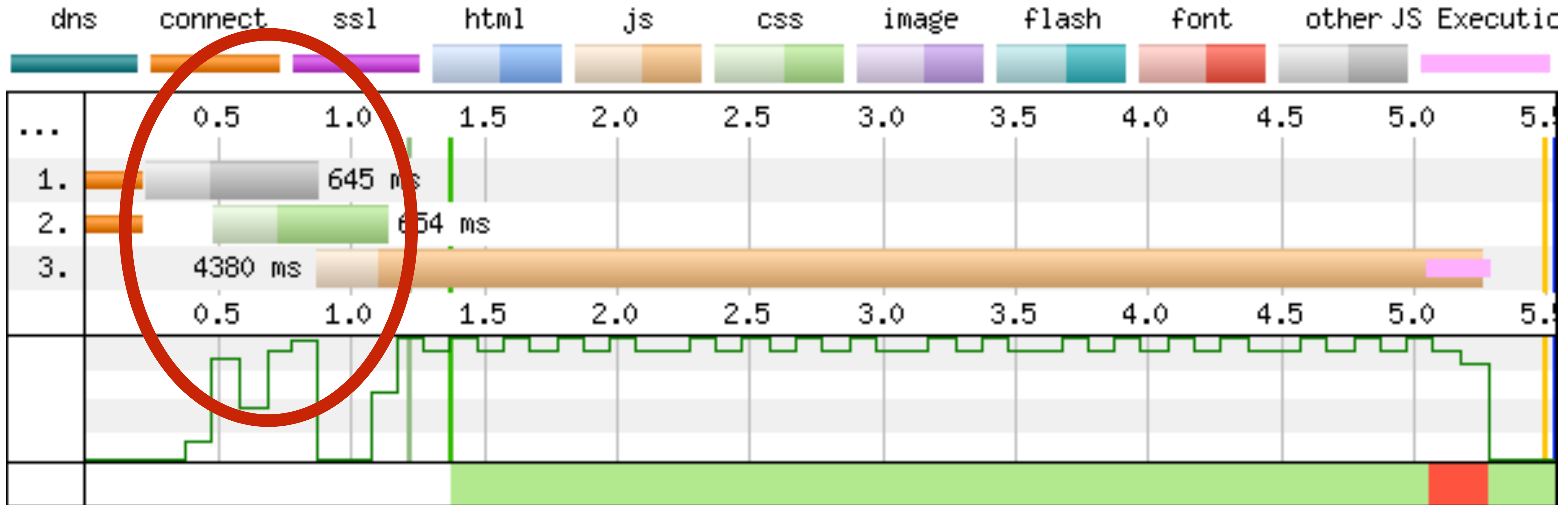
0.0

3-caching



0.0

# -300 MC



**FMP: ~1400 MC**

**TTI: ~4800 MC**

\*

Кэшируйте только для  
**неавторизованных пользователей**

# 4. Prefetch

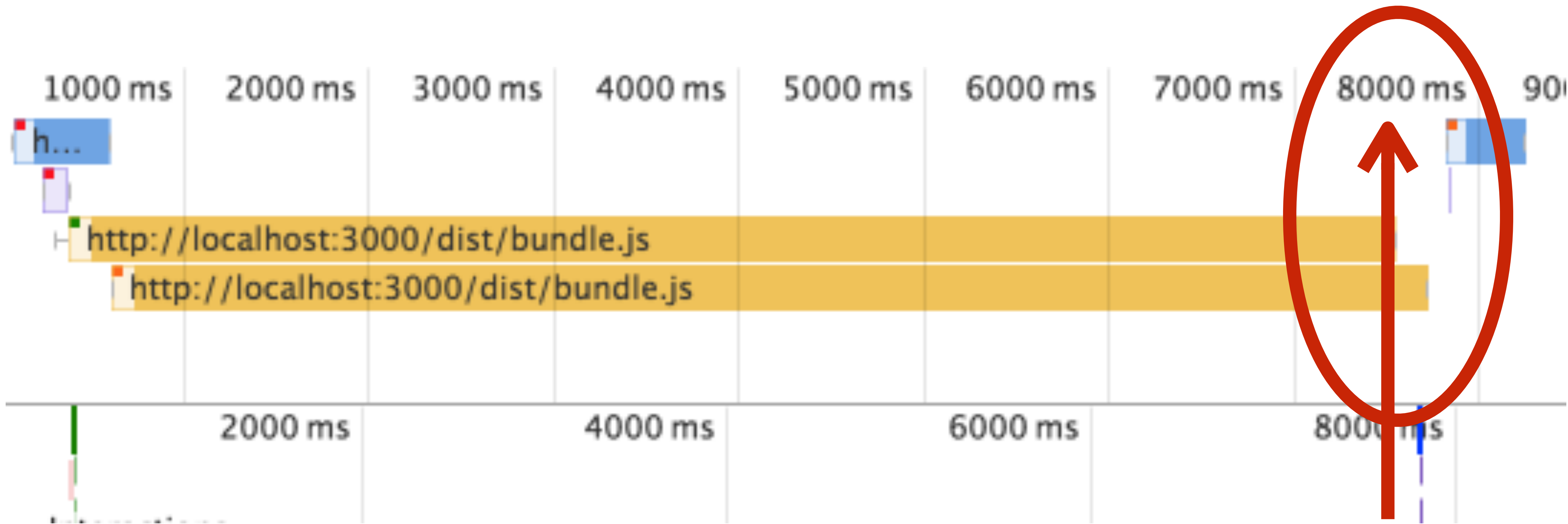


```
{{!-- Document.hbs --}}
```

```
<link rel="prefetch" as="script" href="dist/bundle.js">
```

**MDN: Link prefetching — это браузерный механизм, использующий время простоя браузера для предзагрузки ресурсов, которые могут скоро понадобиться пользователю**

**Xa. Xa. Xa.**



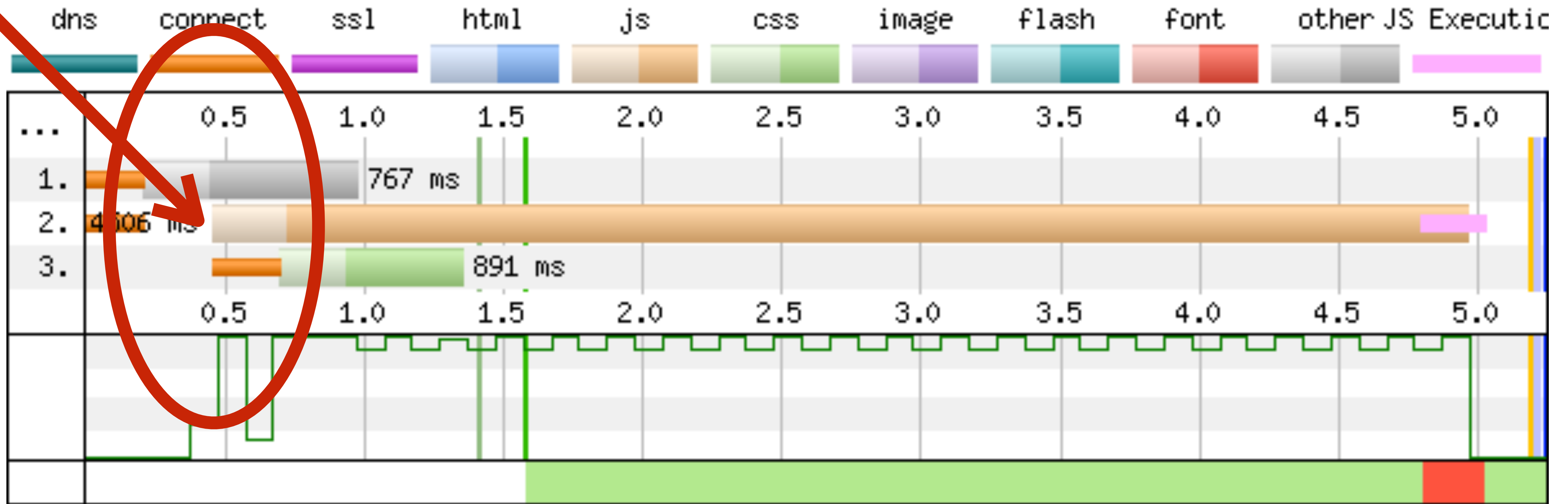
**FMP: ~1400 MC**

**TTI: ~7800 MC**

# 5. Preload

```
{{!-- Document.hbs --}}
```

```
<link rel="preload" as="script" href="dist/bundle.js">
```





**FMP: ~1600 MC**

**TTI: ~5000 MC**

# 6. Streams

```
export class GeneratorStream extends Readable {  
  
  constructor(generator) {  
    super();  
    this.iterator = generator();  
  }  
  
  async readAndEmit() {  
    ...  
  }  
  
}
```

```
let stop = false;
do {
  let {value, done} = this.iterator.next();

  // если текущее значение - Promise - ожидаем его
  if (value && value.then) {
    value = await value;
  }

  // прекращаем отдачу контента, если его больше нет
  // либо .push() вернул false
  // (что означает, что клиент не успевает
  // выкачать данные - backpressure)
  stop = !this.push(done ? null : value) || done;
} while (!stop)
```

```
// DocumentGenerator.js
```

```
export default function * () {  
  yield headerTemplate();  
  
  const app = new Bicycle ({fetch});  
  // Делегируем рендеринг корневому компоненту  
  yield * app.render();  
  
  yield footerTemplate();  
}
```

```
{{!-- Header.hbs --}}
```

```
<html>
```

```
<head>
```

```
  <title>SSR demo app</title>
```

```
  <link rel="stylesheet" href="dist/styles.css">
```

```
  <link rel="preload" as="script" href="dist/bundle.js">
```

```
</head>
```

```
<body>
```

```
<div class="title">
```

```
  <h1>Bitcoin rate</h1>
```

```
</div>
```

```
<div id="container">
```

```
// Bicycle.js
```

```
*render() {  
  const uid = this.getUID();  
  const {count, chunks} = this;  
  
  yield headerTemplate({SSR_HASH_ATTR, uid});  
  
  for (let i = 0; i < chunks; ++i) {  
    yield this.renderPartial({  
      offset: count * i, count  
    });  
  }  
  
  yield footerTemplate();  
}
```

```
{{!-- Footer.hbs --}}
```

```
</div>
```

```
<div class="copyright">
```

```
  Data taken from
```

```
  <a target="_blank" href="/price/">
```

```
    coindesk.com
```

```
  </a>
```

```
</div>
```

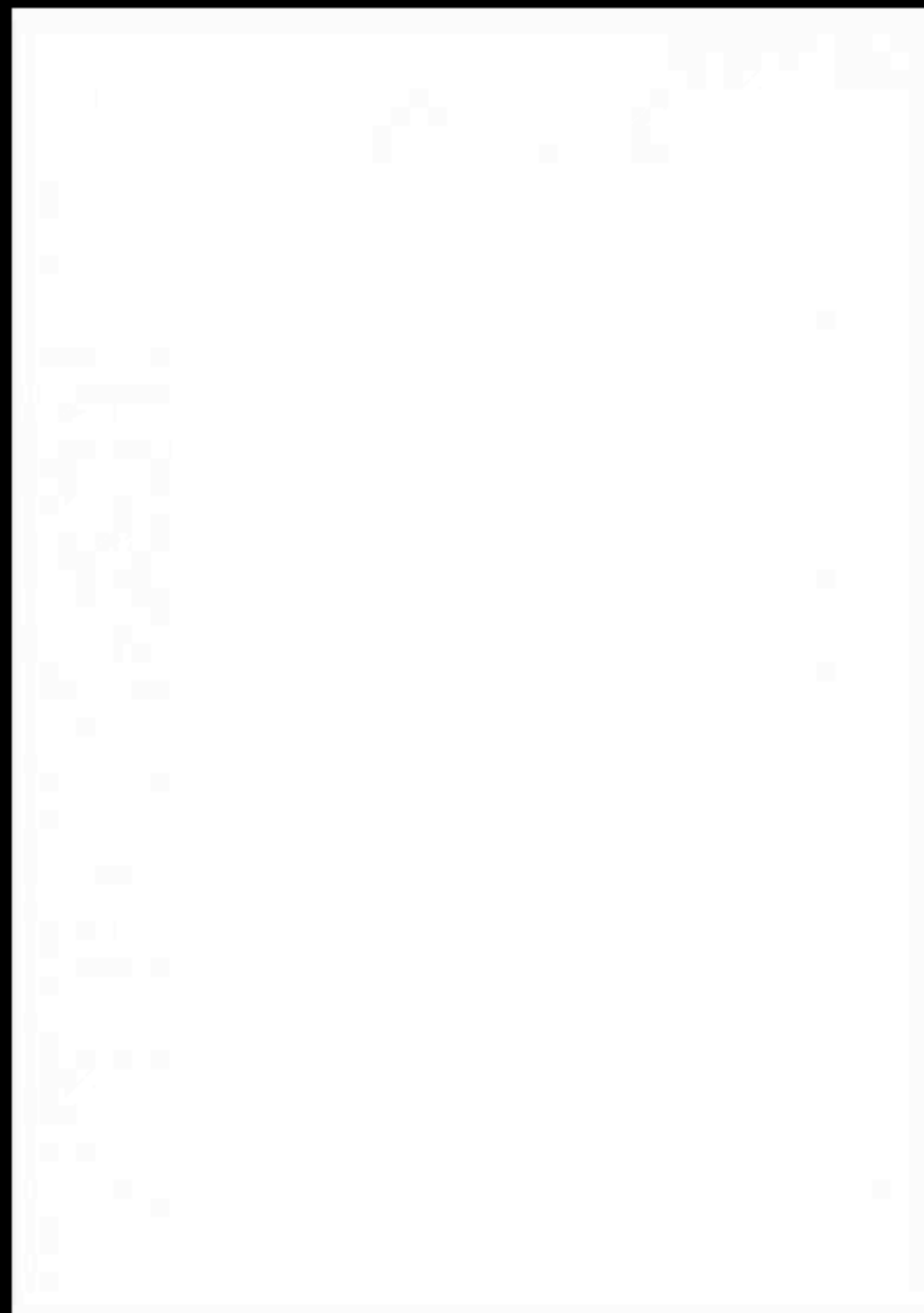
```
<script src="dist/bundle.js"></script>
```

```
</body>
```

```
</html>
```



1-basic

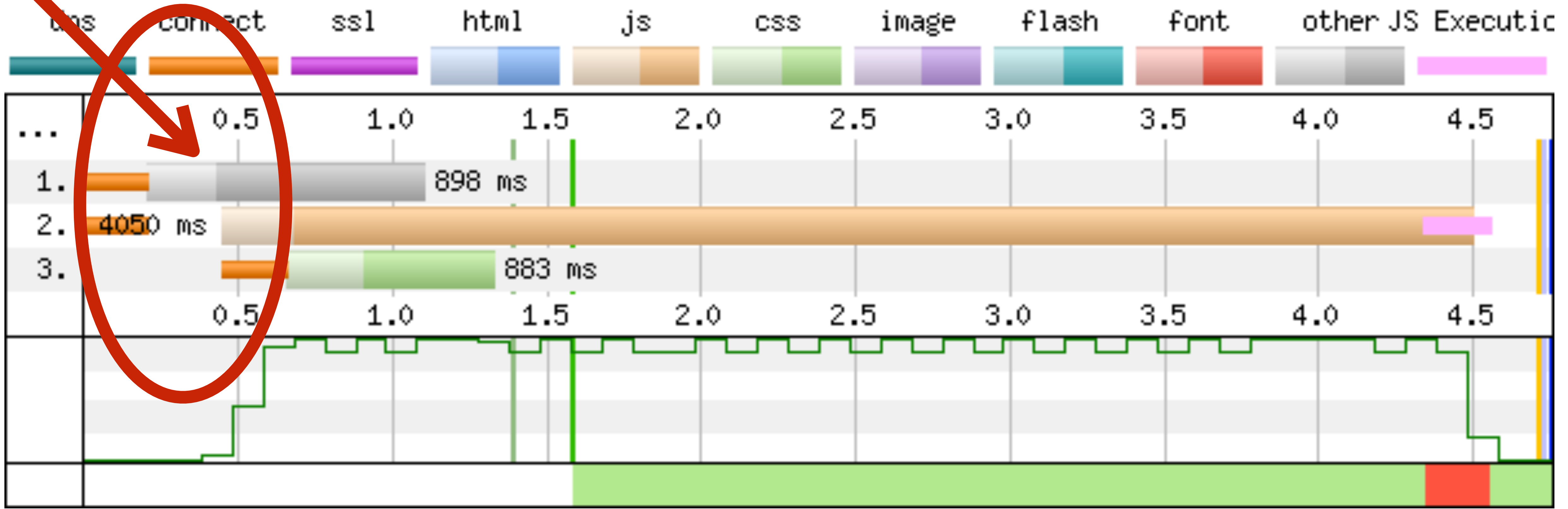


0.0

6-stream



0.0



**FMP: ~1600 MC**

**TTI: ~4600 MC**

\*

- › HTTP-статус всегда 200 :)
- › Меньший размер `gzip_buffers`
- › Поисковики (не ждут)

TL;DR

SPA → SSR

FMP: 6300 → 1600 MC

TTI: 6300 → 4600 MC

НИЧОСИ



Демо: [akwuh.me/ssr-demo](http://akwuh.me/ssr-demo)

Код: [github.com/jakwuh/ssr-demo](https://github.com/jakwuh/ssr-demo)

# Спасибо за внимание

Джеймс Аквух  
Software Engineer

 [t.me/dailytip](https://t.me/dailytip)

 [jakwuh@yandex-team.ru](mailto:jakwuh@yandex-team.ru)

 [jakwuh](https://github.com/jakwuh)

 [jakwuh](https://vk.com/jakwuh)