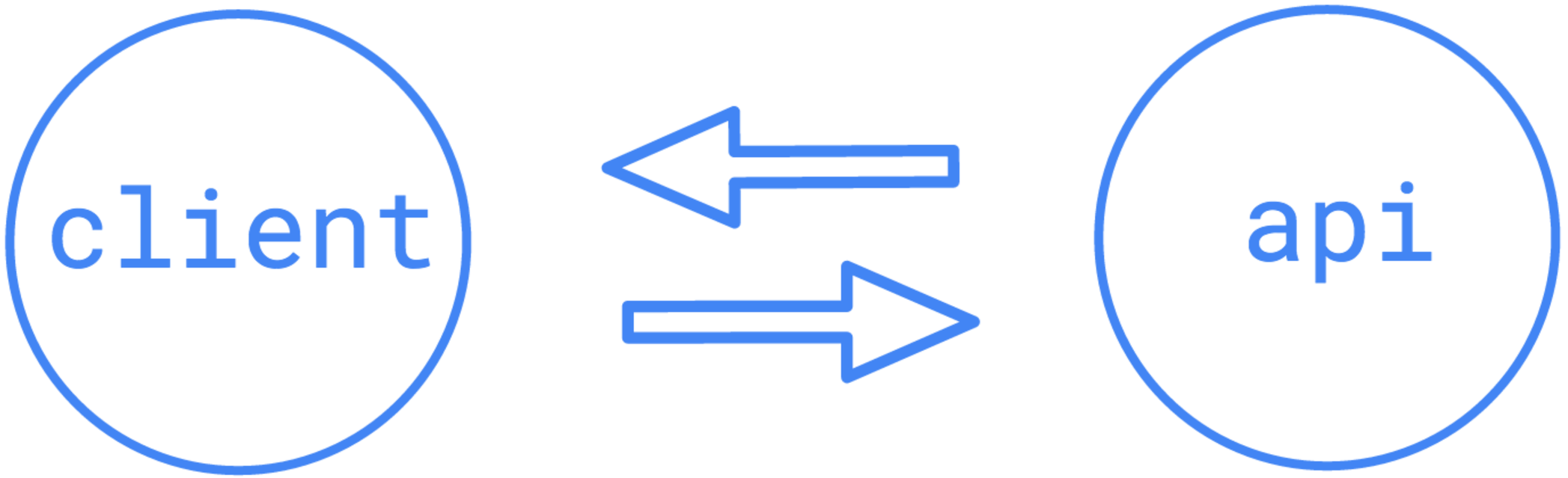

Яндекс

Яндекс

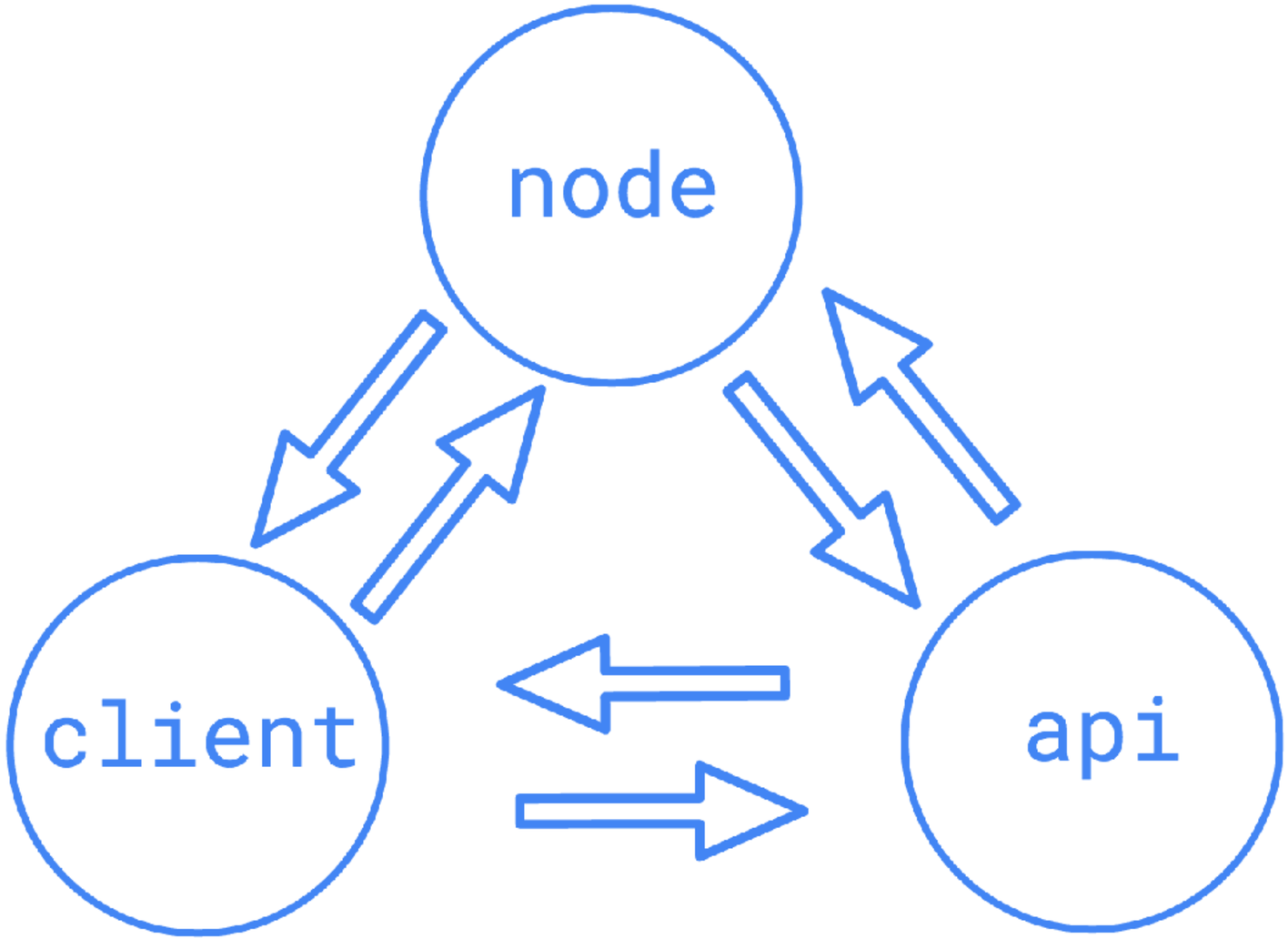
SSR: DIY

James Akwuh, Software Engineer

SPA



SSR



+

- SEO
- Code reuse -> no SPA syndrome
- Graceful degradation for free
- Improves FMP, TTI
- Better caching

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

FMP
TTI

* Regular 3G

0. SPA

https://goo.gl/ch55hU

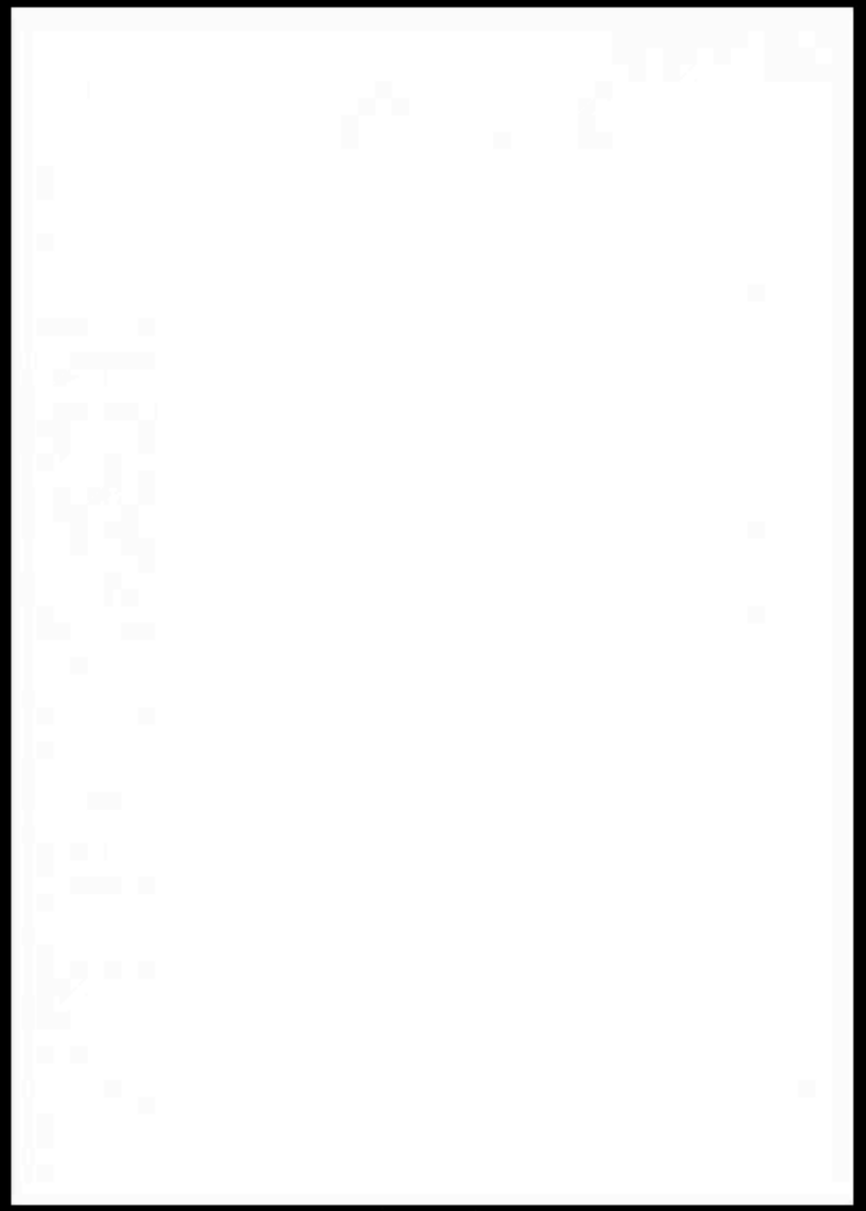
```
{{!-- Document.hbs --}}

...

<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 class="loader"></div>`
    );

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

...
```

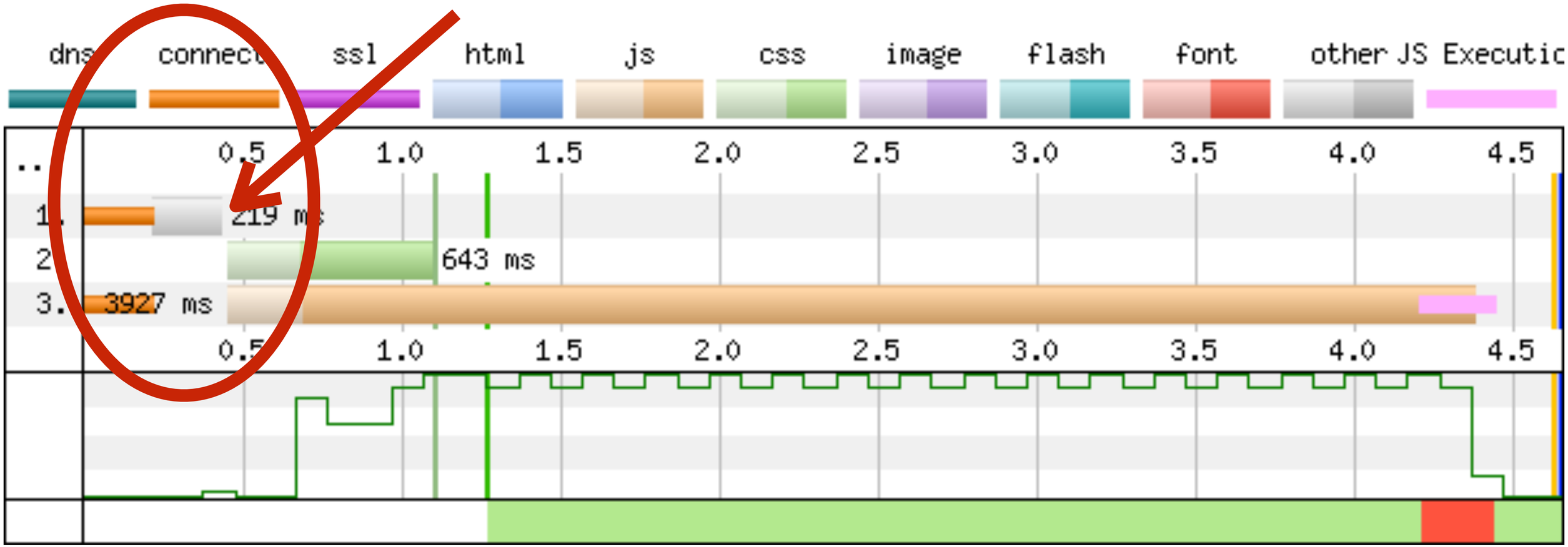
```
// entries/client.js

...

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

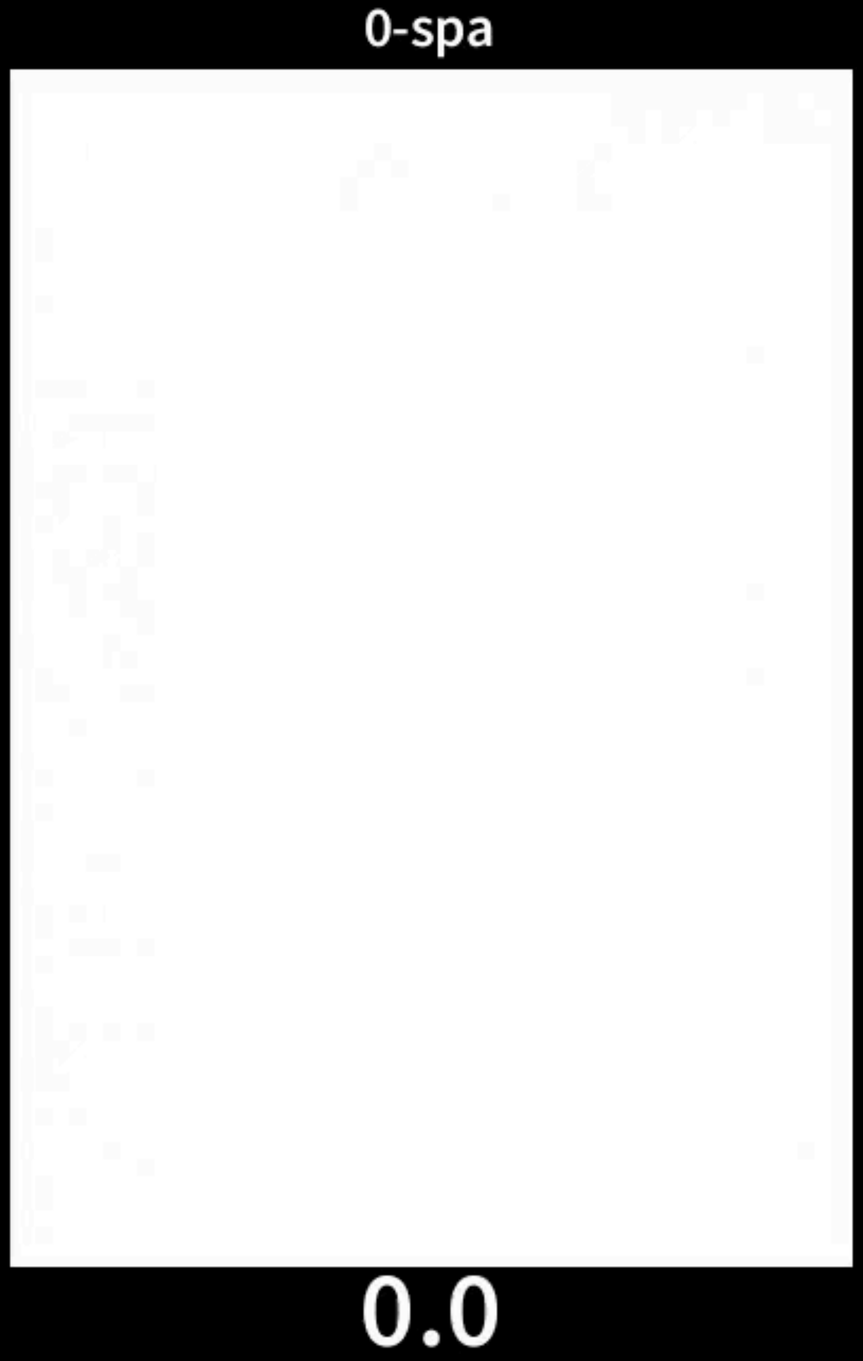

dns
connect
ssl
html
js
css
image
flash
font
other JS Executic
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
219 ms
643 ms
3927 ms

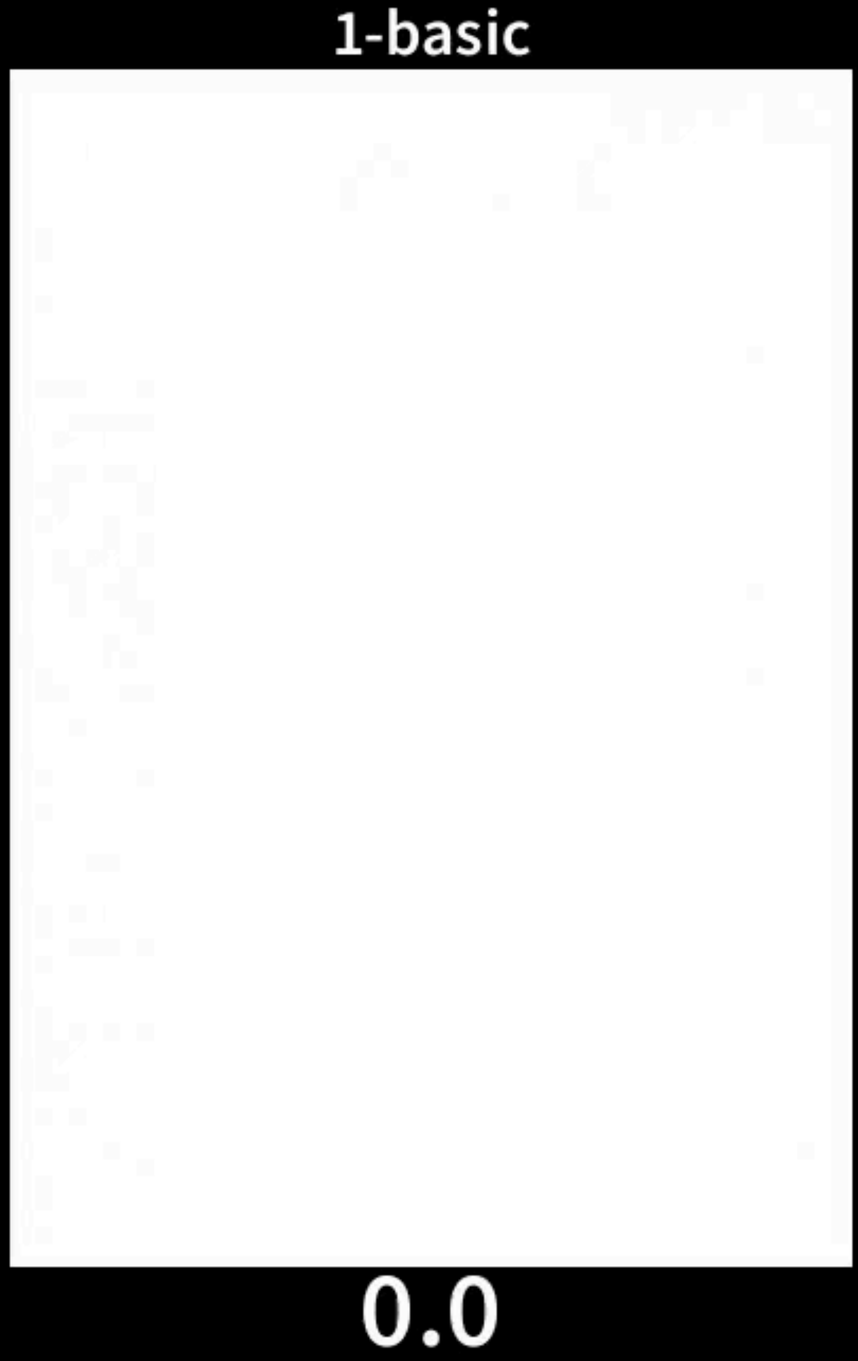
FMP: ~6300ms
TTI: ~6300ms

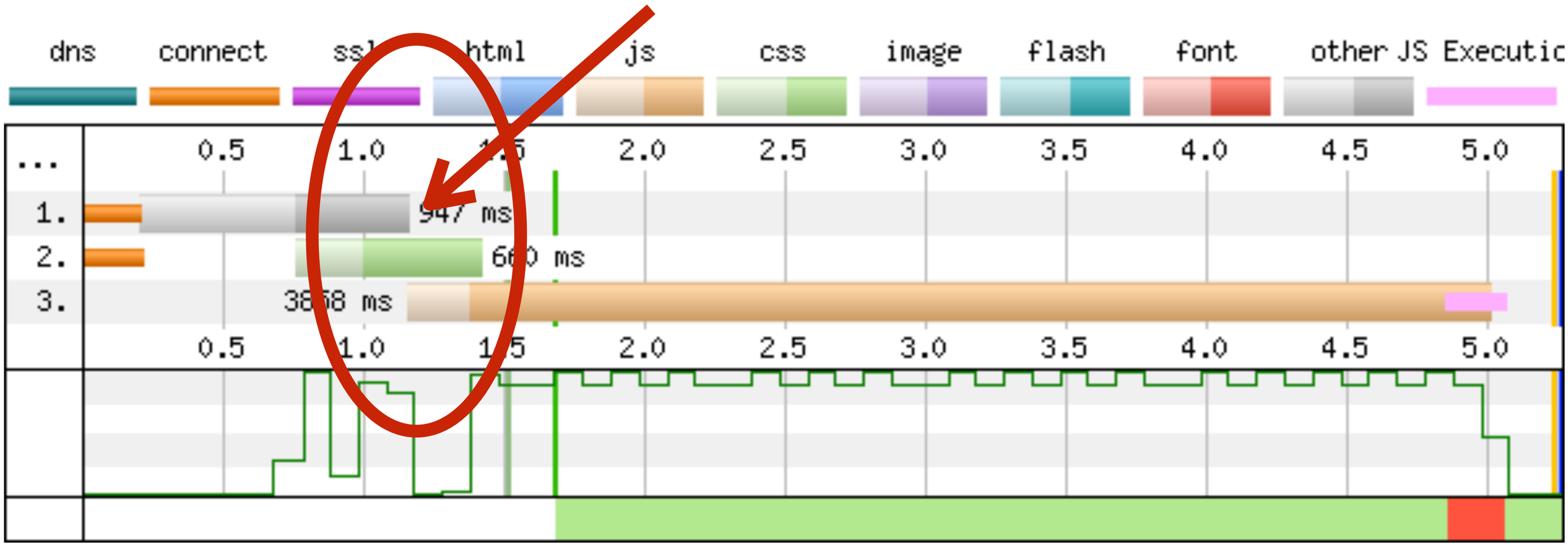
1. SSR

https://goo.gl/mwd8yr

```
// 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


dns
connect
ssl
html
js
css
image
flash
font
other
JS Executic
...
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
1.
947 ms
2.
3.

FMP: ~1700ms
TTI: ~6900ms

*

no direct DOM access
no direct use of browser / node APIs (e.g. document, fetch)
no singletons, global variables
no need in data reactivity (event listeners)

2. Checksum

https://goo.gl/b3140J

```
<div
    data-reactid=".157rq30hudc"
    data-react-checksum="556954499"
>
```

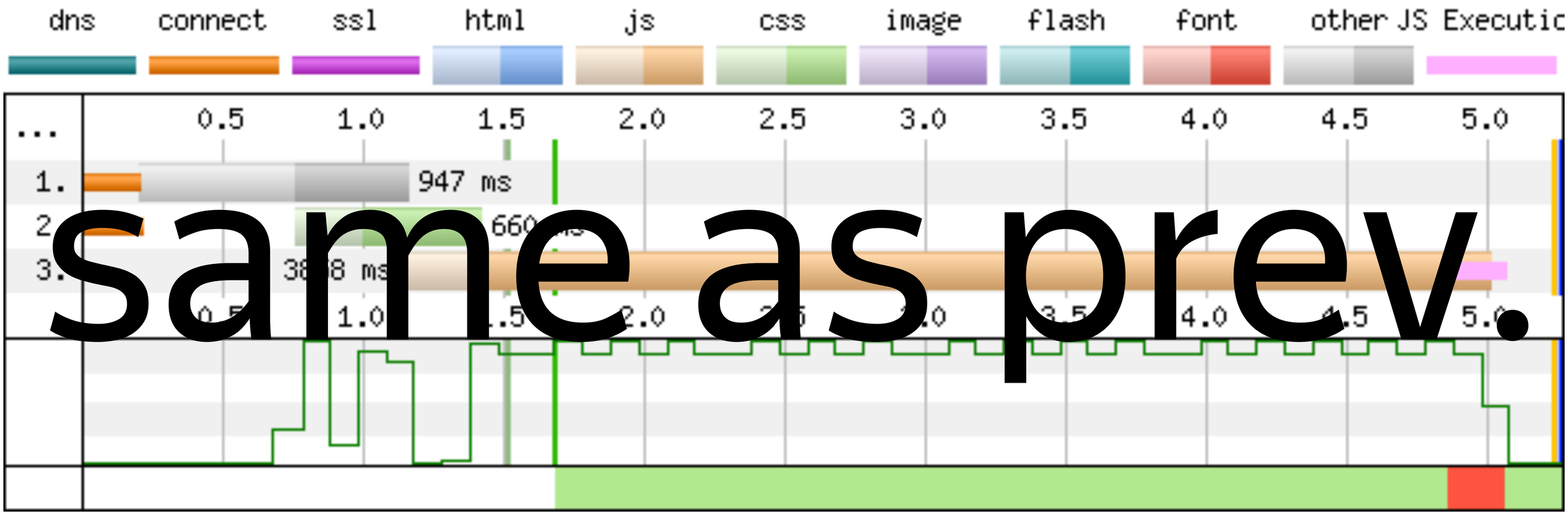
```
{{!--Bicycle.hbs--}}
```

```
<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 => {
            ...
        });
    }
}
```


dns
connect
ssl
html
js
css
image
flash
font
other JS
Executic
...
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
1.
947 ms
2
3.
same as prev.

FMP: ~1700ms
TTI: ~5100ms

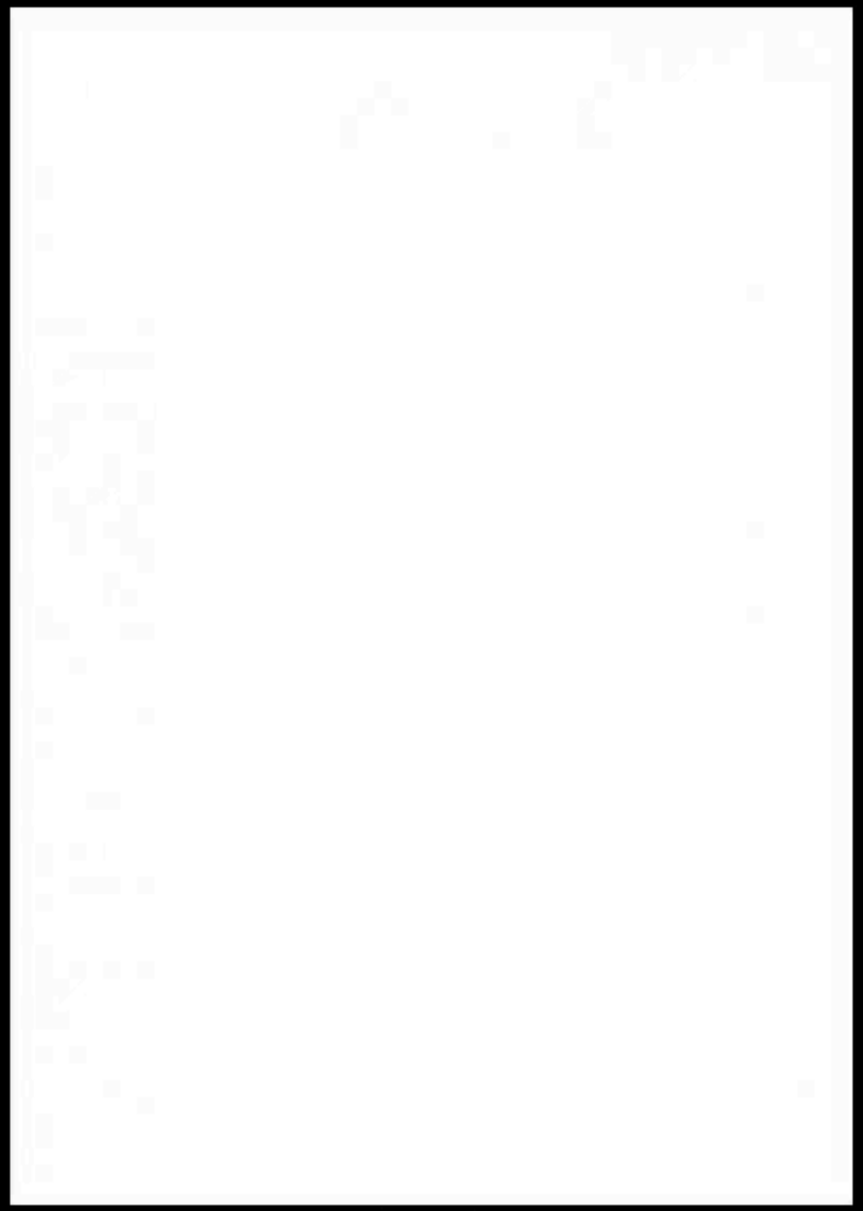
*

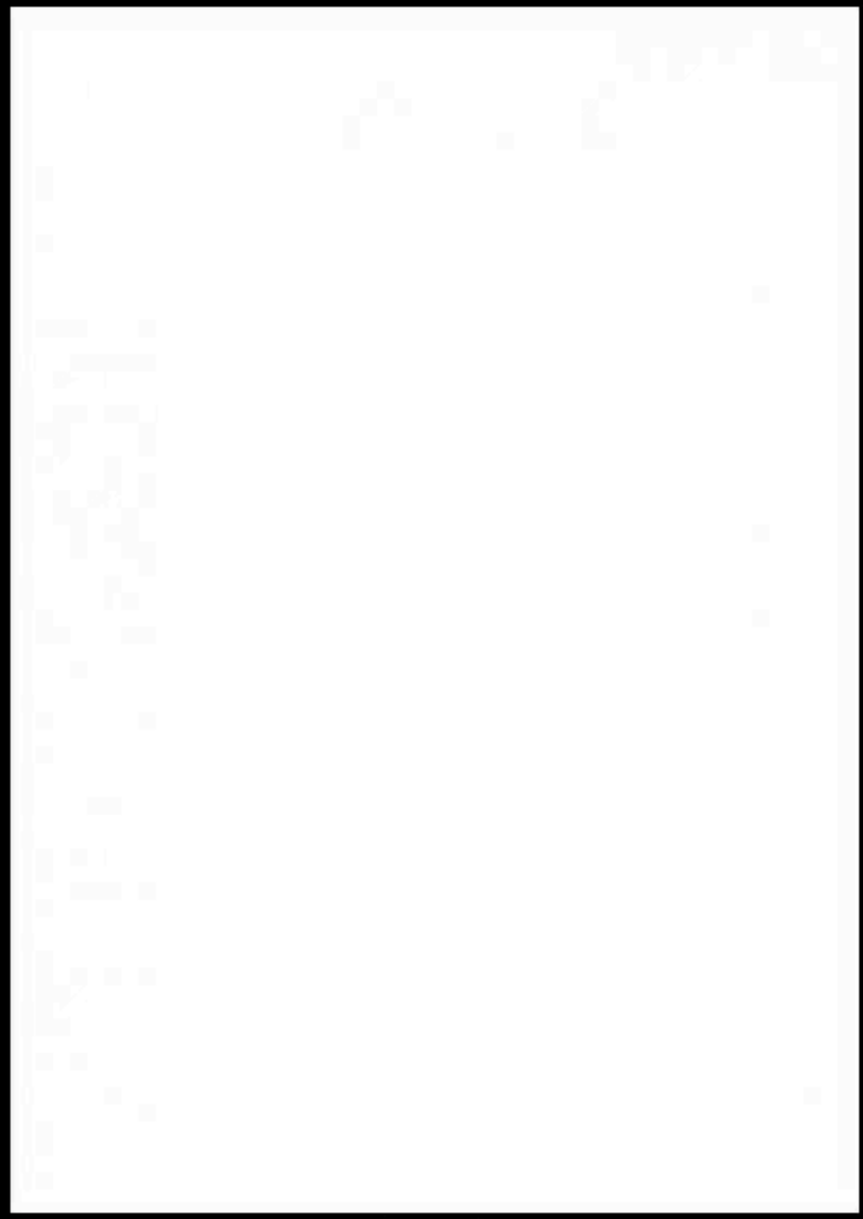
uid is a bad pattern, good is a hash(VDOM)

3. Cashing

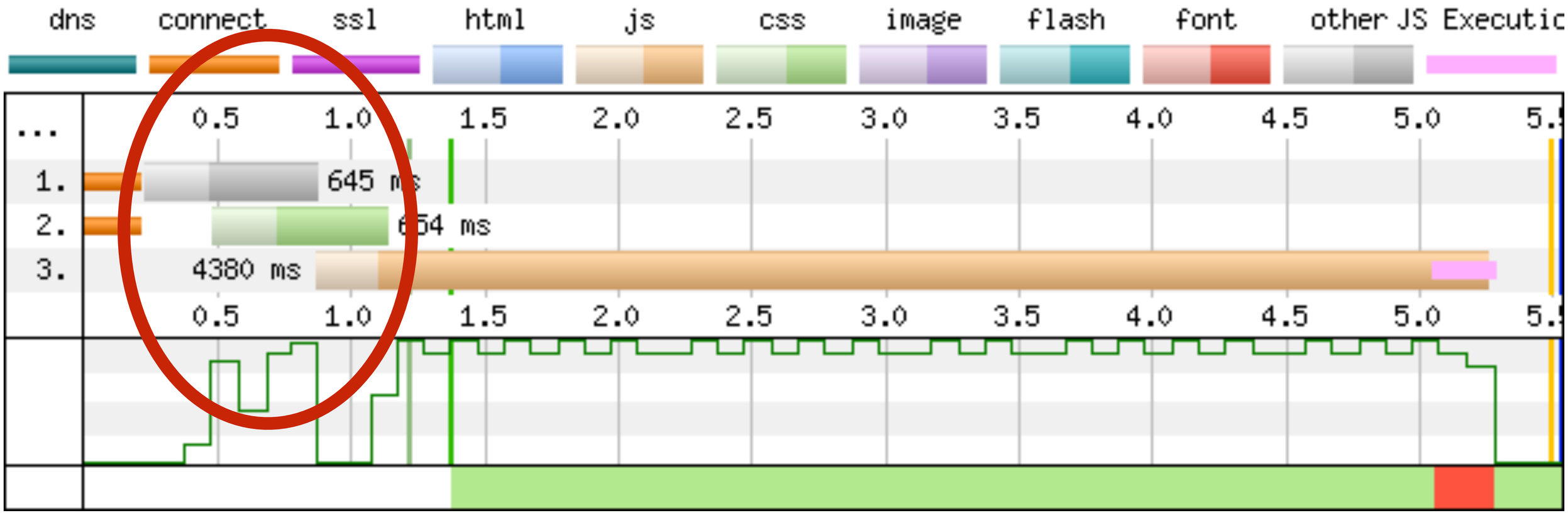
https://goo.gl/GnRRaZ

```
// 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

-300ms


dns
connect
ssl
html
js
css
image
flash
font
other
JS Executic
...
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
1.
645 ms
2.
654 ms
3.
4380 ms

FMP: ~1400ms
TTI: ~4800ms

*

* Better start from guest-only hash

4. Prefetch

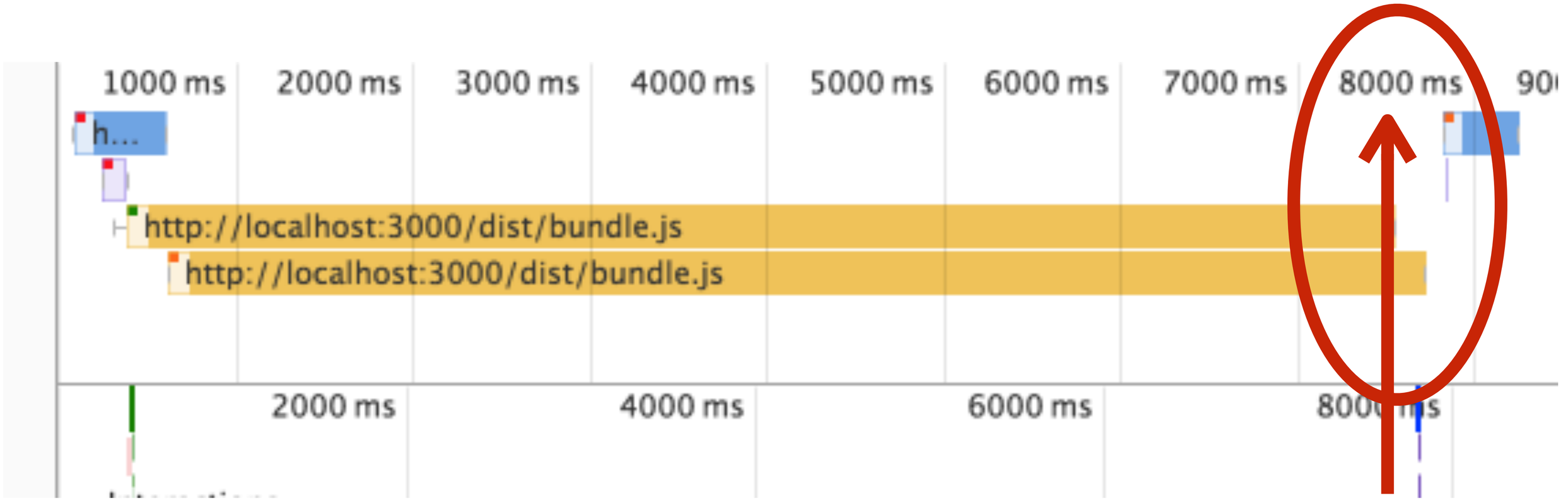
https://goo.gl/k63GLC

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

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

MDN: **Link prefetching** is a browser mechanism, which **utilizes browser idle time** to download or *prefetch* documents that the user might visit in the near future.

ha. ha. ha.


1000 ms
2000 ms
3000 ms
4000 ms
5000 ms
6000 ms
7000 ms
8000 ms
h...
http://localhost:3000/dist/bundle.js
http://localhost:3000/dist/bundle.js
2000 ms
4000 ms
6000 ms

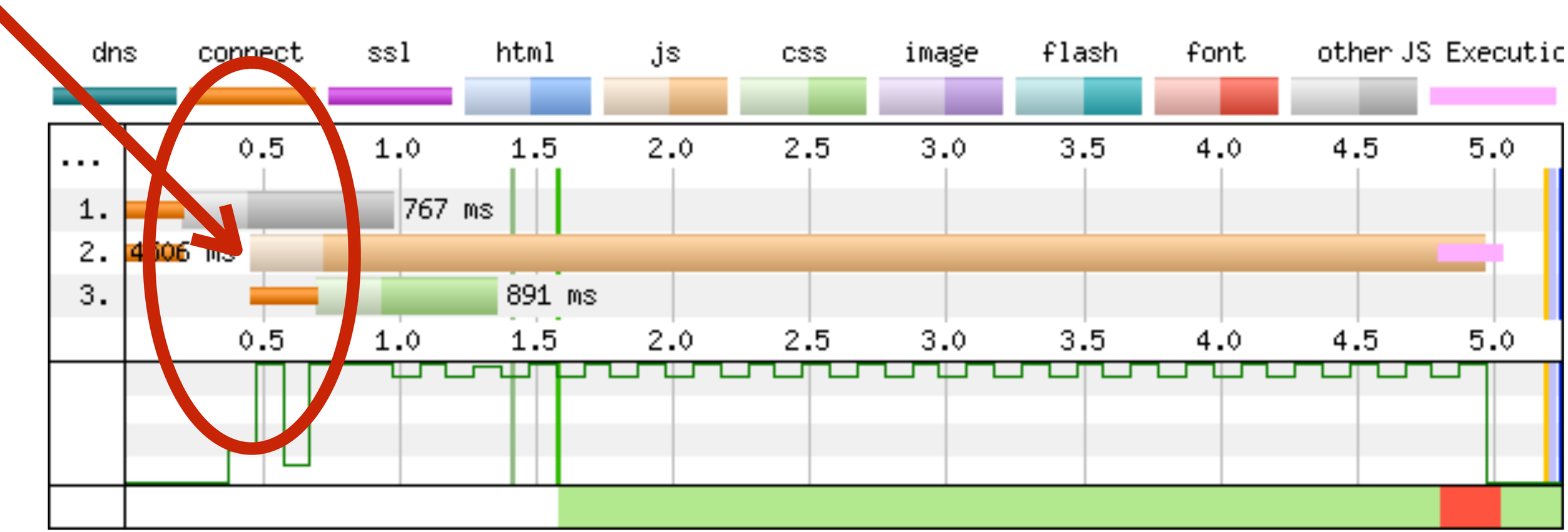
FMP: ~1400ms
TTI: ~7800ms

5. Preload

https://goo.gl/V5oa7O

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

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


dns
connect
ssl
html
js
css
image
flash
font
other JS Executic
...
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
1.
767 ms
2.
3.
891 ms

FMP: ~1600ms
TTI: ~5000ms

6. Streams

https://goo.gl/7MppPe

```
export class GeneratorStream extends Readable {

    constructor(generator) {
        super();
        this.iterator = generator();
    }

    async readAndEmit() {
        ...
    }


}
```

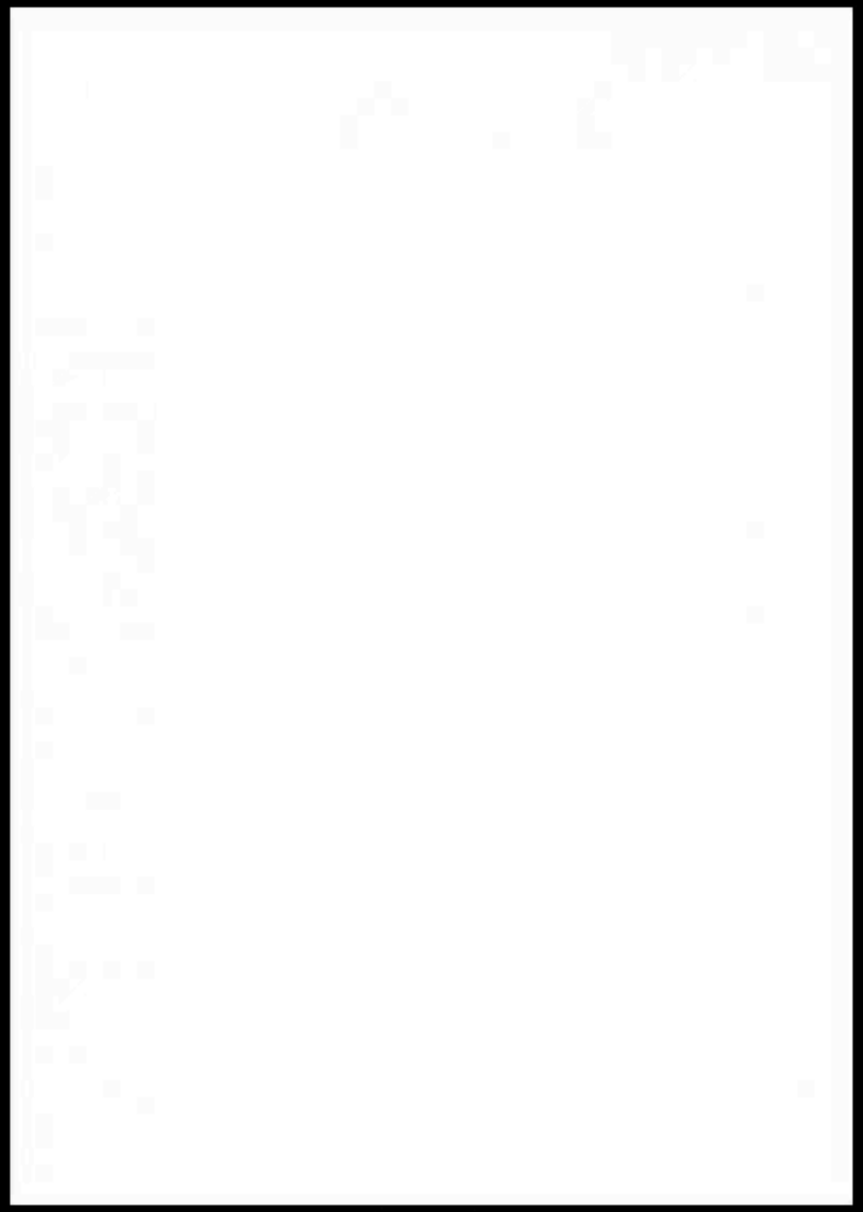
```
let stop = false;
do {
    let {value, done} = this.iterator.next();

    // if current value is promise - await it
    if (value && value.then) {
        value = await value;
    }

    // we should stop if either iterator is done or
    // .push() returns false
    // (which means readable stream internal buffers
    // are full)
    stop = !this.push(done ? null : value) || done;
} while (!stop)
```

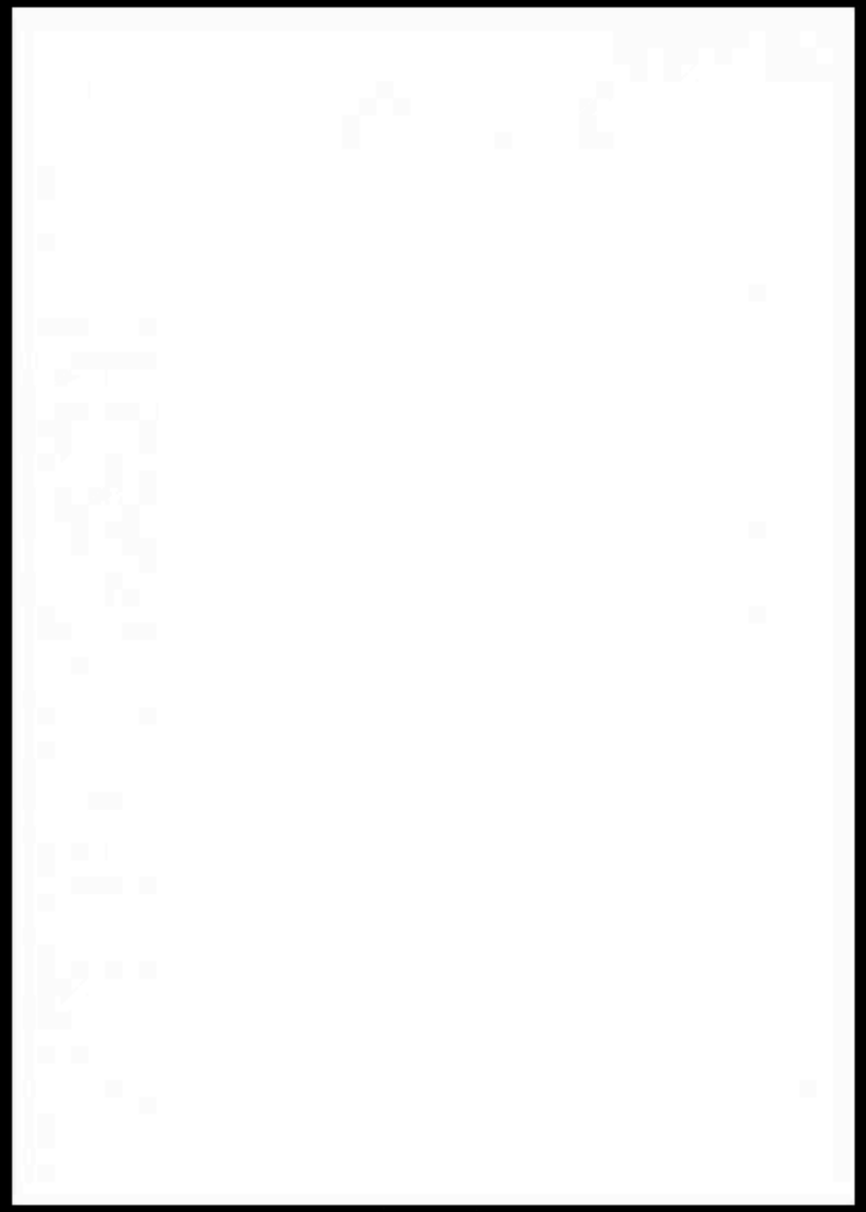
```
// DocumentGenerator.hbs



export default function *() {
    yield headerTemplate();

    const app = new Bicycle({fetch});
    // delegate rendering to root component
    yield * app.render();

    yield footerTemplate();
}
```

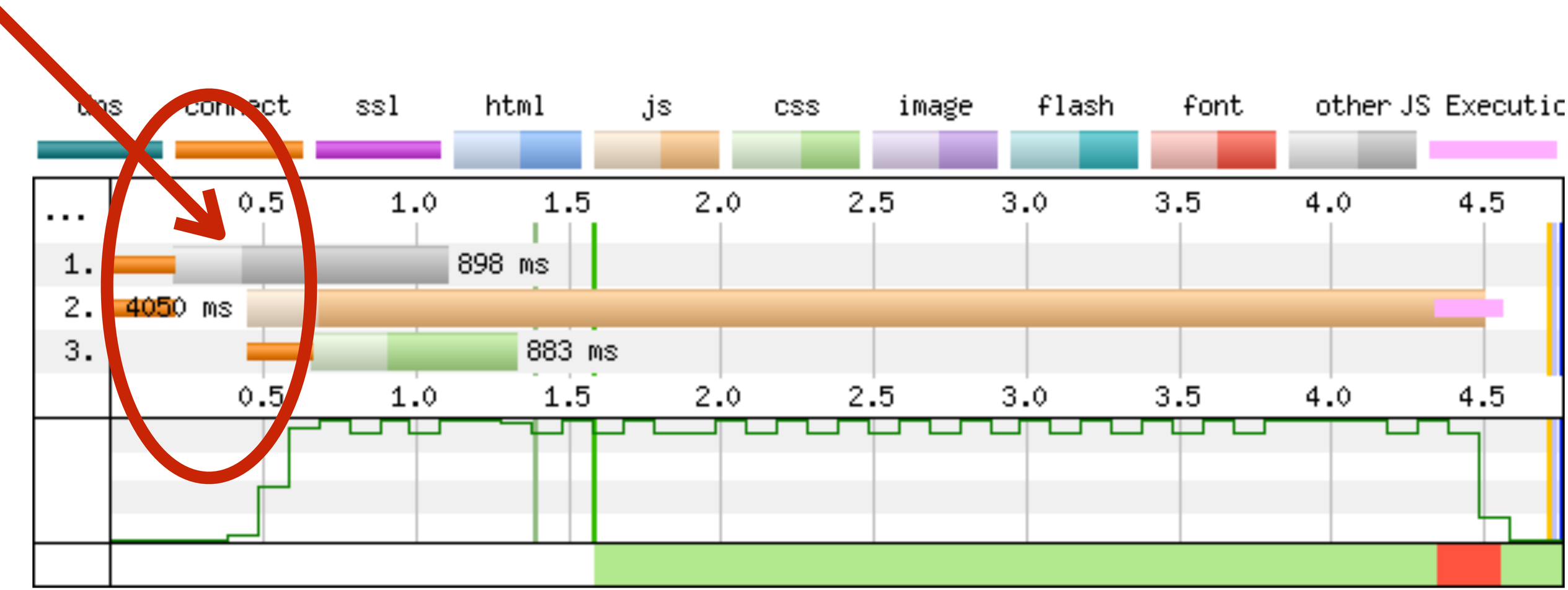
```
{{!--Header.hbs--}}


<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">
```

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



</div>
<div class="copyright">
    Data taken from <a target="_blank" href="http://
www.coindesk.com/price/">coindesk.com</a>
</div>
<script src="dist/bundle.js"></script>
</body>
</html>
```

```
// 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();
}
```


1-basic
0.0


6-stream
0.0


dns
connect
ssl
html
js
css
image
flash
font
other JS Executic
...
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
1.
898 ms
2.
4050 ms
3.
883 ms

FMP: ~1600ms
TTI: ~4600ms

*

〉 HTTP status is always 200 :)

〉 smaller gzip_buffers

〉 search engines (will not wait)

TLDR;

SPA >> SSR

FMP: 6300 >>1600ms

TTI: 6300 >> 4600ms

Demo: akwuh.me/ssr-demo
Source: github.com/jakwuh/ssr-demo

Спасибо за внимание

James Akwuh
Software Engineer

 jakwuh@yandex-team.ru

 jakwuh

 jamesakwuh

 jakwuh