

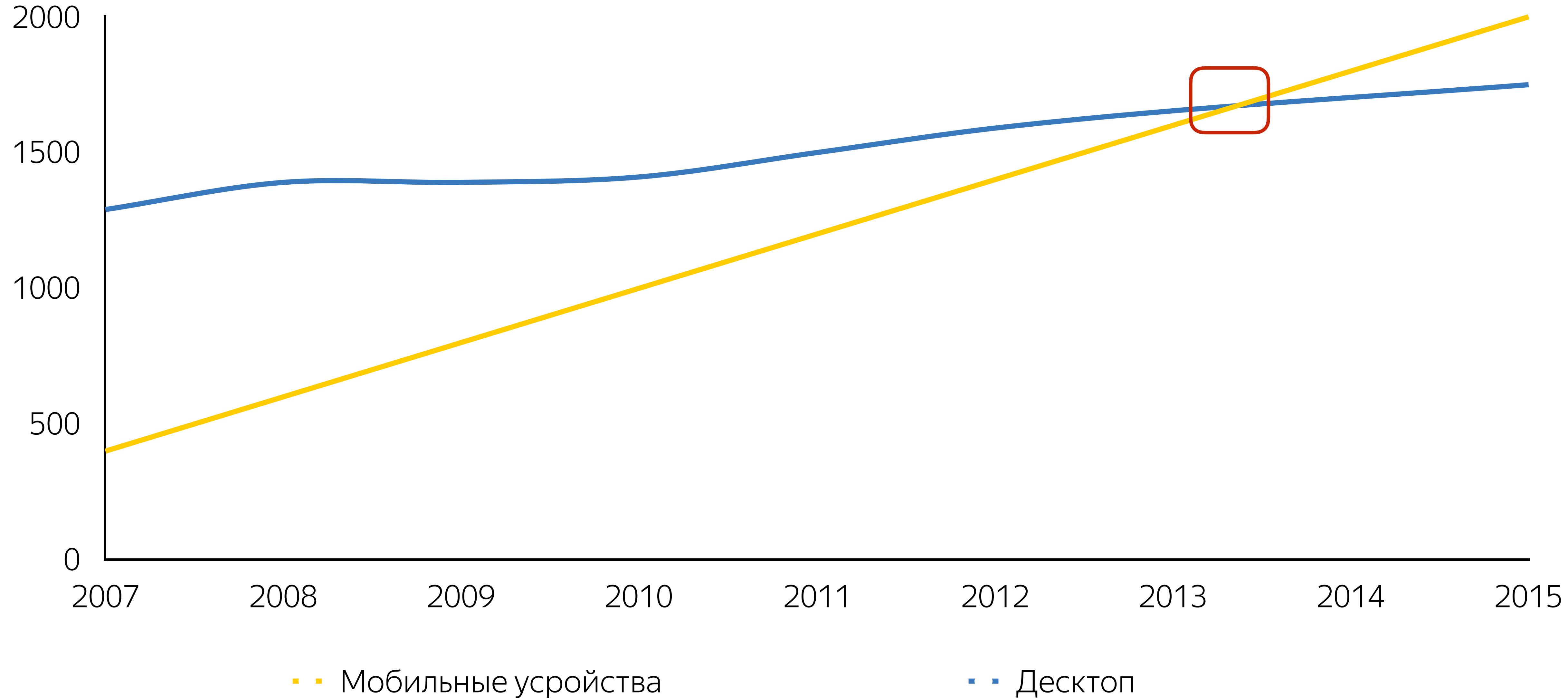
Яндекс Деньги

Ускоряем загрузку
сайтов в эпоху
смартфонов

Андрей Мелихов



Активная база пользователей Интернета в миллионах человек

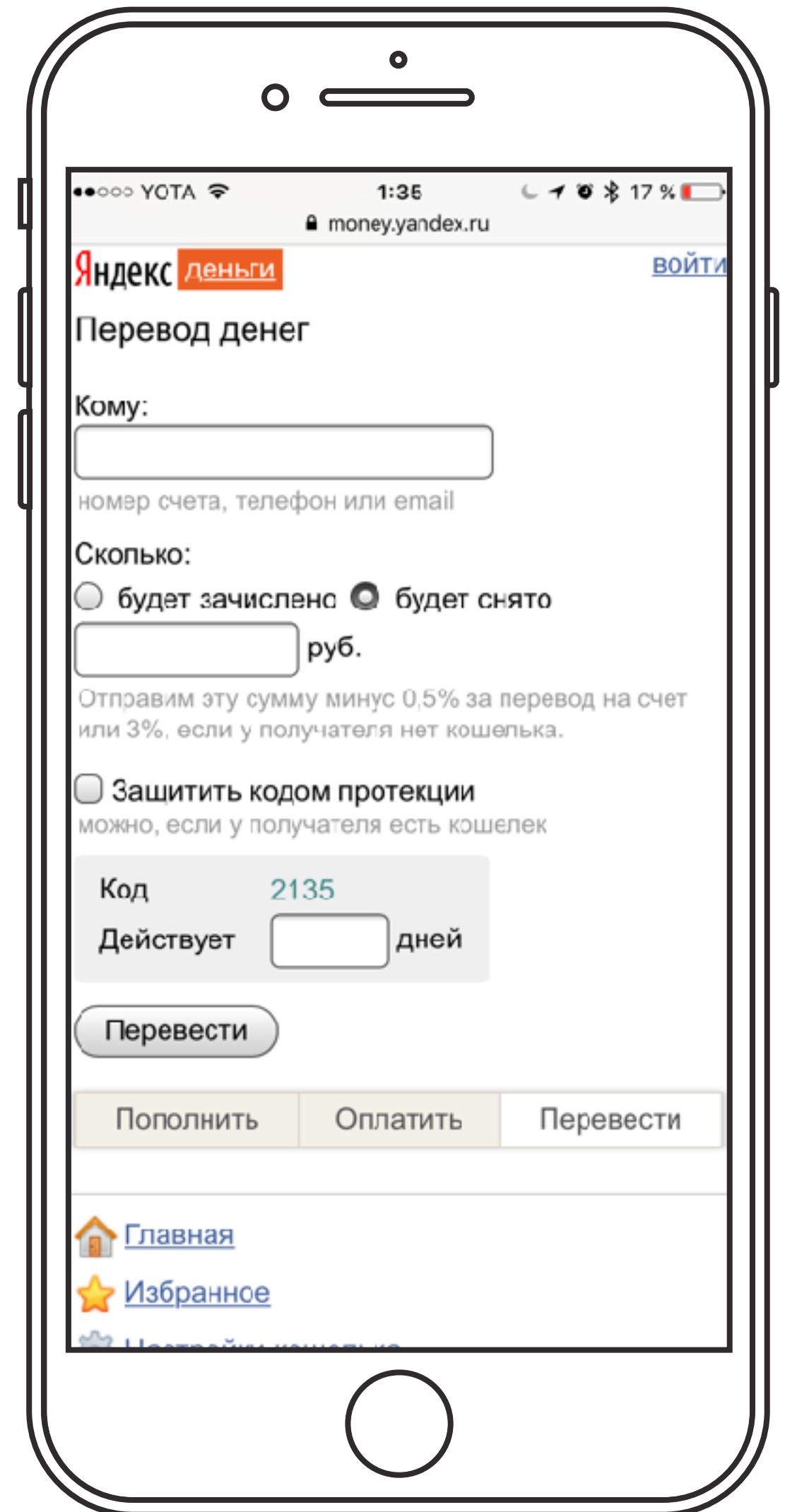


Готов ли ваш сайт для
смартфона?





m.mysupersite.com



« »🔍★

Яндекс Деньги

НайтиОткрыть кошелек➡ Войти

Товары и услуги

Переводы

Банковские карты

Прием платежей

Игры и сервисы

Скидки и бонусы

С кошельком комиссия ниже

Сейчас комиссия за перевод — 1,95%, с кошельком будет 1,5%.

Создать кошелек — минутное дело.

[Создать кошелек](#) [У меня есть кошелек](#)

Перевод денег

Откуда

Сбербанк
MasterCard
Visa

Номер карты

VALID THRU
 / Три цифры с оборотной стороны карты

Куда

Другие способы

На счет в банке

Через Western Union

Через Unistream

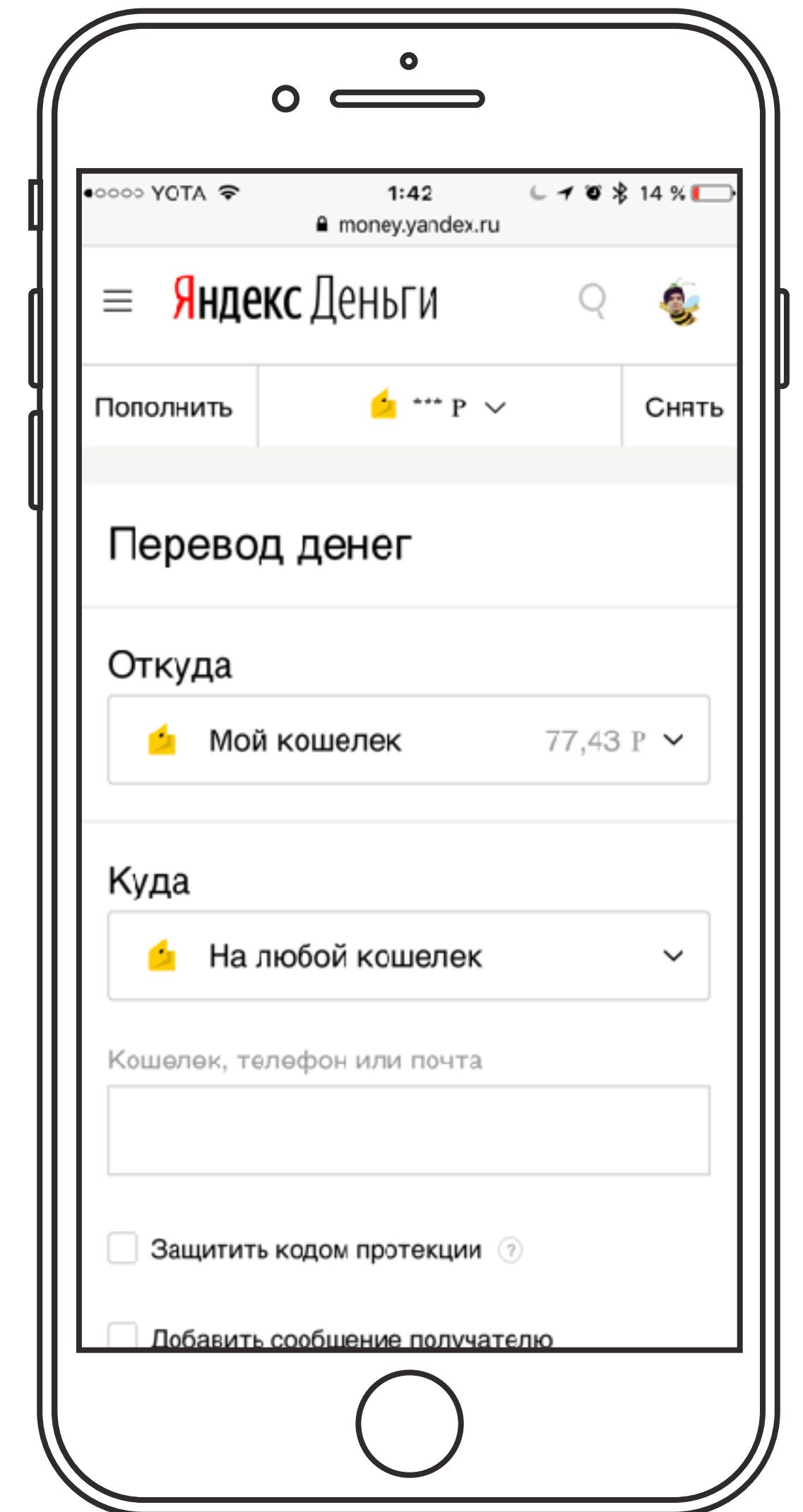
Помощь

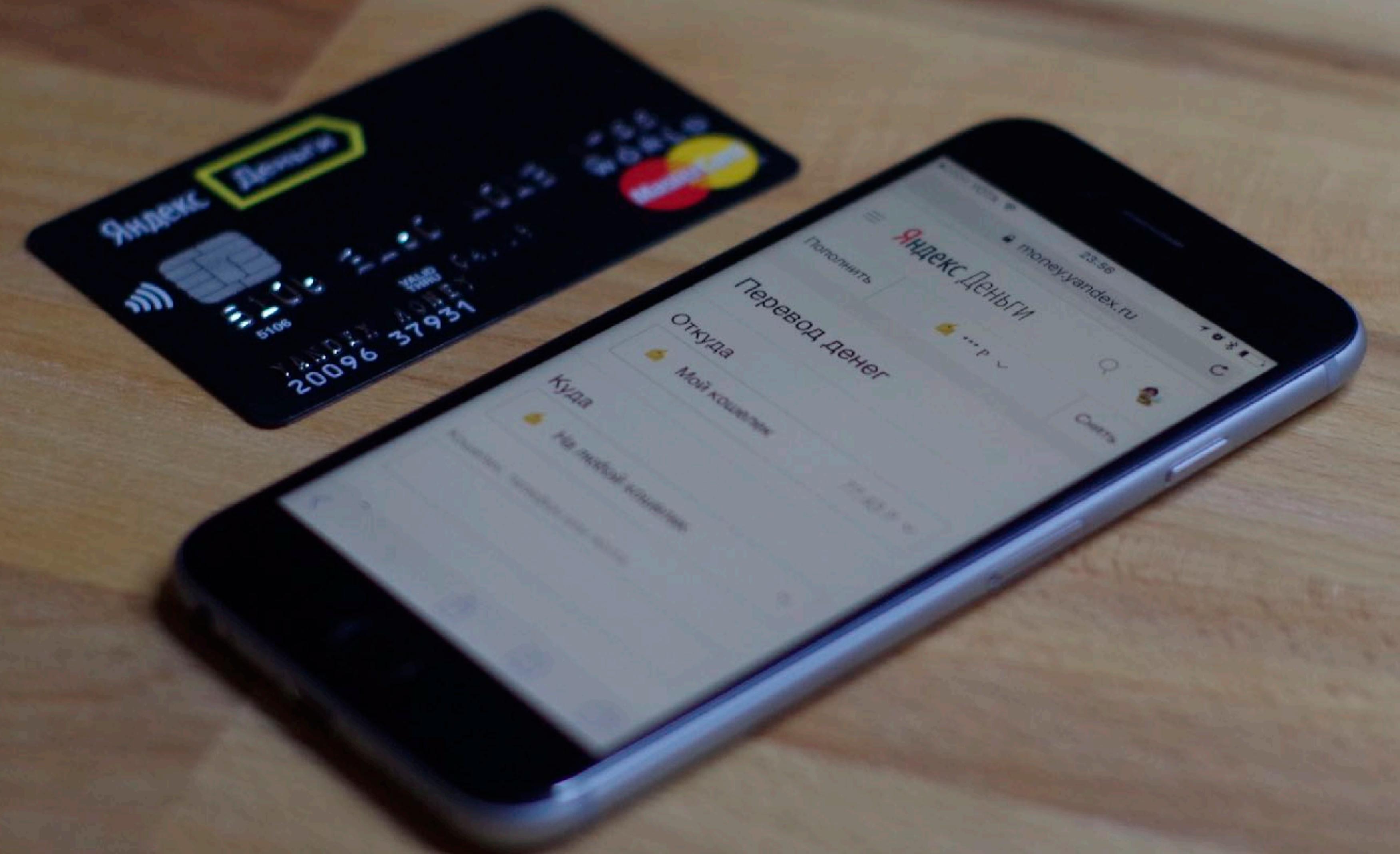
О переводах

Лимиты

Важно знать

Отправить перевод с карты на карту можно, только если обе выпущены в России.







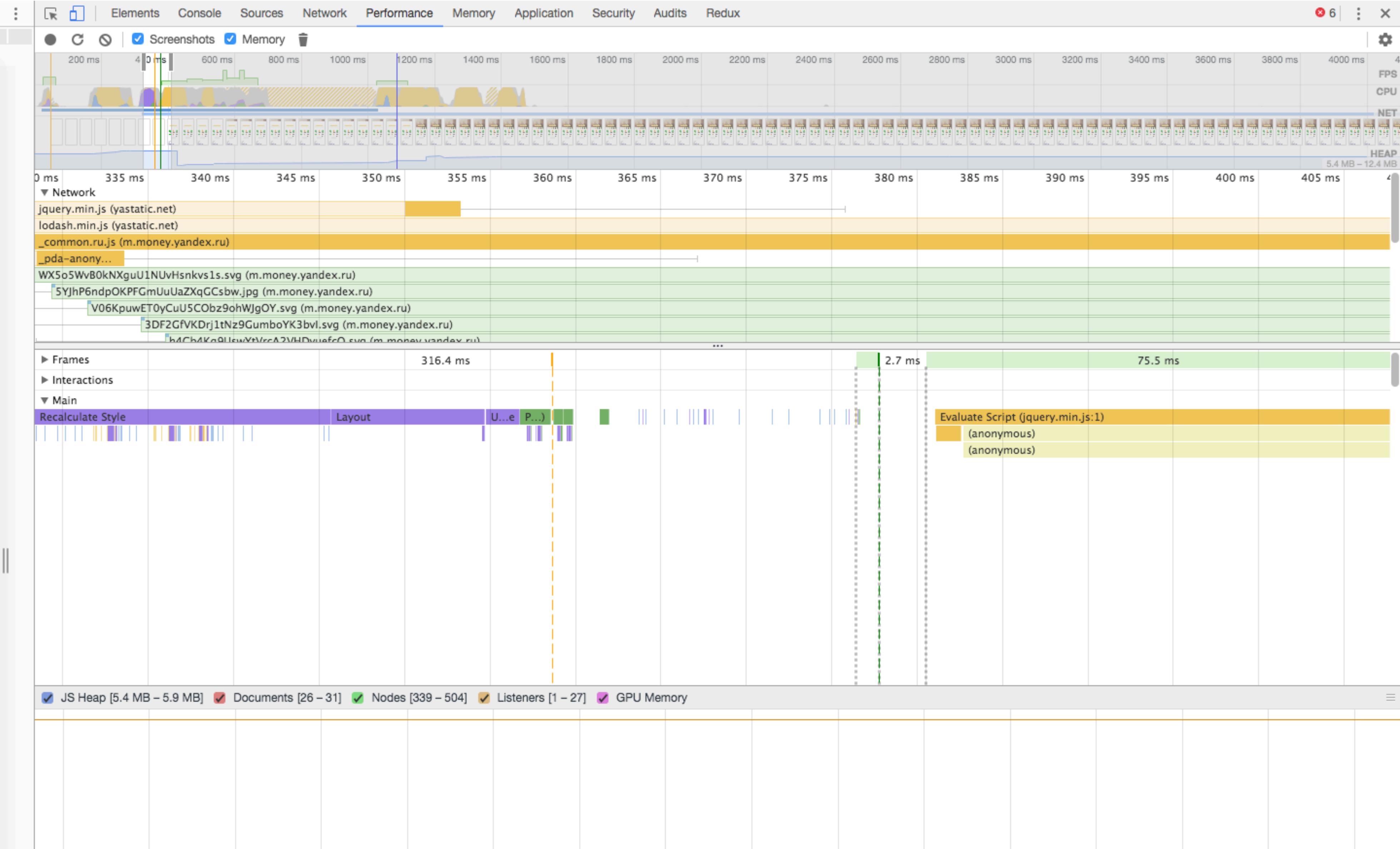
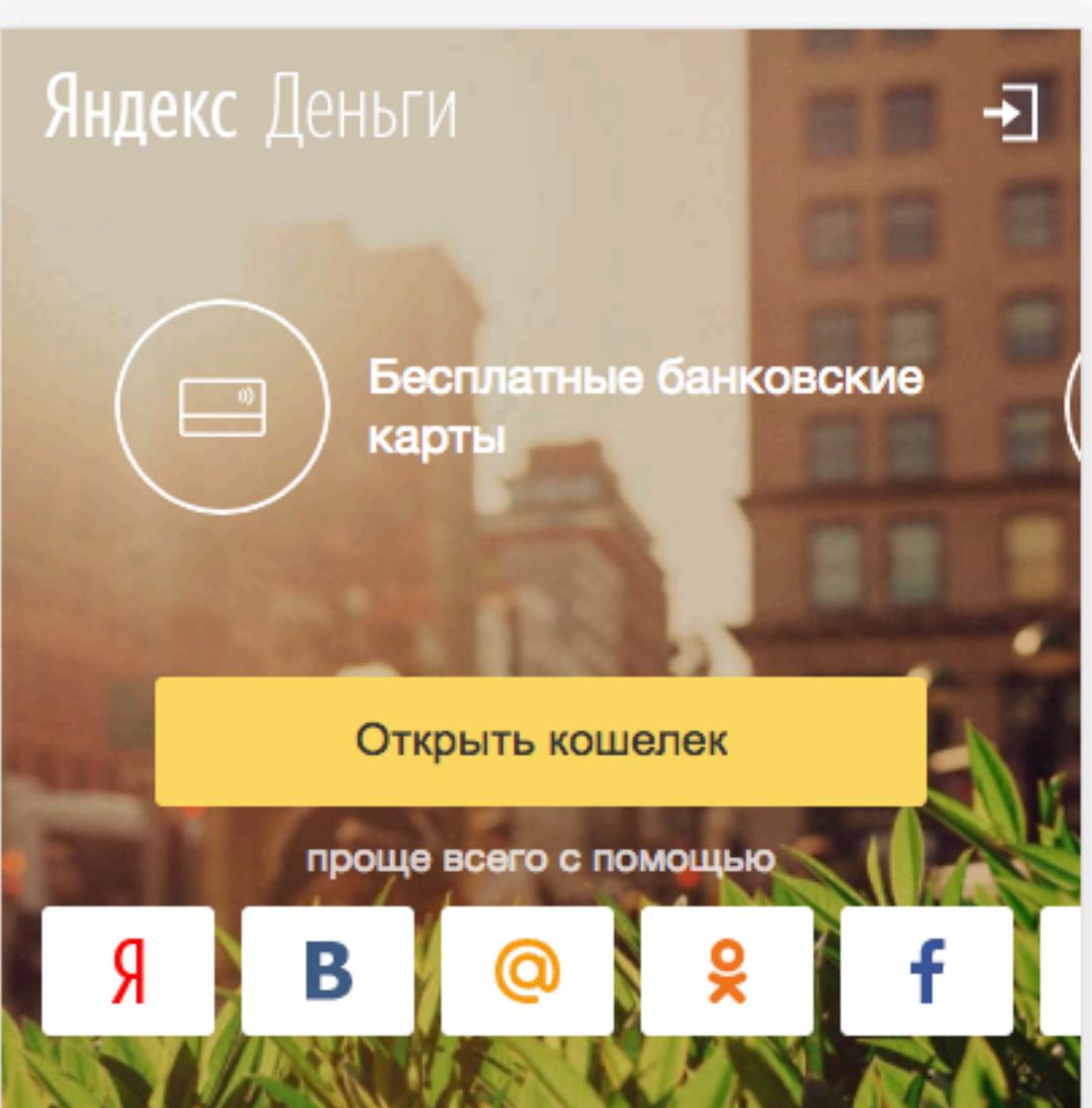
- › В 53% случаев пользователи отказываются от дальнейшей загрузки мобильного сайта, если она требует более 3-х секунд ожидания
- › Каждый второй пользователь ожидает, что страница загрузится менее чем за 2 секунды

«Сайт загрузился» — что
это значит?



Стадии загрузки сайта

1. Первое отображение
2. Первое **значимое** отображение
3. Первая интерактивность



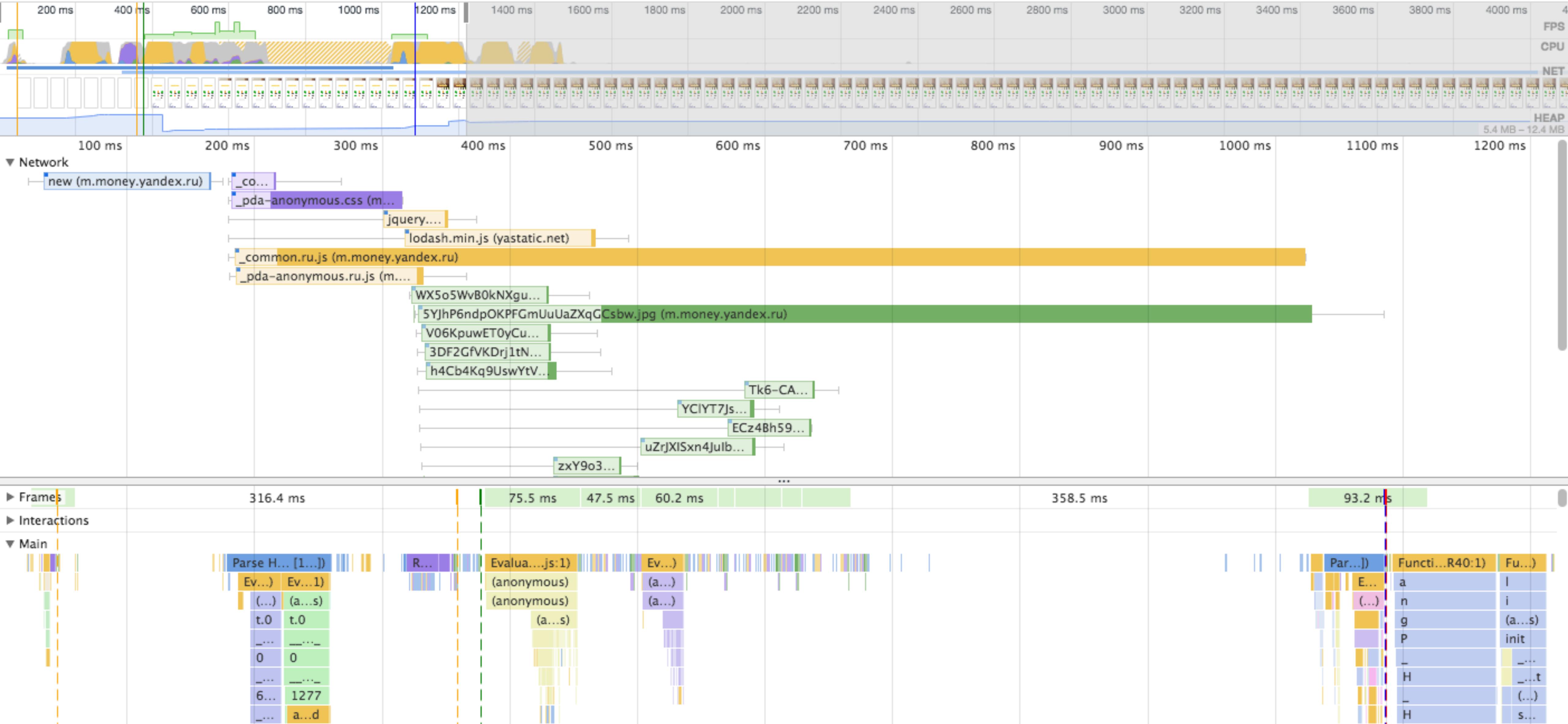
[Полная версия](#)

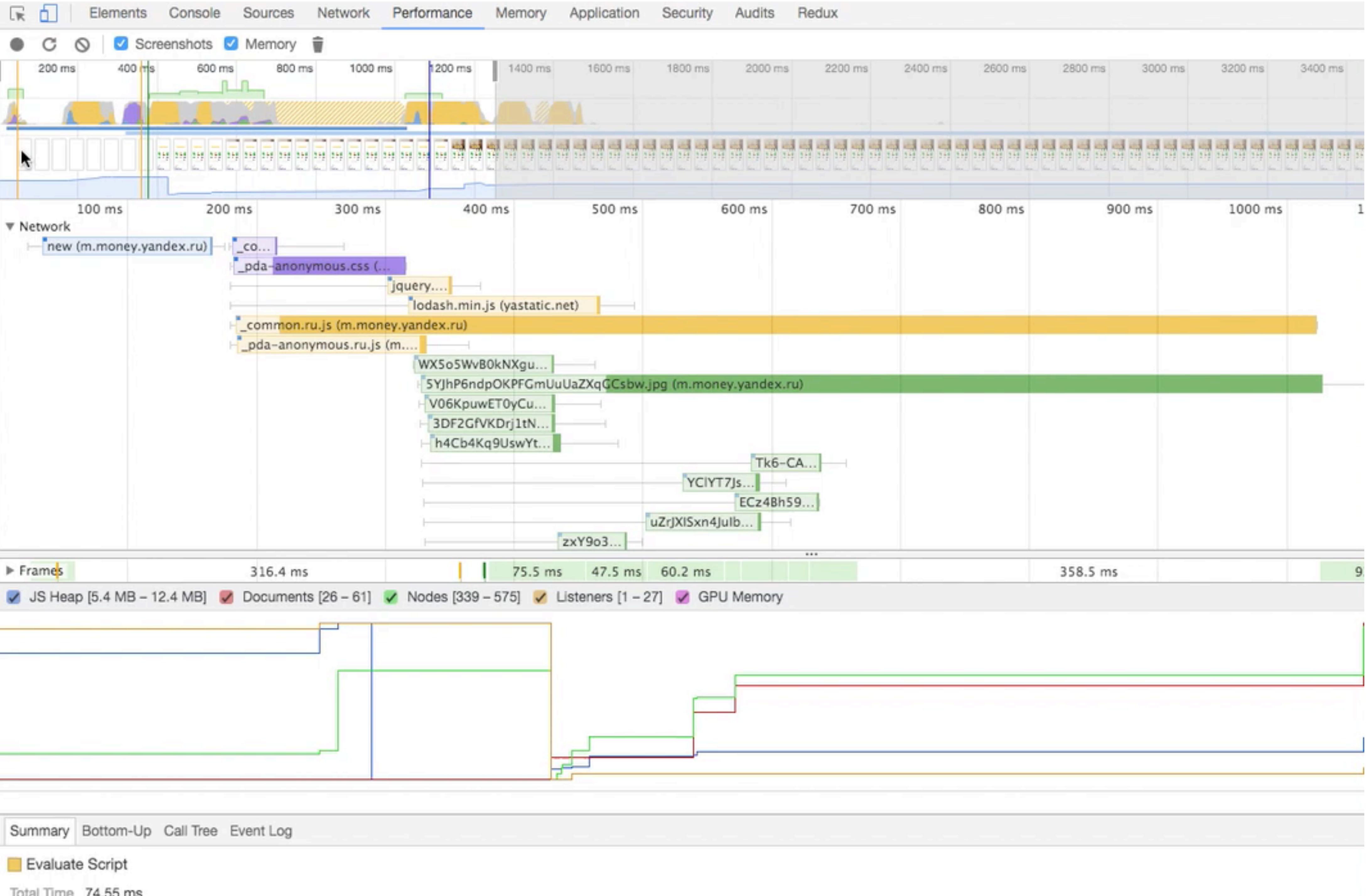
[Помощь](#)

© 2002–2017 «Яндекс»

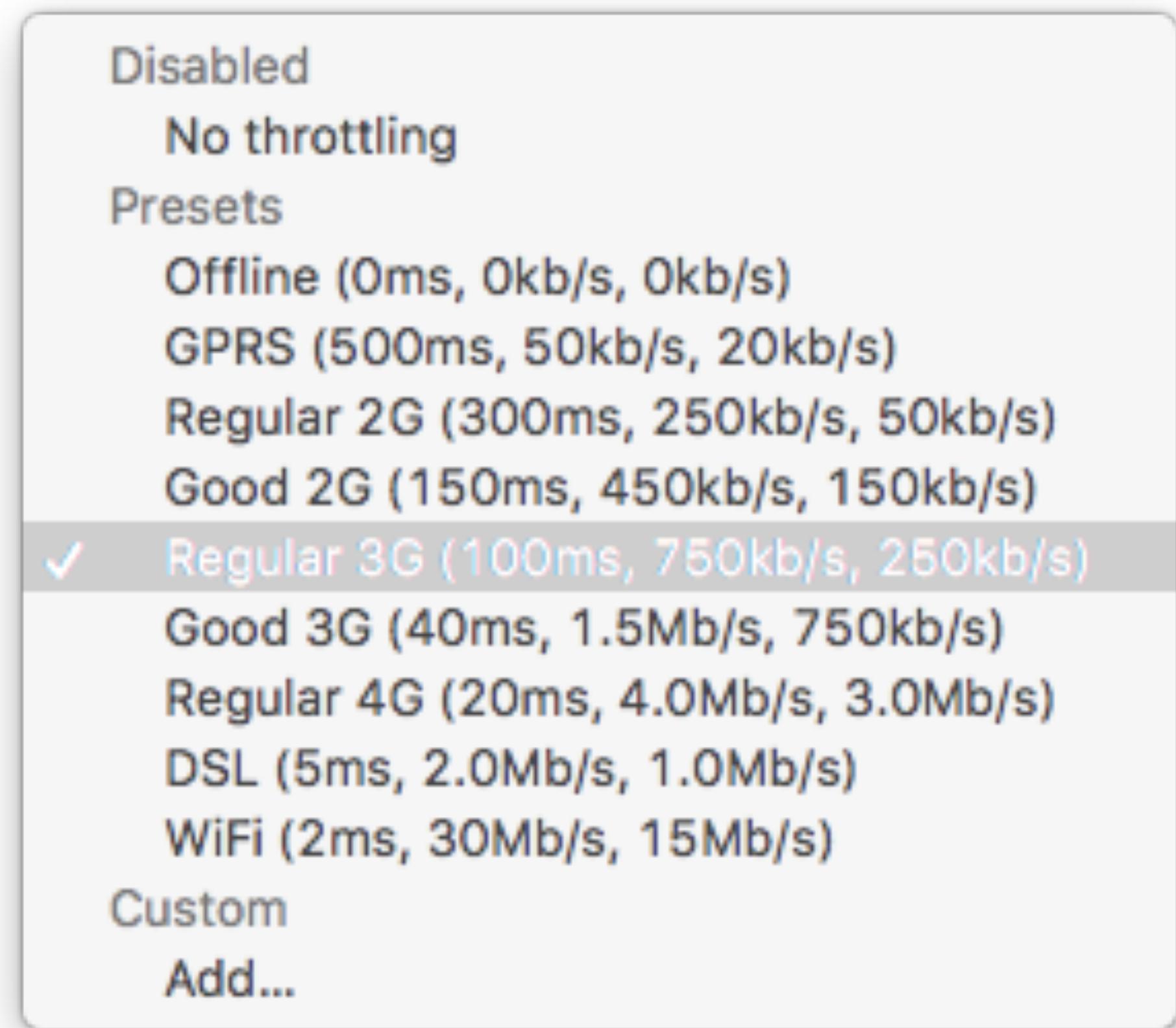
© 2007–2017 ООО НКО «Яндекс.Деньги»

...

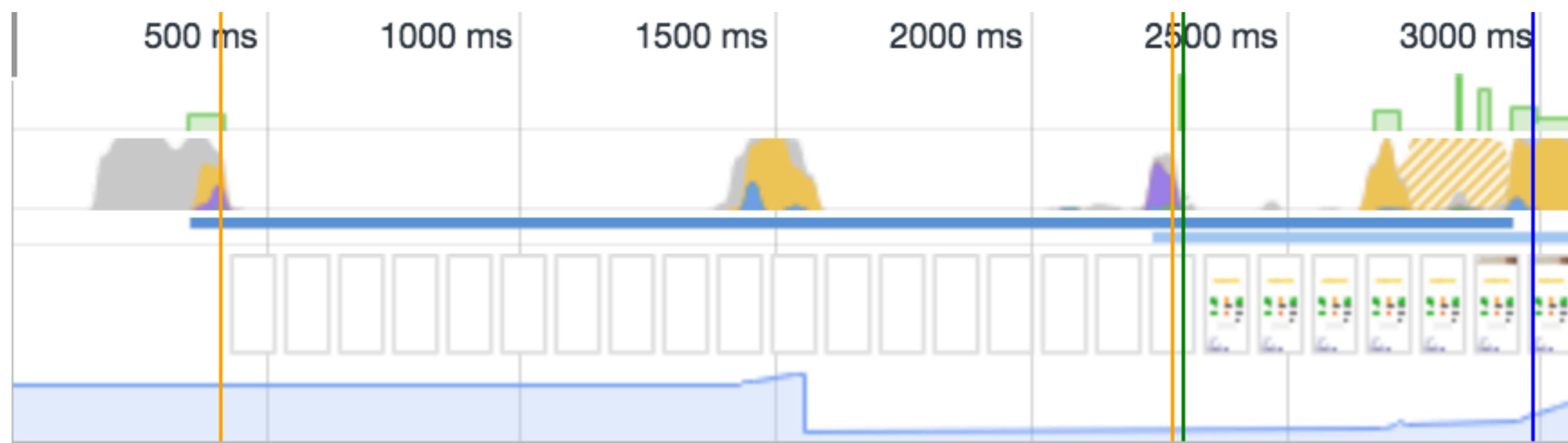




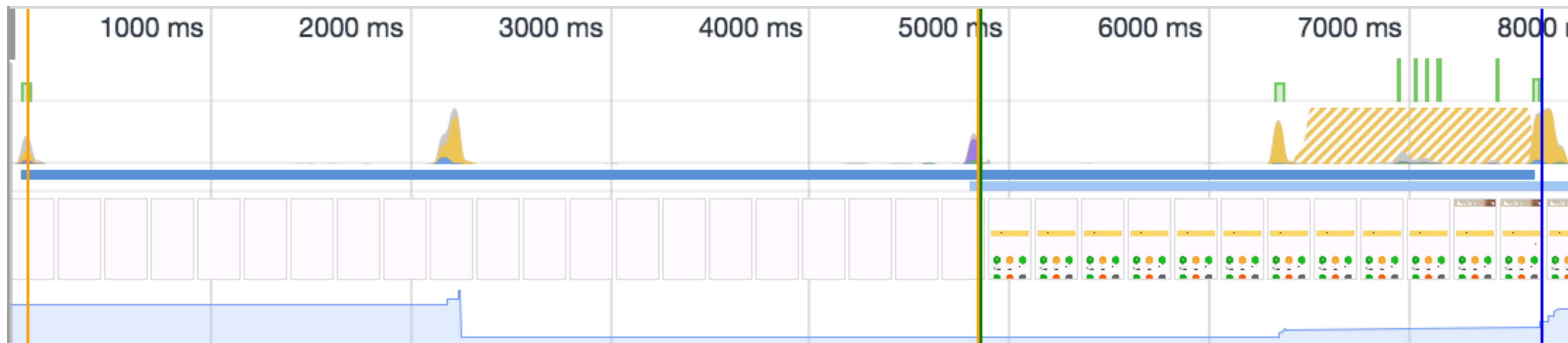
Троттлинг



Fast 3G



Slow 3G

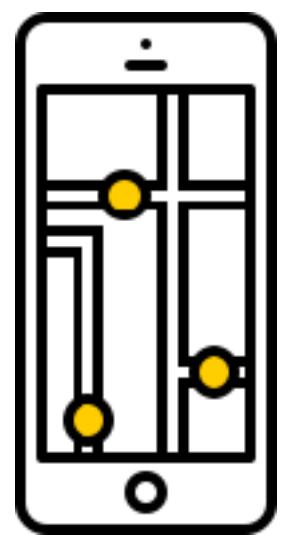


Стадии запроса веб-страницы

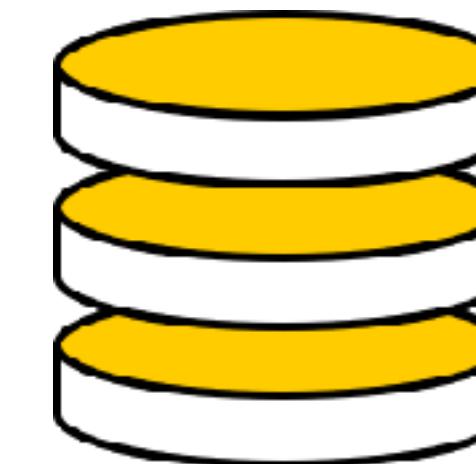


1. Начальный запрос

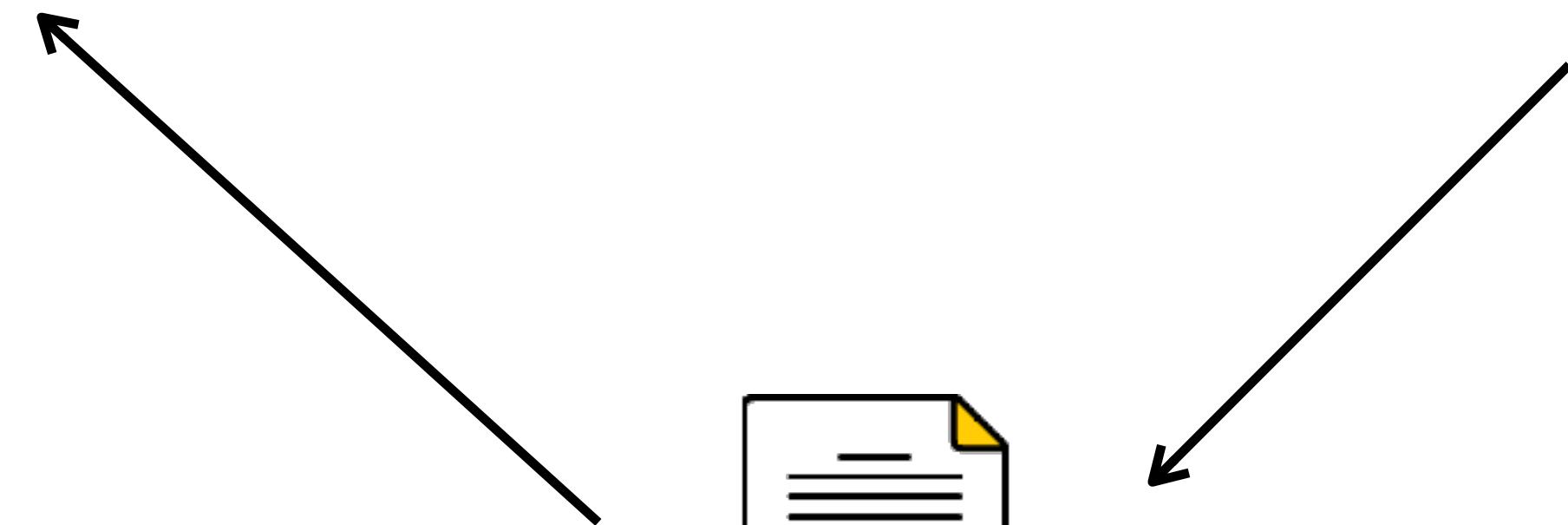
Смартфон



Сервер



HTML



2. Парсинг HTML и загрузка ресурсов





Большинство
браузеров
анализируют HTML,
используя потоковые
парсеры

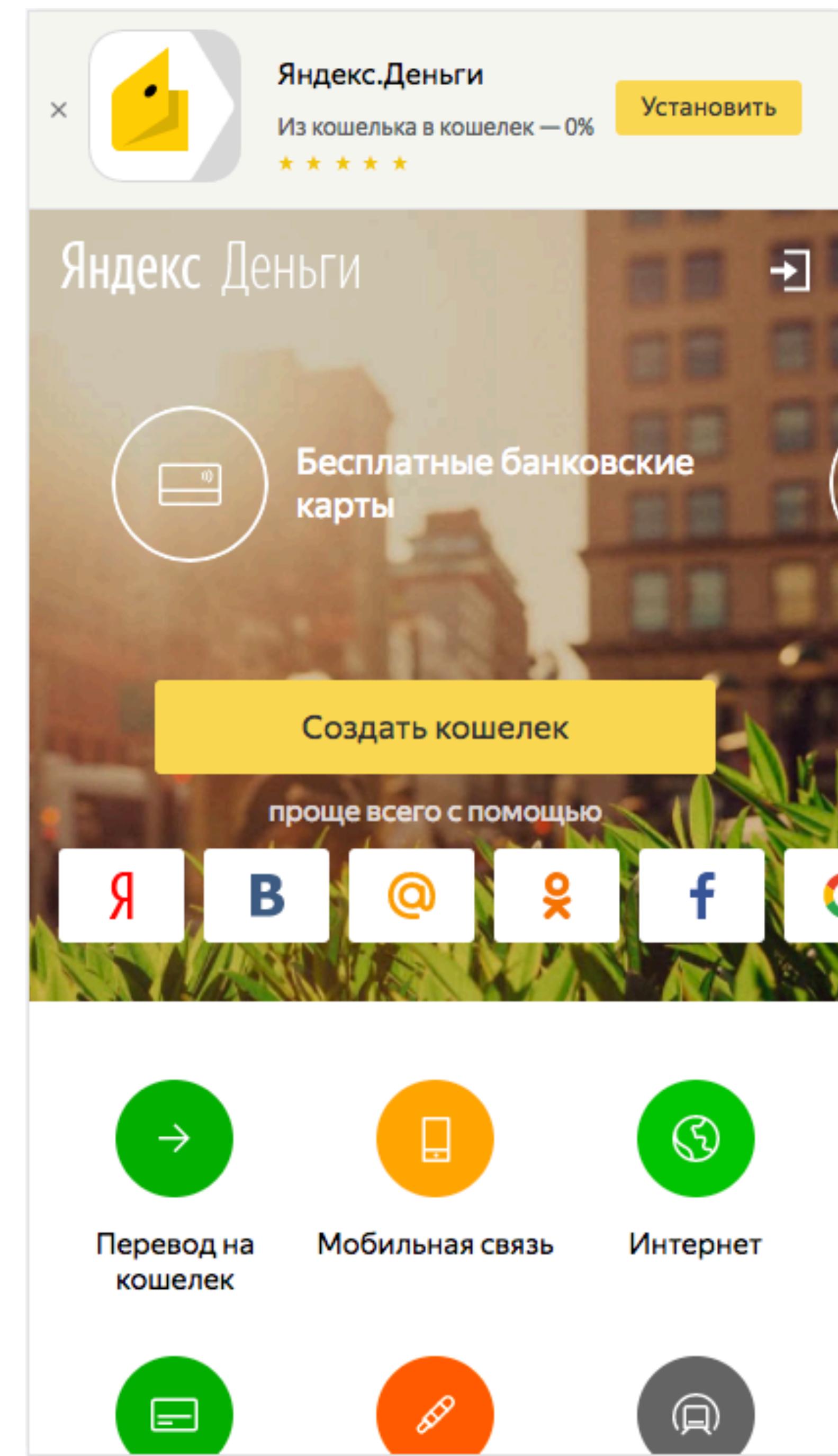
3. Парсинг ресурсов и их исполнение

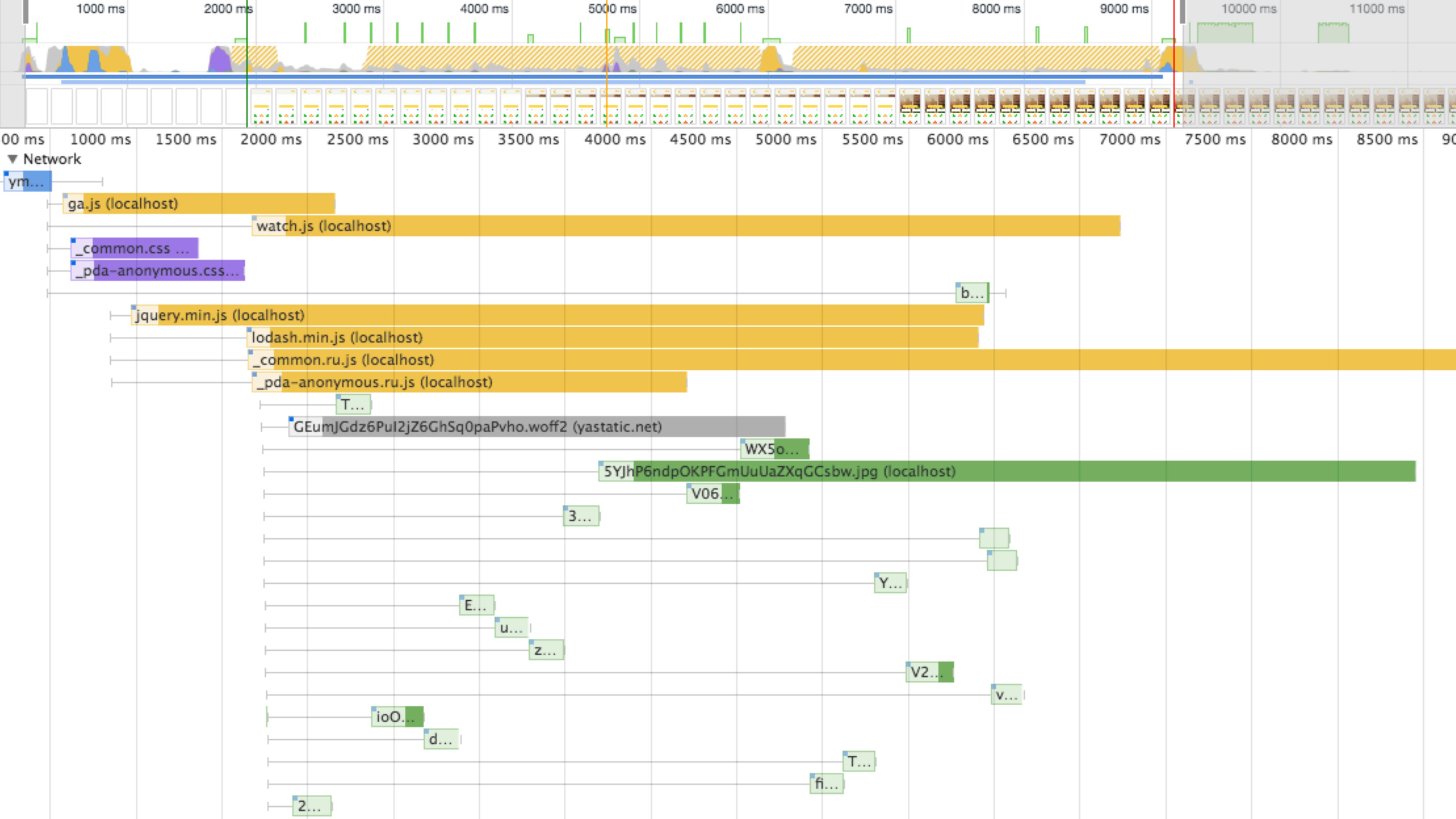
| JavaScript - парсинг и компиляция

| CSS - наложение стилей, рендер

Оптимизация доставки ресурсов



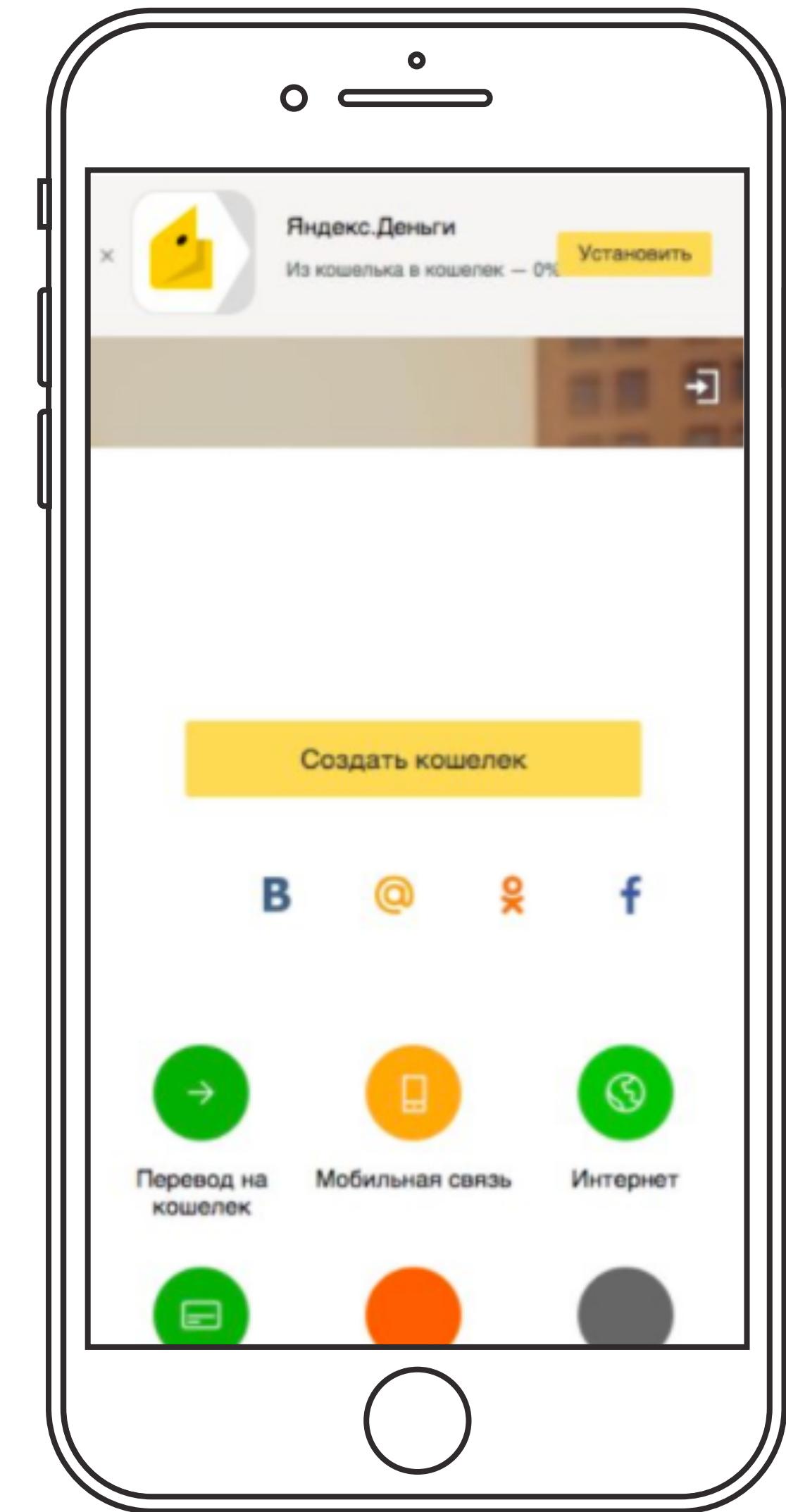




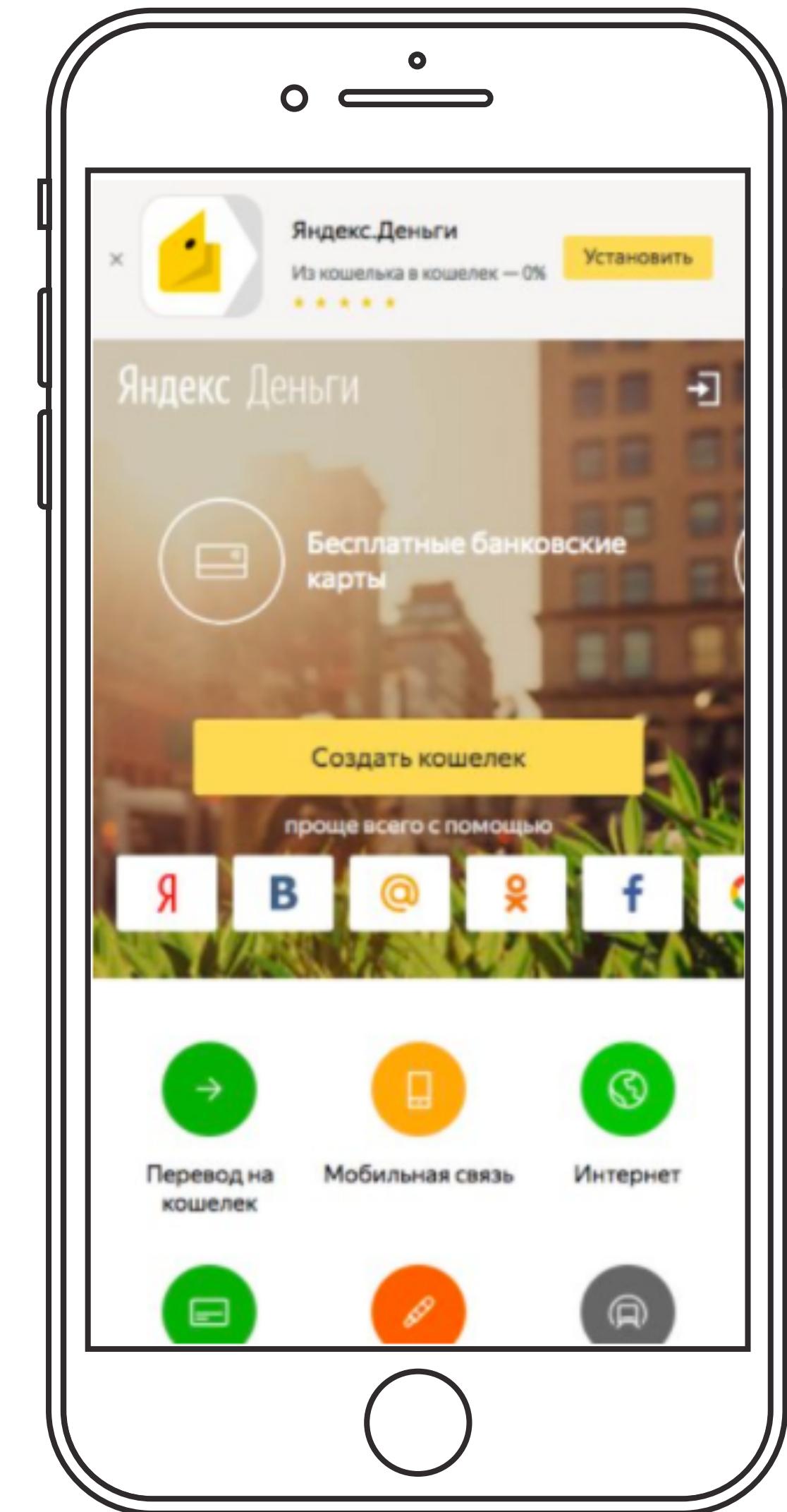
2 секунды



5 секунд



7 секунд

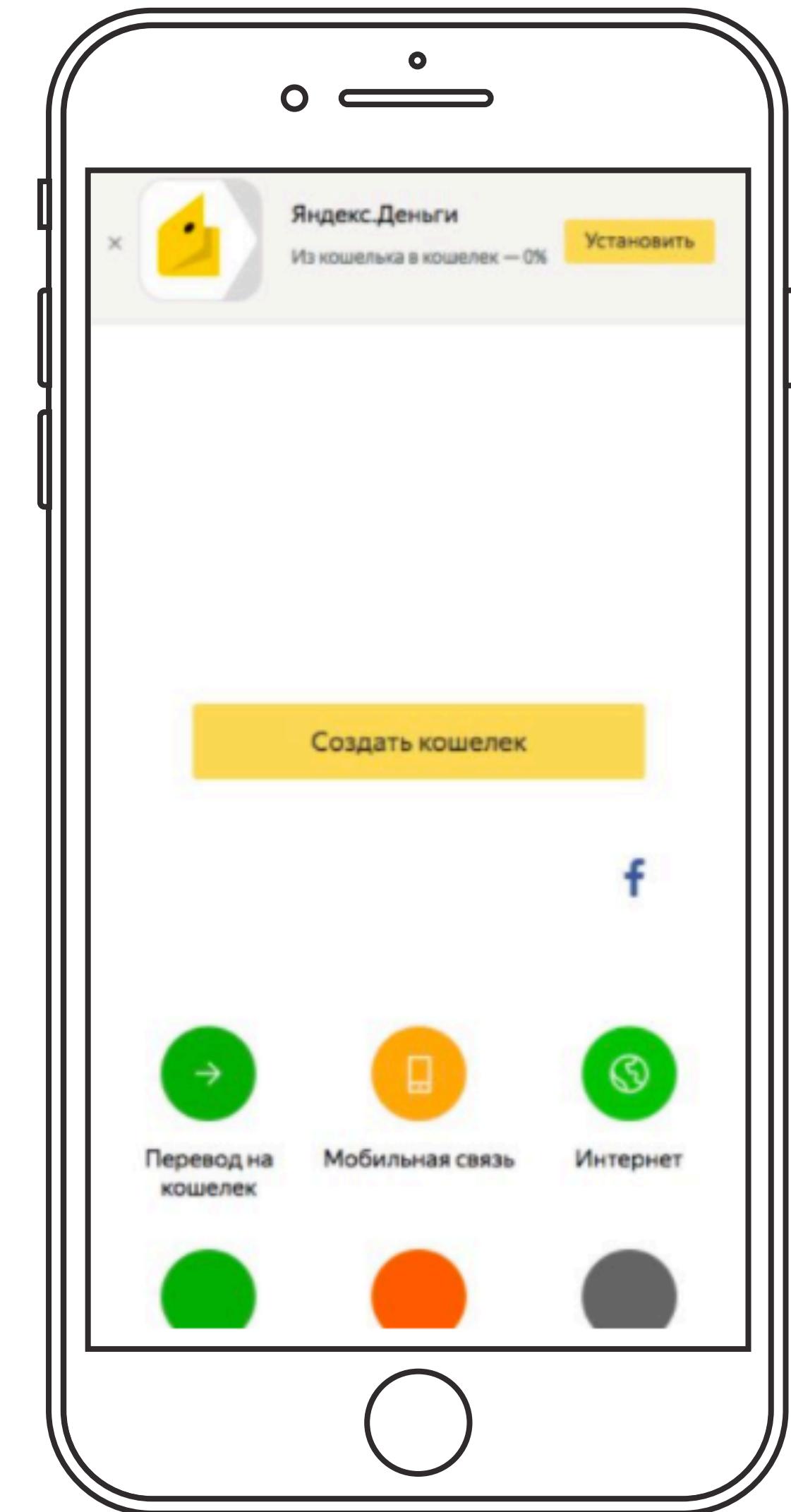


Этот чудесный preload

```
<link rel="preload" href="//yastatic.net/islands/_/GEumJGdz6PuI2jZ6GhSq0paPvho.woff2"  
as="font" type="font/woff2" crossorigin>
```



3.5 секунды

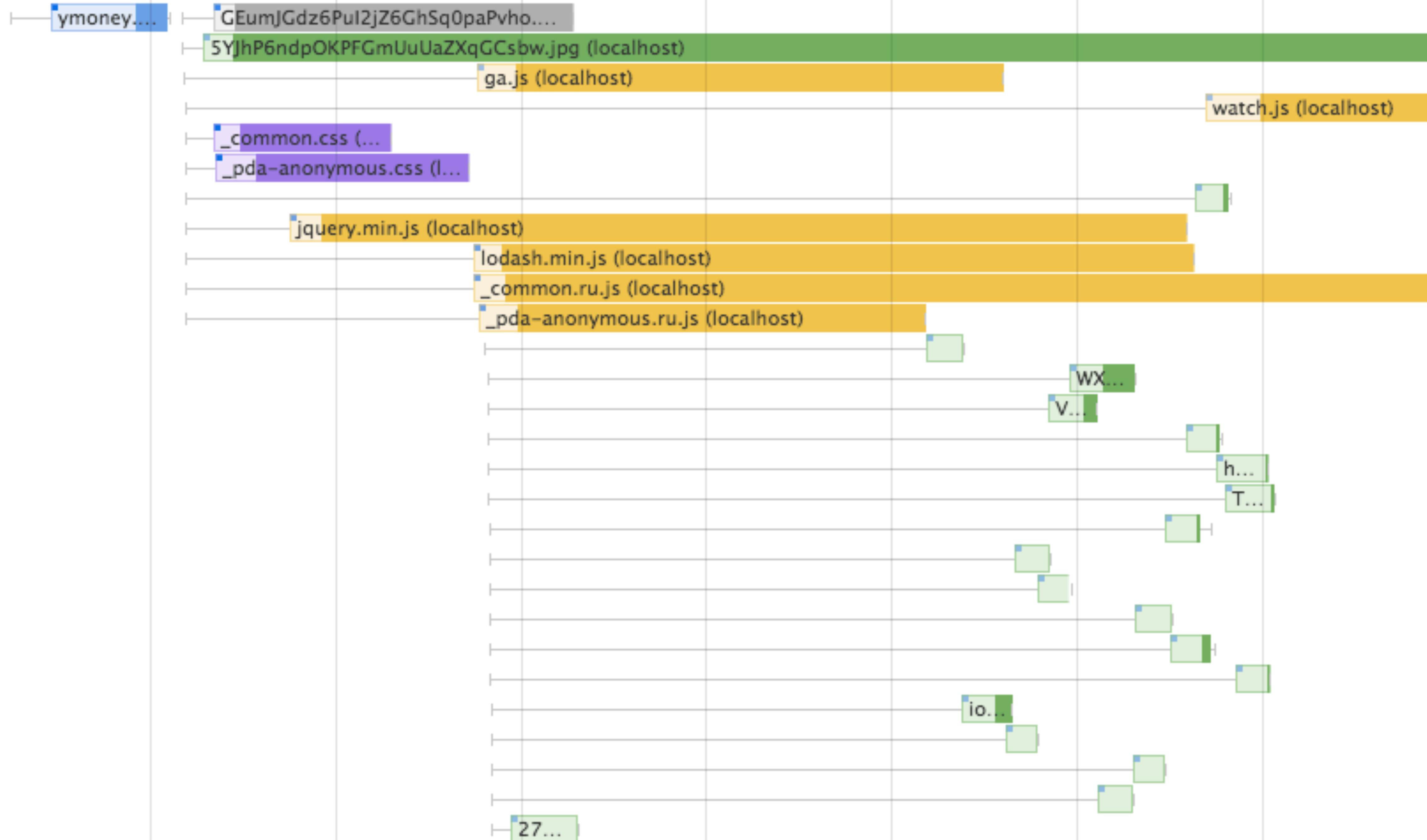


Добавим главное изображение

```
<link rel="preload" href="b/_/5YJhP6ndpOKPFGmUuUaZXqGCsbw.jpg" as="image"  
type="image/jpeg">
```

1000 ms 2000 ms 3000 ms 4000 ms 5000 ms 6000 ms 7000 ms 8000 ms 90

▼ Network







HTTP/0.9 – 1991 год

*HyperText Transfer Protocol



HTTP/1.0 – 1996 год



HTTP/1.1 – 1999 год



Камень в лесу

@kamen_v_lesu

Читать



Сегодня ничего не произошло.

10:53 - 13 окт. 2016 г.

44 ретвита 78 отметок «Нравится»



3

44

78



SPDY — 2012 год



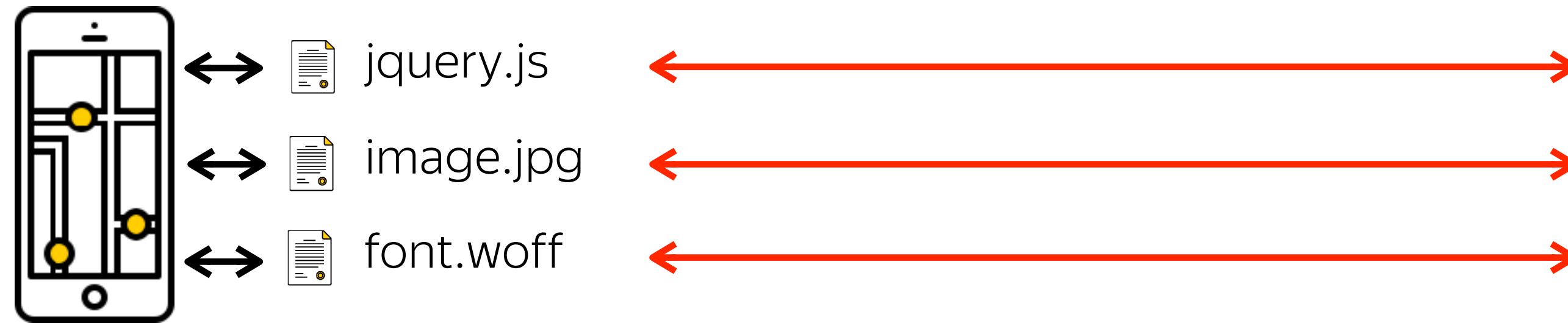
HTTP/2 – 2015 год

Мультиплексирование параллельных запросов

HTTP1.1

TCP-соединения

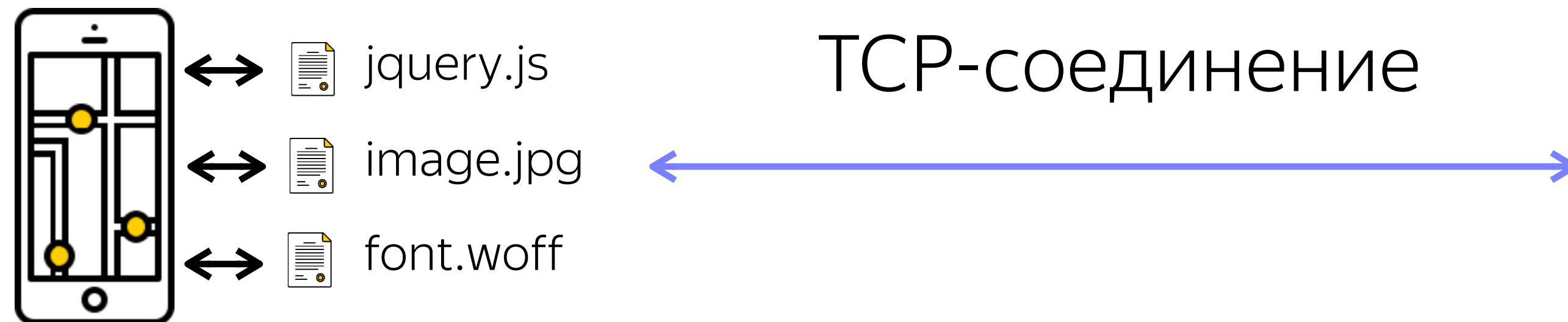
Сервер



HTTP/2

TCP-соединение

Сервер



Node.js 8.4.0 ❤️ HTTP/2

```
const http2 = require('http2');

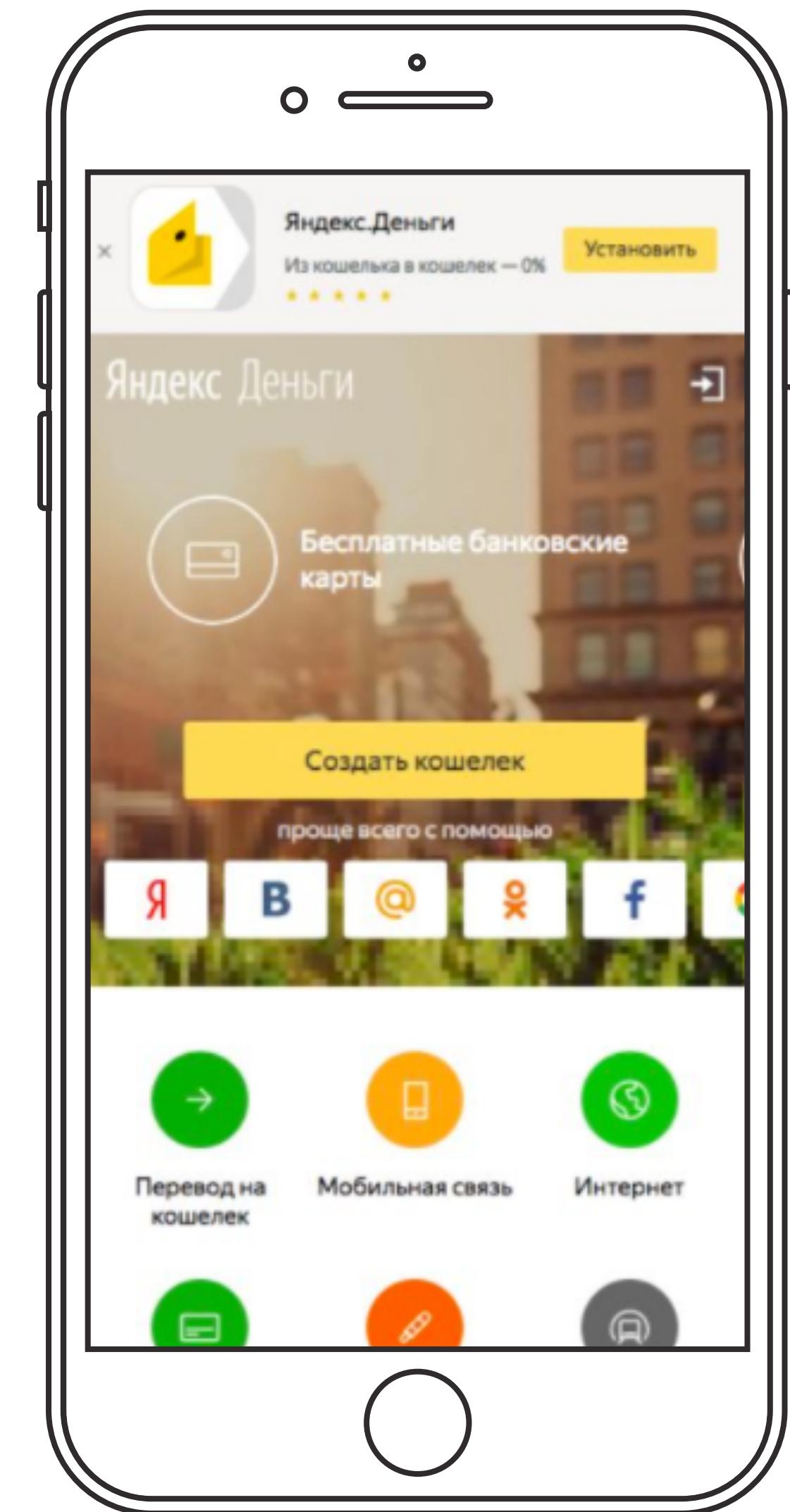
const options = {
    key: fs.readFileSync('/etc/nginx/ssl/key.pem'),
    cert: fs.readFileSync('/etc/nginx/ssl/cert.pem')
};

const server = http2.createSecureServer(options);
server.on('stream', (stream, headers) => {
    /* Читаем файл и создаём readStream */
    readStream.on('open', function () {
        stream.respond({'content-type': mimeType[ext] || 'text/plain', ':status': 200});
        readStream.pipe(stream);
    });
}
server.listen(8800);
```



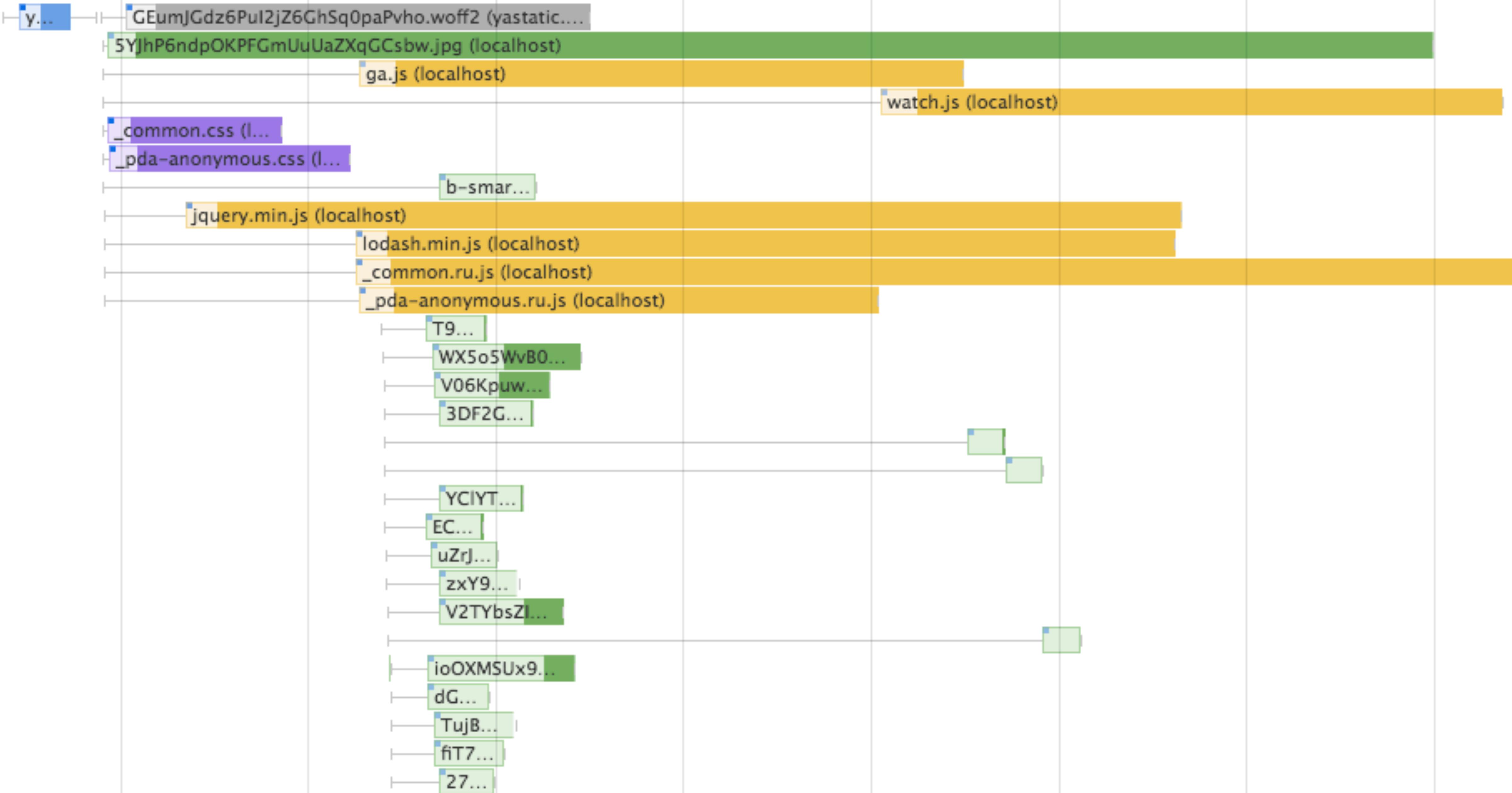
```
node --expose-http2
```

3.4 секунды



1000 ms 2000 ms 3000 ms 4000 ms 5000 ms 6000 ms 7000 ms 8000 ms

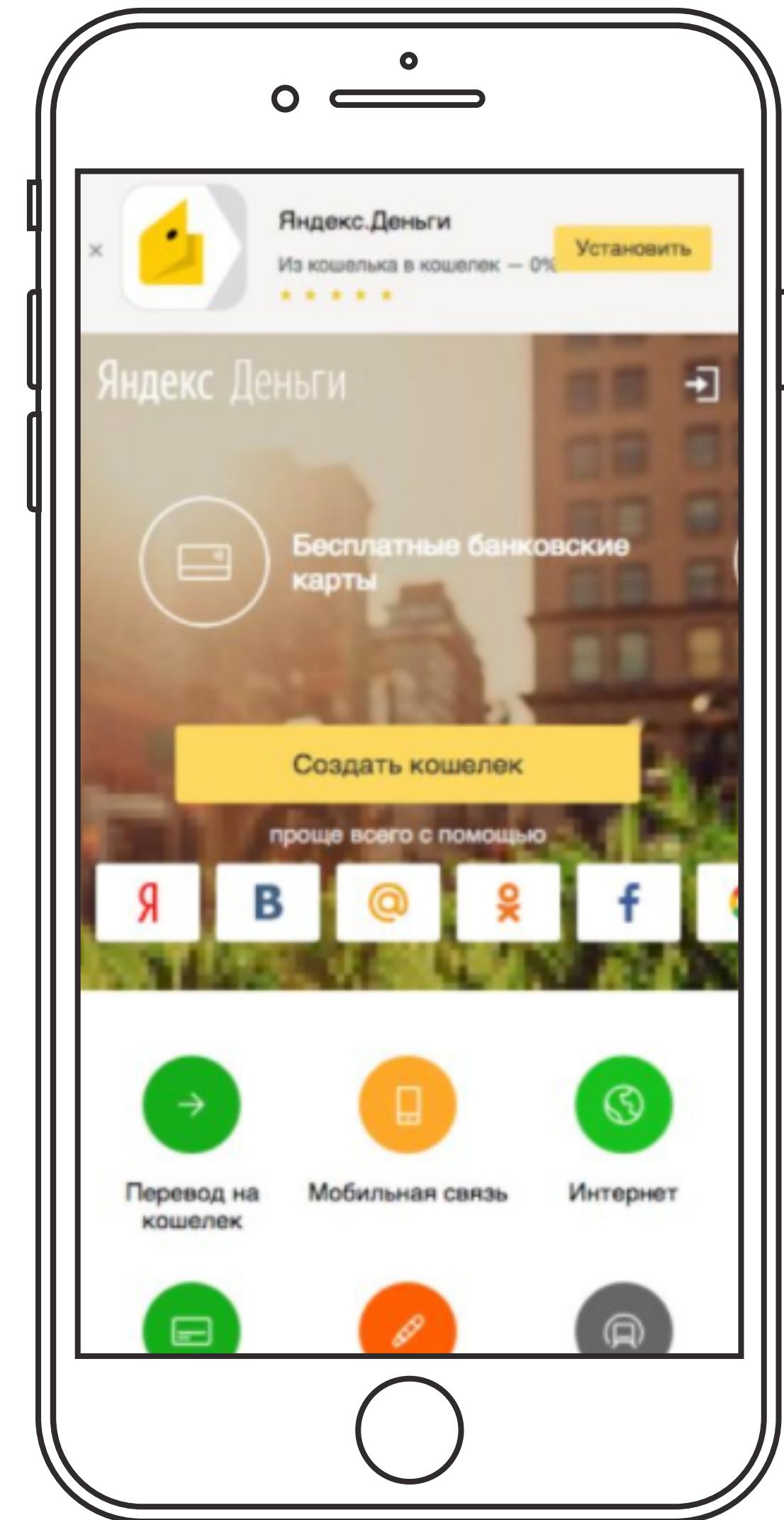
▼ Network





`font-display: swap;`

2.4 секунды





```
font-display: fallback;
```

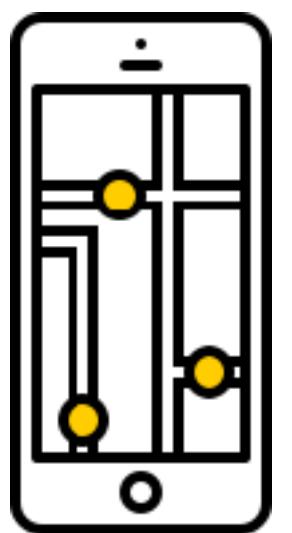


Удаляем data:image?

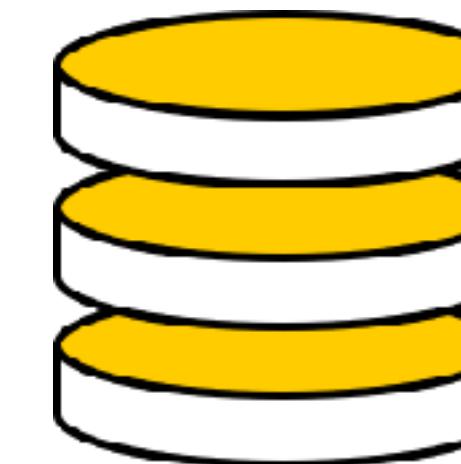
Server Push

Server Push

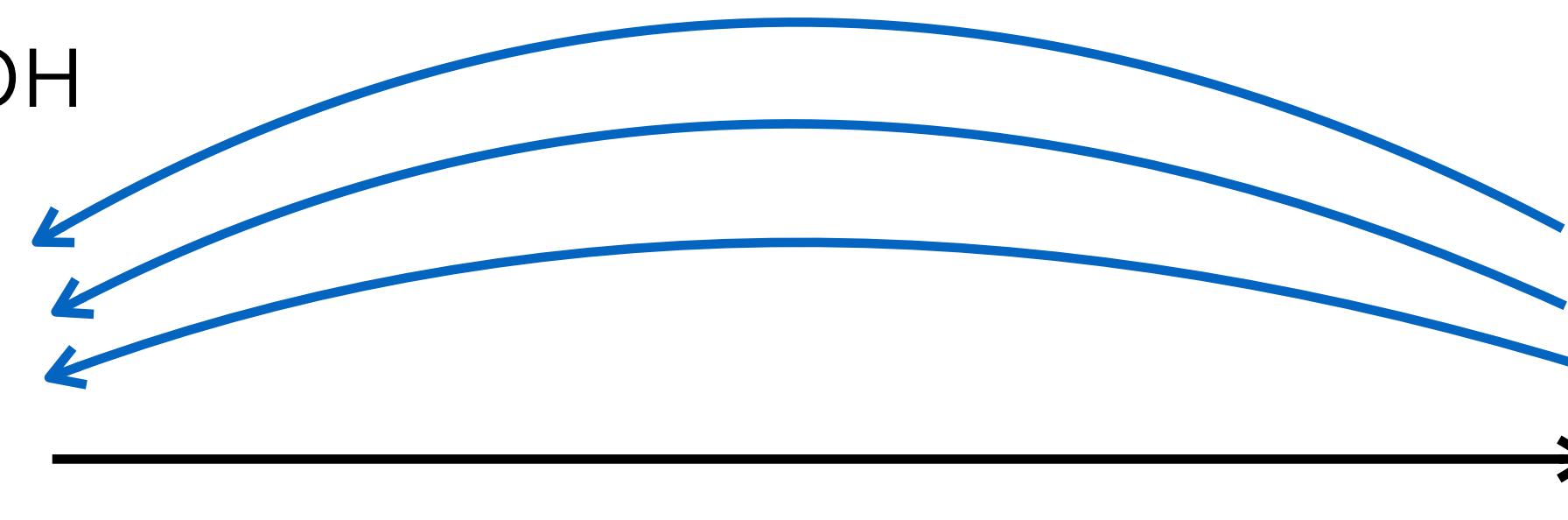
Смартфон



Сервер

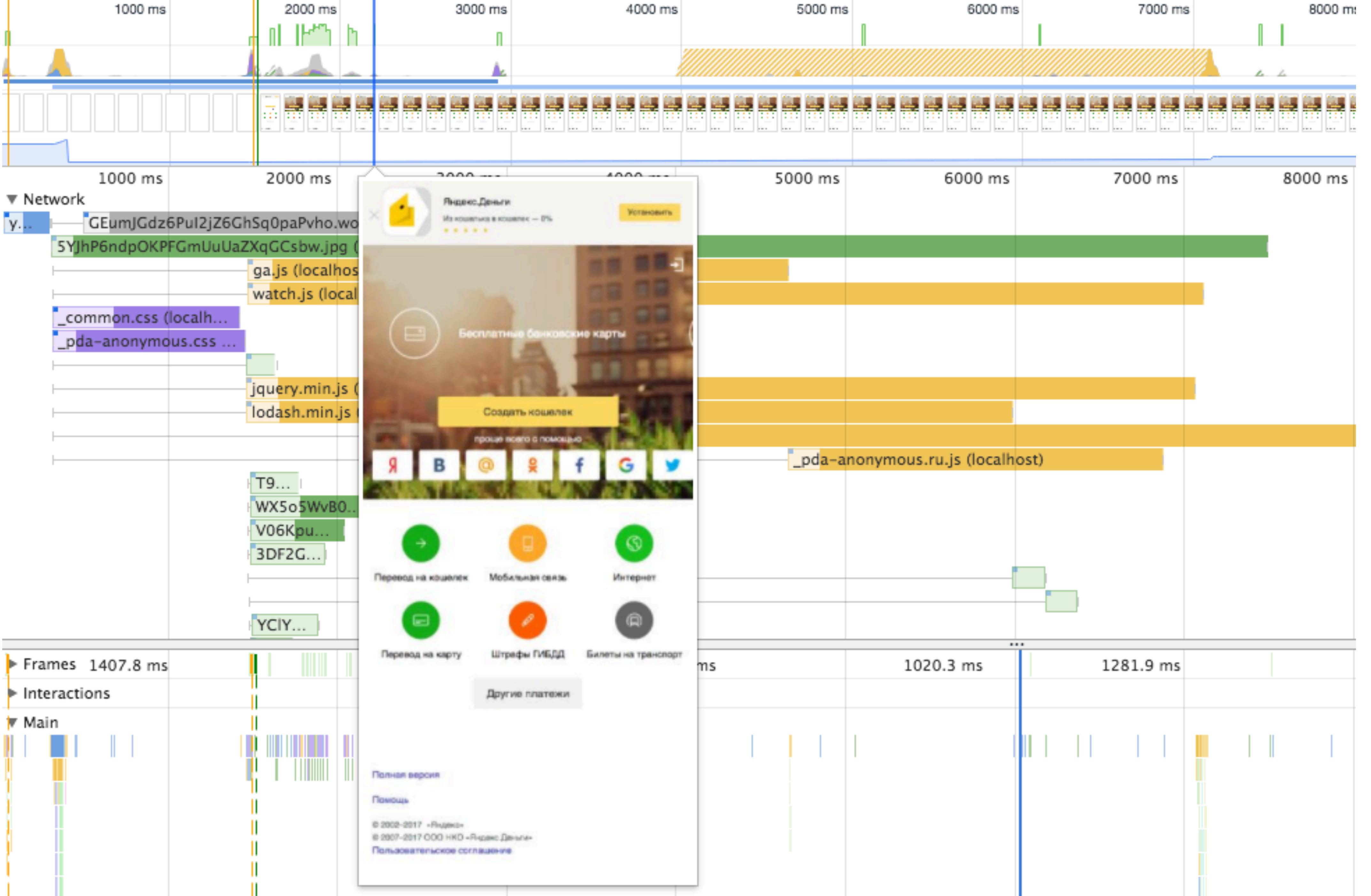


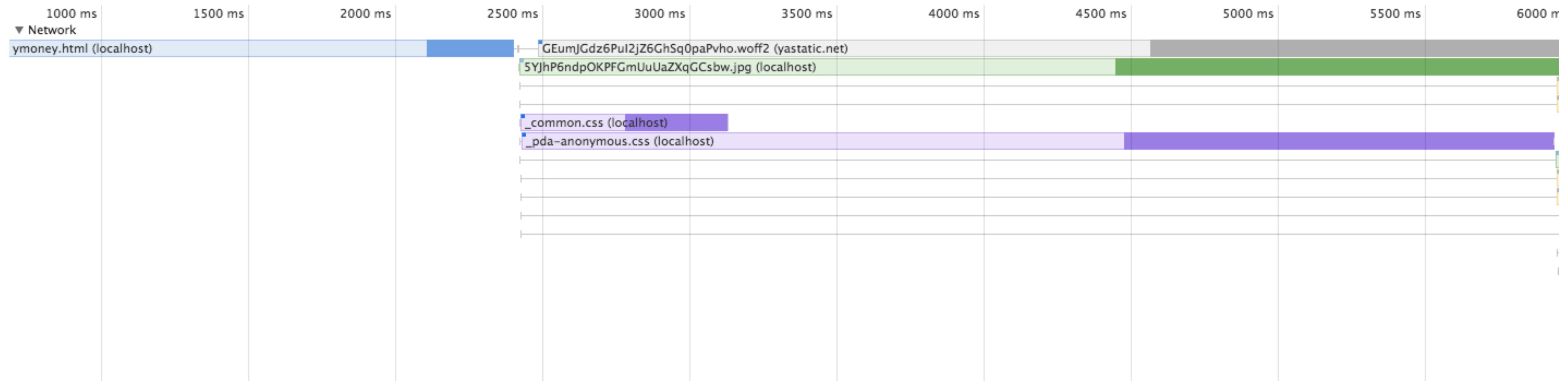
HTML



```
server.on('stream', (stream, headers) => {
  if (headers[':path'] === '/ymoney.html') {
    stream.pushStream({ ':path': '/ymoney_files/_common.css' }, (pushStream) => {
      pushStream.respond({ ':status': 200, 'content-type': 'text/css' });
      pushStream.end(commonCSS, () => {
        });
      });
    const indexPath = path.join(__dirname, pathname);
    const readStream = fs.createReadStream(indexPath);
    readStream.on('open', function () {
      stream.respond({'content-type': mimeType[ext] || 'text/plain', ':status': 200});
      readStream.pipe(stream);
    });
  }
  /* Остальная статика */
});
```

 5YJhP6ndp...	200	jpeg	ymoney.html	1...	7.23 s	120983	
 _common.css	200	styl...	Push / ymoney.html	1...	1.20 s	120983	
 _pda-anony...	200	styl...	ymoney.html	2...	1.19 s	120983	





Name	Initiator	Size	Time	Waterfall	4.00 s	6.00 s	8.00 s	10.00 s	12.00 s	14.
ymoney.html	Other	15.6 KB	2.33 s							
GEumJGdz6Pul2jZ6GhSq0paPvho.woff2	ymoney.html	39.5 KB	4.23 s							
5YjhP6ndpOKPFGmUuUaZXqGCsbw.jpg	ymoney.html	135 KB	8.91 s							
_common.css	Push / ymoney.html	17.8 KB	710 ms							
_pda-anonymous.css	ymoney.html	25.0 KB	3.46 s							

Недостатки Server Push

- › Не знает о браузерном кэше
- › Не поддерживает приоритизацию ресурсов



Server Push ❤ Cookies



H2O

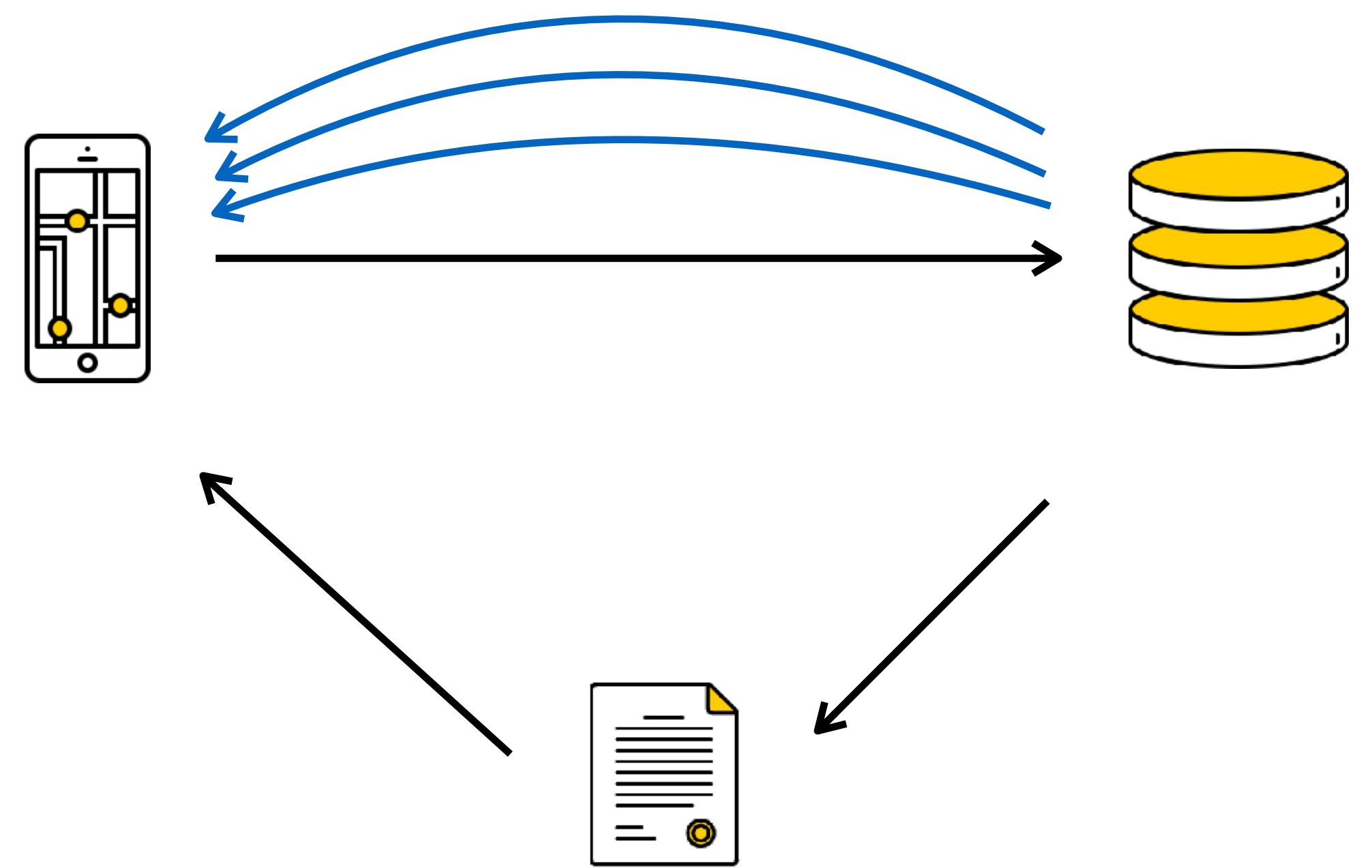
link: <yastatic.net/islands/_/GEumJGdz6PuI2jZ6GhSq0paPvho.woff2>; rel=preload

H2O

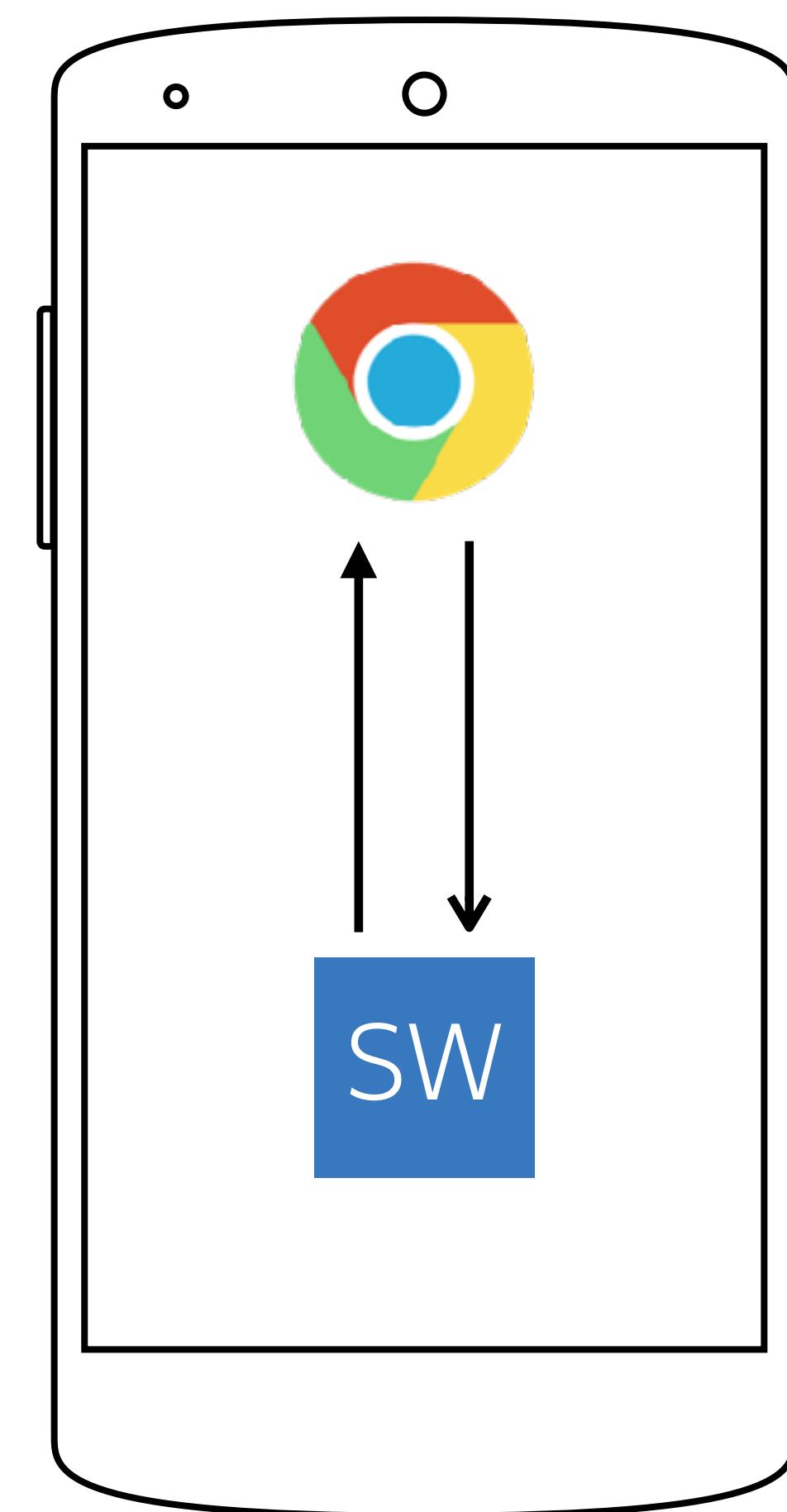
link: <yastatic.net/islands/_/GEumJGdz6PuI2jZ6GhSq0paPvho.woff2>; rel=preload; nopush

Server Push ❤️ SW

Первая загрузка



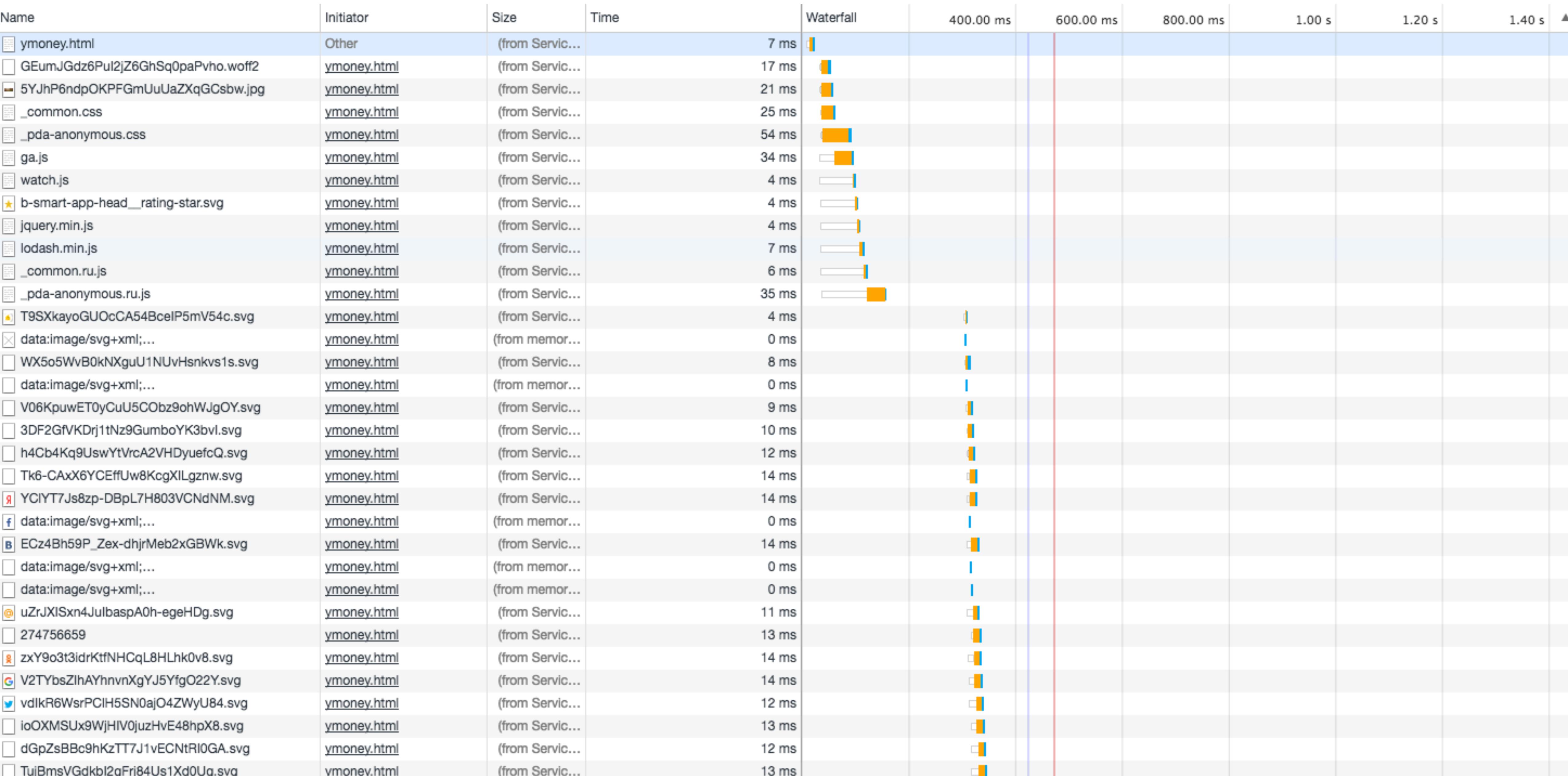
Следующие загрузки



```
let cacheName = 'v1';

this.addEventListener('install', function(event) {
  event.waitUntil(
    caches.open(cacheName).then(function(cache) {
      return cache.addAll([
        'ymoney.html',
        'ymoney_files/_common.css'
      ].map(u => new Request(u, { credentials: 'include' })))
    }).then(()=> console.log('done'))
  );
  this.skipWaiting();
});
```

```
this.addEventListener('fetch', function(event) {  
  console.log('SW Fetching... ')  
  var response;  
  event.respondWith(async function() {  
    const cachedResponse = await caches.match(event.request);  
    if (cachedResponse) return cachedResponse;  
    response = await fetch(event.request);  
    caches.open(cacheName).then(function(cache) {  
      cache.put(event.request, response);  
    });  
    return response.clone();  
  }());  
});
```



Оптимизация JS



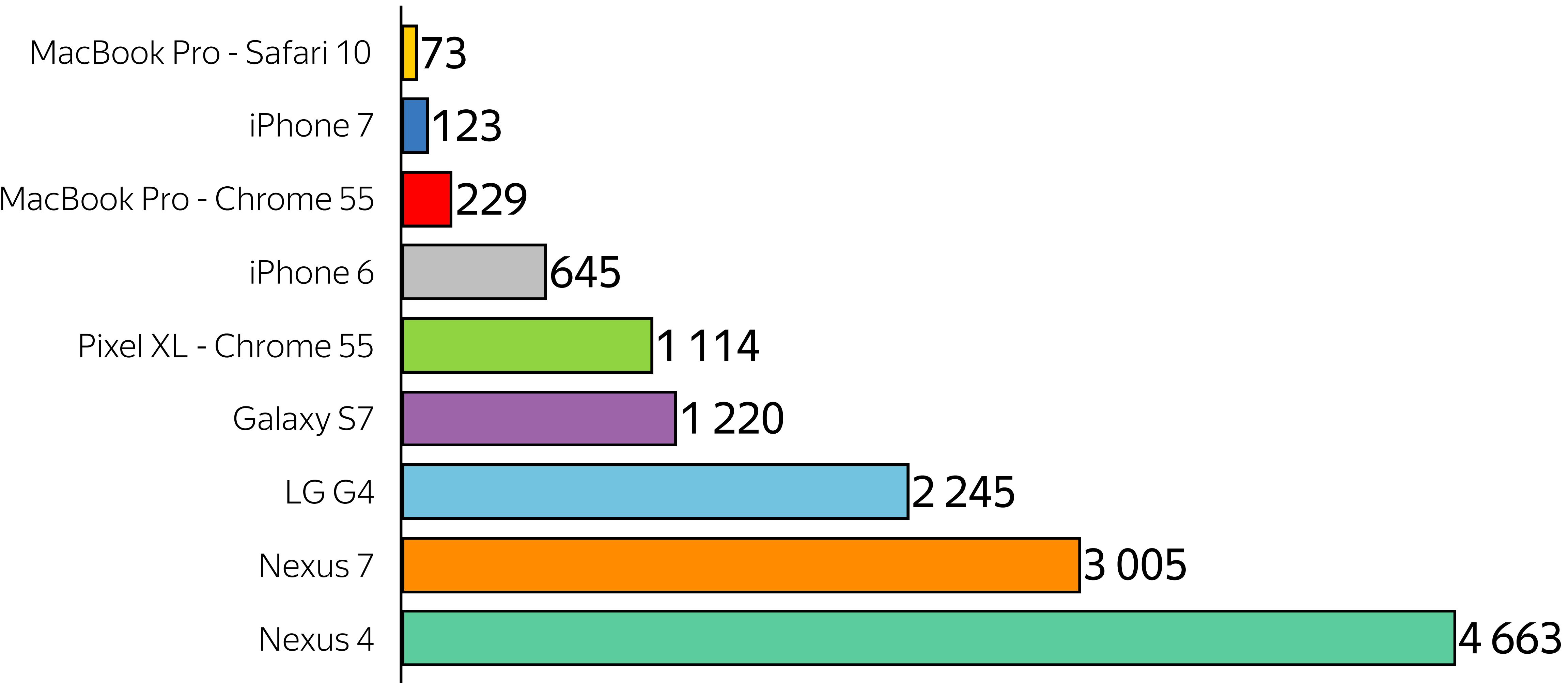
Средний размер загружаемого JS

| 113кб JS в 2010

| 446кб JS в 2017

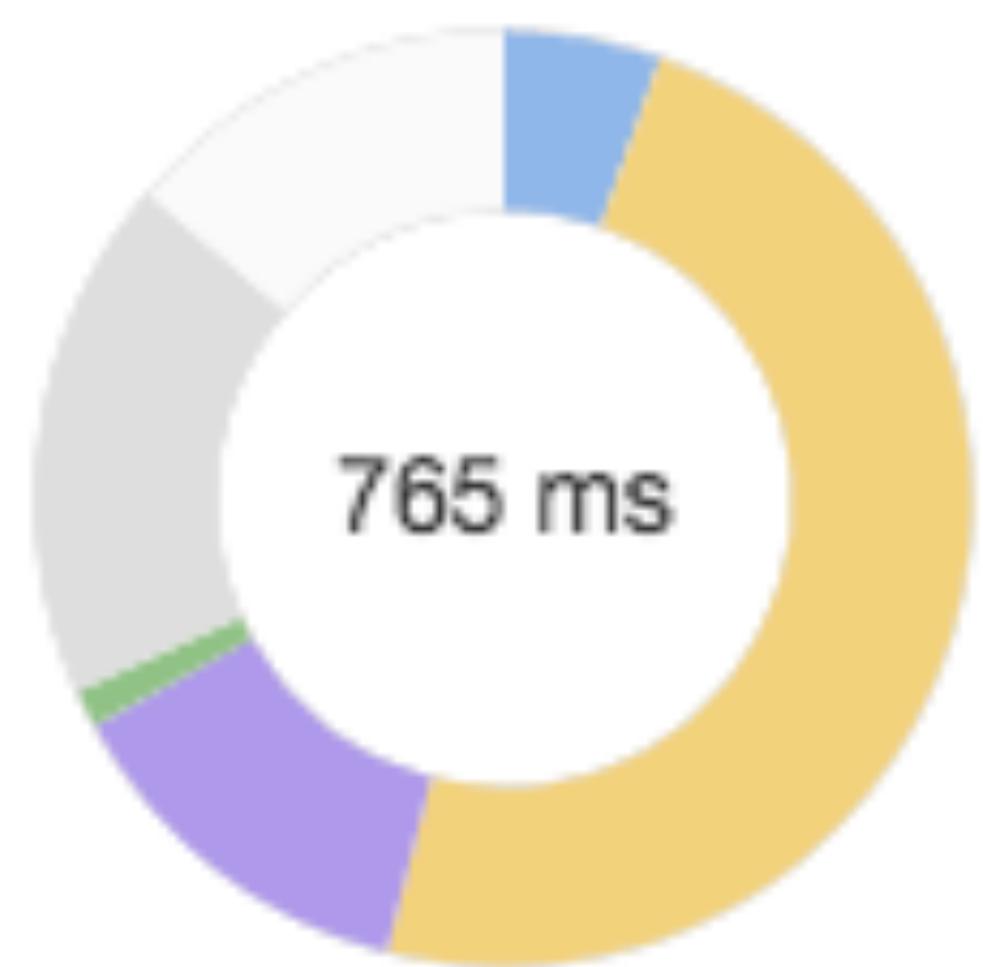
* По данным <http://httparchive.org/>

Время первичного анализа 1mb JS



MacBook Pro

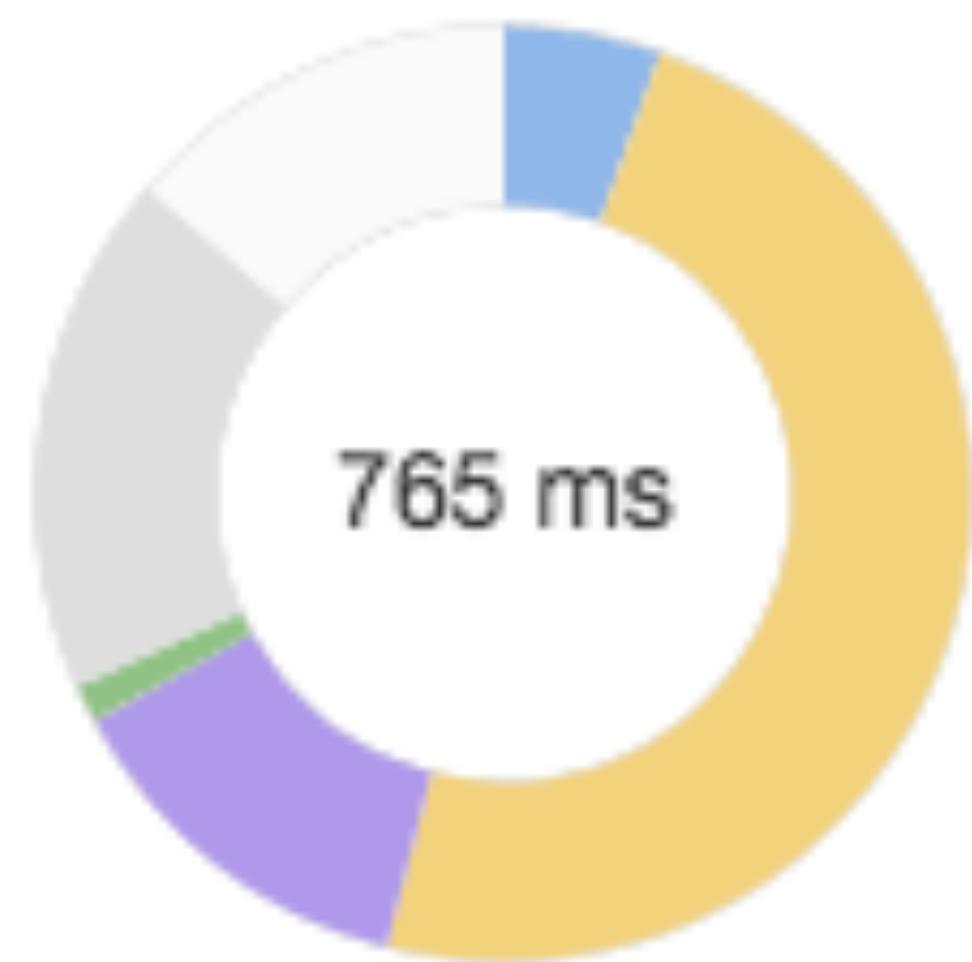
Range: 0 – 765 ms



- 41.5 ms ■ Loading
- 371.7 ms ■ Scripting
- 98.6 ms ■ Rendering
- 10.0 ms ■ Painting
- 137.7 ms ■ Other
- 105.0 ms ■ Idle

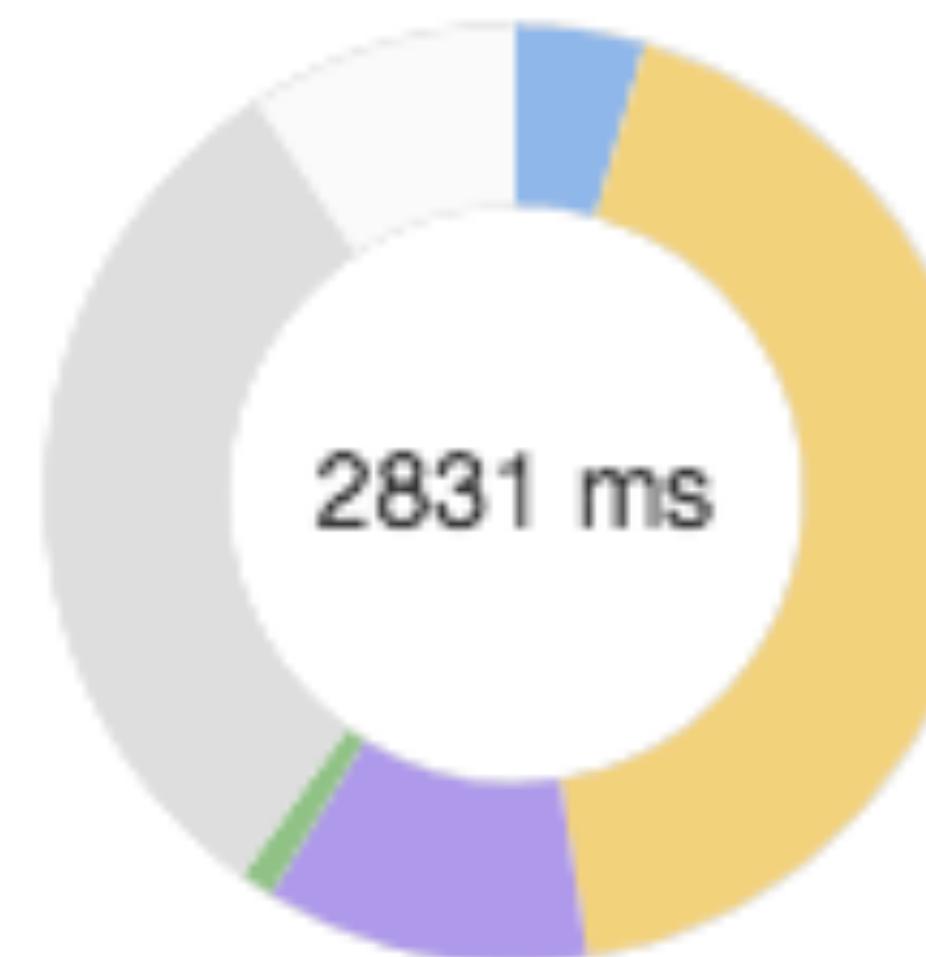
MacBook Pro

Range: 0 – 765 ms

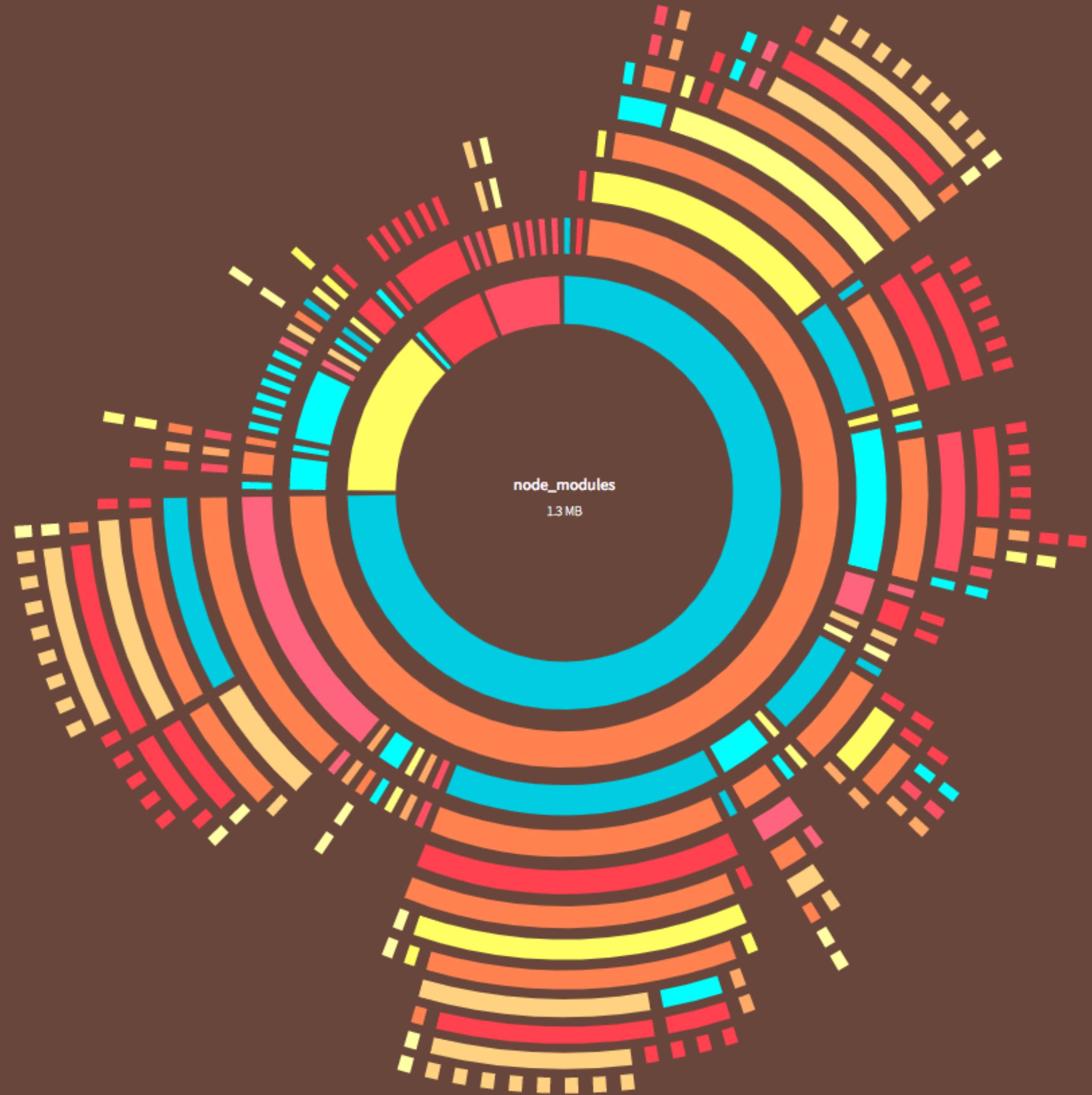


MacBook Pro с троттлингом в 5x

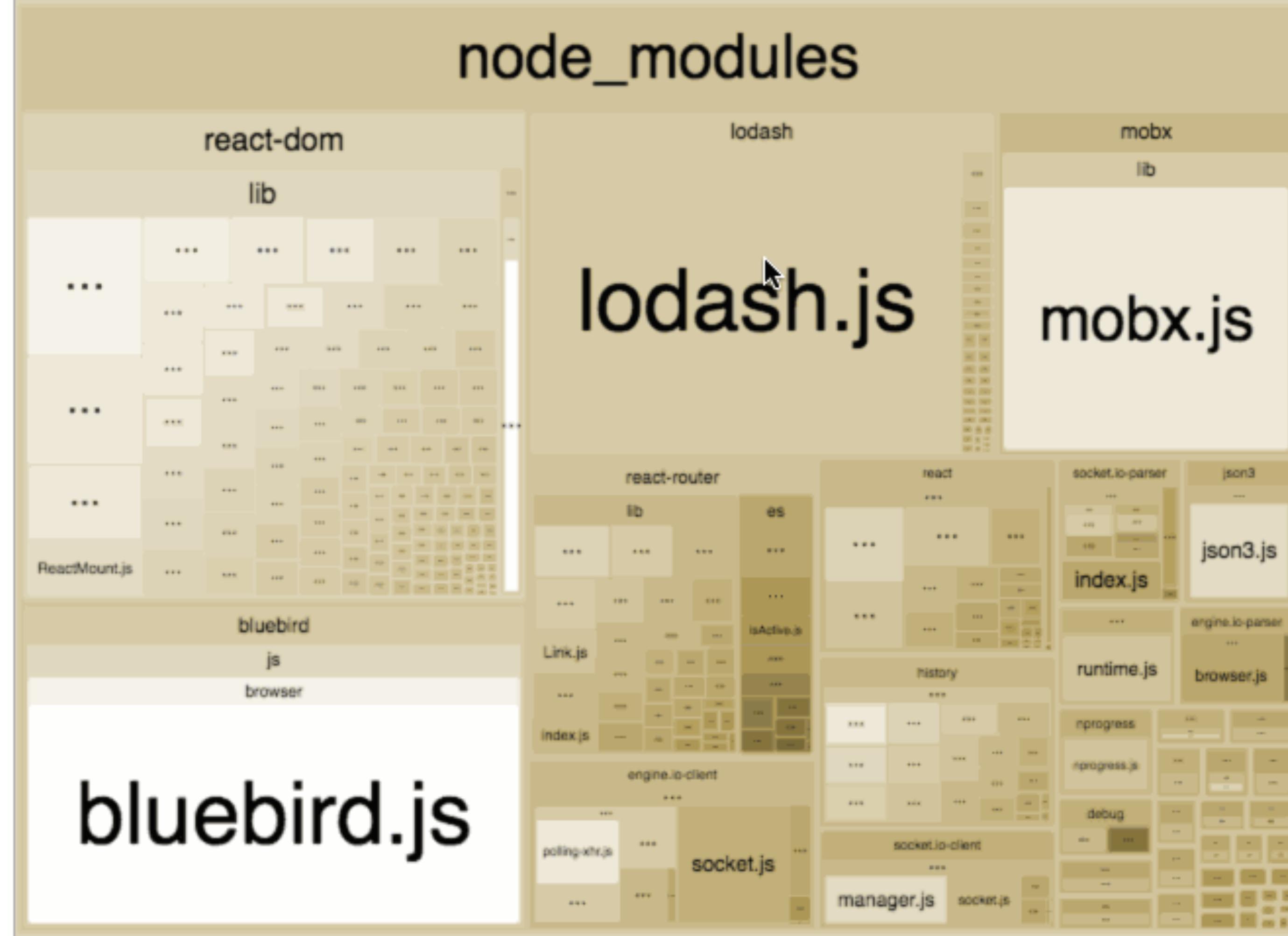
Range: 0 – 2.83 s



126.6 ms	■	Loading
1219.3 ms	■	Scripting
316.8 ms	■	Rendering
32.8 ms	■	Painting
873.0 ms	■	Other
262.5 ms	□	Idle



vendor.f72c2e816850a7014aaa.js



lodash.js

mobx.js

bluebird.js



webpack-bundle-analyzer

Что можно сделать ещё?

- › Побороться с ленивым синтаксическим анализом
- › Разбивать код на несколько бандлов

Следите за метриками



- › Google Lighthouse
- › calibreapp.com
- › Navigation Timing API



Выводы

- › Используйте preload и font-display:swap
- › Используйте HTTP/2
- › Попробуйте Server Push и Service Workers
- › Уменьшайте ваши бандлы
- › Тестируйте на реальных устройствах в реальной сети

One more thing...

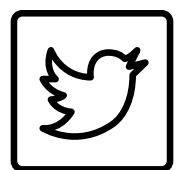
Доступно уже сегодня!

<https://clck.ru/BhGMm>

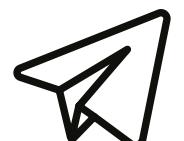


Спасибо за внимание!

Андрей Мелихов
«Яндекс.Деньги»



amel_true



amel_true