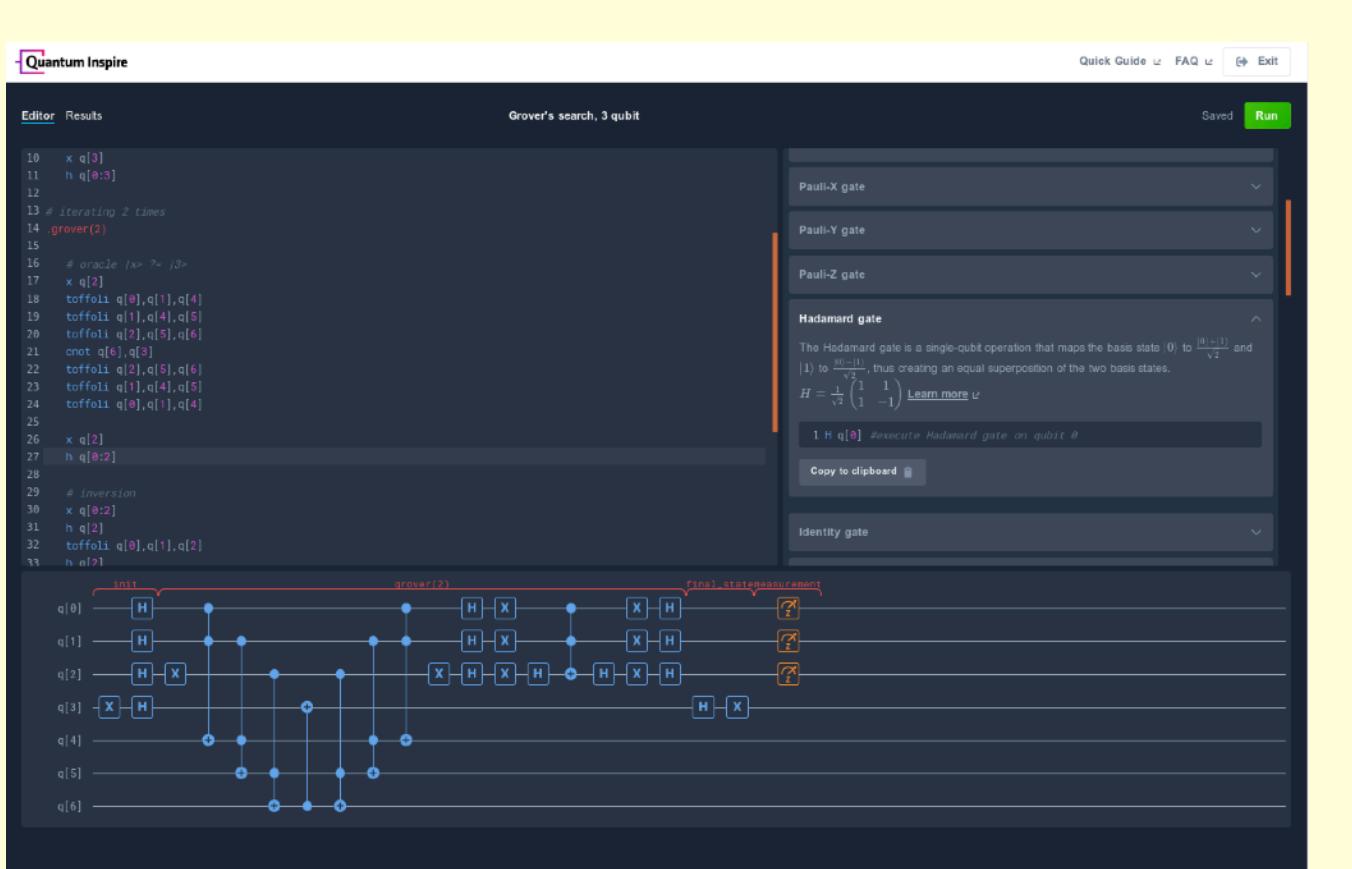




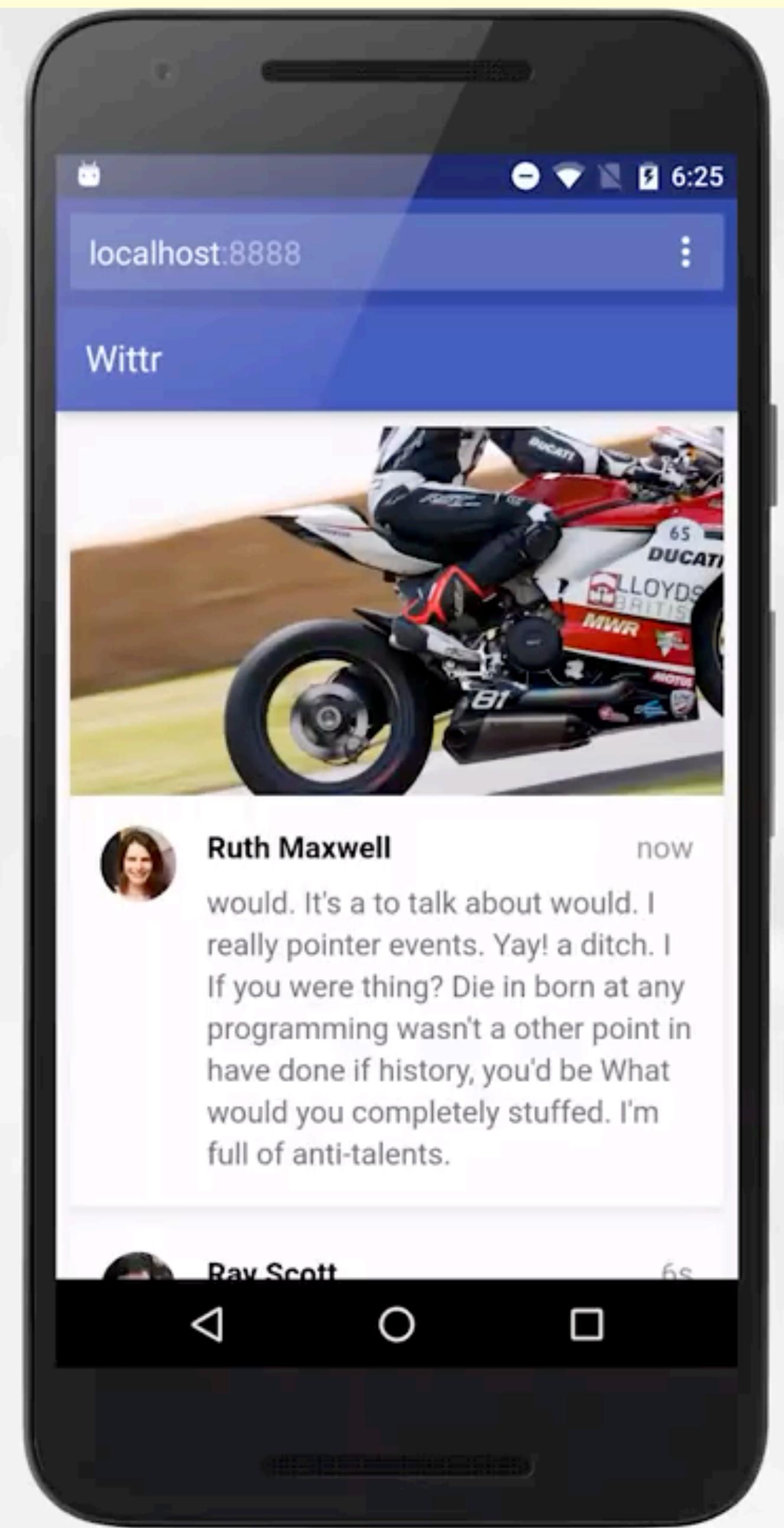
Serverside Rendering

Declan Rek

You might know us from



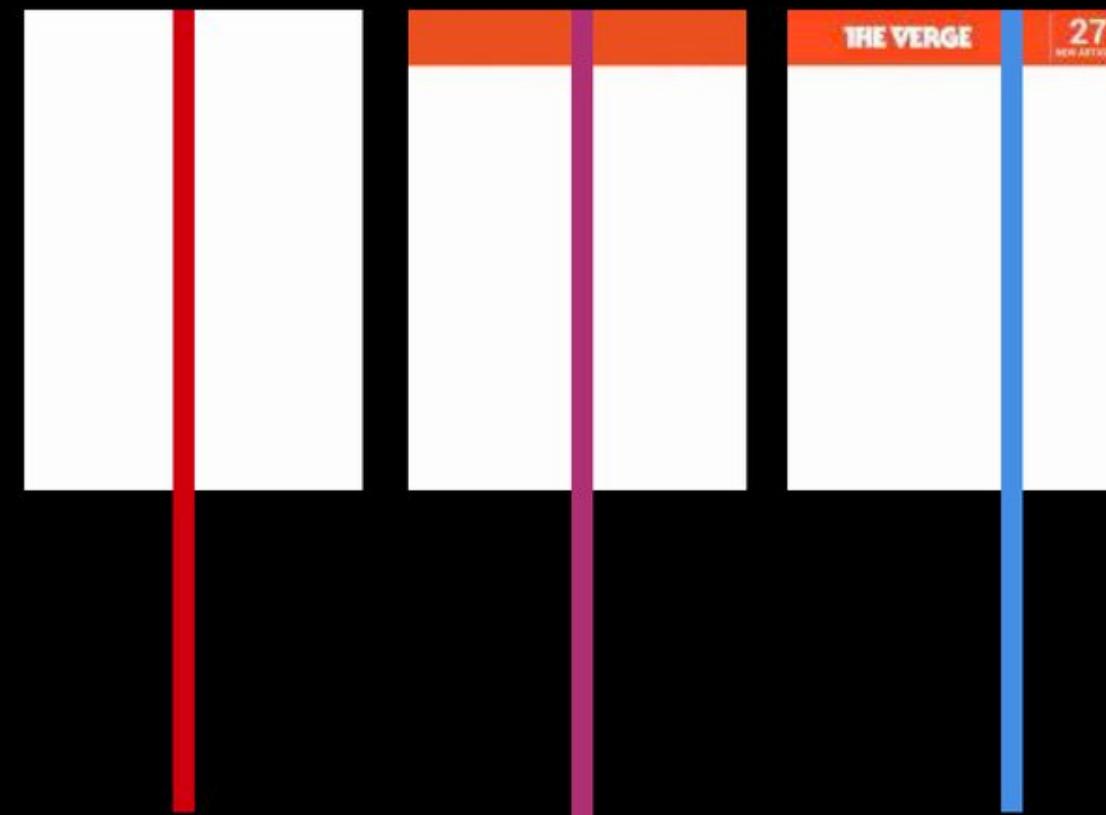
Resilience



We run software on devices we don't own

And we serve our software over networks
we don't control

is it happening?



Navigation
begins

Time to first byte

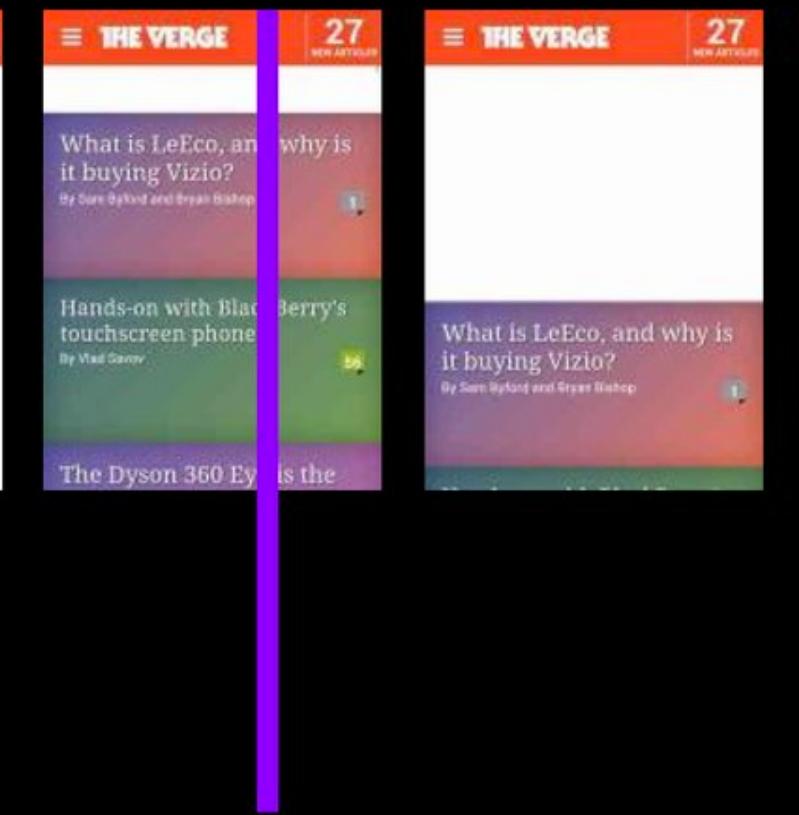
First
Contentful
Paint

Navigation has
successfully started

First Paint

The first non-blank
paint on screen

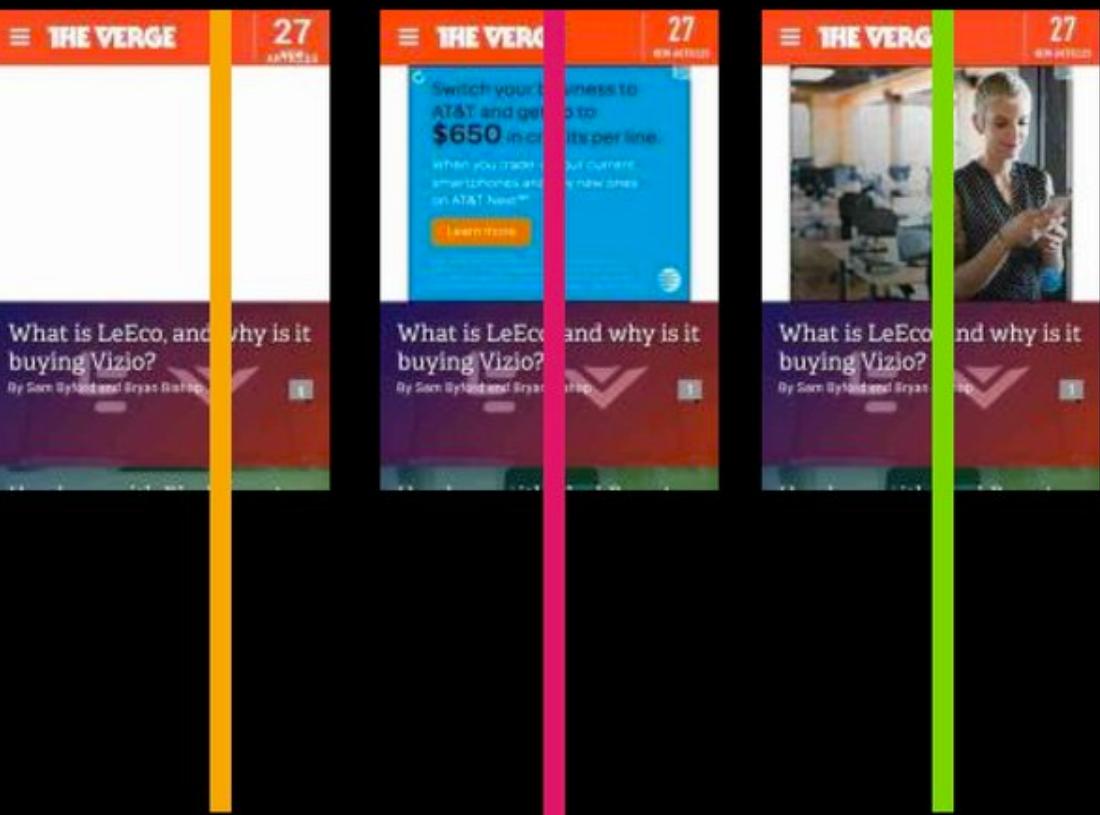
is it useful?



First
Meaningful
Paint

Page's primary
content is visible

is it usable?



Visually
ready

Page looks
nearly done

Fully
Loaded

End of
load lifecycle

Time to
Interactive

Visually usable
and engagable

TTFB: Time to First Byte – seen as the time between clicking a link and the first bit of content coming in.

FP: First Paint – the first time any pixel gets becomes visible to the user.

FCP: First Contentful Paint – the time when requested content (article body, etc) becomes visible.

TTI: Time To Interactive – the time at which a page becomes interactive (events wired up, etc).



Progressive rendering

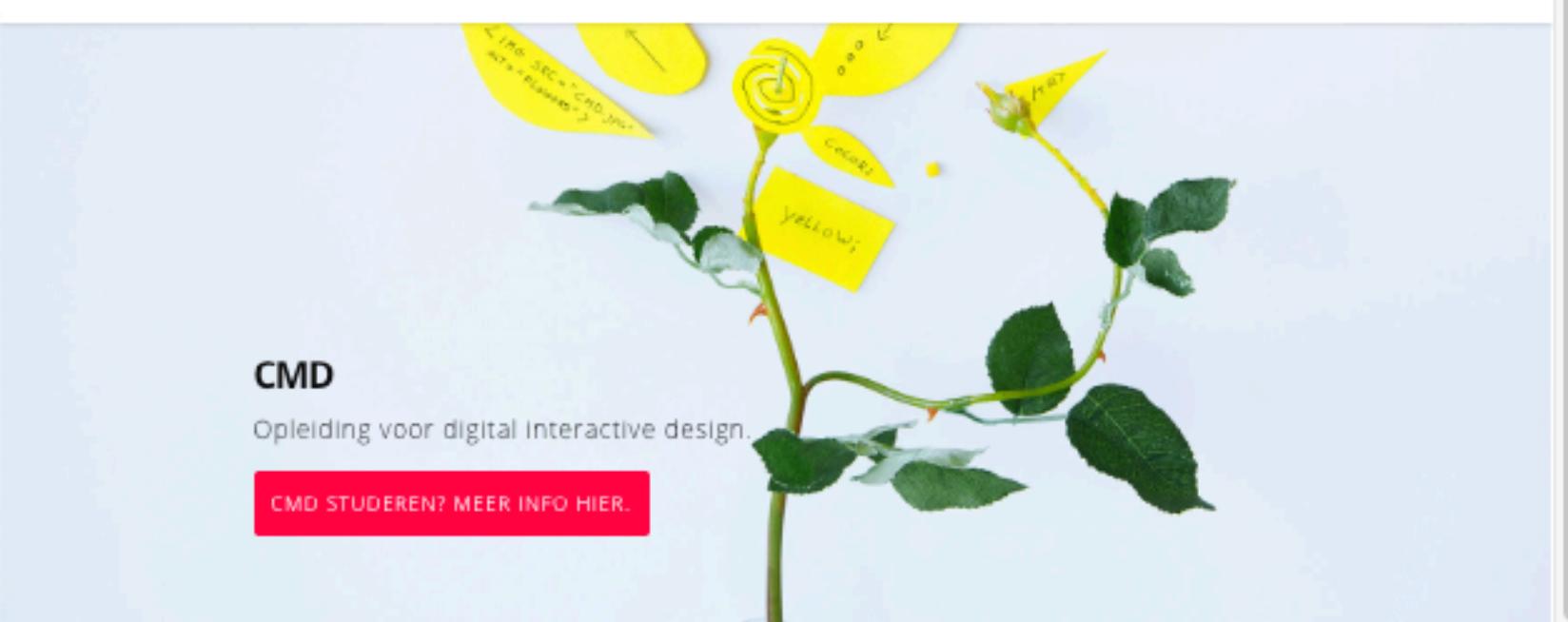
HTML has progressive rendering
built in and is first to arrive

HyperText Transfer Protocol

The protocol to
send and receive data between
client and server

method	URI	http version
POST	/create-user	HTTP/1.1
Host: localhost:3000		
Connection: keep-alive		
Content-type: application/json		
{ "name": "John", "age: 35 }		
http ver.	status	
HTTP/1.1	200	OK
Date: 2017-01-10 12:28:53 GMT		
Server: Apache/2.2.14		
Content-type: text/html		
<h1>Hello World</h1>		

[is a get
request over HTTPS](https://www.cmd-amsterdam.nl/)

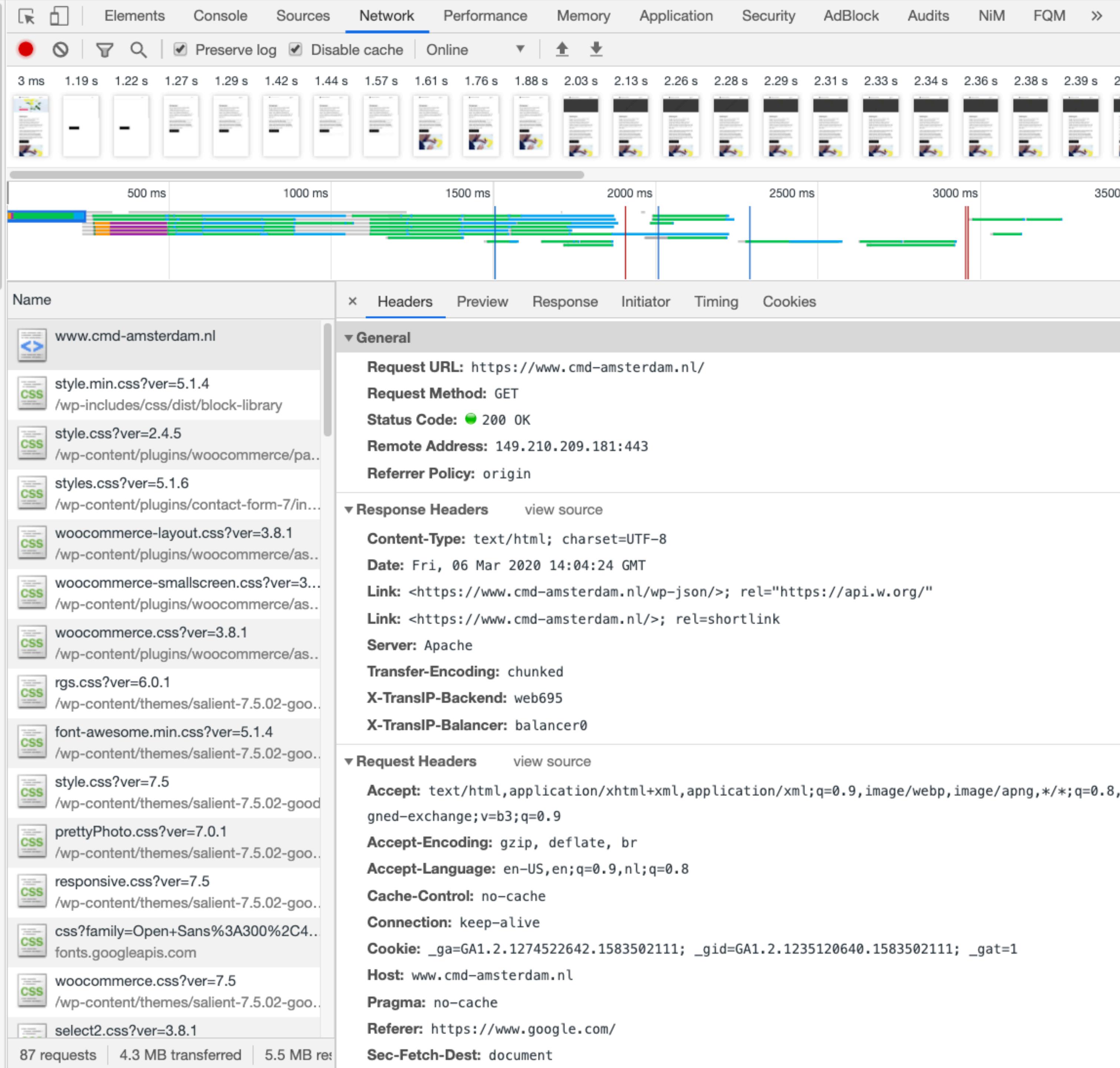
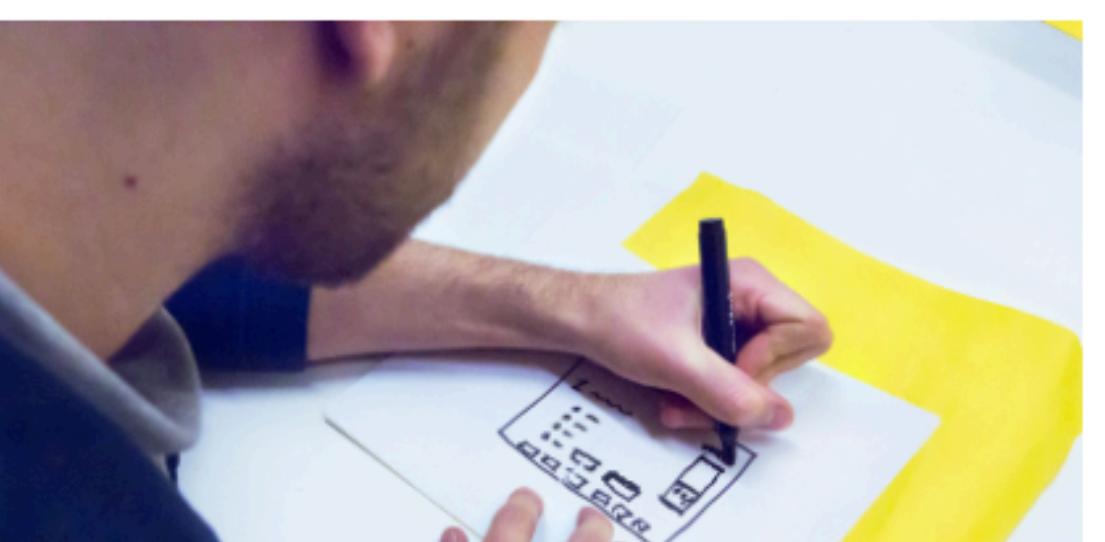


Ontwerpen

Wij zijn Communication and Multimedia Design (CMD). Wij zijn een HBO-ontwerpopleiding voor digital interactive design. Onze studenten ontwerpen en realiseren digitale interactieve oplossingen die optimaal aansluiten bij de behoeften van de gebruikers.

In de kern van de opleiding zit interaction design, visual design en techniek (met name frontend development). Onze vakdocenten hebben een goed netwerk en vertalen continu recente ontwikkelingen naar het gehele onderwijsprogramma. Afgestudeerde studenten zijn daarmee van grote waarde in het huidige werkveld. Daar zijn wij trots op!

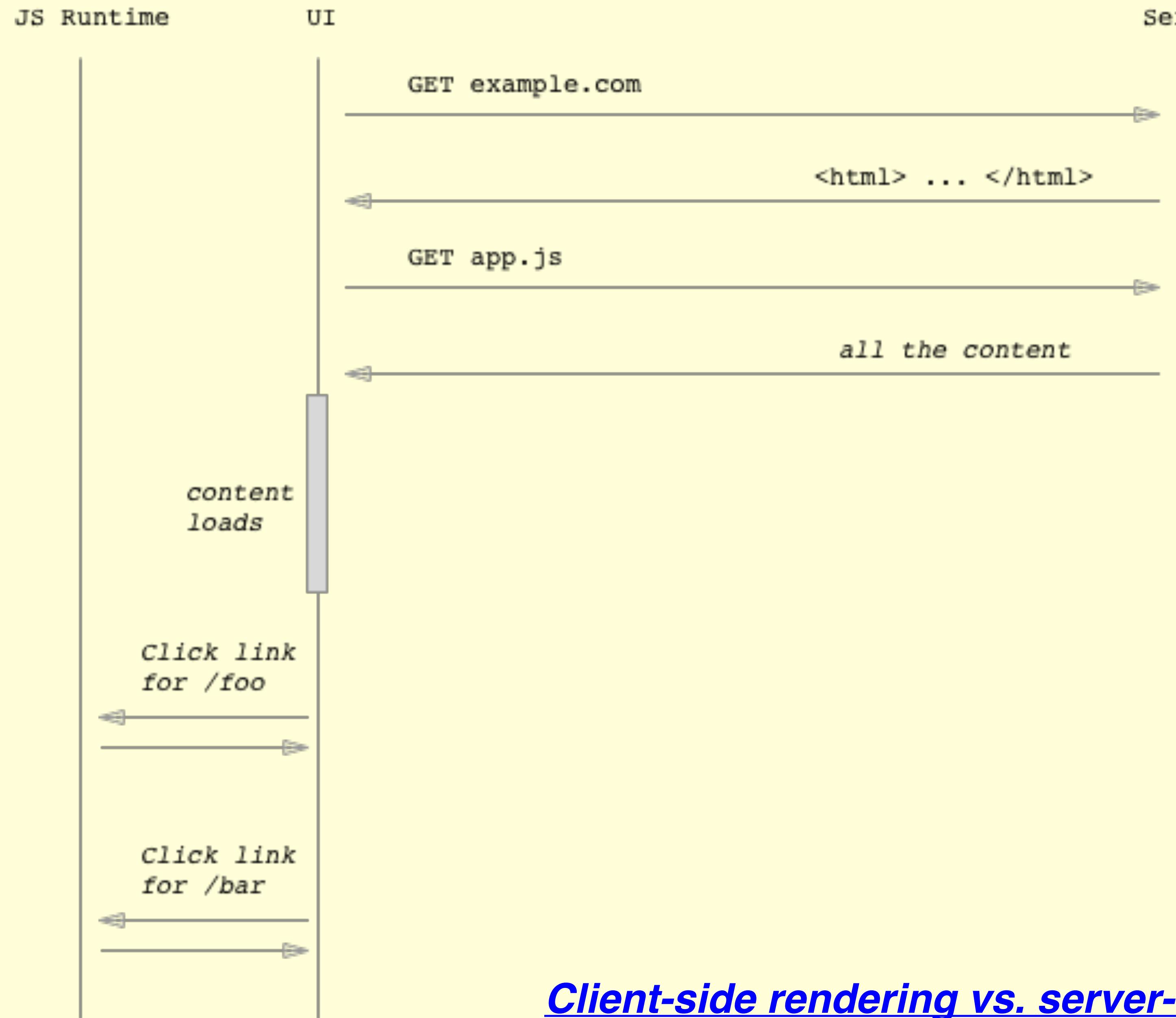
[MEER OVER CMD](#)



Client side rendering

All logic, data fetching, templating and routing are handled on the client

Client side rendering



Client-side rendering vs. server-side rendering

+

Faster when app is loaded and JS is
parsed

“Appy” experience

Static files can deployed to CDN

-

Long initial load

Prone to JS errors

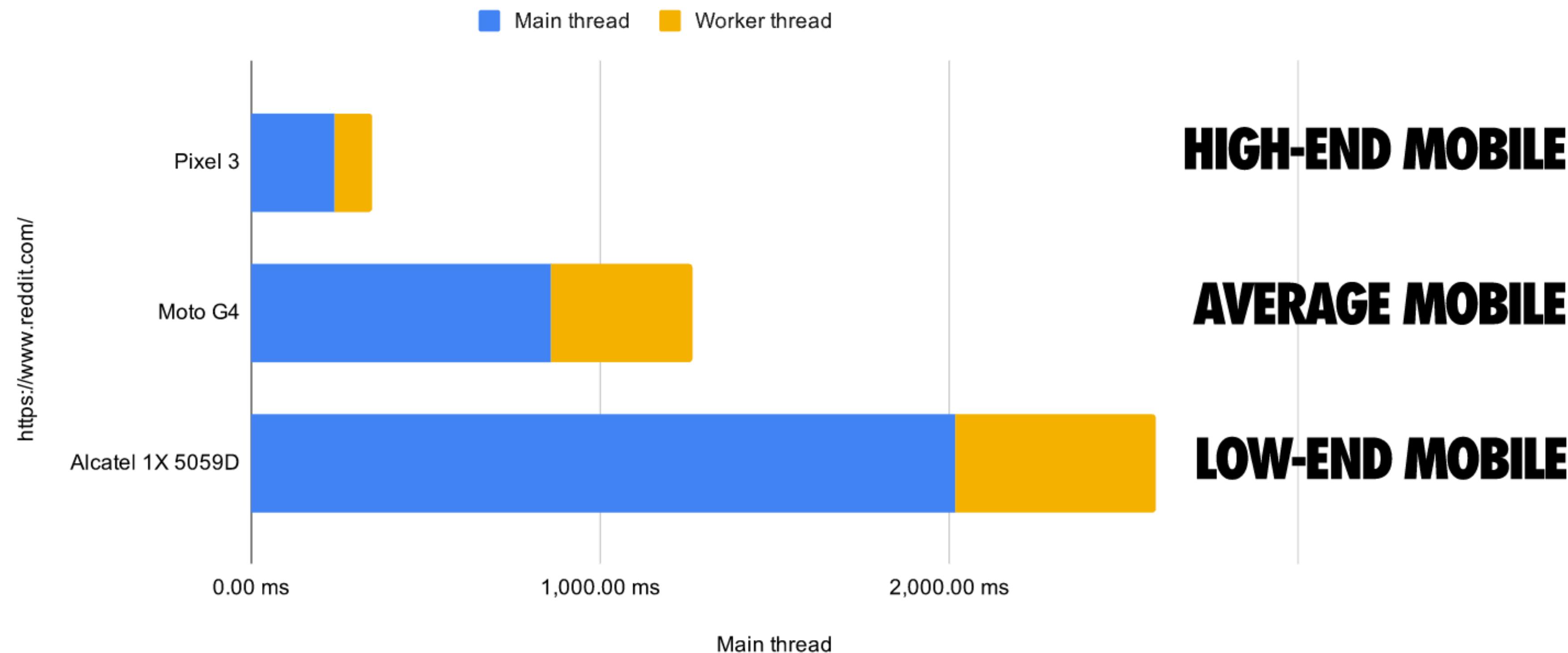
Big difference between high and low end
device

Amount of JS grows as app grows

COST OF JAVASCRIPT

REDDIT.COM

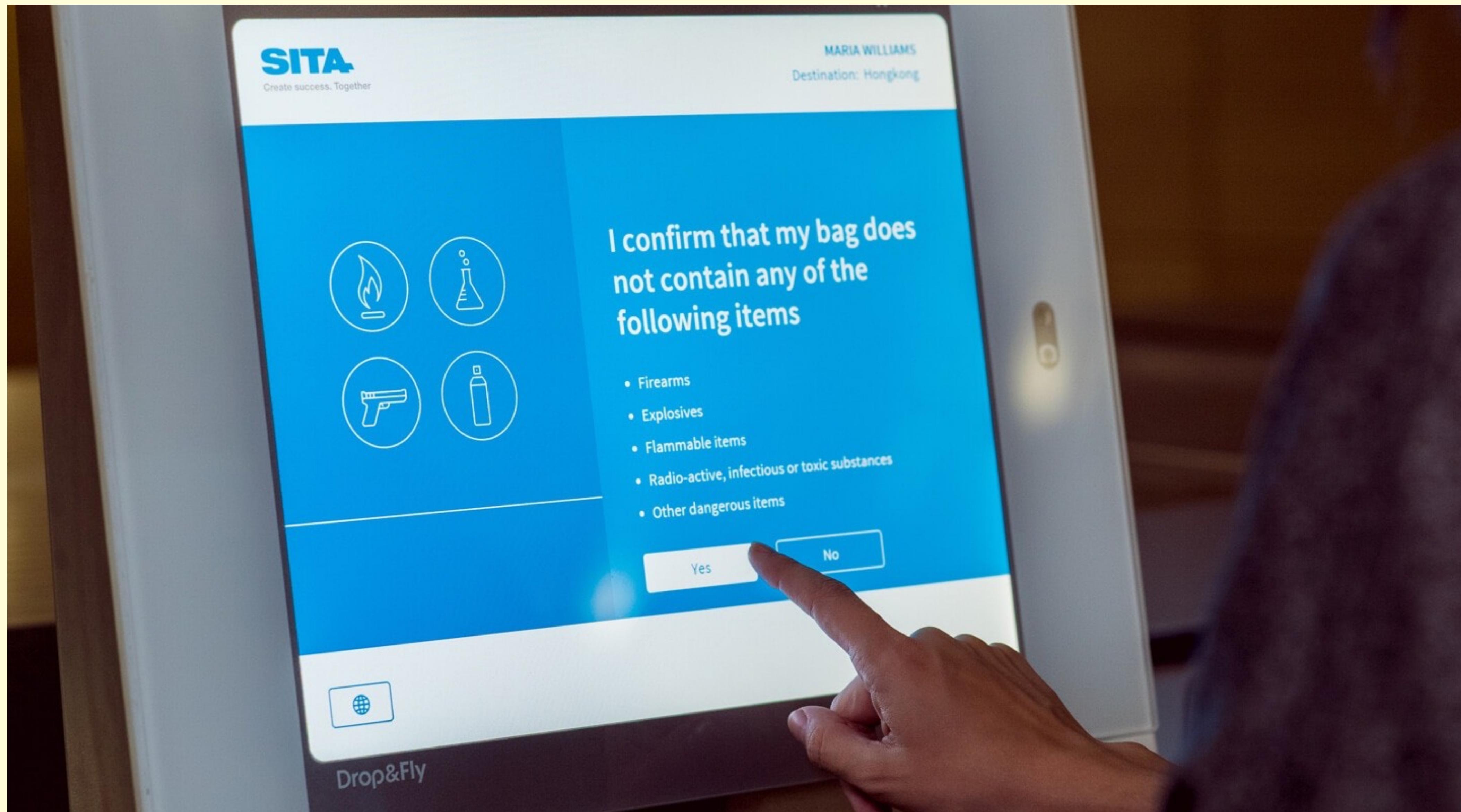
JavaScript Processing times for Reddit.com



The cost of Reddit's JavaScript across a few different device classes (low-end, average, and high-end)

[The Cost Of Javascript](#)

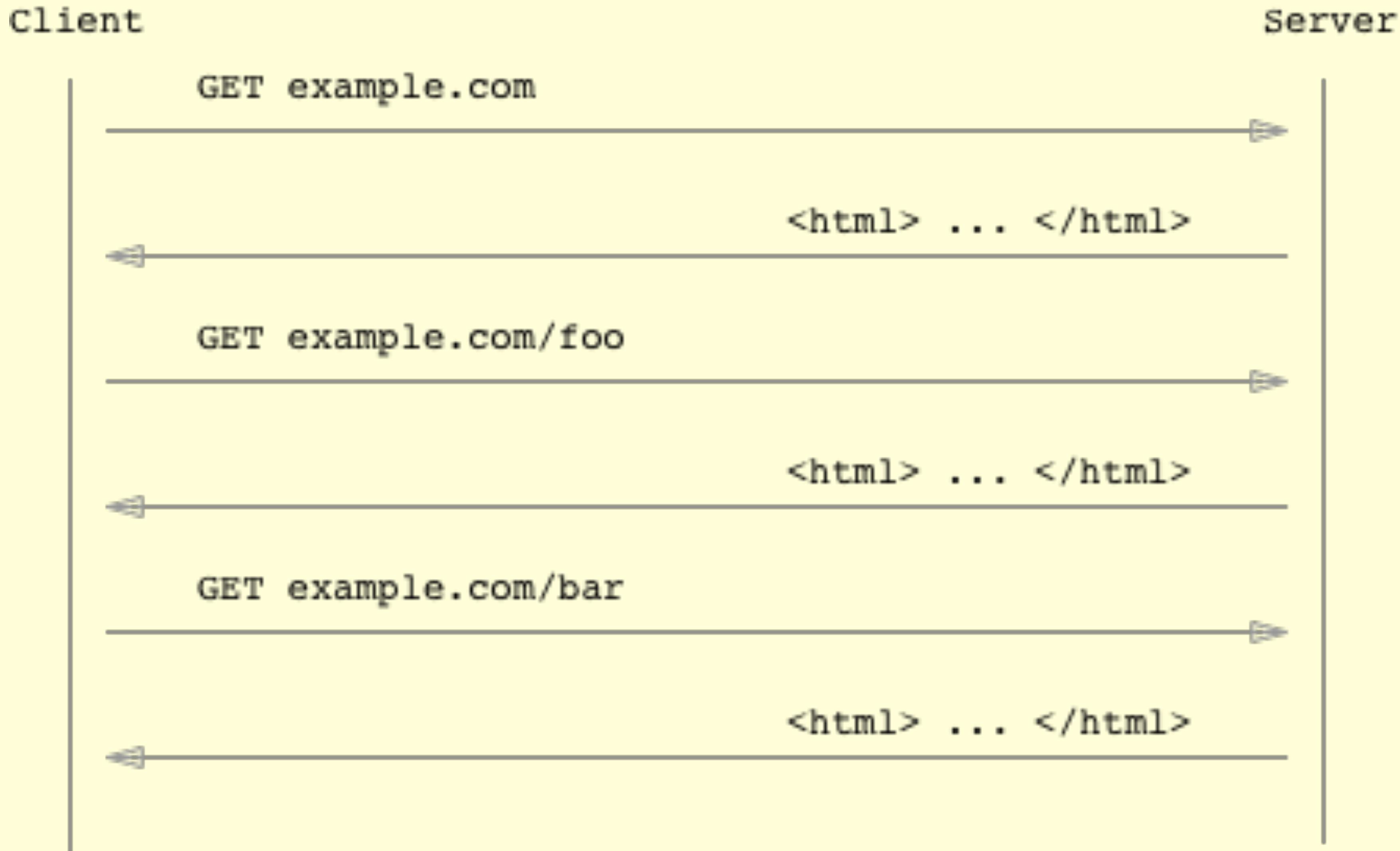
Example: Internal known environment



Server-side rendering

Server rendering generates the full HTML for a page on the server in response to navigation. This avoids additional round-trips for data fetching and templating on the client, since it's handled before the browser gets a response

Server side rendering



[Client-side rendering vs. server-side rendering](#)

Pros

Fast FP, FCP and TTI

Highly dynamic initial HTML

More reliable on the client

Progressive enhancement

Challenges

Hosting

Scaling up

Maintenance

Example: It should work for everyone

The screenshot shows the homepage of the Funda website. At the top, there's a yellow header bar with the Funda logo on the left, a menu icon, 'Inloggen' (Log in) button, and a language switcher for 'NL'. Below the header is a large, colorful illustration of a person riding a bicycle with two children in a cargo bike, set against a background of houses and trees. The text 'Eindelijk weer met je groene vingers in de aarde' (Finally back with your green fingers in the earth) is overlaid on the illustration. The main search area has tabs for 'Koop' (Buy), 'Huur' (Rent), 'Nieuwbouw' (New build), 'Recreatie' (Recreation), and 'Europa'. Below these are search fields for 'Plaats, buurt, adres, etc.' (Place, neighborhood, address, etc.), distance (+ 0 km), price (Van € 0 to Tot Geen maximum), and a 'Zoek' (Search) button. A message 'Laatste zoekopdracht: Amsterdam >' is shown below the search fields. At the bottom of the search area are links for 'Op kaart zoeken' (Search on map) and 'Zoek een NVM makelaar' (Search a NVM broker). The main content area features two images under the heading 'Woning in beeld' (House in image): a traditional Dutch house with a thatched roof and a modern brick house with a balcony. To the right, there's a section titled 'Heb je een vraag?' (Do you have a question?) with the text 'Geen stress. Wij helpen je met het koop- en verkoopproces.' (No stress. We help you with the buying and selling process.) and an illustration of a house with a 'TE KOOP' (For sale) sign.

Eindelijk weer met je groene vingers in de aarde

Koop Huur Nieuwbouw Recreatie Europa

Plaats, buurt, adres, etc. + 0 km Van € 0 Tot Geen maximum Zoek

Laatste zoekopdracht: Amsterdam >

Woning in beeld

Heb je een vraag?

Geen stress.
Wij helpen je met
het koop- en
verkoopproces.

TE KOOP

Prerendering

static rendering means producing a separate HTML file
for each URL ahead of time.

Prerender HTML in build

Serve static HTML

Possibly hydrate (take over) with

Javascript

Pros

Fast FP, FCP and TTI

Deploy static HTML files to CDN

Simple to host / little maintenance

Challenges

Prerendering of highly dynamic pages

Limited functionality

Enhancements need js

Build process

Example: Static prerendering

De Voorhoede

Services Cases Blog Vacatures Over Ons Contact EN | NL

Wij zijn
Front-end developers

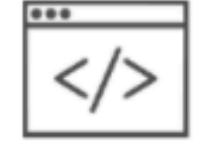
Hoe kunnen we helpen?

SCROLL ↓

The image shows a website layout with a yellow header containing the logo 'De Voorhoede' and navigation links. The main content area has a yellow background. It features text about being front-end developers and a large, bold question 'Hoe kunnen we helpen?'. At the bottom, there's a scroll-down arrow icon.

Keeping static pages up to date

Use web hooks (e.g. save from CMS, push to main branch, daily CRON job) to trigger new build.

Server		Browser			
					
	Server Rendering	“Static SSR”	SSR with (Re)hydration	CSR with Prerendering	Full CSR
Overview:	An application where input is navigation requests and the output is HTML in response to them.	Built as a Single Page App, but all pages prerendered to static HTML as a build step, and the JS is removed .	Built as a Single Page App. The server prerenders pages, but the full app is also booted on the client.	A Single Page App, where the initial shell/skeleton is prerendered to static HTML at build time.	A Single Page App. All logic, rendering and booting is done on the client. HTML is essentially just script & style tags.
Authoring:	Entirely server-side <small>(request-response, HTML)</small>	Built as if client-side <small>(components, DOM*, fetch)</small>	Built as client-side	Client-side	Client-side
Rendering:	Dynamic HTML	Static HTML	Dynamic HTML and JS/DOM	Partial static HTML, then JS/DOM	Entirely JS/DOM
Server role:	Controls all aspects. <small>(thin client)</small>	Delivers static HTML	Renders pages <small>(navigation requests)</small>	Delivers static HTML	Delivers static HTML
Pros:	<ul style="list-style-type: none"> 👍 TTI = FCP 👍 Fully streaming 	<ul style="list-style-type: none"> 👍 Fast TTFB 👍 TTI = FCP 👍 Fully streaming 	<ul style="list-style-type: none"> 👍 Flexible 	<ul style="list-style-type: none"> 👍 Flexible 👍 Fast TTFB 	<ul style="list-style-type: none"> 👍 Flexible 👍 Fast TTFB
Cons:	<ul style="list-style-type: none"> 👎 Slow TTFB 👎 Inflexible 	<ul style="list-style-type: none"> 👎 Inflexible 👎 Leads to hydration 	<ul style="list-style-type: none"> 👎 Slow TTFB 👎 TTI >> FCP 👎 Usually buffered 	<ul style="list-style-type: none"> 👎 TTI > FCP 👎 Limited streaming 	<ul style="list-style-type: none"> 👎 TTI >> FCP 👎 No streaming
Scales via:	Infra size / cost	build/deploy size	Infra size & JS size	JS size	JS size
Examples:	Gmail HTML, Hacker News	Docusaurus, Netflix*	Next.js , Razzle , etc	Gatsby, Vuepress, etc	Most apps

Example: Isomorphic rendering

The screenshot shows a movie streaming interface. At the top, there is a navigation bar with links for 'Home' and 'Genres'. On the right side of the top bar, there is a user account section with an email address 'declan@vo...' and a search icon. The main content area features a large banner for the movie 'Terminator: Dark Fate'. The banner image shows Arnold Schwarzenegger's character, the Terminator, and Linda Hamilton's character, Sarah Connor, in a dramatic, action-oriented pose. Text on the banner includes 'Arnold Schwarzenegger is back' and a promotional tagline: 'Dark Fate gaat verder waar Terminator 2: Judgment Day, de klassieker uit 1991, ophield.' Below the banner is a yellow button with the text '▶ Kijk vanaf €4,99'. To the left of the banner, there is a section titled 'Mijn films' (My movies) featuring a thumbnail for 'Bohemian Rhapsody'. To the right of the banner, there is a section titled 'Top 20' featuring thumbnails for several movies: 'Terminator: Dark Fate', 'Midway', 'Maleficent 2', 'Joker', 'Fuck de Liefde', and 'Gemini Man'. The overall design has a dark background with bright, colorful elements from the movie posters.

It's hard!

Home Genres  Inloggen 

Arnold Schwarzenegger is back
Terminator: Dark Fate

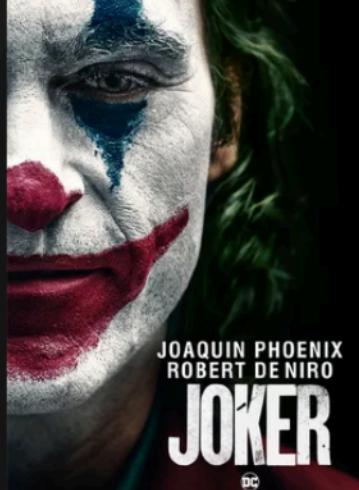
 Kijk vanaf €4,99

Top 20 >


Terminator: Dark Fate


Midway


Maleficent 2


Joker


Fuck de L...

Nieuw >


LUCE

Elements Console Sources Network Performance **Audits** » ⚠ 2 | ⋮ X

+ 15:37:27 - www.pathe-thuis.nl ▾

Clear storage Simulated throttling [Learn more](#)

https://www.pathe-thuis.nl/

68

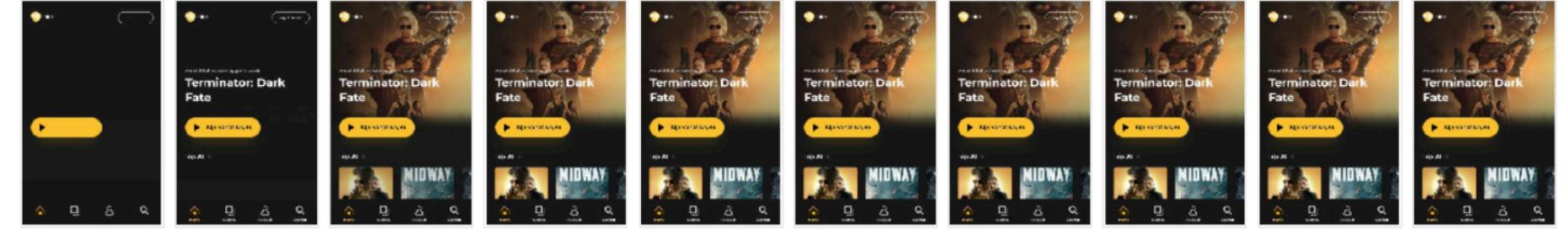
Performance

Metrics 

 First Contentful Paint	1.2 s	 First Meaningful Paint	1.2 s
 Speed Index	3.4 s	 First CPU Idle	7.4 s
 Time to Interactive	8.9 s	 Max Potential First Input Delay	330 ms

[View Trace](#)

Values are estimated and may vary. The performance score is based only on these metrics.



Opportunities — These suggestions can help your page load faster. They don't directly affect the Performance score.

Opportunity	Estimated Savings
 Properly size images	 0.9 s
 Defer offscreen images	 0.54 s
 Preconnect to required origins	 0.3 s
 Efficiently encode images	 0.15 s

Build tooling

Automate repetitive tasks and optimise
your workflow when deploying your app.

Bundling and minifying your Javascript

Using a CSS preprocessor

Generate images from 1 source

Prerendering static HTML files



PARCEL

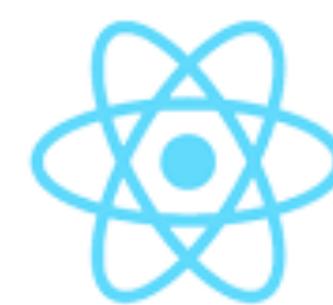
Blazing fast, zero configuration web application bundler



rollup.js



 Vue.js



React



GRUNT



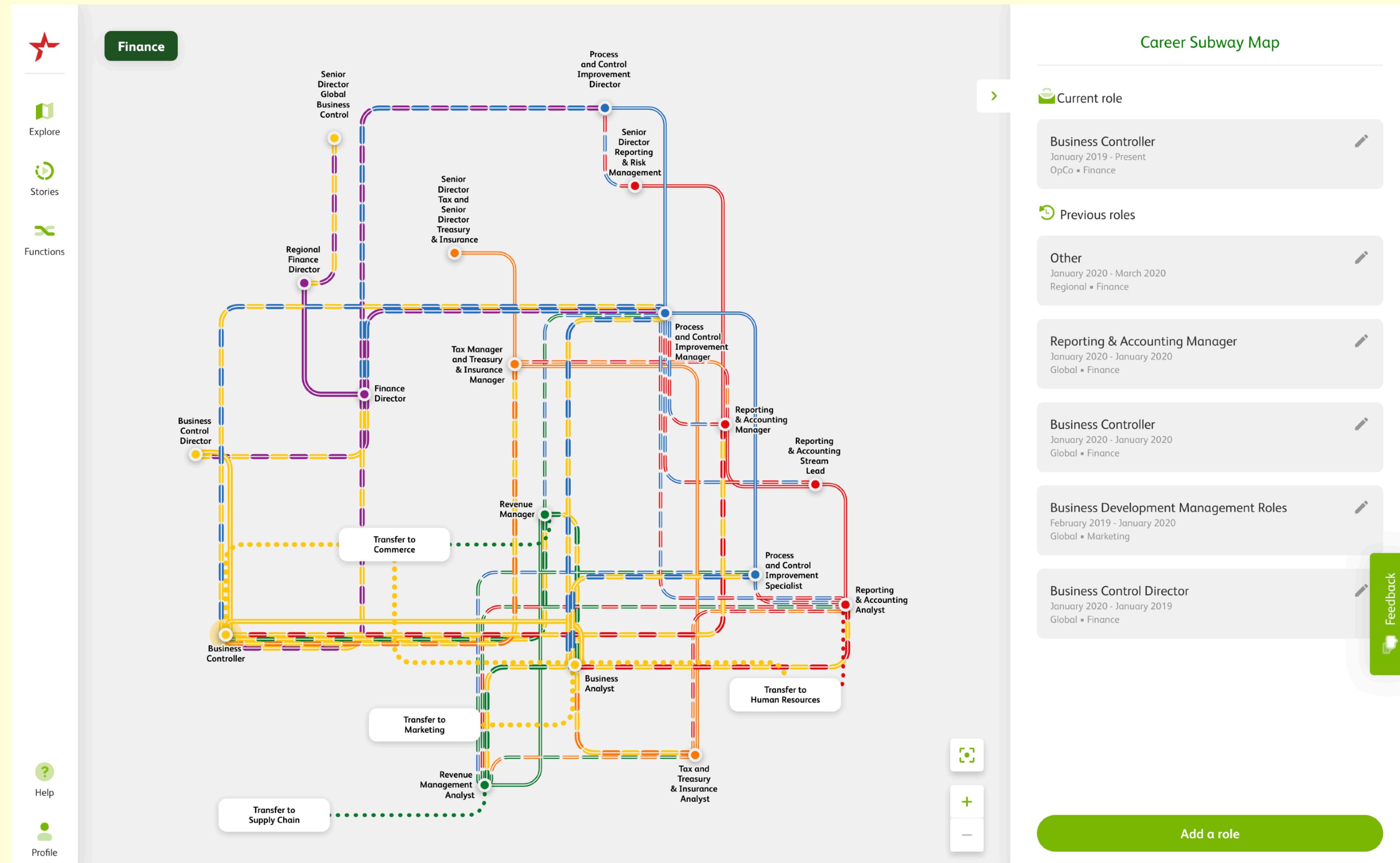
webpack

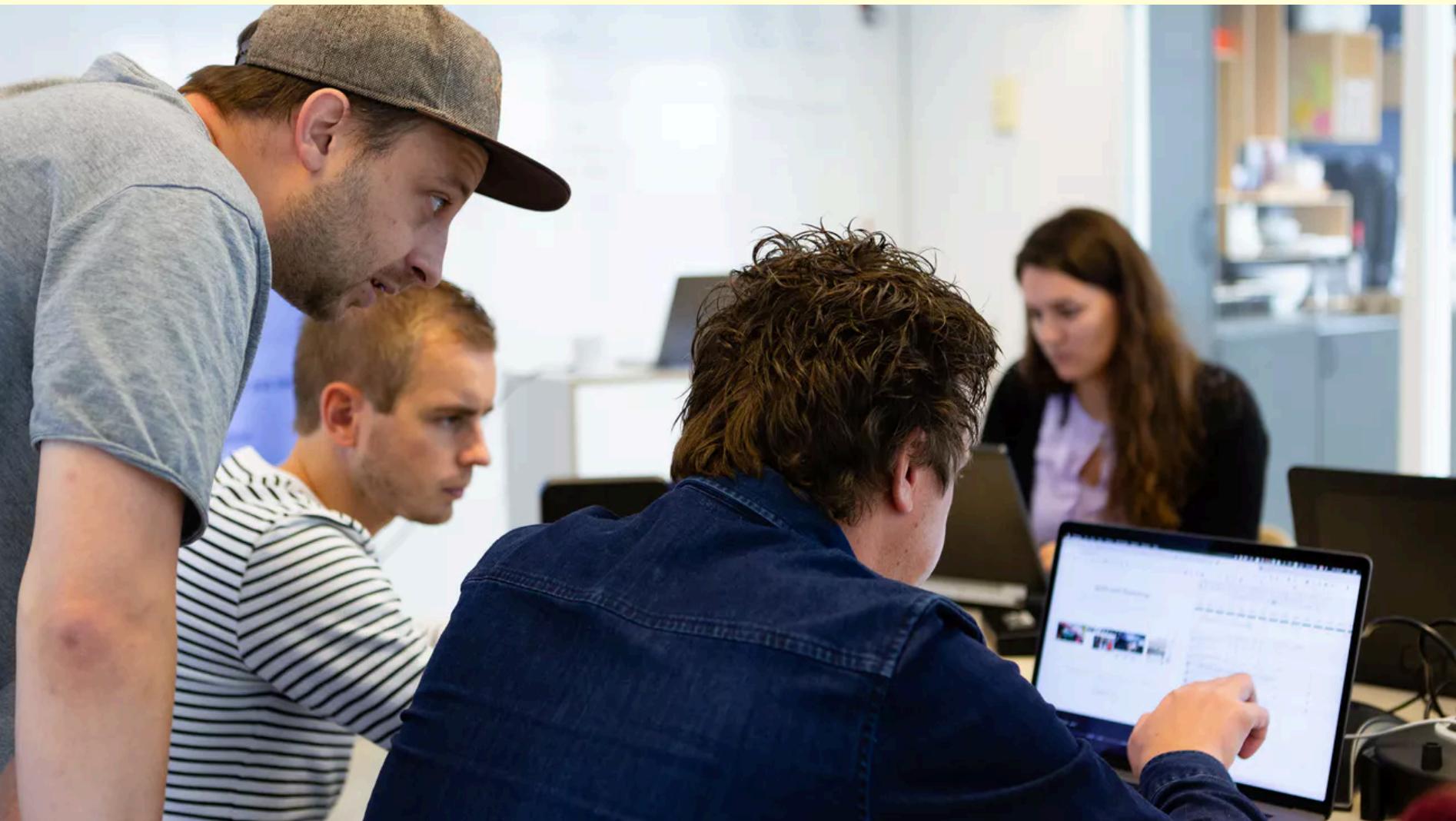


```
{  
  "name": "heineken-career-map",  
  "description": "Explore opportunities and learn what routes others took before you.",  
  "scripts": {  
    "predev": "npm run data",  
    "dev": "npm run watch",  
  
    "prebuild": "npm run data",  
    "build": "vue-cli-service build",  
  
    "predata": "rimraf 'public/data' 'src/static/data'",  
    "data": "run-s data:dump data:build",  
    "data:dump": "node tasks/dump-data.js",  
    "data:build": "node tasks/build-data.js",  
  
    "watch": "run-p watch:*",  
    "watch:client": "vue-cli-service serve",  
    "watch:build": "chokidar 'tasks/*.js' 'lib/*.js' --command 'npm run data:build'",  
    "watch:data": "chokidar 'tasks/get-heineken-functions.js' --command 'npm run data'"  
  },  
  "dependencies": {  
    ...  
  },  
  "devDependencies": {  

```

An example





thanks }
Declan Rek
// @DeclanRek
// voorhoede.nl