THE COST OF JAVASCRIPT

@addyosmani

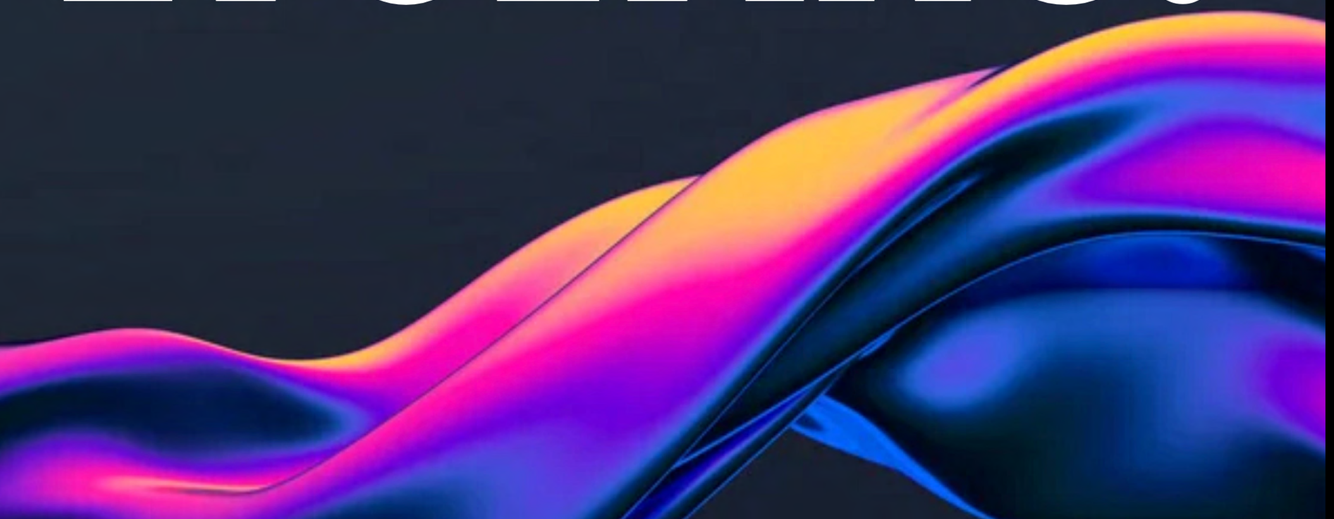
JAVASCRIPT IS GOOD...IN MODERATION

@reactive_dude

IS 0KB OF JAVASCRIPT IN YOUR FUTURE?

bit.ly/0kbjsryan

HOW WE BUILD IS EVOLVING.



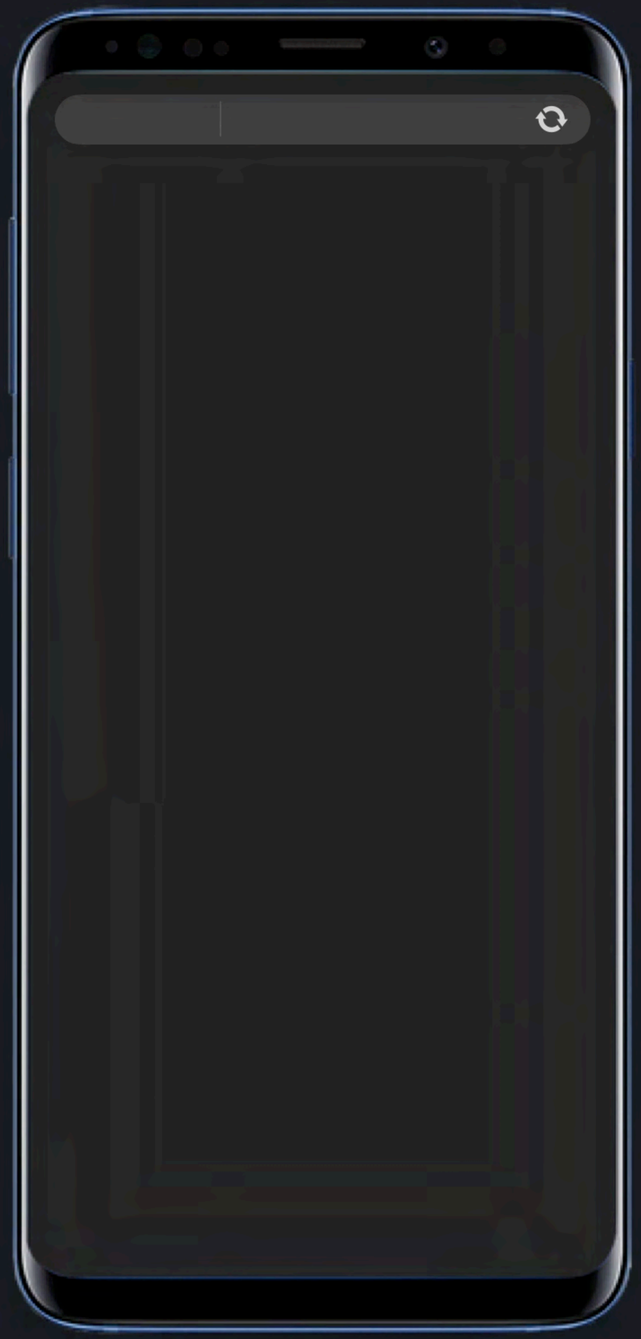
MPA, SPA, MPA++
SSG, SSR, EDGE RENDER
STREAMING RENDERING
HYDRATION
PARTIAL HYDRATION
SERVER COMPONENTS
ISLANDS ARCHITECTURE
RESUMABILITY (QWIK)


PATTERNS!
EVERYWHERE
ALL AT ONCE

Patterns for web applications

	Portfolio	Content	Storefront	Social Network	Immersive
Holotype	Personal Blog	CNN	Amazon	Facebook	Figma
Interactivity	Minimal	Linked Articles	Purchase	Multi-Point, Real-time	Everything
Session Depth	Shallow	Shallow	Shallow - Medium	Extended	Deep
Values	Simplicity	Discover-ability	Load Performance	Dynamicism	Immersiveness
Routing	Server	Server, Hybrid	Hybrid, Transitional	Transitional, Client	Client
Rendering	Static	Static, SSR	Static, SSR	SSR	CSR
Hydration	None	Progressive, Partial	Partial, Resumable	Any	None (CSR)
Example Framework	11ty	Astro, Elder	Marko, Qwik, Hydrogen	Next, Remix	Create React App

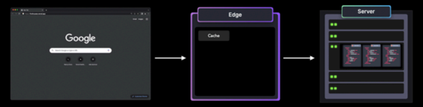
https://dev.to/this-is-learning/patterns-for-building-javascript-websites-in-2022-5a93



Server

Step 1

Client requests HTML from server

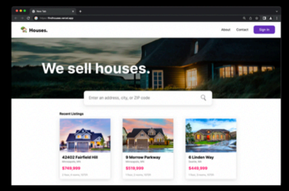


Step 2

Server returns HTML, Edge caches response

Step 3

Browser parses and renders content

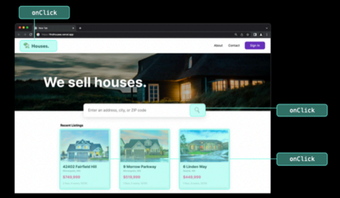


Step 4

Client requests JS bundle from server

Step 5

Browser hydrates elements



TTFB

LCP

FCP

TTI

Network

GET /

GET /bundle.js

Main Thread

Parse HTML

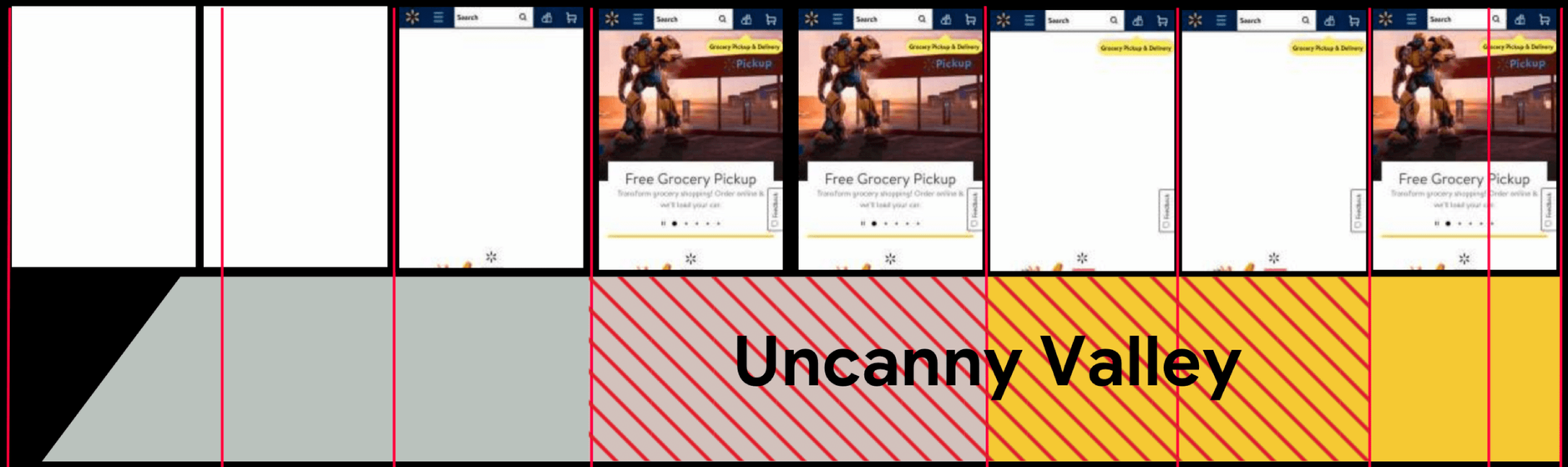
Layout

Paint

Layout

Paint

IS THE PAGE 'READY'?



Initial request

Server returns HTML

Populate stores

Render components

Dehydrate stores

HTML arrives

view painted

JS arrives

JS processed

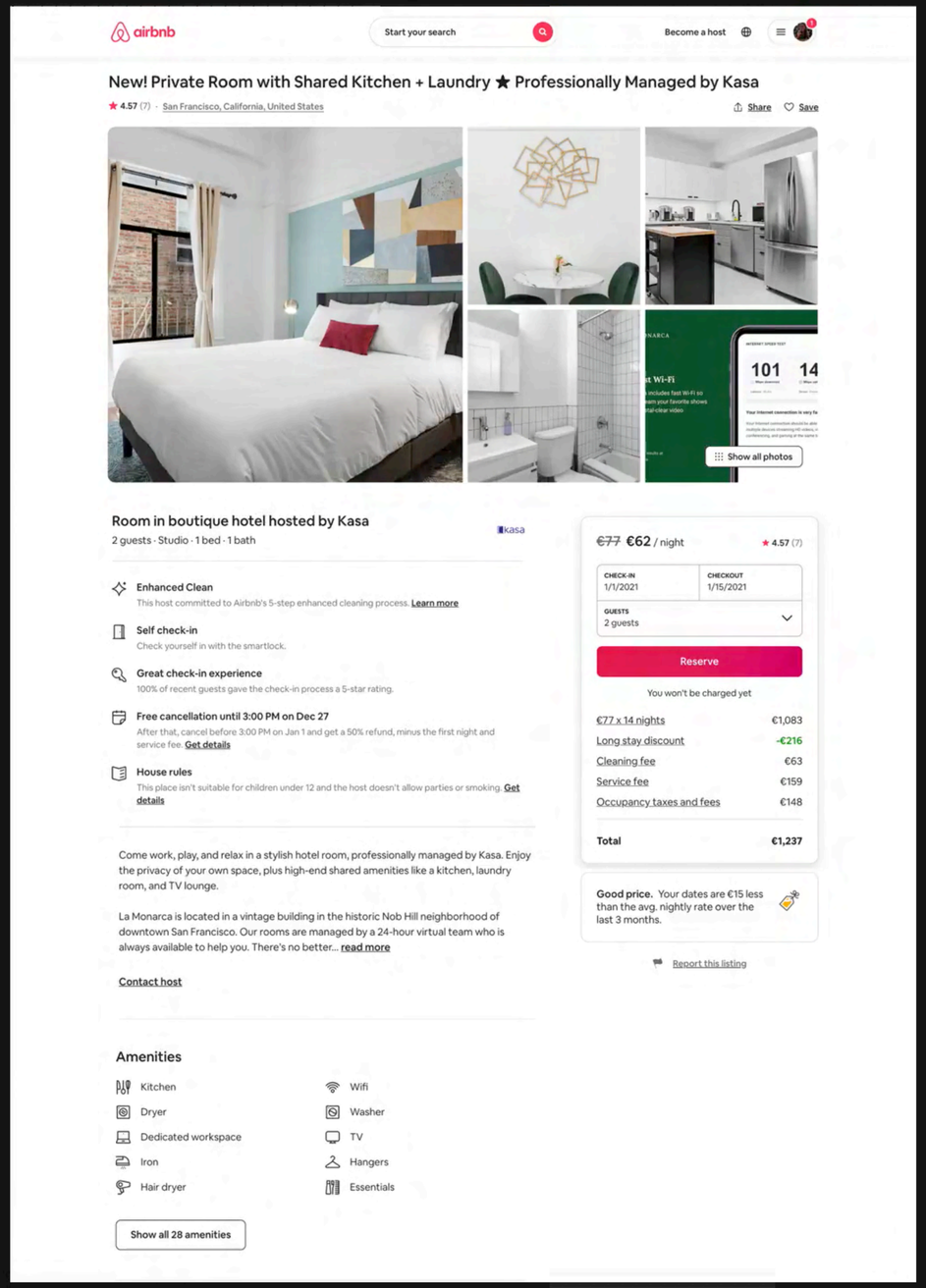
Hydrate stores

REHYDRATION

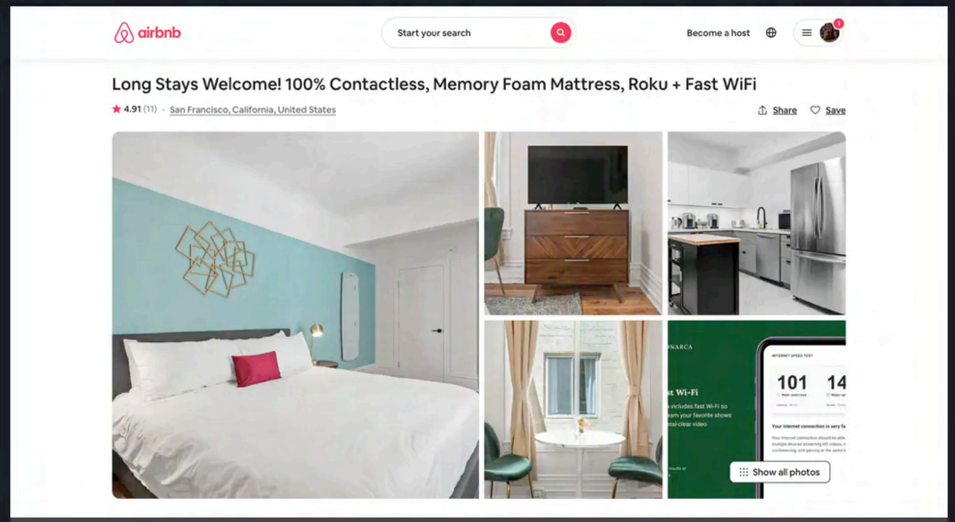
Rehydrate DOM & attach event handlers

UI is interactive

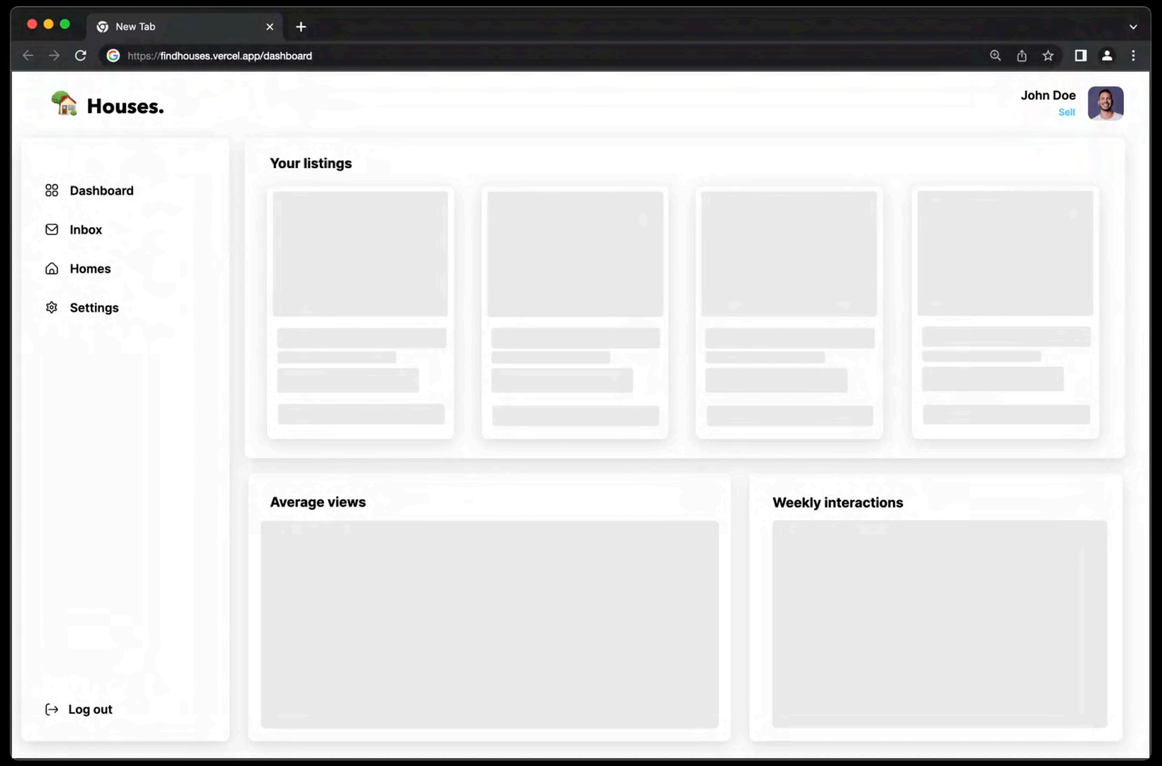
PROGRESSIVE HYDRATION

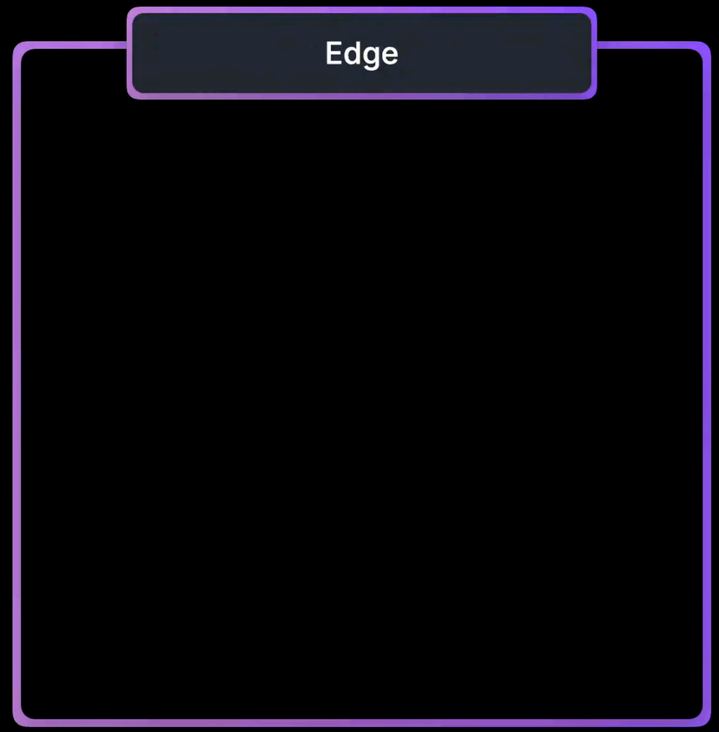


FACTOR IN VISIBILITY



Edge Streaming Server-Side Rendering





CLEANING UP YOUR JAVASCRIPT 6 MONTHS IN

IF JAVASCRIPT DOESN'T BRING USERS JOY, THANK IT, AND THROW IT AWAY

- MARIE KONDO

SMALL JAVASCRIPT BUNDLES

IMPROVE DOWNLOAD SPEEDS, LOWER MEMORY USAGE & REDUCE CPU COSTS

POST-DOWNLOAD,
EXECUTING JAVASCRIPT IS
THE DOMINANT COST.
KEEP IT MINIMAL.

OPTIMIZE FOR NETWORK

OPTIMIZE FOR HARDWARE

OPTIMIZE FOR THE HARDWARE

Hardware (processing power) bounds computationally intensive tasks.

JAVASCRIPT IS CPU BOUND

LET'S TALK ABOUT PHONES

MOBILE IS A SPECTRUM

$30

LOW-END

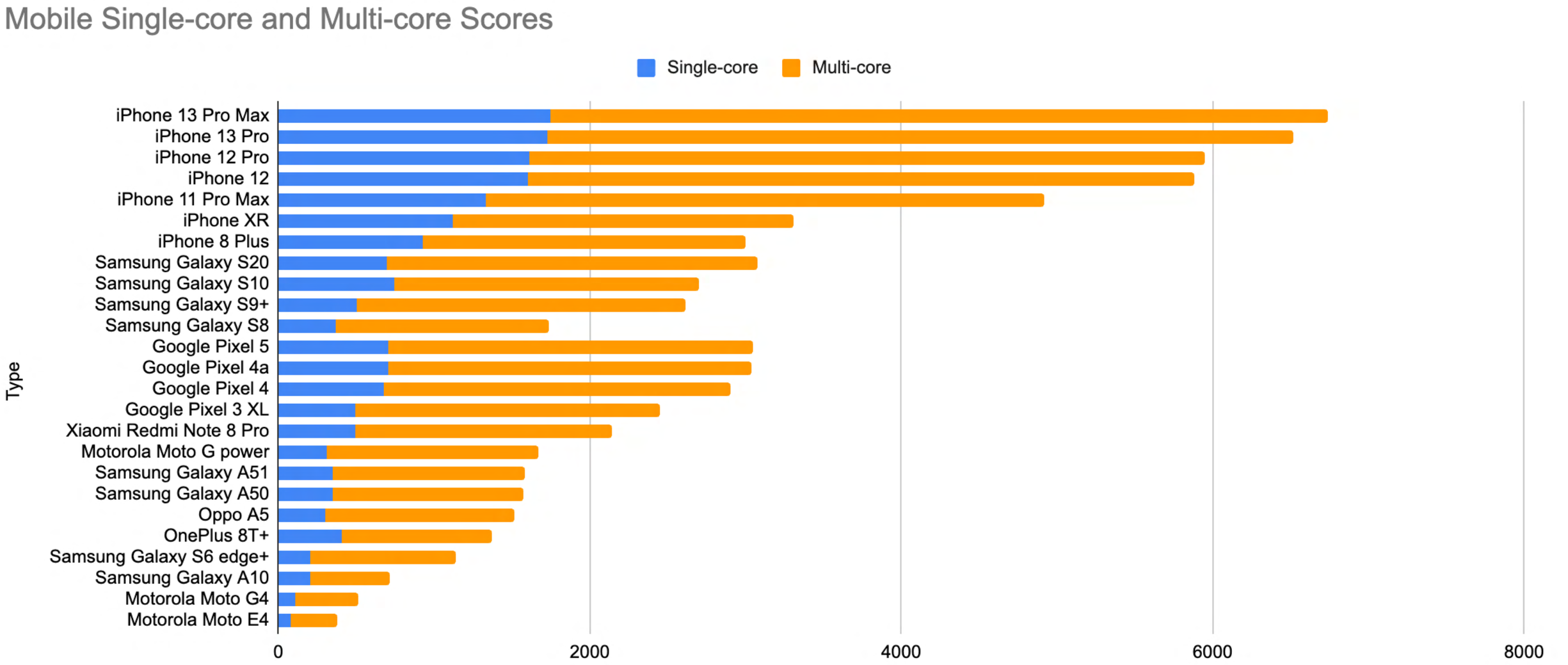
< $200

MEDIAN

$1K

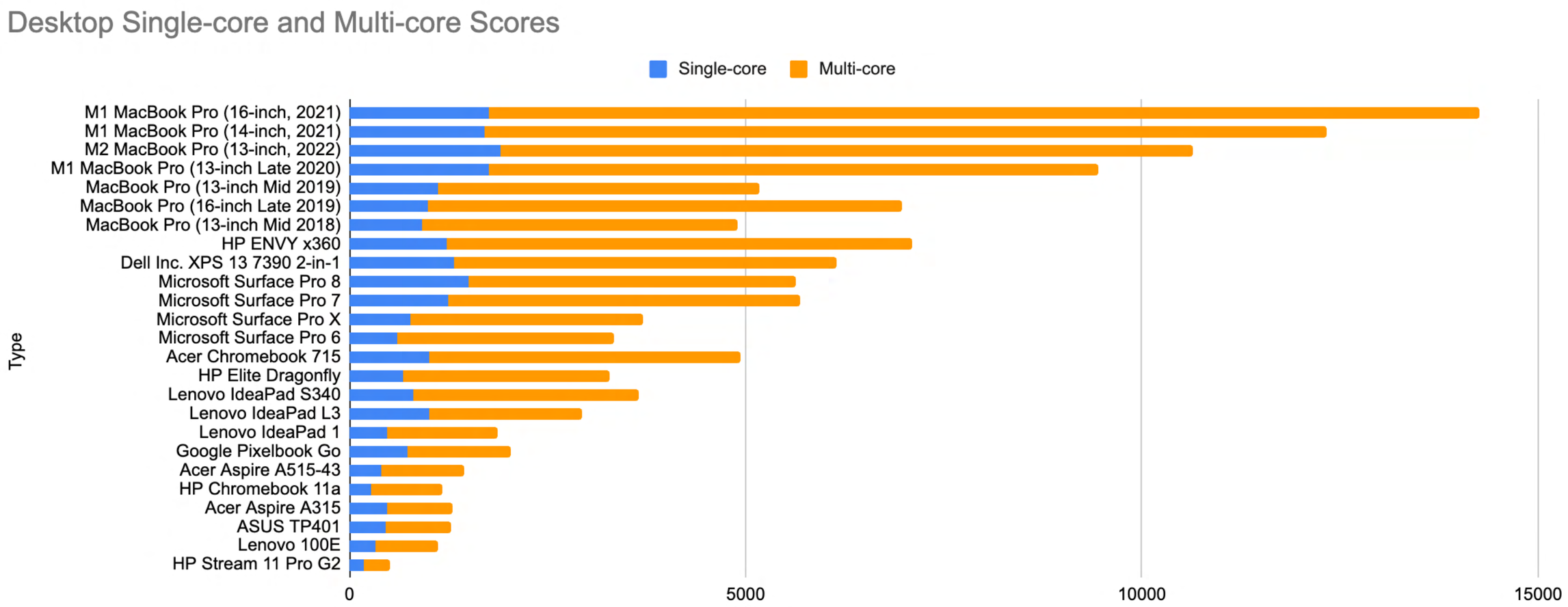
HIGH-END

HIGH VS. LOW-END PHONES

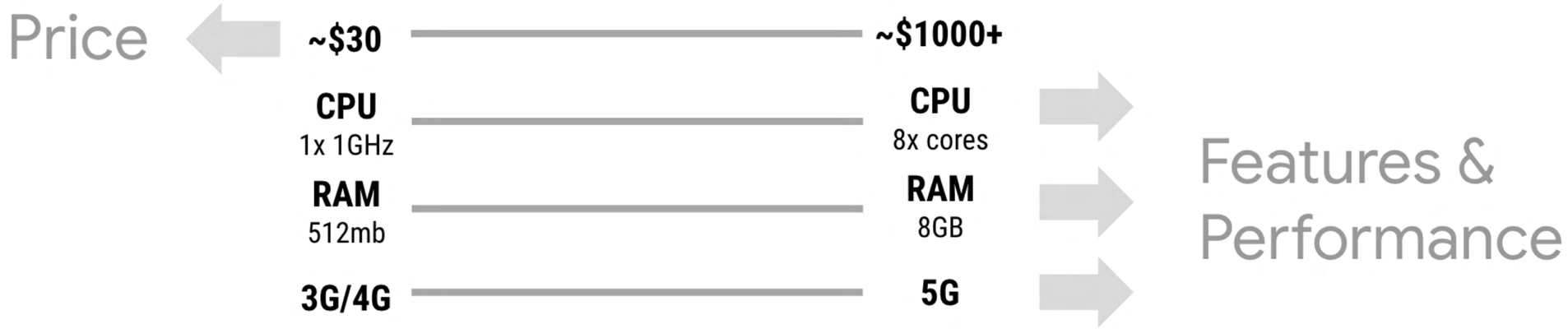


@addyosmani

HIGH VS. LOW-END LAPTOPS



THE PERFORMANCE INEQUALITY GAP CONTINUES TO GROW

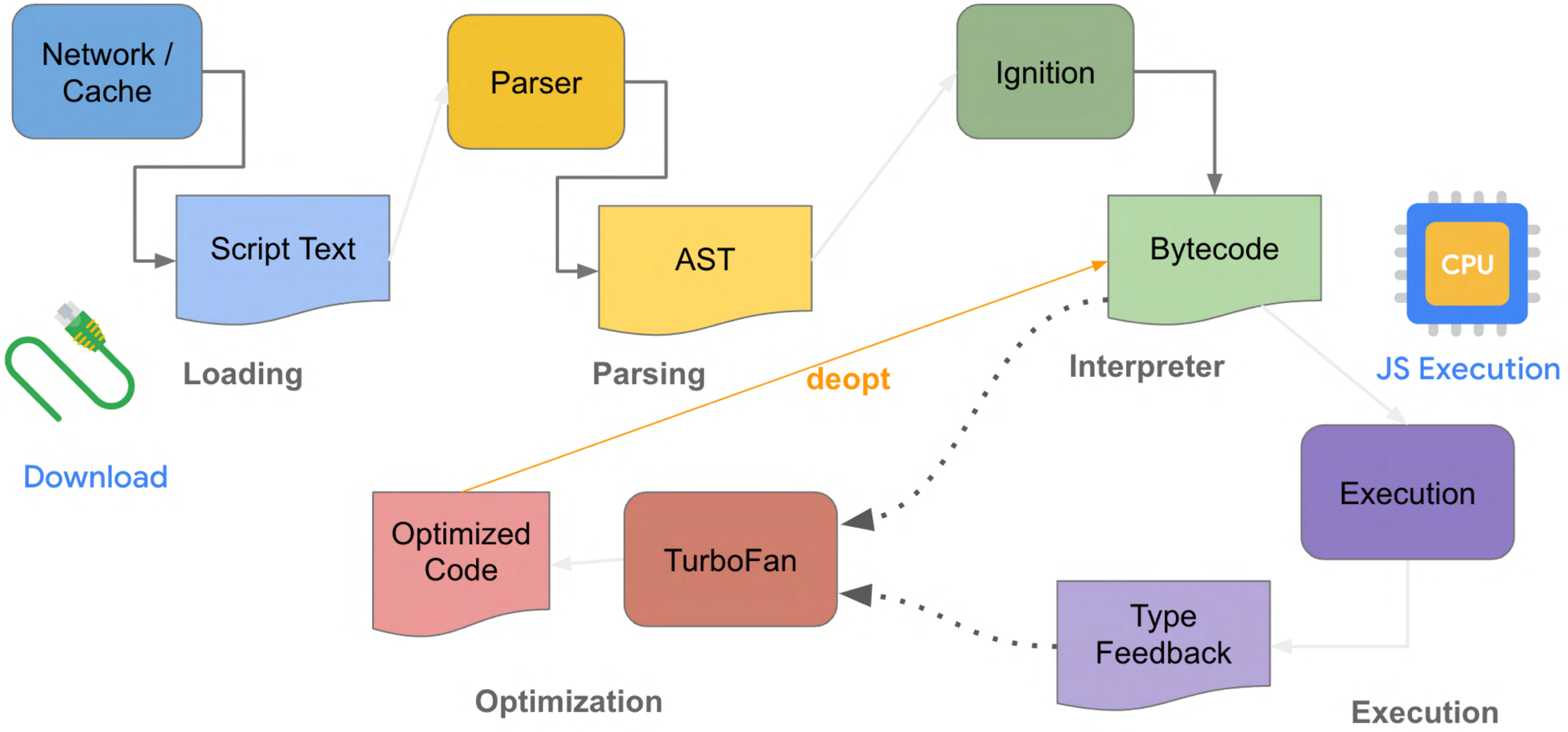


HOW BROWSERS WORK

CSS
JS
IMG
FONT

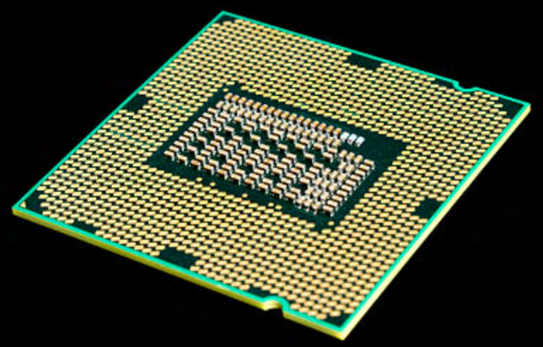
THE MOST EXPENSIVE PART OF YOUR SITE.

LIFE OF A SCRIPT



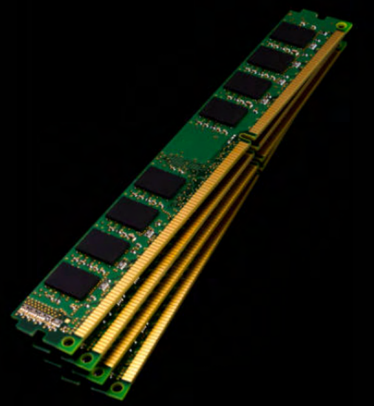
HOW DOES HARDWARE IMPACT PERFORMANCE?

CPU



GPU

RAM



HARDWARE BOUNDS COMPUTATIONALLY INTENSIVE TASKS

Low Computation

High Computation

Low Computation

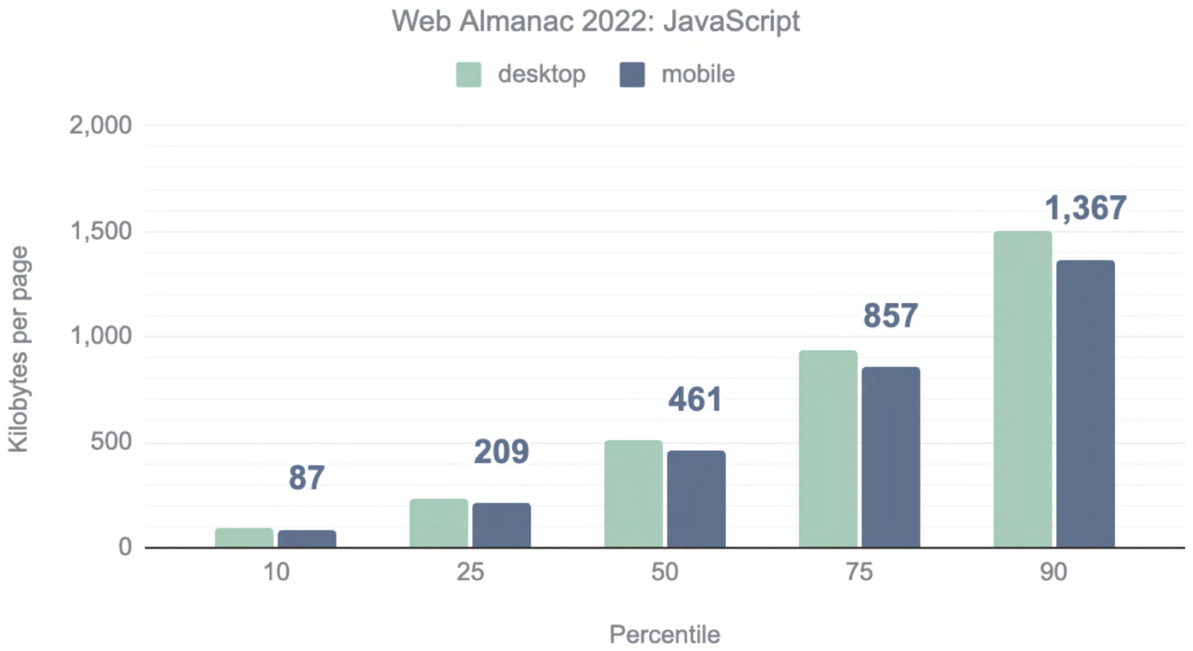
```
function addOne(n) {
    return n+1;
}
```

High Computation

```
// recursive fibonacci

function fib(n) {
    if (n <= 1) return 1;
    return fib(n - 1) + fib(n - 2);
}
```

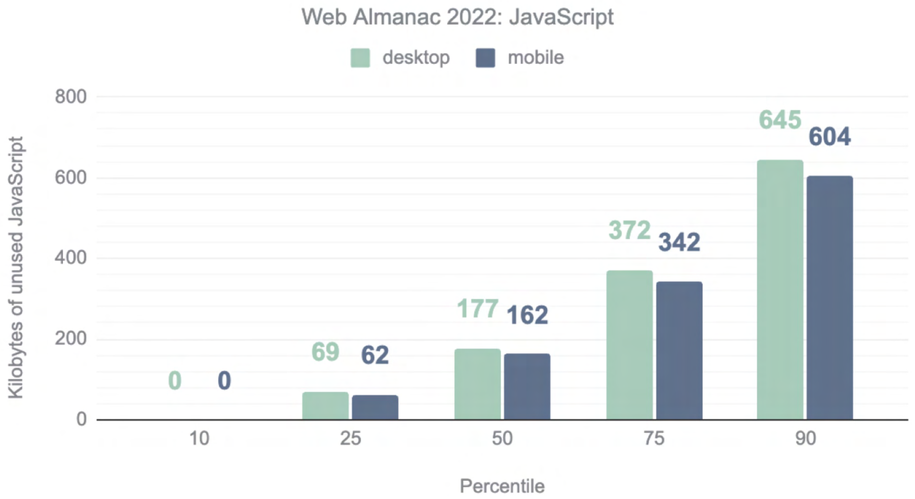
JS BYTES



1.3MB P90

MOBILE

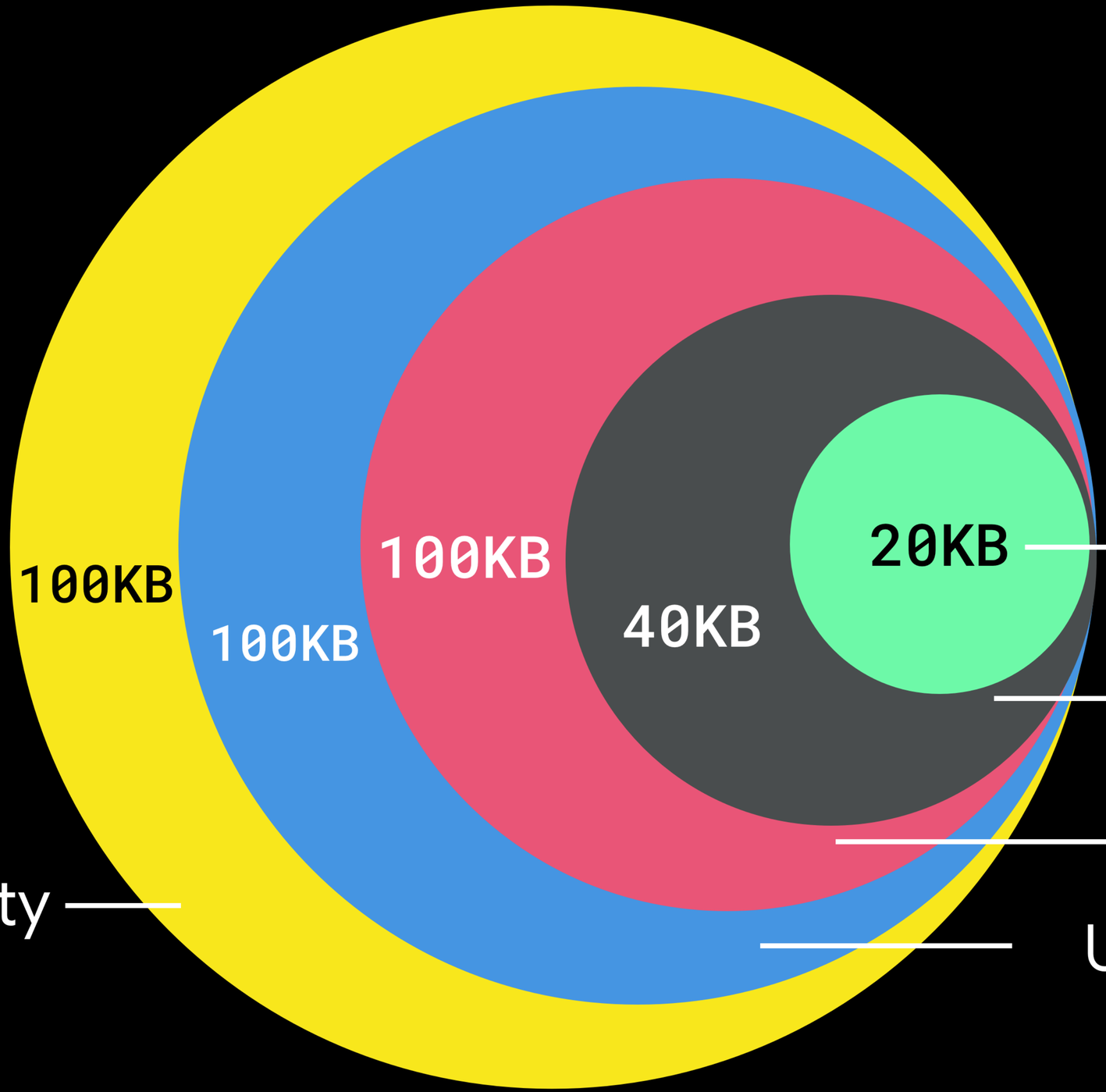
UNUSED JS



604KB P90

MOBILE

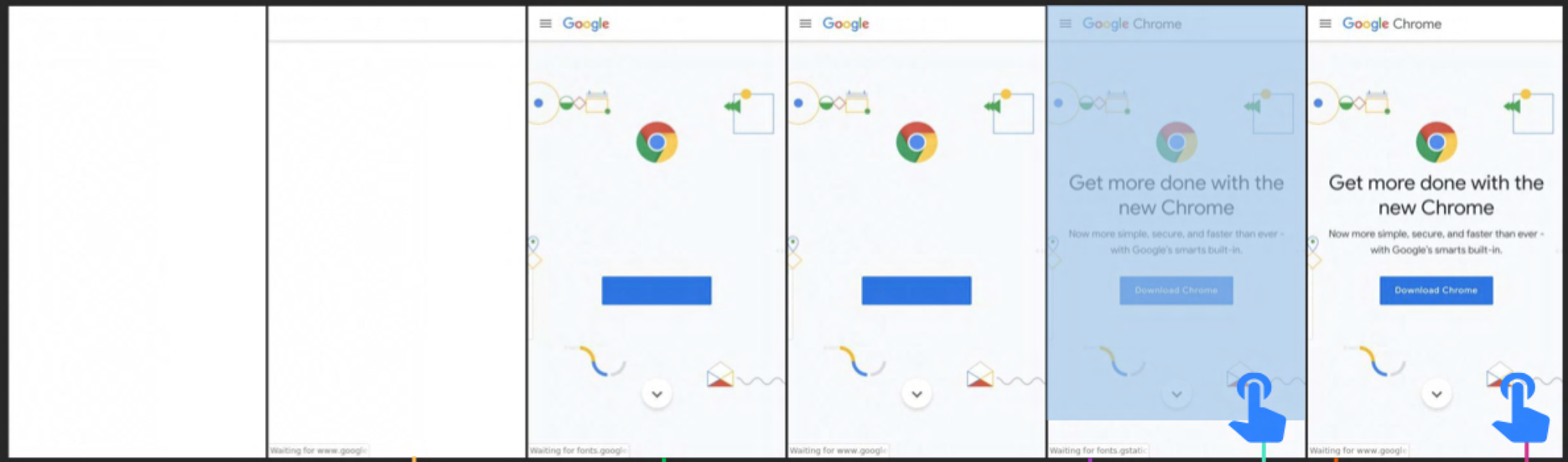
THE JS TAX


100KB
100KB
100KB
40KB
20KB
App Logic
Framework
Utilities
UI Library
Third-party

PAGE LOAD METRICS

When did the user think they could interact?

When could they interact?



First Byte

First Paint

First Contentful Paint

Largest Contentful Paint

First Input Delay

Speed Index

Time to Interactive

INP

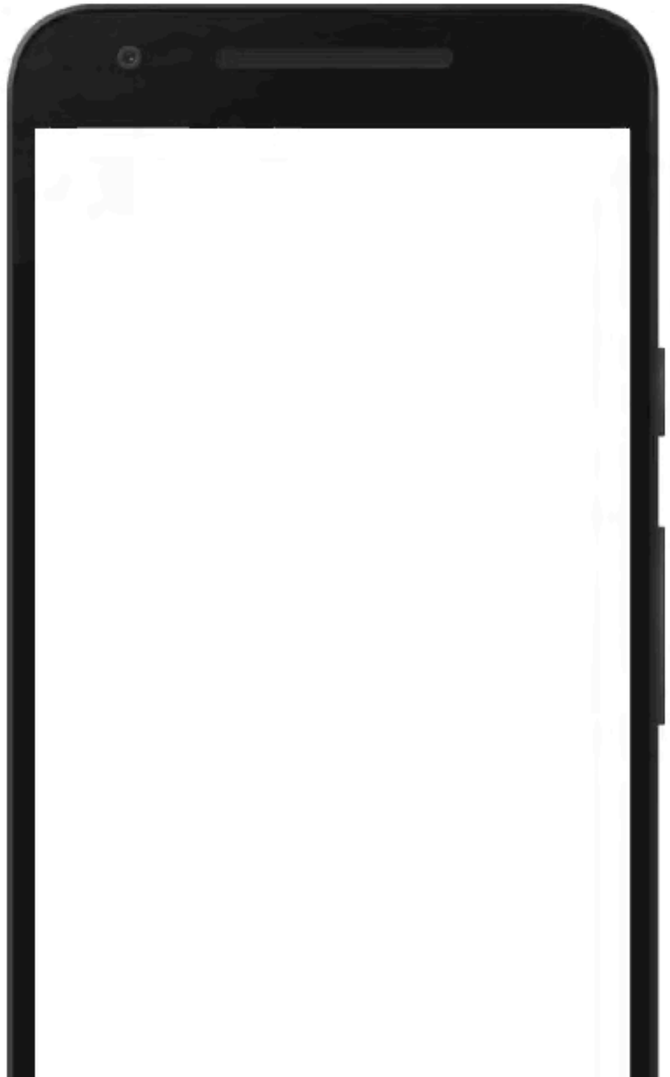
Is it happening?

Is it useful?

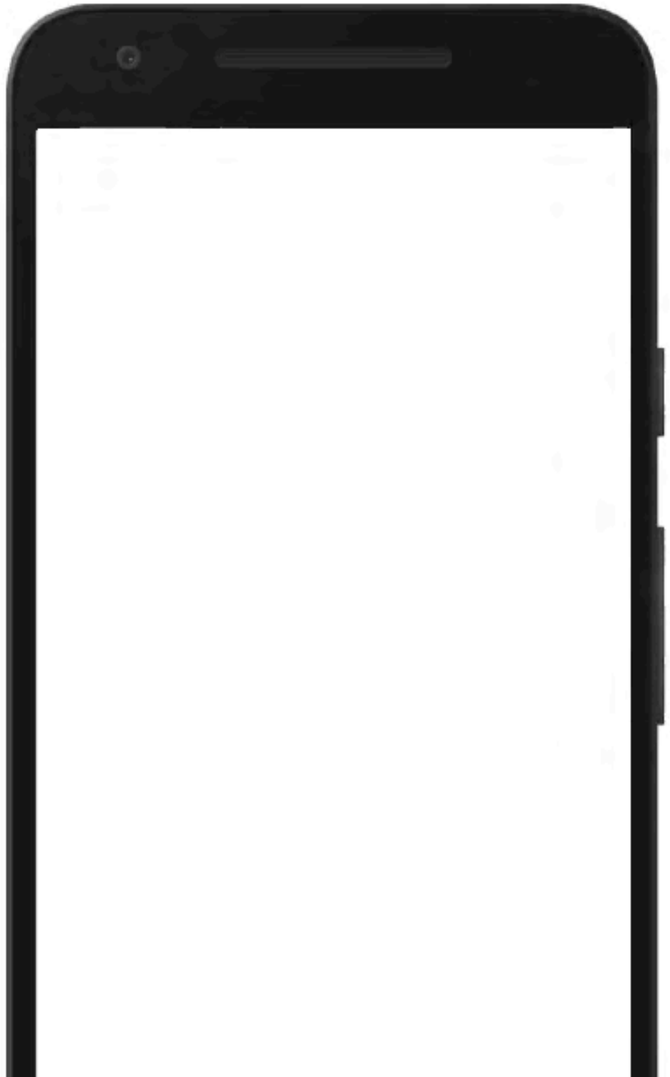
Is it usable?

TIME TO INTERACTIVE

0s 00



0s 00

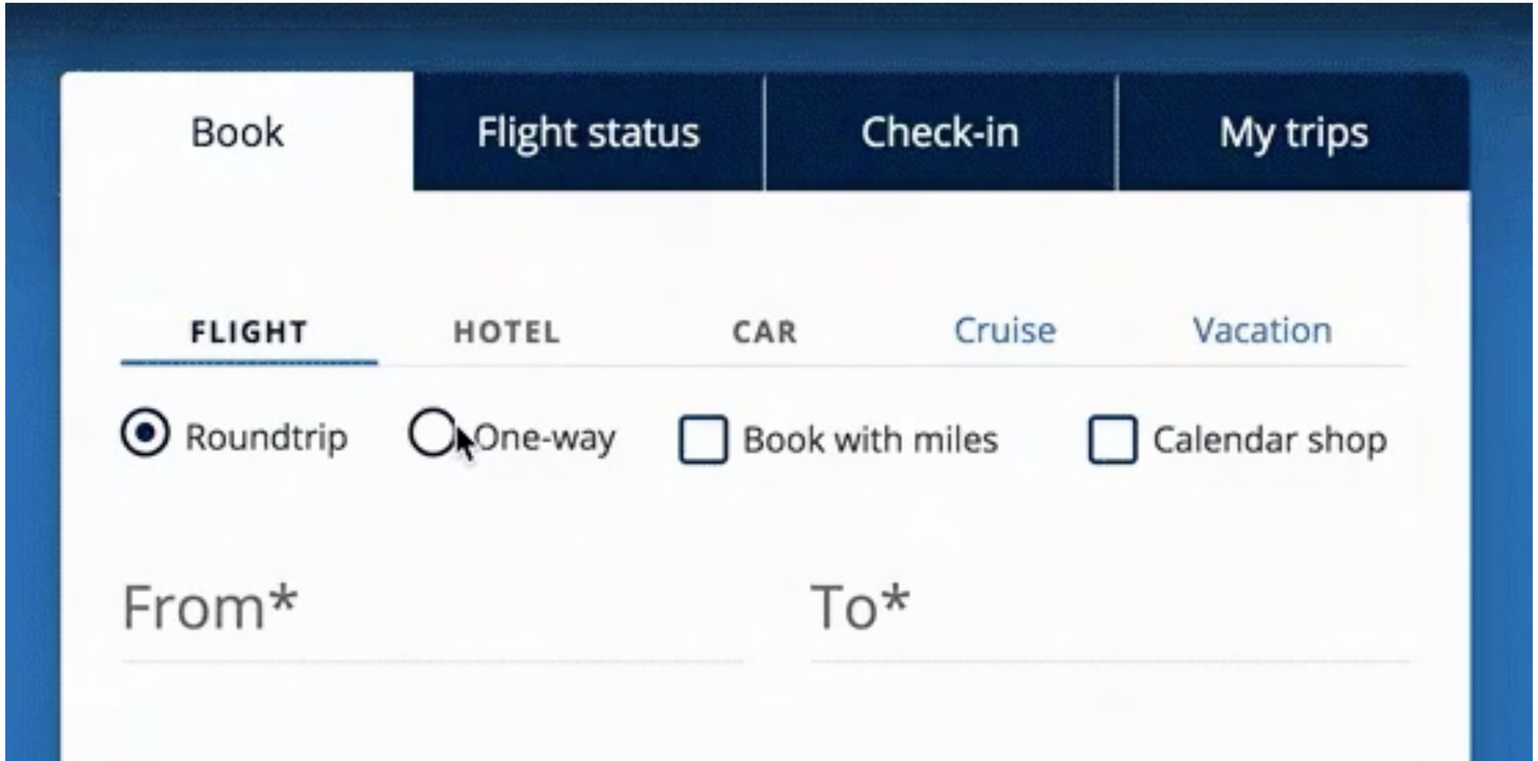


Katie Hempenius
@katiehempenius

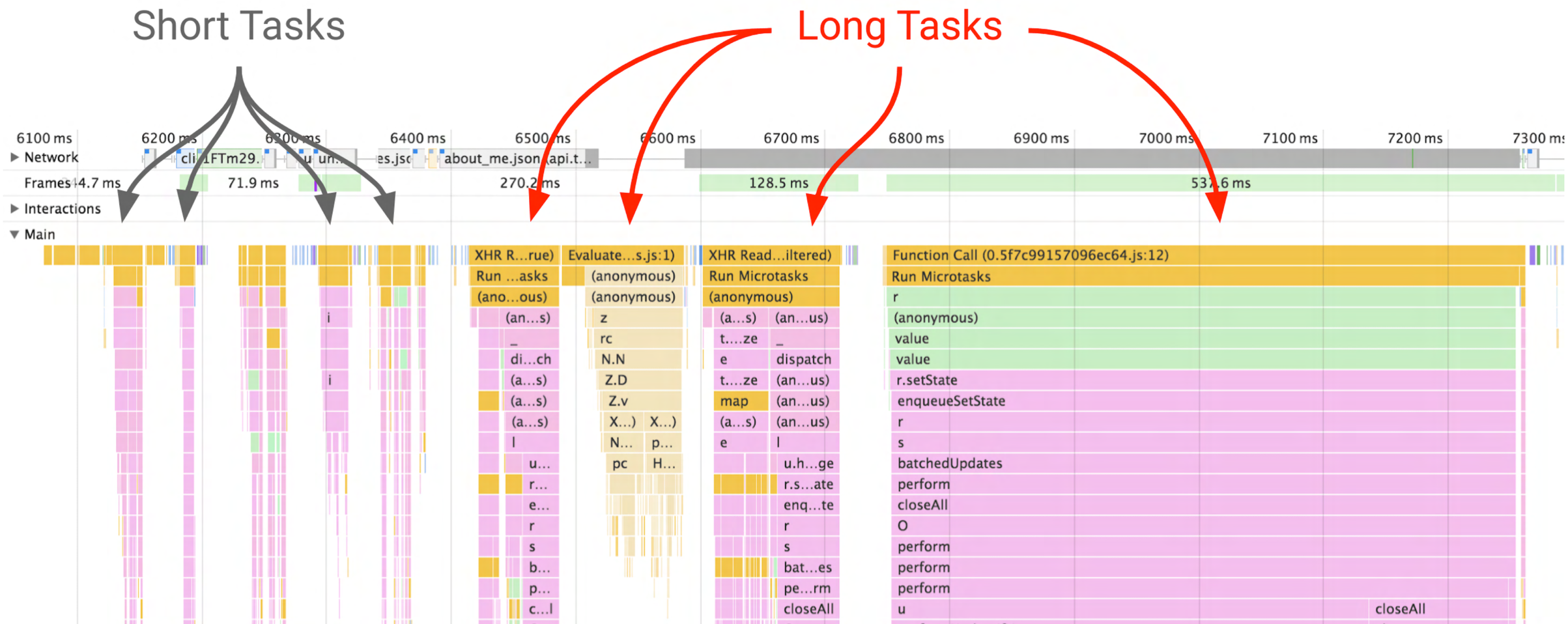
🎞 This is what First Input Delay (FID) looks like.

The page looks ready, but the main thread is busy, so the browser can't respond to my clicks.

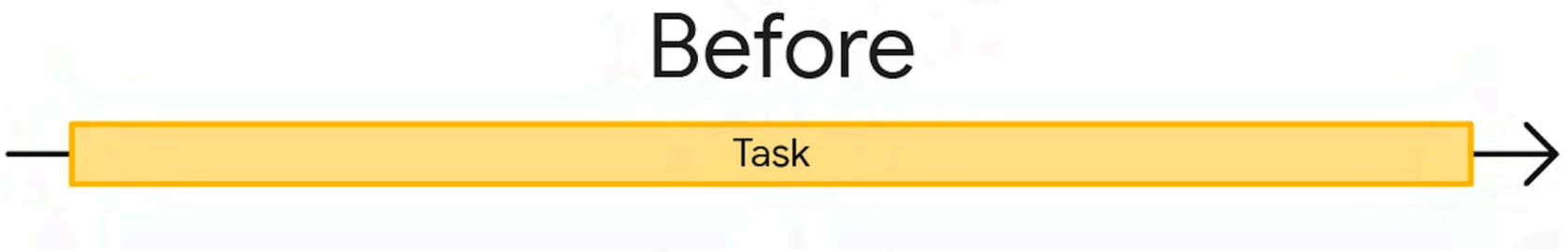
FID approximates this by measuring the time between the first user interaction w/ page & when the browser actually responds.



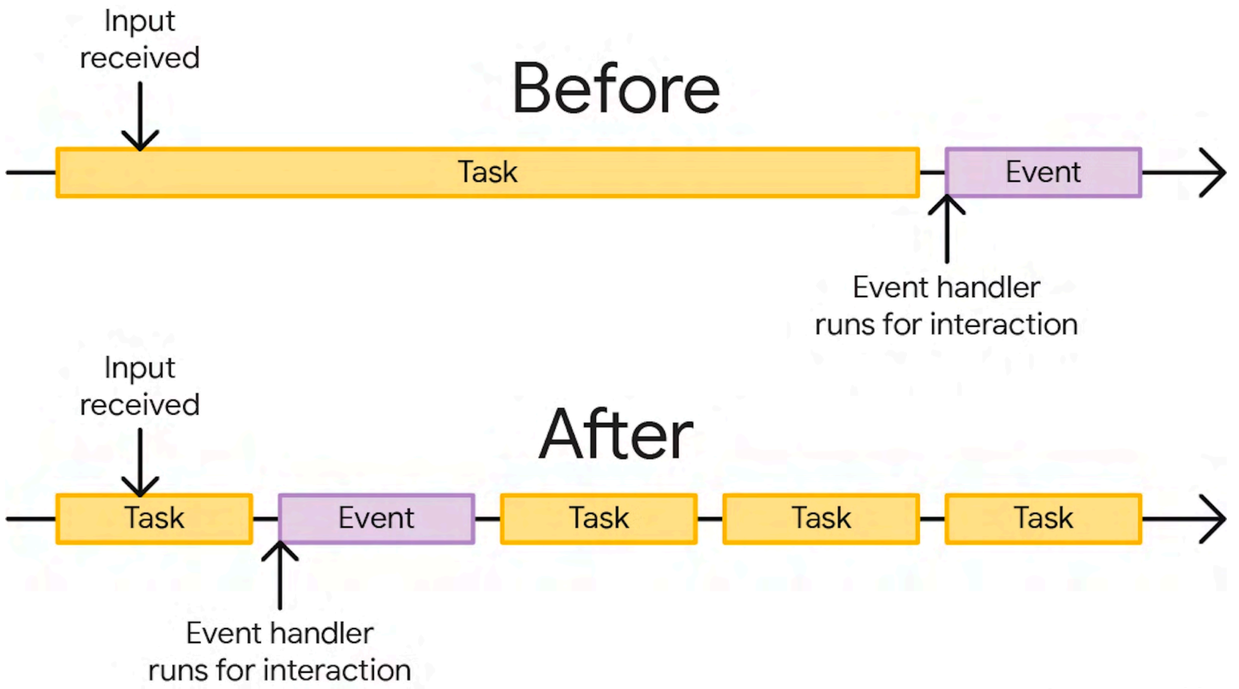
LONG TASKS MONOPOLIZE THE MAIN THREAD. BREAK EM' UP!



A)



B)

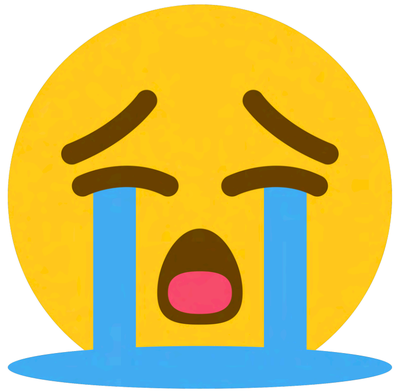


https://web.dev/optimize-long-tasks/

WHEN JAVASCRIPT BLOCKS THE MAIN THREAD, NOTHING ELSE CAN HAPPEN ON IT *

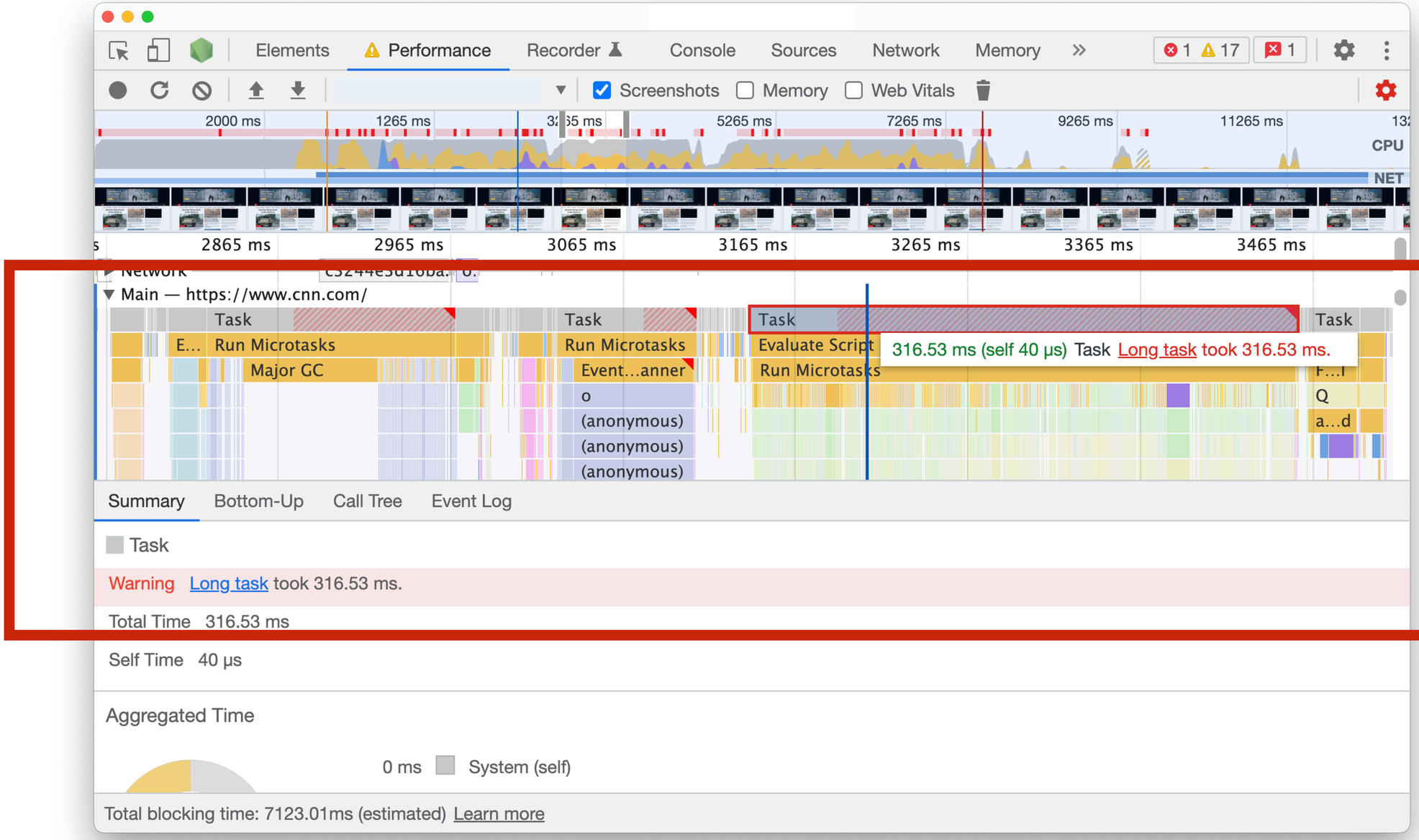
Block the main thread

Element	Example
`<img>`	
`<a>`	https://example.com
`<input>`	Check me Type here



https://philipwalton.com/articles/why-web-developers-need-to-care-about-interactivity/

LONG TASKS IN DEVTOOLS




GET INTERACTIVE
IN 5 SECONDS

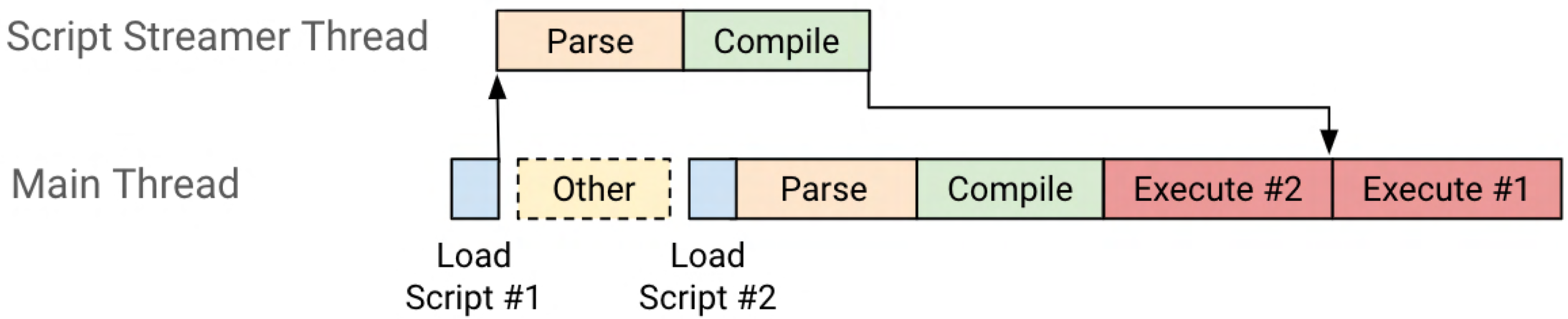
FAST JAVASCRIPT

=

FAST AT

DOWNLOAD
PARSE / COMPILE
EXECUTE

SCRIPT STREAMING



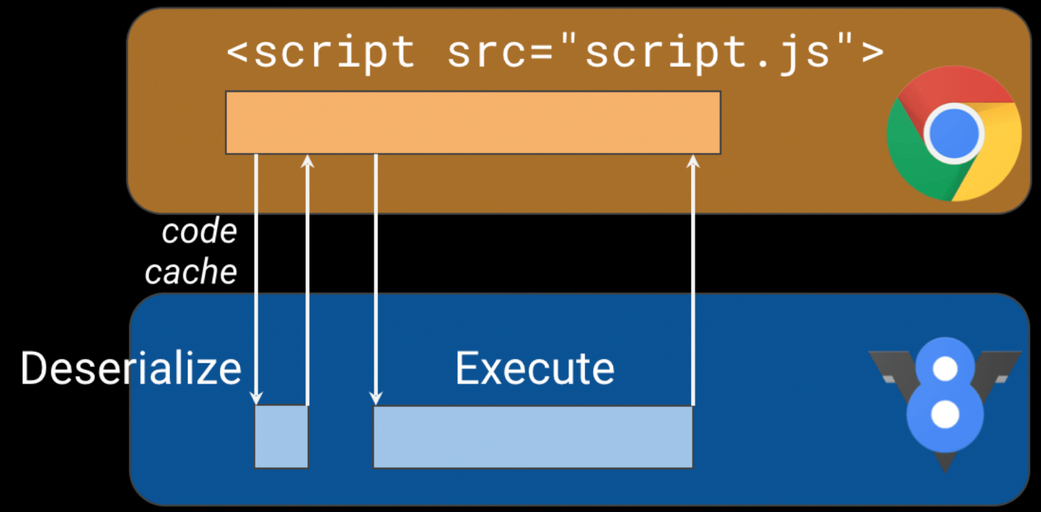
SCRIPTS ARRIVE IN MULTIPLE CHUNKS.
V8 STARTS STREAMING ONCE ITS SEEN AT LEAST 30KB.

* Background compilation: 1-2% improvement to main thread V8 time time reduction. Background parsing and other improvements to parse reduced overall V8 load time by up to 10-20%

CODE CACHING IN V8

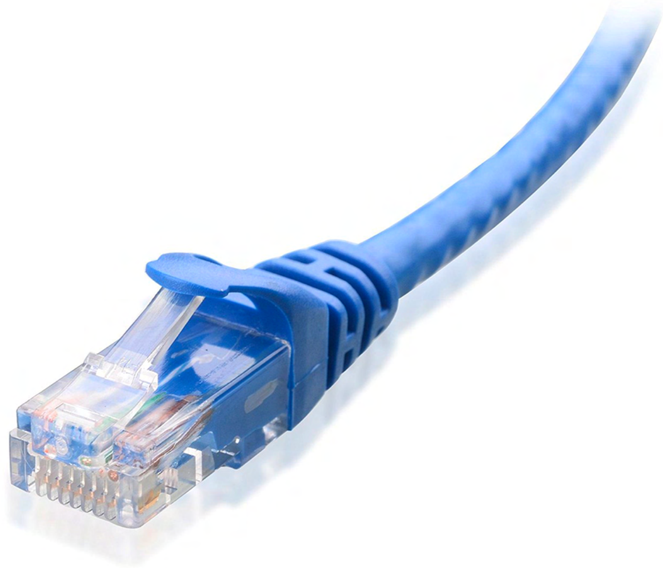
~20-40% reduction in parse and compilation time during load.

	In-memory (same tab*)	On-disk
Cold run ❄	Compile	
Warm run ☀	Use isolate cache	Compile + Serialize
Hot runs 🔥		Deserialize

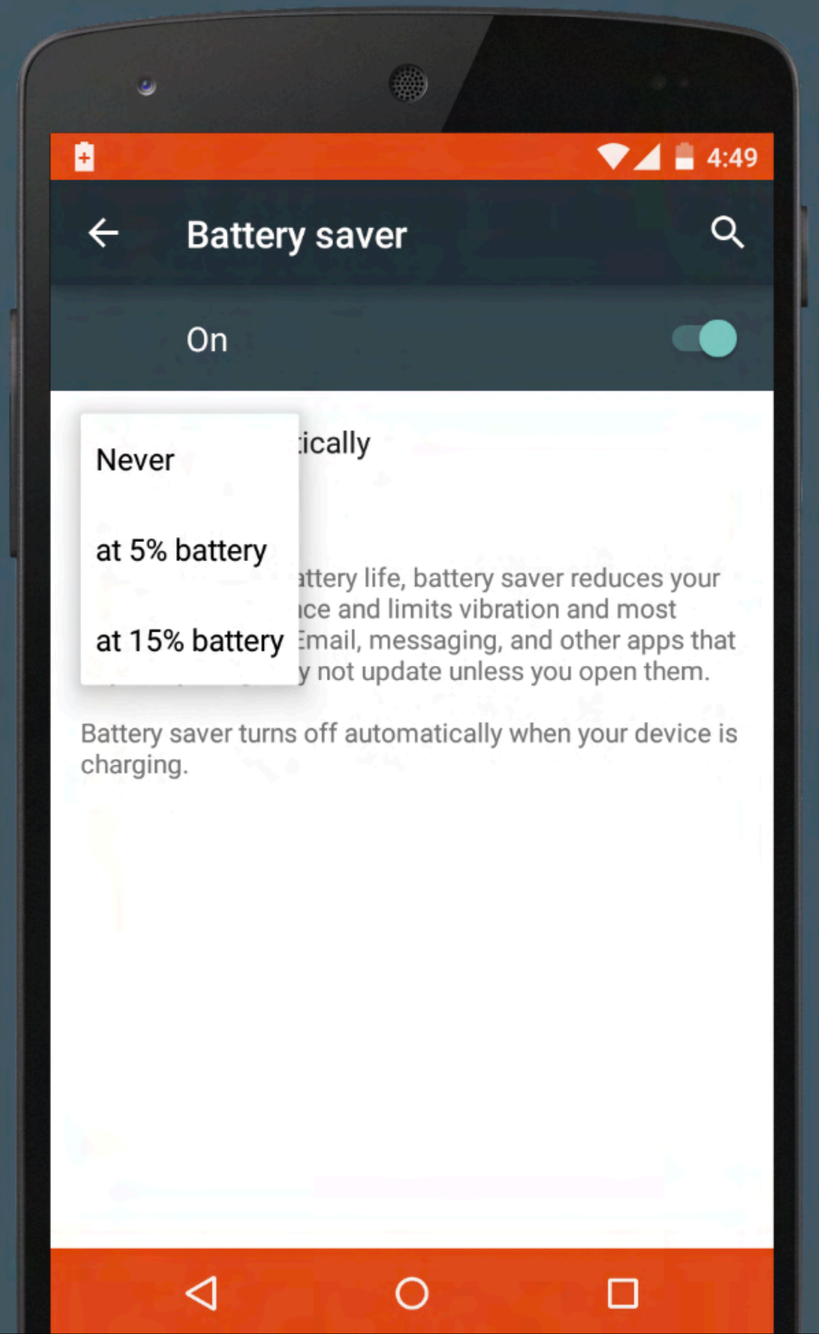


** actually "same V8 isolate", which means same renderer process and same thread*

DOWNLOAD TIMES CRITICAL FOR SLOW NETWORKS



JAVASCRIPT EXECUTION TIME IS CRITICAL FOR DEVICES WITH SLOW CPUS



Low battery?

Expect significantly* slower CPU performance.

Overheating?

Expect significantly* slower CPU performance.

Background process?

Expect slower CPU performance.

...

* local experiments show ~40% variance in CPU benchmarks.

OPTIMIZE FOR THE NETWORK

Network latency and bandwidth impact how soon things get over the wire.

I USE THE FREE WI-FI AT STARBUCKS TO COMPLAIN HOW SLOW THEIR FREE WI-FI IS.

A "FAST" CONNECTION IS NOT ALWAYS FAST.

HOW DO NETWORKS IMPACT PERFORMANCE?

Bandwidth

≈ data throughput (bits/second)

Low Bandwidth

High Bandwidth

Latency

≈ delay due data travel time (ms)

Low Latency

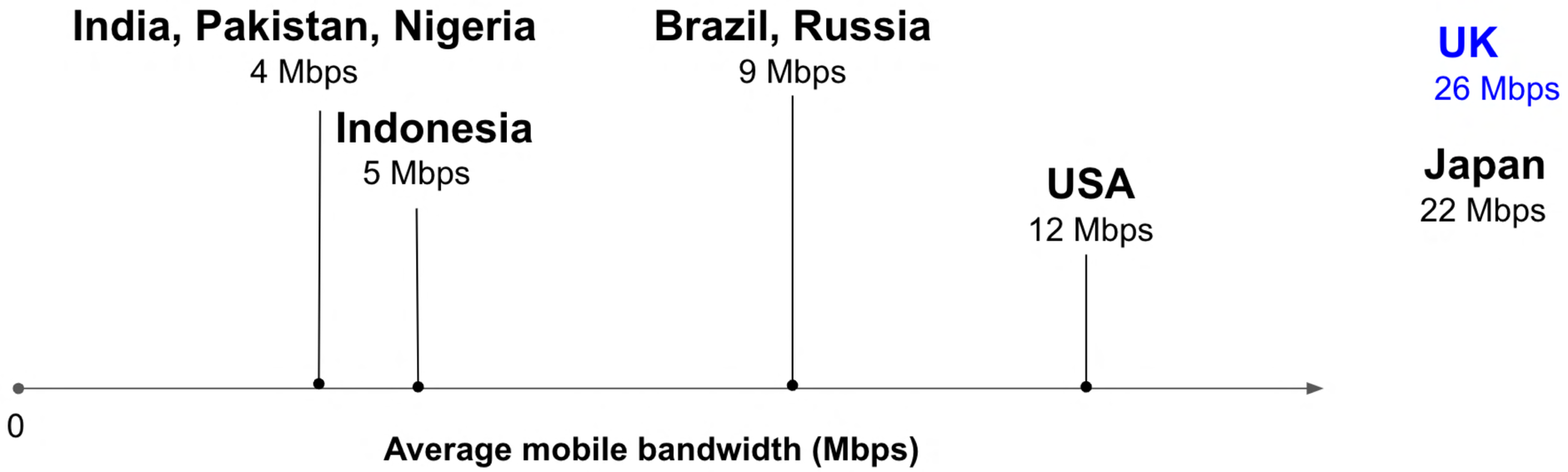
High Latency

ANALOGY: KATIE HEMPENIUS

BANDWIDTH MAY BE SUFFICIENT IN MANY MARKETS

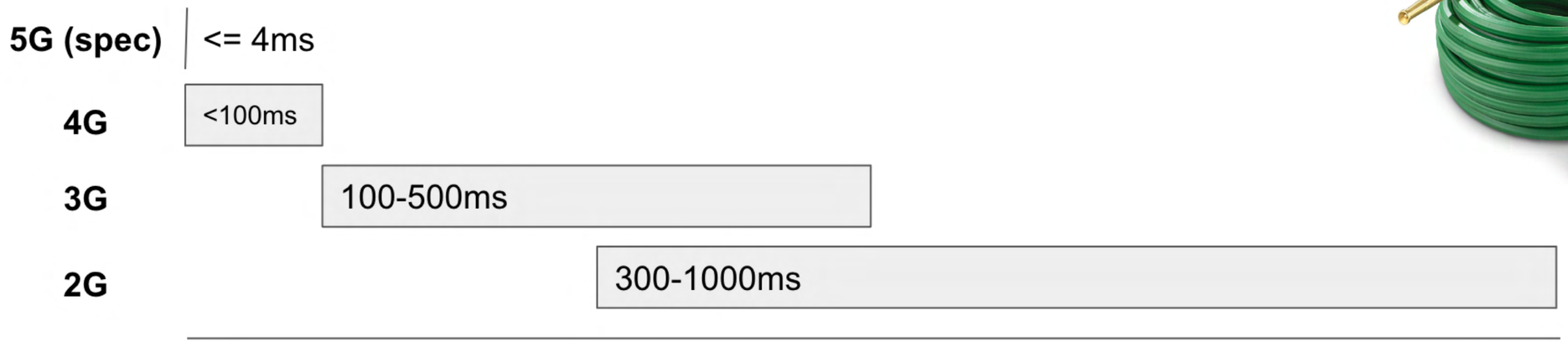
Average Web page size: 3.5Mb

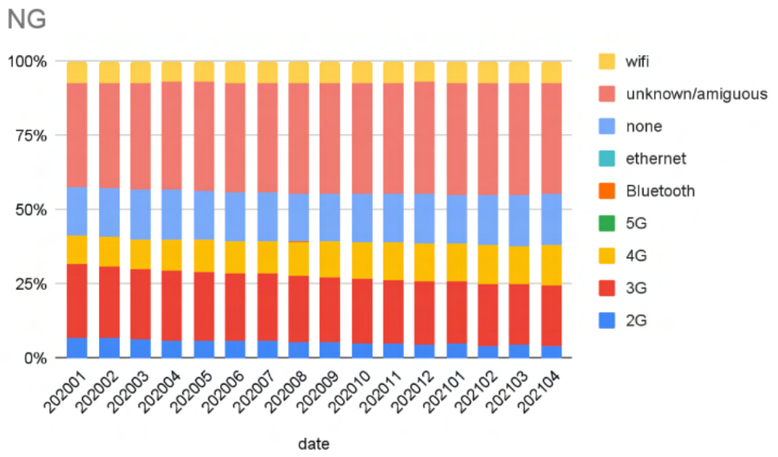
$$\frac{3.5\text{Mb}}{26\ \text{Mbps}} = .13\text{s}$$

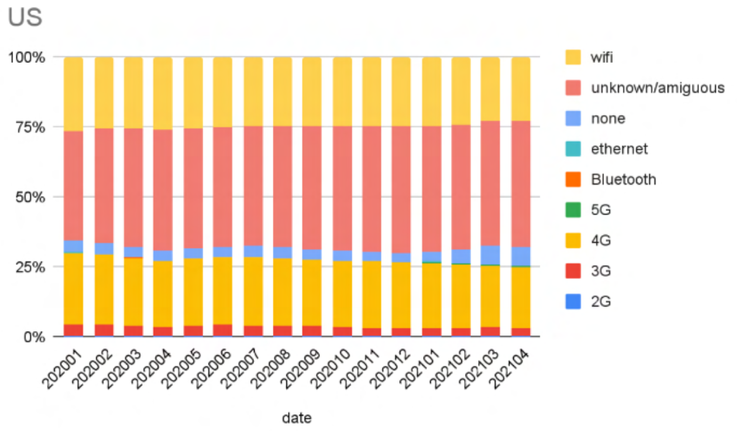


Source(s): Global State of Mobile Networks (February 2017), OpenSignal; HTTP Archive (Jan 2018); 2018 United Kingdom Speed Report (Jan 2018)

LATENCY IS HIGHLY CORRELATED WITH PERFORMANCE

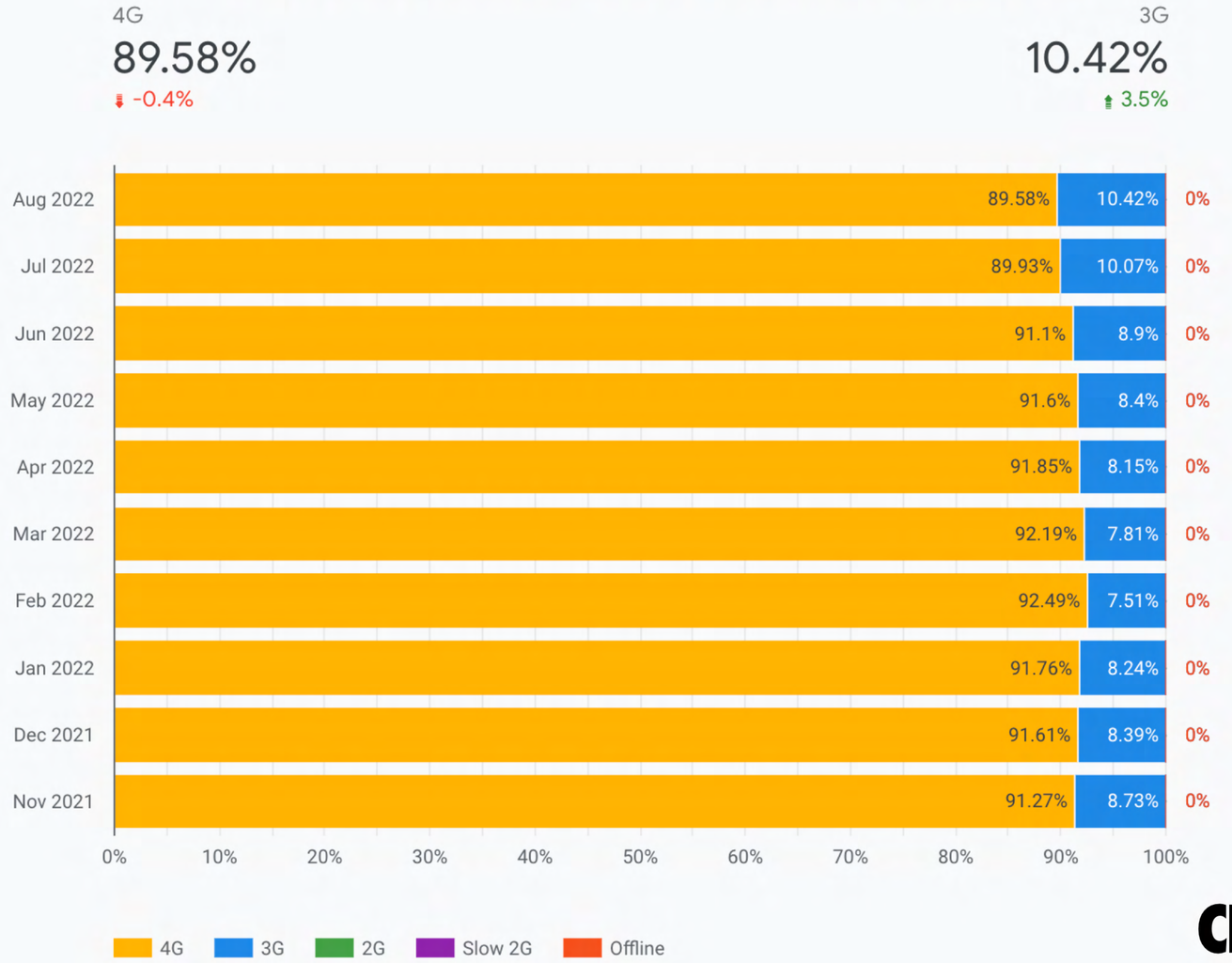






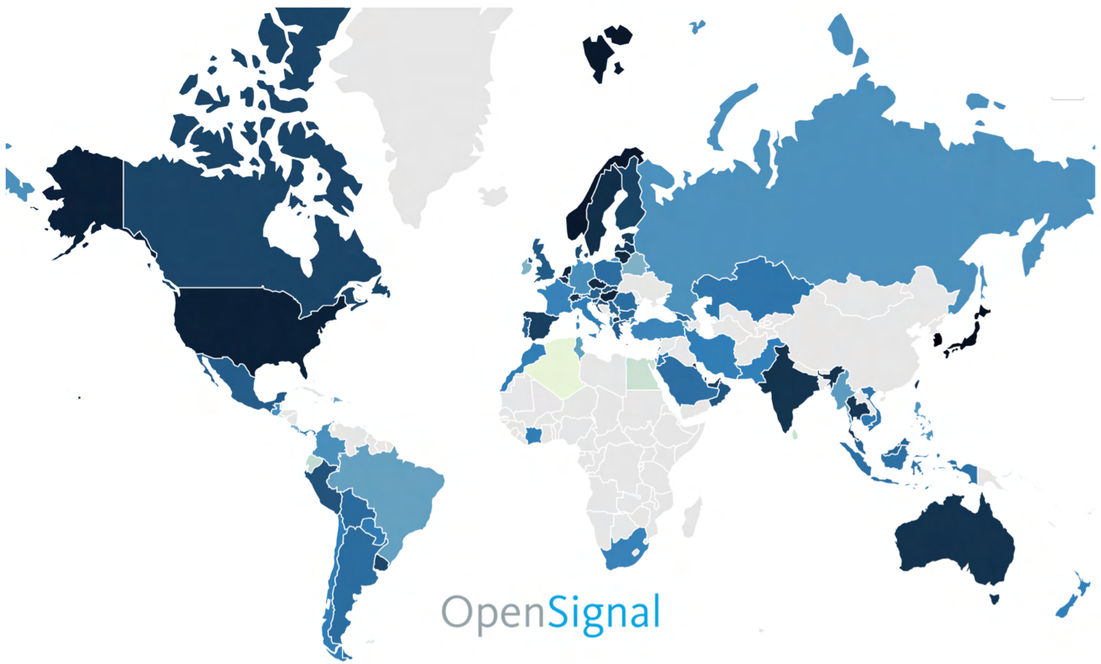
EFFECTIVE CONNECTION TYPE

MOBILE.TWITTER.COM

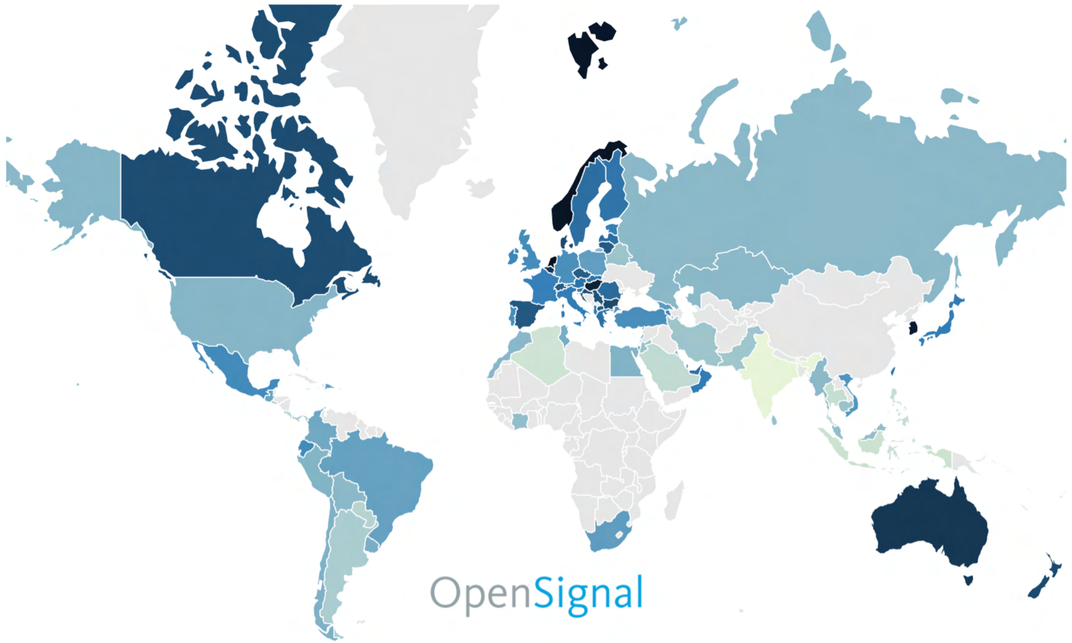


CHROME UX REPORT

GLOBAL 4G AVAILABILITY VS SPEED



4G AVAILABILITY (%) 40.94 — 97.49

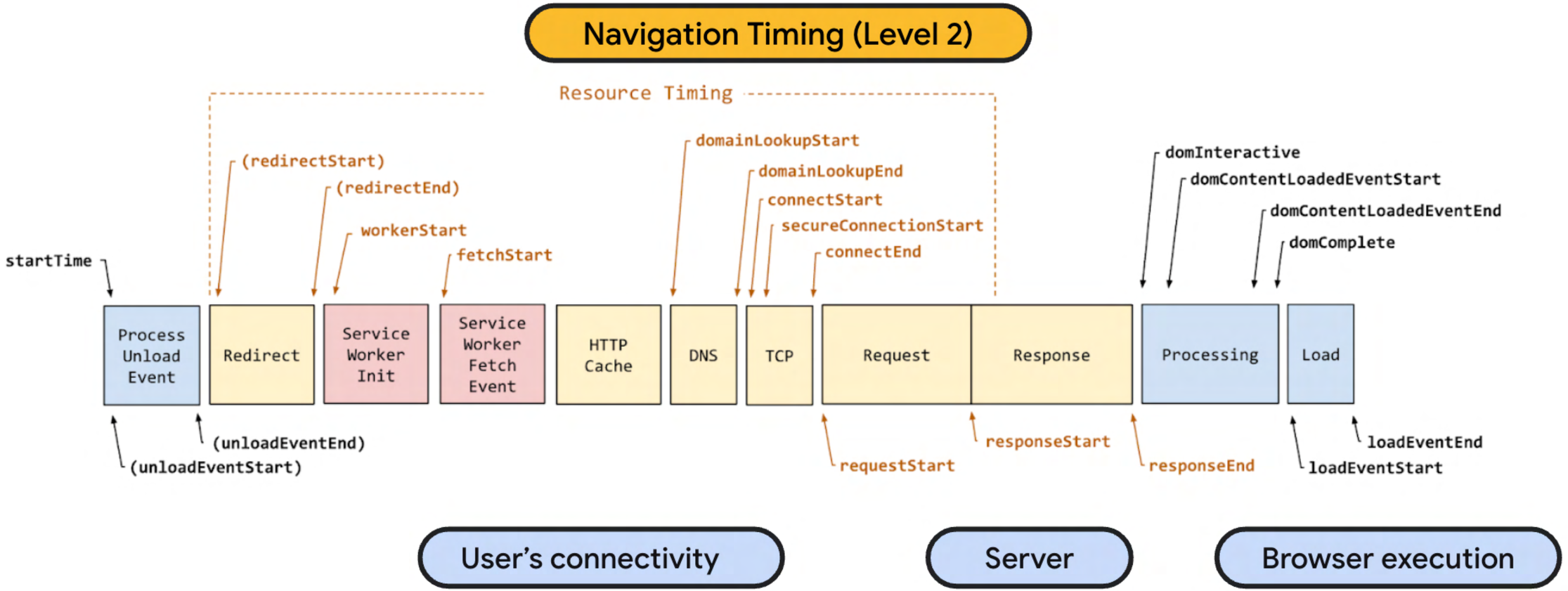


4G SPEED (Mbps) 6.07 — 44.31

RESOURCE LOADING IS HARD

- ⏱️ Performance is tightly coupled to latency
- 🤝 Connection cost is high
- 📈 Congestion control is unavoidable
- 🙈 Critical resources can be hidden
- 💤 Bandwidth is often under-utilised
- ⚡ Script execution is expensive

WHAT'S IN A NAVIGATION?

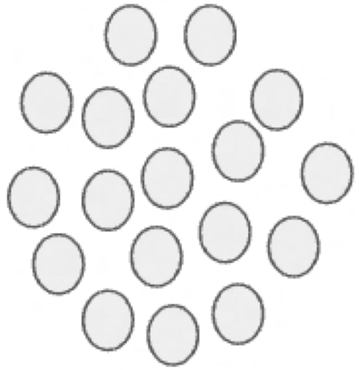


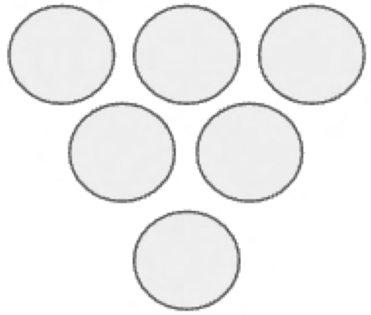
Examples: Time needed to unload previous page, total time required to load from initial navigation request etc.

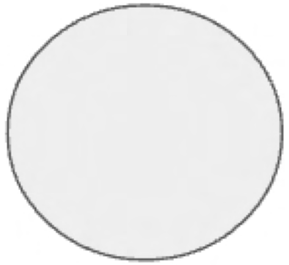
EVERGREEN NETWORK BEST PRACTICES

- ➔ Reduce DNS lookups
- ➔ Reuse TCP connections
- ➔ Use a Content Delivery Network
- ➔ Minimize number of HTTP redirects
- ➔ Eliminate unnecessary request bytes
- ➔ Compress assets during transfer
- ➔ Cache resources on the client
- ➔ Eliminate unnecessary resources

JS COMPRESSION TRADE-OFFS







1 module : 1 script

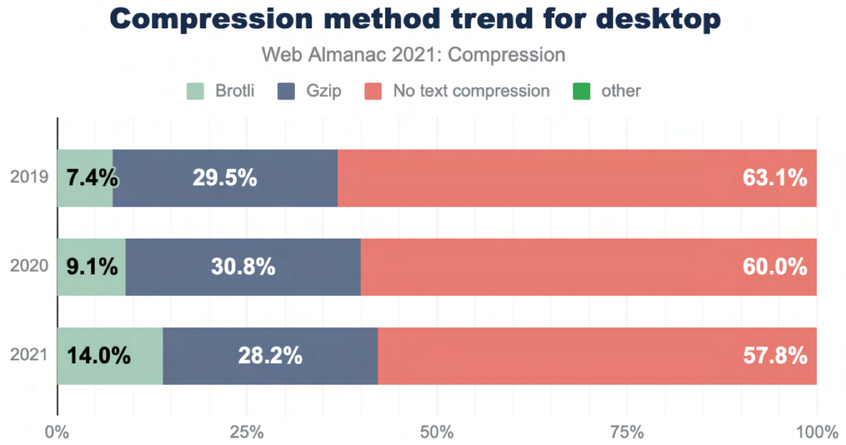
Bundles

1 application : 1 script

- Poor compression
+ Good caching
+ Good granularity

+ Good compression
- Poor caching
- Poor granularity

SOURCE: KATIE HEMPENIUS



COMPRESS.

USE BROTLI

OYO

Compress JS & CSS

15%

JavaScript reduction

37%

Latency Improvement (P50)

Compress API Responses

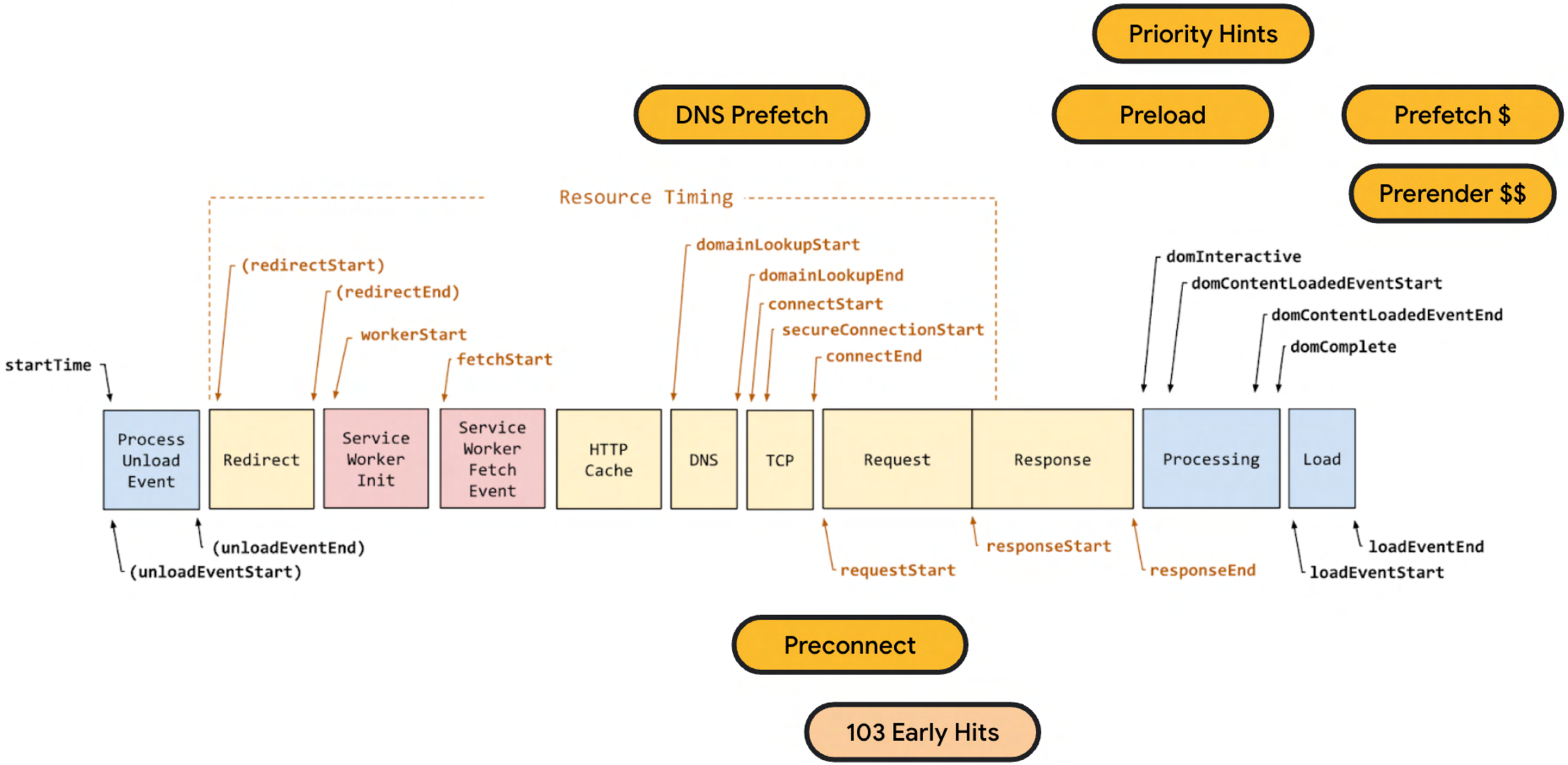
90%

Decrease in payload size for P75 sized payloads

5%

Reduction globally for P95 load latencies. ~15-20% in Emerging Markets

PRELOAD, PREFETCH & PRIORITIES



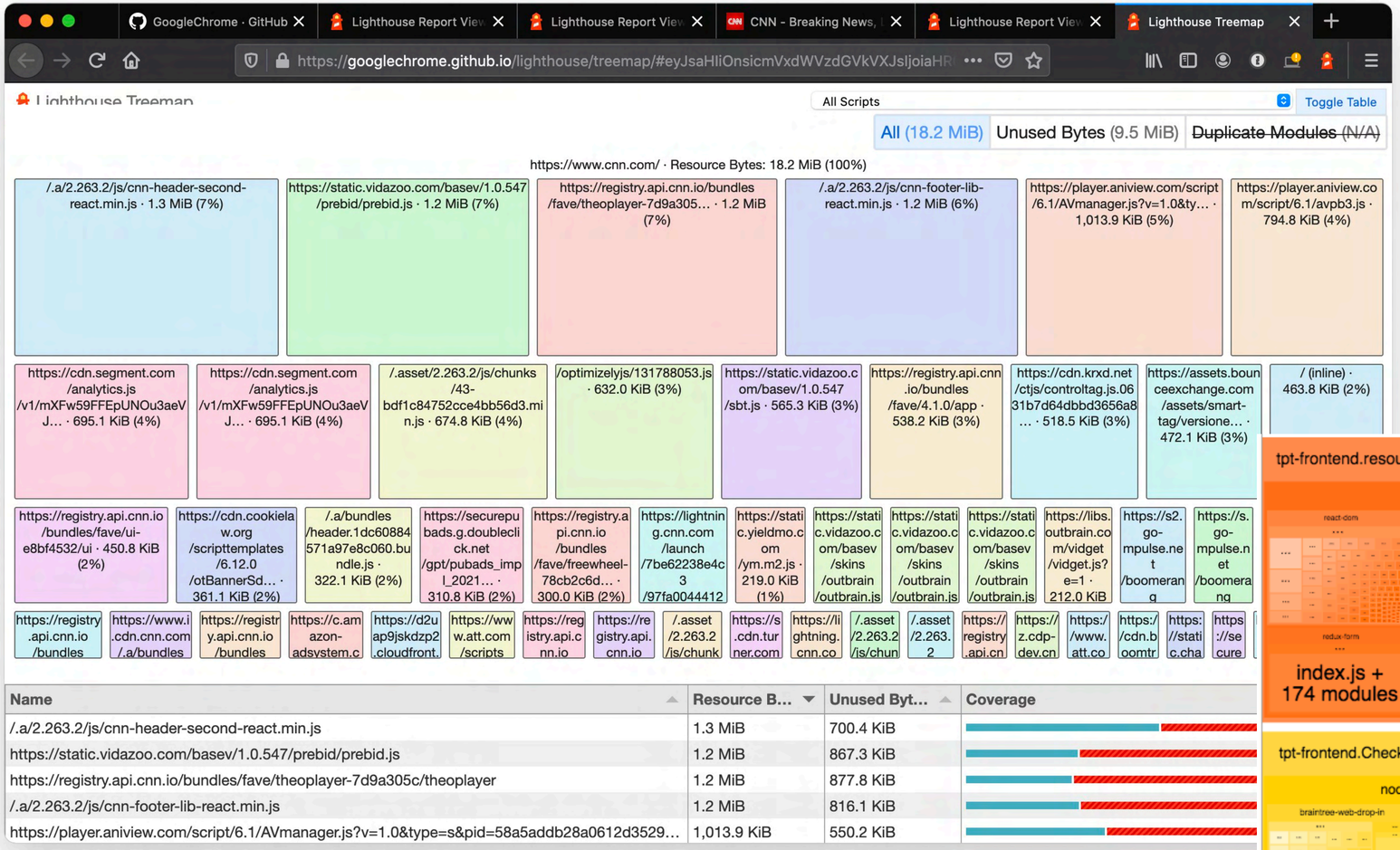
OPTIMIZE LOADING & RENDERING

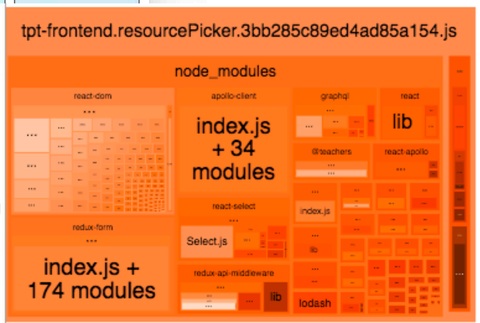
How do we optimize for network and hardware constraints while delivering our desired UX?

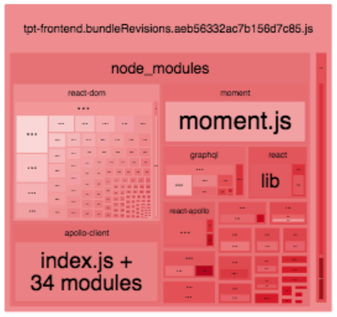
WHAT'S IN YOUR JAVASCRIPT BUNDLE?

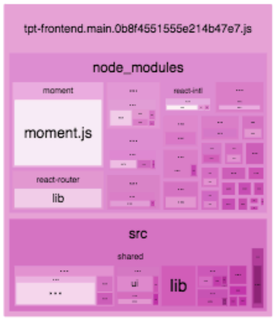
npm install

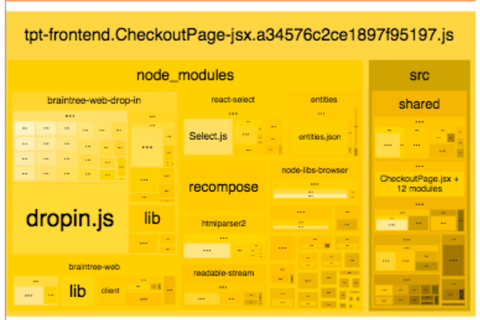
@ChrisArter

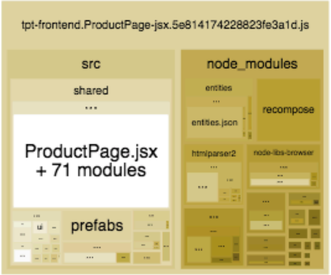

GoogleChrome · GitHub
Lighthouse Report Viev
Lighthouse Report Viev
CNN - Breaking News,
Lighthouse Report Viev
Lighthouse Treemap
https://googlechrome.github.io/lighthouse/treemap/#eyJsaHliOnsicmVxdWVzdGVkVXJsIjoiaHR
Lighthouse Treemap
All Scripts
Toggle Table
All (18.2 MiB)
Unused Bytes (9.5 MiB)
Duplicate Modules (N/A)
https://www.cnn.com/ · Resource Bytes: 18.2 MiB (100%)
/.a/2.263.2/js/cnn-header-second-react.min.js · 1.3 MiB (7%)
https://static.vidazoo.com/basev/1.0.547/prebid/prebid.js · 1.2 MiB (7%)
https://registry.api.cnn.io/bundles/fave/theoplayer-7d9a305… · 1.2 MiB (7%)
/.a/2.263.2/js/cnn-footer-lib-react.min.js · 1.2 MiB (6%)
https://player.aniview.com/script/6.1/AVmanager.js?v=1.0&ty… · 1,013.9 KiB (5%)
https://player.aniview.com/script/6.1/avpb3.js · 794.8 KiB (4%)
https://cdn.segment.com/analytics.js/v1/mXFw59FFEpUNOu3aeVJ… · 695.1 KiB (4%)
https://cdn.segment.com/analytics.js/v1/mXFw59FFEpUNOu3aeVJ… · 695.1 KiB (4%)
/.asset/2.263.2/js/chunks/43-bdf1c84752cce4bb56d3.min.js · 674.8 KiB (4%)
/optimizelyjs/131788053.js · 632.0 KiB (3%)
https://static.vidazoo.com/basev/1.0.547/sbt.js · 565.3 KiB (3%)
https://registry.api.cnn.io/bundles/fave/4.1.0/app · 538.2 KiB (3%)
https://cdn.krxd.net/ctjs/controltag.js.0631b7d64dbbd3656a8… · 518.5 KiB (3%)
https://assets.bounceexchange.com/assets/smart-tag/versione… · 472.1 KiB (3%)
/ (inline) · 463.8 KiB (2%)
https://registry.api.cnn.io/bundles/fave/ui-e8bf4532/ui · 450.8 KiB (2%)
https://cdn.cookielaw.org/scripttemplates/6.12.0/otBannerSd… · 361.1 KiB (2%)
/.a/bundles/header.1dc60884571a97e8c060.bundle.js · 322.1 KiB (2%)
https://securepubads.g.doubleclick.net/gpt/pubads_impl_2021… · 310.8 KiB (2%)
https://registry.api.cnn.io/bundles/fave/freewheel-78cb2c6d… · 300.0 KiB (2%)
https://lightning.cnn.com/launch/7be62238e4c3/97fa0044412
https://static.yieldmo.com/ym.m2.js · 219.0 KiB (1%)
https://static.vidazoo.com/basev/skins/outbrain/outbrain.js
https://static.vidazoo.com/basev/skins/outbrain/outbrain.js
https://static.vidazoo.com/basev/skins/outbrain/outbrain.js
https://libs.outbrain.com/vidget/vidget.js?e=1 · 212.0 KiB
https://s2.go-mpulse.net/boomerang
https://s.go-mpulse.net/boomerang
https://registry.api.cnn.io/bundles
https://www.i.cdn.cnn.com/.a/bundles
https://registry.api.cnn.io/bundles
https://c.amazon-adsystem.c
https://d2uap9jskdzp2.cloudfront.
https://www.att.com/scripts
https://registry.api.cnn.io
https://registry.api.cnn.io
/.asset/2.263.2/js/chunk
https://s.cdn.turner.com
https://lightning.cnn.co
/.asset/2.263.2/js/chun
/.asset/2.263.2
https://registry.api.cn
https://z.cdp-dev.cn
https://www.att.co
https://cdn.boomtr
https://static.cha
https://secure
Name
Resource B…
Unused Byt…
Coverage
/.a/2.263.2/js/cnn-header-second-react.min.js
1.3 MiB
700.4 KiB
https://static.vidazoo.com/basev/1.0.547/prebid/prebid.js
1.2 MiB
867.3 KiB
https://registry.api.cnn.io/bundles/fave/theoplayer-7d9a305c/theoplayer
1.2 MiB
877.8 KiB
/.a/2.263.2/js/cnn-footer-lib-react.min.js
1.2 MiB
816.1 KiB
https://player.aniview.com/script/6.1/AVmanager.js?v=1.0&type=s&pid=58a5addb28a0612d3529…
1,013.9 KiB
550.2 KiB

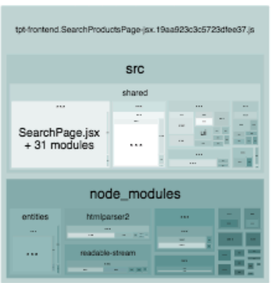

tpt-frontend.resourcePicker.3bb285c89ed4ad85a154.js
node_modules
index.js + 34 modules
index.js + 174 modules
lib
Select.js
lodash

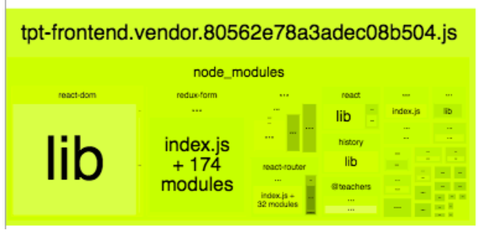

tpt-frontend.bundleRevisions.aeb56332ac7b156d7c85.js
node_modules
moment.js
index.js + 34 modules
lib

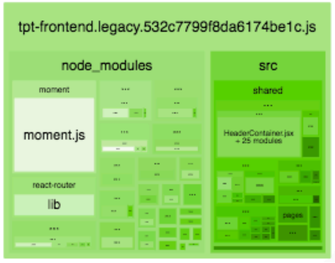

tpt-frontend.main.0b8f4551555e214b47e7.js
node_modules
moment.js
lib
src

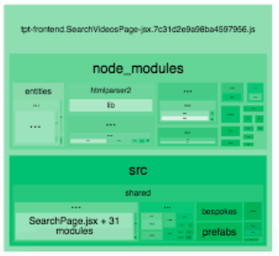

tpt-frontend.CheckoutPage-jsx.a34576c2ce1897f95197.js
node_modules
src
shared
dropin.js
lib
recompose
CheckoutPage.jsx + 12 modules


tpt-frontend.ProductPage-jsx.5e814174228823fe3a1d.js
src
shared
node_modules
ProductPage.jsx + 71 modules
prefabs
recompose

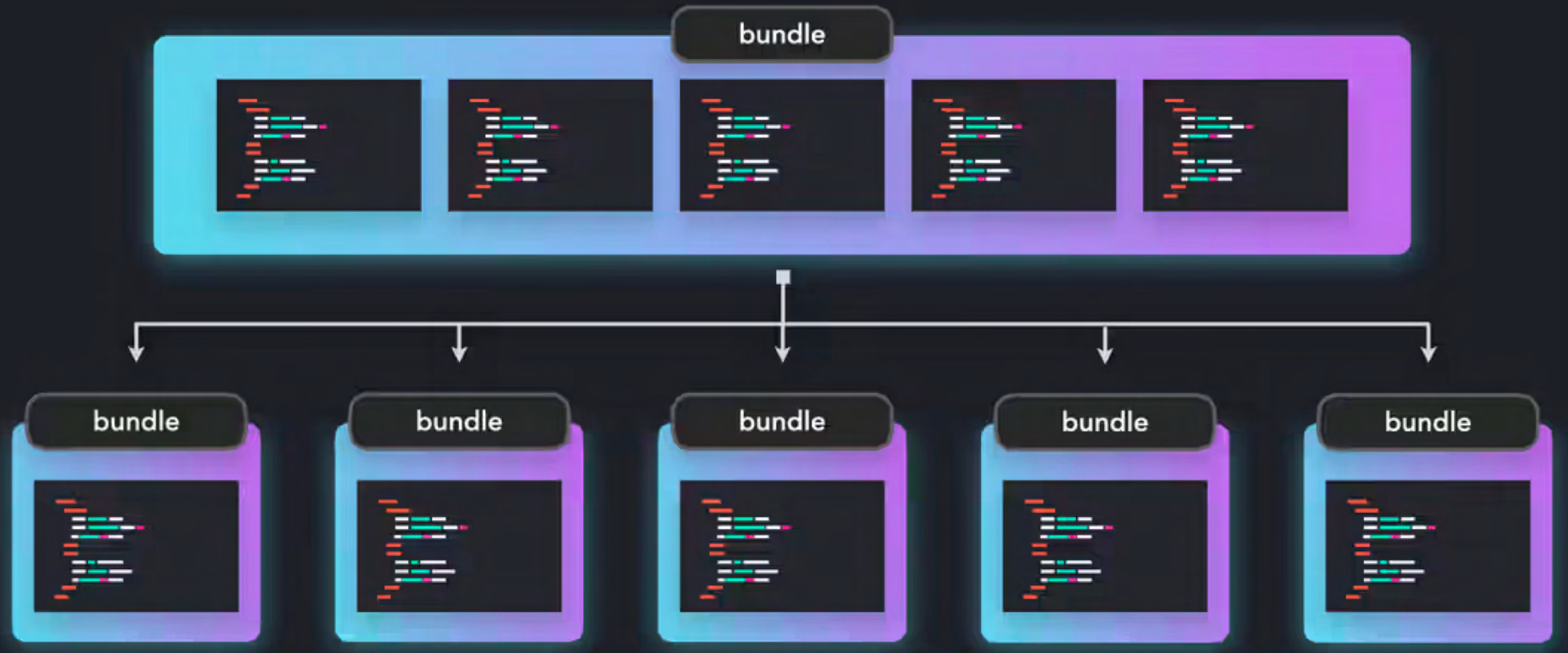

tpt-frontend.SearchProductsPage-jsx.19aa923c3c5723dfee37.js
src
shared
SearchPage.jsx + 31 modules
node_modules


tpt-frontend.vendor.80562e78a3adec08b504.js
node_modules
lib
index.js + 174 modules

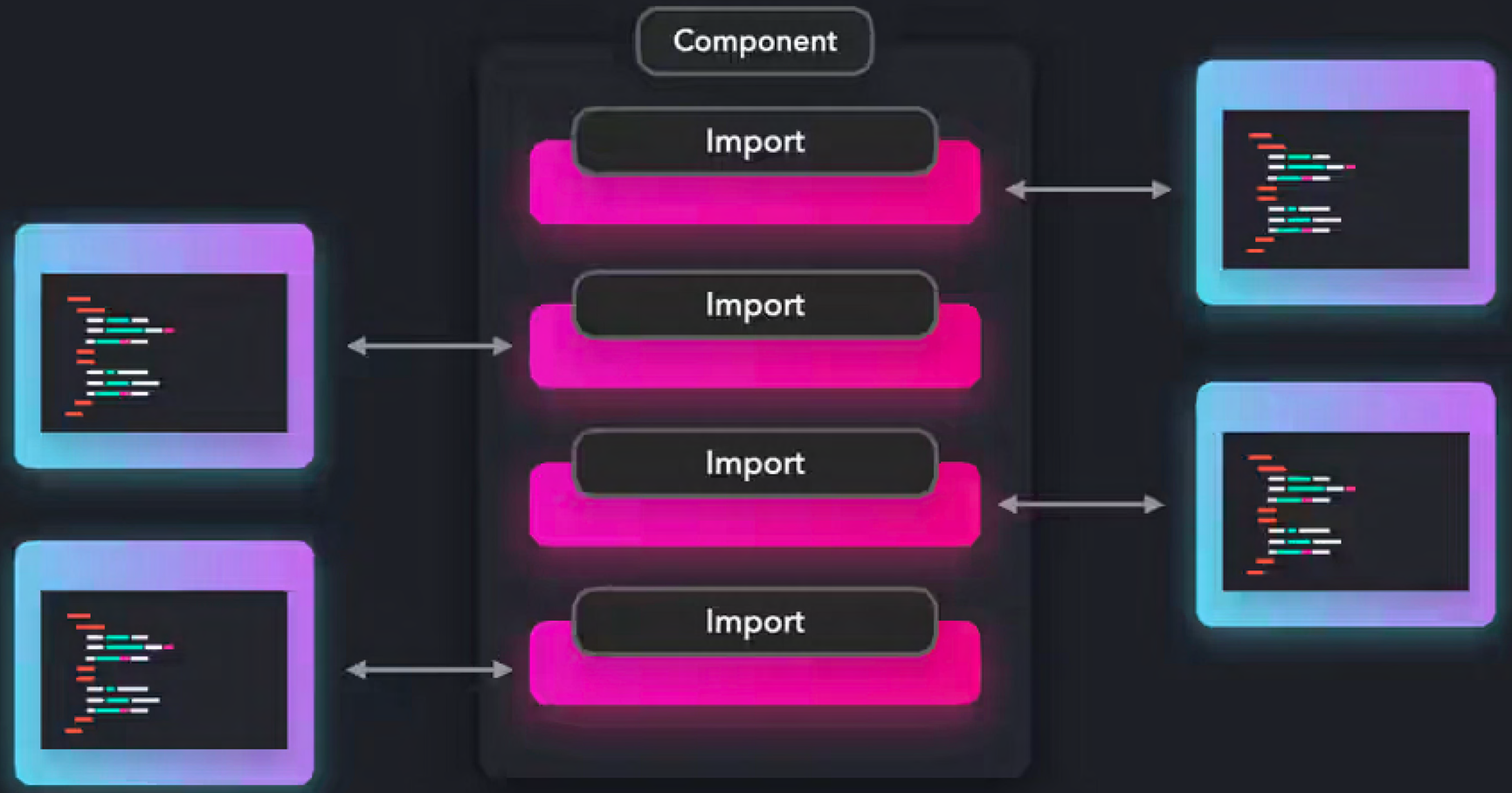

tpt-frontend.legacy.532c7799f8da6174be1c.js
node_modules
src
shared
moment.js
lib


tpt-frontend.SearchVideosPage-jsx.7c31d2e9a98ba4597956.js
node_modules
src
shared
SearchPage.jsx + 31 modules
bespokes
prefabs

CODE-SPLITTING BUNDLES



STATIC IMPORT



Static import

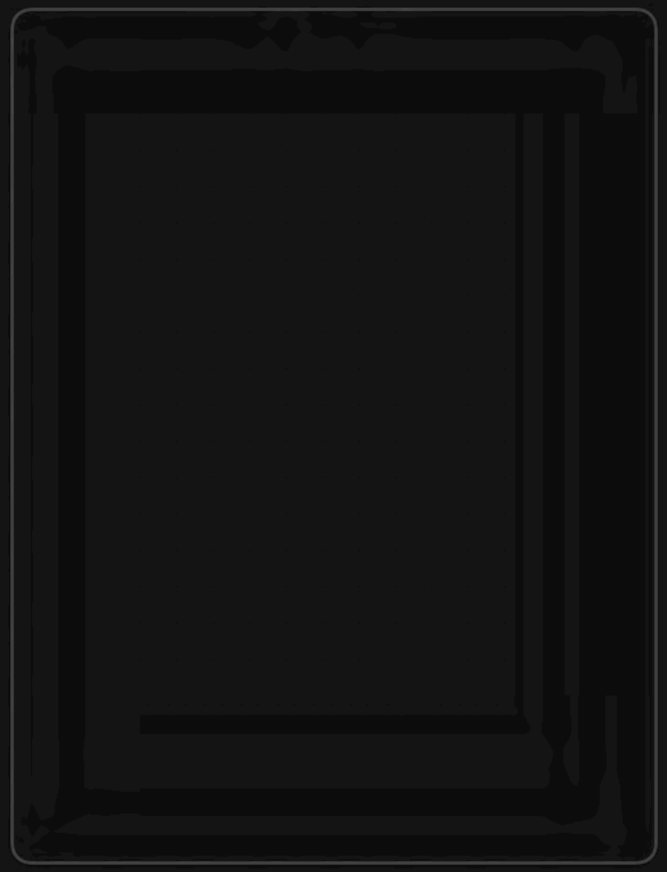
```
import UserInfo from "./UserInfo"
import ChatList from "./ChatList"
import ChatInput from "./ChatInput"

const App = () => { … }
```

```
import EmojiPicker from "./Picker"

const ChatInput = () => { … }
```

Bundle

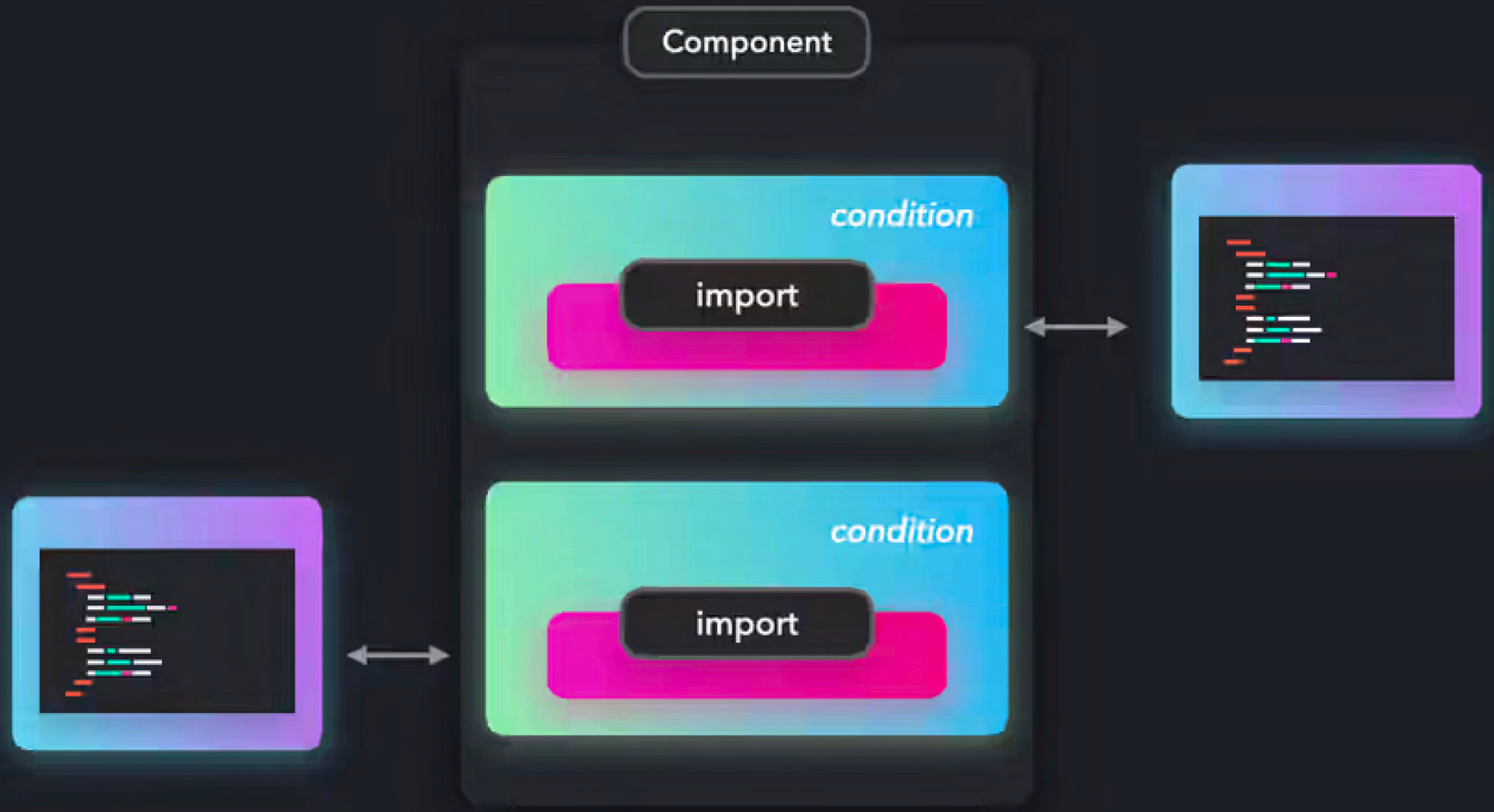


Static import

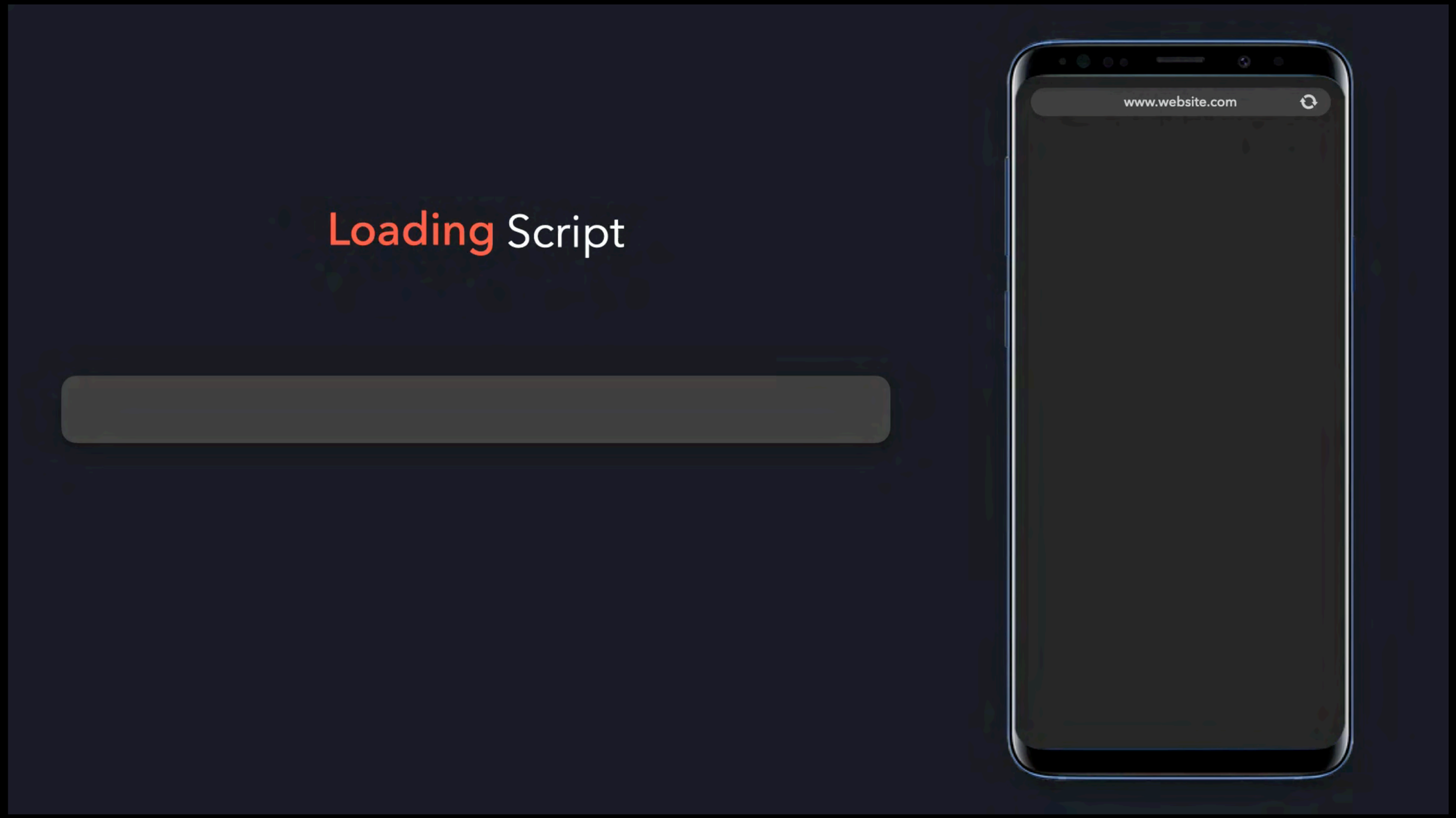
```
Asset              Size      Chunks                Chunk Names
main.bundle.js  1.5 MiB    main  [emitted]  main
```

Code-splitting bundles

DYNAMIC IMPORT



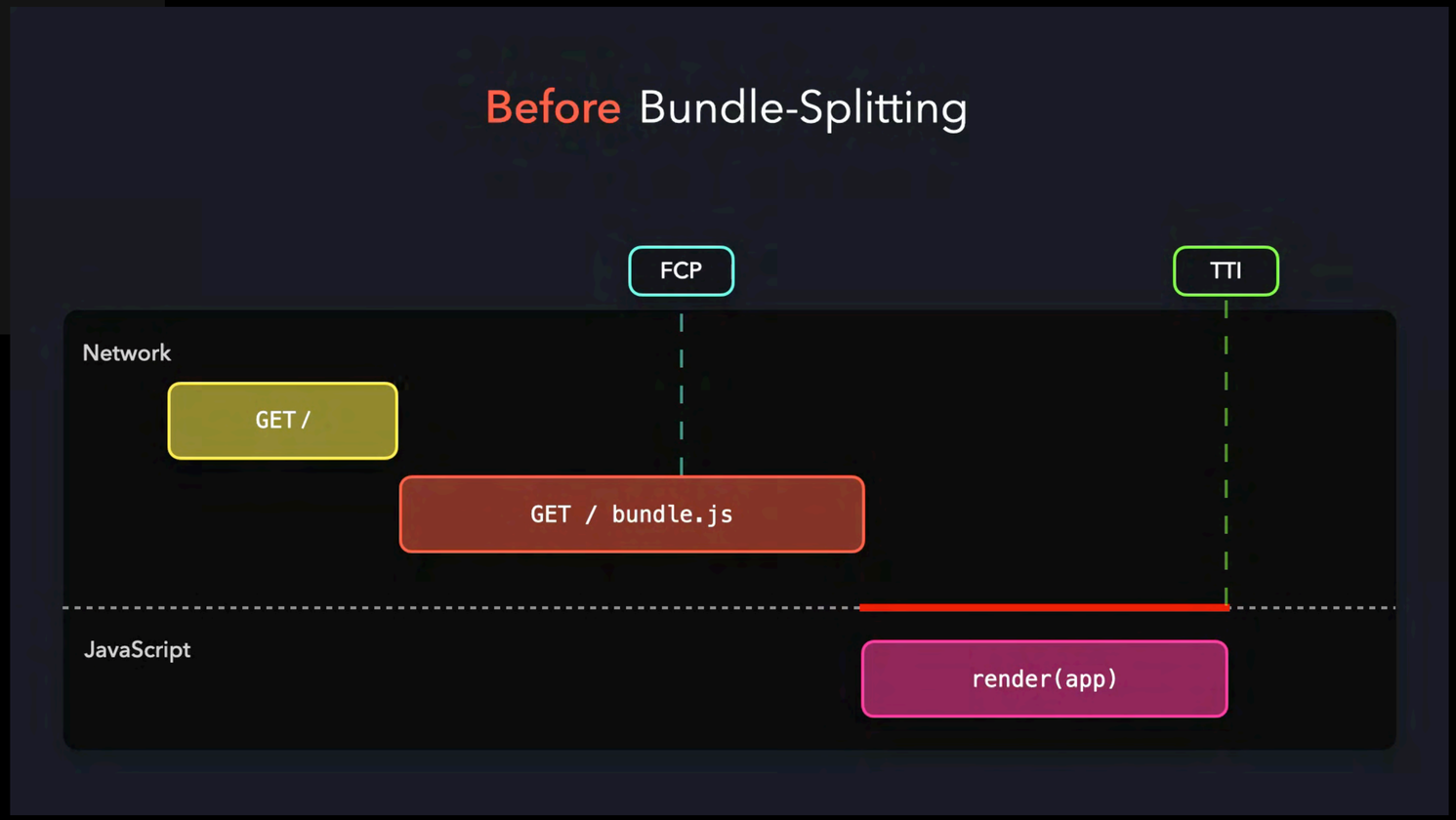
Dynamic Import



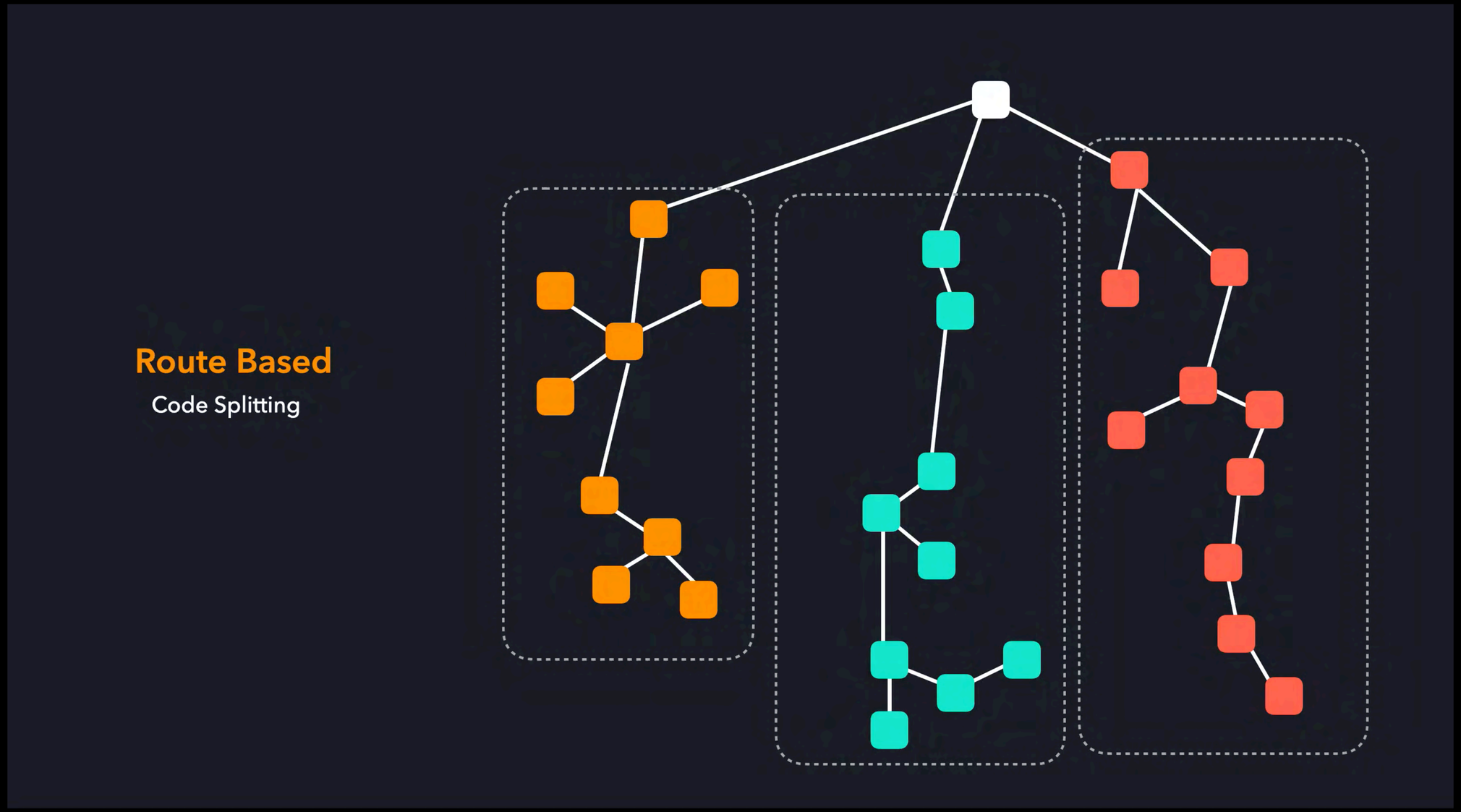
Dynamic Import

```
Asset                          Size        Chunks             Chunk Names
emoji-picker.bundle.js         1.48 KiB    1    [emitted]     emoji-picker
main.bundle.js                 1.33 MiB    main [emitted]     main
vendors~emoji-picker.bundle.js 171 KiB     2    [emitted]     vendors~emoji-pick
```

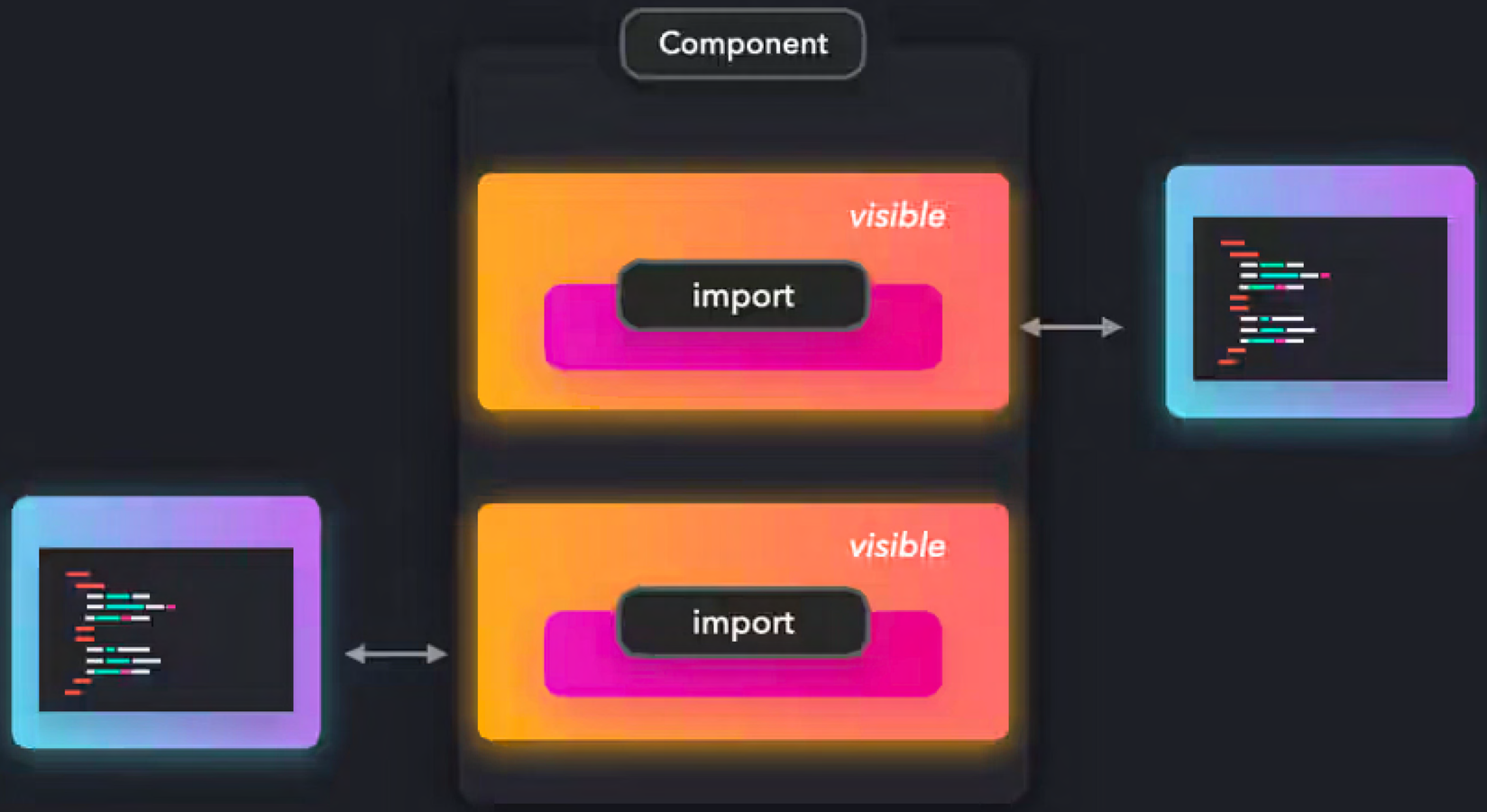
Code-splitting bundles



Route and component splitting



IMPORT ON VISIBILITY



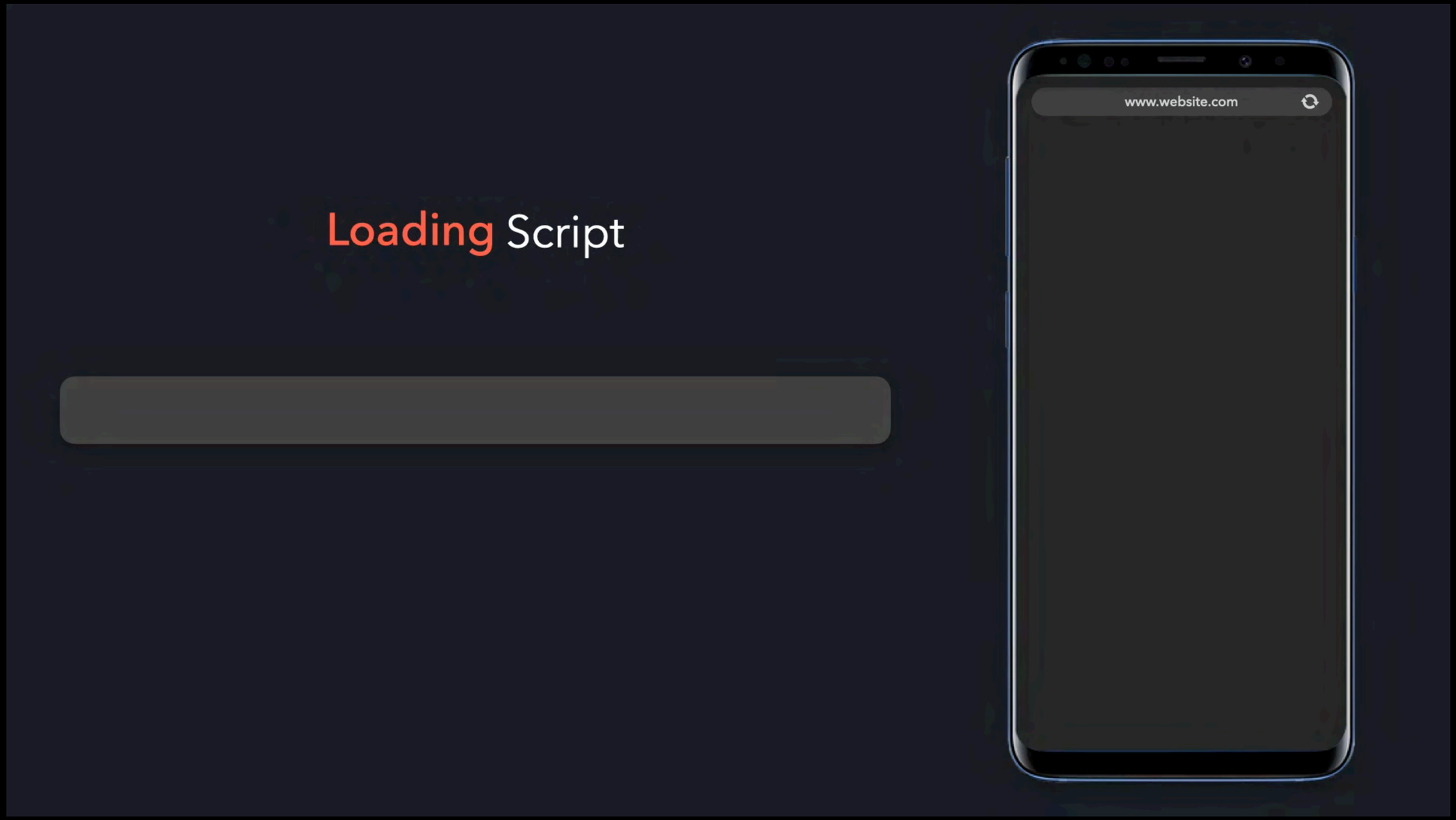
“Invisible”

Visible

Import On Visibility

Name	Waterfall

Import On Visibility



astro DOCS

Learn | Reference

Reference
- Configuration
- CLI
- Runtime API
- Integrations API
- Adapter API
- Template Directives
- NPM Package Format

`client:idle`

- **Priority:** Medium
- **Useful for:** Lower-priority UI elements that don't need to be imme

Load and hydrate the component JavaScript once the page is done w `requestIdleCallback` event has fired. If you are in a browser that do `requestIdleCallback`, then the document `load` event is used.

```
<ShowHideButton client:idle />
```

`client:visible`

- **Priority:** Low
- **Useful for:** Low-priority UI elements that are either far down the p resource-intensive to load that you would prefer not to load them element.

Load and hydrate the component JavaScript once the component has uses an `IntersectionObserver` internally to keep track of visibility.

```
<HeavyImageCarousel client:visible />
```

Eleventy 🎈 v2.0.0-canary.16
@eleven_ty

Peek at a new Partial Hydration/Islands Architecture library (server-framework-independent!) using web components 🏆 on 🎈 Eleventy Weekly №12

<is-land on:visible>
<is-land on:idle>
<is-land on:interaction>
<is-land on:media>
<is-land on:save-data>

youtube.com
Partial Hydration and Islands Architecture—Eleventy 🎈 W...
00:00 Week 1200:33 Community Roundup03:08 Eleventy Client Components03:21 `is-land` Web Components and ...

2:16 PM · May 31, 2022 · Twitter Web App

How and where do I want to render content?

Plain Static Rendering

Best for pages that:

- do **not require request-based data**

On-demand Incremental Static Regeneration

Best for pages that:

- should be **regenerated based on certain events**

Static with Client-Side `fetch`

Best for pages that:

- contain **data that should refresh on every page load**
- contain **stable placeholder components**

Server-Side Rendering

Best for pages that:

- require **request-based data**
- should be **render-blocking**

Incremental Static Regeneration

Best for pages that:

- should be **generated on a certain interval** or **on-demand**

And if:

- you have many pages to pre-render

Streaming SSR

Best for pages that:

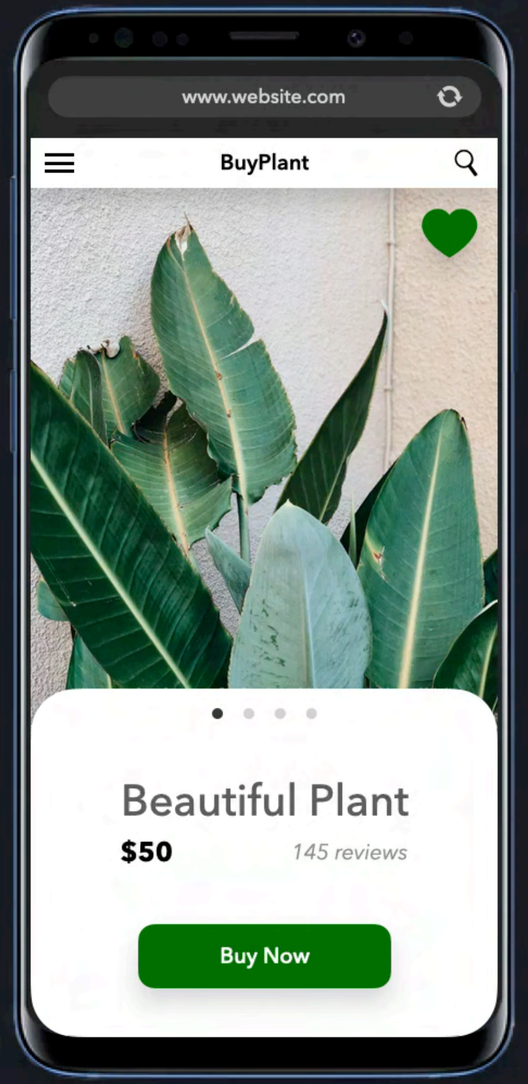
- are server-renderd

Check out Lydia Hallie's great "Rendering Patterns" talk


JS
JS
JS
WHAT IS HYDRATION?
AND WHY CAN IT MAKE YOU CRY?

Rehydration

Rehydration



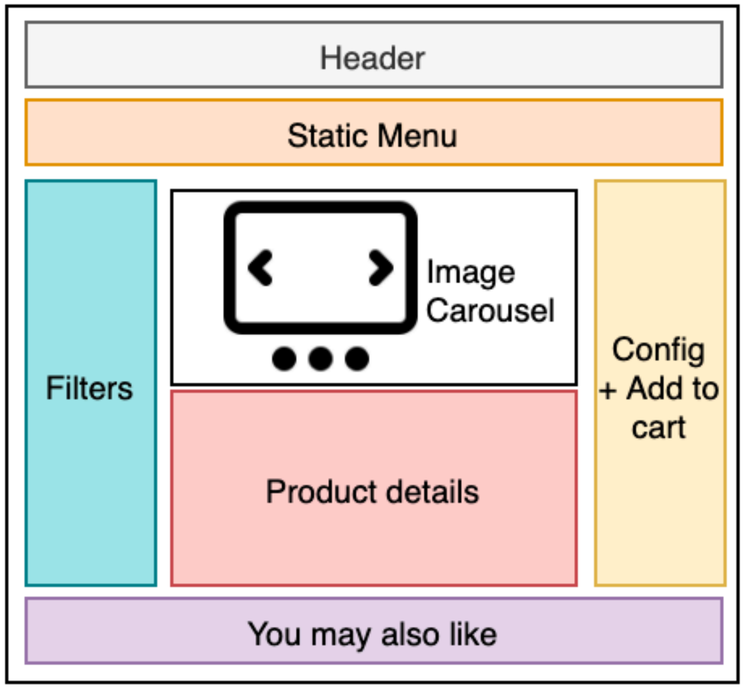
loading...

ISLANDS ARCHITECTURE

Partial Hydration

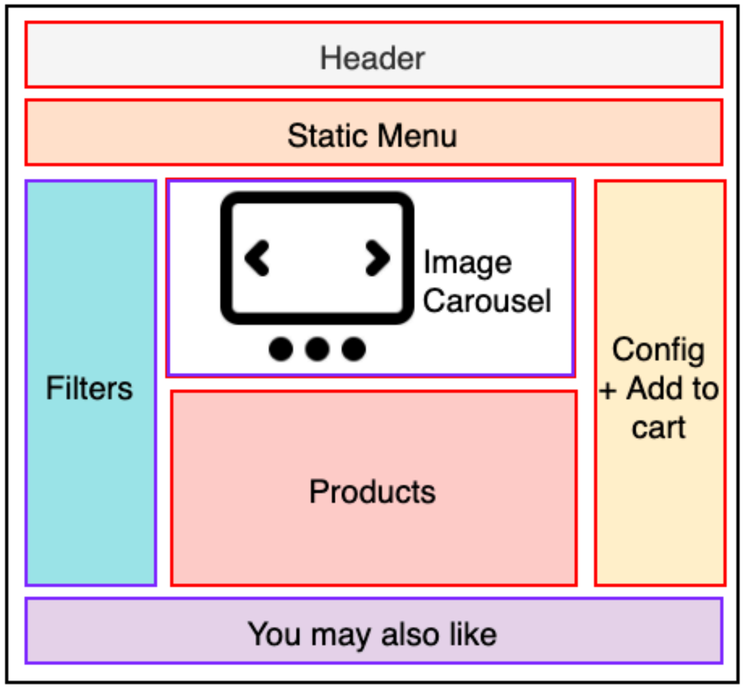
Islands Architecture

SSR



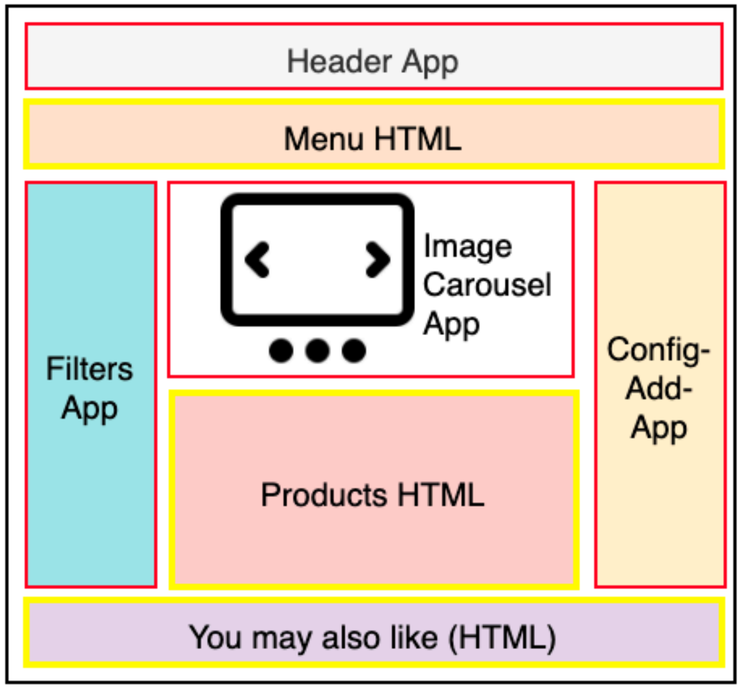
Render all components together and hydrate

Progressive Hydration



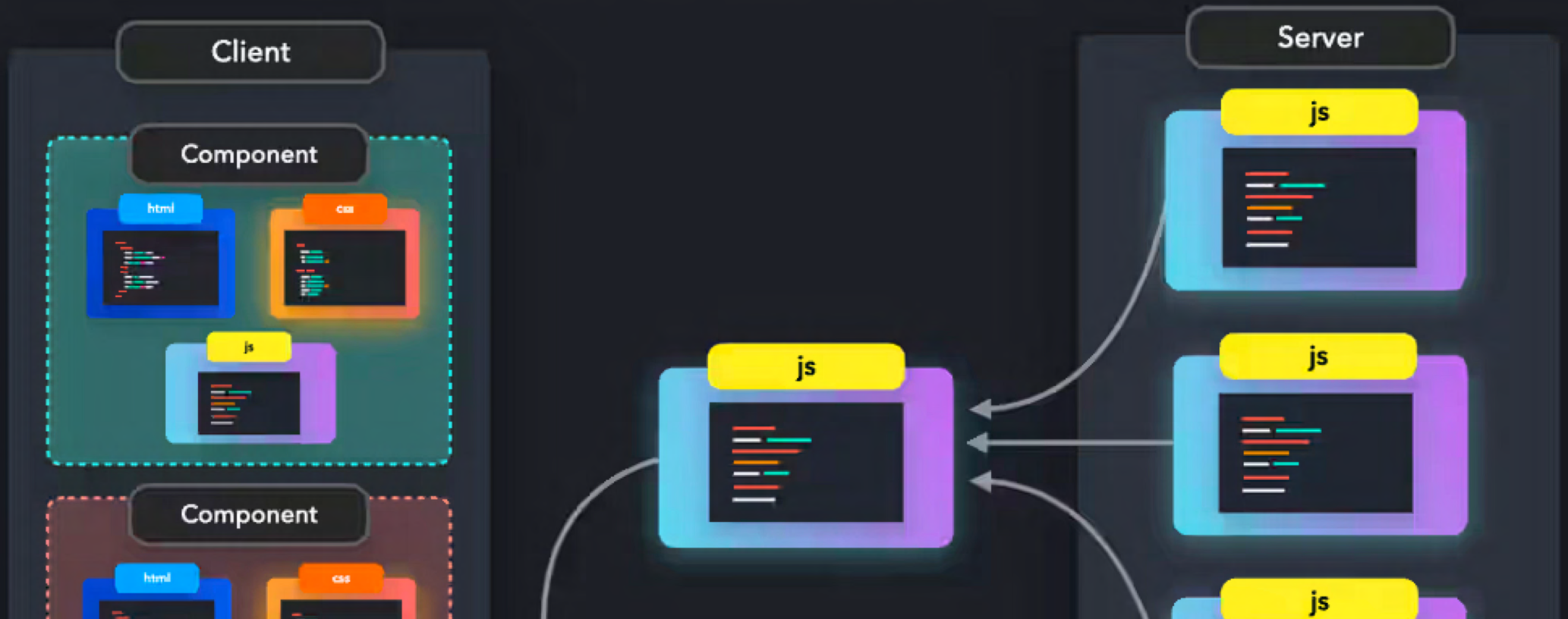
Render all components, hydrate key components first and then progressively hydrate others

Islands Architecture



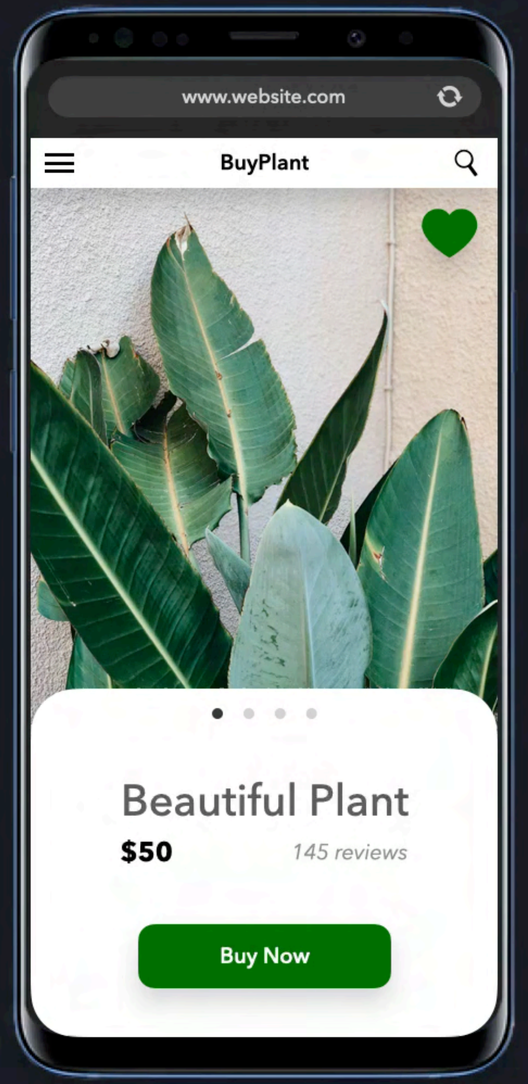
Static components are server rendered HTML. Script is required only for interactive components

PROGRESSIVE HYDRATION



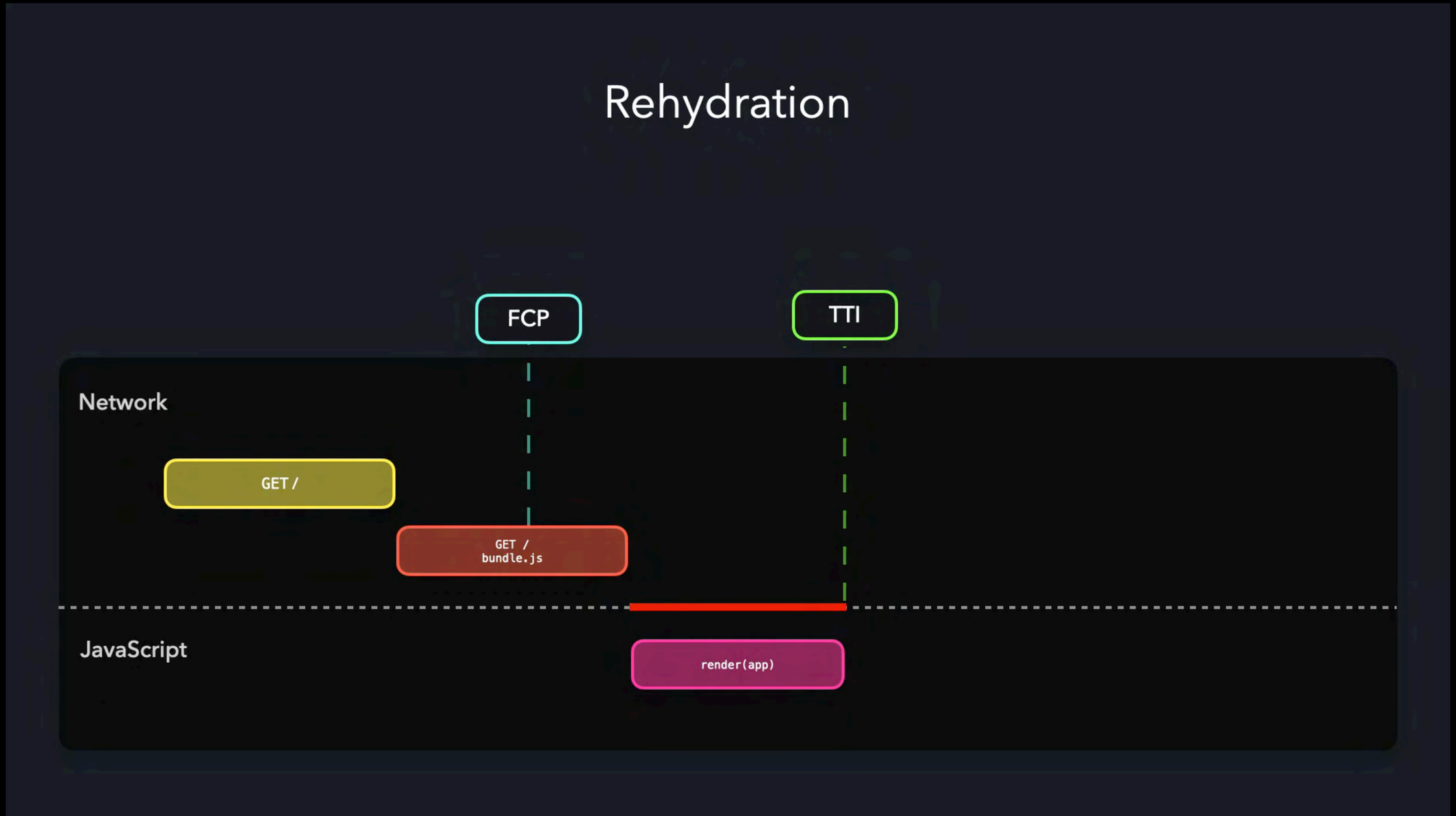
Rehydration

Rehydration



loading...

Progressive Hydration



RESUMABLE HYDRATION

How Resumability Works

Components look familiar:

```
export const Main = component$(() => {
  const state = useStore({
    message: 'hello',
  });

  return (
    <input
      value={state.message}
      onInput$={e => state.message = e.target.value}
    />
  );
});
```

But load in a unique way:

```
<div on:input="./path-to-input-handler.js">
  <input value="hello" />
</div>
```

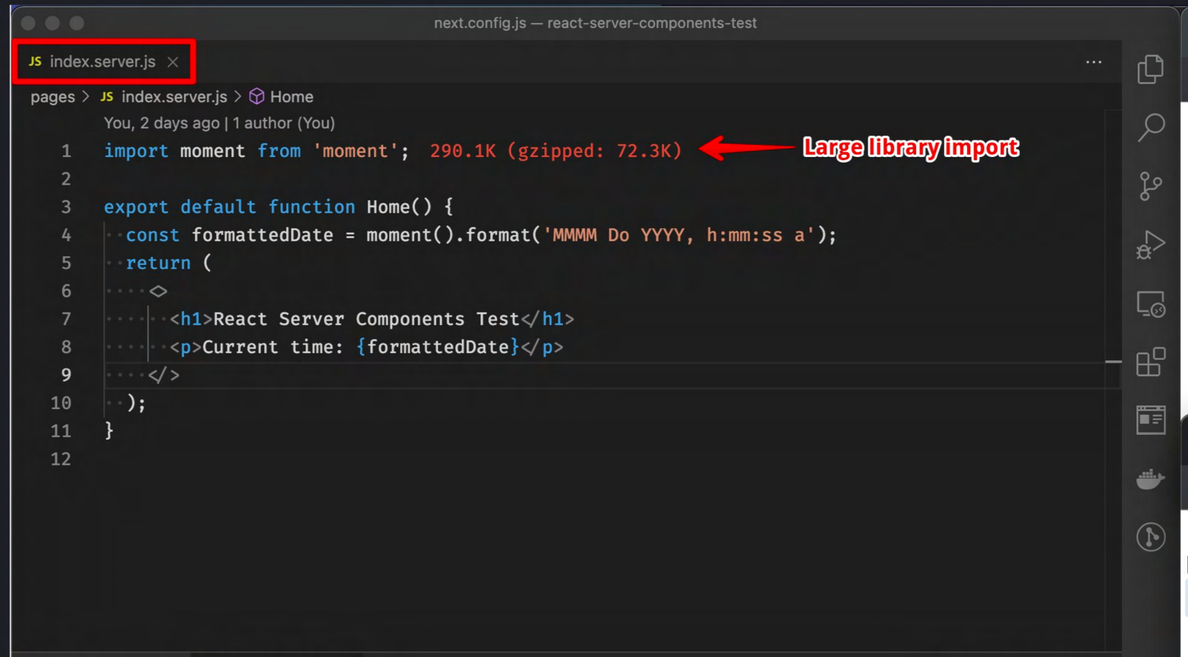
On the server, JS paths are encoded in HTML, so they don't have to download in browser until needed

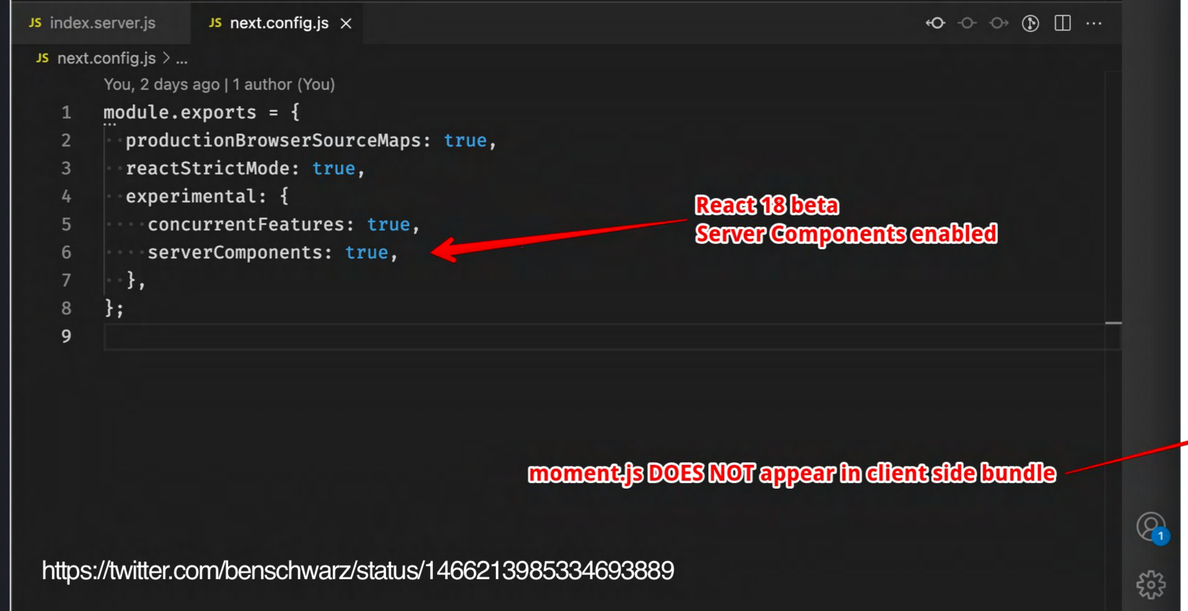
```
<script>
  for (const event of events) {
    document.addEventListener(event, e => {
      const target = e.target.closest(`on:${event}`)
      if (target) {
        const jsPath = target.getAttribute(`[on\\:${event}]`)
        import(jsPath).then(mod => mod.default(e))
      }
    })
  }
</script>
```

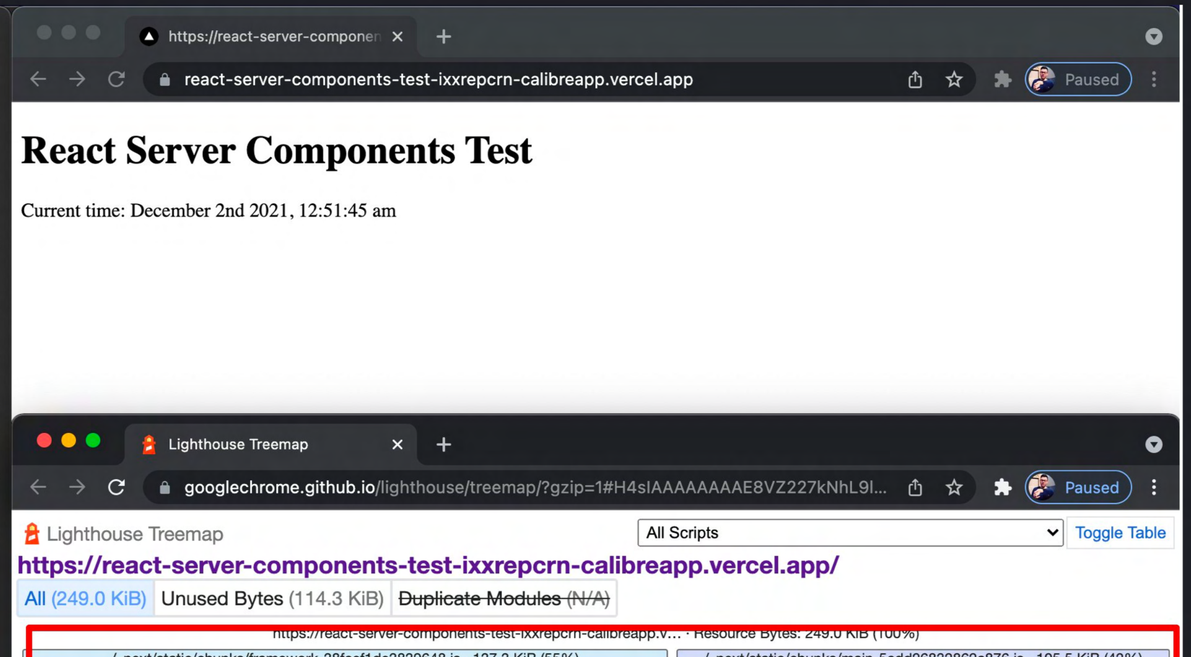
With a tiny bit of code that looks similar to the above, that can be the *only* JS your page needs to become interactive

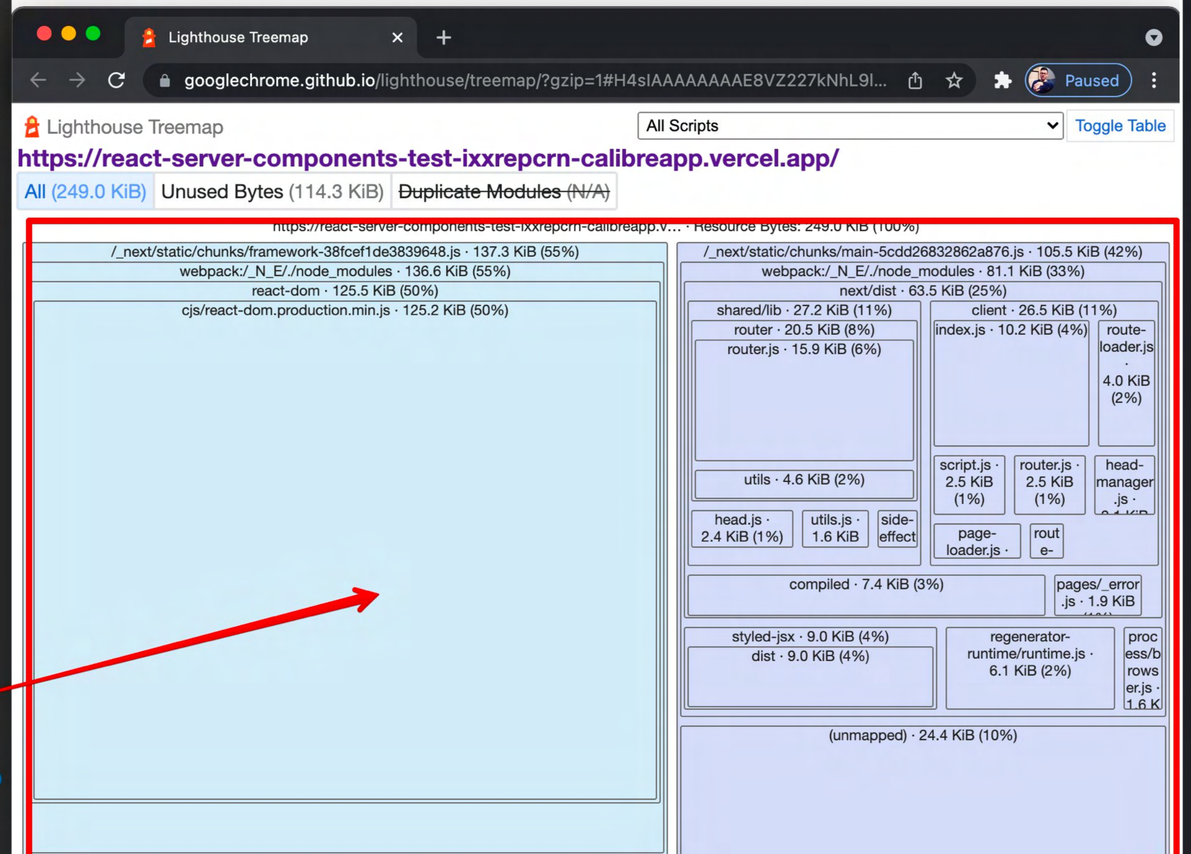
Check out “Hydration is pure overhead” by Misko Hevery

REACT SERVER COMPONENTS



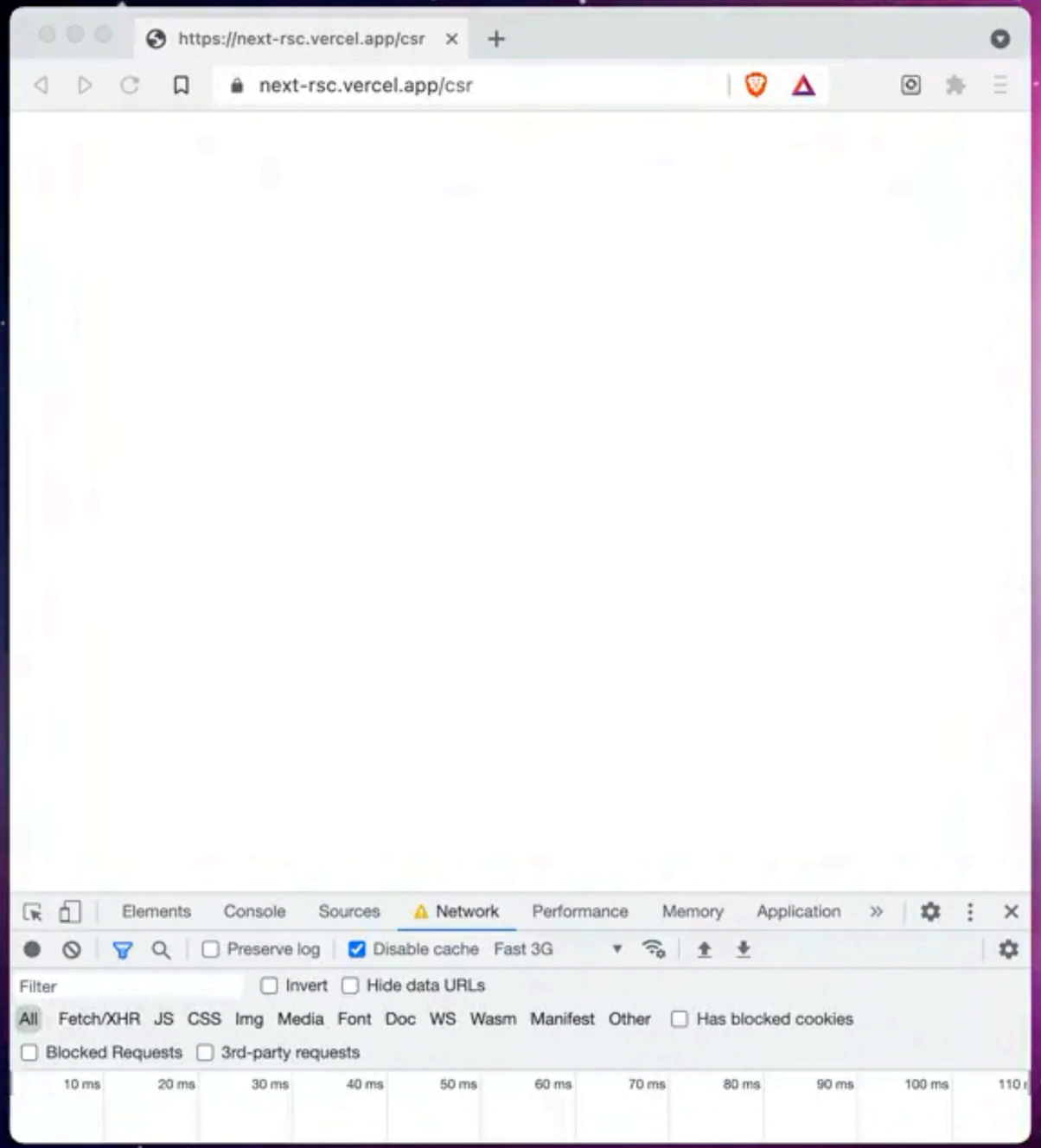


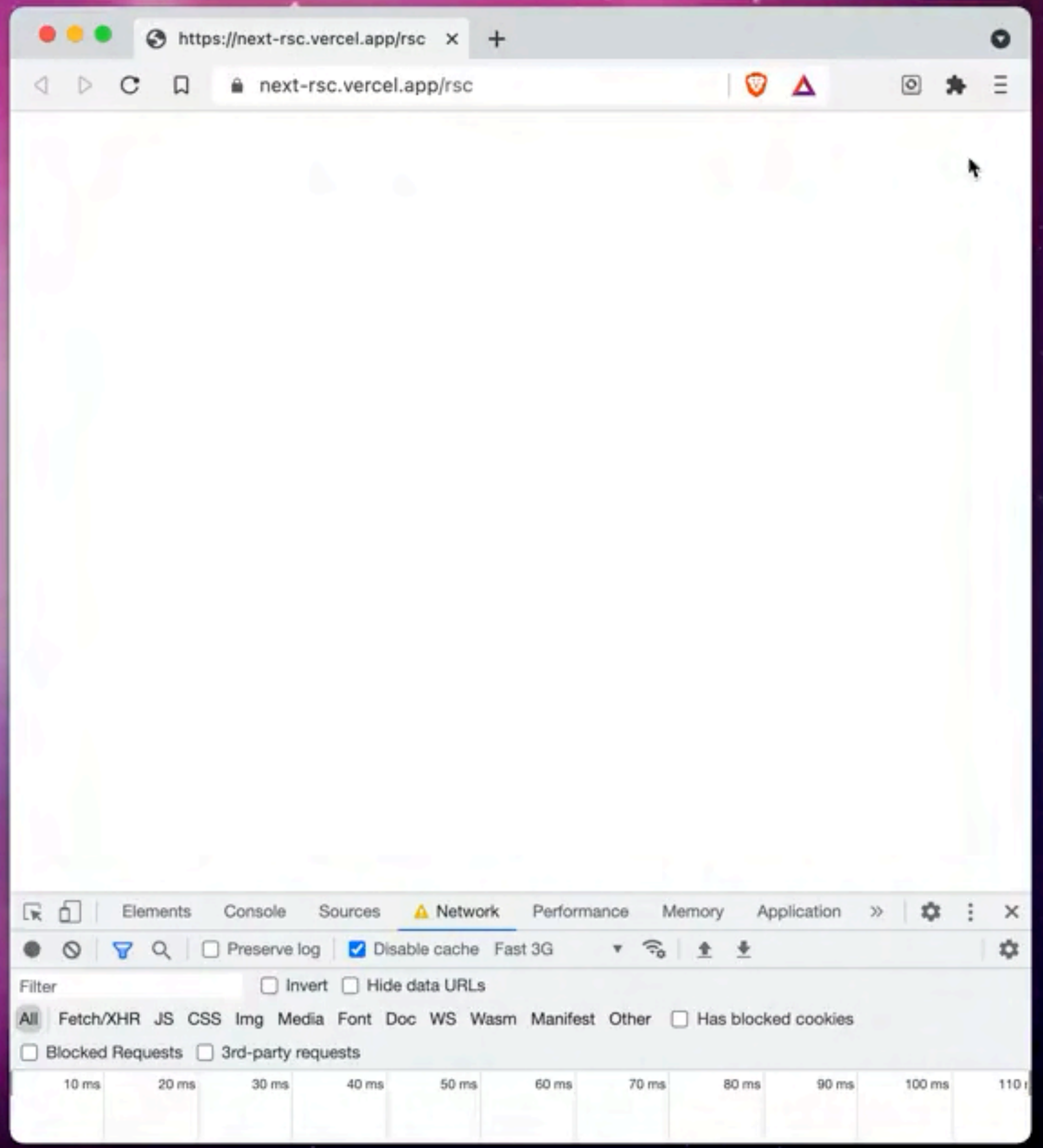




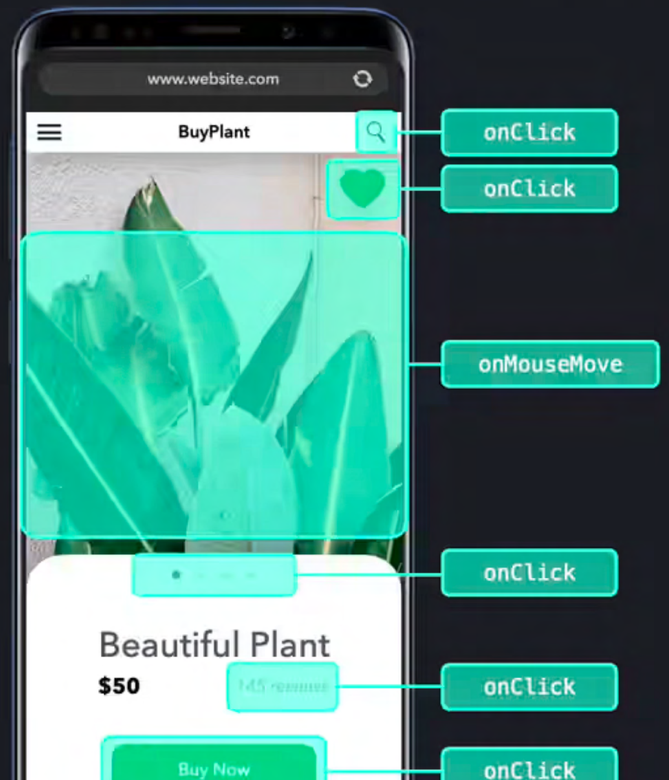
https://twitter.com/benschwarz/status/1466213985334693889

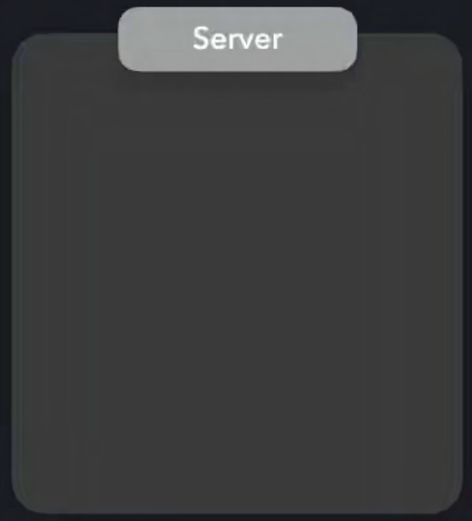
https://twitter.com/shuding_



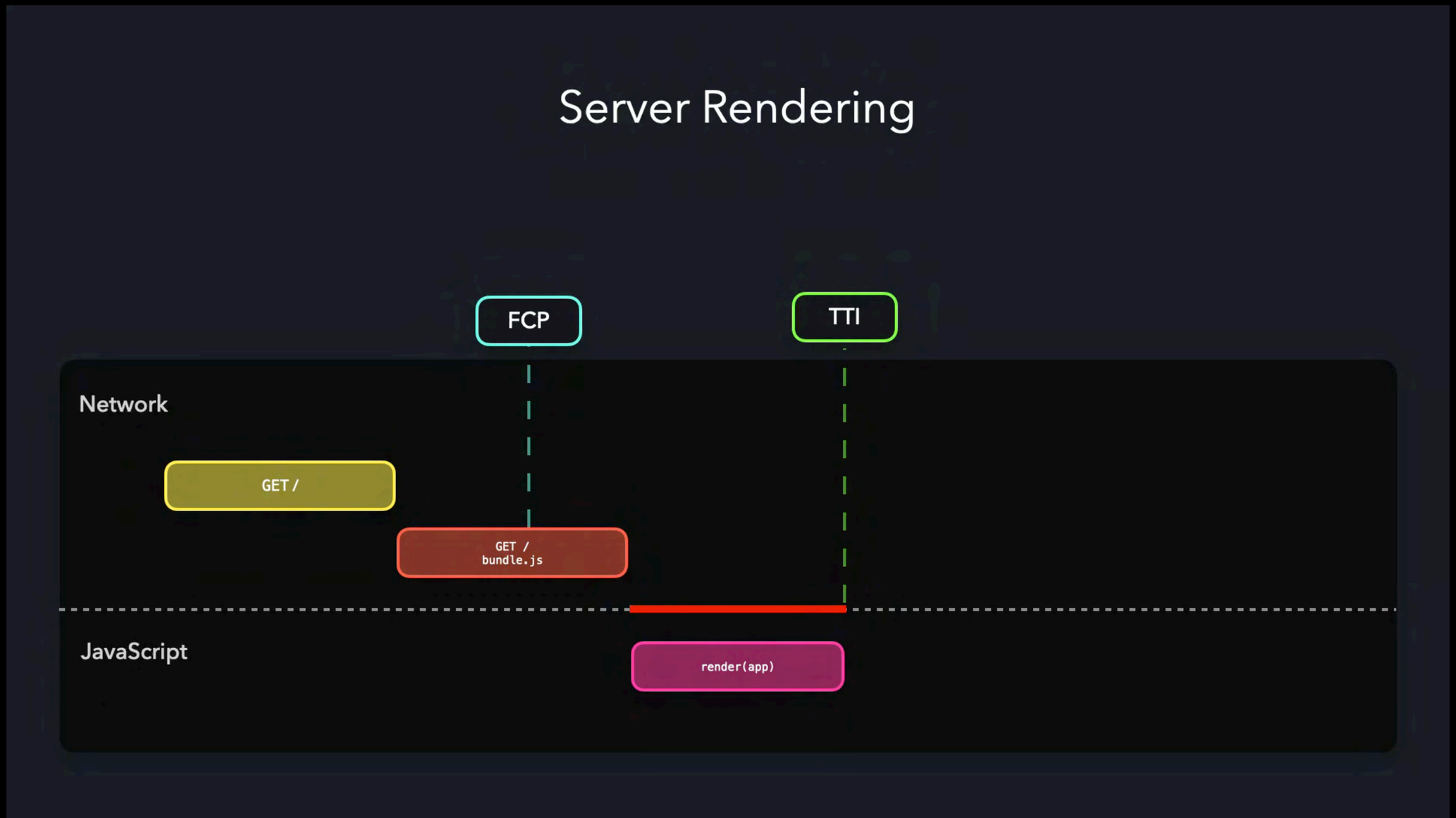


STREAMING SERVER RENDERING

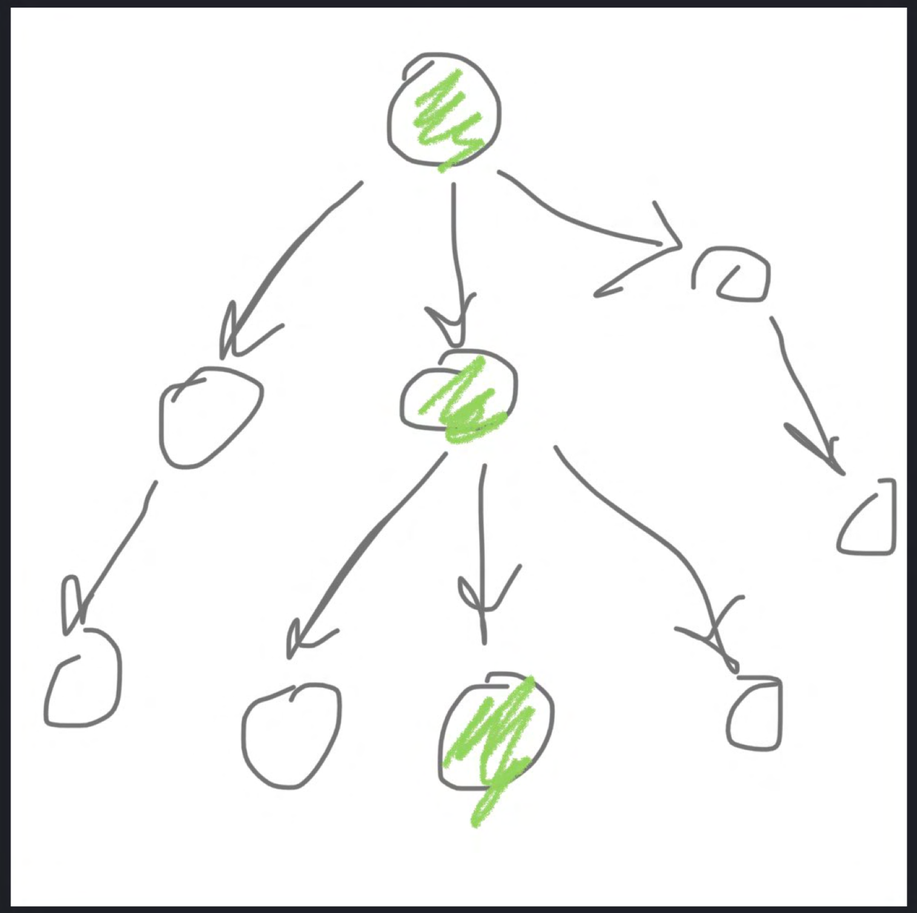


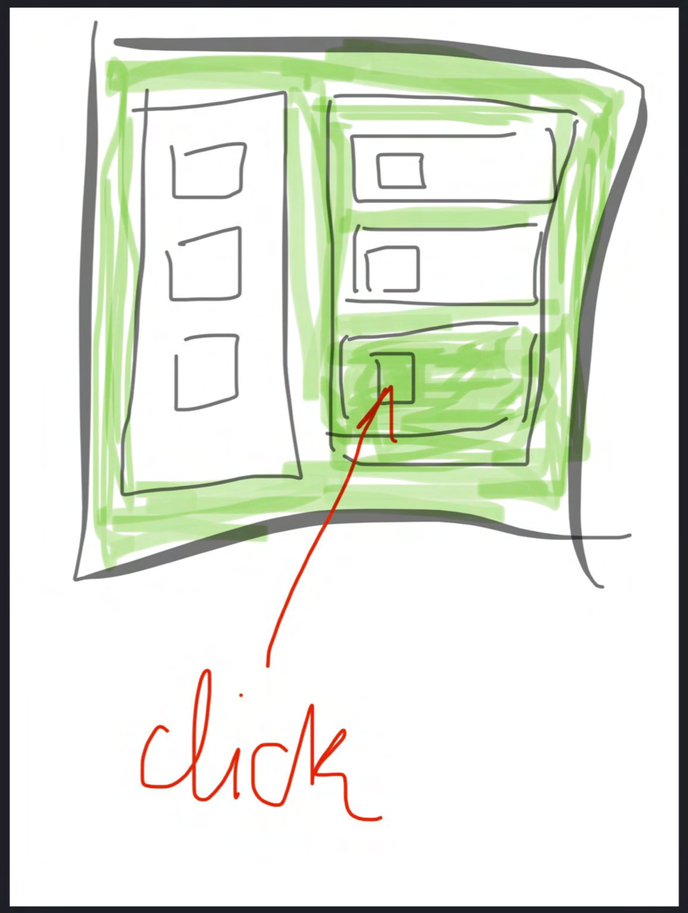


Streaming Server Rendering



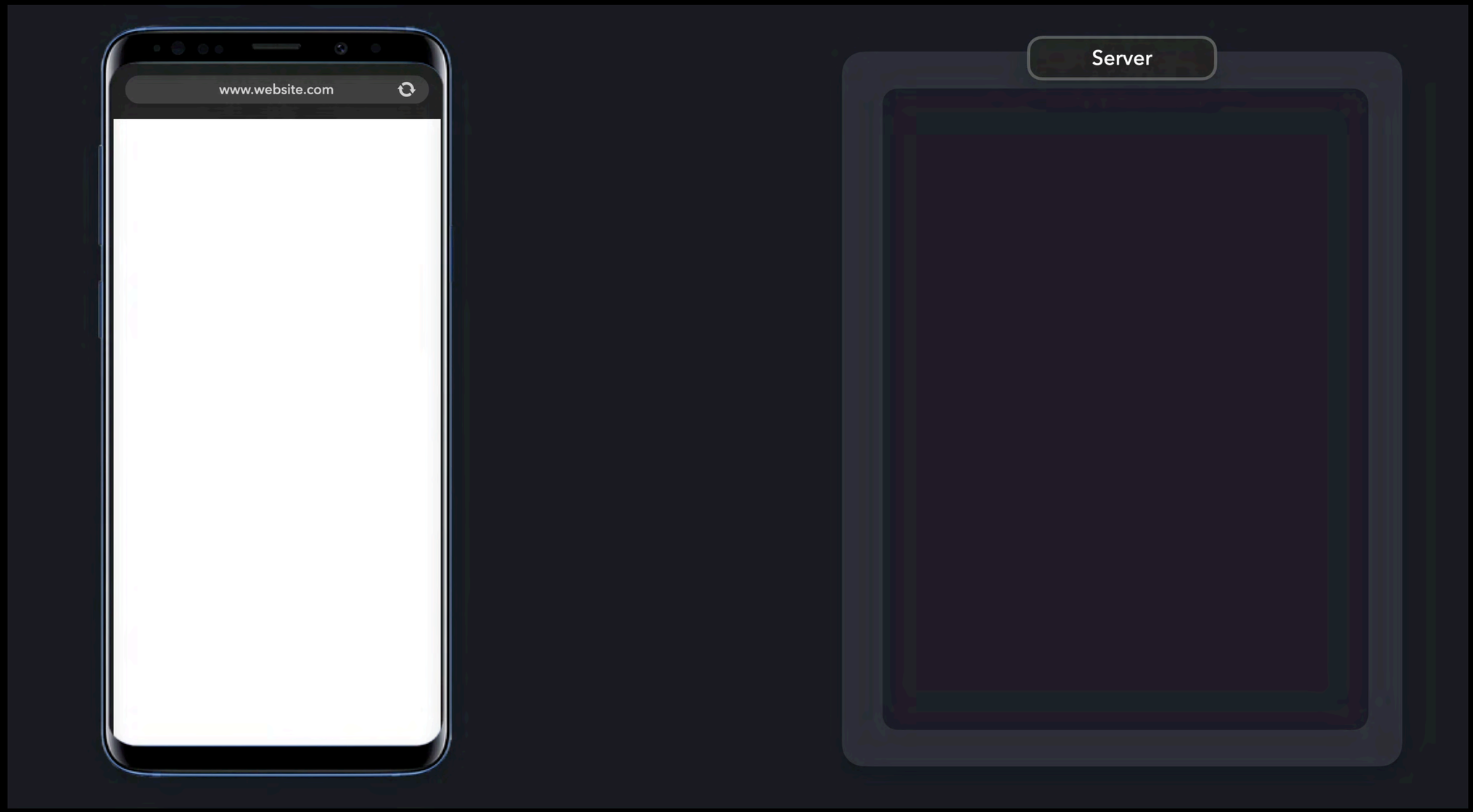
SELECTIVE HYDRATION



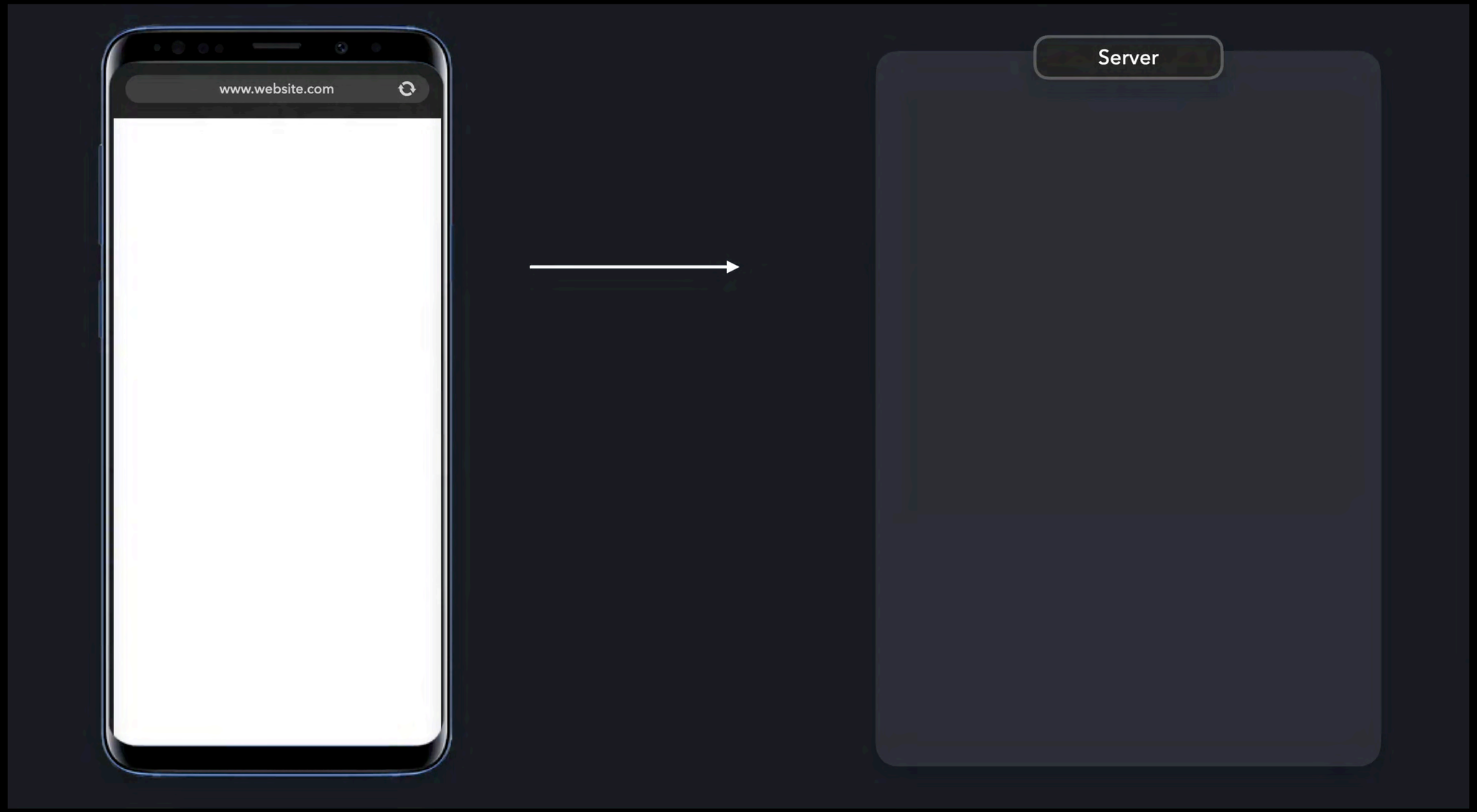


Doodles courtesy Dan Abramov

Selective Hydration: before



Selective Hydration: after



HYDRATE BUT NOT TOO MUCH

IMPROVING PERFORMANCE IS A JOURNEY

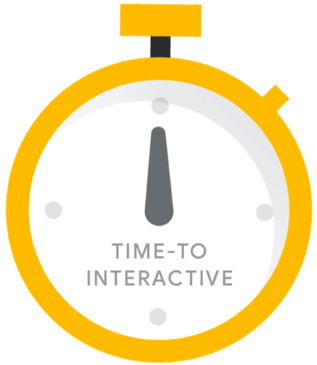

I JUST FIXED THE SITE.
WHY DID YOU RUIN IT?
SMITH
abc

PERF BUDGETS

HOLD THE LINE

PERFORMANCE BUDGET METRICS

< 5s



Milestone

< 170KB

Quantity

Rules

STOP TAKING FAST NETWORKS, CPU & HIGH RAM FOR GRANTED

TEST ON REAL PHONES & NETWORKS

@addyosmani

Learn more over on patterns.dev