THE FUTURE OF WEB DEVELOPMENT

OUT-OF-ORDER STREAMING

OUT-OF-ORDER STREAMING
OUT-OF-ORDER STREAMING
OUT-OF-ORDER STREAMING



THIS WIND POWERED CARGO SHIP IS SET TO CHANGE THE WAY WE SHIP THE GOODS ACROSS OCEANS. THE MODEL IS VERY PRACTICAL AND IS LOOKING AT A POSSIBLE LAUNCH IN 2024. HTTPS://IFT.TT/2MKA9PB

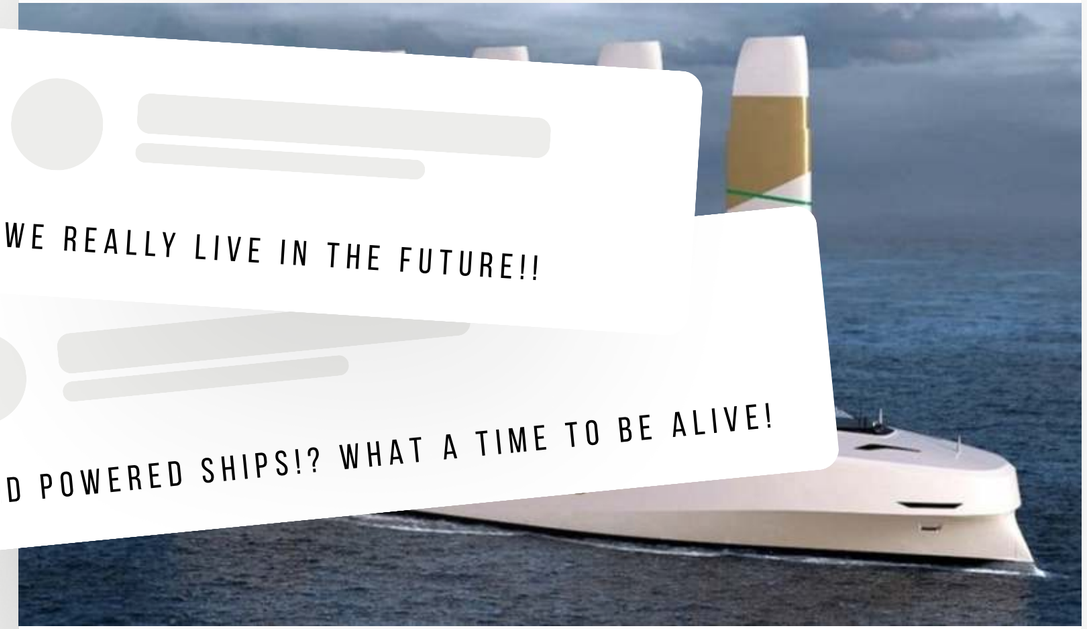


WIND POWERED SHIPS!? WHAT A TIME TO BE ALIVE!

THIS WIND POWERED CARGO SHIP IS SET TO CHANGE THE WAY WE SHIP THE GOODS ACROSS OCEANS. THE MODEL IS VERY PRACTICAL AND IS LOOKING AT A POSSIBLE LAUNCH IN 2024. HTTPS://IFT.TT/2MKA9PB

WE REALLY LIVE IN THE FUTURE!!

WIND POWERED SHIPS!? WHAT A TIME TO BE ALIVE!



THIS WIND POWERED CARGO SHIP IS SET TO CHANGE THE WAY WE SHIP THE GOODS ACROSS OCEANS. THE MODEL IS VERY PRACTICAL AND IS LOOKING AT A POSSIBLE LAUNCH IN 2024. HTTPS://IFT.TT/2MKA9PB


WE REALLY LIVE IN THE
WIND POWERED SHIPS!? W
THIS WIND POWER
THE WAY WE SHIP
IS VERY PRACTICA
IN 2024. HTTPS://IFT.T/2MKA9PB
It is evolving. Just backwards.

A BRIEF HISTORY OF WEB RENDERING

SERVER

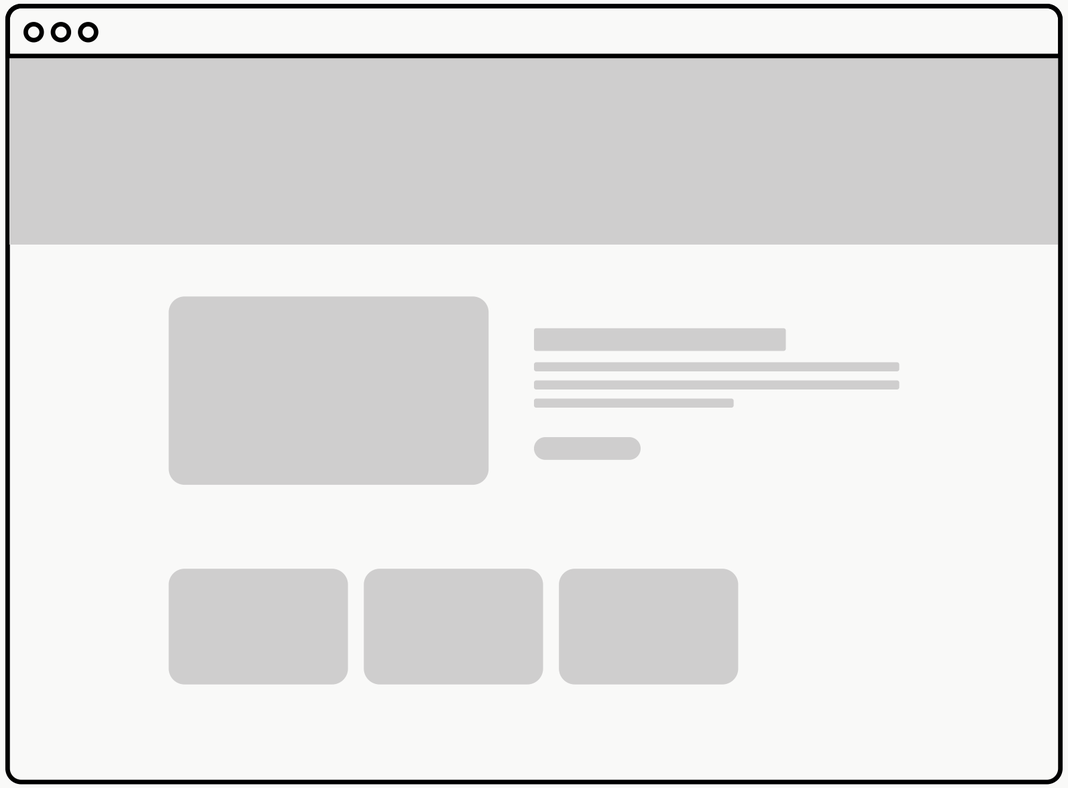
CLIENT

HTML & PHP

BROWSER REQUESTS PAGE



SERVER RESPONSE



SERVER

CLIENT

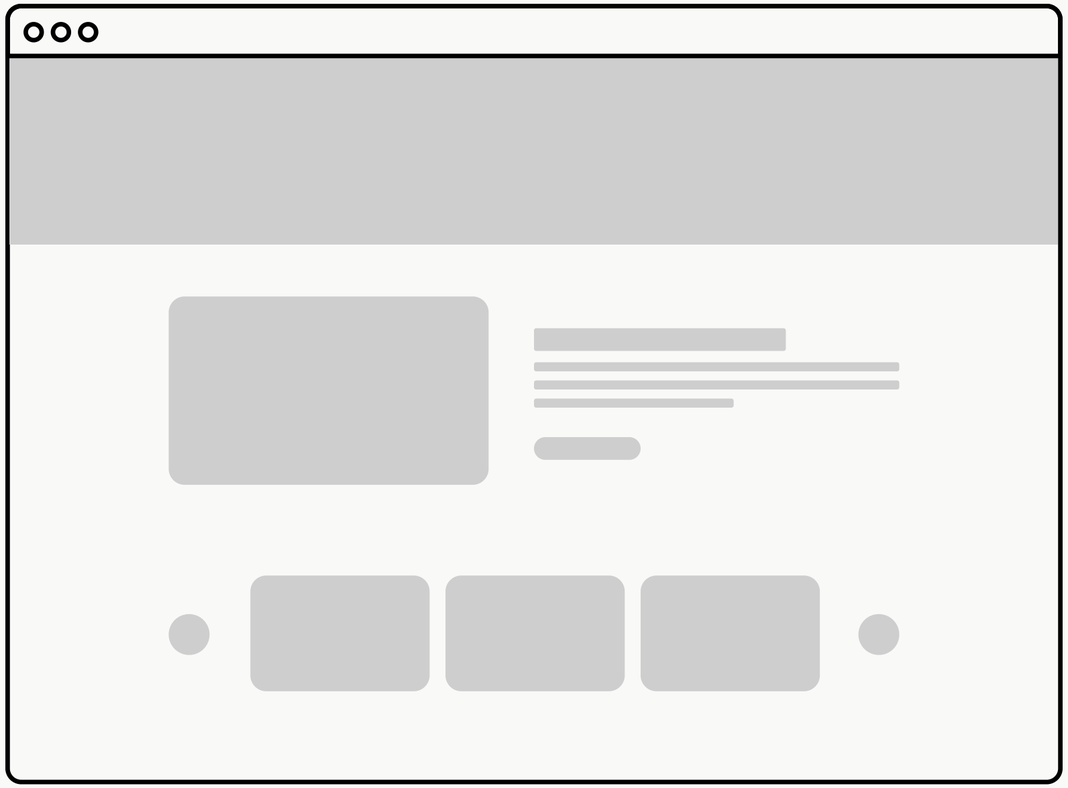
HTML & PHP

JQUERY
& AJAX

BROWSER REQUESTS PAGE



SERVER RESPONSE



JAVASCRIPT LOADED



SERVER

CLIENT

HTML & PHP

JQUERY
& AJAX

SPA

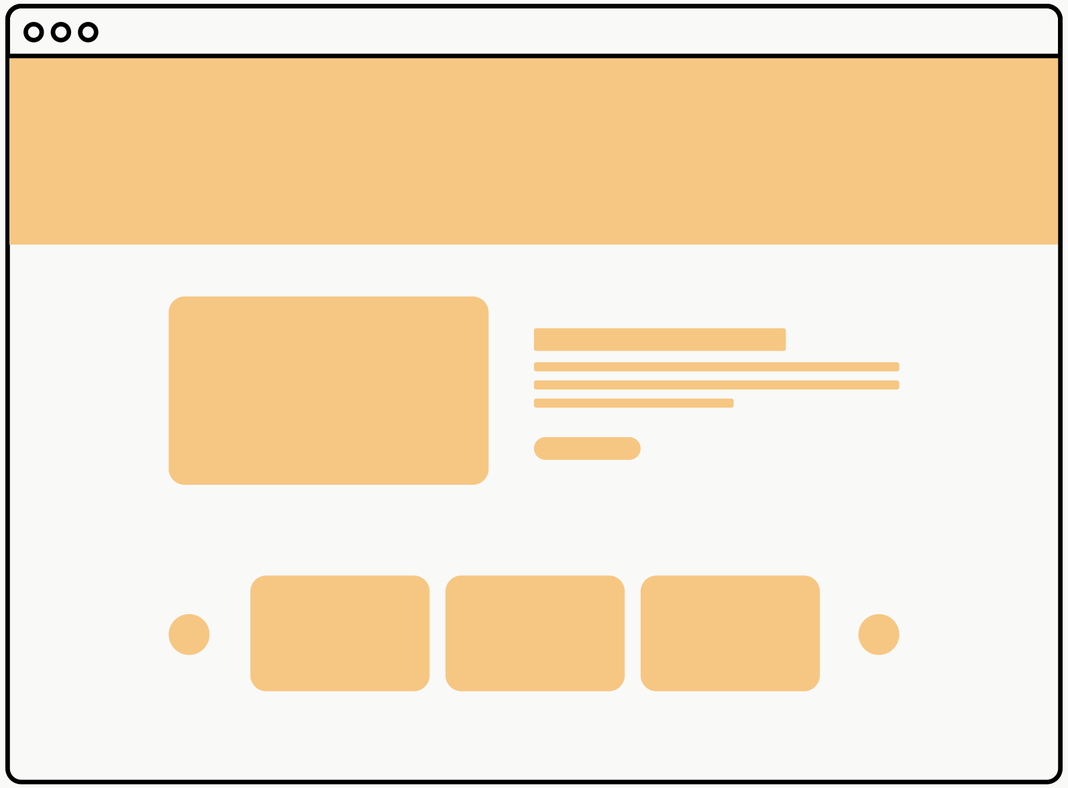
BROWSER REQUESTS PAGE

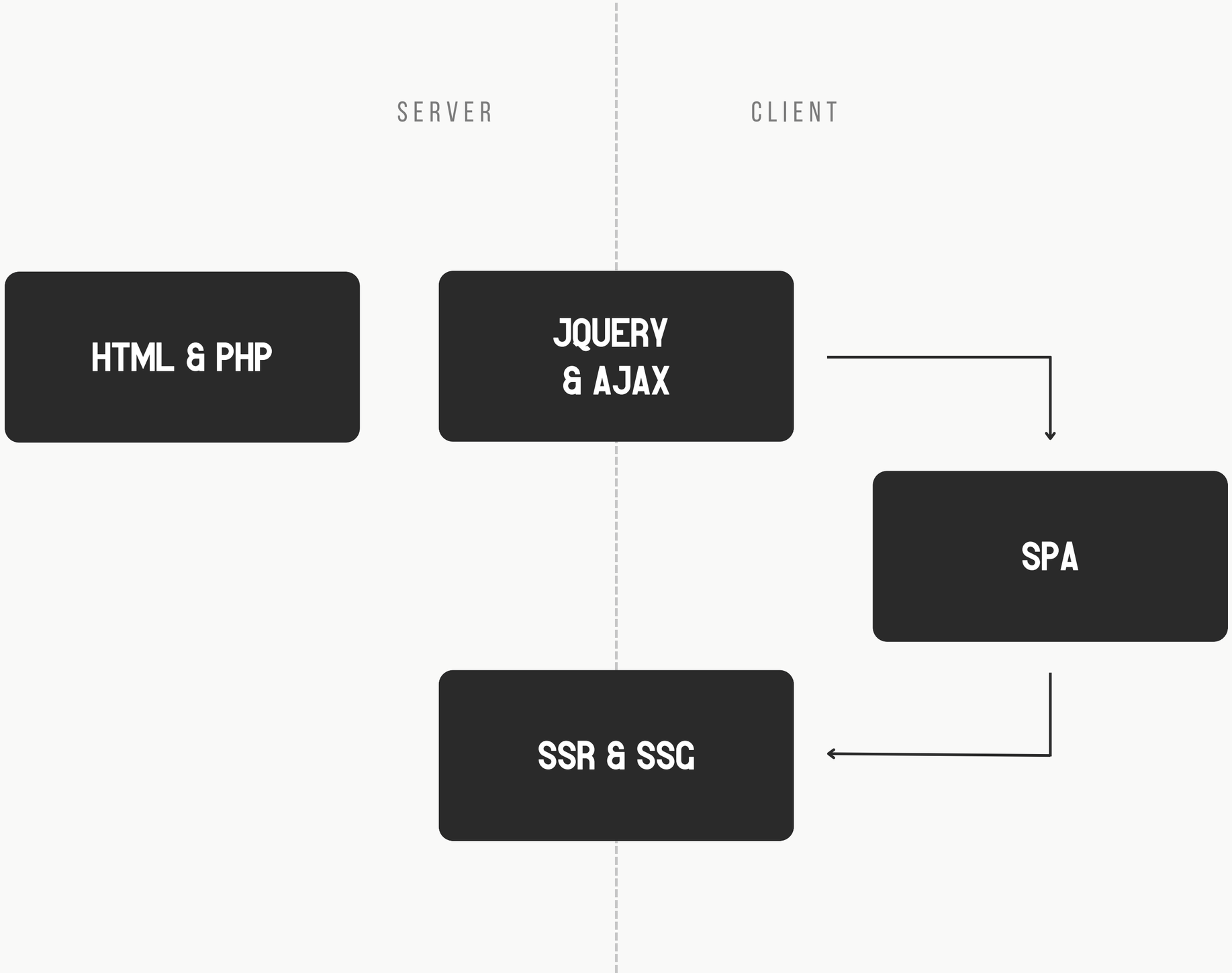


SERVER RESPONSE



JAVASCRIPT LOADED

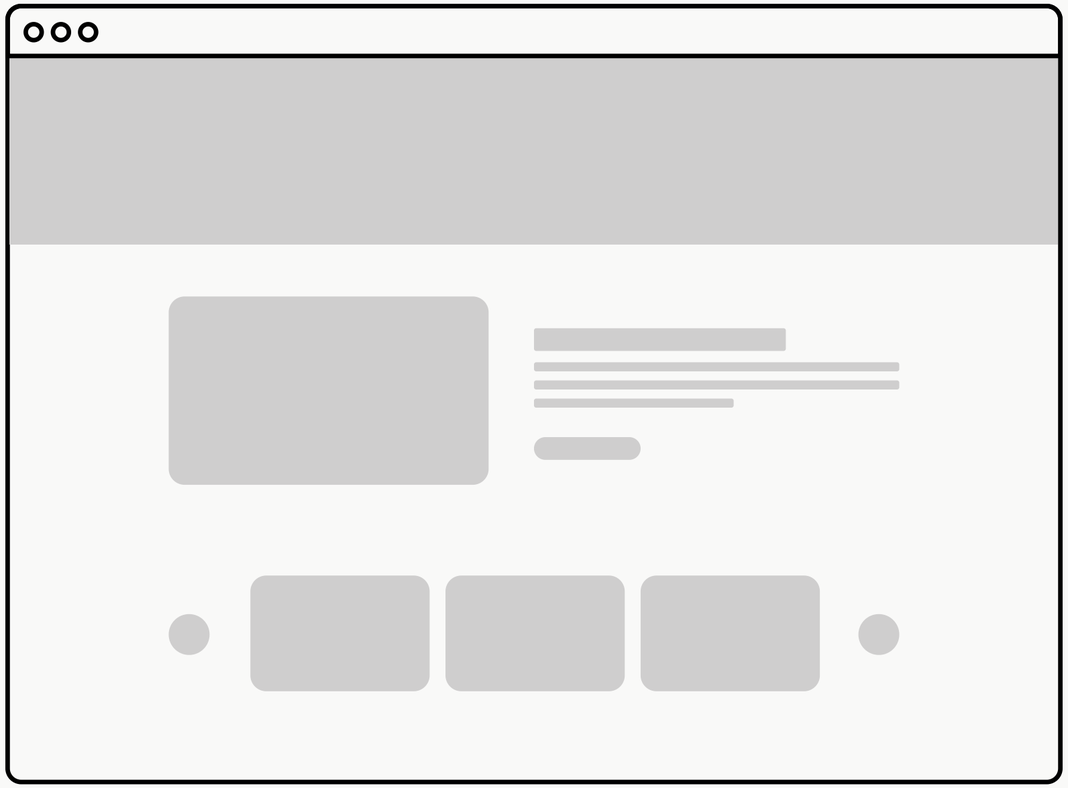



SERVER
CLIENT
HTML & PHP
JQUERY
& AJAX
SPA
SSR & SSG

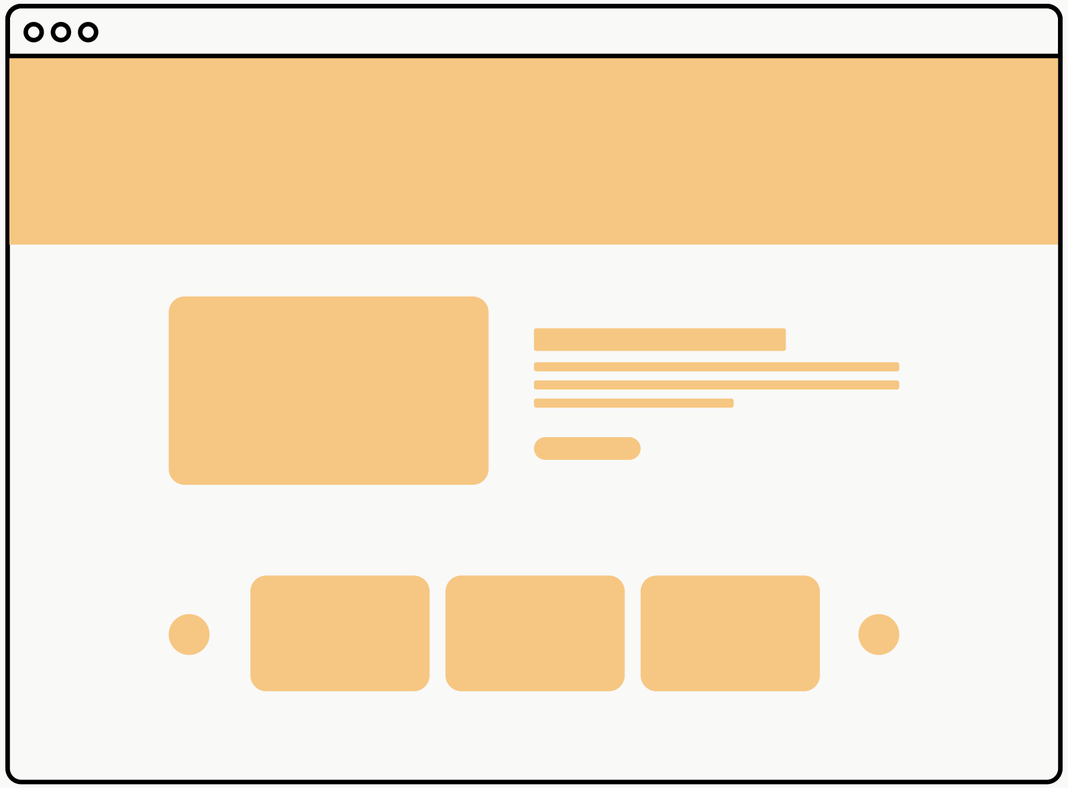
BROWSER REQUESTS PAGE



SERVER RESPONSE



JAVASCRIPT LOADED



SERVER

CLIENT

HTML & PHP

JQUERY
& AJAX

SPA

SSR & SSG

HYDRATION IS
SLOW 🐌

SERVER

CLIENT

HTML & PHP

JQUERY
& AJAX

SPA

SERVER
COMPONENTS

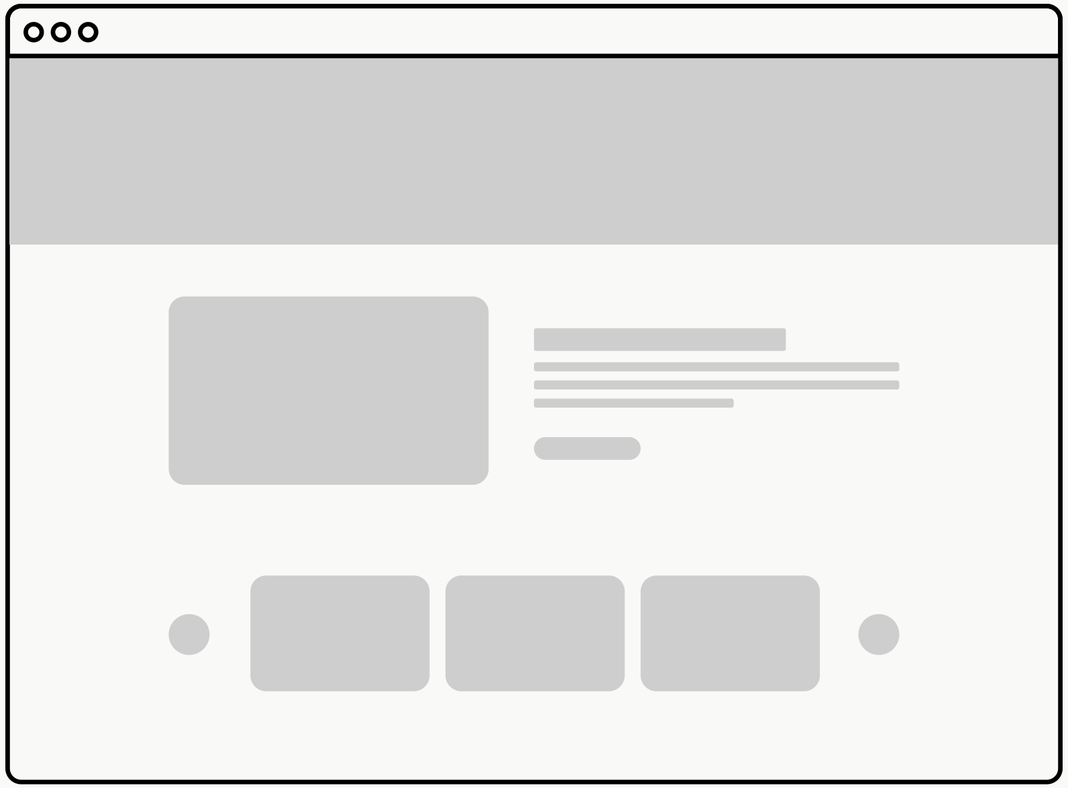
SSR & SSG

HYDRATION IS
SLOW 🐌

BROWSER REQUESTS PAGE

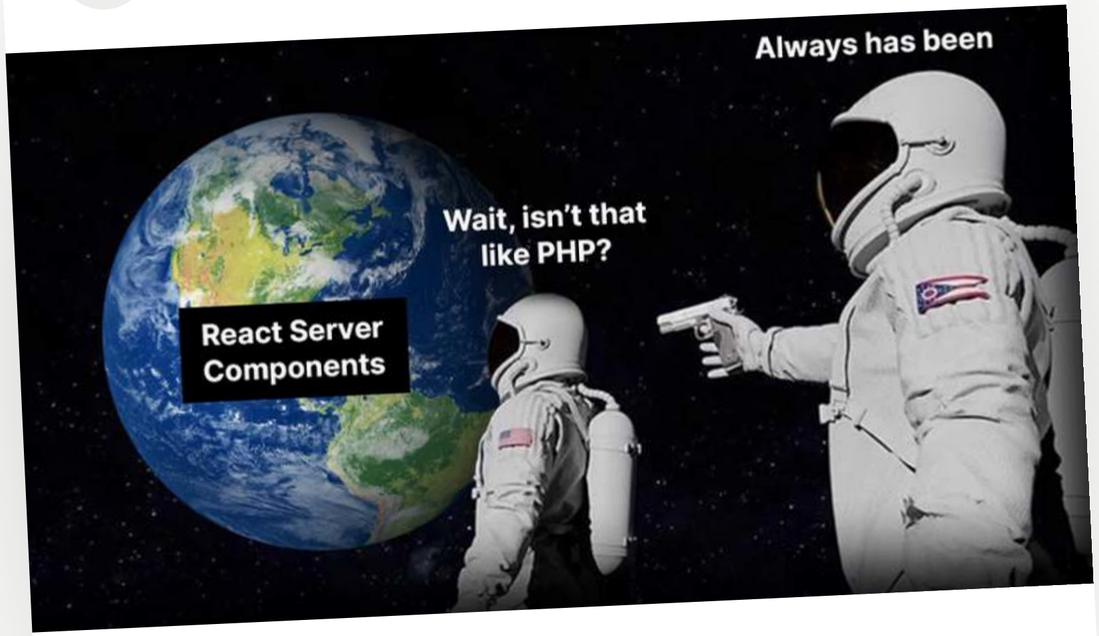


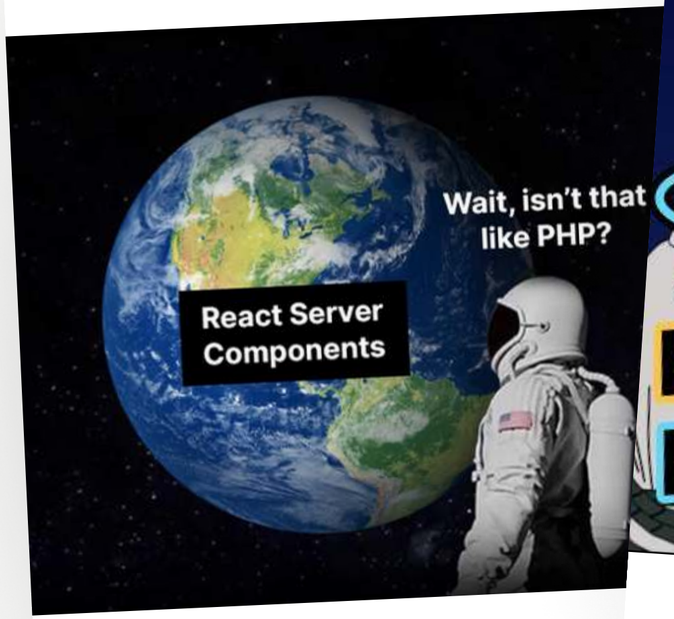
SERVER RESPONSE

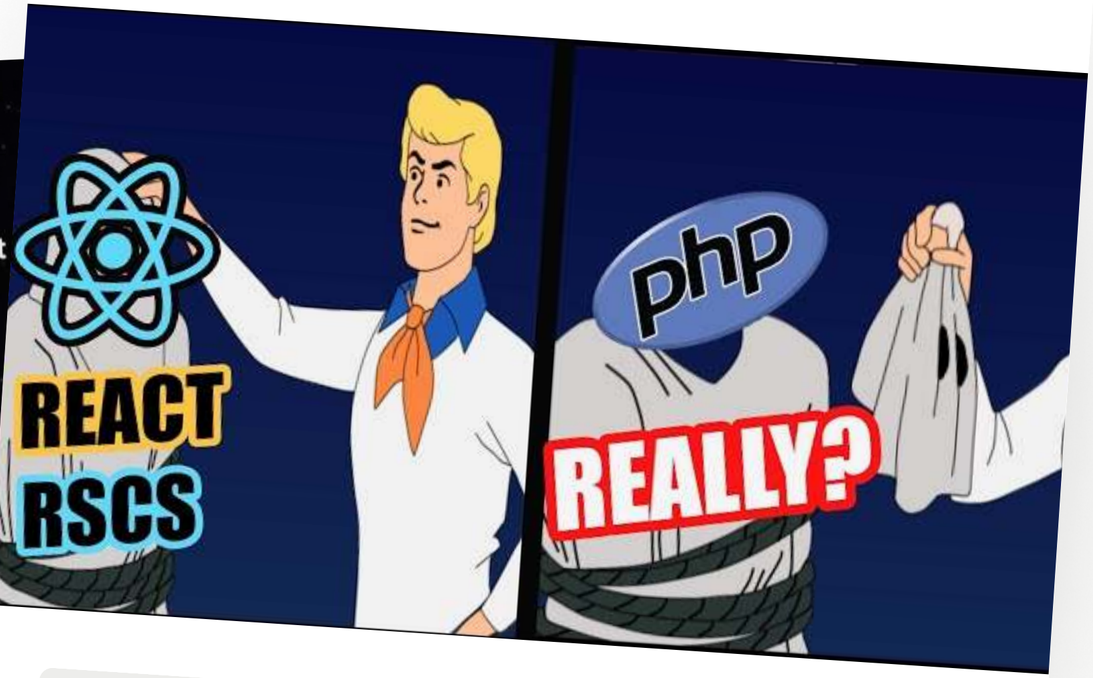


JAVASCRIPT LOADED




Always has been
Wait, isn't that like PHP?
React Server Components


Wait, isn't that like PHP?
React Server Components


REACT
RSCS
php
REALLY?


React S
Compor


It is evolving. Just backwards.


P
LY?

SERVER

CLIENT

HTML & PHP

JQUERY
& AJAX

SPA

SERVER
COMPONENTS

SSR & SSG

HYDRATION IS
SLOW 🐌

HOW CAN WE MAKE THINGS FASTER?

HOW CAN WE MAKE THINGS FASTER?



SERVER-SIDE
RENDERING

HOW CAN WE MAKE THINGS FASTER?



SERVER-SIDE
RENDERING



INTRODUCE
STREAMING

HOW CAN WE MAKE THINGS FASTER?



SERVER-SIDE RENDERING



INTRODUCE STREAMING



OUT-OF-ORDER STREAMING

LET'S WRITE SOME CODE

```tsx
import express from "express";
import { renderToString } from "react-dom/server";

const app = express();
app.use(express.static("out"));

app.get<{}, any, any, { id: string }>("/", async (req, res) => {
  const app = (
    <>
      <Header />
      <Title id={req.query.id} />
      <Details id={req.query.id} />
      <Similar id={req.query.id} />
    </>
  );

  let content = `<!doctype html><html><body>`;

  const contents = await Promise.all(
    app.props.children.map(async (child: JSX.Element) => {
      const element = await child.type(child.props);
      return renderToString(element);
    })
  );

  content += contents.join("");
  content += `</body></html>`;

  res.setHeader("Content-Type", "text/html");
  res.send(content);
});

app.listen(process.env.PORT || 3000);
```

YOU CAN FIND ALL CODE EXAMPLES AT

HTTPS://GITHUB.COM/JULIANBURR/TALK-OUT-OF-ORDER-STREAMING

```tsx
import { fetchTitle } from "../api/title.ts";

export async function Title({ id }: { id: string }) {
  const data = await fetchTitle({ id });
  return (
    <section>
      <img src={data.image} />
      <div>
        <h2>{data.name}</h2>
        <p>
          {data.datePublished.substring(0, 4)} · {data.genre.join(", ")}
        </p>
        <p>{data.description}</p>
      </div>
    </section>
  );
}
```

```tsx
import { fetchDetails } from "../api/details.ts";

export async function Details({ id }: { id: string }) {
  const data = await fetchDetails({ id });
  return (
    <section>
      <h3>Cast</h3>
      <div>
        {data.cast.edges.map((edge: any) => (
          <div key={edge.node?.name?.id}>
            <img
              src={edge.node?.name?.primaryImage?.url}
              alt={edge.node?.name?.nameText?.text}
            />
            <div>
              <p>{edge.node?.name?.nameText?.text}</p>
```

YOU CAN FIND ALL CODE EXAMPLES AT

HTTPS://GITHUB.COM/JULIANBURR/TALK-OUT-OF-ORDER-STREAMING

```
// Introduce streaming

res.setHeader("Content-Type", "text/html");
res.write(`<!doctype html><html><body>`);

for (let child of app.props.children) {
  const element = await child.type(child.props);
  res.write(renderToString(element));
}

res.write(`</body></html>`);
res.end();
```

YOU CAN FIND ALL CODE EXAMPLES AT

HTTPS://GITHUB.COM/JULIANBURR/TALK-OUT-OF-ORDER-STREAMING



```jsx
// Introduce suspense

const app = (
  <>
    <Header />
    <Suspense fallback={<TitleSkeleton />}>
      <Title id={req.query.id} />
      <Details id={req.query.id} />
      <Similar id={req.query.id} />
    </Suspense>
  </>
);
```

```jsx
export let suspended: any = {};

let uuid = 0;
export function Suspense({ fallback, children }: any) {
  const id = uuid++;
  suspended[id] = children;

  return (
    <div data-suspense-id={id} style={{ display: 'contents' }}>
      {fallback}
    </div>
  );
}
```

YOU CAN FIND ALL CODE EXAMPLES AT

HTTPS://GITHUB.COM/JULIANBURR/TALK-OUT-OF-ORDER-STREAMING



```
for (let child of app.props.children) {
  const element = await child.type(child.props);
  res.write(renderToString(element));
}

// Actually rendering suspended components

if (Object.keys(suspended).length) {
  res.write(`
    <script>
      // ...
    </script>
  `);

  await Promise.all(
    Object.entries(suspended).map(async ([key, content]: [any, any]) => {
      const children = Array.isArray(content) ? content : [content];
      const contents = await Promise.all(
        children.map(async (child: any) => {
          const element = await child.type(child.props);
          return renderToString(element);
        })
      );
      res.write(`
        <template>${contents.join("")}</template>
        <suspense-content target-id="${key}"></suspense-content>
      `);
      delete suspended[key];
    })
  );
}
```

YOU CAN FIND ALL CODE EXAMPLES AT

HTTPS://GITHUB.COM/JULIANBURR/TALK-OUT-OF-ORDER-STREAMING

```
// Insert the "magic" bit of Javascript that makes out-of-order-streaming work 🎉

res.write(`
  <script>
    window.customElements.define('suspense-content',
      class SuspenseContent extends HTMLElement {
        connectedCallback () {
          const content = this.previousElementSibling.content;
          const id = this.getAttribute('target-id');
          const target = document.querySelector('[data-suspense-id="' + id + '"]');

          target.innerHTML = '';
          while (content.firstChild) {
            target.appendChild(content.firstChild);
          }
        }
      }
    );
  </script>
`);
```

YOU CAN FIND ALL CODE EXAMPLES AT

HTTPS://GITHUB.COM/JULIANBURR/TALK-OUT-OF-ORDER-STREAMING

DECLARITIVE PARTIAL UPDATES

HTTPS://DEVELOPER.CHROME.COM/BLOG/DECLARATIVE-PARTIAL-UPDATES

```
<main>
  <?start name="details">
    <p>Loading...</p>
  <?end>
</main>

<template for="details">
  <p>The actual loaded <b>content</b></p>
</template>
```

GOOD RESOURCES TO GET STARTED

HTTPS://REACT.DEV/REFERENCE/REACT/SUSPENSE

REACT DOCS — SUSPENSE (ON THE CLIENT)

HTTPS://GITHUB.COM/REACTJS/RFCS/BLOB/MAIN/TEXT/0213-SUSPENSE-IN-REACT-18.MD

RFC DOCUMENT FOR SUSPENSE ON THE SERVER

HTTPS://WWW.YOUTUBE.COM/WATCH?V=MAEBEQHZR84

BEN HOLMES — REACT SERVER COMPONENTS FROM SCRATCH

HTTPS://GAL.HAGEVER.COM/POSTS/OUT-OF-ORDER-STREAMING-FROM-SCRATCH

GAL SCHLEZINGER — OUT OF ORDER STREAMING FROM SCRATCH



https://github.com/julianburr/talk-out-of-order-streaming

https://www.julianburr.de/fullstackconf-2026-slides.pdf

https://bsky.app/profile/julianburr.de

https://linkedin.com/in/julianburr

https://github.com/julianburr