

BACK TO THE SERVER III

JS

Server

Part 3



ABOUT ME

Jonas Bandi

jonas.bandi@ivorycode.com

Twitter: [@jbandi](https://twitter.com/@jbandi)



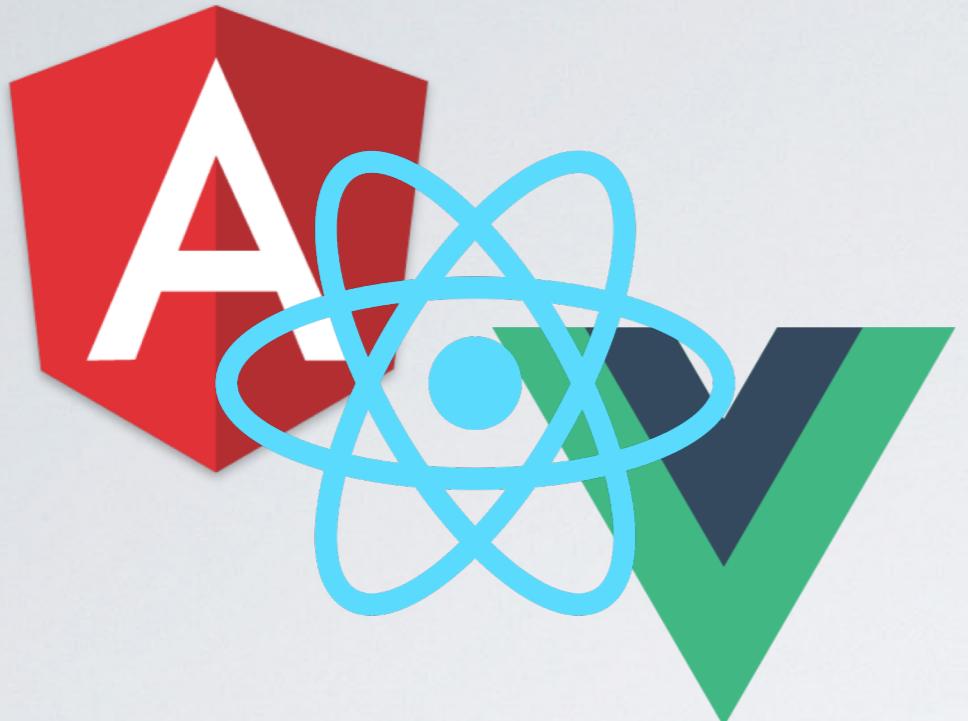
- Freelancer, in den letzten 9 Jahren vor allem in Projekten im Spannungsfeld zwischen modernen Webentwicklung und traditionellen Geschäftsanwendungen.
- Dozent an der Berner Fachhochschule seit 2007
- In-House Kurse & Beratungen zu Web-Technologien im Enterprise: UBS, Postfinance, Mobiliar, AXA, BIT, SBB, Elca, Adnovum, BSI ...



JavaScript / Angular / React / Vue / Vaadin
Schulung / Beratung / Coaching / Reviews
jonas.bandi@ivorycode.com



Reality-Check ...
... who knows this Movie?



Frontend ...

.... On The Server?

Old Hat?



Server Side Rendering (SSR)



... even Angular can do that ...

Using Angular CLI v17

```
> npx @angular/cli@next new ng-ssr
? Which stylesheet format would you like to use? CSS
? Do you want to enable Server-Side Rendering (SSR)
  and Static Site Generation (SSG/PreactRendering)? Yes
```

✓ Packages installed successfully.

```
cd ng-ssr
npm start
```

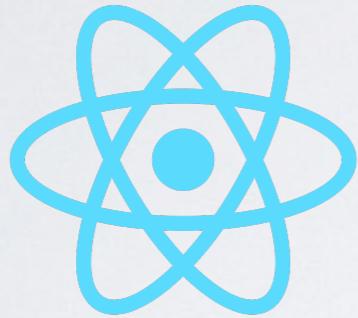
The screenshot shows a browser window with the Angular logo and the title "AwesomeApp". Below it is a "Welcome" page with a "Resources" section containing links to "Learn Angular", "CLI Documentation", "Angular Material", "Angular Blog", and "Angular DevTools". A "Next Steps" section offers options like "New Component", "Angular Material", "Add PWA Support", "Add Dependency", and "Run and Watch Tests". To the right, the browser's developer tools Network tab is open, showing a list of requests. A red box highlights the "localhost" entry in the "Name" column. Another red box highlights the "Response" tab, which displays the raw HTML code of the rendered page. The code includes Angular-specific tags like <app-root>, <ng-content>, and <ng-attr>. At the bottom of the response tab, a message reads: "Angular is running in development mode. Angular hydrated 1 component(s) and 160 node(s), 0 component(s) were skipped. Learn more at ar.io/guide/hydration".

This talk is not about Server Side Rendering ...



(... but I don't tell you the agenda ...)

They say ...



React is Easy!

```
export function Greeter() {  
  return <h1>Hello World!</h1>;  
}
```

The power of React is a Component Model which enables simple & elegant composition ...



Cory House

@housecor

...

I love the simplicity of React's reuse model.

Repeating JSX? Create a component.

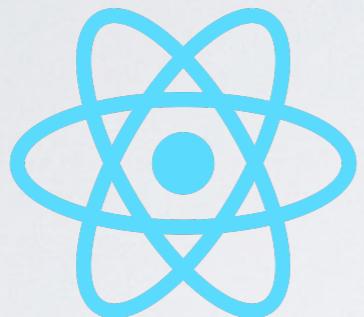
Repeating logic? Create a hook.

I can compose these simple building blocks in infinite ways.

1:08 PM · Nov 25, 2023 · 16.7K Views

<https://twitter.com/housecor/status/1728385239611789758>

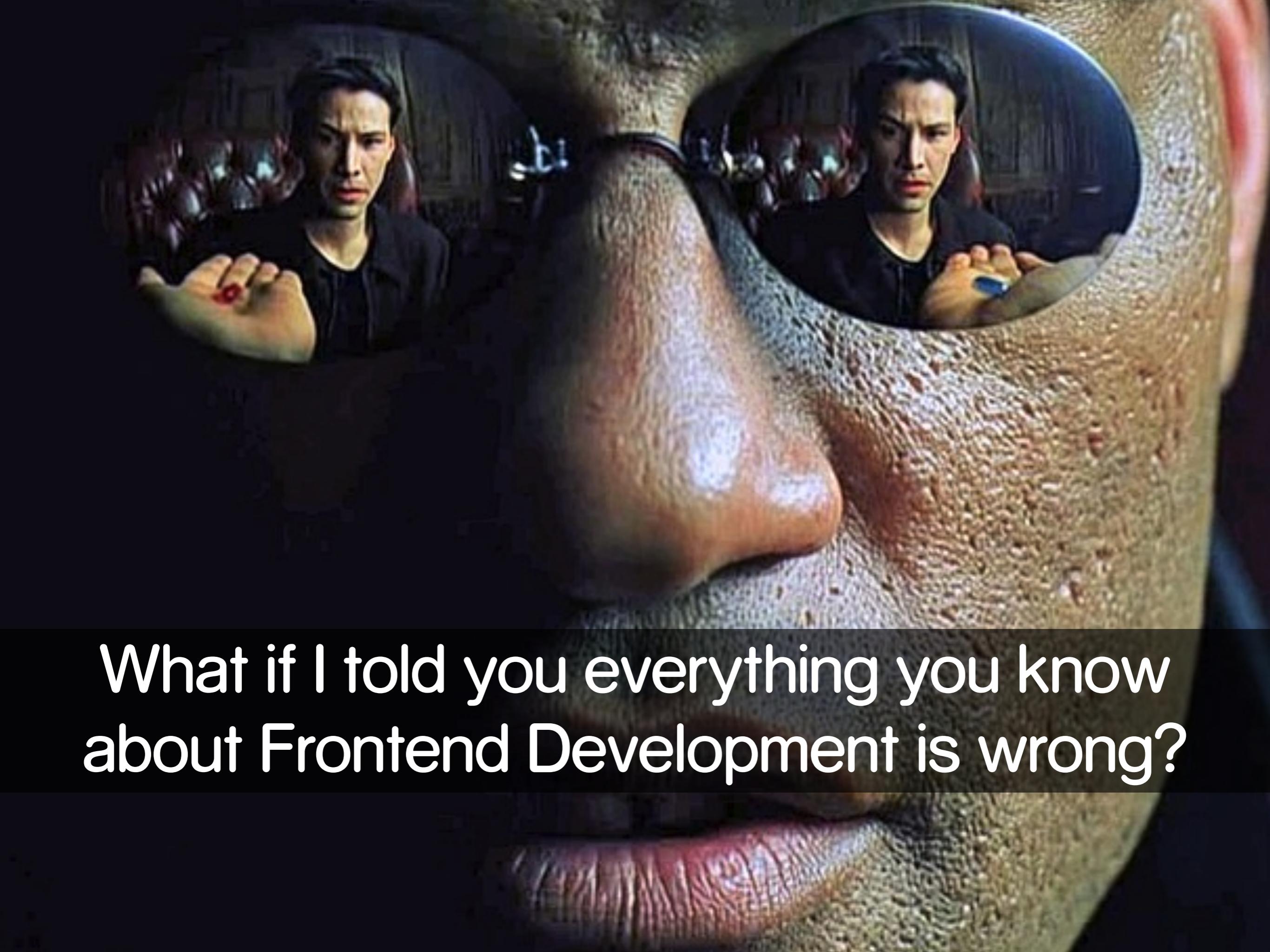
They say ...



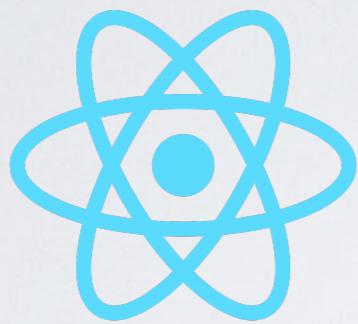
React is Easy!

```
export function Greeter() {  
  return <h1>Hello World!</h1>;  
}
```

... until you start fetching data!



What if I told you everything you know
about Frontend Development is wrong?

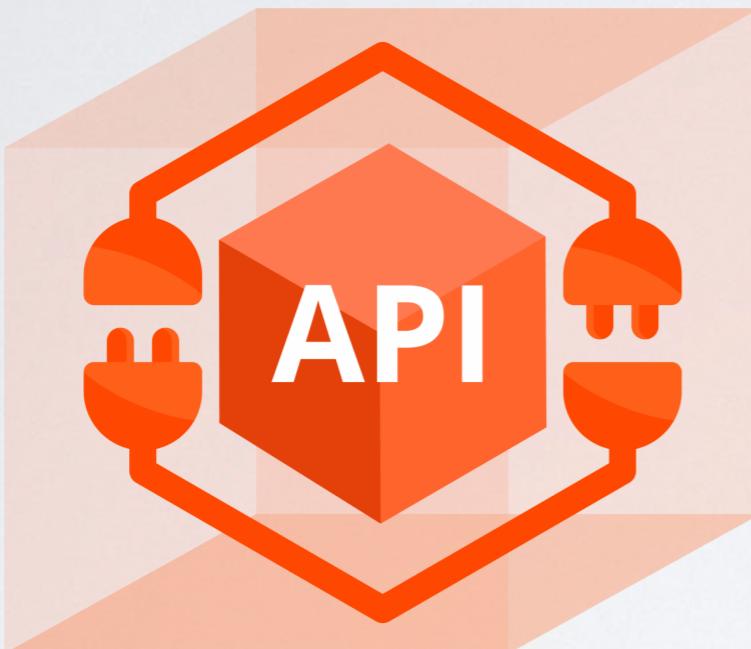


React **is** Easy!

```
export async function Greeter() {  
  const data = await fetchData();  
  return <h1>{data}</h1>;  
}
```

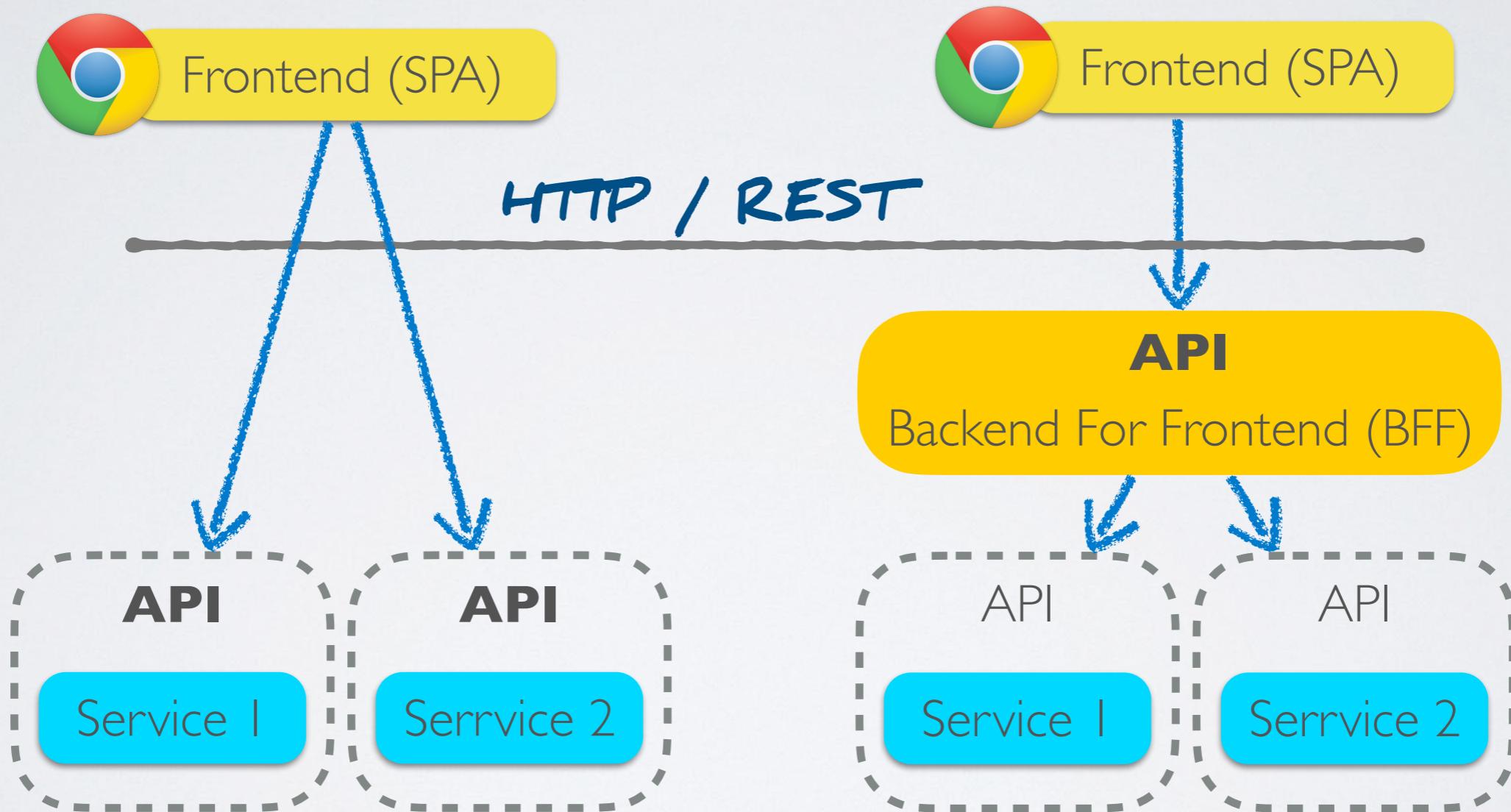


They say ...

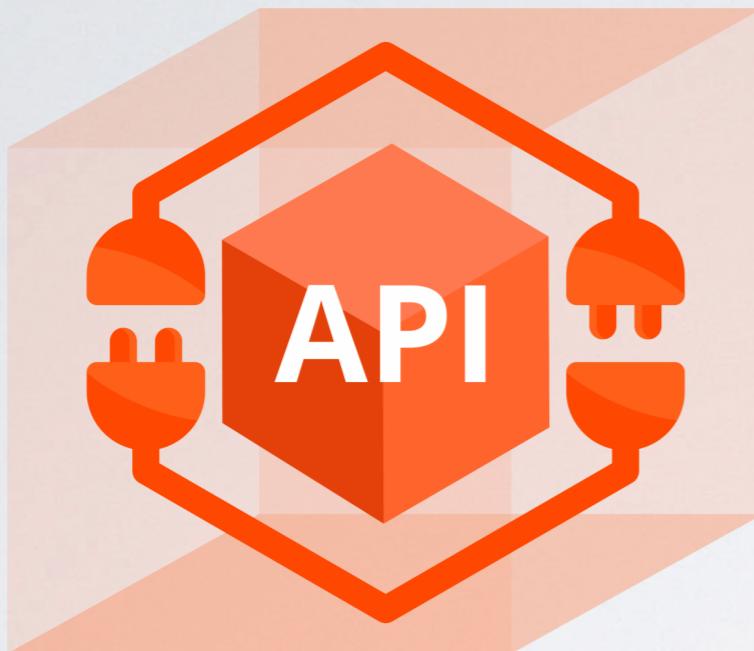


for SPAs thou
must build APIs!

Architecture: APIs for SPAs



They say ...

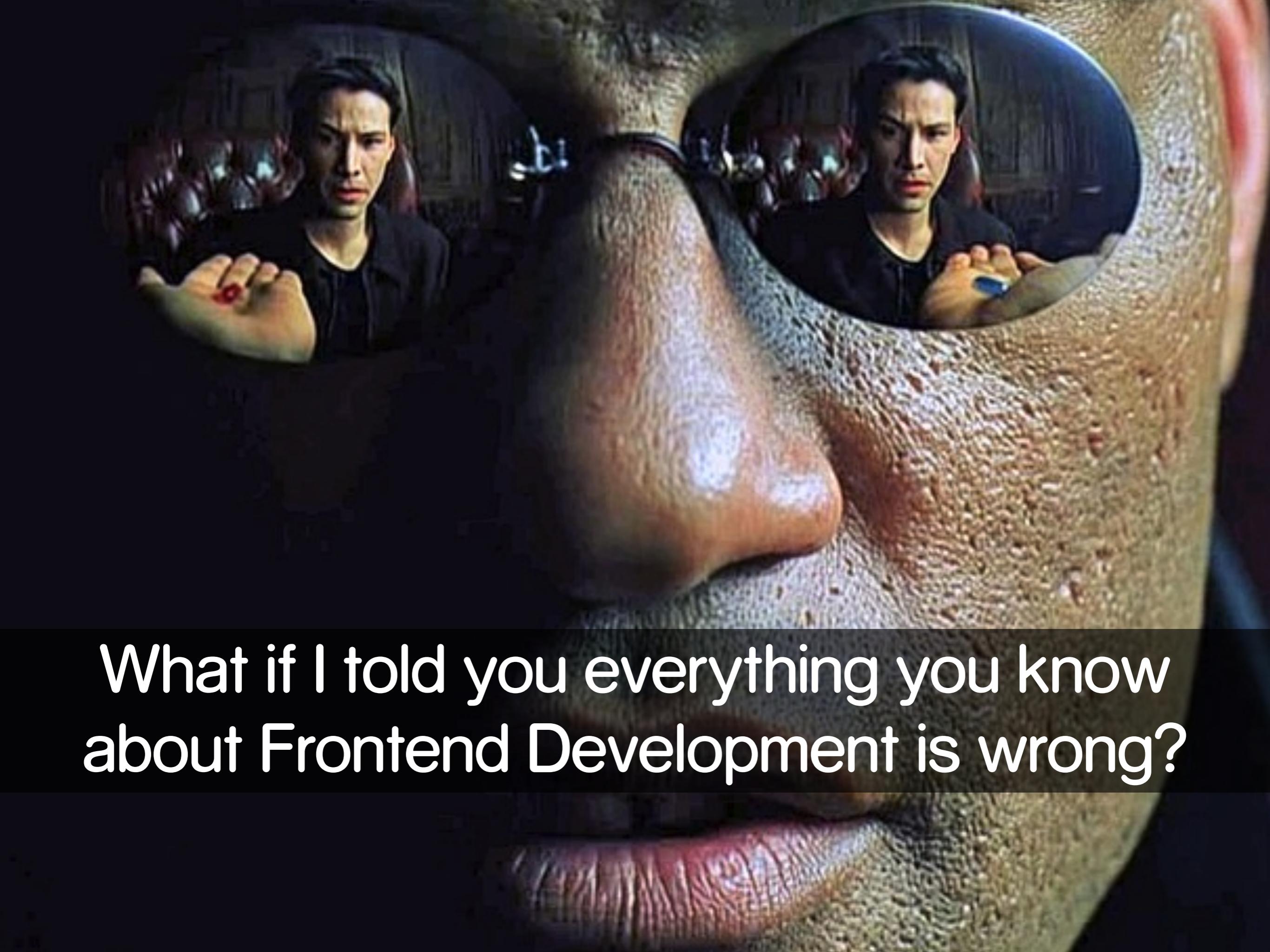


for SPAs thou
must build APIs!

... but maybe you just want access to
server-side data & functionality?

Few organizations are in the business of delivering APIs.

- Stefan Tilkov: Wait, what!? Our microservices have actual human users?
<https://www.youtube.com/watch?v=pUlgXA0rfwc>



What if I told you everything you know
about Frontend Development is wrong?

DEMO



danabra.mov ✅

@dan_abramov

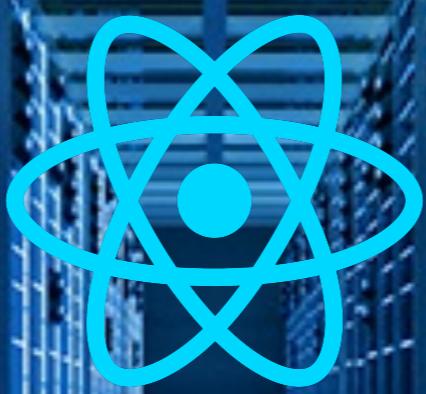
...

never write another API

6:19 AM · Mar 4, 2023 · 39.5K Views

https://twitter.com/dan_abramov/status/1631887155080429569

Wouldn't that be tempting?

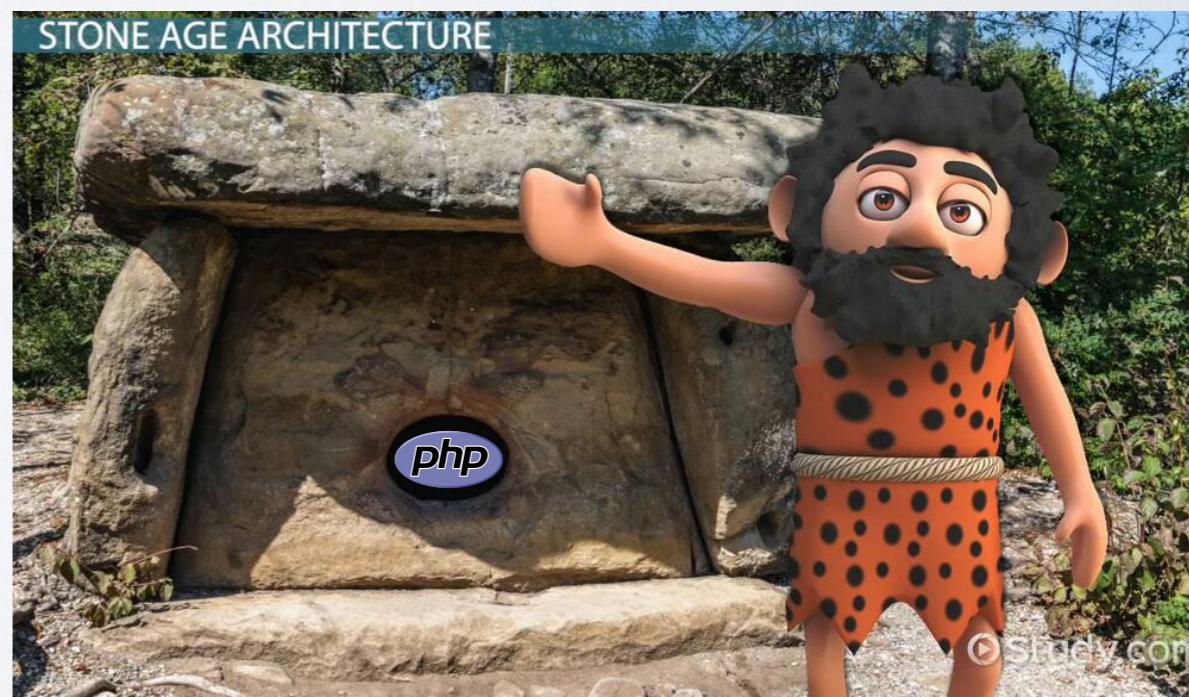


React Server Components (RSC)

NEXT.JS



Back to PHP?

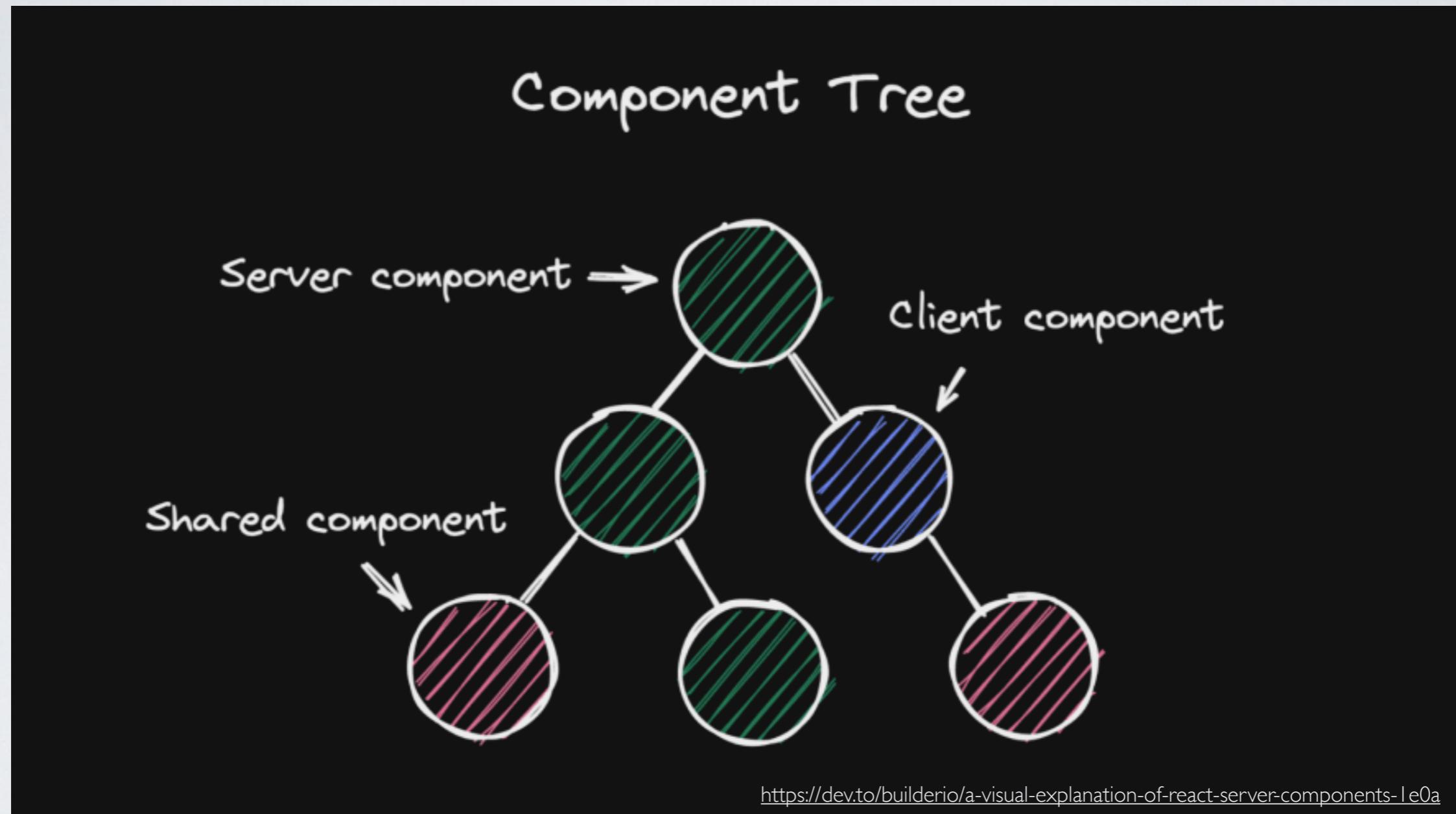


Prepare for ...



DEMO: FormDemo

Your component tree spans across server and client!



Server Components can import Client Components.
Client Components can't import Server Components.
Server Components can be children of Client Components.

Server Components can't:

- can't use `useState / useReducer`
- can't use `useContext`

DEMO: ListDemo - Refreshing Data

It is still a SPA!

Your Code

generate a react tree on the client

Client Component

render instructions
running on the client

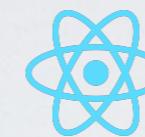
```
export function Greeter() {  
  return (  
    <h1>Hello World!</h1>  
  );  
}
```

render instructions:
1. created on the server

2. sent to the client

Server Component

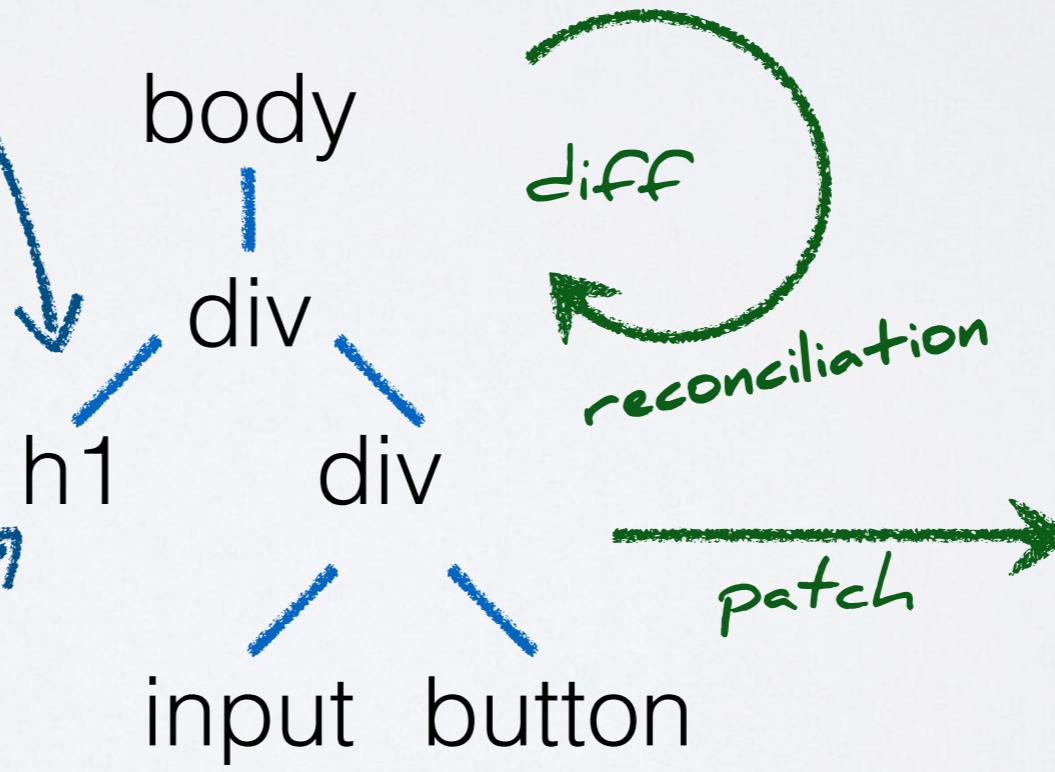
generate a react tree on the server and send it to the client for rendering



React Client Runtime

Virtual DOM

In-Memory, implemented in JavaScript



Browser DOM



```
<body>  
<div>  
  <h1>...</h1>  
<div>  
  <input/>  
  <button>  
    </button>  
</div>  
</div>  
</body>
```

How does it work ...

'use client'
'use server'

danabra.mov  @dan_abramov · Oct 28

“use server” makes a server function callable from the client. it doesn’t change where the function runs (which as you correctly noted is on the server regardless).

1 reply 2 retweets 18 likes 1.5K views

danabra.mov  @dan_abramov · Oct 28

you can think of “use server” as “here’s an entry point into server code”. the opt-in is important because you don’t want arbitrary code to be callable.

1 reply 2 retweets 15 likes 1.7K views

danabra.mov  @dan_abramov

“use server” = server code that can be referenced by the client (becomes API endpoints)

“use client” = client code that can be referenced by the server (becomes <script> tags)

10:23 PM · Oct 28, 2023 · 20K Views
https://twitter.com/dan_abramov/status/1718362813733785970



PHP

Back to PHP?

... yes but in a good way!

The simplicity of PHP combined with the power of React .



Adam Rackis
@AdamRackis

Follow

...

React team: “Hey we cooked up a way to directly invoke server code from the front end. Basically our own RPC. Now you have less need to cobble together http endpoints for every little thing”

The internet: “hurr durr I guess we’re back to asp”

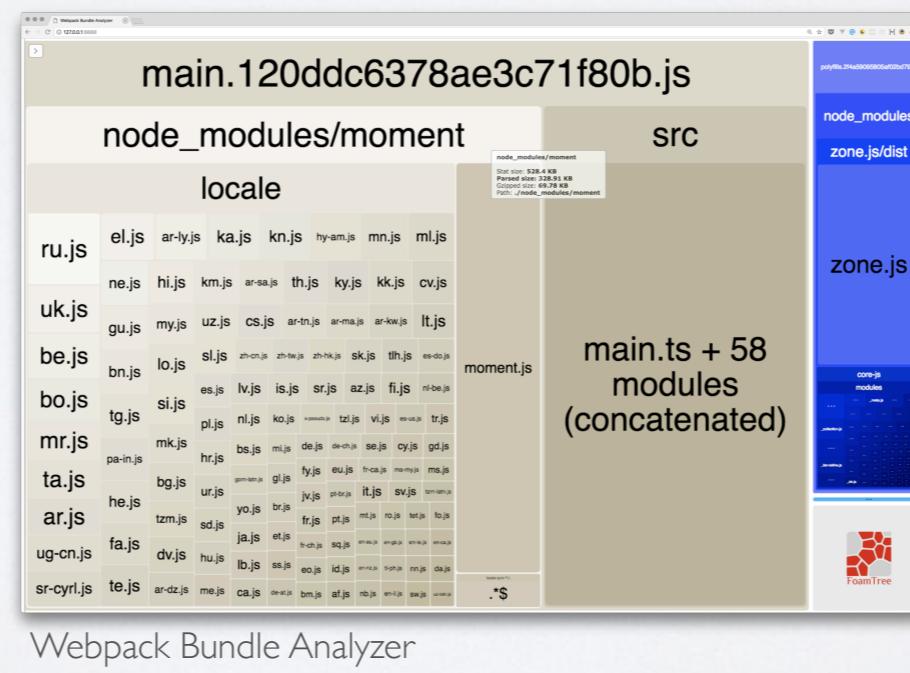
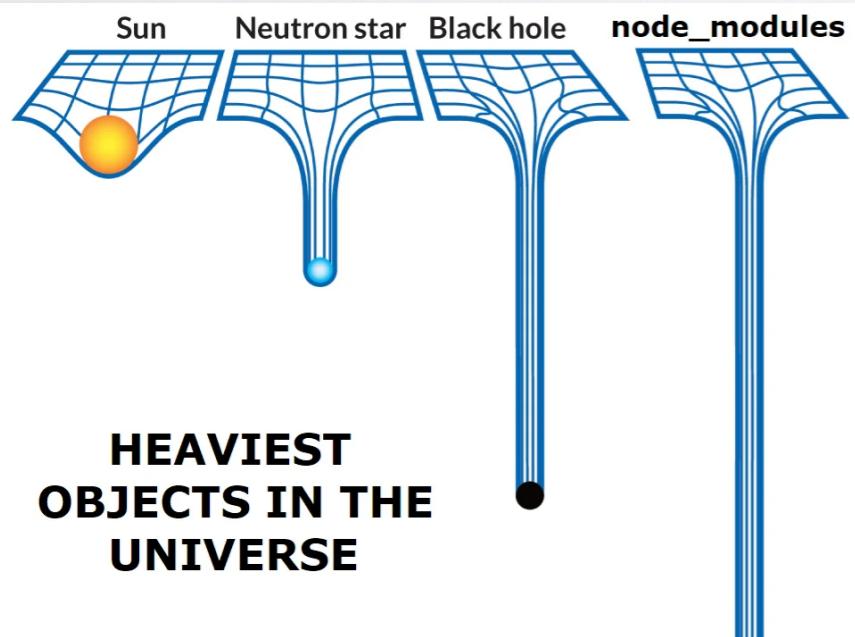
9:41 PM · Oct 27, 2023 · 84.6K Views

<https://twitter.com/AdamRackis/status/1717989813784138225>

They say ...



Thou shalt watch your
dependencies!



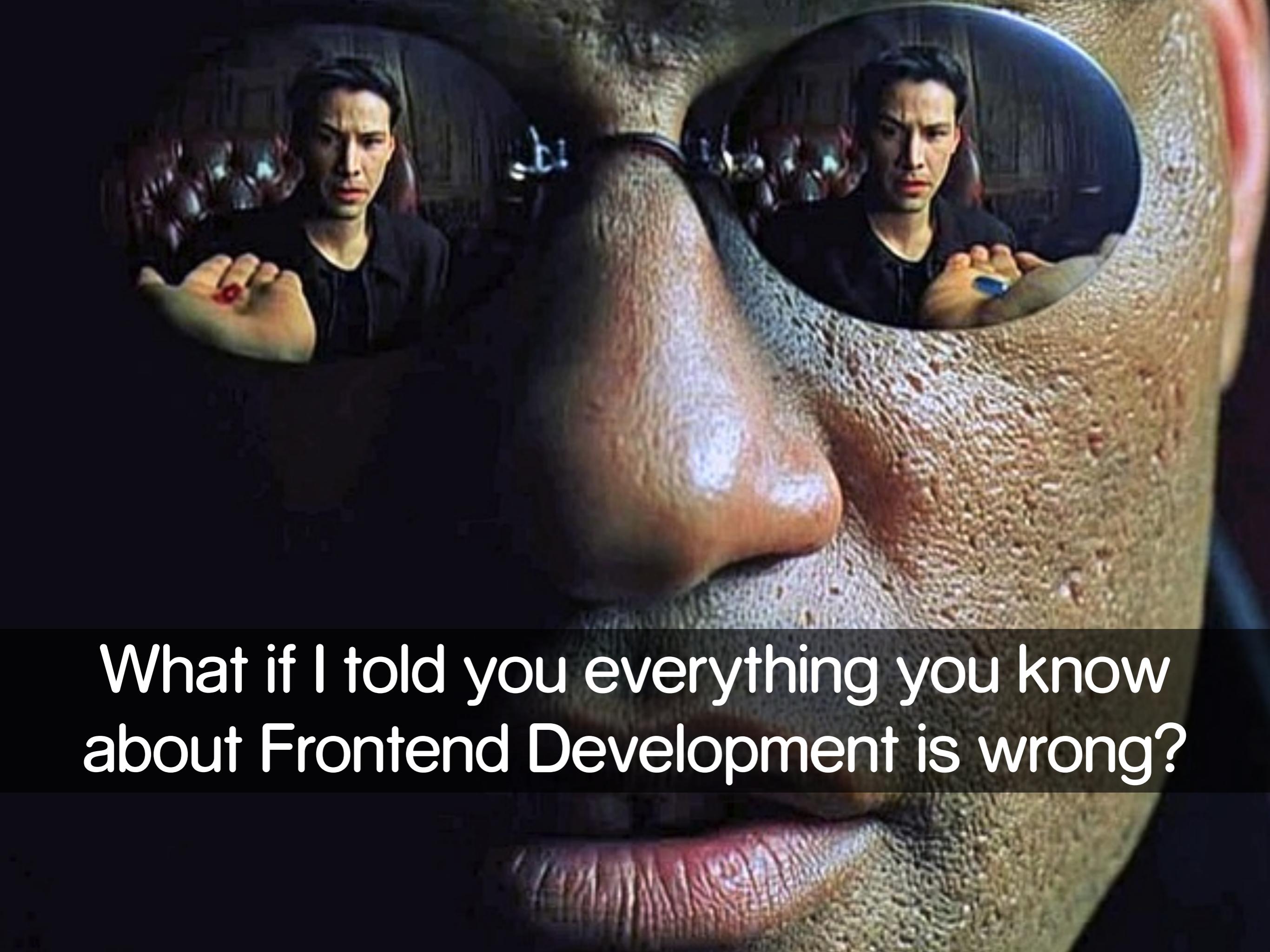
```
import React from 'react'; 44K
import ReactDOM from 'react-dom'; 218.9K
import uniqId from 'lodash/uniqId'; 2.2K
```

VS Code Plugin: Import-Cost



BUNDLEPHOBIA

<https://bundlephobia.com/>



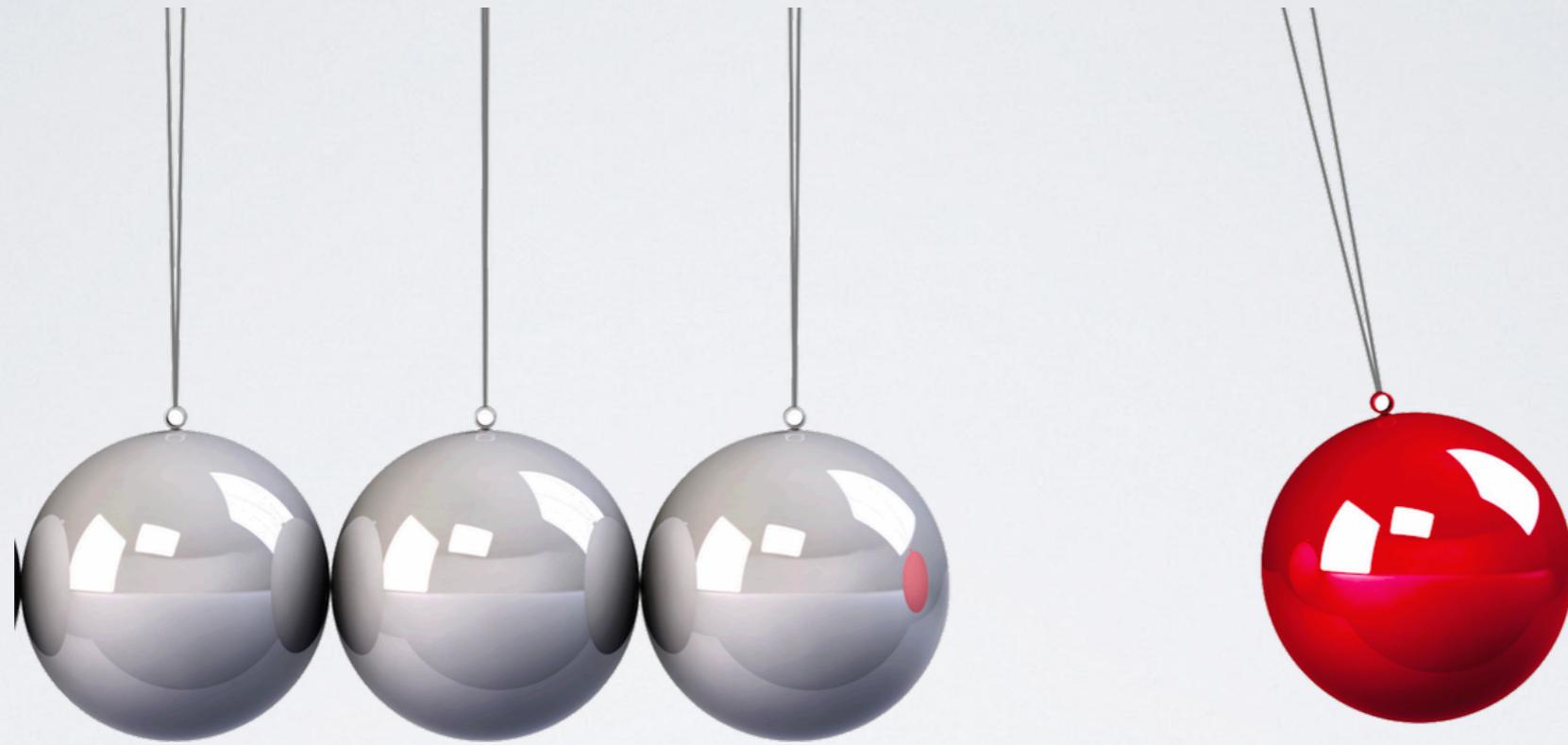
What if I told you everything you know
about Frontend Development is wrong?

DEMO: Chart with D3

```
npm install date-fns  
npm install d3 @types/d3
```



The Pendulum is swinging ...



← Server → Client

... will it ever stop?

My Personal Takeaway

React Server Components try to:

- be an answer for data-fetching and mutations in SPAs
- simplify the full-stack programming model
- improve performance by enabling smaller JS bundles

The power of React is simple & elegant composition ...



Cory House @housecor

...

I love the simplicity of React's reuse model.

Repeating JSX? Create a component.

Repeating logic? Create a hook.

I can compose these simple building blocks in infinite ways.

1:08 PM · Nov 25, 2023 · 16.7K Views

<https://twitter.com/housecor/status/1728385239611789758>

... RSC aim to extend this to the server.

```
function Page() {
  return (
    <Modal> •
    • <NewsStory slug="story-1" />
      <Suspense fallback={<Spinner />}>
        <Carousel> •
        • <RecommendedReads />
        </Carousel>
      </Suspense>
    </Modal>
  );
}
```

Server-side APIs

Client-side APIs



Sam Selikoff at NextJs Conf 2023

<https://www.youtube.com/watch?v=9CN9RCzznZc&t>

Prior Art

Prior Art - Part I: Data Fetching & Mutations

STARWARS
RETURN OF THE RPC

... in other modern frontend frameworks.



<https://remix.run/docs/en/main/route/loader>

<https://remix.run/docs/en/main/guides/data-writes#remix-mutation-start-to-finish>



SOLIDJS

<https://start.solidjs.com/core-concepts/data-loading>

<https://start.solidjs.com/core-concepts/actions>



SVELTE KIT

<https://kit.svelte.dev/docs/load>

<https://kit.svelte.dev/docs/form-actions>



qwik

Qwik City

<https://qwik.builder.io/docs/route-loader/>

<https://qwik.builder.io/docs/action/>

[https://qwik.builder.io/docs/server\\$/](https://qwik.builder.io/docs/server$/)



Also:

<https://trpc.io/>

Prior Art - Part 2: Server Driven SPAs

aka "Live View"



Vaadin



Blazor Server



Phoenix Framework

Enabling SPAs with a server-side programming model and
no need for a REST API.

Demos:

<https://labs.vaadin.com/business/>

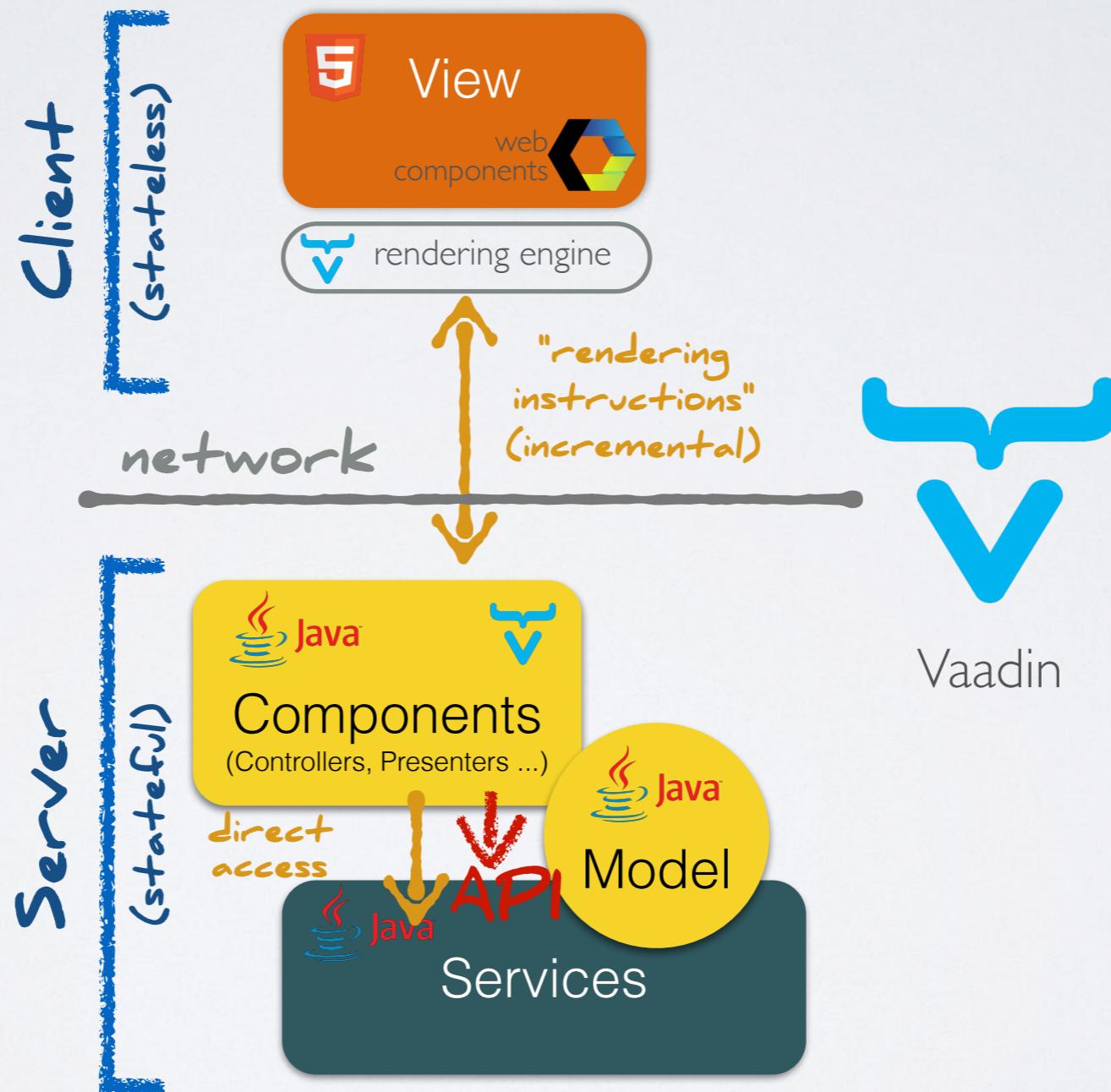
<https://blazor.syncfusion.com/demos/datagrid/overview?theme=bootstrap4>

<https://liveview.zorbash.com/>

Term definition: <https://github.com/dbohdan/liveviews>

Vaadin Architecture

using the browser just as a "rendering engine"



Prior Art - Part 3:

Island Architecture

The islands architecture encourages small, focused chunks of interactivity within server-rendered web pages.



<https://astro.build>



FRESH

<https://fresh.deno.dev>

<https://www.patterns.dev/posts/islands-architecture>

Thank you!

Slides & Code: <https://github.com/jbandi/dev-day-rsc>

Questions? Discussions ...



Jonas Bandi
JavaScript / Angular / React / Vue / Vaadin
Schulung / Beratung / Coaching / Reviews
jonas.bandi@ivorycode.com