

Ask your doctor if

React Server Components

are right for you



Bonnie Schulkin
iJS • Sept 28, 2023

About Bonnie

- in the software industry: **20+ years**
- creating courses full-time: **3 years**
- current gig: Hands-on Web Dev





Image by Nino Carè from Pixabay splash age by Andrew Martin from Pixabay

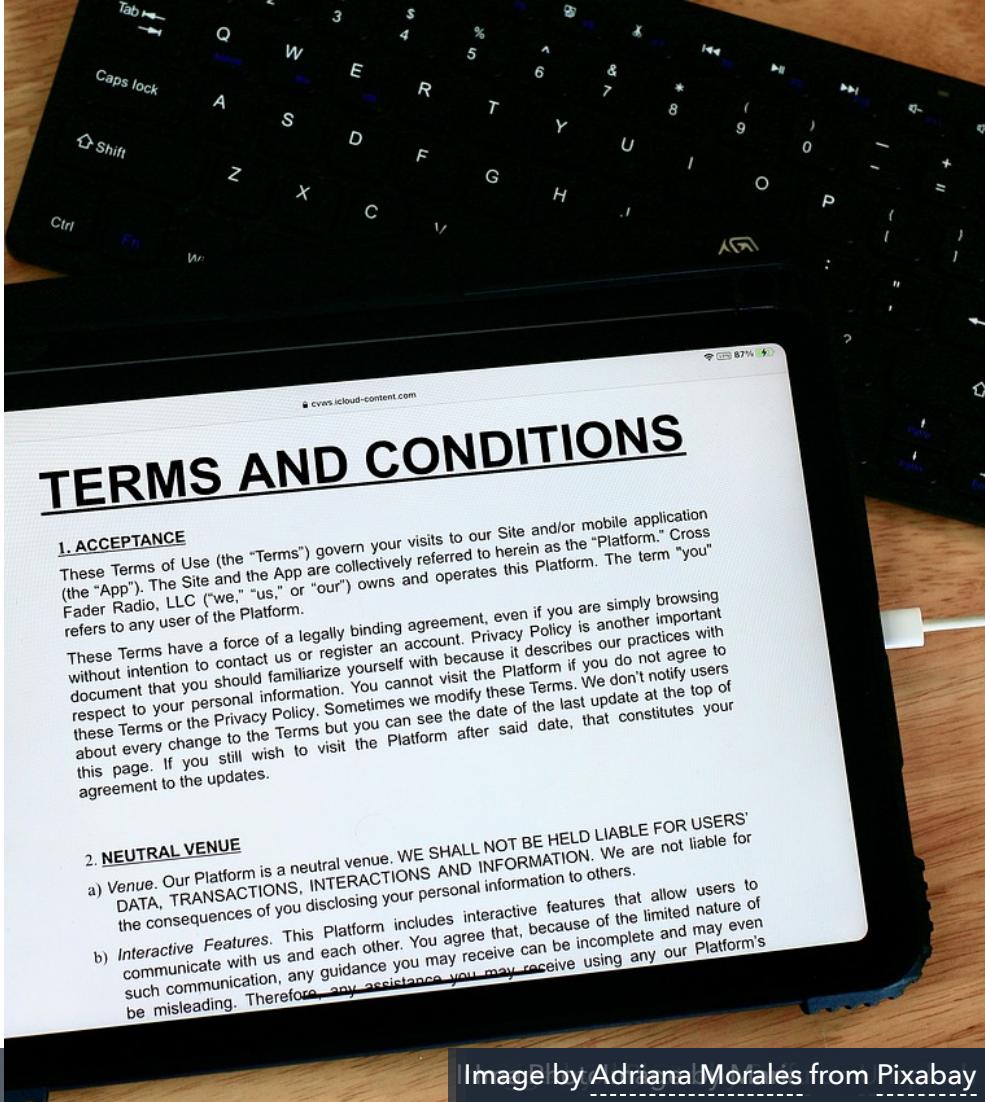
About this talk

- ➡ React and Interactivity
- 🔗 Chain of Communication
- 📦 React Server Components:
Benefits
- ⌚ When to use RSCs
- ✅ Quiz! Server or client?

Image by Adriana Morales from Pixabay

Disclaimer

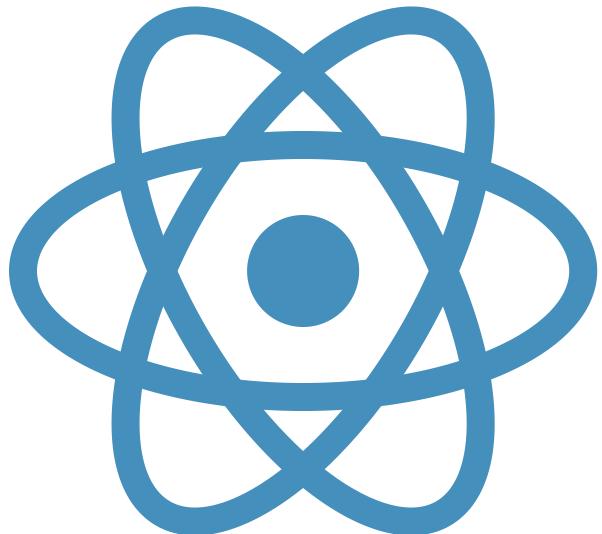
- server components only available via Next.js
- talk covers Next.js app router



React and Interactivity



React



JavaScript library for making user interface (UI)

component component components based on **data** **data** **data**, **props** **props** **props**, and **state** **state**

▫ **component**: re-usable piece of UI

▫ **data**: variable data, usu from a db

▫ **props**: component input



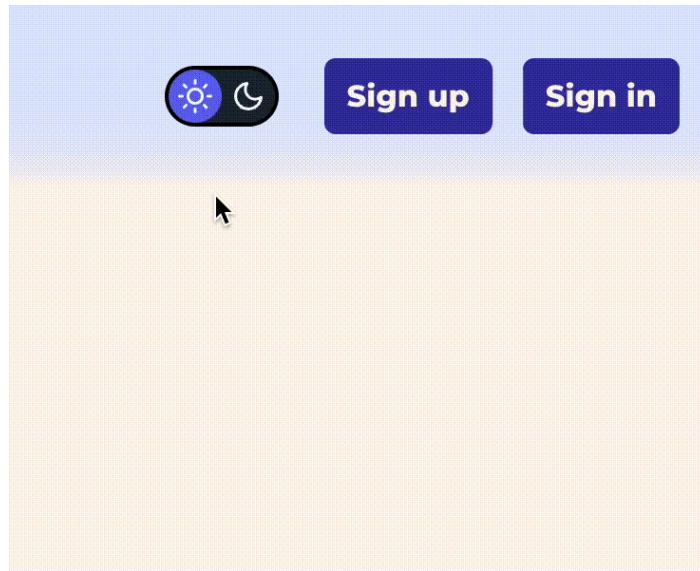
Interactivity

- **Interactivity**
component updates in response to user input
- requires **JavaScript**, for either:
 - event handler (e.g. click), and/or
 - re-render: re-running React code to reflect updates

Quiz! JavaScript required?



JavaScript required?



yes (click handler, state)

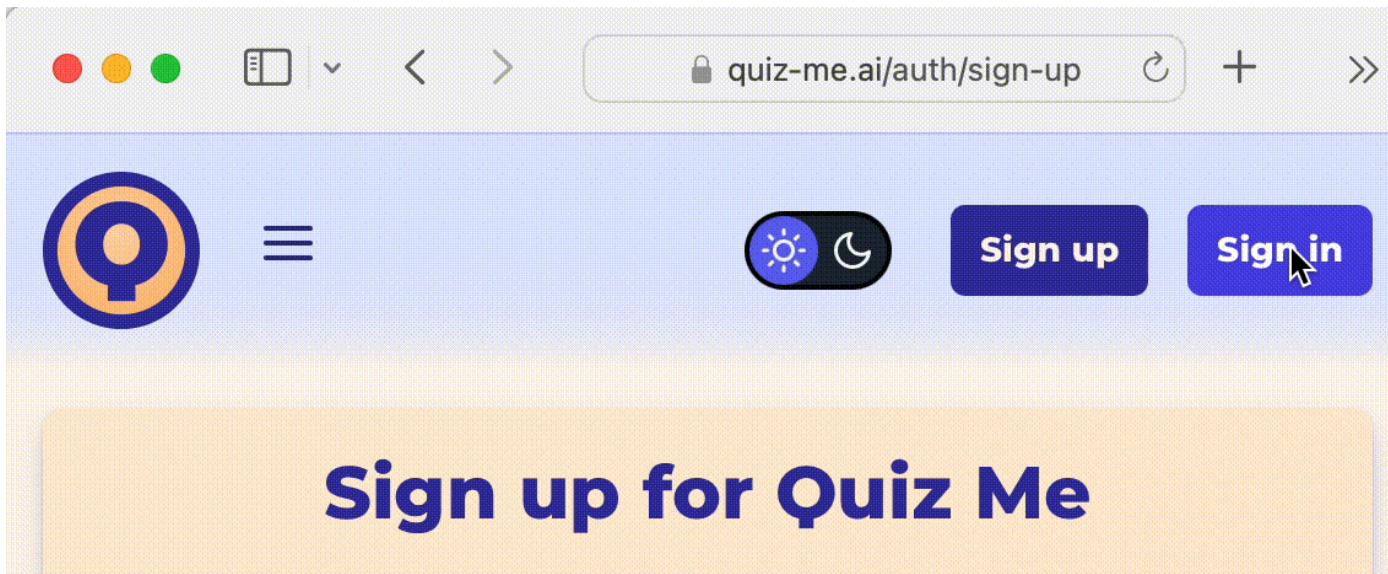
JavaScript required?

Some quiz topic ideas...			
physics	SvelteKit	glass blowing	world history
cricket	oil painting	bhangra dancing	Amelia Earhart
anime	Angular	AI ethics	hiphop
training dragons	cyber security	Australia	React



no (CSS)

JavaScript required?



no (new page)

JavaScript required?

Please enter your ChatGPT API key

API key:

YycT3BlbkFJwx0UONKASjKJeeGFUx9J

Submit



yes (state)

Chain of Communication



Chain of communication

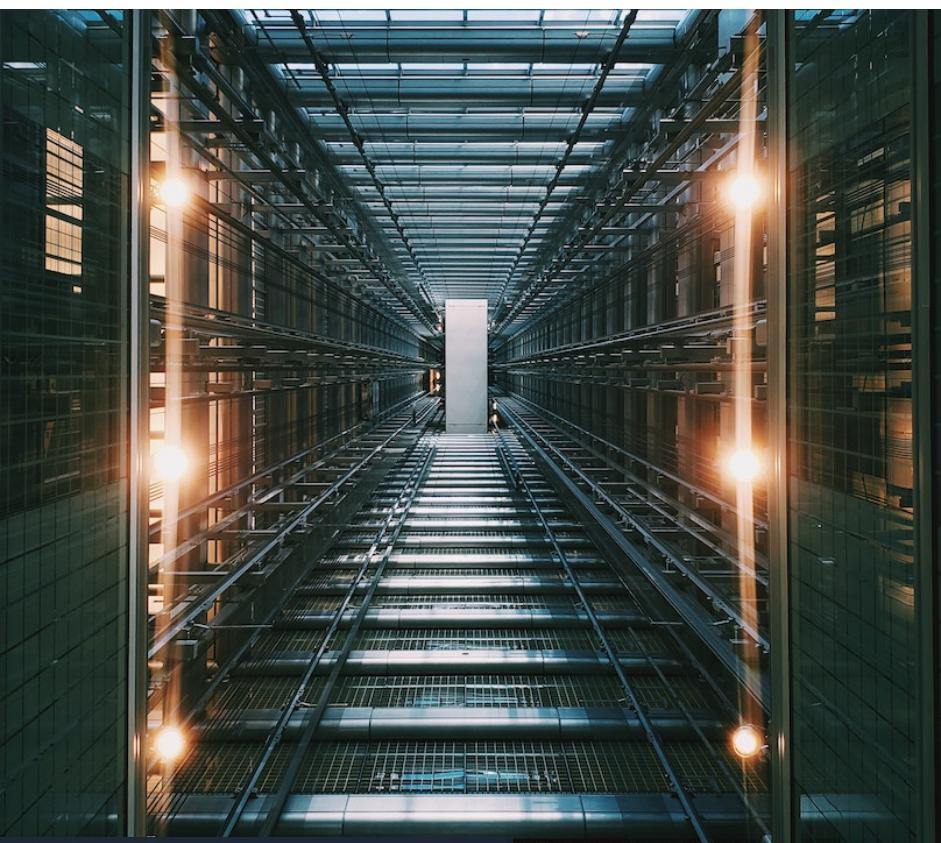


Image by Nino Carè from Pixabay



ay

Image by Adriana Morales from Pixabay

A photograph showing a paved area made of grey cobblestones. There are several small, blue, oval-shaped floor markers scattered across the surface. One marker in the foreground is clearly visible, featuring a white icon of a person standing and the text "ATTENDI QUI" above "WAIT HERE".

Where React code runs

- it matters!
- run on server: no re-renders
- run on client: no db queries
 - need to make separate API call

Why not both?

- SSR: server-side rendering
 - server queries db
 - data passed as props
 - renders on server
 - **hydrates** on client

hydration = add JavaScript
for interactivity



WHY NOT BOTH

SSR, in code (`pages` router)

```
1 import { getHabitatsFromDb } from "../../database-actions";
2 import Habitat from "./Habitat";
3
4 export const getServerSideProps = (async (context) => {
5   const habitats = await getHabitatsFromDb();
6   return { props: { habitats } }
7 });
8
9 export default function DragonHabitats({ habitats }) {
10   return (
11     <>
12       <h1>Dragon Habitats</h1>
13       {habitats.map((h) => {
14         <Habitat key={h.id} data={h}>
15       ))}
16     </>
17   );
18 }
```



SSR: best of both worlds!?

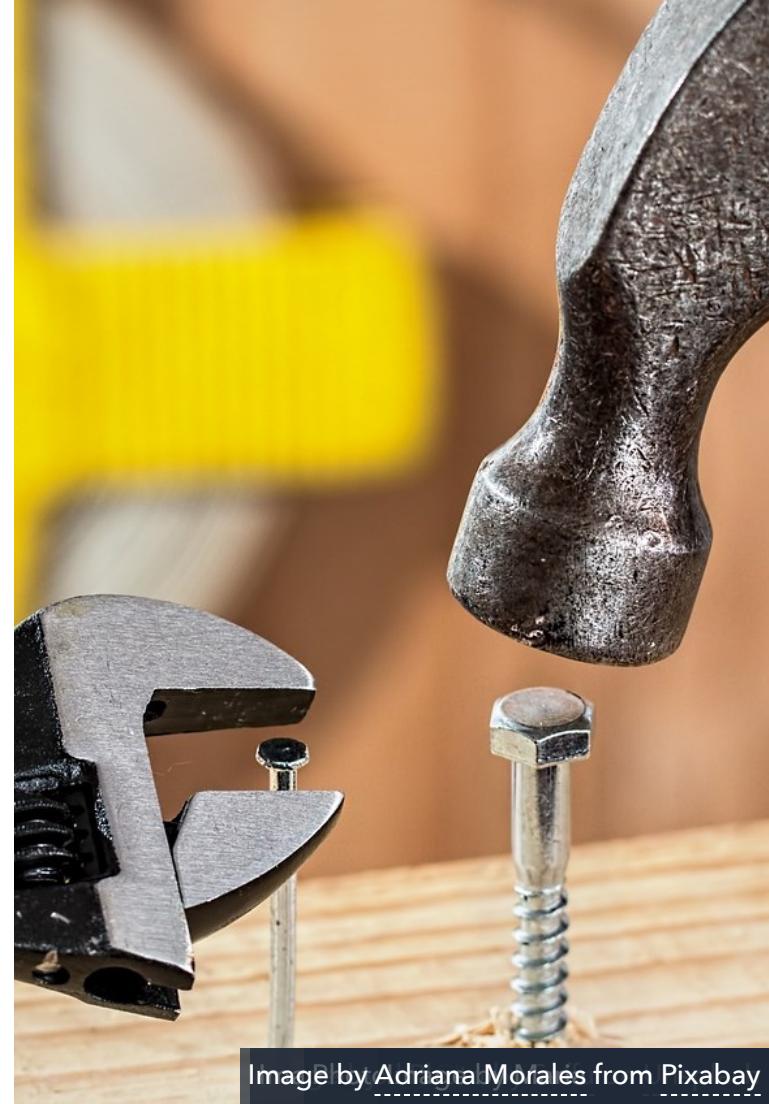
- access to db **and** interactivity

What if there's no interactivity?

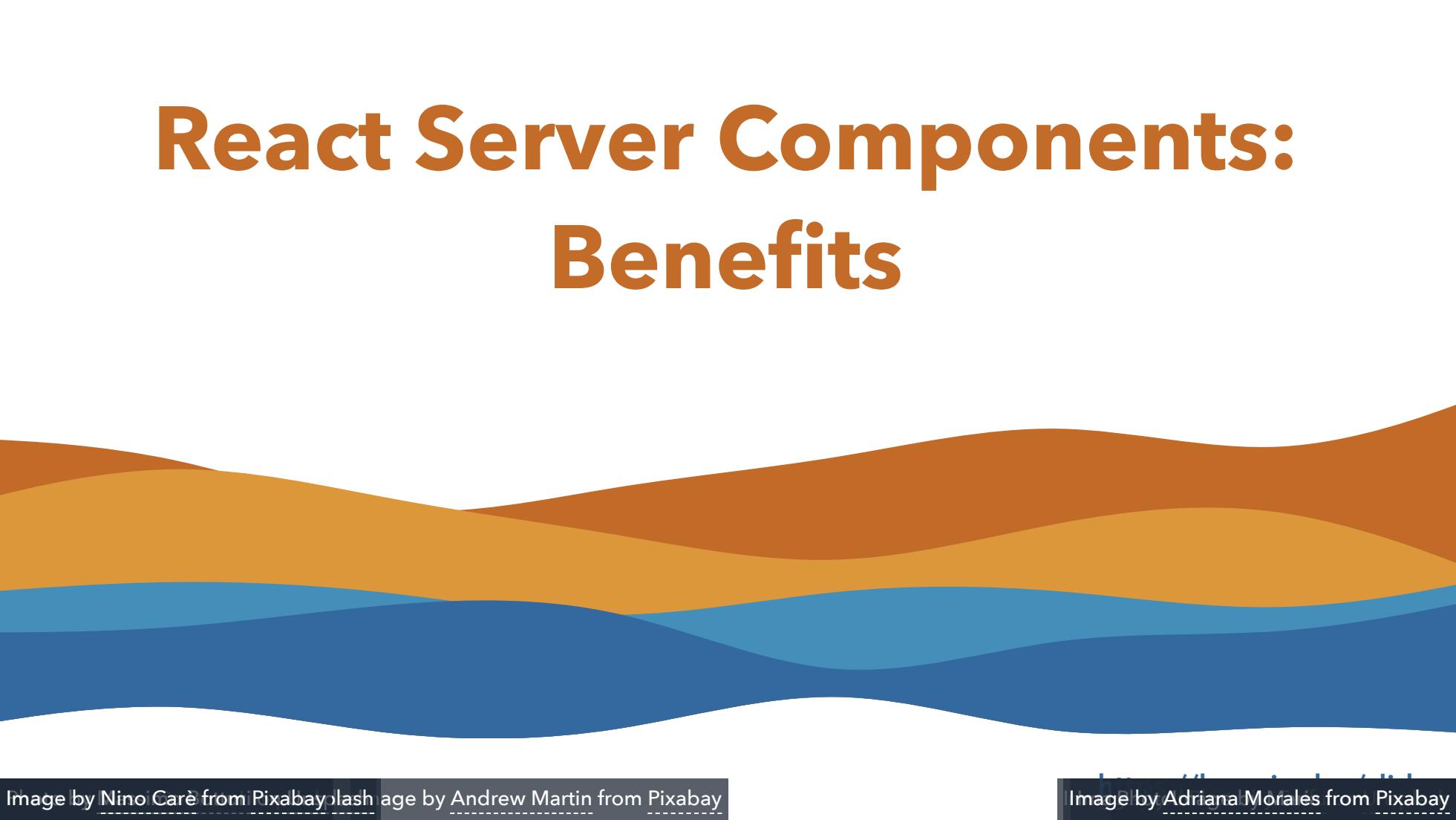
- No re-renders needed!
- SSR sends unnecessary JS
- No hydration necessary

React Server Components

- server render **only**
- choice: server **or** client render
- don't do both everywhere
- instead, do each where needed



React Server Components: Benefits

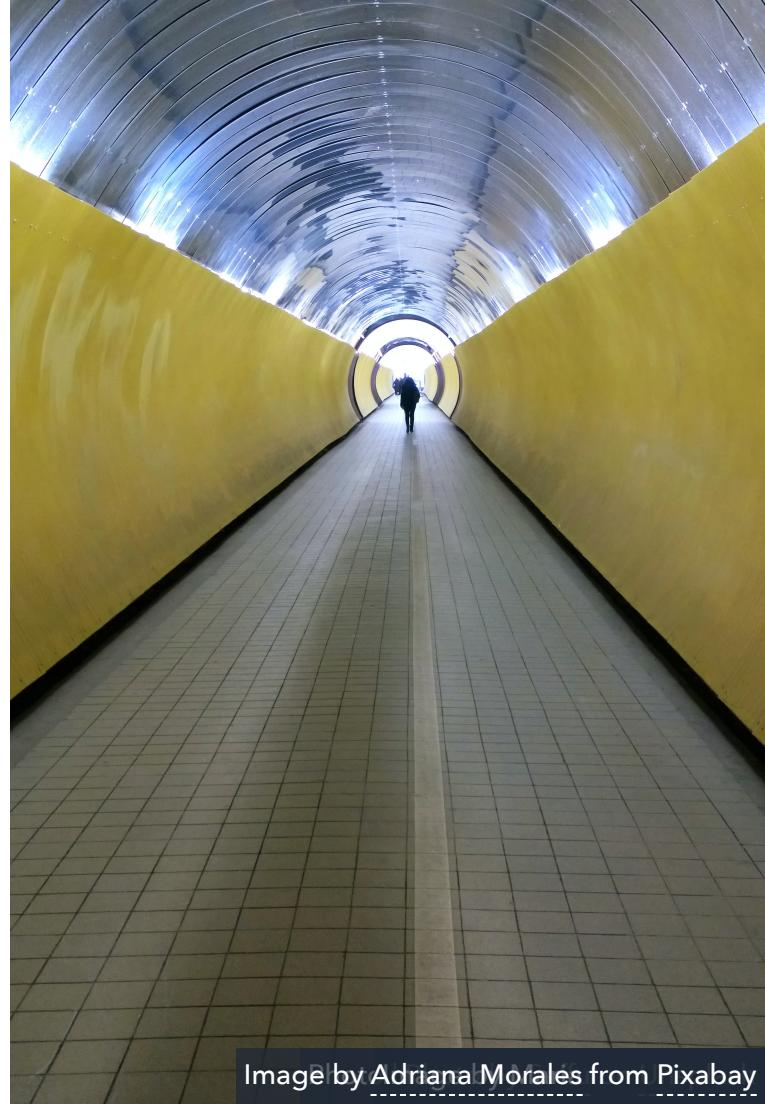


RSC: Benefit #1

Component renders **only** on server

Access to server resources

- database
- filesystem



Data from database

Client component

```
1 import Habitat from "./Habitat";
2
3 function DragonHabitats() {
4   const [habitats, setHabitats] = useState([]);
5   useEffect(() => {
6     fetch("/api/habitats")
7       .then(data => data.json())
8       .then(setHabitats)
9   }, []);
10
11 // display habitat components
12 return (
13   <>
14     <h1>Dragon Habitats</h1>
15     {habitats.map((h) => {
16       <Habitat key={h.id} data={h}>
17     })}
18   </>
19 );
20 }
```

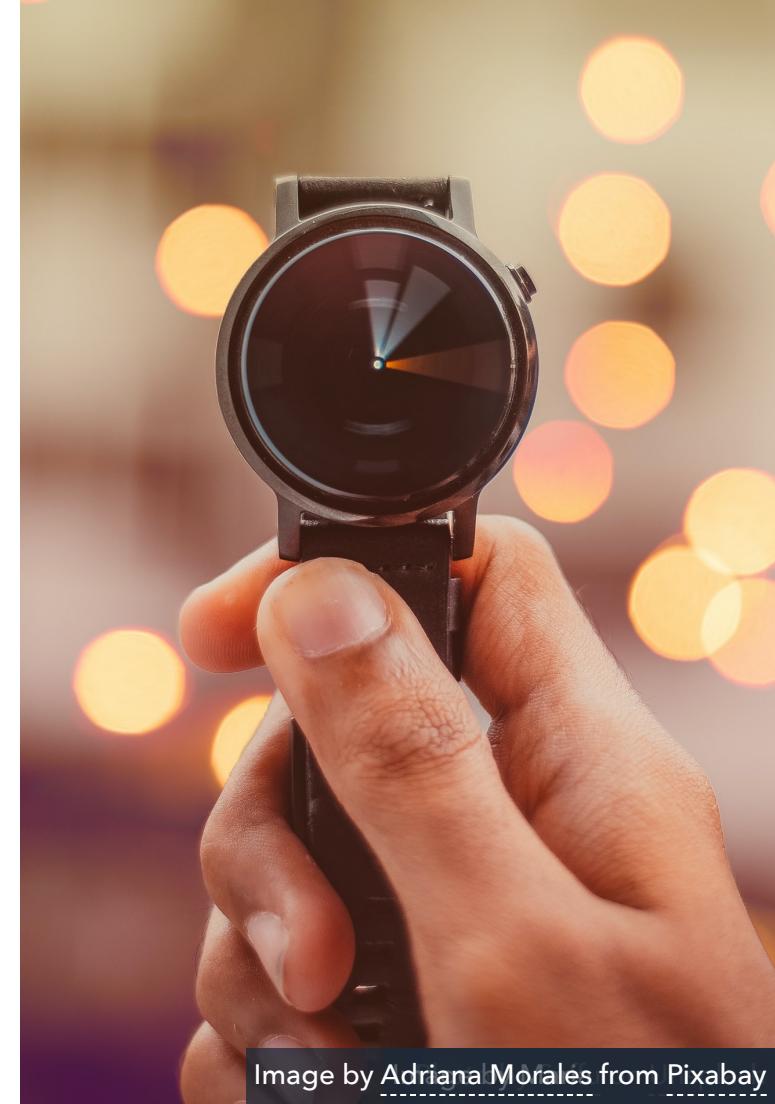
RSC

```
1 import { getHabitatsFromDb } from "../../database-actions";
2 import Habitat from "./Habitat";
3
4 async function DragonHabitats() {
5   const habitats = await getHabitatsFromDb();
6
7   // display habitat components
8   return (
9     <>
10       <h1>Dragon Habitats</h1>
11       {habitats.map((h) => {
12         <Habitat key={h.id} data={h}>
13       ))}
14     </>
15   );
16 }
```

RSC: Benefit #2

Component **never** renders on client
increased website speed

- Server is more powerful (fast!)
- Exclude extra JS from bundle
- Skip hydration step



Server JS Examples

Prism syntax highlighter
~7kB

```
function Equipment() {
  return (
    <div>
      <h1>Equipment</h1>
      <span class="subtitle">
        Everything you need to train your dragon
      </span>
    </div>
  )
}
```

React PDF Renderer
~85 kB

LUKE SKYWALKER

JEDI MASTER

luke@theforce.com



EXPERIENCE

- Jedi Temple, Coruscant | Head Jedi Master A long time ago...
- Started a new Jedi Temple in order to train the next generation of Jedi Masters
 - Discovered and trained a new generation of Jedi Knights, which he recruited from within the New Republic
 - Communicated with deceased Jedi Masters such as Anakin Skywalker, Yoda, Obi-Wan Kenobi in order to learn the secrets of the Jedi Order

Rebel Alliance | General

A long time ago...

- Lead legions of troops into battle while demonstrating bravery, competence and honor
- Created complicated battle plans in conjunction with other Rebel leaders in order to ensure the greatest chance of success
- Defeated Darth Vader in single-combat, and convinced him to betray his mentor, the Emperor

Rebel Alliance | Lieutenant Commander

A long time ago...

- Destroyed the Death Star by using the force to find its only weakness and delivering a torpedo into the center of the ship
- Commanded a squadron of X-Wings into battle
- Defeated an enemy AT-AT single-handedly after his ship was destroyed
- Awarded a medal for valor and bravery in battle for his successful destruction of the Death Star

Tatooine Moisture Refinery | Moisture Farmer

A long time ago...

- Replaced damaged power converters
- Performed menial labor throughout the farm in order to ensure its continued operation

Link

[View profile](#) This is the candidate you are looking for

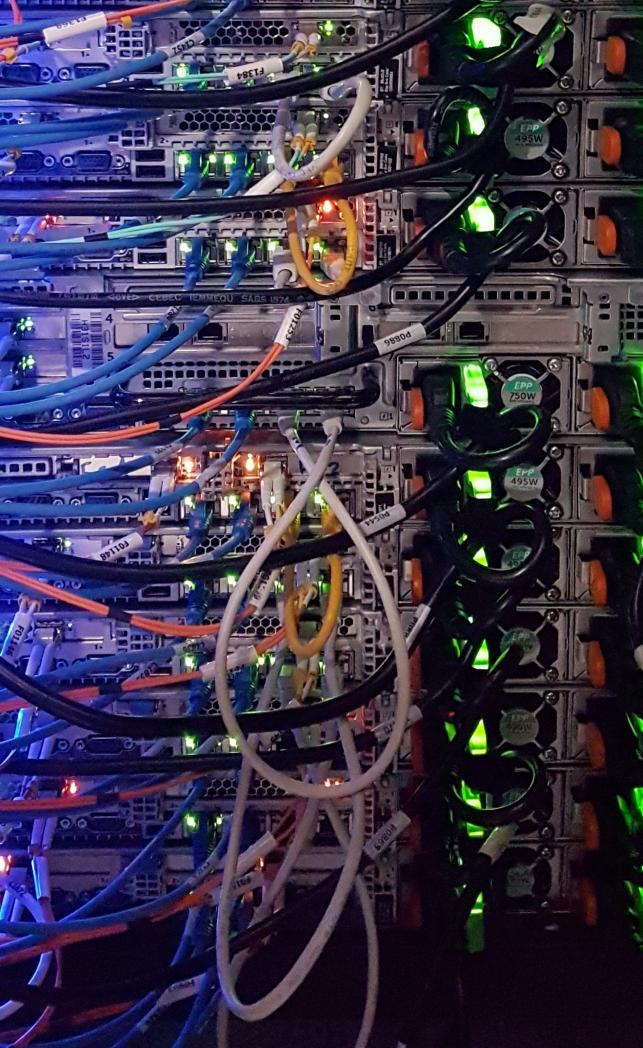
Put another way...

"[T]he server,
far more **powerful**
and **physically closer** to your data sources,
deals with **compute-intensive rendering**
and ships to the client **just the interactive pieces of code.**"

Source: Understanding React Server Components, Vercel Blog

When to use RSCs





React Server Components

for server resources

- database or filesystem access
- computationally intensive renders
- large JS packages not sent to client

Client components

for interactivity

- component that needs to re-render
- changes in state, props, context
- usually due to user interaction



Why not both? but separate

```
import getDataFromDb from "../../database-actions";
import ClientComponent from "./ClientComponent";

async function ServerComponent() {
  const data = await getDataFromDb();

  return (
    <h1>{data.title}</h1>
    <ClientComponent data={data} />
  );
}
```

- pattern: server parent
→ client child
- RSC queries db
- data passed as prop
- interactive child



Example: RSC and client components

- server component queries db for data
- **static UI**: parent server component
- **interactive UI**: child client component

GODOG

Large Skinny Dragons Pet Toy

\$9.99 \$7.00

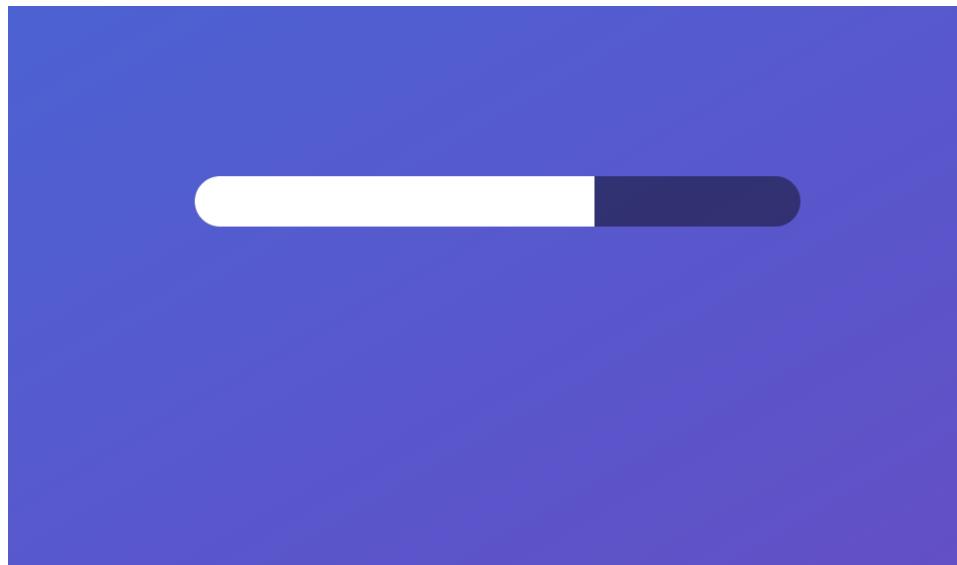
Compare At \$16

Quiz! Server or Client? Or Both?



Server or client component?

Progress bar



💻 client (state)

Server or client component?

LaTeX math notation

$$\sqrt{3x - 1} + (1 + x)^2$$

$$\int_0^{\infty} x^2 dx$$

$$\lim_{h \rightarrow 0} \frac{f(a + h) - f(a)}{h}$$



server (computation, bundle size)

Server or client component?

List of upcoming shows, with links to ticket purchase page

UPCOMING SHOWS

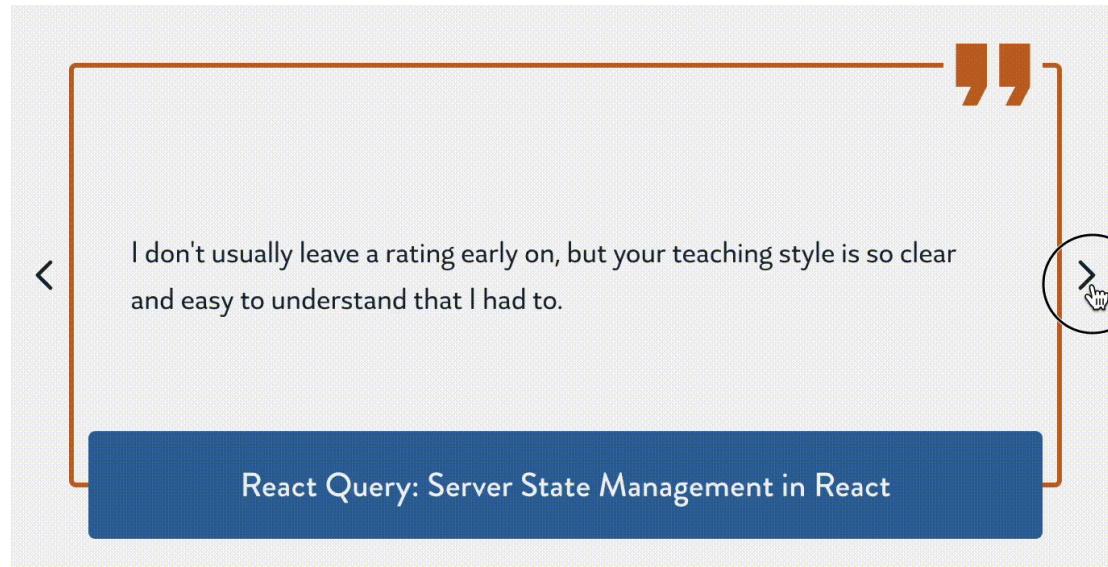
2023 SEP 9	TICKETS	SHAMROCK PETE <i>classic hip-hop supported by a dreamy musical saw</i>
2023 SEP 10	SOLD OUT	avalanche of cheese <i>serious americana featuring iconic harmonies</i>
2023 SEP 11	TICKETS	PICKLE BICS <i>improvisational pop, known for an adventurous water glass orchestra</i>



server (database)

Server or client component?

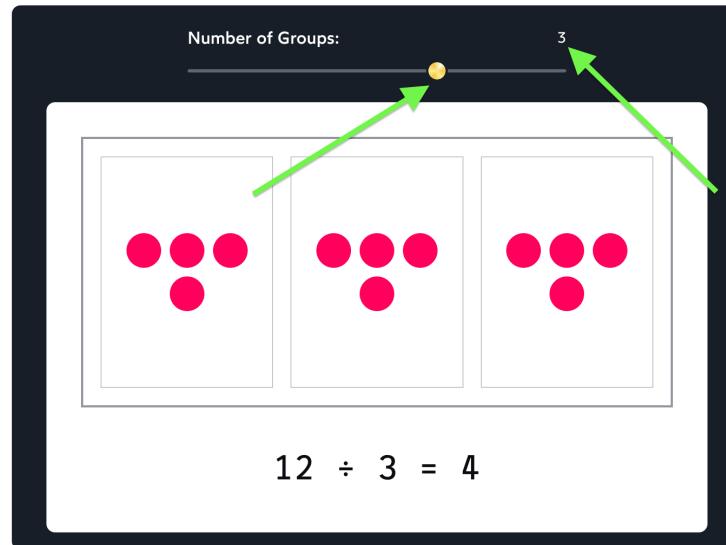
Rotating review component



 both (database and state)

Server or client component?

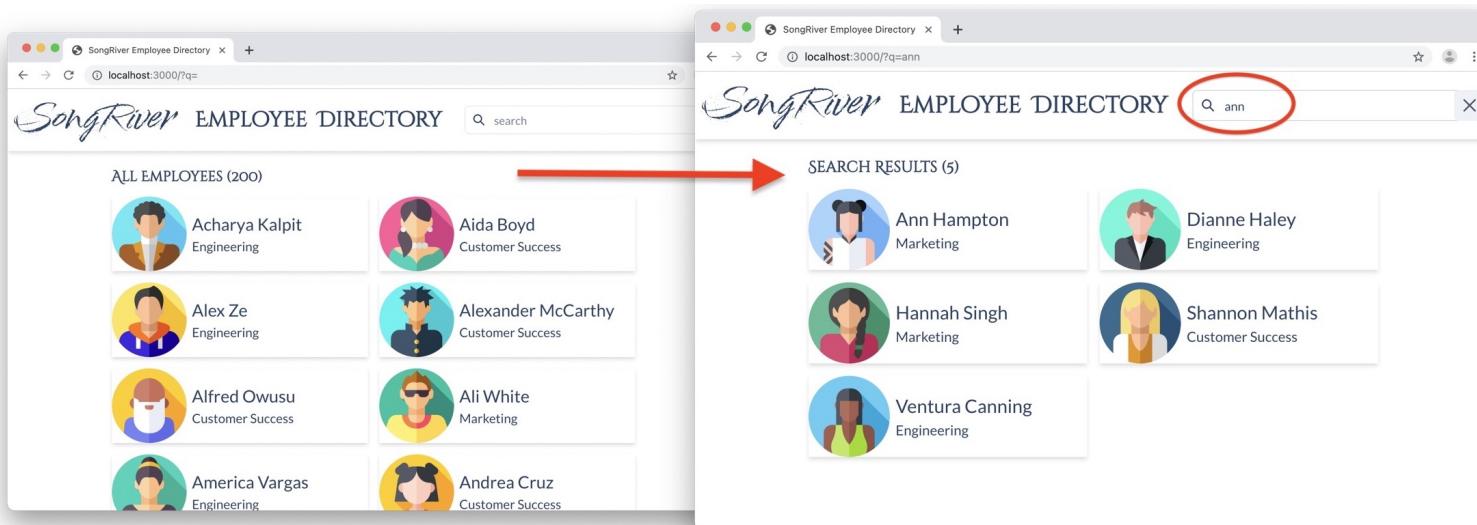
Slider



💻 client (re-render)

Server or client component?

Employee directory results with filter



 both (database and state)

Server or client component?

Component from markdown file

Dragon Habitats

Dragons can survive in a variety of habitats, depending on the dragon species, and the environments available to you.

draconis aquarius, for example, needs access to water, whereas *draconis silva* will not be happy without trees. All dragons require ample room and freedom to fly long distances.



server (filesystem, computation)

I want more!



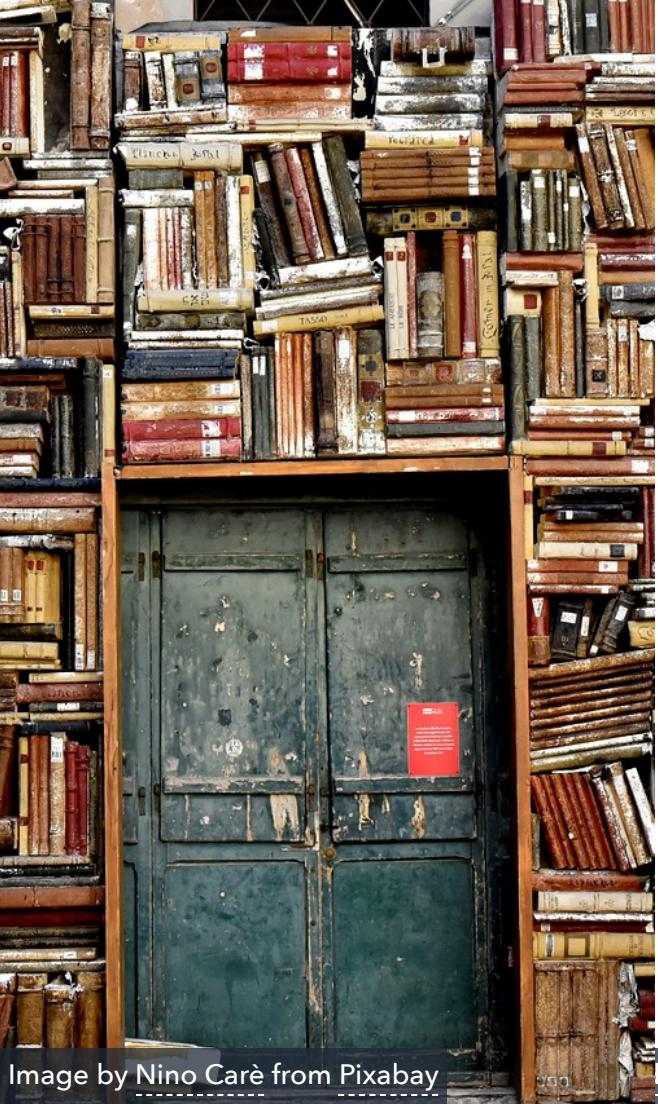
Not covered

streaming / suspense

delivering data as *it becomes available*

Next.js docs





Further reading

Understanding React Server Components

Excellent introductory blog post from Vercel

Making Sense of React Server Components

Useful graphics to visualize sequence of events

Demystifying React Server Components...

Technical discussion with helpful examples

When to use Client and Server components

Table with guidelines, from Vercel docs

More Bonnie!

- <https://bonnie.dev>
 - Blog post of this talk
- <https://howd.dev>
- <https://quiz-me.ai>



Thank you!

