React Data Fetching Patterns

From SPAs to SSG to SSR to RSCs

Kevin Van Cott

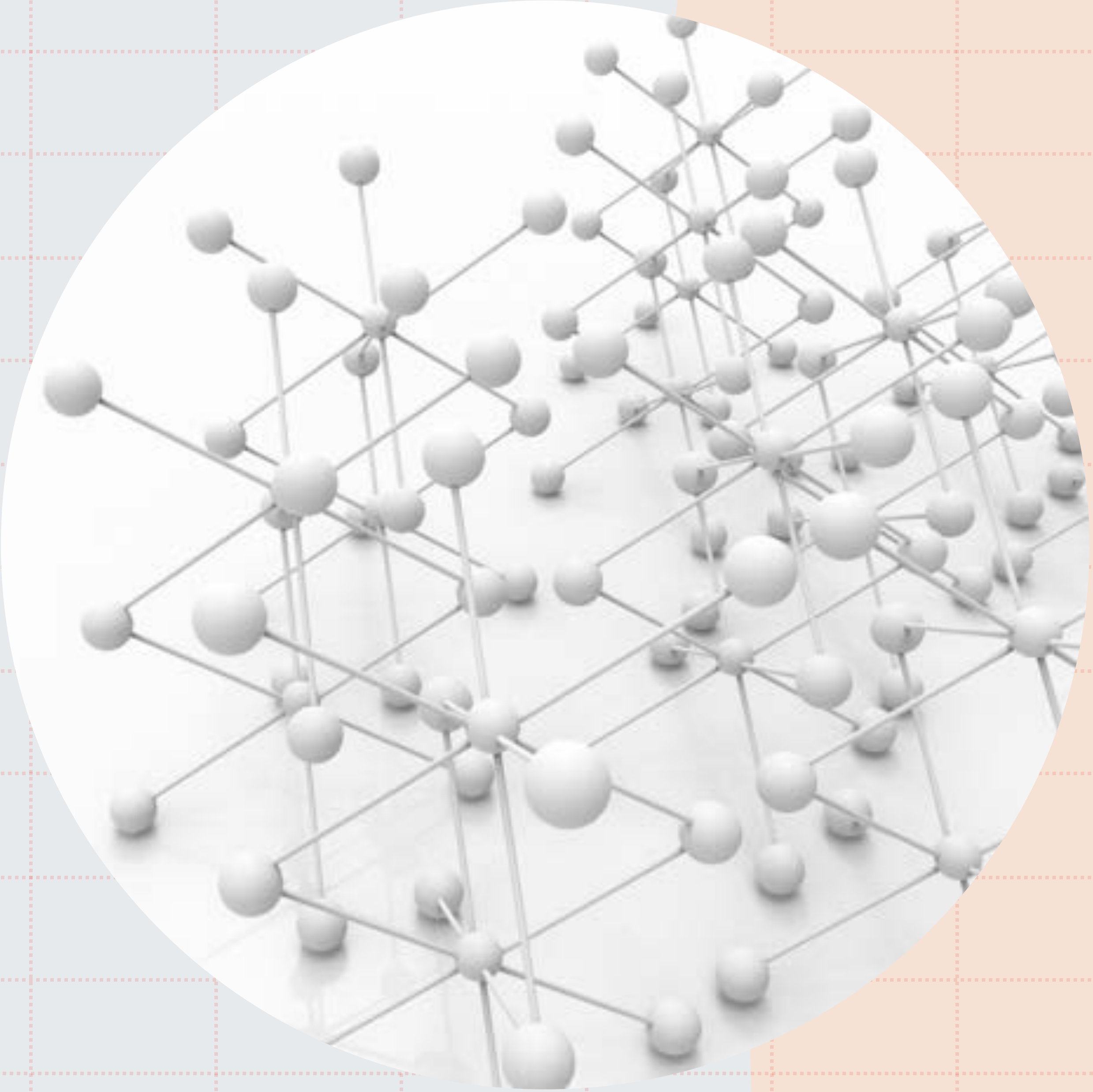
Senior Software Engineer / OSS Maintainer

Who Am I?

- Senior Software Engineer / OSS Maintainer
- I currently work at RentVision in Lincoln, NE
- Other companies I've worked for:
  Manifest Cyber, Fusion Medical Staffing, ALLO Fiber, Talent Plus, Nebraska DHHS
- Open Source Contributor – TanStack Table, MRT
- TanStack Consultant - Dedicated Support | TanStack

What We Will Learn in this Talk?

- What kind of apps/websites can be built with React?
- The most popular ways to build with React (Vite, Next.js, Remix) and how data fetching looks in each
- Best practices for data fetching
- Dig into some code!
- The pros and cons of these different React Frameworks and fetching strategies



What Can We Even Build With React?

"React is a Client-Side JS Library, Right?"

1. React itself is just a client-side JavaScript Library that usually only runs in the browser
2. But Multiple "Meta Frameworks" have popped up over the years to make writing React a apps a full-stack experience

- Websites
- Web Applications
- Mobile Apps (React Native)

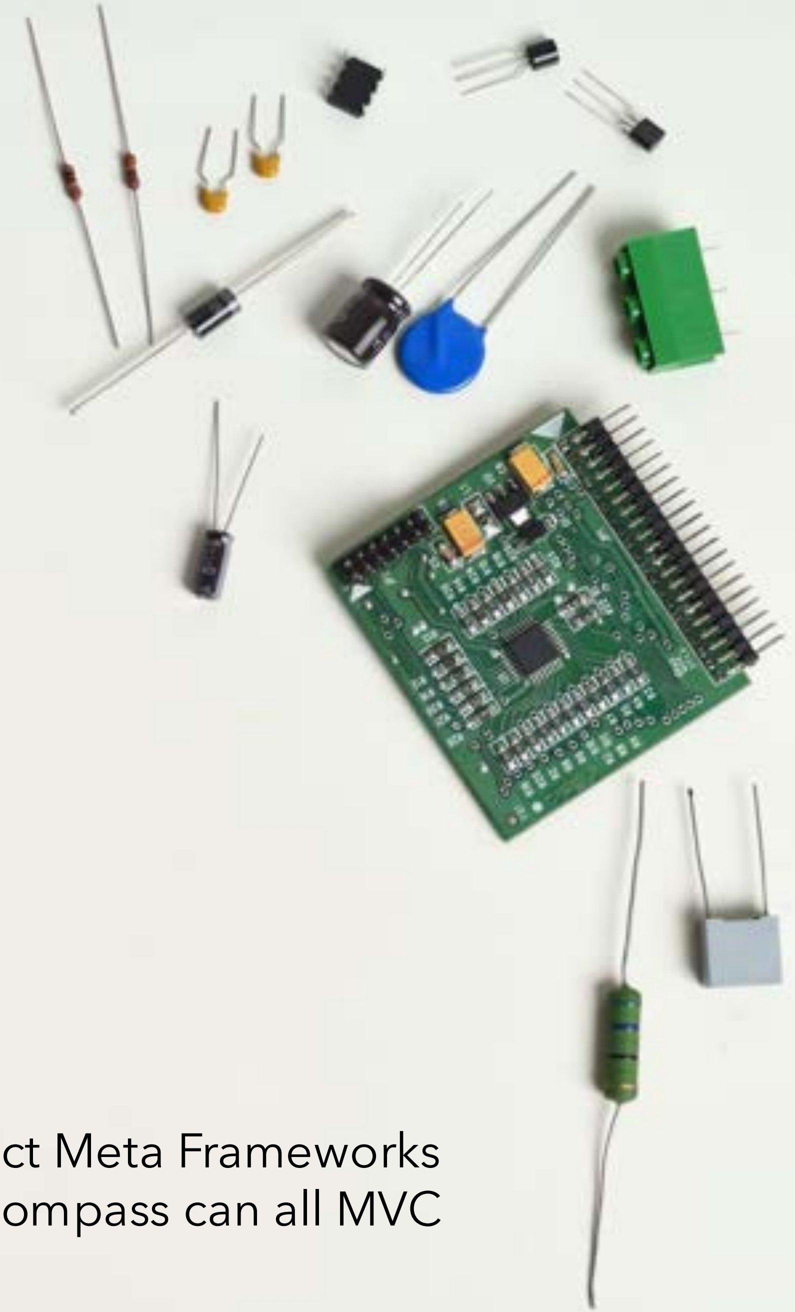
MVC

Model: Data Management

View: UI

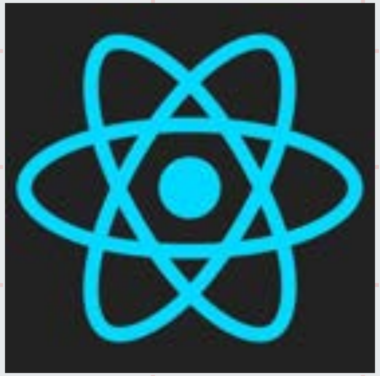
Controller: Communication between Data and UI

React usually goes here



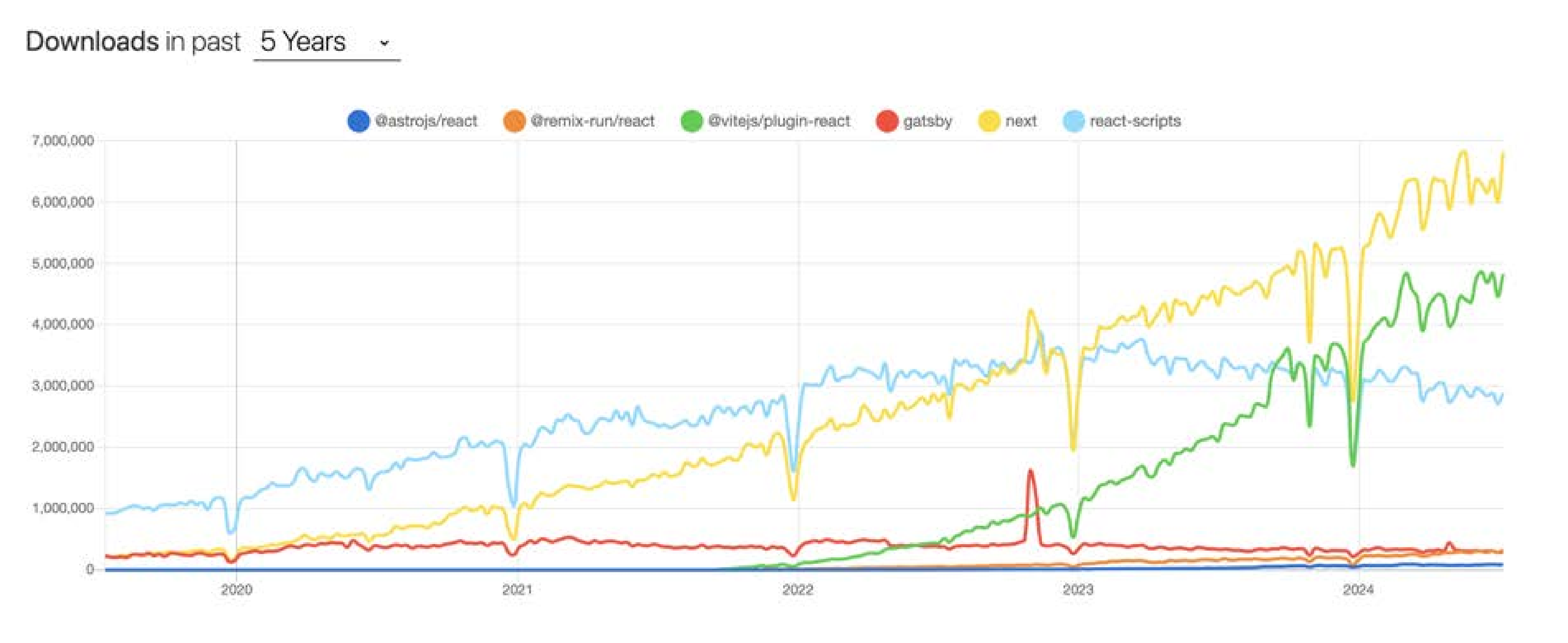
React Meta Frameworks encompass can all MVC

React Frameworks



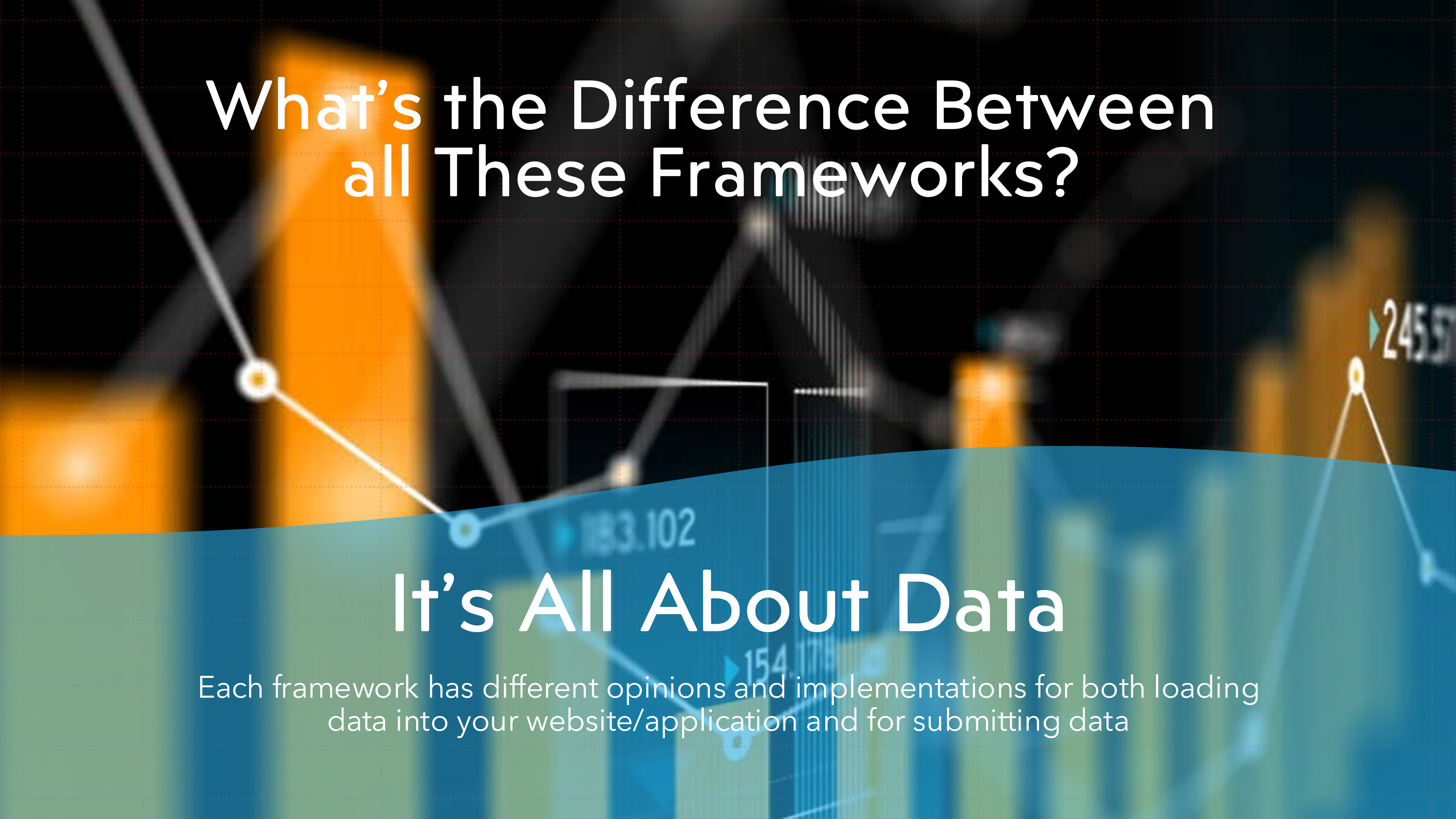
TanStack Start
IT'S ALMOST READY!

React Frameworks over the years



Downloads in past 5 Years

@astrojs/react vs @remix-run/react vs @vitejs/plugin-react vs gatsby vs next vs react-scripts | npm trends



What's the Difference Between all These Frameworks?

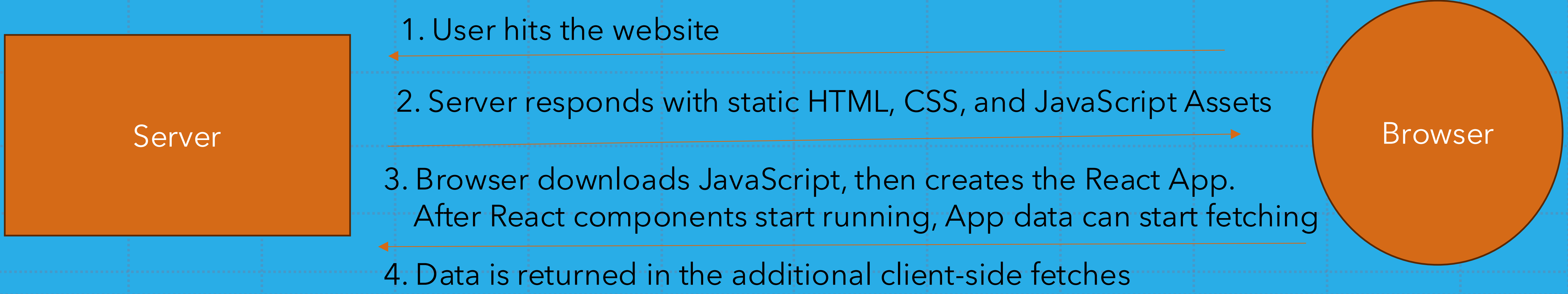
It's All About Data

Each framework has different opinions and implementations for both loading data into your website/application and for submitting data

- What is a SPA? – Single Page Application
- In a traditional SPA, the React project is only part of the view layer.
- All data fetching is initiated client-side

Chapter 1: Client-Side SPAs

	Runs on Server	Runs on Client
Client-Side SPA	✗	✓

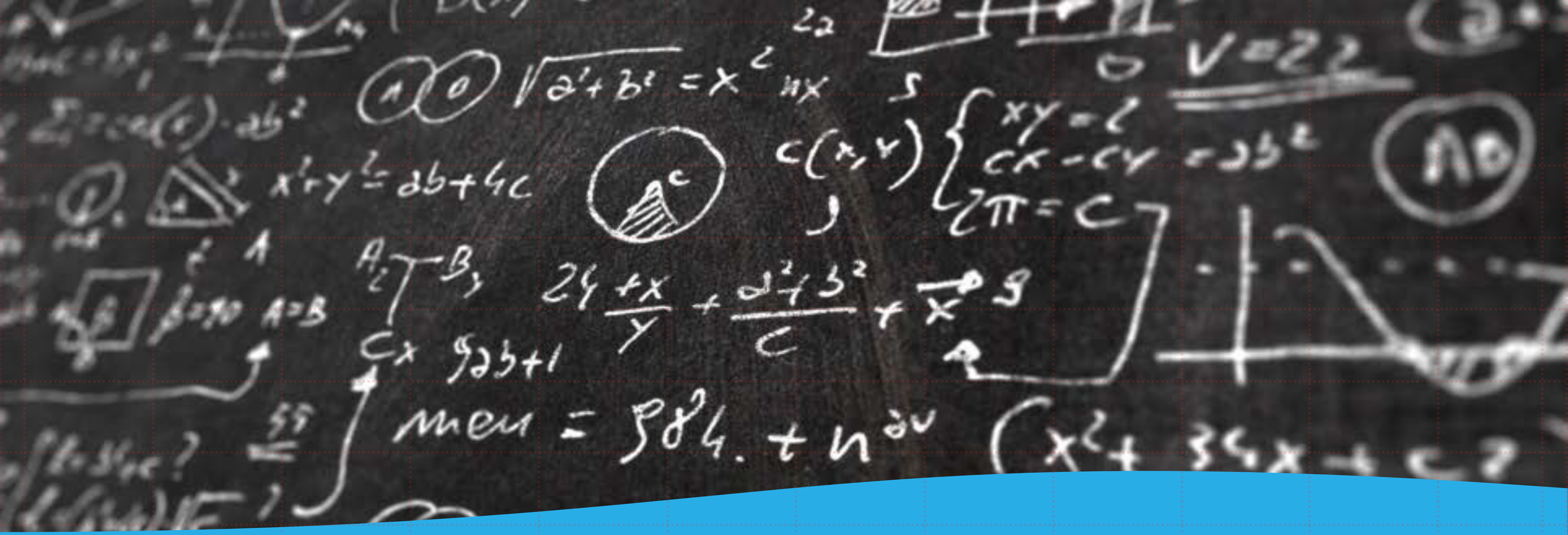


1. User hits the website

2. Server responds with static HTML, CSS, and JavaScript Assets

3. Browser downloads JavaScript, then creates the React App. After React components start running, App data can start fetching

4. Data is returned in the additional client-side fetches

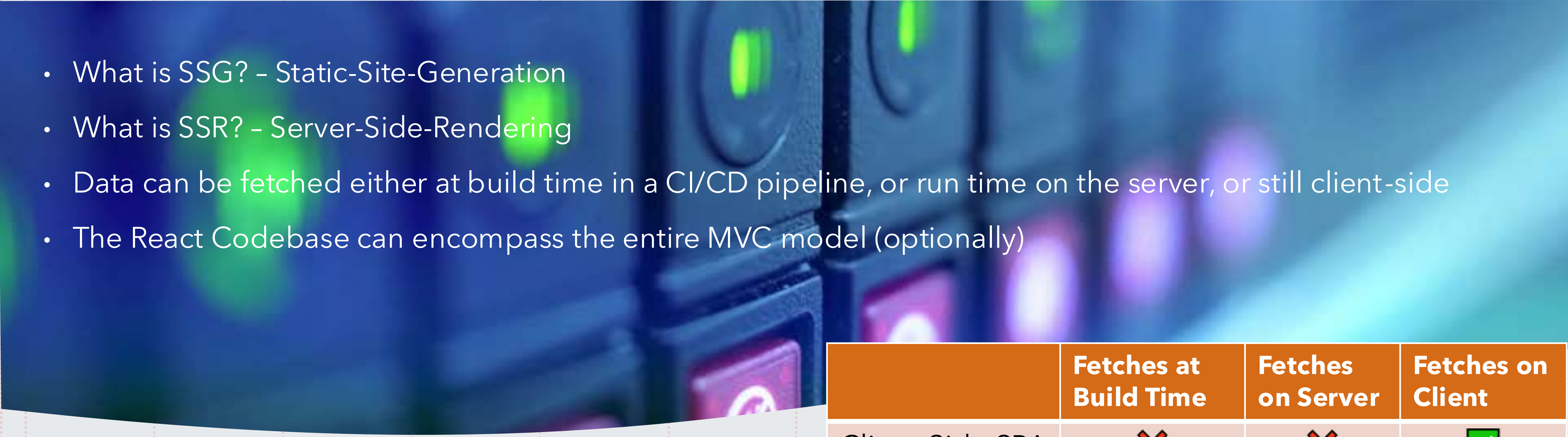


Client-Side SPA examples

- componentDidMount
- useEffect
- React Query (3^{rd} party)

TanStack (React) Query

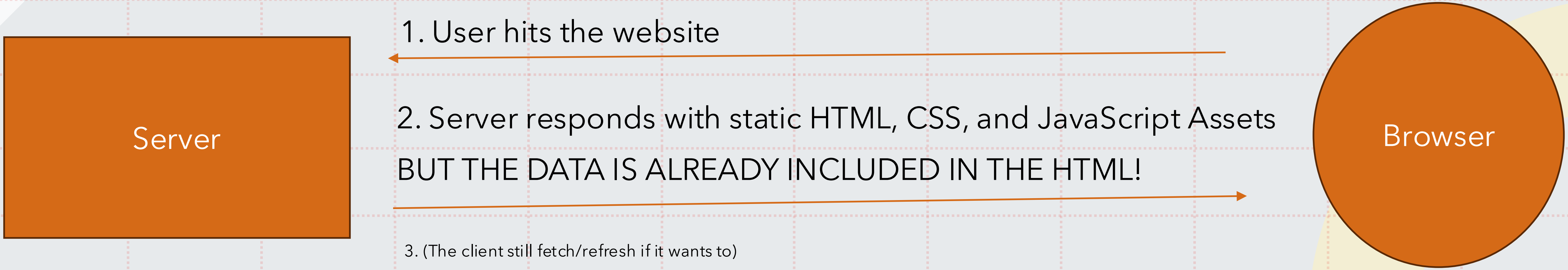
- Why You Want React Query | TkDodo's blog
- We'll take a look at how one of the most popular 3rd party React Libraries can both simplify so much of our code and create better user experiences
- TanStack Query Docs
- Automatic Loading and Reloading states provided
- Automatic Error Handling states
- Adds a caching layer to reduce the total number of fetches
- Prefetching
- Polling and Re-fetching features
- Pagination and infinite scrolling features
- Better memoization re-rendering performance
- Alternatives such as RTK Query, SWC, or Apollo are also good



Chapter 2: SSG and SSR (Back to the Server)

- What is SSG? – Static-Site-Generation
- What is SSR? – Server-Side-Rendering
- Data can be fetched either at build time in a CI/CD pipeline, or run time on the server, or still client-side
- The React Codebase can encompass the entire MVC model (optionally)

	Fetches at Build Time	Fetches on Server	Fetches on Client
Client-Side SPA	❌	❌	✅
SSG	✅	🤔 (ISR)	✅
SSR	❌	✅	✅

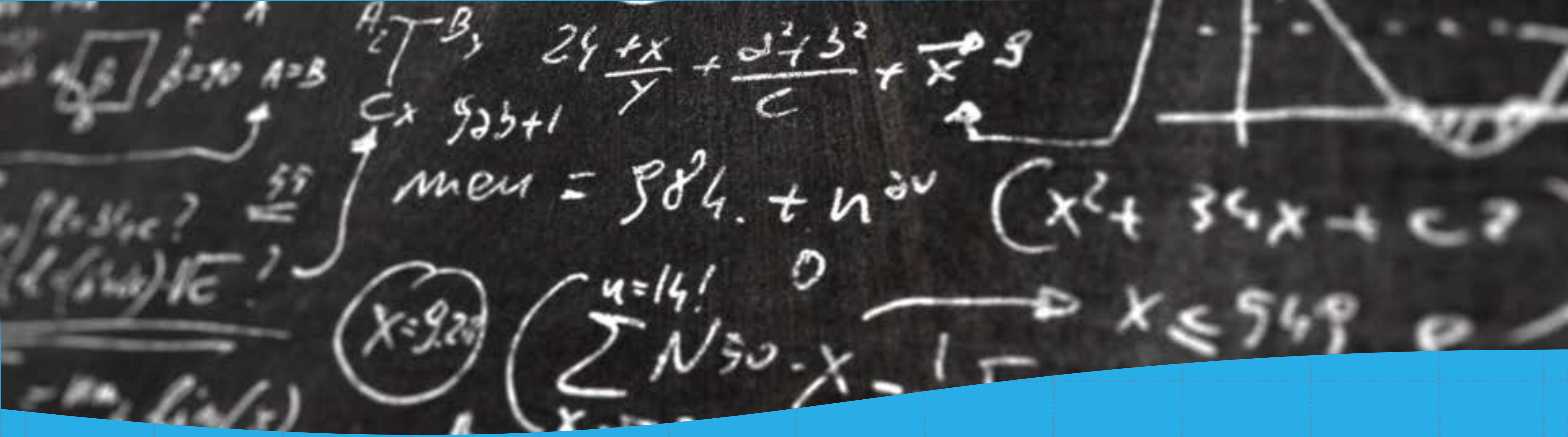


1. User hits the website

2. Server responds with static HTML, CSS, and JavaScript Assets

BUT THE DATA IS ALREADY INCLUDED IN THE HTML!

3. (The client still fetch/refresh if it wants to)



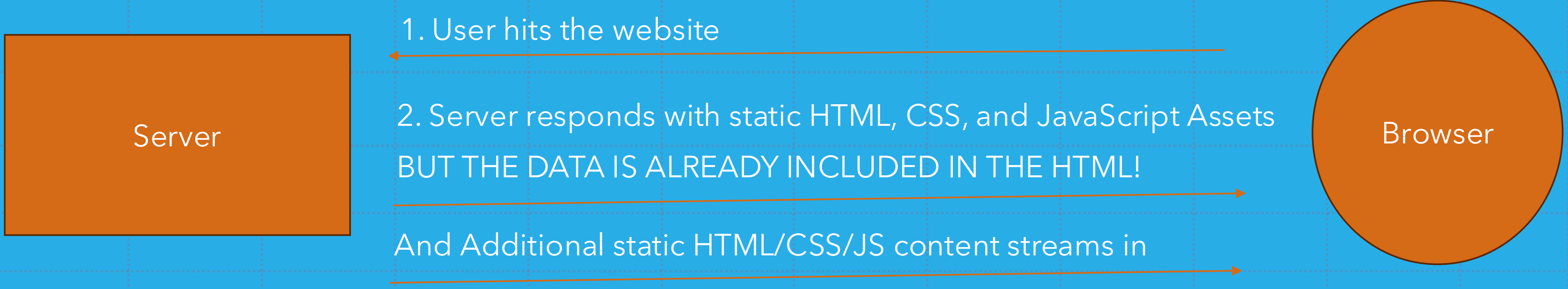
Server-Side Fetching Examples

- Next.js SSG (Static Site Generation
- Next.js SSR (Server-Side Rendering)
- Remix SSR (Server-Side Rendering)

Chapter 3: RSCs (React Server Components)

- What are RSCs? – React Server Components
- React Components that ONLY FETCH ON THE SERVER (And only run on the server in general)
- Fetch on a per server component basis instead of being limited to fetching per page/route
- Results in a reduced bundle size, because RSCs don't run any* client-side JavaScript
- New HTML can stream in (instead of JSON fetches)
- Currently being pioneered by Next.js App Router

	Fetches at Build Time	**Fetches on Server**	**Fetches on Client**
Client-Side SPA	❌	❌	✅
SSG	✅	🤔	✅
SSR	❌	✅	✅
RSCs	🤔	✅	❌ *

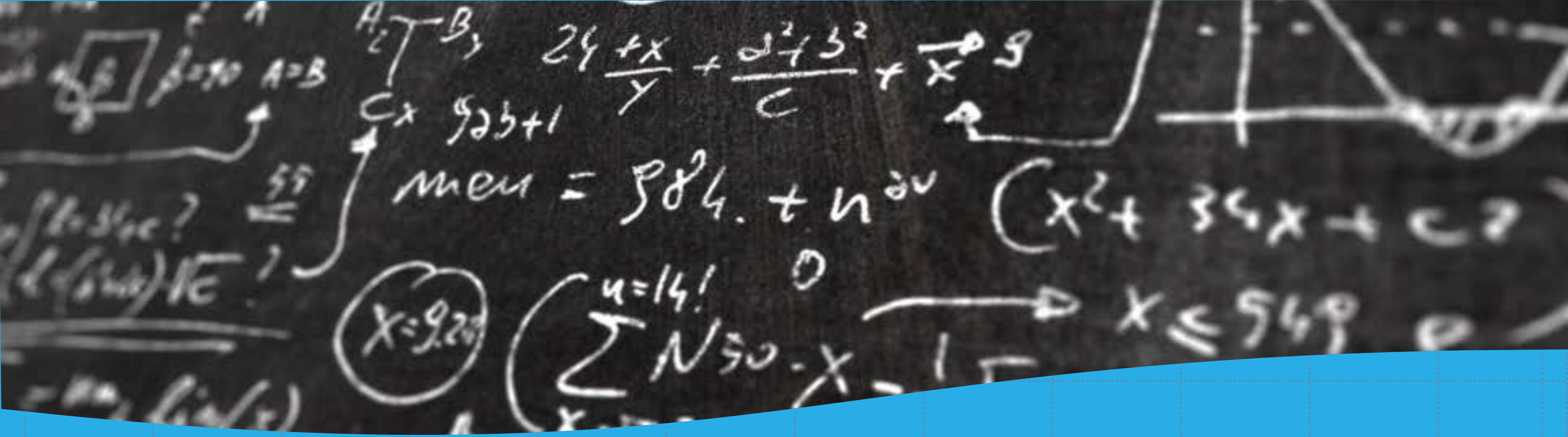


1. User hits the website

2. Server responds with static HTML, CSS, and JavaScript Assets

BUT THE DATA IS ALREADY INCLUDED IN THE HTML!

And Additional static HTML/CSS/JS content streams in



React Server Components example

Conclusions

- What types of Apps should be built as SPAs?
- What types of Apps should be built with SSG?
- What types of Apps should be built with SSR?
- What types of Apps should be built with RSCs?

Ship Faster Websites with React Query

Nebraska JS – September 18

Code and Slides