Software University

HEROES OF JS WORLD 2024

NEXT.JS - нещо повече от Front-End Framework

Цветомир Лазаров, CEO @ QualiSage

It’s 2016

STORY TIME

First Stable React Version

April 2016

- React 15.0.0 released
- ~390k weekly downloads

November 2016

- ~650k weekly downloads

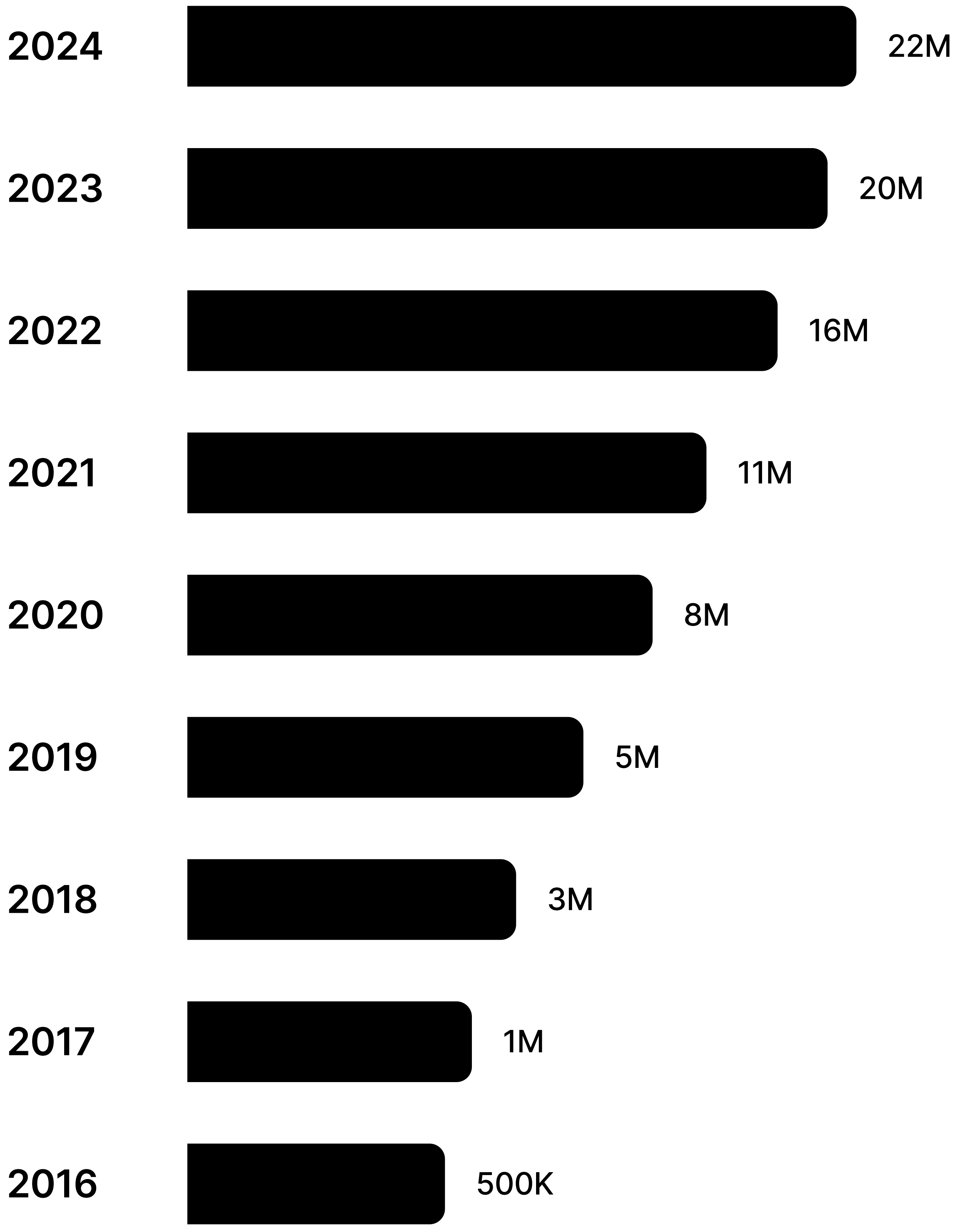
Major Drawback

Works only client-side ☹️

```
<div id="root"></div>
```

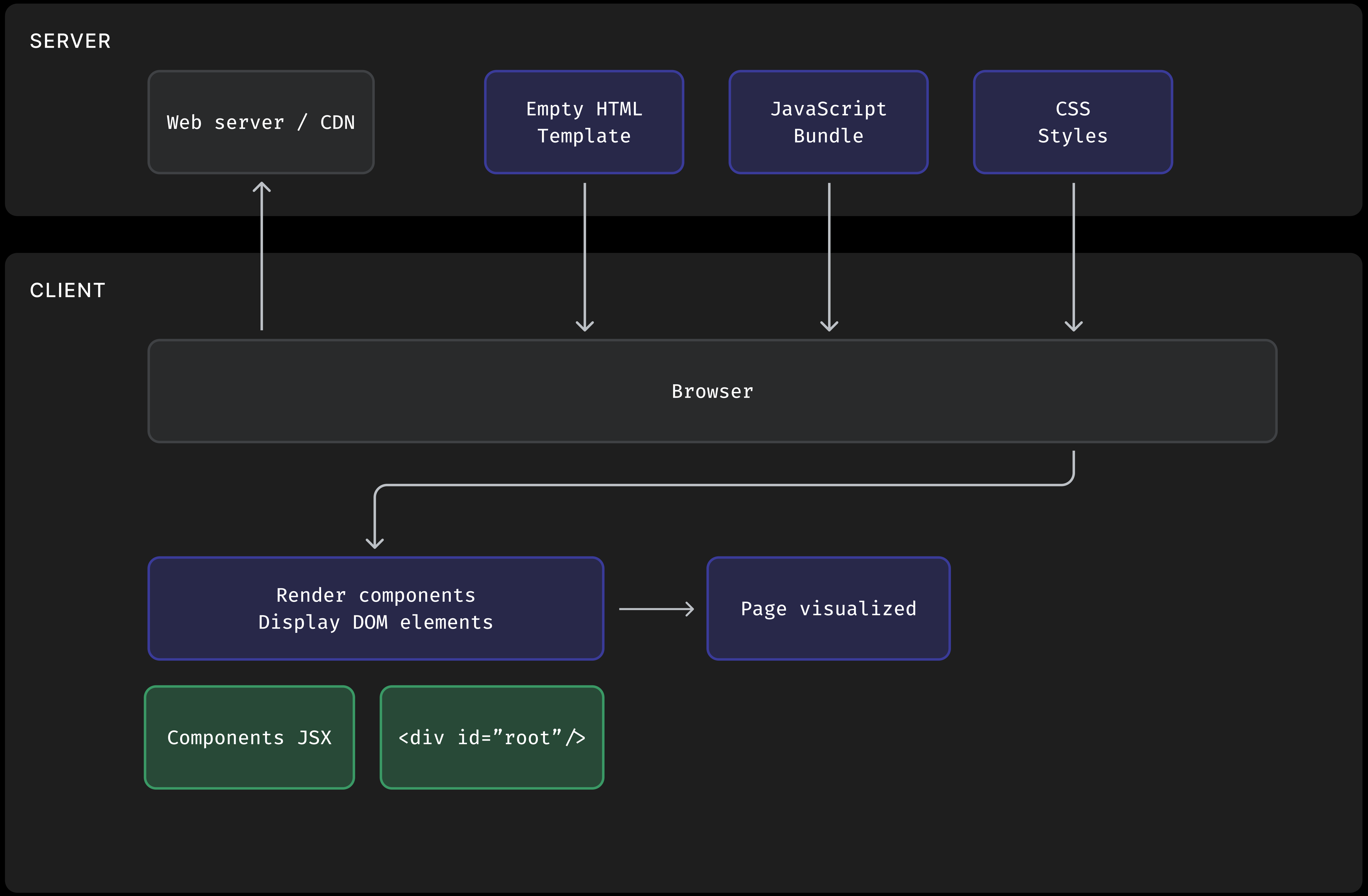
REACT WEEKLY DOWNLOADS

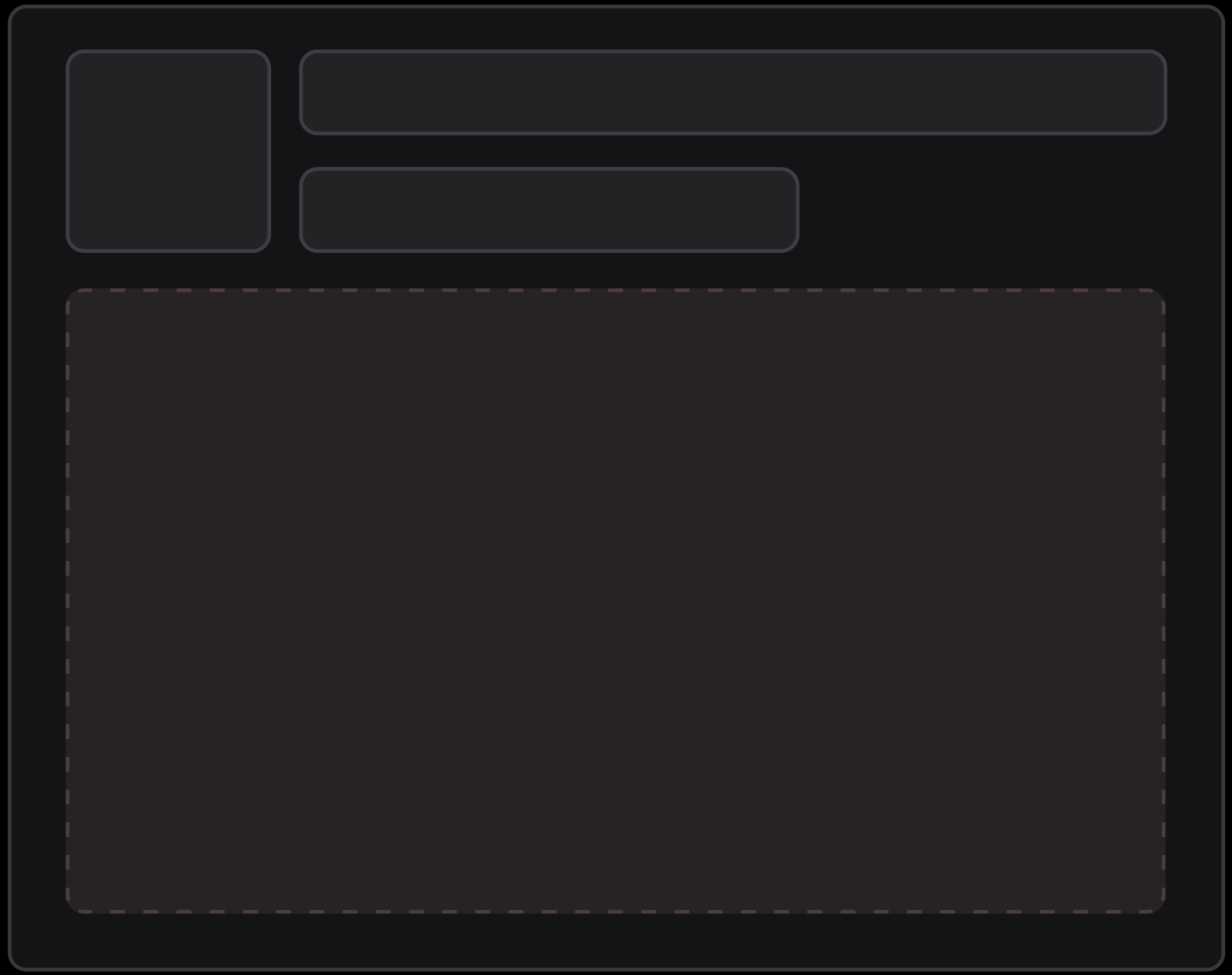
MID-YEAR VALUES, APPROXIMATE



RETROSPECT

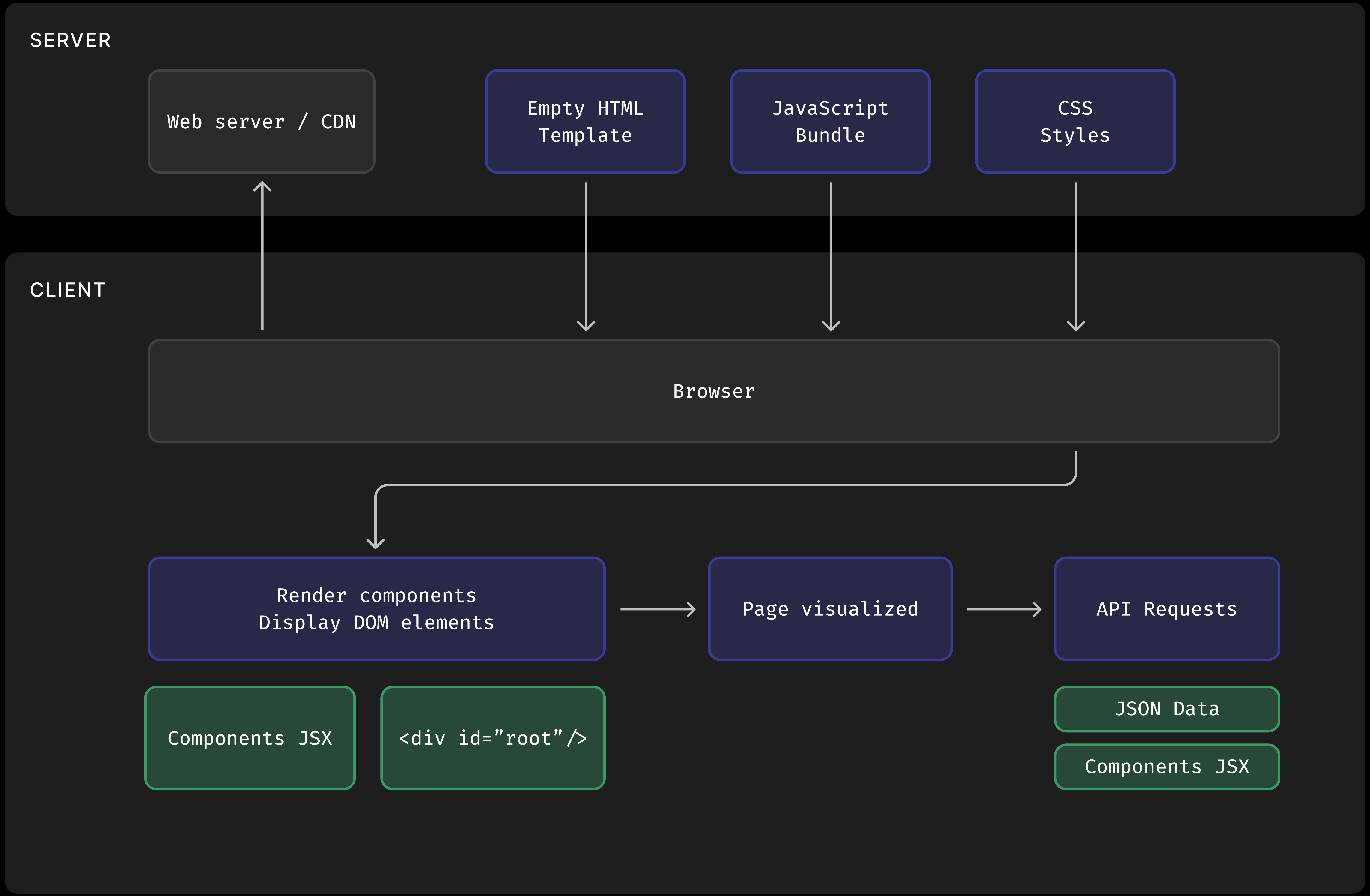
How Does Client-Side Rendering Work?





RETROSPECT

How Does Client-Side Rendering Work?



SERVER

Web server / CDN

Empty HTML Template

JavaScript Bundle

CSS Styles

CLIENT

Browser

Render components
Display DOM elements

Page visualized

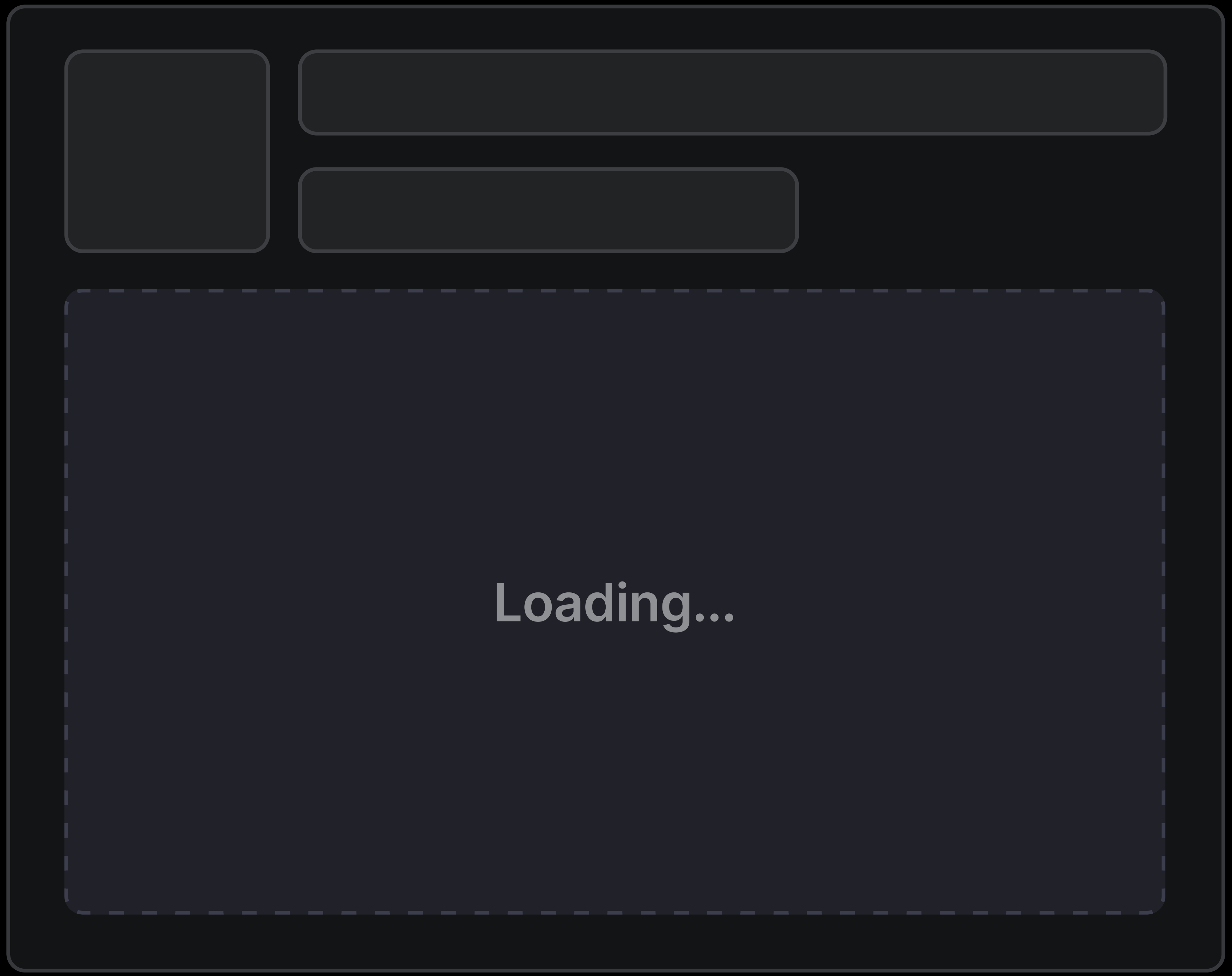
API Requests

Components JSX

<div id=”root”/>

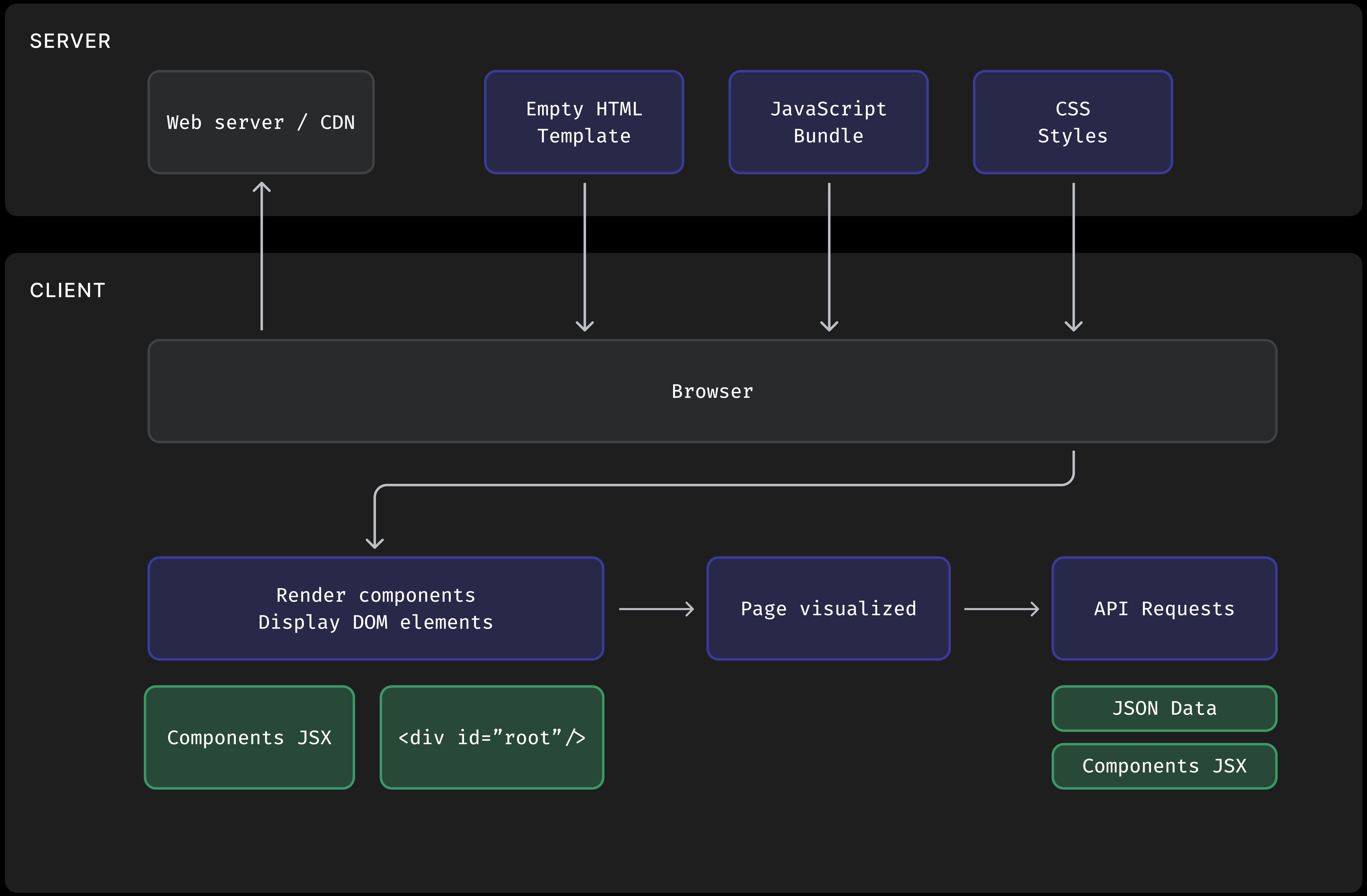
JSON Data

Components JSX



RETROSPECT

How Does Client-Side Rendering Work?



SERVER

Web server / CDN

Empty HTML Template

JavaScript Bundle

CSS Styles

CLIENT

Browser

Render components
Display DOM elements

Page visualized

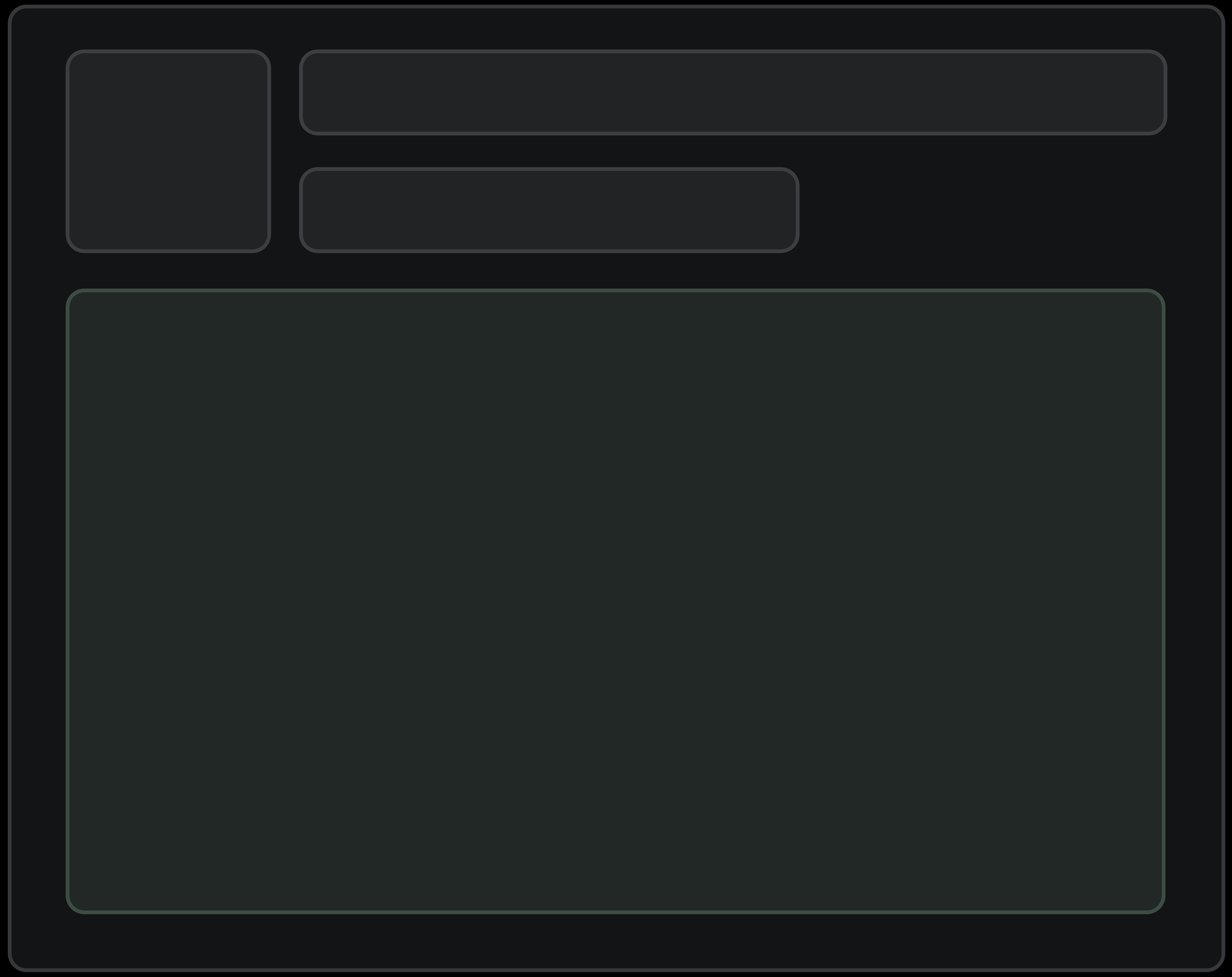
API Requests

Components JSX

<div id=”root”/>

JSON Data

Components JSX



RETROSPECT

Client-Side Rendering Does The Job

🐌 Time to first paint and initial interaction is slow.

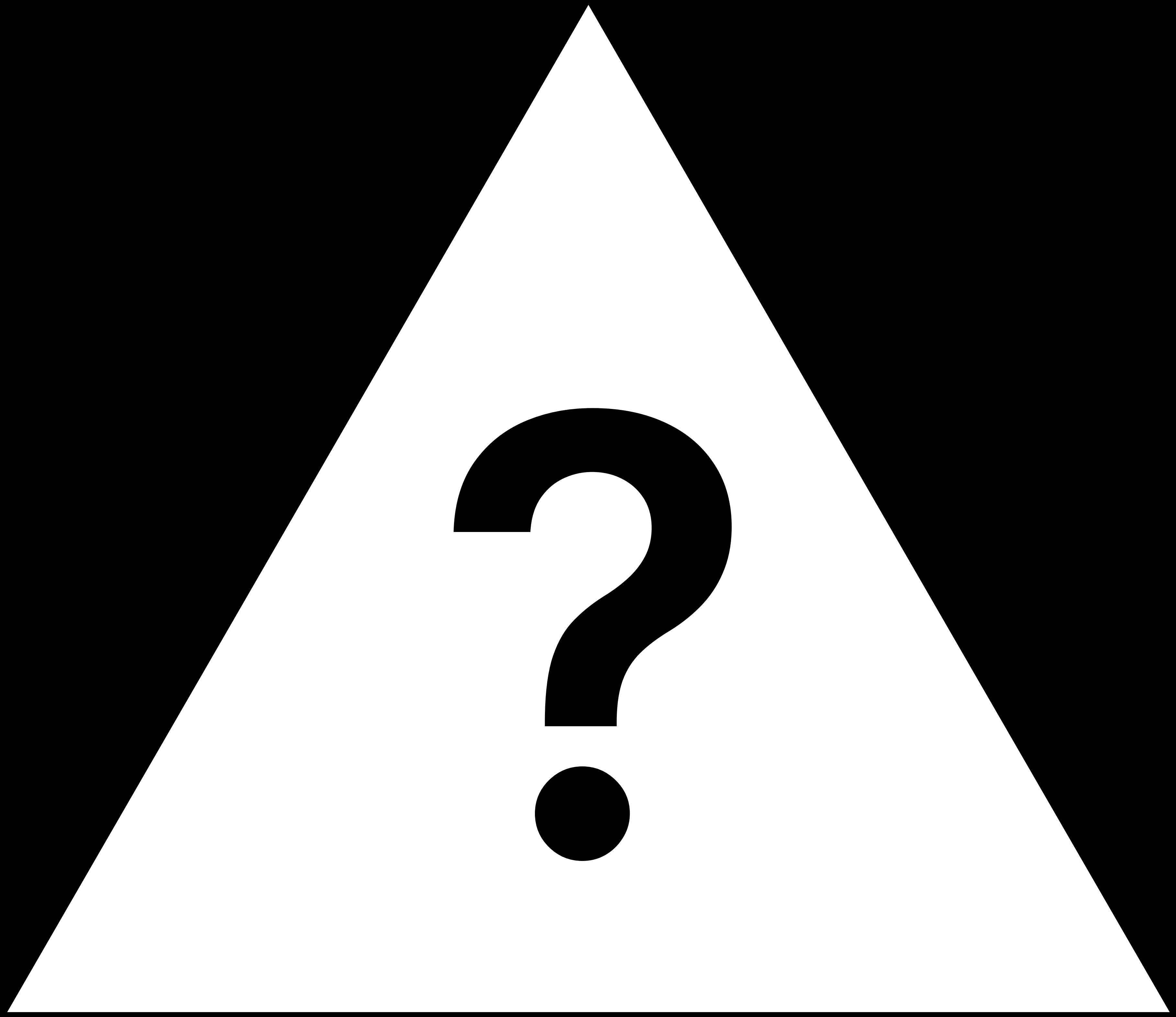
😕 User experience is not great.

🤔 But it works for most developers, so why bother?

¯_(ツ)_/¯

WHY BOTHER?

Think Again



It’s still 2016

STORY TIME

React is not doing enough

ZEIT identified the main pain points of the library:

1. UX (User Experience)

`Performance` `SEO`

2. DX (Developer Experience)

`Setup` `Configuration` `Deployment`

Vercel

Guillermo Rauch, CEO of Vercel (ex ZEIT)

Oct 25, 2016

STORY TIME

ZEIT fixed it and started a revolution

Vercel

Guillermo Rauch, CEO of Vercel (ex ZEIT)

Hello

ABOUT MYSELF

Tsvetomir (Tsetsi) Lazarov

Full-Stack Developer

Team Lead

CTO

CEO

TypeScript

React

Next.js

Node.js

Go

PHP

...

Coding as a <hobby/>

QualiSage

Tsetsi Lazarov, CEO of QualiSage

Hello

ABOUT MYSELF

Tsvetomir (Tsetsi) Lazarov

Full-Stack Developer

Team Lead

CTO

CEO

TypeScript

React

Next.js

Node.js

Go

PHP

...

Coding as a <hobby/>

QualiSage

Tsetsi Lazarov, CEO of QualiSage

AGENDA

What We'll Learn

History of React

History of Next.js

Client-Side Rendering

What Is Next.js

How to Install & Run It

Server vs Client Components

Project Structure

App Router

File-Based Routing

Layout & Pages

Dynamic Routes

Navigation

Loading UI

Error Handling

Route Groups

Route Handlers

Streaming

Fetching Data

Mutating Data

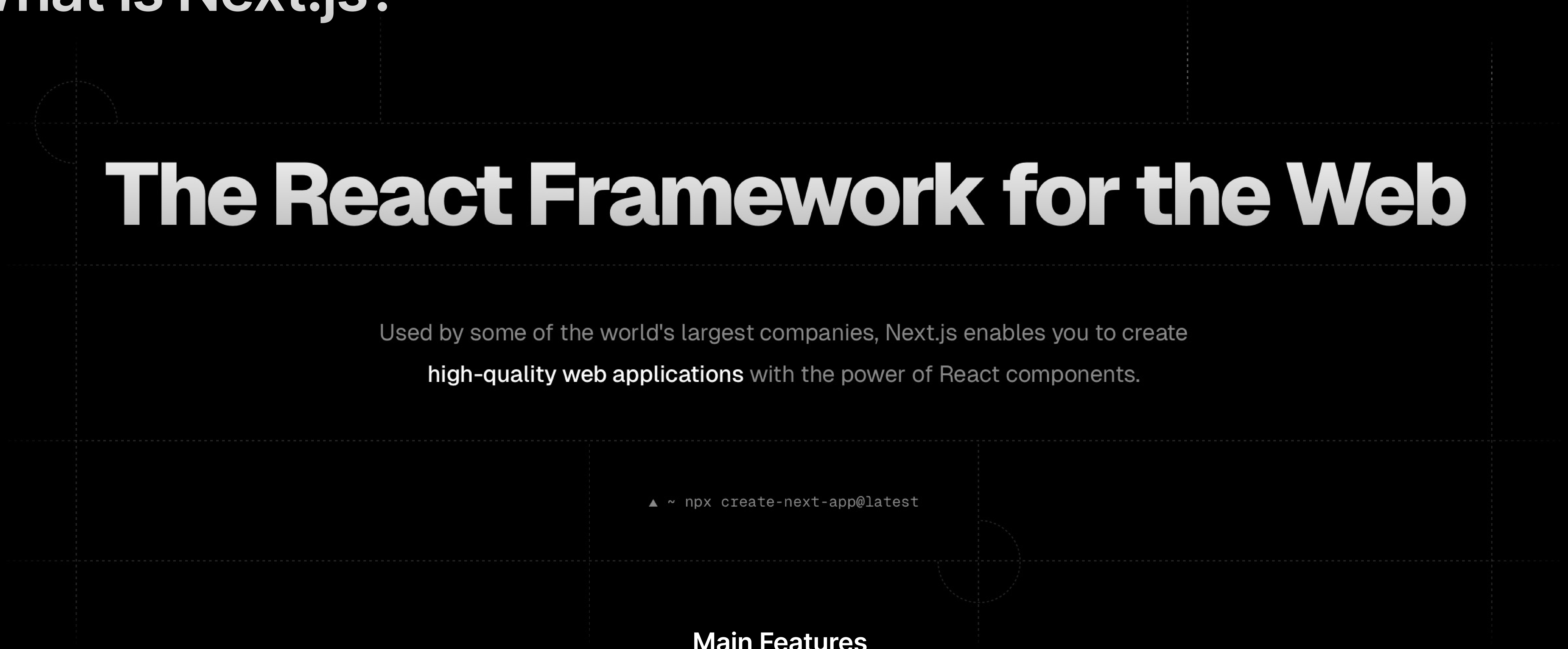
Server Actions

What's New In Next.js 15

The Things I Wished We Had More Time To Talk About

GETTING STARTED

What is Next.js?



The React Framework for the Web

Used by some of the world's largest companies, Next.js enables you to create **high-quality web applications** with the power of React components.

```
▲ ~ npx create-next-app@latest
```

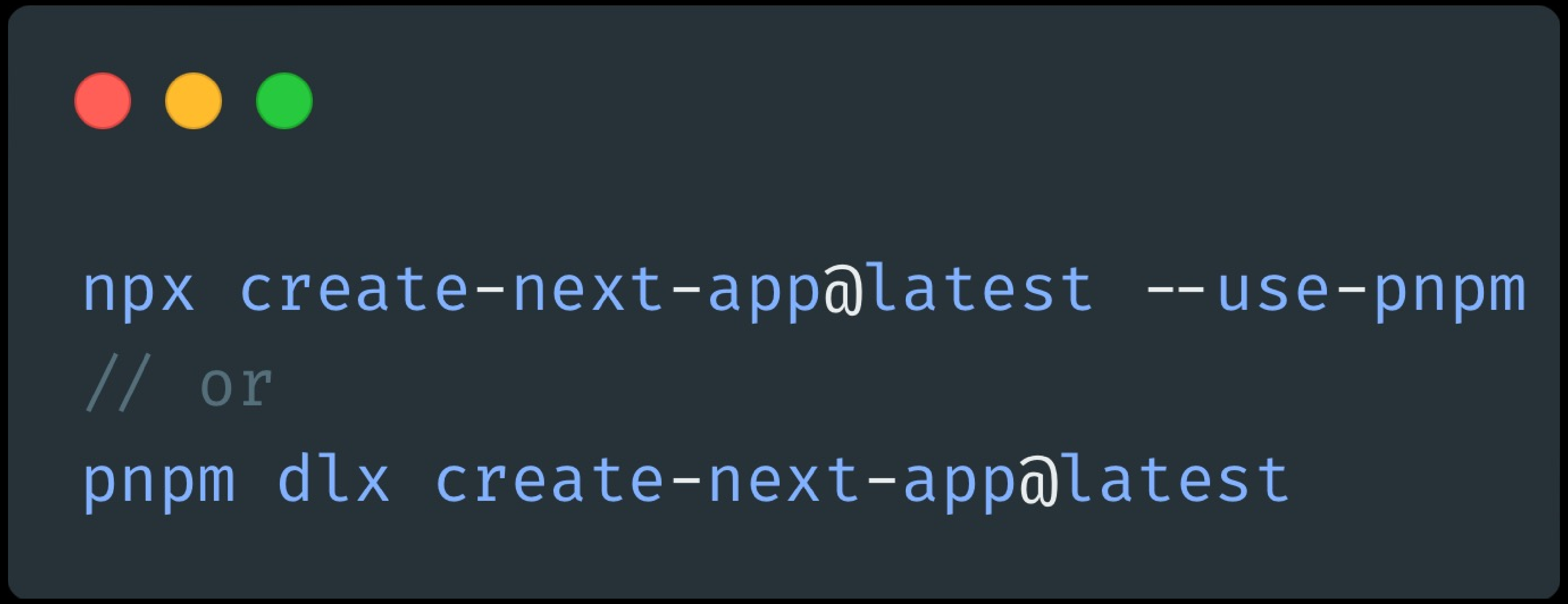
Main Features

Routing | Rendering | Data Fetching | Styling | Optimizations | TypeScript

GETTING STARTED

Installation & Running

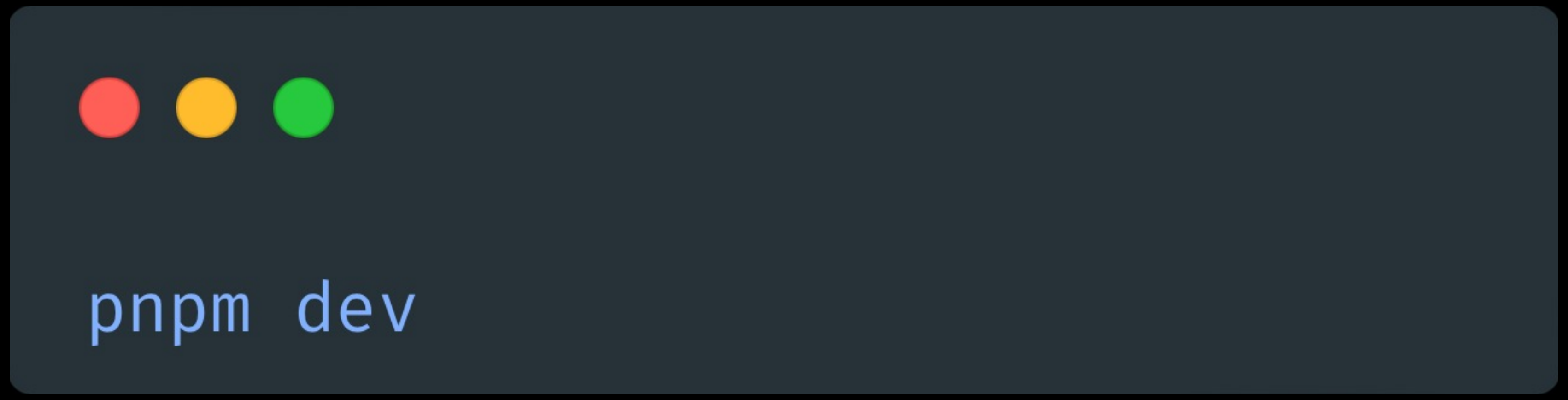
Install with pnpm



```
npx create-next-app@latest --use-pnpm
// or
pnpm dlx create-next-app@latest
```

* Next.js recommends pnpm, because it’s faster & more efficient than npm or yarn.

Run the development server



```
pnpm dev
```

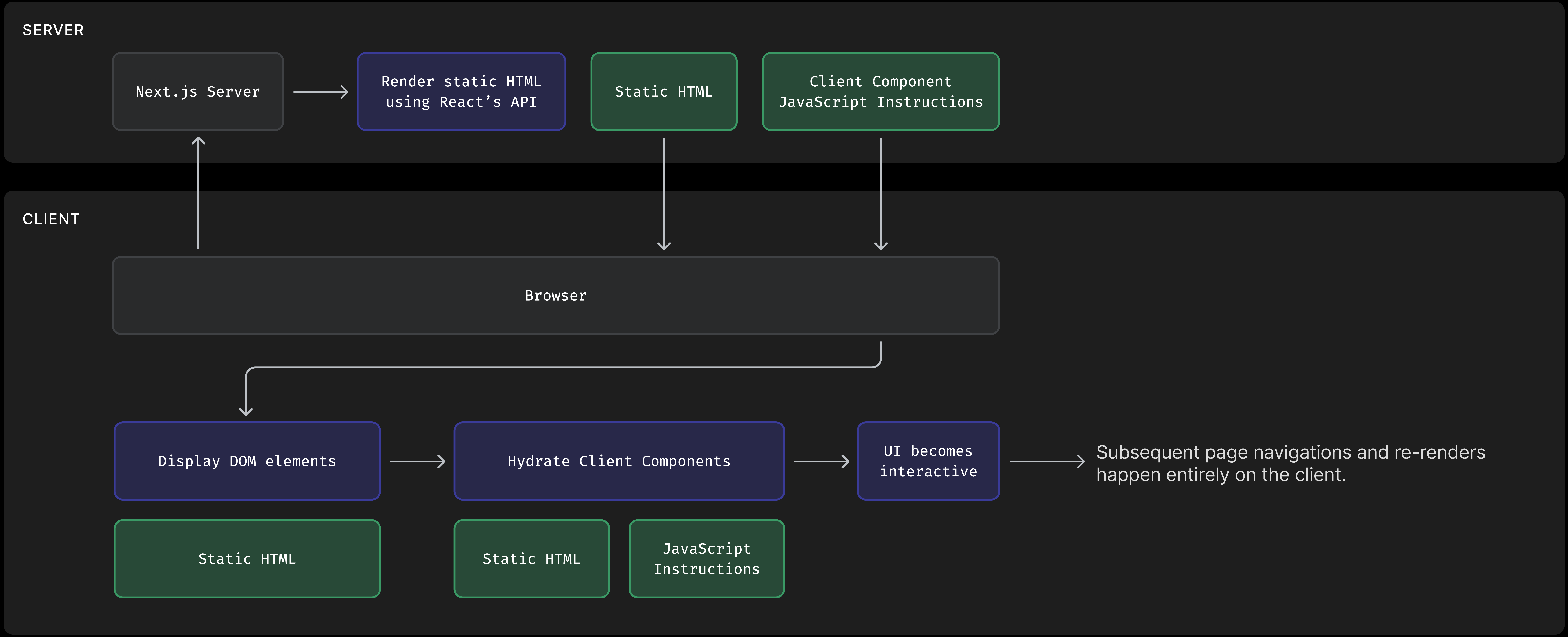
FIRST THINGS FIRST

Server vs Client Components in Next.js

FIRST THINGS FIRST

Server vs Client Components in Next.js

Client components (this is also how all components worked before the app router)

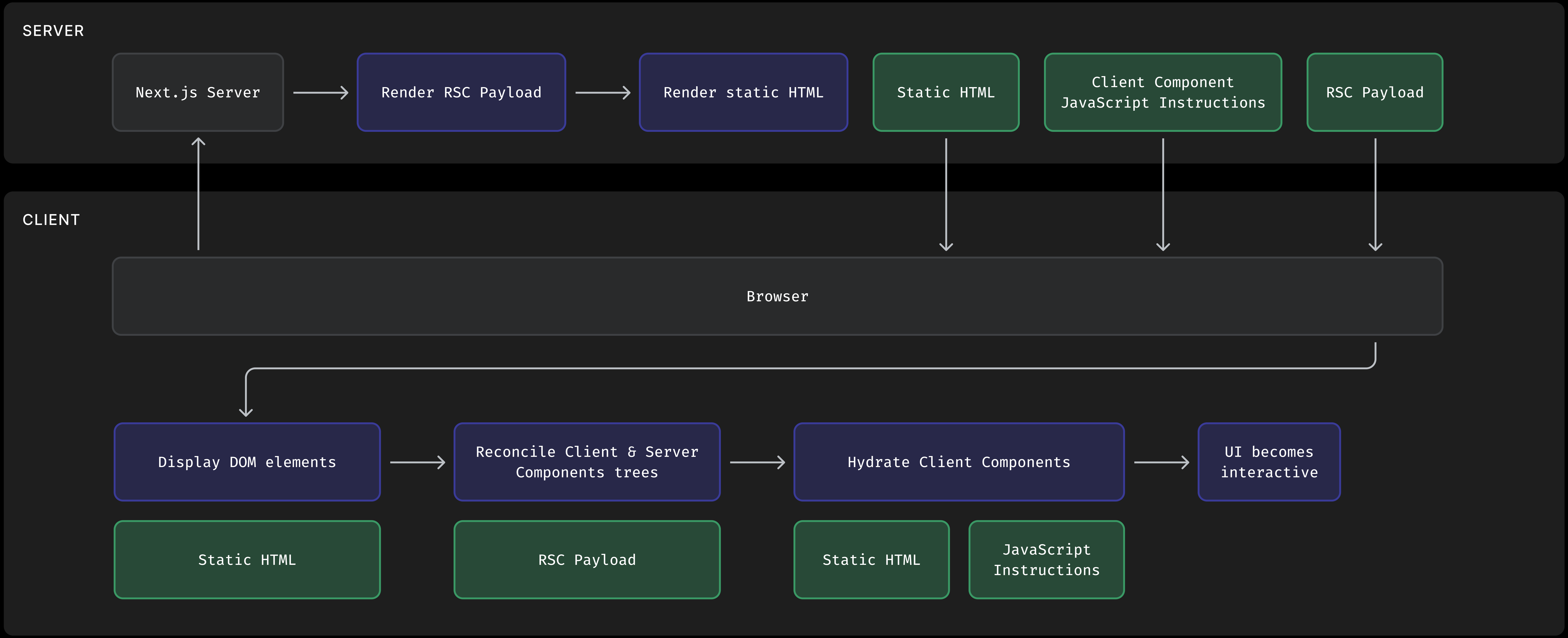


Subsequent page navigations and re-renders happen entirely on the client.

FIRST THINGS FIRST

Server vs Client Components in Next.js

Server components



FIRST THINGS FIRST

Server vs Client Components in Next.js

How to decide?

You need to...	Server Component	Client Component
Fetch data	🟢	❗
Access backend resources directly	🟢	❌
Access sensitive information, e.g. access tokens, API keys	🟢	❌
Install large dependencies that are needed only for non-interactable content	🟢	❌
Add interactivity, e.g. event listeners	❌	🟢
Use React hooks like useState, useEffect, etc.	❌	🟢
Access the browser API	❌	🟢
Use custom hooks that use React hooks or access the browser API	❌	🟢
Use class components 🥲	❌	🟢

TAKING A CLOSER LOOK

Project Structure

Top-level directories and files

Next.js:

- app/
- public/
- src/
- next.config.js
- next-env.d.ts
- middleware.ts
- instrumentation.ts

Tooling:

- .env, .env.*
- .eslintrc.json
- tsconfig.json
- .gitignore
- package.json
- pnpm-lock.yaml
- node_modules

Routing files

Common:

- layout.tsx
- page.tsx
- loading.tsx
- error.tsx
- route.ts

Other:

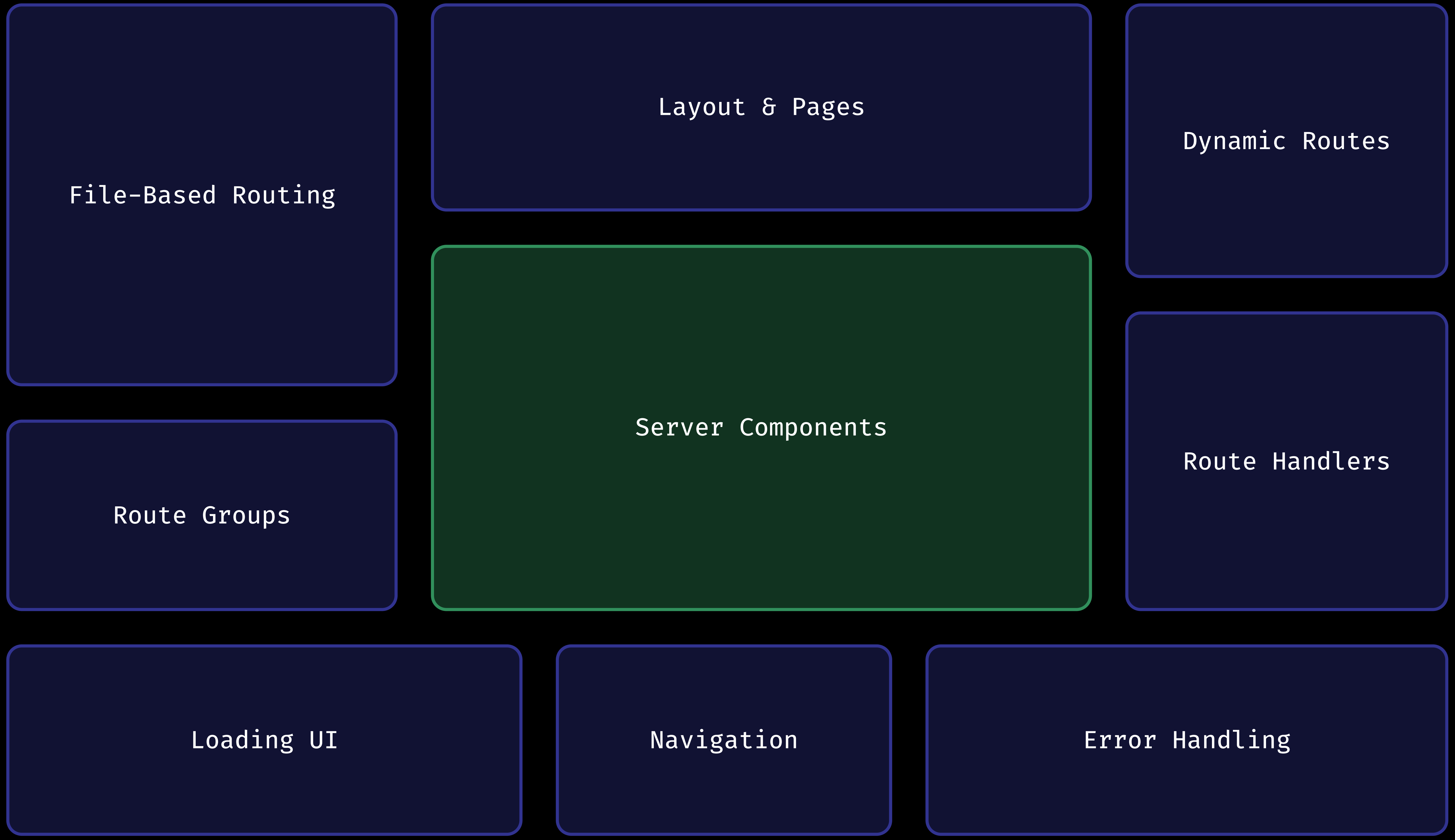
- not-found.tsx
- global-error.tsx
- template.tsx
- default.tsx

Metadata

- Favicon
- OpenGraph & Twitter Images
- Sitemap
- Robots
- Manifest

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

Route Handlers

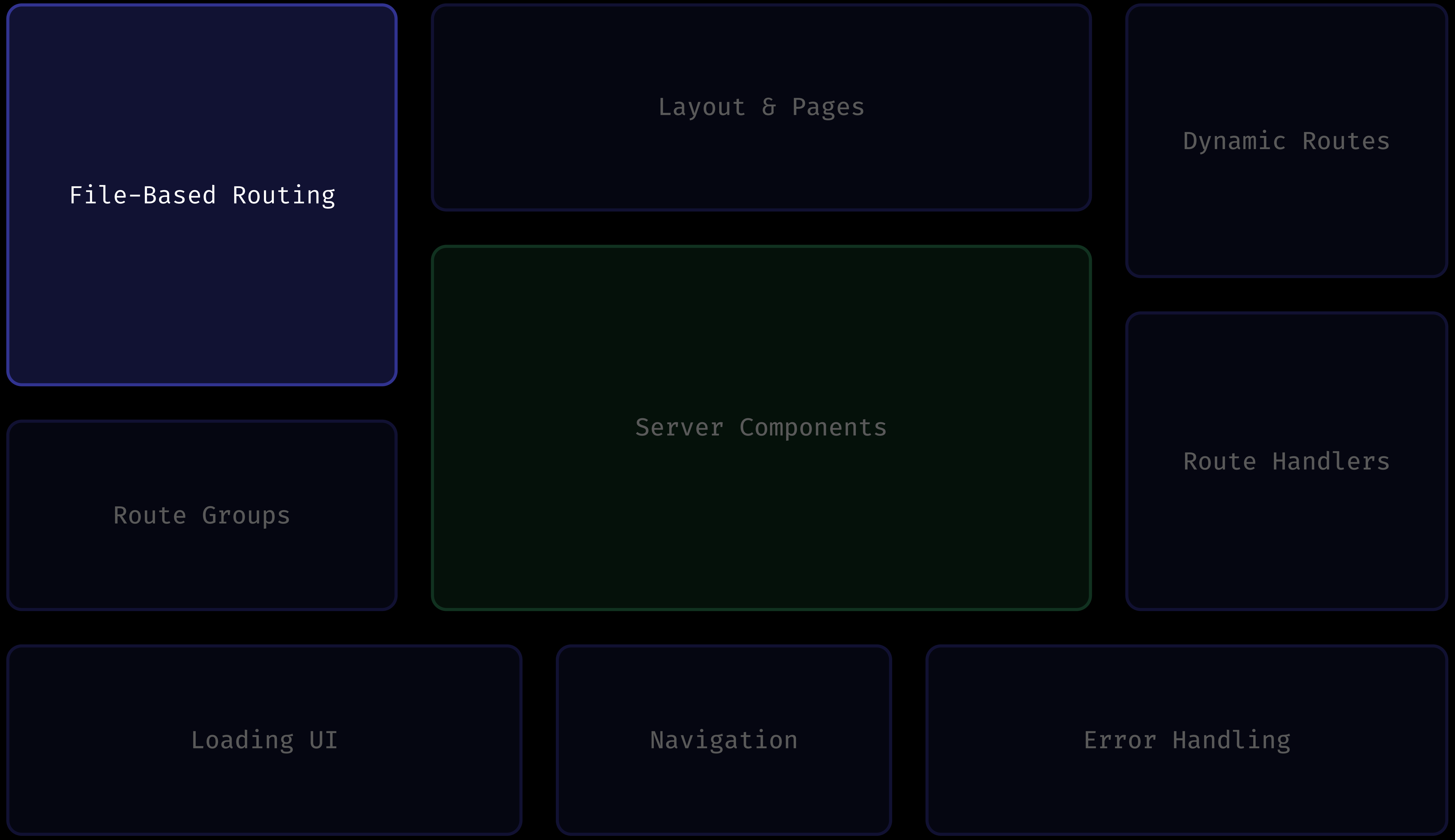
Loading UI

Navigation

Error Handling

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

Route Handlers

Loading UI

Navigation

Error Handling

every page.tsx defines a route

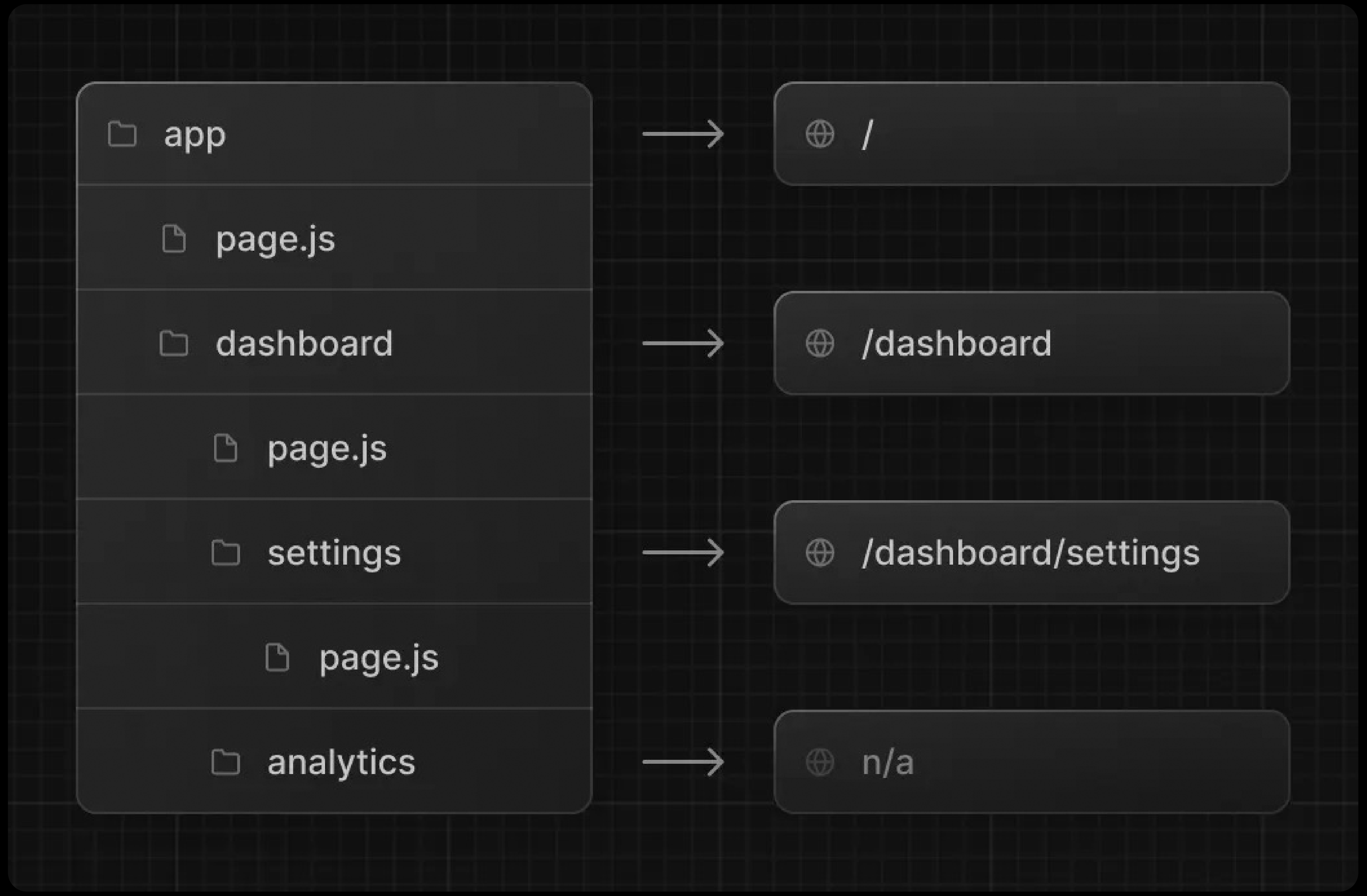
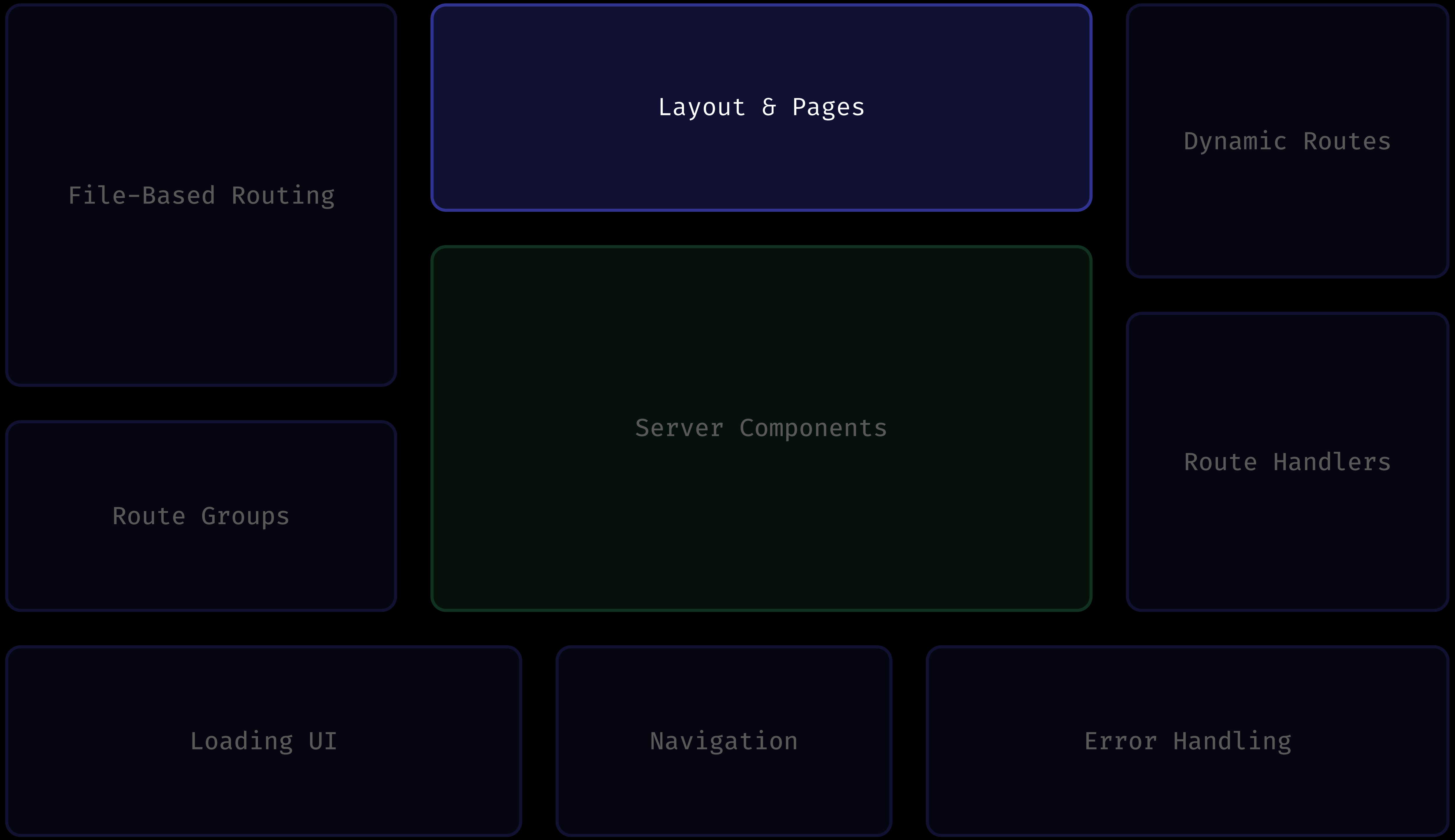


Image © 2024 Vercel, Inc.

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

Route Handlers

Loading UI

Navigation

Error Handling

Layout = Shared UI

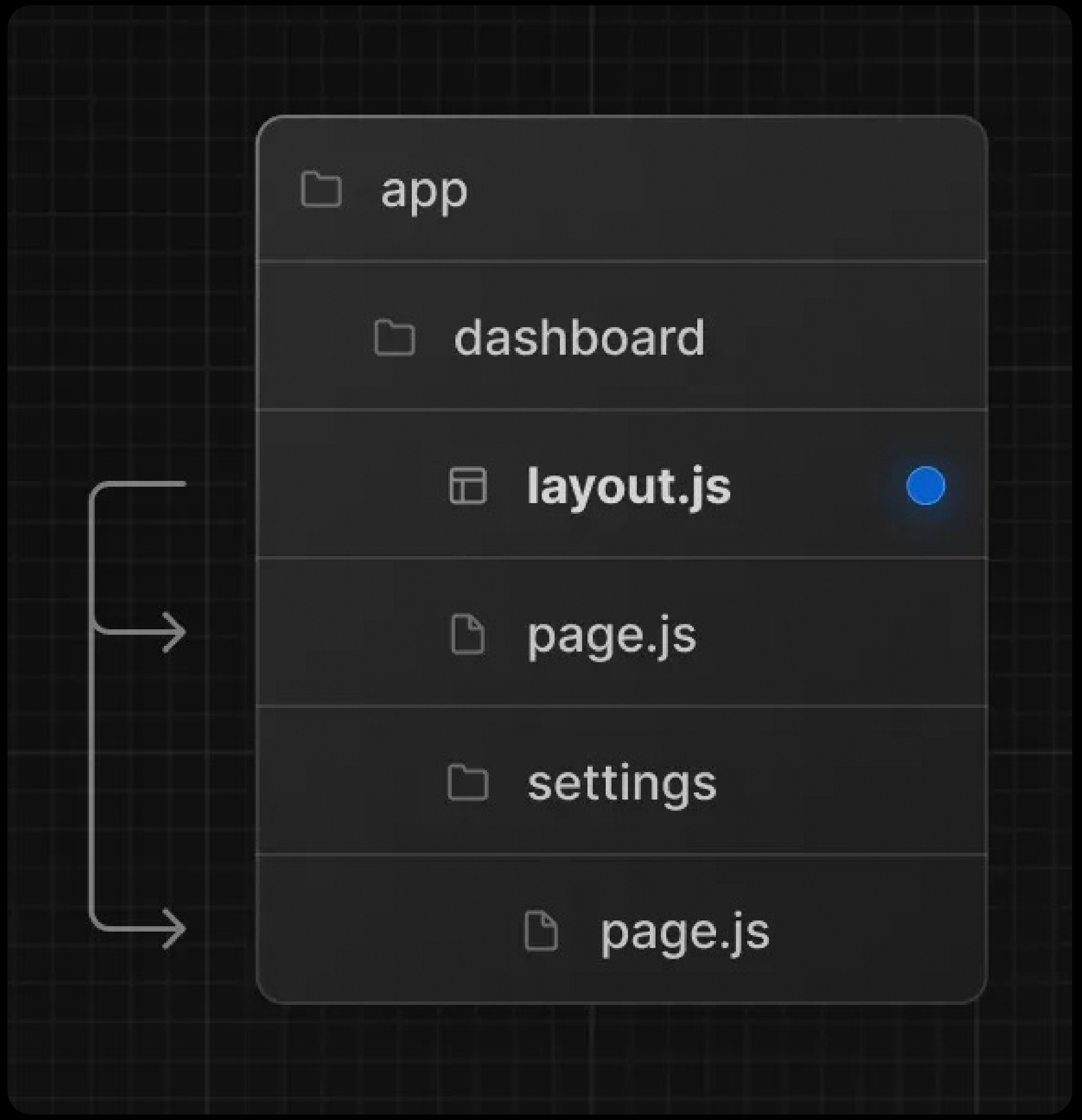
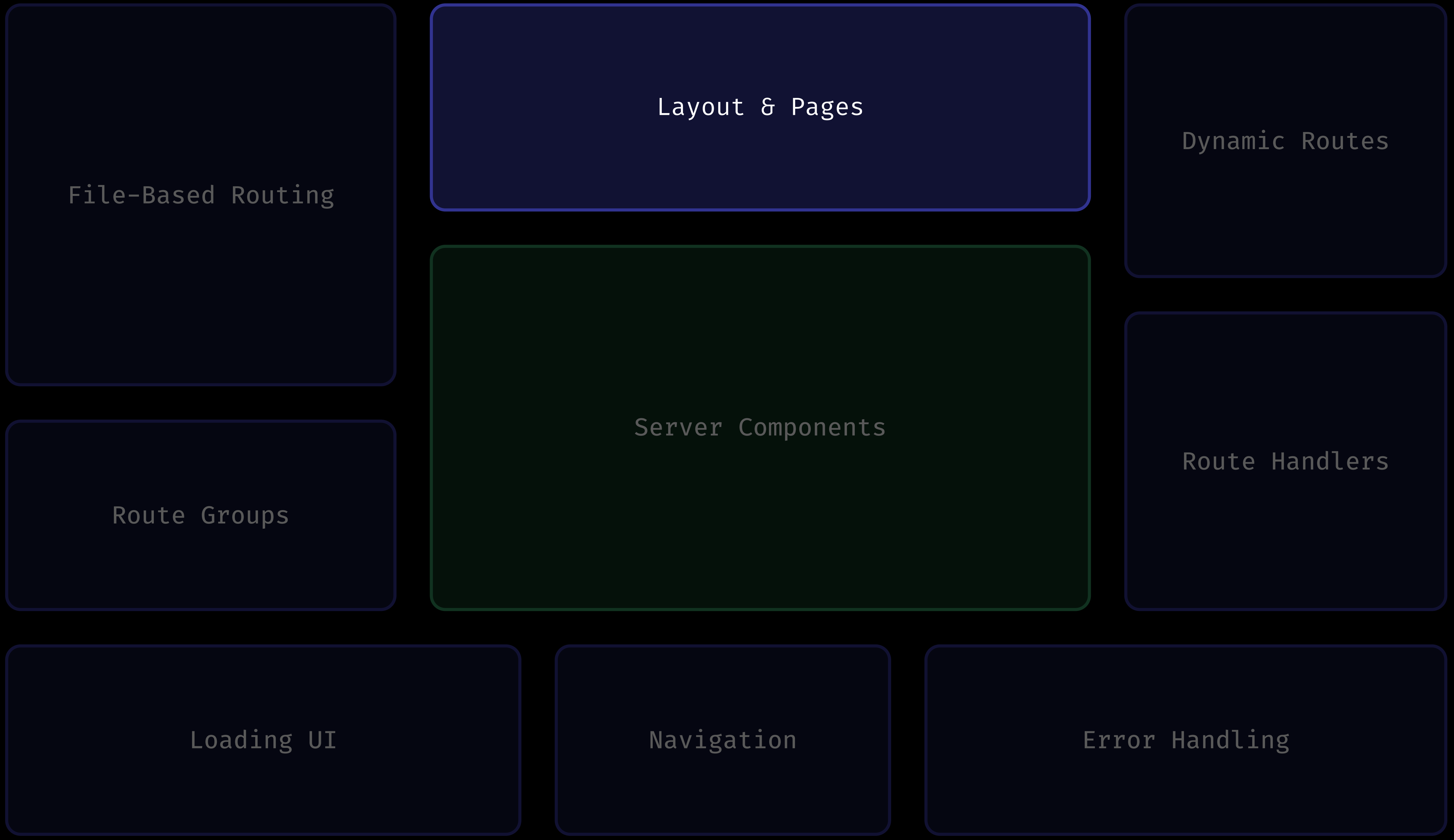


Image © 2024 Vercel, Inc.

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Handlers

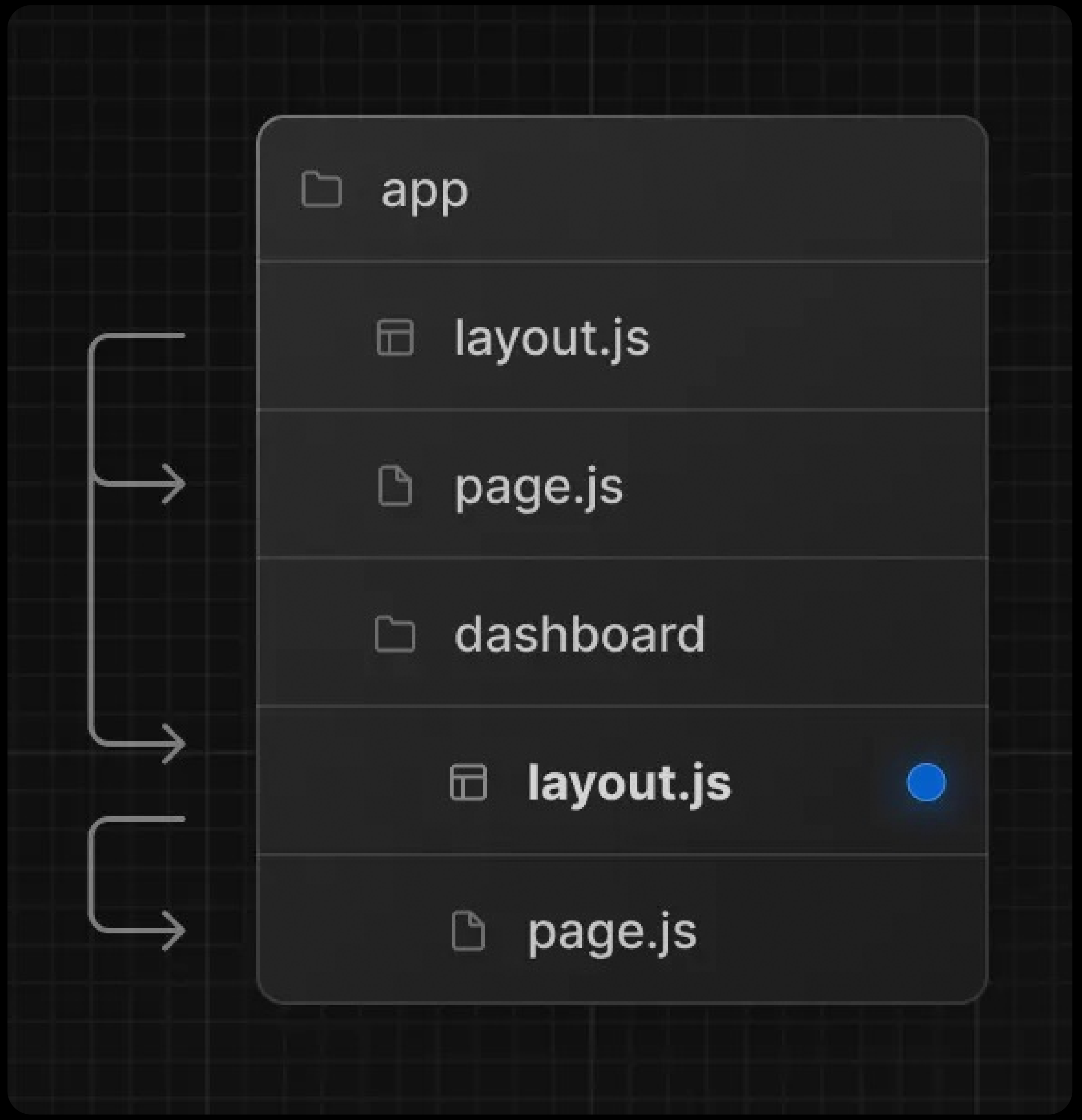
Route Groups

Loading UI

Navigation

Error Handling

Root Layout, Nested Layout



NUTS AND BOLTS

The App Router

File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

Route Handlers

Loading UI

Navigation

Error Handling

Pages define browseable routes

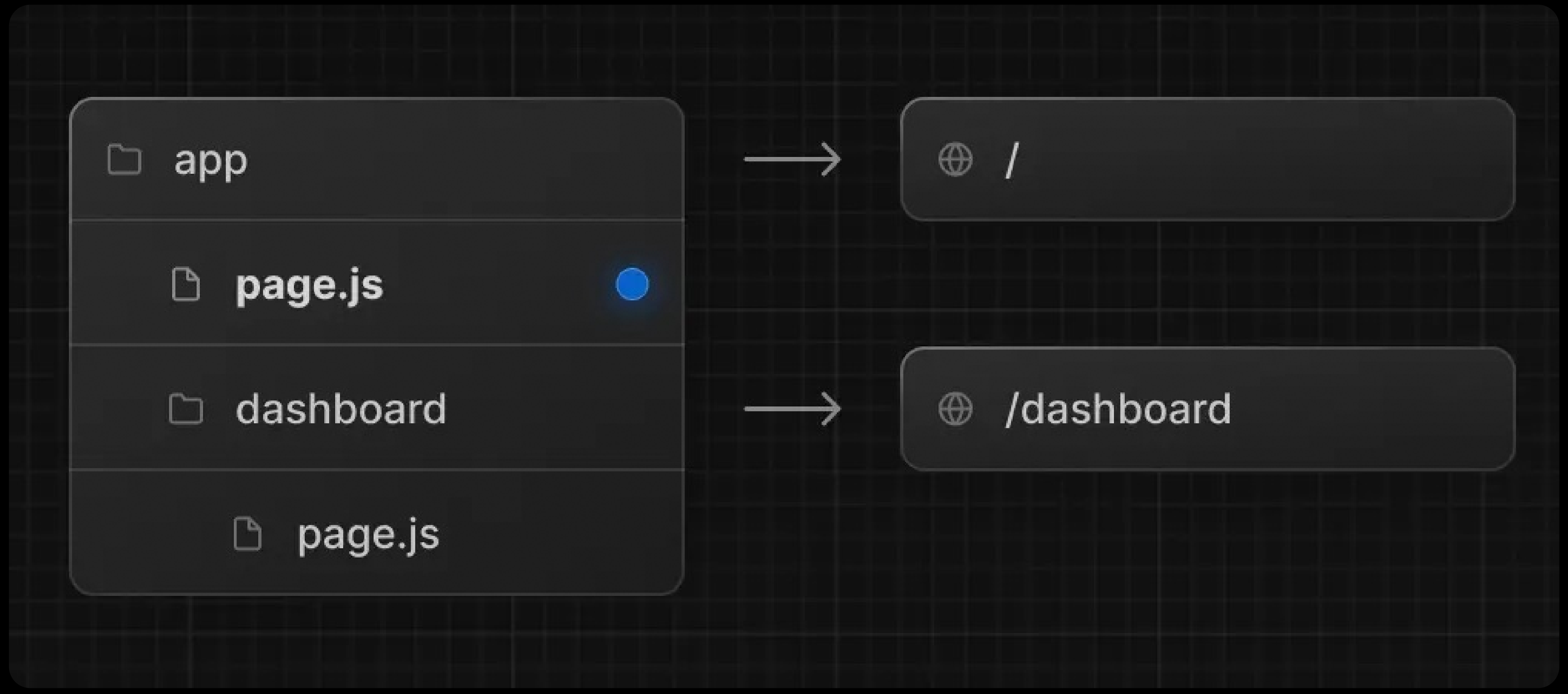
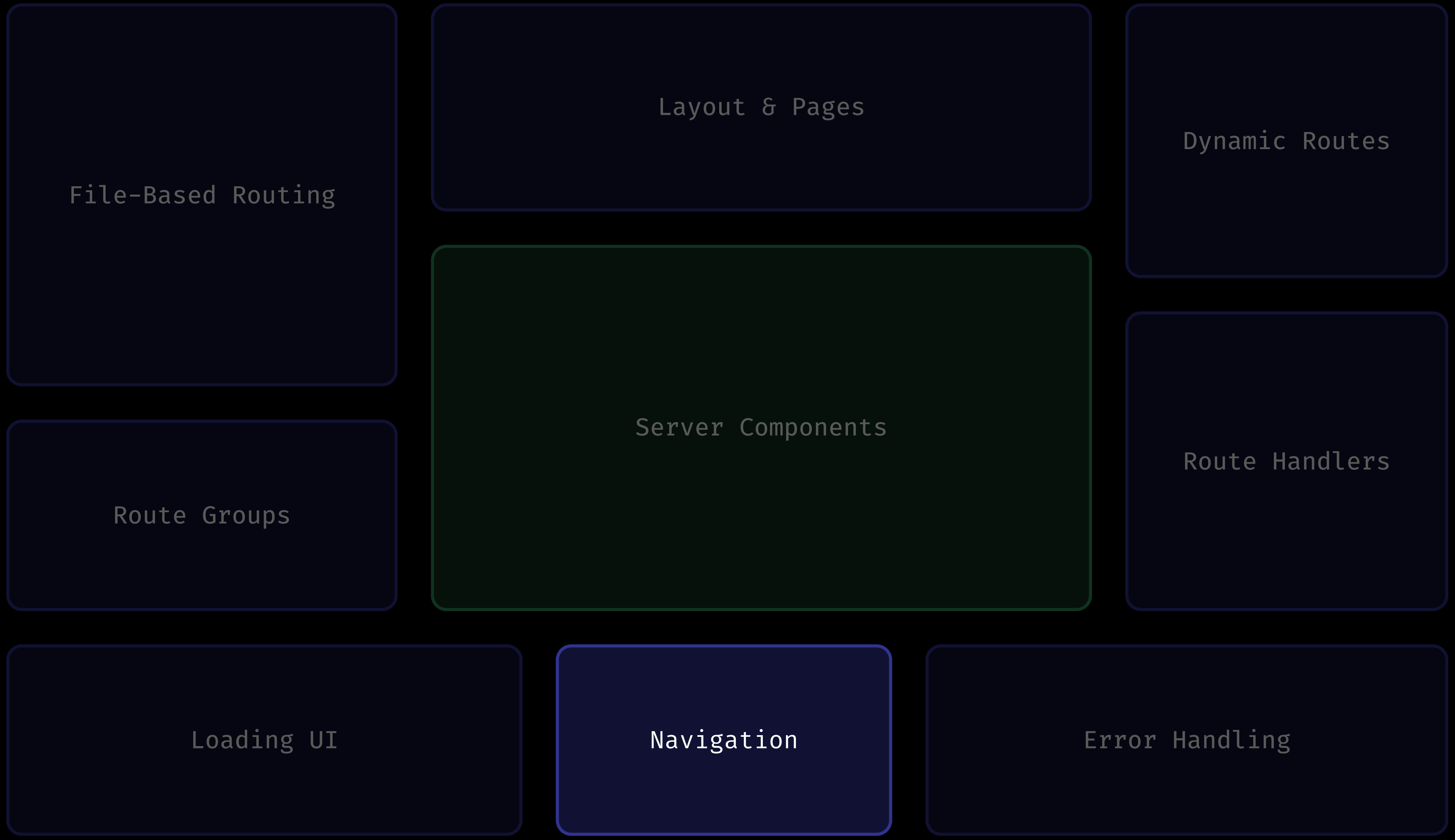


Image © 2024 Vercel, Inc.

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

Route Handlers

Loading UI

Navigation

Error Handling

server client

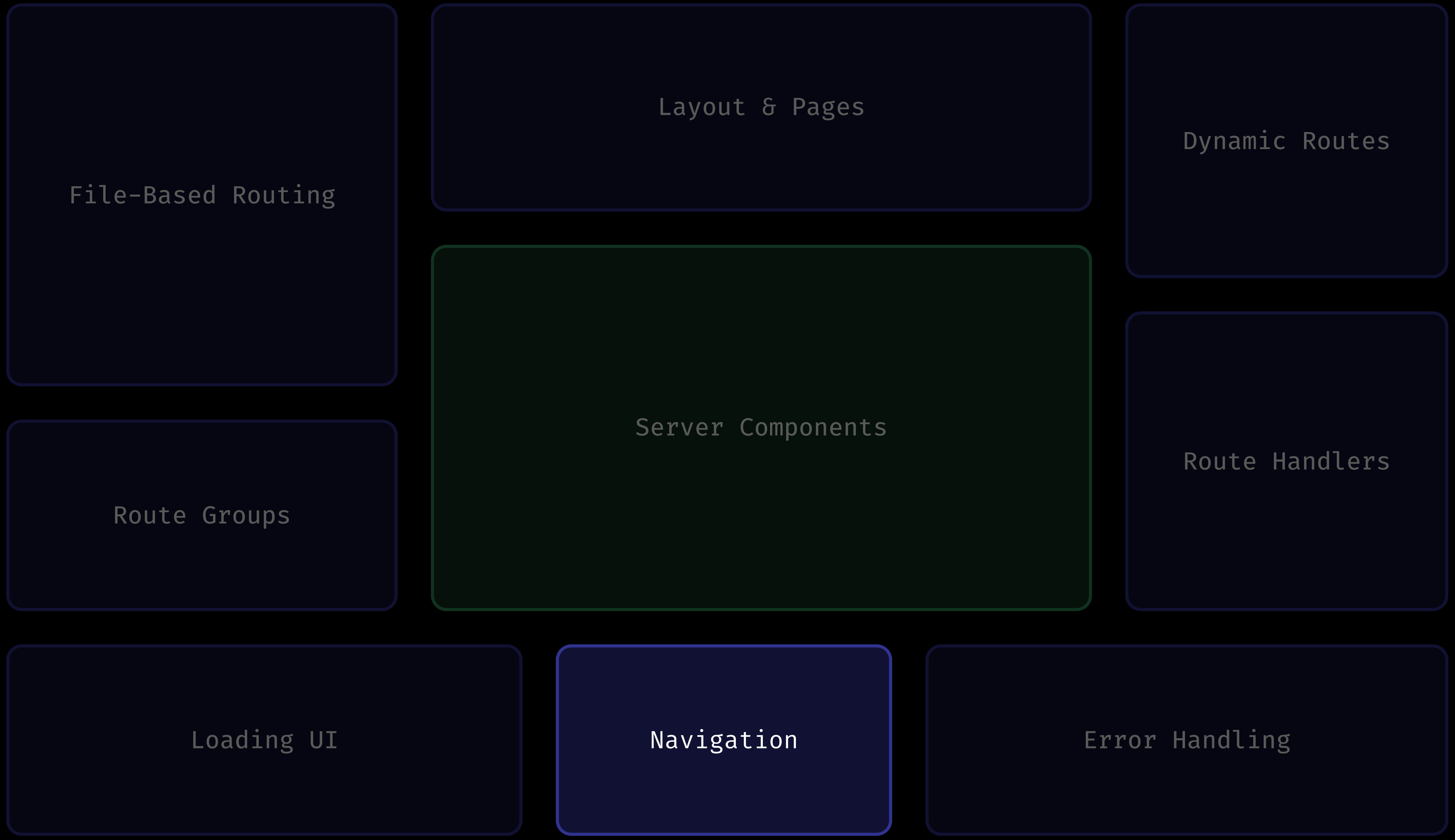
01

<Link> component

HTML anchor tag on steroids.

NUTS AND BOLTS

The App Router


File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

Route Handlers

Loading UI

Navigation

Error Handling

server client

01

<Link> component

HTML anchor tag on steroids.

client

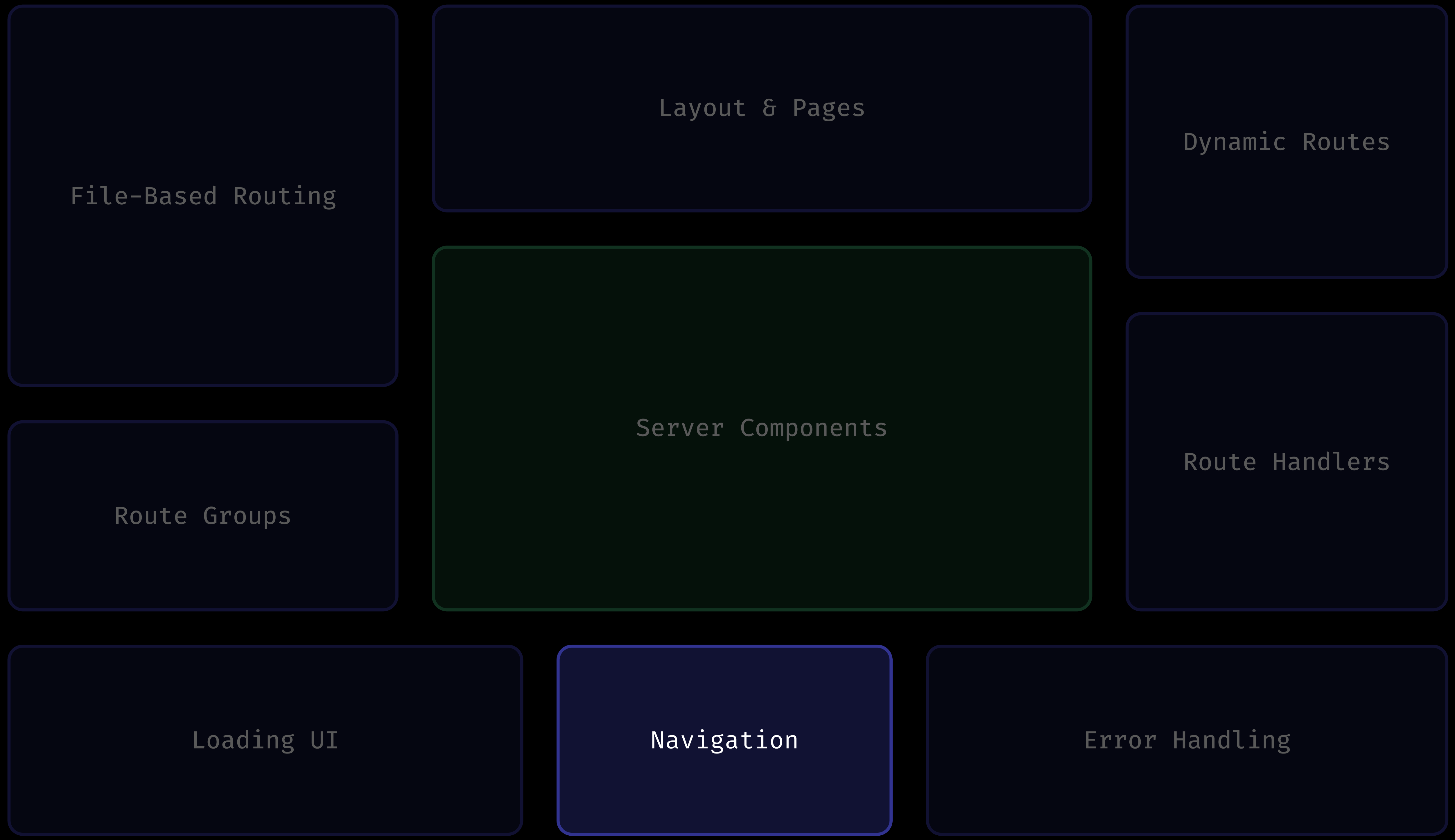
02

useRouter hook

Programmatic navigation.

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

Route Handlers

Loading UI

Navigation

Error Handling

server client

01

<Link> component

HTML anchor tag on steroids.

client

02

useRouter hook

Programmatic navigation.

server client

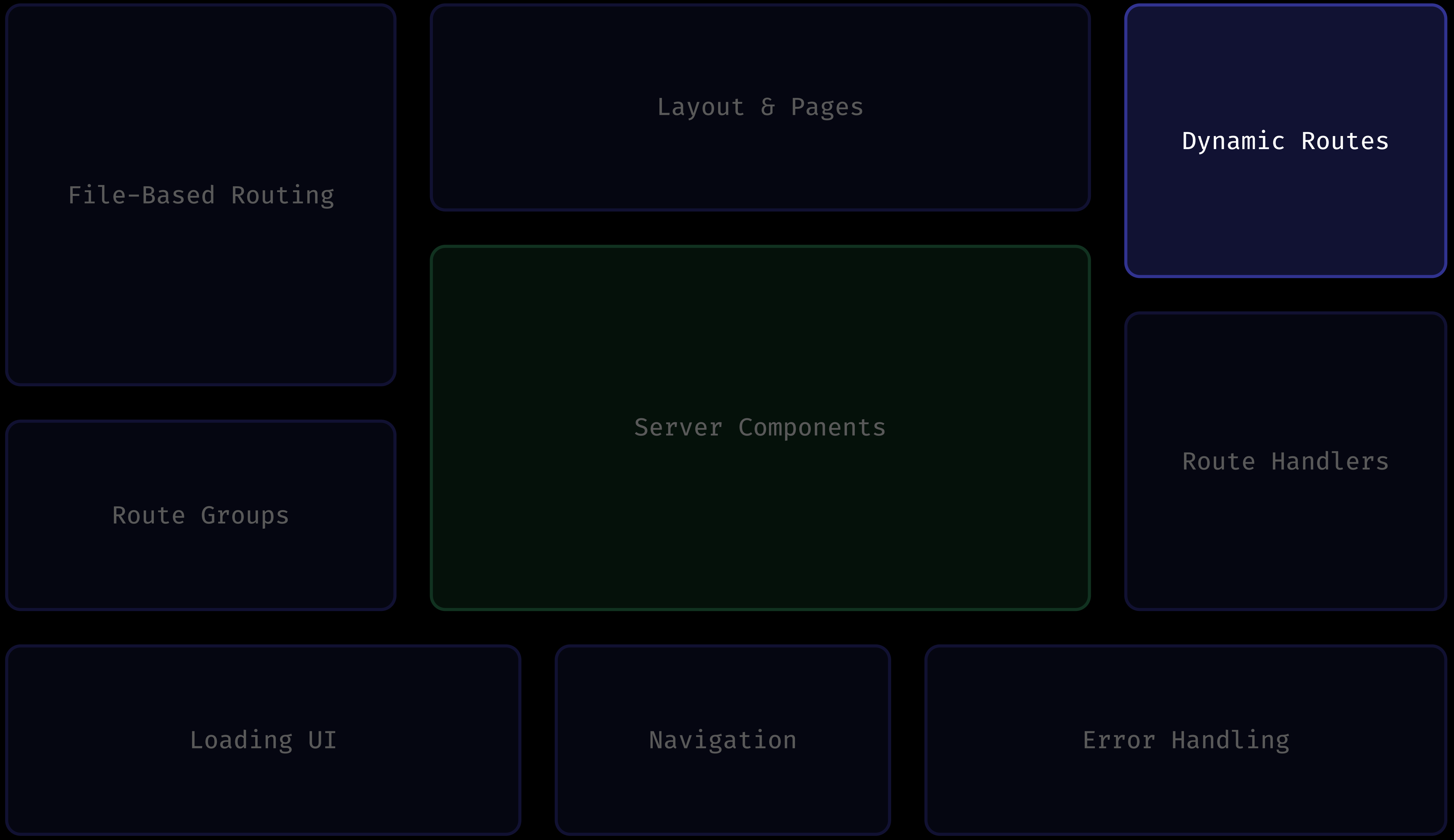
03

redirect() function

Redirect to another page.

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Handlers

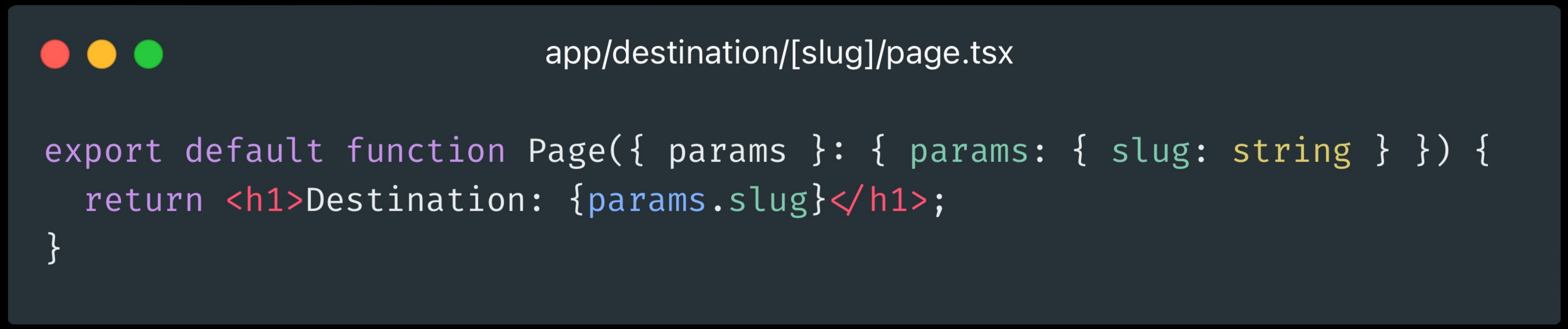
Route Groups

Loading UI

Navigation

Error Handling

Dynamic segment



app/destination/[slug]/page.tsx

```tsx
export default function Page({ params }: { params: { slug: string } }) {
  return <h1>Destination: {params.slug}</h1>;
}
```

✅ /destination/calabria

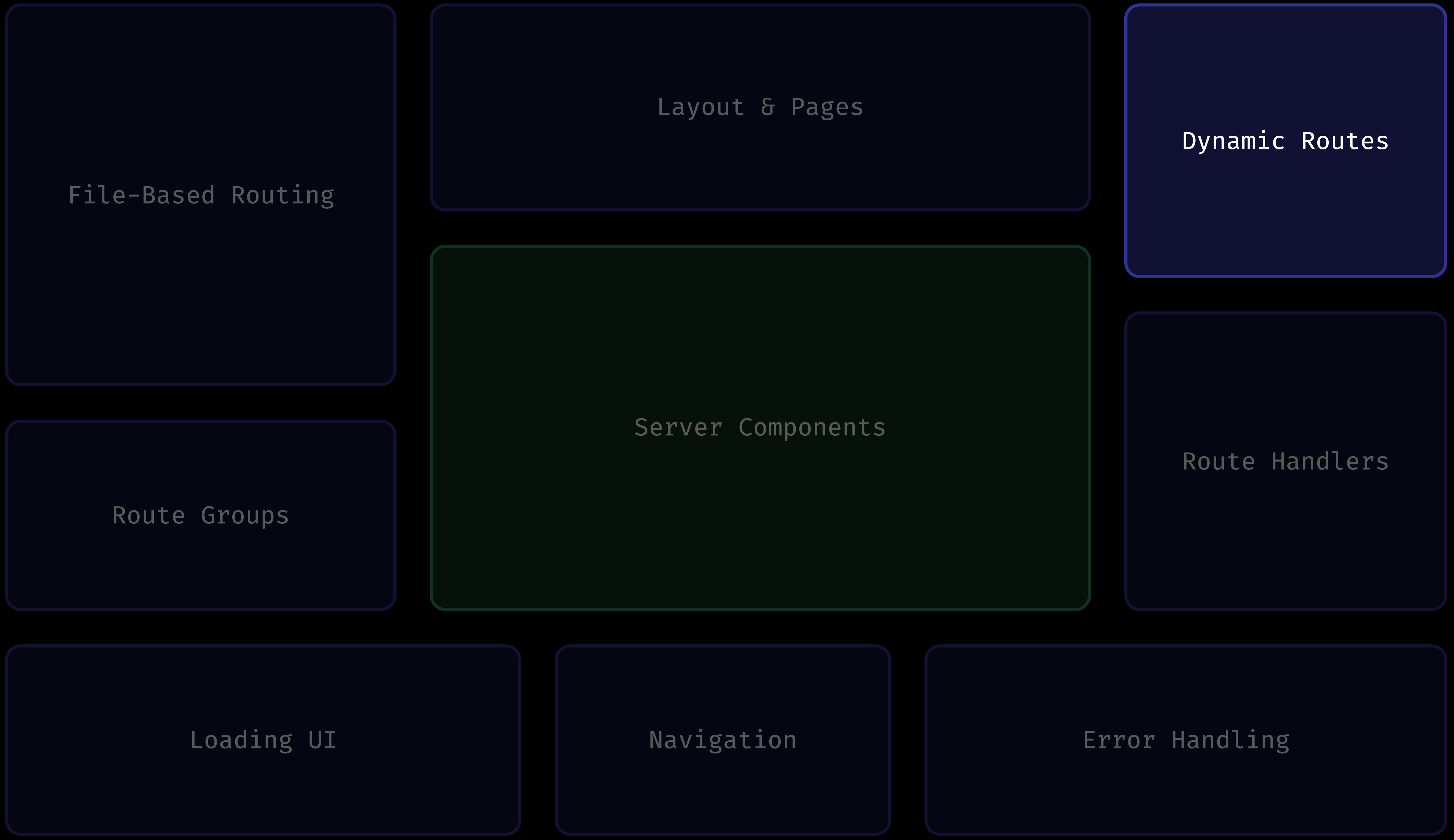
✅ /destination/unknown

❌ /destination/uk/london

❌ /destination

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

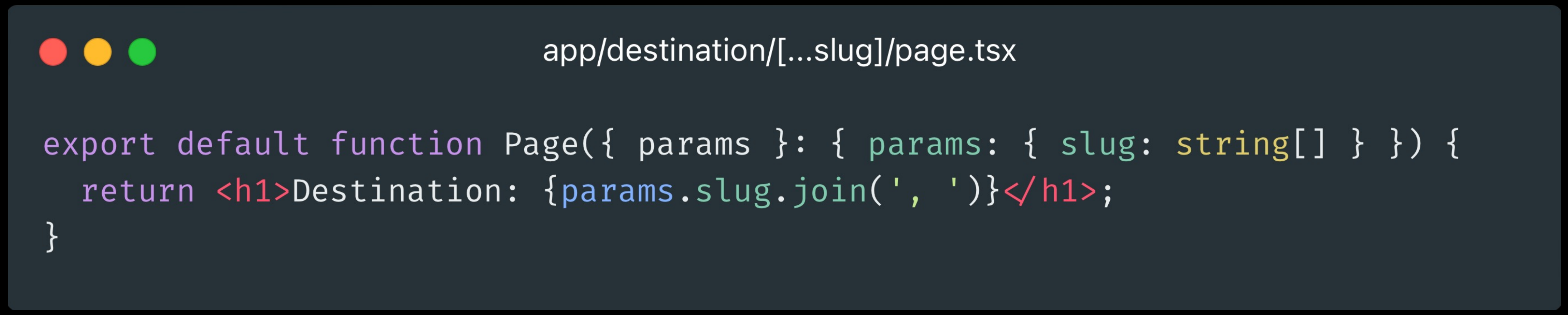
Route Handlers

Loading UI

Navigation

Error Handling

Catch-all segment



app/destination/[...slug]/page.tsx

```
export default function Page({ params }: { params: { slug: string[] } }) {
  return <h1>Destination: {params.slug.join(', ')}</h1>;
}
```

✅ /destination/calabria

✅ /destination/unknown

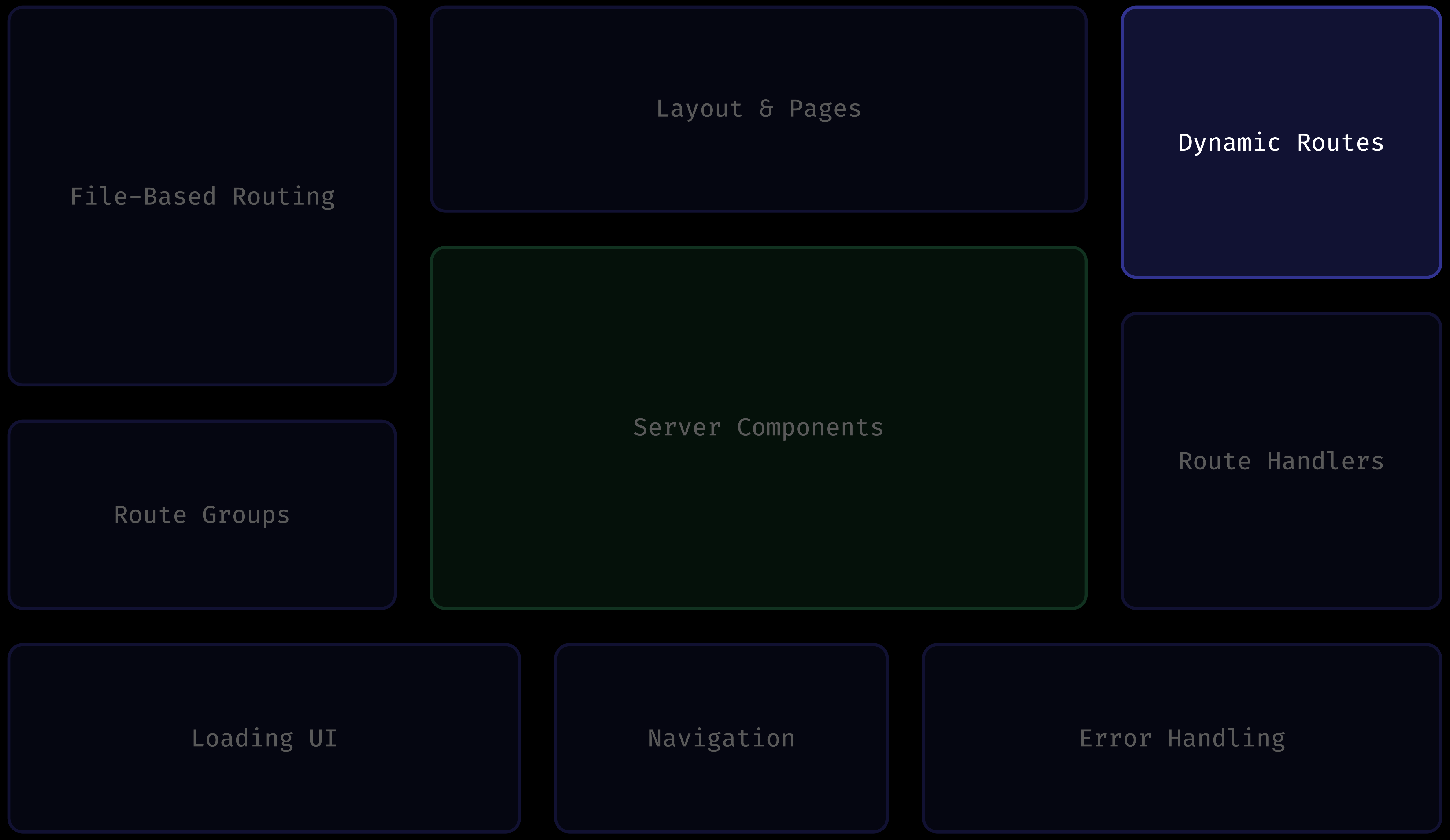
✅ /destination/uk/london

✅ /destination/usa/california/los-angeles

❌ /destination

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

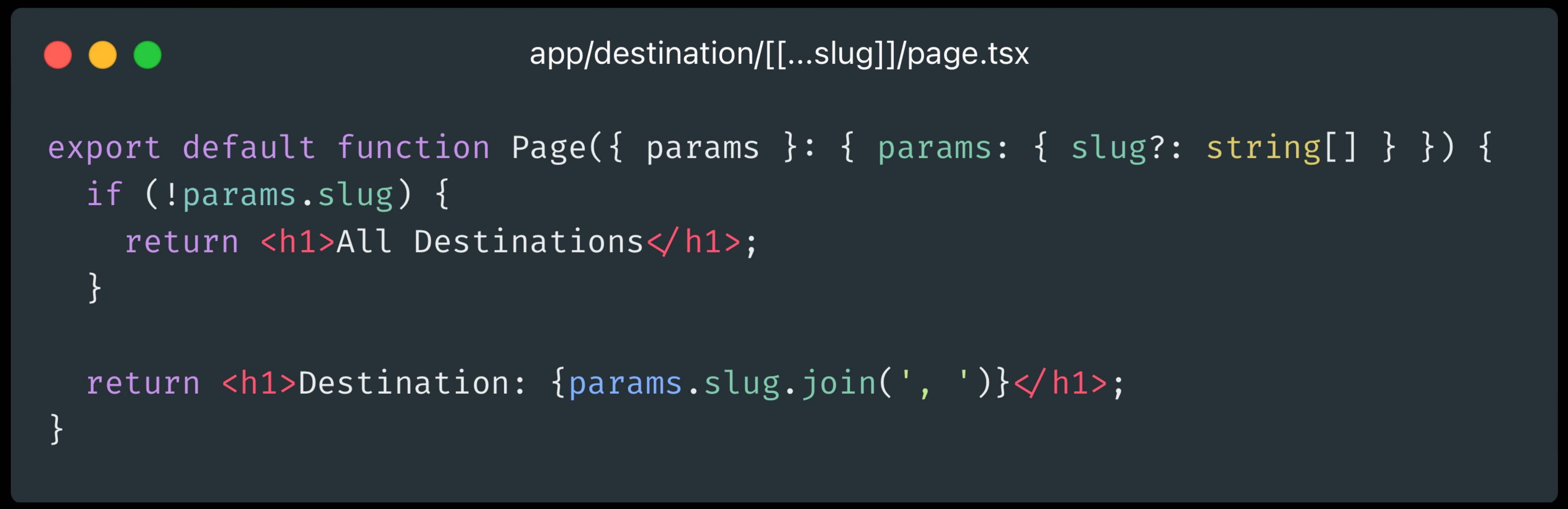
Route Handlers

Loading UI

Navigation

Error Handling

Optional catch-all segment



app/destination/[[...slug]]/page.tsx

```
export default function Page({ params }: { params: { slug?: string[] } }) {
  if (!params.slug) {
    return <h1>All Destinations</h1>;
  }

  return <h1>Destination: {params.slug.join(', ')}</h1>;
}
```

✅ /destination/calabria

✅ /destination/unknown

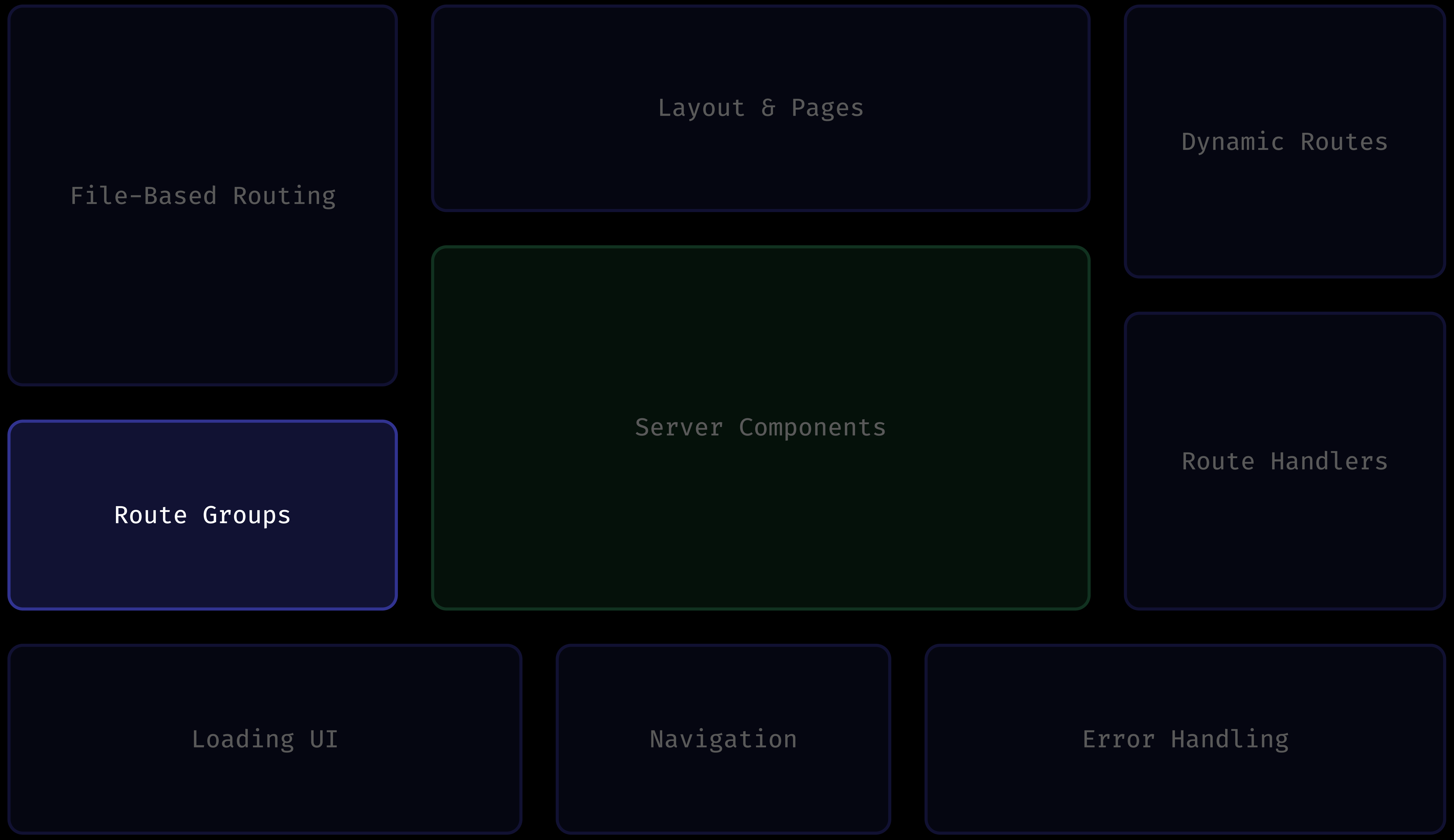
✅ /destination/uk/london

✅ /destination/usa/california/los-angeles

✅ /destination

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

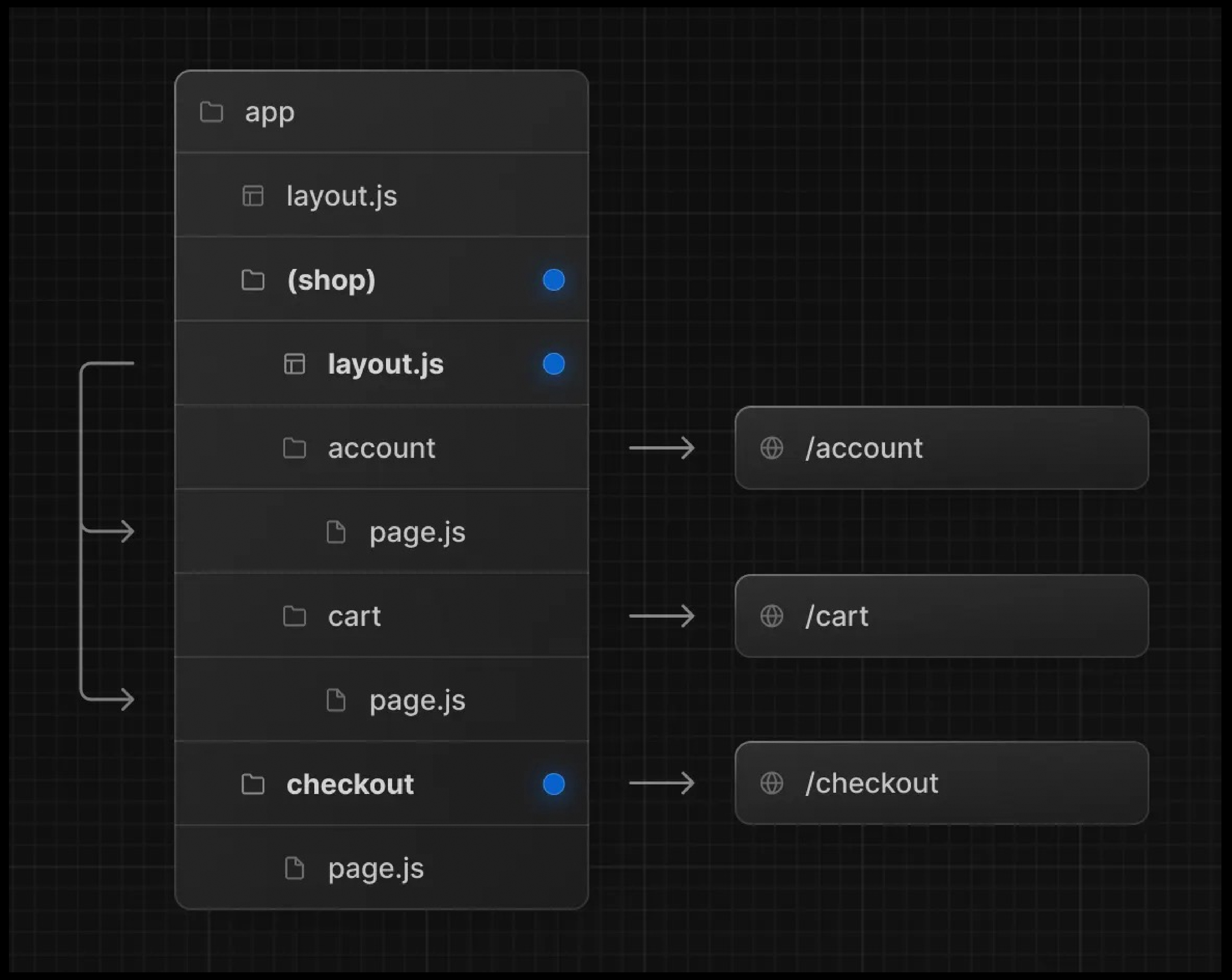
Route Handlers

Loading UI

Navigation

Error Handling

No side effects on URL segments structure



app

layout.js

(shop)

layout.js

account

page.js

cart

page.js

checkout

page.js

/account

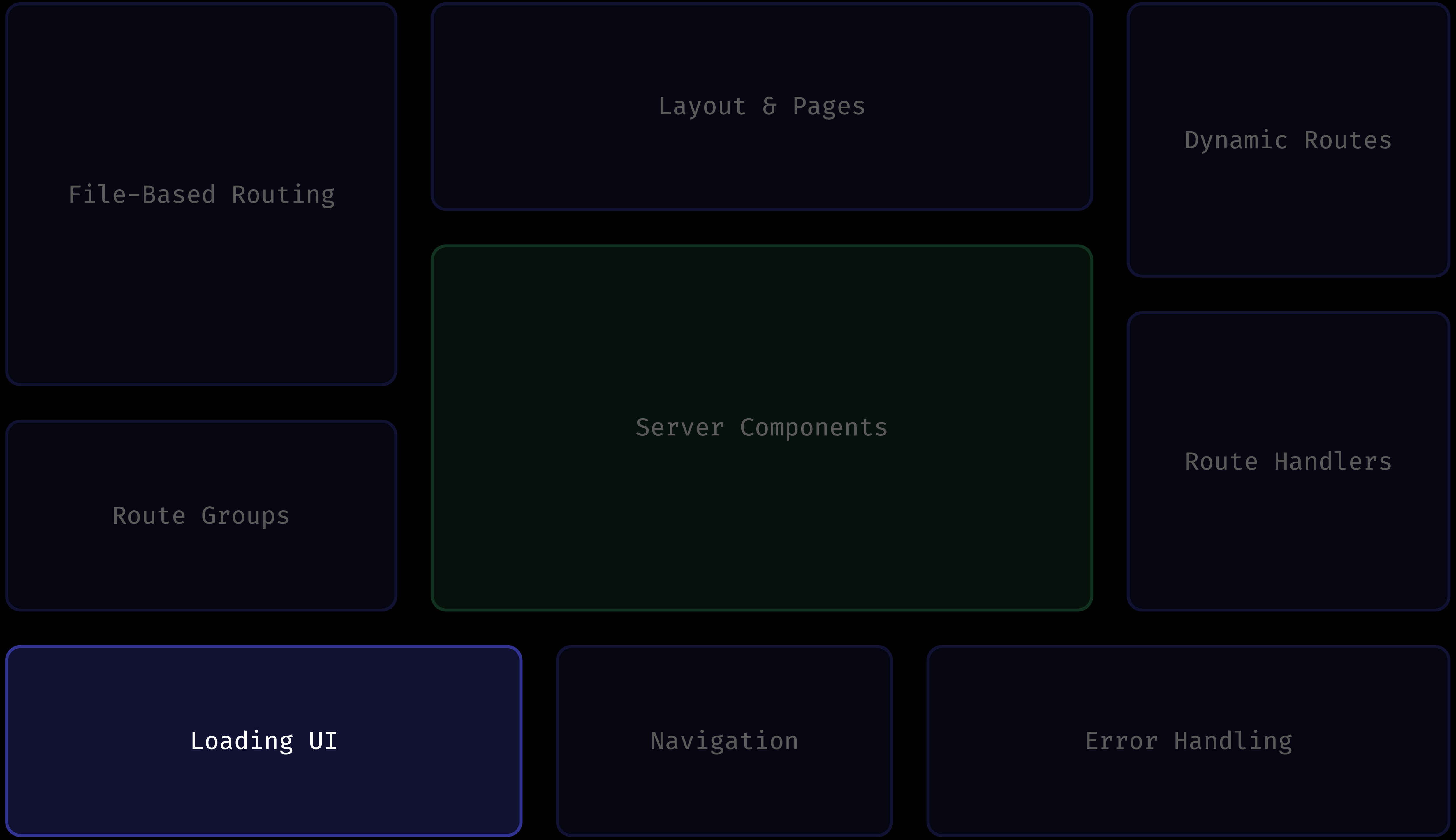
/cart

/checkout

Image © 2024 Vercel, Inc.

NUTS AND BOLTS

The App Router



Loading state for the whole page

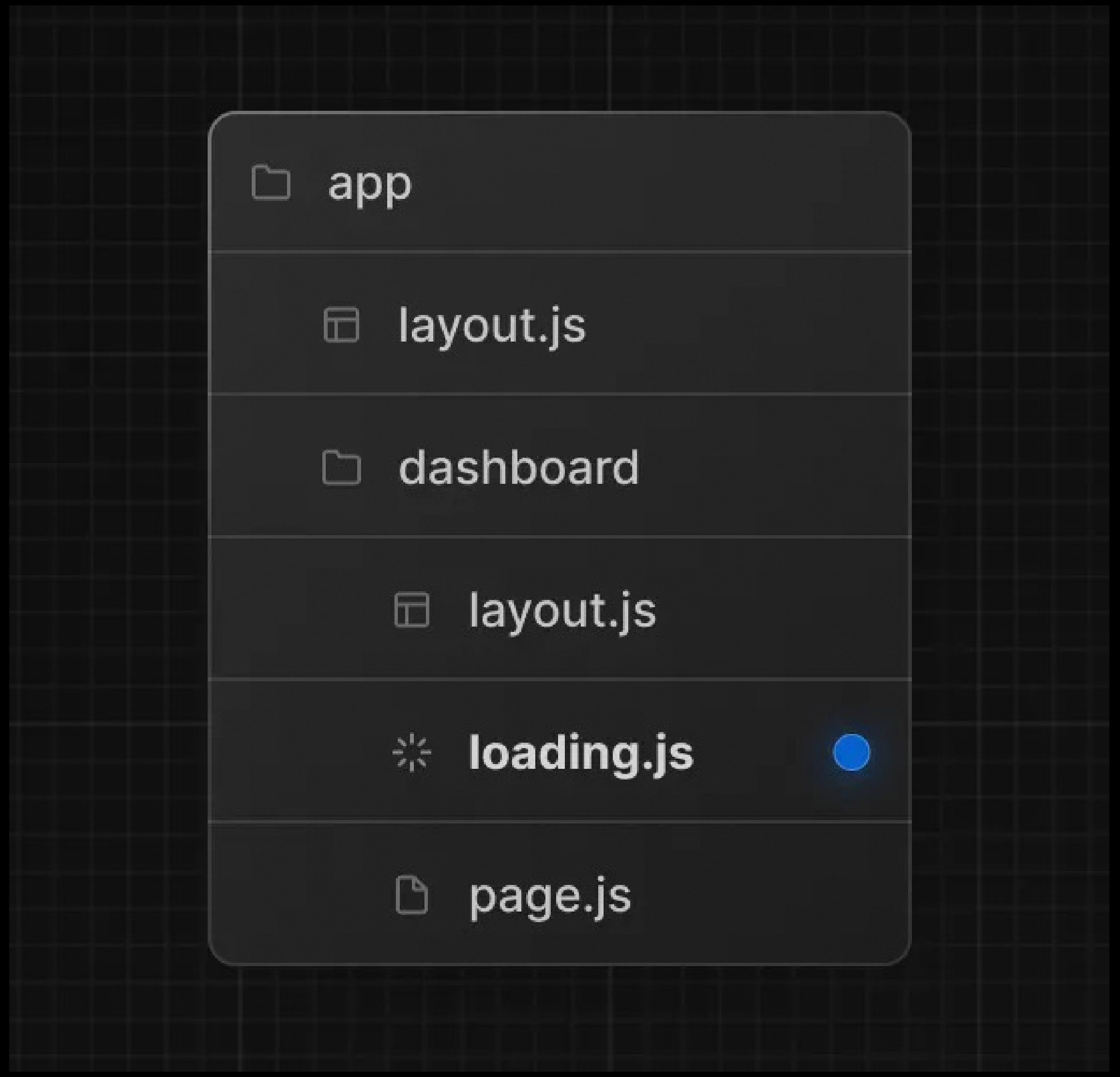


Image © 2024 Vercel, Inc.

NUTS AND BOLTS

The App Router

File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

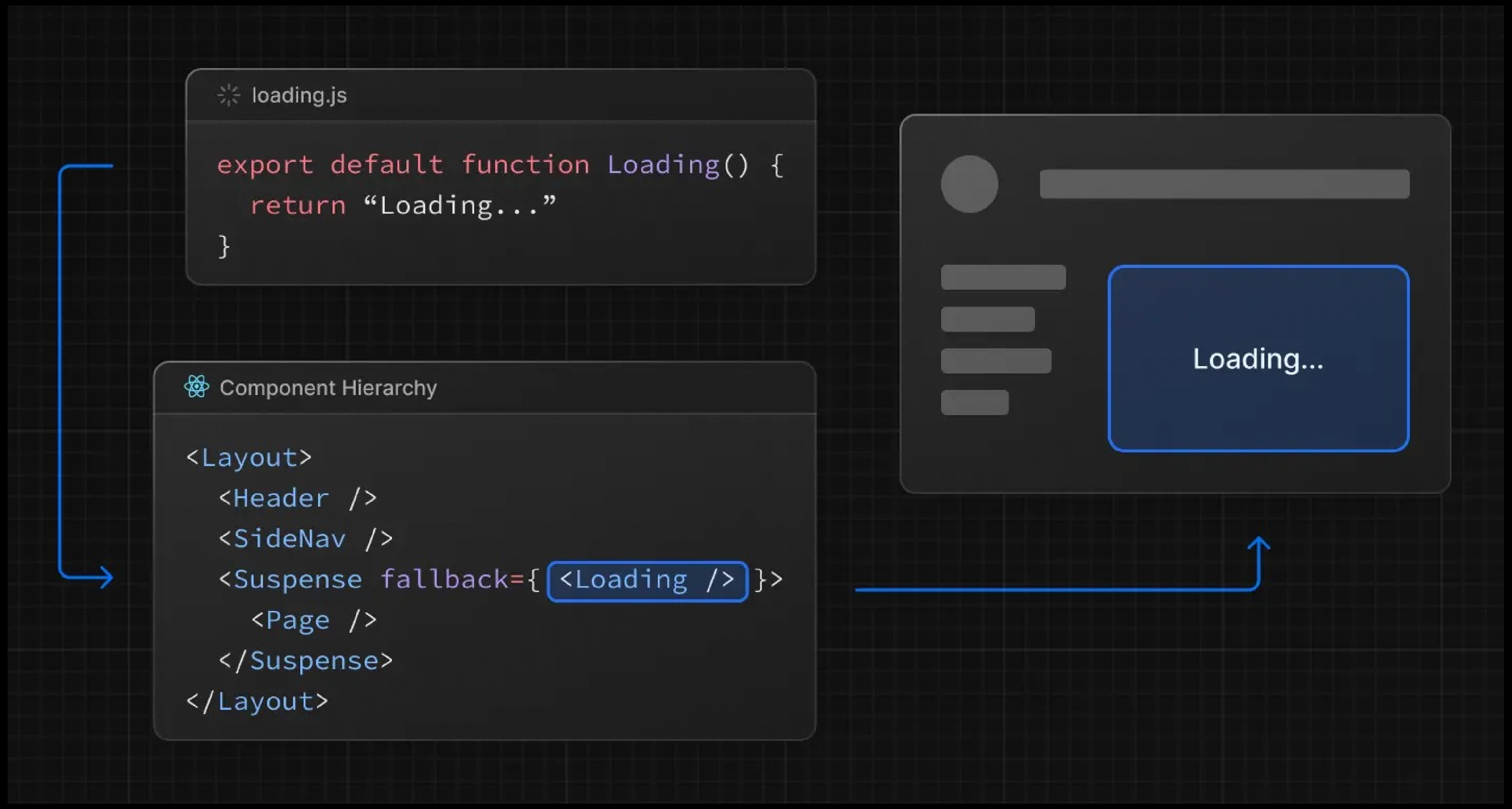
Route Handlers

Loading UI

Navigation

Error Handling

Loading state on only a part of the page? Just wrap in Suspense.



loading.js

```
export default function Loading() {
  return “Loading...”
}
```

Component Hierarchy

```
<Layout>
  <Header />
  <SideNav />
  <Suspense fallback={<Loading />}>
    <Page />
  </Suspense>
</Layout>
```

Image © 2024 Vercel, Inc.

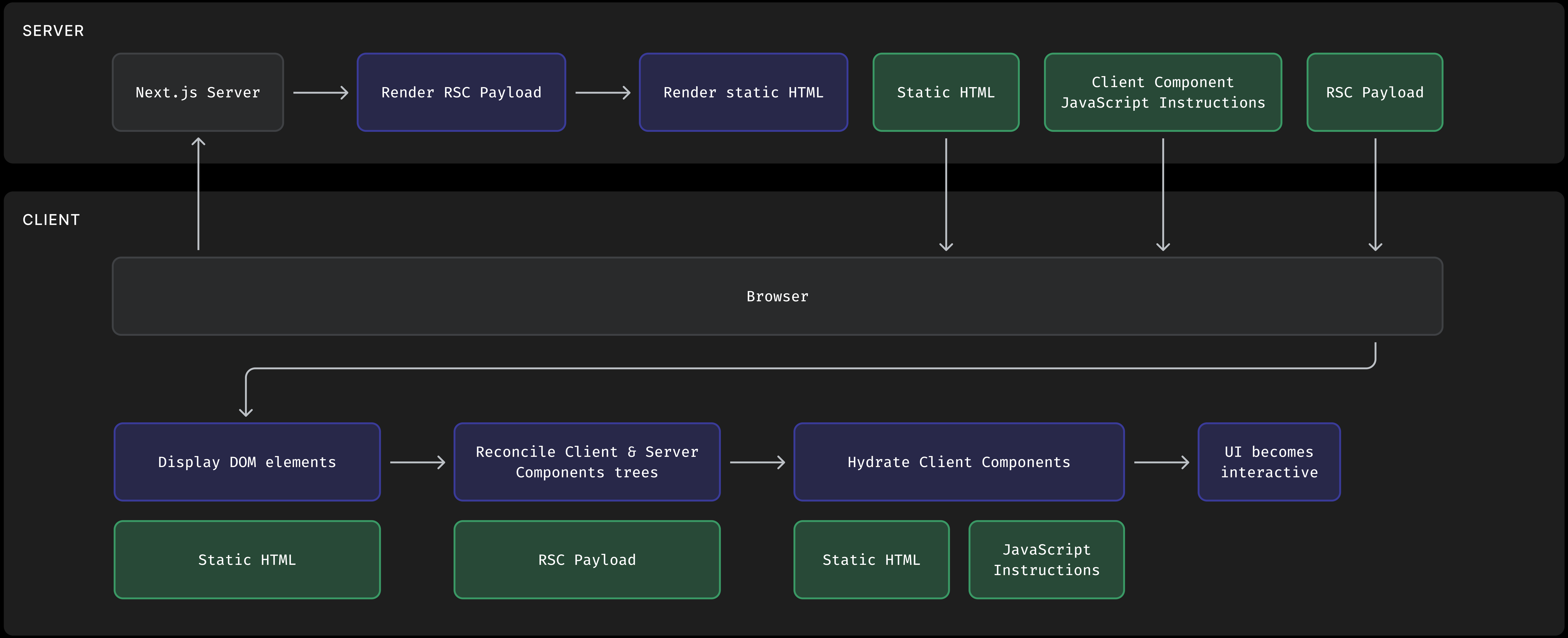
BEFORE WE CONTINUE...

Let’s talk about Streaming

STREAMING

How does streaming work?

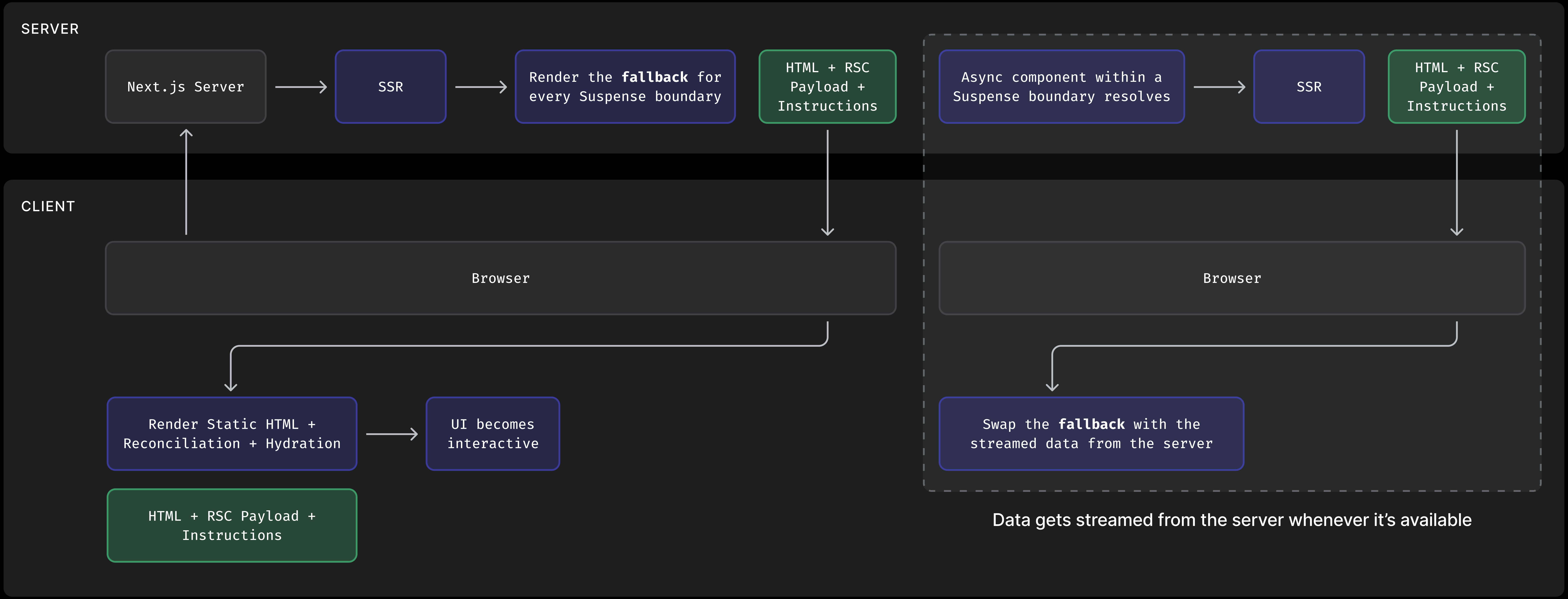
Revisiting SSR and Hydration



STREAMING

How does streaming work?

Streaming

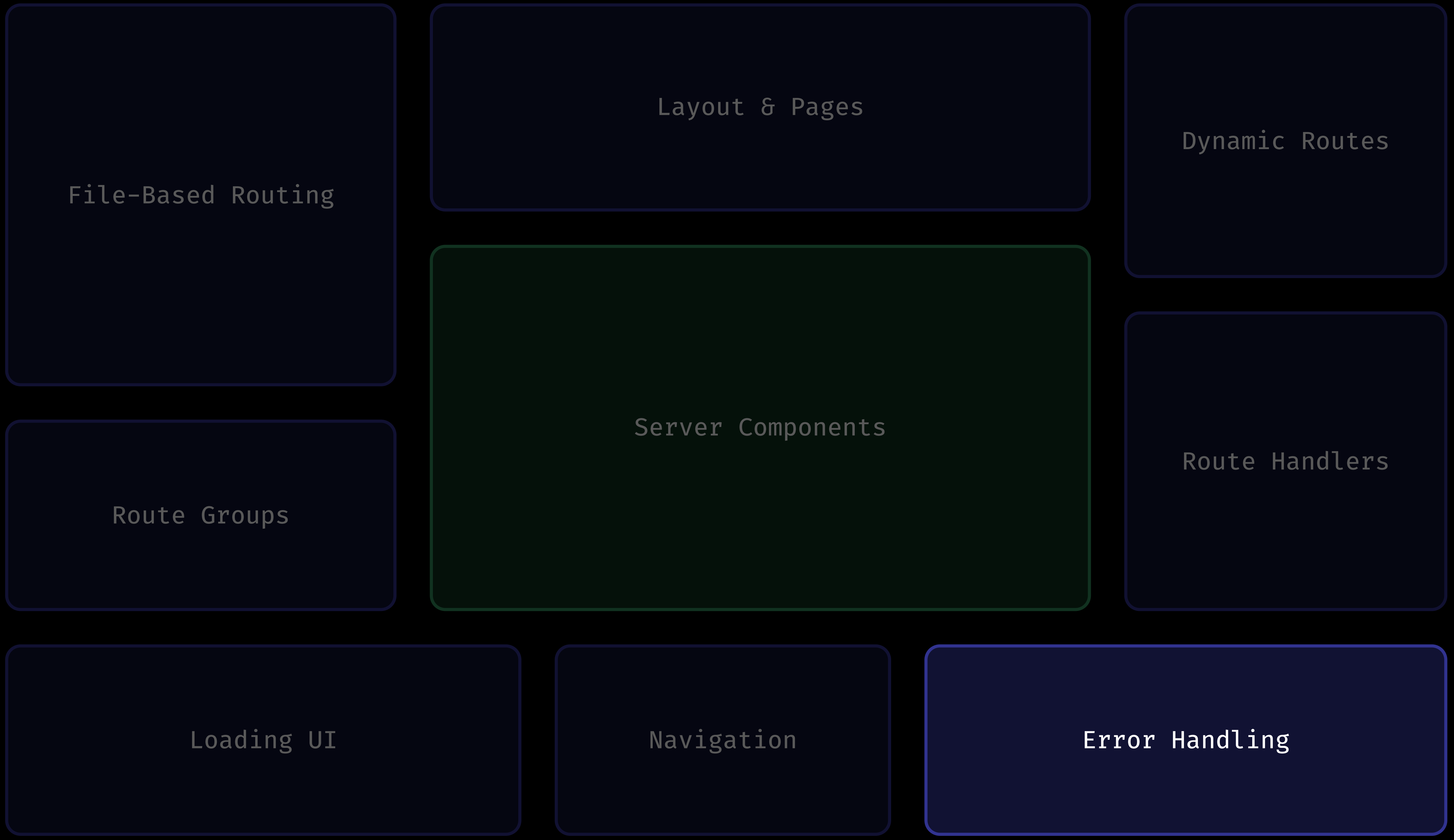

SERVER
Next.js Server
SSR
Render the **fallback** for every Suspense boundary
HTML + RSC Payload + Instructions
Async component within a Suspense boundary resolves
SSR
HTML + RSC Payload + Instructions
CLIENT
Browser
Browser
Render Static HTML + Reconciliation + Hydration
UI becomes interactive
Swap the **fallback** with the streamed data from the server
HTML + RSC Payload + Instructions
Data gets streamed from the server whenever it's available

NOW THAT WE GOT STREAMING OUT OF THE WAY...

Let’s go back

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

Route Handlers

Loading UI

Navigation

Error Handling

Error boundary for the whole page

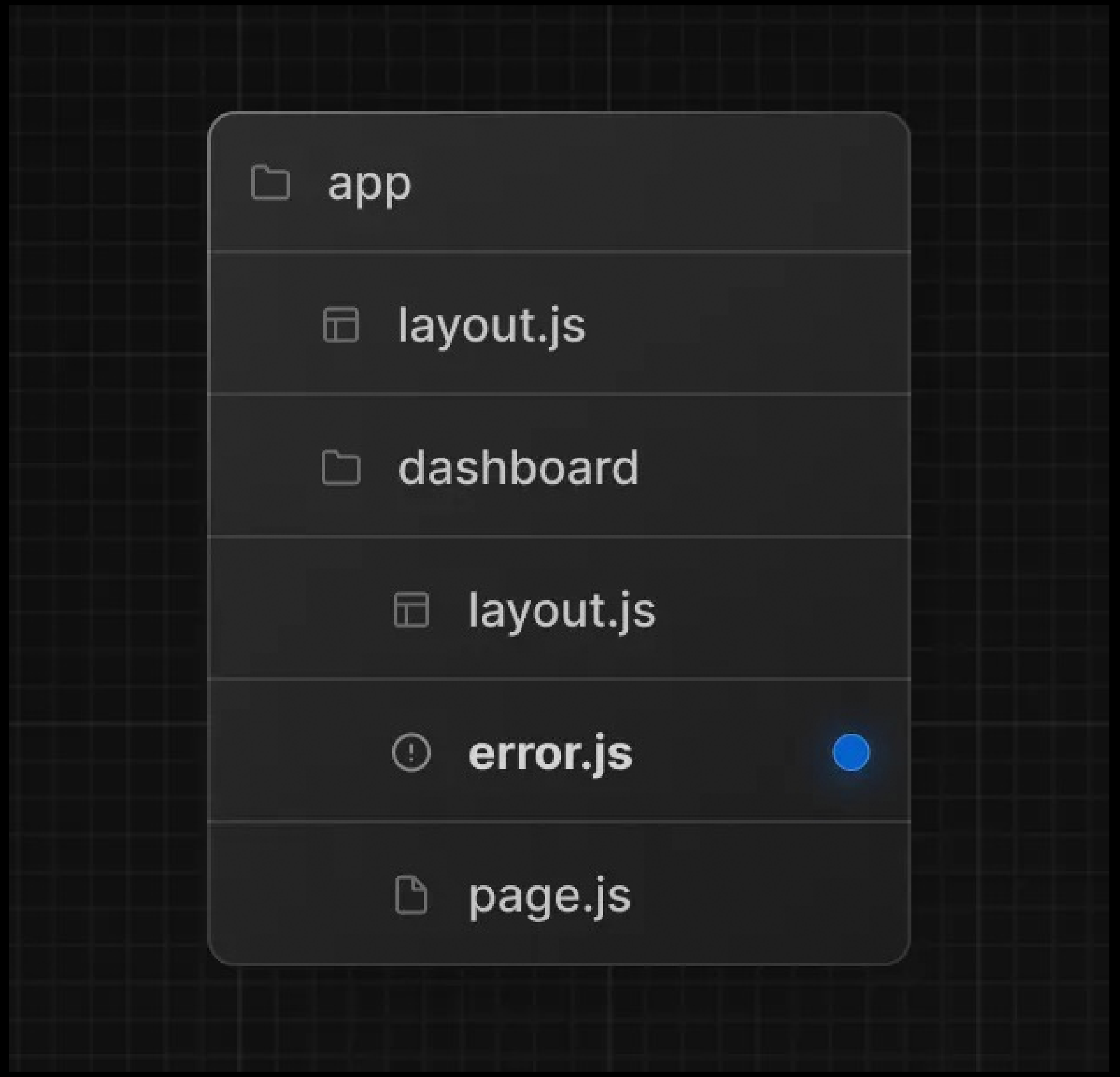
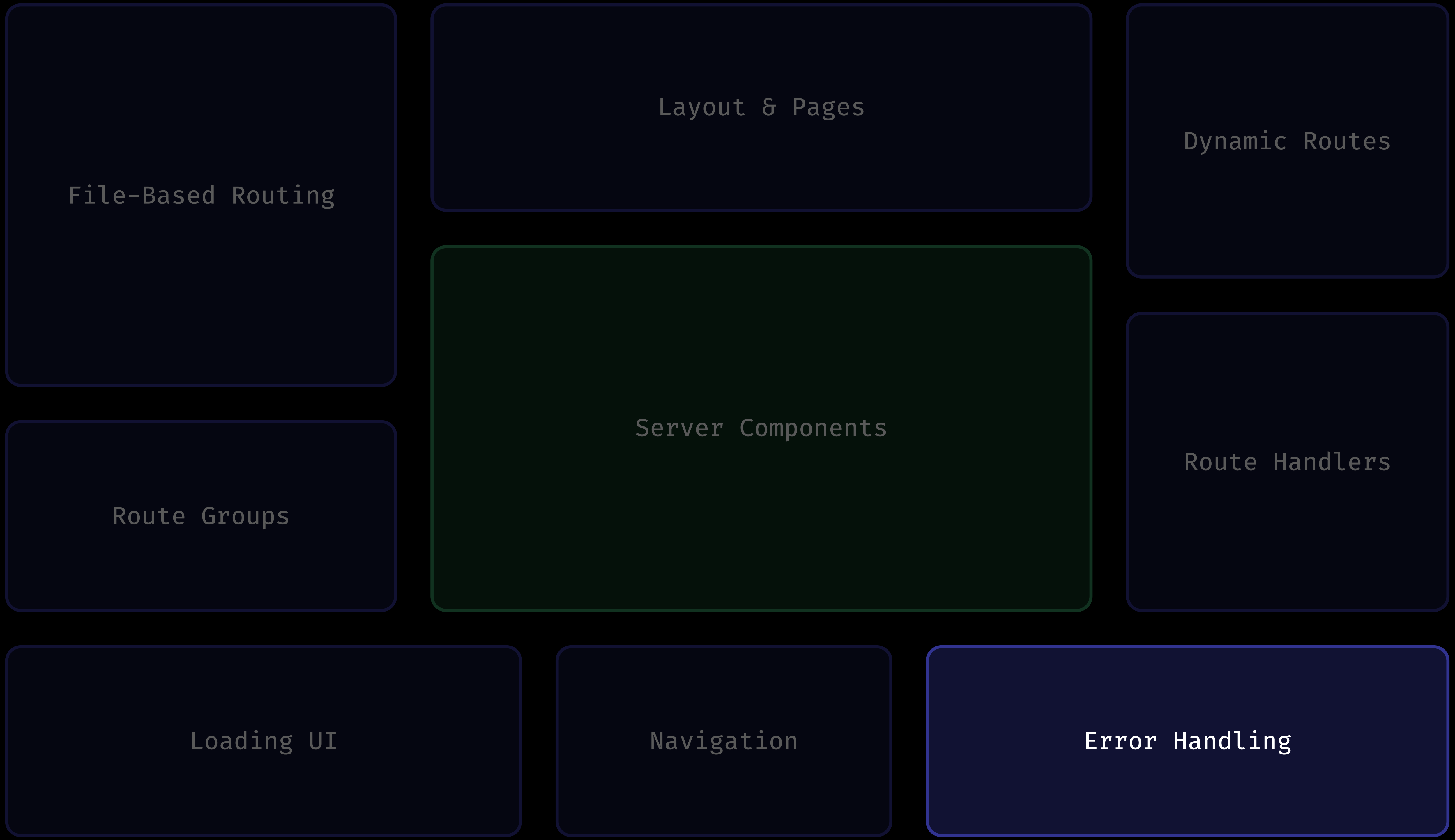


Image © 2024 Vercel, Inc.

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

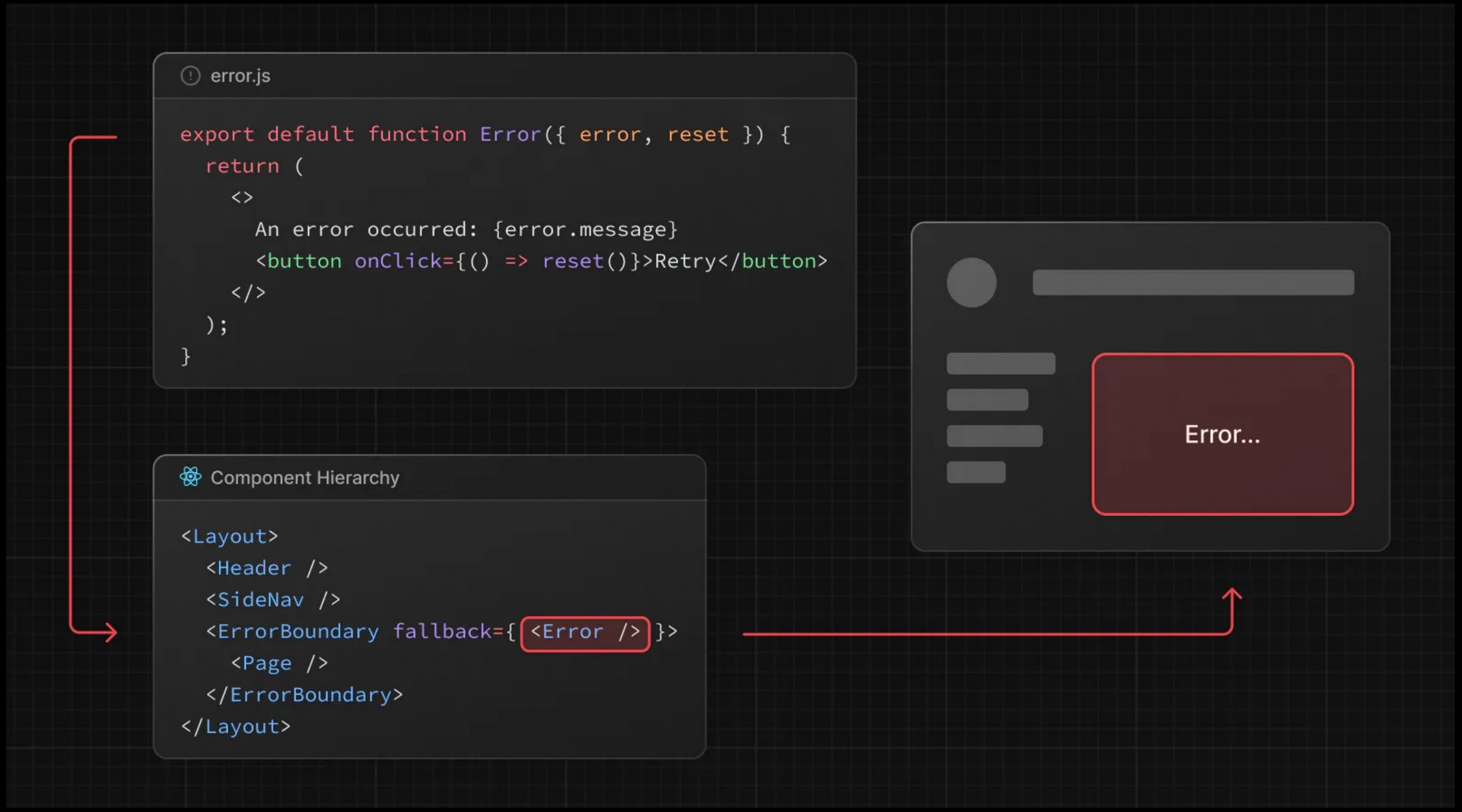
Route Handlers

Loading UI

Navigation

Error Handling

Handle errors only in a specific part of the page? Wrap in ErrorBoundary.



error.js

```
export default function Error({ error, reset }) {
  return (
    <>
      An error occurred: {error.message}
      <button onClick={() => reset()}>Retry</button>
    </>
  );
}
```

Component Hierarchy

```
<Layout>
  <Header />
  <SideNav />
  <ErrorBoundary fallback={<Error />}>
    <Page />
  </ErrorBoundary>
</Layout>
```

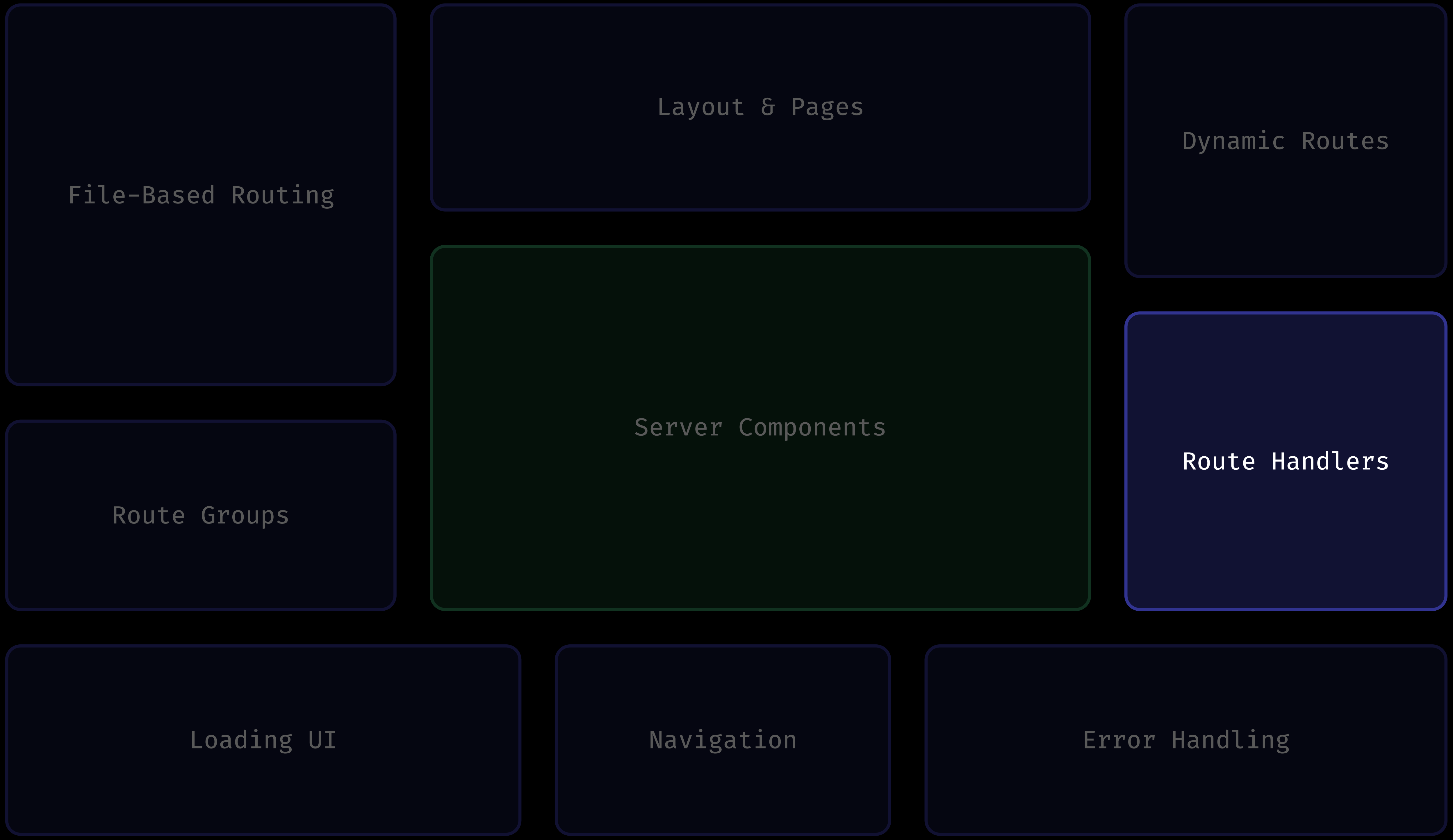
Image © 2024 Vercel, Inc.

*caveat: error boundaries in React are still implemented with class components 😵

solution: install and use **react-error-boundary** package 🎉

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Handlers

Route Groups

Loading UI

Navigation

Error Handling

Just like pages, route files define accessible API routes

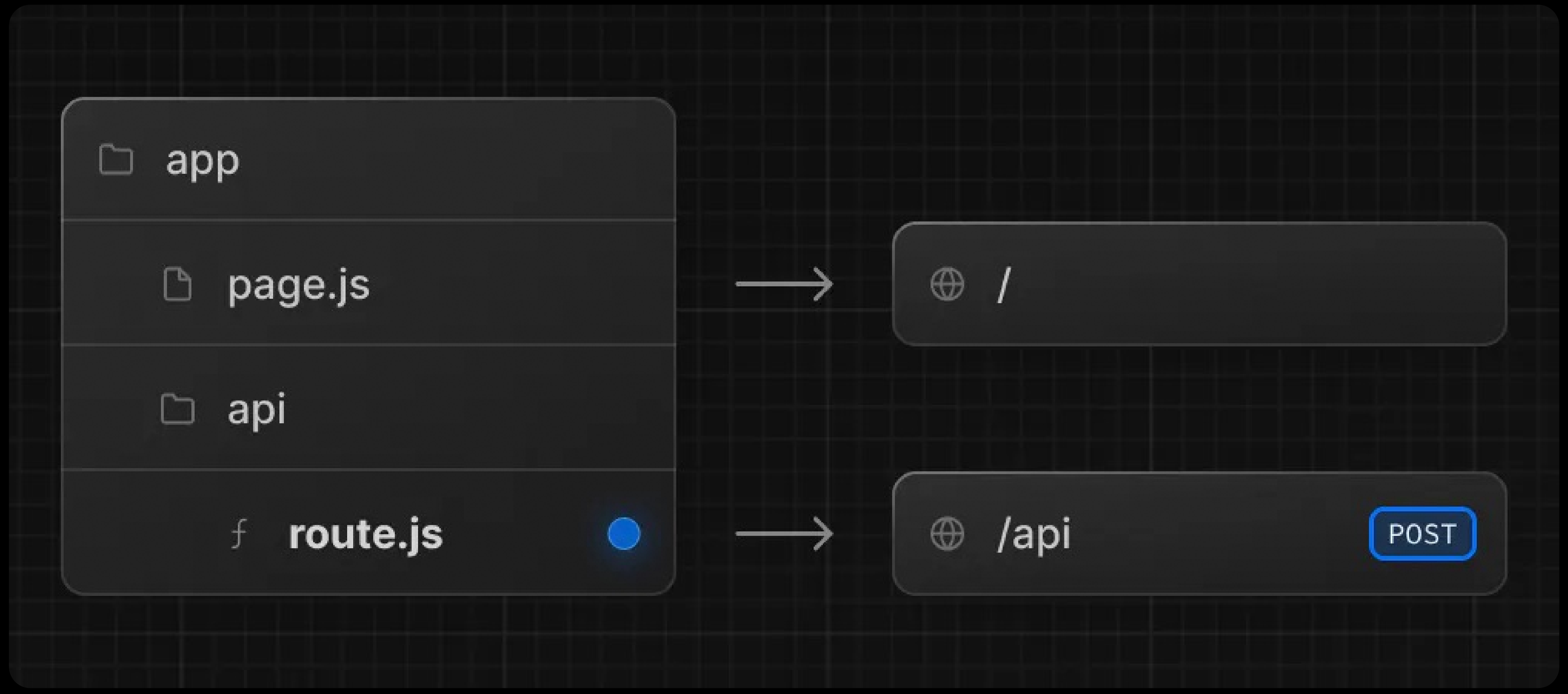
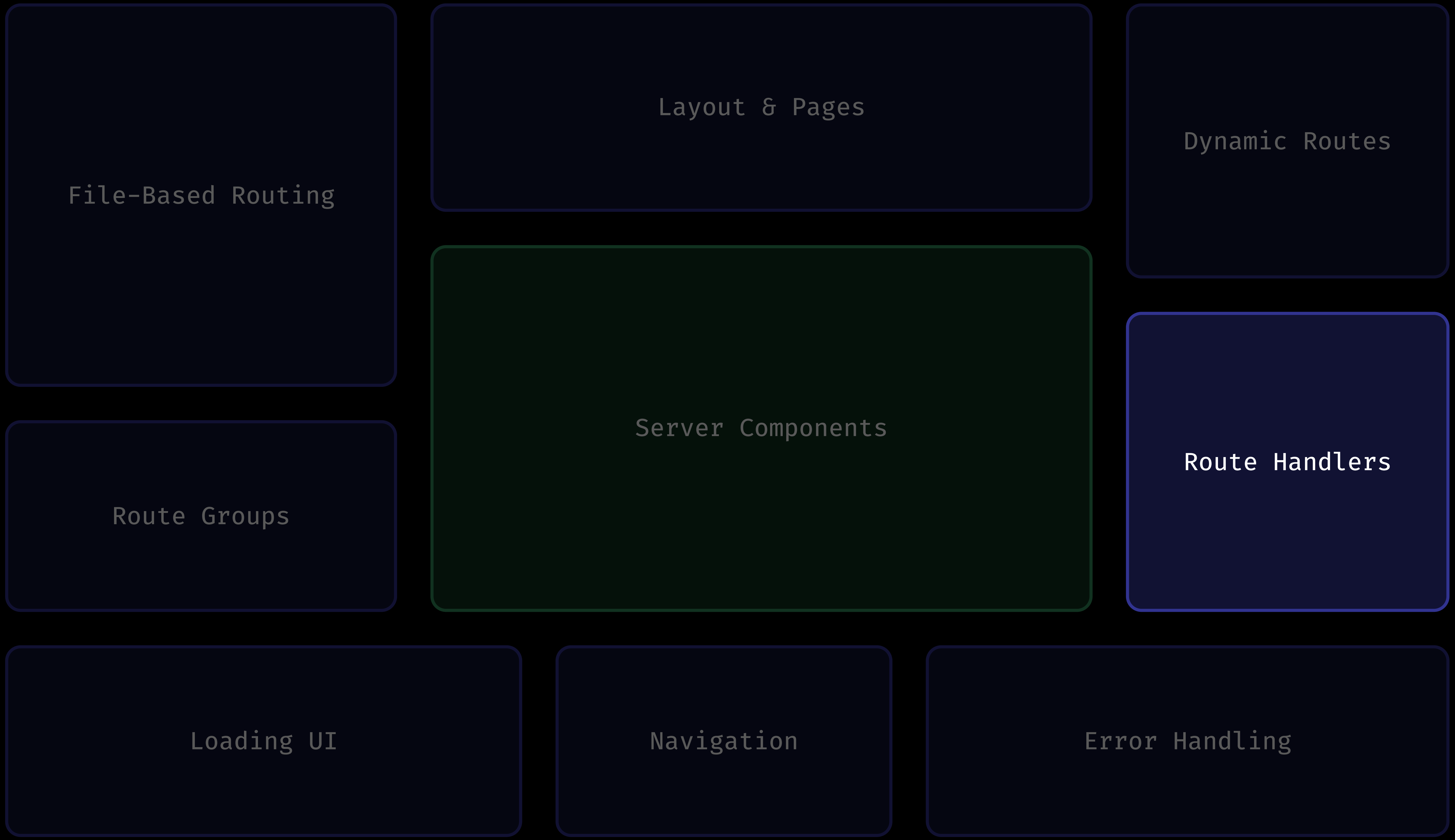


Image © 2024 Vercel, Inc.

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Handlers

Route Groups

Loading UI

Navigation

Error Handling

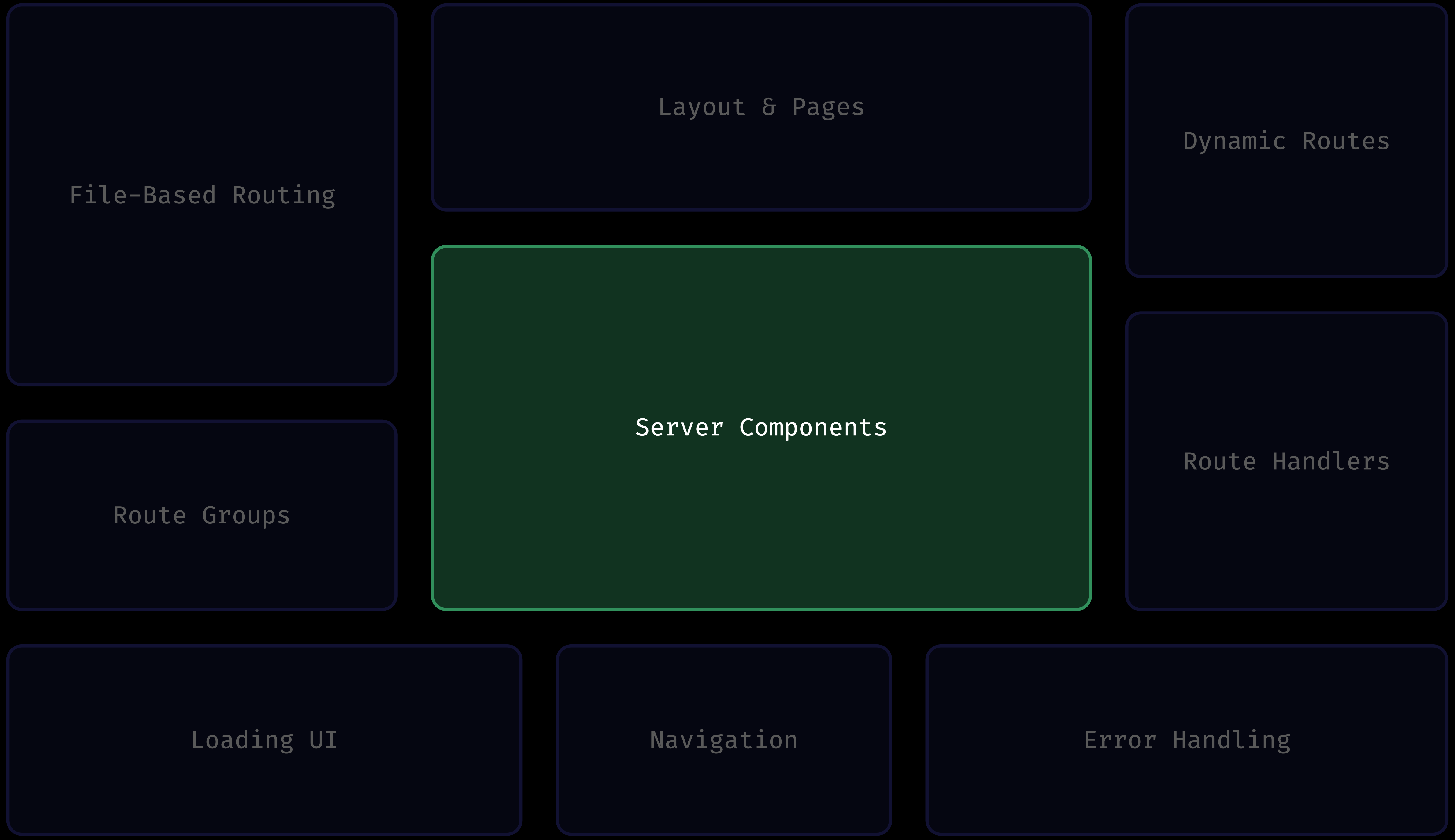
Available

HTTP Methods

GET
POST
PUT
PATCH
DELETE
HEAD
OPTIONS

NUTS AND BOLTS

The App Router



File-Based Routing

Layout & Pages

Dynamic Routes

Server Components

Route Groups

Route Handlers

Loading UI

Navigation

Error Handling

FETCHING DATA

Different Ways Of Fetching Data

01
API Layer

Create API endpoints using Route Handlers.
Suitable for fetching data from the client,
e.g. when using react-query.

02
Server Components

Fetch data directly on the server.
Query the DB directly, or use a 3rd party API
without worrying about exposing tokens.

The recommended approach by the Next.js team

Let’s explore →

FETCHING DATA

Different Ways Of Fetching Data

01

API Layer

Create API endpoints using Route Handlers.
Suitable for fetching data from the client,
e.g. when using react-query.

?

02

Server Components

Fetch data directly on the server.
Query the DB directly, or use a 3rd party API
without worrying about exposing tokens.

FETCHING DATA

Different Ways Of Fetching Data

01

API Layer

Create API endpoints using
Suitable for fetching da
e.g. when using react-que

03

Hybrid Approach

Best of both worlds.
Fetch initially on the server, hydrate the client
and then refetch on demand on the client.

02

Server Components

Fetch data directly on the server.
tly, or use a 3rd party API
g about exposing tokens.

Let’s check it out →

MUTATING DATA

Server Actions

No need for API endpoints

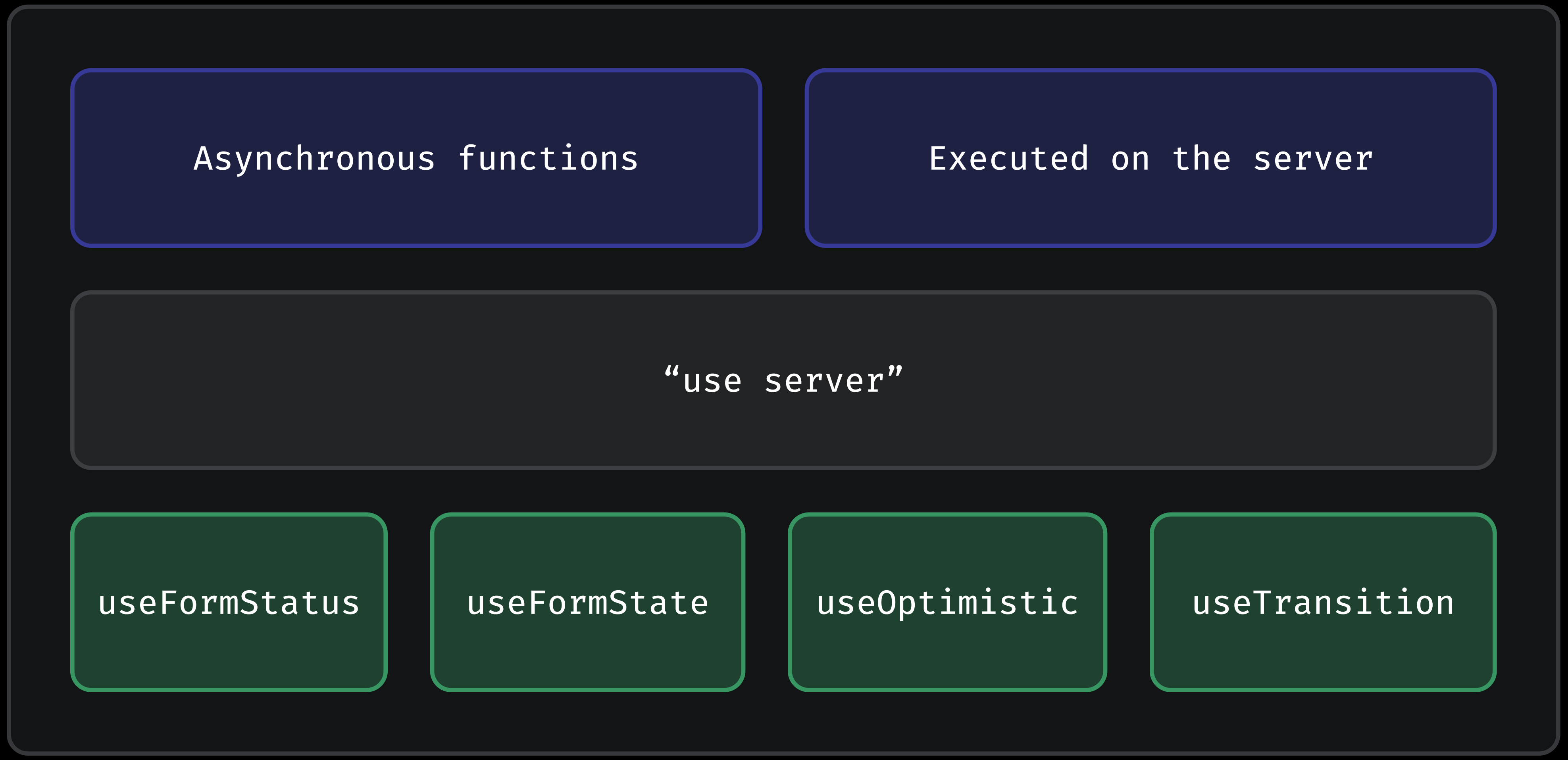
- Mutate data with just a single function
- Invoke from Client or Server components

Not limited to HTML forms

- Invoke from event handlers, e.g. onClick
- Invoke from useEffect

Revalidating cached data

- revalidatePath
- revalidateTag



Asynchronous functions

Executed on the server

“use server”

useFormStatus

useFormState

useOptimistic

useTransition

SERVER ACTIONS

no pun intended 🤫

Let’s see them in action

WHAT'S NEW

Next.js 15 Release Candidate

01
Support for React 19 RC

React Compiler, hydration error improvement

02
Caching is opt-in

Nothing is cached by default anymore

03
Partial Prerendering

Still experimental, but looks promising

04
next/after

Execute code after returning a response

HOW TIME FLIES...

What We Didn’t Cover

- Caching
- Font & Image Optimization
- Metadata
- Configuration
- Testing
- Authentication
- Deployment
- Middleware
- Instrumentation Hook
- Cookies & Headers
- Parallel Routes
- Intercepting Routes
- Internationalization
- Accessibility
- Client-Side Data Fetching With react-query
- Edge Runtime
- Turbopack
- Turborepo
- Vercel
- Vercel Storage
- Vercel Toolbar
- Feature Flagging
- CI/CD & Preview Deployments

The documentation is amazing:

https://nextjs.org/docs

https://vercel.com/docs

https://vercel.com/guides

Let’s Connect

DROP ME A MESSAGE

Tsetsi Lazarov

linkedin.com/in/thexpand

info@qualisage.com

https://www.linkedin.com/in/thexpand/

https://github.com/thexpand/heroes-of-js-world-2024

QualiSage

Tsetsi Lazarov, CEO of QualiSage

TIME TO GET CURIOUS

Q&A Session

> The art and science of asking questions is the source of all knowledge.
>
> *Thomas Berger, American novelist*

THANK YOU

Until NEXT Time