Building a Portfolio Website with Next.js

Sage Abdullah

About Me

- 🌏 Open source enthusiast
- 🧑‍💻 Software Engineer at GudangAda
- 🌞 Google Summer of Code 2019 with Django
- 🌱 Developer Student Clubs – Universitas Indonesia 2019 Lead
- 🧑‍🎓 CSUI 2017

More at laymonage.com/about

Background

The internet is a vast space, full of people.

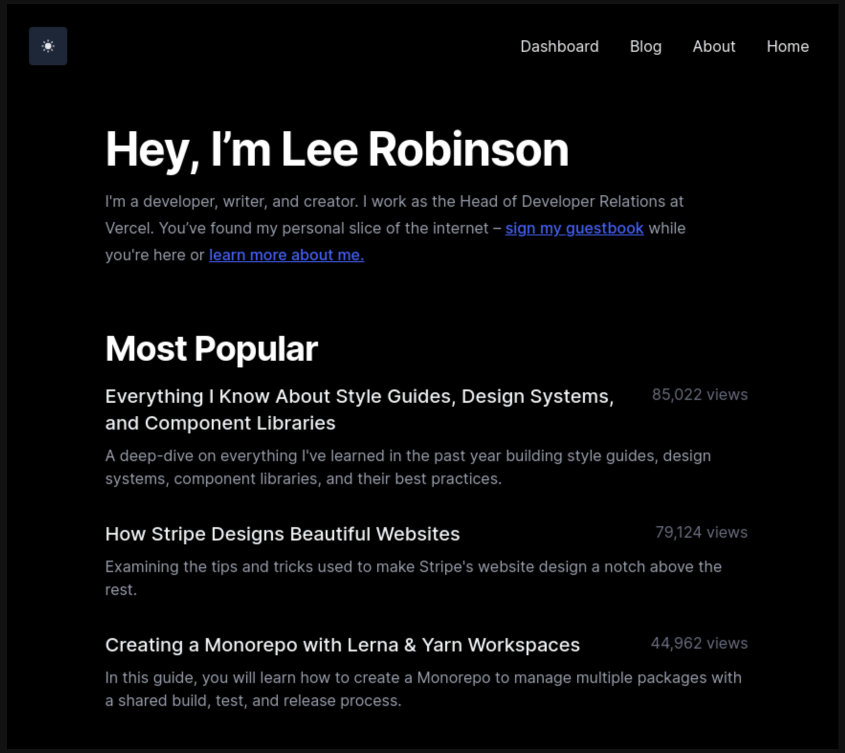
How can others find **you**?

How can others find **your work**?

Portfolio

Your **home** on the internet.

Where people can find **you** and **your work**.



Hey, I'm Lee Robinson

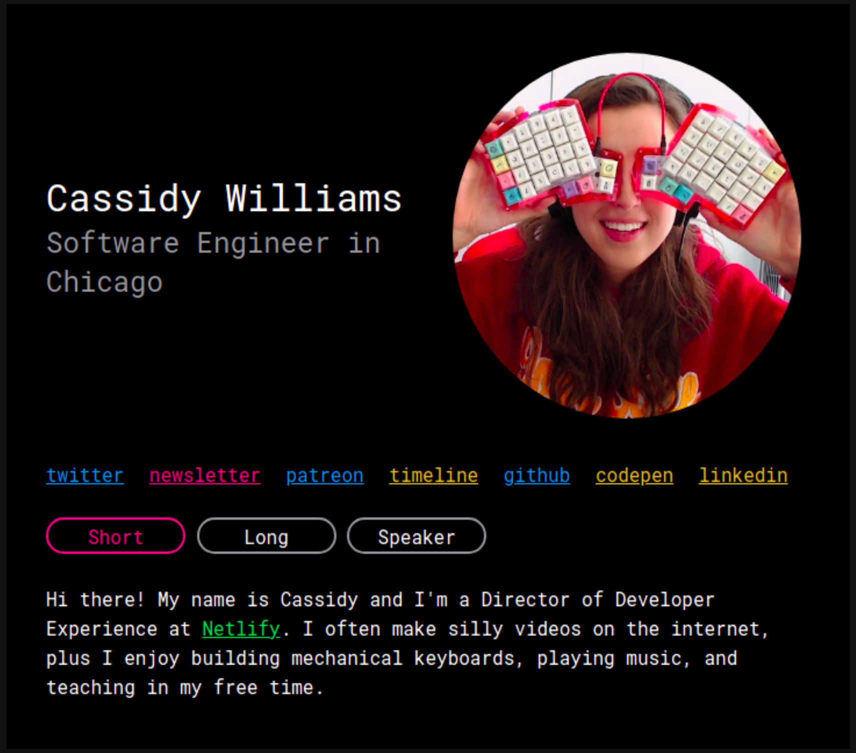
I'm a developer, writer, and creator. I work as the Head of Developer Relations at Vercel. You've found my personal slice of the internet – sign my guestbook while you're here or learn more about me.

Most Popular

Everything I Know About Style Guides, Design Systems, and Component Libraries 85,022 views
A deep-dive on everything I've learned in the past year building style guides, design systems, component libraries, and their best practices.

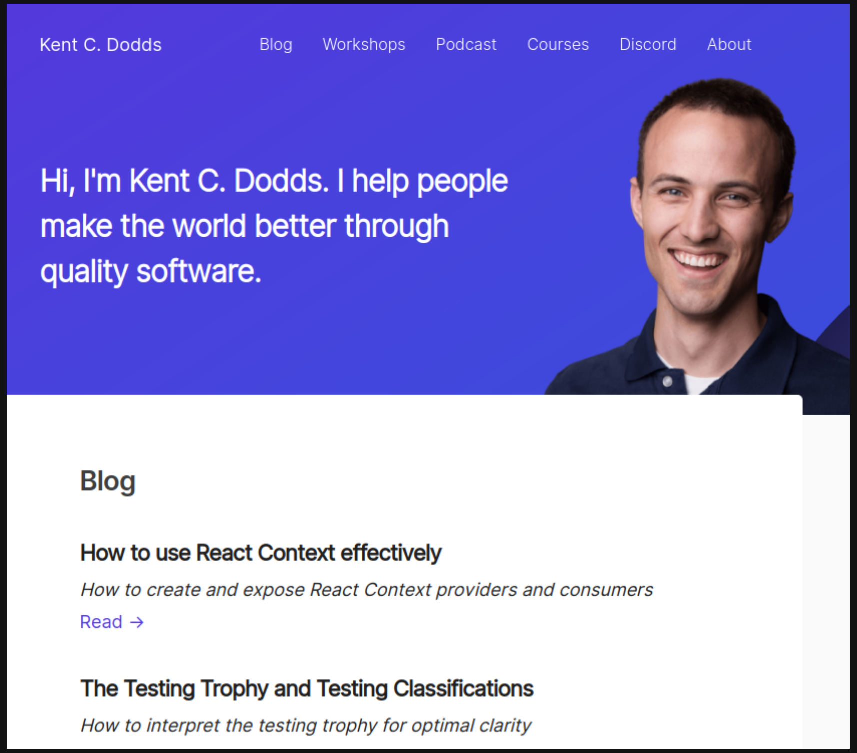
How Stripe Designs Beautiful Websites 79,124 views
Examining the tips and tricks used to make Stripe's website design a notch above the rest.

Creating a Monorepo with Lerna & Yarn Workspaces 44,962 views
In this guide, you will learn how to create a Monorepo to manage multiple packages with a shared build, test, and release process.



Cassidy Williams
Software Engineer in Chicago

Hi there! My name is Cassidy and I'm a Director of Developer Experience at Netlify. I often make silly videos on the internet, plus I enjoy building mechanical keyboards, playing music, and teaching in my free time.

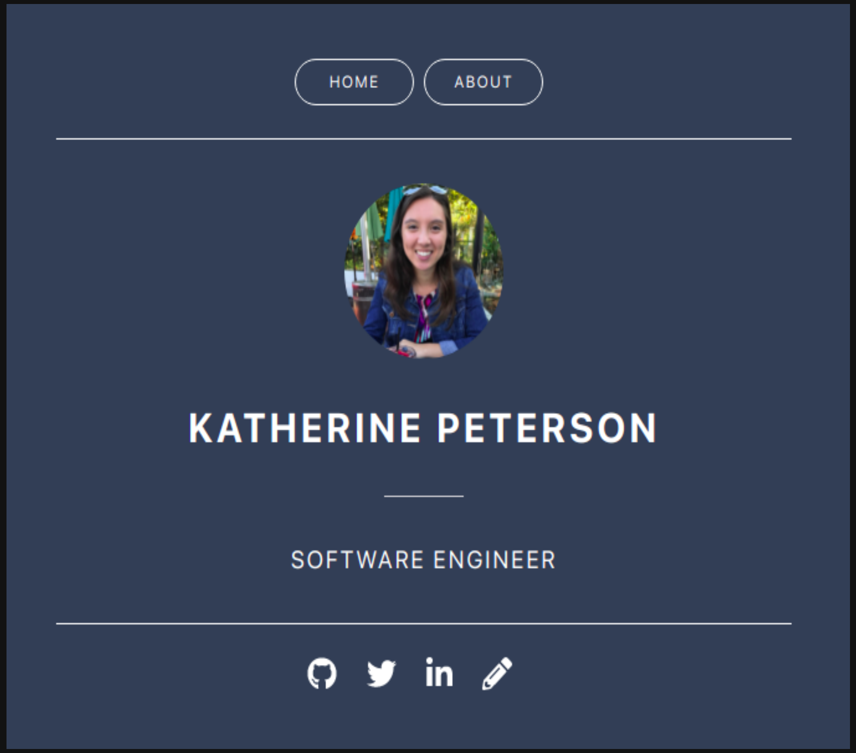


Hi, I'm Kent C. Dodds. I help people make the world better through quality software.

Blog

How to use React Context effectively
How to create and expose React Context providers and consumers

The Testing Trophy and Testing Classifications
How to interpret the testing trophy for optimal clarity

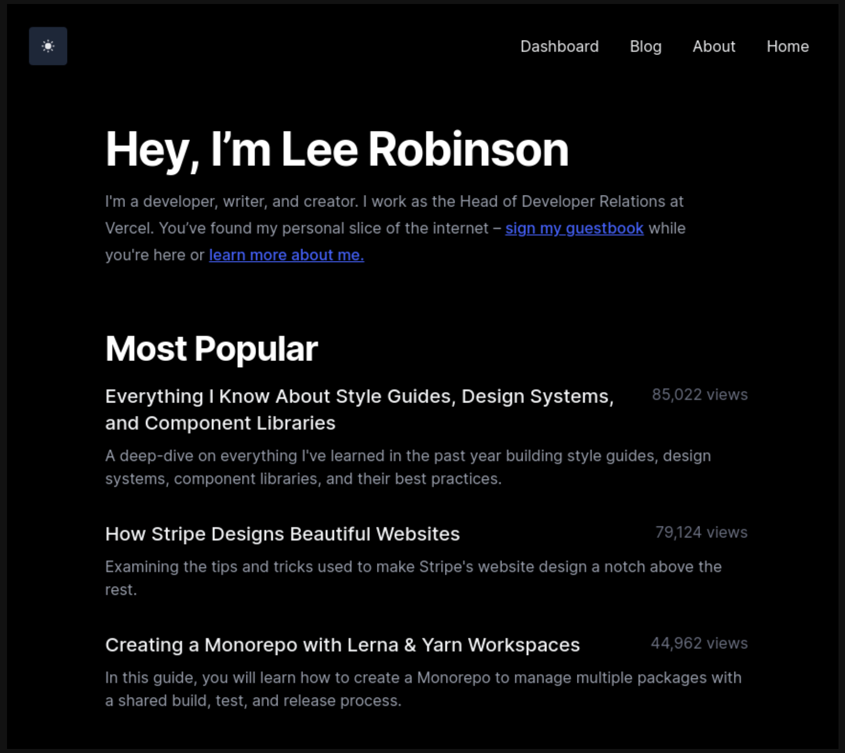


KATHERINE PETERSON

SOFTWARE ENGINEER

Why?

- Show your background
- Prove that you have the know-how
- Convey your creativity and dedication
- Strengthen your personal branding
- Stand out among others



Hey, I'm Lee Robinson

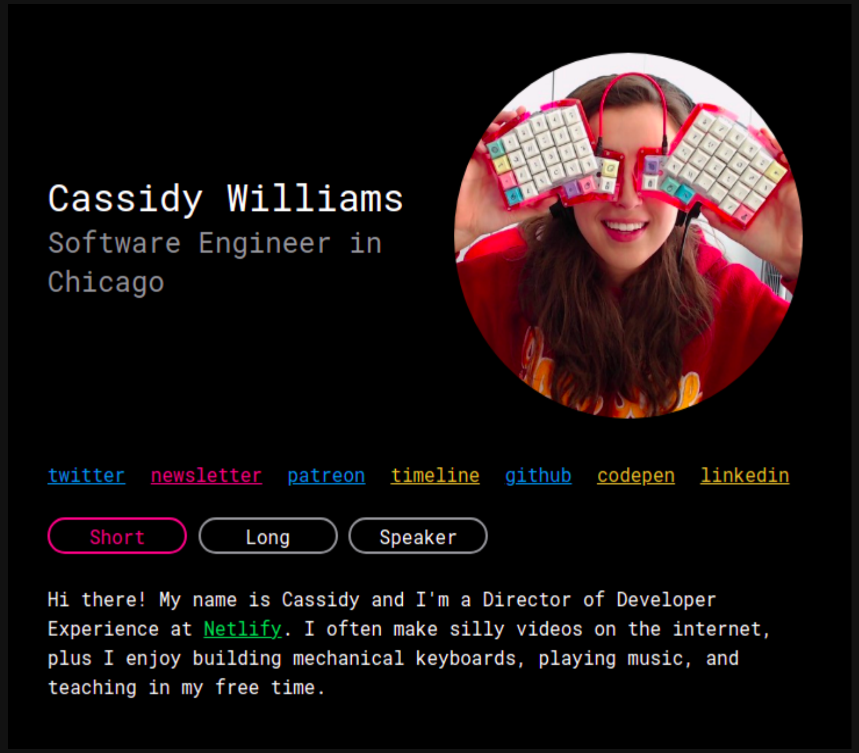
I'm a developer, writer, and creator. I work as the Head of Developer Relations at Vercel. You've found my personal slice of the internet – sign my guestbook while you're here or learn more about me.

Most Popular

Everything I Know About Style Guides, Design Systems, and Component Libraries — 85,022 views
A deep-dive on everything I've learned in the past year building style guides, design systems, component libraries, and their best practices.

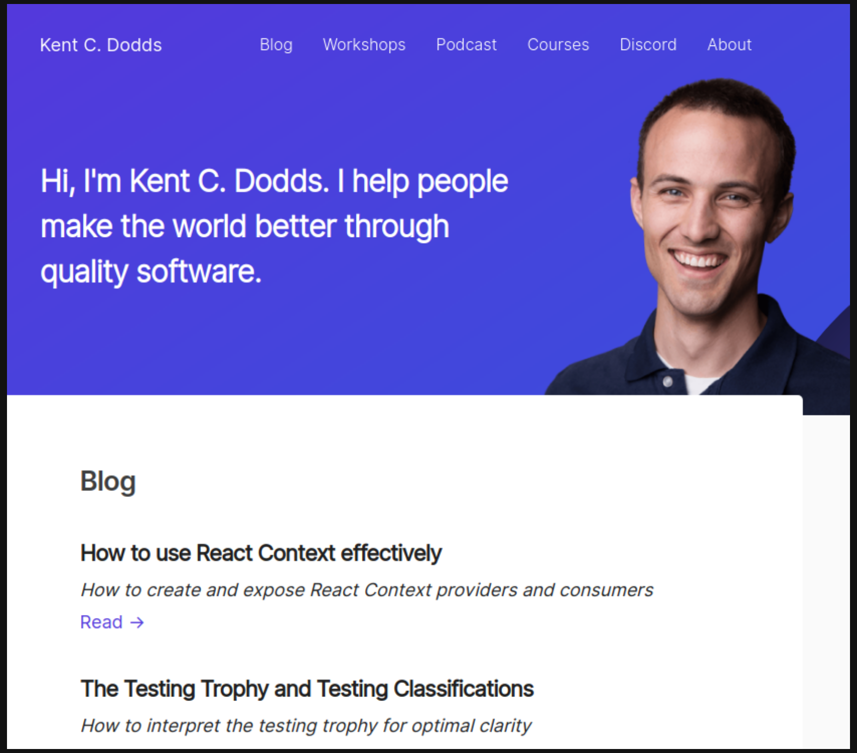
How Stripe Designs Beautiful Websites — 79,124 views
Examining the tips and tricks used to make Stripe's website design a notch above the rest.

Creating a Monorepo with Lerna & Yarn Workspaces — 44,962 views
In this guide, you will learn how to create a Monorepo to manage multiple packages with a shared build, test, and release process.



Cassidy Williams
Software Engineer in Chicago

Hi there! My name is Cassidy and I'm a Director of Developer Experience at Netlify. I often make silly videos on the internet, plus I enjoy building mechanical keyboards, playing music, and teaching in my free time.

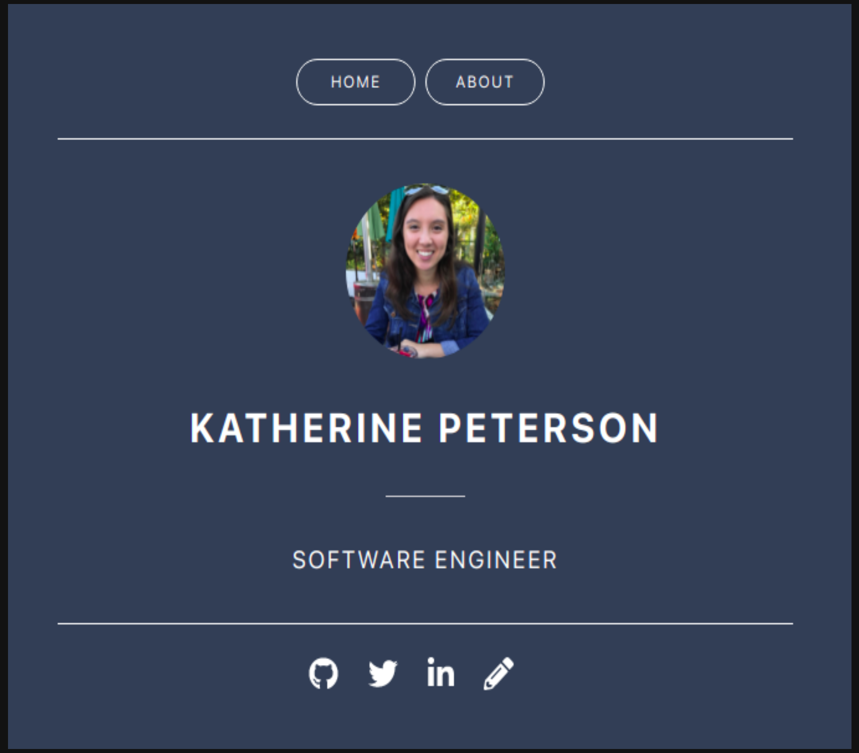


Hi, I'm Kent C. Dodds. I help people make the world better through quality software.

Blog

How to use React Context effectively
How to create and expose React Context providers and consumers
Read →

The Testing Trophy and Testing Classifications
How to interpret the testing trophy for optimal clarity



KATHERINE PETERSON

SOFTWARE ENGINEER

Where to start?

jekyll

HUGO

NEXT.JS

Static vs. Dynamic Website

Static

➕ Blazing fast

➕ Quick to develop

➕ Cheap and easier to host

➖ Can’t be tailored to individual visitors

➖ Functionalities limited by client and premade assets

➖ Impossible to store secrets for external API authentication

Dynamic

➕ Can provide tailored experience

➕ More powerful and flexible

➕ Allows secure authenticated interaction with external APIs

➖ Slower compared to static websites

➖ Require knowledge of backend programming

➖ Expensive and harder to host

Next.js

"The React Framework for Production"

Allows pre-rendering in two forms:

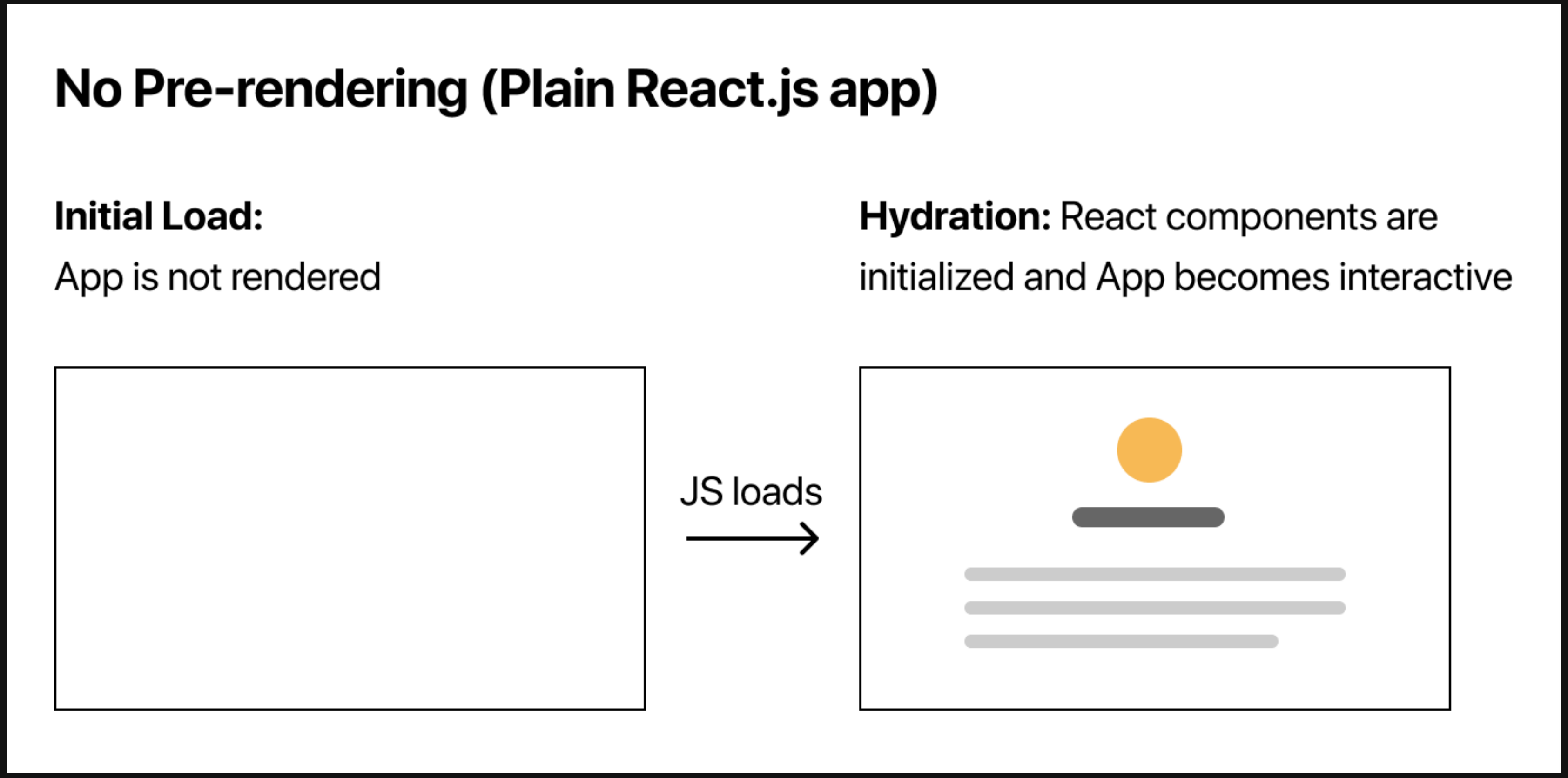
- Static Site Generation (SSG)
- Server Side Rendering (SSR)

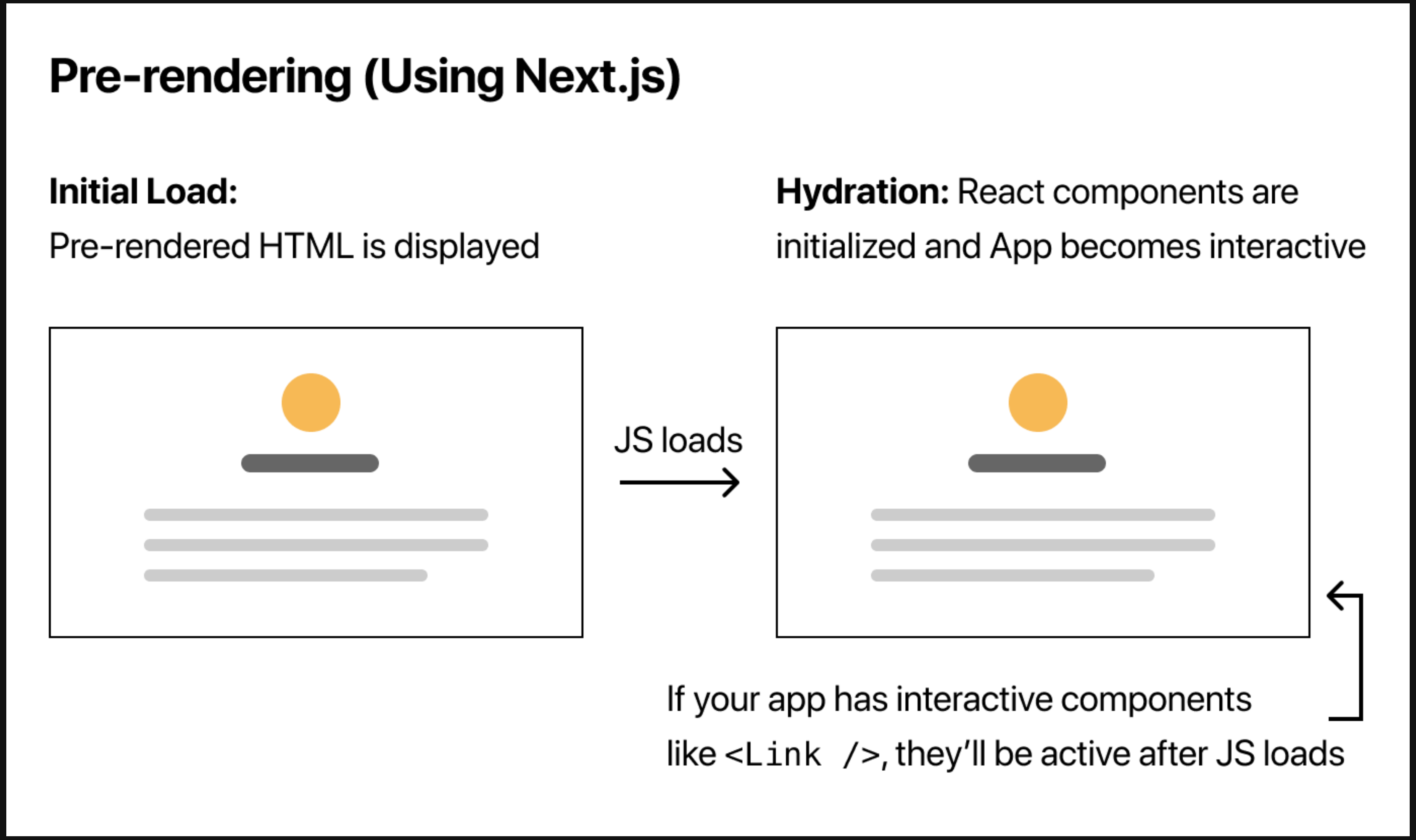
Best of both worlds!

In addition...

- API Routes
- Incremental Static Regeneration (ISR)

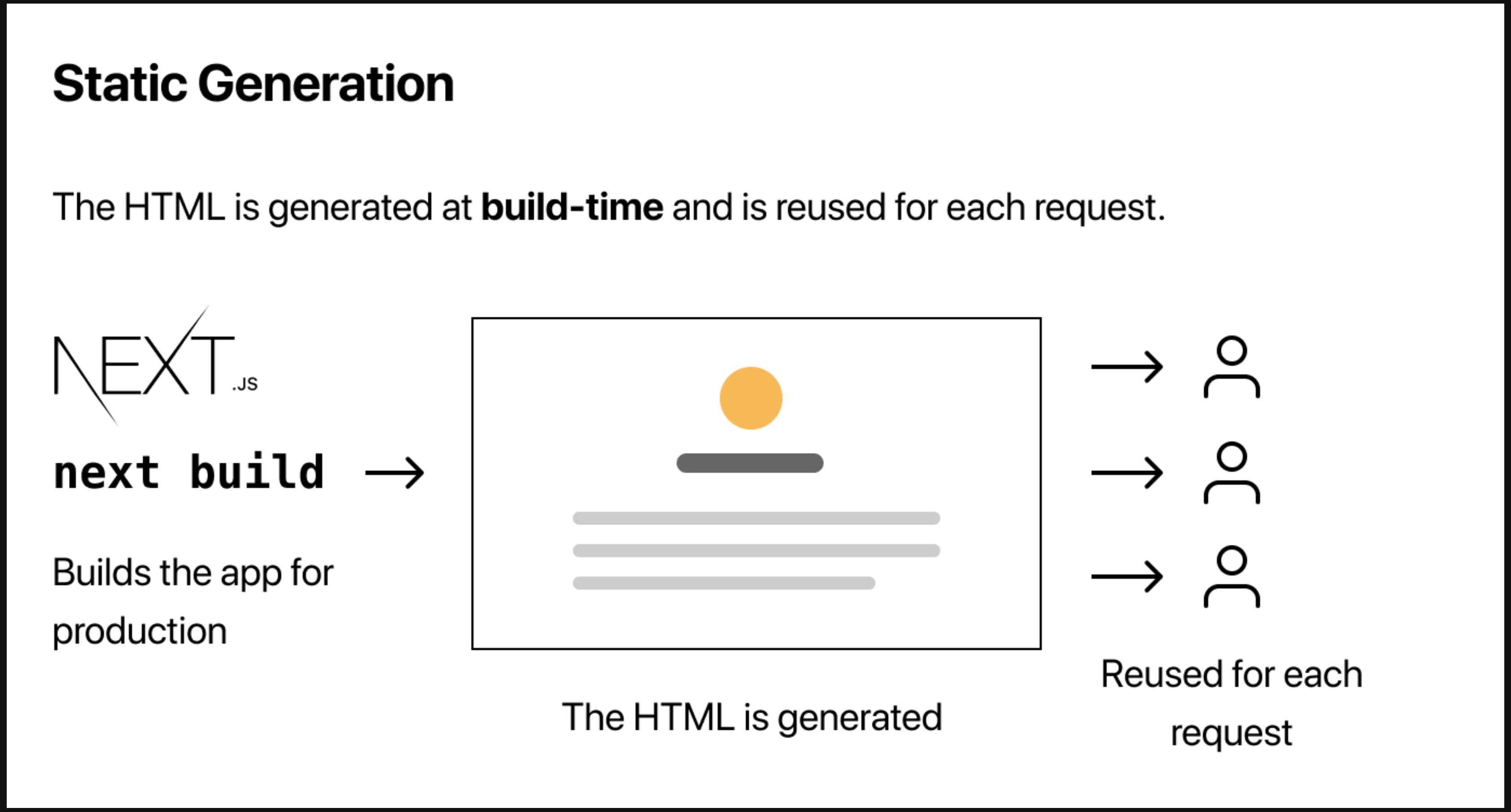
Pre-rendering

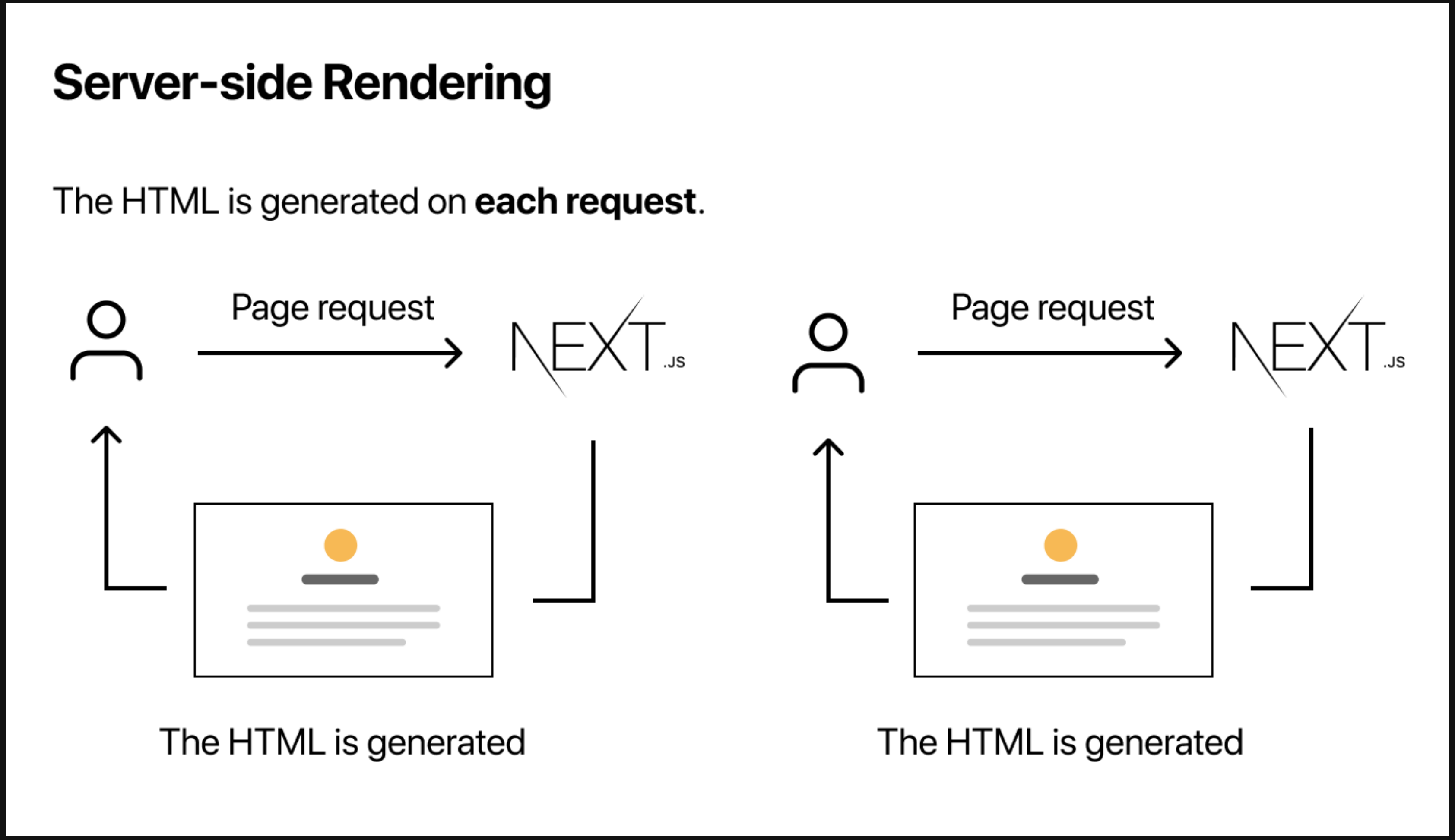

No Pre-rendering (Plain React.js app)
Initial Load:
App is not rendered
Hydration: React components are initialized and App becomes interactive
JS loads


Pre-rendering (Using Next.js)
Initial Load:
Pre-rendered HTML is displayed
Hydration: React components are initialized and App becomes interactive
JS loads
If your app has interactive components like <Link />, they'll be active after JS loads

See nextjs.org/learn/basics/data-fetching/pre-rendering

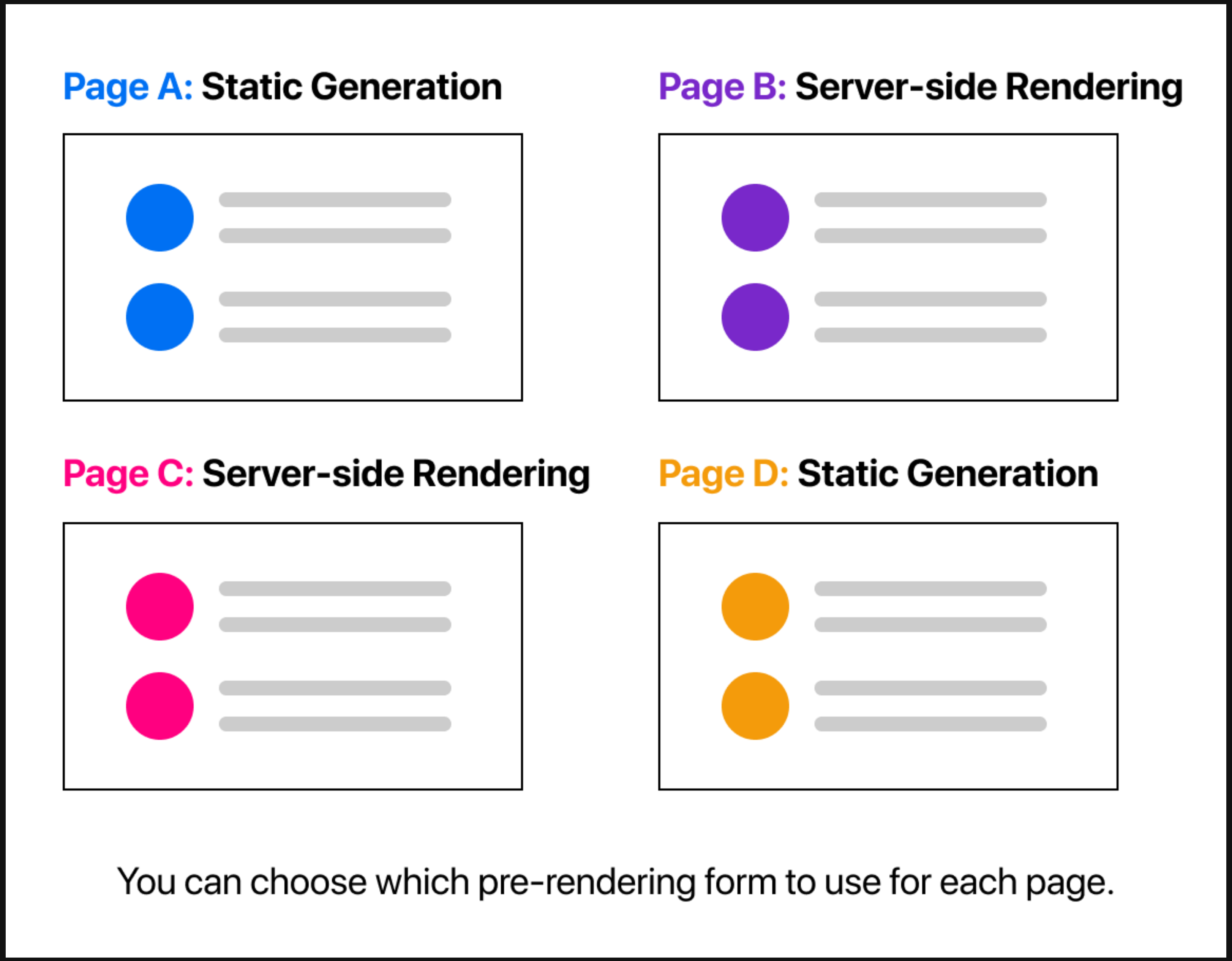
Two Forms of Pre-rendering


Static Generation
The HTML is generated at **build-time** and is reused for each request.
next build
Builds the app for production
The HTML is generated
Reused for each request



See https://nextjs.org/learn/basics/data-fetching/two-forms

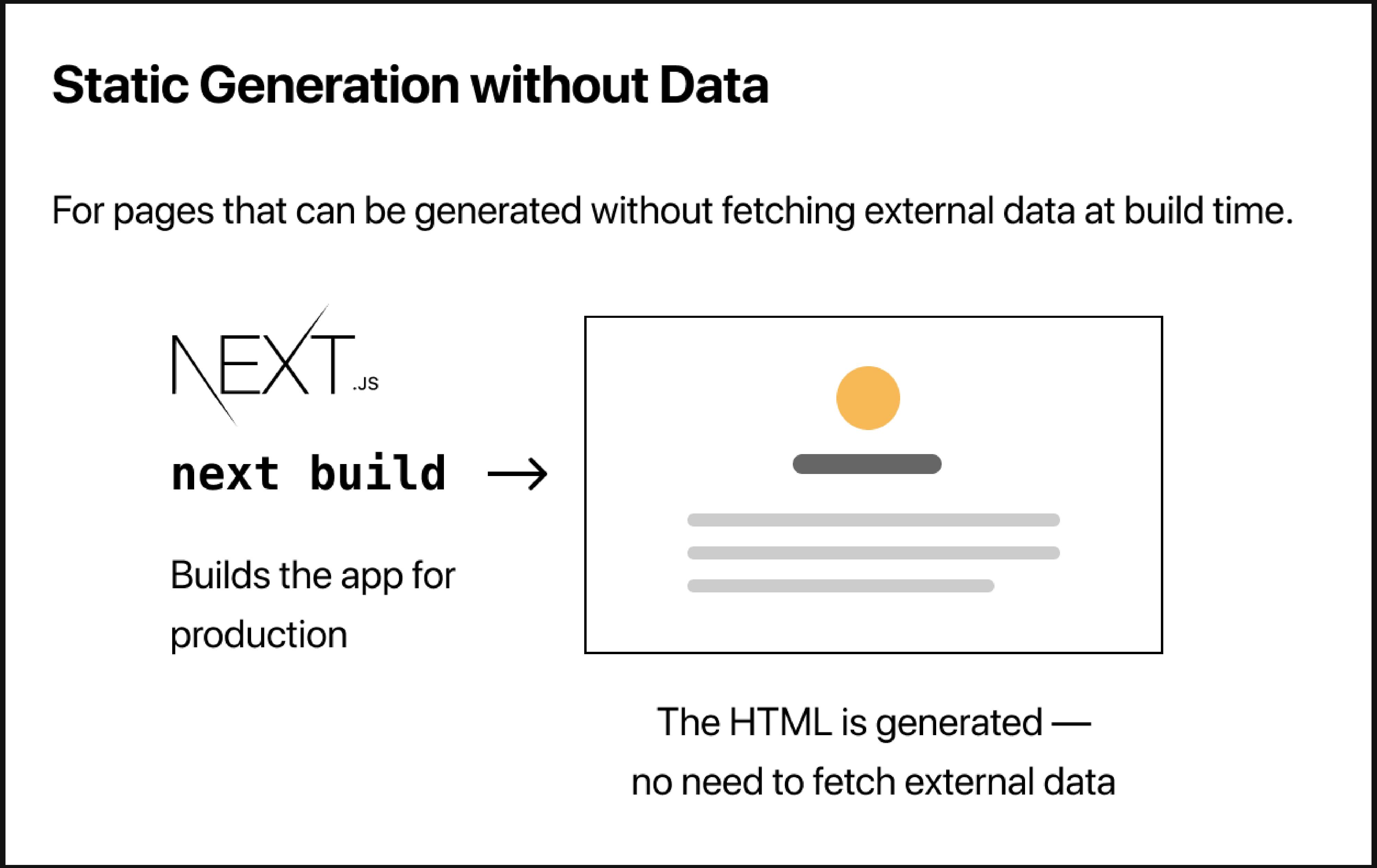
Per-page Basis


Page A: Static Generation
Page B: Server-side Rendering
Page C: Server-side Rendering
Page D: Static Generation

You can choose which pre-rendering form to use for each page.

See https://nextjs.org/learn/basics/data-fetching/two-forms

Static Generation without Data



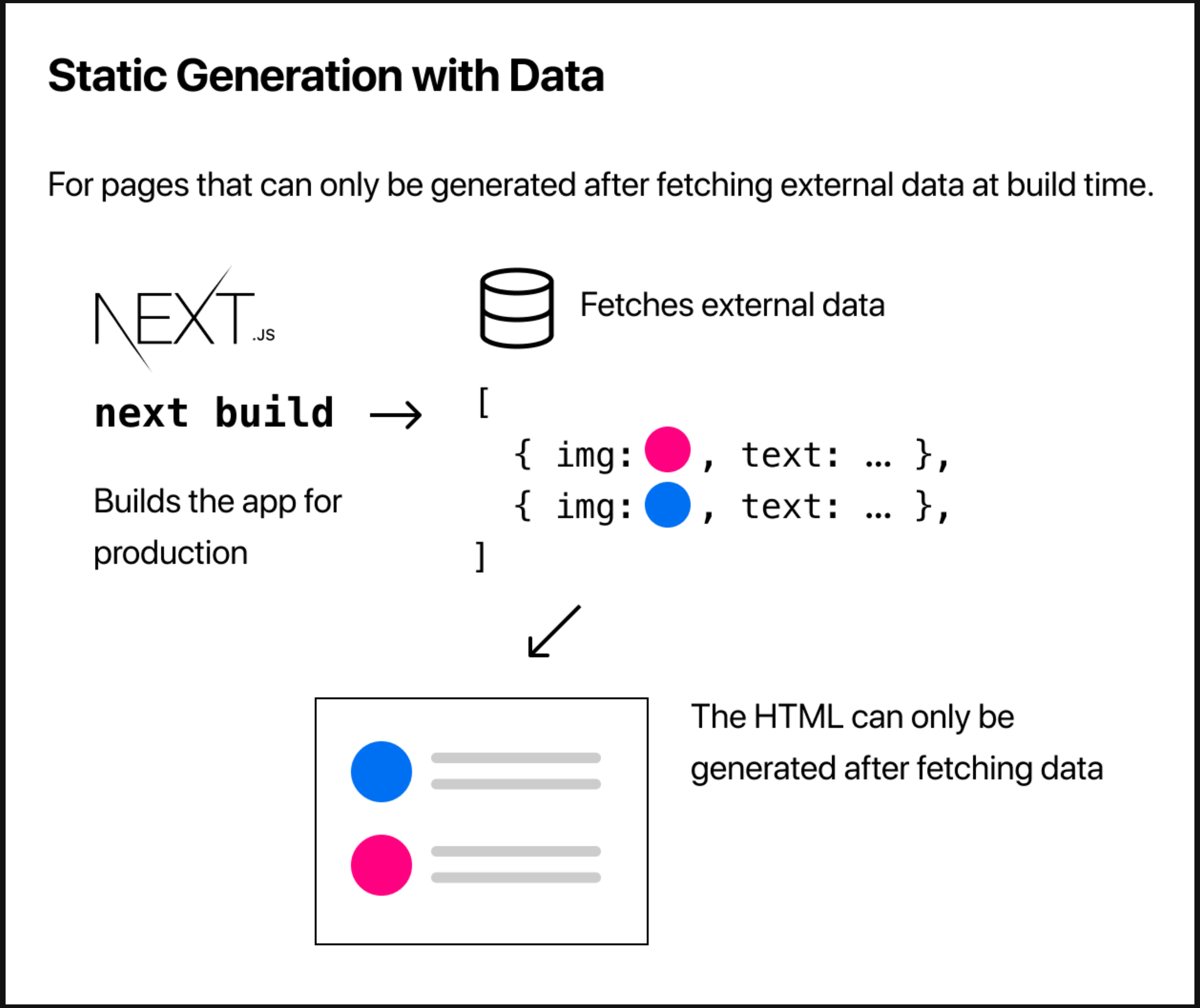
Static Generation without Data

For pages that can be generated without fetching external data at build time.

NEXT.JS
next build →
Builds the app for production
The HTML is generated — no need to fetch external data

See https://nextjs.org/learn/basics/data-fetching/with-data

Static Generation with Data


Static Generation with Data
For pages that can only be generated after fetching external data at build time.
NEXT.JS
next build
Builds the app for production
Fetches external data
[{ img: ●, text: … }, { img: ●, text: … },]
The HTML can only be generated after fetching data

See https://nextjs.org/learn/basics/data-fetching/with-data

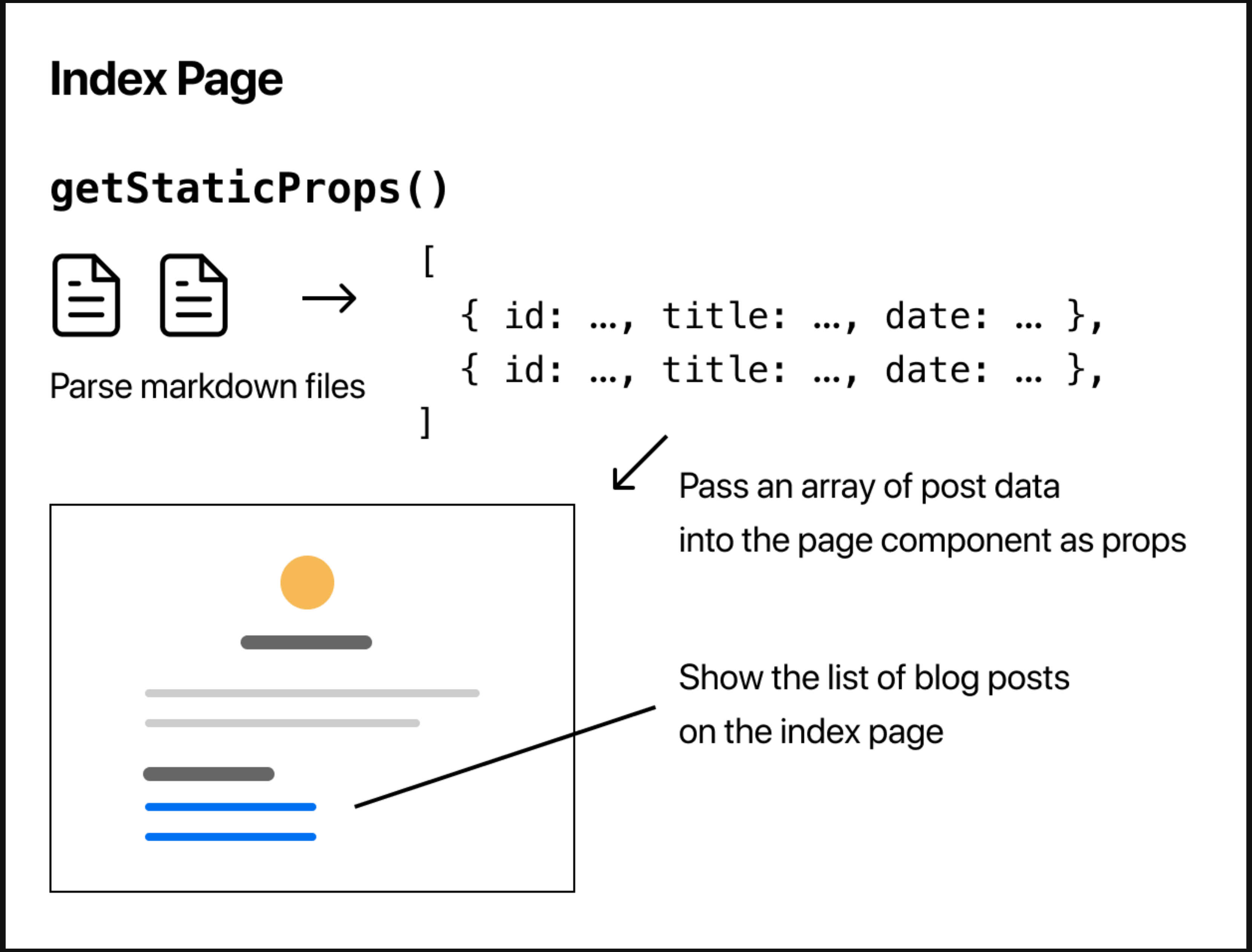
`getStaticProps`

```
export default function Home(props) { ... }

export async function getStaticProps() {
  // Get external data from the file system, API, DB, etc.
  const data = ...

  // The value of the `props` key will be
  //  passed to the `Home` component
  return {
    props: ...
  }
}
```

Example: Blog Data


Index Page
getStaticProps()
Parse markdown files
Pass an array of post data into the page component as props
Show the list of blog posts on the index page

```
[
  { id: …, title: …, date: … },
  { id: …, title: …, date: … },
]
```

See https://nextjs.org/learn/basics/data-fetching/blog-data

Limitations?

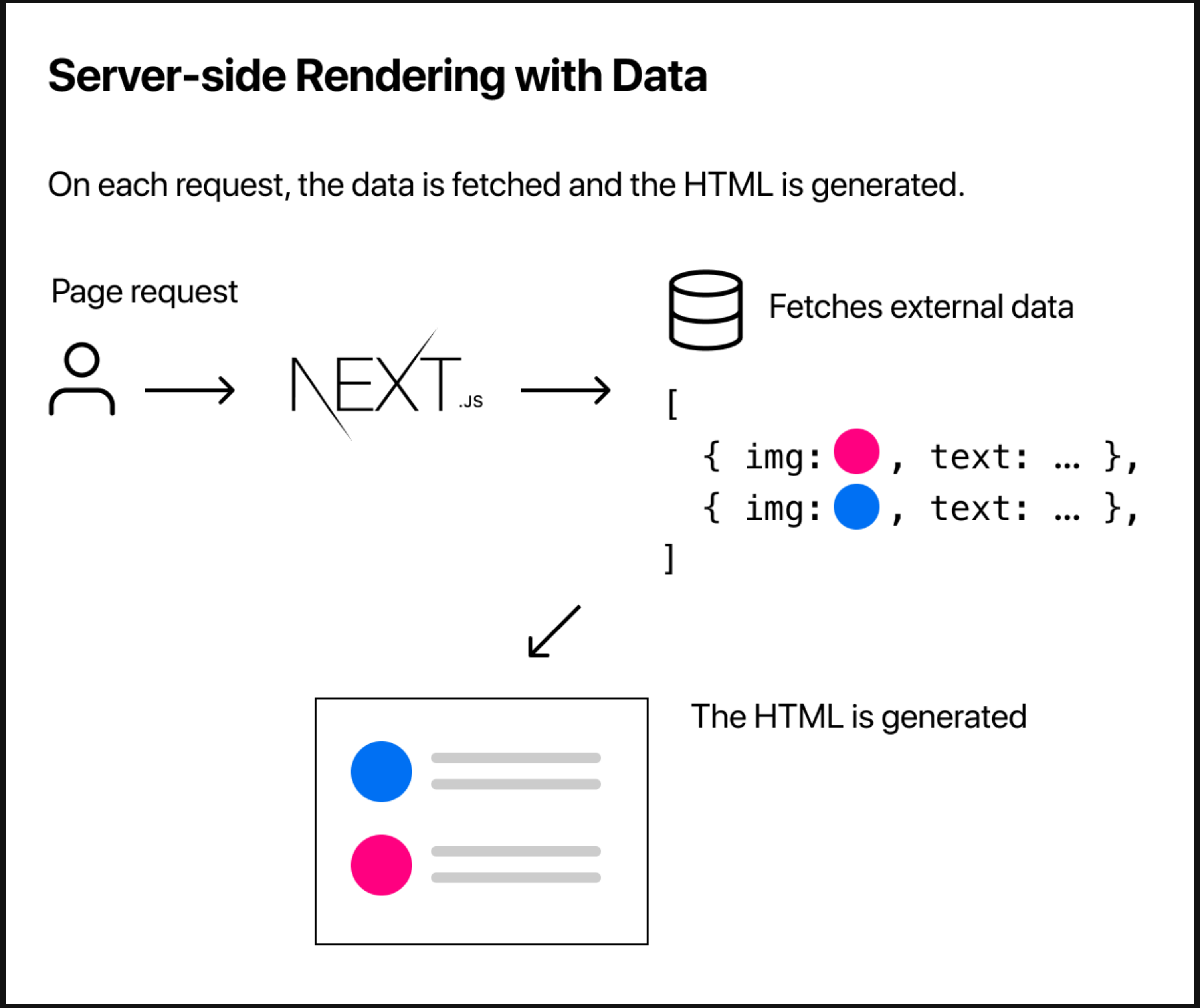
Data is only "fetched" and rendered into the pages at **build time**.

What if the data is constantly updated?

Stale data! 😦

Solution: fetch data at **request time**! 😄

Server-side Rendering with Data



See https://nextjs.org/learn/basics/data-fetching/request-time

`getServerSideProps`

```
export default function Home(props) { ... }

export async function getServerSideProps(context) {
  return {
    props: {
      // props for your component
    }
  }
}
```

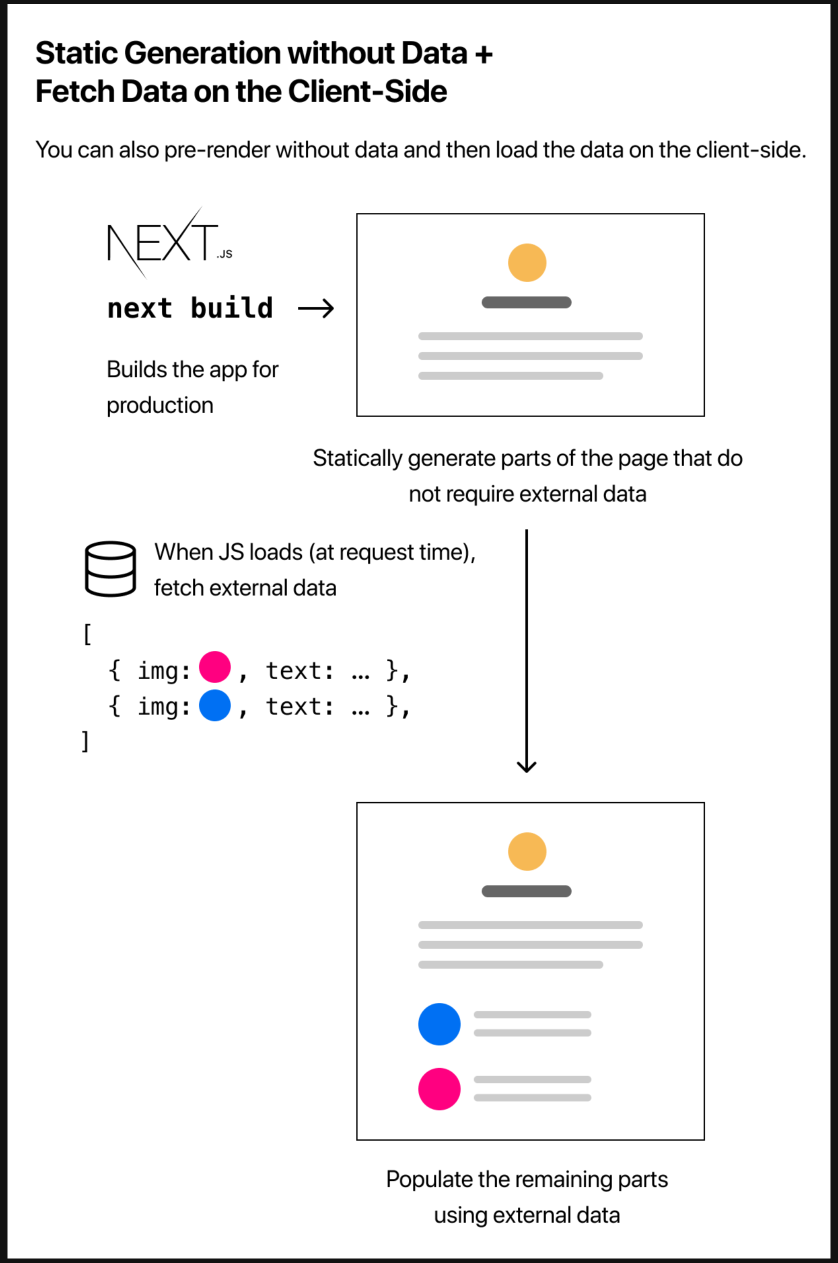
Client-side Rendering?

Server-side rendering might be **costly**

Data might **take a while** to be fetched

Increase **responsiveness** of the page → **client side rendering** 🤔

Client-side Rendering



**Static Generation without Data +
Fetch Data on the Client-Side**

You can also pre-render without data and then load the data on the client-side.

```
next build
```

Builds the app for production

Statically generate parts of the page that do not require external data

When JS loads (at request time), fetch external data

```
[
  { img: ●, text: … },
  { img: ●, text: … },
]
```

Populate the remaining parts using external data

See https://nextjs.org/learn/basics/data-fetching/request-time

More...

- Dynamic Routes
- API Routes
- Incremental Static Regeneration (ISR)

Any questions?

Next: **Next.js Hands-on**

Thank you!

Slides available on slides.laymonage.com/nextjs-portfolio

Slides source code available on github.com/laymonage/slides-nextjs-portfolio

Made with sli.dev

Content based on nextjs.org/learn