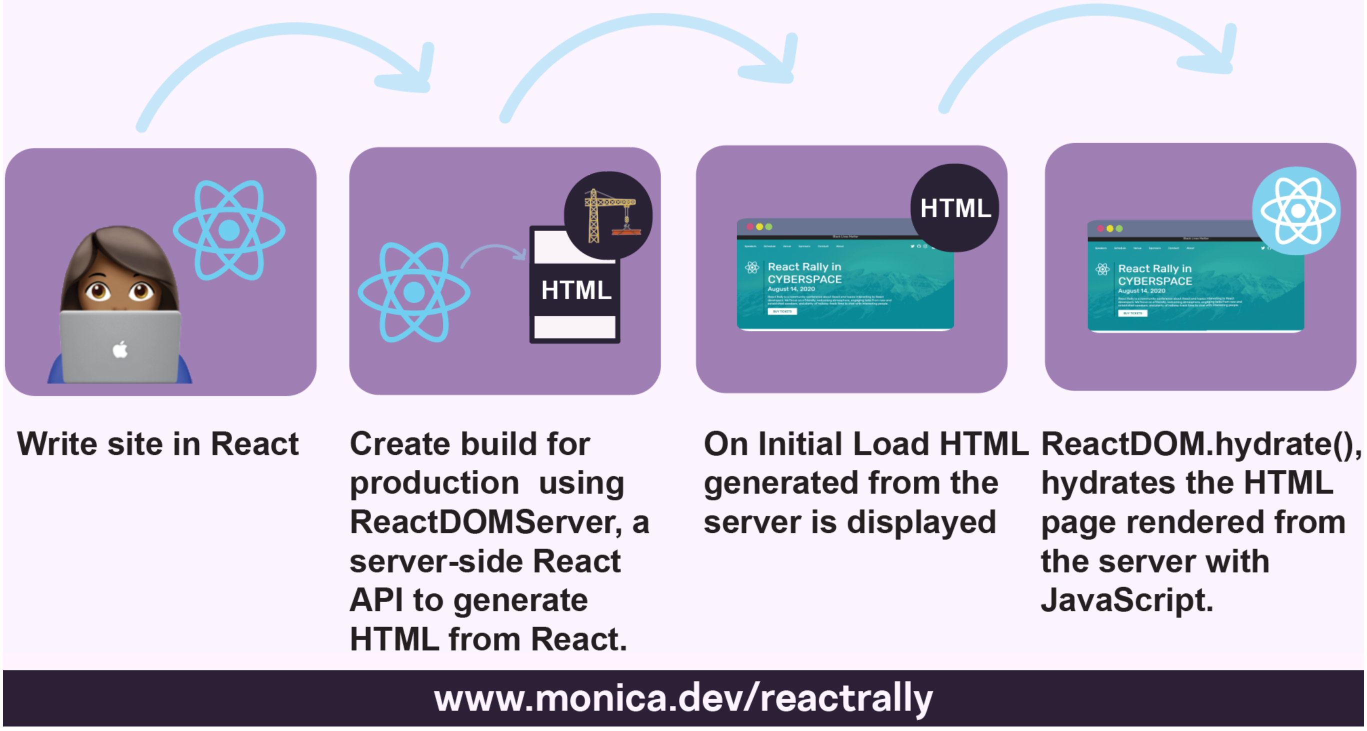
CSR, SSR, SSG and ISR



Write site in React

Create build for production using ReactDOMServer, a server-side React API to generate HTML from React.

On Initial Load HTML generated from the server is displayed

ReactDOM.hydrate(), hydrates the HTML page rendered from the server with JavaScript.

www.monica.dev/reactrally

Source: SSR React Hydrate best practices | aboutmonica.com

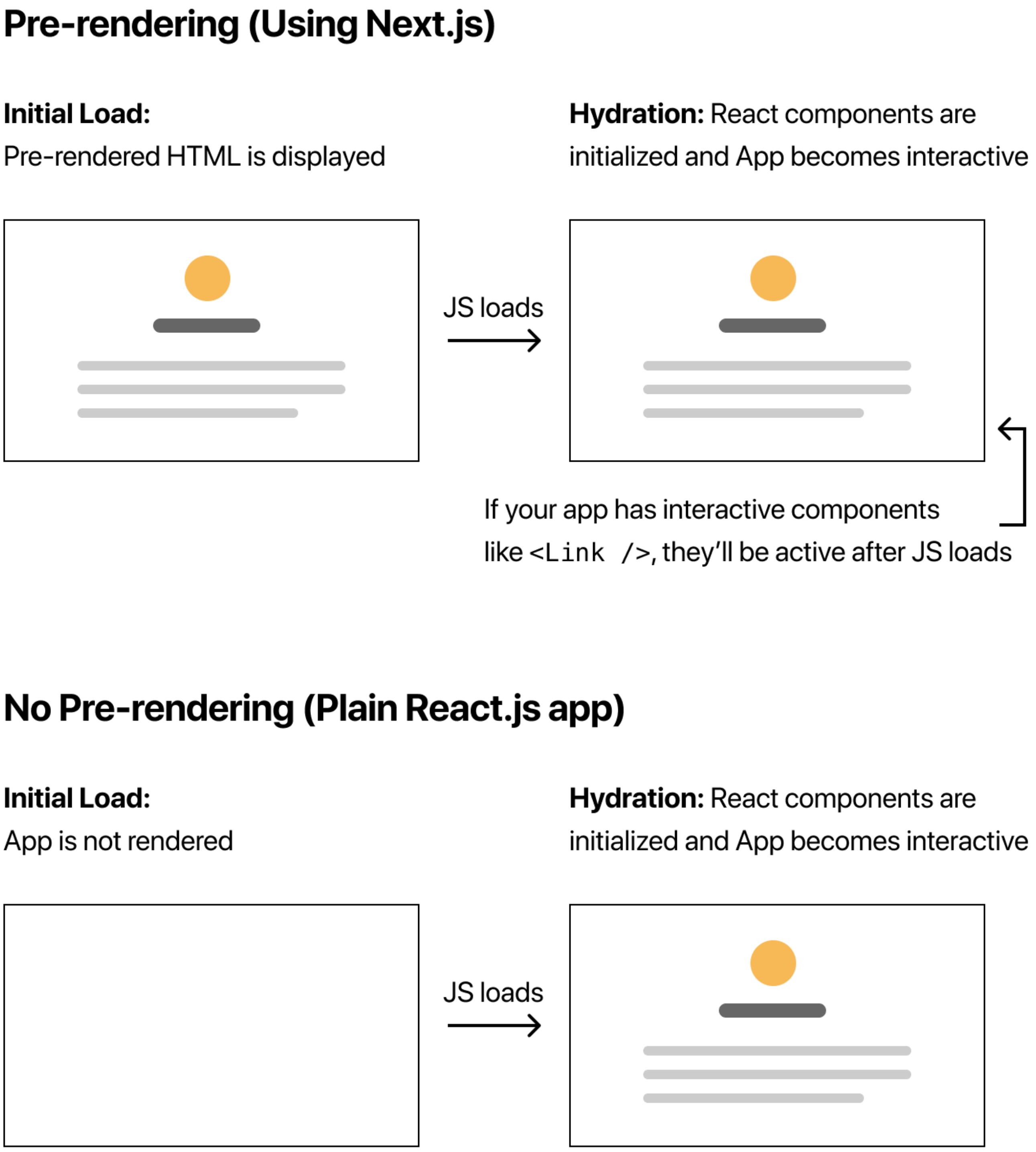
AGENDA

1. Pre-rendering
2. SSG - Static Site Generation
3. SSR - Server Side Rendering
4. CSR - Client Side Rendering
5. ISR - Incremental Static Regeneration

1. Pre-rendering

Two forms of pre-rendering:

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



Pre-rendering (Using Next.js)

Initial Load: Pre-rendered HTML is displayed

Hydration: React components are initialized and App becomes interactive

If your app has interactive components like <Link />, they'll be active after JS loads

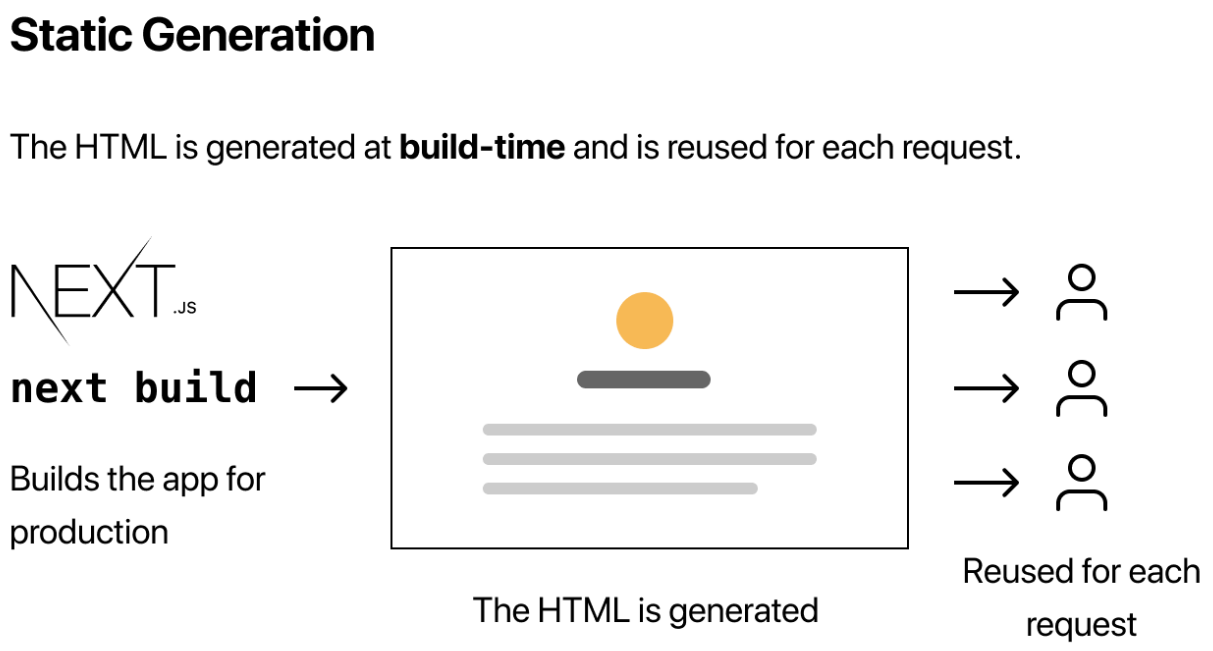
No Pre-rendering (Plain React.js app)

Initial Load: App is not rendered

Hydration: React components are initialized and App becomes interactive

Source: Pre-rendering | NextJS Learn

2. SSG - Static Site Generation

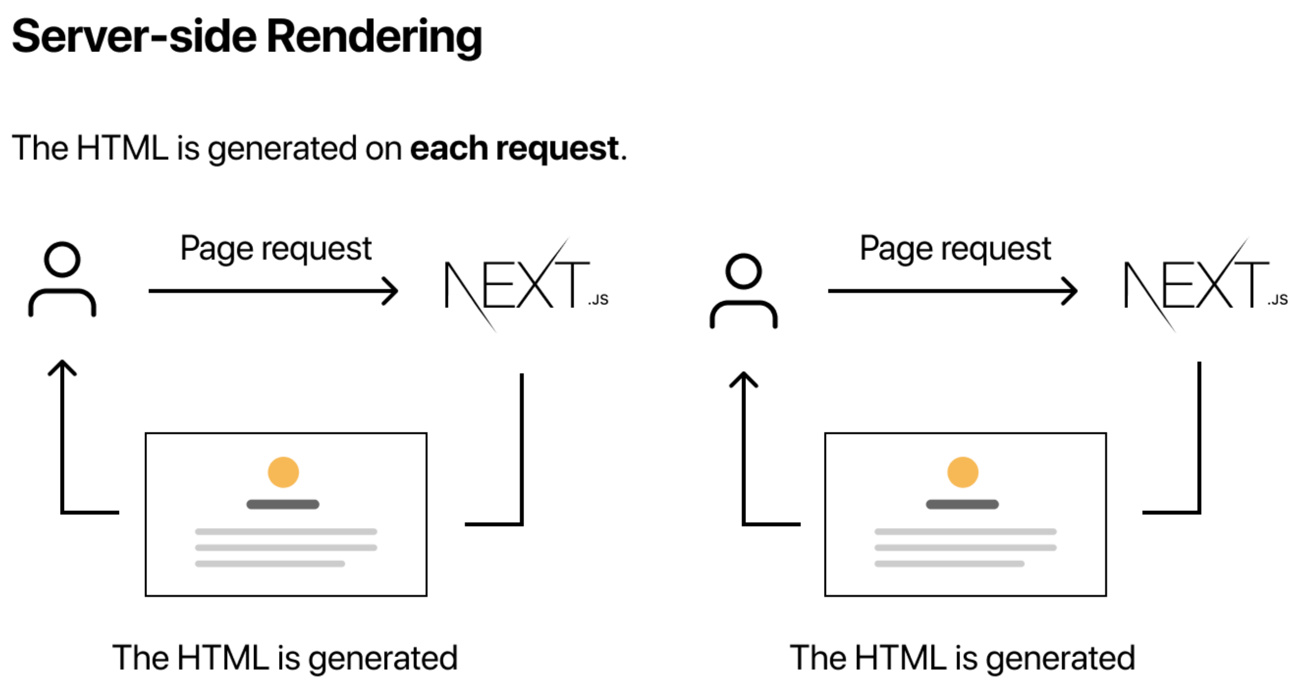


Static Generation

The HTML is generated at **build-time** and is reused for each request.

Source: *Two forms of Pre-rendering | NextJS Learn*

3. SSR - Server Side Rendering

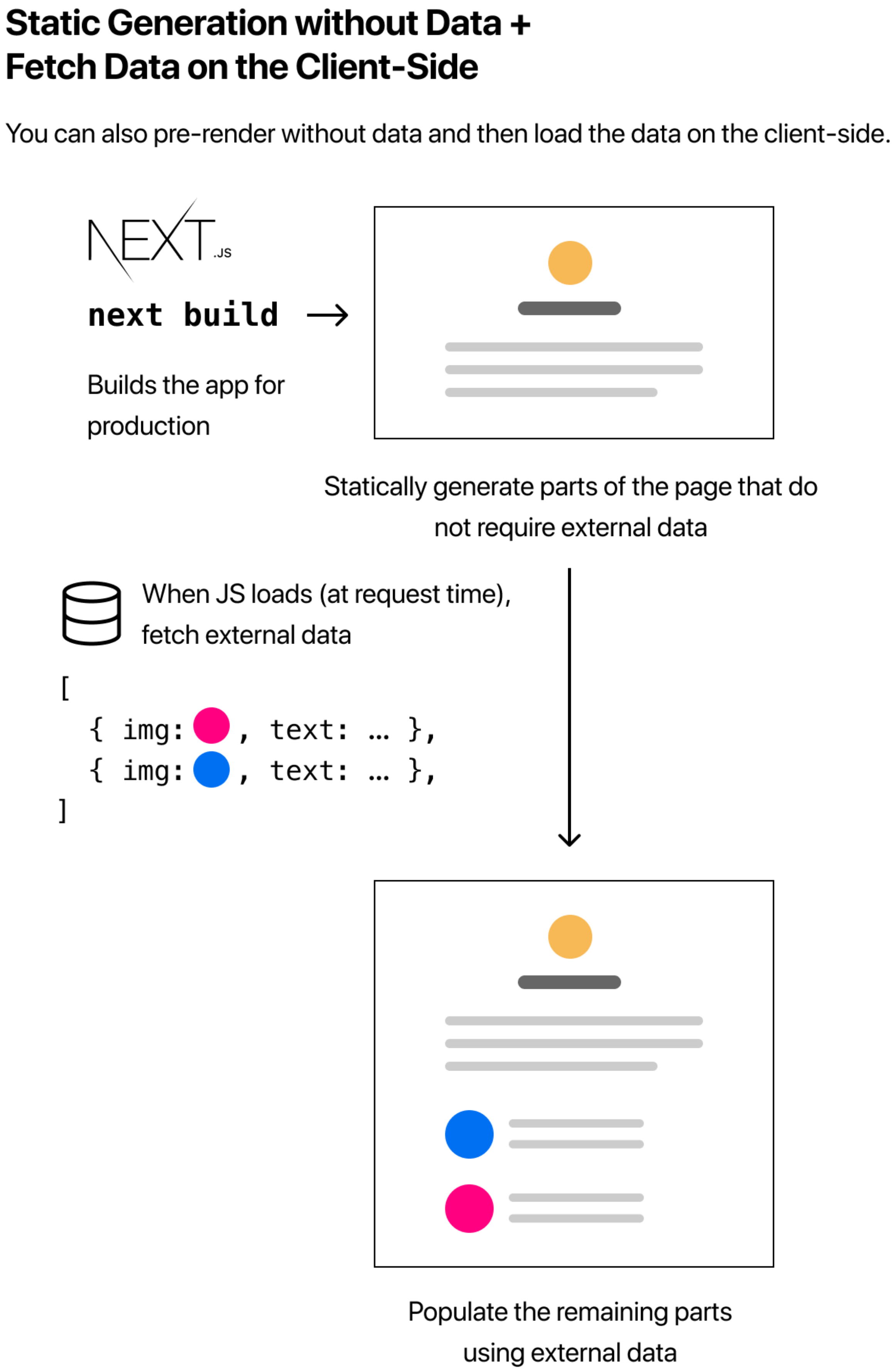


Server-side Rendering

The HTML is generated on **each request**.

Source: *Two forms of Pre-rendering | NextJS Learn*

4. CSR - 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

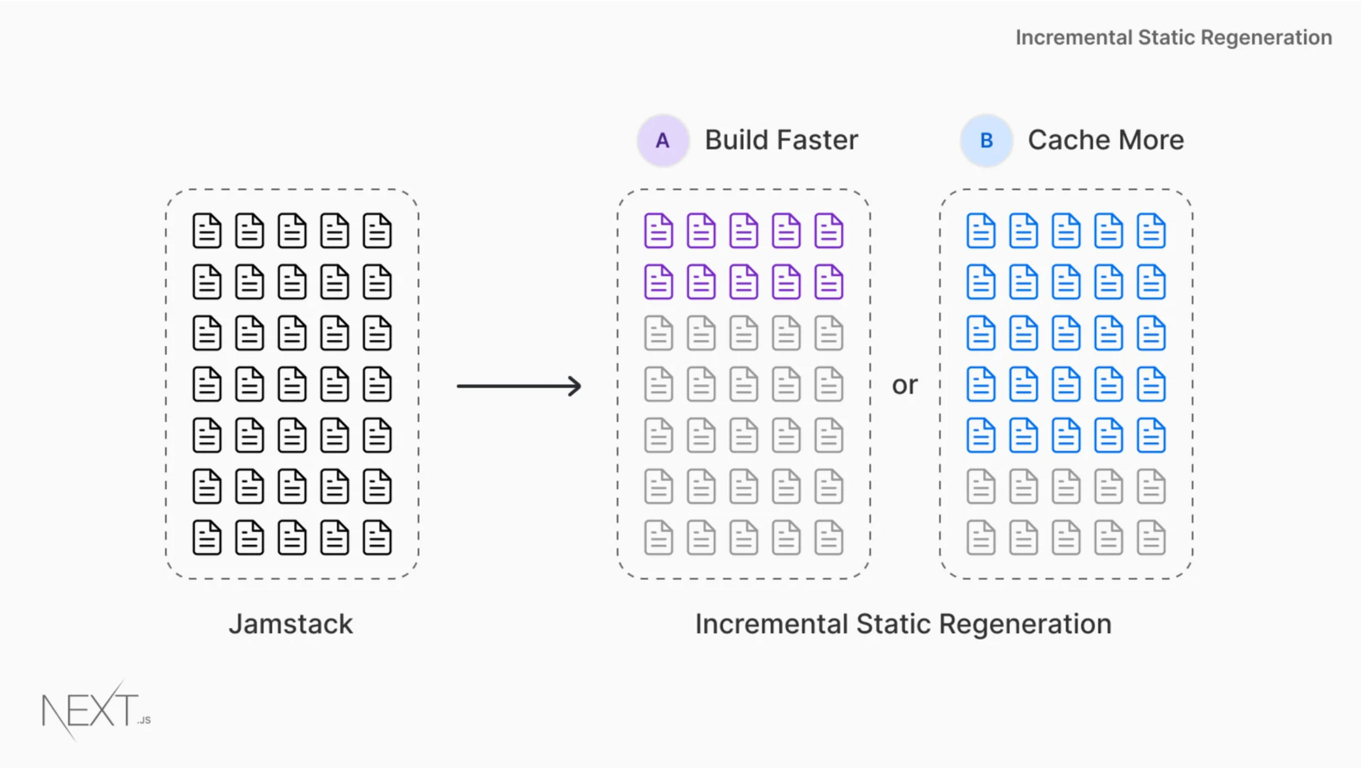
When JS loads (at request time), fetch external data

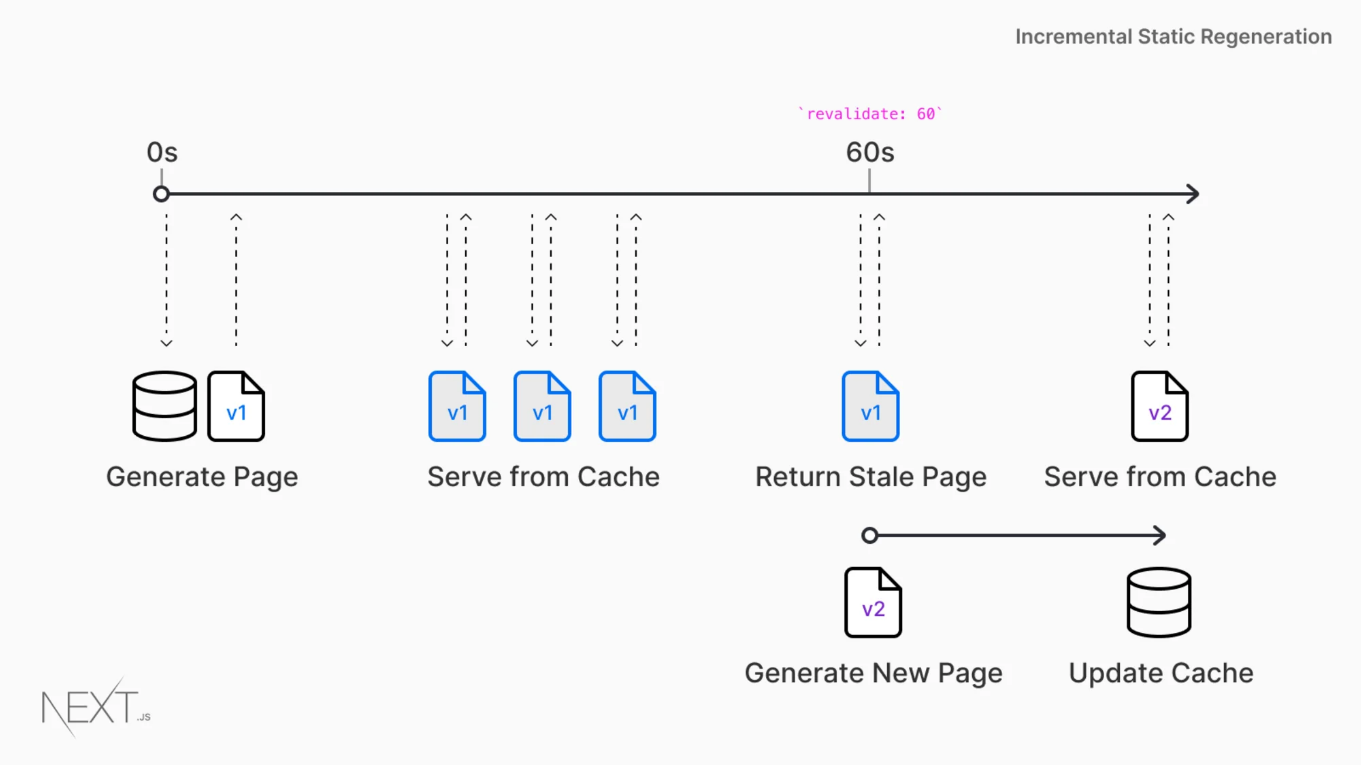
```
[
  { img: ●, text: … },
  { img: ●, text: … },
]
```

Populate the remaining parts using external data

Source: Data Fetching | NextJS Learn

5. ISR - Incremental Static Regeneration

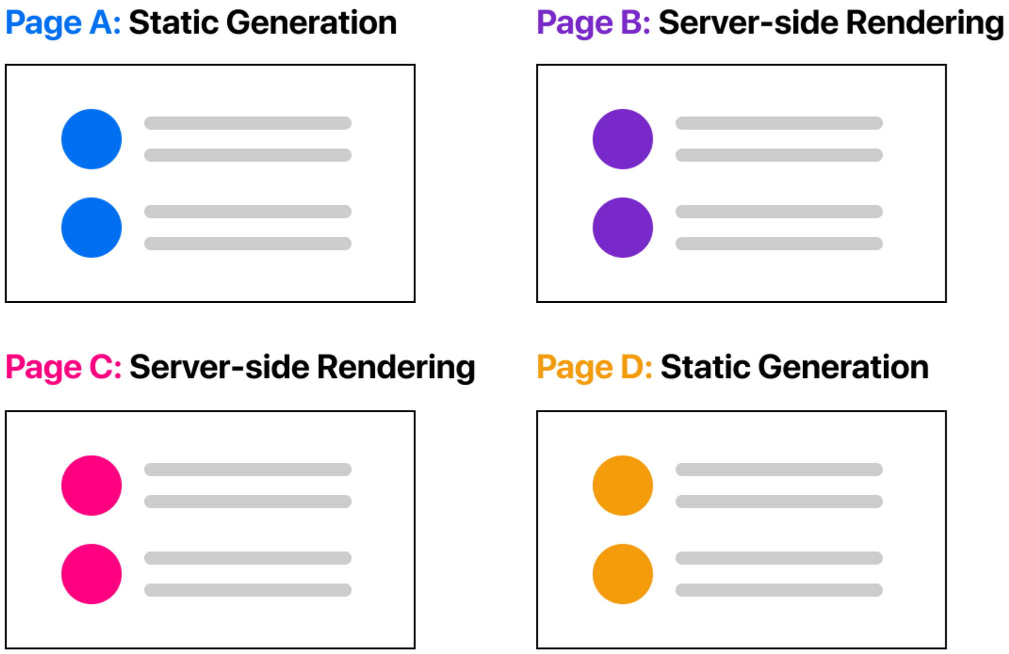

Incremental Static Regeneration
A Build Faster
B Cache More
or
Jamstack
Incremental Static Regeneration
NEXT.JS



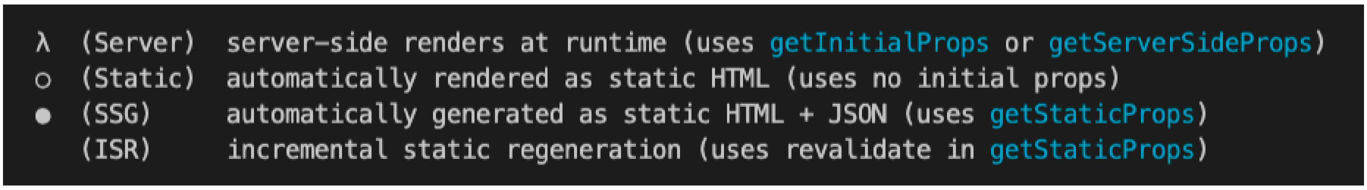
Source: Incremental Static Regeneration | vercel.com

> Read more: https://www.smashingmagazine.com/2021/04/incremental-static-regeneration-nextjs/

Good news! Pre-rendering form is 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.



```
λ  (Server)  server-side renders at runtime (uses getInitialProps or getServerSideProps)
○  (Static)  automatically rendered as static HTML (uses no initial props)
●  (SSG)     automatically generated as static HTML + JSON (uses getStaticProps)
   (ISR)     incremental static regeneration (uses revalidate in getStaticProps)
```

#	Name	Desc
1	Static	automatically rendered as static HTML (uses no initial props)
2	SSG	automatically generated as static HTML + JSON (uses `getStaticProps`)
3	ISR	incremental static regeneration (uses revalidate in `getStaticProps`)
4	SSG + CSR	pre-rendered HTML + data fetching on client side using `useEffect`
5	SSR	server-side renders at runtime (uses `getInitialProps` or `getServerSideProps`)

Tham khảo

- https://github.com/vercel/next.js/discussions/10437#discussioncomment-90459
- https://nextjs.org/learn/basics/data-fetching/pre-rendering
- https://nextjs.org/learn/basics/data-fetching/two-forms
- https://guydumais.digital/blog/next-js-the-ultimate-cheat-sheet-to-page-rendering/

Series - NextJS + Typescript 🎉

- Tác giả: **Hậu Nguyễn**
- Được phát hành trên kênh youtube **Easy Frontend**.
- Tài liệu pdf và videos đều có bản quyền thuộc về Easy Frontend.
- Videos được phát hành cho fan cứng trước, public sau.
- Đăng ký fan cứng để xem series này đầy đủ và sớm nhất nhé.

☎ Kết nối với mình

- ✅ Follow Facebook: https://www.facebook.com/nvhauesmn/
- ✅ Like Fanpage: https://www.facebook.com/learn.easyfrontend
- ✅ Youtube Channel: https://www.youtube.com/easyfrontend