MODERN WEB DEVELOPMENT ON THE

Jamstack

NICOLAS ERNY

▸ Senior Software Engineer at INVIVOO

▸ xcomponent.com

▸ Github / Twitter @nicolaserny

▸ Blog posts: dev.to/nicolaserny

INVIVOO

BEYOND TECH

100-millisecond delay in page load time costs Amazon 1% of revenue

— **Greg Linden**

JAMSTACK?

The modern way to build
[*websites* & apps]
that delivers better performance

JAMSTACK?

▶ **Only a new "buzzword"?**

▶ **Do I have to learn new frameworks all over again?**

Jamstack = JAVASCRIPT + API + MARKUP STACK

CORE PRINCIPLES #1

PRE-RENDERING

▸ **Prebuilt into highly optimized static pages and assets**

▸ **Deploy to** CDN

▸ Client side **vs SSR vs** SSG
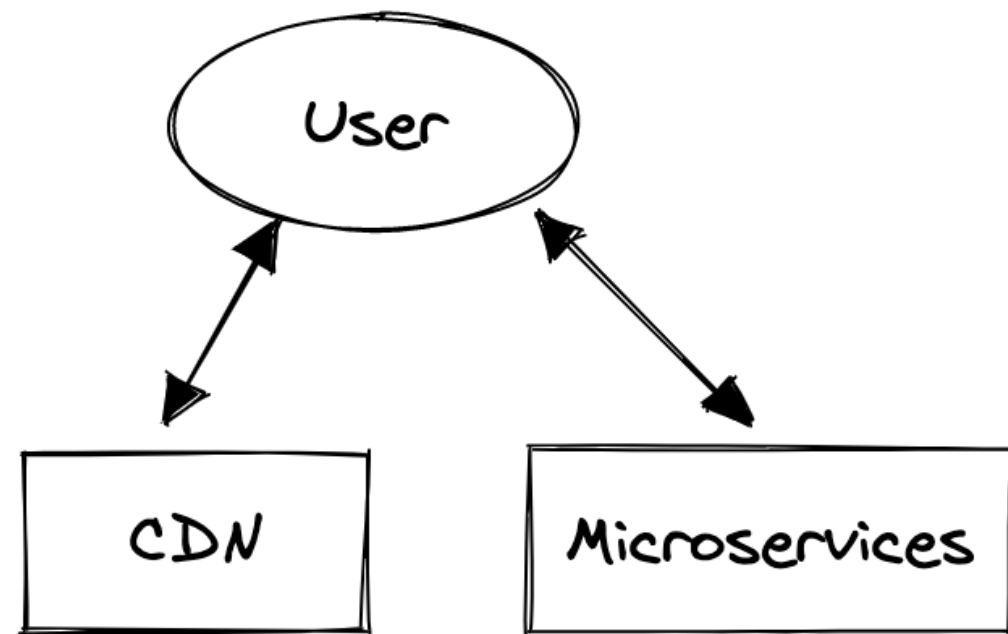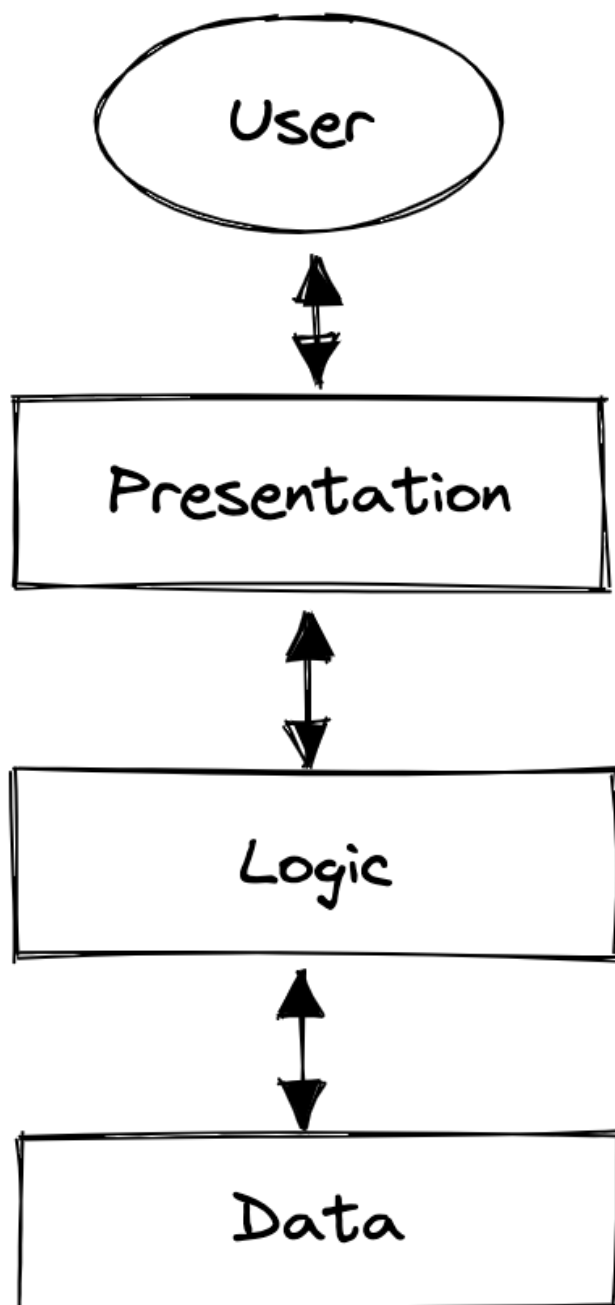
▸ **Only for static sites?**

CORE PRINCIPLES #2

DECOUPLING

- ▶ Use of APIs
- ▶ Flexibility
- ▶ Realtime data

WHY JAMSTACK?

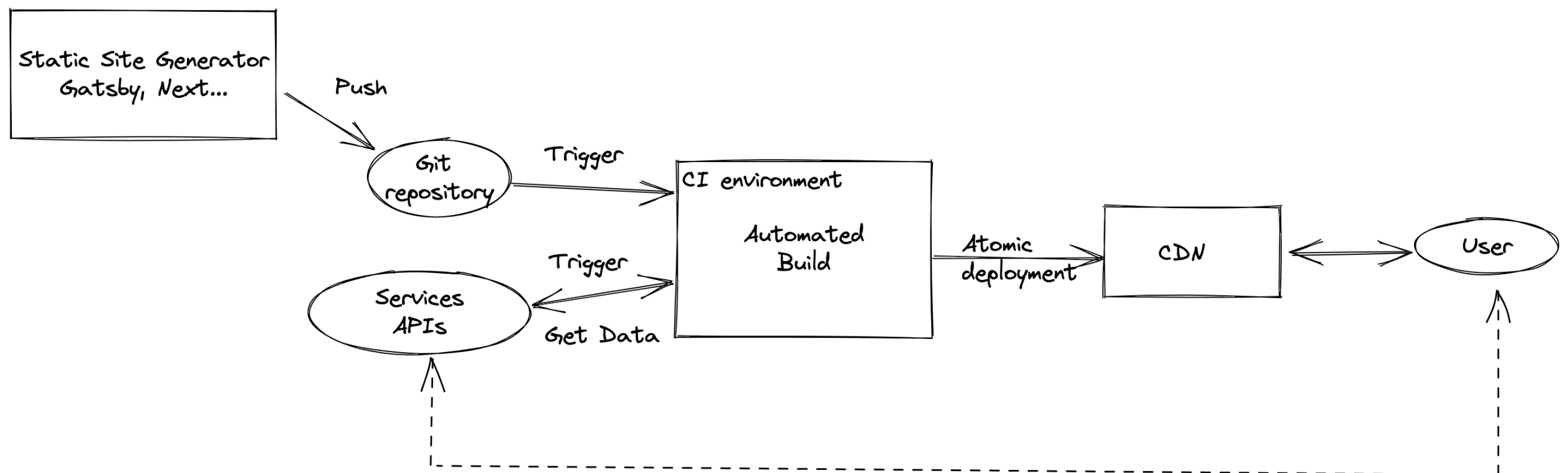- ▸ Security
- ▸ Scalability
- ▸ Performance
- ▸ Maintainability
- ▸ Portability
- ▸ Developer experience
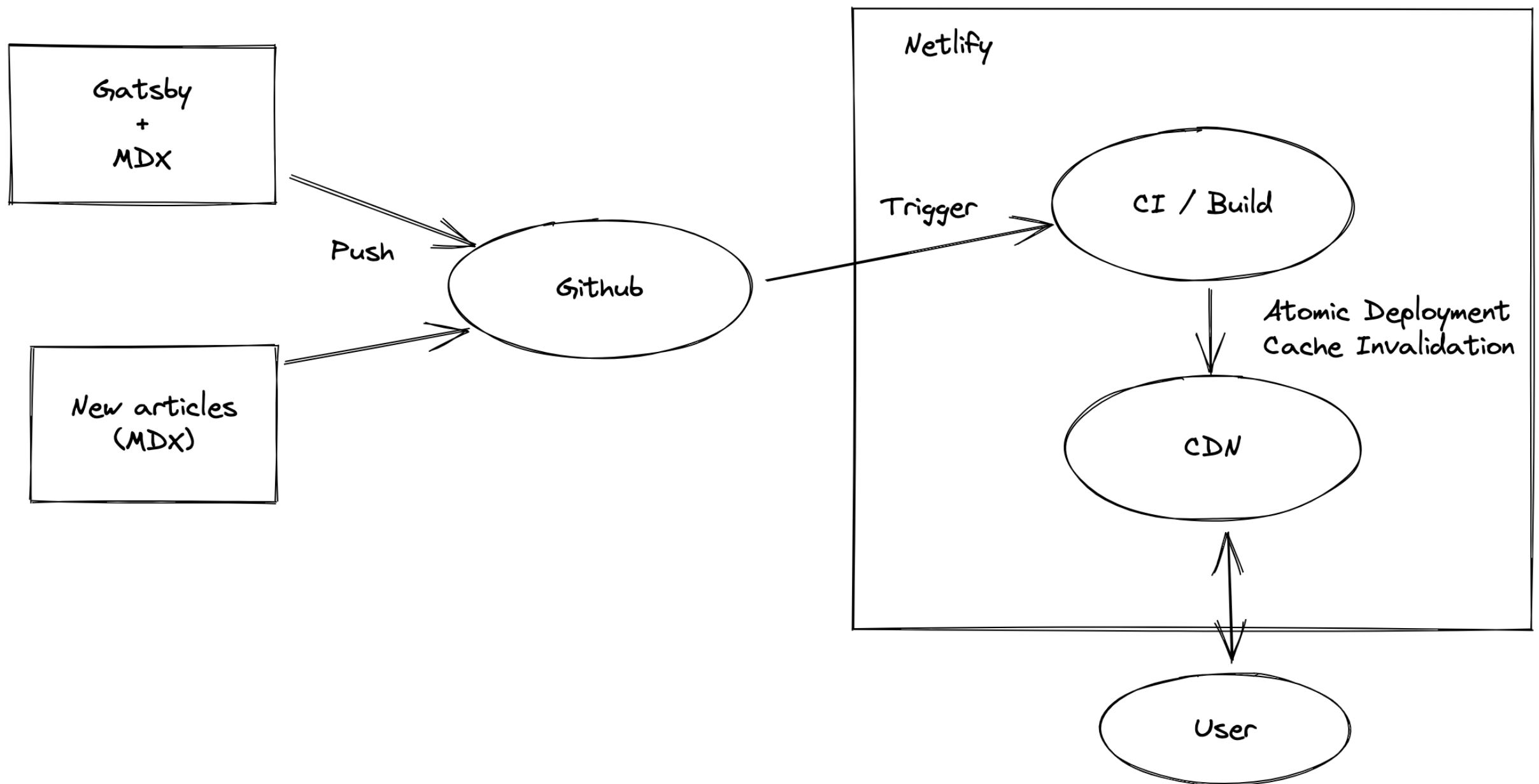
TRADITIONAL WEB VS JAMSTACK

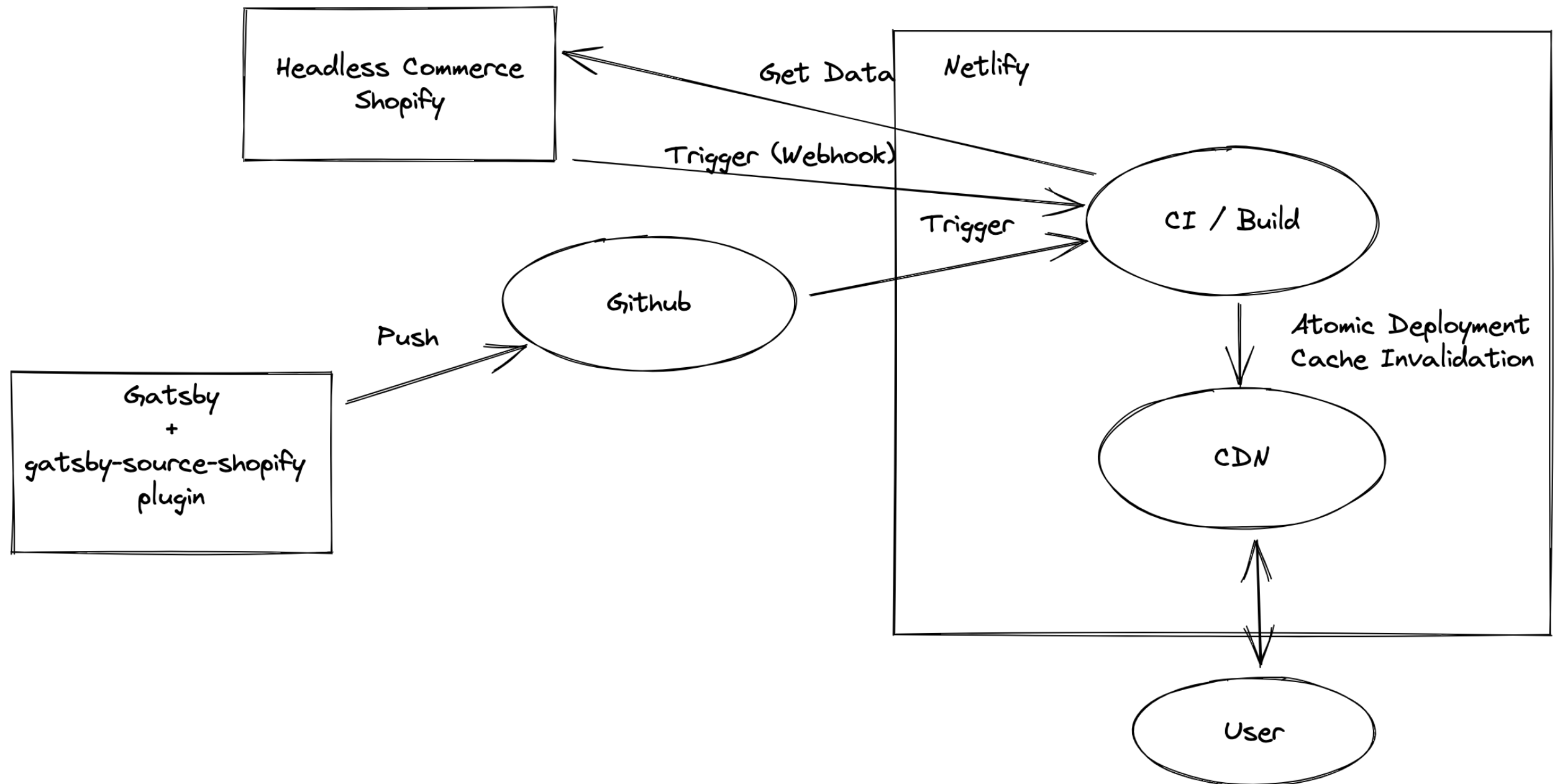# THE FUTURE IS HIGHLY DISTRIBUTED

JAMSTACK ARCHITECTURE

USE CASE: BLOG WEBSITE

USE CASE: BLOG WEBSITE

USE CASE: ECOMMERCE WEBSITE

USE CASE: ECOMMERCE WEBSITE

Headless Commerce
Shopify

Get Data

Trigger (Webhook)

Netlify

CI / Build

Trigger

Github

Push

Gatsby
+
gatsby-source-shopify
plugin

Atomic Deployment
Cache Invalidation

CDN

User

JAMSTACK LIMITATIONS

- ▶ Build time

- ▶ Data freshness

- ▶ Authenticated pages

WHAT'S NEXT?

▶ Incremental builds

▶ Deferred Static Generation (DSG)

▶ Remix

RESOURCES

▸ jamstack.org

▸ gatsbyjs.com/docs/glossary/jamstack

▸ jamstack.wtf

▸ github.com/nicolaserny/talks

Thanks!