The JAM Stack

TONI KARPPI
KRISTIAN ALVAREZ JÖRGENSEN

Javascript

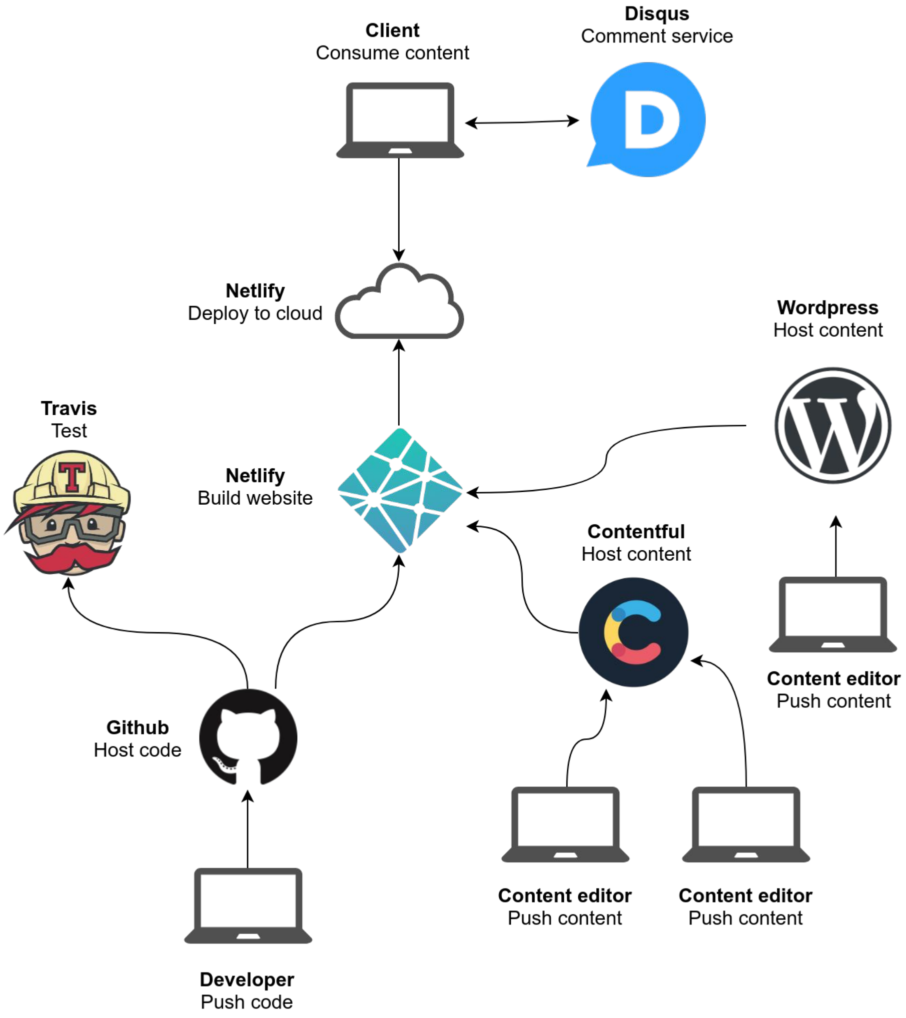
Interactivity provided client side using Javascript.

APIs

Reusable APIs (REST, GraphQL, etc), no direct DB connections. Focus on micro-services.

Markup

The textual content gets turned into HTML during the build.



Developer pushes code to Github.

Travis runs tests, Netlify builds website.

Netlify deploys the website.

Clients connect to the website, and interact with external services if needed.

Site is also rebuilt if a content editor makes a change to the content.

Why JAMStack?

Performance - Served files are prebuilt at deployment time, which leads to very fast load times.

Scaling - Scaling the application just means distributing the files at more places. This is much easier and cheaper than scaling server rendered sites. Integrates well with CDNs.

Security - With server-side processing managed through micro-services, the security of the application can be controlled much more easily.

Separation of concerns - Loose coupling between the application and services allows us to focus on developing features that matter.