San Francisco Search Engine

Schools - Utilizing SF Open Data

Problem

The City of San Francisco has a Schools API they would like to add a frontend to. The API handles the ability to search for keywords or limit results you will be making requests to this API and displaying its results. On top of a frontend, the city would like to save and manage users accessing the site.

Mary

Age Range: 25-35

District: Inner Richmond

Family: Mother of Two

“I would like to know the schools I send my children to are in a safe neighborhood, or at least know the places where they should avoid.”

Suleman

Age Range: 10-15

District: Excelsior

Middle School Student

“One of my classmates got mugged on the way to school and I would know places to avoid so that won’t happen to me either.”

Solution

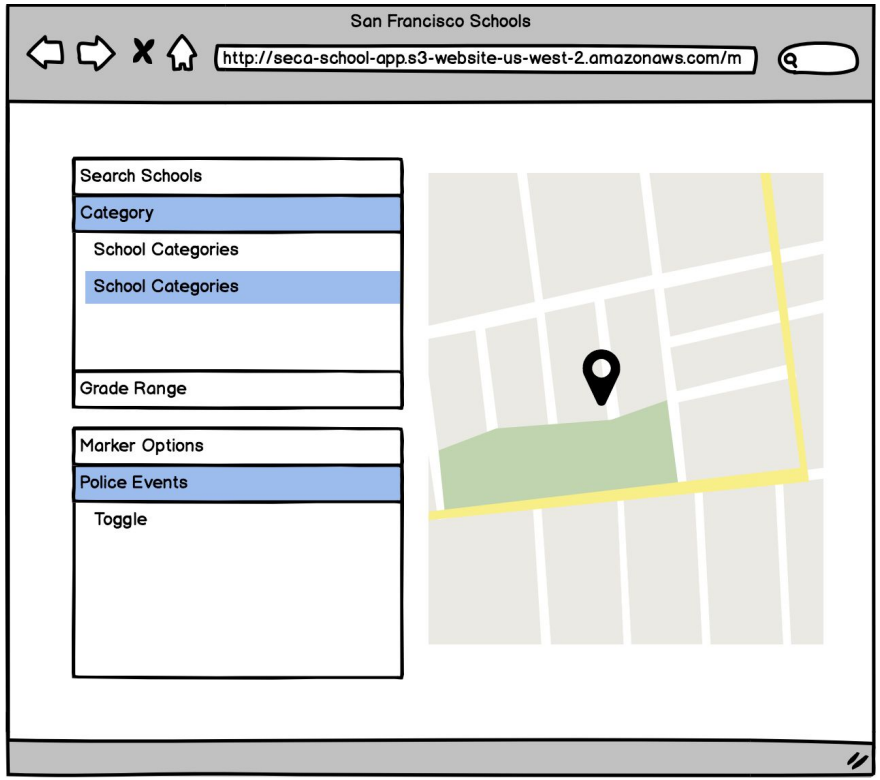
Angular

- Framework for responsive website
- Large ecosystem to pull libraries from (e.g. Angular Maps)

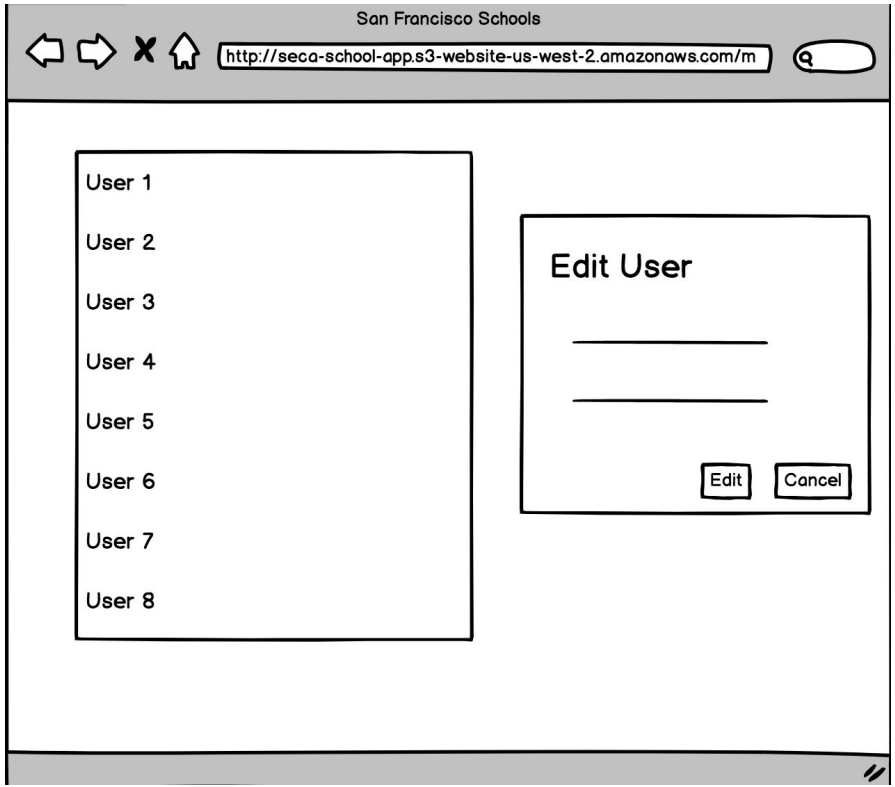
Spring Boot

- Use for the Users API
- Maturity and Security

School Search (Main Feature)



User Management

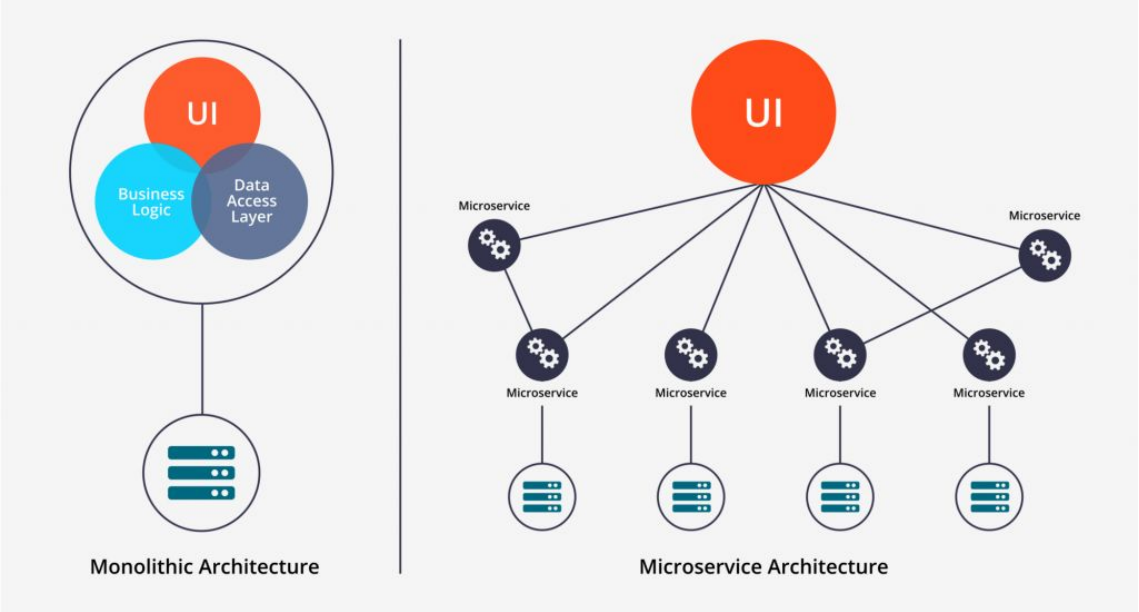


Solution (Cont’d)

Microservices (Backend)

- Isolated Failures
  - Less of a chance to bring the whole system down compared to a monolith

Microservices vs Monolith

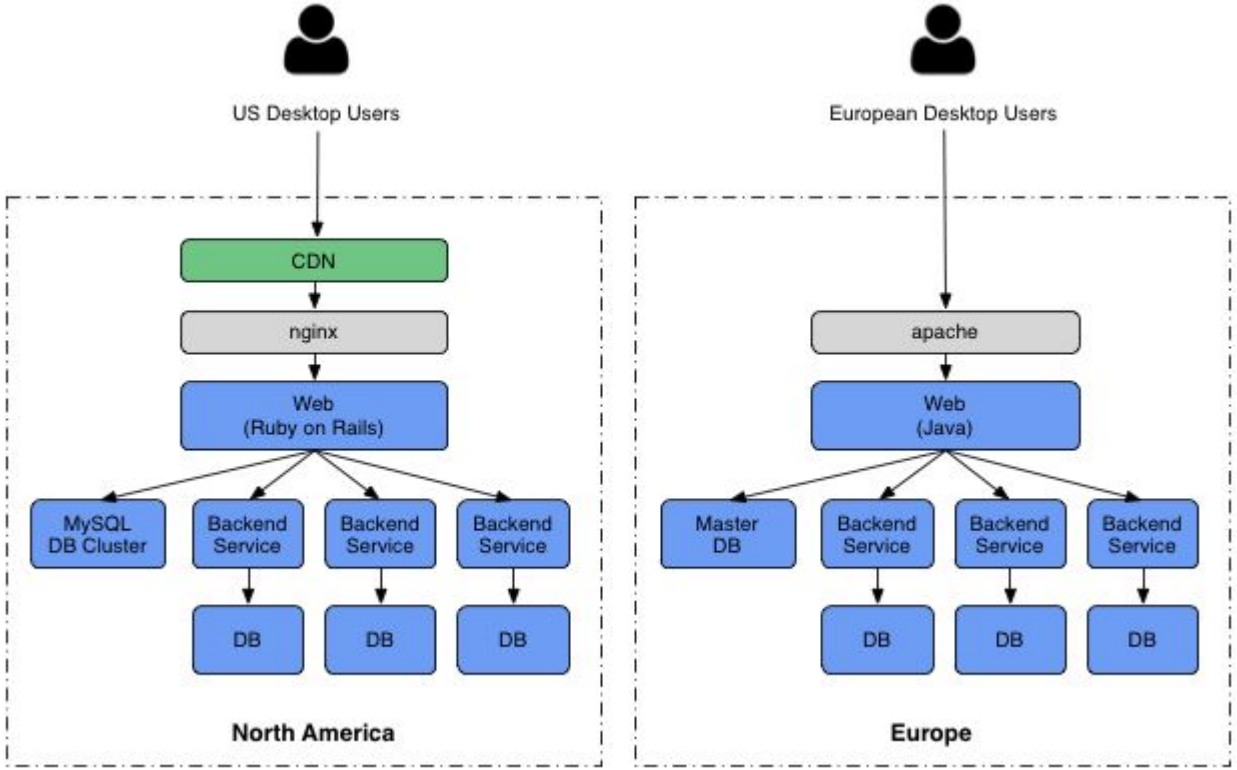


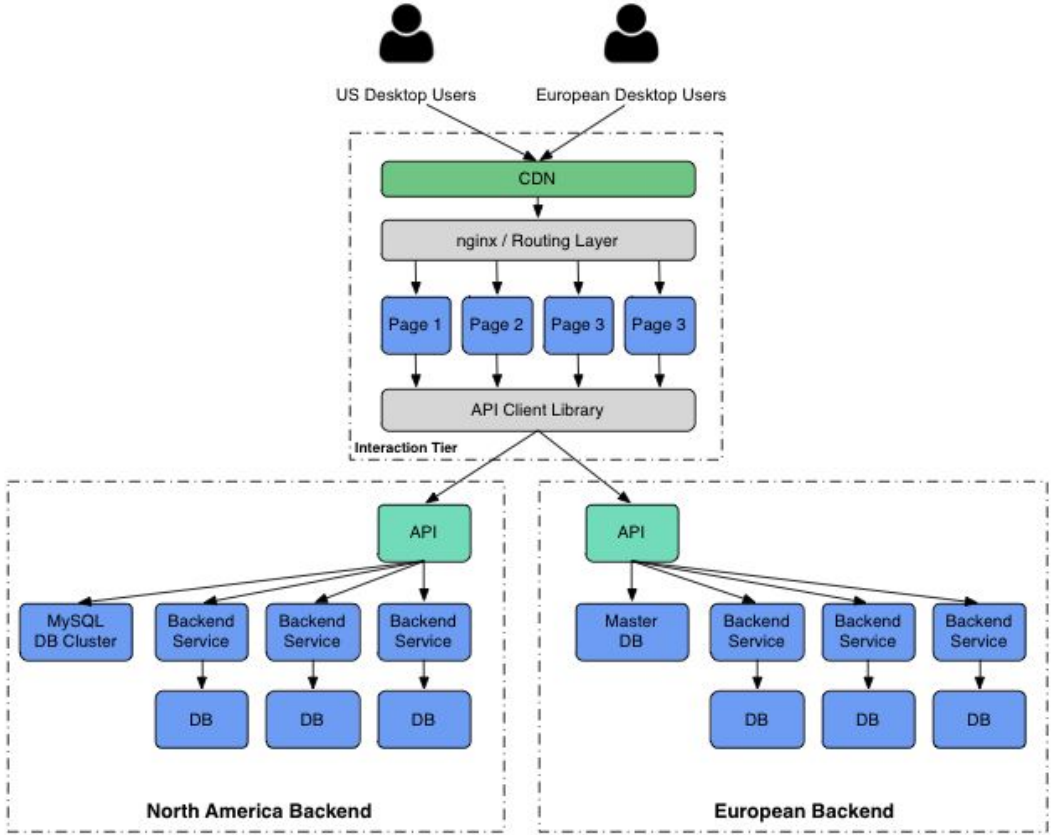
Source: https://www.thesunflowerlab.com/

Microservices Example (Groupon)

- Page loads are faster across the board—typically by 50%. Part of this is due to technology changes and part of this is because we had a chance to rewrite all of our web pages to be much slimmer. And we still expect to make significant gains here as we roll out additional changes.
- We're serving the same amount of traffic with less hardware compared to the old stack.
- Teams are able to deploy changes to their applications independently.
- We've been able to make site-wide feature and design changes much more quickly than we would have been able to with our old architecture.

https://engineering.groupon.com/2013/misc/i-tier-dismantling-the-monoliths/


US Desktop Users
European Desktop Users
CDN
nginx
Web
(Ruby on Rails)
MySQL
DB Cluster
Backend
Service
Backend
Service
Backend
Service
DB
DB
DB
North America
apache
Web
(Java)
Master
DB
Backend
Service
Backend
Service
Backend
Service
DB
DB
DB
Europe


US Desktop Users
European Desktop Users
CDN
nginx / Routing Layer
Page 1
Page 2
Page 3
Page 3
API Client Library
Interaction Tier
API
MySQL DB Cluster
Backend Service
Backend Service
Backend Service
DB
DB
DB
North America Backend
API
Master DB
Backend Service
Backend Service
Backend Service
DB
DB
DB
European Backend