Building a cloud of fares

Ralph Ligtenberg
February 21, 2019

Who am I?

Ralph Ligtenberg
Tech Lead @ Travix
.Net/C#, Go, Google Cloud Platform

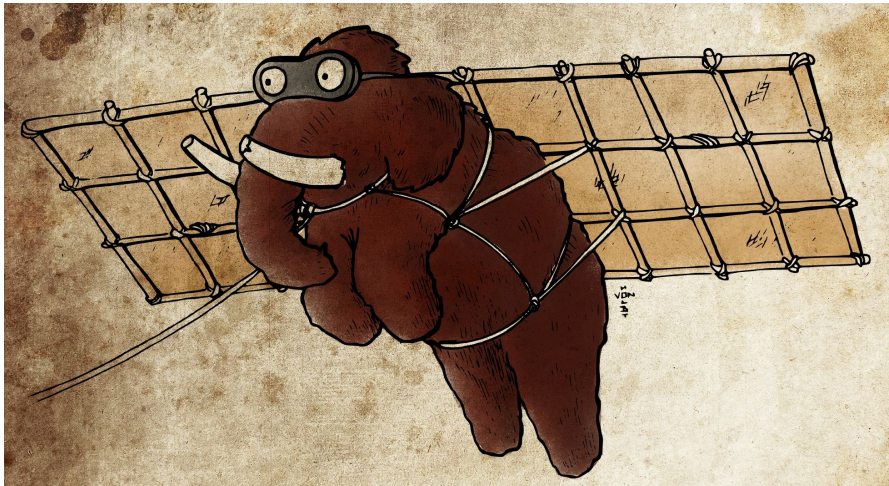
twitter.com/prutswonder
travix.io/@prutswonder
linkedin.com/in/ralphligtenberg

Travix

Travix International

History of the “Fare Cache”

- Key principle: Store all search results ⟶ query ⟶ current cheapest price
- One product, multiple use cases
- Cassandra & .Net
- Released beginning of 2015



Good, but not future-proof

- Scalable, though limited & expensive
  - Micro-monolith
  - Ran on a single machine!
- Cassandra requires advanced Ops knowledge
  - Not maintained by Cassandra expert
  - Tombstoning
  - Replication factor vs cluster size

The cloud idea

- “One Cache To Rule Them All”
- Move data to Google Cloud
- Build horizontally
- Integrate with “old” solution
- New solution saves money!

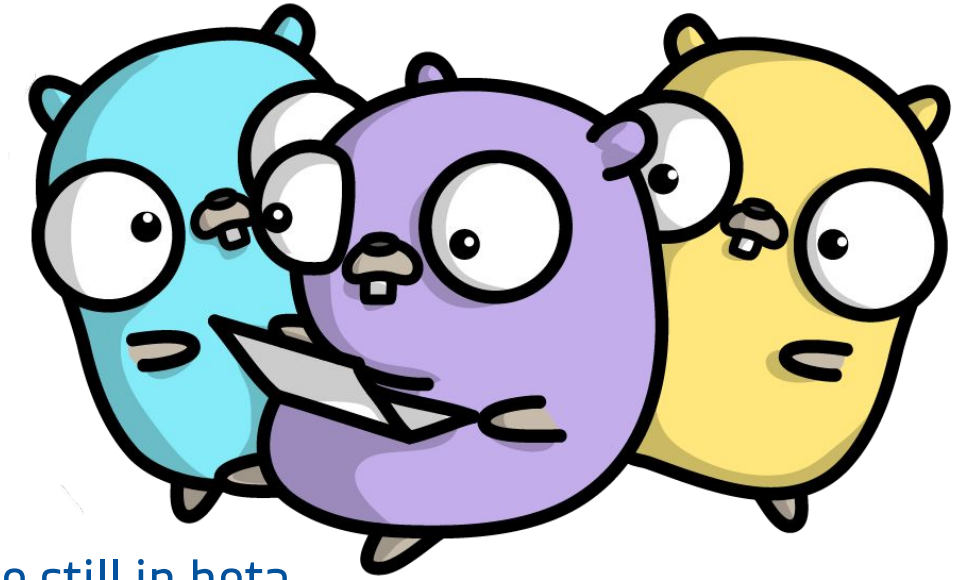
Travix

Enter “Fare Cloud”

- BigTable, Datastore & Dataflow
- Micro-services in Go
- Only specific use-cases
- Implementation: 6 months

Travix

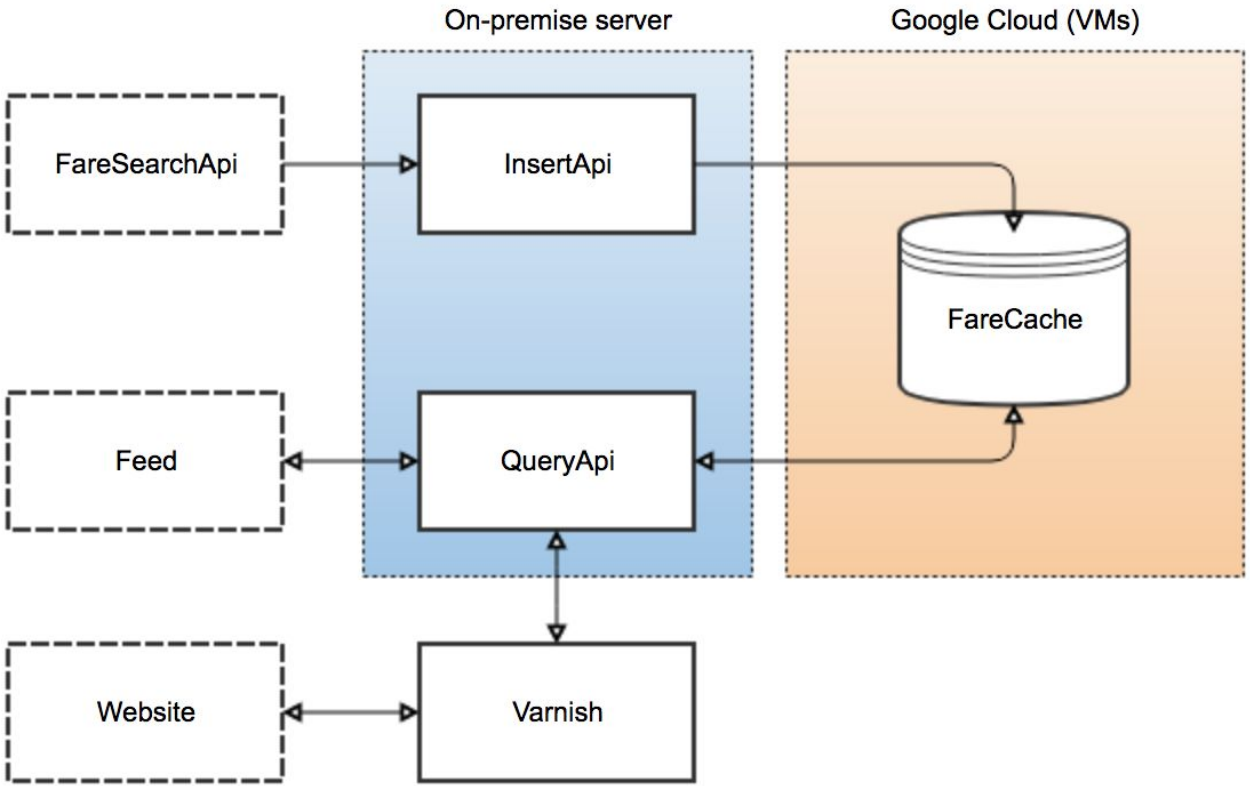
Why use “Go” instead of .Net Core?



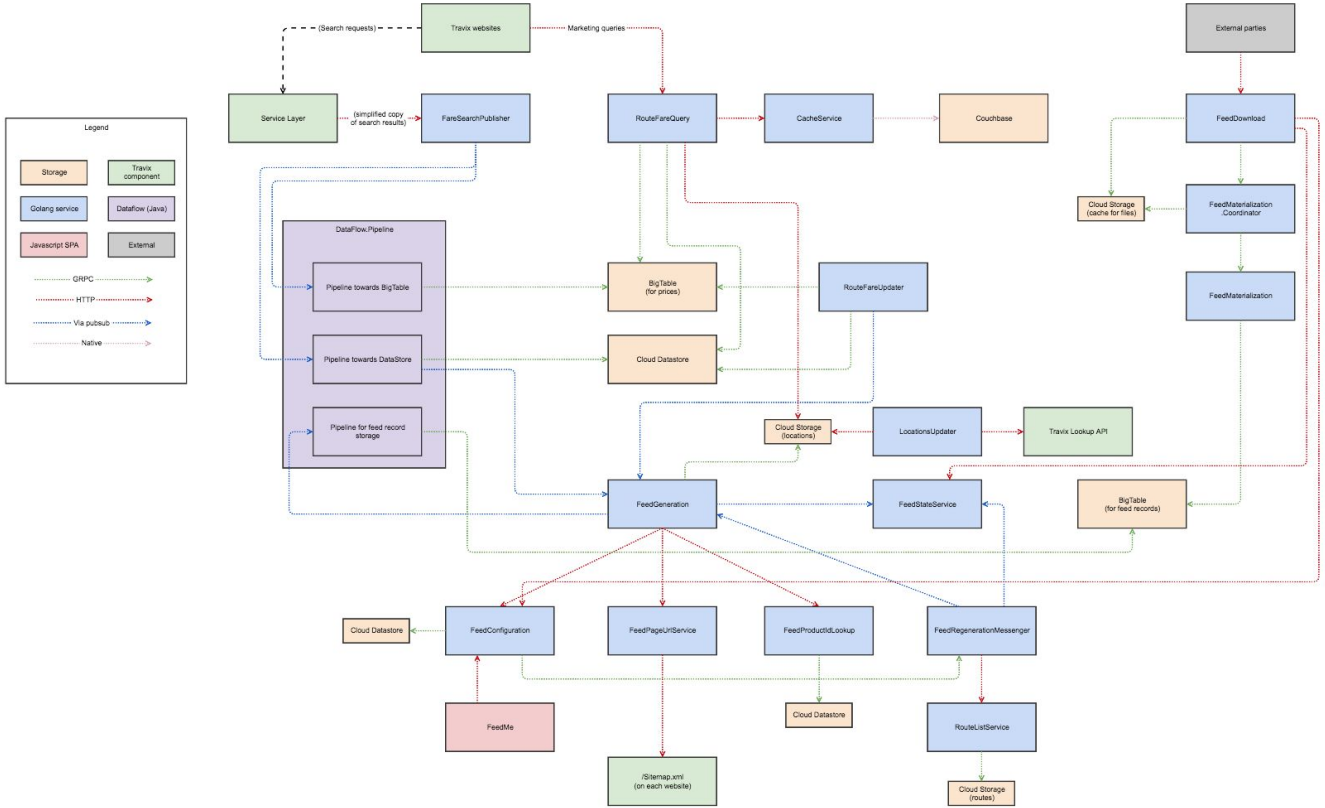
- .Net Core v1.0 just released
- Most Google Cloud drivers were still in beta
- Easy to learn (hard to master)
- PoC was more performant than expected

Travix

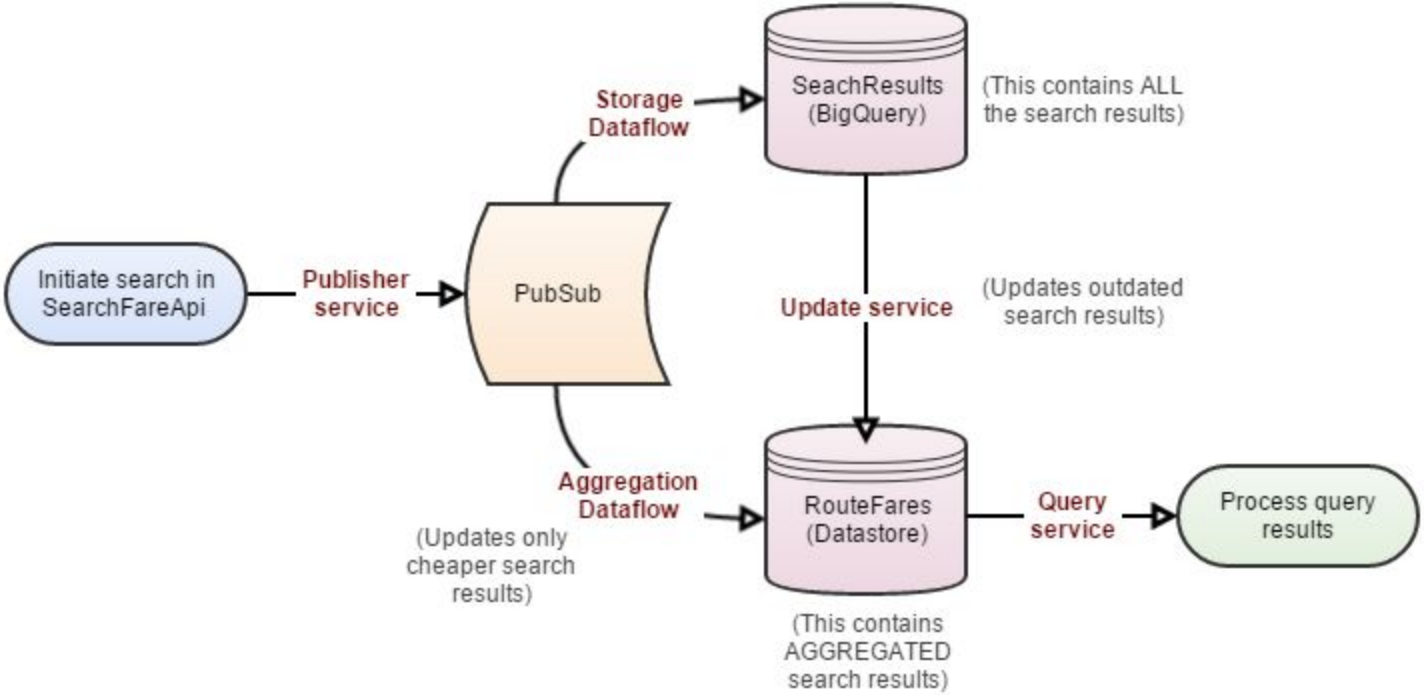
Old FareCache infrastructure



New FareCloud architecture



The “FareCache” bit




Thank you!