

Microservices in GO with **GRPC & Protocol buffers**

There and back again

Boyan Bonev @ Gaida.AI



GAIDA.AI

Who dis???

- A software guy
- Happy Gopher for slightly over 3 years
- Currently with a startup, so, legit here for the free pizza



Today

- Talking microservices
- Learning about GRPC and protobufs
- Experience report from the trenches
- Q&A



To microservice or not to microservice?

A screenshot of a search results page from a search engine. The search bar at the top contains the query "microservices". Below the search bar are navigation links for "All", "Images", "Videos", "News", "Maps", "More", "Settings", and "Tools". A message indicates "About 14,500,000 results (0.41 seconds)". The first result is a snippet titled "What are microservices?" with a link to <https://microservices.io/>. The snippet describes microservices as an architectural style. Below this is a "People also search for" section with links to related terms like "microservices vs api", "microservices spring boot", "microservices c#", "microservices book", "microservices vs soa", and "microservices interview questions".

microservices

All Images Videos News Maps More Settings Tools

About 14,500,000 results (0.41 seconds)

What are microservices?
<https://microservices.io/>

Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of loosely coupled services, ...

People also search for

microservices vs api microservices spring boot

microservices c# microservices book

microservices vs soa microservices interview questions



The options behind^w ahead of us

- Build a monolith
- Build microservices
- Build a modular monolith! (only if you are Simon Brown)



Of course I'm building microservices!

“ *The single biggest problem in communication is the illusion that it has taken place.*

Bernard Show

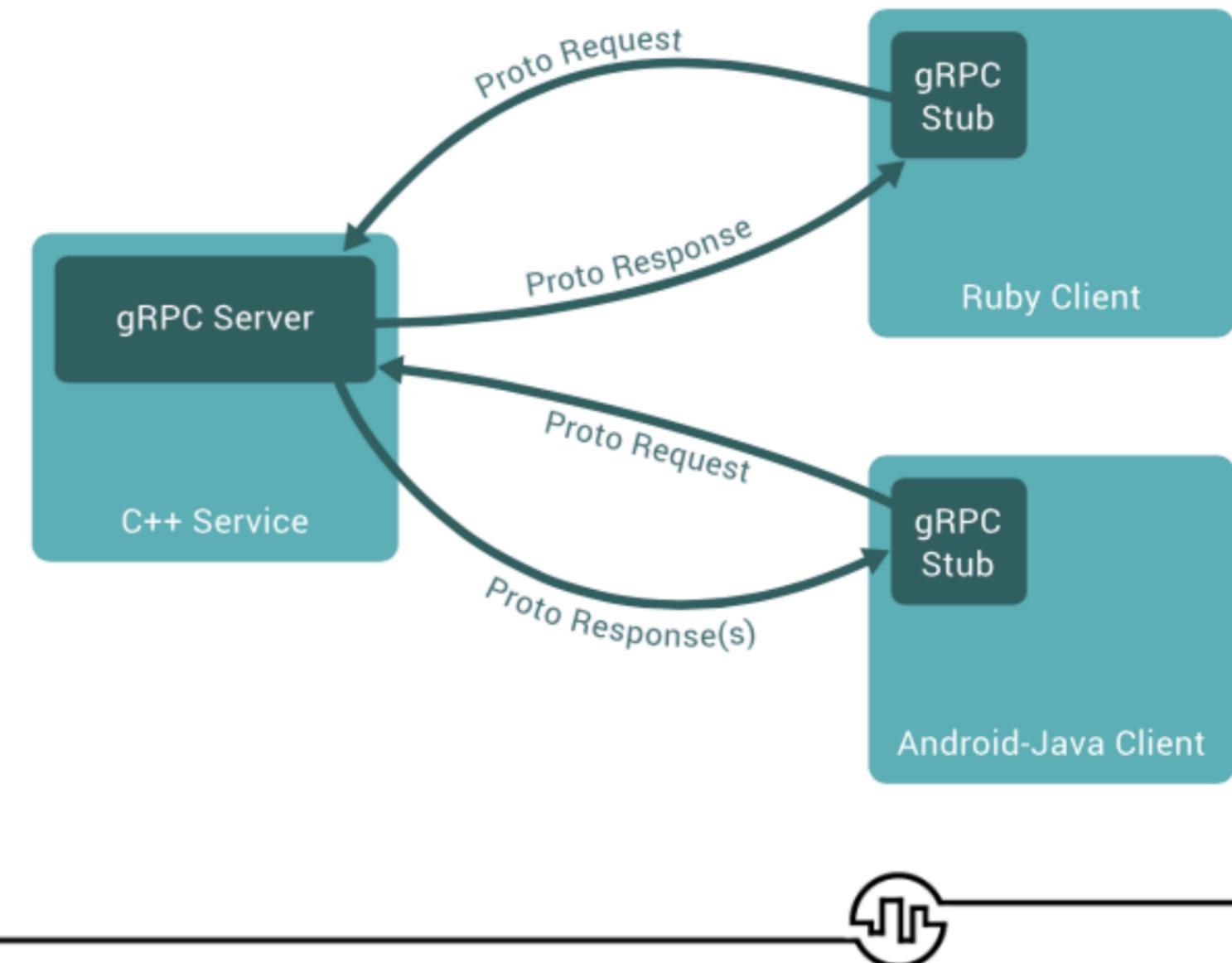
“ *500 Internal Server Error*

The Internet at Large



Et tu, GRPC?

- A framework for fast & robust development of inter-communicating implementation agnostic units
- Excellent performance
- Excellent development speed
- Encourages good practices



Serialize killers

Plain text

- Human readable
- Schema-less
- Ease of ops
- Even Aunt May knows what JSON is

Protocol buffers

- Schema
- Backward compatible
- Language agnostic implementation
- Performance



So, how does it work in GO?



- Excellent support for GRPC
- Excellent support for Protocol buffers
- Excellent tooling, third party libraries etc
- Community that has embraced the technology



Experience report

The state

- We are starting. Code at day 1 = None
- Have an idea on where we are going...
- ...don't **want** to know how we'll get there
- Need to go fast
- Ops isn't scary



Decisions, decisions...

Mindset

- Simple is better than smart
- Confidence is important
- Machines <-> machines
- Open minded, but not adventurous

Explorations

- Sensible microservice units
- Go-kit, micro, bare
- Is production scary?
- Implications of GRPC



The toolbox

- Make
- Gitlab CI
- Docker, Ansible
- Traefik, Graylog, Prometheus, Grafana
- Go, Clojure, Python



This slide with code

```
syntax = "proto3";

package pb;

import "google/protobuf/timestamp.proto";

service Recommendations {
    rpc Get(GetRequest) returns (GetResponse);
    rpc GetList(GetListRequest) returns (GetListResponse);
    rpc Create(CreateRequest) returns (CreateResponse);
    rpc Update(UpdateRequest) returns (UpdateResponse);
}

enum RecommendationStatus {
    pending = 0;
    sent = 1;
    failed = 2;
    delivered = 3;
}
```

```
λ ➤ ~/Development/src/gaida/recommendation-service ➤ master ➤ make proto-build
protoc -I /usr/local/include -I pb/ pb/recommendation.proto --go_out=plugins=grpc:pb
λ ➤ ~/Development/src/gaida/recommendation-service ➤ master ➤
```

```
if tlsCreds != nil {
    grpcConnection, err = grpc.Dial(viper.GetString("recommendation-service.uri"), grpc.WithTransportCredentials(tlsCreds.(credentials.TransportCredentials)))
} else {
    grpcConnection, err = grpc.Dial(viper.GetString("recommendation-service.uri"), grpc.WithInsecure())
}
grpcClient := pb.NewRecommendationsClient(grpcConnection)
```



Lessons learned

- Domain knowledge is a more serious problem than technical ability
- Don't shoot for getting it right the very first time, shoot for getting it working
- GRPC is good and will scale



Thank you!

