The Story of why we migrate to gRPC and how we go about it

KubeCon

CloudNativeCon

Europe 2019

@mattgruter

2019-05-22

“Developers
don’t care
about new
RPC technologies”

-- someone at KubeCon 2018

Why?

Why are we doing this?

How?

How do we get there?

Why?

Spotify's Infrastructure

200M active monthly users

8M QPS at ingress

1k developers

2.5k services

Java, Python, ...

Hermes

Not this Hermes!

Hermes

But this

Hermes

- Written in 2012
- Based on ZeroMQ
- JSON or Protobuf payload
- Not a RPC framework

Hermes works!

Why Change?

Hermes Ecosystem

404 Not Found

From DIY to FOSS

Why gRPC?

The Proto

Code Generation

Java, Golang, Python, Ruby,
Dart, PHP, Node.js,
Objective-C, C#, C++

Schema Management

1. Embrace the **proto**

Schema Management

2. Shared repo for all **protos**

Schema Management

3. Version on the **proto** package

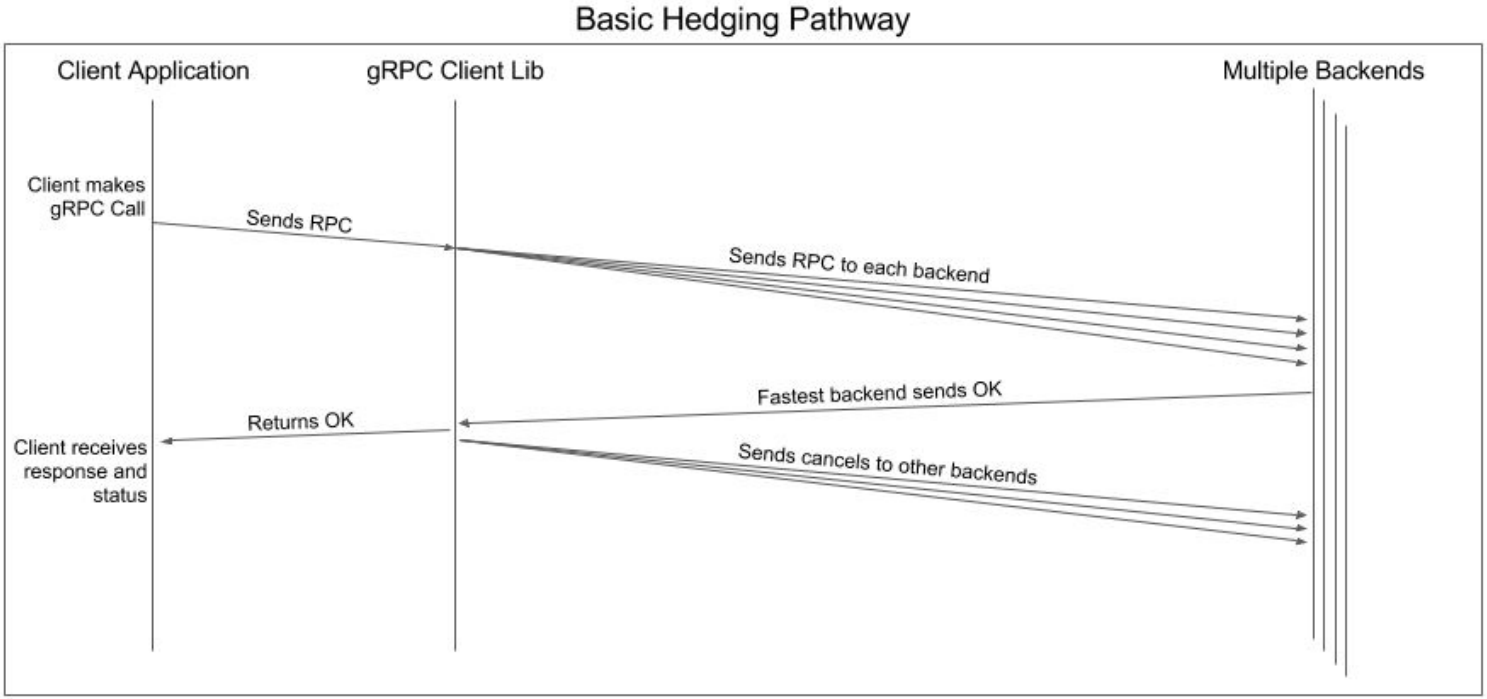
```
package spotify.playlist.v1;
package spotify.playlist.v2beta1;
```

Resiliency

Resiliency:
Retries

Proposal

Retry Hedging



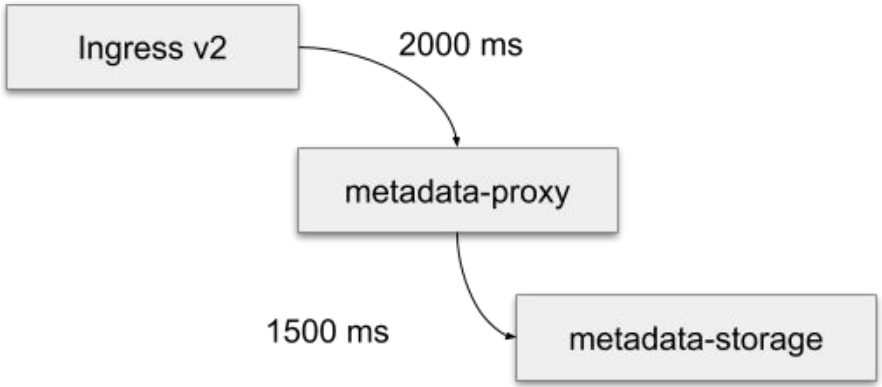
Thundering Herd

Proposal

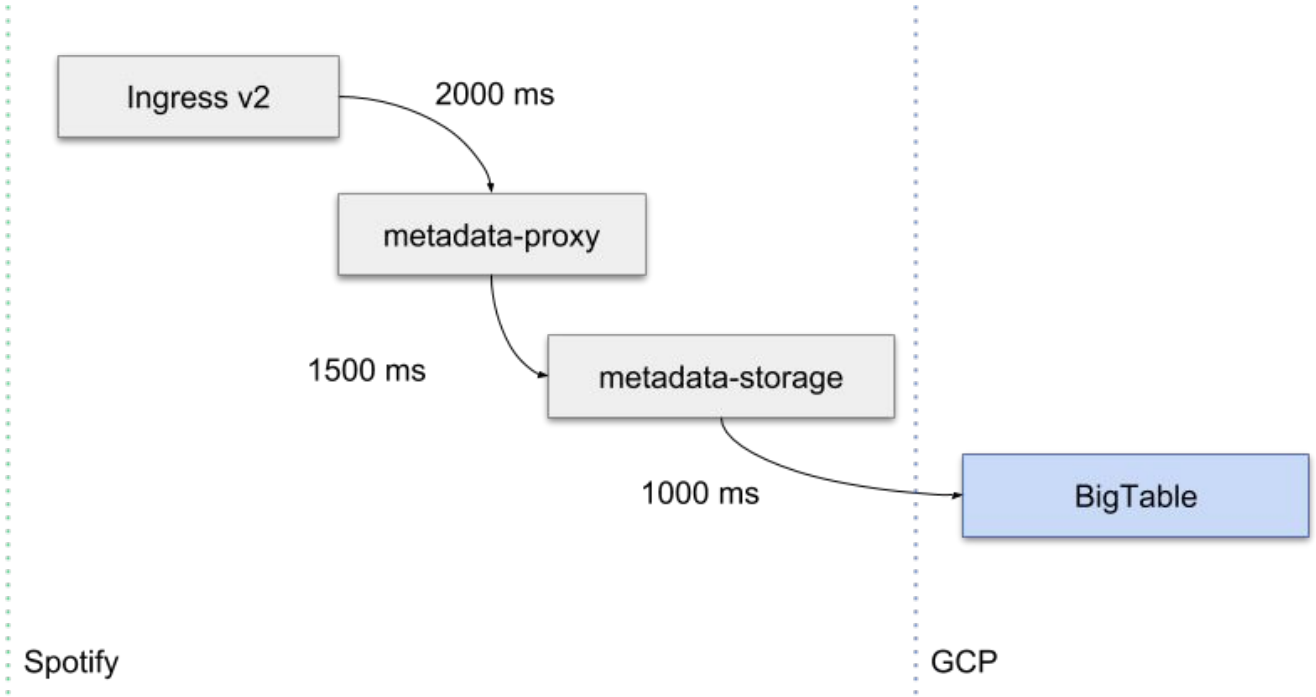
Resiliency:
Retry Throttling & Pushback

Resiliency:
Deadlines

Deadlines



Deadlines


Ingress v2
2000 ms
metadata-proxy
1500 ms
metadata-storage
1000 ms
BigTable
Spotify
GCP

Load Balancing & Routing

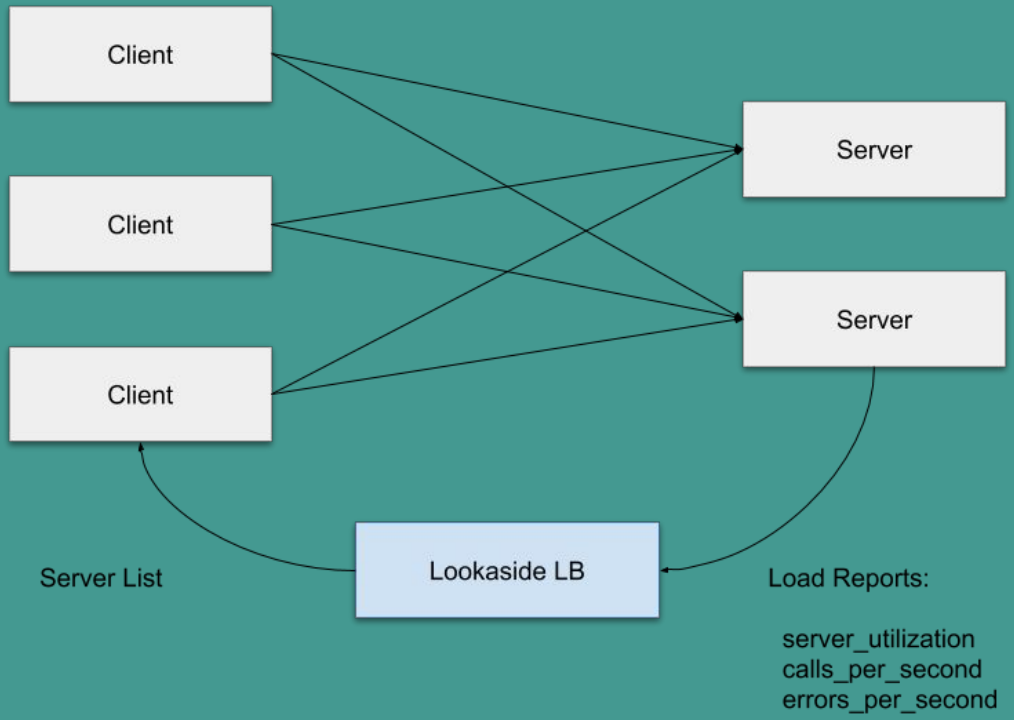
Load Balancing

Server-side	Client-side
proxy	no proxy
dumb client	complex client
extra hop	low latency

Load Balancing

Server-side	Client-side	Lookaside
proxy	no proxy	no proxy
dumb client	complex client	thinner client
extra hop	low latency	low latency

Lookaside Load Balancing



Proxyless RPC mesh

Why gRPC?

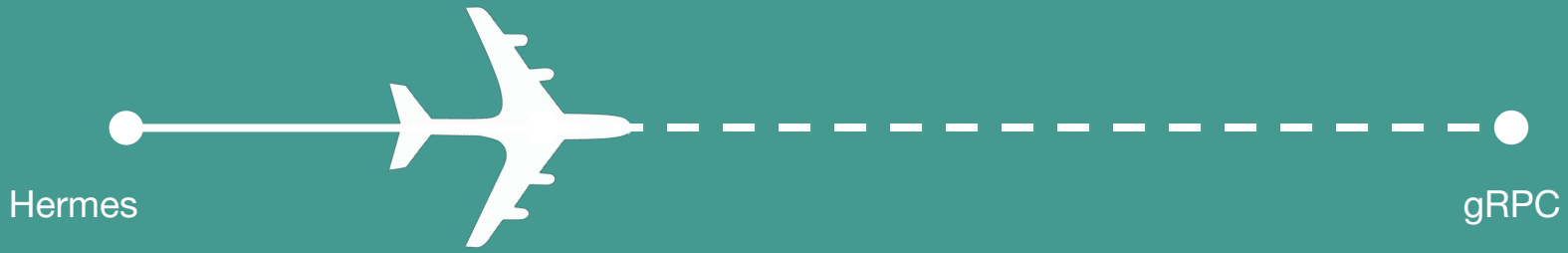
gRPC

TO PROTOC & SERVE

Spotify
gRPC

How?

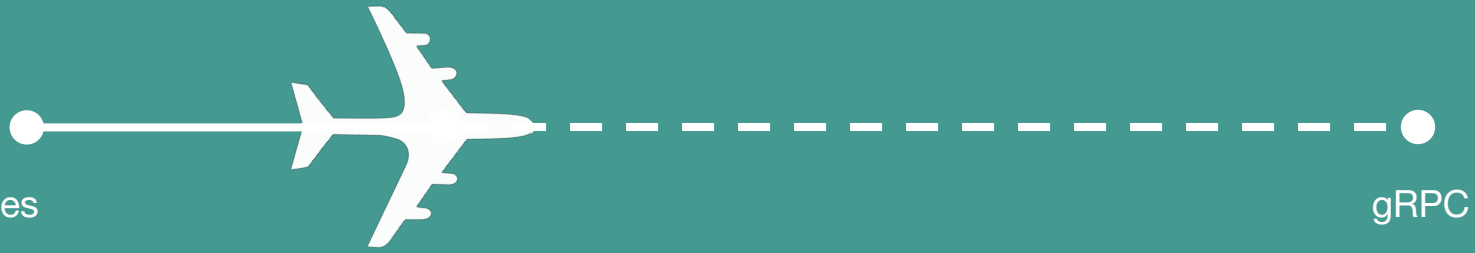
Our Journey



Our Journey

10%
Distance Travelled

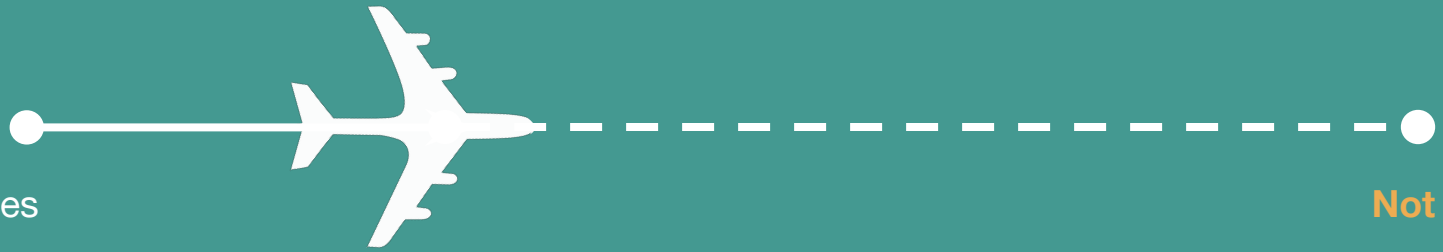
90%
Remaining Distance to Destination



Our Journey

10%
Distance Travelled

90%
Remaining Distance to Destination



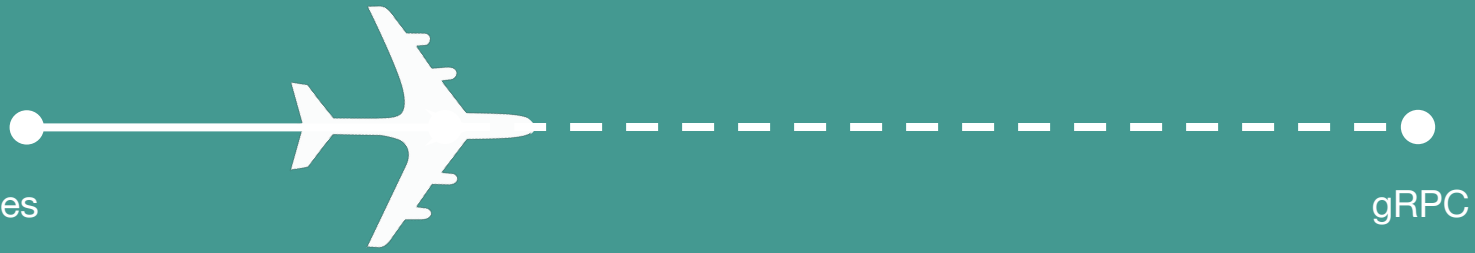
Hermes

Not Hermes

Our Journey

10%
Distance Travelled

90%
Remaining Distance to Destination



Main Challenge

Change is hard

“This is not the protocol you’re looking for”

Main Challenge

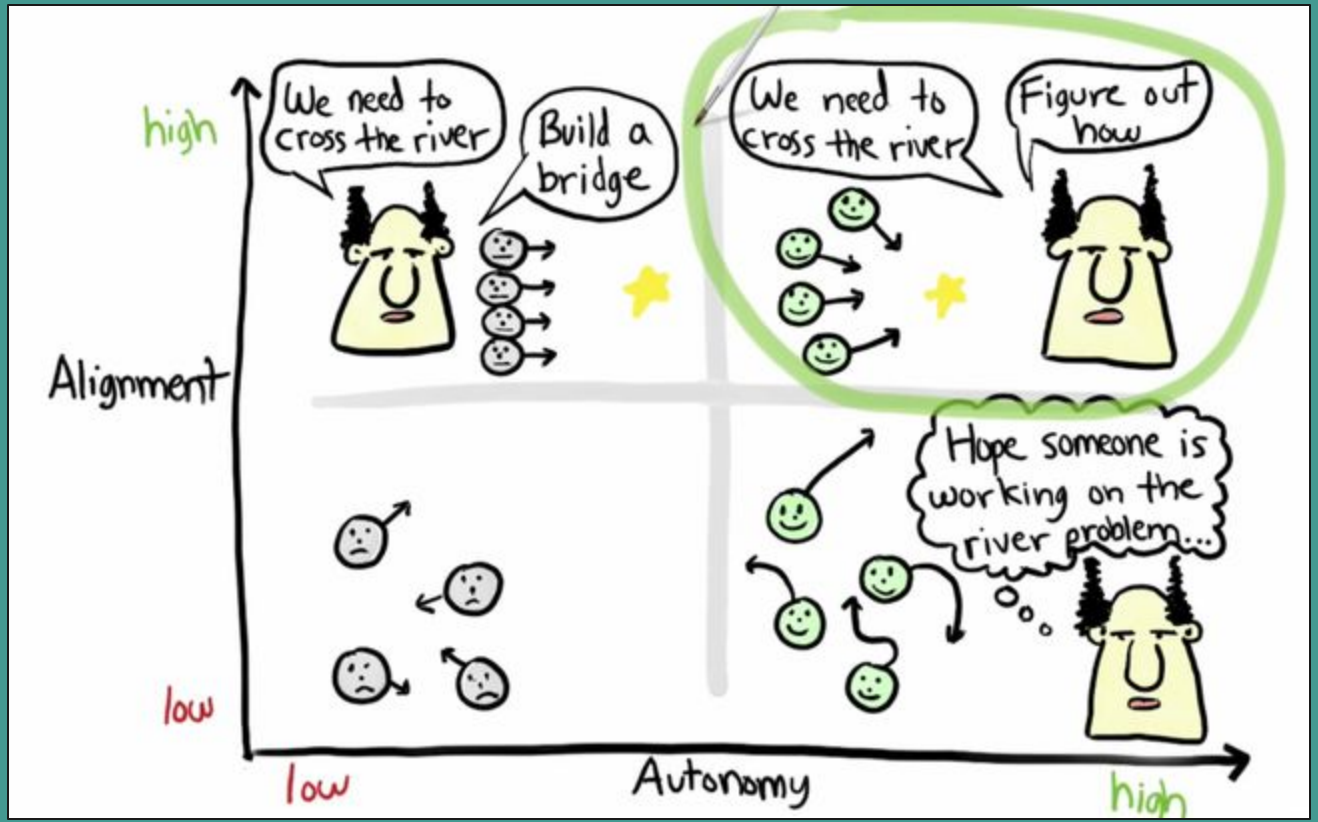
Change is hard

Autonomous Teams

Trust

Autonomous Teams

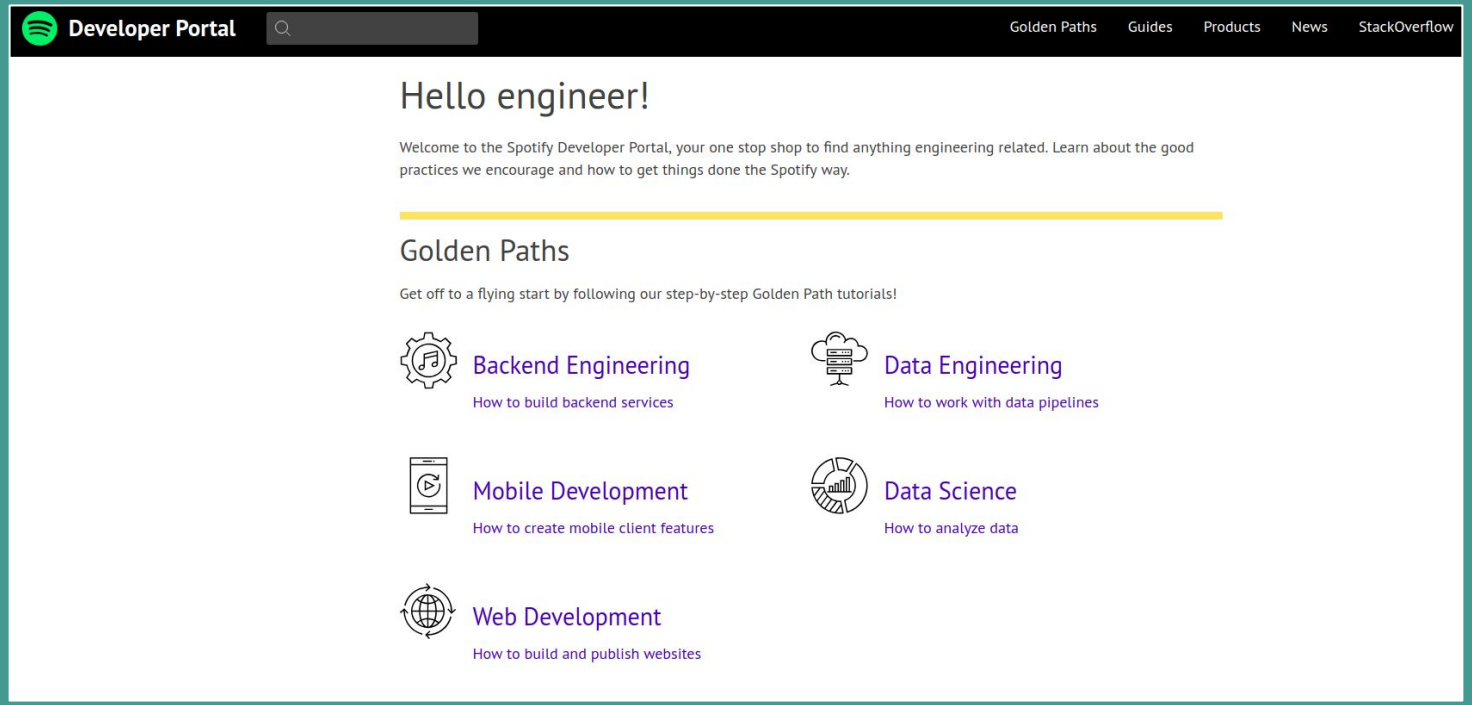
Autonomy != Freedom



Henrik Kniberg / Spotify—Aligned Autonomy

Aligned Autonomy

Spotify's Golden Path



Align by making

the **right** choice

the **easy** choice.

2. Don’t force it!

2. Don’t force it!

Code generation

2. Don’t force it!

Resiliency patterns

Code generation

2. Don’t force it!

Tracing

Resiliency patterns

Code generation

2. Don’t force it!

Tracing

Resiliency patterns

<Fancy Tool X>

Code generation

In case of emergency

aka. never-ending migration
aka. yet another protocol

gRPC

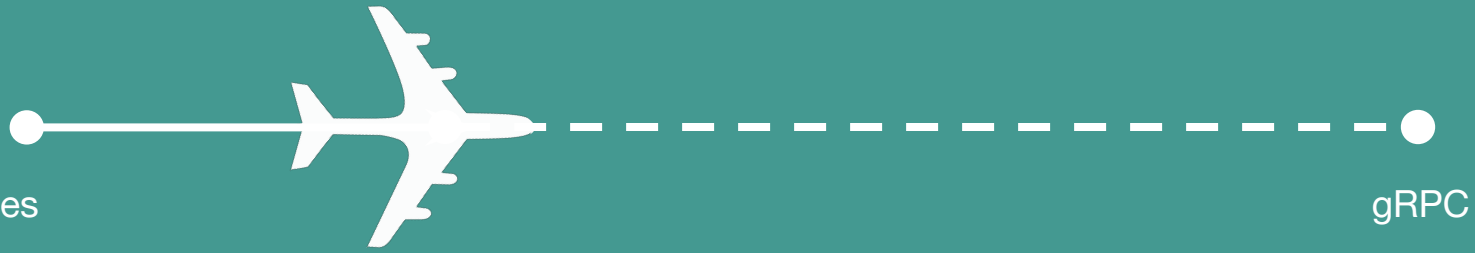
3. Force it!

```
$ sudo migrate
```

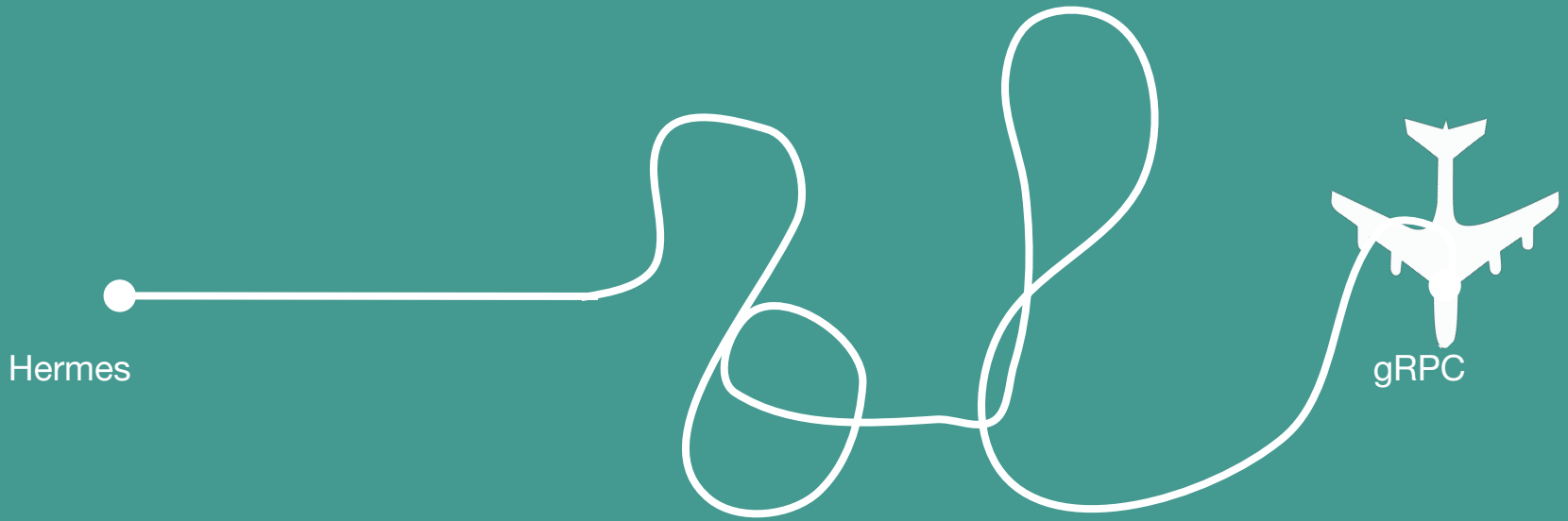
Our Journey

10%
Distance Travelled

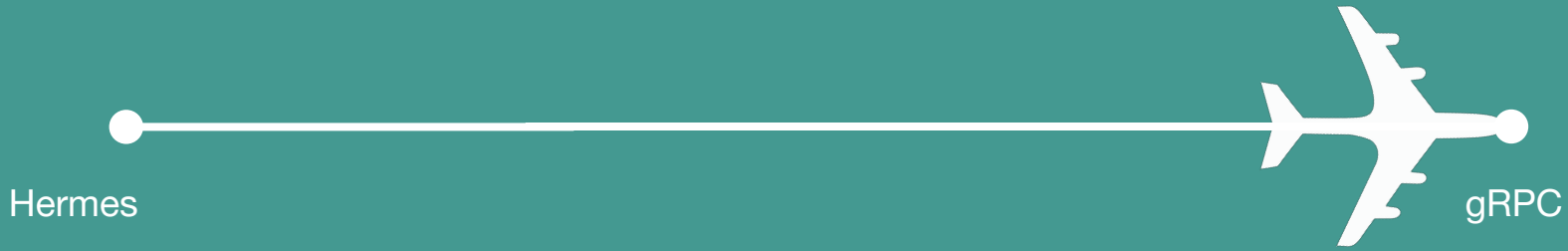
90%
Remaining Distance to Destination



Our Journey



Our Journey



“Developers don’t care about new RPC technologies”

-- overheard at KubeCon 2018 Seattle

“Developers don’t have to care about new RPC technologies”

-- “overheard” at KubeCon 2019 Barcelona

Thank You

Matthias Grüter
@mattgruter