

{ REST }

VS

↑ GRPC ↓

Battle of API's

Who am I?

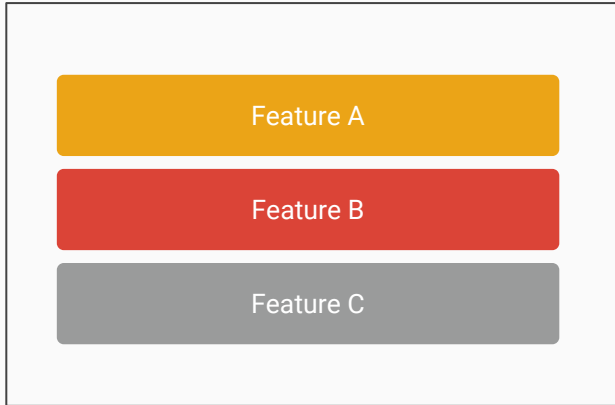


- Software Engineer at Sensedia
- MBA in java projects
- Java and microservice enthusiastic

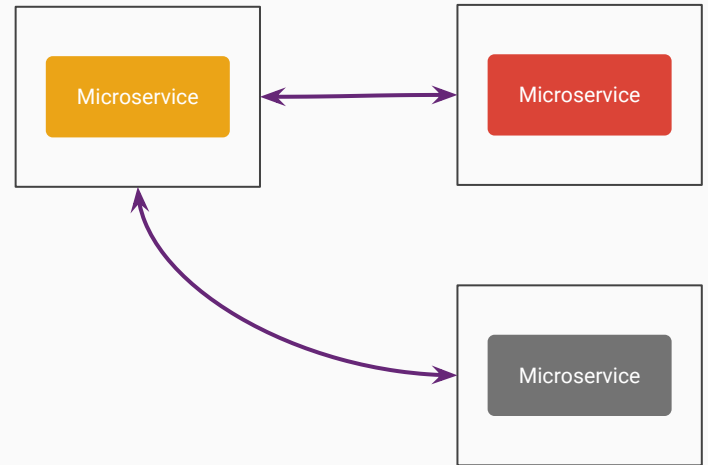
Agenda

- Microservices
- REST
- gRPC
- Demo
- Questions

Monolith

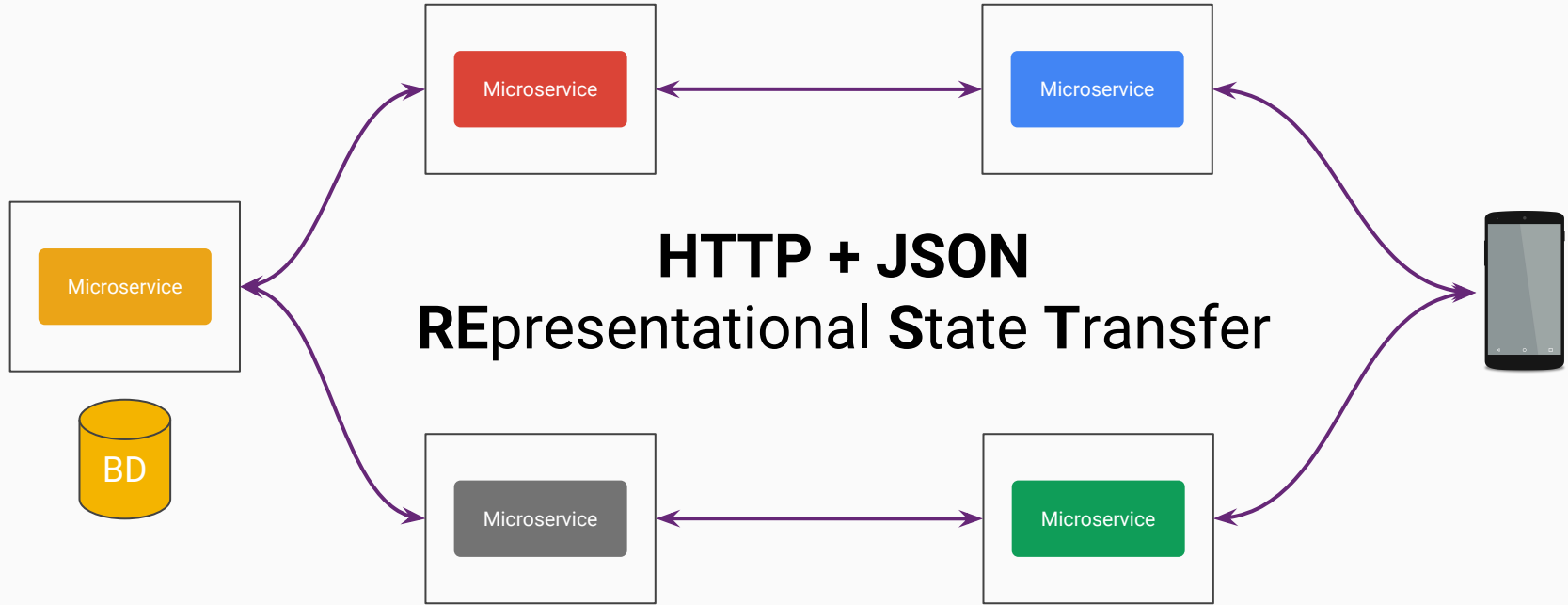


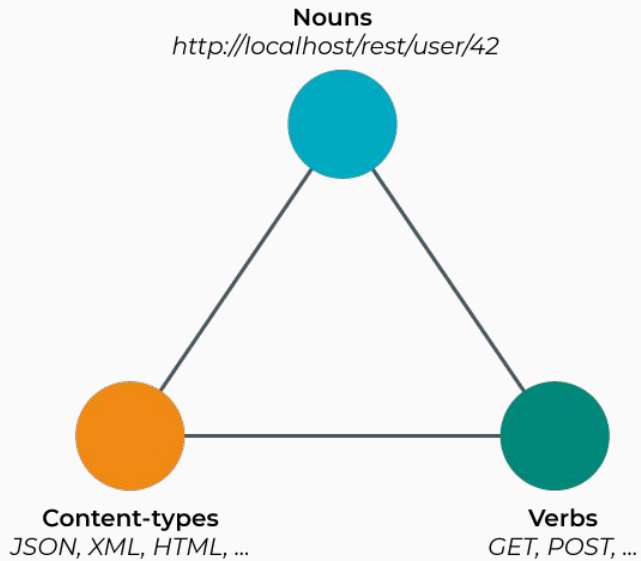
Microservices



Challenges with Microservices

- Communication
- Security / Authentication
- Network Communication Speed
- Language Interoperability





GET

/movies

GET

/movies/{movieid}

POST

/movies

PUT

/movies/{movieid}

DELETE

/movies/{movieid}

Representational State Transfer

- **Good things**
- **Bad things**

- Web already built on top of HTTP
- Easy to understand
- Variety of http implementation in any languages
- Loose coupling between client and server

Representational State Transfer

- Good things
- **Bad things**

- Operations are difficult to model
- Streaming is difficult to implement
- Bi-directional streaming is not possible
- Inefficient, textual representation are not optimal for networks

gRPC is going to fix all of those problems?

Yes!!!

What is RPC and gRPC?

gRPC

- A high-performance open-source universal RPC framework
- Based on “Stubby” (Google’s internal RPC system)
- Part of Cloud Native Computing Foundation

So what makes gRPC so effective?

Protocol Buffers

Protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data.

- Strongly typed
- Rules for making backwards compatible changes
- Efficient binary data representation for network transmission
- Comprehensive style guide

Interface Definition Language

The API definition and structure of the payload messages.



```
syntax = "proto3";
```

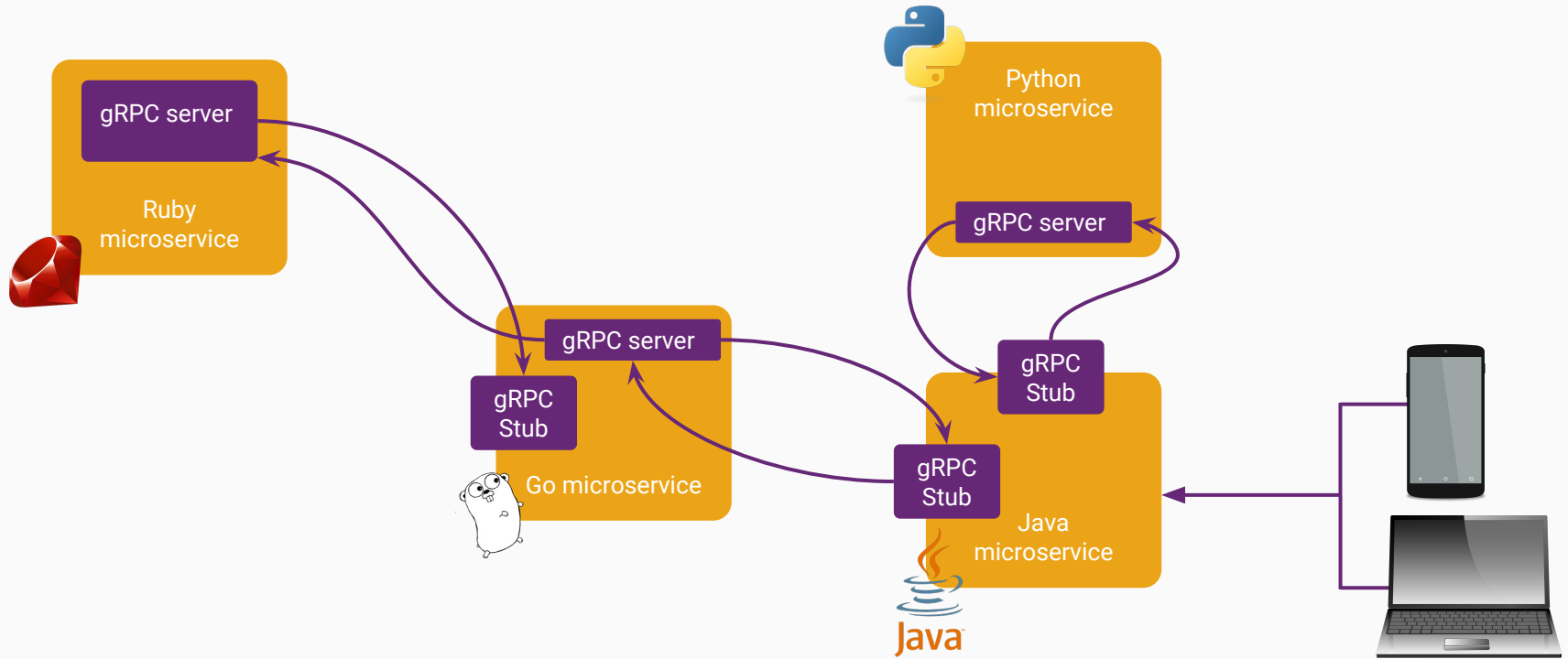
```
message PersonRequest {  
    string name = 1;  
    int32 age = 2;  
}
```

```
message PersonResponse {  
    int32 id = 1;  
    string name = 2;  
    int32 age = 3;  
}
```

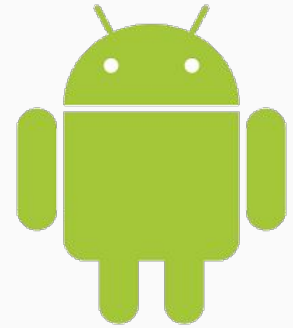
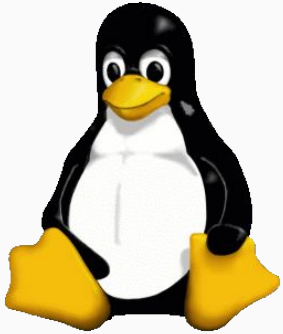
```
service PersonService {  
    rpc create(PersonRequest) returns (PersonResponse);  
}
```


Define once, generates
well-structured code for all
supported language!

Multiple Language Support

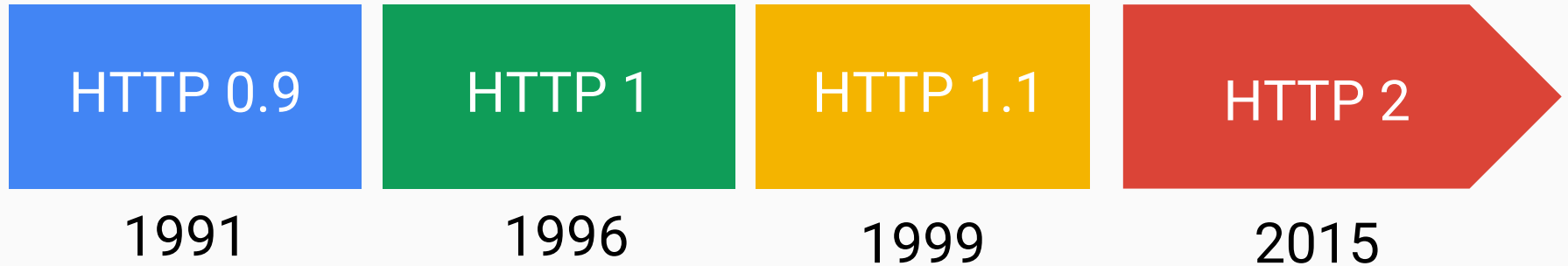


Multiple Platform Support



Works over HTTP 2

16 years without evolution or improvement



HTTP 2

HTTP 2

- Multiplexing
- Bidirectional Streaming
- HTTPS
- Performance



HTTP 1.x

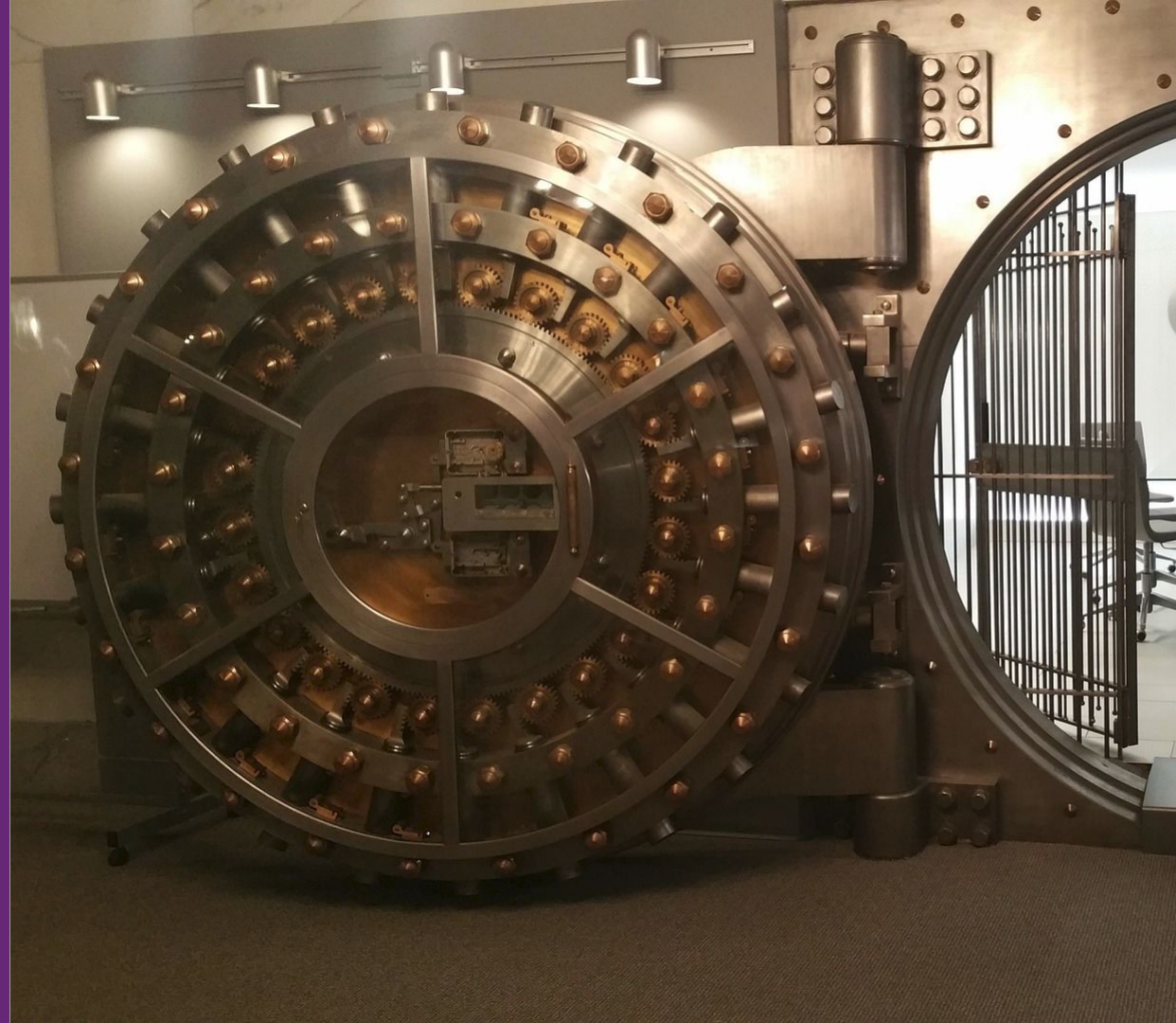
HTTP 2

- Multiplexing
- **Bidirectional Streaming**
- HTTPS
- Performance



HTTP 2

- Multiplexing
- Bidirectional Streaming
- **HTTPS**
 - **SSL/TLS**
 - **Token authentication**
 - **Channel credentials**
 - **Call credentials**
- Performance



HTTP 2

- Multiplexing
- Bidirectional Streaming
- HTTPS
- **Performance**

HTTP 1.1



HTTP 2





Unary RPC

The client sends a single request and gets back a single response

Server streaming RPC

The server sends back a stream of responses after getting the client's request message

Client streaming RPC

The client sends a stream of requests to the server instead of a single request

Bidirectional streaming RPC

The call is initiated by the client calling the method and the server receiving the client metadata, method name, and deadline

Who is using gRPC?



Square

NETFLIX

lyft

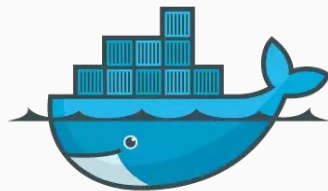


Google

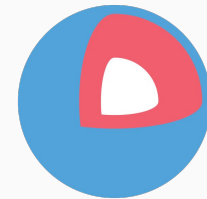


Yik Yak

VENDASTA



docker



Core OS



NETFLIX
OSS



spring



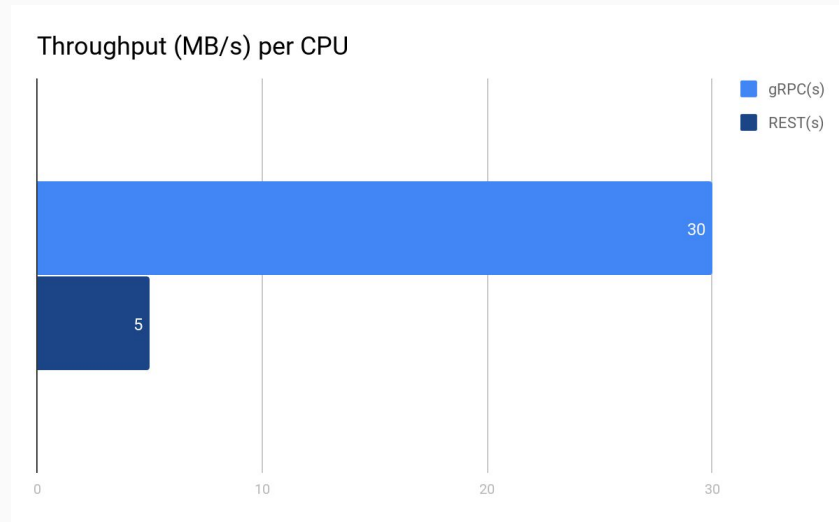
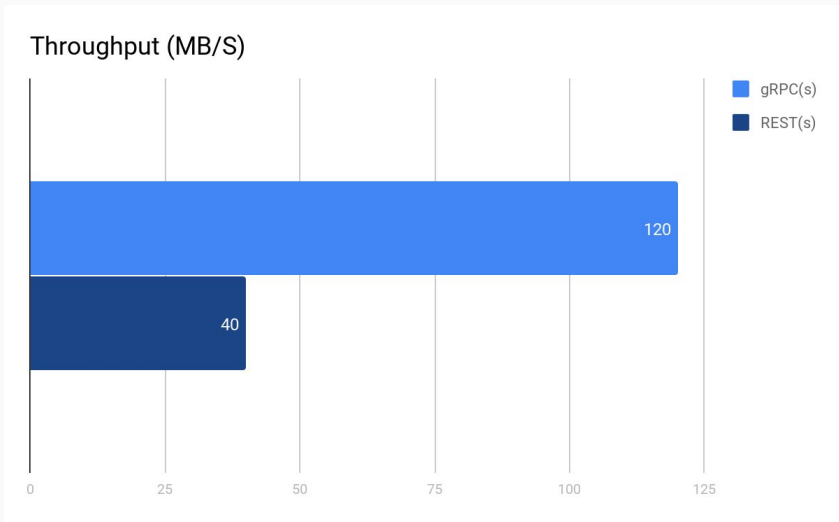
envoy



etcd

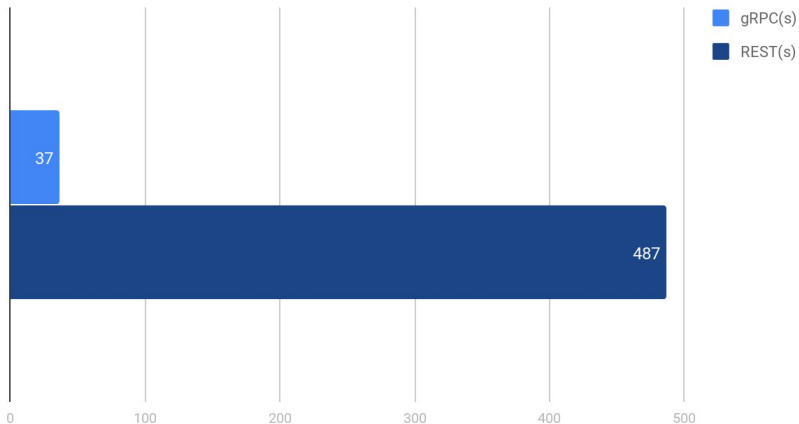
VERT.X

Why mainly Google and Netflix
are using gRPC?

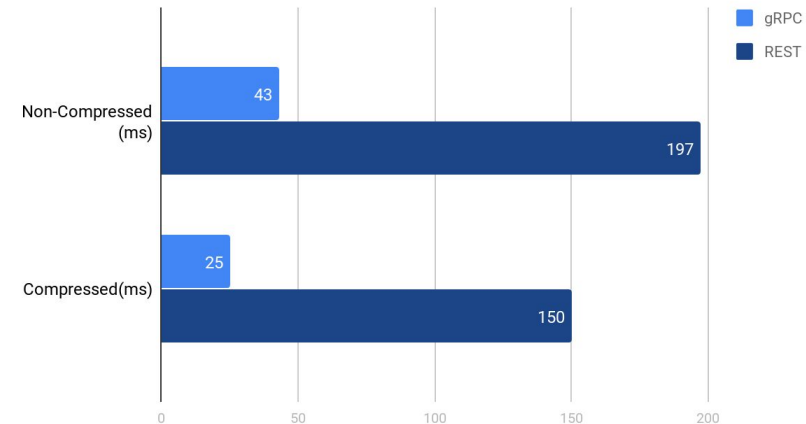


Sources: <https://cloud.google.com/blog/products/gcp/announcing-grpc-alpha-for-google-cloud-pubsub>

Time to make 300k requests



Java to Java Communication



Sources: <https://blog.gopheracademy.com/advent-2015/etcd-distributed-key-value-store-with-grpc-http2>,
<https://auth0.com/blog/beating-json-performance-with-protobuf>



Flexibility & Summary

	Goal	REST (HTTP/JSON)	gRPC
COMPATIBILITY	Single source of truth	X	✓
	Multi-platform + languages built in	X	✓
	Handle non-breaking changes.	X	✓
PERFORMANCE	Network: connection handling	Manual, 1 per call X	Built-in, Multi per conn ✓
	Speed: Transmission of data	Human-readable Text X	Binary ✓
	CPU: Improved resource usage	X	✓
MAINTENANCE	Tracing	Manual X	Easy to plug in ✓
	Logging	Manual X	Easy to plug in ✓
	Monitoring	Manual X	Easy to plug in ✓

Source: <https://www.infoq.com/presentations/grpc>

Demo

Where to use gRPC?



Microservices

gRPC shines as a way to connect servers in service-oriented environments

Client-Server Application

gRPC works just as well in client-server applications, where the client application runs on desktop or mobile devices

Integrations and APIs

gRPC is also a way to offer APIs over the Internet, for integrating applications with services from third-party providers

Browser-based Web Applications

Do not use!
grpc-gateway
grpc-web (oct/18)

Questions?

We're Hiring!
sensedia.com/carreira



sensedia

Consultoria | P&D | Marketing &
Sales | ADM | RH



Campinas | Rio | São Paulo



Thanks a million!



[/larchanjo](#)



[/luram-archanjo](#)