



gRPC en acción

Una alternativa para el desarrollo de microservicios



S P I K E





Hola!

Me llamo Matias Aravena

Machine Learning Engineer en Spike

matias@spikelab.xyz

github.com/maravenag

medium.com/@aravenagamboa



S P I K E



¿Que es gRPC?

- google Remote Procedure Calls (RPC)
- Open-Source <https://github.com/grpc>
- RPC existe desde 1981, el RFC existe desde 1988
<https://tools.ietf.org/html/rfc1050>
- Utiliza [protocols buffers](#) para serializar data estructurada.
- Permite hacer streaming, y métodos de autenticación incluidos.
- Funciona sobre HTTP/2



Protocols Buffers

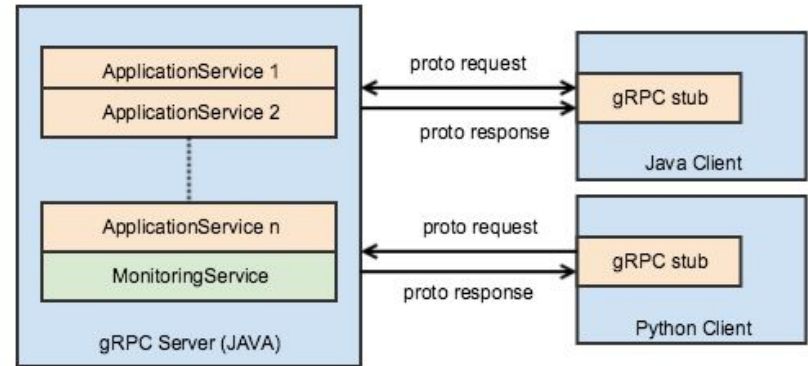
> Protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data

*> You define how you want your data to be structured once, then you can use special generated source code to **easily** write and read your structured data*

```
message SearchRequest {
  string query = 1;
  int32 page_number = 2;
  int32 result_per_page = 3;
  enum Corpus {
    UNIVERSAL = 0;
    WEB = 1;
    IMAGES = 2;
    LOCAL = 3;
    NEWS = 4;
    PRODUCTS = 5;
    VIDEO = 6;
  }
  Corpus corpus = 4;
}
```

Por qué usar gRPC sobre REST?

1. Permite definir un servicio con métodos, inputs y outputs claros y bien definidos.
2. Ideal para el desarrollo de micro-servicios.
3. Protocol buffers tienen mayor performance sobre JSON -> serializar/deserializar.
4. Permite generar auto-clientes en diferentes lenguajes.





Métodos de comunicación

- Unary RPC.
- Server Streaming RPC.
- Client Streaming RPC.
- Streaming Bi-Direccional



Métodos de comunicación

- **Unary RPC.**
- <Server Streaming RPC.
- Client Streaming RPC.
- Streaming Bi-Direccional

Veamos algo de código y cómo funciona!

