



SISTEMAS OPERATIVOS 1

CLASE 7

Leonel Aguilar

aux.leoagUILar@gmail.com

<https://github.com/leoagUILar97/so1-course/tree/main>

Sebastián Sánchez

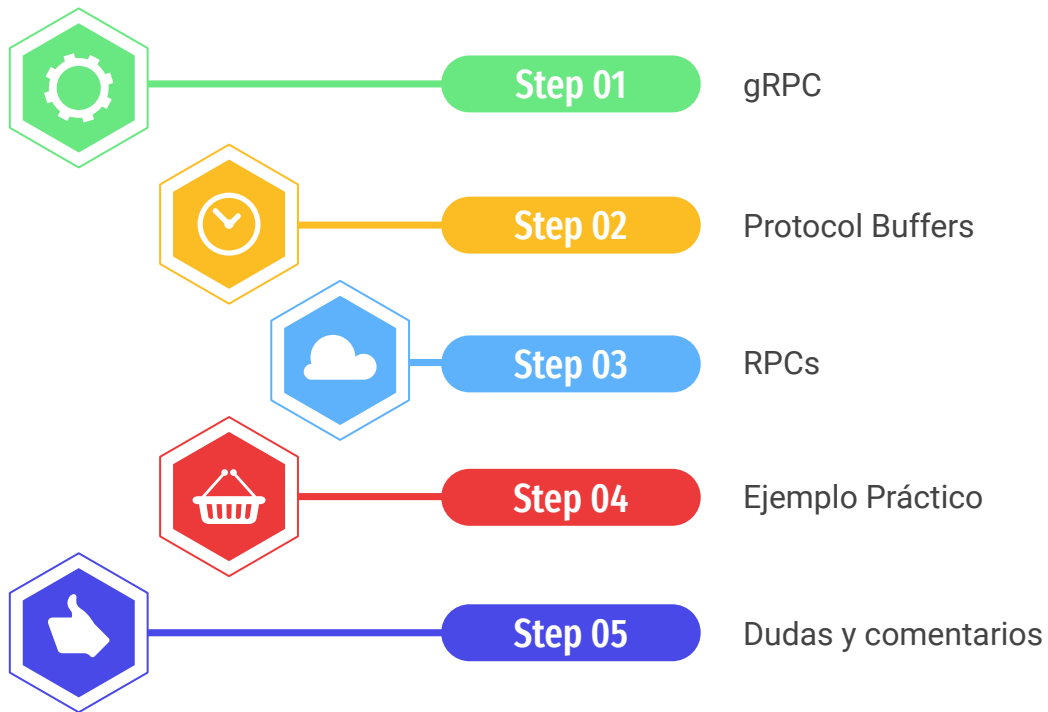
sebastiantuchez0@gmail.com

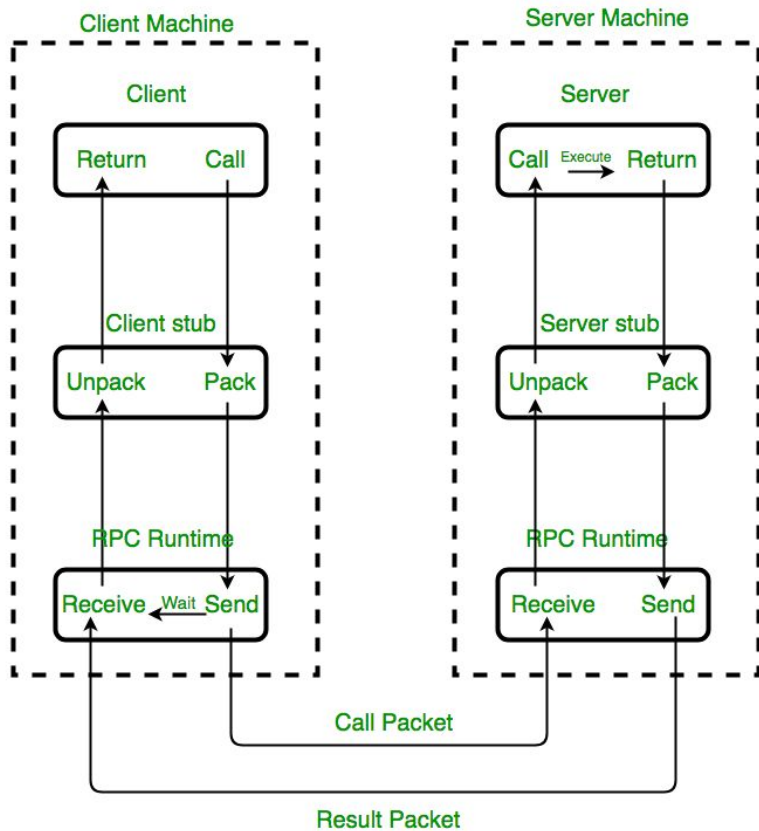
<https://github.com/SebastianSNZ/GRPCExample>

gRPC

Protocol buffers, HTTP/2, RPC

Agenda de hoy





Implementation of RPC mechanism

¿QUÉ ES UN RPC?

- Remote Procedure Call
- El cliente va a buscar código directo al server
- Es cómo llamar a una función directa en el server.

```
// Cliente

// ... otro código

int main(){
    // llamada a código server
    server.sayHello();

    return 0;
}
```

```
# Server

# ... otro código ...

def sayHello():
    print("Llamadita :3")

    return "Hola mundo!"
```

PROTOCOL BUFFERS

01

Lang. Agnostic.



Puede generar código a básicamente cualquier lenguaje.

02

CPU Friendly



Transporta información binaria. Menos tiempo realizando parsing.

03

Definir mensajes



Definir mensajes (data, request, response), nombres de servicios y endpoints de RPC.

04

Liviano



Utilizar protocol buffers significa comunicación más rápida y eficiente.

JSON VS PROTO

```

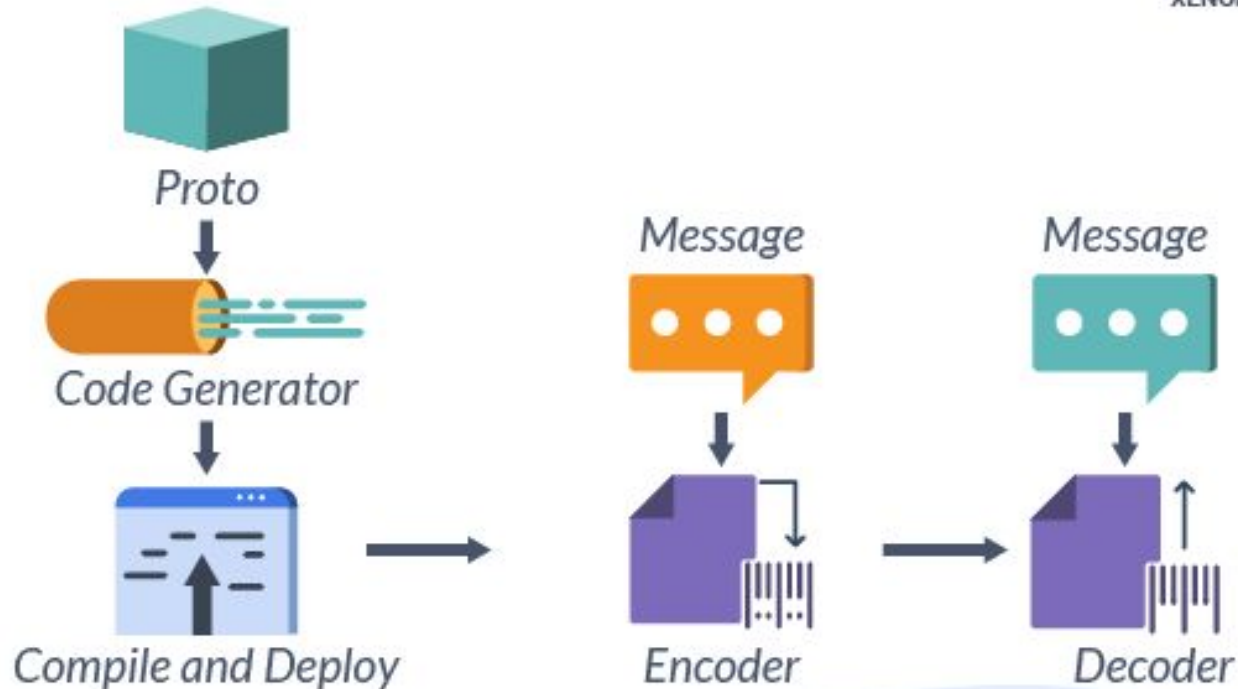
# esta info. ocupa 50 bytes

{
  "first_name": "Leonel",
  "message": "Hola mundo!"
}
```

```

# Esta info solo ocupa 20 bytes
message Greeting {
    string first_name = 1;
    string message = 2;
}
```

Google Protocol Buffers



¿COMO SE VE?

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

```
package greet;  
option go_package="greetpb";
```

```
message Greeting {  
    string first_name = 1;  
    string message = 2;  
}
```

```
message GreetRequest {  
    Greeting greeting = 1;  
}
```

```
message GreetResponse {  
    string result = 1;  
}
```

```
service GreetService{  
    rpc Greet(GreetRequest) returns (GreetResponse) {}  
}
```

HTTP/2

<https://imagekit.io/demo/http2-vs-http1>



HTTP/2

HOW IT WORKS

HTTP/2



HTTP/1.X



HTTP 1.1

vs

HTTP 2



Client

Hello

Hello

May I have a picture of a house please.

Here is the house

May I also have a picture of a car.

Here is the car

May I also have a picture of a dog.

Here is the dog

Thanks, bye

Bye



Apps



Client

Hello

Hello

May I have a picture of a house please.

And a car

And a dog

Here is the house

Here is the car

And a cat

Here is the dog

Here is the cat

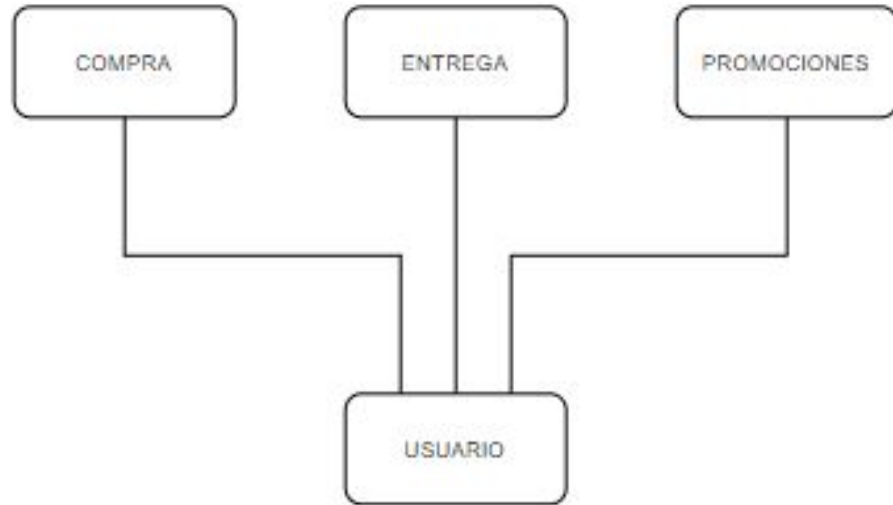
Thanks, bye

Bye

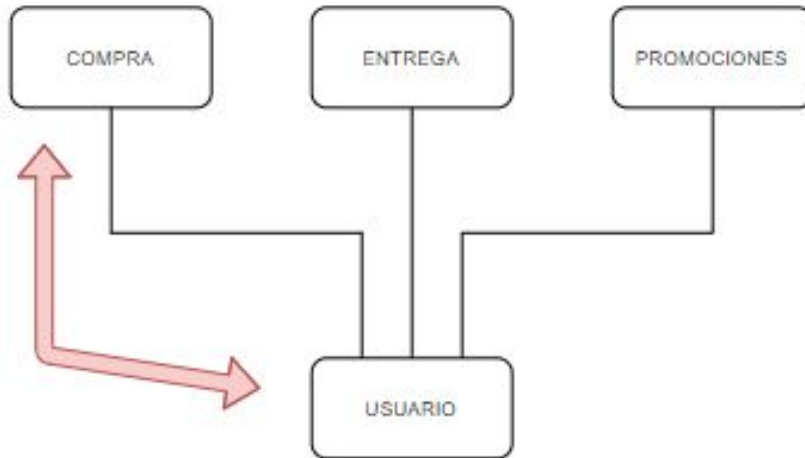


Apps

MICROSERVICES

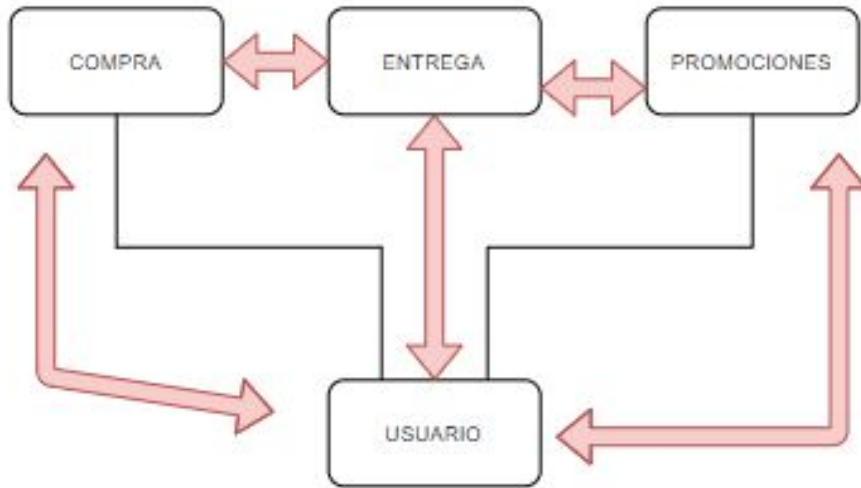


MICROSERVICES

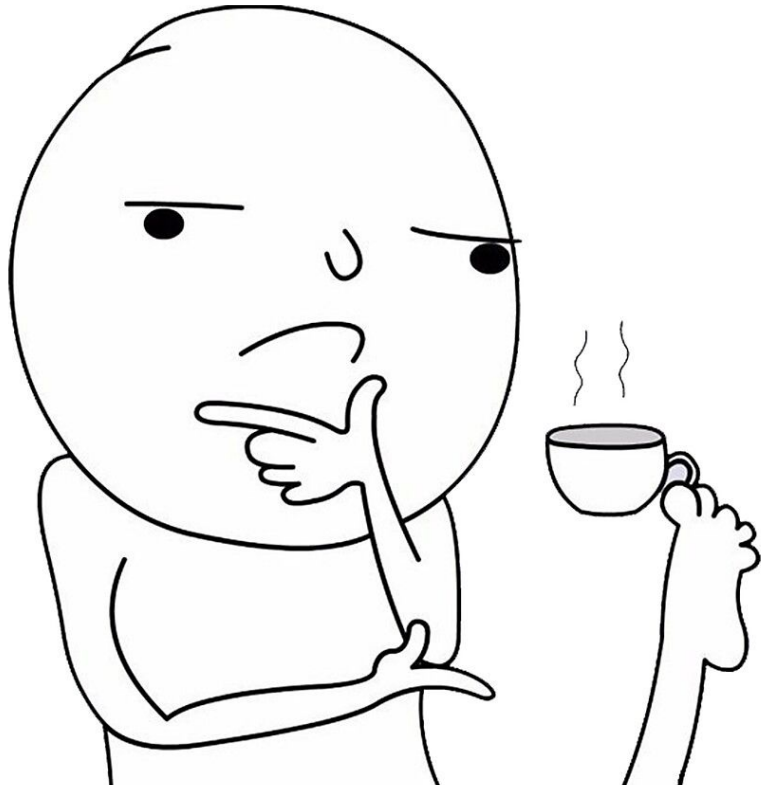


- APIs
 - ¿Qué datos manejaremos?
 - Endpoints
 - Errores
 - Latencia
 - ¿Qué tanto obtendremos de una llamada?
 - ¿Qué tantas llamadas haré?

MICROSERVICES



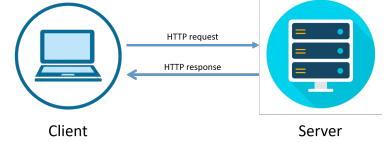
- ¡Empieza a ser poco manejable!
 - Data format
 - Error patterns
 - Load balancing
 - Authentication
 - Logging
 - Monitoring



**¿HABRÁ UNA MANERA
MÁS SENCILLA?**

gRPC

- Significa grpc remote process call.
- Es un framework RPC high performance.
- Parte del Cloud Native Computing Foundation.



Request / Response

- Permite definir peticiones y respuestas (Remote Procedure Calls).
- Maneja todo lo demás por tí.

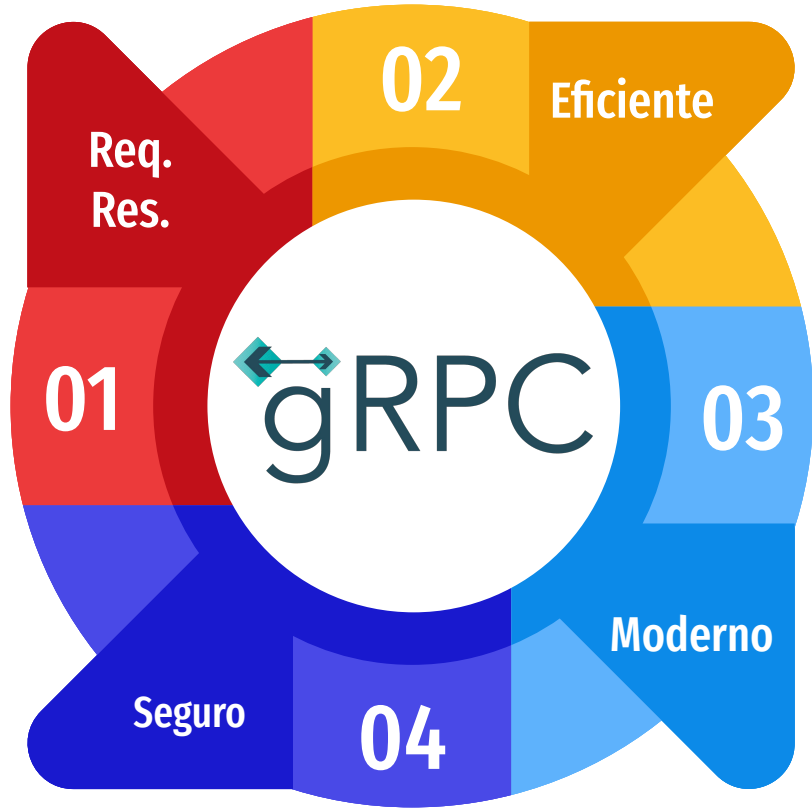


Eficiente

- Es moderno
- Rápido
- Eficiente

25 veces más rápido que REST

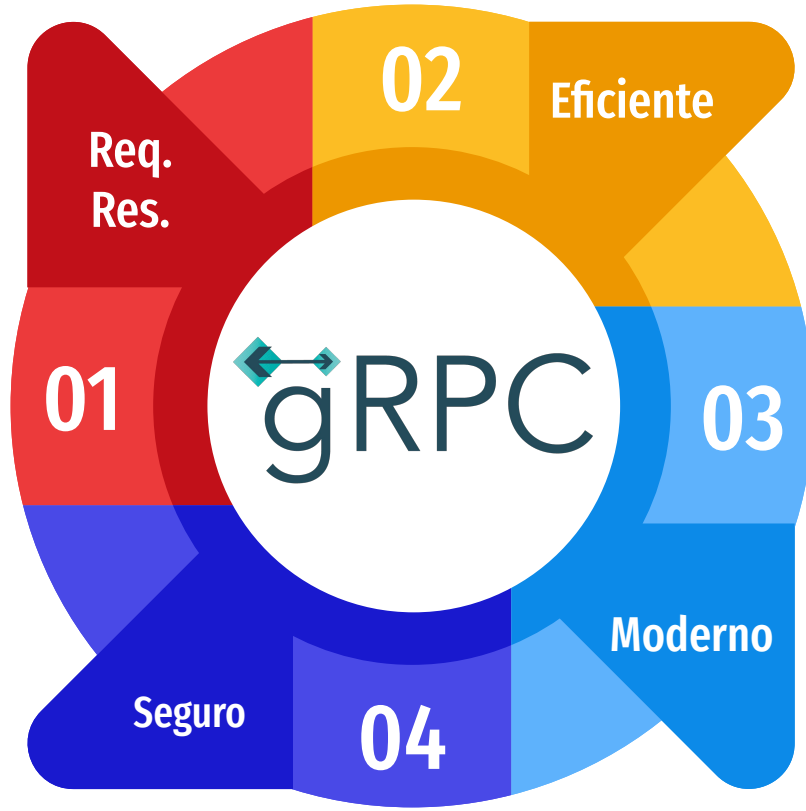
<https://husobee.github.io/golang/rest/grpc/2016/05/28/golang-rest-v-grpc.html>



MODERNO

- Utiliza http/2
- No importa el lenguaje
- Poca latencia

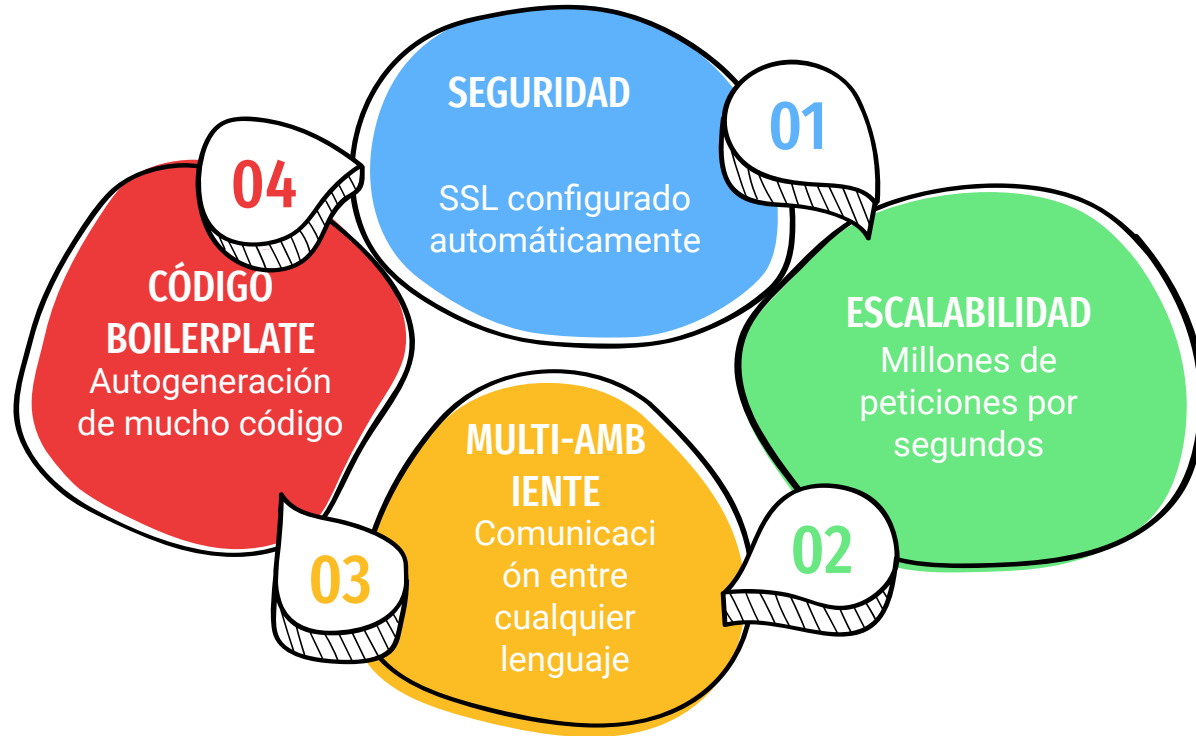




SEGURO

- Utilización de SSL sobre el canal de comunicación.
- Provee de APIs con sus respectivos métodos de encriptación.

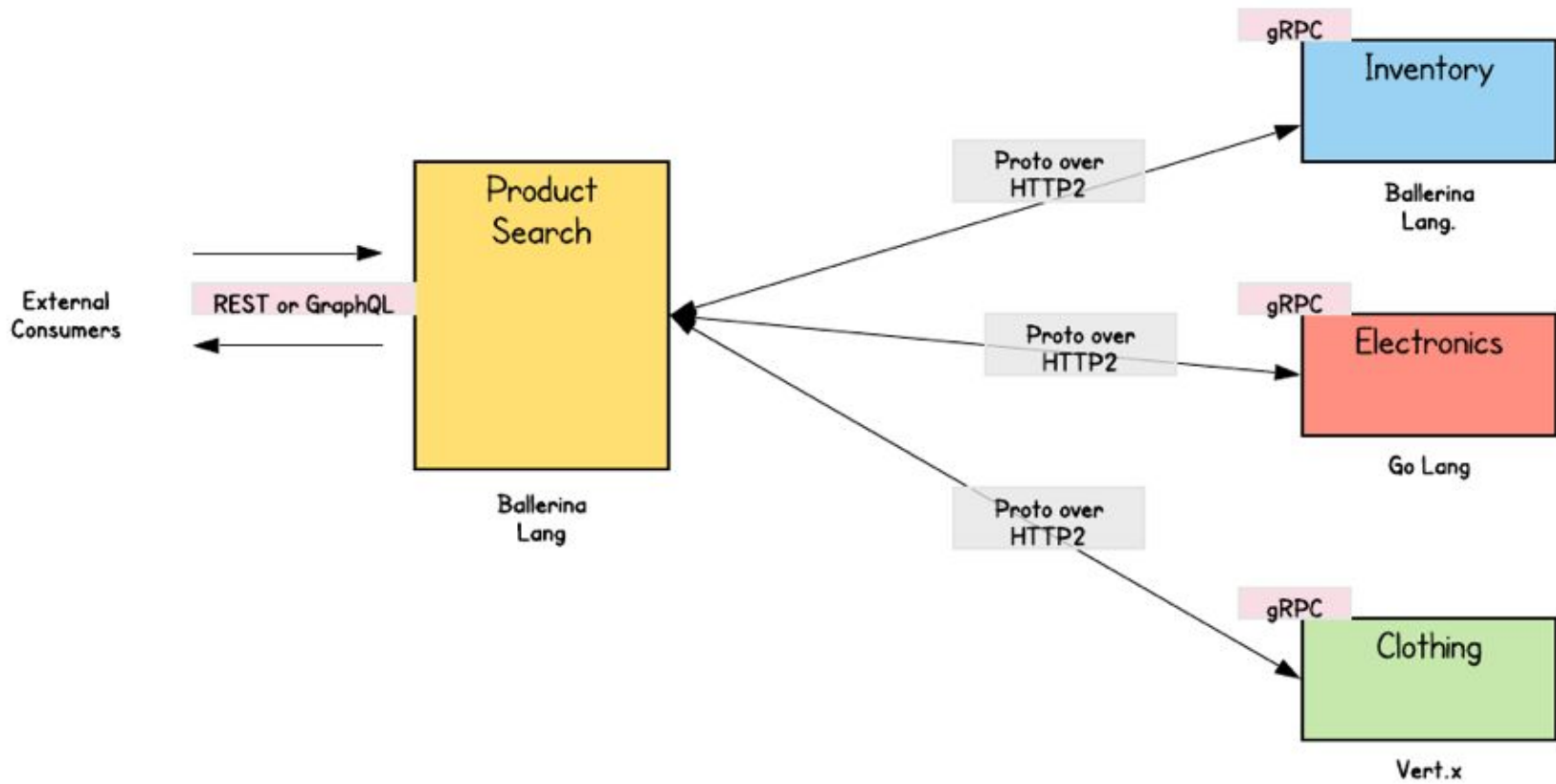
OTROS BENEFICIOS



SCALABILITY

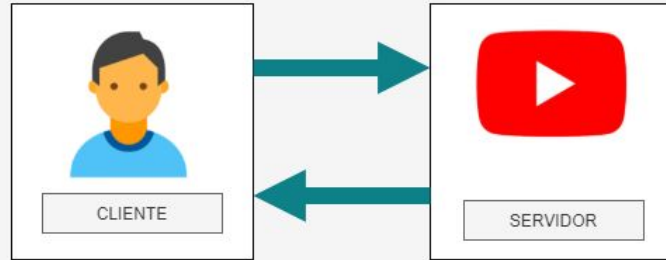
Google maneja **10 miles de millones** de peticiones por segundo con gRPC internamente.



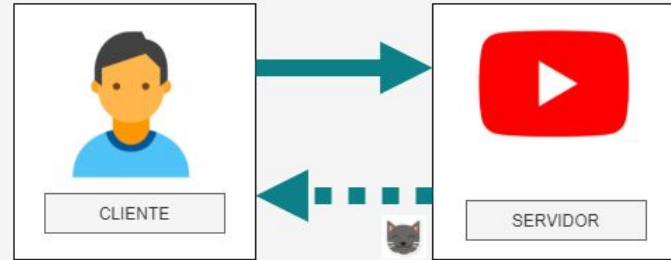


TIPOS DE API GRPC

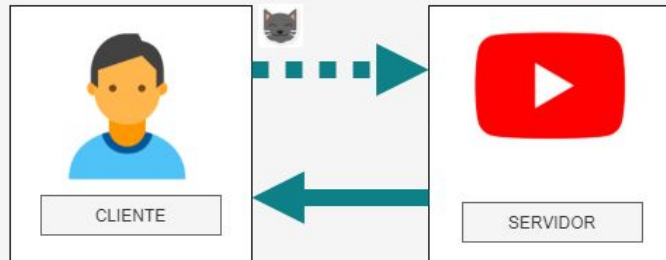
UNARY API



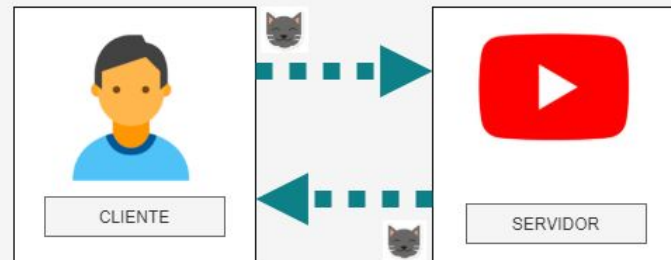
SERVER STREAMING



CLIENT STREAMING



BIDIRECTIONAL STREAMING



gRPC



{ REST }

GRPC

01	Protocol Buffer. Son datos binarios, es pequeño, liviano y rápido.
02	Utiliza HTTP-2 (Tecnología de 2015) Bidireccional y asíncrono
03	Code generation con los protocol buffer. RPC based. Soporta streaming.
04	API Oriented

REST

JSON. Es basado en texto, más lento, pequeño, incrementa la latencia.	01
Utiliza HTTP-1.1 (creado en 1997) Acepta únicamente peticiones, una única respuesta del servidor	02
Code generation utilizando OpenAPI/Swagger.	03
Resource oriented. CRUD.	04

ENTONCES, ¿POR QUÉ GRPC?



FÁCIL

Es sencillo definir y generar código en 11 lenguajes (oficiales)

LATENCIA

Utiliza el moderno HTTP-2

SSL

Seguridad SSL incluida

STREAMING

Puede realizar streaming de 3 formas diferentes.

API

API oriented. Se enfoca en qué quieres hacer.



TURNIP 2020

```
INITIATING... WAIT... BAIT A BIT MORE... ALRIGHT
//////////
---PUNK LUBRICATION VOLUME---
WELCOME TO FUNKTOWN! BRODER
REFINE (PUNK PARAMETERS) SUPER CRUNKY
WORKING
CRUNKYNESS SET TO SETTINGS: SUPER(SUPER CRUNKY)
C:\LOVE_THE_WAY_YOU_MOVE.MP3 -LOAD TO EXT. DEVICE
WRITTEN SUCCESSFULLY TO EXT. DEVICE
C:\DROP_BA_BEAT LIKE (DELOVE THE_WAY_YOU_MOVE.MP3)
HEAT DROPPING...
BEAT SUCCESSFULLY DROPPED. IS IT FRESH? Y/N
Y
TIGHTY. INITIATE FLY SOVIET DANCE KICKS TESTING
PHASE? Y/N
Y
INITIATING... PLEASE WAIT...
SUCCESSFULLY INITIATED.
C:\HASH
HASH 3-24 LX-1RT      MAD HEATS      FLY KICKS
PUNK      BROMPS      CRUNK
DANCE    POPS      LOCKS
HASH 1-2: CR DANCE
```