


{ REST }
is not the best

¿Quién soy?

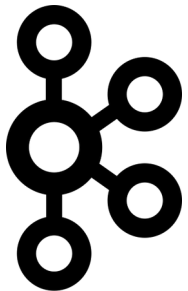
Bernal Varela Gómez.

 bernal@corunet.com

 @bernalvarela

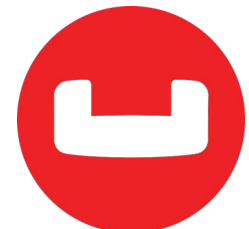


¿Qué hago?

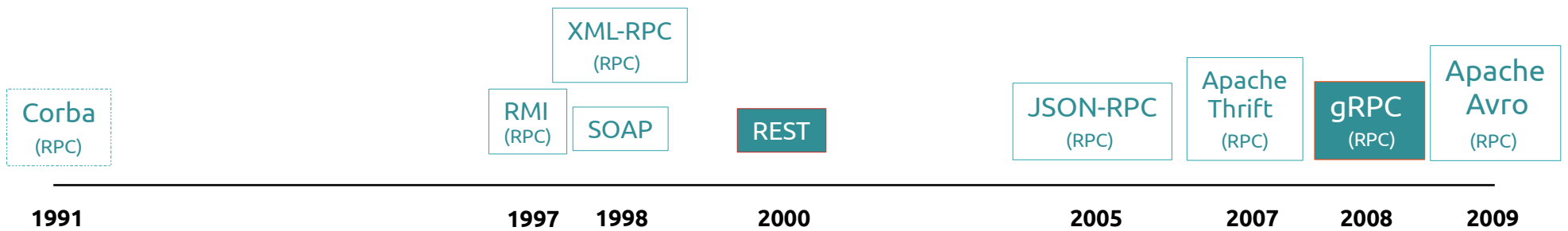


OPENAPI
INITIATIVE

JUnit







{ REST }

FIGHT

↶ GRPC ↷

{ REST }

Representational state transfer { REST }

Arquitectura Cliente / Servidor

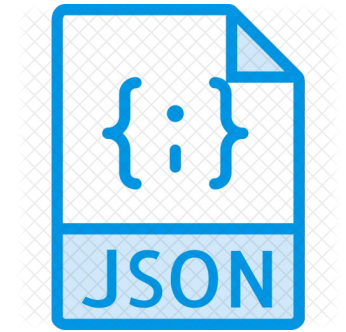
Peticiones sin estado

Operaciones bien definidas

Sintaxis universal para identificar recursos (URI)

Uso de hipertextos

{ REST }



Cuerpo de llamada y respuesta Json

Formato texto.

No comprimido.

Fácil lectura.

Alta transferencia de datos.

Serialización costosa.

```
{  
  "event": "DevFest Asturias",  
  "city": "Gijón",  
  "talk": "Rest is not the best",  
  "speaker": "Bernal Varela",  
  "company": "Corunet",  
  "location": {  
    "name": "Palacio de congresos",  
    "lat": 43.5383048,  
    "lon": -5.651559,  
    "region": "Asturias"  
  }  
}
```


HTTP / 1.1

{ REST }

Protocolo muy complejo.

Una llamada por objeto.

Tcp Handshake necesario para cada petición.

Gran cantidad de peticiones y latencia.

Límite de conexiones al mismo dominio. (6-8).

Handshake

{ REST }



[syn] YO ILL TELL YA WHAT I WANT
WHAT I REALLY REALLY WANT

[syn-ack] ILL TELL YA WHAT I WANT
WHAT I REALLY REALLY WANT-SO TELL
ME WHAT YOU WANT WHAT YOU
REALLY REALLY WANT

[ack] SO TELL ME WHAT YOU WANT
WHAT YOU REALLY REALLY WANT ?



↑ GRPC ↓

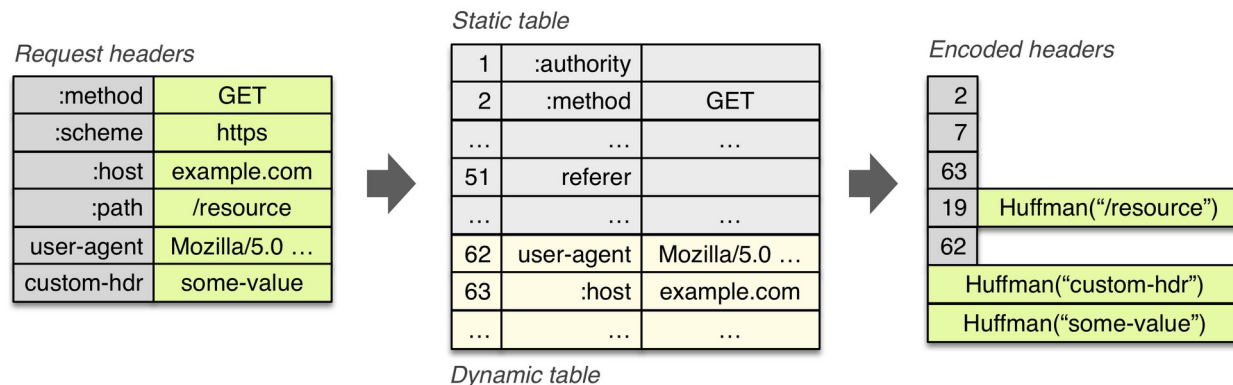
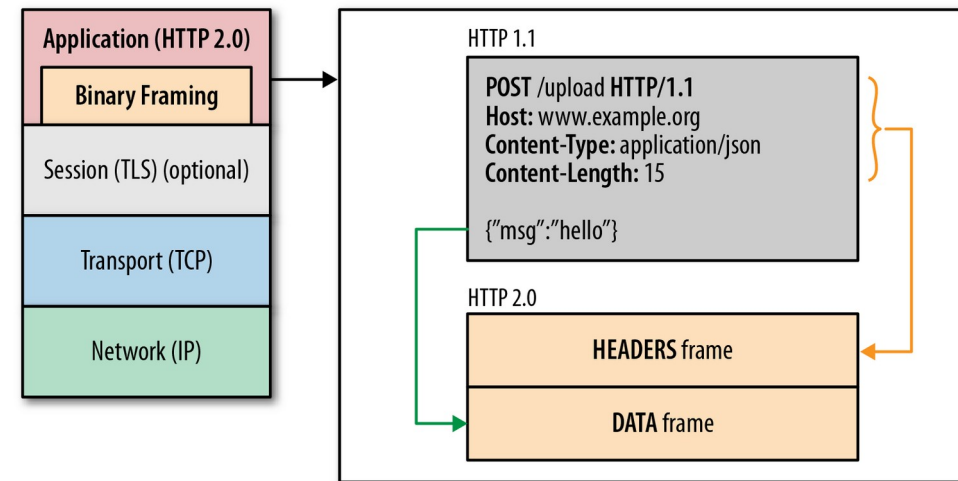


HTTP / 2

Una conexión por origen.

Frames binarios y compresión

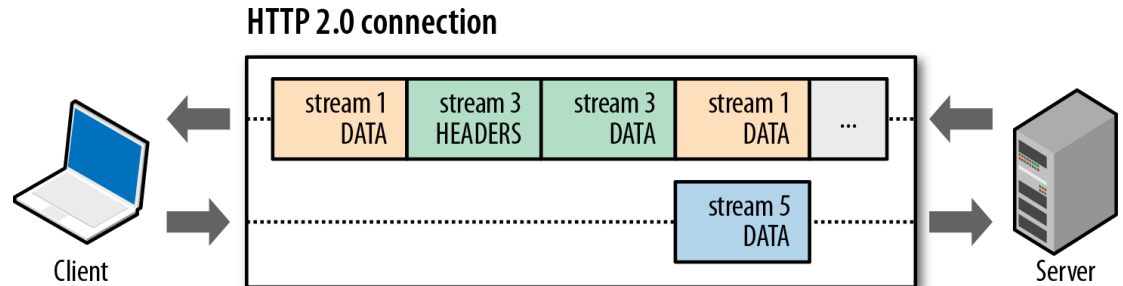
Compresión de cabeceras



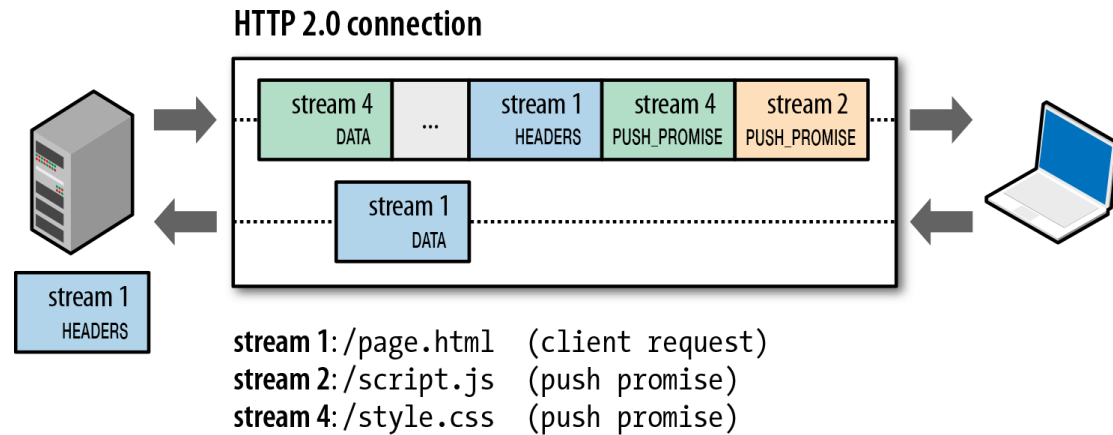


HTTP / 2

Streaming Bidireccional



Server push.



Flujo de Control



Lenguaje creado por Google.

- Contrato que define la estructura de mensajes (IDL).

Flexible y eficiente.

Más sencillo, rápido y de menor tamaño que XML.

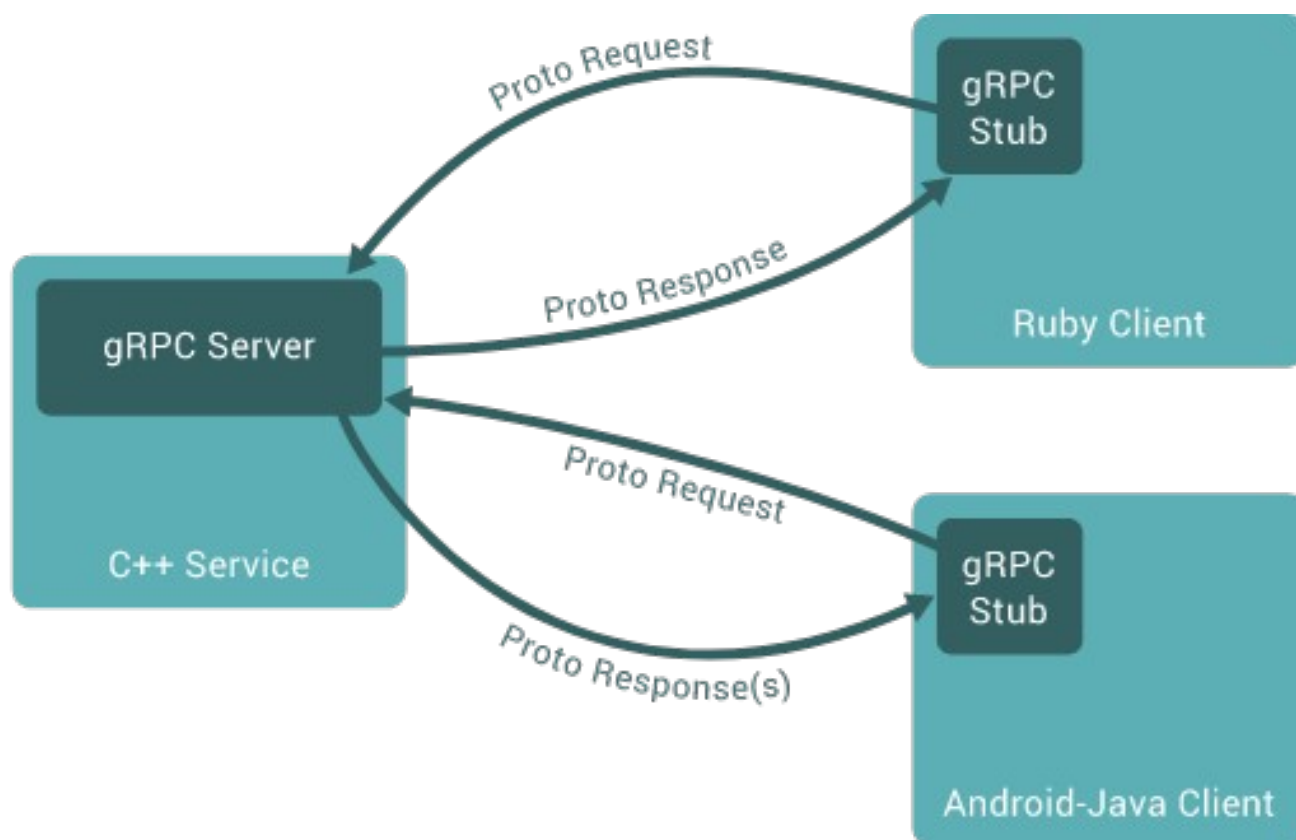
Generaremos código a partir de esta definición.

Se generará stubs/clientes y servicios.

Se ofrece un compilador que permite generar código en diferentes lenguajes.

Un servicio gRPC podrá recibir llamadas de código en diferentes lenguajes.

C++, Java (Android), Python, Go, Ruby, C#, JavaScript, Objective-C, PHP, Dart.



Protobuffer V2



```
syntax = "proto2";  
// the Person message definition ⑤  
message Person { ①  
  ③ required string name = 1; ②  
    required int32 id = 2;  
    optional string email = 3;  
  
    enum PhoneType {  
      MOBILE = 0;  
      HOME = 1;  
      WORK = 2;  
    }  
  
    message PhoneNumber {  
      required string number = 1;  
      optional PhoneType type = 2 ④[default = HOME];  
    }  
  
    repeated PhoneNumber phone = 4;  
}
```

- ① Definición del mensaje
- ② Definición de atributos
Tipo Nombre = Posición
Escalaes
Enumerados
Tipos Complejos
- ③ Características del atributo
Opcional
Required
Repeated
- ④ Valores por defecto
- ⑤ Comentarios

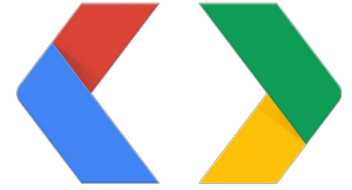
Protobuffer V3



```
syntax = "proto3";  
// the Person message definition  
message Person {  
    string name = 1;  
    int32 id = 2;  
    string email = 3;  
  
    enum PhoneType {  
        MOBILE = 0;  
        HOME = 1;  
        WORK = 2;  
    }  
  
    message PhoneNumber {  
        string number = 1;  
        PhoneType type = 2;  
    }  
  
    repeated PhoneNumber phone = 4;  
}
```

Protobuffer

Definición de servicios



```
rpc SayHello>HelloRequest) returns (HelloResponse){}
```

Cliente envía una petición y recibe una respuesta

```
rpc SayHello>HelloRequest) returns (stream HelloResponse){}
```

Cliente envía una petición y espera hasta recibir todas las respuestas del servidor

```
rpc SayHello(stream HelloRequest) returns (HelloResponse){}
```

Cliente envía una secuencia de mensajes y al finalizar esperará por la respuesta del servidor

```
rpc SayHello(stream HelloRequest) returns (stream HelloResponse){}
```

Streaming bidireccional en el que ambos extremos envían una secuencia de mensajes usando un stream de lectura / escritura.

Ambos streams funcionan de manera independiente, cliente y servidor escribirán en el orden que deseen.

NETFLIX

Definición clara APIs.

Generación de código en distintos lenguajes.

Facilitar la integración de terceros.

Minimizar errores por código de clientes.

Realizar Streaming

Minimizar latencia.

EMPRESA I

Migración de monolito a microservicios.

Millones de peticiones diarias.

Mejorar comunicación entre microservicios.

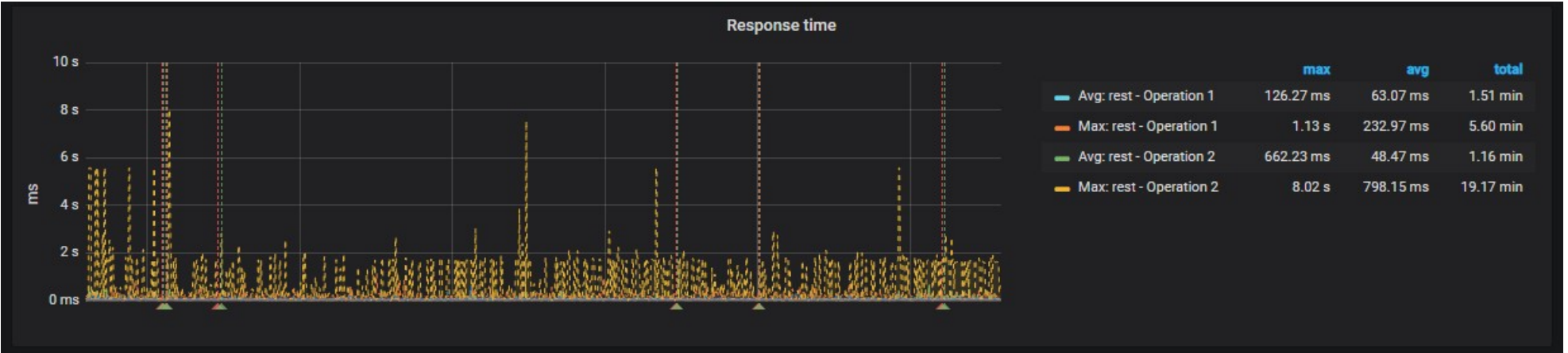
Búsqueda de baja latencia.

Simplificar desarrollo.

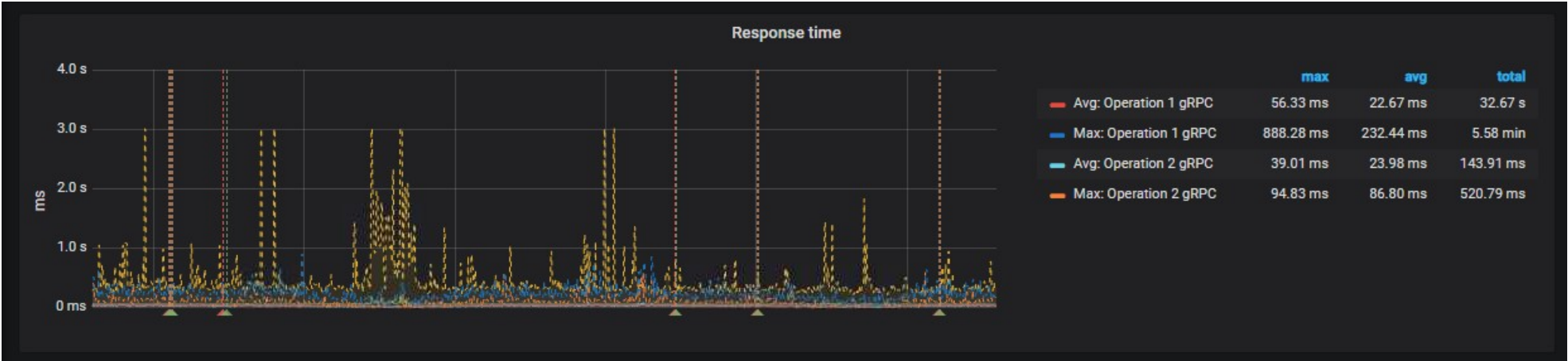
Minimizar errores de clientes.

EMPRESA I

{ REST }



↑ gRPC ↓

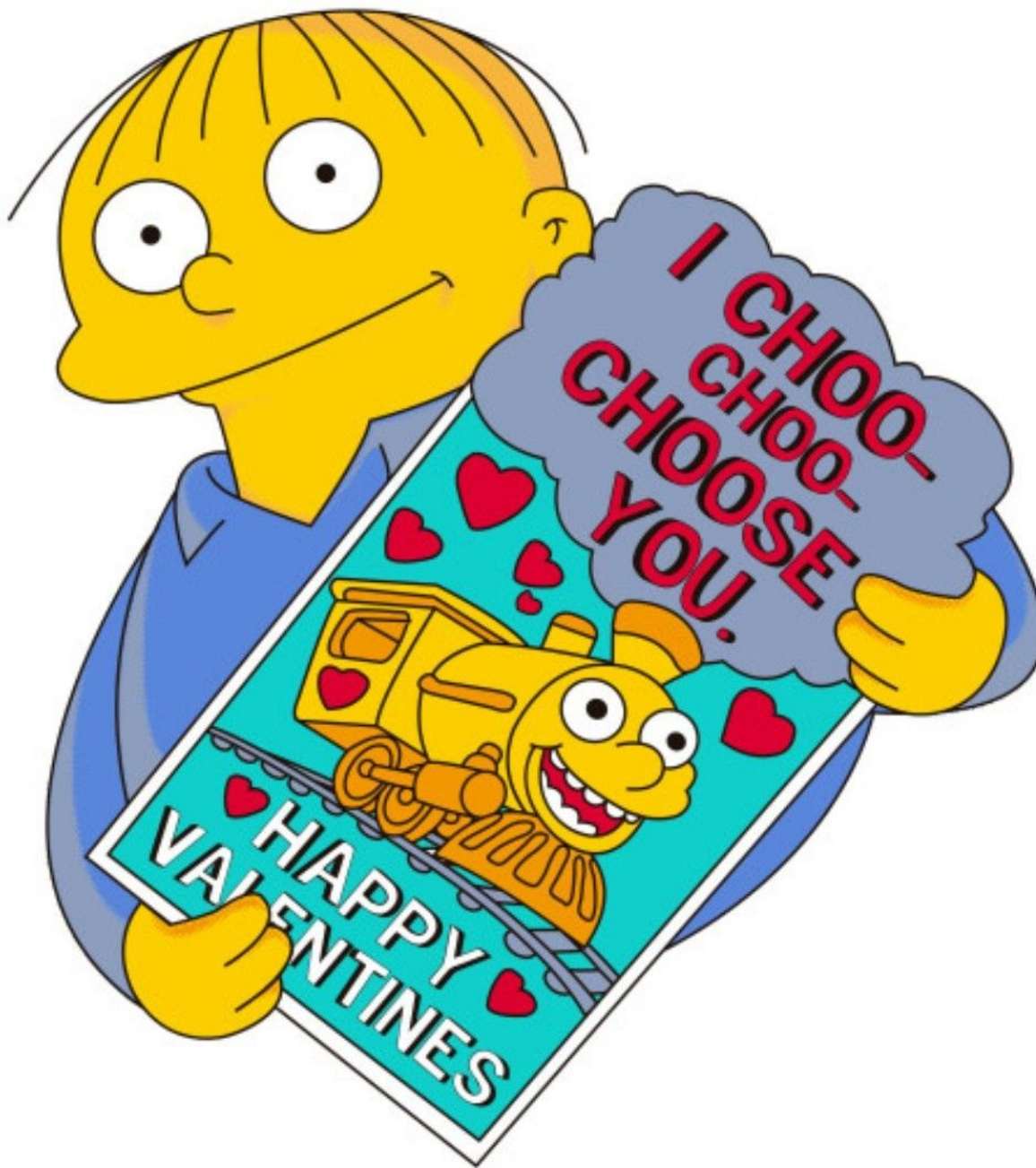




	Feature	gRPC	HTTP APIs with JSON
NETFLIX EMPRESA I	Contract	Required (<i>.proto</i>)	Optional (OpenAPI)
NETFLIX EMPRESA I	Transport	HTTP/2	HTTP
NETFLIX EMPRESA I	Payload	Protobuf (small, binary)	JSON (large, human readable)
	Prescriptiveness	Strict specification	Loose. Any HTTP is valid.
NETFLIX	Streaming	Client, server, bi-directional	Client, server
	Browser support	No (requires grpc-web)	Yes
	Security	Transport (HTTPS)	Transport (HTTPS)
NETFLIX EMPRESA I	Client code-generation	Yes	OpenAPI + third-party tooling

{ REST }

is not the best



<http://bit.ly/restIsNotTheBest>