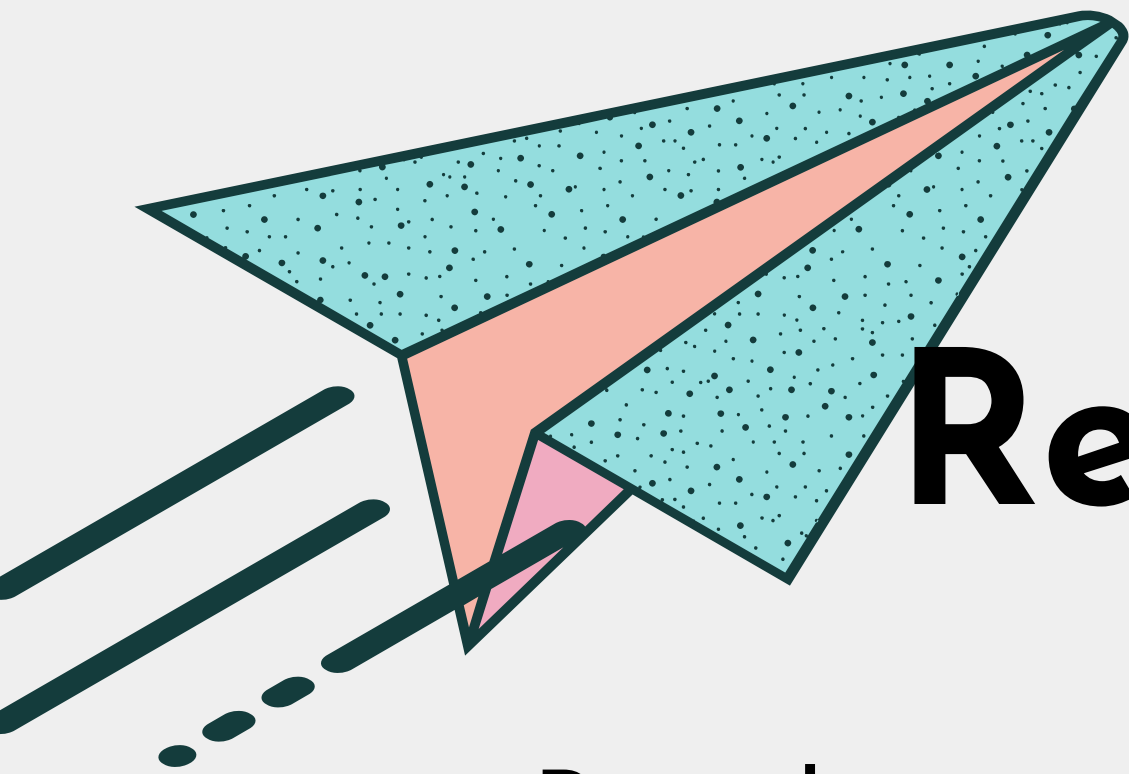




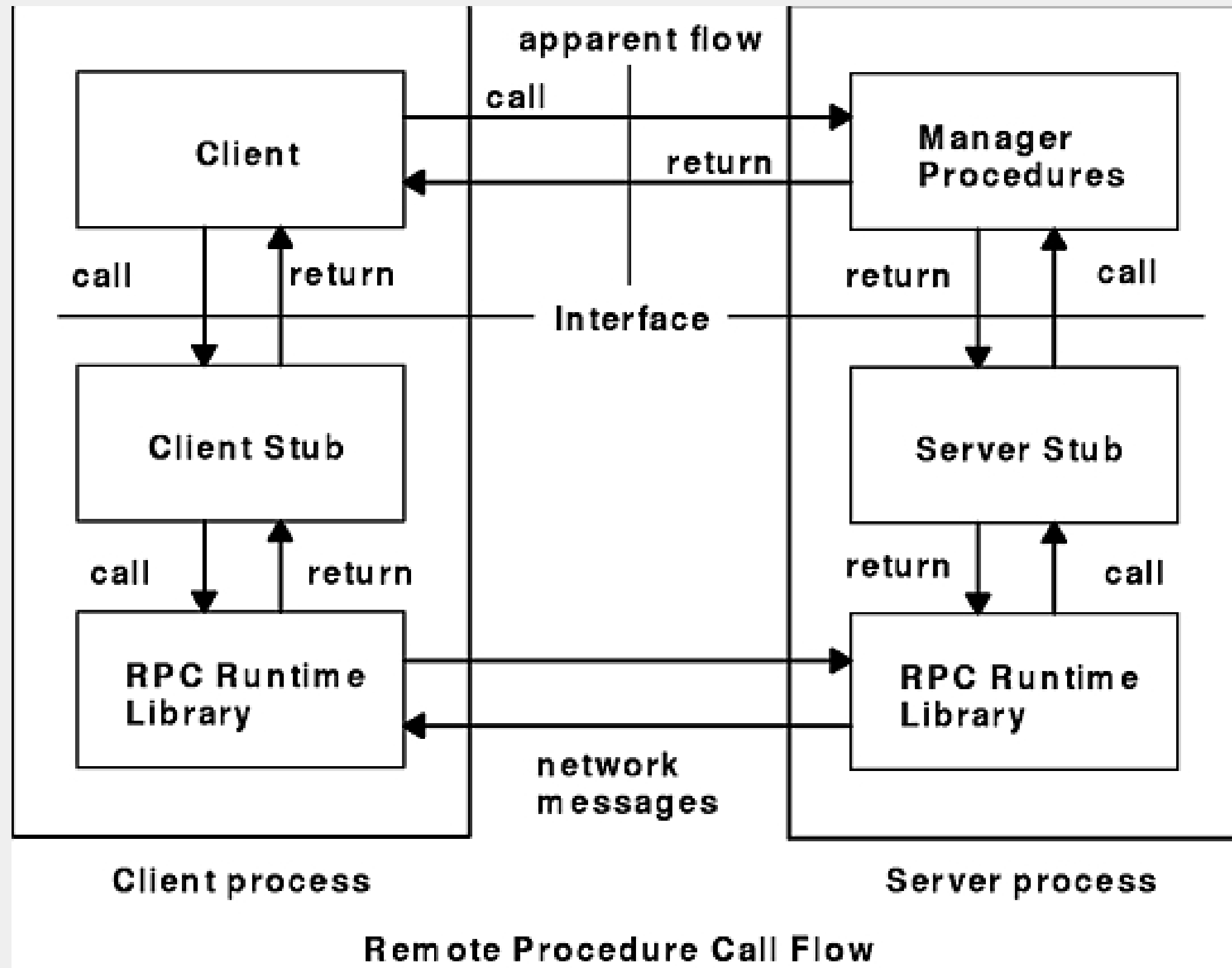
RPC vs gRPC & ProtoBuf



Relembrando o RPC

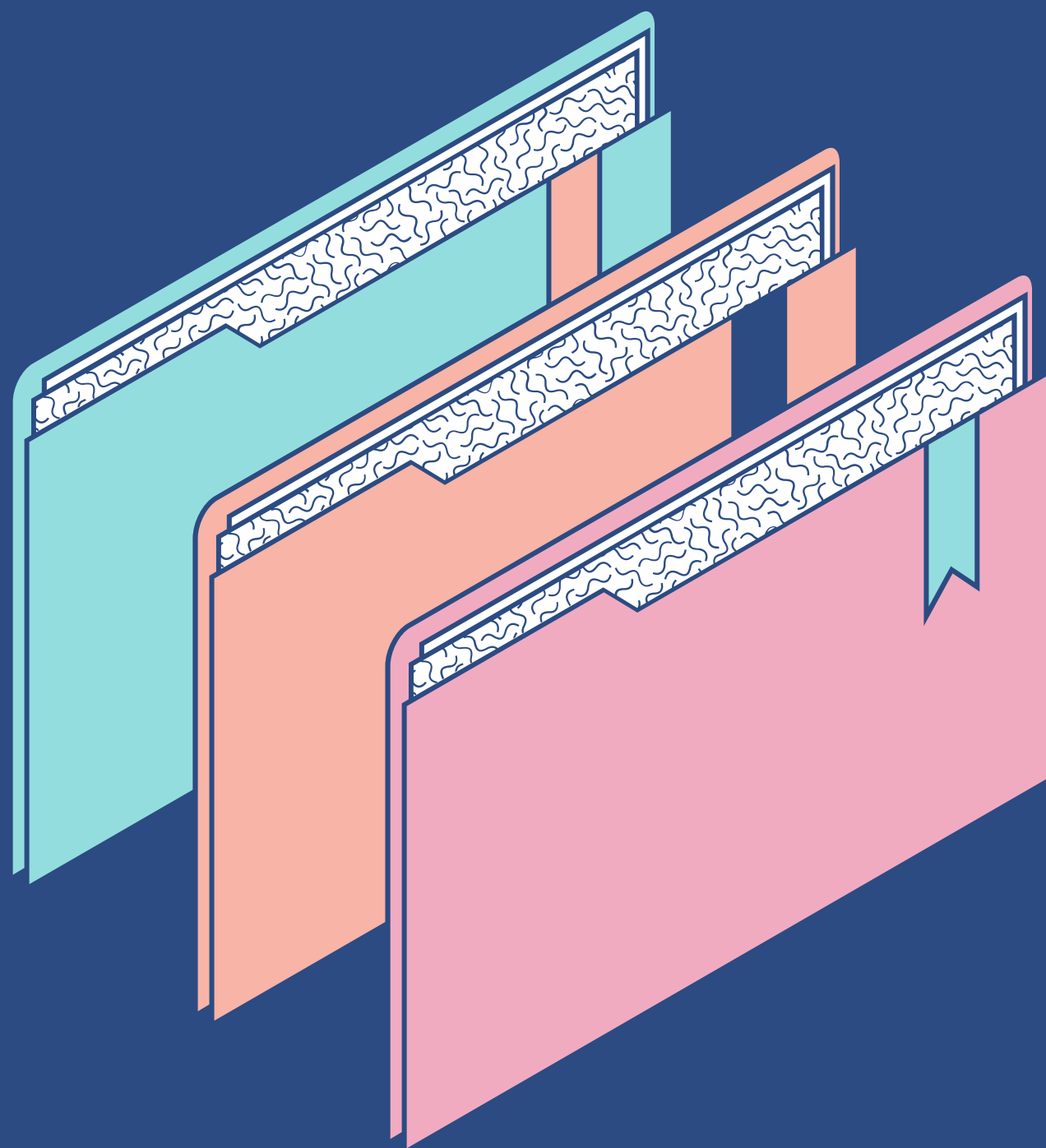
- Paradigma de comunicação de alto nível
- Uso de protocolos de transmissão de baixo nível (TCP/UDP)
- Lógica de comunicação entre cliente e servidor
- Uso do protocolo eXternal Data Representation (XDR)
 - Padronização da representação dos dados
 - Conversão de parâmetros e resultados
- Uso do Port Mapper para localização das portas dos serviços

Arquitetura RPC



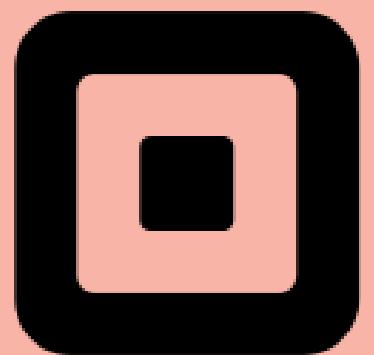
gRPC

O QUE É?



- Um framework open source de RPC para diversas linguagens
- Comunicação de serviços através de servidores e clientes
- Servidores são responsáveis por lidar com as chamadas dos clientes
- Clientes são Stubs que comunicam com os servidores
- Por *default* faz o uso dos protocols buffers como linguagem de definição de interface (IDL) e do HTTP/2 como transporte

NETFLIX

 Square Core OS

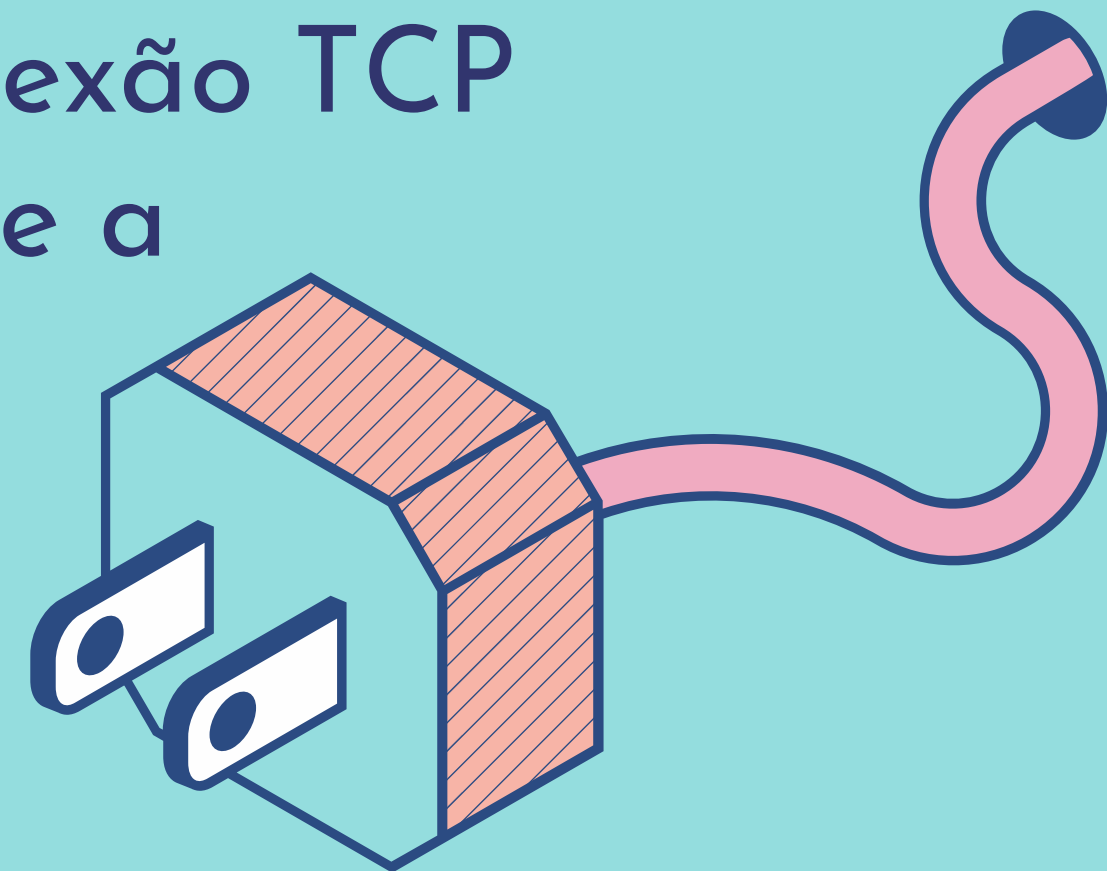
CISCO

Alguns exemplos de aplicações do gRPC

- Microserviços
- *Stream* de telemetria de dados
- *Stream* de configurações de rede

Funcionalidades

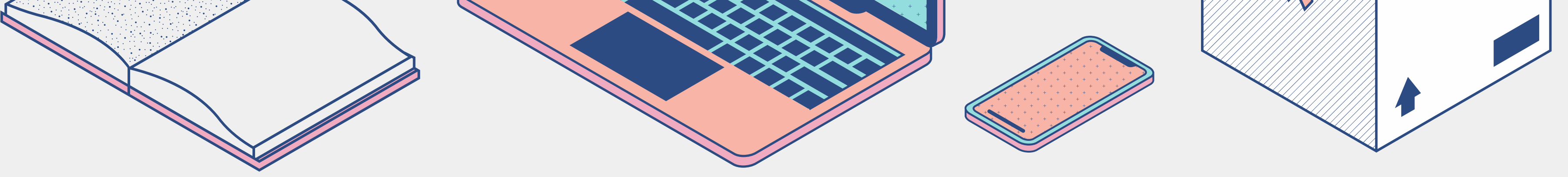
- Chamadas de procedimento remoto em lote
- Chamadas de procedimento em *Broadcast*
- Procedimentos de *Call-back*
- Múltiplos *requests* em uma única conexão TCP
- Permite o sincronismo/assincronismo e a persistência na comunicação



Tipos de Serviços



- Unary RPC
 - Cliente envia uma requisição e o servidor retorna uma resposta
- Server streaming RPCs
 - Cliente envia uma requisição e o servidor retorna uma stream para o cliente ler uma quantidade variada de mensagens como resposta
- Client streaming RPCs
 - Cliente envia uma stream de mensagem como requisição e o servidor lê as mensagens e retorna uma resposta
- Bidirectional streaming RPCs
 - Cliente envia uma stream de mensagem como requisição e o servidor devolve uma stream de mensagem como resposta.

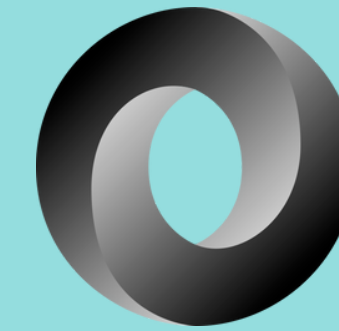


Protocol Buffer

Mecanismo extensível de serialização de dados estruturados.



Protobuf vs JSON



- Transmissão eficiente de dados
- Conjunto de regras que definem a troca de mensagens
- Mensagens no formato binário
- O dado é associado a um esquema
- Usado para troca de mensagens entre serviços internos
- Suporte à vários tipos de dados

- Transmissão lenta de dados
- Define apenas as trocas de mensagens
- Mensagens no formato de texto
- Não possuem um esquema associado
- Usado para troca de mensagens entre serviços web
- Suporte limitado aos tipos de dados

HTTP/2

Redução de latência

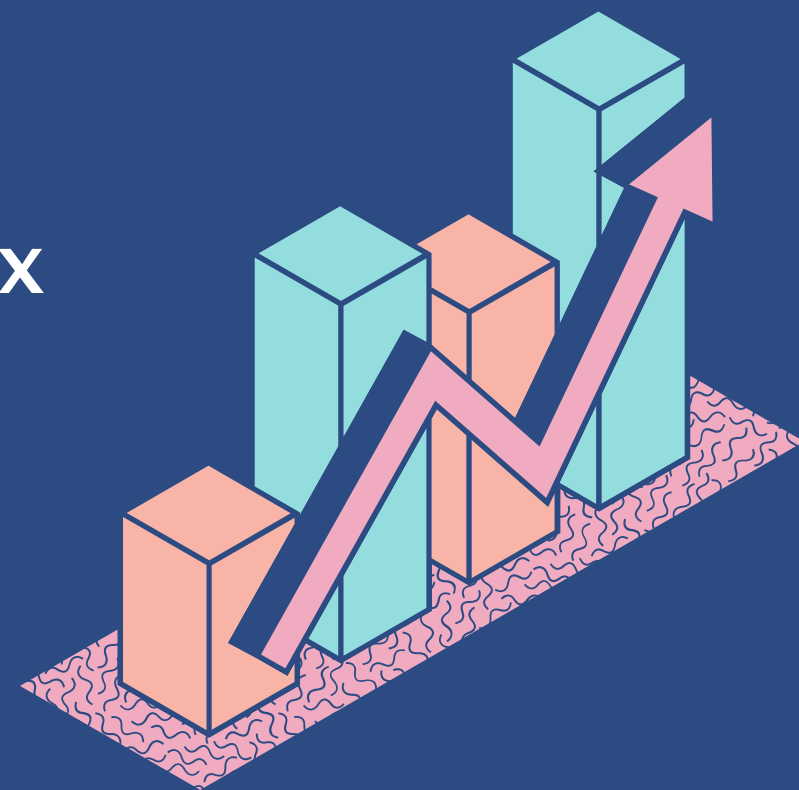
Minimização da carga do protocolo

Adição de suporte a priorização de solicitação

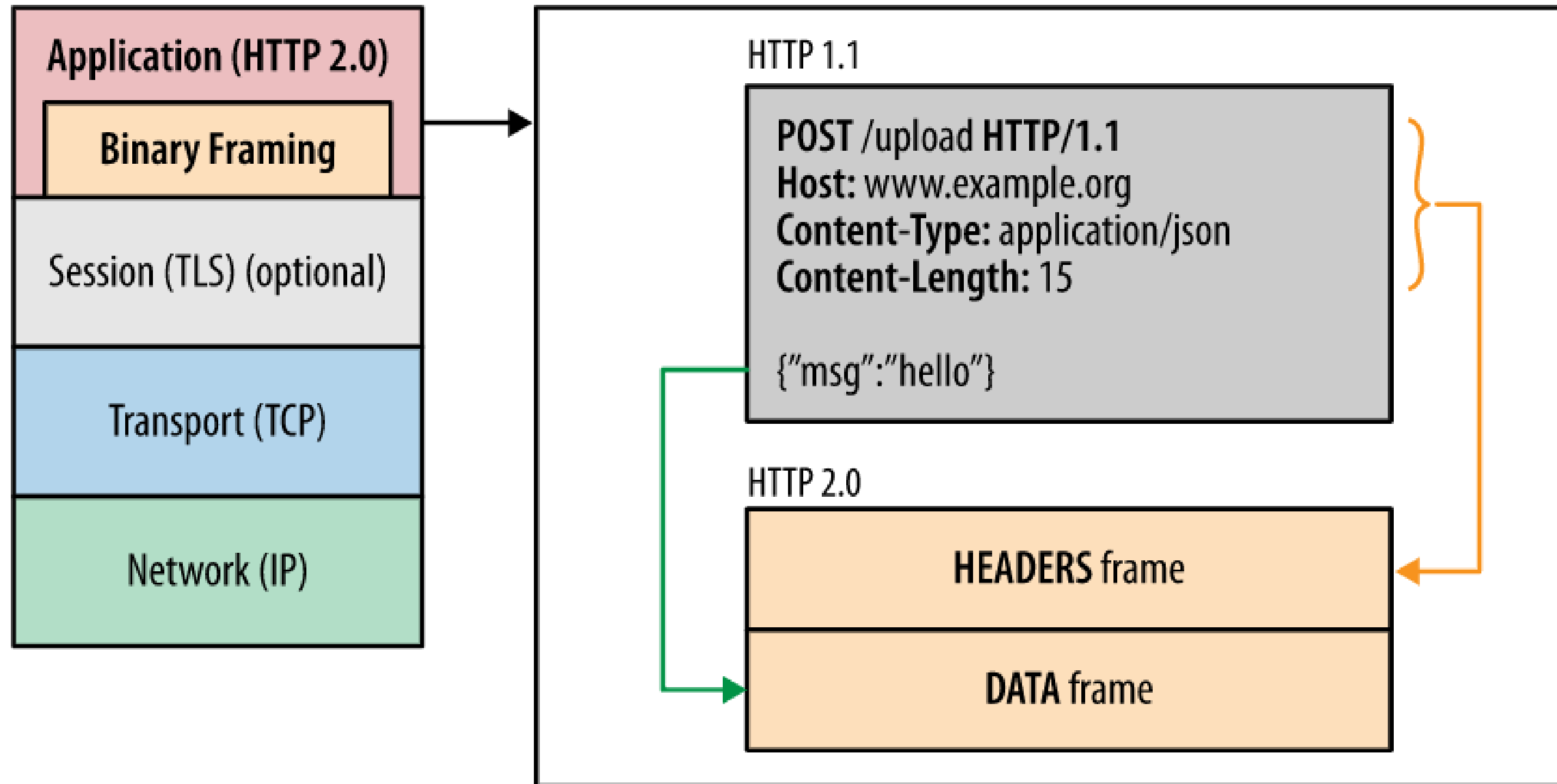
Envio push de servidor

Formatação e transporte de dados

2 porque não é retro compatível com versões 1.x



HTTP 2.0 - Binary framing layer

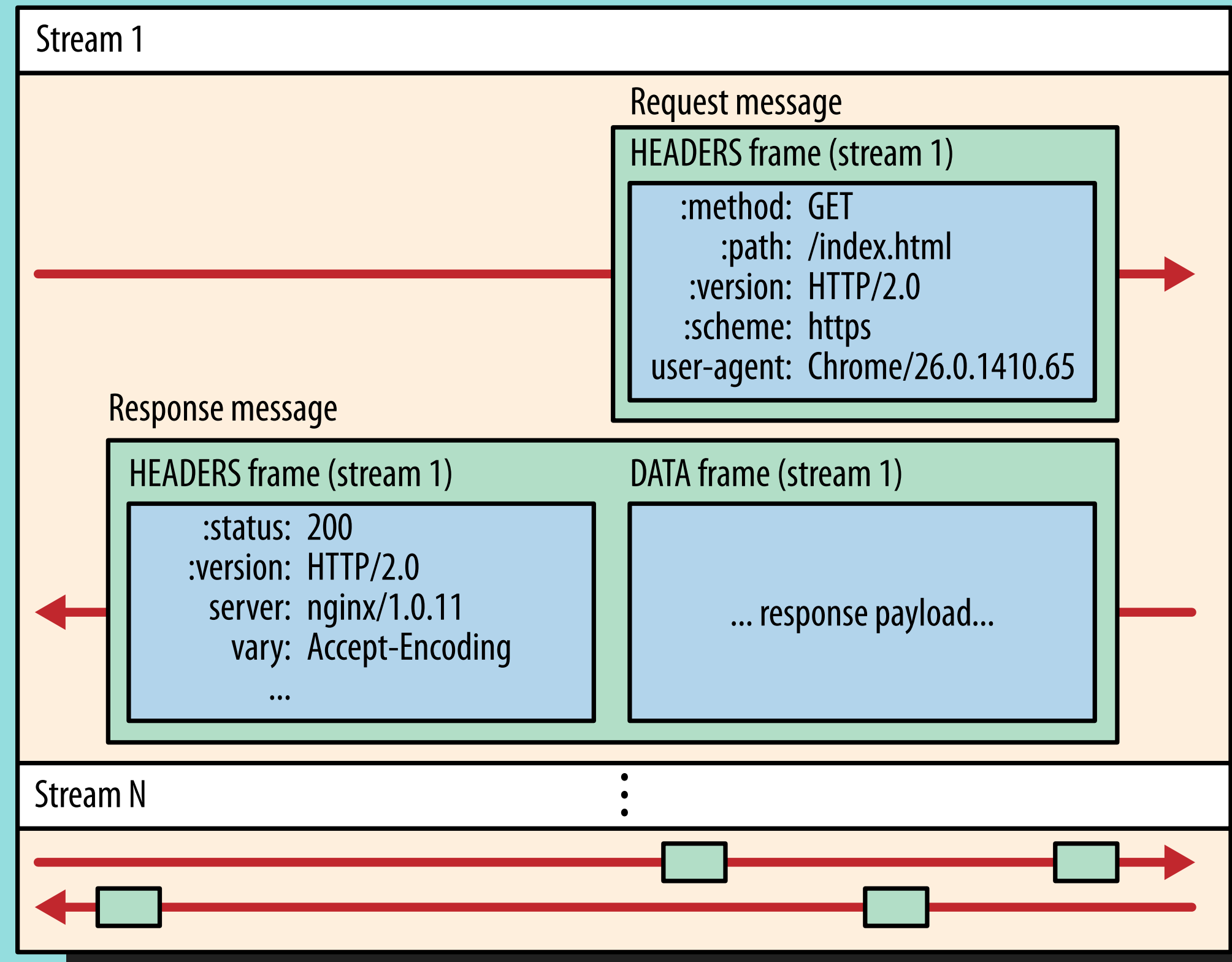


Fonte: Introdução a HTTP/2. Google. Disponível em: <<https://developers.google.com/web/fundamentals/performance/http2>>.

Acesso em: 9 de fev. de 2022.

HTTP 2.0 - Streams, messages, and frames

Connection

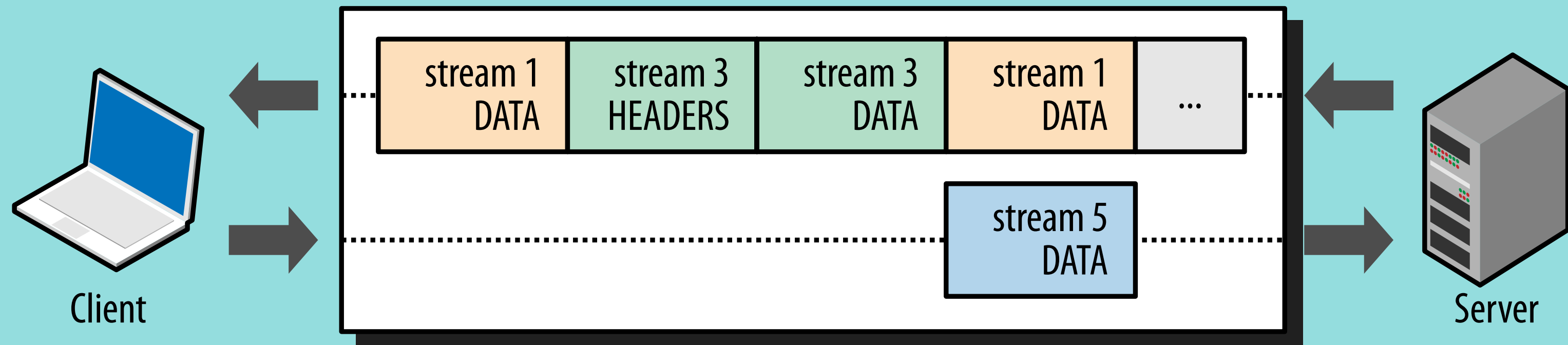


Fonte: Introdução a HTTP/2. Google. Disponível em: <<https://developers.google.com/web/fundamentals/performance/http2>>.

Acesso em: 9 de fev. de 2022.

HTTP 2.0 - Request and response multiplexing

HTTP 2.0 connection

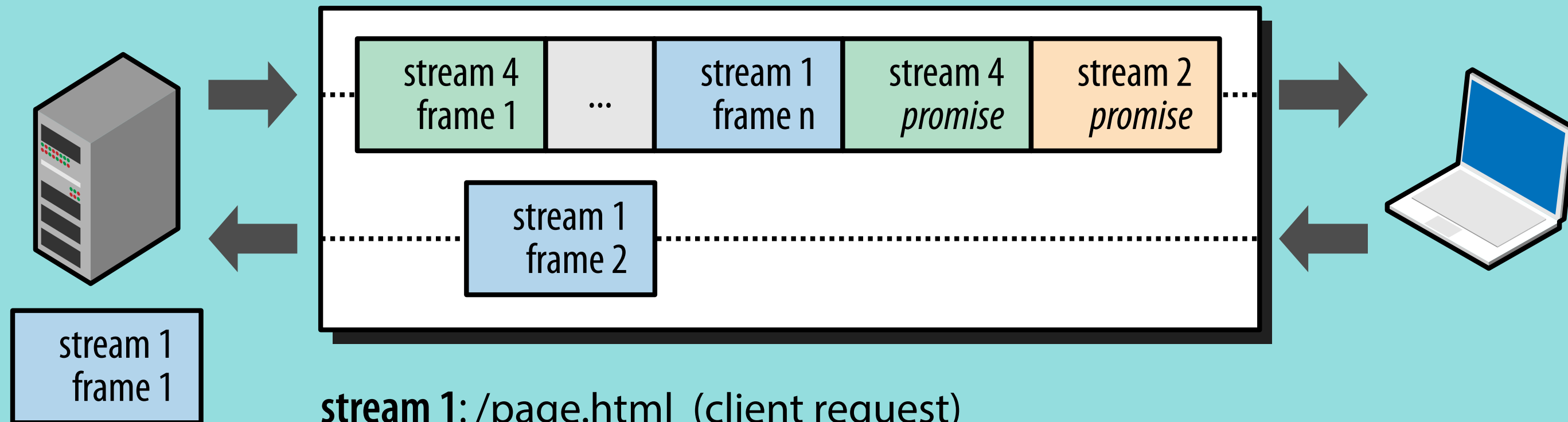


Fonte: Introdução a HTTP/2. Google. Disponível em: <<https://developers.google.com/web/fundamentals/performance/http2>>.

Acesso em: 9 de fev. de 2022.

HTTP 2.0 - Server push

HTTP 2.0 connection



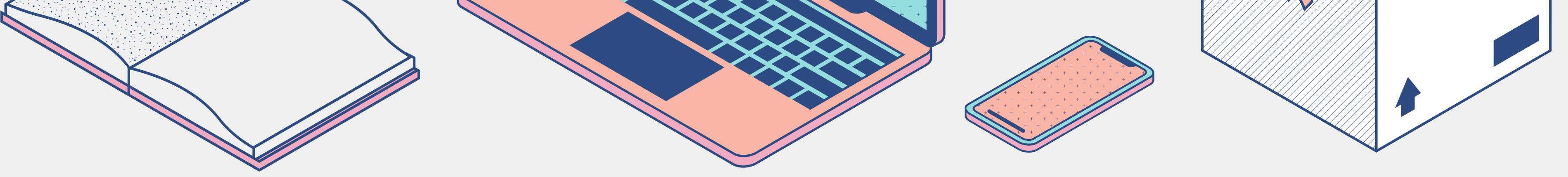
stream 1: /page.html (client request)

stream 2: /script.js (push promise)

stream 4: /style.css (push promise)

Fonte: Introdução a HTTP/2. Google. Disponível em: <<https://developers.google.com/web/fundamentals/performance/http2>>.

Acesso em: 9 de fev. de 2022.



Exemplo

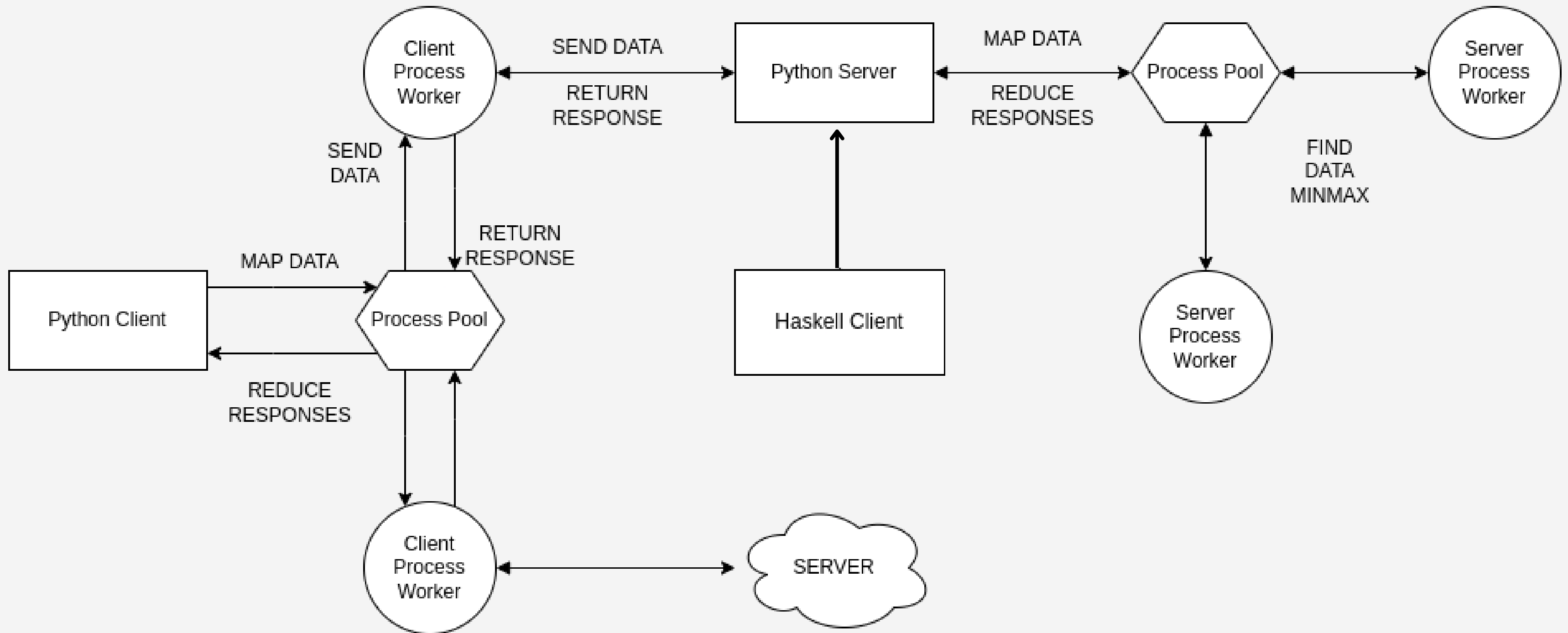
Encontrar o maior e menor número

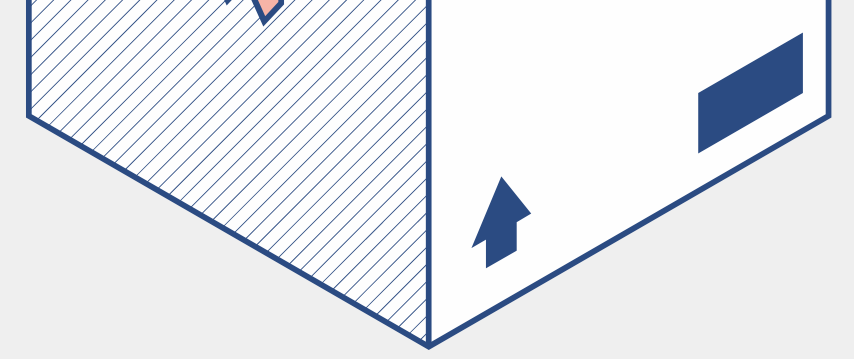
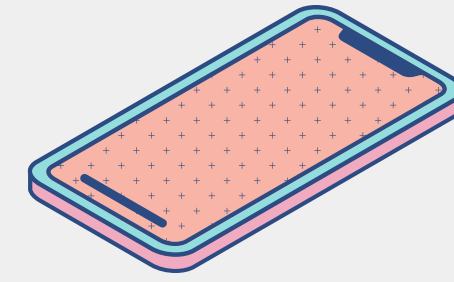
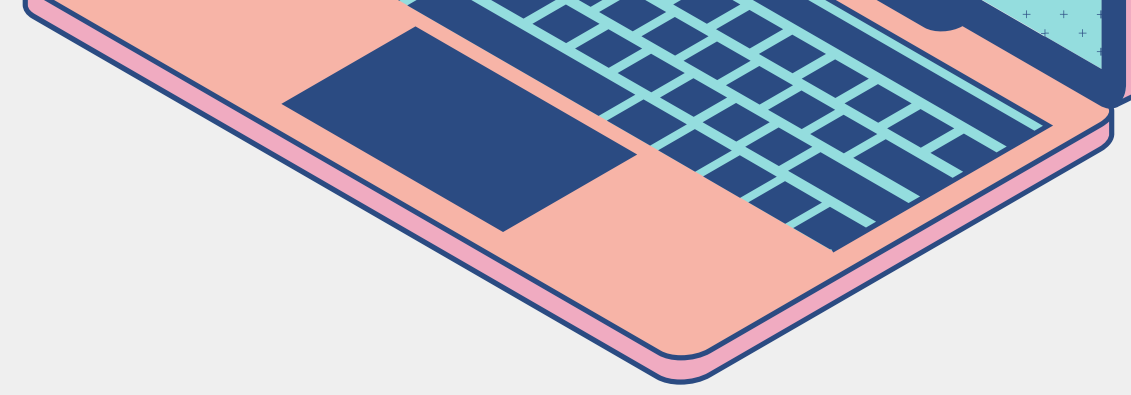
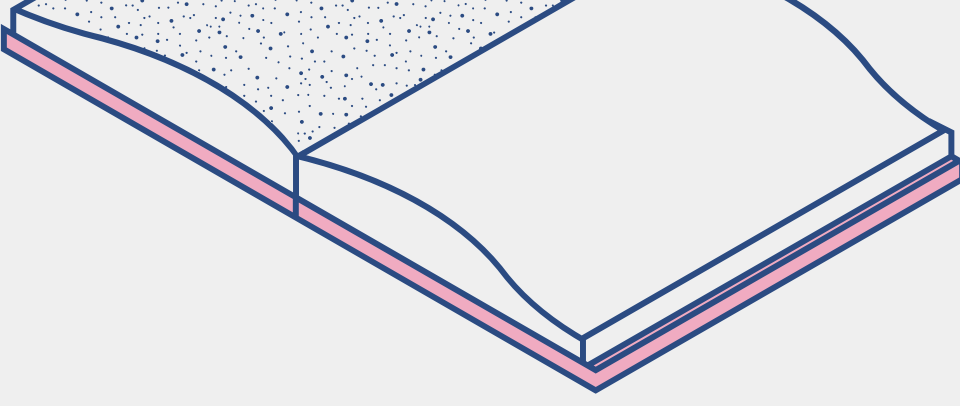
Definição do Serviço



```
syntax = "proto3";  
package minmax;  
  
service MinMax {  
    rpc Find (FindRequest) returns (FindResponse) {}  
}  
  
message FindRequest {  
    repeated float numbers = 1;  
}  
  
message FindResponse {  
    float min = 1;  
    float max = 2;  
}
```


Diagrama da solução





Código



Referências:

Introdução ao HTTP/2. Google. Disponível em: <<https://developers.google.com/web/fundamentals/performance/http2>>. Acesso em: 9 de fev. de 2022.

gRPC, A high performance, open source universal RPC framework. Google. Disponível em: <<https://grpc.io/>>. Acesso em 9 de fev. de 2022.

RPC Features. IBM. Disponível em: <<https://www.ibm.com/docs/en/aix/7.2?topic=call-rpc-features>>. Acesso em 9 de fev. de 2022.

Protocol Buffers. Google. Disponível em: <<https://developers.google.com/protocol-buffers>> Acesso em 9 de fev. de 2022.

ProtoBuf vs JSON. educba. Disponível em: <<https://www.educba.com/protobuf-vs-json/>>. Acesso em 9 de fev. de 2022.