



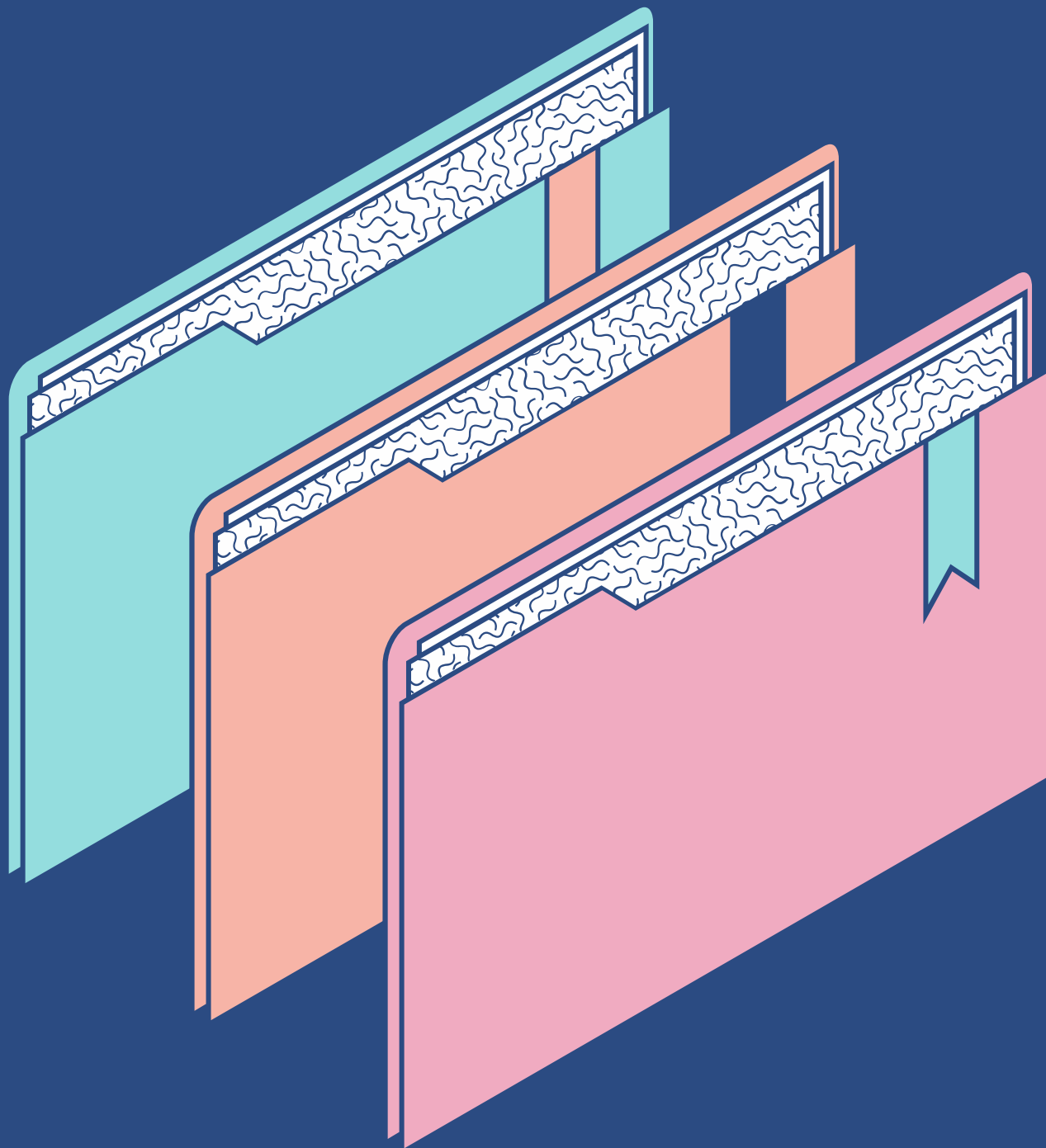
gRPC & ProtoBuf como solução de mensageria

João Pedro Moura Oliveira - 190030879

Matheus Calixto Vaz Pinheiro - 190055201

gRPC

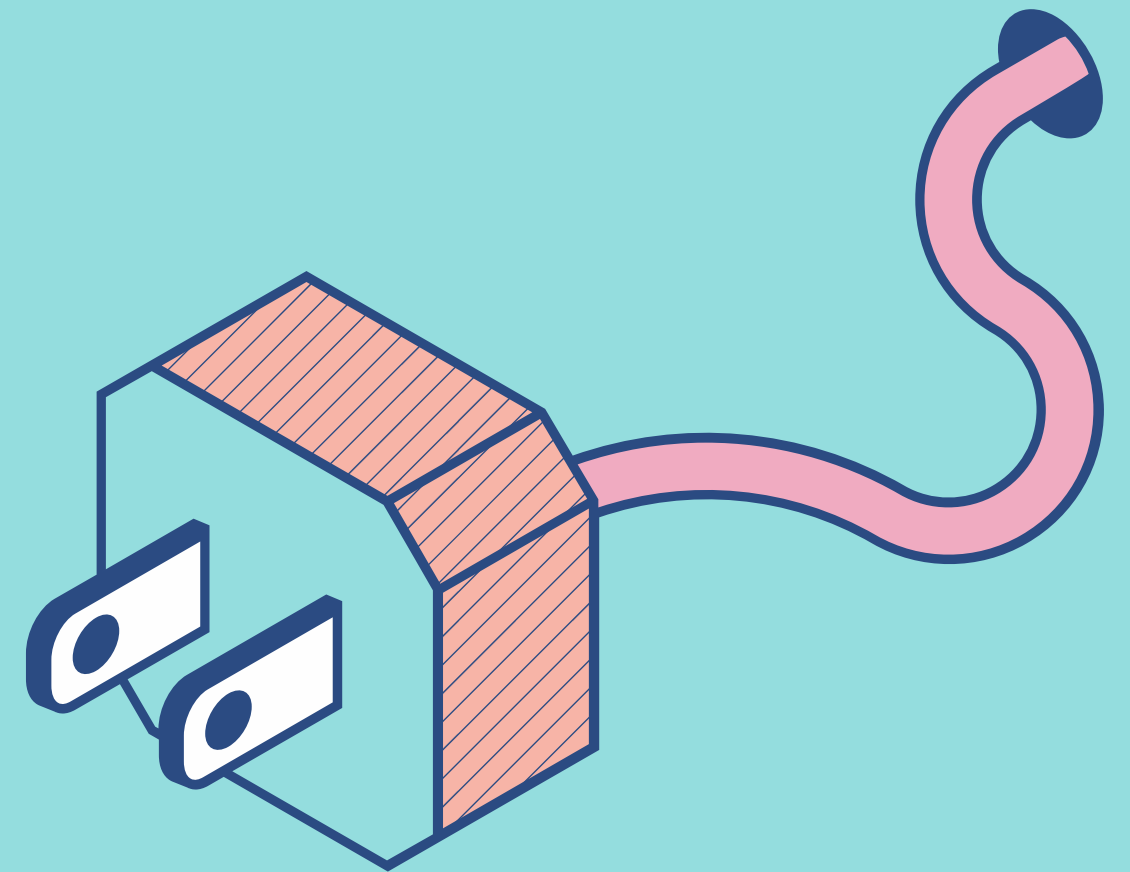
O QUE É?




- 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

Funcionalidades

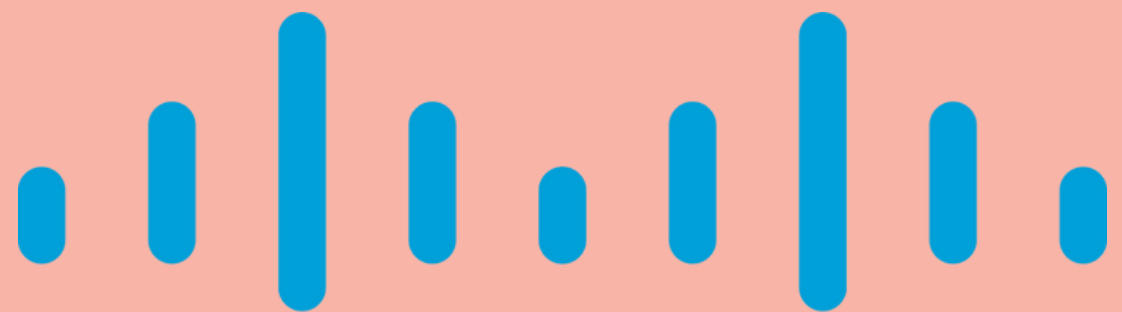
- Chamadas de procedimento remoto em lote
- Chamadas de procedimento em *Broadcast*
- Procedimentos de *Call-back*
- Múltiplos *requests* em uma única conexão TCP
- Permite o sincronismo/assincronismo e a persistência na comunicação



NETFLIX

 Square

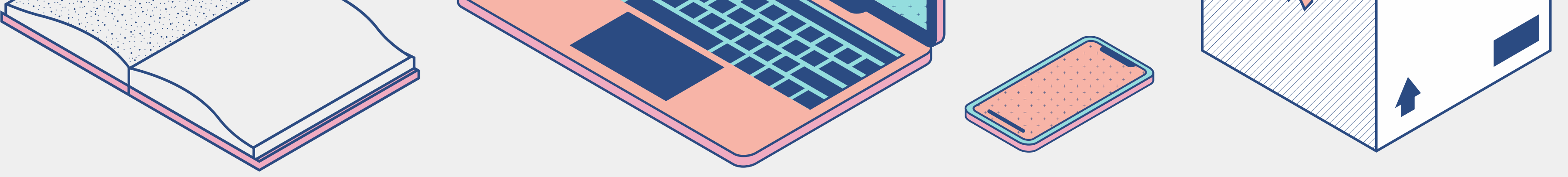
 Core OS



CISCO

Alguns exemplos de aplicações do gRPC

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



Protocol Buffer

Mecanismo extensível de serialização de dados estruturados.

ProtoBuf

Necessita de menos largura de banda

Escrito em binário e incorpora um série de regras para definir e trocar mensagens

Codificação independente de linguagem e plataforma e compatível com JAVA, C, C++, Python, GO, ruby, Kotlin e Dart.

Mais usado para enviar dados entre serviços

Suporta mais tipos de dados

JSON

Necessita de mais largura de banda

Escrito em formato textual legível para humano, não necessitando de regras

Codificação independente e compatível com a maioria das linguagens de programação.

Mais usado em aplicações web

Limitado à alguns tipos de dados

Alguns elementos do ProtoBuf

```
syntax = "proto3";  
import "google/protobuf/empty.proto";  
package exemplo;  
  
service ServExemplo {  
    rpc ExemploOla(MsgExemplo) returns (google.protobuf.Empty);  
}  
  
message MsgExemplo {  
    string nome = 1;  
    singular int32 idade = 2;  
    repeated bool = 3;  
    reserved 2, 10 to 15;  
    reserved "foo", "bar";  
}
```

HTTP/2

Redução de latência

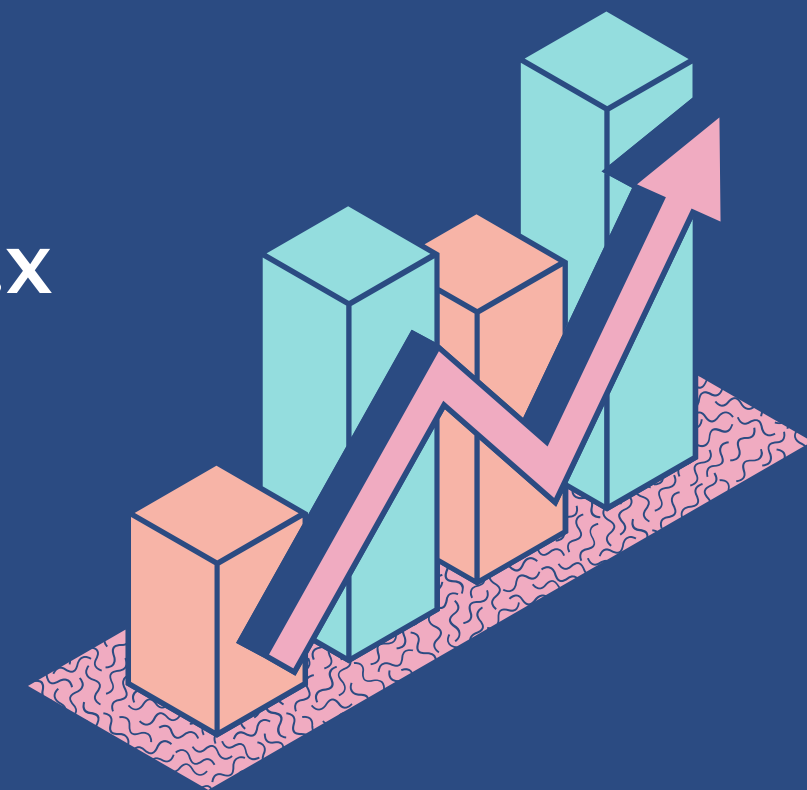
Minimização da carga do protocolo

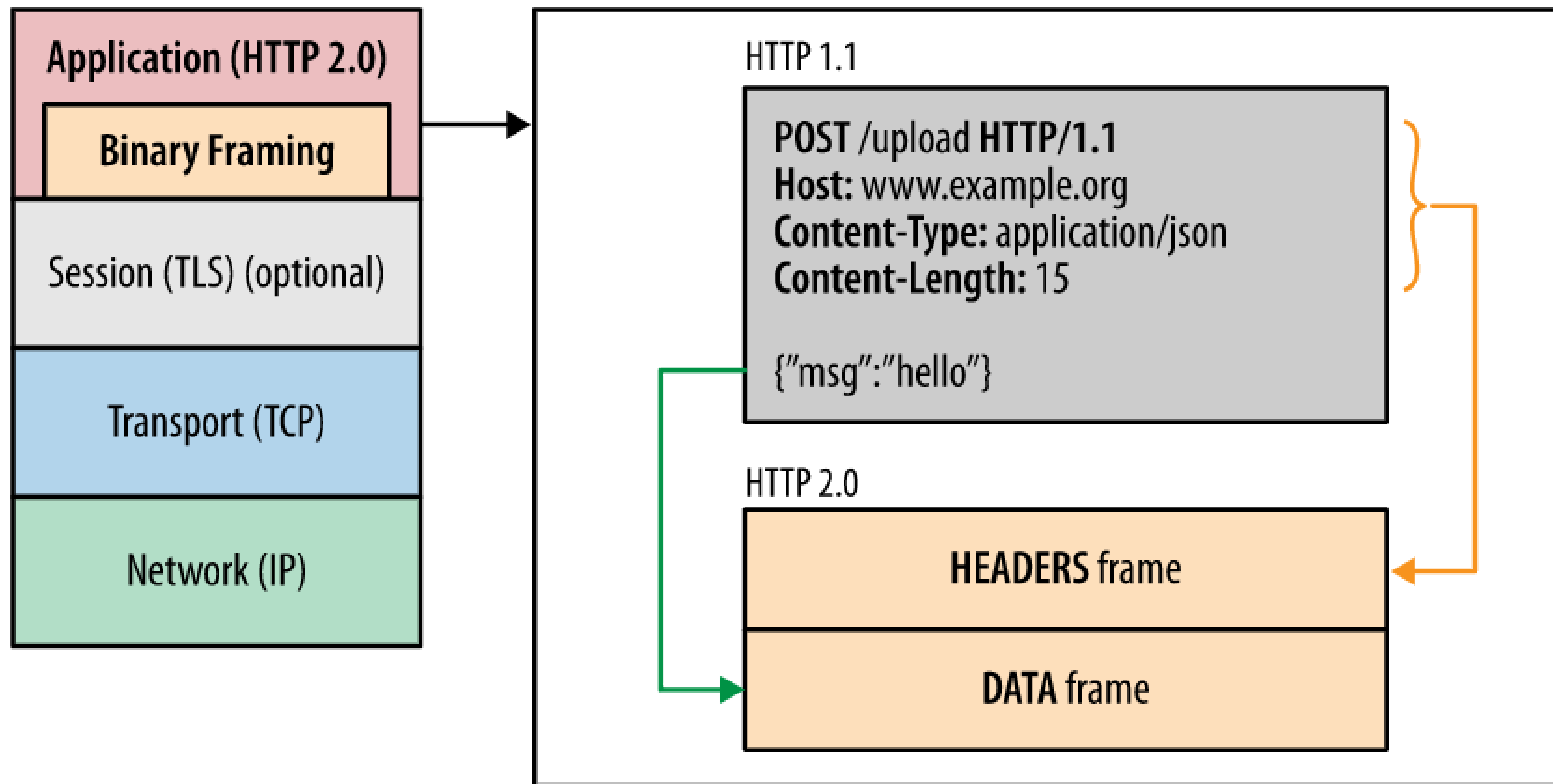
Adição de suporte a priorização de solicitação

Envio push de servidor

Formatação e transporte de dados

2 porque não é retro compatível com versões 1.x





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

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.

gRPC. Devopedia. Disponível em: <<https://devopedia.org/grpc>>. 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.