

gRPC入門



今日話すこと

- gRPCって何？
- gRPCってどういう仕組み？

gRPCって何？



gRPCはRPC(Remote Procedure Call)の一つ

- Googleが開発したRPC

そもそもRPCって？

- RPCはサーバー上に実装された関数(メソッド)をクライアントから実行する仕組み/仕様のこと
- RPCで定められた仕様を満たしていればどんな言語でも開発できる

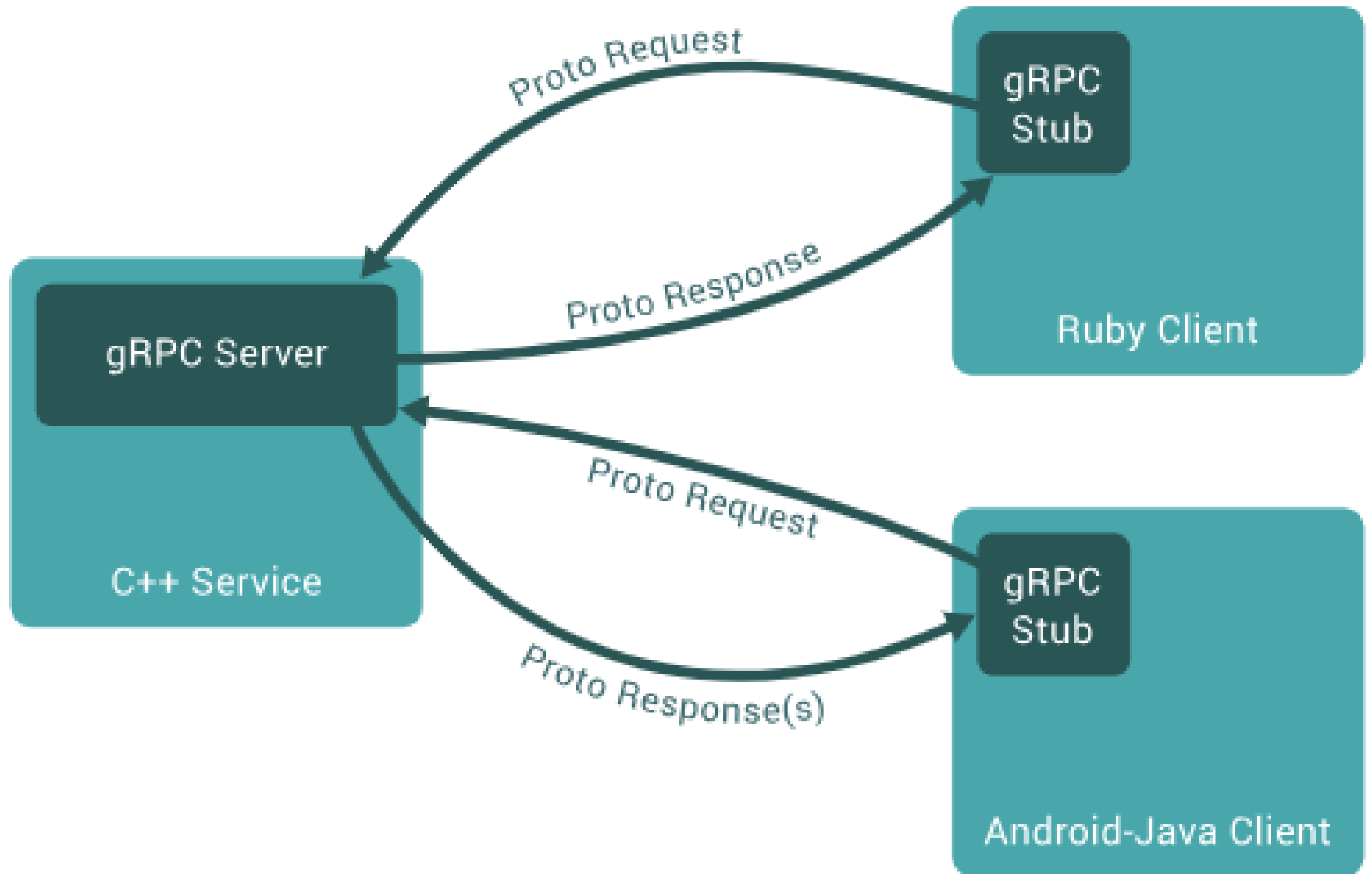
gRPCの特徴

- HTTP/2を前提としていて高速な通信を行える
- ProtocolBuffersを使ってRPCの定義とコードの生成をできる
- Googleをはじめ大規模なサービスで利用されている

gRPC以外に何があるの？

- サーバー間のHTTP通信技術
 - REST: 普段使ってるやつ シンプル
 - GraphQL: サーバーに対してQueryを発行できて柔軟なデータ取得ができる
- RPC
 - JSON-RPC: データのやりとりにJSONを用いる
 - XML-RPC: データのやりとりにXMLを用いる

つまりクライアントとサーバーでのやりとりを行うための仕組み



gRPCってどう
いう仕組み?

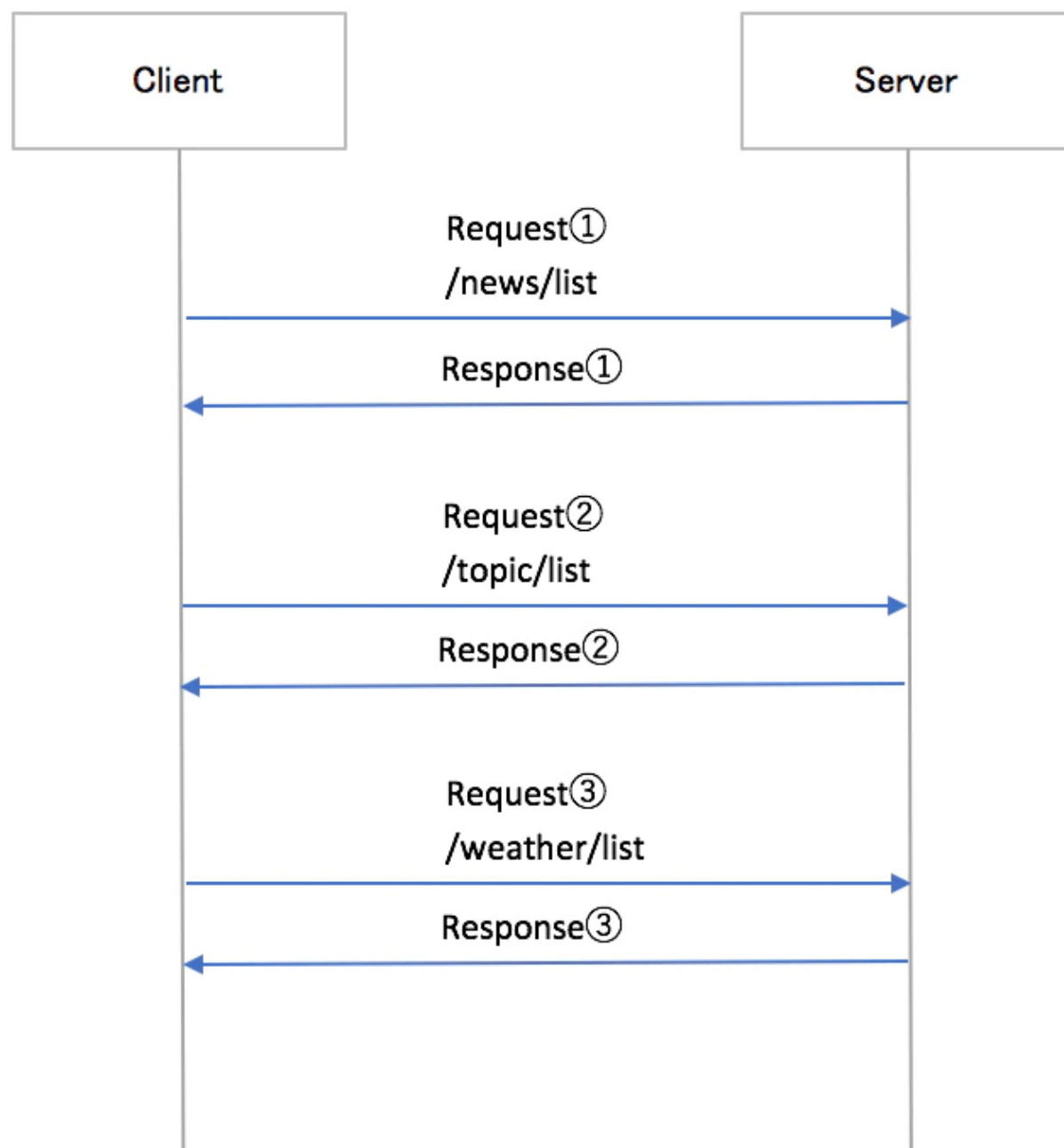


gRPC = HTTP/2 + ProtocolBuffers

HTTP/2による高速な通信

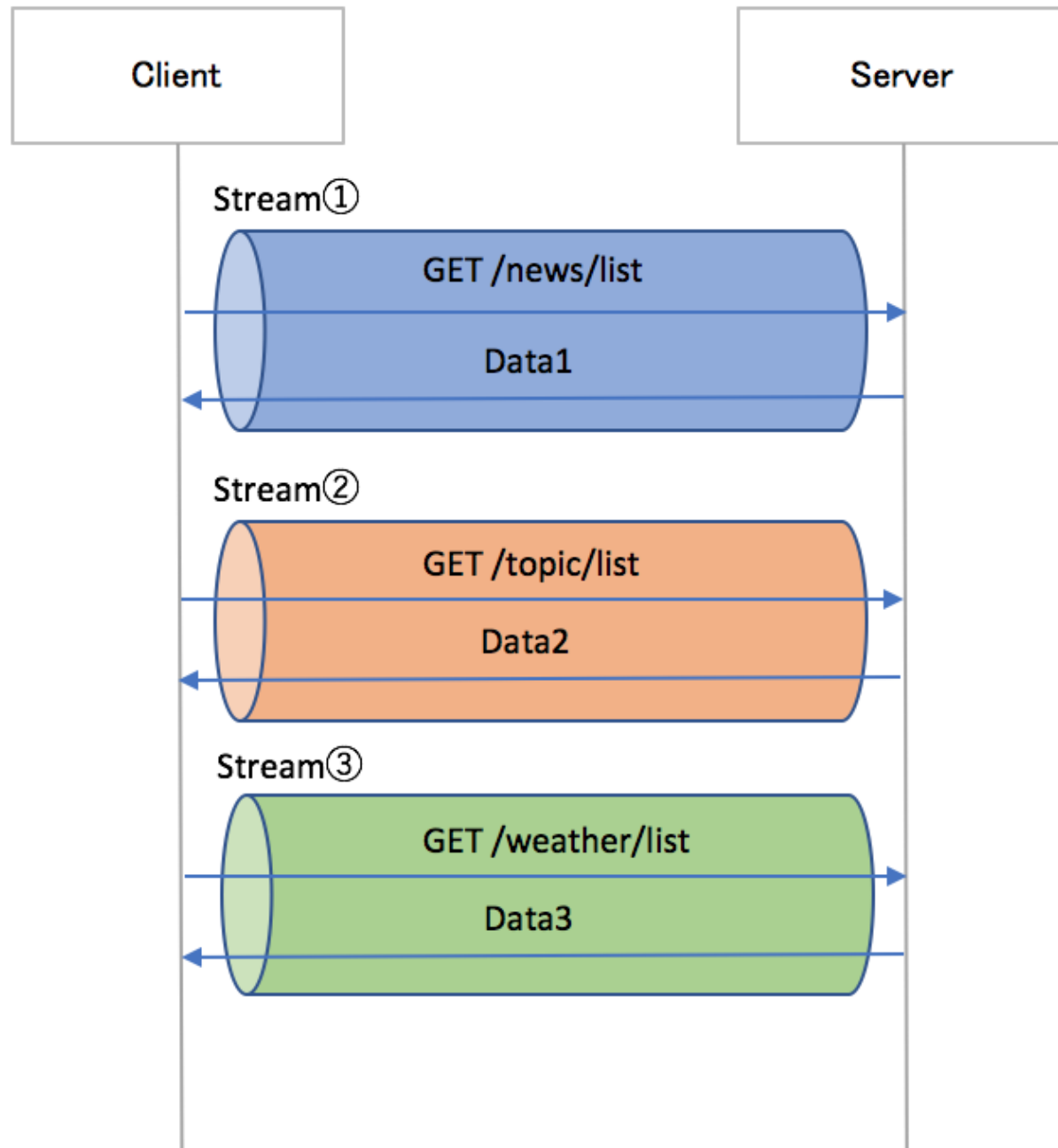
HTTP/1.1の場合...

- レスポンスが返ってくるまで次のリクエストを送れない
- つまりリクエストを並列で実行できない



HTTP/2だと

- ストリームという概念があり同時にリクエストを送受信できる



- さらにHeaderを圧縮できる
 - ハフマン符号化データ圧縮
 - ヘッダーのKey-ValueのIndex化により、以前送信した値は送信データ量を縮小可能

つまり早い



ProtocolBuffers

ProtocolBuffersはスキーマ言語 + データシリアライズ形式

- ProtocolBuffersを使ってRPCの定義を行う

calculator.proto

```
syntax = "proto3";

service Calculator {
    rpc Add(AdditionRequest) returns (AdditionReply);
}

message AdditionRequest {
    int32 added = 1;
    int32 add = 2;
}

message AdditionReply {
    int32 result = 1;
}
```

protoファイルを作ってスキーマの定義を行う

このprotoファイルを元にサーバー/クライアントのコードを生成する

- 公式な対応言語がたくさんある
 - C/C++
 - C#
 - Dart
 - Go
 - Java
 - Kotlin
 - Node
 - PHP(クライアントのみ)
 - 他にも...

ただしブラウザについては完全に対応しているとは言えない...

- grpc-webというJSライブラリがあるが、protobufのシリアライズを扱えないのでjsonでのやりとりになる
 - シリアライズによるデータ量圧縮の恩恵を受けられ無かったり...
 - 一部のgRPC通信方法に対応してなかったり...

シリアライズに関しては利用者が意識しなくていいように隠匿か
されてる

実際にやってみよう

まとめ



- gRPCはクライアント/サーバー間の通信方法の1つ
- protoファイルを使うことで異なる言語間での実装が容易
- HTTP2通信を使うので早い

もっと詳しく知りたい人は電子書籍に入門書があるので見てみてください

