

Inter-Service Communication with gRPC

Niba Tandukar

Date: 14/07/2023



Coverage

- Introduction
- How gRPC works
- Why gRPC use Protocol Buffer?
- Types of gRPC
- Typical use cases for gRPC
- Demo



The motivation of gRPC

Communication between different languages

- Back-end and front-end are written in different languages
- Micro-service might be written in different languages
- They must agree on the API contracts to exchange information

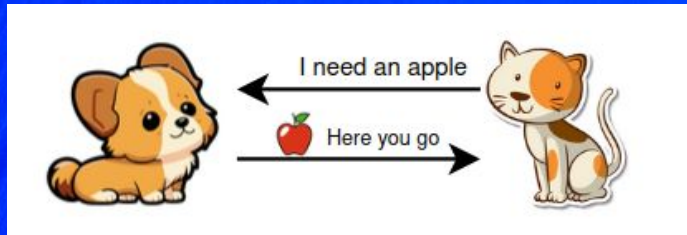


Communication should be efficient

- Huge amount of exchange messages between different micro-services
- Mobile network can be slow with limited bandwidth

Communication should be simple

- Client and server should focus on their core service logic
- Let the framework handle the rest



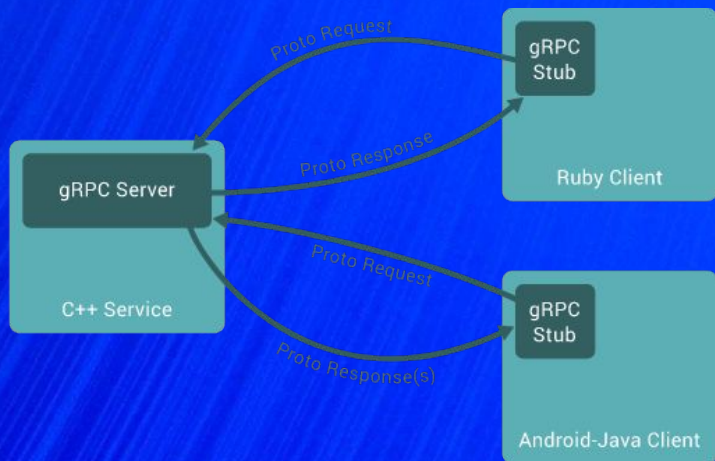
Introduction

- gRPC is a high performance open-source feature-rich RPC framework.
- gRPC is originally developed by Google.
- Now it is a part of the CNCF.
- g stands for different things in each gRPC release: such as good, green, glorious, game,



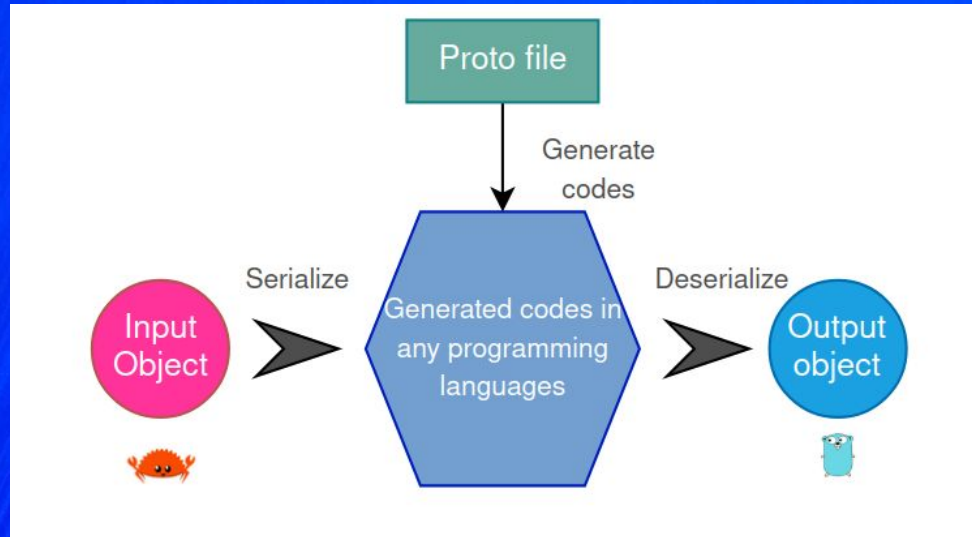
How gRPC works?

- Client has a generated stub that provides the same methods as the server.
- The stub calls gRPC framework under the hood to exchange information over the network.
- Client and server use stubs to interact with each other, so they only need to implement their core service logic.

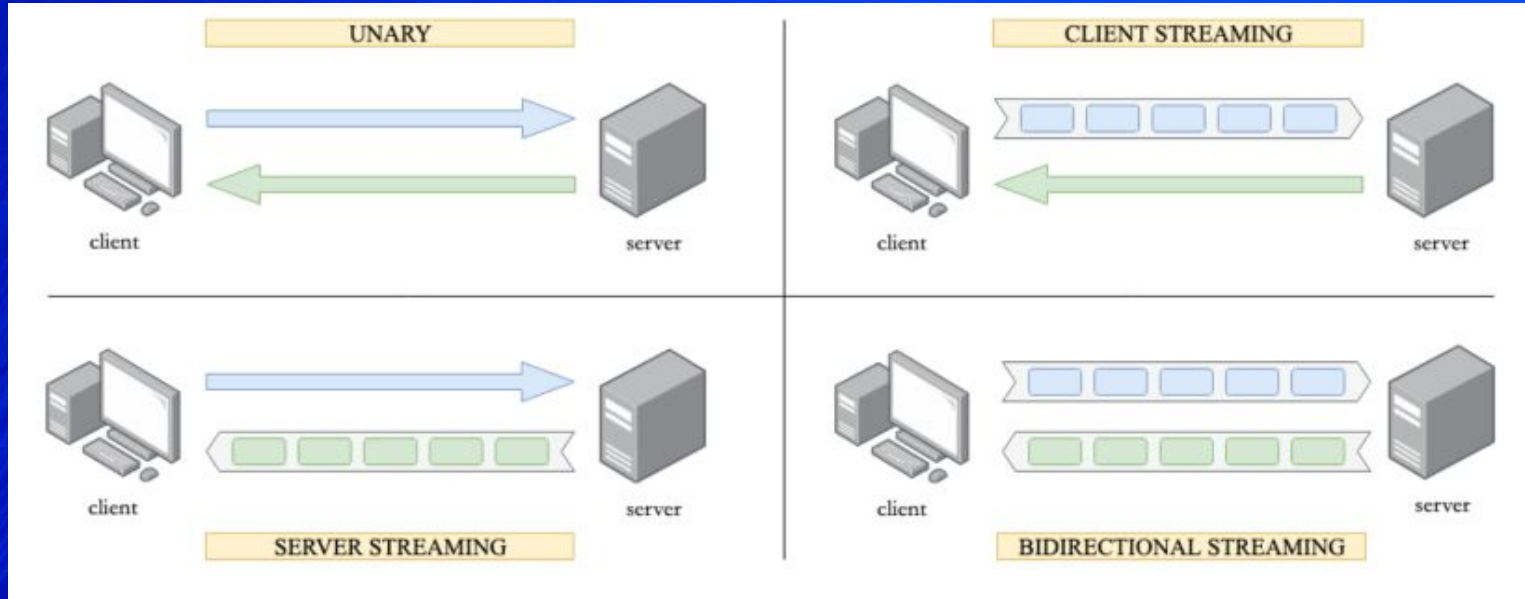


Why gRPC use Protocol Buffer?

- Human-readable interface Definition Language(IDL)
- Programming Language interoperable
- Binary data representation
- Strongly typed contract
- Conventions of API evolution
- Alternative options: Google flatbuffers, Microsoft bond



Types of gRPC



Typical use cases for gRPC

- Microservices
- Distributed systems
- IoT (Internet of Things)
- Real-time applications



Thank you

