Microservices with gRPC

Luka Obradovic • 07.06.2017

Overview

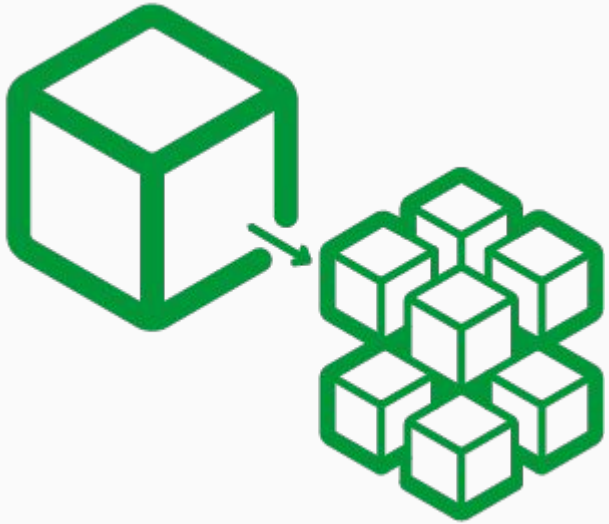
- **Motivation** - Microservices
- **RPC**
- **Protobuf**
- **gRPC**
- **Demo**

What is gRPC ?

A high performance, open-source, universal RPC framework that relies on Protocol Buffers

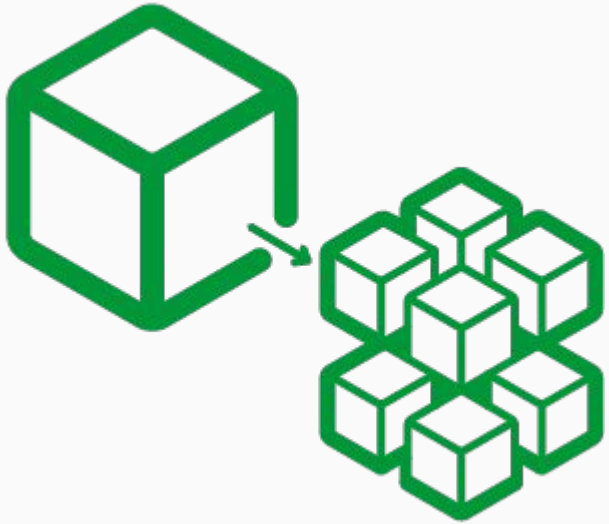
Motivation - Microservices

- main() => classes => libraries



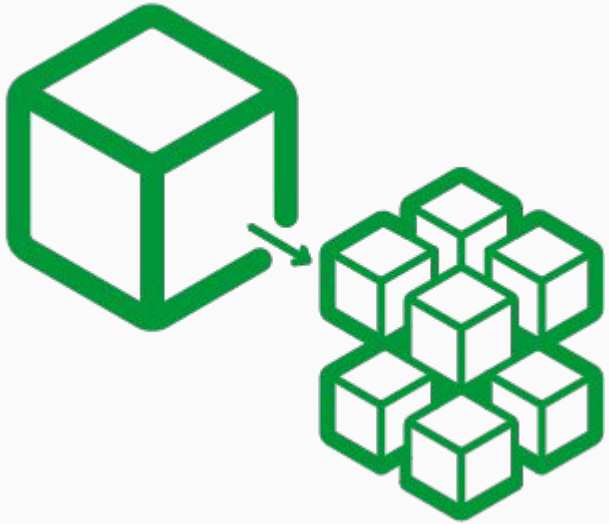
Motivation - Microservices

- main() => classes => libraries
- Easier to maintain



Motivation - Microservices

- main() => classes => libraries
- Easier to maintain
- Easier to deploy



General - RPC

- **R**emote **P**rocedure **C**all

General - RPC

- **R**emote **P**rocedure **C**all
- Stubs

General - RPC

- **R**emote **P**rocedure **C**all
- Stubs
- Like calling a local method

General - RPC

- **R**emote **P**rocedure **C**all
- Stubs
- Like calling a local method

- **I**nterface **D**escription **L**anguage

General - RPC

- **R**emote **P**rocedure **C**all
- Stubs
- Like calling a local method

- **I**nterface **D**escription **L**anguage
- Strictly typed

General - RPC

- **R**emote **P**rocedure **C**all
- Stubs
- Like calling a local method

- **I**nterface **D**escription **L**anguage
- Strictly typed
- Can use other protocols

Protobuf

- Serialising mechanism

Protobuf

- Serialising mechanism
- Schema Language

Protobuf

- Serialising mechanism
- Schema Language
- Language neutral

Protobuf

- Serialising mechanism
- Schema Language
- Language neutral
- Binary, not text

Protobuf

- Serialising mechanism
- Schema Language
- Language neutral
- Binary, not text
- Backwards-compatibility

Protobuf

- Serialising mechanism
- Schema Language
- Language neutral
- Binary, not text
- Backwards-compatibility
- Less Boilerplate Code

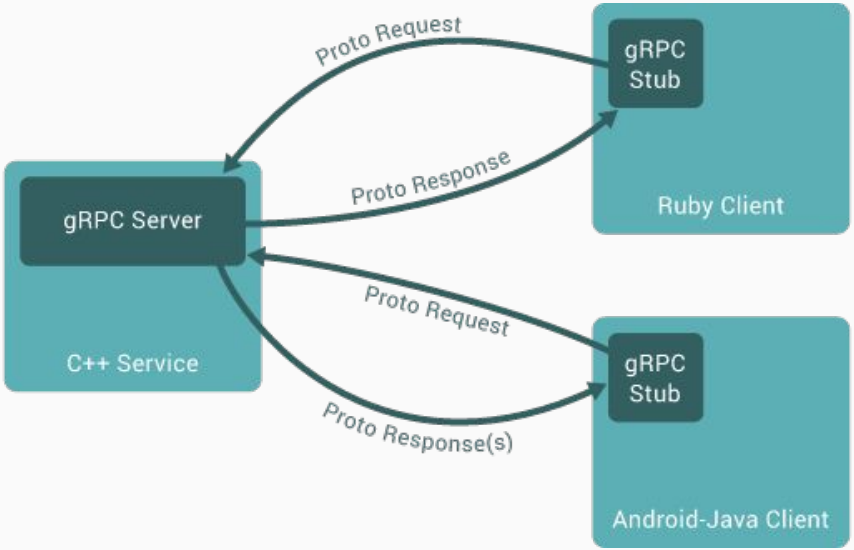
Protobuf [example]

```
message Person {
  string name = 1;
  string family_name = 2;
  int32 id = 3;

  enum Sex {
     MALE = 0;
     FEMALE = 1;
  }

  Sex sex = 4;
}
```

```
final Person person =
    Person
        .newBuilder()
        .setId(12345)
        .setName("John")
        .setFamilyName("Doe")
        .setSex(Person.Sex.MALE)
        .build();

person.writeTo(new FileOutputStream(args[0]));
```

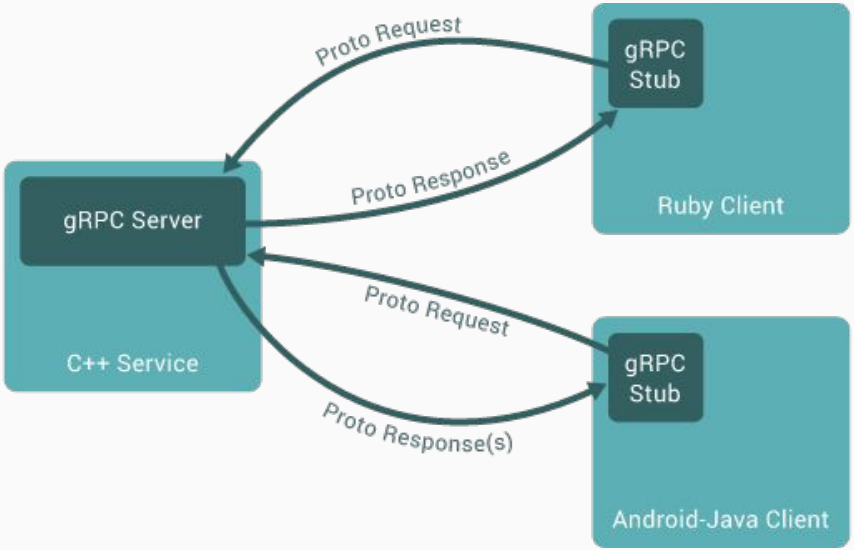
gRPC

- Protobuf to describe the service (IDL)



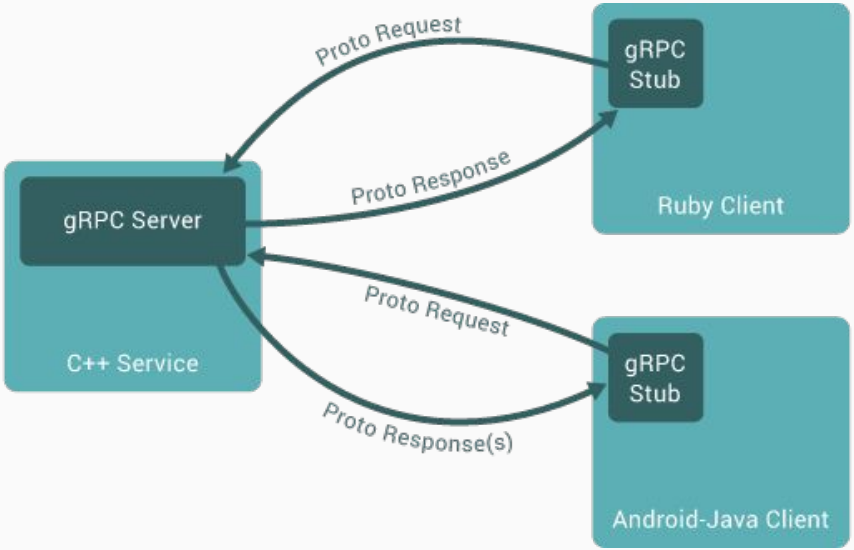
gRPC

- Protobuf to describe the service (IDL)
- Protobuf for messaging



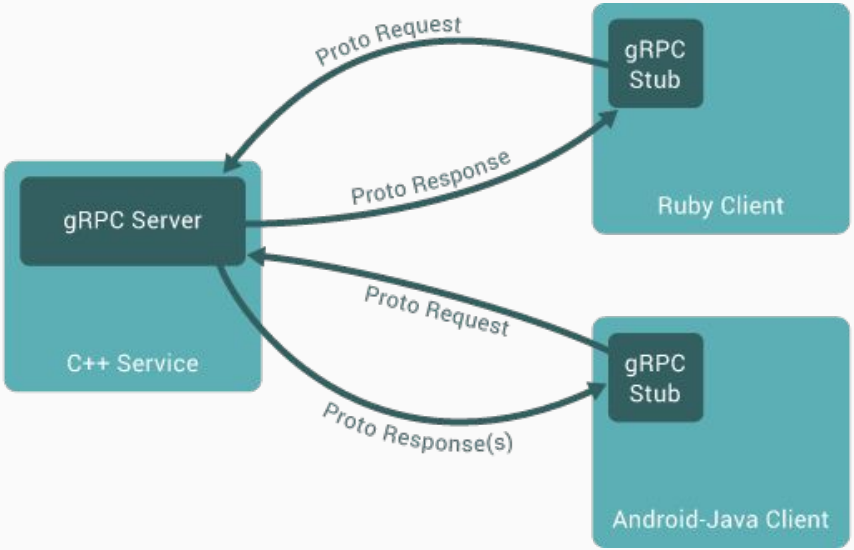
gRPC

- Protobuf to describe the service (IDL)
- Protobuf for messaging
- Generated server and client code



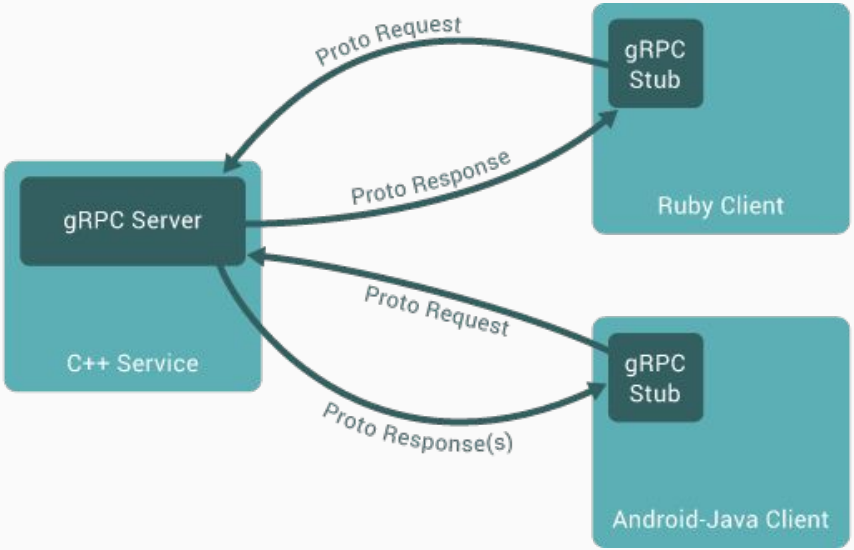
gRPC

- Protobuf to describe the service (IDL)
- Protobuf for messaging
- Generated server and client code
- HTTP/2



gRPC

- Protobuf to describe the service (IDL)
- Protobuf for messaging
- Generated server and client code
- HTTP/2
- Bi-directional streaming



gRPC [example]

```
service Calculator {
   rpc Add (AddRequest) returns (AddResponse) {

   }
}

message AddRequest {
   int32 x = 1;
   int32 y = 2;
}

message AddResponse {
   int32 z = 1;
}
```

Demo