

Chapter 05: gRPC

gRPC เป็น **RPC Framework** ที่ใช้ในการสื่อสารใน distributed system

gRPC

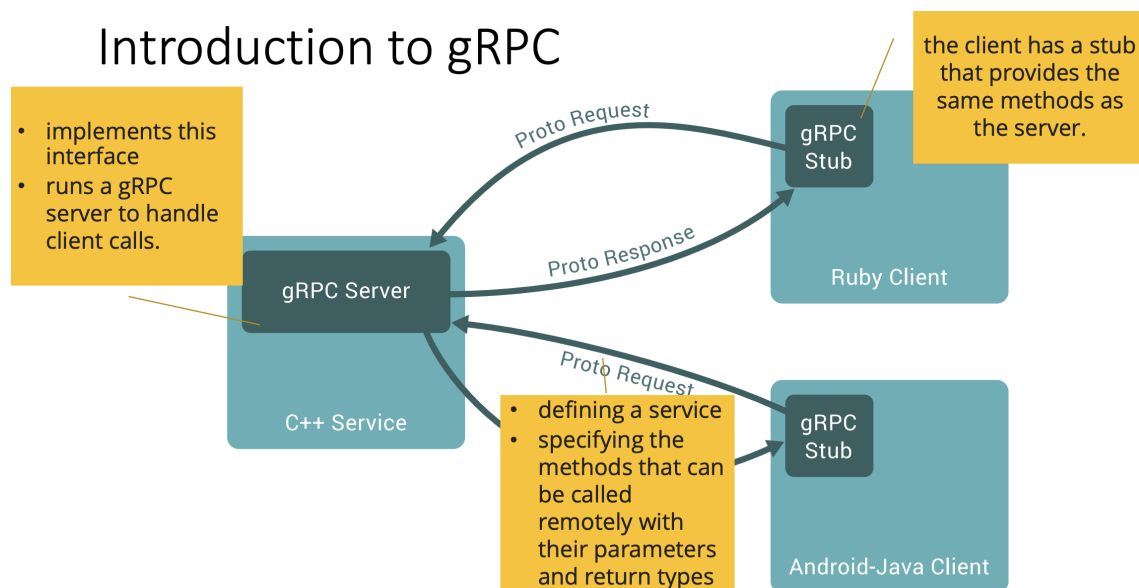
gRPC Server >> ผู้ให้บริการ มีการ implement service ไว้ แล้วให้ Client เรียกใช้

gRPC Client >> ผู้ใช้บริการ

- **gRPC Stub** >> โปรแกรมเล็ก ๆ ที่จำลอง และแปลงการเรียกใช้ service ของ Client ให้กลายเป็น **Proto Request** พร้อมทั้งรับ **Proto Response** จาก Server
 - เวลา Client ติดต่อกับ Server จะทำให้ Client ไม่ต้องรู้ข้อมูลอะไรเกี่ยวกับ Server เลยก็ได้ เพียงแค่เรียกใช้ service เหมือนเป็น service ที่อยู่ในเครื่องของตัวเองเลย
- ก่อนที่ Client จะคุยกับ Server ได้ จะต้องมีการ **define service** ก่อน (เพื่อให้ Client จะได้ว่ามี method อะไร parameter/return type มีอะไรบ้าง)
- define service แล้วก็ต้องมีการระบุ **parameter, return types** ด้วย

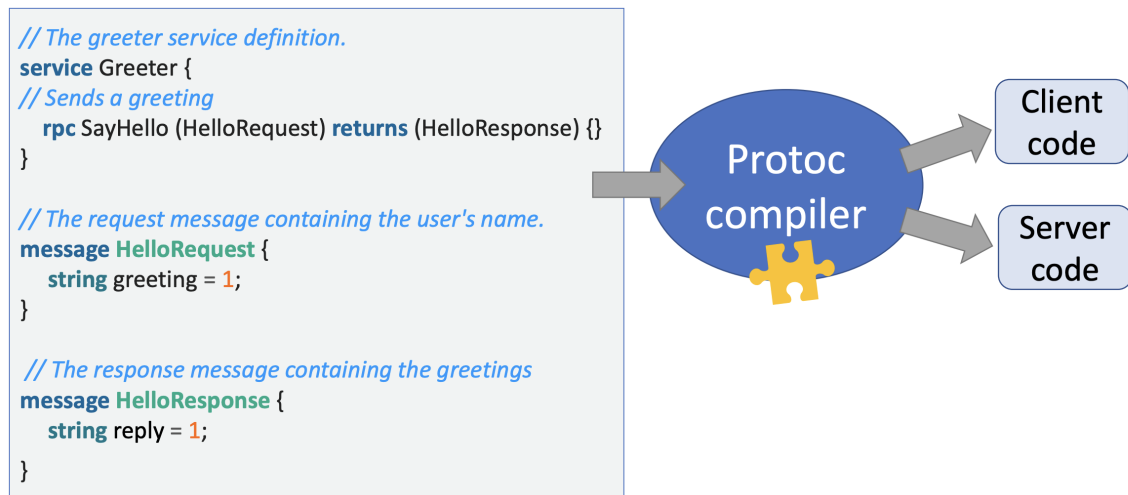
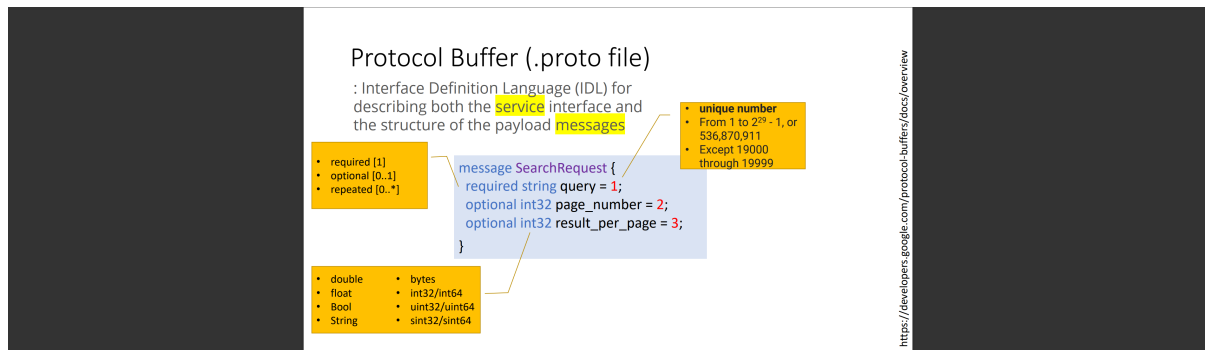


สังเกตว่าทั้ง Client และ Server แต่ละเครื่อง **ไม่จำเป็นต้องเขียนภาษาเดียวกัน**



Protocol Buffer

- **.proto file** >> คือไฟล์ที่ใช้ในการสื่อสาร
 - เป็น **Interface Definition Language (IDL)** ซึ่งคือภาษาที่ใช้สื่อสารที่ระบุ **service** และ **message**
- **message** >> ชนิดของข้อมูล (types ของข้อมูล, required/optional/repeated, unique number)
- **service** >> ระบุ **method** ที่อยากให้มี โดยใส่ 'rpc' ไว้ข้างหน้าชื่อ method
- **protoc** (protocol compiler) >> เมื่อระบุ message แล้ว ก็ต้องนำไปผ่าน protoc เพื่อ **compile message** ให้กลายเป็น class



4 Kinds of gRPC Service Method

แบ่งตามลักษณะการส่งข้อมูลระหว่าง client - server (การันตีลำดับของ message)

1. **Unary RPCs** >> ทั้งคู่ส่งเป็น message เดียว ๆ
2. **Server Streaming RPCs** >> client ส่งเป็น message เดียว ๆ server ส่งกลับเป็น data stream
3. **Client Streaming RPCs** >> client ส่งเป็น data stream server ส่งกลับเป็น message เดียว ๆ
4. **Bidirectional streaming RPCs** >> ทั้งคู่ส่งเป็น data stream