

# Go,Java,C# 的 GRPC + Protocol Buffer 共同大亂鬥

# gRPC

- 遞迴命名法的”RPC”框架：**gRPC Remote Procedure Calls**
- Google與Square公司共同開發後以Apache 2.0授權開源
- [HTTP/2](#)為傳送媒介(也只有支援HTTP/2的環境才能用)
- 可跨語言呼叫，目前支援：
  - Go (Server & Client)
  - Java (Server & Desktop, Android Client)
  - C# (Server & Desktop)
  - C++, PHP, Node.js, Python, Ruby, Objective-C, Swift, Haskell等等...
  - 瀏覽器端的Js要再看看(HTTP/2支援程度問題)
- 使用場景：
  - Microservice 內部服務互相呼叫([etcd](#), [Kubernetes](#), [Tensorflow](#), [proto.actor](#) [平行運算框架](#))
  - Mobile Device和Server端交換資料、Client/Server Streaming
  - GCP(Google Cloud Platform) 提供的雲端API

# Protocol Buffer

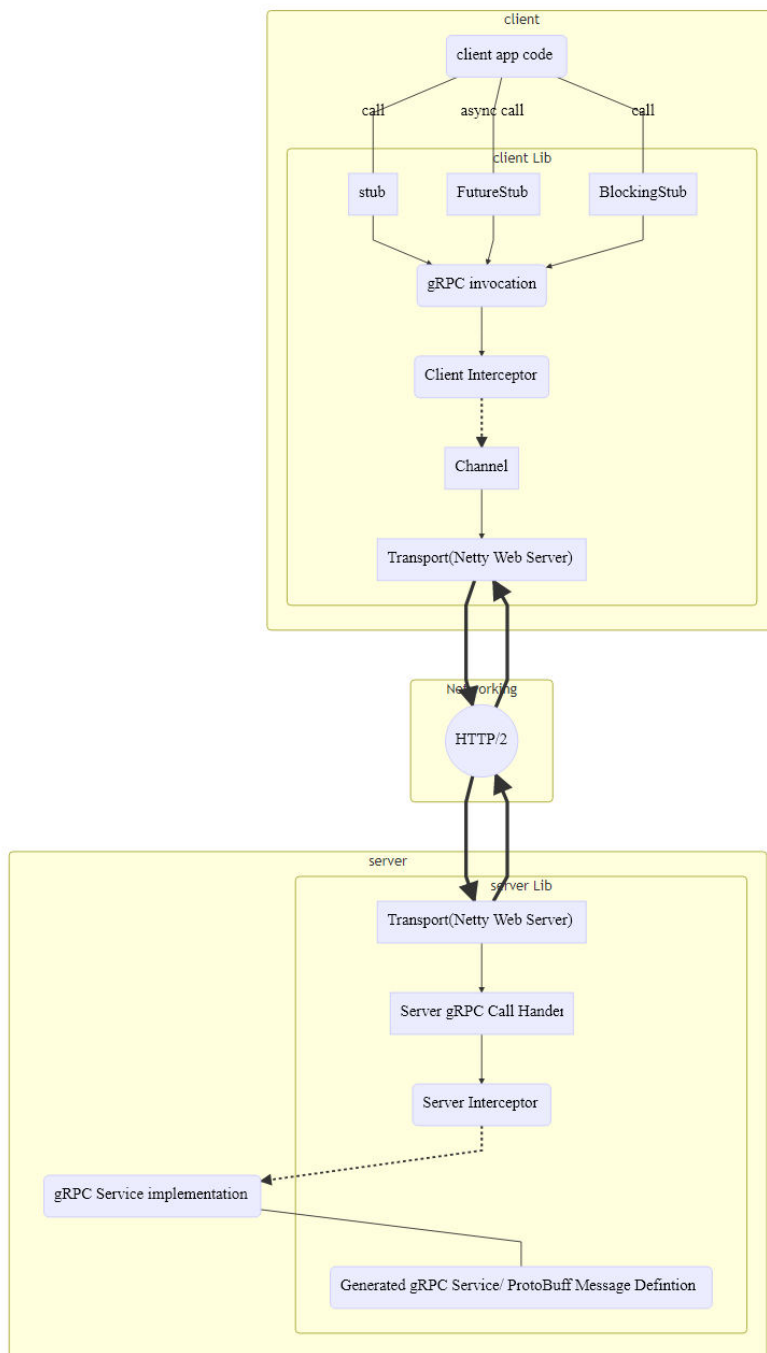
- 一種『序列化』、『反序列化』資料的方法
- 由Google開發用來做內部伺服器快速傳輸資料之用，後來開源出來
- 可跨語言、跨平台使用，官方提供一些[共通資料結構定義](#)
- 資料型態的定義靠 \*.proto的DSL(Domain Specific Language)檔案  
然後再用“protoc”命令列code gen出各種語言可直接使用的程式碼
- gRPC方法的定義也是寫在.proto檔內，一起code gen產生stub物件

# gRPC函式庫常見術語概念解釋

gRPC server/client端官方支援多種程式語言，雖然各種程式語言寫法有所不同，但在函式庫以及文件之中會有幾種共通概念術語：

- **Channel**
- **Deadline/Timeout**
- **MetaData/Context**
- **Streaming**
- **Interceptor**

相關說明可參考[官方文件](#)和[這個](#)



一張圖看懂  
(JAVA為例)

# 開發流程

## 1. 建立敘述gRPC服務的 \*.proto 檔案

- Style guide: <https://developers.google.com/protocol-buffers/docs/style>

## 2. 從 \*.proto 檔產生出配合使用的程式語言的原始碼檔案

- Go 產gRPC程式碼的[protoc plugin](#)要額外安裝。
- Java [gradle plugin](#)套用時仍需建置環境可直接指令列呼叫protoc
- C#安裝[Grpc.Tools Nuget](#)套件有Linux/Mac/Windows現成可用protoc執行檔

產生的程式碼依功能分三種：

- **Request/Response Model :**

Protocol Buffer定義的訊息格式物件建立和屬性存取方法。

- **gRPC Server Code :** gRPC呼叫方法的Interface定義或Abstract Class

- **gRPC Client Code :** 可在客戶端使用的gRPC呼叫用Proxy物件或Stub方法

# 開發流程(續)

## 3.Server端實作gRPC Server Code

- gRPC呼叫方法的Interface定義或是覆寫Abstract Class定義的方法
- gRPC Server物件建立&設定監聽port的code整合至系統運作生命週

## 4.Client端實作gRPC呼叫

- 設置channel連接、加密等屬性設定
- 建立Stub物件呼叫方法執行gRPC呼叫

# .proto檔內容改來改去的相容性

新舊.proto檔內容所產生的程式碼相容的準則：

- 命令列工具protoc的版本儘量一致
- .proto檔定義的資料結構欄位tag號碼只新增、不要更動既有的
- gRPC定義方法名稱和實作Server端新增給新版Client端使用的、保留舊版



# DEMO

[Android Java client + Go Server使用有稍許差異的proto檔](#)

# GCP Speech To Text API

- 提供各種語言的語音辨識成文字的雲端服務(當然還是英文最準)  
<http://cloud.google.com/speech/>
- 提供下列三種使用方式：
  - 1 min語音檔案同步產生辨識結果(REST , gRPC)
  - 80 min語音檔案非同步產生辨識結果(REST, gRPC)
  - 串流即時收音即時辨識的gRPC bi-directional streaming

# DEMO

串流語音辨識Go Server + 遠端 .net Core C# 存word檔

[http://github.com/windperson/coscup2017\\_grpc\\_proto](http://github.com/windperson/coscup2017_grpc_proto)

[http://github.com/windperson/coscup2017\\_grpc\\_golang](http://github.com/windperson/coscup2017_grpc_golang)

[http://github.com/windperson/coscup2017\\_grpc\\_csharp](http://github.com/windperson/coscup2017_grpc_csharp)

Q&A