

2019.10.31 CRL STUDY GROUP

GRPC



CRL Study Group By Schnee

# THE STARTUP GUIDE

---

## TOPICS TO BE COVERED

1. Preface
2. Protobuf
3. GRPC
4. GRPC vs HTTP API
5. GRPC Advantage
6. GRPC Disadvantage
7. GRPC Install
8. GRPC DEMO

# PREFACE

## 1.RPC(REMOTE PROCEDURE CALL)

- 1.是計算機通訊協定。該協定允許執行於一台計算機的程式呼叫另一個位址空間（通常為一個開放網路的一台計算機）的子程式
- 2.RPC是一種伺服器-用戶端（Client/Server）模式，經典實現是一個通過傳送請求-接受回應進行資訊互動的系統。

## 2.GRPC

- 1.簡單來說，就是Google自己推動的一個RPC的協議，他會由Protobuf檔案產生一個伺服器出來，而這個好處，就是你不再需要寫一堆路由。

# PROTOBUF

- 1.全名為：Protocol Buffers
- 2.開發者：Google
- 3.初版發行時間：2008年7月7日
- 4.穩定版本：version3.7.1(2019年3月26日)
- 5.只需定義一次資料結構，自動生成符合你程式語言的檔案
- 6.結構就是文件
- 7.語言支援：
  - (1)proto2：提供一個程式產生器，支援C++、Java和Python  
第三方實作支援JavaScript
  - (2)proto3：提供一個程式產生器，支援C++、Java (包含JavaNano)、Python、Go、Ruby、Objective-C、C#，從 3.0.0 Beta 2 版開始支援JavaScript。  
第三方實作支援Perl、PHP、Scala和Julia
- 8.型態參考：<https://developers.google.com/protocol-buffers/docs/proto3>

# PROTOBUF

```
1  //撰寫格式為 Proto V3
2  syntax = "proto3";
3
4  // User 帶有使用者資料，如帳號、密碼。
5  message User {
6      |   string username = 1;
7      |   string password = 2;
8  }
9
10 //Post 帶有文章編號及內容
11 message Post {
12     |   int64 id = 1;
13     |   string content = 2;
14 }
15
16 //需定義好資料型態及資料
17 //數值代表Protobuf在編碼及解碼時，所使用的編號
18 //這代表即使你移除了其中一個欄位，也不會打亂整個資料結構的編碼與解碼
```

# PROTOBUF

## 1.與JSON的比較，優點

(1)資料輕量化：資料非常輕量，省去了不必要的 { 或 : 累贅。

(2)混淆性：在一般人眼中無法輕易地猜測出資料結構為何，因為有經過編碼。

(3)效能高：處理速度很快。

(4)極具方便性：結構就是你的資料模型，你能夠直接在程式中使用這些結構，而不用建立新的物件來接納、映射（Mapping）這些資料。

(5)清晰明瞭、無需文件：檔案本身就是你的文件，不需要額外撰寫 API 或結構文件來告訴別人你接受怎樣的資料。

# GRPC

---

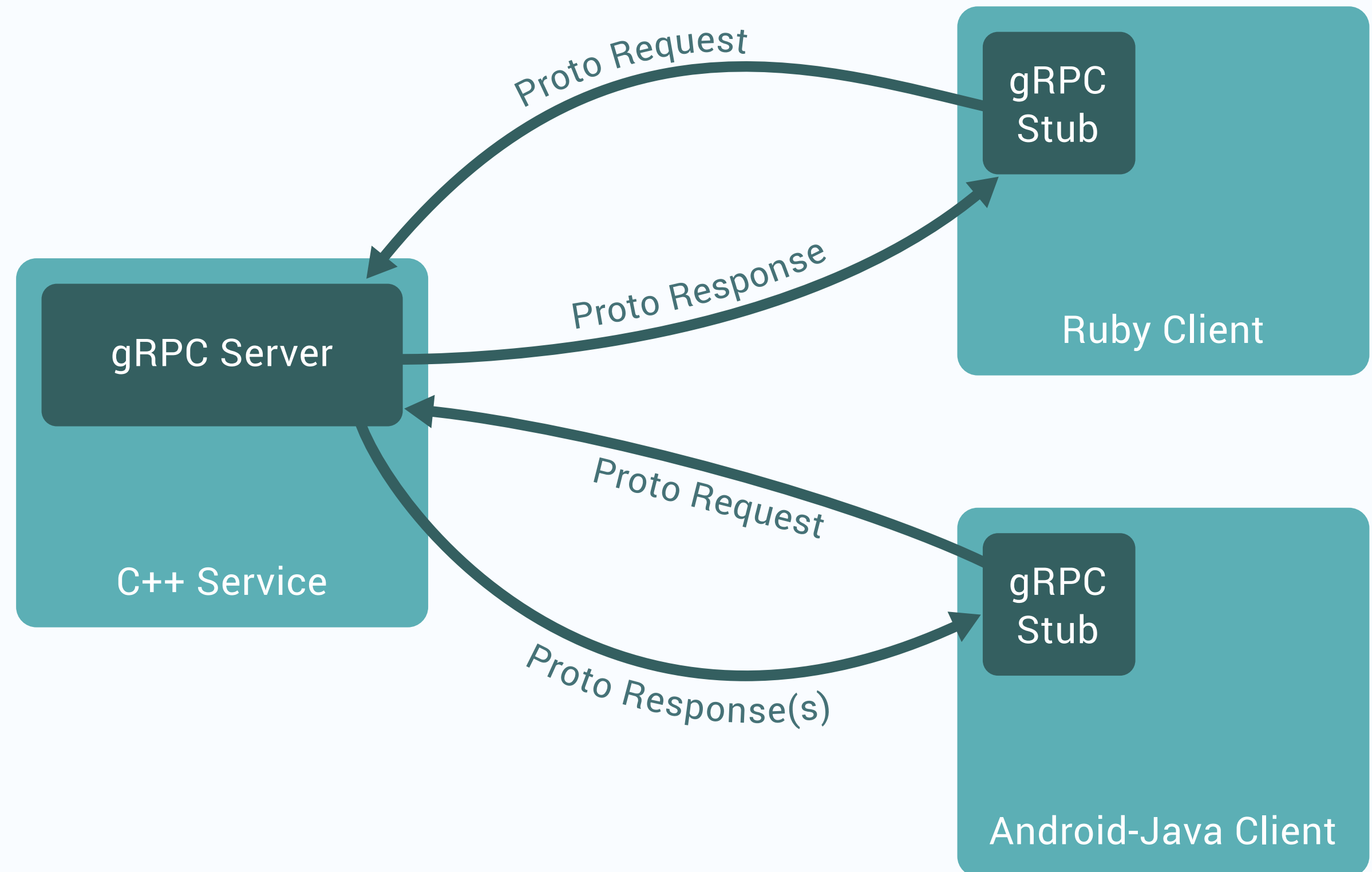
1.開發推行者：Google

2.為CNCF(雲端運算基金會)的育成專案之一  
參考：<https://www.cncf.io/>

3.知名使用者：NETFLIX、CISCO

4.Release：最新版本1.25(2019/10/29釋出)

# GRPC





# GRPC VS HTTP API

功能	GRPC	HTTP API
合約	必要 ( proto )	選擇性 ( OpenAPI )
Transport	HTTP/2	HTTP
承載	Protobuf	JSON
Prescriptiveness	嚴格規格	鬆動. 任何 HTTP 都有效
資料流	用戶端，伺服器，雙向	用戶端，伺服器
瀏覽器支援	否 ( 需要 grpc-web )	是
安全性	傳輸 ( HTTPS )	傳輸 ( HTTPS )
用戶端程式代碼產生	是	OpenAPI + 協力廠商工具

# GRPC ADVENTAGE

## 1.效能

使用Protocol序列化，是一種有效率的二進位訊息格式，且gRPC是針對HTTP/2所設計，是一種可透過HTTP1.X提供更顯著效能優勢的主要修訂版

- (1)二進位框架和壓縮。HTTP/2通訊協定在傳送和接收時都是精簡且有效率的。
- (2)透過單一 TCP 連線進行多個 HTTP/2 呼叫的多工處理。 多工化可避免行標頭封鎖。

## 2.程式碼產生

藉由共用伺服器 and 用戶端之間的proto檔案，可以從端對端產生訊息和用戶端程式代碼。 用戶端的程式碼產生不會在用戶端和伺服器上排除重複的訊息，並為您建立強型別用戶端。 不需要撰寫用戶端，就能在具有許多服務的應用程式中節省大量的開發時間。

# GRPC ADVENTAGE

## 3.資料流

HTTP/2 提供長時間即時通訊資料流程的基礎。 gRPC 提供透過 HTTP/2 進行串流的第一級支援。

- (1)一元(無串流)
- (2)伺服器對用戶端串流
- (3)用戶端對伺服器的串流處理
- (4)雙向串流

## 4.期限/超時和取消

gRPC 可讓用戶端指定他們願意等待 RPC 完成的時間長度。 期限會傳送至伺服器，而伺服器可以決定在超過期限時要採取的動作。 例如，伺服器可能會在超時時取消進行中的 gRPC/HTTP/資料庫要求。透過子 gRPC 呼叫來傳播期限和取消，有助於強制執行資源使用限制。

# GRPC DISADVANTAGE

## 1. 有限的瀏覽器支援

目前無法從瀏覽器直接呼叫 gRPC 服務。gRPC 大量使用 HTTP/2 功能，而且沒有瀏覽器提供 web 要求所需的控制層級來支援 gRPC 用戶端。

但可以使用 gRPC-Web，gRPC-Web 是由 gRPC 團隊提供的額外技術，可在瀏覽器中提供有限的 gRPC 支援。

gRPC 是由兩個部分組成：支援所有新式瀏覽器的 JavaScript 用戶端，以及伺服器上的 gRPC Web proxy。gRPC-Web 用戶端會呼叫 proxy，而 proxy 會在 gRPC 要求上轉送至 gRPC 伺服器。

## 2. 非人類看的懂的

gRPC 訊息會以 Protobuf 編碼。雖然 Protobuf 的傳送和接收效率較高，但其二進位格式卻不容易閱讀。Protobuf 需要指定於 proto 檔案中的訊息介面描述，才能正確還原序列化。需要額外的工具，才能在網路上分析 Protobuf 承載，並以手動方式撰寫要求。

# GRPC INSTALL

## 1.INSTALL GRPC

```
go get google.golang.org/grpc
```

## 2.INSTALL PROTOBUF

```
go get -u github.com/golang/protobuf/proto
```

```
go get -u github.com/golang/protobuf/protoc-gen-go
```

## 3.設定環境變數（該處有坑）

```
export PATH=$PATH:$GOPATH/bin
```

註：

(1)先到Gopath目錄下的Bin資料夾

(2)找到protoc-gen-go 該檔案

(3)mac： 將該檔案放到/usr/local/bin下

## 4.使用PROTOBUF

```
protoc --go_out=plugins=grpc:. *.proto
```

GRPC DEMO

GRPC

DEMO



# A GUIDE FOR FIRST-TIME TALK ABOUT GRPC

---

WRITING BY SCHNEE