

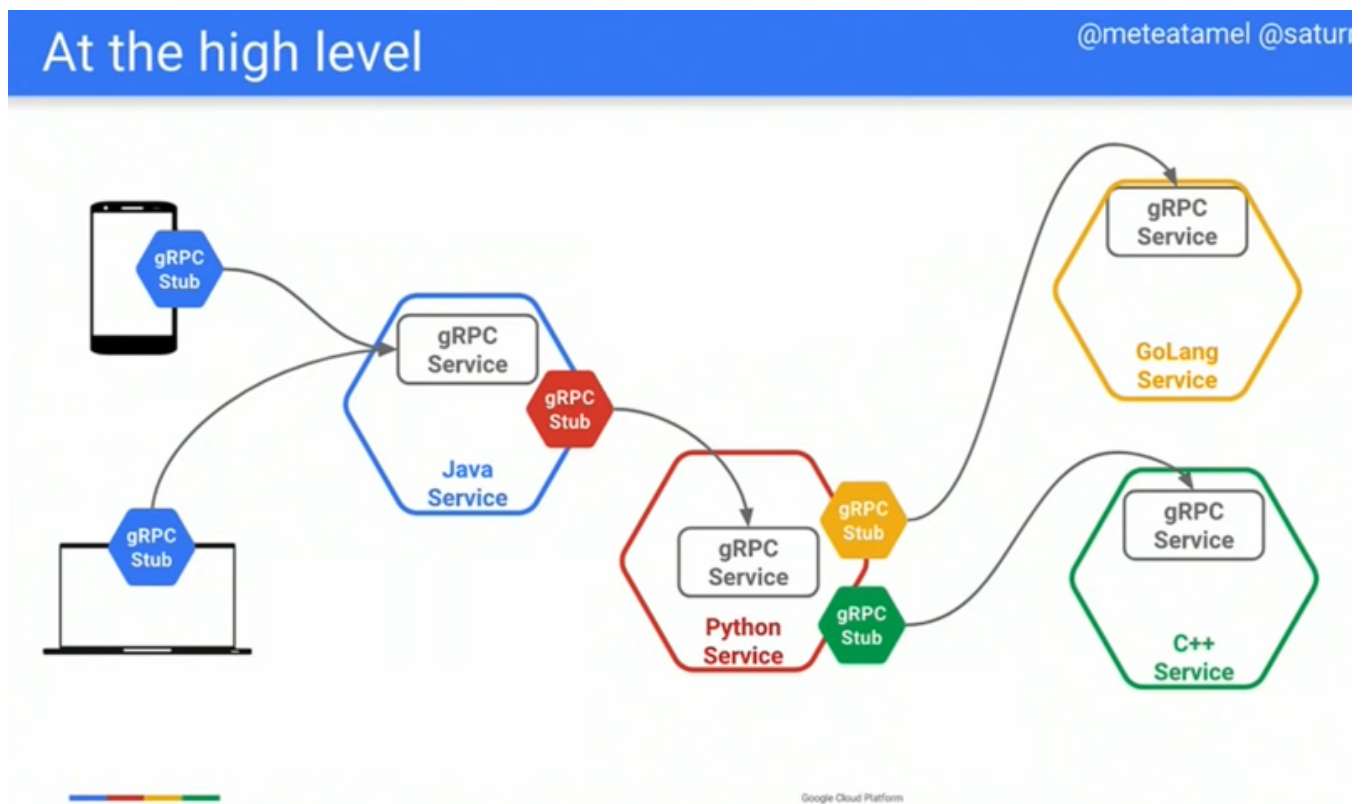
# # gRPC

## Définition

**gRPC** est un framework RPC ([Remote procedure call](#)) open source initialement développé par [Google](#). Il utilise [le protocole HTTP/2](#) pour le transport, [Protocol Buffers](#) comme langage de description d'interface.

## Explication

- High lvl = Vous générer le service grpc pour le service java à partir du 'service definition'
- La communication va se faire au travers des '**stub**' et c'est grpc qui s'occupe de ça. Enfin, vous pouvez avec les services en python, golang, c++ etc



Src : <https://www.youtube.com/watch?v=hNFM2pDGwKI>

## Les modes

- Unary RPC (requête vers le serveur et le serveur répond)

- Server streaming RPC (le serveur s'attend à envoyer une série de réponse)
- Client streaming RPC (le client envoie une série de request)
- Bidirectional streaming RPC (client et serveur envoie une série de request/reponse)

## Definition du message

- Définition de la syntaxe du 'protocole buffer', de base si on spécifie pas c'est proto2. De plus, nous pouvons également spécifier un package, avoir des 'enum' etc

```
syntax = "proto3";
```

```
message Number  
{  
    float value = 1;  
}
```

### Constrution d'un service

1. Permet la création d'une méthode et va retourner un item de type Number

```
service Calculator  
{  
    rpc SquareRoot(Number) returns (Number) {}  
}
```

## Inconvénient

1. Le serveur et les clients dépendent d'un schéma spécifique.