



Europe 2023

The RPC Revolution:

Getting the Most out of gRPC



Richard Belleville (he/him)
Software Engineer, Google
@gnossen

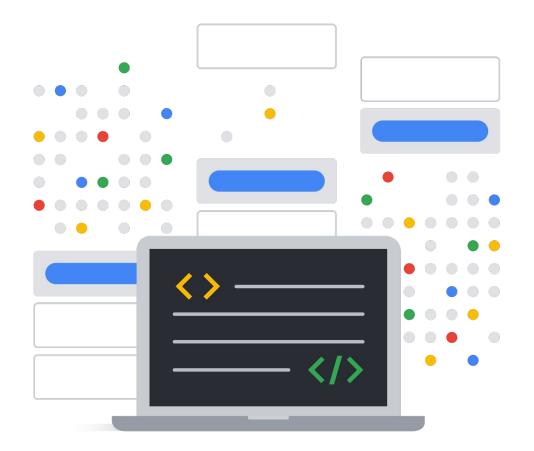


Kevin Nilson (he/him)
Engineering Manager, Google
@javaclimber

gRPC Today



- Used in applications with a worldwide-footprint and millions of users
- Popular with developers, including excellent support within the developer community
- Well rounded feature set with support for many popular programing languages







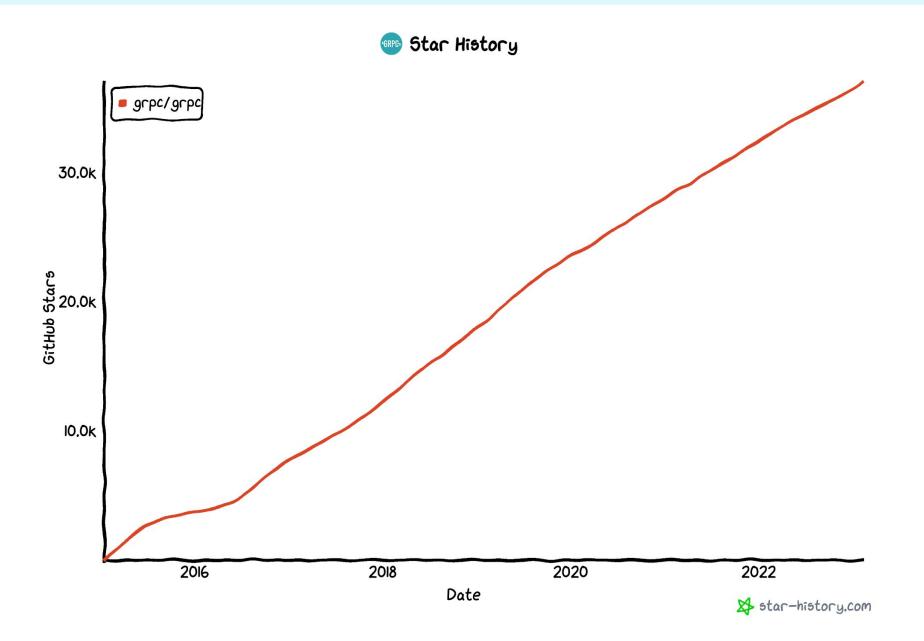
6 million weekly downloads of @grpc/grpc-js



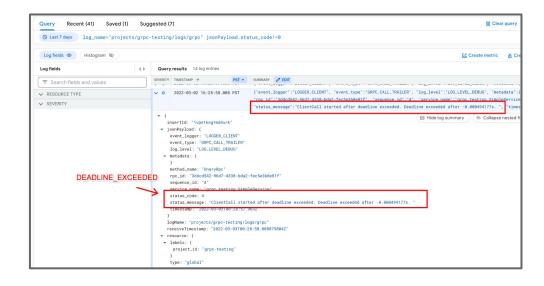
Downloaded 2 million times per day; #59 most downloaded package on PyPi

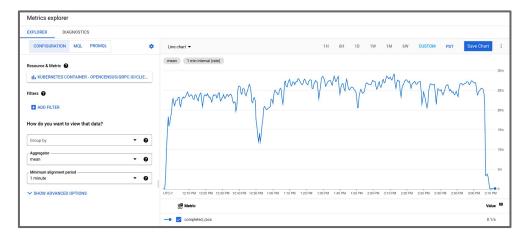


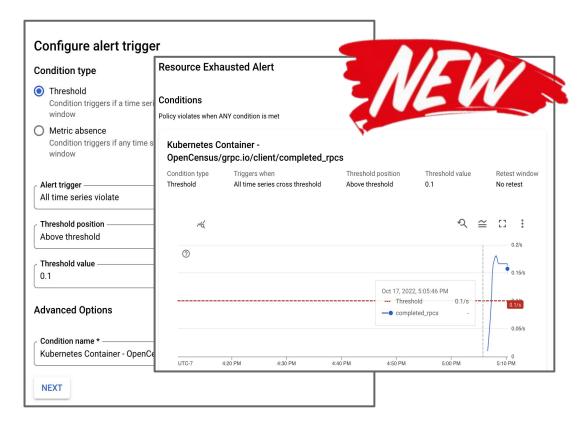
12.5 million Maven downloads per month of Java gRPC Context

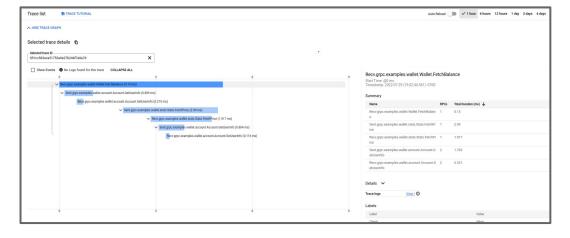


Observability









Areas Of Focus

Near-Term Deliverables









Documentation

We Want Your Help



- We actively review issues and PRs daily
- Issues get reviewed and triaged within one week
- Most PRs get merged
- We are looking for maintainers

Other gRPC Talks



Autoscaling Elastic Kubernetes Infrastructure for Stateful Applications Using Proxyless gRPC and Istio

Friday, 11:00

Topics



- Resource orientation and concurrency considerations
- protoc and protoc plugins
- Debugging utilities

gRPC and REST



{REST}



gRPC and REST



- Client-Server Architecture
- Statelessness
- Cacheability

- Transparency for Proxies
- Uniform Resource Interface



```
message Book {
  option (google.api.resource) = {
    type: "library.acme.com/Book"
    pattern: "books/{book}"
  };
  string name = 1;
  string author = 2 [(google.api.resource_reference) = {
    type: "library.acme.com/Author"
  }];
service Library {
  rpc CreateBook(CreateBookRequest) returns (Book);
  rpc GetBook(GetBookRequest) returns (Book);
  rpc UpdateBook(UpdateBookRequest) returns (Book);
  rpc DeleteBook(DeleteBookRequest) returns (google.protobuf.Empty);
  rpc ListBooks(ListBooksReguest) returns (ListBooksResponse);
```

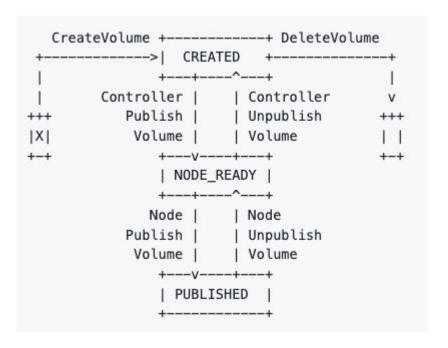


```
revice Controller {
    rpc CreateVolume (CreateVolumeRequest) returns (CreateVolumeResponse);
    rpc ControllerGetVolume (ControllerGetVolumeRequest) returns (ControllerGetVolumeResponse);
    rpc DeleteVolume (DeleteVolumeRequest) returns (DeleteVolumeResponse);
    rpc ListVolumes (ListVolumesRequest) returns (ListVolumesResponse);
    rpc ControllerPublishVolume (ControllerPublishVolumeRequest) returns (ControllerPublishVolumeResponse);
    rpc ControllerUnpublishVolume (ControllerUnpublishVolumeRequest) returns (ControllerUnpublishVolumeResponse);
    rpc ControllerExpandVolume (ControllerExpandVolumeRequest) returns (ControllerExpandVolumeResponse);
}
```



```
type VolumeState int
const (
  Created VolumeState = iota
 NodeReady
  Published
type Volume struct {
 Name string
  State VolumeState
```

```
vol := &Volume{"foo", Published}
controller.UpdateVolume(vol)
```





```
type VolumeController interface {
    Create(name string) *Volume
    Get(name string) *Volume
    Publish(v *Volume)
    Unpublish(v *Volume)
}
```

```
vc := NewVolumeController()
vol := vc.Create("foo")
vc.Publish(vol)
// Use volume
vc.Unpublish(vol)
```



```
message Account {
   string id = 1;
   float value_euros = 2;
}

service Bank {
   rpc CreateAccount(CreateAccountRequest) returns (Account);
   rpc GetAccount(GetAccountRequest) returns (Account);
   rpc UpdateAccount(UpdateAccountRequest) returns (Account);
   rpc DeleteAccount(DeleteAccountRequest) returns (google.protobuf.Empty)
   rpc ListAccounts(ListAccountsRequest) returns (ListAccountsResponse);
}
```

Server

Value: 10

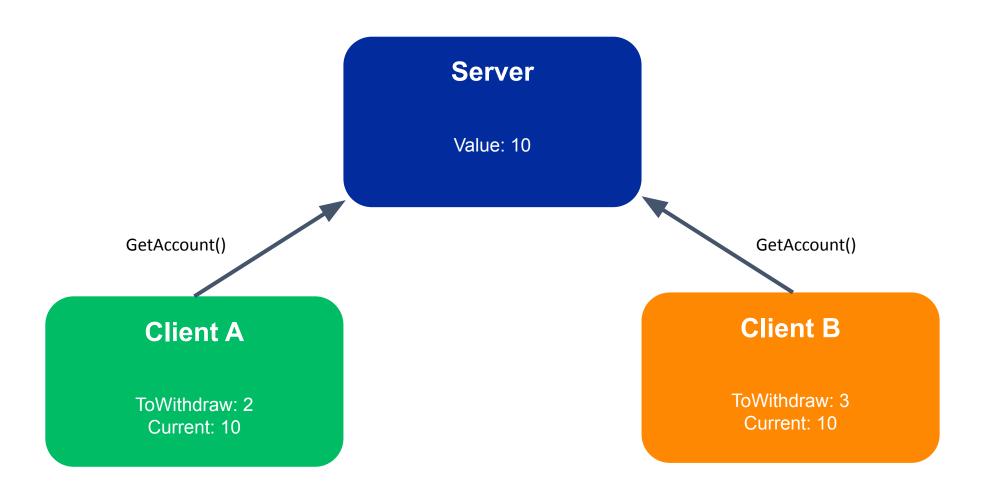
Client A

ToWithdraw: 2

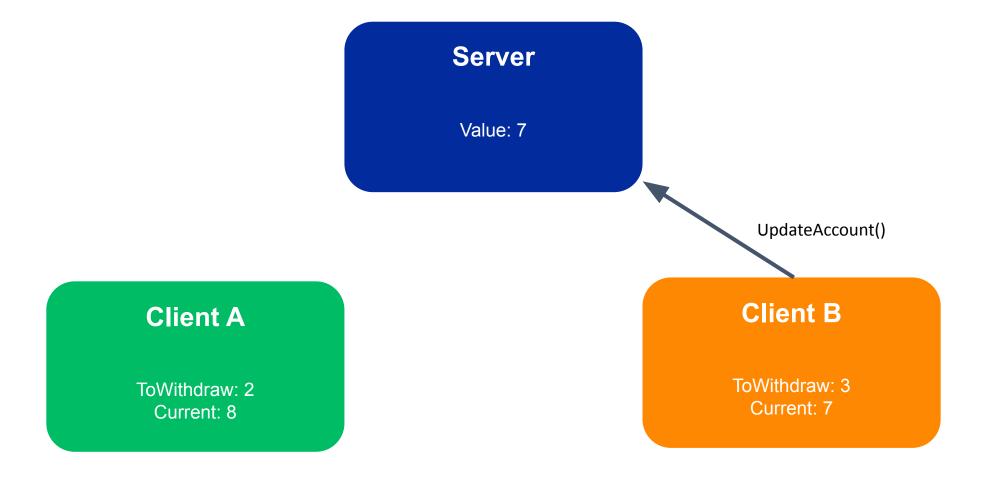
Client B

ToWithdraw: 3











Server

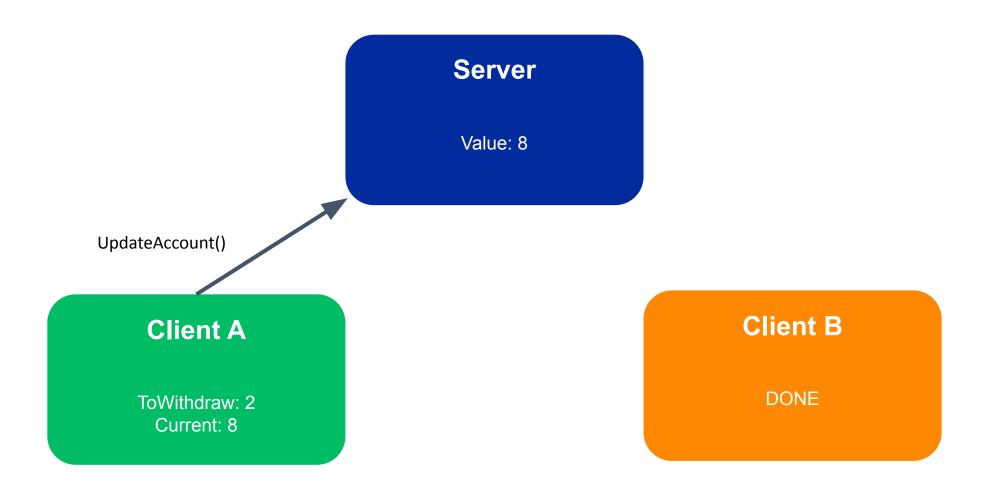
Value: 7

Client A

ToWithdraw: 2 Current: 8 **Client B**

DONE







Server

Value: 8

Client A

DONE

Client B

DONE

Concurrency Control - ETags



```
func Withdraw(id string, eurosToWithdraw float) {
 var etag string
 for {
   account, _ := client.GetAccount(&GetAccountRequest{Id: id})
   current := account.ValueEuros
   etag = account.Etag
   current -= eurosToWithdraw
   _, err := client.UpdateAccount(
     &UpdateAccountRequest{
       &Account{
         Id: id,
         ValueEuros: current,
       },
       etag,
   if err == nil {
     break
```

Concurrency Control - Distributed Locks KubeCon CloudNativeCon





```
message Account {
  string id = 1;
 float value_euros = 2;
service Bank {
  rpc UpdateAccount(UpdateAccountRequest) returns (Account);
  // Blocks until locked.
  rpc LockAccount(LockAccountRequest) returns (google.protobuf.Empty);
  rpc UnlockAccount(UnlockAccountRequest) returns (google.protobuf.Empty);
```

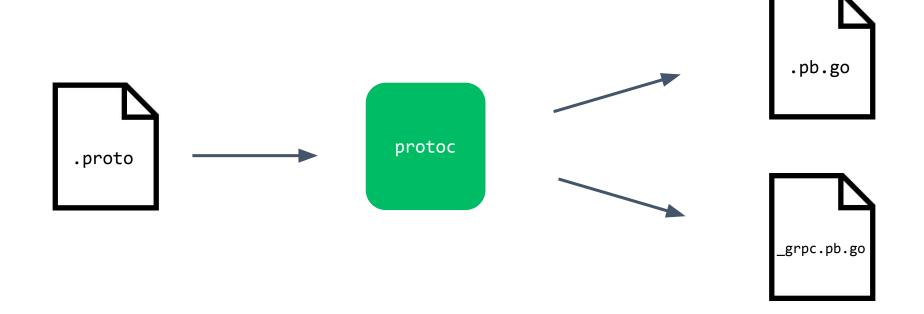
```
func Withdraw(id string, eurosToWithdraw float) {
 client.LockAccount()
 defer client.UnlockAccount()
 account, _ := client.GetAccount(&GetAccountRequest{Id: id})
 current := account.ValueEuros
 current -= eurosToWithdraw
 client.UpdateAccount(
   &UpdateAccountRequest{
     &Account{
       Id: id,
       ValueEuros: current,
```

Concurrency Control - RPCs



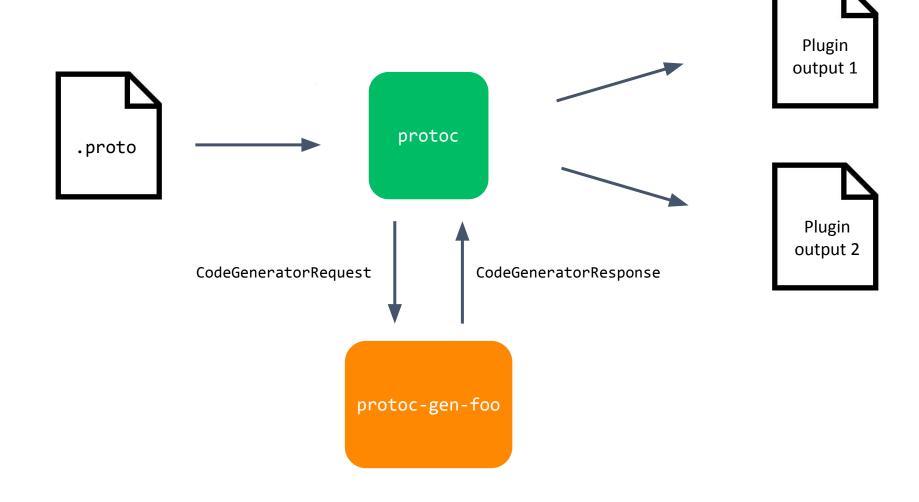
protoc





protoc Plugins





protoc-gen-validate



```
message Person {
  uint64 id = 1 [(validate.rules).uint64.gt = 999];
                                                                        p.Id = 1000
  string email = 2 [(validate.rules).string.email = true];
  string name = 3 [(validate.rules).string = {
             "^[^[0-9]A-Za-z]+([^[0-9]A-Za-z]+)*$",
   max_bytes: 256,
  }];
  Location home = 4 [(validate.rules).message.required = true];
  message Location {
    double lat = 1 [(validate.rules).double = {gte: -90, lte: 90}];
    double lng = 2 [(validate.rules).double = {gte: -180, lte: 180}];
```

```
p := new(Person)
err := p.Validate() // err: Id must be greater than 999
err = p.Validate() // err: Email must be a valid email address
p.Email = "example@example.com"
err = p.Validate() // err: Name must match pattern '^[^\d\s]+( [^\d\s]+)*$'
p.Name = "Protocol Buffer"
err = p.Validate() // err: Home is required
p.Home = \&Location{37.7, 999}
err = p.Validate() // err: Home.Lng must be within [-180, 180]
p.Home.Lng = -122.4
err = p.Validate() // err: nil
```

gRPC-Gateway



```
message StringMessage {
   string value = 1;
}

service YourService {
   rpc Echo(StringMessage) returns (StringMessage) {
      option (google.api.http) = {
        post: "/v1/example/echo"
        body: "*"
      };
   }
}
```



Example: Data Locality



```
syntax = "proto3";
option go_package = "foo.com/account";
import "com/foo/locality.proto";
package com.foo.account;
message Account {
    option (com.foo.locality) = {
        restricted_to_country: "netherlands"
        restricted_to_country: "belgium"
    };
    string id = 1;
    float value_euros = 2;
```

Example: Data Locality



```
syntax = "proto3";
option go_package = "foo.com/account";
import "com/foo/locality.proto";
package com.foo.account;
message Account {
    option (com.foo.locality) = {
        restricted_to_country: "netherlands"
        restricted_to_country: "belgium"
    };
    string id = 1;
    float value_euros = 2;
```

```
syntax = "proto3";
option go_package = "foo.com";
import "google/protobuf/descriptor.proto";
package com.foo;
message LocalityOptions {
    repeated string restricted_to_country = 1;
extend google.protobuf.MessageOptions {
    LocalityOptions locality = 183782;
```

Example: Log Sensitivity



```
syntax = "proto3";
syntax = "proto3";
                                                                     option go_package = "foo.com";
option go package = "foo.com/account";
                                                                     import "google/protobuf/descriptor.proto";
import "com/foo/log_sensitivity.proto";
                                                                     package com.foo;
package com.foo.account;
                                                                     enum LogSensitivityOptions {
message Account {
                                                                        NOT SENSITIVE = 0;
    string id = 1;
                                                                         SENSITIVE = 1;
    float value_euros = 2 [(com.foo.log_sensitivity) = SENSITIVE]; }
                                                                     extend google.protobuf.FieldOptions {
                                                                         LogSensitivityOptions log_sensitivity = 183783;
```

Health Checking



```
message HealthCheckRequest {
  string service = 1;
message HealthCheckResponse {
  enum ServingStatus {
    UNKNOWN = 0;
   SERVING = 1;
   NOT_SERVING = 2;
    SERVICE_UNKNOWN = 3;
  ServingStatus status = 1;
service Health {
  rpc Check(HealthCheckRequest) returns (HealthCheckResponse);
  rpc Watch(HealthCheckRequest) returns (stream HealthCheckResponse);
```

Health Checking



```
message HealthCheckRequest {
  string service = 1;
message HealthCheckResponse {
  enum ServingStatus {
    UNKNOWN = 0;
    SERVING = 1;
   NOT SERVING = 2;
    SERVICE_UNKNOWN = 3;
  ServingStatus status = 1;
service Health {
  rpc Check(HealthCheckRequest) returns (HealthCheckResponse);
  rpc Watch(HealthCheckRequest) returns (stream HealthCheckResponse);
```

```
containers:
- name: etcd
  image: registry.k8s.io/etcd:3.5.1-0
  ports:
  - containerPort: 2379
  livenessProbe:
    grpc:
      port: 2379
    initialDelaySeconds: 10
```

grpcdebug - Health Checking



```
$ grpcdebug localhost:50051 health
```

<0verall>: SERVING

\$ grpcdebug localhost:50051 health helloworld.Greeter

<0verall>: SERVING

helloworld.Greeter: SERVING

grpcdebug - channelz



```
$ grpcdebug localhost:50051 channelz channel 7
Channel ID:
Target:
                   localhost:10001
                   READY
State:
Calls Started:
                   3976
Calls Succeeded:
                   3520
Calls Failed:
                   456
Created Time:
                   6 minutes ago
___
Subchannel ID
                Target
                                  State
                                            Calls(Started/Succeeded/Failed)
                                                                               CreatedTime
8
                localhost:10001
                                  READY
                                            3976/3520/456
                                                                               6 minutes ago
                           Child Ref
                                                           Description
Severity
           Time
                                                           Channel Created
CT INFO
           6 minutes ago
                                                           Resolver state updated: {Addresses:[{Addr:localhost:10001...
CT INFO
           6 minutes ago
                                                           Channel switches to new LB policy "pick_first"
CT INFO
           6 minutes ago
CT_INFO
                           subchannel(subchannel_id:8 )
                                                           Subchannel(id:8) created
           6 minutes ago
CT_INFO
           6 minutes ago
                                                           Channel Connectivity change to CONNECTING
CT INFO
           6 minutes ago
                                                           Channel Connectivity change to READY
```

grpcdebug - channelz



```
$ grpcdebug localhost:50051 channelz socket 9
Socket ID:
Address:
                                 ::1:47436->::1:10001
Streams Started:
                                 4807
Streams Succeeded:
                                 4807
Streams Failed:
                                 0
Messages Sent:
                                 4807
Messages Received:
                                 4243
Keep Alives Sent:
                                 0
Last Local Stream Created:
                                 now
Last Remote Stream Created:
                                 a long while ago
Last Message Sent Created:
                                 now
Last Message Received Created:
                                 now
Local Flow Control Window:
                                 65535
Remote Flow Control Window:
                                 65535
Socket Options Name
                      Value
SO_LINGER
                      [...]:{duration:{}}
                      [...]:{duration:{}}
SO RCVTIMEO
                      [....]:{duration:{}}
SO_SNDTIMEO
TCP_INFO
                      [...]:{tcpi_state:1 tcpi_options:7 tcpi_rto:204000 tcpi_ato:40000 ...
Security Model:
                 TLS
Standard Name:
                 TLS_AES_128_GCM_SHA256
```

grpcdebug - xDS



```
$ grpcdebug localhost:50051 xds config --type=eds
  "dynamicEndpointConfigs": [
      "versionInfo": "1",
      "endpointConfig": {
       "@type": "type.googleapis.com/envoy.config.endpoint.v3.ClusterLoadAssignment",
       "clusterName": "cloud-internal-istio:cloud_mp_1040920224690_6530603179561593229",
        "endpoints": [
           "locality": {
             "subZone": "jf:us-central1-a_7062512536751318190_neg"
            "lbEndpoints": [
               "endpoint": {
                 "address": {
                   "socketAddress": {
                     "address": "192.168.120.26",
                     "portValue": 8080
               "healthStatus": "HEALTHY"
            "loadBalancingWeight": 100
      "lastUpdated": "2021-03-31T01:20:33.936Z",
      "clientStatus": "ACKED"
```

The RPC Revolution:

Getting the Most out of gRPC

- Check out more at <u>https://github.com/grpc-ecosystem/awesome-grpc</u>
- We want to hear your feedback on gRPC
 - Schedule a call at grpc.io/meet
- Join the gRPC mailing list groups.google.com/g/grpc-io





Europe 2023





