

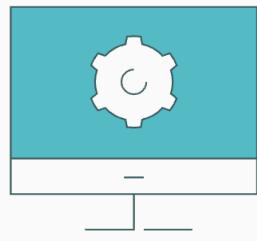


A high-performance, open source universal RPC framework

Type Safety and Productivity

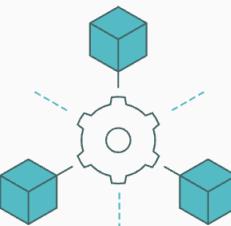
sample project that use Angular 8 on client and SpringBoot (written in Kotlin) on server side

Konstantin Ignatyev kgignatyev@gmail.com



Simple service definition

Define your service using Protocol Buffers, a powerful binary serialization toolset and language



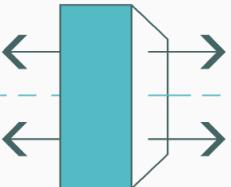
Start quickly and scale

Install runtime and dev environments with a single line and also scale to millions of RPCs per second with the framework



Works across languages and platforms

Automatically generate idiomatic client and server stubs for your service in a variety of languages and platforms



Bi-directional streaming and integrated auth

Bi-directional streaming and fully integrated pluggable authentication with HTTP/2-based transport

```
8  message LatLng {  
9    // The latitude in degrees. It must be in the range [-90.0, +90.0].  
10   double lat = 1;  
11   // The longitude in degrees. It must be in the range [-180.0, +180.0].  
12   double lng = 2;  
13}  
14  
15 message Polygon {  
16   repeated LatLng vertices = 1;  
17}  
18  
19 enum AreaType {  
20   TEAM_CONTROL = 0;  
21   SERVICE_AREA = 1;  
22}  
23  
24 message Area {  
25   string id = 1;  
26   Polygon polygon = 2;  
27   string name = 3;  
28   AreaType area_type = 4;  
29}  
30  
31 message AreasList {  
32   repeated Area items = 1;  
33}  
34  
35 message AreaSearchCriteria {  
36   oneof criteria {  
37     string text = 1;  
38     LatLng contains_point = 2;  
39   }  
40}  
41 service Geography {}  
42   rpc CreateArea (Area) returns (Area);  
43   rpc GetAreaByID (google.protobuf.StringValue) returns (Area);  
44   rpc UpdateArea (Area) returns (Area);  
45   rpc DeleteAreaByID(google.protobuf.StringValue ) returns (google.protobuf.Empty);  
46   rpc FindAreas(google.protobuf.StringValue ) returns (AreasList);  
47   // deprecated, use FindAreasBy  
48   rpc FindAreasContaining( LatLng ) returns (AreasList) {  
49     option deprecated = true;  
50   };  
51   rpc FindAreasBy( AreaSearchCriteria ) returns (AreasList);  
52}
```

From proto file we generate client and server libraries

```
build-stubs.sh <input>
49 gen_proto_docs () {
50   mkdir -p target/docs
51   gen_docs --doc_opt=html,index.html geography.proto
52   cp target/docs/index.html target/docs/index.original.html
53   cat target/docs/index.original.html |sed -e "s/>Table of Contents</> Version:</>${INTERFACE_VERSION}</>${INTERFACE_VERSION}.html"
54   cp target/docs/index.html target/docs/index-$INTERFACE_VERSION.html
55   gen_docs --doc_opt=markdown,interface.md geography.proto
56 }
57
58 mvn clean
59 gen_java_stubs
60 export_version_info
61 source target/version_info
62
63 gen_proto_docs
64
65 gen_ts_stubs
66
67 deploy_to_repo
```

and versioned documentation Protocol Documentation

Version: 1.0.8 generated: Tue Apr 14 09:05:12 PDT 2020

interface
 |.idea
src
 main
 proto
 geography.proto
target
 classes
 docs
 index.html
 index.original.html
 index-1.0.8.html (circled)
 interface.md
 generated-sources
 maven-archiver
 maven-status
 protoc-dependencies
 protoc-plugins
 ts_grpcweb

geography.proto

M Area
M AreaSearchCriteria
M AreasList
M LatLng
M Polygon
E AreaType
S Geography

Scalar Value Types

geography.proto

Area

| Field | Type |
|-----------|----------|
| id | string |
| polygon | Polygon |
| name | string |
| area_type | AreaType |

LatLng

| Field | Type | Label | Description |
|-------|--------|-------|---|
| lat | double | | The latitude in degrees. It must be in the range [-90.0, +90.0]. |
| lng | double | | The longitude in degrees. It must be in the range [-180.0, +180.0]. |

Polygon

| Field | Type | Label | Description |
|----------|--------|----------|-------------|
| vertices | LatLng | repeated | |

AreaType

| Name | Number | Description |
|--------------|--------|-------------|
| TEAM_CONTROL | 0 | |
| SERVICE_AREA | 1 | |

Geography

| Method Name | Request Type | Response Type | Description |
|---------------------|------------------------------|------------------------|-----------------------------|
| CreateArea | Area | Area | |
| GetAreaByID | .google.protobuf.StringValue | Area | |
| UpdateArea | Area | Area | |
| DeleteAreaByID | .google.protobuf.StringValue | .google.protobuf.Empty | |
| FindAreas | .google.protobuf.StringValue | AreasList | |
| FindAreasContaining | LatLng | AreasList | |
| FindAreasBy | AreaSearchCriteria | AreasList | deprecated, use FindAreasBy |

Methods with deprecated option

| Method Name | Option |
|---------------------|--------|
| FindAreasContaining | true |

Documentation for client available inside of NPM module 100% in-sync with API

The screenshot shows a code editor interface with a left sidebar and a right panel.

Left Sidebar:

- Project navigation bar: Project, +, -, MD interface.md
- File tree:
 - geography-client (~dev/kgi/type-safety-end-to-end/geography-cl)
 - .idea
 - e2e
 - node_modules library root
 - .bin
 - .cache
 - @agm
 - @angular
 - @angular-devkit
 - @auth0
 - @babel
 - @fortawesome
 - @improbable-eng
 - @istanbuljs
 - @kgi
 - geography-interface
 - geography_pb.d.ts
 - geography_pb.js
 - geography_pb_service.d.ts
 - geography_pb_service.js
 - interface.md
 - package.json
 - @ng-bootstrap
 - @ngtools
 - @schematics
 - @types
 - @webassemblyjs
 - @xtuc
 - @yarnpkg
 - accepts
 - acorn
 - adm-zip
 - after
 - agent-base

A red oval highlights the `geography_pb_service.js` file, and a yellow arrow points from it towards the right panel.

Right Panel:

interface.md tab is active.

Polygon

| Field | Type | Label | Description |
|-------|--------|-------|---|
| lat | double | | The latitude in degrees. It must be in the range [-90.0, +90.0]. |
| lng | double | | The longitude in degrees. It must be in the range [-180.0, +180.0]. |

Polygon

| Field | Type | Label | Description |
|----------|--------|----------|-------------|
| vertices | LatLng | repeated | |

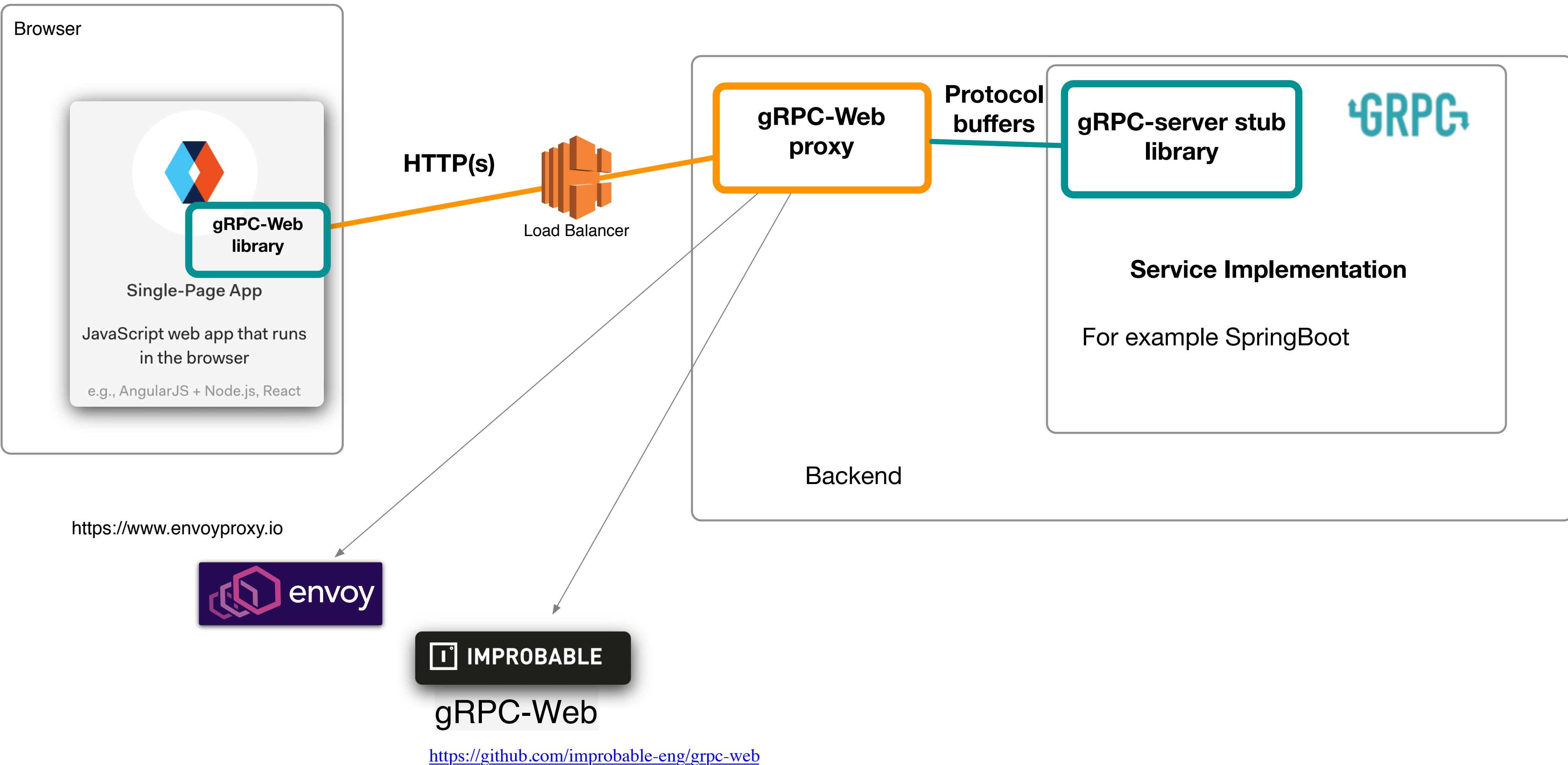
AreaType

| Name | Number | Description |
|--------------|--------|-------------|
| TEAM_CONTROL | 0 | |
| SERVICE_AREA | 1 | |

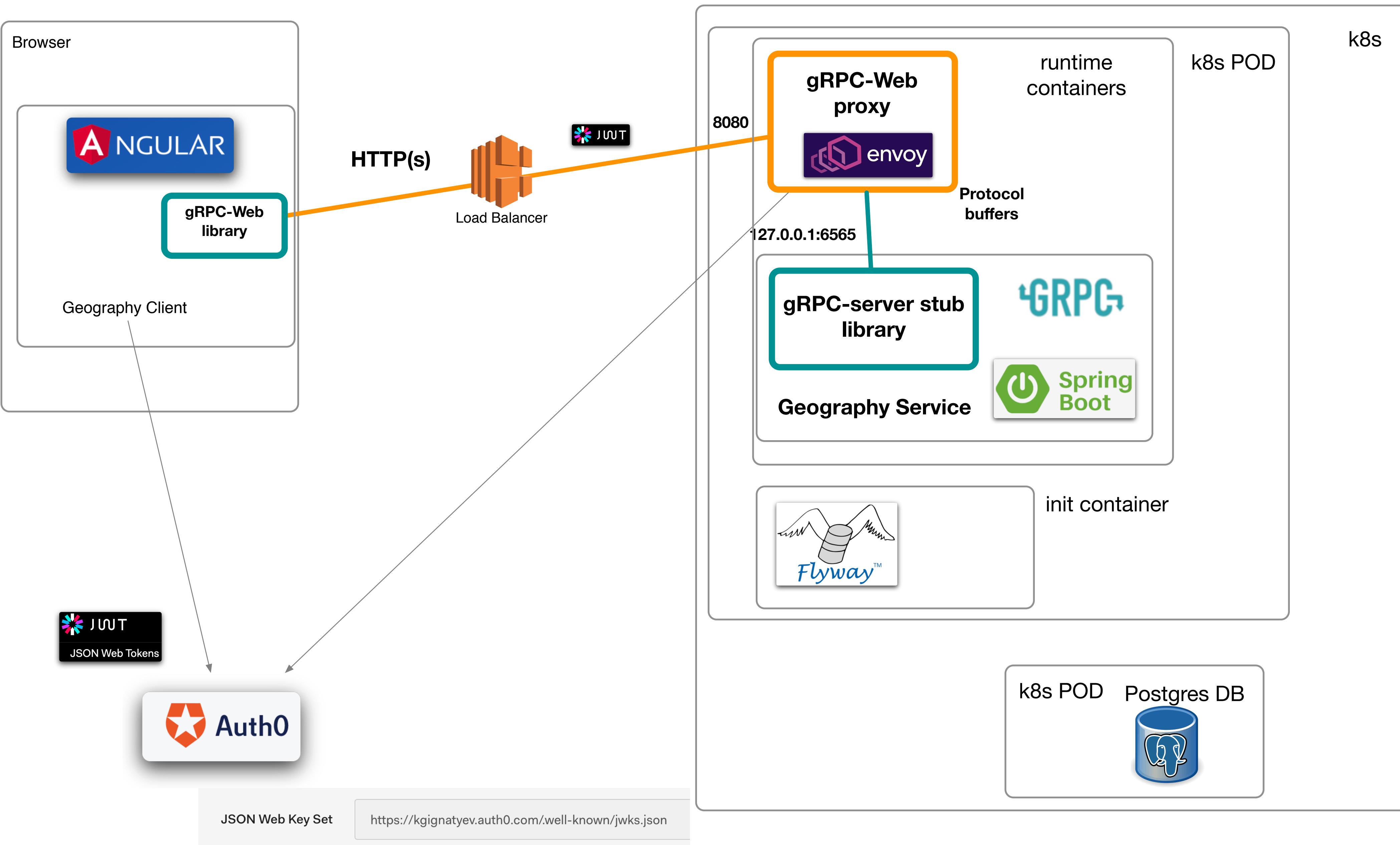
Geography

| Method Name | Request Type | Response Type | Description |
|----------------|------------------------------|------------------------|-------------|
| CreateArea | Area | Area | |
| GetAreaByID | .google.protobuf.StringValue | Area | |
| UpdateArea | Area | Area | |
| DeleteAreaByID | .google.protobuf.StringValue | .google.protobuf.Empty | |
| FindAreas | .google.protobuf.StringValue | AreasList | |

Running gRPC service and exposing it for Web Clients



Running gRPC service in Kubernetes and exposing it for Web Clients



Server Side - simply extend a generated class and implement methods

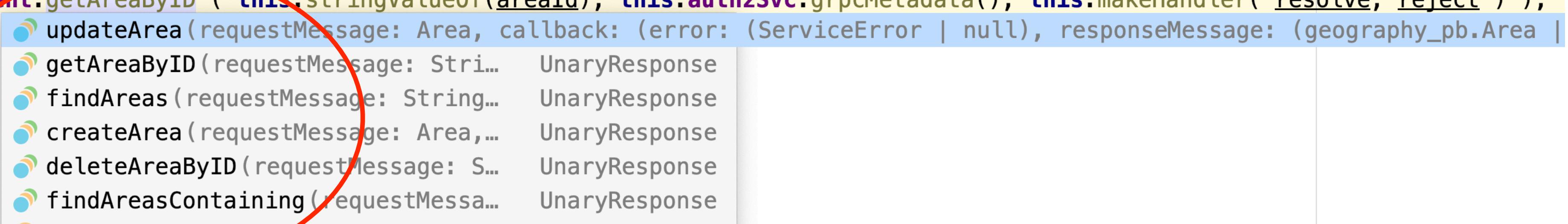
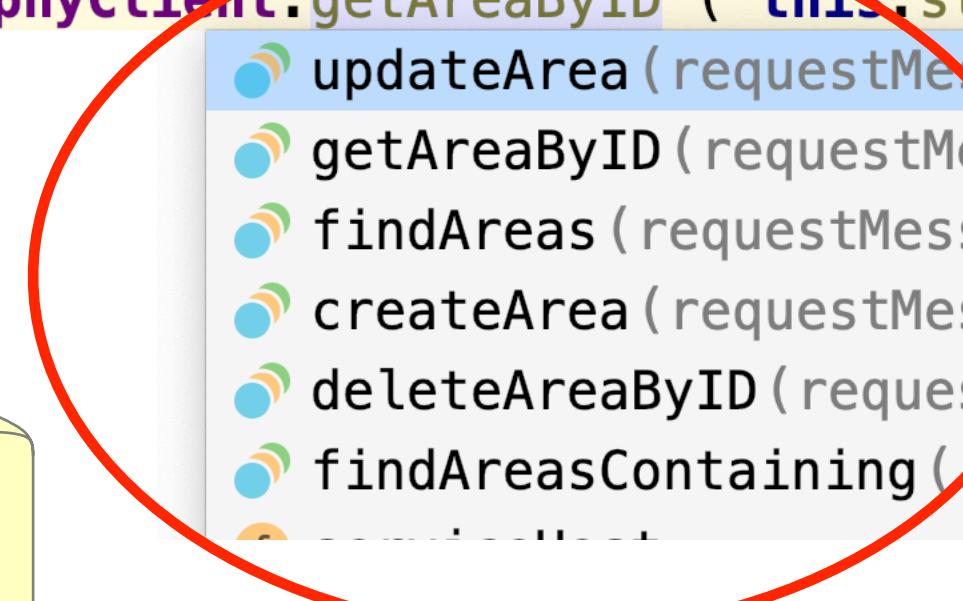
```
14 @GrpcService( interceptors = [ AuthInterceptor::class ] )  
15 class GeographyGRPC: GeographyImplBase() {  
16  
17     @Resource  
18     lateinit var geographySvc: GeographySvc  
19  
20     fun <T> sendResponse(response0bserver: Stream0bserver<T>?, value: T?) {  
21         response0bserver!!  
22         response0bserver.onNext(value)  
23         response0bserver.onCompleted()  
24     }  
25  
26     override fun updateArea(request: GeographyOuterClass.Area, response0bserver: Stream0bserver<GeographyOuterClass.Area>?) {  
27         sendResponse( response0bserver, geographySvc.update( request))  
28     }  
29  
30     override fun createArea(request: GeographyOuterClass.Area, response0bserver: Stream0bserver<GeographyOuterClass.Area>?) {  
31         sendResponse( response0bserver, geographySvc.create( request))  
32     }  
33  
34     override fun getAreaByID(request: StringValue, response0bserver: Stream0bserver<GeographyOuterClass.Area>?) {  
35         sendResponse( response0bserver, geographySvc.getAreaByID( request.value ))  
36     }  
37  
38     override fun deleteAreaByID(request: StringValue, response0bserver: Stream0bserver<Empty>?) {  
39         geographySvc.deleteAreaByID( request.value )  
40         sendResponse( response0bserver, Empty.getDefaultInstance() )  
41     }  
42  
43     override fun findAreasContaining(request: GeographyOuterClass.LatLng, response0bserver: Stream0bserver<GeographyOuterClass.AreasList>?) {  
44         sendResponse( response0bserver, geographySvc.findAreasContaining( request))  
45     }  
46 }
```

```
40
41 service Geography {
42     rpc CreateArea (Area) returns (Area);
43     rpc GetAreaByID (google.protobuf.StringValue) returns (Area);
44     rpc UpdateArea (Area) returns (Area);
45     rpc DeleteAreaByID(google.protobuf.StringValue ) returns (google.protobuf.Empty);
46     rpc FindAreas(google.protobuf.StringValue ) returns (AreasList);
47     // deprecated, use FindAreasBy
48     rpc FindAreasContaining( LatLng ) returns (AreasList) {
49         option deprecated = true;
50     };
51     rpc FindAreasBy( AreaSearchCriteria ) returns (AreasList);
52 }
```



```
55 updateArea( area: Area ): Promise< Area > {
56     return new Promise( [ executor, [ resolve, reject ] ] => {
57         this.geographyClient.updateArea( area, this.authzSvc.grpcMetadata(), this.makeHandler( resolve, reject ) );
58     } );
59 }
60
61 findAreas( text: string ): Promise< AreasList > {
62     return new Promise( [ executor, [ resolve, reject ] ] => {
63         this.geographyClient.findAreas( this.stringValueOf(text), this.authzSvc.grpcMetadata(), this.makeHandler( resolve, reject ) );
64     } );
65 }
66
67 async getAreaById(areaId: string): Promise< Area > {
68     return new Promise( [ executor, [ resolve, reject ] ] => {
69         this.geographyClient.getAreaByID ( this.stringValueOf(areaId), this.authzSvc.grpcMetadata(), this.makeHandler( resolve, reject ) );
70     } );
71 }
72
73
```

auto completion works



```
5 @Component({
6   selector: 'app-areas',
7   templateUrl: './areas.component.html'
8 })
9 export class AreasComponent implements OnInit {
10
11   areas: Array<Area>;
12
13   constructor( private geographySvc: GeographyService) { }
14
15   async ngOnInit() {
16     const areas = await this.geographySvc.findAreas( text: '')
17     this.areas = areas.getItemsList();
18 }
```

```
13 <div class="row" *ngFor="let a of areas">
14   <div class="col-md-4">
15     <a class="nav-link" [routerLink]=[ '/areas', a.getId() ]>{{a.getId()}}</a>
16   </div>
17   <div class="col-md-3">
18     {{a.getName()}}
19   </div>
20 </div>
```

Home Areas About

Areas

| | |
|--|----------|
| 1cec68d9-2632-4f67-8771-868a091f01f9 | kgi12345 |
| e86651c8-f60b-4d88-875c-3c39e49203ab | usa2 |
| 585f1689-f38d-4770-b2cb-12c31266be11 | r23414 |
| ce4991b9-e9f7-4e09-8234-b4e61876aa55 | a1 |
| 9cc8011c-7fe3-431e-ad21-7ea01c1de591 | b2 |
| a89a57a5-50c7-4fc7-b53f-a439ab47d7ae | kgi |
| be78cc59-b0a6-4449-ae84-847e5b4750cf | z3 |

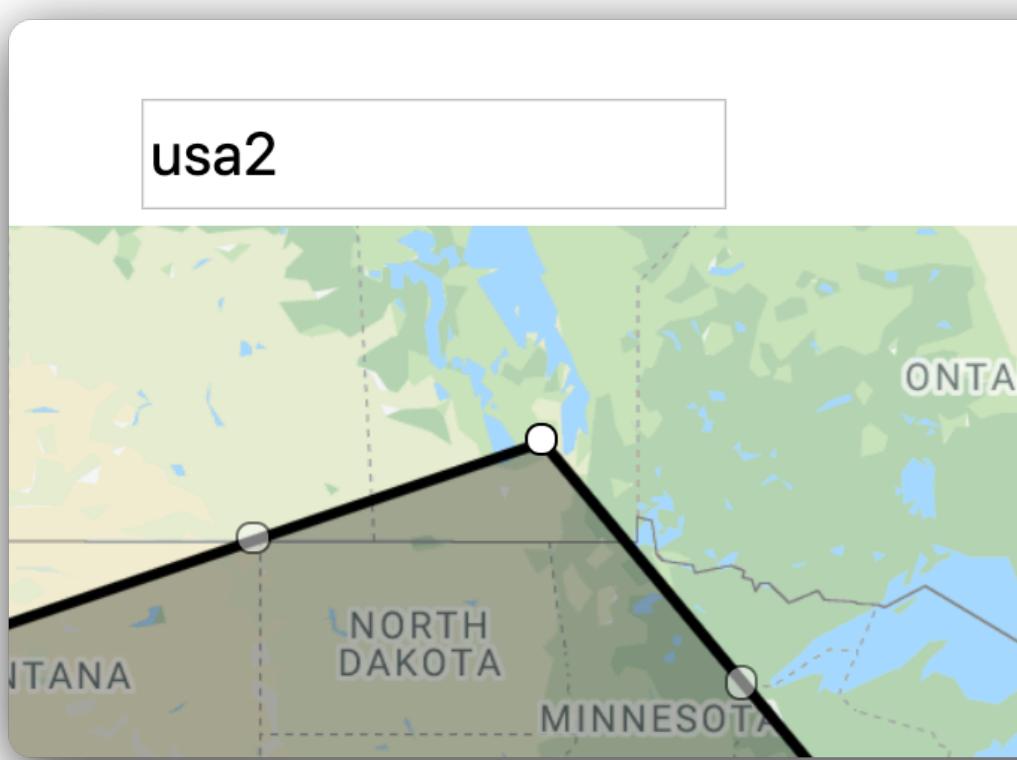
```
19 <div class="col-md-2">
20   <input [ngModel]="area?.getName()" 
21     (ngModelChange)="area?.setName( $event.toString() )" />
22 </div>
23 </div>
```

A screenshot of a map application interface. At the top, there is a form with a label "Name:" and an input field containing "r23414". Below the form is a map of West Africa, specifically showing Senegal, Gambia, Guinea-Bissau, Guinea, Sierra Leone, Liberia, Côte d'Ivoire, Ghana, Burkina Faso, Benin, Togo, Nigeria, and Equatorial Guinea. A black polygon is drawn on the map, roughly encompassing the coastal areas of these countries. The map includes standard features like country names, borders, and a zoom control. At the bottom of the map interface, there are "Cancel" and "Save" buttons, with "Save" being highlighted in blue.

```
150 subscribeTorouteParams() {
151   this.sub = this.route.params.subscribe(next: async params => {
152     let areaId = params.id;
153     if (areaId) {
154       if ('new' == areaId) {
155         this.area = this.newArea()
156       } else {
157         this.area = await this.geographySvc.getAreaById(areaId);
158       }
159     } else {
160       this.area = this.newArea()
161     }
162     this.drawArea();
163   });
164 }
```

```
175 async save() {
176
177   if (this.isNewArea()) {
178     this.area = await this.geographySvc.createArea(this.area)
179   } else {
180     this.area = await this.geographySvc.updateArea(this.area)
181   }
182   this.close();
183 }
```

Client Code - Custom component



TEAM_CONTROL

```
1 <select (ngModelChange)="setAreaType($event)" [ngModel]="getAreaType()">
2   <option *ngFor="let k of keys" [value]="k">
3     {{k}}
4   </option>
5 </select>
```

```
4 @Component({
5   selector: 'app-area-type-select',
6   templateUrl: './area-type-select.component.html'
7 })
8 export class AreaTypeSelectComponent implements OnInit {
9
10  @Input() area: Area;
11
12  constructor() { }
13
14  ngOnInit() {
15    }
```

Class to Import

- Area ("@kgi/geography-interface/geography_pb")
- Area ("@kgi/geography-interface/geography_pb")

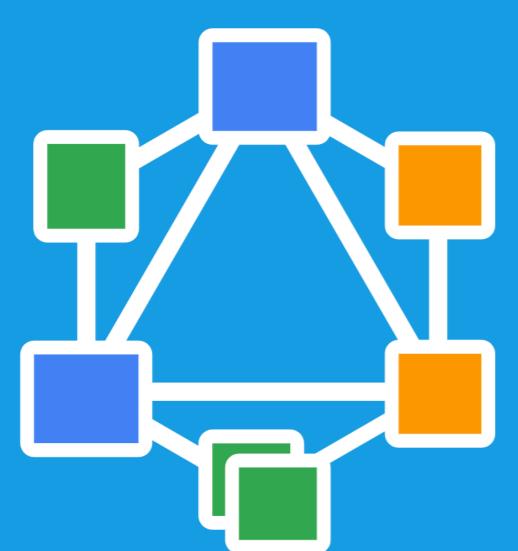
.proto definition

```
19 enum AreaType {
20   TEAM_CONTROL = 0;
21   SERVICE_AREA = 1;
22 }
```

```
12
13  keys: string[] = Object.keys( AreaType );
14  at = AreaType;
15
16  constructor() { }
17
18  ngOnInit() {
19    console.log( this.keys )
20  }
21
22  setAreaType(typeKey: any) {
23    this.area.setAreaType( this.at[typeKey] );
24    console.info(typeKey);
25  }
26
27  getAreaType() {
28    if( this.area ) {
29      return this.keys[ this.area.getAreaType() ]
30    } else {
31      return this.keys[0]
32    }
33  }
34}
```

Rejoiner

Uniform GraphQL API
served over
HTTP and
gRPC



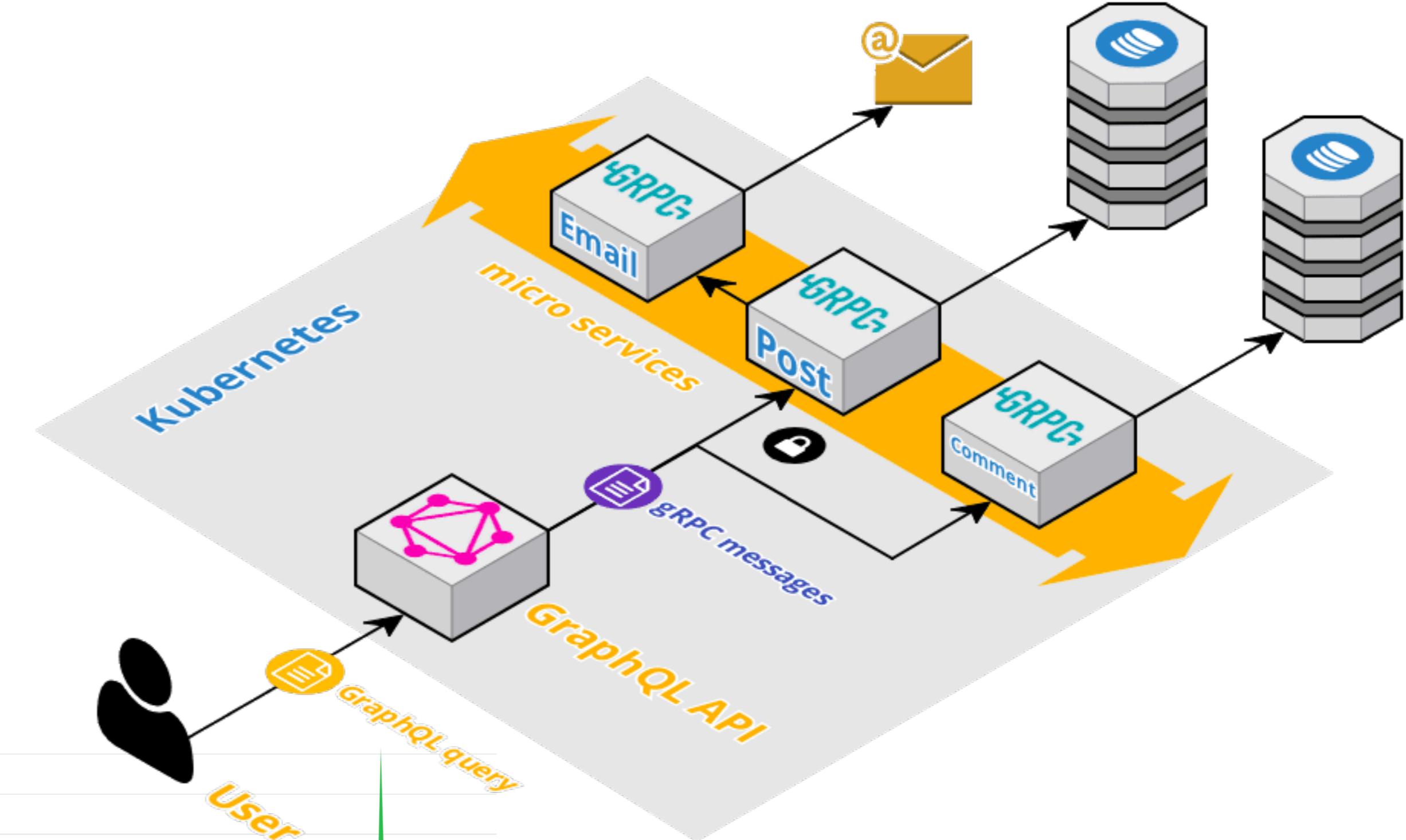
Java
gRPC Microservice



Go
gRPC Microservice



Python
gRPC Microservice



<https://github.com/google/rejoiner>

