

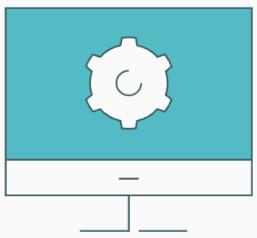
# End-to-End Type Safety and Productivity with



## and codegeneration

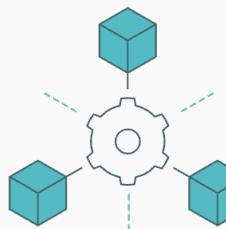
**Konstantin Ignatyev** [kgignatyev@gmail.com](mailto:kgignatyev@gmail.com)

**Demo project uses Angular 8 on client and SpringBoot (written in Kotlin) on server side**



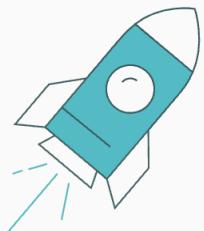
### Simple service definition

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



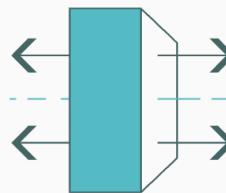
### Start quickly and scale

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

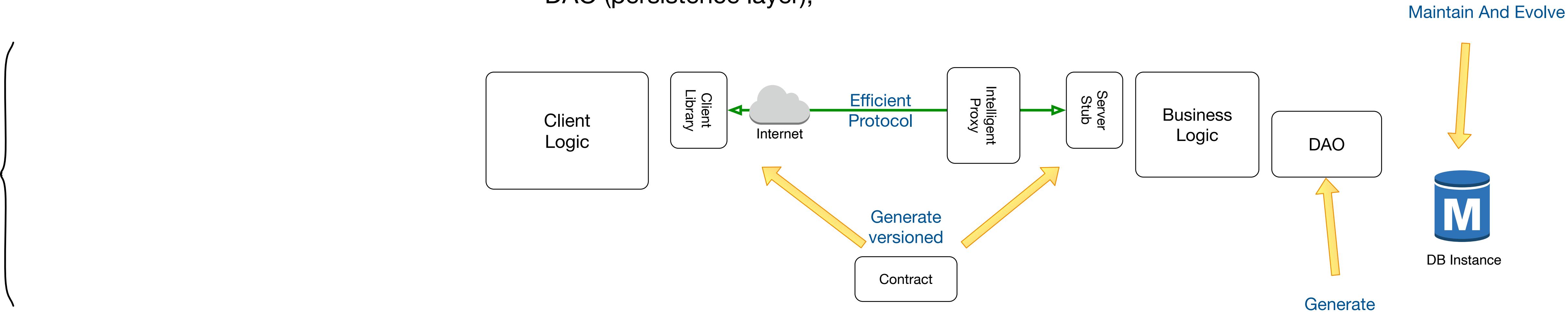
Automatically generate idiomatic client and server stubs for your service in a variety of languages and platforms



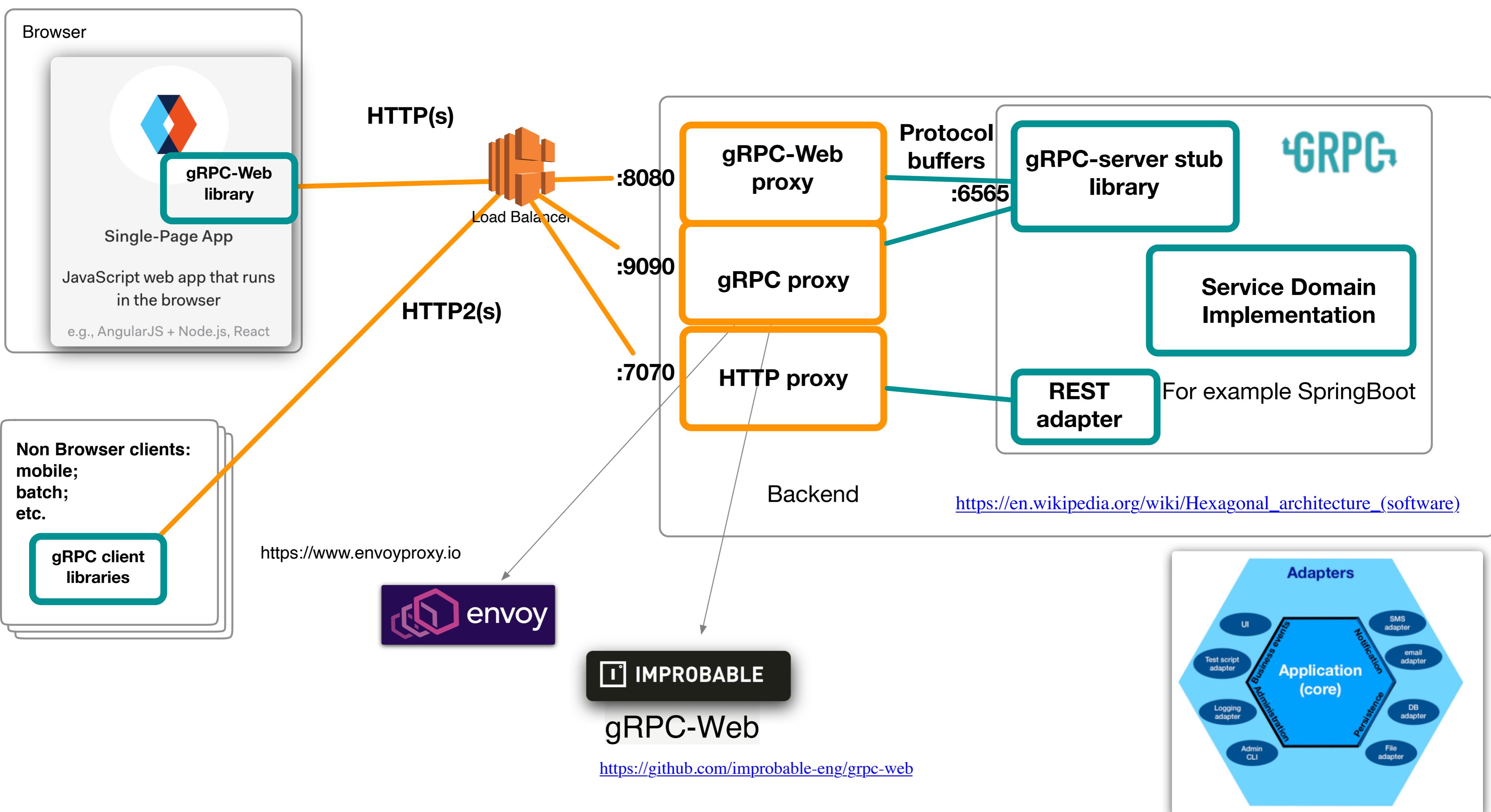
### Bi-directional streaming and integrated auth

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

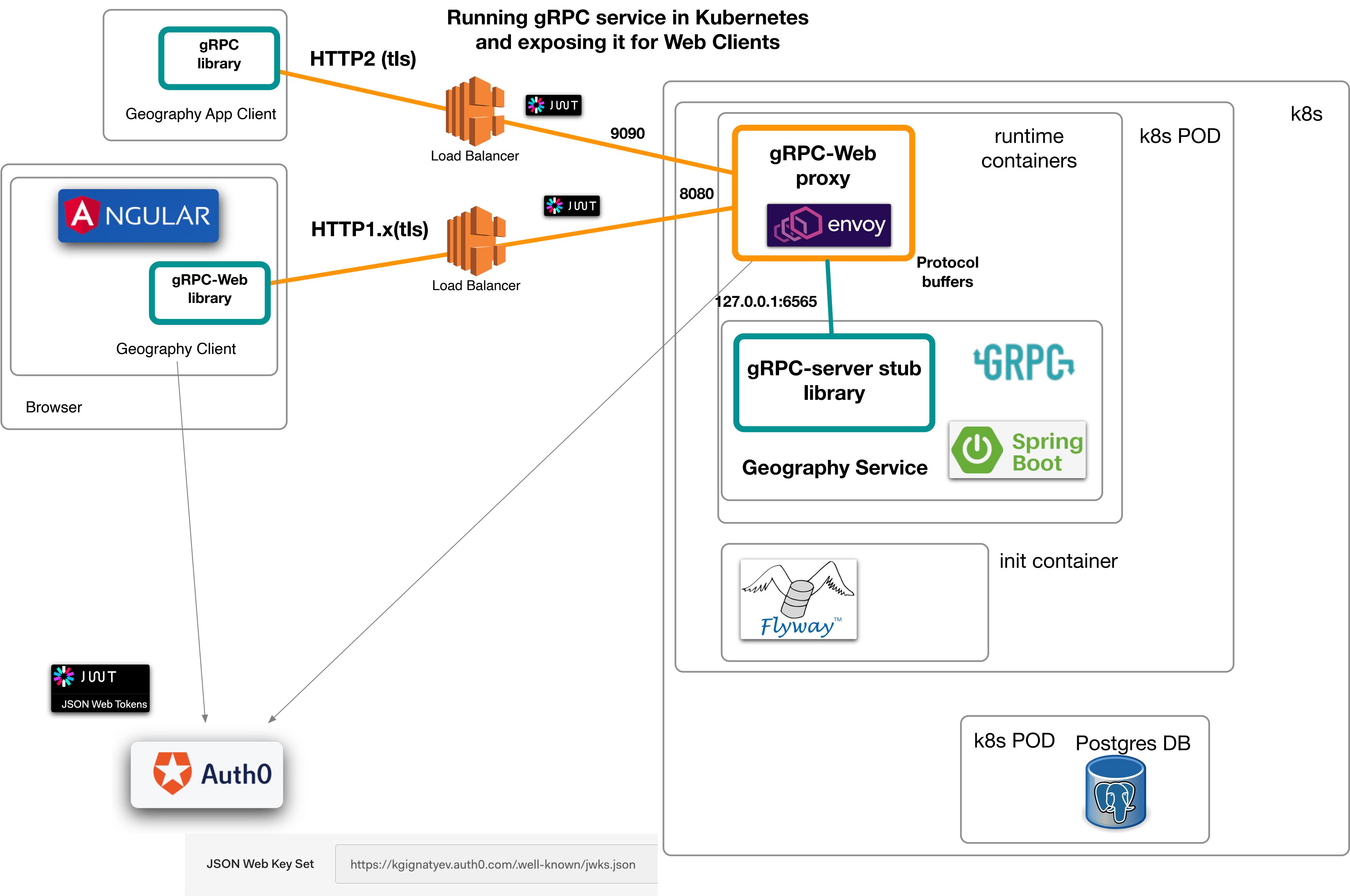
- \* Interface First - architecture
- \* Externalized:
  - authentication;
  - monitoring;
  - traffic management;
- \* Change Planning and Change Propagation
- \* Good-Old code generation
  - client and server protocol handlers;
  - DAO (persistence layer);



# Running gRPC service and exposing it for Web Clients



# Running gRPC service in Kubernetes and exposing it for Web Clients



```

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

```

463    464    465    466    467

*/\* @deprecated \*/*  
*@Deprecated*  
**public void** findAreasContaining(LatLng request, StreamObserver<AreasList> responseObserver) {  
 ServerCalls.asyncUnimplementedUnaryCall(GeographyGrpc.getFindAreasContainingMethod(), responseObserver);  
}

The diagram illustrates the build process for a Geographical API. It starts with a Protobuf file containing message definitions like `LatLng`, `Polygon`, and `Area`. A specific method, `findAreasContaining`, is annotated with `@deprecated` and `@Deprecated`. This method is part of a `Geography` service. The service also includes other RPCs such as `CreateArea`, `GetAreaByID`, `UpdateArea`, `DeleteAreaByID`, `FindAreas`, and `FindAreasBy`.

A red arrow points from the `findAreasContaining` method in the Protobuf file to a yellow box labeled "Artifacts Repository". This box lists various artifact management platforms: APT\*, Composer\*, Conan\*, CPAN\*, Docker, ELPA\*, Git LFS, Helm\*, Maven, npm, NuGet, P2\*, PyPI, R\*, Raw, RubyGems, and yum. Below this is a large yellow arrow pointing to a white box labeled "JFROG ARTIFACTORY". A note in this box states: "Note: we can produce libraries for other languages too if needed".

# 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 with a sidebar and a main content area.

**Project:** geography-client (~dev/kgi/type-safety-end-to-end/geography-cl)

**File Tree:**

- .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` and `interface.md` files. A large yellow arrow points from the sidebar towards the right-hand content area.

**Content Area:**

**interface.md**

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	

## Server Side - we 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 }
```

The screenshot shows a file explorer interface with the following directory structure:

- geography-grpc-client** (~ /dev/kgi/type-safety)
  - .idea
  - config
    - jwt.txt
  - src
    - main
      - kotlin
        - com.kgi.geography\_service.client
          - CallContext.kt
          - GeographyServiceGrpcClient.kt
          - OutgoingCallInterceptor
    - test
      - kotlin
        - com.kgi.geography\_service.client
          - AuthenticatedCallTest
    - resources

```
OutgoingCallInterceptor.kt

rpc:
  client:
    geography-grpc-server:
      address: 'static://localhost:6565'
      enableKeepAlive: true
      keepAliveWithoutCalls: true
      negotiationType: plaintext

    geography-grpc-server-auth:
      address: 'static://localhost:9090'
      enableKeepAlive: true
      keepAliveWithoutCalls: true
      negotiationType: plaintext

  logging:
    level:
      root: "INFO"
```

## **Direct connection to the service port**

# Proxied connection that requires valid JWT token

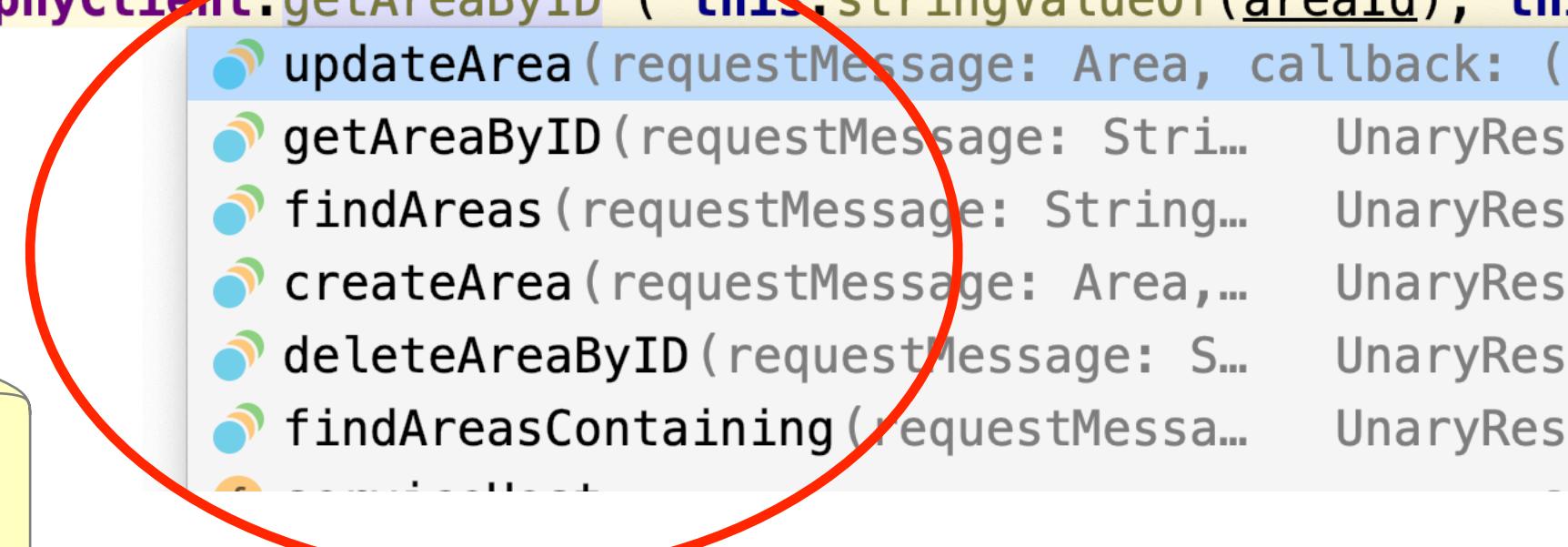
```
56 >     fun testWithStub( stub: GeographyGrpc.GeographyBlockingStub){  
57         val foundAreas :GeographyOuterClass.AreasList! = stub.findAreas(StringValue.of( value: "" ))  
58         foundAreas.itemsList.forEachIndexed { index, a  
59             println("found area ${index}: ${area}")  
60         }  
61     }  
62 }  
63 }
```

## Client Code

### geography.service.ts

```
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



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 }

```
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>
```

ID	Name
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

Note: that even in markup we use strongly typed accessors and mutators, if something changes, for example property is renamed, then we get compile time errors and we have full support for refactoring from IDEs

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

Name: r23414

Senegal  
The Gambia  
Guinea-Bissau  
Guinea  
Sierra Leone  
Liberia  
Côte d'Ivoire  
Burkina Faso  
Togo  
Ghana  
Nigeria  
Cameroon  
Equatorial Guinea  
Gabon  
Congo  
Angola  
Mozambique  
Zambia  
Malawi  
Tanzania  
Kenya  
Uganda  
Sudan  
South Sudan  
Chad  
Central African Republic  
DRC  
Rwanda  
Burundi  
Horn of Africa

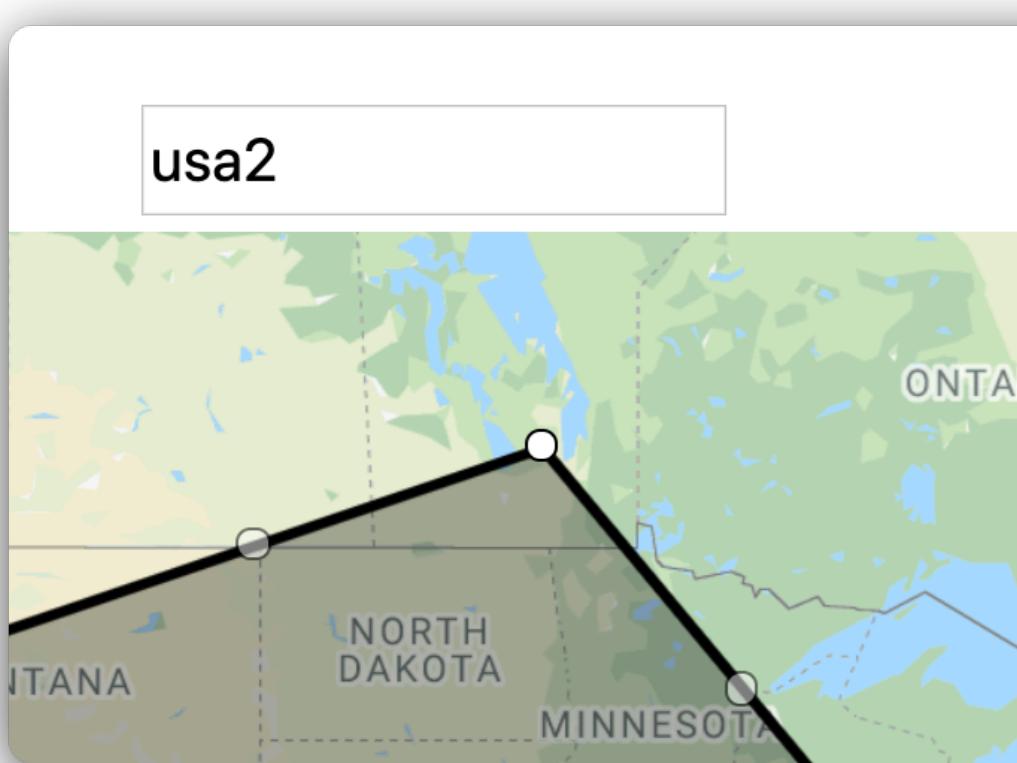
Google Map data ©2020 Google, ORION-ME Terms of Use

Cancel Save

```
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")

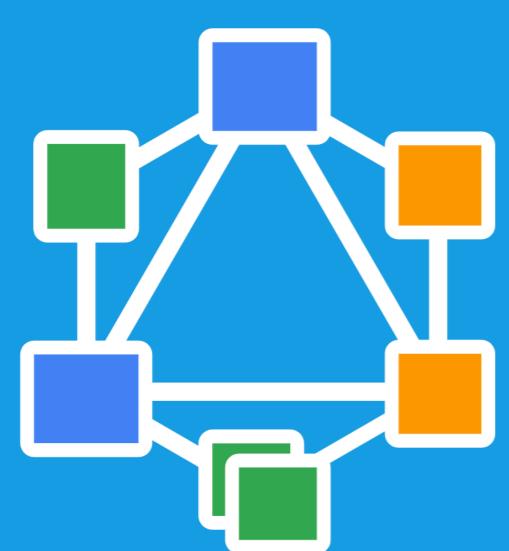
### .proto definition

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

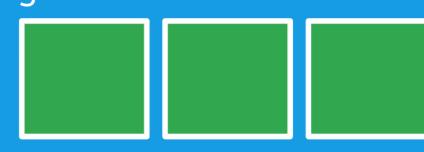
```
12
13  keys: string[] = Object.keys( AreaType );
14  at = AreaType;
15
16  constructor() { }
17
18  ngOnInit() {
19    console.log( this.keys )
20  }
21
22  setAreaType(typeKey: any) {
23    this.area.setAreaType( this.at[typeKey] );
24    console.info(typeKey);
25  }
26
27  getAreaType() {
28  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

## Rejoiner

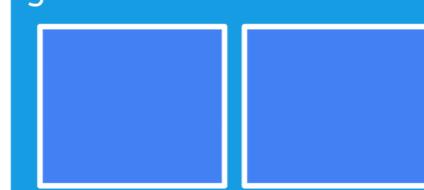
Uniform GraphQL API  
served over  
HTTP and  
gRPC



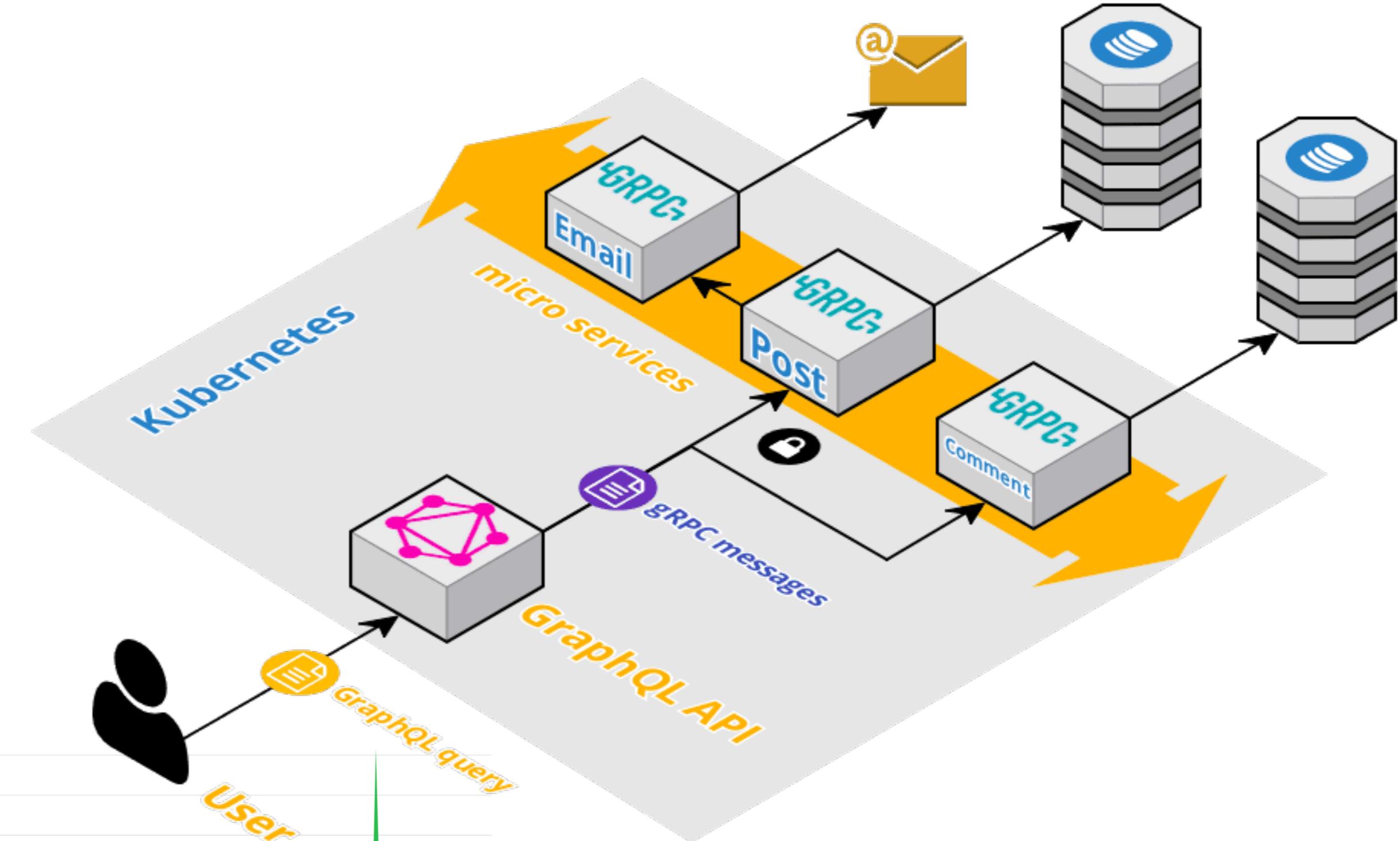
Java  
gRPC Microservice



Go  
gRPC Microservice



Python  
gRPC Microservice



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

