

gRPC

Comment communiquer en gRPC avec Quarkus

gRPC ?

Remote **P**rocedure **C**all développé par Google

D'autres protocoles existent

- RMI (Java **R**emote **M**ethod **I**nvocation) => TCP/IP
- SOAP (**S**imple **O**bject **A**ccess **P**rotocol) => HTTP XML
- REST (**R**epresentational **S**tate **T**ransfert) => HTTP JSON

gRPC => HTTP 2 / **D**onnées **b**inaires

HTTP 2

- Multiplexing
- Streaming
- Gain en performance

Comment ça marche

- protobuf : Interface **D**efinition **L**anguage

```
service TrainPositionService {  
    rpc GetPosition(GetPositionRequest) returns (TrainPosition);  
    rpc ReportPosition (PositionRequest) returns (PositionResponse);  
}
```

```
message GetPositionRequest {  
    int64 trainId = 1;  
}
```

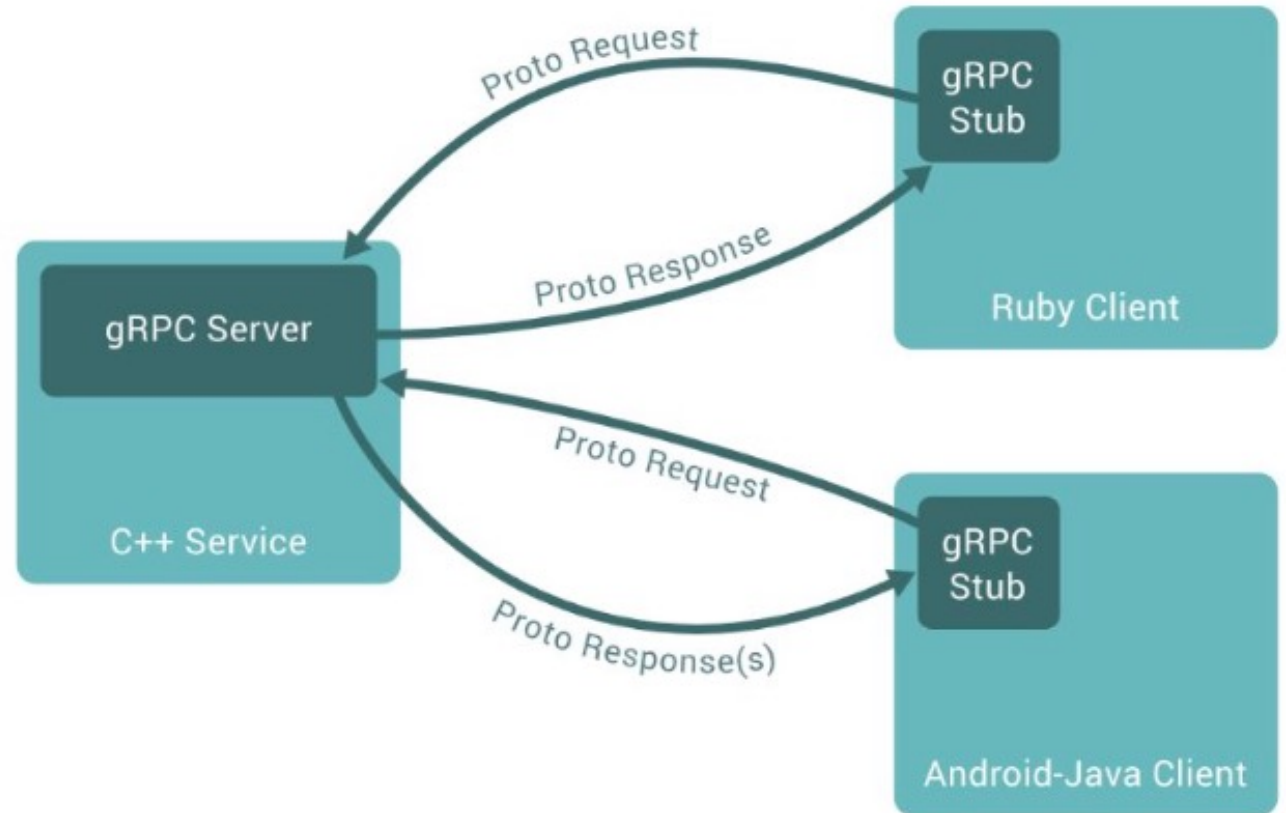
```
message TrainPosition {  
    int64 trainId = 1;  
    int64 trackId = 2;  
    float kilometer = 3;  
}
```

```
message PositionRequest {  
    int64 trainId = 1;  
    int64 trackId = 2;  
    float kilometer = 3;  
}
```

```
message PositionResponse {  
    TrainPosition actualTrain = 1;  
    TrainPosition previousTrain = 2;  
    TrainPosition nextTrain = 3;  
}
```

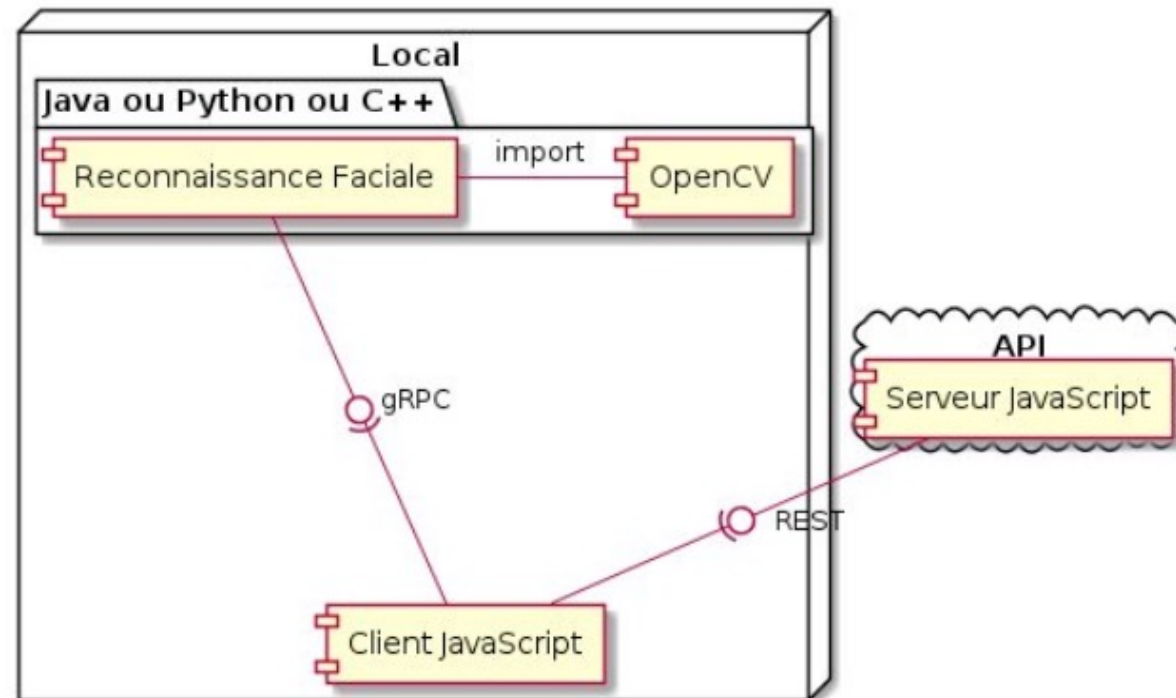
Comment ça marche

- Permet de générer un client / serveur pour le langage ciblé



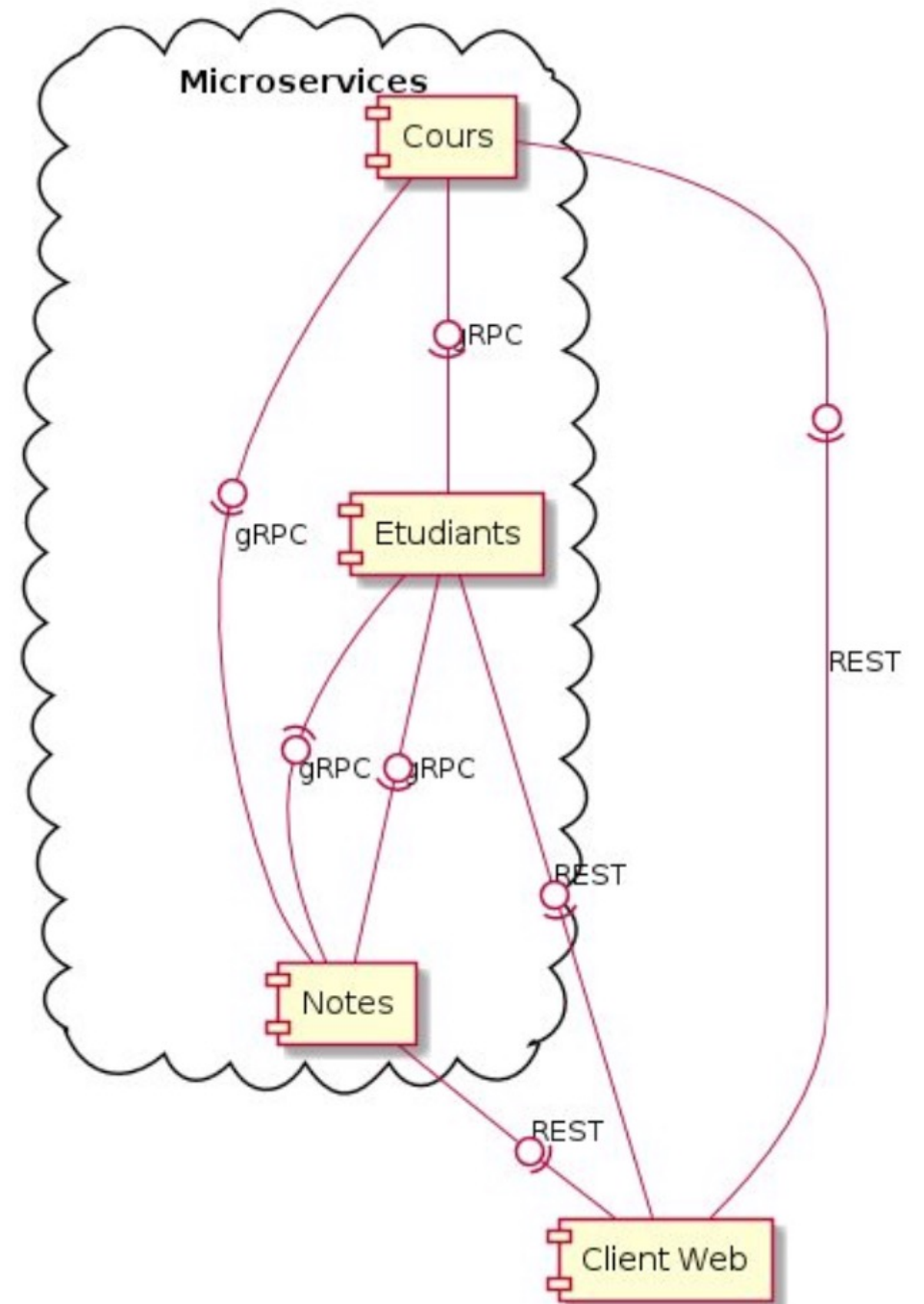
Exemple d'utilisation

- Intégration d'une bibliothèque de reconnaissance d'images dans un langage tiers



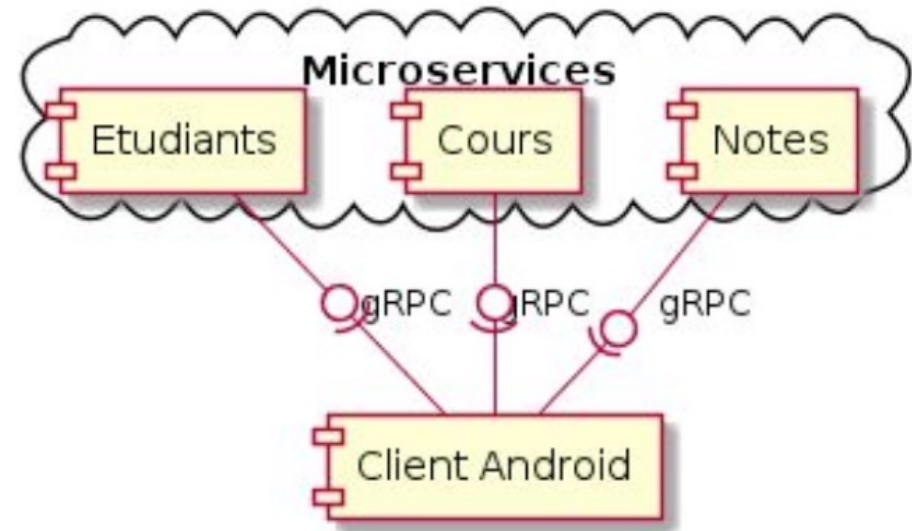
Exemple d'utilisation

- Communication interne dans une architecture micro-services
- Exposition API externes en REST
- Parfait pour du service à service



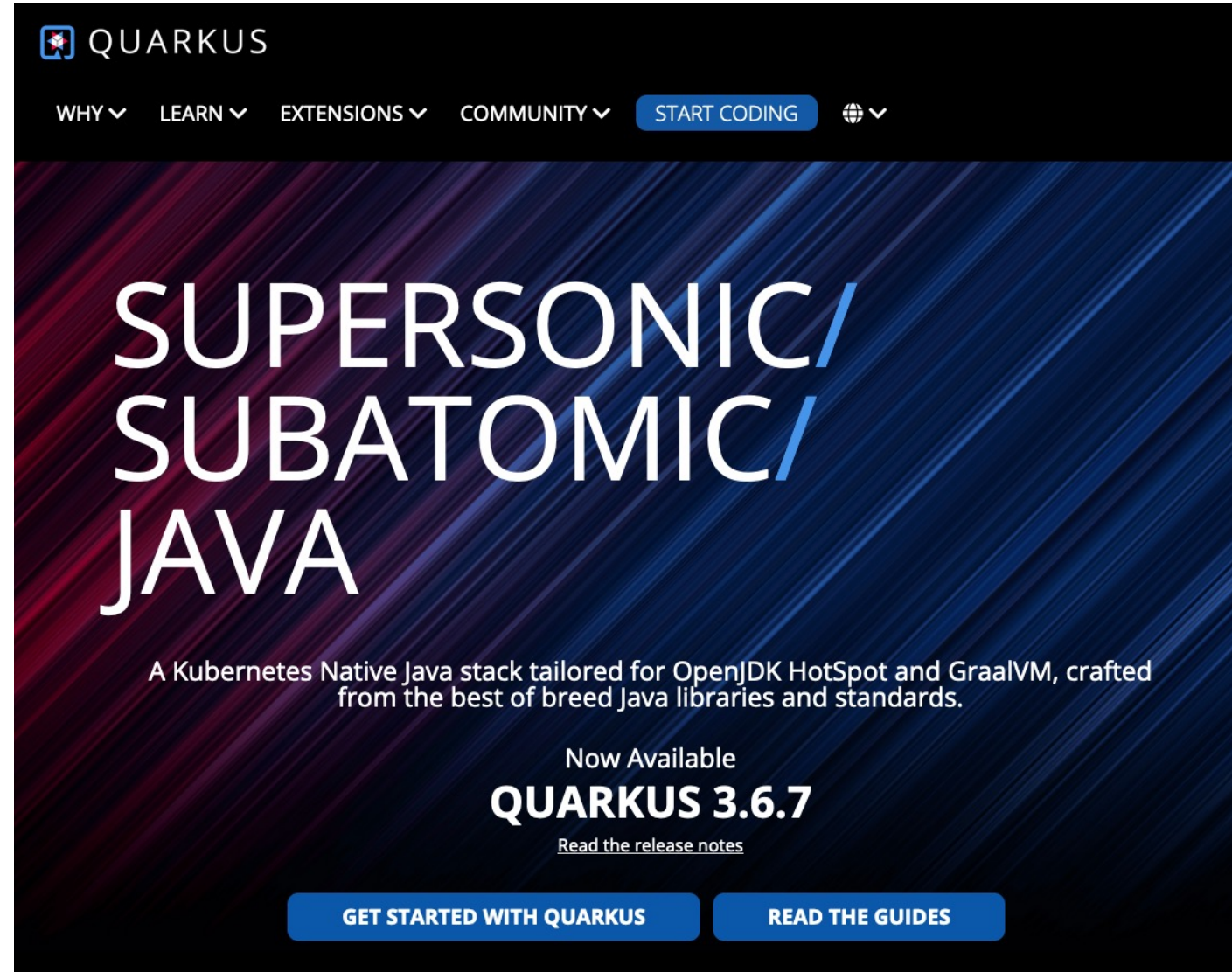
Exemple d'utilisation

- Communication avec application mobile Android / iOS



Quarkus

- Alternative à Spring
 - Plus légère
- Temps de démarrage plus rapide
 - Parfait pour des micro applications sur le cloud
- Compilation native
- Système d'extensions
- Standards Java



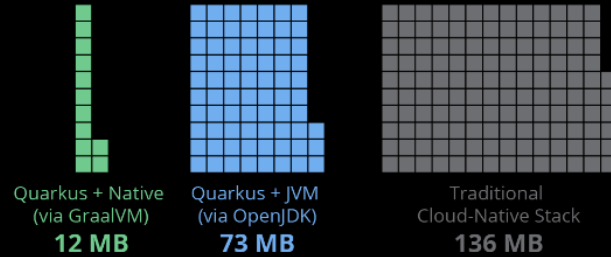
Quarkus

Quarkus offers unequalled performance

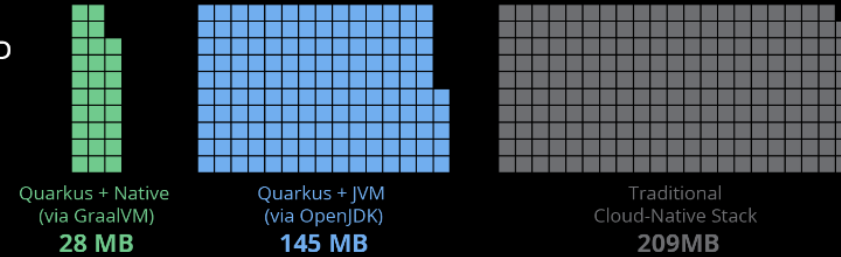
Memory (RSS) in Megabytes*

*Tested on a single-core machine

REST



REST + CRUD



BOOT + First Response Time

REST



REST + CRUD



Quarkus

- <https://code.quarkus.io/>
- Intègre une extension gRPC
- Starter code disponible

The screenshot displays the Quarkus code generator web interface. At the top, the Quarkus logo and version 3.6 are shown, along with a link to 'Back to quarkus.io' and a note about 'Enterprise Support'. The main section is titled 'CONFIGURE YOUR APPLICATION' and contains three input fields: 'Group' (org.acme), 'Artifact' (code-with-quarkus), and 'Build Tool' (Maven). A 'MORE OPTIONS' button is located to the right of these fields. Below the configuration section, there is a search bar with the text 'origin:platform grpc' and a 'Filters' dropdown. The search results show 'Extensions found by origin: 2 in platform 1 in other'. Two extensions are listed: 'gRPC [quarkus-grpc]' (checked) and 'Camel gRPC [camel-quarkus-grpc]' (unchecked). Both extensions have a 'STARTER-CODE' button next to them. On the right side of the interface, there is a blue button labeled 'Generate your application (\` + ↵)' with a dropdown arrow. Below this button, a 'Selected Extensions' panel shows the same two extensions with their respective 'STARTER-CODE' buttons.

QUARKUS 3.6
io.quarkus.platform

Back to quarkus.io Available with Enterprise Support

CONFIGURE YOUR APPLICATION

Group org.acme

Artifact code-with-quarkus

Build Tool Maven MORE OPTIONS

Filters origin:platform grpc

Extensions found by origin: 2 in platform 1 in other

☒ gRPC [quarkus-grpc] STARTER-CODE

Serve and consume gRPC services

☐ Camel gRPC [camel-quarkus-grpc]

Expose gRPC endpoints and access external gRPC endpoints

2 Generate your application (\` + ↵)

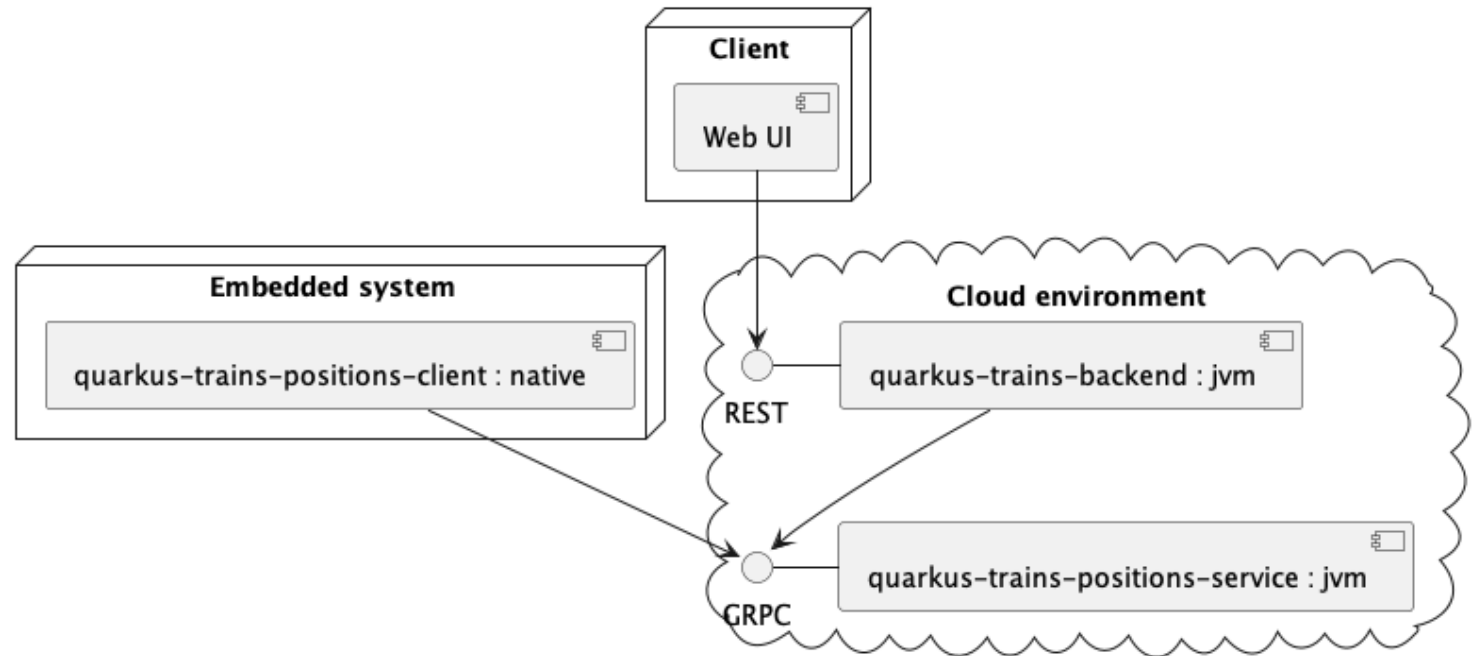
Selected Extensions

- RESTEasy Classic STARTER-CODE

- gRPC STARTER-CODE

Example codé avec Quarkus

- API REST : Liste de trains
- Service gRPC : Positions des trains
- Client gRPC natif : Position de chaque train



<https://github.com/niushapaks/quarkus-trains/>