Create your Digital Twin in days, not months.

How to connect your PLC/Sensors to the ISO 10303 Repository using the Arrowhead IoT/CPS open source software

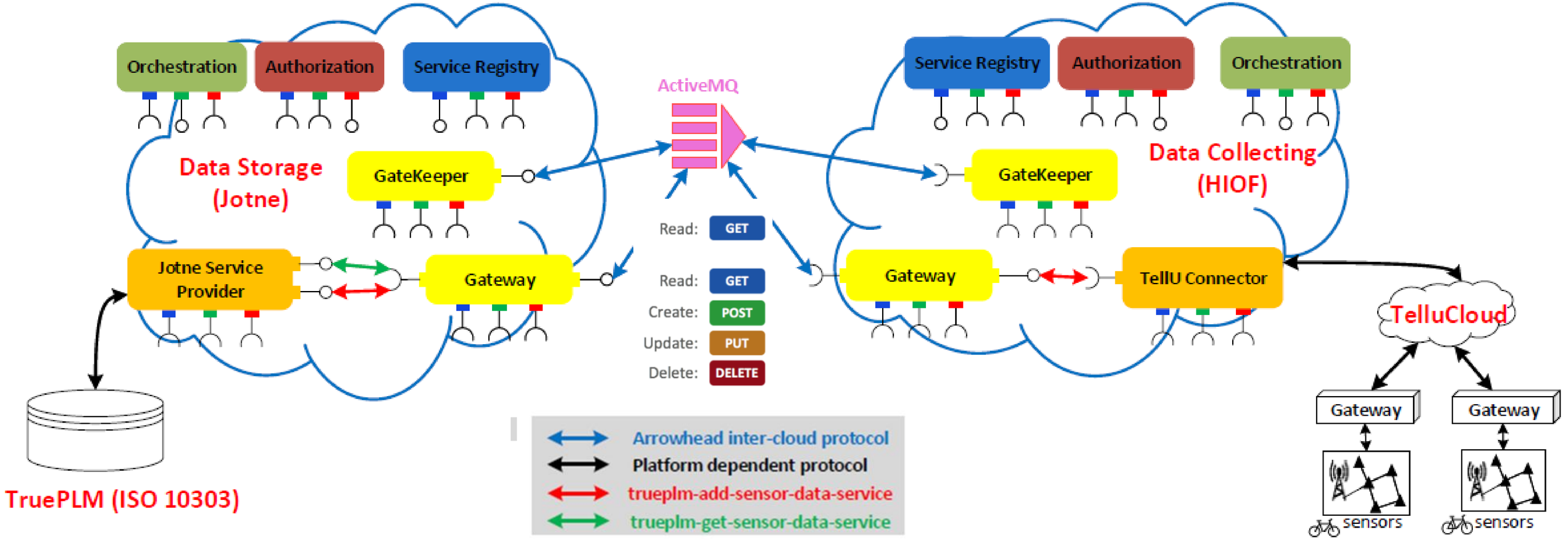
Roman Filatov, Jotne

Roman.Filatov@jotne.com

ARROWHEAD TOOLS
KYKLOS 4.0
CHANGE2TWIN
Supported by advanced EU H2020 projects
DAIS
DRYADS

Table of the content

PLM using Arrowhead 4.2



Setup your own Arrowhead cloud

Use the following links:

https://github.com/arrowhead-f/core-java-spring#quick-start-guide

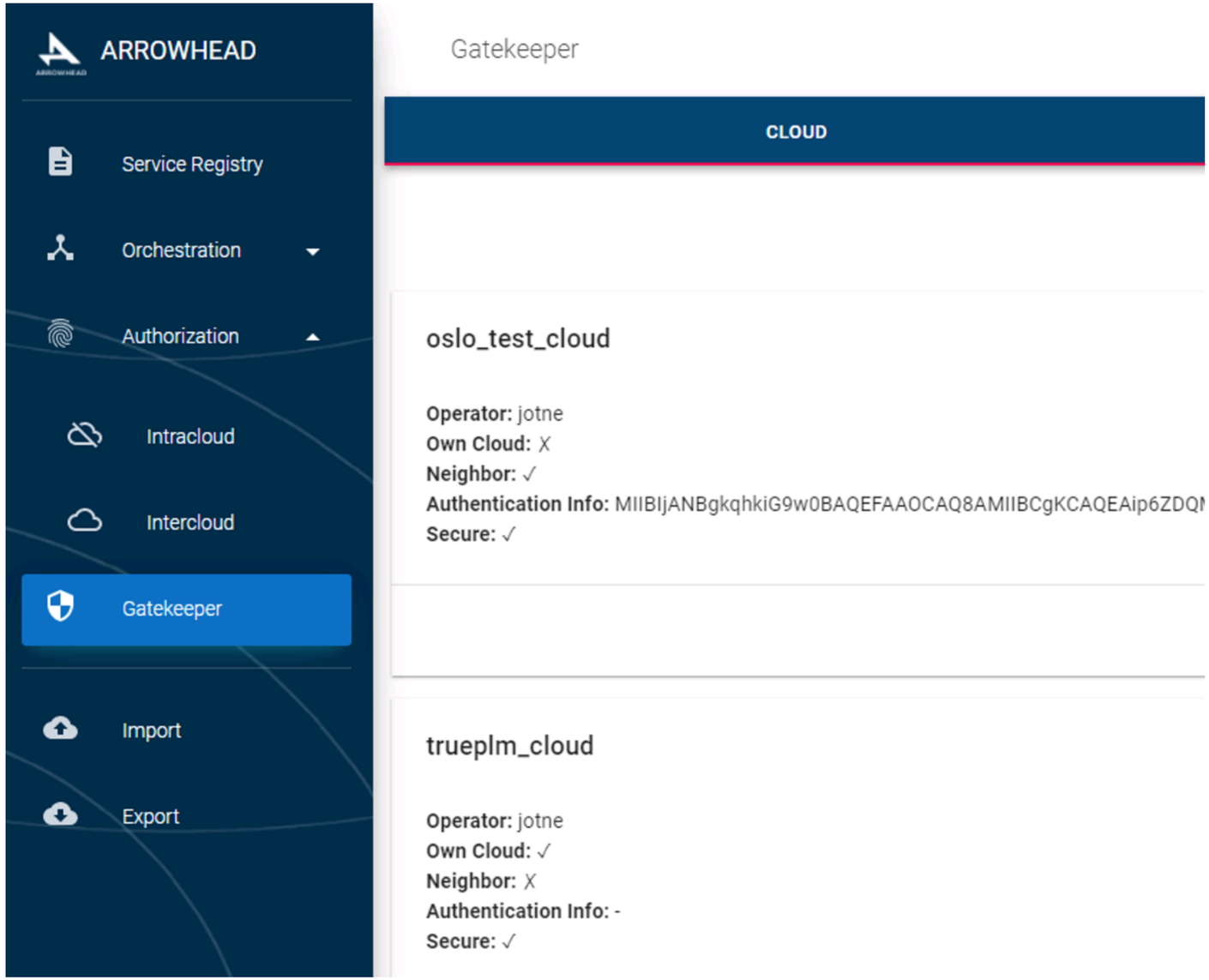
https://github.com/arrowhead-f/core-java-spring

https://github.com/arrowhead-f/core-java-spring#certificates

https://www.arrowhead.eu/

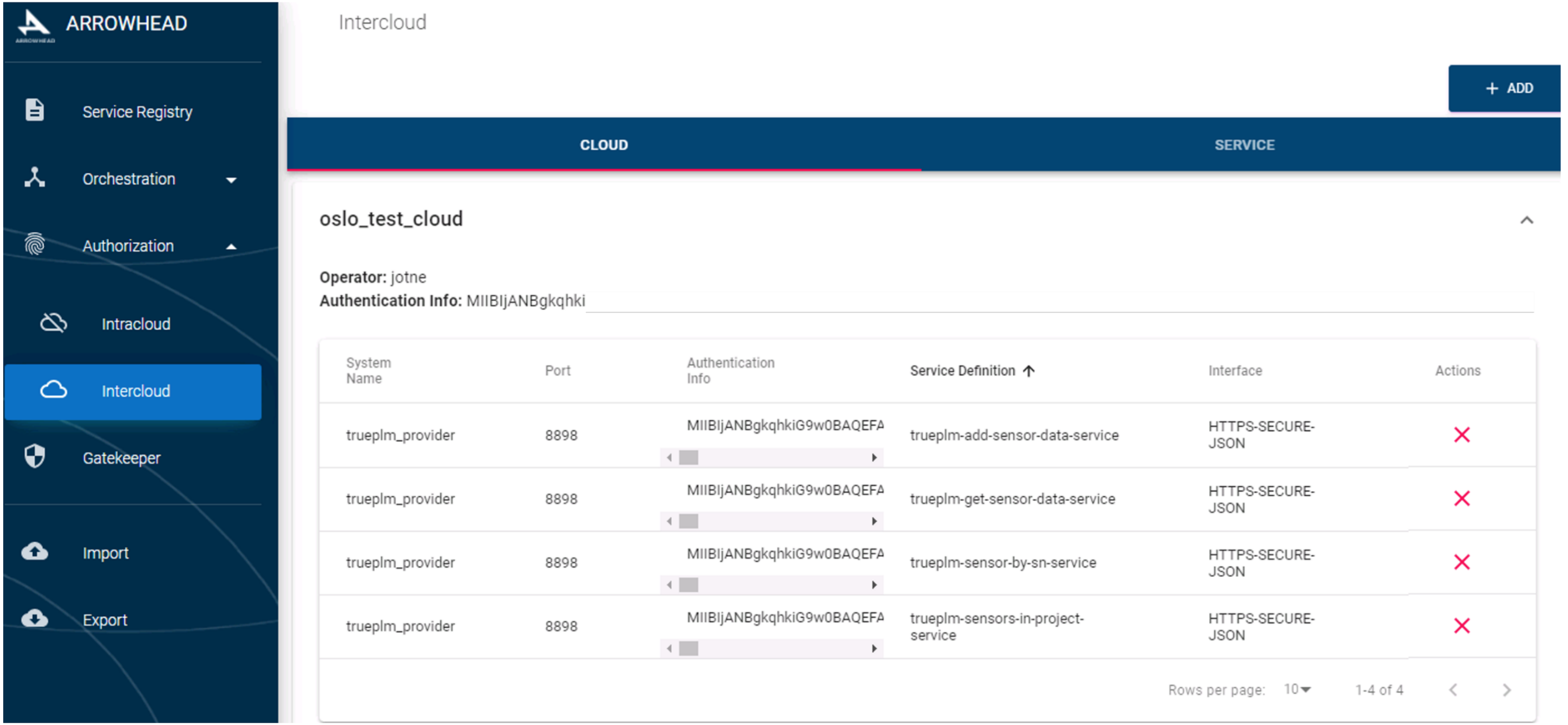
https://www.youtube.com/user/ArrowheadProject

Add Jotne truePLM cloud as the neighbour in the Gatekeeper



and ask Jotne to add your cloud

Ask Jotne to add the authorisation rules to get access to the truePLM provider services from your cloud.

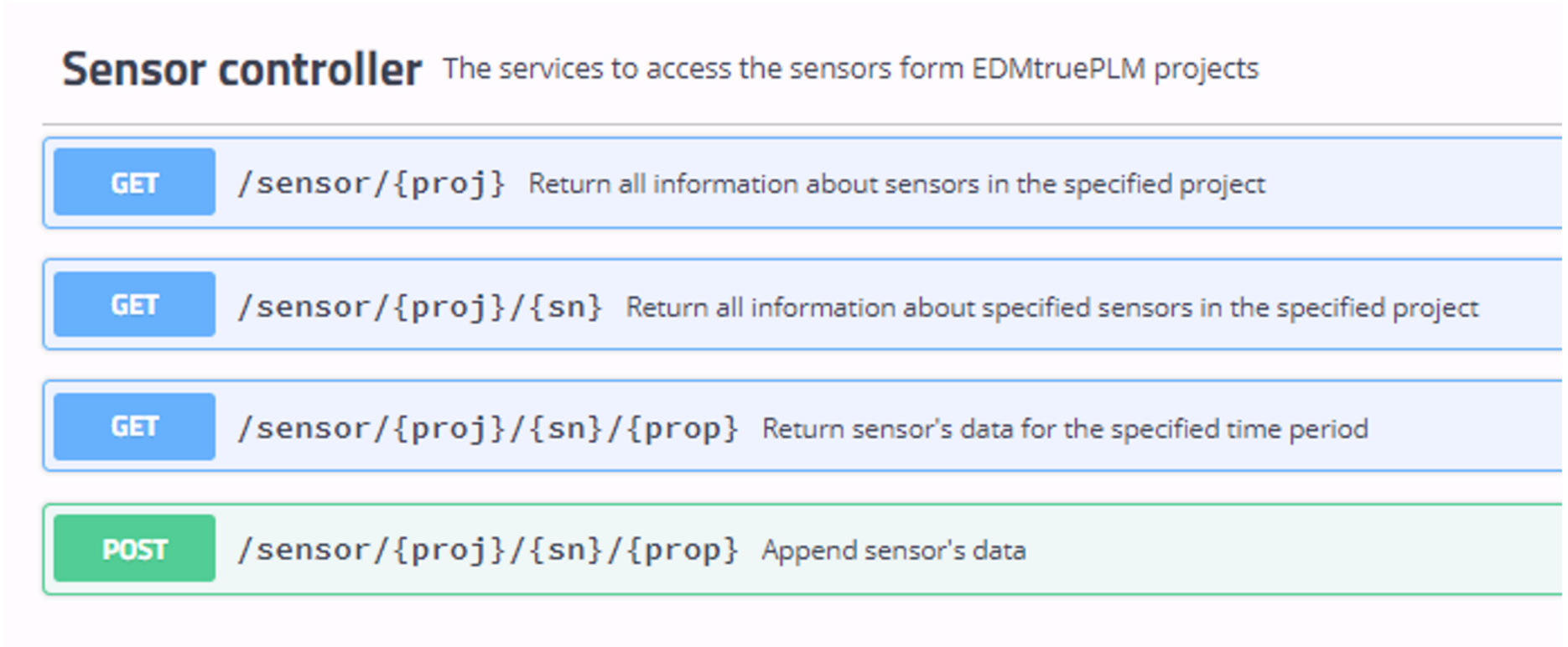


Create your own Consumer to get the data from the sensors and write it to the truePLM database by the truePLM provider services.

Use examples and documentation from the Arrowhead repository

https://github.com/arrowhead-f/sos-examples-spring/tree/master/demo-exchange-rate-intercloud

TruePLM provider services



Append data service

POST `/sensor/{proj}/{sn}/{prop}` Append sensor's data

Parameters

Name	Description
proj * required `string` *(path)*	Project name
prop * required `string` *(path)*	Aggregate property's name
si * required *(body)*	Sensor's data **Example Value** | Model `{ "SensorData": [ { "SensorMeasurement": [ { "Measurement": "string", "value": "string" } ], "timestamp": "string" } ], "SensorType": "string", "id": "string" }` **Parameter content type** application/json
sn * required `string` *(path)*	Sensor's serial number

Get data service

GET `/sensor/{proj}/{sn}/{prop}` Return sensor's data for the specified time period

Parameters

Name	Description
fromDate * required string *(query)*	Start of the time period 11/04/2020
proj * required string *(path)*	Project name Bike
prop * required string *(path)*	Aggregate property's name urn:rdl:Bike:altitude list
sn * required string *(path)*	Sensor's serial number 1180322-02701
toDate * required string *(query)*	End of the time period 12/04/2020

Server response

Code	Details

200	**Response body**
	{
	"id": "1180322-02701",
	"SensorType": "urn:plcs:rdl:ArrowHead:USB_GPS",
	"SensorData": [
	{
	"timestamp": "1587566605",
	"SensorMeasurement": [
	{
	"value": "19.417",
	"Measurement": "altitude"
	},
	{
	"value": "105.488",
	"Measurement": "altitude_err"
	},
	{
	"value": "-0.177",
	"Measurement": "vspeed"
	},
	{
	"value": "210.98",
	"Measurement": "vspeed_err"
	}
	]
	},

Get sensors info service

GET /sensor/{proj} Return all information about sensors in the specified project

Parameters

Name	Description
proj * required string *(path)*	Project name Bike

Server response

Code	Details
200	Response body


```
    "aggrProps": null,
    "aggrPropTypes": null,
    "instanceID": 171798746529
  },
  {
    "name": "13483027",
    "description": "Sensor 2",
    "serialNumber": "13483027",
    "type": "urn:plcs:rdl:ArrowHead:RUUVITAG",
    "aggrProps": null,
    "aggrPropTypes": null,
    "instanceID": 171798748523
  },
  {
    "name": "218991",
    "description": "Sensor1 from RUUVI",
    "serialNumber": "218991",
    "type": "urn:plcs:rdl:ArrowHead:RUUVITAG",
    "aggrProps": null,
    "aggrPropTypes": null,
    "instanceID": 171798748460
  },
```

Get sensor info service

GET `/sensor/{proj}/{sn}` Return all information about specified sensors in the specified project

Parameters

Name	Description
proj * required string *(path)*	Project name Bike
sn * required string *(path)*	Sensor's serial number 1180322-02701

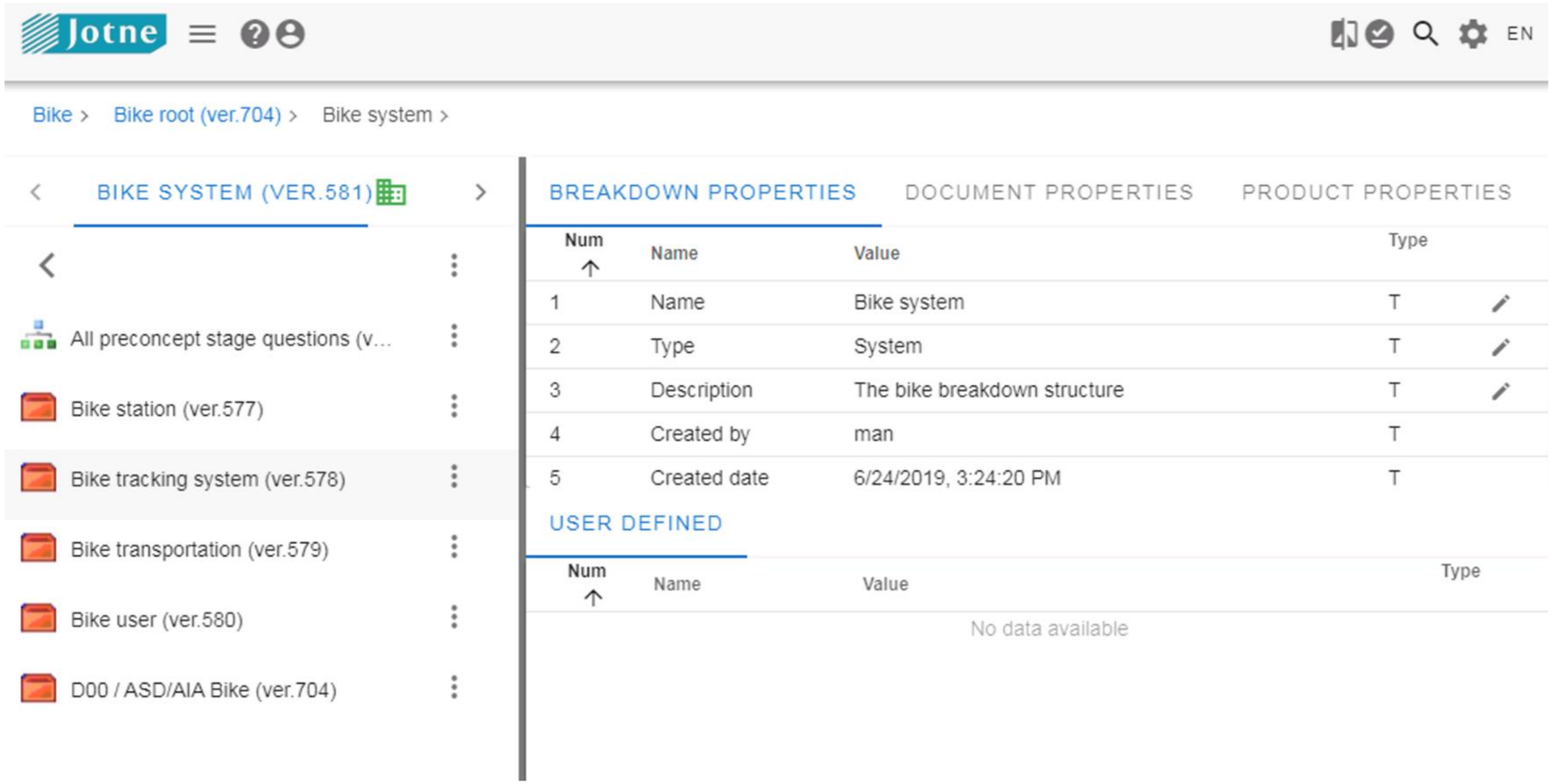
Server response

Code	Details
200	**Response body**

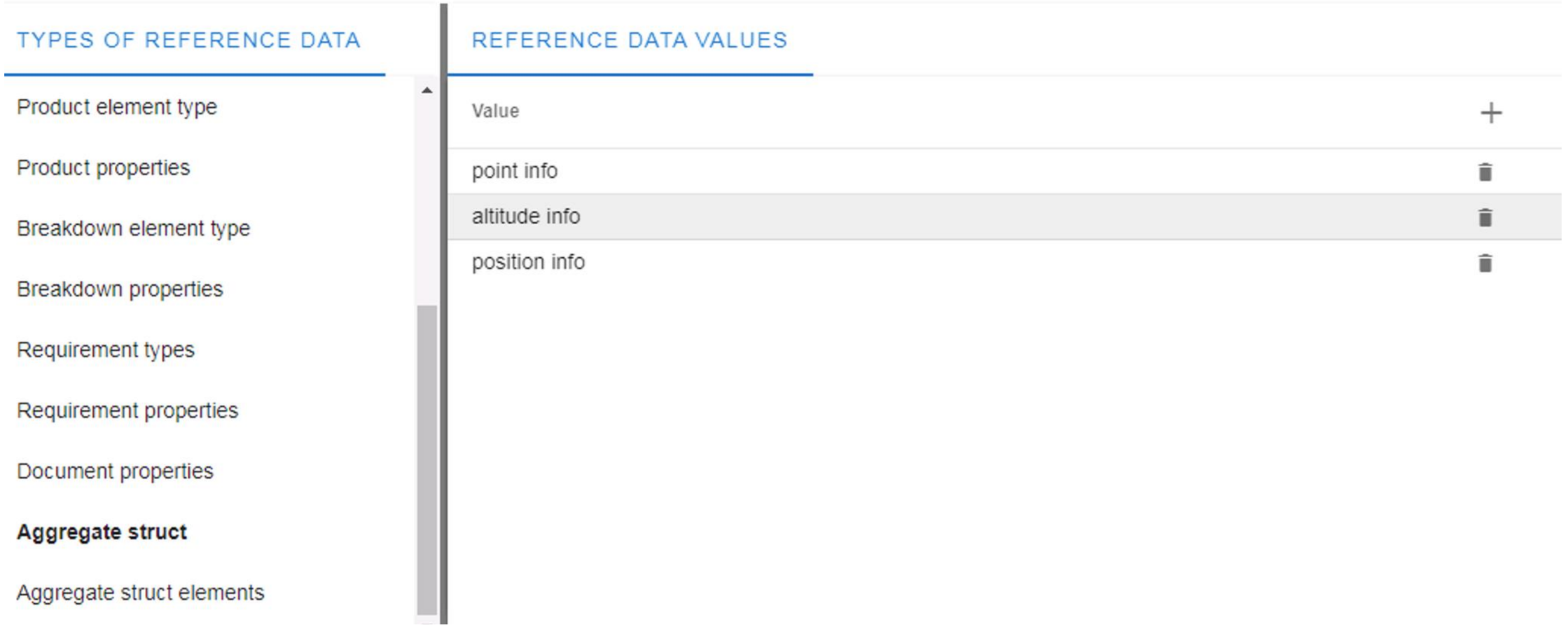

```
{
  "name": "Gateway",
  "description": "Gateway for the ruuvitag sensors",
  "serialNumber": "1180322-02701",
  "type": "urn:plcs:rdl:ArrowHead:USB_GPS",
  "aggrProps": [
    "urn:plcs:rdl:ArrowHead:position_series",
    "urn:plcs:rdl:ArrowHead:altitude_series",
    "urn:rdl:Bike:altitude list",
    "urn:rdl:Bike:position list"
  ],
  "aggrPropTypes": [
    "urn:plcs:rdl:ArrowHead:GPSposition",
    "urn:plcs:rdl:ArrowHead:GPSaltitude",
    "urn:rdl:Bike:altitude info",
    "urn:rdl:Bike:position info"
  ],
  "instanceID": 171798750672
}
```

Create new project in the truePLM

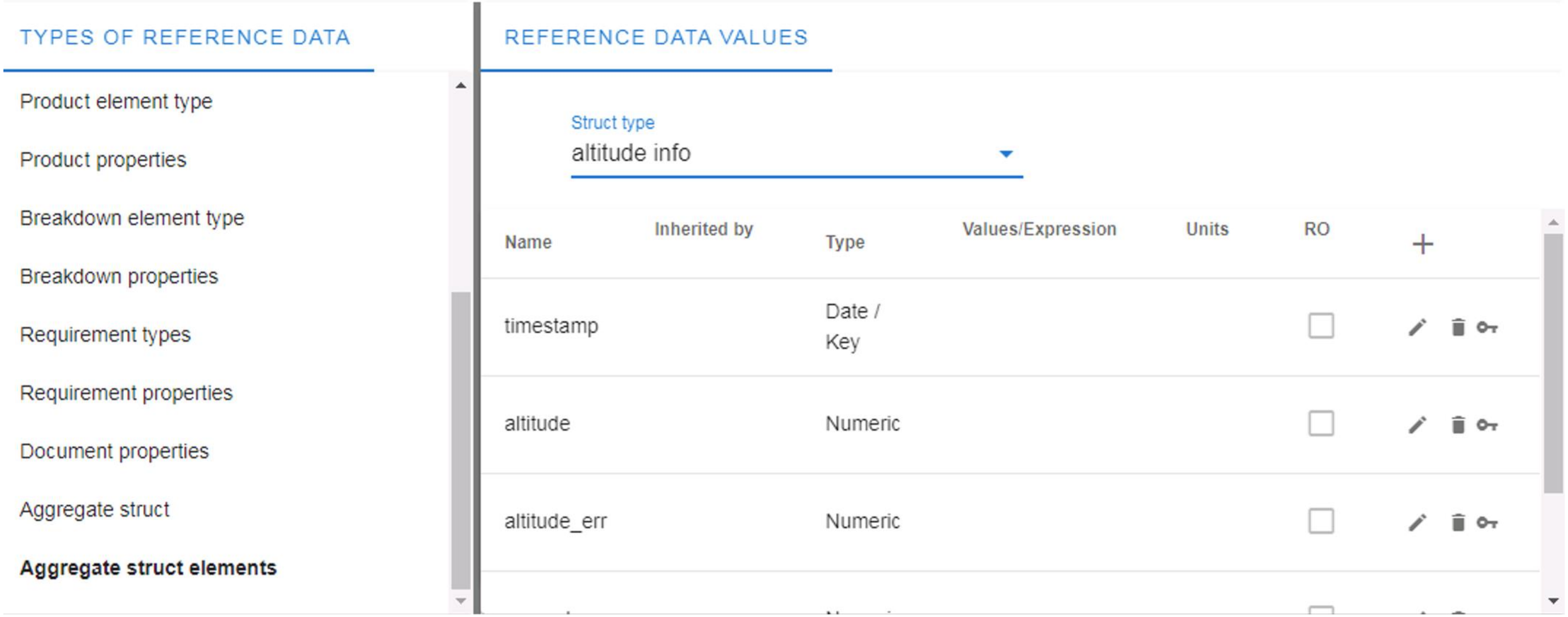
And define the breakdown structure



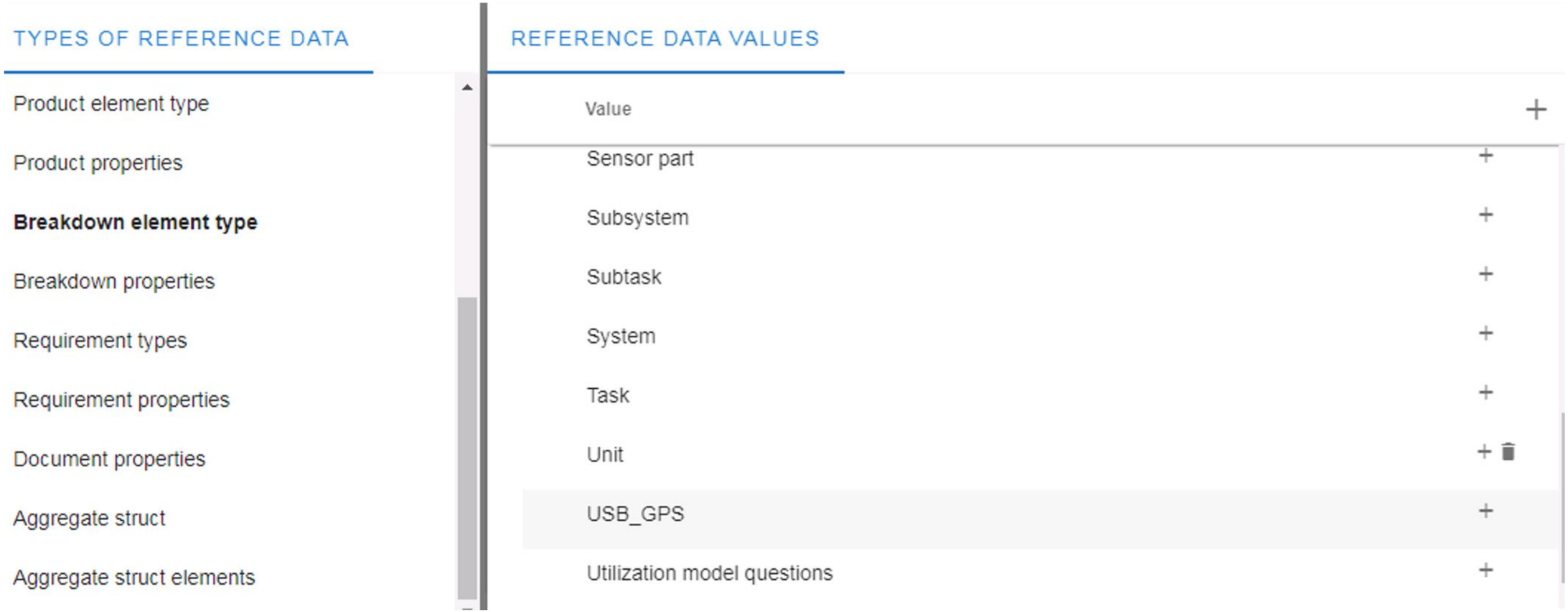
Define the structure for your sensor data values



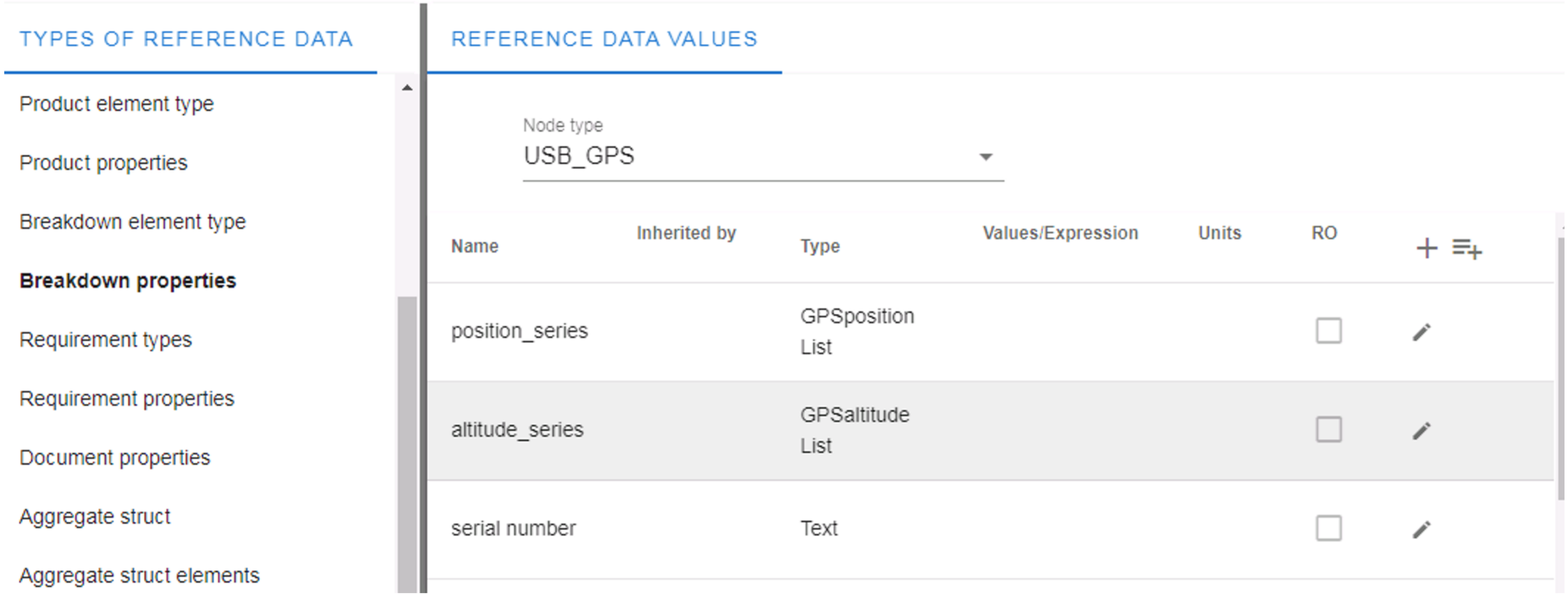
Define the elements of the structure according to the value set of the sensor



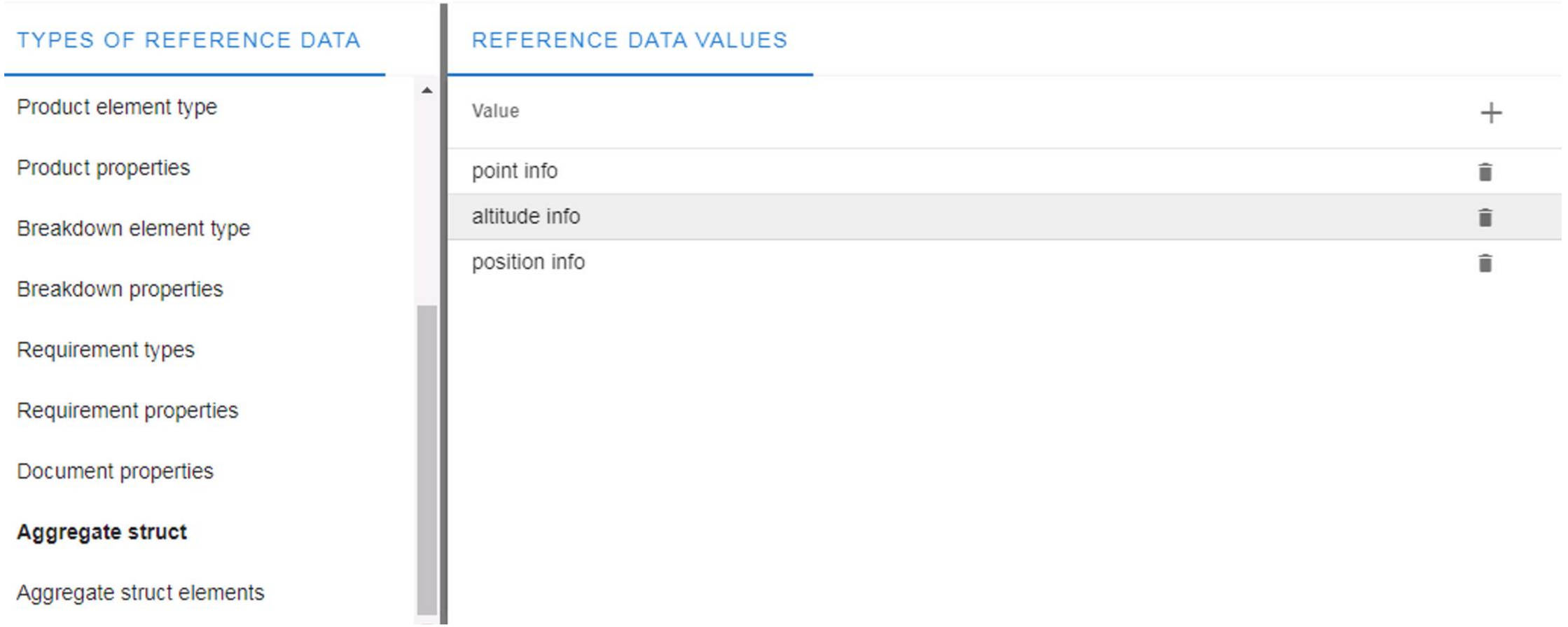
Define the breakdown element type for the sensor



Define the properties for the breakdown element type of the sensor



Define the structure for your sensor data values



Direct connection to the EDMtruePLM system with the REST API

Page URL: https://demo.jotne.com/EDMtruePLM/swagger.html

Api Documentation 1.0

[Base URL: demo.jotne.com/EDMtruePLM]
https://demo.jotne.com//EDMtruePLM/v2/api-docs

Api Documentation

Terms of service
Apache 2.0

Admin controller The common functions for administrative functionality

Authorization controller The common authorization functions

Baseline controller The functions for baseline (also called snapshots) functionality

Breakdown controller The common functions for breakdown structure functionality