

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

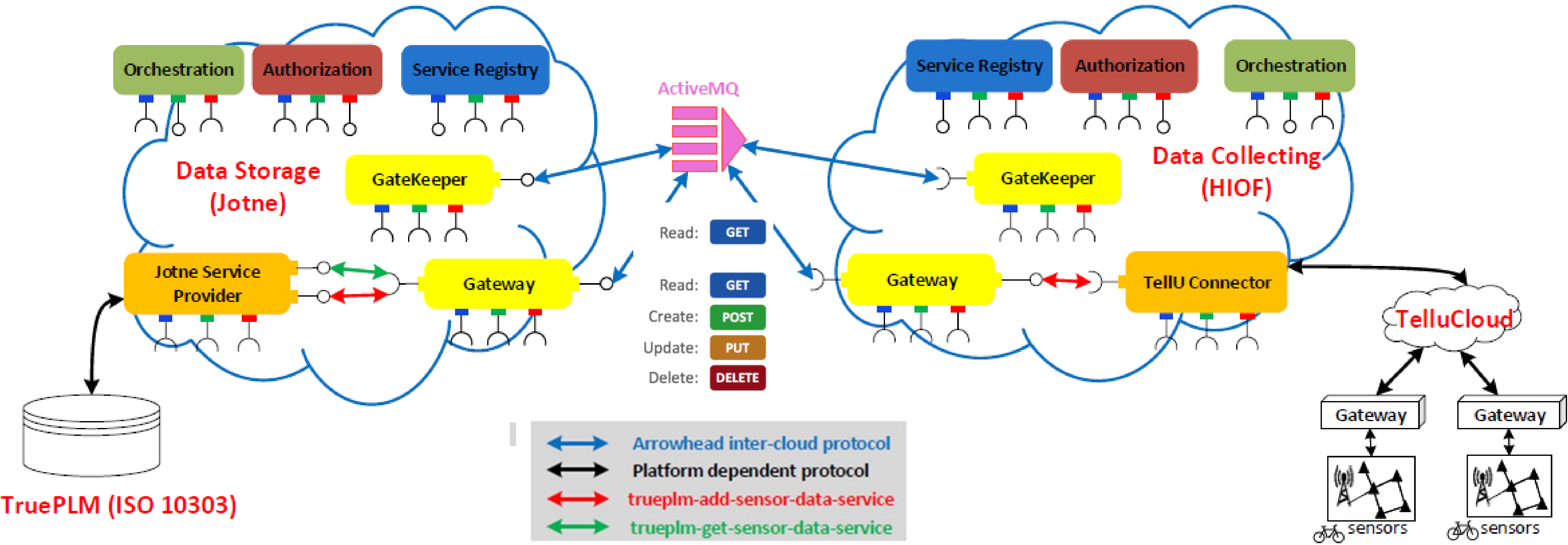


Supported by advanced EU H2020 projects

Table of the content

- [PLM using Arrowhead 4.2](#)
- [Setup your own Arrowhead cloud](#)
- [Add Jotne truePLM cloud as the neighbour in the Gatekeeper](#)
- [Ask Jotne to add the authorisation rules](#)
- [Create your own Consumer](#)
- [TruePLM provider services](#)
- [Append data service](#)
- [Get data service](#)
- [Get sensors info service](#)
- [Get sensor info service](#)
- [Create new project in the truePLM](#)
- [Define the structure for your sensor data values](#)
- [Define the elements of the structure](#)
- [Define the breakdown element type for the sensor](#)
- [Define the properties for the breakdown element type of the sensor](#)
- [Create the breakdown element for the sensor](#)

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>

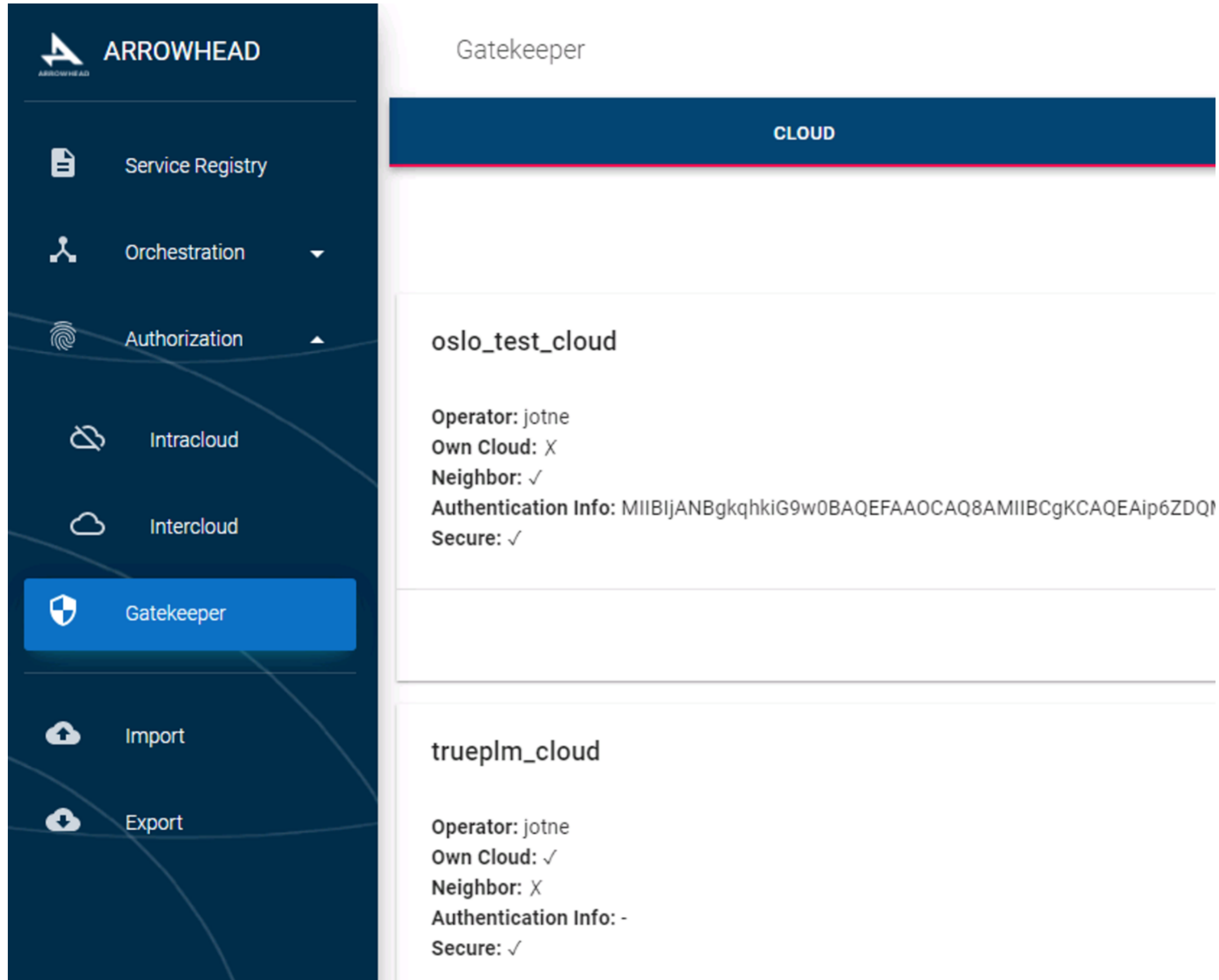
<https://github.com/arrowhead-f/core-java-spring>

<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



The screenshot shows the Arrowhead Gatekeeper interface. The left sidebar contains navigation options: Service Registry, Orchestration, Authorization, Intracloud, Intercloud, Gatekeeper (highlighted), Import, and Export. The main content area is titled 'Gatekeeper' and has a 'CLOUD' header. It displays two cloud configurations:

- oslo_test_cloud**
 - Operator: jotne
 - Own Cloud: X
 - Neighbor: ✓
 - Authentication Info: MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAIp6ZDQI
 - Secure: ✓
- trueplm_cloud**
 - Operator: jotne
 - Own Cloud: ✓
 - Neighbor: X
 - Authentication Info: -
 - Secure: ✓

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.

The screenshot shows the Arrowhead Intercloud interface. On the left is a dark blue sidebar with navigation options: Service Registry, Orchestration, Authorization, Intracloud, Intercloud (highlighted), Gatekeeper, Import, and Export. The main content area is titled 'Intercloud' and features a '+ ADD' button in the top right. Below this is a header with 'CLOUD' and 'SERVICE' tabs. The 'oslo_test_cloud' section is active, showing 'Operator: jotne' and 'Authentication Info: MIIBljANBgkqhkiG9w0BAQEFA'. A table lists four services from 'trueplm_provider' on port 8898, each with a red 'X' in the Actions column. The table has columns for System Name, Port, Authentication Info, Service Definition, Interface, and Actions. At the bottom right, there is a pagination control showing 'Rows per page: 10' and '1-4 of 4'.

System Name	Port	Authentication Info	Service Definition ↑	Interface	Actions
trueplm_provider	8898	MIIBljANBgkqhkiG9w0BAQEFA	trueplm-add-sensor-data-service	HTTPS-SECURE-JSON	✗
trueplm_provider	8898	MIIBljANBgkqhkiG9w0BAQEFA	trueplm-get-sensor-data-service	HTTPS-SECURE-JSON	✗
trueplm_provider	8898	MIIBljANBgkqhkiG9w0BAQEFA	trueplm-sensor-by-sn-service	HTTPS-SECURE-JSON	✗
trueplm_provider	8898	MIIBljANBgkqhkiG9w0BAQEFA	trueplm-sensors-in-project-service	HTTPS-SECURE-JSON	✗

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

Sensor controller The services to access the sensors form EDMtruePLM projects

GET

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

GET

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

GET

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

POST

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

POST

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

Parameters

Name

Description

proj * required

Project name

string
(path)

prop * required

Aggregate property's name

string
(path)

si * required

Sensor's data

(body)

Example Value | Model

```
{
  "SensorData": [
    {
      "SensorMeasurement": [
        {
          "Measurement": "string",
          "value": "string"
        }
      ],
      "timestamp": "string"
    }
  ],
  "SensorType": "string",
  "id": "string"
}
```

Parameter content type

application/json

sn * required

Sensor's serial number

string
(path)

Append data service

Get data service

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

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

Parameters

Name

Description

fromDate * required

Start of the time period

string
(query)

11/04/2020

proj * required

Project name

string
(path)

Bike

prop * required

Aggregate property's name

string
(path)

urn:rdl:Bike:altitude list

sn * required

Sensor's serial number

string
(path)

1180322-02701

toDate * required

End of the time period

string
(query)

12/04/2020

Get sensors info service

Server response

Code	Details
200	<p>Response body</p> <pre>"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 }, }</pre>

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

Parameters

Name	Description
proj * required string (path)	Project name <input type="text" value="Bike"/>

Get sensor info service

GET

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

Parameters

Name

Description

proj * required

Project name

string
(path)

Bike

sn * required

Sensor's serial number

string
(path)

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

The screenshot shows the Jotne PLM software interface. The top navigation bar includes the Jotne logo, a menu icon, a help icon, and a user profile icon. On the right side of the navigation bar, there are icons for a mobile device, a checkmark, a search icon, a settings gear, and the language code 'EN'. Below the navigation bar, the breadcrumb path is 'Bike > Bike root (ver.704) > Bike system >'. The main content area is divided into three tabs: 'BREAKDOWN PROPERTIES', 'DOCUMENT PROPERTIES', and 'PRODUCT PROPERTIES'. The 'BREAKDOWN PROPERTIES' tab is selected and displays a table with the following data:

Num	Name	Value	Type
1	Name	Bike system	T
2	Type	System	T
3	Description	The bike breakdown structure	T
4	Created by	man	T
5	Created date	6/24/2019, 3:24:20 PM	T

Below the table, there is a section titled 'USER DEFINED' with a sub-table that is currently empty, displaying 'No data available'.

Define the structure for your sensor data values

TYPES OF REFERENCE DATA	REFERENCE DATA VALUES
Product element type	Value +
Product properties	point info 🗑️
Breakdown element type	altitude info 🗑️
Breakdown properties	position info 🗑️
Requirement types	
Requirement properties	
Document properties	
Aggregate struct	
Aggregate struct elements	










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

TYPES OF REFERENCE DATA


- Product element type
- Product properties
- Breakdown element type
- Breakdown properties
- Requirement types
- Requirement properties
- Document properties
- Aggregate struct
- Aggregate struct elements**

REFERENCE DATA VALUES

Struct type
altitude info

Name	Inherited by	Type	Values/Expression	Units	RO	
timestamp		Date / Key			<input type="checkbox"/>	  
altitude		Numeric			<input type="checkbox"/>	  
altitude_err		Numeric			<input type="checkbox"/>	  

Define the breakdown element type for the sensor

TYPES OF REFERENCE DATA	REFERENCE DATA VALUES
Product element type	Value +
Product properties	Sensor part +
Breakdown element type	Subsystem +
Breakdown properties	Subtask +
Requirement types	System +
Requirement properties	Task +
Document properties	Unit + 
Aggregate struct	USB_GPS +
Aggregate struct elements	Utilization model questions +

Define the properties for the breakdown element type of the sensor

TYPES OF REFERENCE DATA

- Product element type
- Product properties
- Breakdown element type
- Breakdown properties**
- Requirement types
- Requirement properties
- Document properties
- Aggregate struct
- Aggregate struct elements

REFERENCE DATA VALUES

Node type
USB_GPS

Name	Inherited by	Type	Values/Expression	Units	RO	+ ≡+
position_series		GPSposition List			<input type="checkbox"/>	
altitude_series		GPSaltitude List			<input type="checkbox"/>	
serial number		Text			<input type="checkbox"/>	

Define the structure for your sensor data values

TYPES OF REFERENCE DATA	REFERENCE DATA VALUES
Product element type	Value +
Product properties	point info 🗑️
Breakdown element type	altitude info 🗑️
Breakdown properties	position info 🗑️
Requirement types	
Requirement properties	
Document properties	
Aggregate struct	
Aggregate struct elements	

Direct connection to the EDMtruePLM system with the REST API

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



<https://demo.jotne.com/EDMtruePLM/v2/api-docs>

Explore

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
