

FROST

Fraunhofer Opensource SensorThings

21.07.2023

Hylke van der Schaaf

Why FROST-Server?

Because we needed it

- 2015: The (H2020) OpenIoT Project software... didn't work as hoped
 - For a follow-up project we needed something that did work
 - Right around the time STA v1.0 was being finalised
 - There was no OpenSource server implementation yet
- We started work on a SensorThings Server implementation

Why Open Source?

Because it makes sense

■ Fraunhofer IOSB

- Research Institute
- Financed through Projects
- Renowned as domain experts
- Not in the business of selling software...

■ Open Source (LGPL)

- Using FROST?
→ OK
- FROST as library in a closed-source product?
→ OK
- Changing FROST?
→ Release changes!

History

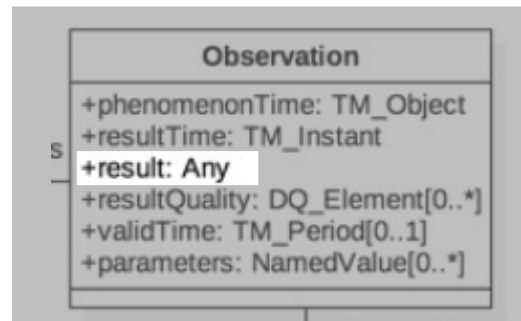
Seven years and counting...

- 2016-02: Start of development
 - Goal: A full implementation of the STA
- 2016-07: published on GitHub
- 2016-11: v1.0 – CRUD, DataArray, MQTT
- 2016-11: MultiDatastream
- 2017-01: JSON filtering
- 2017-09: Docker support
- 2018-01: StringID & UUID Backends
- 2018-02: FROST has a name!
Batch Processing
 - Goal reached!
- 2018-04: Horizontal scalability
- 2018-04: Client-specified IDs
- 2018-08: HELM chart
- 2019-01: Authentication
- 2019-07: Actuation
- 2020-02: CSV result format
- 2020-08: GeoJSON result format
- 2021-03: Deep & Distinct select
- 2022-05: Version 2.0
 - Data Model as Plugin
 - OData Support
 - Direct Data Streaming
- 2023-08: Fine-Grained Authorisation

Data Type Handling

Optimised for precision

- Observation/result has type Any
- Any? Anything that is valid in JSON
 - Number: 1.23e-3
 - String: "cloudy"
 - Object: {"temp": 1.2, "clouds": true}
 - Array: [1.2, 1.3, 0.9]
 - Boolean: true|false
 - No-Value: null



■ FROST:



- Truly type-conserving
- Type-specific ordering
- Type-safe filtering
- Precision-conserving
 - 2.00 stays 2.00

ID Handling

Numbers are so sequential

■ Supported ID types in FROST Server

- Long (default) `{"@iot.id": 12345}`
- UUID `{"@iot.id": "123e4567-e89b-12d3-a456-426655440000"}`
- String `{"@iot.id": "http://example.org/ontology/superThing"}`

■ ID generation methods

- Server defined (default)
- User defined
- Mixed

■ Both configurable per EntityType

Complex Filtering

Searching...

■ Calculating with times and intervals

Observations?

```
$filter=phenomenonTime gt now() sub duration'PT1H'
```

■ Compare observation result to a property of its Datastream

v1.0/Observations?

```
$filter=result gt Datastream/properties/threshold
```

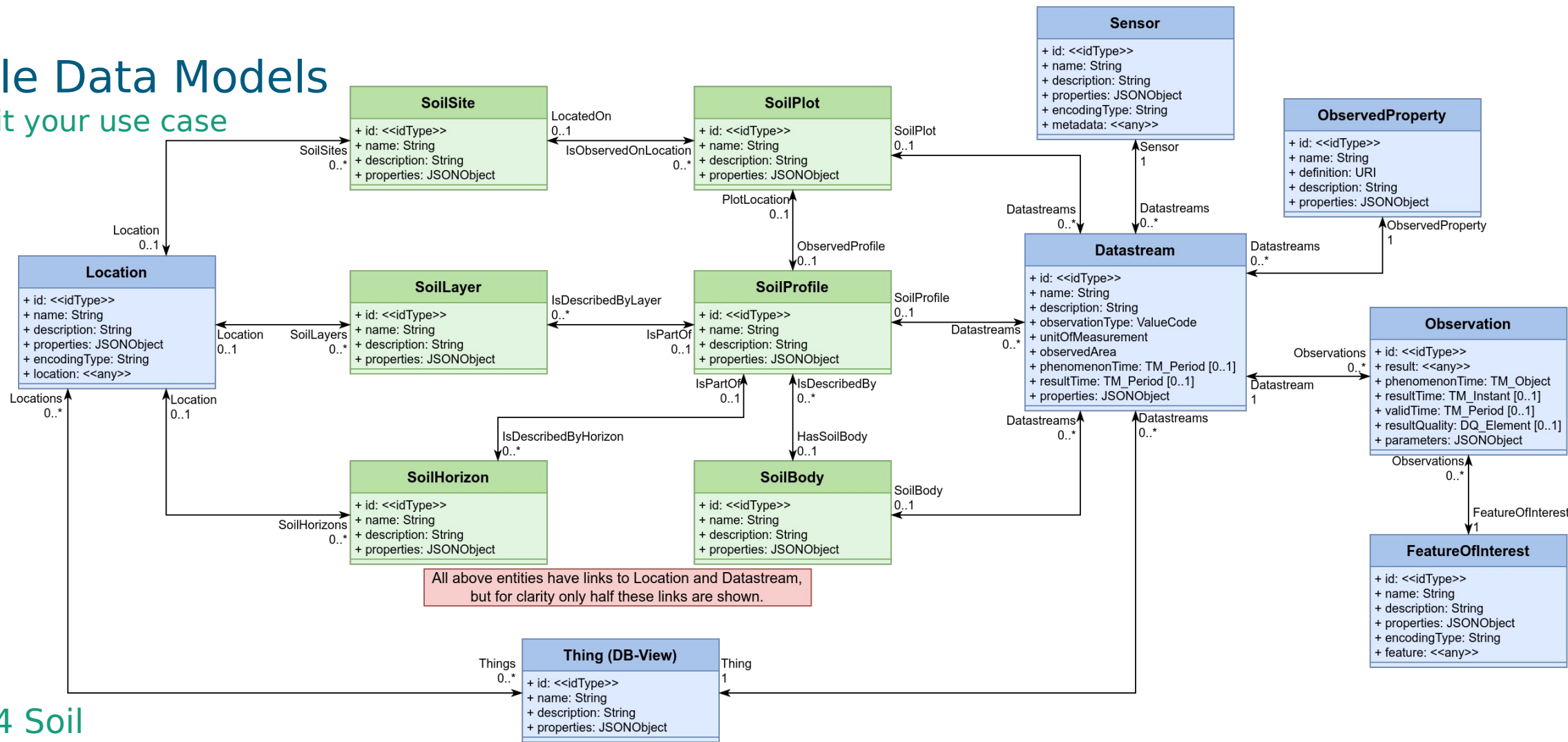
■ Fetch Observations for the last n days where n is specified in the Datastream

v1.0/Observations?

```
$filter=phenomenonTime gt now() sub duration'PT1D' mul  
Datastream/properties/fetchDays
```

Flexible Data Models

Make it fit your use case



■ STA 4 Soil

■ INSPIRE soil Model

■ Specialised Things

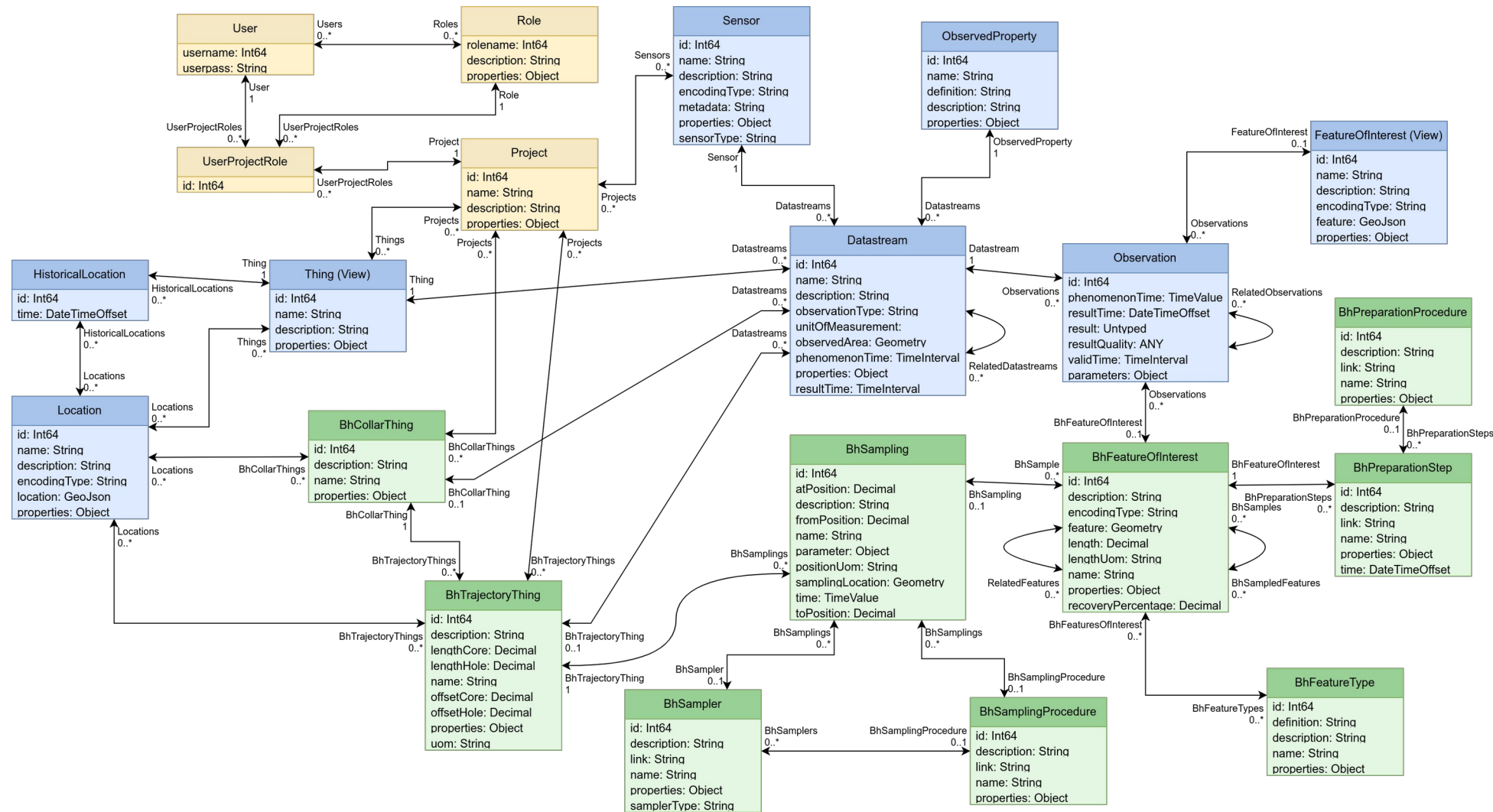
Flexible Data Models

Make it fit your use case

■ STA 4 GeoTech

■ Sampling

■ Linear referencing



Deployment

Every use case is different

- Tomcat

- Docker

- Docker images

- <https://hub.docker.com/r/fraunhoferiosb/frost-server>

- <https://hub.docker.com/r/fraunhoferiosb/frost-server-http>

- <https://hub.docker.com/r/fraunhoferiosb/frost-server-mqtt>

- Docker-compose examples

- <https://github.com/FraunhoferIOSB/FROST-Server>

- HELM charts (for deployment on Kubernetes)

- <https://github.com/FraunhoferIOSB/helm-charts>

Deployment

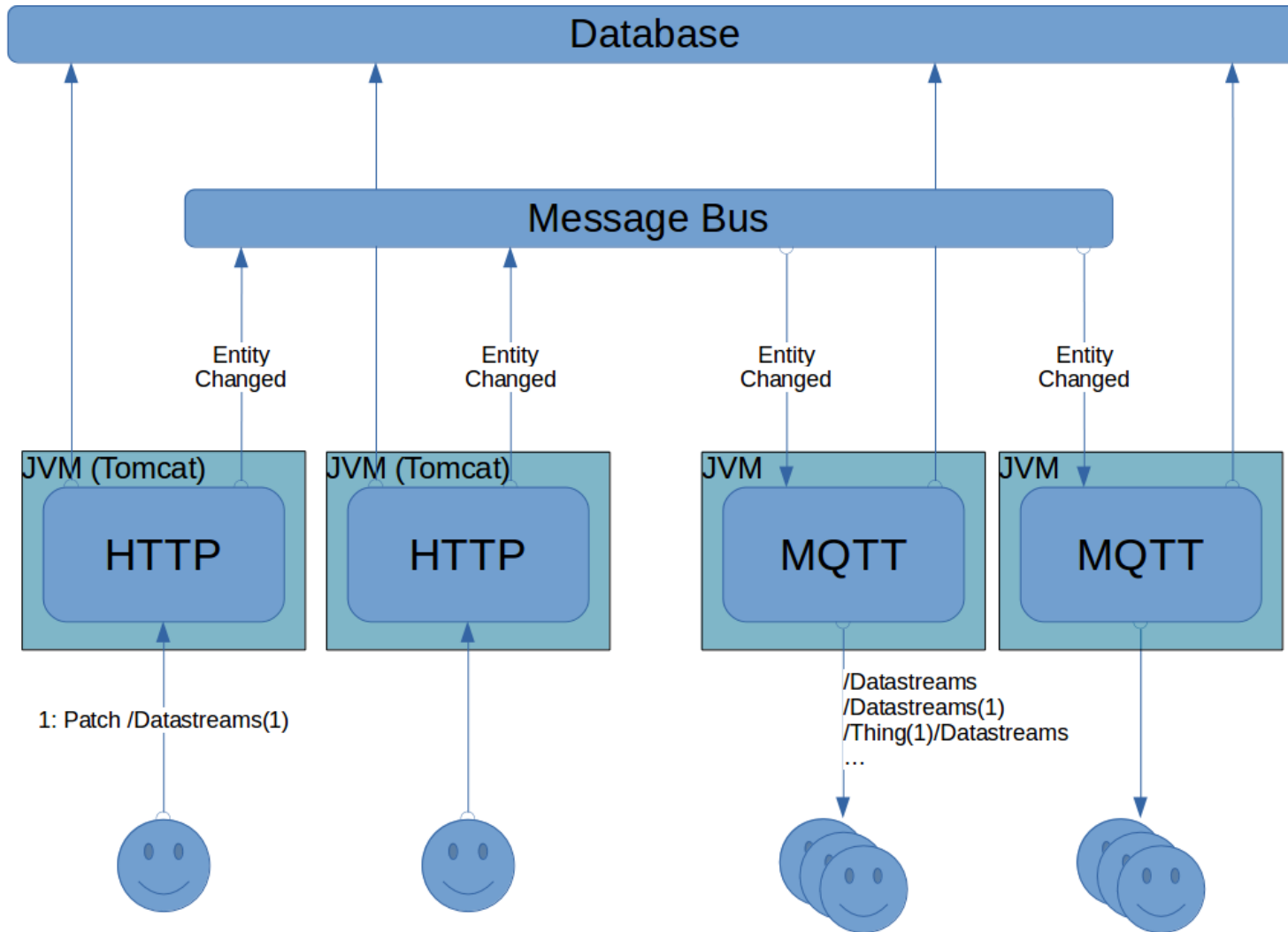
Do it yourself

Install FROST Server in only 3 simple steps NOW!

- > `wget https://github.com/FraunhoferIOSB/FROST-Server/blob/master/docker-compose.yaml`
- > `docker-compose up`
- open `http://localhost:8080/FROST-Server/v1.0`

Scalability

For when your data grows



Performance

A few numbers

■ Hamburg

- High volume:
~2000 – 5000 per second
- Short retention ~1 Week
- Running on an auto-scaling Kubernetes cluster in Azure

■ Air Quality Demo

- Low volume: 12000 per hour
- Long retention → ~700M Observations
- Running on one node in our test Kubernetes cluster (On a SSD disk)

Air Quality

Outdoor, but indoor also works

- All European Countries
- Harvested from EEA CSV files
- ~700M Observations after 2018

AirQuality

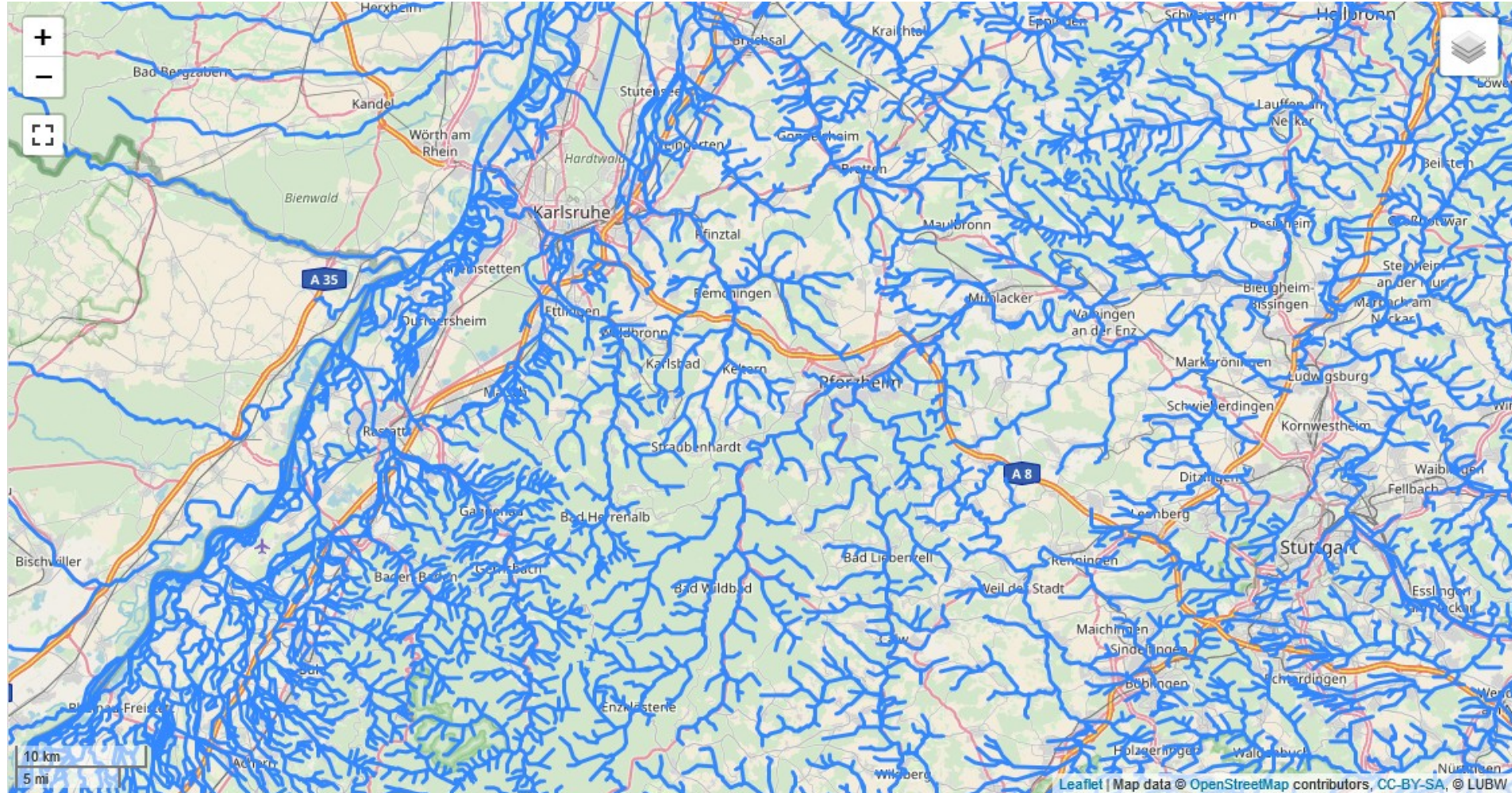


<https://api4inspire.k8s.ilt-dmz.iosb.fraunhofer.de/servlet/is/127/>

Rivers in Baden-Württemberg

Locations can also be lines or polygons

- ~20000 Rivers and streams

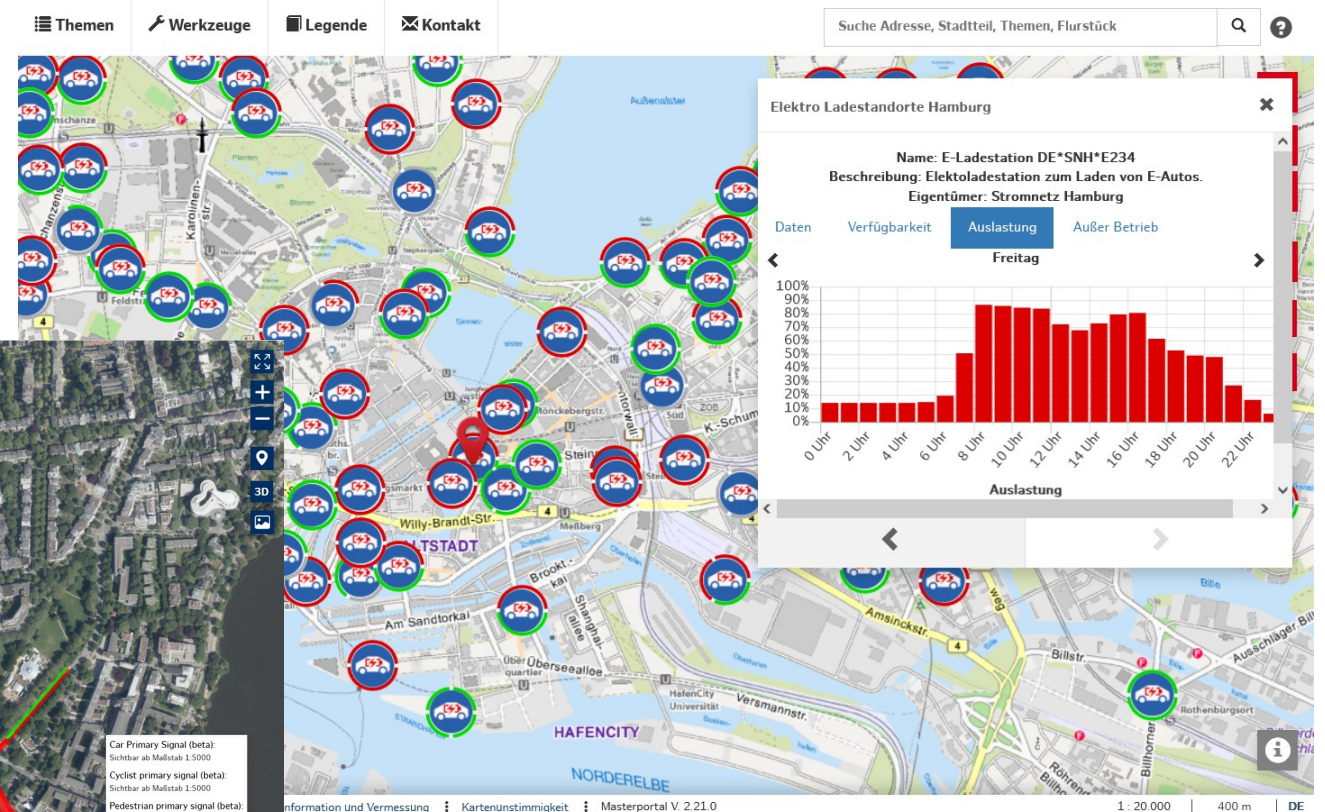
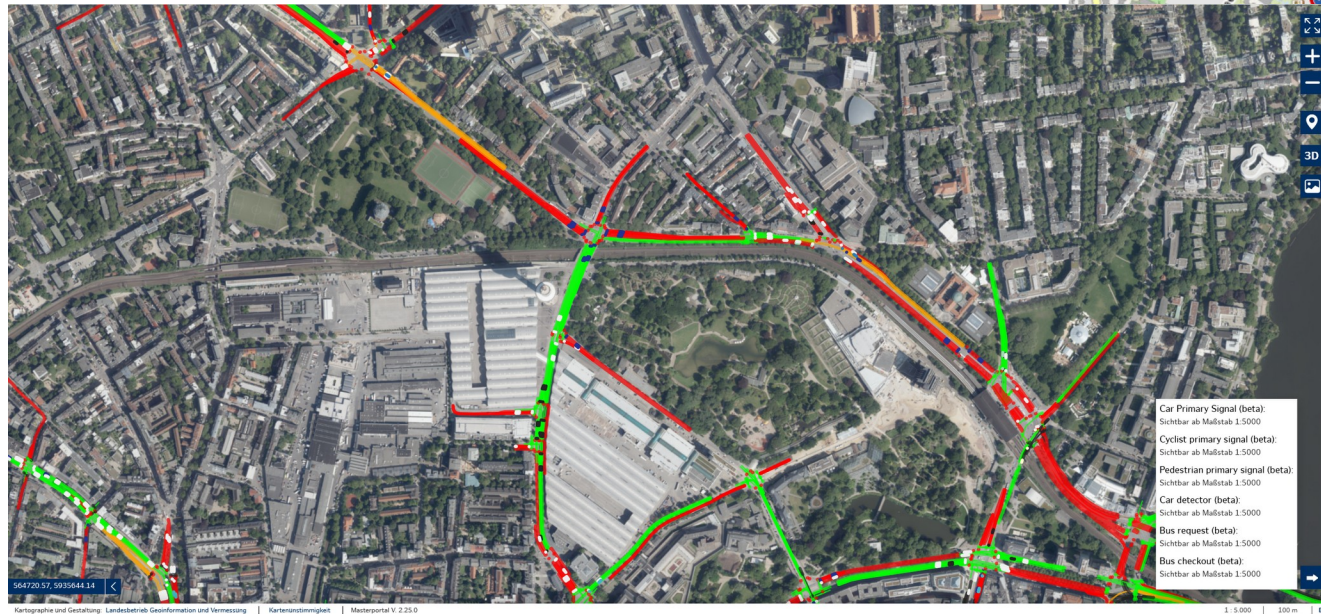


<https://api4inspire.k8s.ilt-dmz.iosb.fraunhofer.de/servlet/is/107/>

Urban Data Plattform Hamburg

Our first large user

- Car charging stations
- Bike lending stations
- Live Traffic Light Status



STA plus

A new Data Model Extension

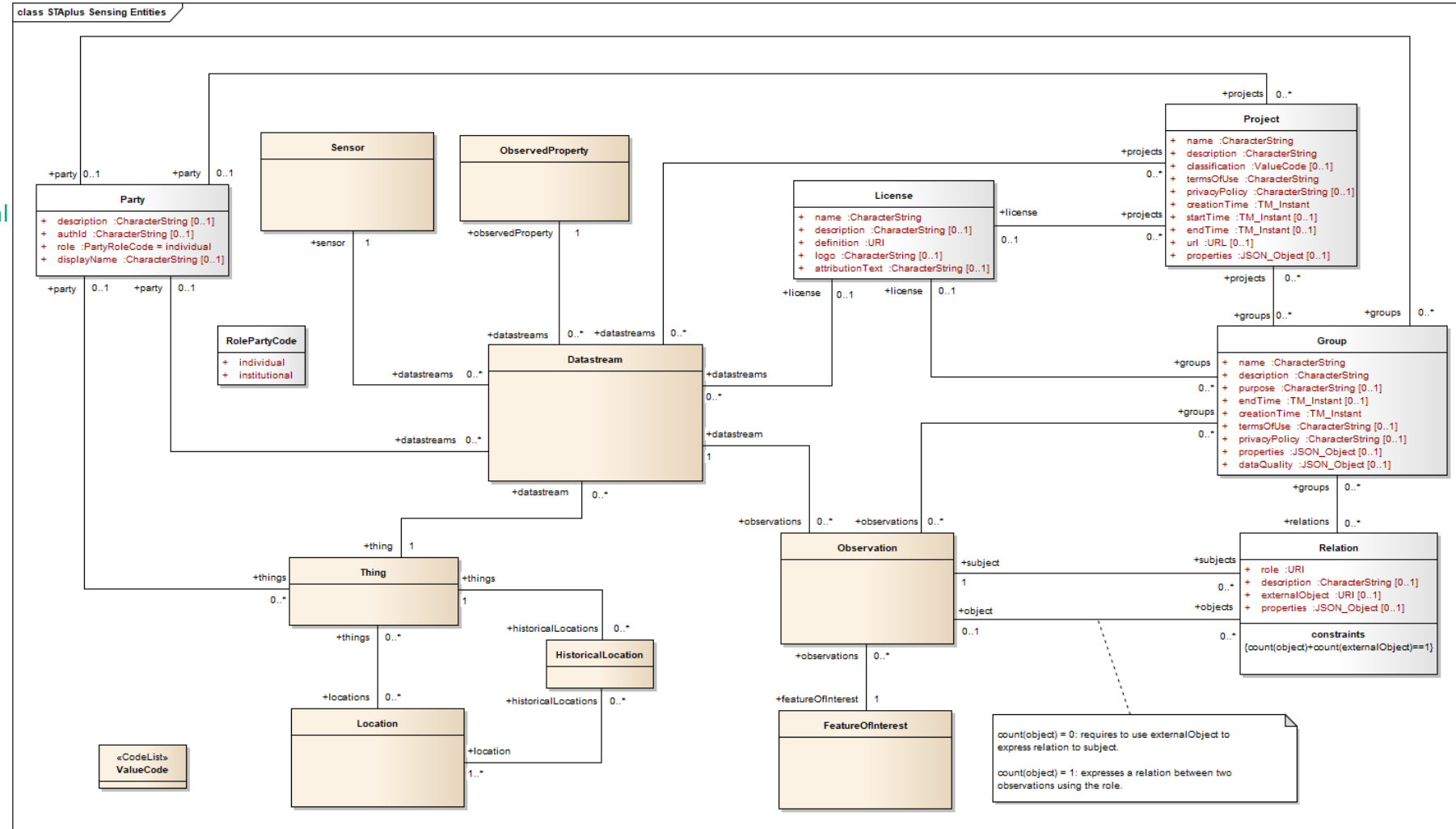
- STA For Citizen Science

- Currently under vote

<https://docs.ogc.org/DRAFTS/22-022.html>

- Adds

- Project
- License
- Group
- Relation
- Party



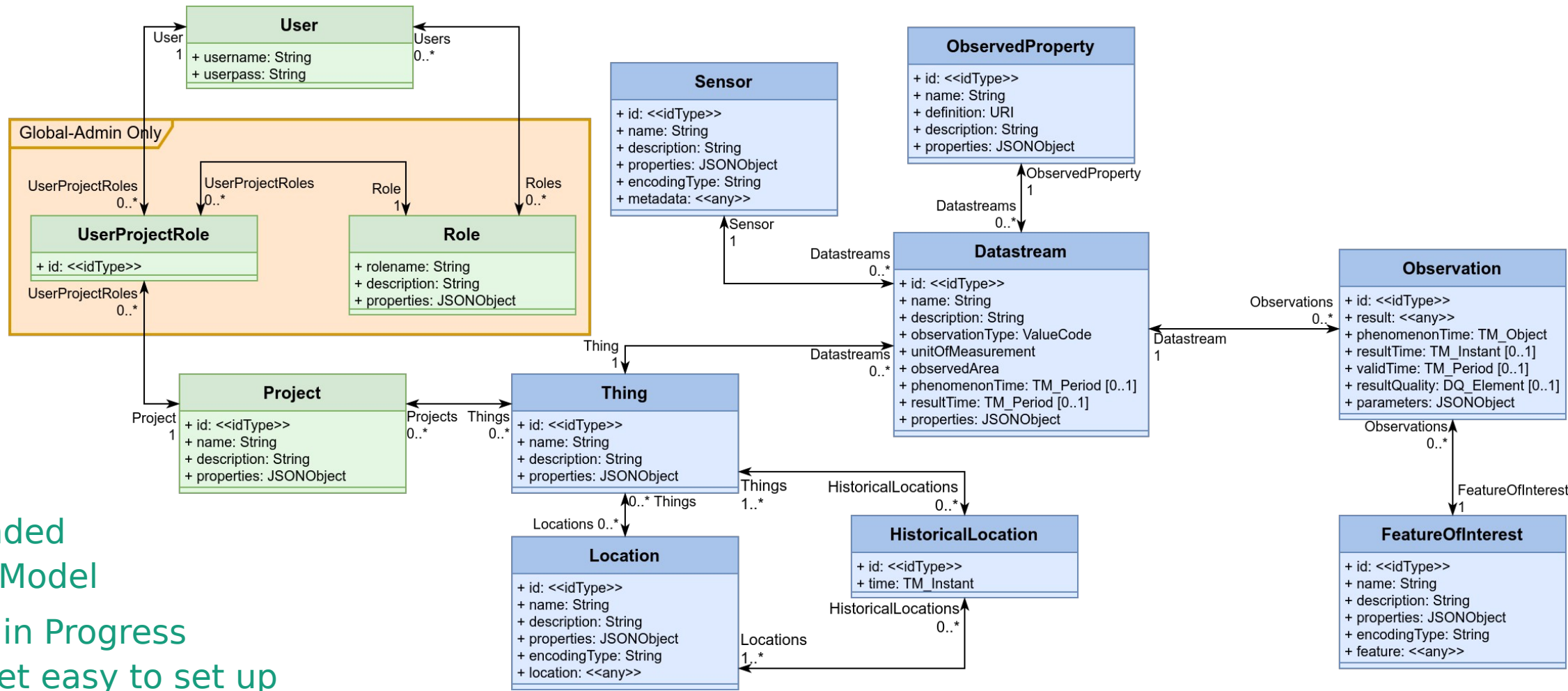
Fine-Grained security

A very fresh feature

- User A can only insert Observations in certain Datastreams
 - User B can edit entities linked to a certain Project
 - normal users can only read non-restricted data
- Can a User
 - Create Entities of EntityType-X (new Observations)
 - Link new Entity-X to Entity-Y (new Observation in DS-1)
 - Update properties of Entity-X (Patch/Put Observation-1)
 - Change a link of Entity-X from Entity-Y1 to Entity-Y2 (Move Observation-1 from DS-1 to DS-2)
 - Delete Entity-X

A Fine Grained Security Prototype

Your use case is probably different!



- Extended Data Model
- Work in Progress
- Not yet easy to set up

What do we Offer

The Source is Open

- Partner in European projects
- On-Line & On-Site
 - Introduction to SensorThings API
 - Data Modelling your use case
 - Setting up FROST
- Technical Support
- Custom extensions

Contact

Dr. Hylke van der Schaaf
Information Management and Production Control
hylke.vanderschaaf@iosb.fraunhofer.de

Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB
Fraunhoferstraße 1
76131 Karlsruhe, GERMANY
www.iosb.fraunhofer.de

