



ESTABLISH THE CONNECTION OF CITIZEN OBSERVATORIES RESOURCES WITH CENTRAL CATALOGUE

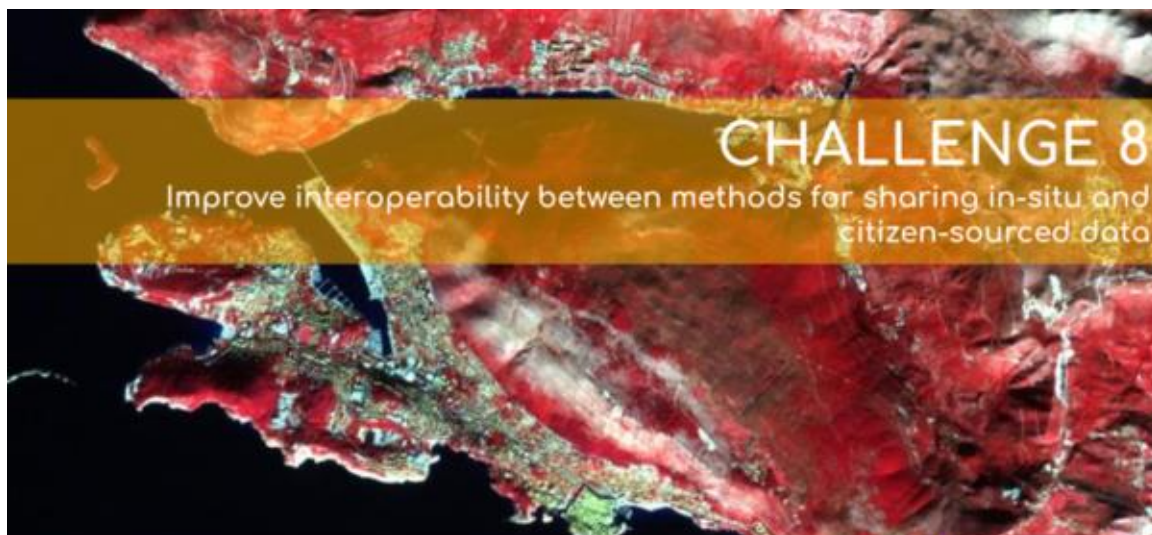
Dubrovnik INSPIRE Hackathon 2020 – Challenge Final Report

March 30 – June 11

Mentors/Team leaders

Michal Kepka, University of West Bohemia

Valantis Tsiakos, Institute of Communication and Computer Systems



Earth Observation + Citizen Science =
Empowered Society



Overview & process

The goal of the challenge is to make available datasets provided by H2020 Citizen Observatories as well as other citizen-science projects and initiatives, through the use of OGC [SensorThings API](#) standard and develop and test tools to provide combined visualization and mapping of data coming from different sources. This involves also sharing of environmental measurements coming from different IoT devices and in-situ monitoring sensor networks, aiming to establish combined use of data and services among different platforms towards improved environmental monitoring.

The overall outcome of this challenge involves the development of an alignment methodology (inspired by the INSPIRE harmonisation process) and the implementation of data translators and connectors that will facilitate the conversion of resources to OGC SensorThings API compatible schemas.

The main activities realised in the context of this challenge are as follows:

- ❖ Analysis of the OGC SensorThings API and exploitation of data provided from [SCENT Citizen Observatory](#) by third-party application.
- ❖ Establishment of a methodology towards the utilisation of a standardised approach for sharing crowd-sourced and citizen-science data. This involves the compilation of a template to allow the mapping between the SensorThings API and other schemas.
- ❖ Implementation of a translation process aiming to facilitate combined use of heterogeneous data sources.
- ❖ Integration of SCENT Citizen Observatory data with existing environmental monitoring platforms (SensLog).

The [SCENT Harmonisation platform](#) was utilised as reference implementation in the context of this challenge. The API for the SensorThings API standard was implemented as a standalone Tomcat application using the Jersey RESTful Web Services framework. The implementation supports all the requests described in the standard as well as all the filtering capabilities.

The root URI of the service that was made available for the challenge is the following: <https://mariaisawesome.iccs.gr:8443/SensorThing/v1.0>

Each entity has a unique identifier and one-to-many properties. Also, in the case of an entity holding a relationship with entities in other entity sets, this type of relationship is expressed with navigation properties. Therefore, in order to perform CRUD actions on the resources, the first step is to address to the target resources through URI. There are three major URI components used here, namely (1) the service root URI, (2) the resource path, and (3) the query options.

The root URI of the service provides as a response a JSON object with a property named value. The value of this property is a JSON Array containing one element for each entity set of the SensorThings Service. Each element is a JSON object with at least two name/value pairs, one with name containing the name of the entity set (e.g., Things, Locations, Datastreams, Observations, ObservedProperties and Sensors) and one with name url containing the URL of the entity set. The following image depicts the returned JSON.

```

value:
  0:
    name: "Things"
    url: "https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Things"
  1:
    name: "Locations"
    url: "https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Locations"
  2:
    name: "HistoricalLocations"
    url: "https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/HistoricalLocations"
  3:
    name: "Datastreams"
    url: "https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Datastreams"
  4:
    name: "Sensors"
    url: "https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Sensors"
  5:
    name: "Observations"
    url: "https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Observations"
  6:
    name: "ObservedProperties"
    url: "https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/ObservedProperties"
  7:
    name: "FeaturesOfInterest"
    url: "https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/FeaturesOfInterest"

```

FIGURE 1: SCREENSHOT OF THE JSON RETURNED FROM THE SERVICE ROOT URI

Results

Exploitation of resources from SCENT Harmonisation platform

Interaction with the SCENT Harmonisation platform involved the creation of various GET requests, enabling the exploitation of the existing citizen-science resources while also harnessing the API capabilities. Such examples that showcase the functionalities of the API and were utilised in the context of this activity are presented in the following table.

<u>Functionality</u>
<ul style="list-style-type: none"> ❖ Select an observed property (i.e. water velocity) and get all the Datastreams given the selected observed phenomenon ❖ Then take all the observations under a Datastream
<u>API call</u>
https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/ObservedProperties(4)/
https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/ObservedProperties(4)/Datastreams
https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/ObservedProperties(4)/Datastreams(402)/Observations?\$filter=phenomenonTime%20ge%202018-11-

[15T00:00:00.000Z%20and%20phenomenonTime%20le%202018-11-17T00:00:00.000Z&\\$count=true](https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/ObservedProperties(4)/Datastreams?$expand=Observations($expand=FeatureOfInterest)$filter=phenomenonTime%20ge%202018-11-17T00:00:00.000Z&$count=true)

Functionality

- ❖ Select all the observations for an observed phenomenon for a citizen science campaign

API call

[https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/ObservedProperties\(4\)/Datastreams?\\$expand=Observations\(\\$expand=FeatureOfInterest\)\\$filter=phenomenonTime%20ge%202018-11-15T00:00:00.000Z%20and%20phenomenonTime%20le%202018-11-17T00:00:00.000Z&\\$count=true&\\$count=true](https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/ObservedProperties(4)/Datastreams?$expand=Observations($expand=FeatureOfInterest)$filter=phenomenonTime%20ge%202018-11-15T00:00:00.000Z%20and%20phenomenonTime%20le%202018-11-17T00:00:00.000Z&$count=true&$count=true)

Functionality

- ❖ Get all available Things (or else Data Providers)
- ❖ Get all historical locations (both time and coordinates) for the Thing (Data Provider in the case of SCENT) that has been chosen in the previous step

API call

<https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Things>

[https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Things\(268\)/HistoricalLocations?\\$expand=Locations](https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Things(268)/HistoricalLocations?$expand=Locations)

Functionality

- ❖ Get all Datastreams for the Thing that has been chosen previously. Sensor and Observed Property entities info are also being fetched in order to populate the second dropdown list

API call

[https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Things\(268\)/Datastreams?\\$expand=Sensor,ObservedProperty](https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Things(268)/Datastreams?$expand=Sensor,ObservedProperty)

Functionality

- ❖ Get all observations along with their coordinates (through the FeatureOfInterest entity) that are associated with the given Datastream

API call

[https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Datastreams\(438\)/Observations?\\$expand=FeatureOfInterest](https://mariaisawsome.iccs.gr:8443/SensorThing/v1.0/Datastreams(438)/Observations?$expand=FeatureOfInterest)

Functionality

❖ Get all the values from an in-situ sensor that exceed a certain value
API call
https://mariaisawesome.iccs.gr:8443/SensorThing/v1.0/Datastreams(3)/Observations?\$filter=phenomenonTime%20ge%202018-11-15T00:00:00.000Z%20and%20phenomenonTime%20le%202018-11-17T00:00:00.000Z%20and%20result%20gt%200.36&\$count=true
Functionality
❖ Get all the values from both the in-situ and citizen scientists that exceed a certain value during campaign
API call
https://mariaisawesome.iccs.gr:8443/SensorThing/v1.0/ObservedProperties(1)/Datastreams?\$expand=Observations(\$filter=phenomenonTime%20ge%202018-11-15T00:00:00.000Z%20and%20phenomenonTime%20le%202018-11-17T00:00:00.000Z%20and%20result%20gt%200.36)&\$top=5&\$count=true

Mapping between OGC SensorThings API & SensLog data model

Core of the SensLog data model is based on the ISO Observations & Measurements standard thus mapping between OGC SensorThingsAPI standard and SensLog data model was feasible during this hackathon. Extensions of the SensLog data model in the field of sensor-unit hierarchy, user hierarchy and alerts that are out of the ISO O&M standard were not mapped on the OGC SensorThingsAPI data model.

The main idea of the mapping procedure is to match the most of attributes between models without additional operations. The simplified diagram of mapping entities between both models is shown in Figure 2. In the most cases each entity in one model can be mapped on corresponding entity in the another model. Only one exception of compiling attributes from 2 entities was the entity Datastream which covers collection of same ObservedProperty and Sensor entities. This Datastream entity contains attributes that are covered by 2 entities in SensLog model - Phenomenon and UnitToSensor. The opposite situation is when entity from one model doesn't have corresponding entity in the another model. This case is FeatureOfInterest from SensorThingsAPI model that doesn't have counterpart in the SensLog data model.

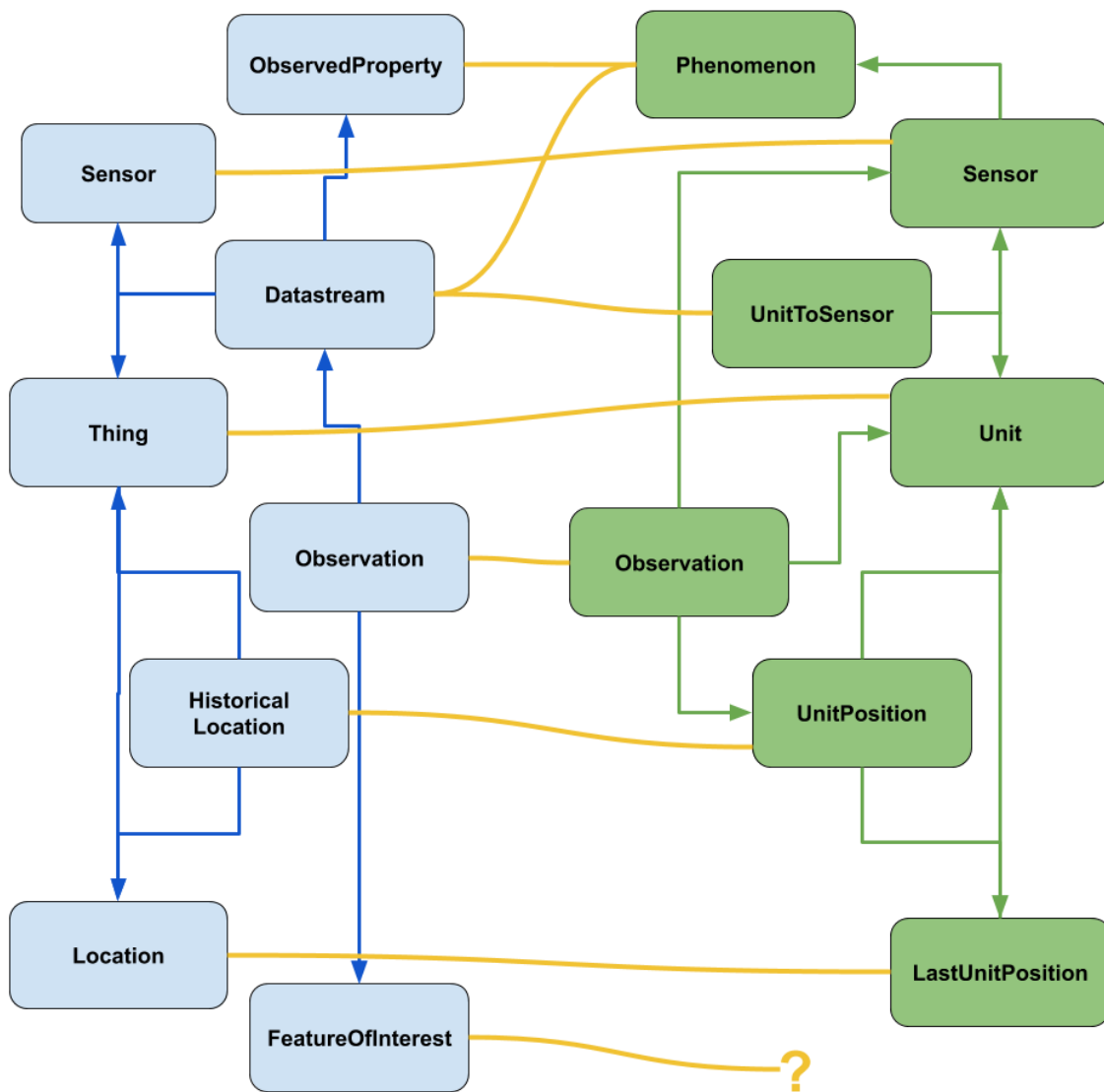


FIGURE 2: DIAGRAM OF ENTITIES MAPPING BETWEEN SENSLOG MODEL AND SENSORTHINGSAPI

A template with mapping entities between both data models was described as the result of this subtask.

Ingestion of data from SCENT Citizen Observatory to SensLog platform

To increase interoperability of SensLog, system of Connectors was designed in previous development. SensLog Connector is a component translating source original REST API to the SensLog API and vice versa. In defined task a version of SensLog Connector was designed where public interface is implementing services only to get data from the OGC SensorThings API interface of SCENT platform. Scheme of components for this data ingestion is shown in Figure 4.

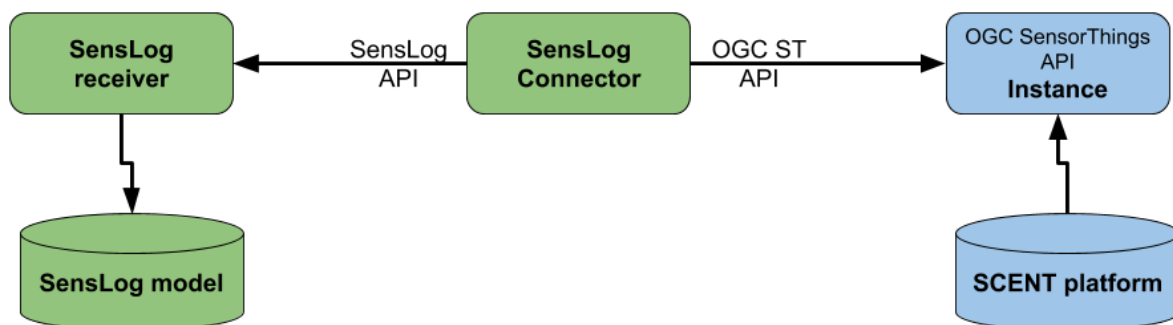


FIGURE 4: SCHEME OF COMPONENTS TO TRANSFER DATA FROM SCENT TO SENSLOG

Testing example data mapped from SensLog entities to OGC SensorThingsAPI entities are described in the [Spreadsheet](#).

Impact

Cross sectoral and boundary interoperability

The SensorThings API was chosen for the modeling of IoT enabled, crowd-sourced and in-situ measurements collected for two main reasons. Firstly, it allows the efficient modeling of ‘moving sensors’. Portable sensors (and IoT devices) are used to collect measurements from different areas and are equipped by multiple users. Also, volunteers may use multiple sensors within a campaign to collect measurements. The flexibility of the SensorThings API, that models independently the sensor, from the volunteer and from the location that the measurement was collected at, is ideal for these types of data and use cases. What is more, the SensorThings API can support heterogeneous measurements collected from the same volunteer using the same sensor. This is very important as in the case of a Citizen Observatory like SCENT, a volunteer is using a smartphone to collect information about land cover/land use, water level and water velocity and the same portable sensor to collect air temperature and soil moisture.

In addition, the SensorThings API standard has some key advantages over similar standards, such as the OGC Sensor Observation Service (SOS). To begin with, the data format exchange is based on the JSON encoding which is lightweight and easy to create and exchange over the network minimizing the overhead that the XML schema adds to the data objects. Also, the SensorThings API is based on a Resource Oriented Architecture, offering standards and guidelines about the data models and the entities that structure the information. In addition, the standard is based on the REST Web services, meaning that it is stateless and clearly separates the concerns of client and server. Furthermore, the standard supports HTTP POST,

DELETE, PUT and PATCH allowing the creation, update and deletion of entities as needed. Last but not least, it allows the retrieval of the information using pagination, ensuring scalability and short response times.

Sustainable Development Goals (SDGs)

Citizen science could make contributions in different types of processes linked to the SDGs (i.e. monitoring progress and implementing action). For instance, tracking progress towards the SDGs requires high-quality, timely and accessible data, often in areas or for sectors where little or no data exists. Traditional means to gather such data (i.e. via household surveys) but these are costly, and they are not collected frequently enough to support policy making. In addition, gaps exist in the required data. Citizen science projects can contribute to fill these gaps, or to disaggregate national-level data in order to provide clearer pictures of geographic, gender, socio-cultural or other inequalities while also complementing traditional household surveys via additional heterogeneous data (river and coastal management information, etc), that are important for the implementation of SDGs. In this context, the outputs of this challenge are further contributing to the integration and utilisation of citizen-science data towards monitoring and implementing SDGs. The interoperability considerations that were addressed, aim to support and streamline the uptake and combination of citizen-science data with existing information systems and legacy data sources and subsequently lead to improved monitoring of relevant indicators.

Contacts

In case you want to improve exploitation and use of your citizen-science resources following the data sharing principles employed in this challenge, feel free to contact: info@weobserve.eu

- END OF DOCUMENT -



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 776740.