OGC Citizen Science engeneering report

Publication Date: YYYY-MM-DD

Approval Date: YYYY-MM-DD

Submission Date: YYYY-MM-DD

Reference number of this document: OGC YY-nnn

Reference URL for this document: http://www.opengis.net/doc/PER/t14-D012

Category: Public Engineering Report

Editor: Joan Masó

Title: OGC Citizen Science engeneering report

OGC Engineering Report

COPYRIGHT

WARNING

LICENSE AGREEMENT

complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Table of Contents

Chapter 1. Summary

This Engineering report describes the first phase of the citizen science Interoperability experiment organized by the EU H2020 WeObserve project under the OGC innovation program and supported by the EU H2020 NEXTGEOSS project. The activity covered aspects of data sharing architectures for citizen science data, data quality, data definitions and user authentication.

The final aim was to propose solutions on how Citizen Science data could be integrated in the Global Earth Observation System of Systems (GEOSS). The solution is necessarily a combination of technical and networking components, being the first ones the focus of this work. The applications of international geospatial standards in current citizen science and citizen observatory projects to improve interoperability and foster innovation is one of the main tasks in during the experiment to achieve the final aim.

The main results of the activity was to demonstrate that Sensor Observing Services can be used for citizen science data (as proposed in the OGC SWE4CS discussion paper) by implementing it in several clients and server that has been combined to show Citizen Science observations. In addition an authentication server was used to create a federation between three projects. This federated approach is part of the proposed solution for GEOSS that can be found in the last chapter. Many open issues has been identified and are expected to be addressed in the second phase of the experiment, including the use of a definitions server.

1.1. Requirements & Research Motivation

This experiment was designed to demonstrate how current ICT-based tools can be applied together to allow better citizen participation in CS projects and enable better reuse of the data gathered. Citizen Science is highly transdisciplinary and heterogeneous by nature, and current standardization efforts already occur in the OGC (e.g. addressing data model and sharing issues) as well as outside the OGC (primarily addressing project descriptions and data set metadata). Citizen Science projects might benefit from concrete examples and best practices required to achieve the full benefits of interoperability. OGC is in the ideal position to develop and provide such best practice guidance to the international community. Developed solutions in this IE should be applicable to most Citizen Science projects. Findings from this IE will be generalized in this manner and might set the basis for additional experimentation in the future.

The FP7 Citizen Observatory Web (COBWEB) project was the first to propose the use of SWE in CS. This work resulted in an OGC public discussion paper available on the OGC website (OGC 16-129). The discussion paper describes a data model for the standardized exchange of citizen science sampling data based on SWE standards. This discussion paper was the initial motivation for this IE.

On the other hand, the Citizen Science Association’s International Working Group on Citizen Science Data and Metadata that developed the (PPSR-Core), the European Citizen Science Association (ECSA) has a working group that recognizes the value of standardization in the CS activities (supported by a COST Action). However, these activities could benefit from some experimentation that would be able to suggest common best practices while recognizing the particularities and current approaches in different thematic domains, such as biodiversity monitoring. Citizen Science can complement authoritative in-situ observations and fill the information gaps in numerous scientific disciplines that could be essential for informed decision making. In that sense, the way Citizen Science can be integrated into The Global Earth Observation System of Systems (GEOSS) (including GEOSS-Data Core as the pool to promote and share open and free data) is still under investigation.

The 'Ecosystem of Citizen Observatories (CO) for Environmental Monitoring' — WeObserve project is a Horizon 2020 funded project focused on improving the coordination between existing COs and related regional, European and international activities. WeObserve tackles three key challenges that face COs: awareness, acceptability and sustainability. The CoP3 is about Interoperability of Citizen Science projects. WeObserve project – via its CoP activities – has represented an opportunity to promote interoperability experiment in collaboration with the OGC. Such collaboration would address questions raised in the SWE4CS discussion. In addition, it offers the possibility to directly feed the results into the OGC relevant standards and promotes their usage within GEOSS (as an important user community of OGC standards).

In anticipation of the 50th Anniversary of Earth Day in 2020, Earth Day Network, the Woodrow Wilson International Center for Scholars, and the U.S. Department of State, through the Eco-Capitals Forum, announce Earth Challenge 2020, a Citizen Science Initiative. This initiative is in collaboration with Connect4Climate – World Bank Group, Conservation X Labs, Hult Prize, National Council for Science and the Environment (NCSE), Open Geospatial Consortium (OGC), Reset, SciStarter, UN Environment and others to be announced. Earth Challenge 2020 will help engage millions of global citizens in collecting one billion data points in areas including air quality, water quality, biodiversity, pollution and human health. Earth Change 2020 data will be shared through the GEOSS Portal.

A precise descriptions of the requirements that have been addressed by the work documented in this Engineering Report; together with the research motivation that answers the fundamental question: What motivated us to address this topic in this report?

1.2. Prior-After Comparison

This is the first Citizen Science interoperability experiment conducted by the OGC. Prior to this activity there was a proposed discussion paper on how to apply the SWE standards in Citizen Science (SWE4CS). This experiment positively tested the proposed route using Sensor Observing Services but also has open the door to future exploration fo SensorThings API. Prior this activity there was a H2020 project with a Authentication Service and after the activity three H2020 formed a bigger federation demonstrating the route to federating and aggregating Citizen Science projects to contribute to GEO objectives. This work is relevant to the OGC Citizen Science Domain Working Group.

1.3. Recommendations for Future Work

This OGC Interoperability Experiment ended on June 2019, but a second IE is foreseen for next year.

New possible topics for next IE to be discussed among the members include the following:

- There is a need for clarifying how to coordinate infrastructures for citizen science in Europe, and adopt standard procedures for data sharing and single sign on. Solving this issue will help in connecting CitSci to GEO. Steffen Fritz (IIASA) has proposed a side event in the next EuroGEOSS workshop to discuss this with the relevant players. This is emerging as a new activity in the WeObserve Interoperability Community of Practice that is related but not directly connected to the IEs.
- The WMO (World Meteorological Organization) is concerned about the amount of different CitSci activities that are being organized by meteorological organizations. It is looking for ways of taking advantage of this new data stream but problems of standardization of what is measured and how data is being shared arise. WMO has detected the potential of these data streams and would like to harmonize the situations to make data more useful for weather predictions in the future.
- OGC is promoting a new generation of web services based on OpenAPI. It is unclear how this could impact the use of OGC standards by CitSci projects but it is seen as an opportunity to make OGC standards more usable and compatible with IT mainstream. A hackathon to develop OGC API specifications is organized on June 20-21 in London.

The definition of the follow up IE will start in June when the first one will finalize and the ER is produced.

This section should answer the question: What does this ER mean for the Working Group and OGC in general? What aspects shall be addressed next? In any specific order? What actions are necessary?

This is a write up for why this ER should be important to the working group and OGC. This paragraph provides recommendations on how to further proceed with the achievements documented in this ER.

1.4. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Contacts

Name	Organization
Joan Maso	UAB-CREAF
Andy Cobley	University of Dundee
Valantis Tsiakos	Institute of Communication & Computer Systems (ICCS)
Simon Jirka	52 North
Sven Shade	Joint Research Center

Initiators

Organization
Universtat Autònoma de Barcelona - CREAF (UAB-CREAF)
International Institute for Applied Systems Analysis (IIASA)
Joint Research Center (JRC)
European Space Agency (ESA)
Woodrow Wilson International Center for Scholars (Wilson Center)

The WeObserve project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under grant agreement No. 776740. This presentation reflects only the author’s views and the EU Agency is not responsible for any use that may be made of the information it contains.

1.5. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 2. References

The following normative documents are referenced in this document.

- OGC 10-004r3, Topic 20: Observations and Measurements [http://portal.opengeospatial.org/files/?artifact_id=41579]
- OGC 10-025r1, Observations and Measurements - XML Implementation [http://portal.opengeospatial.org/files/?artifact_id=41510]
- OGC 15-100r1, OGC Observations and Measurements – JSON implementation [https://portal.opengeospatial.org/files/64910]
- OGC 12-006, OGC® Sensor Observation Service Interface Standard v2.0 [https://portal.opengeospatial.org/files/?artifact_id=47599]
- OGC 08-094r1, OGC® SWE Common Data Model Encoding Standard v2.0 [http://portal.opengeospatial.org/files/?artifact_id=41157]
- 15-042r5 OGC TimeseriesML – XML Encoding of the Timeseries Profile of Observations and Measurements v1.2 [http://docs.opengeospatial.org/is/15-042r5/15-042r5.html]

Even if the following is an OGC discussion paper that is not an OGC standard and cannot be considered strictly a normative reference, it is actually the bases for several sections of this document and should be considered as and important background:

- OGC 16-129, Standardized Information Models to Optimize Exchange, Reusability and Comparability of Citizen Science Data (SWE4CS) [https://portal.opengeospatial.org/files/?artifact_id=70328]

Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard OGC 06-121r9 [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

- Citizen Observatory (CO)

```
COs are community-based environmental monitoring and information systems,
that invite individuals to share observations, typically via mobile phone
or the web (from: https://www.weobserve.eu/about/citizen-observatories)
```

- Citizen Science The collection and analysis of data relating to the natural world by members of the general public, typically as part of a collaborative project with professional scientists (from: https://www.uen.org/crowdandcloud/citizen.shtml)
- Citizen Science Association

```
A network that seeks to promote and advance citizen science in a region
or around the world. Examples are the American Citizen Science
Association (CSA), The European Citizen Science Association (ECSA) or
even the Citizen Science Global Partnership (CSGP)
```

- Citizen Science Federation

```
A network of Citizen Science that aims to aggregate innovative EO
technologies, mobile devices, community-based environmental monitoring,
data collection, interpretation and information delivery systems to
empower communities to monitor and report on their environment. An
example of this is the The LandSense Federation
```

- Community of Practice (CoP)

```
Community which work to consolidate practice-based knowledge of COs
sharing information and resources as well as developing guidelines and
toolkits for COs (from: https://www.weobserve.eu/cops/)
```

3.1. Abbreviated terms

NOTE: The abbreviated terms clause gives a list of the abbreviated terms and the symbols necessary for understanding this document. All symbols should be listed in alphabetical order. Some more frequently used abbreviated terms are provided below as examples.

- CitSciIE Citizen Science Interoperability Experiment
- CO Citizen Observatory
- CoP Community of Practice
- COST European Cooperation in Science and Technology
- CS Citizen Science
- CS DWG Citizen Science Domain Working Group
- CSGP Citizen Science Global Partnership
- ECSA European Association of Citizen Science
- EO Earth Observation
- ICT Information and Communication Technologies
- IE Interoperability Experiment
- O&M Observation and Measurements
- PPSR Public Participation in Scientific Research
- SOS Sensor Observation Service
- SSO Single Sign On
- SWE Sensor Web Enablement
- SWE4CS Sensor Web Enablement for Citizen Science
- TC Technical Committee
- TIE Technology Integration Experiments
- WPS Web Processing Service

Chapter 4. Overview

This Engineering Report focuses on the findings of the first phase of the Citizen Science Interoperability Experiment (CitSciIE).

The primary focus of the OGC CitSciIE experiment was to demonstrate the interoperability of Citizen Observatories and Citizen Science (CS) projects and the way OGC standards can be applied to Citizen Science, including possible relationships to other relevant standards from the community. In particular, a subset the originally proposed topics were being addressed based on the participant organizations:

- The use of OGC standards (e.g. Sensor Web Enablement for Citizen Science SWE4CS) to support data integration among CS projects, and with other sources, esp. authoritative data;
- The use of ISO standards, and OGC publications, and community resources to document data quality aspects (e.g. UncertML, QualityML);
- The integration of CS projects/campaigns in a Single Sign-On system (SSO) federation;
- The relationships between OGC standards and data and metadata standards currently used by Citizen Science projects.

The desired outcome of this experiment was to:

1. Successfully demonstrate how OGC standards (e.g. SWE) are applicable to Citizen Science, document available supporting tools, identify the challenges of using OGC SWE standards (or Internet of Things equivalent solutions) within current Citizen Science projects, and propose a way forward. Make recommendations to the Earth Science 2020 initiative on which OGC standards should be utilized to underpin interoperable data collection and sharing.
2. Successfully demonstrate how to estimate Citizen Science data quality and make the quality indicators and conformity available in the document and in supporting tools, and link them to the OGC SWE standards (or Internet of Things equivalent solutions) within current CS projects, and propose a way forward.
3. Determine the security considerations and the available tools to support an SSO federation that helps users in participating in several projects by using a single user account.
4. Assess the possible relationships of OGC standards (e.g. SensorML) with other existing standards in the field (e.g. Public Participation in Scientific Research (PPSR) - Core, the ontology developed by the COST Action on Citizen Science, and the Citizen Science Definition Service (CS-DS) developed in the NextGEOSS project).
5. Satisfy and document the necessary requirements to integrate Citizen Science into Global Earth Observations System of Systems (GEOSS) by using OGC standards.

This IE has been promoted by the OGC Citizen Science Domain Working Group, the WeObserve and NextGEOSS H2020 projects, and The Earth Challenge 2020 project as supported by National

Geographic Society. This IE contributes not only to the interoperability and possibly standardization program of the OGC, but also to the Global Earth Observation System of Systems (GEOSS). This work is also relevant to the foundational objectives of the Citizen Science Global Partnership (CSGP). Regional and national Citizen Science Associations will equally benefit from the results of this OGC IE.

4.1. Structure of the activities

The official kick off meeting for the OGC CitSciIE experiment was held on Friday 14th September 2018 at the OGC TC meeting in Stuttgart. Activities have continued until March 2019.

During the Kick off meeting of the Experiment the following subgroups emerged:

- V: Vocabularies for organizing Citizen Science projects. There was a discussion on essential variables but also on other kind of practices that can be associated to vocabularies, i.e. on how to publish vocabularies (PublishingDefs) or on defining a list of vocabularies that could be useful to experiment with (observations, project descriptions, general glossaries of terms).
    - Working item V.1: A list of the current projects that Wilson Center knows that can align with the Earth Challenge topics (air and water quality, pollution, human health and eventually biodiversity) and extraction of a common set of variables they usually cover.
    - Working item V.2: Analysis of data models that contributors in the experiment can bring in: Air quality (HackAir), Biodiversity (Atlas of Living Australia & Natusfera), Mosquito (CREAF), Land Use (IIASA), Phenology (CREAF), Invasive Alien Species (JRC).
    - Working item V.3: Consider the COST action metadata model for inclusions as another vocabulary, this might include a set of definitions of phenomena that are being addressed by CS initiatives (based on the inventory of citizen science activities for environment policies).
- D: Data sharing using OGC standards such as O&M and SOS. A pool of services were identified for participating in an interoperability experiment, including SOS services and clients and citizen science project databases and APIs.
    - Working item D.1: A set of instructions on how a CS project can easily setup an SOS service. It could include 52North implementation and might include MiraMon SOS (with some work in the implementation). It should address the case of a small project contributing to the Earth Challenge 2020.
    - Working item D.2: Create an SOS endpoint for HackAir data with minimum resources
    - Working item D.3: Define the requirements for a data provider that could assist Wilson Center in setting up the challenge database. It should consider upload of data into the system. It seemed to go for a harvest system instead of a federated system. It could describe a possible architecture to allow the dialog between the central database and the small contributing projects. It should impose data sharing requirements (services o APIs) on the central database.

- S: Connection between Landsense federation and JRC user system.
  - Working item S.1: Interoperability test on the integration of LS-SSO and JRC-SSO
- Q: Data quality
  - Working item Q.1: Write a document on perspectives of the different quality aspects: Quality assessment (ISO 19157-QualityML), Quality improvement, Quality plan, Data Management principles (ISO 8001), Quality documentation, Quality communication
  - Working item Q.2: Perfect the quality measurement system based on WPS and SOS harvest by demonstrating the concept in practice. Also include in the SOS harvesting the possibility to have a query for assessing the quality of "views"/"selections"/"fragments" of a dataset.
    - Connection with: D.2
  - Working item Q.3: Refine the QualityML vocabulary with new entries considering the work done in Australia
    - Connection with: D.3
  - Working item Q.4: Add new entry point the QualityML for other common vocabulary formats like TTL etc.
    - Connection with: V

For each of the subgroups a chair and the main participants and contributors were identified. Responsible persons were also assigned to each of the working items.

4.2. Results detailed in next sections

These are the main activities and outcomes of the interoperability experiment detailed by activity.

Data sharing using OGC standards such as O&M and SOS

This activity has been the most active one. During the testbed the following servers have been deployed: MiraMon SOS server, Grow SOS, DLR istSOS SOS and 52north SOS. Three clients have also been produced: MiraMon SOS browser, Grow SOS data viewer and 52north Helgoland. In the last meeting at the EGU, the group was able to demonstrate interoperability by connecting the SOS clients to the SOS services and showing the data on clients, sometimes mixing data form different services and datasets in a single view. This is the most significant result of the experiment and is being extensively documented in this Engineering Report in sections 5, 6 and 7

Data quality

Two quality vocabularies have been detected: an Australian work done by Peter Brenton team (https://github.com/tdwg/bdq) and the QualityML vocabulary developed by CREAF in the GeoViQua project. The intention was to do a comparison of both approaches but we were not able to do that in the timeframe of the first interoperability experiment. Section 8 describes the current status of the activity. It is foreseen that the second IE will continue what was started here.

Definition server for organizing Citizen Science projects

The objective of this activity was to support the Earth Challenge 2020 research questions. The questions were defined during the first month of the experiment and now it is time to analyse them in terms of data needs and thematic vocabularies they are going to use. That is the reason this activity has not resulted in tangible outputs and will be reintroduced in the second interoperability experiment. Details of this development are described in section 9.

Connection between Landsense federation and JRC user system

Andreas Matheus SME (Secure Dimensions) was very active in providing demonstrations and information on how the LandSense federation works and how other projects can be included in the federation and use the single Sign on facility. Unfortunately, no other member of the CoP had the resources to apply the SSO on their services or clients and take advantage of the Andreas offering. The activity resulted in a video demonstrations that are publicly available here: https://portal.opengeospatial.org/files/?artifact_id=81550. Section 10 Details the current status of the activity

Other

Section 11 summarizes the lessons learned that can be applied to GEOSS.

In addition to these activities, another activity about quality annotating scientific documentation in a standard way was proposed by Lucy Bastin. A video was recorded that summarizes the idea: https://portal.opengeospatial.org/files/?artifact_id=82544.

Chapter 5. O&M for Cit Sci

In a *feature model* all characteristics of a feature are considered properties of the feature and are not semantically separated at the abstract level.

The Observations and Measurements(O&M) standard (OGC 10-004r3, Topic 20: Observations and Measurements) defines a data model that observations where. main concepts are separated as represented in the Figure 1:

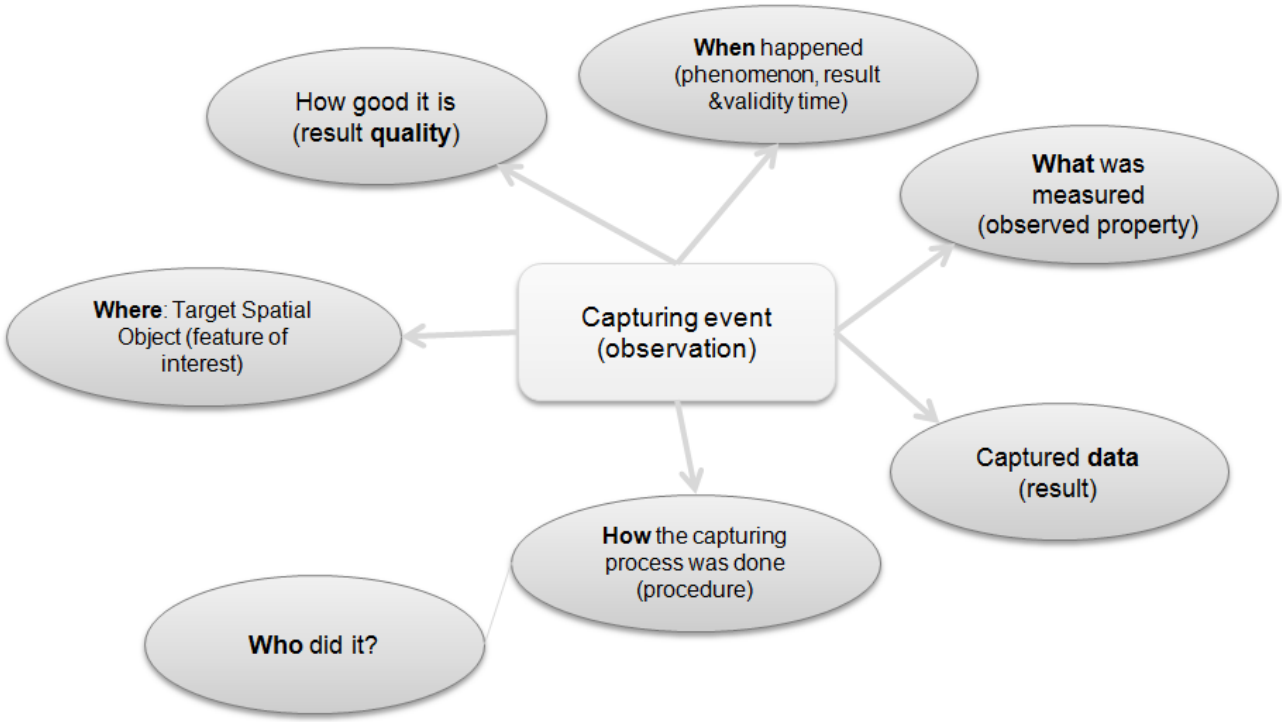


Figure 1. Observation main concepts

For each observation, the O&M allows us to document the following characteristics:

- Where the observation is located: Even if the observations was done remotely with a camera or a drone, it is commonly more relevant to know the position of the observed phenomena (the sensor position can also be recorded).
- When the observation took place and what time period it represents: Even if samples were collected and analyzed later, it is commonly more relevant to know the instant or period of the observed phenomenon.
- How the observation was done: This will describe the procedure and instrument used to capture the phenomenon.
- Who did the observation: The procedure and instrument used to capture the phenomenon was installed or used on side by someone. In citizen Science, where many observers contribute small pieces of information that together will form a dataset, it is particularly important to record at least an observer identifier.
- What was measured: This will define the property names and units of measure of the

variables observed.

- What data was collected: This will record the actual values of the properties measured.
- What is the expected quality of the observation: If an estimation of the quality of the observation was done, it is important to document it.

In the O&M data model, there aspects are clearly separated semantically as shown in Figure 2. This is the main value of the O&M model and its usage SOS (or the SensorThingsAPI that uses a very similar approach to model the data), but it is also the main handicap in applying the standard.

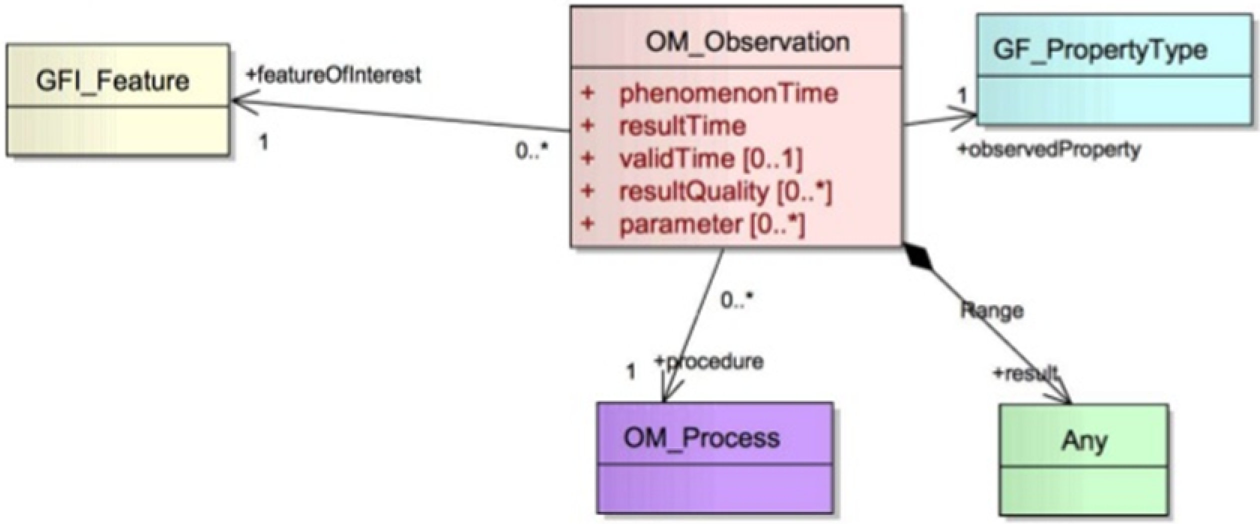


Figure 2. Common Model For Oservations and Measurements

Concept	O&M	type
Where	featureOfInterest	GFI_Feature
When	phenomenonTime, resultTime	
How	procedure	OM_Process
Who	procedure	OM_Process
What	observedProperty	GF_PropertyType
Data	result	Any
Quality	resultQuality	

Table 1. Concepts in an observation mapped to the O&M model

Even if this aspects are separated, the O&M model gives a lot of flexibility in defining the properties and this flexibility can condition interoperability when trying to combine data from different sources. The standard give us freedom to select among the different geometries provided by GML to define the featureOfInterest. The standard gives us even more freedom on the data collected that can have any imaginable structure.

That is the reason why the data model used to represent the data gathered by a Citizen Observatory need to be carefully considered before even starting the first campaign. Data models can be designed in UML for clarity but they are later encoded in XML. XML is the only official encoding that O&M referenced in the OGC website (OGC 10-025r1 Observations and Measurements - XML Implementation v2.0). Nevertheless, there is JSON alternative discussed in a OGC discussion paper (OGC 15-100r1 OGC Observations and Measurements – JSON implementation) that does not represent an official position of the OGC but can be implemented anyway. As it will be discussed latter, the interpretation of long XML files might be to slow in web browsers, and, in this cases, a JSON encodings is regarded as a good alternative. This alternative can be the proposed OGC 15-100r1 or the alternative proposed by the SensorThings API.

5.1. The GT20 examples

In the Ground Truth 2.0 project we have been using the MiraMon implementation of O&M. This implementation assumes a simplified situation that considers that each observation can be represented by a single row in a CSV or in a single record of a database table. Coordinates are represented as a single point. In this situation, we select which column names represent the *phonomenonTime*, the *procedure* (that actually is including the user name) and the *featureOfInterest* (the coordinates). The rest of the columns are considered part of the data record that needs to be provided as the *result*.

The section 8.2.1 of the OGC 08-094r1, OGC® SWE Common Data Model Encoding Standard v2.0 describes a way to encode a *DataRecord* as an array of fields that can numbers, strings, dates, etc. In our simplified assumption, this results ideal to wrap the properties of the observations that cannot be mapped to any other O&M aspect. This practice is consistent with the section 7.2.8 of the SWE4CS discussion paper.

The following is an example of how a water quality observation is represented following the O&M model and encoded in XML:

Example of XML encoding of the water quality observation

```
<om:OM_Observation gml:id="vatten-fokus_2_1">
    <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_ComplexObservation"/>
    <om:procedure xlink:href="http://www.opengis.uab.cat/vatten-
fokus/procedure/22655"/>
    <om:observedProperty xlink:href="http://www.opengis.uab.cat/vatten-
fokus/observedProperty"/>
    <om:featureOfInterest xlink:href="http://www.opengis.uab.cat/vatten-
fokus/featureOfInterest/2"/>
    <om:result xsi:type="swe:DataRecordPropertyType">
        <swe:DataRecord>
            <swe:field name="CREA_DATE">
                <swe:Text definition="http://www.opengis.uab.cat/vatten-
fokus/field/Creation_Date">
```

```
                    <swe:value>07/12/2018 17:23</swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="SITE_NAME">
                <swe:Text definition="http://www.opengis.uab.cat/vatten-
fokus/field/Site_name">
                    <swe:value>Dunkershall. V¤gtrumma uppst¤ms.</swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="LAND_USE">
                <swe:Text definition="http://www.opengis.uab.cat/vatten-
fokus/field/Land_use_in_the_immediate_surroundings">
                    <swe:value>Agriculture</swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="BANK_VEGE">
                <swe:Text definition="http://www.opengis.uab.cat/vatten-
fokus/field/Bank_vegetation">
                    <swe:value>Grass</swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="NITRATE">
                <swe:Quantity definition="http://www.opengis.uab.cat/vatten-
fokus/variable/NITRATE">
                    <swe:uom/>
                    <swe:value>1.50</swe:value>
                </swe:Quantity>
            </swe:field>
            <swe:field name="PHOSPHATE">
                <swe:Quantity definition="http://www.opengis.uab.cat/vatten-
fokus/variable/PHOSPHATE">
                    <swe:uom/>
                    <swe:value>0.075</swe:value>
                </swe:Quantity>
            </swe:field>
            <swe:field name="WATER_COLOR">
                <swe:Text definition="http://www.opengis.uab.cat/vatten-
fokus/field/Estimated_water_colour">
                    <swe:value>Colourless</swe:value>
                </swe:Text>
            </swe:field>
        </swe:DataRecord>
    </om:result>
</om:OM_Observation>
```

The following is an example of how two air quality observations are represented following the

O&M model and encoded in JSON:

Example of encoding of the air quality observation


```
{
    "id":"meet-mee-mechelen_1_0",
    "type" : "http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_ComplexObservation",
    "phenomenonTime" : "2017-11-19 17:20:00+01",
    "resultTime" : "2017-11-19 17:20:00+01",
    "procedure" : "http://www.opengis.uab.cat/meet-mee-mechelen/procedure/5",
    "observedProperty" : "http://www.opengis.uab.cat/meet-mee-
mechelen/observedProperty",
    "featureOfInterest" : "http://www.opengis.uab.cat/meet-mee-
mechelen/featureOfInterest/1",
    "result": {
        "type":"DataRecord",
        "field":[
            {
                "name" : "CAMPAIGN",
                "type" : "Text",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/field/CAMPAIGN",
                "value" : "Oct-Nov2017"
            },
            {
                "name" : "bc_aggr",
                "type" : "Quantity",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/bc_aggr",
                "value" : "3155"
            },
            {
                "name" : "bc_aggr_mi",
                "type" : "Quantity",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/bc_aggr_mi",
                "value" : "80"
            },
            {
                "name" : "bc_aggr_ma",
                "type" : "Quantity",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/bc_aggr_ma",
                "value" : "16413"
            },
            {
```

```
                "name" : "bc_aggr_st",
                "type" : "Quantity",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/bc_aggr_st",
                "value" : "3398"
            },
            {
                "name" : "uncertaint",
                "type" : "Quantity",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/uncertaint",
                "value" : "0.50"
            }
        ]
    }
},
{
    "id":"meet-mee-mechelen_2_1",
    "type" : "http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_ComplexObservation",
    "phenomenonTime" : "2017-11-19 17:20:06+01",
    "resultTime" : "2017-11-19 17:20:06+01",
    "procedure" : "http://www.opengis.uab.cat/meet-mee-mechelen/procedure/5",
    "observedProperty" : "http://www.opengis.uab.cat/meet-mee-
mechelen/observedProperty",
    "featureOfInterest" : "http://www.opengis.uab.cat/meet-mee-
mechelen/featureOfInterest/2",
    "result": {
        "type":"DataRecord",
        "field":[
            {
                "name" : "CAMPAIGN",
                "type" : "Text",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/field/CAMPAIGN",
                "value" : "Oct-Nov2017"
            },
            {
                "name" : "time_first",
                "type" : "Text",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/field/time_first",
                "value" : "2017-11-06 08:00:18+01"
            },
            {
                "name" : "bc_aggr",
                "type" : "Quantity",
```

```
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/bc_aggr",
                "value" : "3382"
            },
            {
                "name" : "bc_aggr_mi",
                "type" : "Quantity",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/bc_aggr_mi",
                "value" : "80"
            },
            {
                "name" : "bc_aggr_ma",
                "type" : "Quantity",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/bc_aggr_ma",
                "value" : "17256"
            },
            {
                "name" : "bc_aggr_st",
                "type" : "Quantity",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/bc_aggr_st",
                "value" : "3663"
            },
            {
                "name" : "number_of_",
                "type" : "Quantity",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/number_of_",
                "value" : "25"
            },
            {
                "name" : "number_o_1",
                "type" : "Quantity",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/number_o_1",
                "value" : "13"
            },
            {
                "name" : "mean_numbe",
                "type" : "Quantity",
                "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/mean_numbe",
                "value" : "7"
            },
            {
```

```
                  "name" : "uncertaint",
                  "type" : "Quantity",
                  "definition" :"http://www.opengis.uab.cat/meet-mee-
mechelen/variable/uncertaint",
                  "value" : "0.50"
              }
          ]
      }
}
```

These examples were produced by SOS requests to this URL: http://www.ogc3.uab.cat/cgi-bin/CitSci/MiraMon.cgi?. A client connecting to this service can be found here: http://www.ogc3.uab.cat/gt20/

5.2. HackAir examples

To illustrate the flexibility of the O&M we have included this air quality report that shows how HackAir data is presented by a 52North SOS implementation. In this case the *result* presents a single numerical value while the other information is provided as parameters. This approach is consistent with section 7.2.2.5 of the O&M standard.

Example of encoding of the water quality observation

```xml
<om:OM_Observation gml:id="o_499">
    <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_Measurement"/>
    <om:phenomenonTime>
        <gml:TimeInstant gml:id="phenomenonTime_499">
            <gml:timePosition>2019-01-01T00:00:12.000Z</gml:timePosition>
        </gml:TimeInstant>
    </om:phenomenonTime>
    <om:resultTime xlink:href="#phenomenonTime_499"/>
    <om:procedure xlink:href="sensors_arduino_1000"/>
    <om:parameter>
        <om:NamedValue>
            <om:name xlink:href="PM2.5_AirPollutantIndex"/>
            <om:value xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type=
"xs:string">bad</om:value>
        </om:NamedValue>
    </om:parameter>
    <om:parameter>
        <om:NamedValue>
            <om:name xlink:href="http://www.opengis.net/def/param-name/OGC-
OM/2.0/samplingGeometry"/>
            <om:value xmlns:ns="http://www.opengis.net/gml/3.2" xsi:type=
```

```
"ns:GeometryPropertyType">
                <ns:Point ns:id=
"Point_sp_45C0E376C40E98E8EC0D48C05F7558C2FFD15245">
                    <ns:pos srsName=
"http://www.opengis.net/def/crs/EPSG/0/4326">52.063269625917
4.5077472925186</ns:pos>
                </ns:Point>
            </om:value>
        </om:NamedValue>
    </om:parameter>
    <om:parameter>
        <om:NamedValue>
            <om:name xlink:href="source"/>
            <om:value xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type=
"xs:string">sensors_arduino</om:value>
        </om:NamedValue>
    </om:parameter>
    <om:parameter>
        <om:NamedValue>
            <om:name xlink:href="user"/>
            <om:value xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type=
"xs:string">sID :1000</om:value>
        </om:NamedValue>
    </om:parameter>
    <om:observedProperty xlink:href="PM2.5_AirPollutantValue" xlink:title=
"PM2.5_AirPollutantValue"/>
    <om:featureOfInterest xlink:href="sensors_arduino_1000"/>
    <om:result xmlns:ns="http://www.opengis.net/gml/3.2" uom="µg/m3"
xsi:type="ns:MeasureType">130.67</om:result>
</om:OM_Observation>
```

A service producing this type of results can be seen here: https://nexos.demo.52north.org/52n-sos-hackair-webapp/service

5.3. GROW example

In the GROW project the SME Hydrologic has develop an SOS service that uses a O&M observation. In this case, a single number is provided as the *result* of the observation and additional parameters are transported.

```xml
<OM_Observation xmlns="http://www.opengis.net/om/2.0">
    <type gml:remoteSchema="http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_Measurement" />
    <phenomenonTime>
        <gml:TimePeriod>
            <gml:beginPosition>2018-09-03T09:01:38.000Z</gml:beginPosition>
            <gml:endPosition>2018-09-03T09:01:38.000Z</gml:endPosition>
        </gml:TimePeriod>
    </phenomenonTime>
    <resultTime>
        <gml:TimeInstant>
            <gml:timePosition>2018-09-03T09:01:38.000Z</gml:timePosition>
        </gml:TimeInstant>
    </resultTime>
    <procedure>Grow.Thingful.Sensors_je47sfac</procedure>
    <observedProperty nilReason=
"Thingful.Connectors.GROWSensors.AirTemperature" />
    <featureOfInterest nilReason="je47sfac" />
    <result>20.64</result>
</OM_Observation>
```

5.4. Future work

So far we have seen 3 servers using 2 different approaches to represent the result. That is not a problem for a web service (that only outputs data) but it is not the best situation to ensure interoperability at the client side where an integrated client will need to react to any possible encoding variation and deliver the best result.

5.4.1. How to encode the procedure.

The SWE4CS discussion paper suggest that we use an approach to encode the procedure that takes into account a recommendation extracted from section 6.18.1 of Timeseries Profile of Observations and Measurements standard (OGC 15-042r5) that suggests an encoding for both the observation *process* and the *operator* of the sensor (the *citizen* doing Citizen Science) that is based on ISO metadata. This will ensure an uniform way to report on this two important aspects of the observation.

NOTE	This approach has not been implemented during the interoperability experiment but it is considered something we can experiment with in the future. An example of this procedure is provided in the SWE3CS document and reproduced here for convenience

Example of encoding of the procedure (including process and operator) extracted from the SWE4CS document

```
<om:procedure>
  <tsml:ObservationProcess gml:id="op1">
    <!-- processType defines observation performed by human with sensor -->
    <tsml:processType
xlink:href="http://www.opengis.net/def/waterml/2.0/processType/Sensor"/>
    <!-- processReference defines sampling protocol -->
    <tsml:processReference
xlink:href="https://dyfi.cobwebproject.eu/skos/JapaneseKnotweedSamplingProtoc
ol"/>
    <!-- if a sensor is used, provide the link to the sensor definition here.
Use
SensorML if possible -->
    <tsml:parameter>
      <om:NamedValue>
        <om:name xlink:href=
"http://www.opengis.net/def/property/OGC/0/SensorType"/>
        <om:value>http://www.motorola.com/XT1068</om:value>
      </om:NamedValue>
    </tsml:parameter>
    <!-- operator defines the citizen scientist producing this observation
-->
    <tsml:operator>
      <gmd:CI_ResponsibleParty>
        <gmd:individualName>
          <gco:CharacterString>Ingo Simonis</gco:CharacterString>
        </gmd:individualName>
        <gmd:organisationName>
          <gco:CharacterString>OGC</gco:CharacterString>
        </gmd:organisationName>
        <gmd:role>
          <gmd:CI_RoleCode
    codeList=
"http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml"
    codeListValue="resourceProvider"/>
        </gmd:role>
      </gmd:CI_ResponsibleParty>
    </tsml:operator>
  </tsml:ObservationProcess>
</om:procedure>
```

The result is quite verbose what might affect performance when many data is transmitted.

5.4.2. Avoiding verbosity by defining a data streem

An approach based on providing a coma separated recordset that is described only once at the beginning should be more compact and efficient to parse.

The section 8.4.3 of the OGC 08-094r1, OGC® SWE Common Data Model Encoding Standard v2.0 describes a way to encode a *DataStream* only once and then send the data directly as a CSV format using HTTP of other protocol. A similar solution could be worth to be tested in the future to increase performance.

Chapter 6. SOS architectures

In this chapter we describe three architectures tested in the Interoperability Experiment that demonstrate end to end architectures as well as interoperability among servers and clients.

6.1. Architecture 1: SOS services integrated in a SOS client

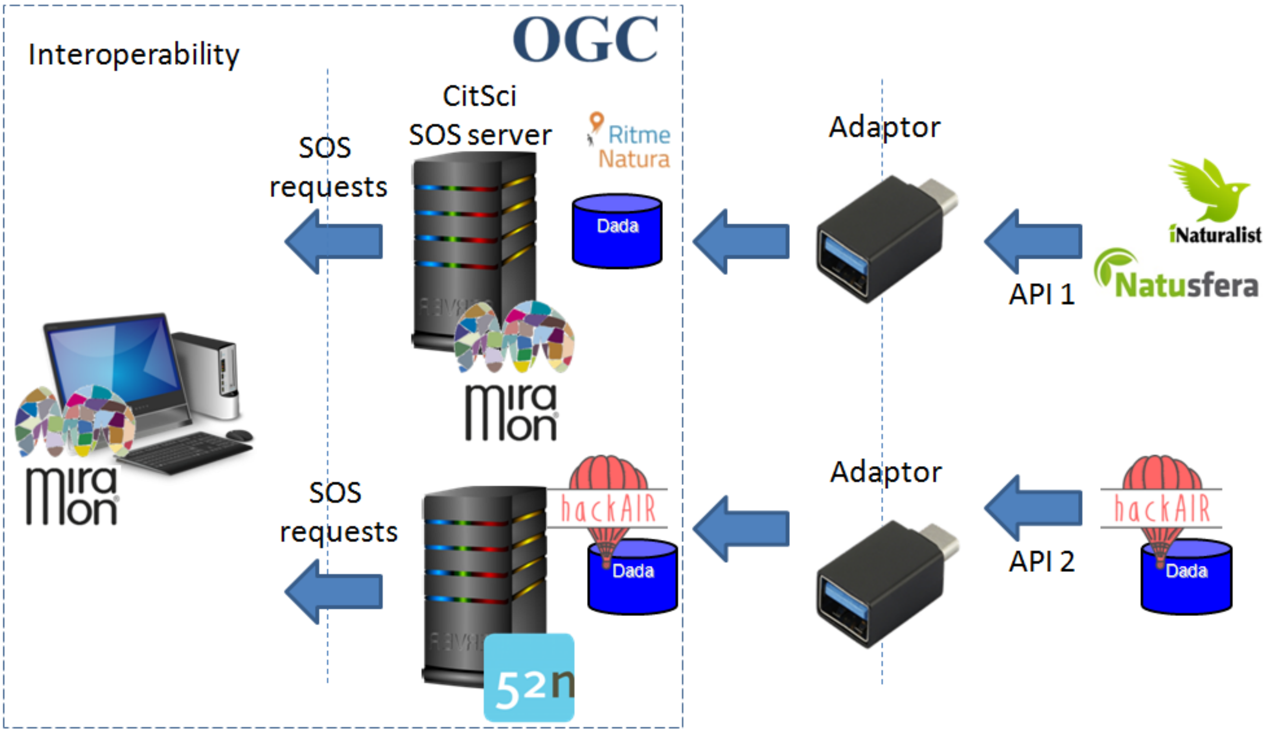


Figure 3. MiraMon SOS Architechture

In this architecture ([MiraMonSOSArchit]) the client access directly two different SOS services. It formulates a GetFeatureOfInterest to determine the positions of the individual observations and a GetObservation each time it needed to show a complete description of a single point (the user triggers this event by clicking on an icon) or if it needs to represent different icons as a function of the value of the observation. In this case, interoperability happens directly in the client. Since the SOS requests are communicated to the Internet, this client is exposing requests and response, allowing people to explore the SOS protocol with both the map browser console as well as the browser developing tools.

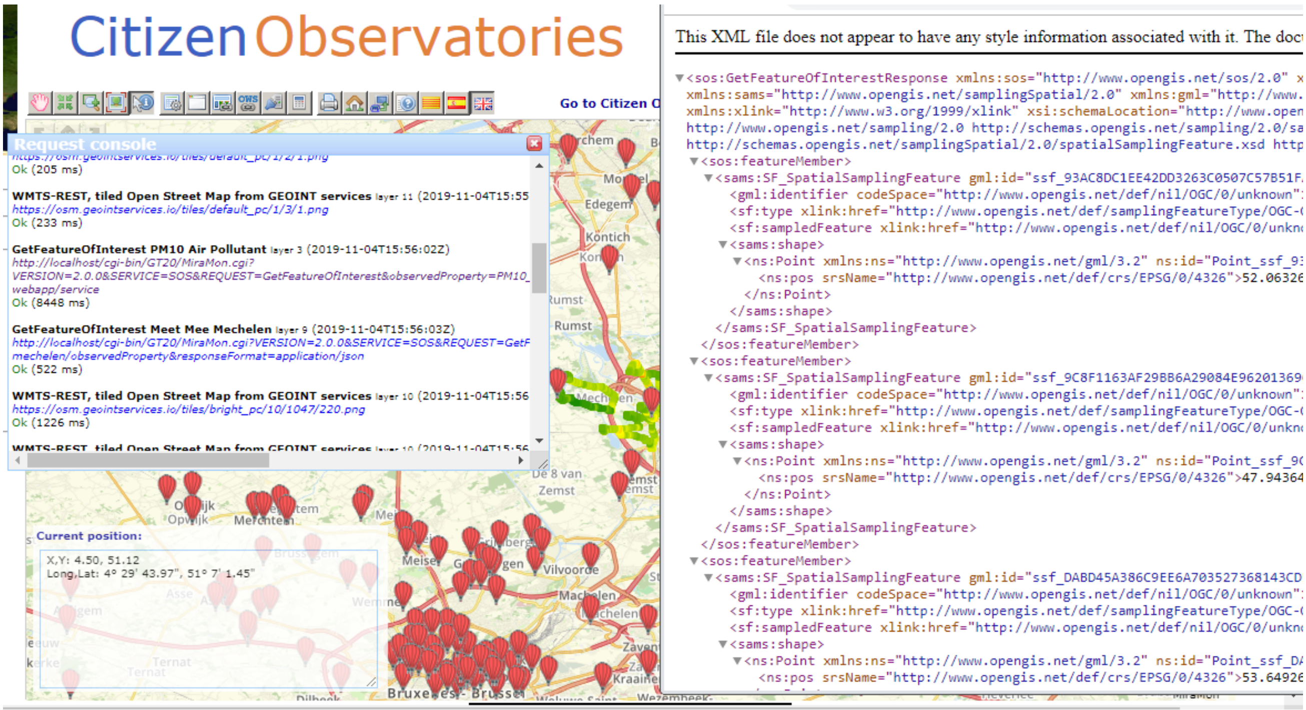


Figure 4. MiraMon Map Browser console with both server requests

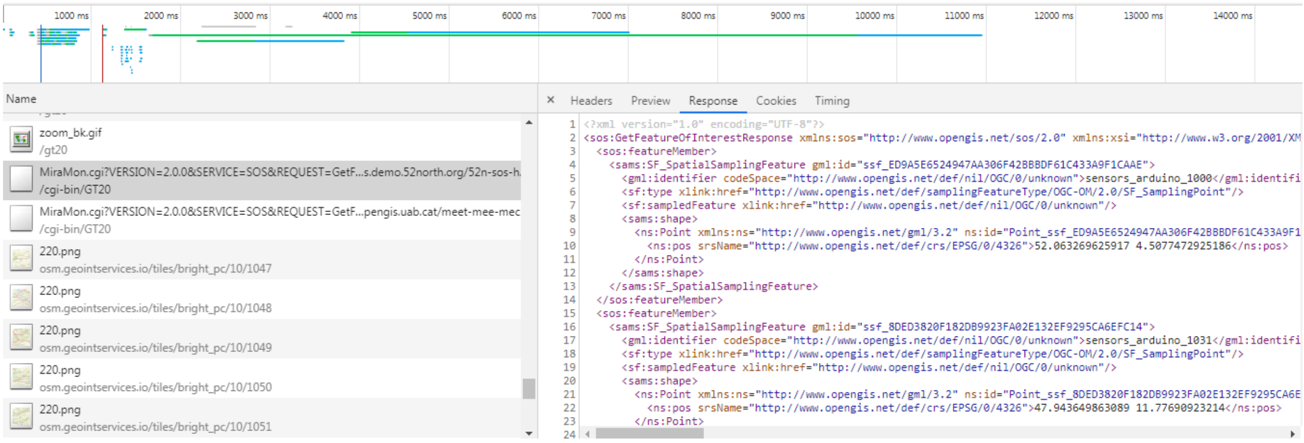


Figure 5. SOS requests as shown in the Chrome development tool

It is worth mentioning that this architecture is only possible if both services are declaring their willingness to be combined in the header of the responses. By default, programatically reading XML or JSON data coming from a Internet domain different from the client itself is not allowed except if the server states in the header that this is allowed. This is known as Cross-Origin Resource Sharing (CORS). The following headers will allow CORS with anybody.

```
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE
```

In the case a SOS server does not allow CORS, our client is still able to force a solution by redirecting the request to our server with an extra parameter *ServerToRequest*. In this case, our server will cascade the request to the specified server and return the response back to the client as if there was only one server involved in only one domain.

6.2. Architecture 2: SOS services integrated in a combined agile service

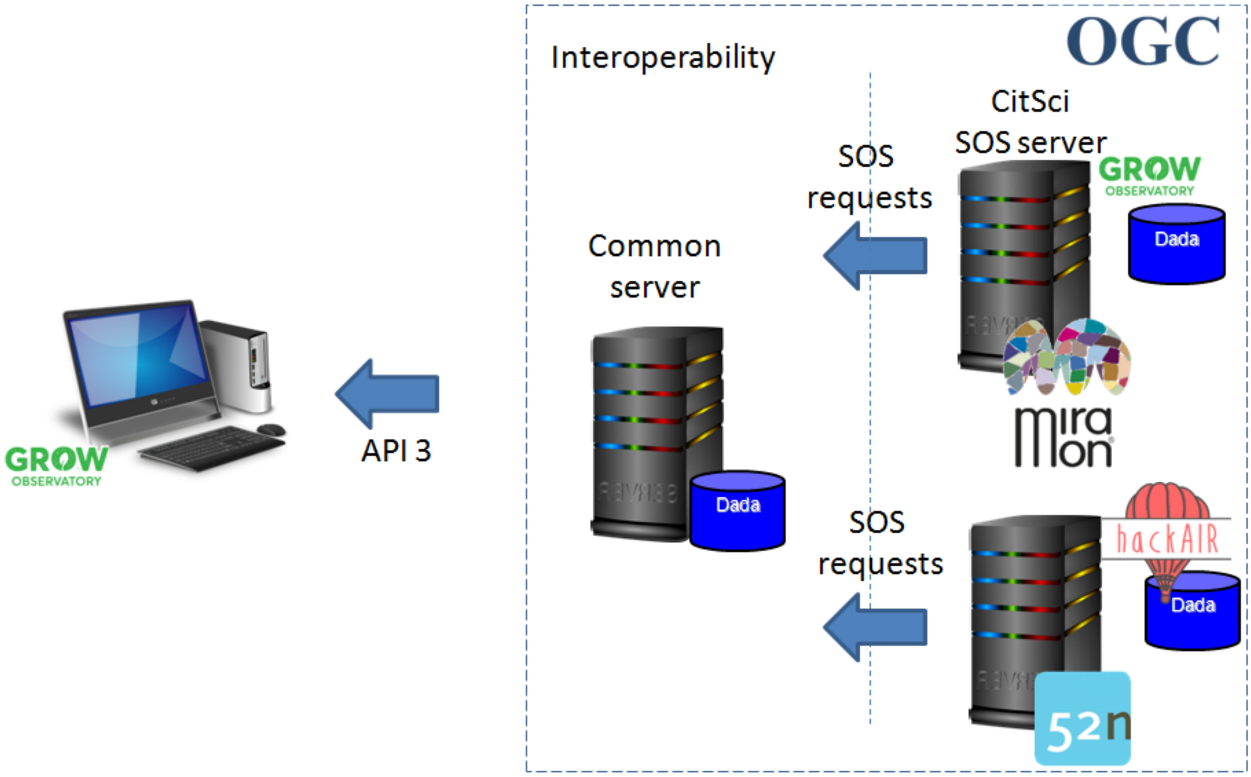


Figure 6. Grow SOS Architechture

In this architecture ([GrowSOSArchit]) a common server pulls two or more SOS requests into a central tabular datastore. This datastore records only the information from the returned SOS data that is required for the final visualisation and removes information that is redundant creating a data warehouse representing only one version of the data. In this approach the interoperability happens internally in the datastore and the SOS requests and responses are not explosed to the final client. In the diagram below ([GRowDataFlow]), data is stored in the datawarehouse and Microsoft’s PowerBI does the heavy lifting for the visualisation of the combined data sources.

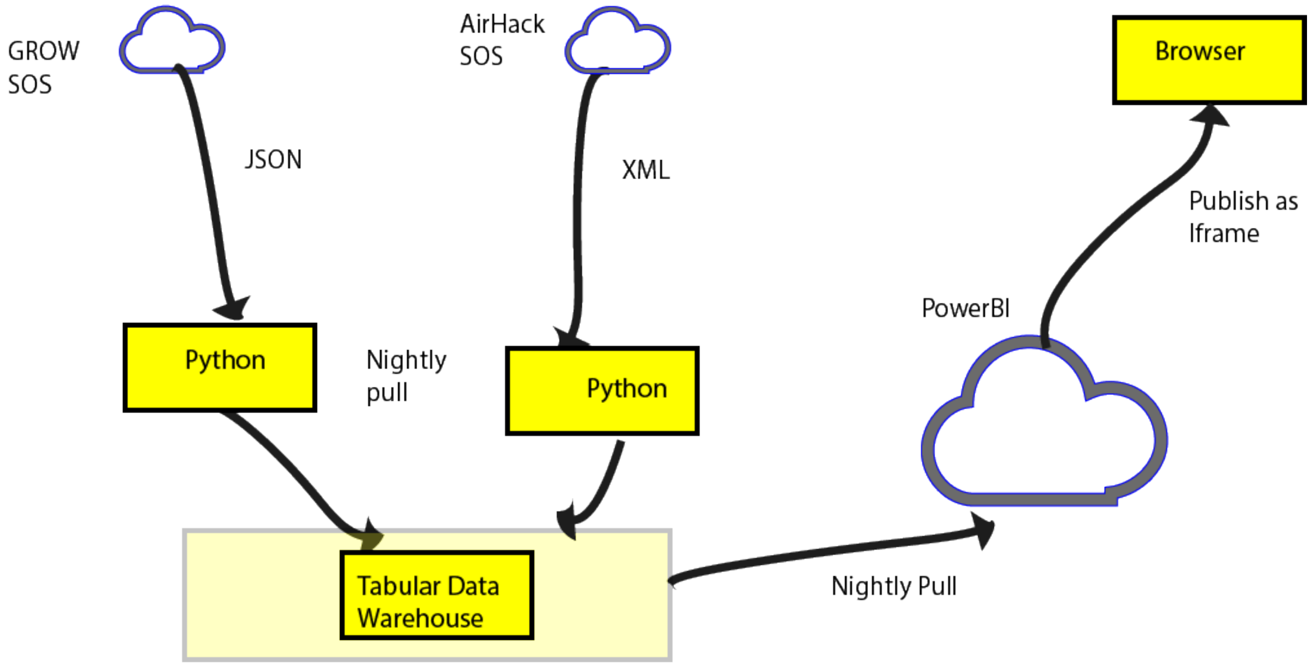


Figure 7. Grow SOS Data Flow

The common server would typically be a cloud server, but for some clients this is not necessary, in the case of PowerBI a data bridge is created between the data source and the visualisation tool before it is published to a web client. Other visualisation tools (such as Tableau) will have their own methods of connecting to the data warehouse and publishing the results to a web based client. In this architecture, data is only as up to date as the latest data pull from the SOS servers; in the case of GROW this is done nightly but this could be made more frequent or moved towards real time using a data log pipeline in a kappa architecture.

6.3. Architecture 3: SOS service for interoperability and JSON API for fast client

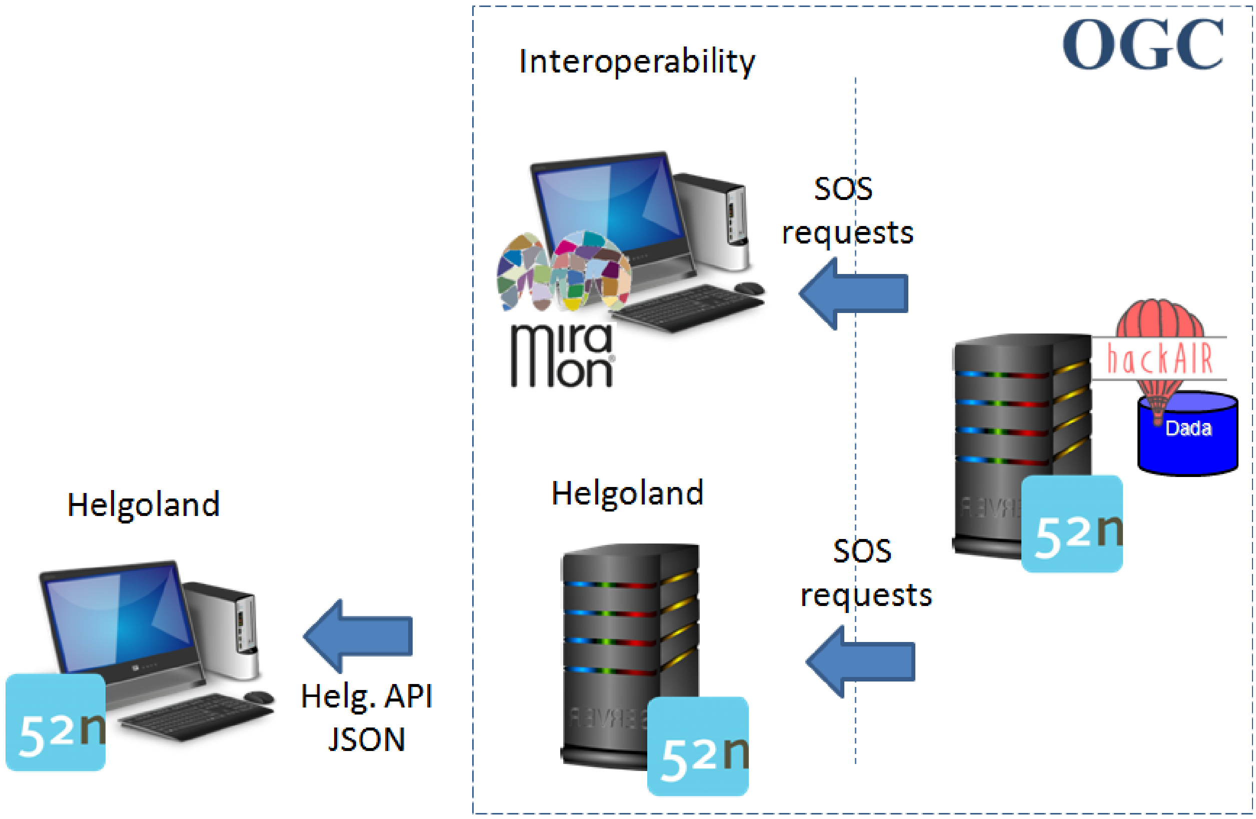


Figure 8. Helgoland SOS Architechture

This architecture was especially optimised to support the development of lightweight Sensor Web applications. This is achieved by avoiding the direct XML encoding/decoding on the client device. Instead, the interactions between the client (in this case the 52°North Helgoland Sensor Web Viewer) and the server components is achieved via a REST and JSON interface (the 52°North Sensor Web API).

This API can be directly exposed by Sensor Web servers such as the 52°North implementation. Alternatively, an available proxy component is also capable to encapsulate existing OGC SOS servers behind the lightweight interface of the 52°North Sensor Web API.

The advantage of this approach is a more lightweight communication pattern to be implemented on the client side. In addition the 52°North Sensor Web API offers further convenience methods as well as functionalities for reducing the transferred data volume (by generalising observation data) and improving the data visualisation (e.g. providing rending hints). A drawback of this approach is a less direct interaction with SOS servers, so that for integrating new SOS servers, a proxy component has to be configured/adjusted.

Chapter 7. SOS servers

In this chapter we describe four SOS servers tested in the Interoperability Experiment.

7.1. 52 North solution

The 52°North Sensor Web Server comprises several server-side modules which closely interact to provide different kinds data access functionality. In detail, this comprises the following elements:

- Data storage: The database for storing the observation data is integrated through an object-relational mapping layer based on the Hibernate framework. This allows the flexible integration of different types of database management systems (e.g. PostgreSQL, Oracle, MS SQL Server, MySQL) and data models. For this Interoperability Experiment, PostgreSQL was used.
- For the access to observation data, the server offers three dedicated modules, which use the same common Sensor Web database.
  - SOS: The SOS module offers a comprehensive implementation of the OGC Sensor Observation Service 2.0 standard (including beyond the core several extended functionalities, transactional and result handling operations). It also offers several interoperability enhancements such as a support of the INSPIRE Technical Guidance on the SOS as a Download Service.
  - SensorThings API: In addition to the SOS support, a dedicated module is available for supporting the OGC SensorThings API Part 1: Sensing (not yet evaluated as part of this Interoperability Experiment).
  - 52°North Sensor Web API: Complementary to the previous modules, also the 52°North Sensor Web API is offered. This API offers an additional but optional convenience layer for building client applications. While both the SOS and the SensorThings API standards are well suited for enabling the interoperable access to observation data, the Sensor Web REST-API allows to provide additional functionality that significantly facilitates the development of client applications. Typical examples of this additional functionality comprise: generalisation of observation data (important for developing mobile applications), provision of rendering hints (e.g. styling information for time series), conversion of data to mainstream formats such as CSV.

The URL of the instance used in this IE was: https://nexos.demo.52north.org/52n-sos-hackair-webapp/service.

More information about the initiative can be found here: https://52north.org/software/software-projects/sos/

7.2. istSOS

istSOS (Istituto Scienze della Terra Sensor Observation Service) is an OGC SOS server implementation written in Python. istSOS allows for managing and dispatch observations from monitoring sensors according to the Sensor Observation Service standard.

istSOS evolved in time from being a SOS service provider to complete data management system. But the standard does not account for a number of functionalities that were later included in the software. Some of the extending capabilities are:

- Handle of irregular time series
- On-the-fly aggregation of observed measures with no-data management.
- Capability to filter observations based on partial observed property names (LIKE filtering support).
- Native support for data validation and data quality index associated with each observation

The project also provides also a Graphical user Interface that allows for easing the daily operations and a RESTFul Web API for automatizing administration procedures.

The URL of the instance used in this IE was: http://artemis.geogr.uni-jena.de/istsos_ie/soil?service=SOS&request=GetCapabilities&version=1.0.0 More information about the initiative can be found here: http://istsos.org/

7.3. MiraMon SOS

MiraMon Server is a stand alone CGI application that runs on Windows operating systems that can be used in combination with a web server such as Internet Information Server or Apache for Windows. It is the ideal solution for people that already uses MiraMon professional on desktop because it uses the same MiraMon formats in the back-end. MiraMon server is based on the same libraries that are used by MiraMon professional and has the same capabilities in terms of CRS support, interpolation algorithms, MMZX compression etc. One particularity of the software is the internal tiled schema required to serve maps and tiles in a fast an scalable way. MiraMon Server uses OGC web services as a baseline for the interaction to the client. Currently, MiraMon server provides support for the following standards: * Web Map Service (all versions) * Web Map Tile Service (all versions) * Web Coverage Service (version 1.0) * Web Feature Service (version 2.0) * Web Processing Service (version 1.0) * Sensor Observing Service (version 2.0)

The Sensor Observing Service capacity is used in tandem with the Web Feature Service and uses the same MiraMon topologically structured formats in the back-end. It has been developed in the Ground Truth 2.0 project to server interoperable data from the Citizen Observatories created during the project. The current implementation is incomplete and only supports GetFeatureOfInterest and GetOBservation operation with limited capabilities. The objective of the minimum capabilities developed was to repont to the requirements of a viewer client needs to represent a map of the features of interest provided by the service and to allow for a query in a point to get more information about the observations there. Each dataset in MiraMon becomes a observedProperty in the SOS service. Each observation is a position in a PNT file that has a DBF record associated that is transformed to a O&M DataRecord automatically. Internally, it is possible to mark field names in the DBF that are associated to concepts in the O&M such as the phenomenon time and the user name.

This is the internal format for the small REL5 document necessary to include the extra information that the server requires.

```
file name: C:\inetpub\SIWeb\gt20\VattenFokus\VattenFokusT.dbf
Last update on: 24-01-2019
Number of records: 254
Number of fields per record: 32
Character set: Windows ANSI (88, 0x58)

Field characteristics:
-------------------------------------------------------------------------------
--------------
NUM |               NAME |                                        DESCRIPTOR |
T | SIZ | REL
-------------------------------------------------------------------------------
--------------
  1 |          ID_GRAFIC |       Identificador Gràfic ID | N |   3 |
  2 |            USER_ID |                       User ID | N |   5 |
  3 |          SAMPLE_ID |                     Sample ID | N |   5 |
  4 |          CREA_DATE |                 Creation Date | C |  16 |
  5 |          CHAN_DATE |             Modification date | C |  16 |
  6 |         SAMPLEDATE |                   Sample date | C |  16 |
  7 |           GROUP_ID |                      Group ID | C |  34 |
  8 |          SITE_NAME |                     Site name | C | 111 |
  9 |   Sample_date_time |              Sample date/time | C |  16 |
 10 |      N_PARTICIPANT |  Total number of participants | N |   3 |
 11 |              NOTES |                         Notes | C | 276 |
 12 |         WATER_TYPE |          Freshwater body type | C |   7 |
 13 |   OTHER_WATER_TYPE |    Other freshwater body type | C |  50 |
 14 |           LAND_USE |      Land use in surroundings | C |  17 |
 15 |     OTHER_LAND_USE | Other land use in surrounding | C |  84 |
 16 |          BANK_VEGE |               Bank vegetation | C |  36 |
 17 |    OTHER_BANK_VEGE |         Other bank vegetation | C |  74 |
 18 |           ON_WATER |          On the water surface | C |  30 |
 19 |      POPUT_SOURCES |     Pollution in surroundings | C |  46 |
 20 |          WaterUses |        Evidence of water uses | C |  33 |
 21 |    OTHER_WATER_USE |  Other evidence of water uses | C |  10 |
 22 |       AQUATIC_LIVE |      Evidence of aquatic life | C |  69 |
 23 | OTHER_AQUATIC_LIVE |    Other evidence of aq. life | C |  38 |
 24 |              ALGUE |                Algae presence | C |  16 |
 25 |         WATER_FLOW |      Estimated the water flow | C |   7 |
 26 |        WATER_LEVEL |         Estimated water level | C |   7 |
 27 |            NITRATE |                       Nitrate | N |   4 |
 28 |          PHOSPHATE |                     Phosphate | N |   5 |
 29 |          TURBIDITY |       Water Quality Turbidity | C |   7 |
 30 |             RESULT |                        Result | N |   3 |
 31 |        WATER_COLOR |        Estimated water colour | C |  10 |
 32 |  OTHER_WATER_COLOR |  Other estimated water colour | C |  43 |
```

Internal REL5 format to expose a MiraMon topologically structured file as a SOS offering.

```
[VERSIO]
Vers=5
SubVers=0

[GetObservation]
GetObsservation_Vers=5
GetOBservation_SubVers=0
Fitxer=MeetMeeMechelenT.rel
CampDataHoraFenomen=time_last
CampNomSensor=street_nam
```

Final representation as XML O&M of the same structure:


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<sos:GetObservationResponse xmlns:sos="http://www.opengis.net/sos/2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink=
"http://www.w3.org/1999/xlink" xmlns:om="http://www.opengis.net/om/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:swe=
"http://www.opengis.net/swe/2.0">
    <sos:observationData>
        <om:OM_Observation gml:id="vatten-fokus_2_1">
            <om:type xlink:href=
"http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_ComplexObservation
"/>
            <om:procedure xlink:href="http://www.opengis.uab.cat/vatten-
fokus/procedure/22655"/>
            <om:observedProperty xlink:href=
"http://www.opengis.uab.cat/vatten-fokus/observedProperty"/>
            <om:featureOfInterest xlink:href=
"http://www.opengis.uab.cat/vatten-fokus/featureOfInterest/2"/>
            <om:result xsi:type="swe:DataRecordPropertyType">
                <swe:DataRecord>
                    <swe:field name="SAMPLE_ID">
                        <swe:Quantity definition=
"http://www.opengis.uab.cat/vatten-fokus/variable/SAMPLE_ID">
                            <swe:uom/>
                            <swe:value>45821</swe:value>
                        </swe:Quantity>
                    </swe:field>
                    <swe:field name="CREA_DATE">
                        <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Creation_Date">
                            <swe:value>07/12/2018 17:23</swe:value>
```

```
                        </swe:Text>
                    </swe:field>
                    <swe:field name="CHAN_DATE">
                        <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Modification_date">
                            <swe:value>07/12/2018 17:23</swe:value>
                        </swe:Text>
                    </swe:field>
                    <swe:field name="SAMPLEDATE">
                        <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Sample_date">
                            <swe:value>07/12/2018 15:00</swe:value>
                        </swe:Text>
                    </swe:field>
                    <swe:field name="GROUP_ID">
                        <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Group_ID">
                            <swe:value>Dunkern, Group ID: 38438</swe:value>
                        </swe:Text>
                    </swe:field>
                    <swe:field name="SITE_NAME">
                        <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Site_name">
                            <swe:value>Dunkershall. V¤gtrumma uppst¤
ms.</swe:value>
                        </swe:Text>
                    </swe:field>
                    <swe:field name="Sample_date_time">
                        <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Sample_date/time">
                            <swe:value>07/12/2018 15:00</swe:value>
                        </swe:Text>
                    </swe:field>
                    <swe:field name="N_PARTICIPANT">
                        <swe:Quantity definition=
"http://www.opengis.uab.cat/vatten-fokus/variable/N_PARTICIPANT">
                            <swe:uom/>
                            <swe:value>1</swe:value>
                        </swe:Quantity>
                    </swe:field>
                    <swe:field name="NOTES">
                        <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Notes">
                            <swe:value>+2 grader C.</swe:value>
                        </swe:Text>
                    </swe:field>
                    <swe:field name="WATER_TYPE">
```

```
                    <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Freshwater_body_type">
                        <swe:value>Other</swe:value>
                    </swe:Text>
                </swe:field>
                <swe:field name="OTHER_WATER_TYPE">
                    <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Other_freshwater_body_type">
                        <swe:value>Dike</swe:value>
                    </swe:Text>
                </swe:field>
                <swe:field name="LAND_USE">
                    <swe:Text definition=
"http://www.opengis.uab.cat/vatten-
fokus/field/Land_use_in_the_immediate_surroundings">
                        <swe:value>Agriculture</swe:value>
                    </swe:Text>
                </swe:field>
                <swe:field name="OTHER_LAND_USE">
                    <swe:Text definition=
"http://www.opengis.uab.cat/vatten-
fokus/field/Other_the_land_use_in_the_immediate_surroundings">
                        <swe:value></swe:value>
                    </swe:Text>
                </swe:field>
                <swe:field name="BANK_VEGE">
                    <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Bank_vegetation">
                        <swe:value>Grass</swe:value>
                    </swe:Text>
                </swe:field>
                <swe:field name="OTHER_BANK_VEGE">
                    <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Other_bank_vegetation">
                        <swe:value></swe:value>
                    </swe:Text>
                </swe:field>
                <swe:field name="ON_WATER">
                    <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/On_the_water_surface">
                        <swe:value>None</swe:value>
                    </swe:Text>
                </swe:field>
                <swe:field name="POPUT_SOURCES">
                    <swe:Text definition=
"http://www.opengis.uab.cat/vatten-
fokus/field/Pollution_sources_in_the_immediate_surroundings">
```

```
                    <swe:value>Other</swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="WaterUses">
                <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Evidence_of_water_uses">
                    <swe:value></swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="OTHER_WATER_USE">
                <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Other_evidence_of_water_uses">
                    <swe:value></swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="AQUATIC_LIVE">
                <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Evidence_of_aquatic_life">
                    <swe:value></swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="OTHER_AQUATIC_LIVE">
                <swe:Text definition=
"http://www.opengis.uab.cat/vatten-
fokus/field/Other_evidence_of_aquatic_life">
                    <swe:value></swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="ALGUE">
                <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Algae_presence">
                    <swe:value>No algae</swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="WATER_FLOW">
                <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Estimated_the_water_flow">
                    <swe:value>Surging</swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="WATER_LEVEL">
                <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Estimated_water_level">
                    <swe:value>Average</swe:value>
                </swe:Text>
            </swe:field>
            <swe:field name="NITRATE">
```

```
                        <swe:Quantity definition=
"http://www.opengis.uab.cat/vatten-fokus/variable/NITRATE">
                            <swe:uom/>
                            <swe:value>1.50</swe:value>
                        </swe:Quantity>
                    </swe:field>
                    <swe:field name="PHOSPHATE">
                        <swe:Quantity definition=
"http://www.opengis.uab.cat/vatten-fokus/variable/PHOSPHATE">
                            <swe:uom/>
                            <swe:value>0.075</swe:value>
                        </swe:Quantity>
                    </swe:field>
                    <swe:field name="TURBIDITY">
                        <swe:Text definition=
"http://www.opengis.uab.cat/vatten-
fokus/field/Water_Quality_Secchi_Tube_(Turbidity).">
                            <swe:value>&lt;14</swe:value>
                        </swe:Text>
                    </swe:field>
                    <swe:field name="RESULT">
                        <swe:Quantity definition=
"http://www.opengis.uab.cat/vatten-fokus/variable/RESULT">
                            <swe:uom/>
                            <swe:value></swe:value>
                        </swe:Quantity>
                    </swe:field>
                    <swe:field name="WATER_COLOR">
                        <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Estimated_water_colour">
                            <swe:value>Colourless</swe:value>
                        </swe:Text>
                    </swe:field>
                    <swe:field name="OTHER_WATER_COLOR">
                        <swe:Text definition=
"http://www.opengis.uab.cat/vatten-fokus/field/Other_estimated_water_colour">
                            <swe:value></swe:value>
                        </swe:Text>
                    </swe:field>
                </swe:DataRecord>
            </om:result>
        </om:OM_Observation>
    </sos:observationData>
...
</sos:GetObservationResponse>
```

7.4. GROW SOS

The GROW SOS service is tightly integrated into the GROW platform based around Hydrologic's existing Hydronet 4 platform. The SOS 2.0 service runs concurrently with the GROW standards API in the HydroNET GROW Server.

The service implements two .net packages that disseminate GROW data to the SOS 2.0 standard within the GROW instance of HydroNET 4 Server. A SOS package covers the mapping of SOS 2.0 requests to GROW requests and the mapping of GROW data structures to SOS 2.0 standards. A second package Ogc.Wrapper.Entities contains the SOS 2.0 entity definitions.

The Base URL of the GROW SOS 2.0 service: http://grow-beta-api.hydronet.com/api/service/sos

SOS 2.0 knows four core operations: GetCapabilities, DescribeSensor, GetObservation and GetFeatureOfInterest.

7.4.1. GetCapabilities

This operation lists all available metadata in the service and provides a detailed list of all other operations that are provided in the service itself. It provides the information you need to execute other operations within the SOS 2.0 effectively.

7.4.2. GetObservation

This operation gives the client access to observation data from sensors. What data is returned is dependent on the parameters you give as a client.

The GetObservation operation requires the following parameters:

- Procedure: The identifier of the sensor. The procedure can be found in the GetCapabilities response.
- ObservedProperty: The parameter that you want to query data for. A sensor can provide data for multiple parameters (e.g.: soil moisture, temperature). Which ObservedProperies are available for this sensor can be found in the GetCapabilities response, or the DescribeSensor response of the relevant sensor.
- TemporalFilter: The timespan for which you want to query data. Datetimes follow the ISO 8601 standard. The full timespan of data that the sensor provides can be found in the GetCapabilities or DescribeSensor (for the relevant sensor) response.

7.4.3. GetFeatureOfInterest

This operation provides information about features of interest (name, description, coordinates etc.) of a sensor or an observation.

The GetFeatureOfInterest operation requires the following parameters:

- procedure:The identifier of the sensor. The procedure can be found in the GetCapabilities response

The SOS 2.0 service in GROW provides two response types: xml and json. The client can provide ResponseFormat in the parameters to define which response is desired. If no ResponseFormat is given the service returns xml by default.

Chapter 8. SOS clients

In this chapter we describe three SOS integrated clients tested in the Interoperability Experiment.

8.1. Grow Client

The GROW client is a demonstrator application showing the how data sources can be combined in a tabular data warehouse and off the shelf visualisation tools can be used to create rich visualisations. In the case of this demonstrator, Microsoft's PowerBi was used although commercial tools such as Tableau, Spotfire can be used. In addition, free tools such as Grafana, Rawgraphs or Apache Superset can of course be employed.

The diagram below shows a typical output using PowerBi, showing GROW sensor locations combined with Airhack sensor locations.

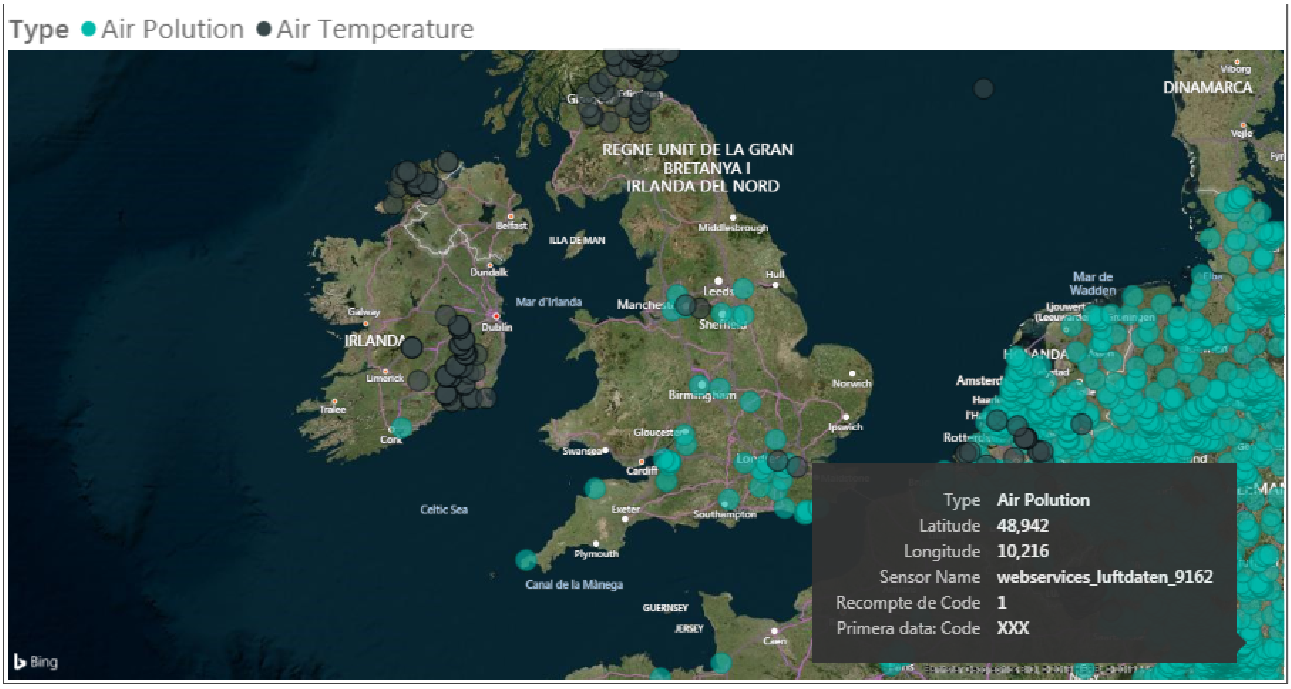


Figure 9. GROW SOS Client

In this application data is pulled from two SOS sources (GROW and AirHack) via a python script, usually overnight. This data is stored in a tabular data warehouse, in this prototype this is just flat files, however a full scale data warehouse such as Microsoft's Analysis services or a tabular database such as Cassandra could be used. Data from the warehouse is pulled nightly into Microsoft's cloud and then published to a web page that makes heavy use of javascript to provide an rich exporable interface.

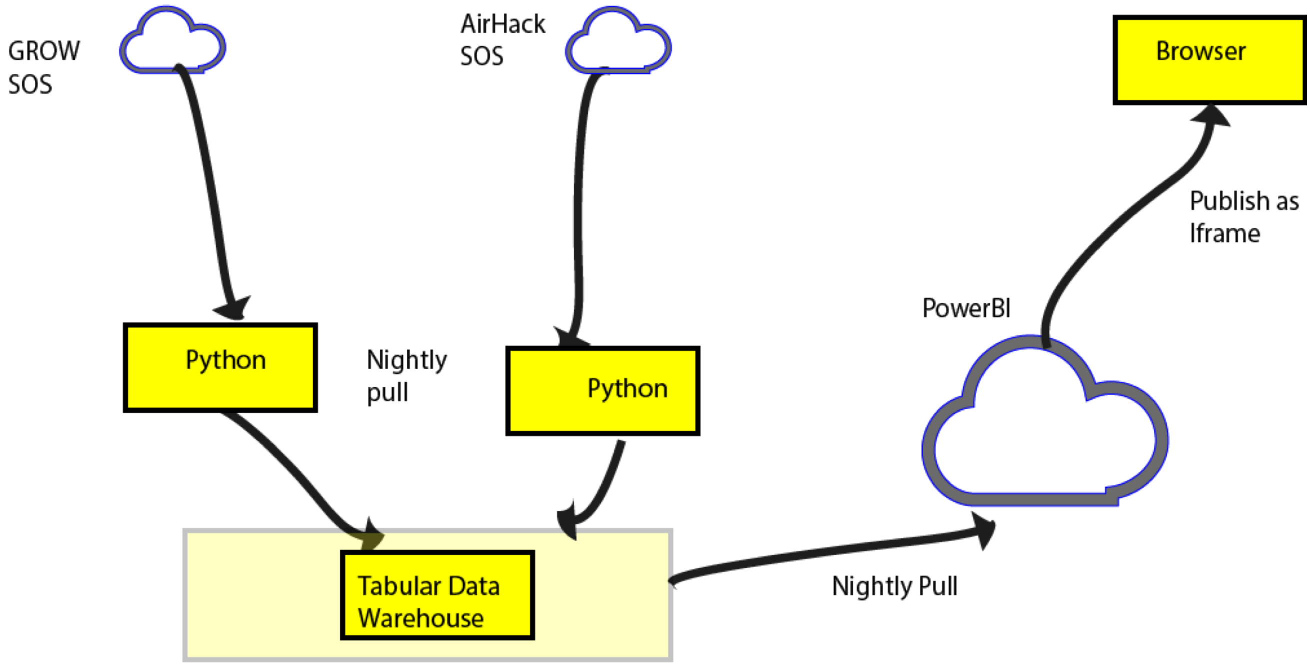


Figure 10. GROW data follow

This approach can be extended to mix different types of data into one visualisation. In the figure below gridded data (locations of sensors) is combined with time series data so that the user can explore the data more fully. In this application when a user click on a sensor, the time series data from that sensor is displayed to the user. In this case the tool takes the time series data and displays the average, minimum and maximum of the time series data.

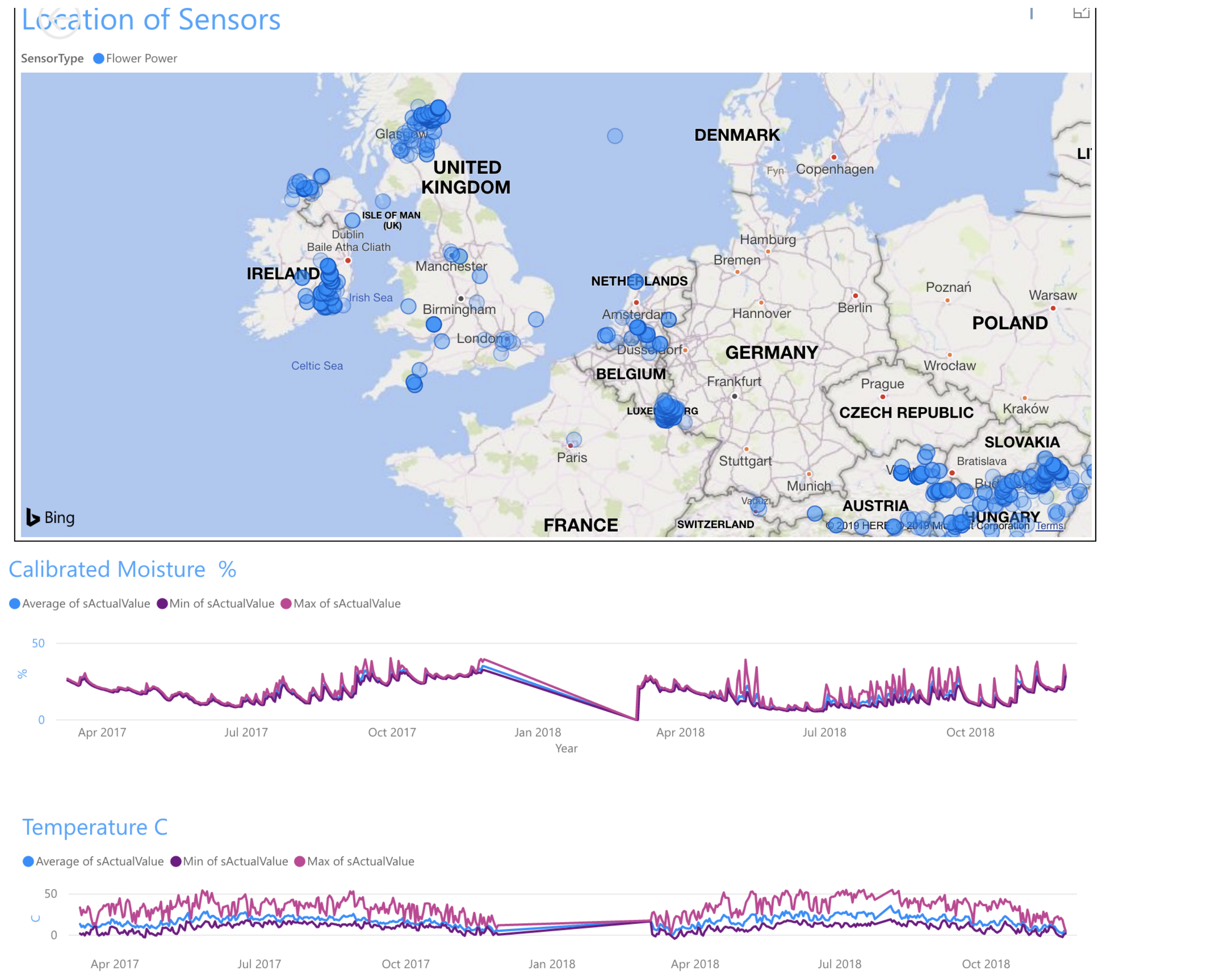


Figure 11. GROW SOS Client data

8.2. MiraMon Client

The MiraMon map Browser is a long term developing effort to create a visualization, analysis and download tool that runs in modern web browsers. Only based on HTML5 and JavaScript it uses Open Geospatial Consortium web service protocols to connect to web services and show the information to the user. The objective of the development is to assign to web browser and the JavaScript engine as much work as possible, limiting interactions to the server to the minimum possible and the transfer of information to a format that is as raw as possible. This approach can be surprising, these days were many application prefer to perform processing functionalities in the cloud and not by the client machine. Most of the time, the MiraMon map Browser is the direct responsible to create the visualization on the flight based on the raw data, allowing the user to change visualization properties, perform analysis, statistics or build time series in the client side directly. In the Data quality estimations in the client side section we will show how the same principle can be used to compute overall data set quality can be computed on the flight too.

This are the main functionalities and standards used to achieve them:

- Raster visualization and query by location is possible using OGC Web Map Service and Web Map Tile Service.

- Raster analytics is possible by using the OGC Web Map Service in an special way that transmits binary arrays of values (raw or RLE encoded) instead of pictorial representations. Pixel based analysis is performed directly in JavaScript in the client side and visualized on the flight.
- Vector visualization and query by location is possible using OGC Web Feature Service and Sensor Observation Service. The client accepts both XML an JSON formats.
- Data download is limited to the use of Web Coverage Service v.1.0.

Several layers of data coming from different servers and using different protocols can be overlaid simultaneously. Some layers represent data from a single dataset while other can be a virtual datasets computed on the flight for each zoom and pan, and created by combining data form more than one dataset and server.

Both WFS and SOS visualization are currently limited to points that are represented in a HTML5 canvas as icons, circles and texts and could be easily extended to lines and polygons in the near future. This functionalities where particularly useful to show occasional observations done by citizens in different places. We where able to show together Ground Truth 2.0 observations with HackAir observations using the SOS protocol directly in XML and in JSON. Some of the visualization functionalities were actually improved during the Interoperability Experiment such as the capability to condition the color of a circle by an attribute (or an observed *result*) of the feature. This way it was possible to represent the level of concentration of a pollutant as a colored circle using color code that was represented in the legend.

Representing the positions of the observations will only require the GetFeatureOfInterest operation except if the visualization of the feature should depend on the *result* of the observation. In the later case, a GetObservation is performed and all the information on the features is loaded in the client making query bay location non dependent on the server anymore.

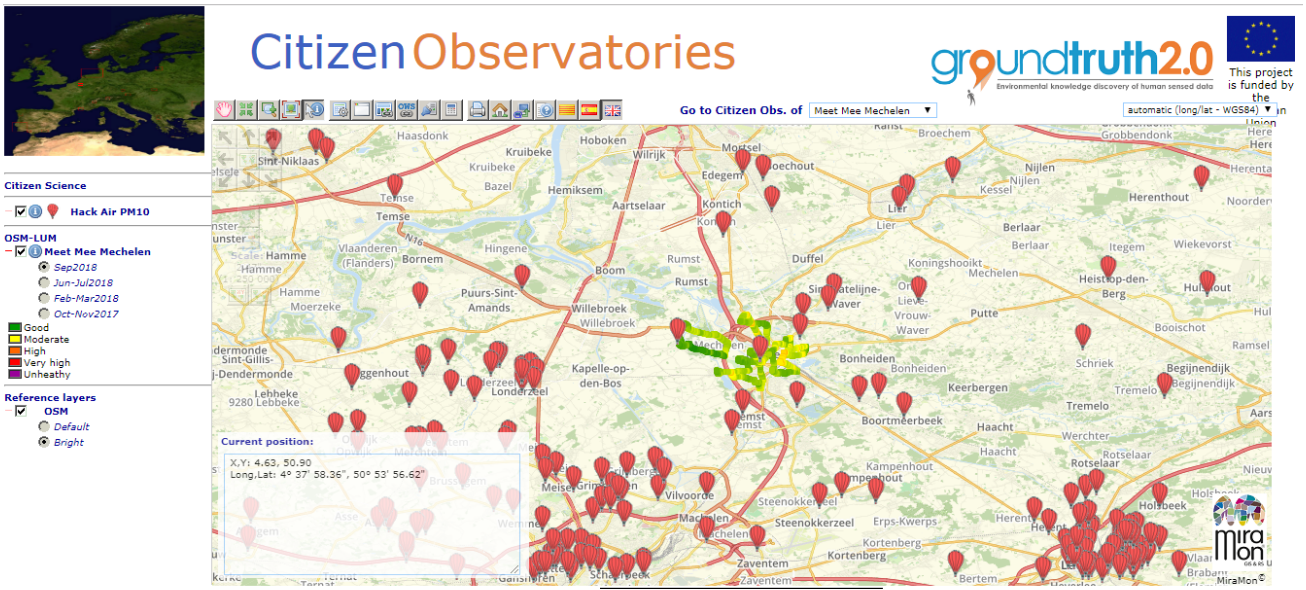


Figure 12. MiraMon SOS Client

8.3. 52°North Helgoland Sensor Web Viewer

The Helgoland Sensor Web Viewer developed by 52°North is an open source visualisation tool for different kinds of Sensor Web data. It allows to explore available observation data sets and to visualise the actual data (e.g. time series) as diagrams.

Figure 1 shows the map view of the Helgoland Sensor Web Viewer. In this case, a sample data set of the hackAIR project is explored. The map view shows the locations at which air quality sensors are located.

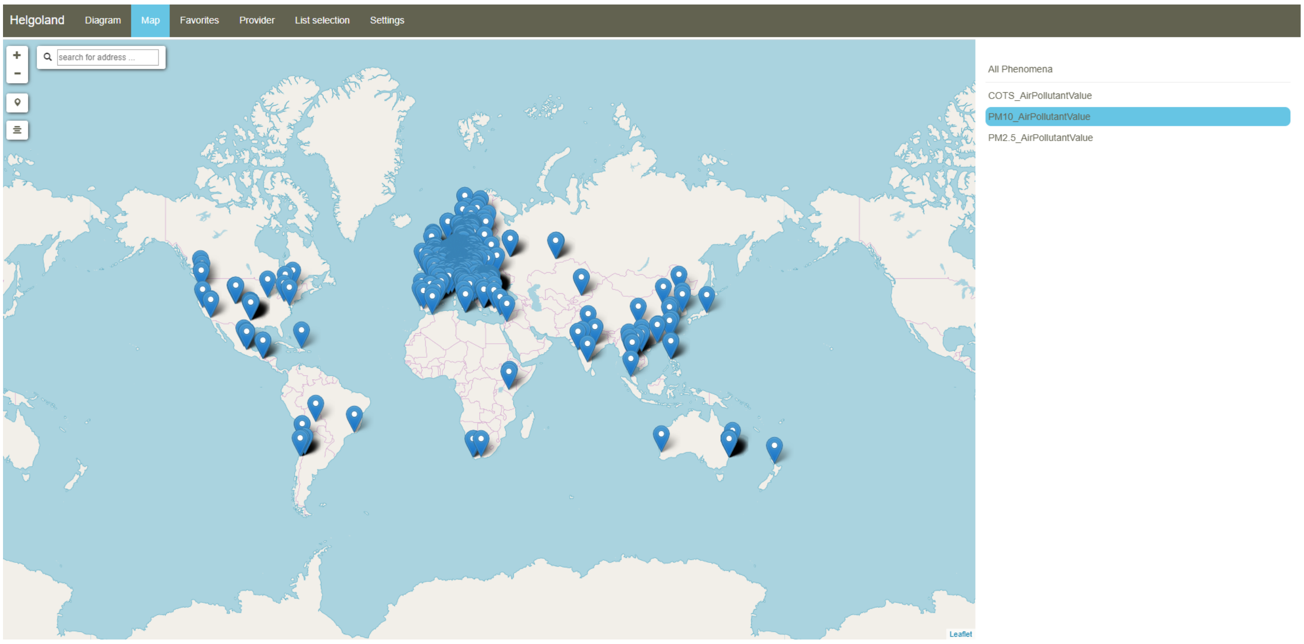


Figure 13. 52°North Helgoland Sensor Web Viewer - Map View

After selecting a specific measurement location, the data can be visualised as a diagram (see Figure 2). It is possible to combine data from multiple sensors, multiple observed properties and even from different providers into a single diagram.

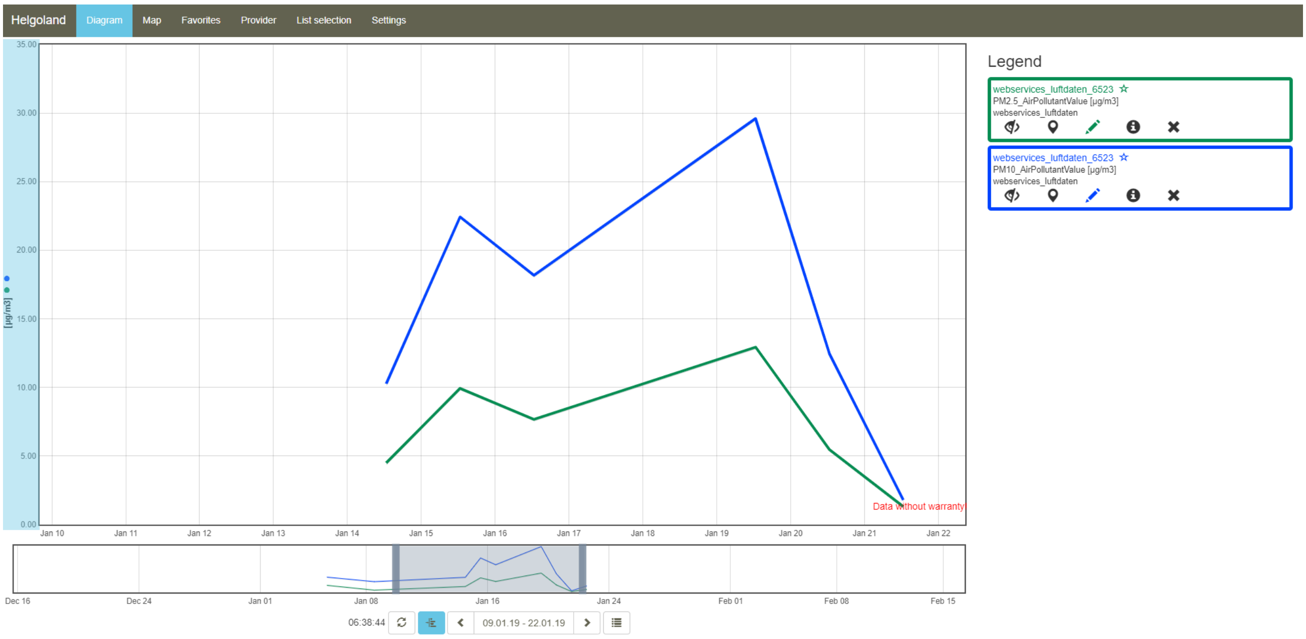


Figure 14. 52°North Helgoland Sensor Web Viewer - Diagram View

8.4. SOS Technology Integration Experiments

During the IE a set of Technology Integration Experiments (TIE) were conducted. In the client and server architecture a TIE is test that combines a server with a client and demonstrates that communication between client and server is possible and the user (operating the client) is able to see or get some data. The following table summarizes the tests conducted and the degree o success achieved.

Servers	Data	MiraMon Client	Helgoland Client	Grow Client
MiraMon	Ground Truth 2.0	Yes		
Helgoland	HackAir		Yes	
istSOS	HackAir	Yes		Yes
Grow	Grow			Yes

Table 2. Technology Integration Experiment results

Chapter 9. Data quality estimations in the client side

One of the main concerns in using and adopting citizen science based data is the quality of observations. Citizen Observatories (and, by extension, Citizen Science) are particularly sensible to data quality because the number of contributors is bigger and more heterogeneous than in a traditional data survey campaign. An additional difficulty is that active Citizen Observatories are receiving continuous inputs and updates from citizens. GroundTruth 2.0 has developed a tool to document well the quality of datasets in order to increase the trust on the information collected by citizens integrated in the MiraMon Map browser.

The tool requires that data is exposed in the Web as a service using Sensor Observation Service (SOS). It presents a set of tests like positional accuracy, attribute consistency or confusion matrix that can be applied to a complete dataset or to an area the user is visualizing. Results include an overall quality indicator for the dataset.

The Ground Truth 2.0 Data Quality tool uses an interoperable approach based on QualityML that allows to parametrize the different statistics that are used to assess the quality of the data, and it focus on data quality indicators for Citizen Science datasets from the QualityML list. The quality module is encoded in JavaScript and has been made available as part of the web based MiraMon Map Browser (https://github.com/joanma747/MiraMonMapBrowser).

9.1. Quality estimation on vector data

The SOS protocol and the GetObservation operation enables a client to retrieve all the information about the results of the observations. With this data, the client can perform all sorts of analysis on the observations including to apply some quality checks. This section will discuss a pilot that was done in the GroundTruth 2.0 project that demonstrates this capability in some practical cases.

The selected cases and its implementation is based on the QualityML vocabulary. The scenario of rapidly growing geodata catalogues requires tools focused on facilitate users the choice of products. QualityML is a dictionary that contains hierarchically structured concepts to precisely define and relate quality levels: from quality classes to quality measurements. This levels are used to encode quality semantics for geospatial data by mapping them to the corresponding metadata schemas. The benefits of having encoded quality semantics, in the case of data producers, are related with improvements in their product discovery and better transmission of their characteristics. In the case of data users, they would better compare quality and uncertainty measures to take the best selection of data as well as to perform dataset intercomparison. Also it allows other components (such as visualization, discovery, or comparison tools) to be quality-aware and interoperable. On one hand, the QualityML is a profile of the ISO geospatial metadata standards (e.g. ISO 19157) providing a set of rules for precisely documenting quality measure parameters that is structured in 5 levels. On the other hand, QualityML includes semantics and vocabularies for the quality concepts. Whenever possible, it uses statistic expressions from the UncertML dictionary (http://www.uncertml.org) encoding. However it also extends UncertML to provide a list of alternative metrics that are commonly used to quantify quality beyond the uncertainty concept.

9.1.1. How data quality is presented

Datasets can have precomputed data quality indicators associated. This is part of the metadata of the datasets, but in the map browser it has a prominent place in the *quality* option in the context menu of the layer name in the legend.

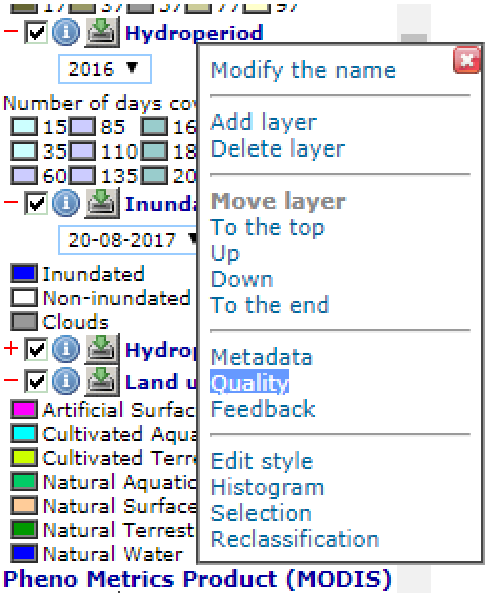


Figure 15. Data Quality context menu

Data quality indicators are presented following the QualityML model. For a quality class (in the Figure 16 "Thematic classification correctness"), there could be one or more quality measures (in the Figure 16 "Misclassification"), that are done by applying some metrics (in the Figure 16 "MeanAbsolute and StandardDeviation") over a domain (in the first entry of Figure 16 "Omission Error" over the categories "water" and "no water")

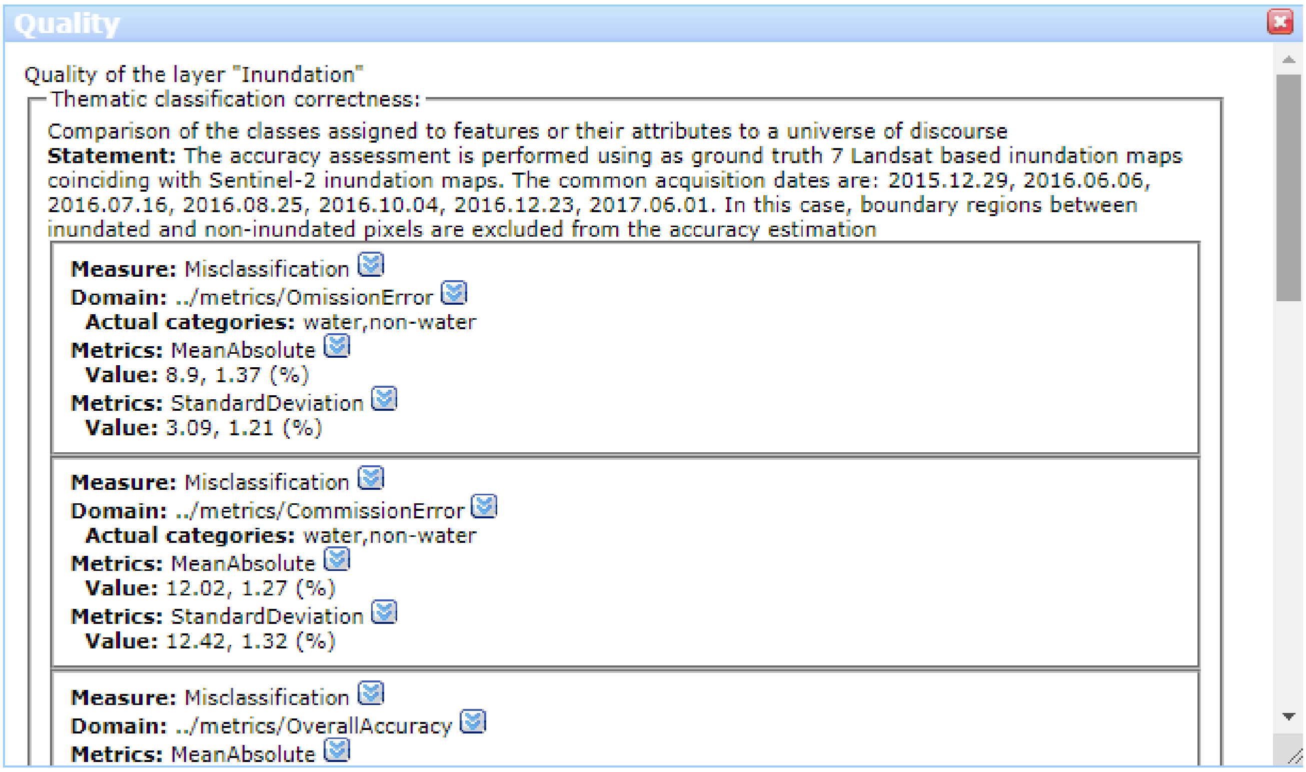


Figure 16. Data Quality indicators presented following the QualityML data model.

Every concept used here is connected to the QualityML vocabulary to know more details about it.

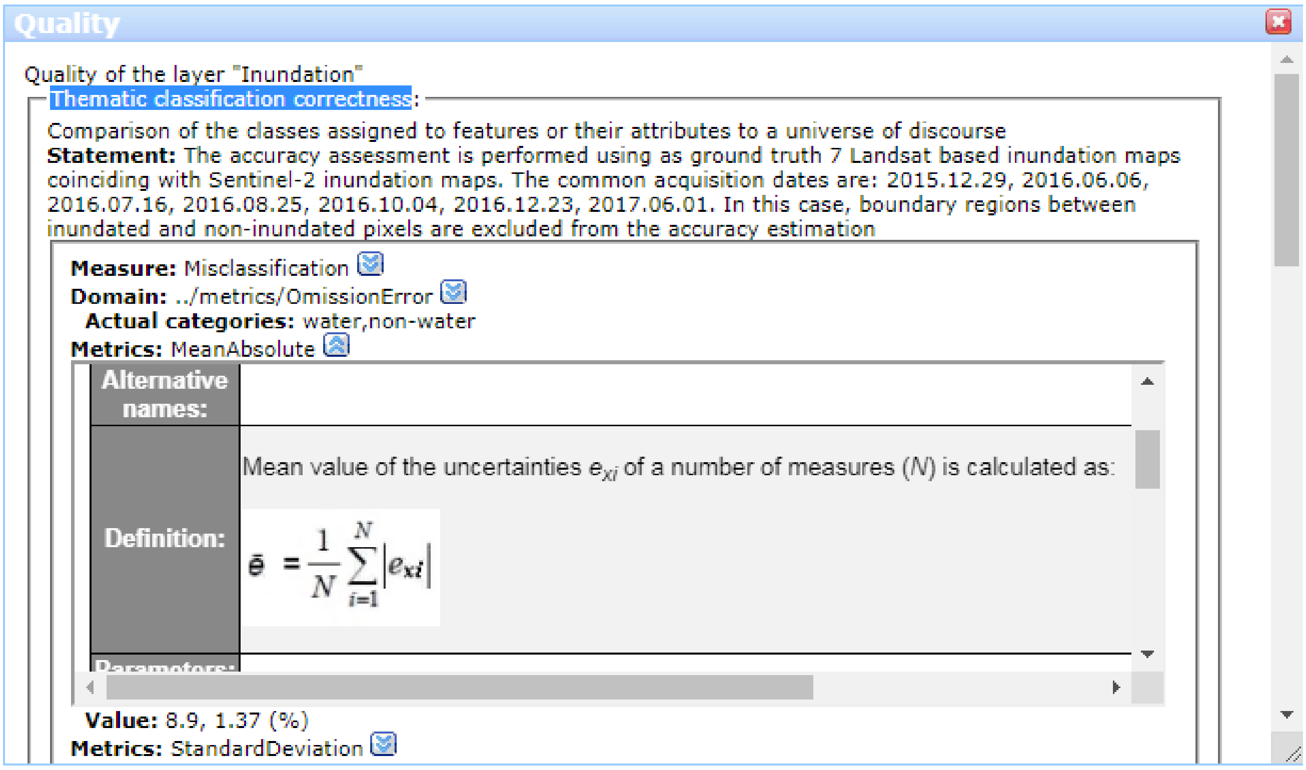


Figure 17. Connection to the QualityML dictionary.

9.1.2. How to start computing data quality

The MiraMon Map Browser described in the section SOS clients allows for computing some data quality indicators. To start the process, we should select the right option in the context menu by clicking in the layer name in the legend.

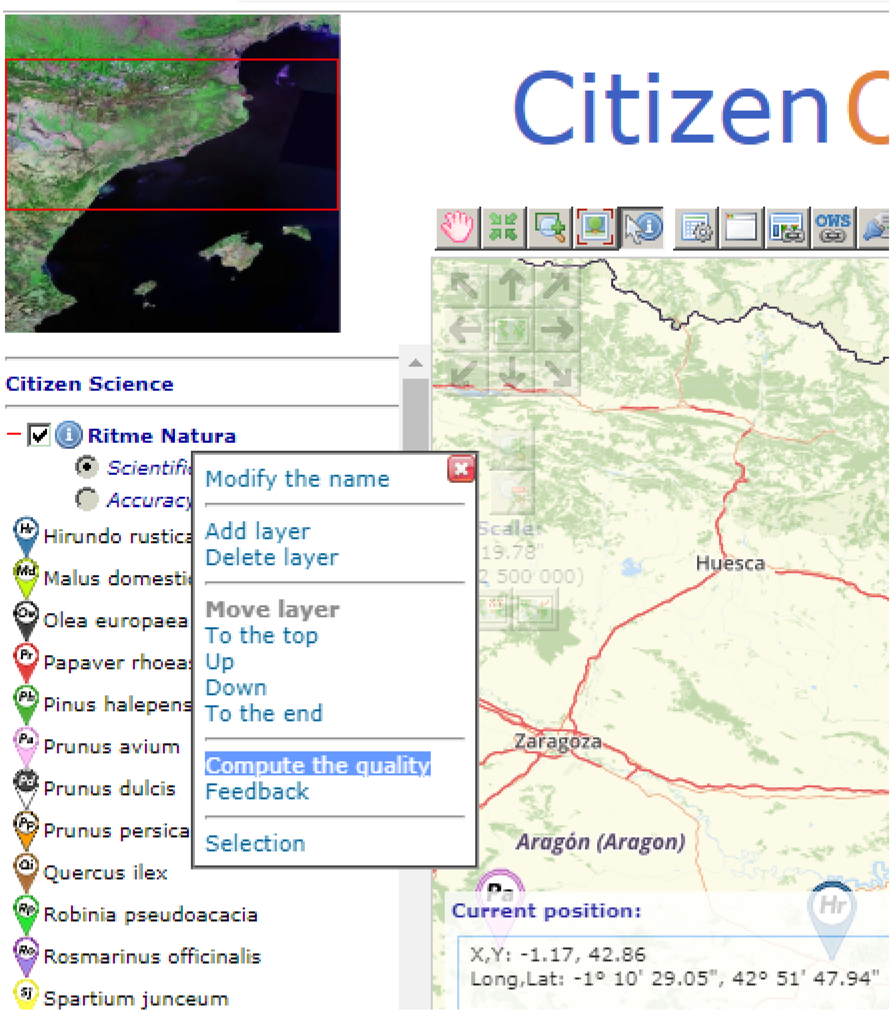


Figure 18. Data Quality compute context menu

This option opens a dialog box that offers a short list of four quality indicators that will grow with new tests.

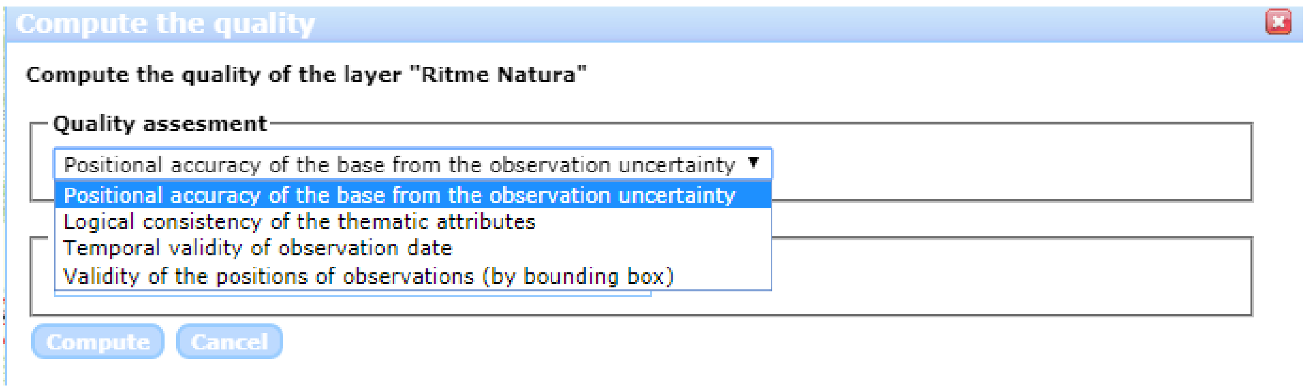


Figure 19. Data Quality indicator list

9.1.3. Case 1: Positional accuracy of the layer from observation uncertainties

Many citizen Science projects, use a mobile phone to get observations. In this process they use the location capabilities of the phone, including GPS, 3G triangulation, Wify antenna location or

IP address registration. Each of this methods has different known positional accuracies and the phone is able to estimate that ad the same time as it estimates the position. In this case we will assume that the individual observations has got a position an some estimation of the positional uncertainty and this are recorded by the service and offered as properties of the observation.

To compute this indicator we should select the property associated with the observation that contains the positional uncertainty.

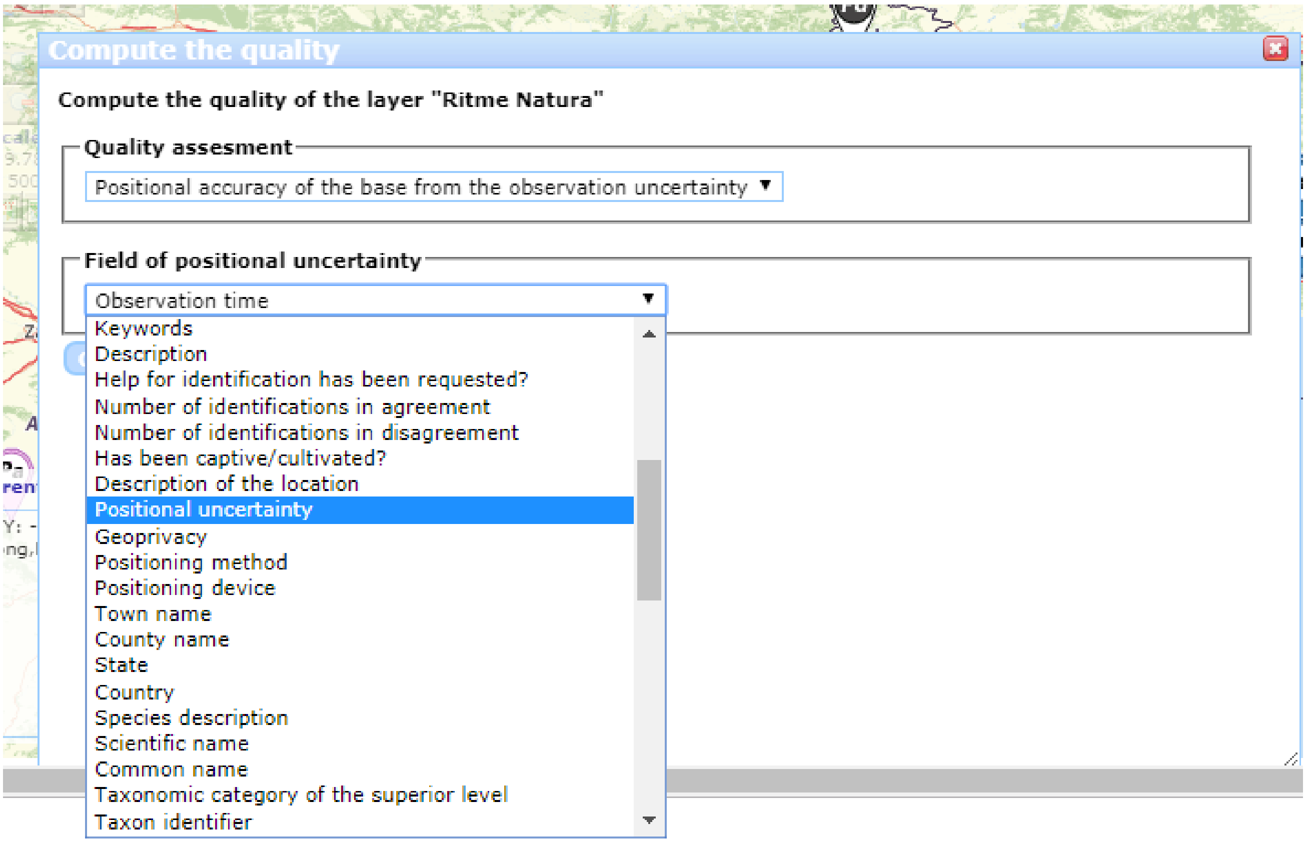


Figure 20. Selection of the positional uncertainty field

The calculated data quality parameter is not shown immediately but added to the previous recorded data quality indicators.

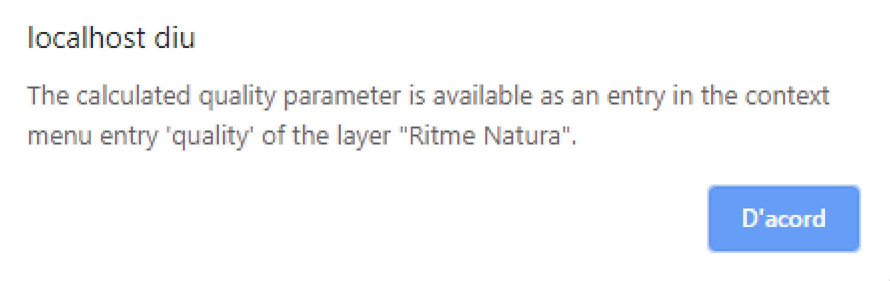


Figure 21. Calculation complete message

The result is a quality report that can be found in the `quality` option in the context menu by clicking in the layer name in the legend.

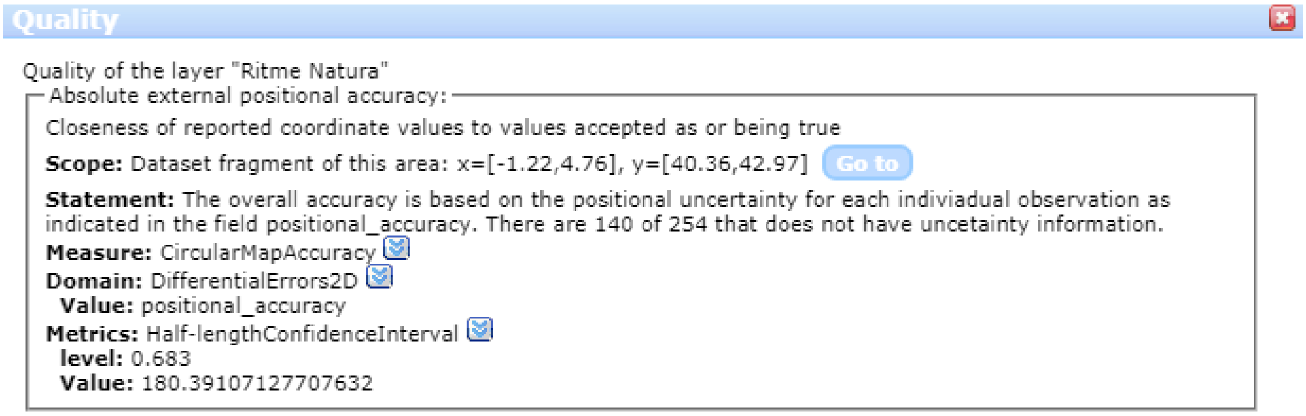


Figure 22. Calculated positional accuracy

Several things can be commented here. The first one indicates that the scope is not the full dataset but the view used for calculating the quality indicator: "Dataset fragment of this area: x=[-1.22,4.76], y=[40.36,42.97]". Secondly, the statement reports that not all observations has positional uncertainties: "There are 140 of 254 that does not have uncertainty information". The accuracy is reported as a *half-length confidence interval* with a confidence *level* of 0.683. An uncertainty of 180.39m is not particularly good indicating the heterogeneity of the methods used to calculate the positions of the observations, some of the with big uncertainties.

9.1.4. Case 2: Logical consistency of the thematic attributes

Many Citizen Science projects provide the citizens with comprehensive instructions on how to conduct some observational tasks. In some cases observations are limited to a set of possibilities for a list. In cases more complex, once selected an option in the first list, only some values are possible in a second list. Sometimes apps control user inputs preventing citizens to input a value that is not listed in the instructions, but in some cases (such as bulk input form a csv) there might be no controls and unwanted values or incompatible value combinations could end up in the database.

In the case of RitmeNatura citizen observatory, we rely on Natusfera software that is designed for biodiversity in general allowing any possible scientific name while RitmeNatura is asking for a limited set of species. Obviously, if nobody filters them, there is a chance that observations report on species not contemplated by the RitmeNatura subset.

The logical consistency test can count how many observations are not consistent with a controlled list of possibilities. To compute this indicator we should select the property (or properties) associated with the observation that are affected by a controlled list of possibilities and list the possible combinations of attributes. In this simple case, we will test if the `scientific name` is compatible with the list of possibilities described in the legend.

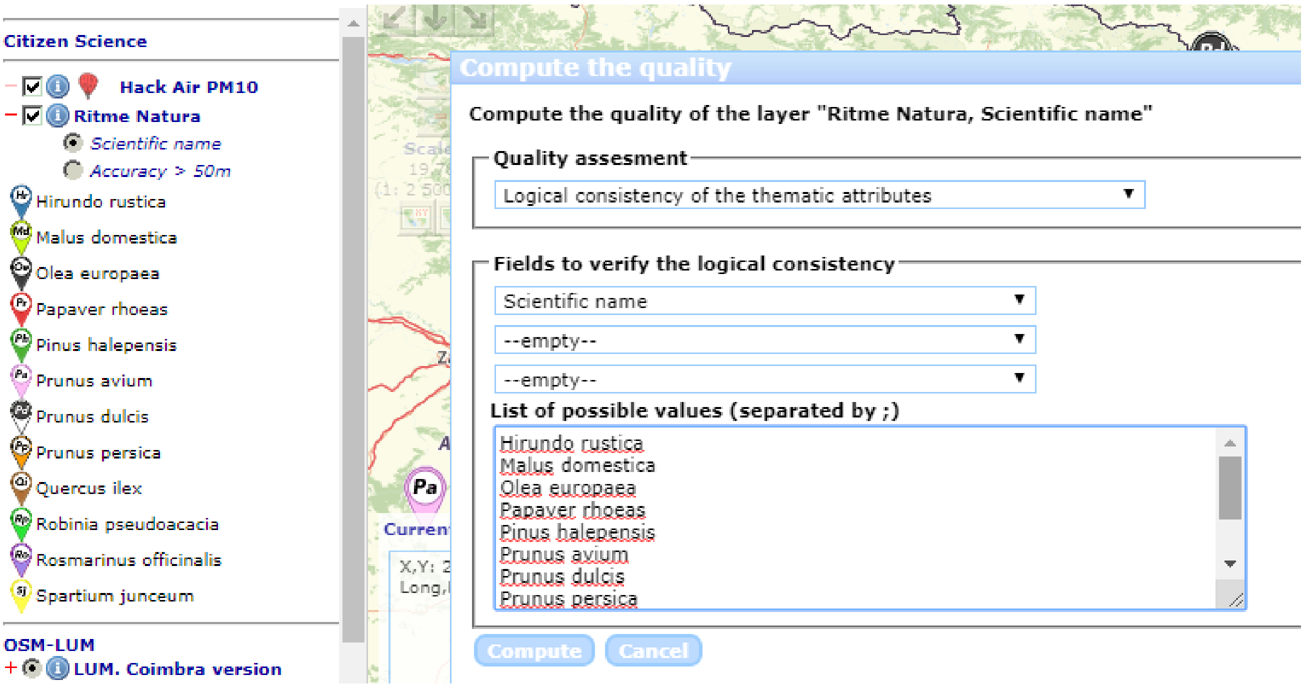


Figure 23. Computation of logical consistency

A new quality indicator will be added to the list of quality indicators related to this layer.

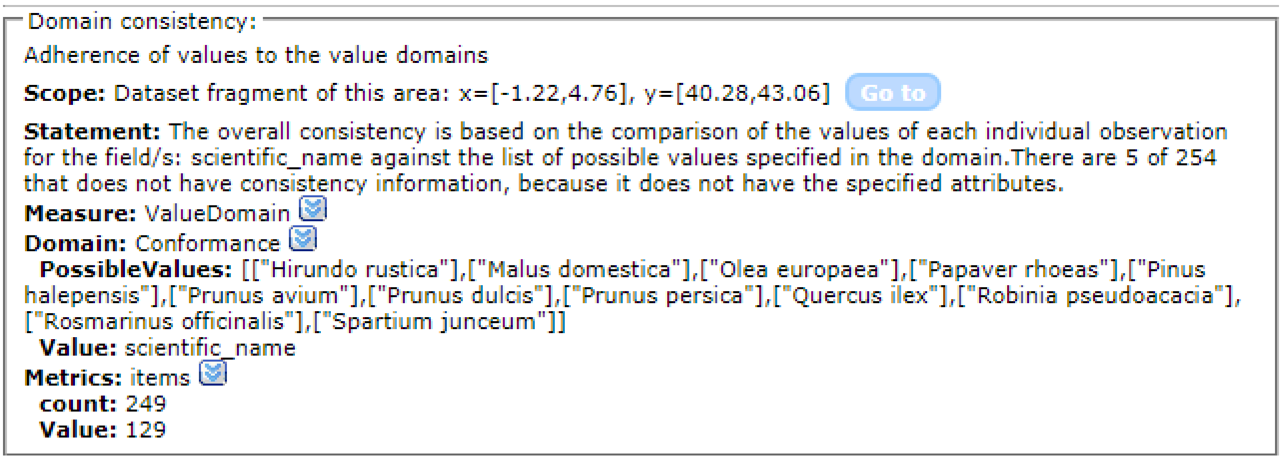


Figure 24. Domain consistency result

Only 129 of the 249 species scientific name are consistent with the legend. In addition, 5 observations have no scientific name (probably because the observer did not know it).

9.1.5. Case 3: Temporal validity of the observation date

One very simple quality control that can be performed is to check if the observations has an associated data, if the date is in the right format and if the date is in a range of plausible values.

In this example, we test if the observations where done after the year 2000 because we know there should not be observations before this date.

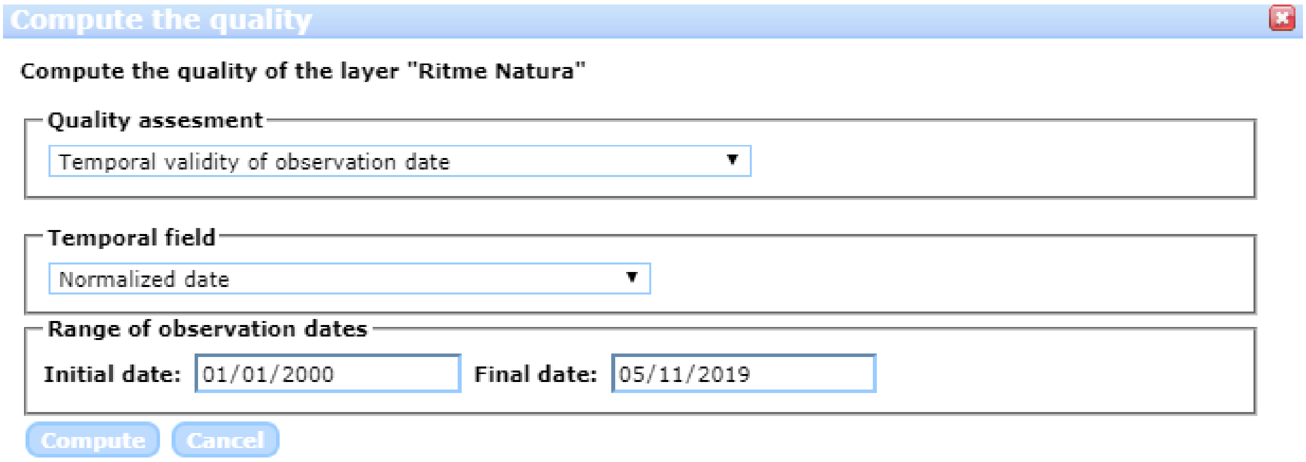


Figure 25. Computation of temporal validity

A new quality indicator will be added to the list of quality indicators related to this layer.

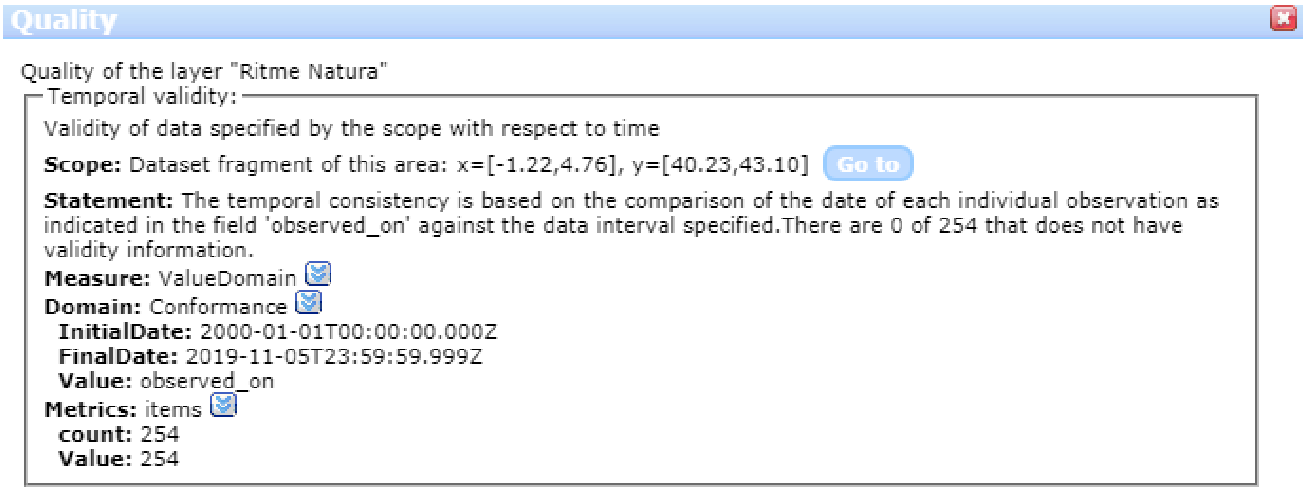


Figure 26. Temporal validity result

In this case we see that all the observations has passed the test.

9.1.6. Case 4: Validity of the positions of observations (by bounding box)

One very common mistake in data gathering projects is the presence of observations in places that does not have much sense. Tipical mistakes is swap latitude and longitude values or simply have them in the middle of the Atlantic ocean at the 0,0 position.

In this cases we are going to run a test to find how many observations are in the Catalonian bounding box.

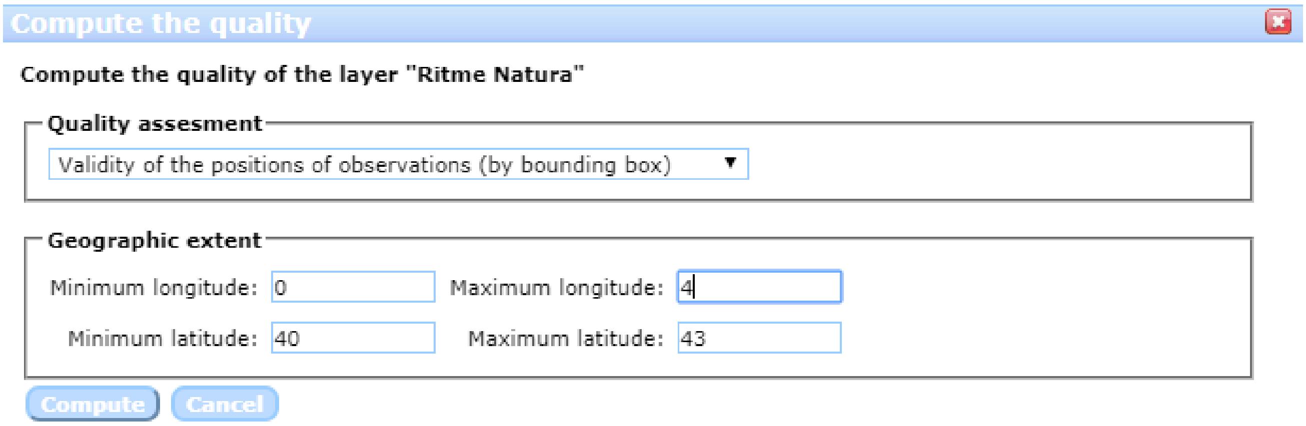


Figure 27. Computation of positional validity

A new quality indicator will be added to the list of quality indicators related to this layer.

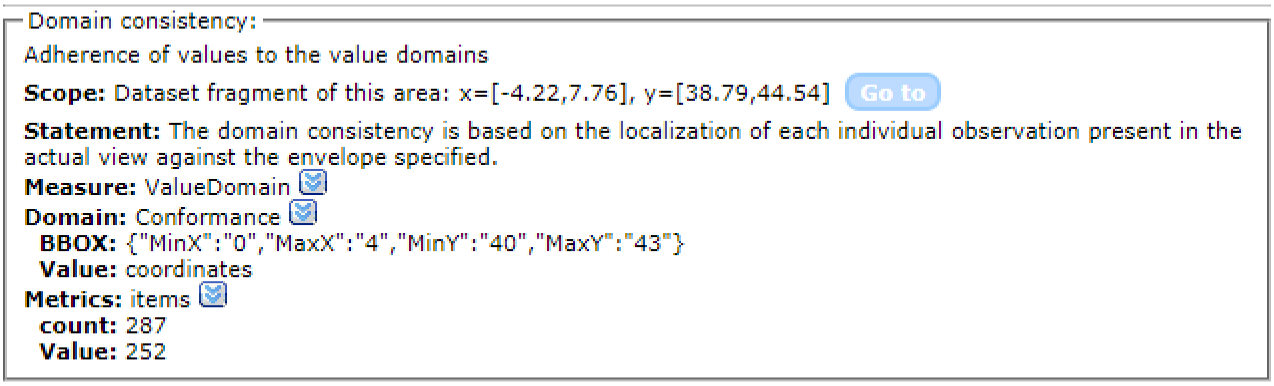


Figure 28. Positional validity result

The result identifies 35 observations in this view that are clearly outside the boundaries of Catalonia.

9.2. Quality estimation on raster data

As explained before, the WMS protocol can be used to transport binary arrays instead of pictures. During this Interoperability experiment, we have implemented a comparison functionality that can be used to compare two categorical maps with the same legend. This comparison results in a new map with all combinations of the two maps categories allowing us to discover changes in this maps.

This can be used to compare maps but also to quality control maps if we assume that one map represents the truth.

9.2.1. Confusion matrix

In this exercise we will combine one land cover map created from Open Street Map with another one created by remote sensing.

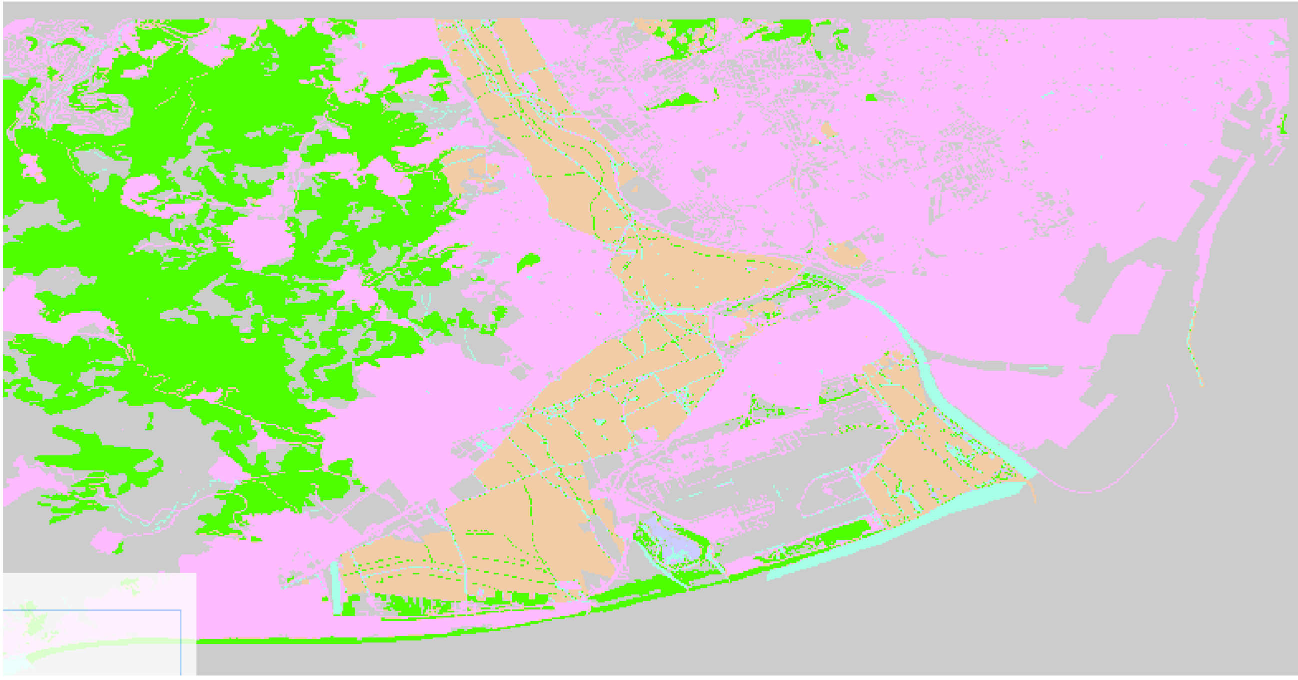


Figure 29. Open street map version of the land use map

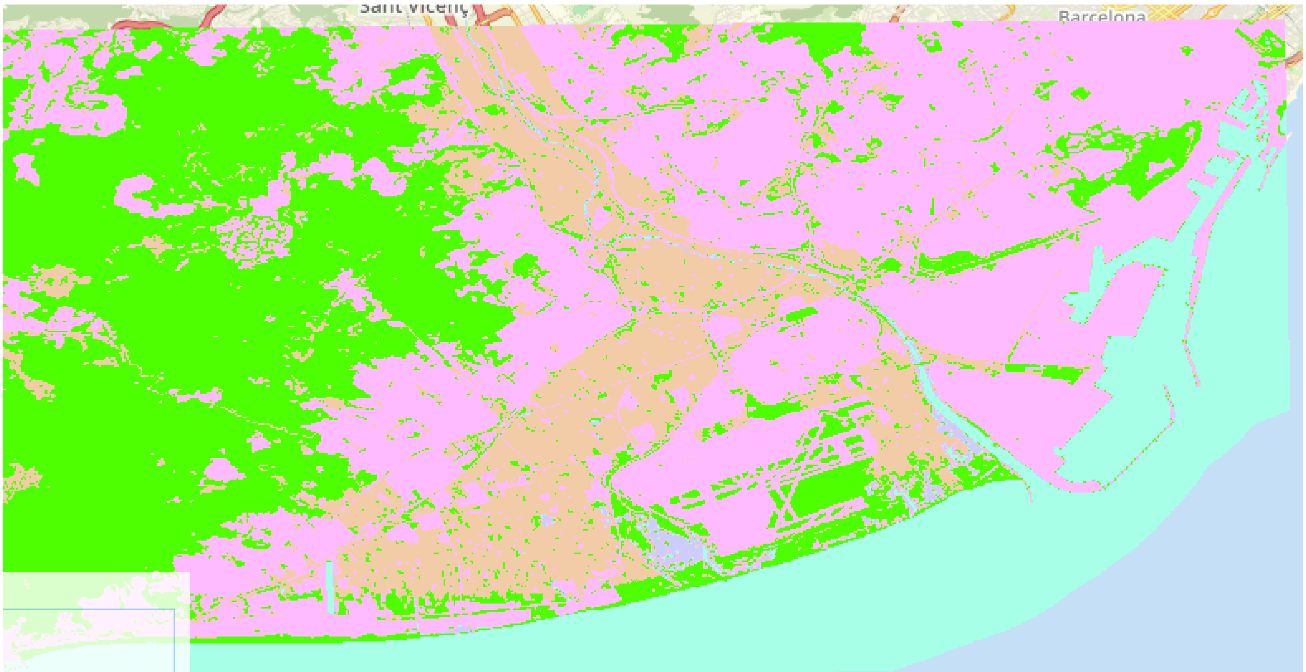


Figure 30. Remote sensing version of the land use map

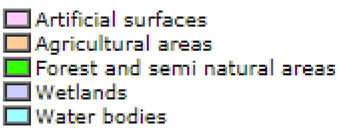


Figure 31. Land use map legend

The process of creating a confusion matrix starts by requesting the combination of both maps in a single layer those pixels will contain classes that are all possible permutations of the legend. In the Figure 32 the Coimbra version is the one generated from OSM while the CREAF-RS version is the one created by remote sensing. The result of the combination is shown in Figure 32. In principle even the number of combinations is 25, there are only 5 many colors present, corresponding to the classes that are the same in both maps.

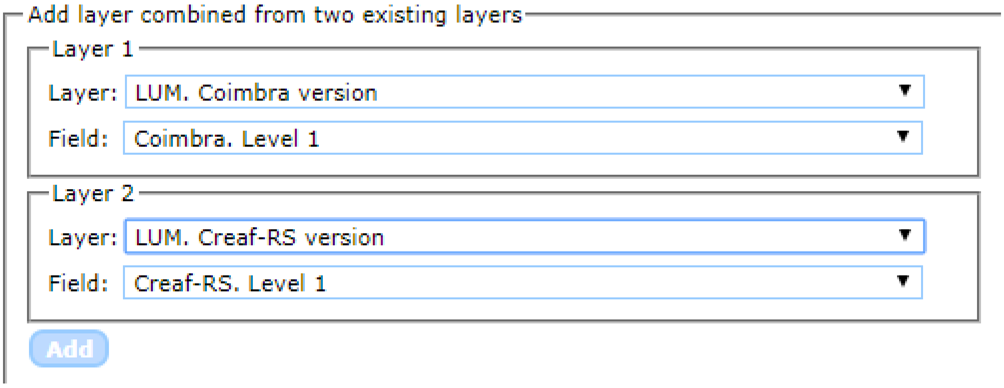


Figure 32. Request for a layer combination of both land use maps

Now we can request the confusion matrix as a statistical summary of the combination by selecting the option in the context menu.

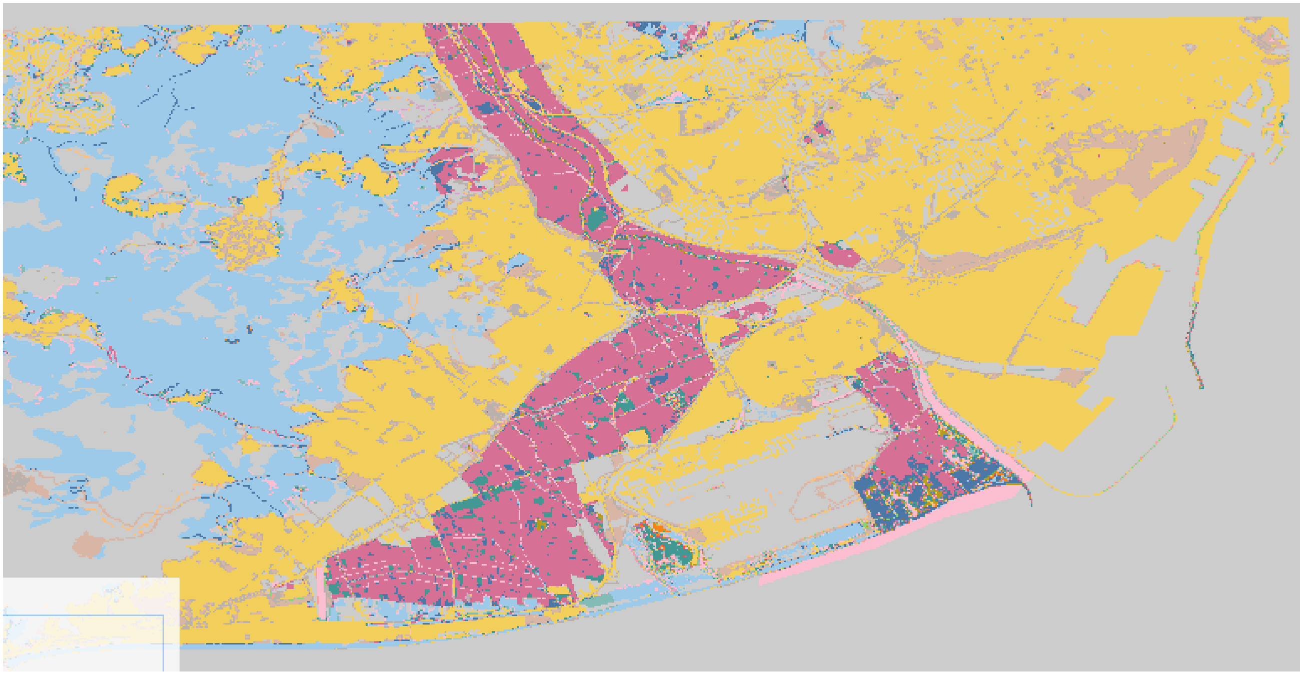


Figure 33. Layer combination of both land use maps

The diagonal values of the matrix (represented in green) correspond to the pixels that have the same value in both maps. The non-diagonal values are the pixels that has different classes in both maps. We can also see some information about the most similar classes (*artificial surfaces* and *forest and semi natural areas*) as well as the Kappa coefficient that is 0.81 (the closes to 1 the better).

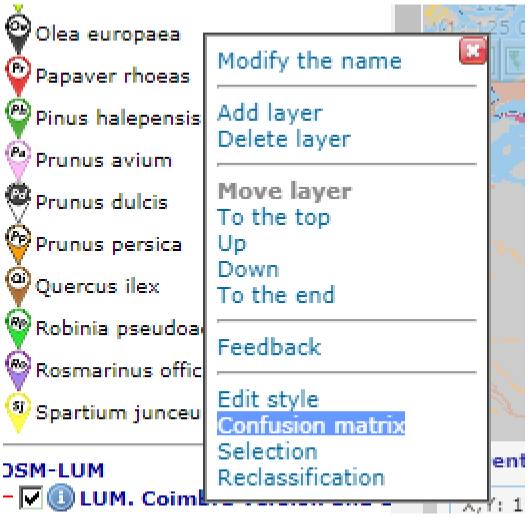
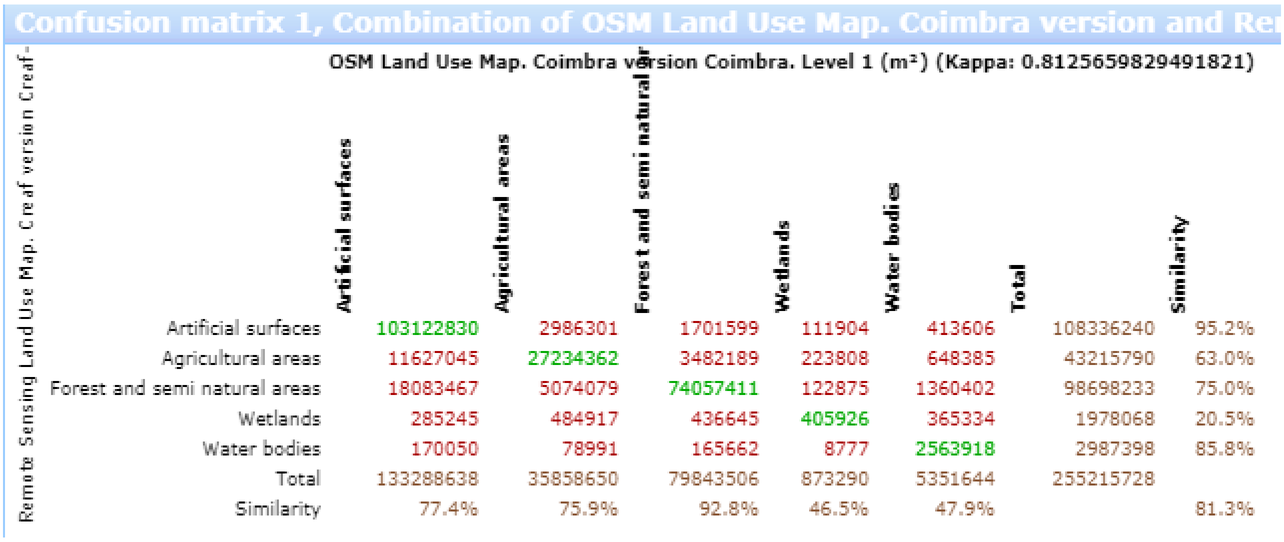


Figure 34. Request for the confusion matrix

A manual exploration of the dataset allow to discover a big purple area the is *artificial surfaces* from the OSM and *forest and semi natural areas* from the RS map.



Confusion matrix 1, Combination of OSM Land Use Map. Coimbra version and Re

OSM Land Use Map. Coimbra version Coimbra. Level 1 (m²) (Kappa: 0.8125659829491821)

Remote Sensing Land Use Map. Creaf version Creaf-

	Artificial surfaces	Agricultural areas	Forest and semi natural	Wetlands	Water bodies	Total	Similarity
Artificial surfaces	103122830	2986301	1701599	111904	413606	108336240	95.2%
Agricultural areas	11627045	27234362	3482189	223808	648385	43215790	63.0%
Forest and semi natural areas	18083467	5074079	74057411	122875	1360402	98698233	75.0%
Wetlands	285245	484917	436645	405926	365334	1978068	20.5%
Water bodies	170050	78991	165662	8777	2563918	2987398	85.8%
Total	133288638	35858650	79843506	873290	5351644	255215728	
Similarity	77.4%	75.9%	92.8%	46.5%	47.9%		81.3%

Figure 35. Request for the confusion matrix result

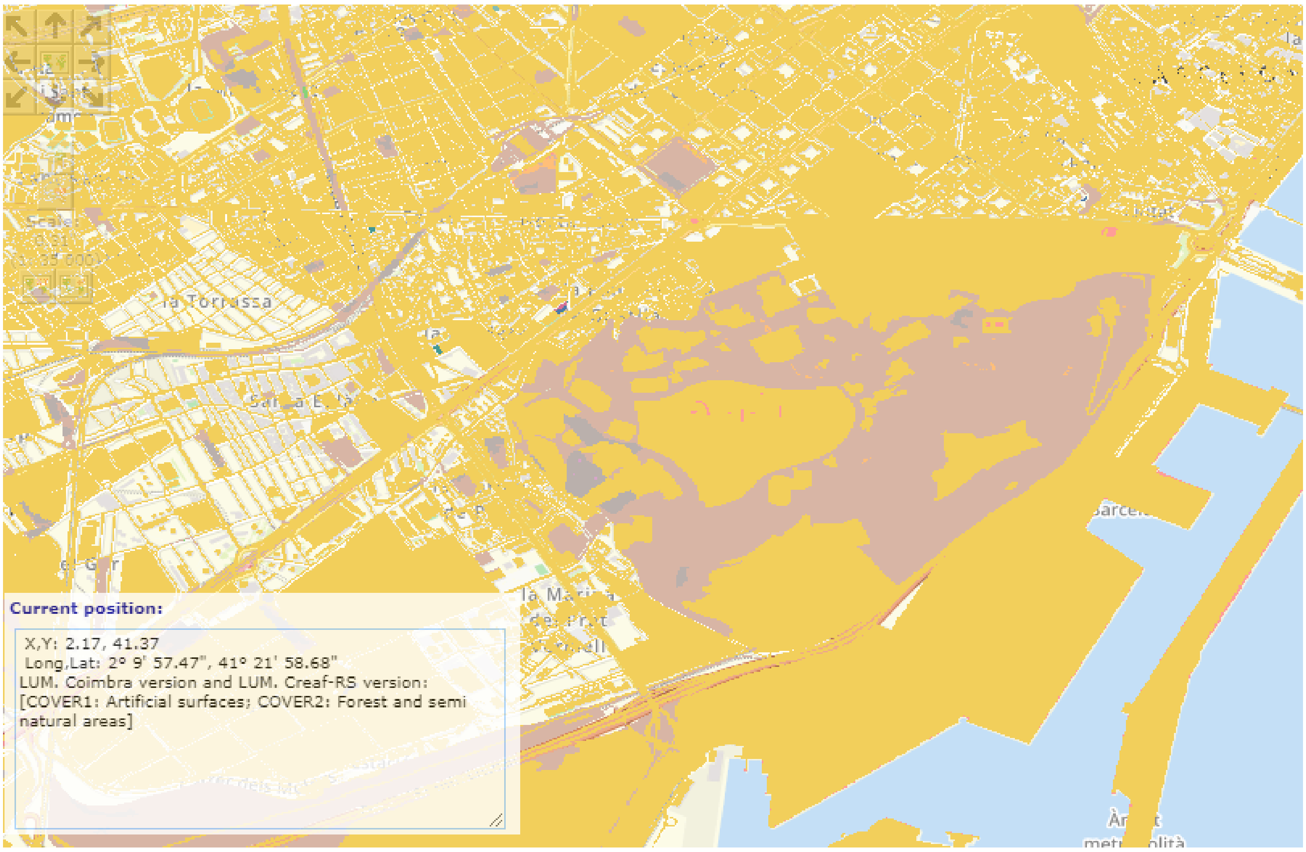


Figure 36. Zoom to an area of discrepancies

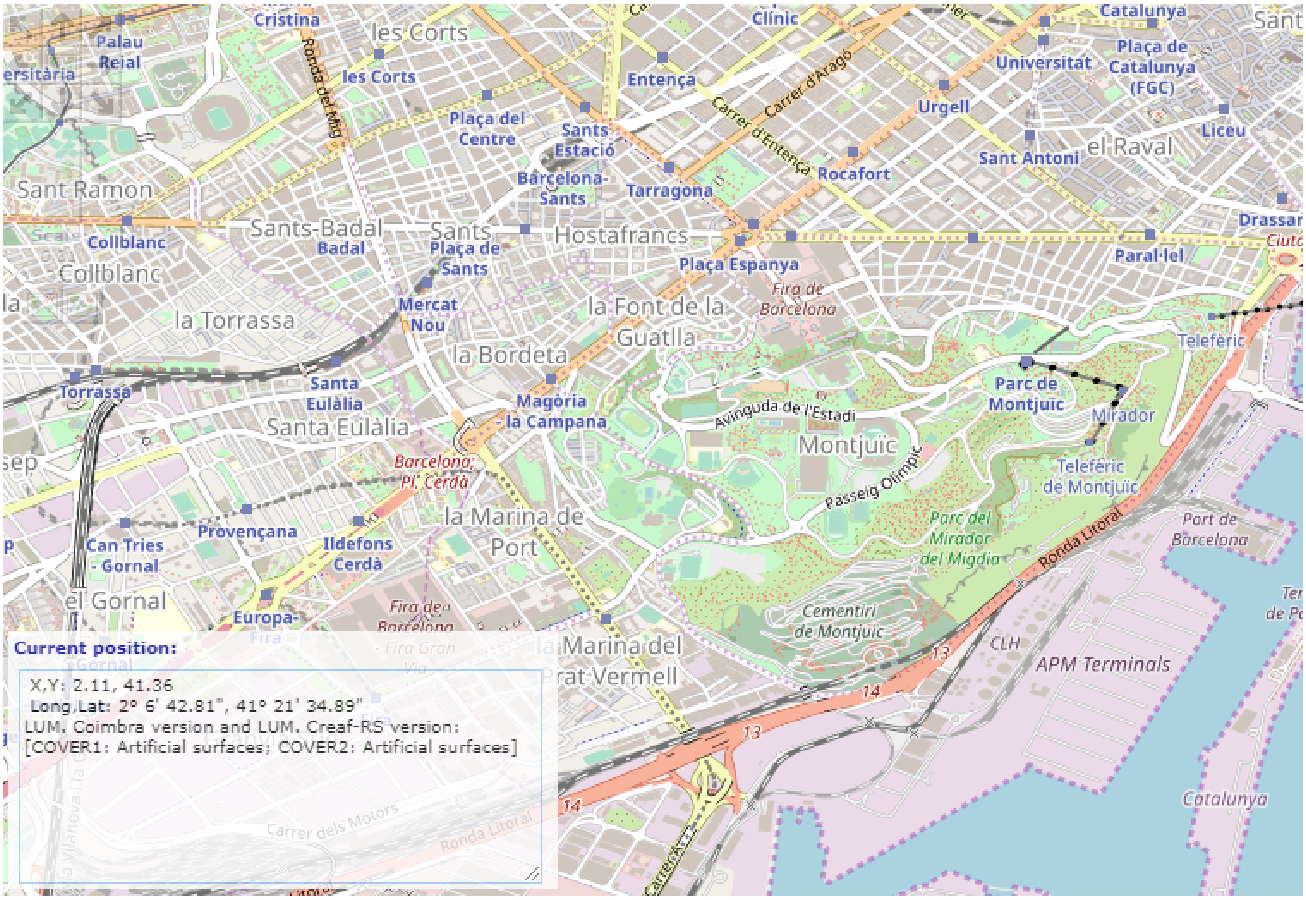


Figure 37. Reason for the discrepancies

The discrepancy makes sense. A big park in the city is identified as artificial in the OSM version that is more focus on land use while is seen as a forest area from remote sensing due to its green land cover.

9.3. Future work

There are some points the authors of this chapter believe it is worth to develop or explore.

- In the implementation of the confusion matrix there is no connection to the QualityML. It should be done.
- Highlight the observations that were detected as less accurate could be an interesting feature to have.
- We would like to be able to share the quality assessments with other users. One possibility is usinge the OGC Geospatial user feedback to report data quality assessments and share them with other users. Saving the quality report in the NiMMbus database (www.opengis.uab.cat/nimmbus) implemented in the NextGEOSS project will allow that.
- The computations done in the MiraMon map browser are just an small subset of the QualityML vocabulary. We would like to extend the implementation to cover a better range of possibilities.
- QualityML is a vocabulary for data quality. The OGC definitions server presented in Definition Server is a generic tool to share vocabularies. Translating QualityML into a format that can be ingested by the Definitions Server should be a priority of the next interoperability experiment.

Chapter 10. Definition Server

The definition server is service that allows for storing, querying and linking definitions. The definition server provides a common way to resolve terms published by the OGC to get details of definitions (instead of downloading large complex documents in varying formats).

Currently the definitions server has a complete set of terms that have been defined by the OGC since the inception of the OGC Naming Authority - which aims to keep all such URL references consistent. In Citizen Science we can find hundreds of projects dealing with similar topics but it is difficult to know if they are collecting variables that can be directly compared. By trying to link to a previous definition on the server they become connected to other projects. By exposing their definitions in the definition server other citizen science projects can reuse the same definitions and methodologies.

It has the ability to get machine readable versions of these details (e.g. JSON to allow simple integration of details into Web and mobile applications). It has a flexible capability to cross-link between terms and the ability to use any information model to extend available details. It allows for per-term or as-package download.

In this Interoperability Experiment the defintion server was improved and presented by the OGC to the other participants in the IE. As of Nov 2019 he API is undergoing an upgrade to comply with the emerging W3C Recommendation for "Content Negotiation by Profile" [https://www.w3.org/TR/dx-prof-conneg/]. In the next phase of the IE, we will test the applicability of the definition server for citizen Science purposes.

10.1. What the definition server does.

The OGC Definitions Server is a Web accessible source of information about things ("concepts") the OGC defines or that communities ask the OGC to host on their behalf. It applies FAIR principles to the key concepts that underpin interoperability in systems using OGC specifications. These things can be anything that is important in the course of interoperability around spatial information where the OGC plays a role in facilitating common understanding - either through publishing specifications or assisting communities to share related concepts. OGC uses stable web addresses (URIs) to unambiguously identify concepts in its specifications. The Definitions Server makes those URIs "work" - i.e. makes them dereference to a definition that can be used.

The OGC Naming Authority manages the Definitions Server to ensure all URIs are stable with transparent governance. These identifiers can thus be safely used in external context. All content is freely available for re-use. Re-use is envisaged largely through the machine-readable versions

Examples include the OGC glossary, technical terms from application schemas, for example the HY schema from the https://www.opengis.net/def/appschema/hy_features/hyf/HY_HydroFeature [hydrology domain], or many others.

Even though currently a limited search capability is provided, the Definitions Server is implemented using Linked Data principles - so the combination of stable URIs allowing references to be made from outside, and "follow your nose" navigation via links from one concept to related concepts provides enhanced findability.

The Definitions Server does not make any assumptions about the client software that may be used now or in the future, other than the use of HTTP protocols. This enhances accessibility for different environments.

The "Web-friendly" way of using an identifier (i.e. a URL) to get more information is augmented by "content negotiation" - the Definitions Server can deliver both user friendly Web pages and other forms of resource representations, e.g. JSON-LD or Turtle (TTL).

Figure 38 shows different views of a resource HY_Feature. The left panel shows an HTML representation, the middle shows the same information using TTL, and the right using JSON. All three representations have the same content, but differ in its serialization/format. This allows both human users to explore the OGC Definition Server, as well as machines to process its content.

Figure 38. Various representations of the same content (fragments, HTML left, TTL middle, JSON right)

10.2. Interoperability in the definition server

The interoperability of these resources is a key goal. There are several aspects of this handled using different mechanisms:

- Content model: can the client understand how the data is structured?
- Encoding: can the client parse the response
- Interaction: how can a client ask for the form it needs?

10.2.1. Content interoperability

The identifiers mentioned above, i.e. the URLs that can deliver content to the user, are termed Concepts and are organized into ConceptSchemes - and Collections. Concept, ConceptScheme, and Collections are defined by SKOS. SKOS, the Simple Knowledge Organization System, is a common data model for sharing and linking knowledge organization systems via the Web. It is a W3C recommendation.

So why SKOS? Many knowledge organization systems, such as thesauri, taxonomies, classification schemes and subject heading systems, share an almost similar structure, and are used in almost similar applications. Even though they might even share exact semantics, you need to learn that by explicitly discovering, accessing, and evaluating the content. Without a standardized interface, this endeavor is labor-intensive and can hardly be executed by machines.

SKOS captures much of this similarity and makes it explicit. It enables data and technology sharing across diverse applications by providing a lightweight, intuitive language for developing and sharing knowledge. In most cases, existing knowledge can be transformed into SKOS, because the SKOS data model provides a standard, low-cost migration path for porting existing knowledge organization systems.

10.2.2. Encoding Interoperability

The Definitions Server currently offers a range of encodings for all terms: 1. HTML 2. JSON (using JSON-LD augmentations to specify URLs) 3. RDF (as XML,TTL or JSON-LD) 4. Plain text

Where applicable certain types of resources are also available in the original or additional formats. For example Application Schemas will be made available as XML schema (XSD) and UML (XMI) forms.

10.3. Using the Definitions Server

10.3.1. URI access

Access definitions by following any URI supported.

The server will respond with a HTTP 303 URI redirect to the current service interface appropriate to the requested profile(view) and format.

http://www.opengis.net/def/docs/03-003r10 ⇒ HTTP 303 Location: http://defs.opengis.net/elda-common/ogc-def/resource?uri=http://www.opengis.net/def/docs/03-003r10

(the actual final resource URL may change as we improve the interface - but the original URI will always work)

10.3.2. ConceptSchemes, Collections and Semantics

Every term belongs to a "ConceptScheme" which will usually be part of the path

http://www.opengis.net/def/docs/03-003r10 skos:inScheme http://www.opengis.net/def/docs

Each part of the path ending with "/" will represent a Collection that contains a list of members http://www.opengis.net/def/docs policy:collectionView http://www.opengis.net/def/docs/

Terms may also a non-overlapping set of broader/narrower relationships, with the top of each hierarchy linked via skos:hasTopConcept from the ConceptScheme

This supports the following linkages: 1) ConceptSchemes are the "unit of governance" where metadata and download links for sets of definitions can be accessed 2) Collections are a flexible nested way of listing related subsets of terms - where lists may overlap - but do not state semantic relationships between terms 3) Terms are the basic resources with definitions 4) Terms may be semantically related using broader/narrower and other match (e.g. skos:exactMatch)

10.3.3. Search

A basic search capability is provided via the underlying interface e.g. http://defs.opengis.net/elda-common/ogc-def/concept?labelcontains=Catchment

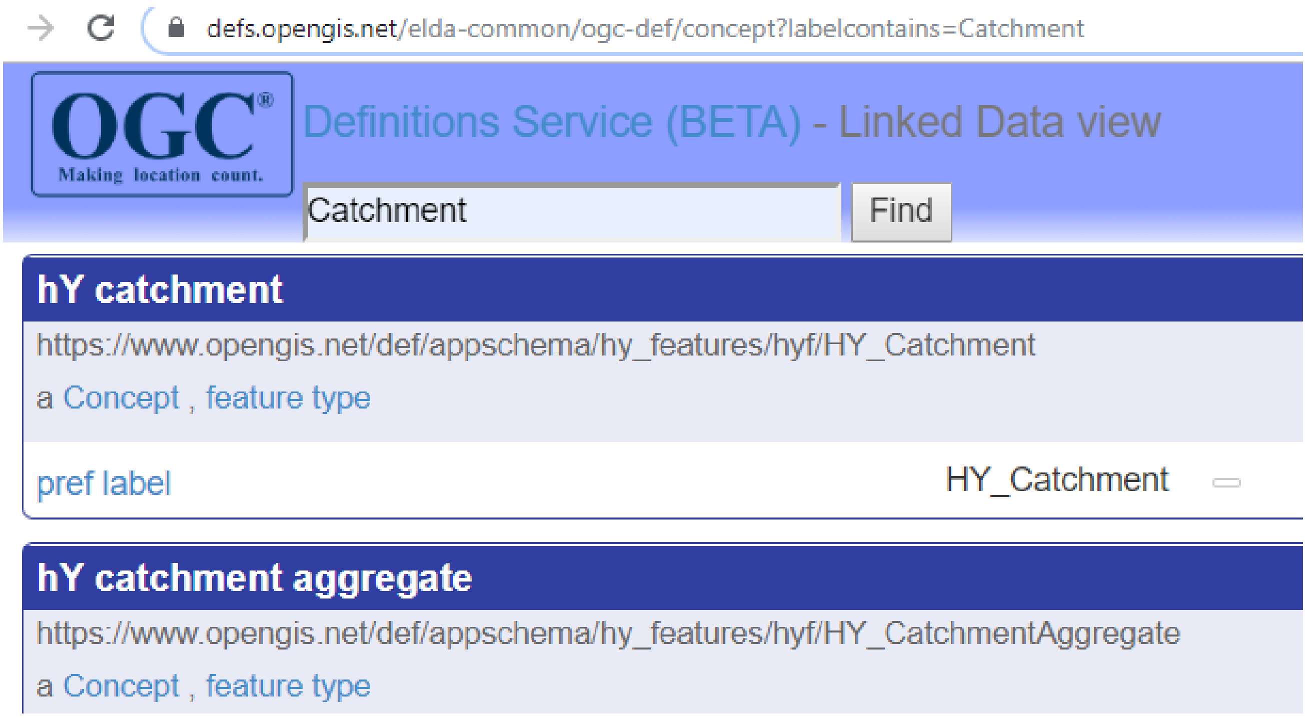


Figure 39. Search option

This provides machine readable outputs if requested via the _format parameter or the HTTP Accept: header. https://defs.opengis.net/elda-common/ogc-def/concept?labelcontains=Catchment&_format=ttl

Searches may be constrained to a specific concept scheme:

https://defs.opengis.net/elda-common/ogc-def/concept?labelcontains=Profile&scheme=http://www.opengis.net/def/docs

(note URL encoding is required for parameters with URI values - browsers tend to do this automatically)

10.3.4. Downloading Data

Every term includes a link to an "alternates" view

see also	HY_CatchmentAggregate?_view=alternates

Figure 40. Alternates link

(This link can be accessed by qualifying any Definitions Server hosted URIs with _view=alternates or _profile=alternates. A W3C compliant view for the specific concept (not the dataset as a whole) can be accessed with _profile=all) This view lists available formats for both the individual term and the collection or package that defines it:

Information resources available for specified object: https://www.opengis.net/def/appschema/hy_features/hyf/HY_CatchmentAggregate

View	Description	Resources
alternates	lists available information resources for a concept identifer	[json] [json-ld] [rdf] [xml] [html] [ttl]
_default	Default description view of identified object	[json] [json-ld] [rdf] [xml] [html] [ttl]

Dataset related links: https://www.opengis.net/def/appschema/hy_features/hyf

View	Description	Resources
alternates	lists available information resources for a concept identifer	[json] [json-ld] [rdf] [xml] [html] [ttl]
_default	Default description view of identified object	[json] [json-ld] [rdf] [xml] [html] [ttl]
iso190150	OWL-ISO19150 (as TTL via OGC Git)	[ttl]
owlbasic	OWL (as TTL via OGC Git)	[ttl]
xmi	UML (as XMI via OGC Git)	[xml]
conceptscheme	SKOS (as TTL via OGC Git)	[ttl]

Figure 41. Available alternate representations

ConceptSchemes offer download options for original sources of definitions - for example an Application Schema will have a download link for the canonical UML model file.

Collections allow list of concepts to be downloaded.

Concepts allow simple packages of information about the concept itself to be accessed.

Chapter 11. User authentication and applications federation

In the Interoperability experiment, we experimented with the federated identity provider developed in the project H2020 Landsense as authorization server that enables a federation of applications.

The Landsense project contributed the Engagement Platform (https://lep.landsense.eu/Project/LEP), the H2020 Scent project contributed the Scent Harmonisation Platform Visualisation Site (https://scent-harm.iccs.gr/) and the H2020 NextGEOSS contributed the NiMMbus Geospatial user feedback system (https://www.opengis.uab.cat/nimmbus/). The three platforms were able to work together and use the Landsense authorization server to authenticate users and create a Single Sign On experience. From the user perspective, once logged in one of the three platforms it can use the other two in a transparent way without having to authenticate again.

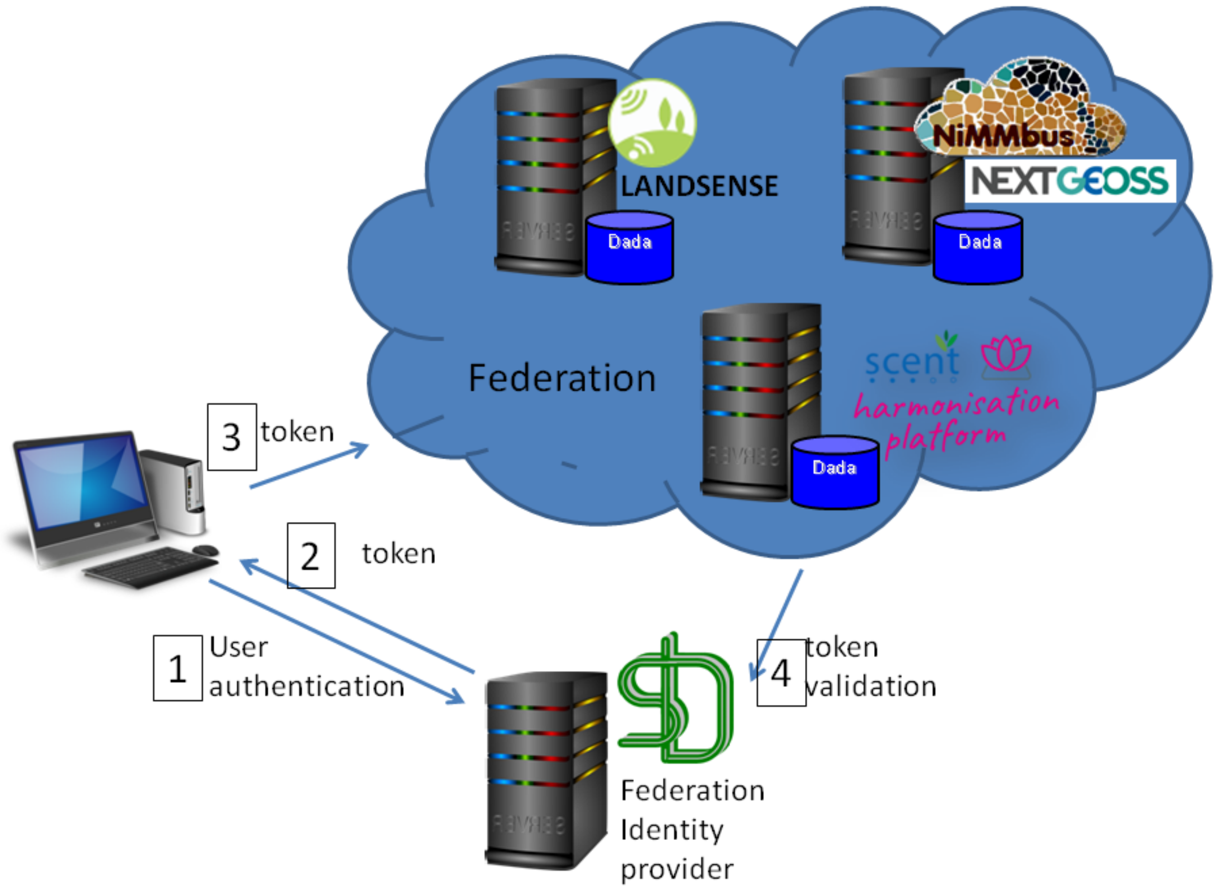


Figure 42. Federation diagram

The federation is designed in a way that is complain with the GDPR EU regulation and the user is in full control of which information is releasing, to whom and with that purpose. In an extreme case, the user can decide to do not release any adentity information to an application in the federation.

11.1. The LansSense Authorization Server

With the Authorization Server (AS), the federation supports users to authenticate from a variety of login providers including social media, organizations and Academic institutions participating in eduGAIN. Based on the trust in the login providers, services, tools and Application Programming Interfaces (APIs) can be provided to the platform by either operating a compliant SAML2 Service Provider or an Open Authorization 2 (OAuth2) Resource Server that accepts JSON Web Token (JWT) Access Tokens from any LandSense compliant OAuth2/OpenID Connect Provider.

The platform is extensible to any other login providers as long it is compliant with the Federation requirement regarding the participation as a login provider: deployment of a Security Assertion Markup Language v2 (SAML2) compliant Identity Provider. The LandSense Coordination Centre digitally signs and hosts the SAML metadata of the Engagement Platform by which trust is established between the SAML2 Identity and the Service Providers (the LandSense Federation).

The LandSense Authorization Server acts as a GDPR compliant broker between the personal information received after a user's login and registered applications based on user approval. In order to honour GDPR data minimisation, the AS requests from the Identity Provider (IdP) at login only that amount of personal information, as it is required by a registered application. This amount (and which attributes in detail) is controlled by the registration / login level. The AS provides five levels of which the first two do not enable an application to obtain personal information: AUTH, CRYPTO, PROFILE, EMAIL, PROFILE+EMAIL.

11.1.1. Five levels of personal information in the federation

AUTH: Any application that is registered with this level must not be GDPR compliant, as there is no information about the user other than "yes we know that you have successfully logged in with one of the LandSense IdPs). After login with Level AUTH you will not see any personal information.

CRYPTONAME: Any application that is registered with this level will receive a crypto name for the user. This crypto name is unique in LandSense and generated after a successful login. The crypto name is not stored which ensures that no personal information can be obtained based on the single possession of the crypto name. This allows applications to cluster (group) user contributions without knowing the real identity of the user. Because of that, any registered application processing the crypto name must not be GDPR compliant. After login with Level CRYPTO you will see your crypto name as value of the personal claim "sub".

PROFILE: Any application that is registered with this level will be able to receive personal information as defined in the OpenID Connect specification for the scope profile (provide URL) after the user has given their approval. Any application operating on this level must be fully GDPR compliant, which means that the registration process requires to provide a URL to the privacy statement of the application. This privacy statement defines which personal information is requested, for which purpose and which operators will be able to also process the personal information. After login with Level PROFILE you will see the crypto name plus all available personal information that fall into the scope profile.

EMAIL: Any application that is registered with this level will be able to receive personal information as defined in the OpenID Connect specification for the scope profile (provide URL) after the user has given their approval. Any application operating on this level must be fully GDPR compliant, which means that the registration process requires to provide a URL to the privacy statement of the application. This privacy statement defines which personal information is requested, for which purpose and which operators will be able to also process the personal information. After login with Level EMAIL you will see the crypto name plus all available personal information that fall into the scope profile.

PROFILE+EMAIL: This is a combination of scopes PROFILE and EMAIL. After login you see your crypto name, email address, whether it is validated and all the personal information received for scope profile.

11.1.2. How add an web application into the federation

Any application (mobile or web browser based) can be included if they support OpenID connect and the are registered with the LandSense OpenID Connect provider. Assuming successful registration, the application can then be used on own provisioning and offerings from the Federation.

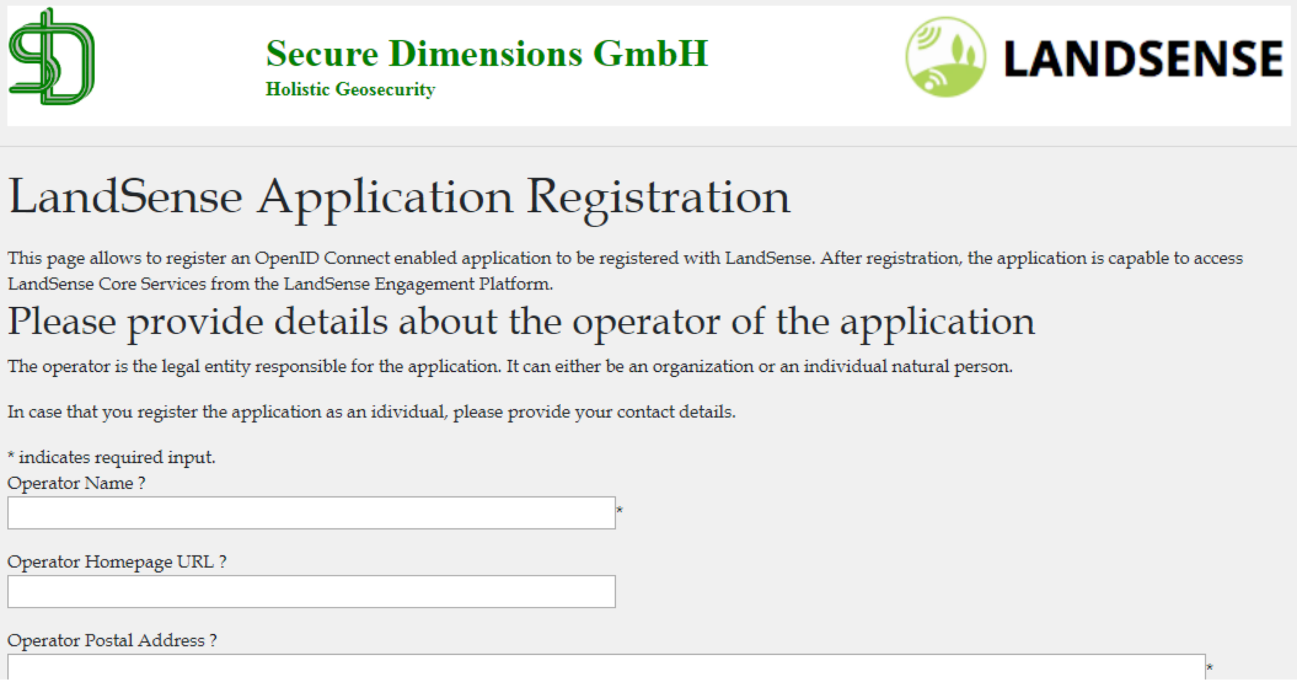


Figure 43. Federation Application Registration

11.2. LandSense Engagement Platform

In a nutshell, the LandSense Engagement Platform (https://lep.landsense.eu) is to become the marketplace where citizens can participate in the various Land Use and Land Cover (LULC) related campaigns and interested parties can reuse existing services and register new applications.

The first version of the LandSense Engagement Platform was realized based on the existing tools, services and platforms from LandSense partners as well as new applications built for the Demo Cases.

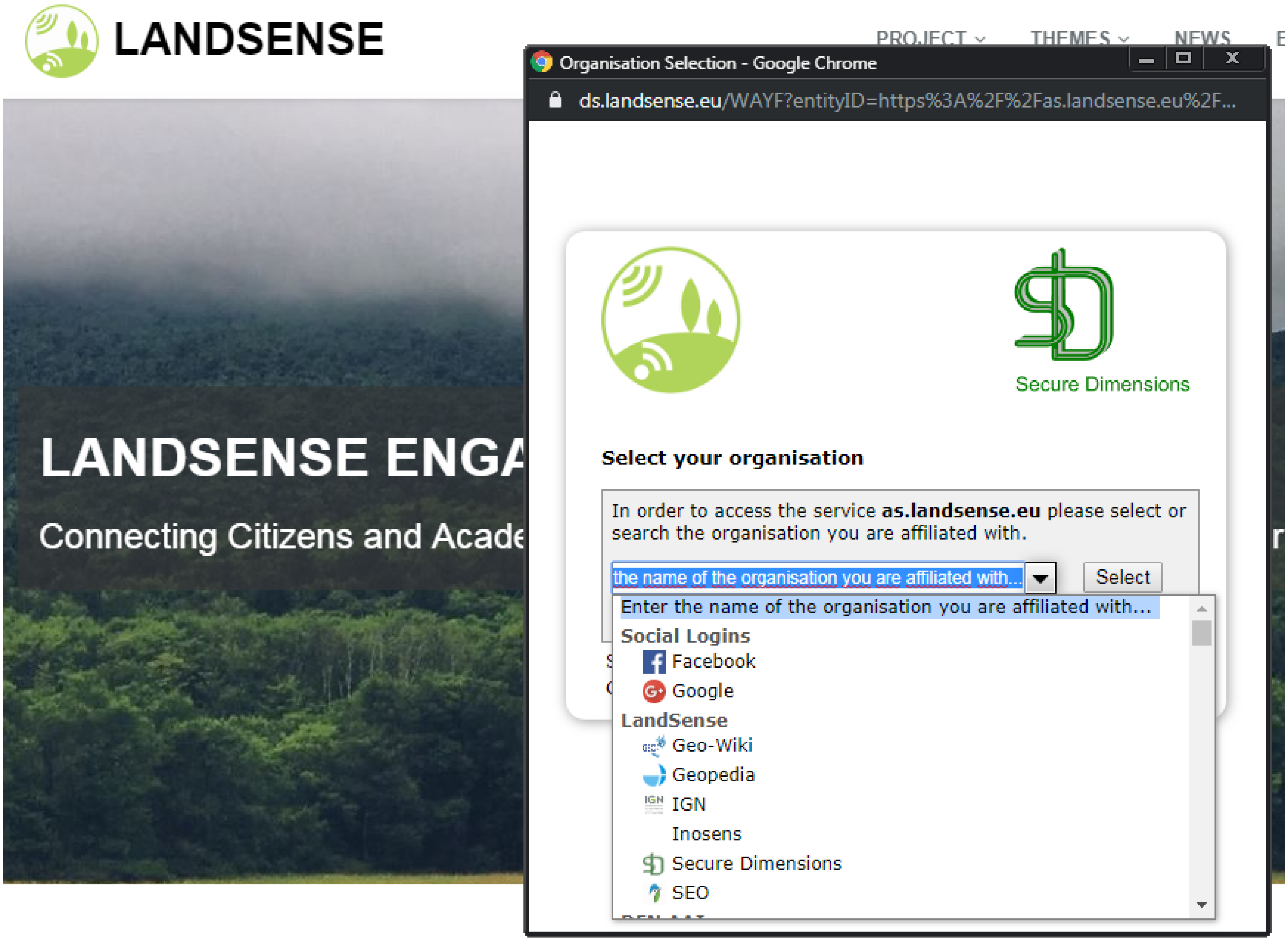


Figure 44. LandSense Engagement Platform login

11.3. Scent Harmonization Platform

Scent Harmonisation Platform Visualisation Site (https://scent-harm.iccs.gr/) is a client application tailored for the purposes of inspecting and visualising traditional in-situ and citizen-generated observations.

The Visualisation site constitutes a custom innovative application that exposes the resources made available from the Scent Harmonisation Platform. The application conforms to OGC SensorThings API standard and it consists of the following main characteristics:

- User-friendly interfaces enabling both time-series analysis and spatial representation of SensorThings API resources with support to time-series analysis, metadata and visualisation and a dashboard.
- An interactive campaign dashboard that enables the spatial visualisation and graphic representation of the images of Land Cover/ Land Use elements that have been collected from the volunteers

Scent Harmonisation Platform manages a variety of citizen-generated data as well as environmental data that been collected through in-situ monitoring stations in Kifisos river basin, Attica, Greece. All the data are being maintained and have been structured according to widely accepted standards, such as the Open Geospatial Consortium (OGC) in order to be compliant with open and unified frameworks (such as SensorThings API).

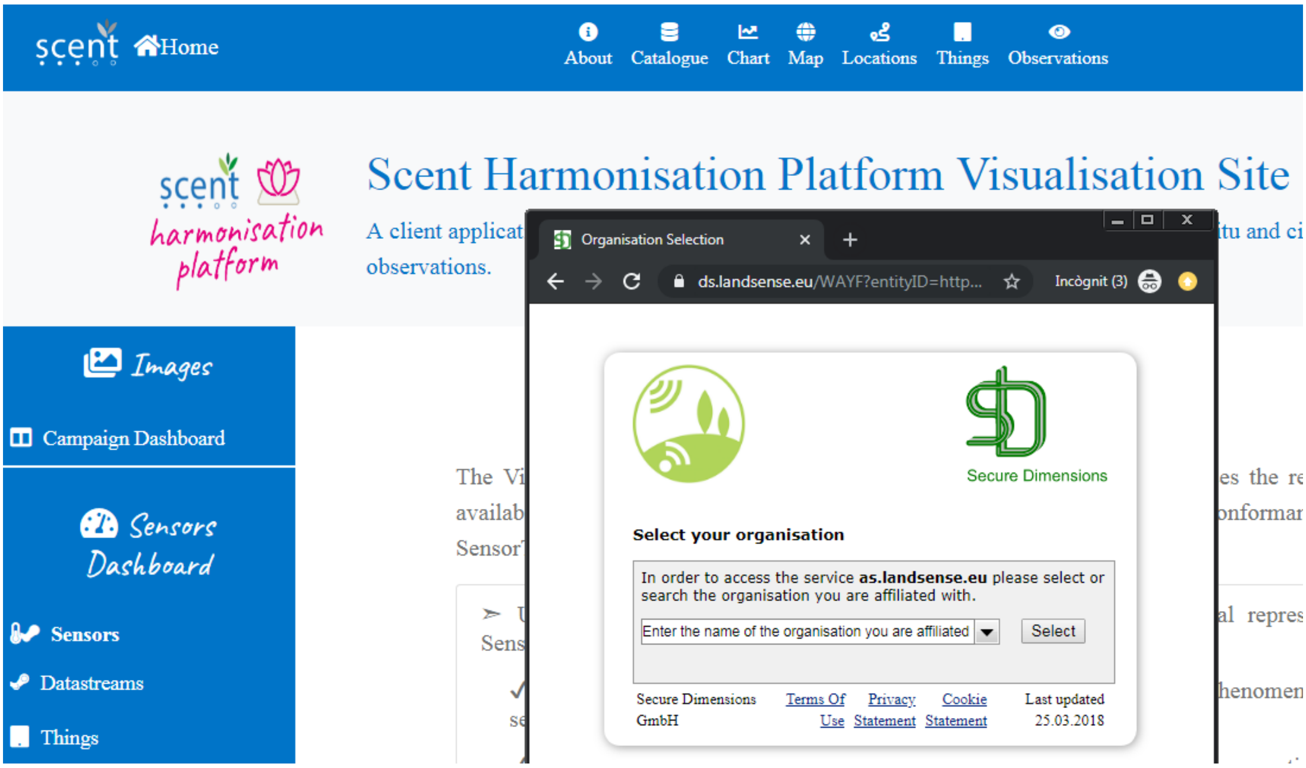


Figure 45. Scent Harmonization Platform Login

11.4. NiMMBus Geospatial User Feedback

The NiMMBus web portal records geospatial user feedback about existing geospatial resources. The user is able to provide comments, rates, quality reports and publications related to a geospatial resource. It can be used to comment on datasets but also on individual observations. The system allows to creating a citation of an external resource (in an external catalogue or repository) and associate feedback items about it. It builds upon a service developed in the H2020-funded NextGEOSS project, adding the capability of integration into the LandSense federation authentication.

The system is based on the NiMMbus; a solution for storing geospatial resources on the MiraMon cloud. The system implements the Geospatial User Feedback (GUF) standard developed in the OGC GUF (and started in the FP7-funded GeoViQua project).

The solution is composed by three elements: the open source code for a JavaScript the client, a server that stores the feedback information and a well document API that allows for interacting with the client.

Figure 46. NiMMbus NiMMBus Geospatial User Feedback Login

Chapter 12. Connecting Citizen Science data sets to GEOSS

In addition to the developments outlined above, this Interoperability Experiment also examined possibilities for the Citizen Science community to make their projects discoverable and accessible via the Global Earth Observation System of Systems (GEOSS).

NOTE	The worldwide effort to build GEOSS is led by the Group on Earth Observations (GEO). GEO is an intergovernmental organization working to improve the availability, access and use of Earth observations for the benefit of society. GEO works to actively improve and coordinate global Earth Observation systems and promote broad, open data sharing.

We see a major benefit in establishing this connection because GEOSS already provides an established process to clarify data policies together with established data management principles, and to provide the minimum metadata required to access the data sets created by Citizen Science projects. This activity thereby helps to surface and to mobilize already existing data sets - with clear acknowledgment of the Citizen Science contributions. We expect that this work provides concrete contributions to the increasing discussions on how Citizen Science could connect to GEOSS and, more generally, how more in-situ data and derived knowledge could become available at the global level.

Within this Interoperability Experiment, we particularly explored the possible technical connection with the GEOSS Platform facilitated by OGC standards. We also identified organizational structures that would be required in order to provide a more flexible and scalable solution to Citizen Science projects, both big successful ones and small ones. We consider this as a major need for the future evolution of the Citizen Science contribution because of the high number of already existing projects that could potentially be connected to GEOSS. OGC standards can play an essential role to facilitate this connection. This elaboration would be also generalized to ongoing debates on increasing the availability of in-situ datin GEOSS.

Notably, we see the connection with GEOSS as one highly promising way to make Citizen Science data better accessible and more widely used. For example, complementary efforts might be undertaken, to increase the findability via mainstream search engines, or – more generally – to provide the machine readable information about projects resources that is required for automated harvesting by web-crawlers. We come back to this issue in the final part of this section, when outlining possible follow up activities.

With our exercise so far, we follow the next overarching principles and carry them to the Citizen Science community:

- GEO Data Management Principles [https://www.earthobservations.org/documents/dswg/201504_data_management_principles_long_final.pdf], addressing issues such as discovery, access, traceability, quality documentation and preservation.

- Not favoring any silos, i.e. need for open data and leveraging open solutions, which in this context means the provision of data as part of the GEOSS Data Core [https://www.earthobservations.org/geoss_dsp.shtml], but also the application of the GEO Architecture Principles, which in addition advocate flexibility, scalability, etc.

- Distributed, standards based, and flexible to support interoperability while meeting a range of use cases and needs, which translates to the use of the GEOSS Discovery and Access Broker (GEO DAB) [https://www.geodab.net/] that includes – among other - the support to a large range of OGC standards.

12.1. Connecting a single Citizen Science project to GEOSS

The most straight forward way to connect Citizen Science data sets to GEOSS is the inclusion of the project that collects these data sets at the GEOSS Platform. This process essentially requires the registration of the project in the GEOSS 'Yellow Pages' and it can be used to register also multiple data sets from one single project (see also Figure 47 below).

Figure 47. Sketch on registering a single project at the GEOSS Platform

The entry of the Yellow Pages requires information (metadata) such as:

- *Lead organization* (name, description, URL, geographical coverage, GEO affiliation, contact points, etc.)

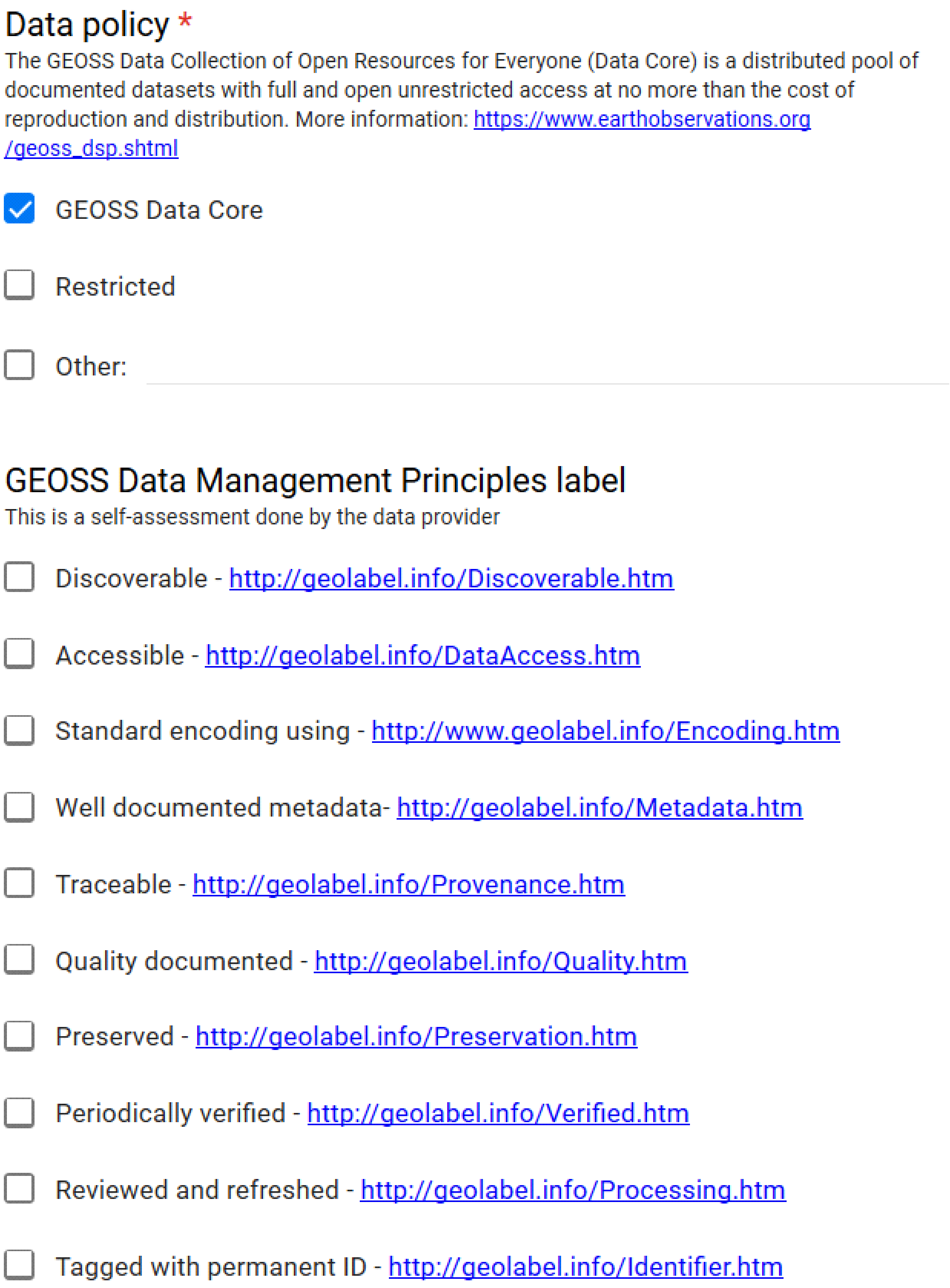
- *Type of* provided online *resource* (data, vocabulary, model, algorithm, etc.) – in the context of this Interoperability Experiment we focus on data

- *Data policy*, including the option to declare as free and open by choosing "GEOSS Data Core" (see Figure 48 below)

- Indication of GEOSS *data management principles* that are implemented (see Figure 48 below)

- Relevance for the *Sustainable Development Goals*

- A service *endpoint*



Data policy *

The GEOSS Data Collection of Open Resources for Everyone (Data Core) is a distributed pool of documented datasets with full and open unrestricted access at no more than the cost of reproduction and distribution. More information: https://www.earthobservations.org/geoss_dsp.shtml

- [x] GEOSS Data Core
- [ ] Restricted
- [ ] Other:

GEOSS Data Management Principles label

This is a self-assessment done by the data provider

- [ ] Discoverable - http://geolabel.info/Discoverable.htm
- [ ] Accessible - http://geolabel.info/DataAccess.htm
- [ ] Standard encoding using - http://www.geolabel.info/Encoding.htm
- [ ] Well documented metadata- http://geolabel.info/Metadata.htm
- [ ] Traceable - http://geolabel.info/Provenance.htm
- [ ] Quality documented - http://geolabel.info/Quality.htm
- [ ] Preserved - http://geolabel.info/Preservation.htm
- [ ] Periodically verified - http://geolabel.info/Verified.htm
- [ ] Reviewed and refreshed - http://geolabel.info/Processing.htm
- [ ] Tagged with permanent ID - http://geolabel.info/Identifier.htm

Figure 48. Parts of the items asked for when registering in the GEOSS Yellow Pages

Once completed, the information provided by the Citizen Science project is passed on to the GEO DAB team. Members of this team check the entries and run some tests (for example, if the provided endpoint actually serves the intended data, if it needed to implement the required standards (from the OGC or comparable alternatives) and if it indeed follows the indicated principles and applies the indicated data policy). This testing might entail a dialogue with the registering Citizen Science project in order to make the project's offer fit to the promises in the entry for the GEOSS Yellow Pages.

SCENT Citizen Observatory [https://scent-project.eu/] successfully undertook the process to offer its citizen-generated data to GEOSS. Following the administrative registration that was described in detail above, the Interoperability Registration and Brokering workflow took place. This was the second phase and consisted of the following steps:

- Technological Information was provided to the DAB team. More specifically the SCENT web server URL, the endpoints of the SCENT OGC web services (WMS, WFS) were sent to the DAB team along with the accompanied descriptions. In addition, information was provided about WFS and WMS services versions, the supported relevant operations through the APIs (e.g. GetCapabilities, DescribeFeatureType, GetFeature, etc.), the various feature types, the SCENT data to be integrated to GEOSS (i.e. event, images, video metadata) and the associated geographic regions that they cover (i.e. Kifisos river basin, Attica Greece and Danube Delta, Romania).
- DAB team conducted a set of interoperability tests with SCENT WFS and WMS services, including discoverability, accessibility and visualisation use cases. A test report (GEO DAB/SCENT web server Brokering Test Report) was generated as an output of this process.
- Notification was received from DAB team regarding the successful conduction of the tests and that no interoperability issue was found. Taking into consideration also suggestions indicated in the test report, metadata information of the datasets provided was further enriched.
- As a final step of the process, the DAB team proceeded with the successful integration of SCENT web server into GEOSS Portal (offering its services to production environment). Thus, users can access SCENT resources via the GEOSS Portal catalogue (https://www.geoportal.org/?f:sources=wfsscentID%2CwmsSCENTID) as presented in the Figure 49.

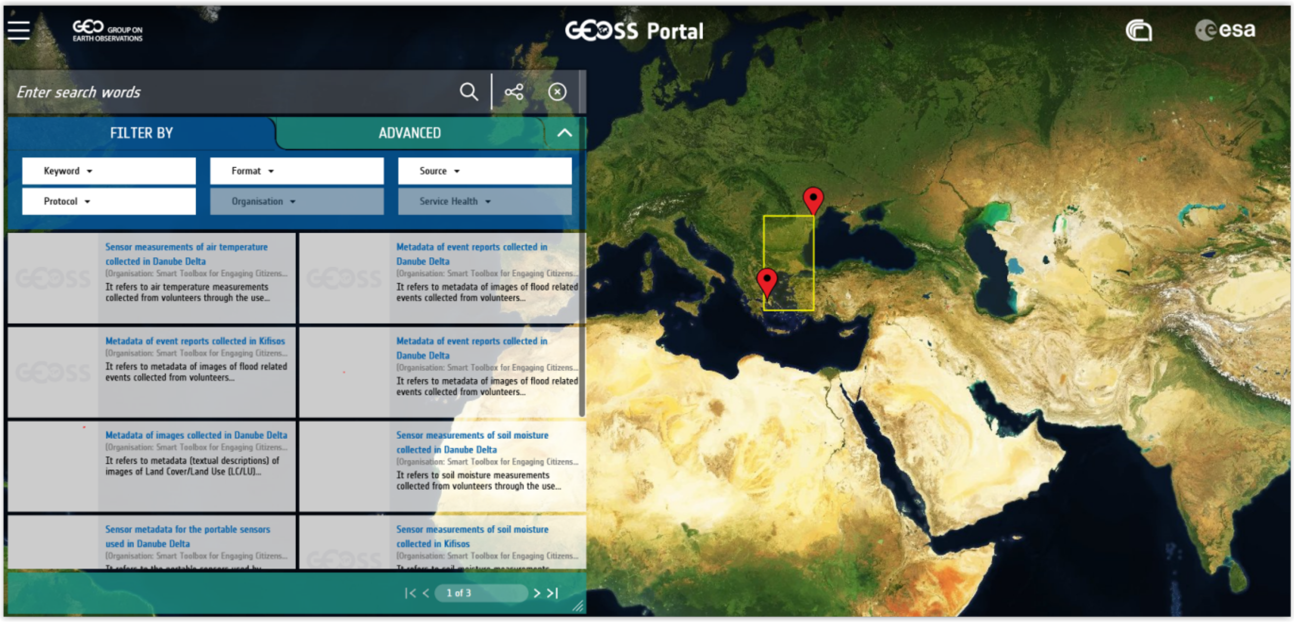


Figure 49. Scent resources available in GEOSS portal

Overall, we proved that Citizen Science projects indeed can implement the free and open data policy of the GEOSS Data Core, and that all of the GEOSS Data Management Principles are followed by Scent. Furthermore, out of the OGC Citizen Science Domain Working Group (DWG) we offer examples and guidance on how this can be realized with OGC standards. However, realistically speaking, and in order to see concrete progress, it appears more feasible that projects first enter what they already support, and at least fulfill the needs to make their data sets discoverable from within the GEOSS Platform [http://geolabel.info/Discoverable.htm] using well-documented metadata [http://geolabel.info/Metadata.htm], and including information about data quality [http://geolabel.info/Quality.htm]. Following the brokering approach of GEOSS, this entails that the projects provide a minimum of required information, and ideally follow one of the multiple standards that are already supported by the GEOSS platform. If it should not be immediately possible to also provide one of the multiple options to make data accessible [http://geolabel.info/DataAccess.htm] or to provide that data in an already recognized encoding [http://www.geolabel.info/Encoding.htm], it might still be considered to register a projects and then update the record in the Yellow Pages once the additional functionalities for harmonized data access are put in place by the project.

12.2. Why a case-by-case registration is not the best way forward

Although the possibility to register Citizen Science projects of any size in GEOSS exists and has been illustrated in the previous section, we do not recommend that each and every project go ahead and register by itself now. Potentially, the number of relevant contributions is (at least) in the hundreds, see for example, this project inventory [http://data-staging.jrc.it/dataset/jrc-citsci-10004] and the related prototype of a project catalogue [https://ec-jrc.github.io/citsci-explorer/] (Figure 50).

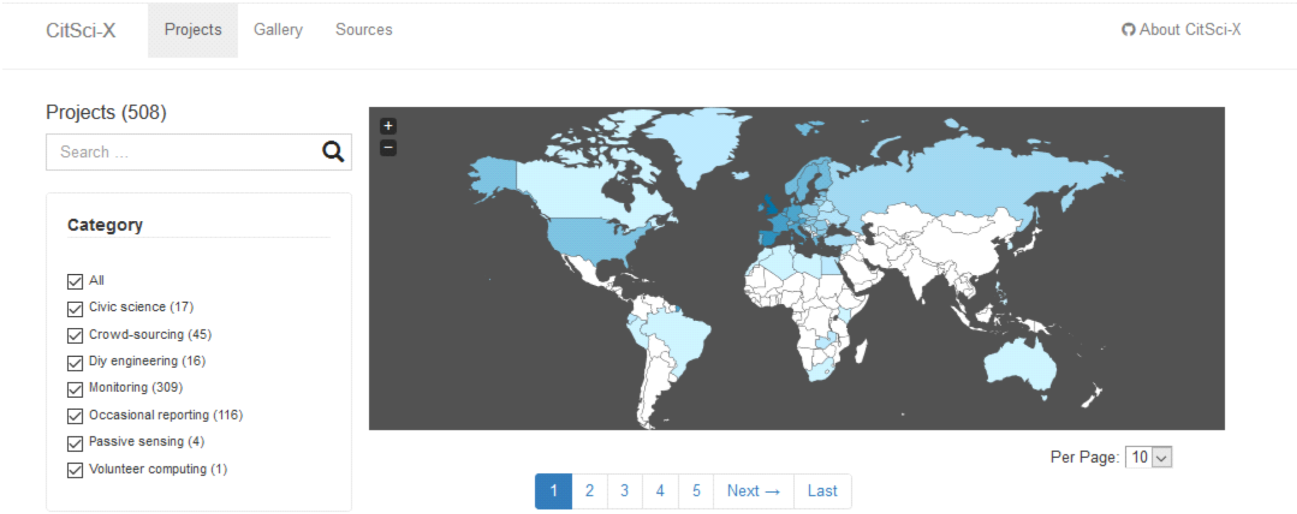


Figure 50. Example Citizen Science explorer providing access to 508 different projects (status: 13 June 2019)

It becomes obvious that a case-by-case registration per project (which each might want to register one data set or more) would create a bottle neck towards the GEO DAB and the team that is responsible to evaluate and test entries in the GEOSS Yellow Pages. As a result of our investigations within this Interoperability Experiment, we therefore suggest to elaborate on and develop an **intermediate layer** that provides the required organizational and technical support to the Citizen Science community so that their data sets become better discoverable, accessible, and potentially more widely used - thereby also amplifying visibility and impact of the individual projects.

We consider such structures particularly important in view of larger mobilization campaigns of Citizen Science projects, as, for example, planned within the context of the Earth Challenge 2020 (EC2020). Again, also here the two/multiple-step approach - where project resources become discoverable first and commonly accessible in a second stage - might be most realistic in order to progress more quickly and to have intermediate results.

12.3. How to improve the connection of Citizen Science into GEOSS

In order to move ahead, we identified requirements that we are grouping in different approaches that complement each other.

12.3.1. Provide technical support to connect to the GEOSS platform

There appears a need to slot a technical support for Citizen Science projects in ahead the GEO DAB. This additional support should remove the potential bottle neck and help to scale up the number or Citizen Science projects and their data sets in the GEOSS Platform (and ideally in the GEOSS Data Core). Requirements for this support entail:

- Support Citizen Science projects in filling the GEOSS Yellow Pages.

- Proving examples and guidance on the use of OGC standards for implementing GEOSS requirements for data discovery, quality descriptions, data access, data encodings, etc.
- Pre-testing of yellow page entries before registration in GEOSS.
- If necessary, interaction with individual projects to correct their entries for the Yellow Pages.
- Liaise with the GEO DAB team in order to actually register the new entries.
- Establish a capacity building mechanism, capable to support and equip existing initiatives with the necessary skills to apply data management principles related to the accessibility, discoverability, re-usability and curation of their resources.

12.3.2. Federate multiple Citizen Science projects and their endpoints into a single access point

To reduce the number of endpoints connected to the GEOSS Platform, federations of citizen Science projects could act as hubs that would in fact cluster multiple Citizen Science projects and their endpoints into a single access point, which is then registered within the GEOSS Platform (see Figure 51).

This federations could be thematic or regional and take advantage of the current structure of activities in the GEO work program.

Considering the Earth Challenge 2020 we could imagine the following architecture: EC2020 will collect new data and offer it via a dedicated API. At the same time, several already existing Citizen Science projects partner with EC2020 and also provide access to their data (in different forms). For the connection to GEOSS, EC2020 could provide a gateway that federates the newly collected data and the offerings of the different partners to a single discovery service and a single data access service. These two endpoints would be registered via the Yellow Pages with the GEOSS Platform only once and thereby make the EC2020 resources more widely visible, together with a clearly defined and well-known data policy and following most recent data management principles. The figure below depicts this setting.

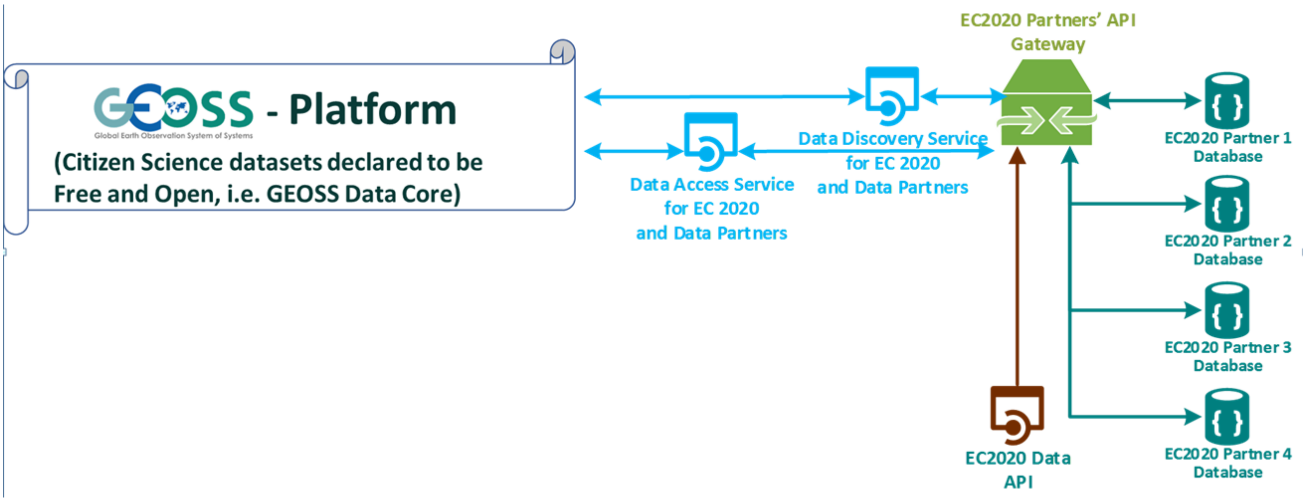


Figure 51. Sketch of a possible connection between EC2020 and the GEOSS Platform (one possible scenario), as an example of a more coordinated structure to make Citizen Science data discoverable and accessible via GEOSS

NOTE	It is important to realize that we are not proposing a single federation maintained by GEO but a collection of self organized federation that act as a aggregation point and provide services to the citizen science projects. This federations can have different scales and can be thematic or regional.

Extra considerations for Citizen science federations

A federation of services can provide extra services to the citizen science projects. This is a short list of examples that can grow with time.

Standard translation

Currently the GEO DAB does not fully support Sensor Web Enablement (SWE) standards such as OGC Sensor Observing System and OGC SensorThingsAPI. A service in the federatino can provide the translationof services to other supported ones such as WFS or WMS allowing for an harmonized access via the GEOSS Platform.

Data agregation

Some compatible projects can be aggregated into bigger virtual datasets that can be served on demand.

Federated authentication

As discussed in User authentication and applications federation, a federation could provide a mechanism to authenticate user that can them provide observations to several projects in the federation from a single app or from multiple apps used in parallel. This can be useful on the data input case (data capture) but could be also used for a pull of experts validating data coming from different projects. A federated authentication can also protect the privacy of the citizens as will be discussed in Citizen privacy and protection

Data preservation

A federation can provide a service that allow for archiving data from project campaigns from ephemeral citizen science projects or projects that can no longer be maintain.

Common definitions

Sharing common definitions (with tools such as the definition server proposed in Definition Server) will be essential to ensure data integration and should be a part of the federation.

12.3.3. Improve networking and capacity building

On the other hand, and because the offerings made above alone would not be enough to actually advance from the current situation, also the networking the Citizen Science community deserves dedicated attention. The current Earth Observations Citizen Science community activity in the

GEO work program could provide additional help. The following requirements has been identified:

- Mobilizing existing data sets, i.e. reaching out to the Citizen Science community and let them know about this work and the linked offering of increased visibility and possibly impact, providing guidelines and practical examples on what would need to be done from their side, offering support in establishing the connections.
- Help in preparing new data sets, i.e. be available to consult Citizen Science projects during their set-up phase, and let the community know about this offer.
- Promote FAIR data management and GEOSS as a practical way to get there.
- Provide access and training for (OGC) standard-based tools that the community can use to make the connection, and implement the desired data policy and data management principles.

For further discussions and possible realizations, it should be considered if the support outlined above could be provided in a coordinated but decentralized way. We could imagine that the above mentioned support could be set of for different geographic regions, thematic areas, or other sensible divisions (e.g. with a retaliation that is specific for EC2020, which would still need to be discussed). Such settings could also help to disseminate good practices, for example, on the use of OGC standards in this context.

12.4. Future work regarding to the GEOSS integration

The experiment helps us to identify current possibilities, and to shape parts of the way forward. However, it has also left a few questions unanswered and raised some new issues. We should develop different scenarios to meet the identified organizational requirements exposed before. From our experiences, we see particular needs to further investigate the following aspects:

- Acknowledging that Citizen Science data is already included in GEOSS today, i.e. systematically flagging where Citizen Science already contributed to a knowledge resource on the GEOSS Platform (GEOSS Data Core, ideally).
- Develop detailed examples and guidance on how CS projects can implement the different GEO Data Management Principles by using the many already supported OGC standards.
- Consider promoting Sensor Web Enablement (SWE) standards such as OGC Sensor Observing System and OGC SensorThingsAPI to be considered by the GEO DAB, because both standards appear to be taken up by several Citizen Science projects, but at the moment to write this lines the are not supported by the GEO DAB, so other standards (such as WFS or WMS) need to be implemented in addition to allow harmonized access via the GEOSS Platform.
- Consider Citizen Science not only as a data source, but also explore the possibilities and use of OCG standards when it comes to the engagement of Citizen Scientists as part of data validation.

- Also consider Citizen Science as part of the processing capacity, collective intelligence, data cubes, relationship to Web Processing Service (WPS), work on Artificial Intelligence, etc.

While focusing on the connection to GEOSS here, we should also investigate how this work related to the provision of metadata for ‘flat’ online searches (e.g. Google search) and the accessibility to automatic web crawlers. We might want to address both in a single go. If we will work towards intermediate organizational structures the help the Citizen Science community in using OGC standards and the GEOSS Platform for improved data policies and management, can these intermediaries – and the tools and services they provide – also automatically cover these complementary needs?

12.4.1. Citizen privacy and protection

The aspect of citizen privacy and personal data protection is a serious issue that should not be undermined. There has been recent examples of commercial companies using social media companies personal data and citizen profiles for unethical purposes of for their own profit. In extreme cases companies business model was based on collecting, and integrating personal data of their users to then sell to third parties personalized databases and services for commercial or political targeting. Accidentally allowing citizen science data to be gathered by these platforms, thus opening the door to their use of the personal data of those who have collected the data and those who use them, without their consent. This is clearly against the data protection regulations in Europe and other areas in some but still done by some if technically possible.

This important issue needs to be addressed by the individual citizen science projects the emerging federations and the GEOSS platform at large. This is a real problem that should be included in an GEO architecture ensuring a good balance between the necessary anonymity of the citizens personal data as well as the acknowledgement of their individual contributions when participating in the Citizen Science activities. The proposed federation discussed in the User authentication and applications federation has an embedded component taking care of this aspect (see Five levels of personal information in the federation) with two levels that ensure absolute privacy while other levels allow for some degree of acknowledgement and recognition. This needs to be complemented with the way hosts, manage and own data

It is our responsibility to raise this issues within the GEO community and find the right solution that will probably require a composition of technical, management and legal aspects.

Appendix A: Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
June 15, 2019	Joan Maso	.1	all	initial version
October 6, 2019	Joan Maso	.9	all	preparation for publication

Table 3. Revision History