Metrology for Integrated marine MaNagement and Knowledge-transfer nEtwork

Call H2020-INFRAIA-2020-1
Topic INFRAIA-02-2020
Integrating Activities for Starting Communities

Deliverable 3.2

Best practice guide on data harmonisation

09 30 2022,

Version 1.0

EUROPEAN COMMISSION	Project funded by the European Commission within the Horizon 2020 Programme (2014-2020)
Grant Agreement No.	101008724

Type		
R	Document, report excluding the periodic and final reports	X
DEM	Demonstrator, pilot, prototype, plan designs	
DEC	Websites, patents filing, press & media actions, videos, photos, etc.	
SOF	Software, technical diagrams, etc.	
OTHER	Procedures, management tools, flyers, etc.	

Dissemination level		
PU	Public, fully open.	X
CO	Confidential, restricted under conditions set out in Grant Agreement	
CI	Classified	

Revision history

R#	Date	Description/Reason of change	Deliverable contributors
R0.1	20220926	Draft for review	Christian Autermann, Simon Jirka, Markus Konkol, Enoc Martinez, Joaquin del Rio, Albert Garcia
R1.0	20220930	Review and internal approval	

Authors

52N: Christian Autermann, Simon Jirka, Markus Konkol, Enoc Martinez, Joaquin del Rio, Albert Garcia

Citation

This report can be cited as:

MINKE consortium D3.2 (2022), Best practice guide on data harmonisation, Christian Autermann, Simon Jirka, Markus Konkol, Enoc Martinez, Joaquin del Rio, Albert Garcia, Metrology for Integrated Marine Management and Knowledge-Transfer Network, MINKE.

License and attribution

Executive Summary

This document is MINKE's Deliverable 3.2 "Best practice guide on data harmonisation". It provides comprehensive guidance documentation on how to enable data harmonisation via specific interoperability standards and vocabularies.

This deliverable is the result of a dedicated task on the evaluation of relevant interoperability standards (T3.2) to be recommended for usage in the MINKE project. For creating this deliverable, different aspects of data harmonisation were considered:

- Standards for encoding observation data
- Standards for describing sensor data (metadata)
- Internet of Things protocols for transmitting data from sensing devices
- Interfaces for data access

Consequently, this document provides an overview about relevant standards that are recommended to harmonise the exchange of measurement data within the MINKE project but also beyond (e.g., via the European Open Science Cloud). While the identified standards cover mainly syntactic aspects, also the semantics of the encoded content is discussed. For this purpose, an overview of relevant vocabularies is outlined.

Specific consideration was given to approaches for handling observation and sensor quality information. For this purpose, additional elements for the identified data and metadata standards are proposed to enable a better determination of data quality and deriving its associated uncertainty.

Index

Executive Summary 4
Index 5
1 Introduction 7
2 Approach 8
    2.1 Questionnaire for MINKE Partners 8
    2.2 Main Findings 8
        2.2.1 Participants 8
        2.2.2 Results 9
            Captured parameters 9
            Spatial and temporal coverage and resolution of the data 9
            Update frequency & estimated data size 9
            Derivative data 10
            Data formats 10
            SensorML descriptions and Vocabularies 10
            Tools for data collection and management 10
            Calibration 10
            Open data and licensing 10
3 Relevant Standards for Harmonised Data Sharing 11
    3.1 Metadata Standards 11
        3.1.1 Sensor Model Language 11
            EMSO Generic Instrument Module 12
            Real-time Underwater Noise Measurements 12
        3.1.2 ISO 19115 14
    3.2 Standards for Observation Data 14
        3.2.1 OGC/ISO Observations and Measurements 14
        3.2.2 Relevant Technical Guidance from the INSPIRE framework 15
    3.3 Data Access Interfaces 15
        3.3.1 OGC SensorThings API 16
            SensorThings API Use Case at OBSEA 16

- 3.3.2 OGC Sensor Observation Service 17
- 3.3.3 Relevant Technical Guidance from the INSPIRE framework 17
- 3.4 Internet of Things Standards 18
  - 3.4.1 MQTT 18
  - 3.4.2 AMQP 19
- 4 Semantic Interoperability 20
  - 4.1 Parameter Vocabularies 20
  - 4.2 SensorML Vocabularies 20
- 5 Additional Considerations to Handle Quality Information 22
  - 5.1 Overview of Relevant Metadata Elements 22
    - 5.1.1 Sensor metadata 23
    - 5.1.2 Calibration and Testing Metadata 23
    - 5.1.3 Metadata for Observations 24
    - 5.1.4 Metadata for Data Quality Control 24
- 6 Summary and Outlook 26
- 7 References 27
- Annex I - Overview of Quality Related Metadata Elements 1

1 Introduction

This deliverable aims at providing guidance on how to achieve harmonised distribution of observation data, related metadata, as well as information on data quality. For this purpose, this deliverable introduces relevant standards covering those aspects which need to be considered to achieve interoperability.

In section 2 the approach used for assessing the needs for data harmonisation is introduced. This comprises mainly a survey conducted among all MINKE partners in order to determine which types of observation data are handled by the different partners and which information is available to describe the quality of these data sets as well as the underlying measurement processes.

After this, in section 3, the different data and metadata standards recommended for the use within the MINKE project are introduced. This comprises especially the following aspects:

- Metadata describing observation data sets as well as the underlying sensors/measurement processes
- Encodings for the observation data itself
- Additional information to describe data quality

In addition, in section 4, a first overview and mapping to relevant vocabularies is given which may be used to achieve not only syntactic but also semantic interoperability.

Finally, section 5 discusses additional aspects needed for handling data quality information. As this aspect is usually addressed to a lesser degree by the standards introduced in section 3, we propose which additional metadata elements should be included in order to provide the necessary information to assess the quality of an observation data set. Basis for these recommendations is the feedback received from the MINKE partners via the questionnaire as well as exemplary calibration reports typically used by the partners.

2 Approach

In order to identify relevant standards for harmonised data sharing, we first aimed to survey already existing and datasets that are planned to be created or collected in MINKE. For this purpose, we sent an online questionnaire to the MINKE partners. In the following, we first describe the questionnaire and then report on the main findings.

2.1 Questionnaire for MINKE Partners

First, the questionnaire collected background information from the participant, i.e., the associated partner, email address for further correspondence, and dedicated work packages. Second, the survey investigated the datasets and asked for their title, description and purpose, parameters, spatial/temporal coverage, and resolution. Furthermore, we asked for the update frequency of each dataset and whether it is derived from another dataset. Next, we requested information on the estimated size of the dataset and its format. We also wanted to know whether they have SensorML descriptions of the used sensors and whether they use standard vocabularies. Then, we asked the participants to list the tools used for data collection and management and whether they plan to provide information on calibration processes and reference data. Optionally, the participant could upload a calibration report. Finally, we asked for copyright, if the data can be made openly available, and which license will be used (especially in order to determine if specific consideration needs to be given to certain aspects of access control).

Most questions could be answered using free text. Participants collecting or providing multiple datasets were asked to complete this survey for every dataset. We created the survey using google form and distributed it via email. The survey was available for four weeks. A reminder was sent three days before the deadline.

2.2 Main Findings

2.2.1 Participants

Two participants (response #2, #11) were removed from the survey because they stated they do not collect data. Two participants (#3, #8) reported on the same dataset and were treated as one response. Two participants (#9, #10) reported on the same dataset but gave inconsistent responses. However, since in response #10 nine questions were answered with “I don’t know”, we assume #9 to be more reliable and removed #10 from the survey. The final dataset contains 13 responses from 12 participants (one participant completed the survey for two different datasets #13, #14) across 12 different partners and nine work packages (WP 1-6, 8-10).

2.2.2 Results

Captured parameters

The mentioned parameters refer to the *Essential Ocean Variables* (EOV, e.g., salinity) and other environmental parameters (e.g., relative humidity). Others described observational data that is more complex (e.g., "*numbers of macrolitter items (> 2.5 cm) counted within 3m x 3m sampling stations [...], grouped according to different material categories*") or more general (e.g., "*Raw acoustic data*"). Besides a collection of parameters, the responses revealed further insights proving the need for the use of standards. Several parameters were indicated differently (e.g., "*air temperature*", "*temperature (in situ)*"). Thus, it is unclear whether these parameters refer to the same or a different phenomenon. Furthermore, the parameters naming does not yet follow a certain convention (e.g., "*salinity*" vs. "*Sea surface salinity*" or "*Subsurface salinity*" as defined by the Global Ocean Observing System[1]). Hence, it remains open what kind of salinity is meant, which is a typical semantic issue. Such problems might be solved through human inspection but require expertise. We will address these issues in the chapters on semantic interoperability and data harmonisation.

Spatial and temporal coverage and resolution of the data

Information on spatial/temporal coverage and resolution was provided differently, which is not surprising given the free text response option. The responses are heterogeneous (e.g., "*10 years*" vs. "*since 2007*"), ambiguous ("*high resolution*"), and sometimes inaccurate ("*it is planned to cover beaches from around the world*"). Also, it is not always clear if a response referred to the temporal coverage or resolution (e.g., "*2 weeks*"). Again, these issues are probably caused by the free text and the way the question was asked but they also highlight the need for standards.

Update frequency and estimated data size

The update frequency and estimated data size should be considered together, since a low update frequency can still become an issue with large datasets and a small dataset can become a problem if updated frequently. From the 13 datasets, four (size: "*unknown yet*", "*<1 GB*", "*depending on length of experiment*", "*few KB*") will not be updated and for two datasets ("*<500 MB*", "*5 MB*") there are no update plans yet. The remaining datasets will become updated yearly ("*1 MB*", "*2 MB*"), monthly ("*108687 records*" - no size information), daily ("*240 KB*"), every three hours ("*few MB*"), and every four years ("*<1 MB*"). In one case, the data might be updated in real-time ("*tens of GB to a few TB*").

Derivative data

Only two datasets are derived from another dataset. It would be interesting to further investigate why the others have not derived their dataset from an existing one, e.g.,

---

[1] The Global Ocean Observing System:
https://www.goosocean.org/index.php?option=com_oe&task=viewDocumentRecord&docID=17470

because existing datasets do not exist or are not findable, accessible, interoperable, and reusable (FAIR). Both issues are strong arguments for the MINKE objectives.

Data formats

Fortunately, most participants make use of open data formats, such as plain text files (5 participants mentioned this format), csv files (4), netCDF (3), and xlsx (2). One participant mentioned “excel”, which can be the closed version of an Excel file (xls) or the open format (xlsx). One participant collects acoustic data using .wav files. These numbers are good news for MINKE since achieving platform-independent semantic and syntactic interoperability requires open data formats.

SensorML descriptions and Vocabularies

In most cases (ten), the participants do not yet have SensorML descriptions of the used sensors whereas one has. In two cases, sensors were not used to collect data.
Four participants do not use standard vocabularies, whereas three participants do so but have not mentioned which ones. The remaining datasets are based on the vocabularies from BODC/NERC (including ODIP, SeaDataNet, OceanSITES,) and MMISW.

Tools for data collection and management

According to one participant, observations are uploaded to a Sensor Observation Service. Three participants make use of ERDDAP. Two participants make use of Internet of Things approaches (MQTT, SensorThings), one of them additionally mentioned the Sensor Web Enablement framework without providing further details. Two participants use internal or own storage options (it remains unclear what this means). Three participants remained vague and stated, e.g., “*adapted commercial data base*”.

Calibration

Data from calibration processes or reference devices will be provided in eight cases. In two of the eight cases, the participants could not provide information on the calibration process. The others mentioned, for example, “*standard CTD calibration procedures*”. Two participants provided a link to a calibration report.

Open data and licensing

Four datasets are already openly available under an open license (CC-BY, Open Database License). In seven cases, the institution/partner is the copyright holder but the data can be published openly (two of them after publication, one requires permission from third-party).

3 Relevant Standards for Harmonised Data Sharing

Standardised formats and services are essential to ensure the harmonised preservation of and access to ready-to-use data generated in the context of MINKE (and beyond). In the following, we briefly review relevant standards for metadata and observations as a starting point for our considerations regarding semantic interoperability and quality information.

3.1 Metadata Standards

Data without descriptive information is hardly (if at all) FAIR, i.e., findable, accessible, interoperable and reusable (Wilkinson et al., 2016). Several initiatives invested effort in standardizing this information. For our purposes, we can build on top of two already existing metadata standards.

3.1.1 Sensor Model Language

The OGC Sensor Model Language (SensorML) provides a framework to encode descriptions of sensors and sensor-related processes within XML files. Its main goal is to enhance interoperability, making sensor descriptions understandable by machines and shareable between intelligent nodes (Botts & Robin, 2014). Moreover, additional information related to specific deployments can also be encoded using this standard. Thus, both sensor configuration, measurement operations and contextual information can be defined with the SensorML standard.

This standard is highly flexible and modular as it can describe almost every property related to a sensor or sensor-related process. However, this flexibility and modularity can prove a double-edged sword, as the same feature can be encoded in different ways, increasing the difficulty to generate smart processes capable of interpreting SensorML definitions.

A real-world example of the usefulness of the SensorML metadata standard is the plug and play integration of scientific instruments into data infrastructures. Using this standard detailed metadata about a sensor's characteristics, deployment details, communication protocol, setup routines and more can be encoded in a machine-readable format. Thus, a software tool able to interpret SensorML descriptions could interface a sensor and inject its data (and metadata) into a data infrastructure on-the-fly.

An example of such a SensorML-enabled tool is the SWE Bridge, a cross-platform universal driver. It is a software tool able to retrieve SensorML descriptions, interpret them and set up an acquisition chain accordingly, streamlining sensor data directly into SWE-based services (Martínez, Toma, Jirka & Del Río, 2017). One of its key features is its ability to interface scientific instruments without any previous knowledge by means of SensorML descriptions, regardless of its vendor-specific protocols and interfaces. It can manage sensor communications in almost any format, ranging from plain ASCII communications through serial port to Ethernet high frequency binary streams (e.g., hydroacoustic data) (Martínez, García-Benadí, Toma et al., 2021).

EMSO Generic Instrument Module

An example of the use of SensorML to achieve a sensor abstraction architecture is the real-time data acquisition system for the EMSO Generic Instrument Module (EGIM). The EGIM is a compact-size, multi-sensor module aimed to measure a variety of physical, biogeochemical, biological and ecosystem variables consistently, in a range of marine environments (Lantéri, Ruhl, Gates et al., 2022). This module contains an instrument pack, consisting of several commercial off-the-shelf sensors, which use heterogeneous interfaces and communication protocols.

In order to handle and abstract sensor heterogeneity, the SensorML standard has been used. By using SensorML instrument descriptions their characteristics, protocols and interfaces were unambiguously defined, providing a unified integration layer with the SWE Bridge universal driver. Then, the acquired data was ingested into a SOS service at the shore station. This architecture is depicted in figure 1.

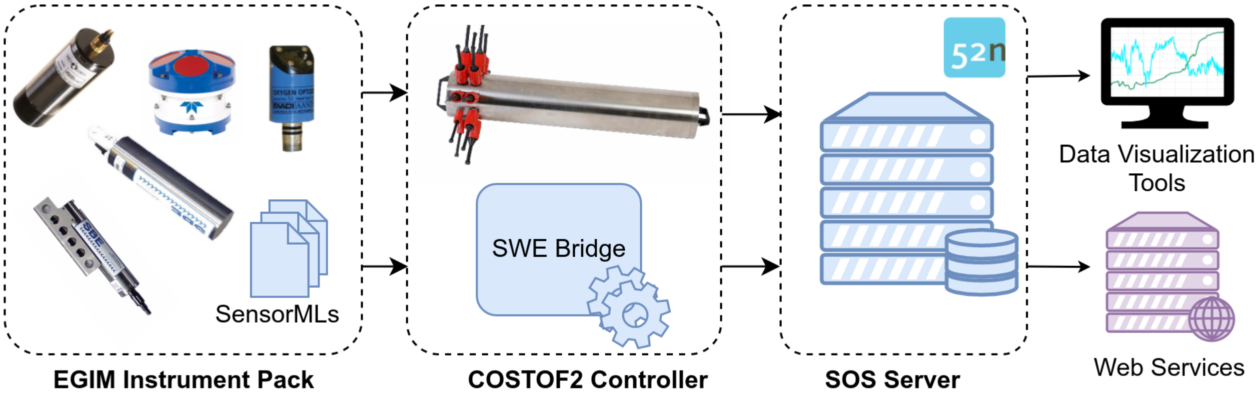


Figure 1 - Architecture of the EMSO Generic Instrument Module (source: UPC)

Real-time Underwater Noise Measurements

SensorML abstraction capabilities go far beyond simple ASCII sensors. Due to its modularity and flexible schema, it is also possible to unambiguously describe much more complex systems such as hydrophones (Martínez, García-Benadí, Toma et al., 2021). Hydrophones metadata is particularly complex, since all internal parameters to their internal acquisition chain are required to properly scale the results. In the following diagram an example on how to encode the whole acquisition chain of a NAXYS Bjørge hydrophone is depicted.

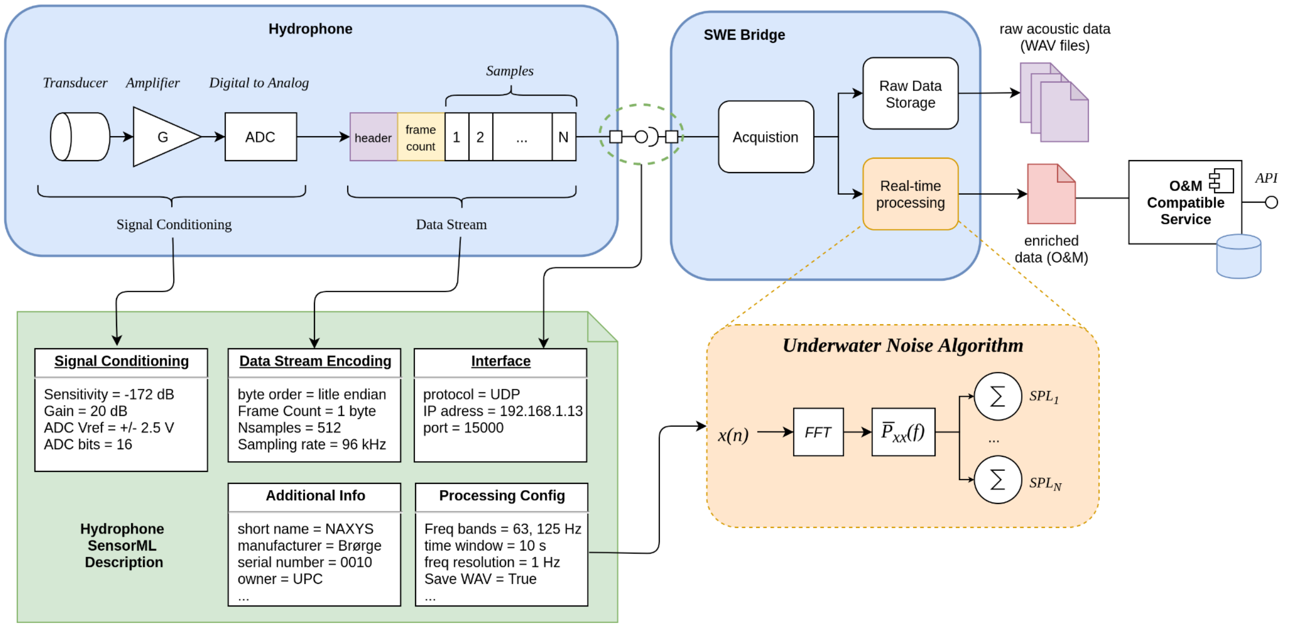


Figure 2 - Schematic overview of the acquisition chain of a NAXYS Bjorge hydrophone (source: UPC)

The green box in the diagram represents a SensorML document with all the information of the signal conditioning stage of the hydrophone, the digital encoding of the data streams as well as the interface and protocol details. Furthermore, the desired processing to be applied can be encoded in a SensorML document. In this particular case a Sound Pressure Level algorithm to detect underwater noise in the third-octave bands of 63 and 125 Hz is configured.

This setup is used at the OBSEA underwater observatory to interface, acquire and process underwater sound data, based on a SensorML abstraction layer and the SWE Bridge software. Results are stored into standardised data services such as SensorThings API and ERDDAP and shared with relevant data infrastructures in the ocean observing community.

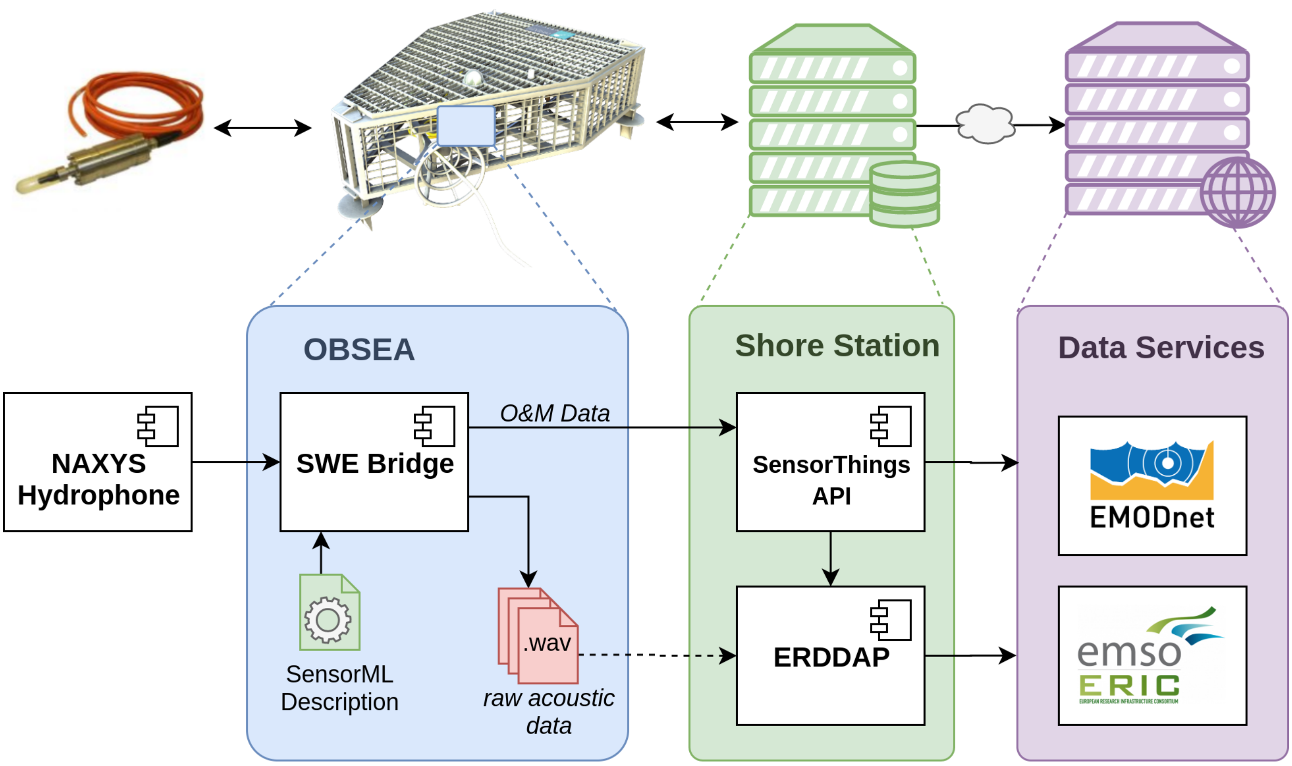


Figure 3 - Overview of the OBSEA data flow (source: UPC)

In summary we expect that SensorML will have a very high relevance for the MINKE project in order to provide metadata about sensors, measurement processes and the generated observation data set. This is expected to provide very useful input to the assessment of the quality of sensor data sets.

3.1.2 ISO 19115

ISO 19115 is a metadata standard to describe geospatial services and data (Habermann, 2019). Besides some typical elements (e.g., title, keywords), the model pays particular attention to spatial and temporal information in the form of geometries. For MINKE, ISO 19115 is relevant because it provides information on granularity and reference measures, which are relevant to interpret and observations quality and uncertainty. Moreover, the standard also incorporates information on the provenance of the data, such as the origin and how it was processed.

3.2 Standards for Observation Data

While the aforementioned metadata standards describe the context of the data, also the observations themselves need to be encoded in standardized way to facilitate interoperability.

3.2.1 OGC/ISO Observations and Measurements

The Observation and Measurements (O&M) standard is another key component of the Sensor Web Enablement framework and comprises a set of elements to encode observations coming from a sensor (Cox, 2013). The description of an observation follows the O&M standard if it provides information on the senor (or algorithm, process etc.) that produced the observation (***Procedure***), the measured parameter, for example, temperature (***ObservedProperty***), the abstracted real-world object to which the observation is associated (***FeatureOfInterest***), and the actual value supplemented by its unit (***Result***). Furthermore, the O&M standard also considers several timestamps to describe an observation's temporal properties, such as the time when the observation was made (***ResultTime***), the time to which the observation applies (***PhenomenonTime***), and during which time interval the observation is valid (***ValidTime***). If an observation is produced by a mobile sensor, it also contains information on the geometry. Finally, information on the observation's quality can be encoded too (***ResultQuality***). However, as this ResultQuality element is highly complex while at the same time missing some basic properties, it is rather recommended to use SensorML as the basis for describing the quality of observation data sets. In addition, the so called O&M ***Parameter*** element may be used to encode observation specific information such as quality flags.

3.2.2 Relevant Technical Guidance from the INSPIRE framework

Besides the standards developed by international standardisation organisations such as the Open Geospatial Consortium (OGC) and the International Organisation for Standardisation (ISO), there are also relevant guidelines on the European Level. This comprises especially the Technical Guidance documents supporting the European INSPIRE Directive which aims at improving the availability of spatial data resources across Europe especially considering environmental aspects.

For modelling and encoding sensor observation data, the Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE (INSPIRE MIG sub-group MIWP-7a, 2016a) offer recommendations on how to encode different types of observation data in an INSPIRE compliant manner. This covers both, the conceptual models as well as encoding examples. For the purposes of the MINKE project, we especially recommend to rely on the conceptual aspects of this document, including a taxonomy of different observation types (e.g., in-situ vs. remote, point vs. coverage measurements). This consideration is valuable in order to derive efficient and extensible data models.

At the same time, the encoding examples of this guidance document are mainly focused on the XML encoding of the OGC Observations and Measurements standard. Thus, here we recommend to rather rely on the more lightweight encodings of the OGC SensorThings API (see 3.3.1) by transforming the XML examples from the technical guidance to equivalent JSON representations.

3.3 Data Access Interfaces

Besides a unified description and preservation of observations, it is also important to ensure standardized access to them.

3.3.1 OGC SensorThings API

The OGC SensorThings API provides an open, geospatial-enabled and unified way to interconnect the Internet of Things (IoT) devices, data, and applications over the Web. It provides a standard way to manage and retrieve observations and metadata from heterogeneous sensor systems.

The SensorThings API standard provides a flexible and powerful framework to encode, archive and share sensor data. Furthermore, its data model (depicted in figure 4) provides a very high granularity, with all the elements and relationships involved in the sampling process defined, achieving a very high data traceability. Especially due to its more lightweight character and high degree of flexibility, the OGC SensorThings API is the recommended interface for enabling interoperable data access within the MINKE project.

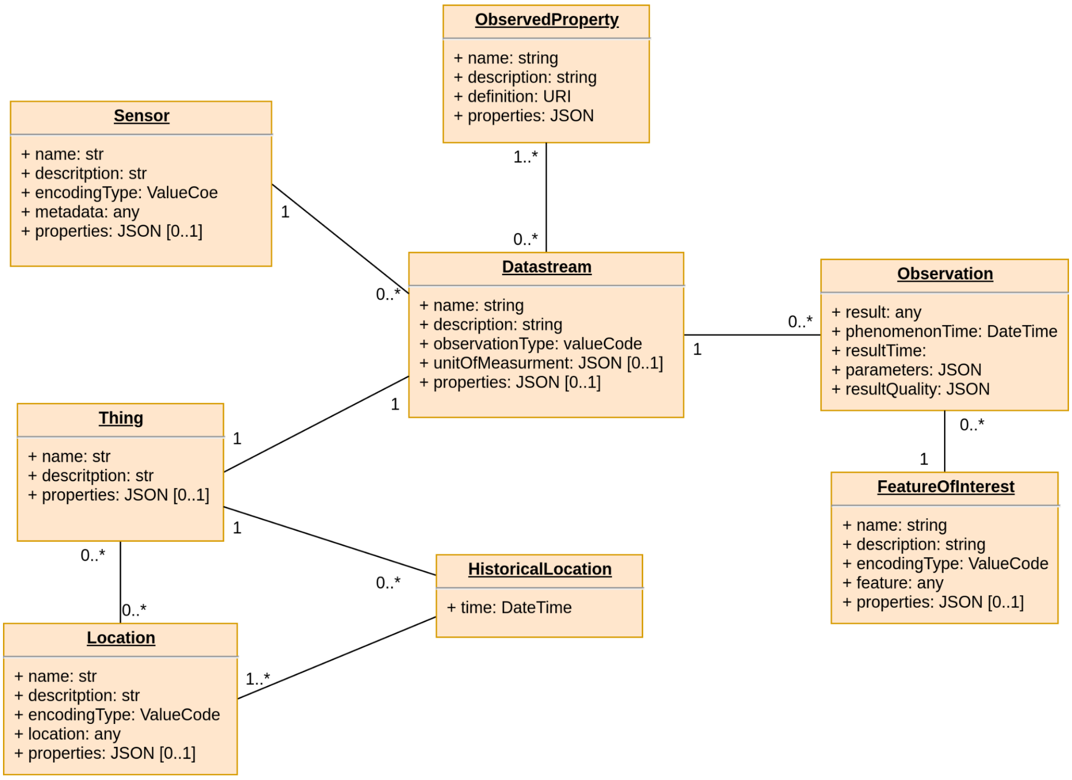


Figure 4 - SensorThings API data model (source: UPC)

SensorThings API Use Case at OBSEA

Due to its benefits, the SensorThings API has been chosen as the core data system of the OBSEA Underwater Observatory, depicted in figure (5). Data streams coming from the sensors are checked by an automated quality control procedure to assign a flag with the estimated quality of the data (see section 5.1.4). The raw sensor data from the sensor is merged with the data quality information, as well as sensor's metadata (sensor model, station, etc.) to generate every observation. Then this rich observation is injected into the SensorThings service. Further data services such as Grafana, ERDDAP or CKAN can access to the data exposed by SensorThings API for formatting, visualisation and processing purposes

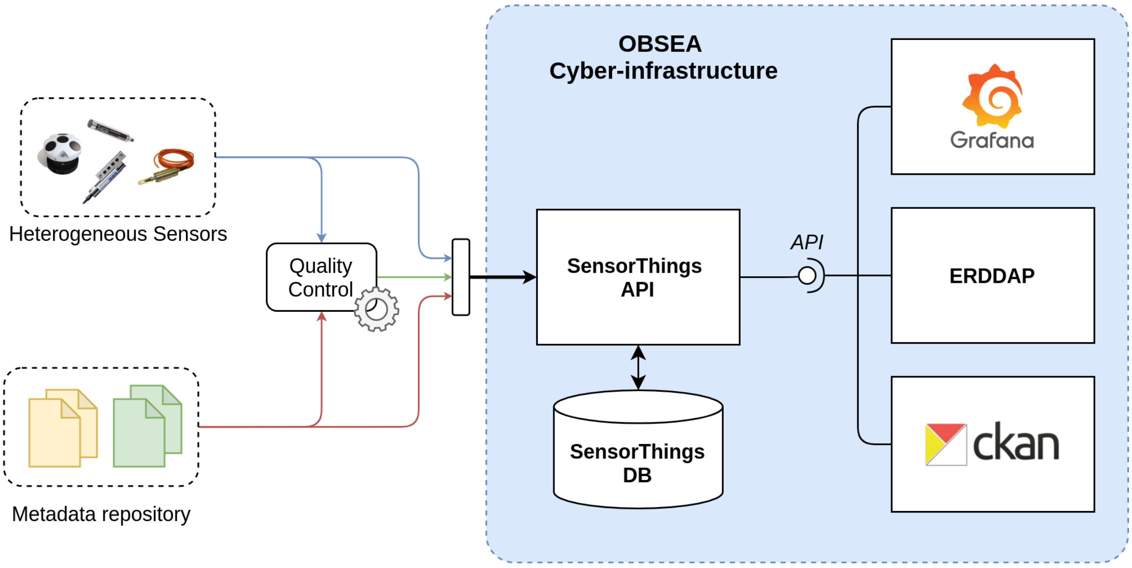


Figure 5 - Core data system of the OBSEA Underwater Observatory (source: UPC)

When an observation is retrieved from the SensorThings API all the quality and traceability information is easily accessible. Thus, for every single data point it is possible to trace back the sensor that produced the observation, the station where it was deployed, the coordinates of the station at that moment, the units of measurements, the parameter measured and also the quality associated with that measurement.

3.3.2 OGC Sensor Observation Service

The Sensor Observation Service (SOS) is the core service of the SWE suite (OGC, 2012). Users (or in general clients) can request information on observations and sensors in an interoperable and standardized way via three core operations: *GetCapabilities* provides information on the SOS (e.g, list of sensors, possible operations); Information on a particular sensor encoded in SensorML can be requested via *DescribeSensors*; *GetObservation* can be used to retrieve information on observations in the aforementioned O&M standard. Other operations are optional, for example, *GetFeatureOfInterest*. Compared to the OGC SensorThings API, the SOS interface is more complex and relies on an older technological approach (XML RPC vs. REST/JSON). Thus, we recommend the use of the OGC SensorThings API instead of the OGC SOS interface.

3.3.3 Relevant Technical Guidance from the INSPIRE framework

With regard to data access interfaces, especially the Technical Guidance on Download Services in the context of the Directive on an Infrastructure for Spatial Information in the European Community (INSPIRE) needs to be considered.

After reviewing the available Technical Guidance documentation, especially the two following guidance documents are considered relevant:

- INSPIRE Good Practice: OGC SensorThings API as an INSPIRE download service (European Commission Joint Research Centre): This Good Practice guide described how the OGC SensorThings API specification (see 3.3.1) may be used for enabling the download of geospatial observation data using lightweight technologies.
- Technical Guidance for implementing download services using the OGC Sensor Observation Service and ISO 19143 Filter Encoding (INSPIRE MIG sub-group MIWP-7a, 2016b): This document described how to use the OGC Sensor Observation Service for INSPIRE compliant data sharing.

Both documents cover similar functional aspects, however, using different technologies. Due to the more lightweight character of the OGC SensorThings API specification, we recommend within the MINKE Project to focus on the INSPIRE Good Practice on using the OGC SensorThings API as an INSPIRE download service.

3.4 Internet of Things Standards

IoT applications have their own needs for which HTTP is often not appropriate. Thus, we also need to consider messaging protocols that may be useful in the context of the MINKE project. While HTTP works in a pull-based mode, the recommended IoT protocols enable a push-based data delivery. This means that data is actively delivered to consumers as soon as it is available. In comparison to this, pull-based approaches such as HTTP require the consumer to actively query for data before it is returned as a response. Thus, push based protocols help to minimize the latency of data delivery.

Please note, that these protocols do not comprise specifications for the data payloads they are transporting. Thus, the data sent via these protocols should follow the recommendations given in sections 3.1 and 3.2.

3.4.1 MQTT

Message Queuing Telemetry Transport (MQTT) is an open OASIS protocol standard organizing the efficient communication between a publisher (e.g., a sensor) and subscribers (e.g., mobile devices) via a broker. It is particularly beneficial in resource-restricted use cases, for example, due to small bandwidths and low resources.

MQTT organises data in so called topics which resemble to channels via which information is send. By using a hierarchical topic structure that also supports several types of wildcards, MQTT offers a high degree of flexibility.

A highly interesting aspect is the inclusion of the MQTT protocol into the OGC SensorThings API specification. On the one hand this integration enables the direct ingestion of sensor data streams into an OGC SensorThings API server. On the other hand, it supports the active delivery of new observation data to subscribers with a minimum latency.

In practice, MQTT is widely adopted in the IoT community and also includes a good support of available implementations. Drawbacks are a slightly lesser focus on sophisticated security and reliability features. However, for most use cases the features of MQTT will be sufficient so that this protocol is recommended for usage in the MINKE project, if push-based data delivery shall be enabled.

3.4.2 AMQP

Advanced Message Queuing Protocol (AMQP) is similar to MQTT but was originally designed for business purposes. Hence, it is less lightweight than MQTT and requires more resources but has a stronger focus on reliability and security.

However, due to its lower adoption rate within the community, we recommend to use MQTT and to rely on AMQT only, if specific requirements towards a more comprehensive set of security features are needed.

4 Semantic Interoperability

The standards recommended in section 3 mainly address the topic of syntactic interoperability. That means that these standards define the structure in which observation data and metadata shall be provided. However, in order to also have interoperability regarding the content of the transmitted data and metadata, a common terminology is also needed. Thus, this section provides a compact overview of relevant vocabularies that are recommended to be used within the MINKE project.

4.1 Parameter Vocabularies

Observed parameters should be described using controlled vocabularies. The two most notable vocabularies are the SeaDataNet Parameter Discovery Vocabulary and the BODC Parameter Usage Vocabulary. The former offers around 120 measurement phenomena and is suitable for parameter discovery in most use cases. The latter follows a more complex semantic model where each concept has the structure “a PROPERTY of an OBJECT in RELATION to a MATRIX by a METHOD” where PROPERTY, OBJECT, RELATION, MATRIX and METHOD in turn are also part of a vocabulary. This allows a very detailed semantic description of the measured property. Through this approach the vocabulary has over 1100 different concepts, which makes it hard for users to find the exact concept needed for their own situation. To solve this, there is a faceted search interface and a parameter thesaurus. More concepts can be created by registered users using the Semantic model Vocabulary Builder. Units of measurement should be described using the BODC-approved data storage units vocabulary.

4.2 SensorML Vocabularies

Through the soft-typed approach of SensorML, controlled vocabularies should be used in the various metadata sections.

SensorML Vocabularies				
SensorML Metadata section	**Description**	**Examples**	**Definition Vocabulary**	**Value Vocabulary**
Identification	Metadata to identify the sensor.	Model name, Serial number, Unique ID	SensorML Identification Section Terms	

Classification	Metadata to categorise the sensor.	Platform Type, Instrument Type	SensorML Classification Section Terms	SeaVoX Platform Categories, SeaDataNet device categories
Capabilities	Properties of the sensor that further qualify the output of the process	Accuracy, Sensitivity, Operating depth	SensorML Capability Section Terms	
Characteristics	Properties of the sensor that do not directly qualify the output.	Physical dimensions, Housing material	SensorML Characteristic Section Terms	
Contacts	Contact information of organisations and individuals.	Manufacturer, Operator, Technical Coordinator	SensorML Contact Section Terms	
History Event Classification	Properties that categorise the history event.	Calibration, Decommissioning, Maintenance	SensorML Event Classification Terms	SensorML History Event Types

Table 1. Overview of recommended vocabularies to provide semantically referenced metadata

The above listed vocabularies already offer a wide range of terms that can be used to describe sensors. For the special use case of describing calibration and test reports for sensors, these vocabularies need to be extended. This will be a continuous process throughout the duration of the MINKE project.

5 Additional Considerations to Handle Quality Information

Users making decisions or developing policies based on data need to be able to evaluate the quality of the data, ideally on the granularity of individual observations. However, determining the quality of an observation is not a trivial task but depends on the use case and requires information on a number of aspects. We checked several resources to distil factors relevant to assess the quality of measurements coming from a sensor. To achieve that, we took into account an exemplary test report checking the functionality of an instrument measuring temperature and conductivity (provided as part of the questionnaire introduced in section 2). In brief, the report describes the “as-received” condition of the sensor (e.g., damaged), then a set of test measurements without calibration coefficients, with old coefficients generated in a previous test, and new coefficients based on the measurement errors. The report concludes with an assessment confirming or refuting adherence to the instrument’s specification. In addition, we considered the data quality flags defined by Devaraju et al. (2015), which range from 0 (= no quality check performed) over 4 (= bad data quality) to 9 (= no data, missing value). Moreover, we incorporated the quality criteria identified by Rodriguez & Servigne (2013), including the quality of the data source (e.g., the sensor’s accuracy), completeness and accuracy with respect to space and time, and consistency between measured and expected values. Finally, we built on top of the data management plan (D3.1) and results from the questionnaire (see 2.1) covering issues such as spatial-temporal coverage of the data collected in MINKE and the use of calibration procedures.

As a result, we identified three main categories (sensor, test report, and observations) for which we describe the relevant metadata elements in the next sections. The goal of the listed metadata is to collect information that help users interpret the quality of datasets on the level of single observations. The idea is not to confront users with a single static value that indicates an observation’s quality, but to provide a rich set of information that can be used in different formulas with varying weightings to calculate a quality value depending on the use case and preferences of a user. Hence, the information is specified considering the use by client applications in different scenarios (e.g., crisis management).

5.1 Overview of Relevant Metadata Elements

The following sections list all the relevant metadata elements we identified related to the sensor and its specification, the test reports describing the sensor’s calibration, and the individual observation. The metadata elements are highlighted in ***bold and italic***. An example of a complete metadata object is available in the supplements in JSON format.

5.1.1 Sensor metadata

The quality of an observation strongly depends on the sensor's properties as declared by the manufacturer, i.e., ***Accuracy*** and ***Precision*** (e.g., ± 0.002 °C), and the range of possible observations (***DetectionLimit***, e.g., -5 to 45 °C). Also, ***BatteryCharge*** and ***StorageLoad*** (e.g., load in %) might affect the quality of an observation. ***MeasurementRate*** and ***TransmissionRate*** (e.g., samples per second) provide information on the temporal completeness, whereas ***Coordinates*** (e.g., latitude/longitude in WGS84) indicate spatial completeness. Temporal completeness is not a quality criterion for a single observation but for a set of observations. For example, a time series missing several observations might indicate a problem with the sensor and thus negatively affect the reliability of an individual observation. Spatial completeness can only be assessed in combination with other sensors. Hence, temporal and spatial completeness are needed to assess the quality of a dataset.

The ***Placement*** of the sensor contains a description of the deployment (e.g., two meters above ground in a valley). The ***QualityLevel*** (e.g., none, checked, adjusted) is inferred from the last test that was carried out to calibrate the sensor. The sensor element also contains a list of ***TestReports*** (see 5.1.2) and ***Observations*** (see 5.1.3). The overall ***SensorUncertainty*** depends on the use case and is calculated based on the elements mentioned above. The outcome of such a calculation can in addition be one of the quality flags introduced by Devaraju et al. (2015).

5.1.2 Calibration and Testing Metadata

This element contains a list of test reports. Each report starts with ***AsReceived***, which describes the condition of the sensor before the test and comprises two aspects. First, information on the ***Physical*** condition comprising the overall ***Condition*** of the sensor (e.g., damaged), ***Photographs*** showing the condition, ***Activities*** carried out to improve the sensor's condition (e.g., repaired), and a more detailed description of these activities (***Workflow***). Second, ***Communication*** includes the ***State*** of the transmission check (e.g., flawed) and further ***Details*** describing the outcome of the check.

A test report also contains a list of conducted ***Tests*** of a certain ***TestType*** (e.g., OldCalibration, i.e., a test conducted with old calibration coefficients and NewCalibration, i.e., a test conducted with new calibration coefficients). Each test entry contains a description of the ***Procedure*** and a ***Date*** when the test was carried out, also showing that sensors tested and calibrated long time ago might generate unreliable observations. The ***AmbientConditions*** element comprises details describing the environmental circumstances during the test by listing the environmental ***Parameter*** (e.g., humidity), the ***Unit*** (e.g., %) of the parameter and the ***Value*** (e.g., 75%). The ***Measurements*** include a list of test observations (***MeasuredValue***), reference values (***ReferenceValue***), and ***Deviations*** between the two supplemented by the tested ***Parameter*** (e.g., temperature) and ***Unit*** (e.g., Celsius). The average error of the test observations against the reference value is provided by ***MeanDeviation***, which makes the sensor pass if the deviation is low enough and fails otherwise (***Satisfactory***). If a sensor comes with ***CalibrationCoefficients***, they are described in the ***Before*** element. New coefficients are provided by the ***After*** element. Further details about the conducted test can be looked up under the URL in the linked ***CalibrationFile***.

5.1.3 Metadata for Observations

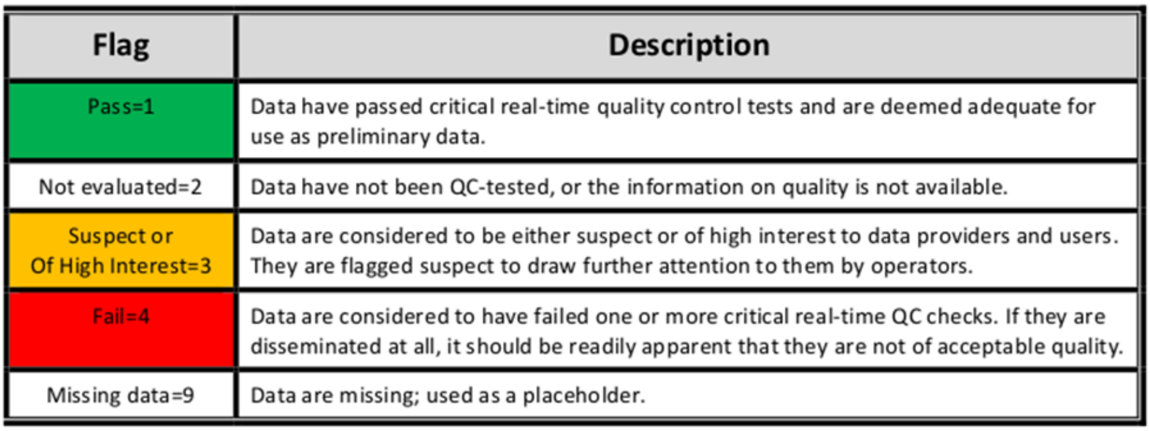
The third element needed to assess an observation's quality is the observation itself. Besides a ***Timestamp***, an observation contains a ***Measurement*** element, including the ***Parameter*** (e.g., temperature), the ***Unit*** (e.g., Celsius) of the parameter and the actual ***Value*** (e.g., 20). Each observation has a ***Validity*** element, e.g., bad if it is inconsistent or beyond acceptable thresholds. Furthermore, information on the processing of the data is provided via the ***DataProcessing*** element, including the ***Level*** with details on what kind of processing was carried out (e.g., raw, adjusted) and, if applicable, the ***Provenance*** describing how it was processed (e.g., using a source code script written in R). The final element is the ***ObservationUncertainty***, which is calculated based on the uncertainty of the sensor (see 5.1.1) and the observation's validity and processing. Again, the result of a such a calculation can be a quality flag (Devaraju et al. 2015).

5.1.4 Metadata for Data Quality Control

Oceanographic data are employed for a wide variety of applications and users. Some applications/users may require that only data of the highest quality be used, and others may seek an indication that a data point is questionable. Some users may prefer the delivery of all data, to be quality controlled using their own criteria. Operators of observing systems may be best suited to determine the quality of their observations and to document their findings by generating metadata to accompany the observations. Information generated by software in real-time about the data quality is referred to as data quality flags, which become an embedded part of the output data stream (U.S. Integrated Ocean Observing System, 2020a).

Within the IOOS QARTOD initiative a number of guidelines to apply QC test to various oceanographic variables have been published over the years. Some of these variables are: pH data, Passive Acoustics data, wind data, in-situ surface wave data, ocean optics, temperature and salinity, dissolved oxygen and in-situ currents among others.

All QC tests provide quality Information for every observation. The quality of each observation is encoded as data quality flags (or simply flag), marking the quality of the data. Following the QARTOD / UNESCO IOC 54:V3 convention, the flag value and definitions are:



Flag	Description
Pass=1	Data have passed critical real-time quality control tests and are deemed adequate for use as preliminary data.
Not evaluated=2	Data have not been QC-tested, or the information on quality is not available.
Suspect or Of High Interest=3	Data are considered to be either suspect or of high interest to data providers and users. They are flagged suspect to draw further attention to them by operators.
Fail=4	Data are considered to have failed one or more critical real-time QC checks. If they are disseminated at all, it should be readily apparent that they are not of acceptable quality.
Missing data=9	Data are missing; used as a placeholder.

Figure 6 - QARTOD / UNESCO IOC 54:V3 flagging scheme (source: U.S. Integrated Ocean Observing System, 2020a)

Quality Control involves follow-on steps that support the delivery of high-quality data and requires both automation and human intervention. QC practices include such things as data integrity checks (format, checksum, timely arrival of data), data value checks (threshold checks, minimum/maximum rate of change), neighbour checks, climatology checks, model comparisons, signal/noise ratios, the mark-up of the data, the verification of user satisfaction, and generation of data flags (U.S. Integrated Ocean Observing System, 2020b).

6 Summary and Outlook

This deliverable has provided guidance on how to handle different aspects of data harmonisation within the MINKE project and beyond. It provides recommendations on relevant standards that shall be used to model and encode:

- Metadata about observation data sets
- Metadata about sensors and data acquisition processes
- Observation data
- Interfaces for data access and delivery

This has been complemented by an overview of relevant vocabularies that shall be used to ensure semantic interoperability through a common terminology within data sets and corresponding metadata.

Special consideration was given to the description of different properties that help to assess the quality of an observation data sets. This comprises not only the description of the data itself but also quality related aspects of data acquisition processes.

We consider this deliverable as a first baseline for the work to be conducted in the MINKE project. Based on the experiences gained during the implementation process we expect further refinements of this guidance. Furthermore, the work on semantic interoperability has until now been focused on the identification of relevant vocabularies. Here we expect that contributions to these vocabularies may arise from the project work. Especially in the context of data quality we expect that further necessary vocabulary entries will be identified.

7 References

1. Botts, M. & Robin, A. (2014): OGC SensorML: Model and XML Encoding Standard 2.0, Open Geospatial Consortium, Wayland, MA, USA. https://portal.opengeospatial.org/files/?artifact_id=55939.
2. Bröring, A., Echterhoff, J., Jirka, S. et al. (2011): New Generation Sensor Web Enablement. Sensors 11(3): 2652-99. http://www.mdpi.com/1424-8220/11/3/2652.
3. Cox, S. (2013): OGC Abstract Specification - Geographic information — Observations and measurements, Open Geospatial Consortium, Wayland, MA, USA. http://portal.opengeospatial.org/files/?artifact_id=41579.
4. Devaraju, A., Jirka, S., Kunkel, R. & Sorg, J. (2015): Q-SOS—A Sensor Observation Service for Accessing Quality Descriptions of Environmental Data. ISPRS Int. J. Geo-Inf. ISPRS International Journal of Geo-Information 4(3). https://doi.org/10.3390/ijgi4031346.
5. European Commission Joint Research Centre. INSPIRE Good Practice: OGC SensorThings API as an INSPIRE download service. European Commission Joint Research Centre. Accessed 16th June 2022. https://inspire.ec.europa.eu/good-practice/ogc-sensorthings-api-inspire-download-service
6. Habermann T. (2019): Mapping ISO 19115-1 geographic metadata standards to CodeMeta. PeerJ Computer Science 5:e174. https://doi.org/10.7717/peerj-cs.174
7. INSPIRE MIG sub-group MIWP-7a. (2016a). Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE - Version 3.0. INSPIRE Maintenance and Implementation Group (MIG). https://inspire.ec.europa.eu/id/document/tg/d2.9-o&m-swe
8. INSPIRE MIG sub-group MIWP-7a. (2016b). Technical Guidance for implementing download services using the OGC Sensor Observation Service and ISO 19143 Filter Encoding—Version 1.0. INSPIRE Maintenance and Implementation Group (MIG). https://inspire.ec.europa.eu/id/document/tg/download-sos
9. Lantéri, N., Ruhl, H. A., Gates, et al. (2022). The EMSO Generic Instrument Module (EGIM): Standardized and interoperable instrumentation for ocean observation. Frontiers in Marine Science, 205.
10. Liang, S., Huang, C.-Y. & Khalafbeigi, T. (2016): OGC SensorThings API Part I: Sensing. OGC Implementation Standard, Open Geospatial Consortium, Wayland, MA, USA. http://docs.opengeospatial.org/is/15-078r6/15-078r6.html.
11. Martínez, E., Toma, D. M., Jirka, S., & Del Río, J. (2017). Middleware for plug and play integration of heterogeneous sensor resources into the sensor web. Sensors, 17(12), 2923.
12. Martínez, E., García-Benadí, A., Toma, D. M. et al. (2021). Metadata-driven universal real-time ocean sound measurement architecture. IEEE access, 9, 28282-28301.
13. Rodriguez, C. G. & Servigne, S. (2013): Managing Sensor Data Uncertainty: a data quality approach. International Journal of Agricultural and Environmental Information Systems (IJAEIS) 4(1). https://doi.org/10.4018/jaeis.2013010104

14. Toma, D. M., Martínez, E., Del Rio, J. et al. (2017). Seamless integration of EMSO Generic Instrument Module into the internet using sensor web components based on OGC SWE framework. In OCEANS 2.017-Aberdeen (pp. 1-5). IEEE
15. U.S. Integrated Ocean Observing System (2020): Manual for the Use of Real-Time Oceanographic Data Quality Control Flags, Version 1.2. https://doi.org/10.25923/w8y6-d298
16. U.S. Integrated Ocean Observing System (2020): Manual for Real-Time Quality Control of In-situ Temperature and Salinity Data Version 2.1: A Guide to Quality Control and Quality Assurance of In-situ Temperature and Salinity Observations. https://doi.org/10.25923/x02m-m555
17. Wilkinson, M., Dumontier, M., Aalbersberg, I. et al. (2016): The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data* **3**. https://doi.org/10.1038/sdata.2016.18

Annex I - Overview of Quality Related Metadata Elements

Component	Metadata Element	Description	Example	How to encode?	XPath
Sensor	Accuracy	Accuracy declared by manufacturer	± 0.002 °C (-5 to to 35 °C); ± 0.01 °C (35 °C to 45 °C)	Intervall	sml:AbstractPhysicalProcess/sml:capabilities/sml:CapabilityList/sml:capability/swe:DataArray
Sensor	Precision	Precision declared by manufacturer	± 0.002 °C (-5 to to 35 °C); ± 0.01 °C (35 °C to 45 °C)	Intervall	sml:AbstractPhysicalProcess/sml:capabilities/sml:CapabilityList/sml:capability/swe:DataArray
Sensor	DetectionLimit	Range of values that can be observed by sensor	-5 to 45 °C	Intervall	sml:AbstractPhysicalProcess/sml:capabilities/sml:CapabilityList/sml:capability/swe:QuantityRange
Sensor	BatteryCharge	Current charge of the battery	90%	Load in percent	sml:AbstractPhysicalProcess/sml:characteristics/sml:CharacteristicList/sml:characteristic/swe:Quantity
Sensor	StorageLoad	Storage occupied by observations	50%	Load in percent	sml:AbstractPhysicalProcess/sml:characteristics/sml:CharacteristicList/sml:characteristic/swe:Quantity
Sensor	Measurement Rate	How often values are measured	0,5	samples per second	sml:AbstractPhysicalProcess/sml:capabilities/sml:CapabilityList/sml:capability/swe:Quantity
Sensor	Transmission Rate	How often values are transmitted	0,1	transmissions per second	sml:AbstractPhysicalProcess/sml:capabilities/sml:CapabilityList/sml:capability/swe:Quantity
Sensor	Placement	Description of the deployment and environment	The sensor is deployed 2m above ground on a meadow.	text	sml:AbstractPhysicalProcess/sml:position/swe:Text
Sensor	Coordinates	Location of the sensor	51.934957,7.6516061	lat/lon (WGS84)	sml:AbstractPhysicalProcess/sml:position/gml:Point
Sensor	Observations	List of observations	[{observations}]	array of observations	sml:AbstractPhysicalProcess/sml:outputs/sml:OutputList/sml:output/sml:DataInterface/sml:data/swe:DataStream
Sensor	QualityLevel	State of the testing	none, quality_checked, adjusted, …, etc.	text	sml:AbstractPhysicalProcess/sml:capabilities/sml:CapabilityList/sml:capability/swe:Category
Sensor	TestReports	List of test reports	[{report1}, {report2},…]	array of reports	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event
Sensor	Sensor Uncertainty	Calculated based on test report + accuracy + precision	*to be specified in the next iteration*	*to be specified in the next iteration*	sml:AbstractPhysicalProcess/sml:capabilities/sml:CapabilityList/sml:capability/swe:Quantity

		+ battery charge + placement etc.			
TestReport	AsReceived	Description of the sensor’s condition before the test	{physical: {<condition etc.>}, communication:{ <state etc.>}}	object	-
TestReport	Physical	Description of the sensor’s physical condition before the test	[{condition: <condition>, photographs: <photos>, activities: <activities>, workflow: <workflow>}]	array of objects	-
TestReport	Condition	State of the sensor's condition	damaged, dirty, modified, testable, ..., etc.	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:Text
TestReport	Photographs	Photos showing condition(s).	photos	photo	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:documentation/sml:DocumentList/sml:Document/gmd:CI_OnlineResource
TestReport	Activities	Activities carried out to improve the sensor's condition	repaired, cleaned, none, ..., etc.	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:Text
TestReport	Workflow	Description of the activities carried out to improve the sensor's condition	We repaired the sensor using a hammer.	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:Text
TestReport	Communication	Description of the communication (transmission?) capability	{state: <state>, details: <details>}	object	-
TestReport	State	Outcome of check	successful, flawed, not checked, ..., etc.	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:Category
TestReport	Details	Description of the outcome of the check	A communications check was conducted	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:Text

			without meeting any problems.		
TestReport	Tests	List of conducted tests.	[{test1}, {test2},...]	array of test objects	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecords
TestReport	Type	Type of the test	AsReceived (Testing of the sensor without calibration coefficients), OldCalibration (Test with old calibration coefficients), NewCalibration (Test with new calibration coefficients).	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='type']/swe:Category
TestReport	Procedure	Description of the test procedure	T Ref. = the calibration bath set-point temperature, measured using the laboratory temperature reference standard.	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='procedure']/swe:Text
TestReport	Date	Date of the test	2022-09-19T05:56:18+00:00	ISO Timestamp	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='date']/swe:Time
TestReport	Ambient Conditions	List of ambient conditions during test	[{parameter: <parameter>, value: <value>, unit: <unit>}]	array of objects	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='ambientConditions']/swe:DataRecord
TestReport	Parameter	Parameter of the ambient condition	temperature, relative_humidity, ..., etc.	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='ambientConditions']/swe:DataRecord/swe:field/*[@definition]

TestReport	Value	Value of the ambient condition parameter	18	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='ambientConditions']/swe:DataRecord/swe:field/*/swe:value
TestReport	Unit	Unit of the ambient condition parameter	celsius	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='ambientConditions']/swe:DataRecord/swe:field/*/swe:uom[@code]
TestReport	Measurements	List of measurement and reference values	[{parameter: <parameter>, unit:<unit>, referenceValue: <refVal>, measuredValue:<mesVal>, deviation:dev}]	object	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='measurements']/swe:DataRecord
TestReport	Parameter	Parameter of the measurement	temperature, relative_humidity, …, etc.	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='measurements']/swe:DataRecord/swe:field[@name='referenceValue']/*[@defintion]
TestReport	Unit	Unit of the reference value	celsius	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='measurements']/swe:DataRecord/swe:field[@name='referenceValue']/*/swe:uom[@code]
TestReport	ReferenceValue	Reference value as the benchmark	20	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='measurements']/swe:DataRecord/swe:field[@name='referenceValue']/*/swe:value
TestReport	MeasuredValues	Measured values	21	array of values	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='measurements']/swe:DataRecord/swe:field[@name='measuredValues']/swe:DataArray/swe:values
TestReport	Deviations	Deviations between measured and reference value	1	array of values	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='measurements']/swe:DataRecord/swe:field[@name='measuredValues']/swe:DataArray/swe:values

TestReport	MeanDeviation	Mean deviation between measured and ref. value	1	text	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='meanDeviation']/*/swe:value
TestReport	Satisfactory	Is the deviation low enough	pass, fail, ...,etc.	text/boolean	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:property/swe:DataRecord/swe:field[@name='satisfactory']/*/swe:value
TestReport	Calibration Coefficients	Calibration coefficients to adjust sensor	{before: <before>, after: <after>}	object	
TestReport	Before	Old calibration coefficients	{a0: -1.154321e-05, a1: ...}}	object	// Old coefficients can be found in the previous calibration event
TestReport	After	New calibration coefficients	{a0: -1.1234e-05, a1: ...}}	object	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:configuration/sm:Settings/sml:setArrayValues[@ref='parameters/calibrationCoeffients']/sml:ArrayValues/sml:value
TestReport	CalibrationFile	URL to calibration file	www. ... .de	URL	sml:AbstractPhysicalProcess/sml:history/sml:EventList/sml:event/sml:Event/sml:documentation/sml:DocumentList/sml:Document/gmd:CI_OnlineResource
Observation	Timestamp	Timestamp of the observation	2022-09-19T05:56:18+00:00	ISO Timestamp	om:Observation/om:phenomenTime
Observation	Measurement	Value of the observation	{parameter: <parametrer>, unit: <unit>, value: <value>}	object	
Observation	Parameter	Parameter of the observation	temperature, relative_humidity, ..., etc.	text	om:Observation/om:observedProperty
Observation	Unit	Unit of the observation	celsius	text	om:Observation/om:result[@uom]
Observation	Value	Value of the observation	20	text	om:Observation/om:result

Observation	Validity	Validity of the observation	good, unevaluated, suspicious, bad (inconsistent, malfunction, erroneous spikes, values beyond acceptable thresholds)	text	om:Observation/om:parameter/om:NamedValue
Observation	DataProcessing	If and how the data was processed	{level: <level>, provenance: <provenance>}	object	-
Observation	Level	What kind of processing was carried out	raw, processed, transformed, derived, adjusted, interpolated, …, etc.	text	om:Observation/om:parameter/om:NamedValue
Observation	Provenance	How data was processed, transformed, derived, adjusted, interpolated, etc.	scource code script	source code	om:Observation/om:parameter/om:NamedValue
Observation	Observation Uncertainty	Depends on SensorUncertainty + DataProcessing + Validity	*to be specified in the next iteration*	*to be specified in the next iteration*	om:Observation/om:parameter/om:NamedValue