



D1.7 TECHNOLOGICAL CHOICES FOR DENSE NETWORKS OF SMALL SENSORS

WORK PACKAGE 1 – NEW SENSOR TECHNOLOGIES: INNOVATION AND SERVICES

LEADING BENEFICIARY: IFREMER

Author(s):	Beneficiary/Institution
Rolin Jean-François	Ifremer
Thomas Loubrieu/Laurent Delauney	Ifremer
Carotenuto Federico	ANAAE / CNR IBIMET
Philippe Dhaussy	CNRS/ENSTA Bretagne
Fadi Obeid	CNRS/ENSTA Bretagne
Delory Eric	PLOCAN

Accepted by: Laurent Delauney (WP 1 leader)

Deliverable type: REPORT

Dissemination level: PUBLIC

Deliverable due date: 30.10.2018/M42

Actual Date of Submission: 17.06.2019

ABSTRACT

Small, generally low-cost sensors, that are deployed in unsupervised networks (or remote locations such as the ocean) are becoming more and more important across RIs and across domain.

Distributed infrastructures require efficient data transmission through a more strict effort on standards for sensor registration, sensor web enablement, sensor-embedded treatment capacities. Network-enabled sensors offer a strong potential. This deliverable will present ongoing initiatives and proves the maturity and security of the smart sensor techniques applied at large scale.

Project internal reviewer(s):

Project internal reviewer(s):	Beneficiary/Institution
Technical Reviewer	Justin J H Buck (NOC)

Document history:

Date	Version
24.10.2018	Chapters for synthesis
21.03.2019	Draft for comments
26.03.2019	Draft for review
02.04.2019	Corrected version
17.06.2019	Accepted by Laurent Delauney

PROJECT SUMMARY

ENVRIplus is a Horizon 2020 project bringing together Environmental and Earth System Research Infrastructures, projects and networks together with technical specialist partners to create a more coherent, interdisciplinary and interoperable cluster of Environmental Research Infrastructures across Europe. It is driven by three overarching goals: **1)** promoting cross-fertilization between infrastructures, **2)** implementing innovative concepts and devices across RIs, and **3)** facilitating research and innovation in the field of environment for an increasing number of users outside the RIs.

ENVRIplus aligns its activities to a core strategic plan where sharing multi-disciplinary expertise will be most effective. The project aims to improve Earth observation monitoring systems and strategies, including actions to improve harmonization and innovation, and generate common solutions to many shared information technology and data related challenges. It also seeks to harmonize policies for access and provide strategies for knowledge transfer amongst RIs. ENVRIplus develops guidelines to enhance transdisciplinary use of data and data-products supported by applied use-cases involving Research Infrastructures (RI) from different domains. The project coordinates actions to improve communication and cooperation, addressing Environmental RIs at all levels, from management to end-users, implementing RI-staff exchange programs, generating material for RI personnel, and proposing common strategic developments and actions for enhancing services to users and evaluating the socio-economic impacts.

ENVRIplus is expected to facilitate structuration and improve quality of services offered both within single RIs and at the pan-RI level. It promotes efficient and multi-disciplinary research offering new opportunities to users, new tools to RI managers and new communication strategies for environmental RI communities. The resulting solutions, services and other project outcomes are made available to all environmental RI initiatives, thus contributing to the development of a coherent European RI ecosystem.



TABLE OF CONTENTS

1) Introduction.....	5
2) New standard protocols and technologies in instrument and data sharing for sensor networks for ocean in-situ sensor systems	6
2.1) Standardisation landscape	6
2.2) European and Global Context	7
2.3) Research Infrastructure Projects.....	8
2.4) Further Activities	11
2.5) Standards Baseline	12
2.5.1) Introduction: OGC Sensor Web Enablement Overview	12
2.5.2) Sensor Web Encodings and Data Models	13
2.6) Sensor Web Interfaces	14
2.7) SWE and Best Practices	19
2.8) Results and discussion	20
2.8.1 – 2.8.10) SensorML Editor (SMLE) Manual	20-40
2.9) Outlook of SENSOR WEB ENABLEMENT.....	40
3) Inter-RI METHODOLOGY TO SUPPORT SENSOR MANAGEMENT.....	41
3.1) Introduction	41
3.2) Present situation.....	42
3.3) Effective tools.....	45
3.4) Standardized interfaces	45
3.5) Conclusion.....	46
4) NETWORK SECURITY OF SENSORS in a cross-domain infrastructure under project. Secure Communication Protocol for Large Number of Distributed Sensors.....	47
4.1) Context of the study.....	47
4.2) Background in computer science	49
4.3) Principles of the Protocol.....	49
4.3.1) Initial Concept	50
4.4) Application	53
4.4.1) Initial Insecure Model	53
4.4.2) Secure Model	54
4.5) Robustness	55
4.6) Experimentations	56
4.7) Achievements	57
5) Conclusions.....	58
5.1) Limitations	58
5.2) Main RESULTS	59
6) References	60
7) Acronyms.....	62

Technological choices for dense networks of small sensors

1) INTRODUCTION

The Task 1.4 “Network enabled sensors” plays the role in ENVRIPLUS to show that instrumentation technology and deployment methods (Addressed by WP1 as a whole) must include the interface between sensor and Research Infrastructure systems. Each sensor interface is important to gain data quality and ensure cost efficient operation. It includes electronics and the first software layers but also innovative technologies ensuring the management of connected objects.

The previous report D1.6 “RESULTS AND RECOMMENDATION FROM THE COMPARISON EXERCISE OF SENSOR EMBEDDED PROCESSING PRACTICES” underlined some results from the experience agronomy, air pollution monitoring and marine instrumentation: a) The technology of small sensors integrating digitalization and data communication is now mature for the requirements of Environment Research Infrastructures. b) Their networking in large numbers is still at the level of prototype deployment or first operational operation. c) The link with metrology is unavoidable to provide quality standard of sensor output and address homogeneity of data sets from many sensors. d) In marine instrumentation, the sensors are not used yet as “dense” networks but the contribution to ENVRIPLUS shows that the specifications of several Marine RIs are well determined and provide a clear vision of the key topics (technology and organization) to be addressed. e) in the terrestrial domain, it has been possible to demonstrate several cases with many sensors although they are still dependent on the technical skill of the research teams.

An important input to Task 1.4 in ENVRIPLUS project has been to orientate one of the tasks of WP3 not only to the communication issues but thanks to a wider vision by the partners of this task, lead by MARUM, to the data interface and the sensor web enablement. (see Deliverable D3.3 “*Report for best practices on robust telecom/data transmission* » - Huber et al.)

The management of ENVRIPLUS has been able to address the potential risk to lose the opportunities of synergies between Work Packages. Since Month 16, a Use Case of Theme 2 has been identified to provide a support by Data Base development engineers to the sensor interface topic addressed in the Task 1.4. It is entitled “Use case TC4 **Sensor registry**” (Deliverable 9.1 section 3.3) and can be as well consulted up to a video available at <https://www.youtube.com/watch?v=4QxTZ2iiznk&feature=youtu.be>

In the deliverable hereunder, we can benefit from these inputs and propose advanced studies which were seen as critical for the design and operation of RIs based on sets of sensors. Ownership cost is the key indicator. Such studies are not only valid for one RI but also for clusters of RIs, RI Domains or cross Domain. The **New standard protocols and technologies** are addressed for ocean in-situ sensor systems (Marine Domain). The **Methodology to support sensor management** is proposed already for a group of RIs in the Marine Domain; it uses analysis and results of the Use case “Sensor Registry”. The **Network Security of sensors** is proposed for a cross domain application Marine/Solid Earth more precisely addressed by the Task 4.3 of ENVRIPLUS: “*Solid Earth-marine common operation of platforms – benthic stations* ».

To also highlight and further promote to other non-marine RIs the progress made on the standard registration of sensors for web-enablement and particularly the efforts made to improve user-friendliness, this document embeds relevant parts of a new sensor registry user interface, described in detail in FixO3 Deliverable D2.10. This enables the reader to get all necessary information out of one document.

2) NEW STANDARD PROTOCOLS AND TECHNOLOGIES IN INSTRUMENT AND DATA SHARING FOR SENSOR NETWORKS, for ocean in-situ sensor systems

The section of this report includes the description of the standardisation landscape with respect to sensor interoperability, applied to in-situ ocean observations but the concepts and tools are applicable/transferable to other domains. A rationale is derived that promotes the use of a set of standards and leads to the development of a user-friendly interface to enable the implementation. The results are presented by means of a description of the developments performed in collaboration with other initiatives and building on projects (e.g. ESONET, I3 FixO3, NeXOS) and current projects like AtlantOS. The description meant to be pragmatic and therefore shows similarities to a user-manual format, so that the community can have clear instructions for proper use of the tools, while at the same time “see” the actual result. An implementation example has been performed as a joint mission of EnvriPLUS and AtlantOS in 2018 in Gran Canaria [Toma, D.M.2018],[Delory, E., 2019]. We acknowledge herein the contribution of the following actors in the design, implementation and testing of the technologies: PLOCAN, Ifremer, NOCS, 52°North, UPC, SLR Consulting, Geomar, OGS, MARUM and HCMR. As the report highlights, the presented tools and technologies are applicable to non-marine sensor systems as the fundamental concepts, resulting formats and services (except for some semantics) are field-agnostic.

2.1 Standardisation landscape

Ocean observations data sets are often collected on a project-basis and, when available, are not always interoperable, and therefore difficult to exploit in broader contexts. Fragmentation, availability gaps, duplication, identification problems, access and usability are common issues. In response, the spatial data infrastructure (SDI) approach is envisioned for the development of marine sensor networks to create environments that enable users to access and retrieve spatial datasets in an easy and intuitive way. Implementing marine sensor networks using an SDI approach will ease the use and sharing of spatial information and services to support spatial-temporal decision making for multiple purposes [Strain, L., et al., 2006] [Bröring, A., Stefan B., 2010]. Other benefits are the improved accessibility to data and interoperability between data sets. However, interoperability can only be achieved through extensive use of international standards. They specify regulations for data access, content, and exchange. The most important international standards promoted in the last years for the development of SDIs are: the set of standards developed by Open Geospatial Consortium (OGC), the geographic information standards developed by International Organization for Standardization (ISO) and the IEEE 1451 set of smart transducer interface standards developed by the Institute of Electrical and Electronics Engineers (IEEE) Instrumentation and Measurement Society’s Sensor Technology Technical Committee.

OGC (e.g. Sensor Web Enablement DWG, SWE-IoT SWG, MetOceans DWG)

The Open Geospatial Consortium (OGC) is a non-profit de facto international standards body for geographic information. The new standards developed by OGC represent a conceptual model for Observations and Measurements (O&M) [Botts, M., et al., 2008]. This standard describes how an *observation* is an action whose *result* is an estimate of the value of some *property* of a *feature of interest*, obtained using a specified *procedure*. The O&M conceptual model has been substantially developed since 2002 through such initiatives (OWS-1.2 and OWS-3). It was finally approved in 2008 as version 1.0 for publication by the OGC, and was progressed as version 2.0 to the ISO standard 19156. The abstract O&M model may be applied across the spectrum of sensor applications and deployments, and provides a framework for building exchange standards and service interfaces for accessing sensor data and contextual information. For example, the Sensor Observation Service (SOS) provides a Web service interface for retrieving filtered observations or related information (feature-of-interest, sensor parameters, observation results). Individual sensor observations may be aggregated within one service into combined 'observation offerings' and multiple services may be federated into single access points. The Sensor Model Language (SensorML) is an XML language for describing observation procedures and sensor types. Other related standards include the Sensor Planning Service (SPS) for tasking and scheduling observation

requests with sensor systems (e.g. by satellite remote-sensing instruments), and eventing components for setting up notification subscriptions for specific sensor events.

Within Europe, the Directive (2007/2/EC) established the 'Infrastructure for Spatial Information in Europe' (INSPIRE) to integrate environmental data across all member states. The 'Copernicus' partnership between the European Commission and the European Space Agency will establish core operational services (e.g. ocean forecasting, land cover monitoring, emergency response) for the global environment and civil security. These three global-scale initiatives all promote the use of OGC standard information models and network services for integrating sensor data - both in situ and remotely sensed (spaceborne and airborne).

ISO

The responsible for the ISO geographic information series of standards is the ISO/TC 211 Geographic information/Geomatics [ISO 19115, 2011]. ISO/TC 211 establish a structured set of standards for information concerning objects or phenomena that are directly or indirectly associated with a location relative to the Earth. These standards are collectively referred to as the ISO 19100-series. These standards specify for the field of geographic information different methods, tools and services for managing (including definition and description), acquiring, processing, analysing, accessing, presenting and transferring such data in digital/electronic form between different users, systems and locations.

The overall objectives of ISO/TC 211 are:

- to increase the understanding and usage of geographic information;
- to increase the availability, access, integration, and sharing of geographic information;
- to promote the efficient, effective, and economic use of digital geographic information and associated hardware and software systems;
- to contribute to a unified approach to address global ecological and humanitarian problems.

IEEE

The IEEE 1451 Smart Sensor Interface Standard (superseded by ISO/IEC/IEEE 21451-7-2011) provides a common communications architecture with sensors over different communication protocols at the physical level. Different protocols are addressed by different branches of the standard, for example, 1451.2 for RS232, I2C, and SPI; 1451.4 for analog sensors, 1451.6 for controller area network (CAN), and so on [IEEE1451, 2007]. This standard has not yet been widely used, especially in marine sensors, and its adoption is hampered by a lack of software tools for implementation. However, it has some capabilities that may be useful in marine networks. Though there have been many incremental steps toward instrument integration as it applies to marine sensor networks, there has not been a standards-based architecture that offers practical end-to-end plug & work capability. Hence, the architecture we propose provides a practical end-to-end solution using OGC-SWE standards to implement distributed and geospatially referenced sensor networks.

2.2) European and Global Context

INSPIRE

The Infrastructure for Spatial Information in the European Community (INSPIRE) directive is an important European legislative foundation. It defines the foundations of a spatial information infrastructure on a European level in order to strengthen and facilitate the exchange of geographic information. The INSPIRE Directive entered into force on 15th May 2007 and is currently in the process of implementation.

The INSPIRE framework comprises several elements regulating the exchange of geographical information. On the one hand it defines obligations for public administration which data has to be made accessible in a INSPIRE compliant manner. On the other hand, it specifies how this data shall be provided. These specifications include both, the specification of data/metadata models as well as service types for discovering and accessing these data sets.

Currently a process is ongoing to enhance the INSPIRE technical guidance to better support the provision of observation data. In the upcoming years several data topics defined in the INSPIRE Technical Guidelines Annex II and III will need to be delivered by responsible agencies in a INSPIRE compliant manner. This will also comprise oceanographic geographical features and sea regions. In the current process a proposal is in development describing how the OGC SOS can be incorporated into the INSPIRE framework as a so-called Download Service (i). Thus, to ensure INSPIRE compliance of the Sensor Web architecture, it is important to consider these activities when selecting an interface for the provision of observation data.

Furthermore, INSPIRE relies on the ISO/OGC O&M standard for the modelling and encoding of observation data.

GEOSS

The Global Earth Observing System of Systems (GEOSS) is a global initiative with the participation of more than 70 countries. The aim of GEOSS is to facilitate and strengthen the exchange of earth observation data to support the work on topics such as the management of disasters, human health, water supply, climate change, biodiversity, and the protection of ecosystems (including marine ecosystems).

For several years, there have been ongoing activities to agree on standards for the exchange of relevant data sets. This includes for example also several workshops and discussions in the field of Sensor Web. Within GEOSS, a framework such as the OGC SWE framework will be an important means for the provision of in-situ measurement data.

Several test beds (the so called GEOSS Architecture Implementation Pilots (AIP)) have been conducted to evaluate and demonstrate potential technologies to implement GEOSS. During these test beds the use of OGC SWE standards has been shown. Furthermore, documentation has been developed to offer data providers guidance how these standards could be applied in the context of GEOSS.

Finally, European projects such as EO2HEAVEN and GEOWOW [GEOWOW, 2014] have addressed the topic of observation data in GEOSS. Also here, SWE-based software components (e.g. the 52°North SOS) and concepts were developed.

Thus, it can be concluded, that the implementation of SWE standards will ensure an alignment with GEOSS activities.

Copernicus

Copernicus, formerly known as GMES (Global Monitoring for Environment and Security), is a European Programme that aims at establishing European capacity for Earth Observation. It addresses both, in-situ as well as remote sensing observation data. At the same time it is a European contribution to GEOSS.

Copernicus covers the following thematic domains: land, marine, atmosphere, climate change, emergency management and security. For these topics, Copernicus has the aim to deliver relevant observation data as well as services based on these data sets to policy and decision makers.

In the context of Copernicus, the MyOcean activities are of special relevance. This project (which consists of two subsequent projects) aimed at an integrated pan-European capability for ocean monitoring and forecasting.

2.3) Research Infrastructure Projects

13 EUROWFLEETS 2

EUROWFLEETS2 (ii) (New operational steps towards an alliance of European research fleets) follows the EUROWFLEETS 1 project. The main goal of these two projects is a collaborative coordination effort of European Research Fleet owners.

Before the EUROWFLEETS project, the European marine research had a broader fragmentation and a lack of cohesion and strategic vision. EUROWFLEETS 1 started in September 2009 with the aim to bring the existing European Fleet owners together and strengthen their coordination. The main objectives of this project were:

- agreement on a common strategic vision for the European research fleets
- development of a way that research vessels are operated and their interoperability capacities
- promotion of interoperable and standardized research tools for the vessels and underwater vehicles

EUROFLEETS 2 built up on the achievements of the preceding project. It started in March 2013 and had a duration of 4 years. This project increased the cooperation and coordination structure, facilitated the access to highly performing research vessels to a larger community of researchers, and created more links with industry. The main objectives in this project phase was:

- increase the participation of research vessels both the global/ocean and the regional
- integrate a common polar in the strategic vision of EUROFLEETS 1
- promote the exchange of mobile equipment on board of European research vessels
- larger and ambitious research missions by coordinate multi-vessels experiments
- operational experimental tests to demonstrate the interoperability of the European fleets

Within the EUROFLEETS Infrastructure, now supported by a third EC funded project, the OGC SWE technology (for example, SensorML is used to create Ship Summary Reports) is considered for data exchange purposes. This is in line with the aims of the Sensor Web approach.

ESONET NoE

ESONET (iii) stands for European Seas Observatory NETwork and is a scientific and technical network for deep sea observations. It was funded for 4 years by the European Commission in the 6th Framework Programme (FP6). It promoted the implementation and the management of a network for long-term multidisciplinary ocean observatories in deep waters around Europe.

An ESONET observatory is defined as a deep sea station with marine sensors that are linked to the shore by acoustic or cable connection and provide the measured data in near-real time. These oceanographic and climatological data sets are offered in a high frequency. The deep sea observatories are complementary options to satellite images, research vessels or moorings. This approach has the following advantages regarding the previously mentioned options:

- really long-term observations (for at least 10 years)
- monitoring throughout the water column
- the power supply for the measurements is managed with cables from the shore

In ESONET WP2 (Standards and Interoperability), led by Christoph Waldmann (MARUM), a specific activity has consisted in the development of a first version of a sensor registration interface, based on the SensorML 1.0.1 standard. The tool was developed by dBscale Technologies with contributions from Marum, Ifremer and 52North. 52North took on the developments to redesign a new interface, meant to be user-friendly and most notably open-source, based on the latest version of the standard (SensorML 2.0) – this work.

SEADATANET

The SeaDataNet (iv) (pan-European infrastructure for ocean and marine data management) project provides an efficient distributed Marine Data Management Infrastructure. This infrastructure manages, indexes, and provides access to large and diverse marine and ocean datasets derived from in-situ and remote, sea and ocean observations.

The European marine observing system is highly fragmented and the data is measured using various sensors on board of research vessels, submarines, air planes and satellites. The measured physical, geological, biological or chemical parameters are neither easily accessible nor standardized. SeaDataNet is based on an interconnected network of data centres and provides the data in a unique virtual management system. Thus, the objective in the SeaDataNet project is to provide the collected data in online integrated databases with a standardized quality. The access to the in-situ measurements and metadata is offered by a unique portal and was implemented in the first phase of this project.

The implementation of a Common Data Index (CDI) metadata service (Sea-Search project) leads to a linking between discovery and delivery of datasets. A CDI interface provides a filter mechanism (e.g. availability, geographical) to specialise the search results. To define an extended metadata format to support the CDI SensorML profiles are used to describe instruments and sensors. Also, the O&M standard comes to the front to adapt the marine observation data.

The objective in the second phase of SeaDataNet was to upgrade the current SeaDataNet infrastructure to an operational robust and state-of-the-art pan-European infrastructure. This second phase started in October 2011 for a duration of 4 years. The second phase has been fulfilled by:

- setting, adopting and promoting common data management standards
- realizing technical and semantic interoperability with other relevant data management systems and initiatives on behalf of science, environmental management, policy making, and economy

SeaDataNet RI is now supported by SeaDataCloud project where the implementation of the standards into a Data Cloud is addressed.

GROOM

The GROOMv (Gliders for Research, Ocean Observation & Management) project addresses a study to evaluate the requirements to set up a sustainable European glider infrastructure. This project has started in October 2011 with a duration of 3 years. The development of the glider infrastructure should create continuous observations of individual operating glider as well as fleets of gliders. The goal is to coordinate the operations with benefits for both fundamental marine research and operational oceanography. The study should reveal that a close coordination of a distributed architecture of “gliderports” around the European seas and overseas is required. In combination with existing observing systems leads this to a better value of money.

The GROOM program focuses on the following points:

- integration of gliders into the existing global and regional/coastal ocean observing system
- raising issues by this infrastructure regarding the Law of the Sea and maritime traffic
- strategic location network with coordination of existing observation activities
- use existing data management infrastructure framework to collect and publish valid and quality-controlled datasets
- support an open access option to the glider data to enhance the educational view on the oceans and the importance for the climate and resources

Standardisation of sensor interface on board gliders has been attempted by US scientific users with a Puck solution but then stopped by the manufacturers. A new attempt has been successful in Europe thanks to NeXOS project.

NeXOS (vi)

To achieve a maximum level of interoperability, it is important to establish application profiles for data exchange standards. NeXOS developed such profiles for the ocean community since within its consortium the domains of marine knowledge and technical expertise merge. These profiles will be based on existing work, such as has been started in the ESONET project.

They have been designed as best practices for marine sensor network providers and submitted by NeXOS to the OGC standardization process. In addition to the development of profiles, NeXOS contributes with practical experience and evaluation mechanisms to the next evolution of SWE specifications, which will shape SWE 3.0. In particular, NeXOS addressed specifications to improve the processes of marine data acquisition. Utilization of the standardized SWE specifications and development of the required profiles, will allow seamless integration with existing international initiatives such as GEOSS and Copernicus.

X-Domes – Earthcube

Short for “Cross-Domain Observational Metadata for Environmental Sensing”, X-Domesvii objective is the development of a standards-based description of environmental sensor metadata, including provenance. Leveraging existing relationships with large National Science Foundation (NSF) -funded data management programs, EarthCube building blocks and working groups, and environmental sensor manufacturers and consortia, X-Domes will establish a community of sensor manufacturers and other stakeholders to provide a unifying approach to describing sensors and observations across geo-science domains. Built on an existing sensor metadata model that references registered, standards-based vocabularies, the X-DOMES pilot project will provide a suite of tools, built upon community-adopted standards of the Open Geospatial Consortium (OGC) and World Wide Web Consortium (W3C) to demonstrate and facilitate the generation of metadata documents that are discoverable and accessible on-line and/or directly from onboard sensor descriptions. The project will also demonstrate mechanisms to associate the data with the metadata through standards-based web services. With vendor-ready tools implemented throughout a broad-based community, the X-DOMES Network will lay the foundation for the development of and adoption of interoperable access to much needed content-rich sensor metadata. Considered as a parallel US initiative to the work described in this report, X-Domes further demonstrates the growing interest in the development of tools for the implementation of the OGC SWE standards, and here more specifically the SensorML standard specification for encoding sensor metadata.

EMSO

EMSO (viii) (European Multidisciplinary Seafloor and water column Observatory) is a European infrastructure. This network of specific sites off European coastline includes deep sea cabled and open-ocean observatories that monitor environmental processes, real-time and delayed-mode. A unique organisation at European level managed these long-term observations related to ecosystem and global changes. EMSO decided to implement core standards of the OGC, such as Catalogue Service for Web (CS-W). Besides this, efforts to use the SWE standards SOS and O&M were undertaken for the EMSO Generic Instrumentation Module (EGIM) [Lanteri, N; 2017] and are planned in the EMSOLink project as a core standardization topic.

2.4) Further Activities

Other Initiatives (e.g. OCEANSITES, ODIP, IOOS)

OceanSITES (ix) is a system of worldwide deep water stations. It relies on long-term measurements and monitors the full depth of the ocean down to 5000 meters including sea-air interactions. Since 1999, the observations are collected by moorings and research vessels and satellite telemetry enables near real-time access to the produced data. The main observation parameters are meteorology, physical oceanography, water transportation, biochemistry carbon cycle relevant parameters. Besides satellite imagery and other in-situ observations OceanSITES is an inherent part of the Global Ocean Observing System (GOOS). The focus in this project is the gathering of measurement data, so the gathered data is provided in a self-developed NetCDF (Network Common Data Form) data set format. Besides this, the sensor metadata descriptions are based on the OGC SensorML standard.

As ocean and marine data are recognised as valuable resources which have a high cost of acquisition the main goal of Ocean Data Interoperability Platform (x) (ODIP) is to provide easy access of oceanic data across scientific domains and international boundaries. The major involved organisations are the EU, US and Australia and the project is funded within EU FP7 for a duration of 36 months from October 2012 to September 2015. International workshops under the organisation of this platform raise the development to reach the goal of interoperable used datasets. Another objective is to harmonise the diversion in the regional systems. A prototypical implementation is planned. This prototype should adopt the main

standards of the OGC SWE working group, which are the encodings to describe the provided data sets (SensorML and O&M) as well as the SOS protocol to provide access to modelled data sets.

The Integrated Coastal and Ocean Observation System (ICOOS) Act of 2009 manifests the development of the U.S. IOOS (xi) (Integrated Ocean Observing System). This project is focused on an easier and better access of measured data in the ocean, the coastal and the Great Lakes region. The IOOS is composed in a national-regional partnership, which works on new tools and forecasts to improve safety, enhance economy and protect the environment. With the information and data which is collected in retrospectively and near real-time the objective is to improve the prediction of coastal events, for example storms, wave heights and sea level changes. For the implementation of the self-defined goals of interoperable data access the IOOS uses the Sensor Observation Service standard with small modifications for their needs.

Practical applications

Besides the research projects mentioned in the previous sections, the Sensor Web technology, and especially the OGC SWE framework has already been applied in various domains. The following list gives a short overview of project examples:

- German Federal Waterways and Shipping Administration: SOS servers and clients for providing/visualising a broad range of hydrological measurement data
- European Environment Agency (EEA): Collection of air quality data from member states and provision of these collected data sets in a standardised manner via the OGC SOS
- British Antarctic Survey: Provision of meteorological measurement data from Antarctica via a SOS server
- The U.S. Integrated Ocean Observing System (IOOS): Operated by the US National Oceanic and Atmospheric Administration (NOAA); provision of oceanographic measurements via SOS servers.
- IRCEL-CELINE (Belgian environment agency): Provision of air quality data through the SOS standard
- Barcelona Port Authority: Access to ship tracking data via a SOS server

This list shall only provide a general overview of possible SWE application. It illustrates, that SWE is not only relevant for oceanography but for a broad range of domains dealing with environmental observations. As a result, the usage of SWE standards will not only be in line with oceanographic projects. In addition it will also ensure the interoperable data exchange with other environmental research fields.

2.5) Standards Baseline

2.5.1) Introduction: OGC Sensor Web Enablement Overview

The Open Geospatial Consortium (OGC) comprises over 400 companies, governmental agencies and universities and acts as a non-profit organisation. The main goals are the development of standards for data models and web services in a spatiotemporal area.

The Sensor Web Enablement (SWE) domain working group is a working group of the OGC which develops standards for sensor data and metadata in the geospatial web. Generally, the standards in the SWE framework can be divided in information and interface standards. The information subgroup defines standards for data models and sensor web encodings while the interface subgroup are responsible for the different sensor web service interface specifications.



- Data Models and Encodings [Robin, A., 2011]:
 - ISO/OGC Observations and Measurements models the gathered measurement data
 - OGC Sensor Model Language describes the sensors and processes which have acquired a certain observation data set (→ provision of metadata)
- Service Interface:
 - OGC Sensor Observation Service [Echterhoff, J, 2011]: interoperable access to measurement data and sensor metadata
 - OGC Sensor Planning Service: interface to send tasks and configurations to sensors
 - Event Services [Echterhoff, J, 2008]: subscription to defined events and alerting if the defined events are detected

ISO/OGC Observations and Measurements (O&M)

- the process (e.g. a sensor) is constituted by the **procedure** which has performed a specific observation
- the **observed property** refers to the parameter which is observed (e.g. water level)
- the **feature of interest** stands for the geospatial object (e.g. a river section) in the real world for which the property is observed
- and the observation's **result** which is the value of the measurement (in this case a water level value in centimetres)

Core to this work, the OGC Sensor Model Language (SensorML) is defined to describe the sensor or process metadata. Similarly, to O&M the SensorML 2.0 standard [Botts, M, et al., 2014] describes the processes by data models and with the help of an XML encoding. A process is either a measurement procedure or processing of previously gathered data. A SensorML document may comprise for example the following parameters:

- 

- **identification:** identifiers to ensure a unique reference of the sensor
- **classification:** classifiers for discovery
- **validTime:** time range, for which the sensor description is valid
- **inputs:** inputs for the process
- **outputs:** resulting outputs to the process
- **characteristics:** additional properties of the process that do not further qualify the output values (e.g. component dimensions, battery life)
- **capabilities:** properties which clarify or qualify the measurement process (units for the output, observed area)
- **contact:** institution, which operates the described sensor instance

The characteristic parameter may also describe for example the temporal availability and the position of the described process or sensor.

2.6) Sensor Web Interfaces

OGC Sensor Observation Service

The OGC Sensor Observation Service (SOS) is the best known Web service of the SWE working group and is manifested in SOS 2.0 standard [Bröring, A., 2012] [Jirka, S., 2011]. The SOS delivers pull-based sensor metadata and associated data. This service works as middleware between clients and a measurement archive or sensor network. Through the SOS, clients can access the data of heterogeneous data sources via a standardized interface. Requested measurements are encoded in O&M while sensor is represented by SensorML documents. The operations in the SOS are divided in the core functionality and extensions. The core operations of the SOS interface are:

- **GetCapabilities** delivers the metadata to the service instance
- **DescribeSensor** describes the metadata for sensors or processes
- **GetObservation** delivers the measurement data

The transactional extension allows the insertion of new sensor metadata (**RegisterSensor** operation) and measurements (**InsertObservation** operation).

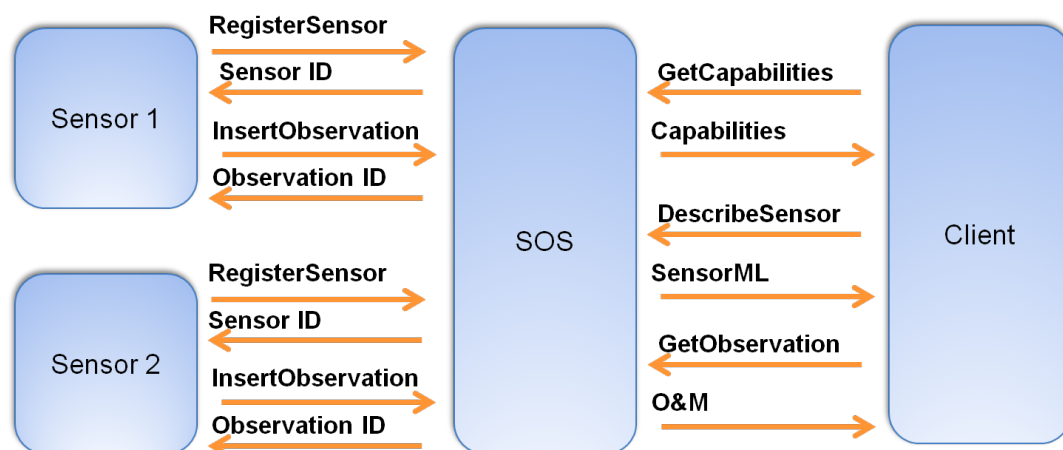


FIGURE 2.2: WORKFLOW FOR INTERACTING WITH SOS SERVERS

Figure 2.2 shows two sensors which communicate use the transactional operations to push their measurement data into a SOS server. On the consumer side a client first requests the metadata about the service (GetCapabilities) and then of the individual sensors. Through the GetObservation operation the client finally obtains the measurements of the sensors.

OGC Sensor Planning Service

The OGC Sensor Planning Service (SPS) [Simonis, I., Echterhoff, J., 2011] offers an interface through which sensors can be tasked or parameters of sensors can be modified (if supported even at runtime). Like the

SOS, the SPS is also a Web service interface specification. It allows clients to control sensors with predefined tasks.

An SPS server provides the following operations to plan and control sensor tasks:

- **GetFeasibility** checks whether a task for a specific sensor is feasible or not
- with the **Submit** operation a task is sent to a sensor
- by **GetStatus** a submitted task can be checked
- **DescribeTasking** delivers information how to formulate tasking requests and how the syntax looks like
- With **DescribeResultAccess** the consumer gets the information how to access the gathered sensor data.

Eventing

The eventing components realize push-based, asynchronous delivery of sensor data. This eventing mechanism is based on a publish/subscribe communication pattern, which is different from the conventional request/response communication as used in the SOS for example. A consumer subscribes for a notification and the service forwards the new incoming measured data to the subscribed consumer in near real-time. This also saves periodical request/response communication.

Recently, a new OGC standard for this functionality was released: The OGC Publish/Subscribe Interface Standard^{xii}. However, as this standard is rather new, the number of implementations is currently still limited. An on-going effort is the development of a Publish/Subscribe Sensor Web service by 52°North.

OGC SensorThings API

To cover the specific conditions of the Internet of Things (i.e. low bandwidth, limited processing power) the OGC SWE IoT Standards Working Group was founded. The idea behind this working group is to develop specialised variants of the OGC SWE standards that are optimised for the Internet of Things. Main characteristics of the resulting candidate standard, the OGC SensorThings API, are the use of JSON as an encoding and REST as a binding.

The resulting standards framework is based on a REST API which allow users to perform CREATE, READ, UPDATE, and DELETE (CRUD) actions. This functionality can be applied to two main functional areas:

- **Sensing Profile:** The sensing profile defines different resources types (including data model) for publishing observation data. This functionality is similar to the OGC Sensor Observation Service.
- **Tasking Profile:** The tasking profile defines different resource types for controlling sensors. This functionality is similar to the OGC Sensor Planning Service.

Compared to the regular OGC SWE standards, the SWE IoT API is less comprehensive (e.g. no complex feature model, less options for time stamping) so that it can be realistically be implemented by lightweight, resource-constrained devices.

Recently, the SensorThings API standardisation activities were completed in a first version^{xiii}.

Marine SWE Profile

A marine profile of the OGC Sensor Model Language 2.0 (SensorML 2.0) is being developed allowing to provide metadata for different levels (e.g. observatory, instrument, and detector) and sensor types. The latter will enable metadata of a specific type to be automatically inherited by all devices/sensors of the same type. The application of further standards such as OGC PUCK will benefit from this encoding, too, by facilitating the communication with instruments. For delivering observation data, the ISO/OGC Observations and Measurements 2.0 (O&M 2.0) standard serves as a good basis. Within an O&M profile, recommendations will be given on needed observation types that cover different aspects of marine sensing (trajectory, stationary, or profile measurements, etc.). Besides XML, further O&M encodings (e.g. JSON-based) will be considered. A profile of the OGC Sensor Observation Service 2.0 (SOS 2.0) standard will be

specified to offer a common way on how this web service interface can be used for requesting marine observations and metadata. At the same time this will offer a common interface to cross-domain applications based upon tools such as the GEOSS DAB. Lightweight approaches such as REST will be considered as further bindings for the SOS interface. The profile will consider the existing observation systems so that migration paths towards the specified profiles can be offered. We will present the current state of the profile development. In particular, a comparative analysis of SWE usage in different projects, an outline of the requirements, and fundamental aspects of profiles of SWE standards will be shown.

Further Relevant Standards

IEEE 1451

In Figure 2.3 the main components of the IEEE 1451 standard approach are illustrated. IEEE 1451 uses the term transducer interface module (TIM) to refer to a sensor or actuator, and a network capable application processor (NCAP) to mean a controller interfacing to one or more TIMs. In this case, data coming from IEEE 1451.X instruments are processed to inject data into an IEEE 1451.0 server. This server publishes data using the HTTP 1451 standard.

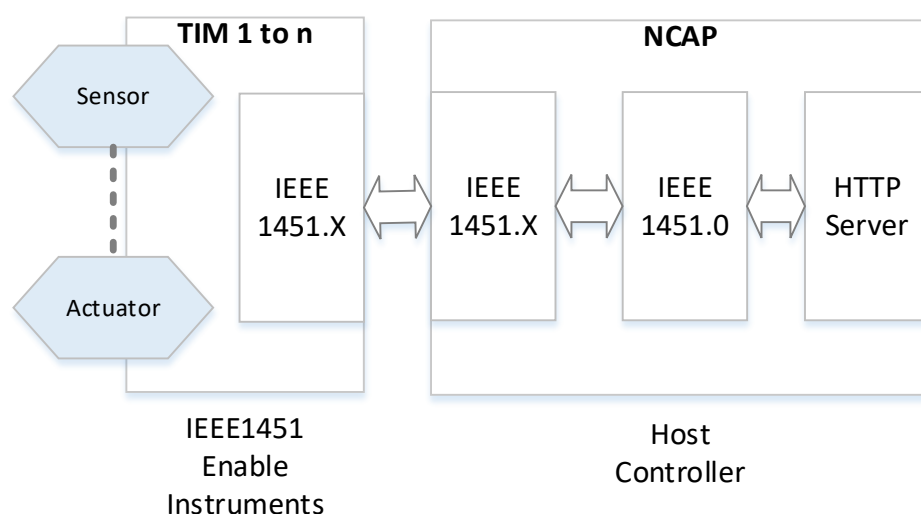


FIGURE 2.3: IEEE1451 SYSTEM ARCHITECTURE

Marine instrumentation most commonly uses serial links, so IEEE 1451.2 would apply. To simplify the use of IEEE 1451 for serial instruments, in 2012 a new draft of IEEE 1451.2 has been released. This new version of IEEE 1451.2 is fully compatible with an RS232 instrument, using the communication and measurement services described in IEEE 1451.0. This will allow manufacturers to implement the protocol in their new generation of instruments.

Another important feature of IEEE 1451 applicable to marine instrumentation is the definition of a standard transducer electronic datasheet (TEDS). IEEE 1451 specifies many standard templates for describing sensors and actuators with a TEDS, and IEEE 1451.4 promotes the idea of storing TEDS information within the device itself. The system we describe includes this approach as well, with PUCK protocol used to store and retrieve the instrument description, whether as an IEEE 1451 TEDS, or in some other format. Java distributed data acquisition and control (JDDAC), created by Agilent Technologies Inc. (Santa Clara, CA, USA) and SunMicrosystems Inc. (Redwood Shores, CA, USA), is another related effort that used the IEEE 1451 TEDS.

Standard Commands for Programmable Instruments (SCPI)

The Standard Commands for Programmable Instrumentation (SCPI) is an industry standard that provides software-level syntax and commands for operating instruments over IEEE-488, RS-232, Ethernet and USB [IEEE 488]. Its aim has been to promote a common language and syntax suitable for all programmable

instruments. Today, SCPI is supported by most of the manufacturers of programmable instruments including Agilent (HP), Tektronix, Keithley, Fluke, and Racal. SCPI is an ASCII language that consists of configuration and query commands that are specific to the instrument and a set of IEEE 488.2 operations and commands that are common to all SCPI-based instruments. SCPI commands have long and short forms, where the long form is very descriptive, and the short form is an abbreviation: “TRIGGER:COUNT” can be “TRIG:COUN”. Although this standard is widely used in the industrial instrumentation, up to now the marine instrumentation manufacturers did not adopt this standard for their developments.

CAN/CAN-OPEN standard family

Controller Area Network (CAN) was originally developed as a bus architecture for automobiles, but today is used in a wide variety of applications. The CAN-bus network provides a very efficient and robust platform for deterministic real-time applications of distributed sensors and actuators. Key advantages provided by CAN-bus include robust and efficient error detection and message transmission protocols. CAN-bus is based on OSI Reference Model layers 1 and 2 (physical and data link layers) and is standardized in ISO 11898 [ISO 11898-1, 2003] [ISO 11898-2, 2003] [ISO 11898-3, 2006] [ISO 11898-4, 2004] [ISO 11898-5, 2016] [ISO 11898-6, 2013]. Several application-level standards have been developed to run on CAN-bus, notably the CANopen communication protocol and device profile specification. Several oceanographic applications that use CAN-bus and CANopen for onboard communications have been implemented, including autonomous underwater vehicles and buoys, and at least one manufacturer supplies oceanographic instruments for CAN-bus. An important step towards achieving interoperability between different ocean observatory systems is to use a basic set of terms to describe the collected data. A number of vocabularies already exist. However, CANopen standard has been defined having different applications in mind. For instance there are definitions on the format of electronic data sheets that are described with the CanOpen or OGC-SWE standards that address similar specification needs but are not identical in format. A harmonization between these arrangements appears necessary for instance by defining a basic set of terms that can be easily mapped between the different standards. Integration of CAN with OGC-SWE standards will be possible, e.g. by “mapping” CANopen electronic data sheets to SensorML.

OGC PUCK

The OGC PUCK is a simple protocol that works with other standards to enable interoperability and automatic configuration. OGC PUCK makes it possible for instruments to carry information that enables sensor networks to use the instrument and its data. OGC PUCK defines a simple protocol to store and retrieve information from an instrument over RS-232 and Ethernet [O’Reilly, T., 2014]. As shown in Figure 2.4, this information consists of a minimal instrument datasheet that includes a universally unique instrument serial number, a manufacturer ID, and a small amount of other metadata. OGC PUCK protocol also allows an optional payload consisting of sensor metadata or any information needed by an observing system. OGC PUCK can be implemented in instrument firmware augmenting the “native” instrument command set.

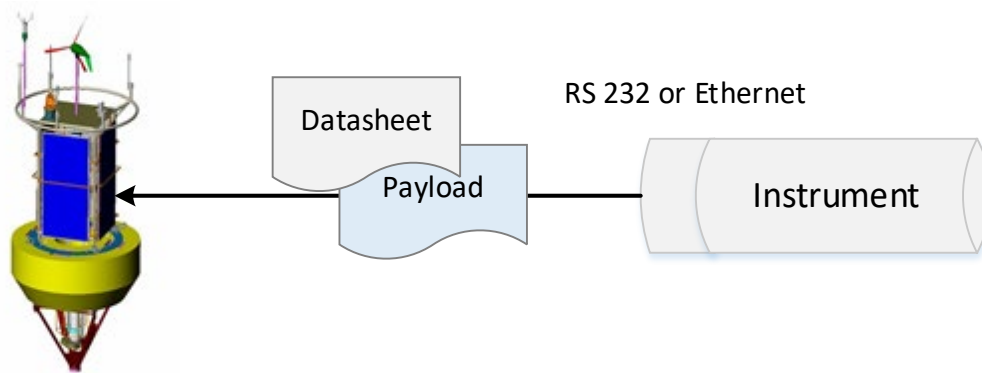


FIGURE 2.4: HOST ISSUES OGC PUCK COMMANDS TO RETRIEVE DATASHEET AND PAYLOAD

A PUCK-enabled RS232 instrument can operate in an “instrument mode” or a “PUCK mode.” In the instrument mode, the device responds to the protocol defined by its manufacturer, including “nonstandard” commands, while in the PUCK mode, the device responds to the standard PUCK protocol. The PUCK mode is typically used when the instrument is first connected to the host or the host is rebooted. To switch a PUCK-enabled device into the PUCK mode, a “PUCK soft break” command is sent to the device at several baud rates until a valid PUCK response is received by the host while in the PUCK mode, the host can retrieve the PUCK datasheet and any optional payload information, from which the host can infer the instrument model and the manufacturer. The host then switches the device into the “instrument mode” and begins issuing instrument- specific protocol commands to it as shown in Figure 2.5.

The OGC PUCK standard specifies PUCK over IP (“IP PUCK”). In addition to the basic PUCK protocol, IP PUCK specifies the existing Zeroconf standard as a means to automatically assign the instrument’s link-local IP address and hostname, and a mechanism for hosts to discover the instrument address and “PUCK port” number. Thus, an IP PUCK instrument can automatically acquire an IP address and name when it is physically installed in a network. A host can then use Zeroconf’s service discovery protocol to discover PUCK-enabled instruments in the marine IP network and retrieve their metadata and payloads using PUCK commands issued to the instrument’s specified “PUCK port”.

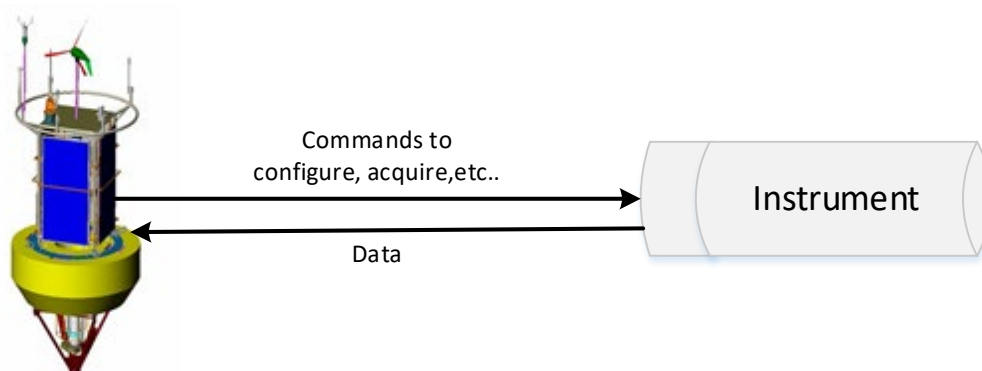


FIGURE 2.5: BASED ON INFO RETRIEVED IN THE FIRST STEP, HOST USES INSTRUMENT’S “NATIVE” PROTOCOL TO CONFIGURE AND ACQUIRE DATA.

One key advantage of implementing the OGC PUCK is that the standard it is already developed by several marine instrumentation manufacturers and it enables in a very easy manner the automatic instrument integration into sensor network (‘Plug and Work’) [Toma, D.M., et al., 2011], [SmartInterface].

ESONET Yellow pages (O3YP)

The Open Ocean Observatories Yellow Pages' (O3YP's) aim is to organize the information concerning off the shelf products, which are provided by the private sector for the development and maintenance of open-ocean observatories.

This includes a range of equipment, from simple isolated sensor parts, to communication systems and even integrated observatories. The O3YP's also aims to provide feedback from the scientific community in what concerns the experience with a specific product, addressing reliability for long-term operations and the use in real deep-sea conditions.

As the O3YP evolved from the ESONET Yellow Pages (EYP), which was mainly for deep-sea observatories, they need to be updated through several steps before they are fully functional. The steps for this first update are described in this Report. The internet name stays as Esonet yellow pages however the Open Ocean yellow pages will be an integration of Esonet Yellow pages, JERICO and EMSO.

Part of the activities in FIXO3 have consisted in linking to the O3YO's to automatically create commercial sensor templates in order to ease the use of the developed SensorML editor.

Esonet Yellow Pages » Sensors » ADCPs

ADCPs → found 12 sensors [How to use this table](#)

Picture	Model	Manufacturer	EC project reference*	Frequency	Max. profiling range	Cell size
	Aquadop Profiler 1000	Nortek AS		1000 kHz	20 m	0.3 - 4 m
	Aquadop Profiler 2000	Nortek AS		2000 kHz	10 m	0.1 - 2 m
	Aquadop Profiler 400	Nortek AS		400 kHz	90 m	2 - 8 m
	Aquadop Profiler 600	Nortek AS		600 kHz	40 m	1 - 4 m
	FlowQuest 1000	LinkQuest Inc.		1000 kHz	40 m	0.25 - 2 m
	FlowQuest 150	LinkQuest Inc.		150 kHz	500 m	2 - 16 m
	FlowQuest 300	LinkQuest Inc.		300 kHz	230 m	1 - 8 m

10 Page 1 of 2 Displaying 1 to 10 of 12 items

FIGURE 2.6: SCREENSHOT OF THE O3YP WEB INTERFACE, AN UPGRADE OF THE ESONET YELLOW PAGES (EYP)

2.7) SWE and Best Practices

A summary of the benefits of SWE technologies is provided in the FixO³ Handbook of Best Practices:

"Sensor Web technologies may close the information gap between sensor data producers and sensor data consumers. The Sensor Web also provides means to remotely operate sensors via the internet and to provide additional metadata such as provenance or quality information. At the moment, best practices and common architectures to utilize Sensor Web technologies in the Oceans domain are being investigated and implementations of those are developed. Thus, SWE may ease the work of sensor operators, domain scientists and modellers."

For more details we refer to the corresponding deliverable of FixO3 project [Coppola, L., et al., 2017].

2.8) Results and discussion

The section presents the newly developed web-based sensor registration interface in a user manual format. The first section links to the source code repository, from which the architecture can be derived.

SensorML Editor (SMLE) source code

The editor was developed under an open-source framework. The code is available on Github at the following URL: <https://github.com/52North/smle>. Free registration to Github may be required to access the repository.

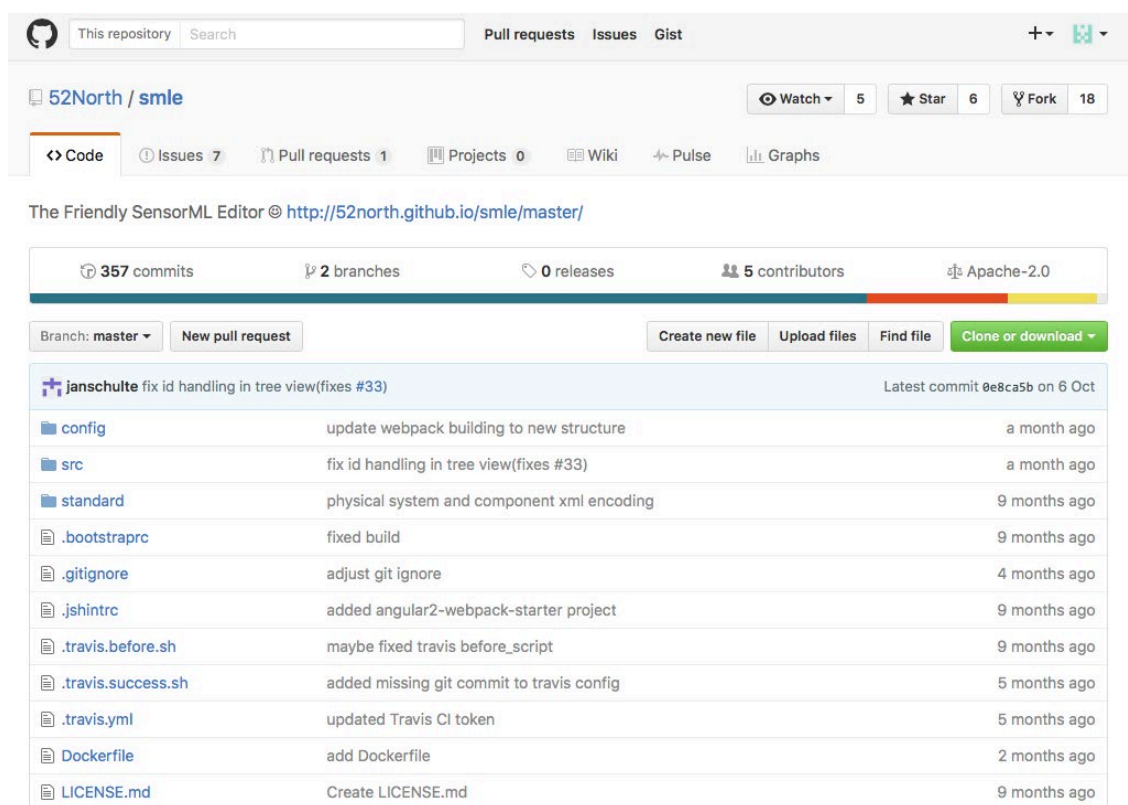


FIGURE 2.7: SMLE EDITOR SOURCE CODE

2.8.1) SensorML Editor (SMLE) Manual

This document provides helpful information on how to use the **SensorML Editor (SMLE)** Web application provided by **52°North**. It uses a tutorial-based manner to guide you through the basic functionalities. This documentation is based on the SMLE version from October 2016. The most current development of **SMLE** is available from the [official GitHub page of SMLE](#).

2.8.2) Basic Visual Interface of SMLE

This section provides general information about the Web application SMLE. As the full name indicates, the SensorML Editor is a Web application allowing users to manage SensorML documents of a preconfigured Sensor Observation Service (SOS). This SOS instance is configured during installation by an IT-expert and is currently not changeable at runtime. **SMLE** provides a visual interface for various tasks such as viewing/editing/deleting existing SensorML documents or creating new sensor descriptions. Without logging in, the application only allows to view existing sensor descriptions. Only after a **successful login** users may perform the administrative tasks *create*, *edit/update* and *delete*. To log in you can use your **GitHub credentials**.

As an exemplar instance of *SMLE*, this guide uses the application from <http://pilot.52north.org:3000/#/>. The landing page greets you as shown in the subsequent figure.



FIGURE 2.8: SMLE HOME PAGE

The dark grey navigation bar contains the available functionalities of *SMLE*, which are accessible by clicking the corresponding button. Via **New** and **Create from Template**, new SensorML documents can be created. Using **Edit/View**, existing sensor descriptions can be viewed and edited. However, as mentioned above, the administrative functionalities can only be performed once logged in. In consequence, as a first step, you should log in via the **login** button located at the right of the navigation bar. This will open a popup window to enter your GitHub credentials. **Note that you may have to enable popups within your browser!**

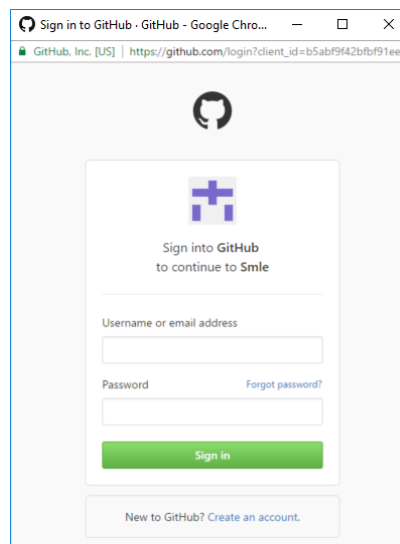


FIGURE 2.9: GITHUB CREDENTIALS

After signing for the first time, you will have to authorize *SMLE*, allowing the application to access your GitHub account. This step is not required for subsequent logins. When logged in, the **login** button is replaced by your GitHub name and the option to logout. In addition, the navigation bar is enriched with a new button called **Delete** allowing you to delete your sensor descriptions. Having logged in, all functionalities become available. Each is introduced subsequently within its own section.

2.8.3) Modifiable Items and Restrictions

When logged in, you are able to create, edit or delete Sensor descriptions. However, you are only allowed to modify content that you have previously created by yourself. To be more precise, you are allowed to *view* any available sensor description from any user. Also, you can *create* arbitrary new SensorML documents, providing a new unique identifier. But you are not permitted to *update* or *delete* sensor descriptions from another user. Only your own documents that are linked to your account are open for modification. Due to this restriction, the application prevents users to manipulate content without permission.

Although not being allowed to modify documents of other users, the visual interface allows to edit an existing sensor description and even click on the **Publish** and **Update** buttons (see section Edit your own

SensorML Descriptions for more detailed workflow). Only then *SMLE* will notice that you tried to change the description of another user and fail. This is due to technical reasons. As the underlying SOS instance stores, which sensor description was created by which user, *SMLE* builds and executes a *SOS UpdateSensorDescription* request. If the logged in user does not have the permission to update that specific sensor description, the SOS operation fails. As an enhancement, future version of *SMLE* might inspect earlier, whether users have permission to edit a certain document and inform them via suitable means (e.g. hide the edit button on a missing permission).

2.8.4) Create SensorML Document from Template

You need to be logged in in order to publish a new SensorML description.

Find and Select Template

The application allows creating new sensor description based on an existing template that already fills parts of the SensorML structure. Select **Create from Template** from the navigation bar to open a new menu where you can browse and search for available SensorML templates provided by the [ESONET YELLOW PAGES](#). As of October 2016 380 templates are available that can be filtered through appropriate keywords. To browse all templates, simply hit the **Search** button without any keyword. To filter the templates, enter any keyword and hit the **Search** button. The following example shows all entries for the keyword *salinity*. From the list of returned templates you may select the target template, which is used to create a new SensorML document. E.g. when selecting the template *Hydrolab Conductivity Sensor*, a description of the selected sensor as well as the option to **create a sensor description of this template** is provided. Optionally, you may already enter a new unique identifier for the new SensorML document, either manually or by clicking the **Create identifier** button, which generates a system-wide unique identifier. However, the identifier may be defined later as well.

FIGURE 2.10: SMLE TEMPLATE SELECTION

Instantiate and Edit new SensorML Document based on selected Template

After selecting a template (and providing a new identifier) you may click on the button **Create sensor description from template**. This will instantiate a new SensorML document using the selected template to provide the necessary SensorML elements and insert predefined values. The application switches to the edit view, where the elements of the SensorML document are displayed. On the left side, the edit view

allows to *add* new metadata within various elements of the sensor description or *edit* existing properties. On the right side, you may toggle the tree view to gain an overview of the specified metadata as a tree structure.

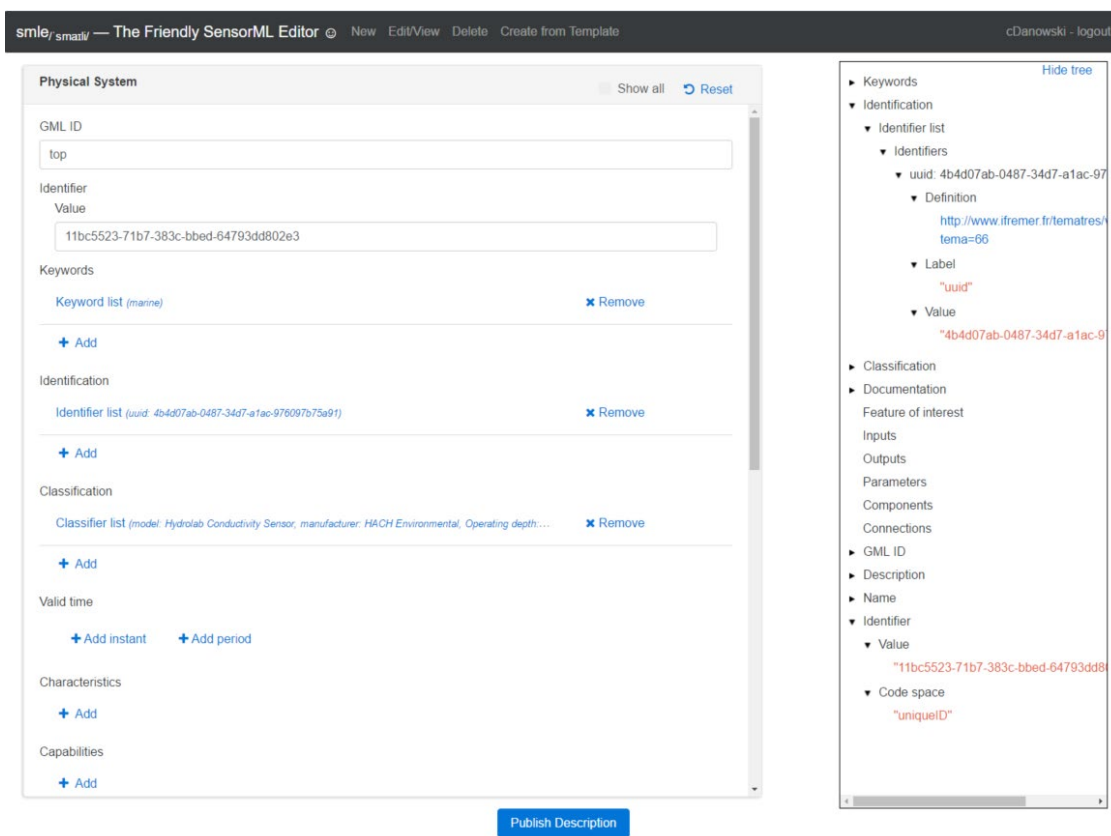


FIGURE 2.11: SMLE INSTANTIATION

If specified, the value of the second entry *Identifier* stores the unique identifier, which can still be modified. Optionally, you can activate the expanded view of *all available elements* by setting the checkbox of **Show all** located at the top of the edit window. If activated, additional element fields are revealed to be edited.

The process of adding new information uses a nested window design to resemble the hierarchy of the added item. As an example, the subsequent series of figures shows how to add *contact information*. Notice how with each new hierarchy level a new nested window appears, where you can add/enter new information. To close a nested window, you may either use the **Close** button on the top right of the window or click on the vertically oriented name of a previous hierarchy element on the left.

- 1 Navigate to the **Contacts** element of the SensorML document. Click on the **Add** button, which adds the item *Contact list*.

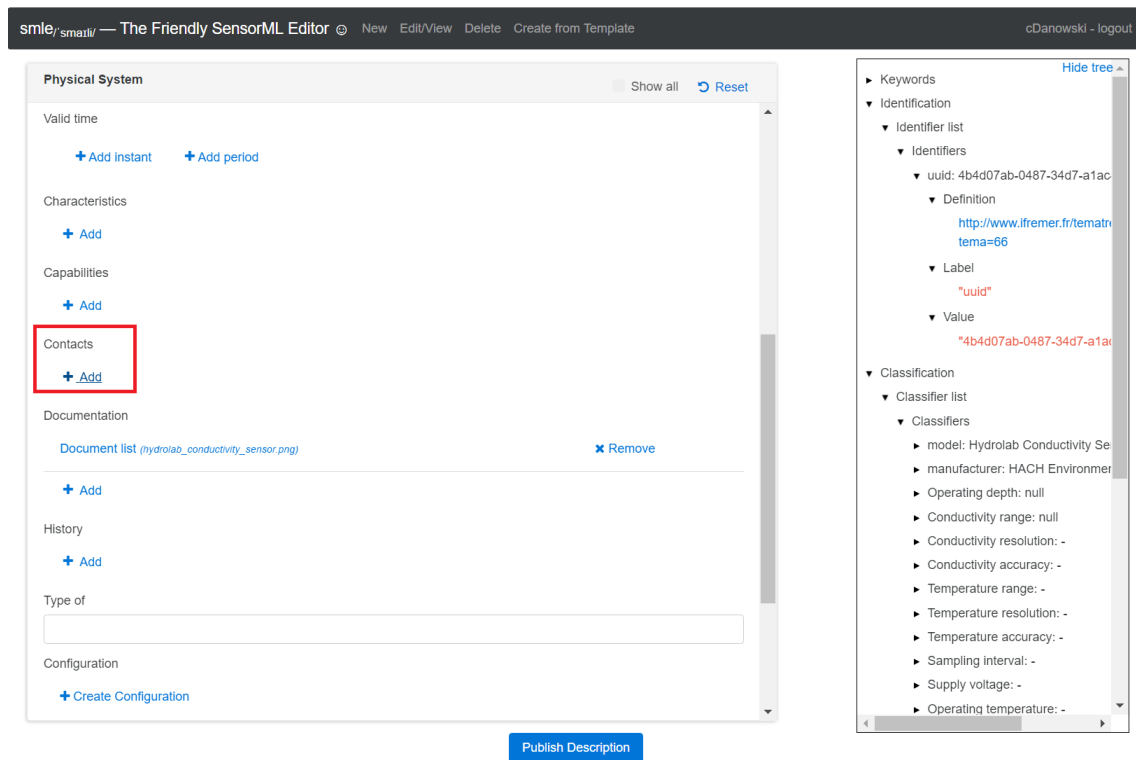


FIGURE 2.12: SMLE INSTANTIATION - STEP 1

- 2 Click on the newly created *Contact List* element to open it in a new nested window.

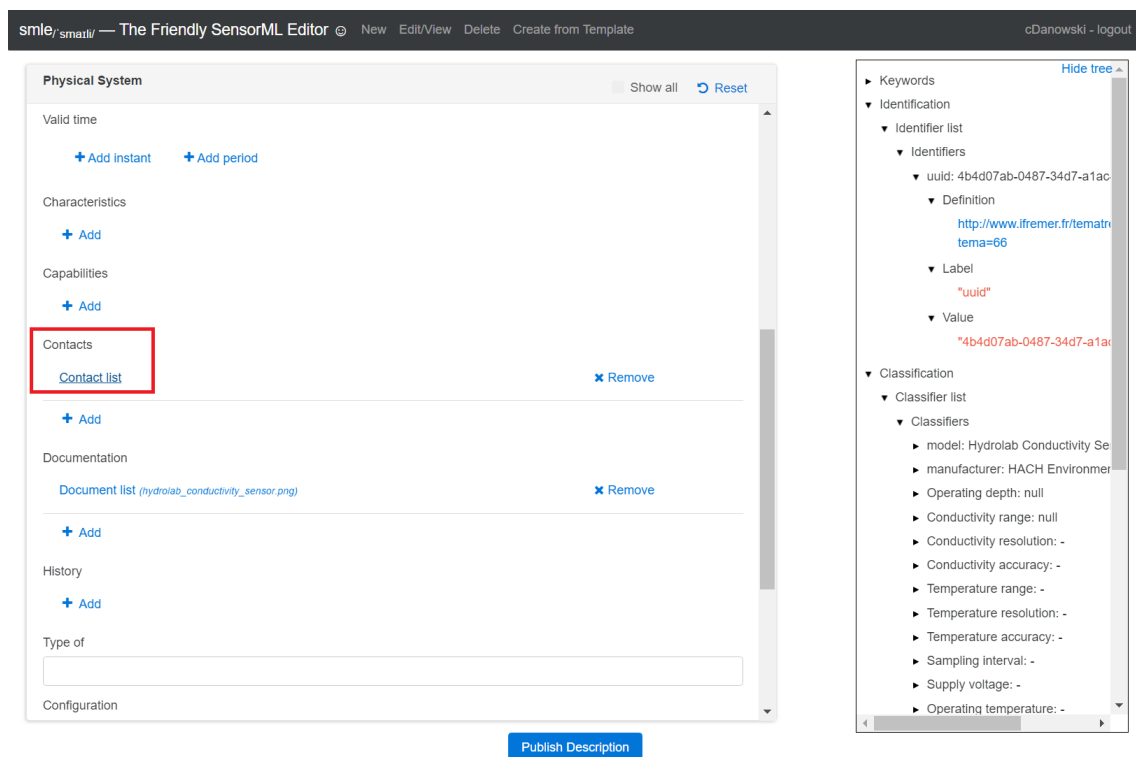


FIGURE 2.13: SMLE INSTANTIATION - STEP 2

- 3 Click the **Add** button to create the new item *Responsible party*.

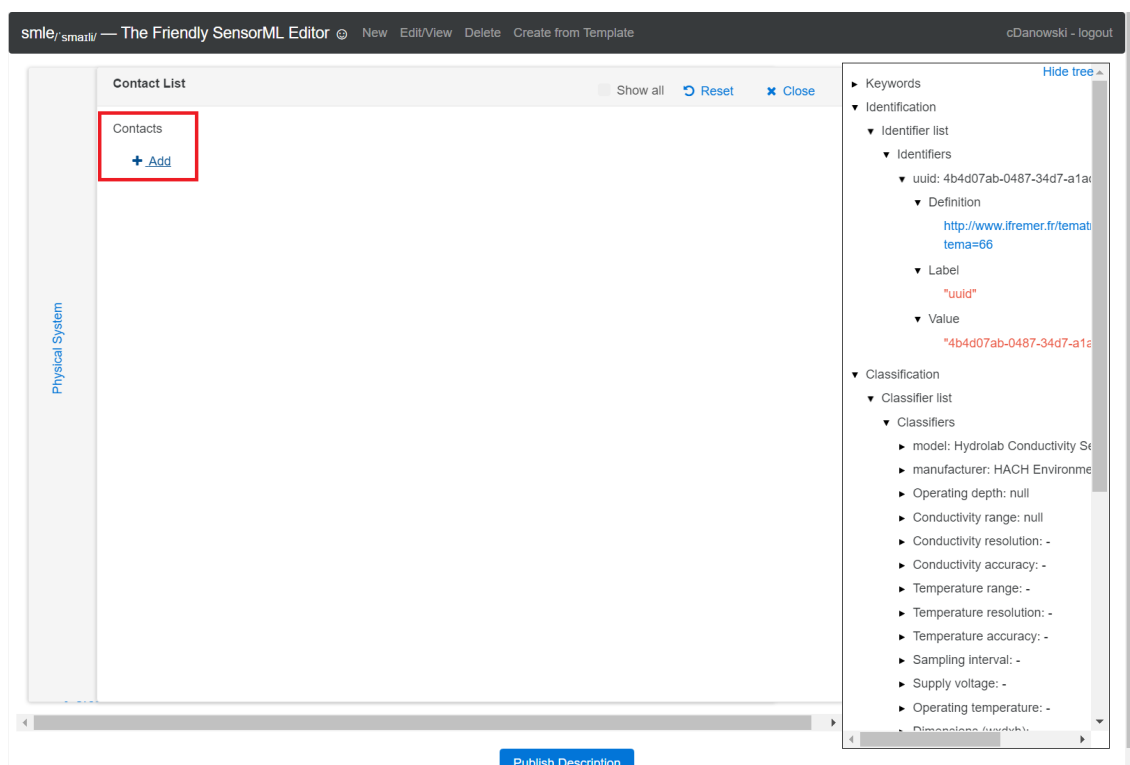


FIGURE 2.14: SMLE INSTANTIATION - STEP 3

- 4 Click on **Responsible party** to open and edit it in a new nested window.

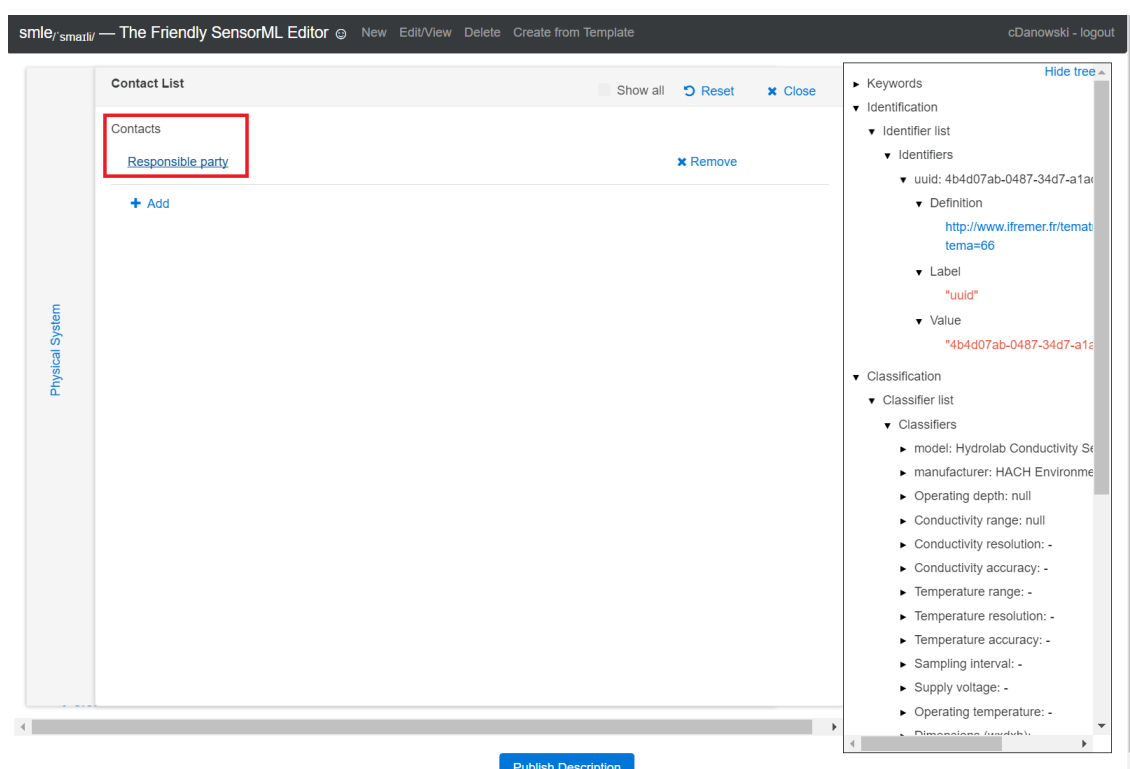


FIGURE 2.15: SMLE INSTANTIATION - STEP 4

- 5 Fill the form fields of the **Responsible party** element and, in addition click on **Create Contact Info**, which adds a new item called *Contact Info*. Whenever you edit any form field, the corresponding content is immediately updated with the new value. Switching back to previous hierarchy level (in

this case *Physical System* or *Contact List*) is allowed at any time. All changed values have been recognized and applied by *SMLE*.

The screenshot shows the 'Responsible Party' form in the SMLE interface. The form has the following fields:

- Individual name: My Name
- Organisation name: Org Name
- Position name: My Position
- + Create Contact Info (highlighted with a red box)
- Role: Resource Provider (selected from a dropdown menu)

The Role dropdown menu is open, showing the following options:

- Resource Provider (highlighted in blue)
- Custodian
- User
- Originator
- Point Of Contact
- Principal Investigator
- Processor
- Publisher
- Author
- Owner

The interface also includes a sidebar with 'Physical System' and 'Contact List' tabs, and a 'Publish Description' button at the bottom.

FIGURE 2.16: SMLE INSTANTIATION - STEP 5

- 6 Click on Contact Info to open and edit it in a new nested window.

The screenshot shows the 'Responsible Party' form in the SMLE interface. The form has the following fields:

- Individual name: My Name
- Organisation name: Org Name
- Position name: My Position
- Contact Info (highlighted with a red box)
- Role: Resource Provider (selected from a dropdown menu)

The Role dropdown menu is open, showing the following options:

- Resource Provider (highlighted in blue)
- Custodian
- User
- Originator
- Point Of Contact
- Principal Investigator
- Processor
- Publisher
- Author
- Owner

The interface also includes a sidebar with 'Physical System' and 'Contact List' tabs, and a 'Publish Description' button at the bottom.

FIGURE 2.17: SMLE INSTANTIATION - STEP 6

- 7 When you reach this point, you should have understood, how to edit the properties of a SensorML document using *SMLE*. Entries highlighted using *blue* colour reveal editable properties within a new nested window. Form fields represent the values of a certain property. E.g., within the **Contact**

element, there are three additional elements and two form fields to be edited. Within this guide only the **Address** element is configured subsequently. So click on it to open it as a new nested window.

The screenshot shows the SMLE (The Friendly SensorML Editor) interface. The top bar includes the title 'smle / smali - The Friendly SensorML Editor' and navigation links: 'New', 'Edit/View', 'Delete', and 'Create from Template'. The user 'cDanowski' is logged out. The main workspace is divided into three vertical panels: 'Physical System', 'Contact List', and 'Responsible Party'. The 'Contact' window is open, displaying a list of elements: 'Phone', 'Address' (highlighted with a red box), 'Online resource', 'Contact instructions', and 'Hours of service'. The 'Address' element is selected, and its form fields are visible. A 'Publish Description' button is located at the bottom of the workspace.

FIGURE 2.18: SMLE INSTANTIATION - STEP 7

- 8 Fill out the form fields of the **Address** element with your values. Of particular interest are the fields *Delivery point* and *E-mail*. Both have additional interactions called *Add* and *Clear*. The *Clear* button clears the entered value while the *Add* button must be clicked to persist the entered value for the property. As these specific properties allow multiple values, it is not sufficient to only enter the new value within the form field. Instead you have to explicitly *add* it.

smle/smali — The Friendly SensorML Editor New Edit/View Delete Create from Template cDanowski - logout

Address Show all Reset Close Show tree

City
My City

Postal code
12345

Country
My Country

Administrative area
My Administrative Area

Delivery point
My Address
Add Clear

E-mail
mymail@example.com
Add Clear

Publish Description

FIGURE 2.19: SMLE INSTANTIATION - STEP 8

- 9 After explicitly adding the values of *Delivery point* and *E-mail*, you may enter a second value or remove any value.

smle/smali — The Friendly SensorML Editor New Edit/View Delete Create from Template cDanowski - logout

Address Show all Reset Close Show tree

City
My City

Postal code
12345

Country
My Country

Administrative area
My Administrative Area

Delivery point
My Address Remove
Add Clear

E-mail
mymail@example.com Remove
Add Clear

Publish Description

FIGURE 2.20: SMLE INSTANTIATION - STEP 9

- 10 To finish editing the contact info you may either use the *Close* button within each nested window (located on the top right) or click on the vertically oriented name of a previous hierarchy level on the left. Using the latter, you may directly return to the leftmost window (in this case *Physical System*), which is the top-level of the SensorML Document. To verify the applied changes, you may

use the *tree* view on the right and expand it at the relevant elements. Alternatively navigate to the respective element within the left menu to inspect it and perform additional edits.

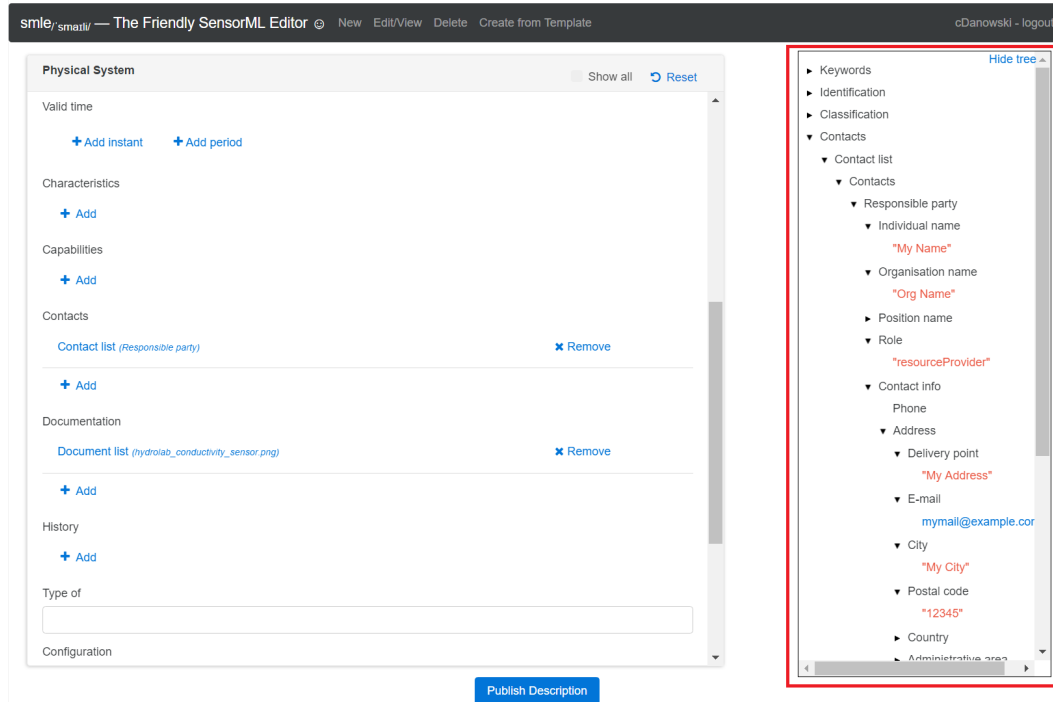


FIGURE 2.21: SMLE INSTANTIATION - STEP 10

- 11 At this point, you should have the knowledge to edit any property of the SensorML description. You may *add* new elements or *edit* the values of a certain element in a nested window. Clicking on any item, which is highlighted using *blue* colour, will open that element as a new nested window to reflect the hierarchy of the SensorML document structure.

Preset Structure and Values of the Template

As you may have noticed, when using a template, certain elements are already pre-set with values from the template. While some values represent template-specific standard values that do not require changes, certain other elements are instantiated using *null* values. However, when relying on the template, you should have the knowledge to identify those elements and provide suitable values.

As an example, using the *Hydrolab Conductivity Sensor* template, the *Classification* element is pre-set with certain classifiers. Open this element by clicking on **Classifier list (...)**.

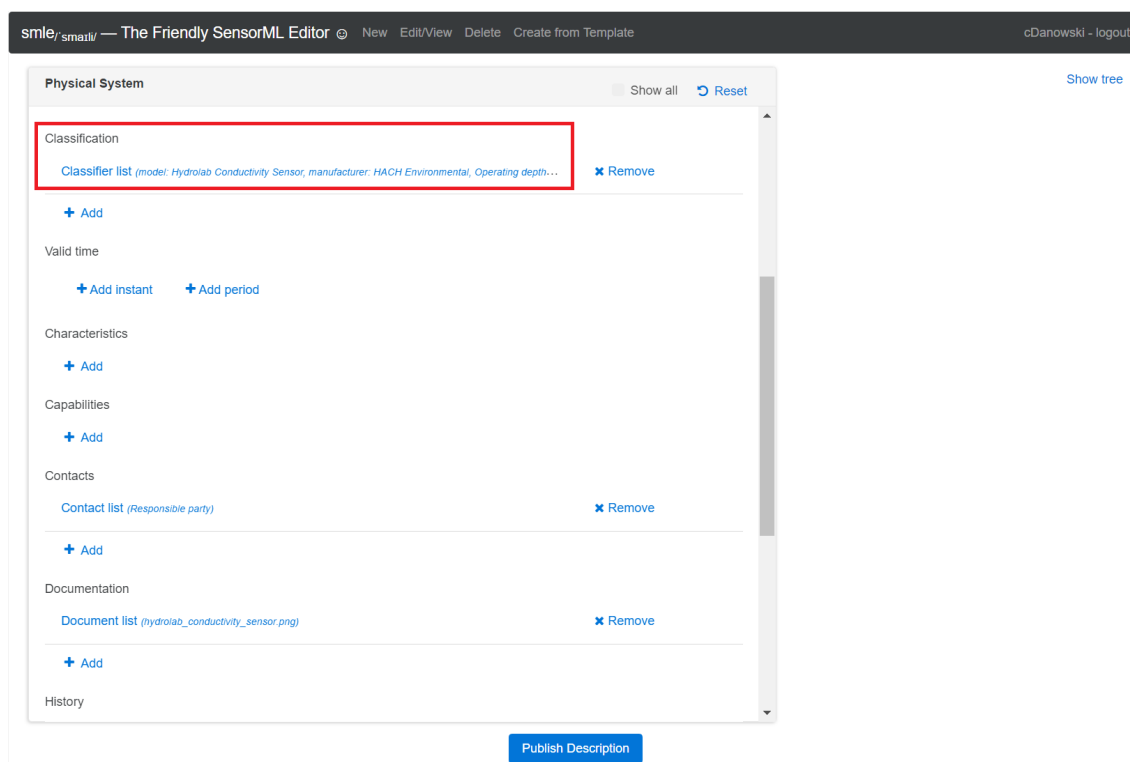


FIGURE 2.22: REPLACING/OVERRIDING VALUES 1

This will reveal a nested window with all pre-set classifiers of the template. As you can see, the value of most of the *classifiers* is set to **null** or “-”. However, as they describe important operational characteristics of the chosen sensor, you should fill them with applicable real values. To replace/override any value, click on the corresponding item. E.g. click on **Operating depth** to open it in a new nested window.

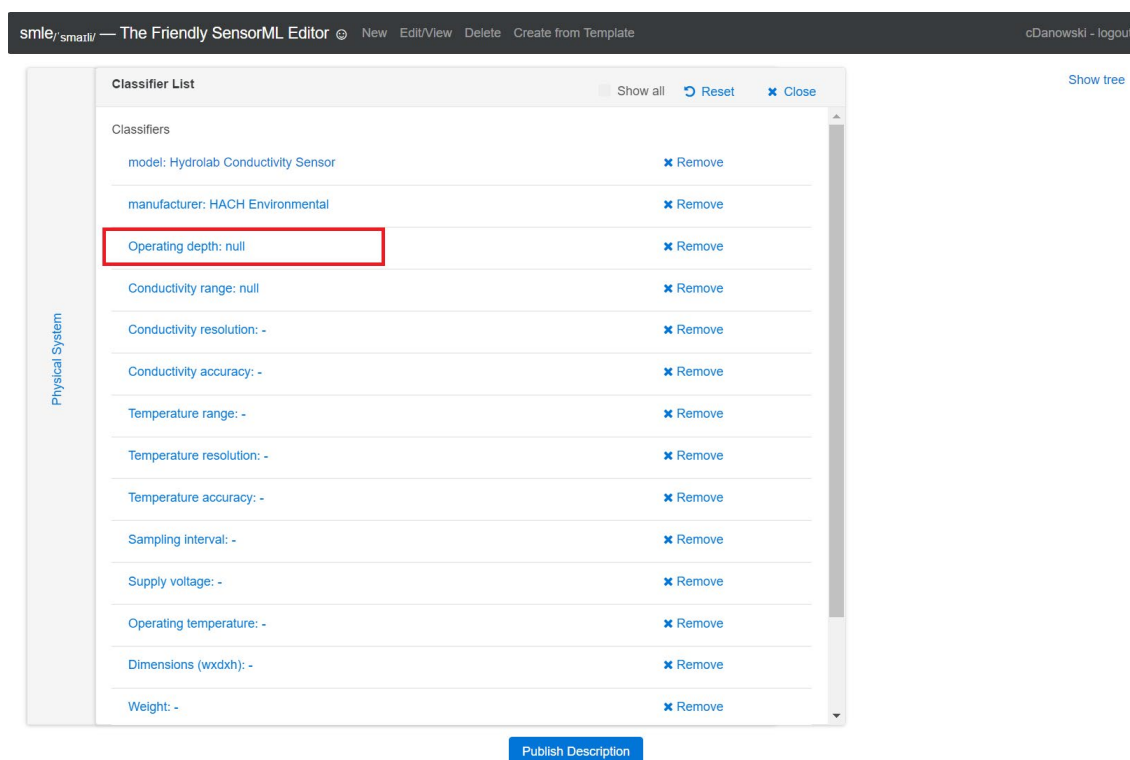


FIGURE 2.23: REPLACING/OVERRIDING VALUES 2

Within the edit window you might change the label and in particular the value of the displayed item. As soon as you replace the **value** property, you may close the *Term* edit window.

The screenshot shows the 'Term' dialog box in the 'The Friendly SensorML Editor'. The 'Label' field is 'Operating depth' and the 'Value' field is 'newDepthValue'. A red box highlights the 'Value' field. The background shows a sidebar with 'Physical System' and 'Classifier List' sections. At the bottom, there is a 'Publish Description' button.

FIGURE 2.24: REPLACING/OVERRIDING VALUES 3

Back on the hierarchy level *Classifier list* you should verify that the edited property indeed carries the new value.

The screenshot shows the 'Classifier List' dialog box in the 'The Friendly SensorML Editor'. The 'Operating depth: newDepthValue' entry is highlighted with a red box. The background shows a sidebar with 'Physical System' and 'Classifier List' sections. At the bottom, there is a 'Publish Description' button.

FIGURE 2.25: REPLACING/OVERRIDING VALUES 4

As shown by this simple example, you should navigate through all pre-set values and edit them to the de facto values of the sensor for which you create the SensorML document. *SMLE* will **NOT** inform you about any property that still carries any *null* values. So you will have to carefully inspect each property by yourself. Should you wish to remove any property, you are free to do so. Basing on a template is up to your choice

and only provides recommendations on how to model sensor descriptions using the SensorML standard. In theory, you are free to include the necessary information about your sensor using other properties/elements of SensorML. However, using templates and its recommended pre-set properties allows better comparability to descriptions of similar sensors.

Publish and persist new SensorML Document

After adding or overriding the necessary properties to reflect your sensor characteristics, click on the **Publish Description** button located below the edit window. Note that you have to be logged in to see the button! If you have not specified any *identifier* for the new document you are prompted to enter it now. Again, you may let the system generate a new identifier using the **Create identifier** button, as shown in the subsequent figure. Alternatively, enter a manual value.

The screenshot shows the top navigation bar of the 'smle/smaile' web application. Below the navigation bar, a message states: 'The sensor description needs an ID to be published'. There is an input field labeled 'Identifier' and a 'Create identifier' button. Below these, there is a blue button labeled 'Use identifier to publish description'.

FIGURE 2.26: PUBLISH NEW SENSOR INSTANCE 1

When publishing the document, *SMLE* will provide you with a final uneditable view of the XML structure of the new SensorML document for verification purposes. Here you should proofread all the edited properties/elements. When detecting an error or some other reason to return to the edit view, use the **Edit Description** button located above the XML view. If you are satisfied with the preview of the new SensorML document, you find a notification below the XML preview. *SMLE* contacts the SOS instance to check, whether the identifier of the new SensorML instance already exists within the SOS. If not, it informs you about this and offers you a button called **Add Description** to persist the created document within the SOS.

The screenshot shows the 'Sensor description with ID: 11bc5523-71b7-383c-bbed-64793dd802e3' view. It includes an 'Edit Description' button. The main content area displays the XML structure of the sensor description, including details about the Hydrolab Conductivity Sensor. Below the XML preview, a green notification box states: 'The SOS has no description with the identifier '11bc5523-71b7-383c-bbed-64793dd802e3'. Do you want to add the description now?'. A blue 'Add Description' button is located below the notification.

FIGURE 2.27: PUBLISH NEW SENSOR INSTANCE 2

After clicking the button, you are notified whether the process was successful. (e.g. note the success message at the bottom of the following figure. Occurring errors are described in sub-section Error Handling below.)

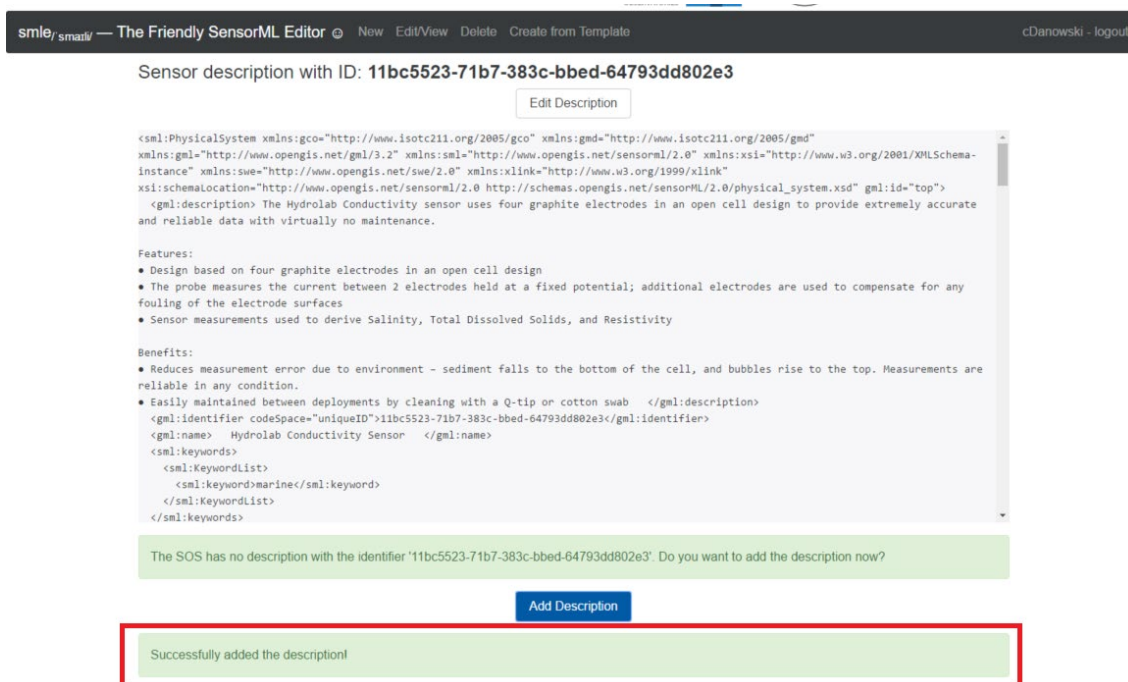


FIGURE 2.28: PUBLISH NEW SENSOR INSTANCE 3

Should the identifier already exist within the SOS, You are asked if you want to *update* the existing document using an **Update Description** button. Here you should carefully decide what to do. If you created a new SensoML instance for a new sensor, the *identifier* should be new as well. If you receive a notification that the *identifier* already exists, you should edit the document and alter the property.

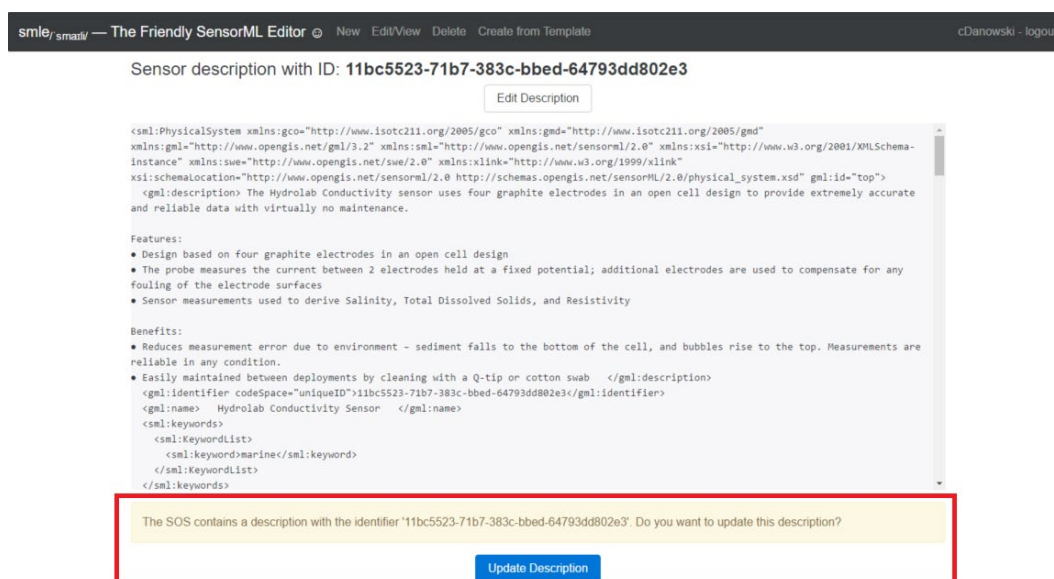


FIGURE 2.29: PUBLISH NEW SENSOR INSTANCE 4

Once persisted within the SOS, you are still able to edit the document using the **Edit Description** button and re-publish the document. In this case the identifier should remain untouched to only *update* the existing document within the SOS.

2.8.5) Error Handling

When trying to publish a SensorML document, the underlying SOS might return with an error message in case the SensorML document does not conform to the requirements of the SOS and SensorML standard. An error message may have several reasons. To mention a few: a needed element may not be specified, empty elements may not be allowed or provided property values do not pass validity checks. If the SOS

rejects the SensorML document for whatever reason, *SMLE* displays the error message at the bottom of the publish view, according to the next figure.

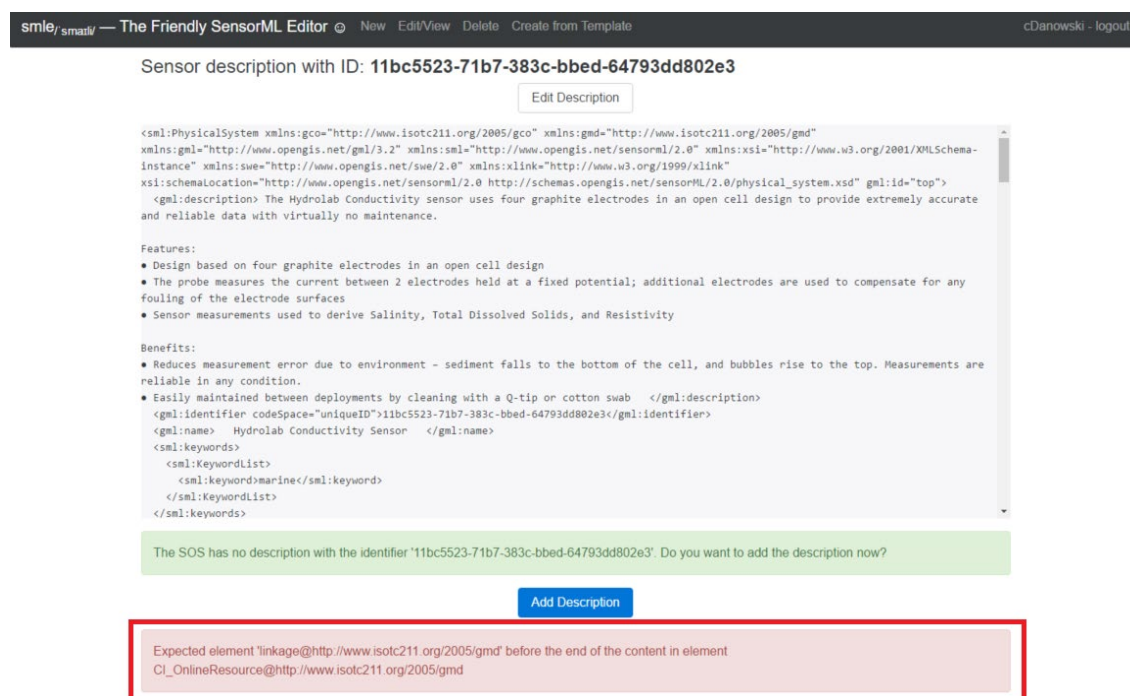


FIGURE 2.30: ERROR HANDLING 1

Currently, the displayed error message is forwarded from the SOS instance and should indicate what kind of error occurs. Hopefully, the message contains a hint on how to edit the SensorML document to make it valid. In this case the error message notifies you of a missing required element called *linkage* within the higher-level element *CI_OnlineResource*. Sadly, there is no indication on where to find the latter. To troubleshoot the problem in worst case scenario, edit the document and inspect every hierarchy level to find the required element and provide a proper value for it. For instance, the created *Contact* element defines an element **Online Resource** with the property **Linkage**. So, navigate to *Contact List - Responsible Party - Contact (Contact Info) - Online Resource* to enter a suitable value, as shown in the following figure.

The screenshot shows the 'Online Resource' form in the SensorML Editor. The form is titled 'Online Resource' and has a 'Show all' button, a 'Reset' button, and a 'Close' button. The form contains the following fields:

- Linkage:
- Protocol:
- Application profile:
- Name:
- Description:
- Function:

At the bottom of the form is a blue button labeled 'Publish Description'.

FIGURE 2.31: ERROR HANDLING 2

After that hit the **Publish Description** button again to open the publish view and add/update the new sensor description. If no other error occurs, the SOS will then save the transmitted document.

2.8.6) Create new SensorML Document

In contrast to creating a new SensorML instance using a template, you are free to create a completely new document via the menu **New** from the navigation bar. As first step you are prompted to choose the *descriptionType* of the new document instance. Currently you can choose between *PhysicalSystem* for single sensor devices and *PhysicalComponent* for sensor components within a multisensor component-based setting.

The screenshot shows the 'Select the descriptionType' dropdown menu in the SensorML Editor. The menu is open, showing two options: 'PhysicalSystem' and 'PhysicalComponent'.

FIGURE 2.32: CREATE NEW SENSOR 1

After selection you enter the *edit view*, where you can add necessary sensor information using dedicated elements and properties of the SensorML standard. The process of adding/editing information as well as the process of publishing the document has already been described in the previous section Create SensorML Document from Template. Thus, it is skipped here.

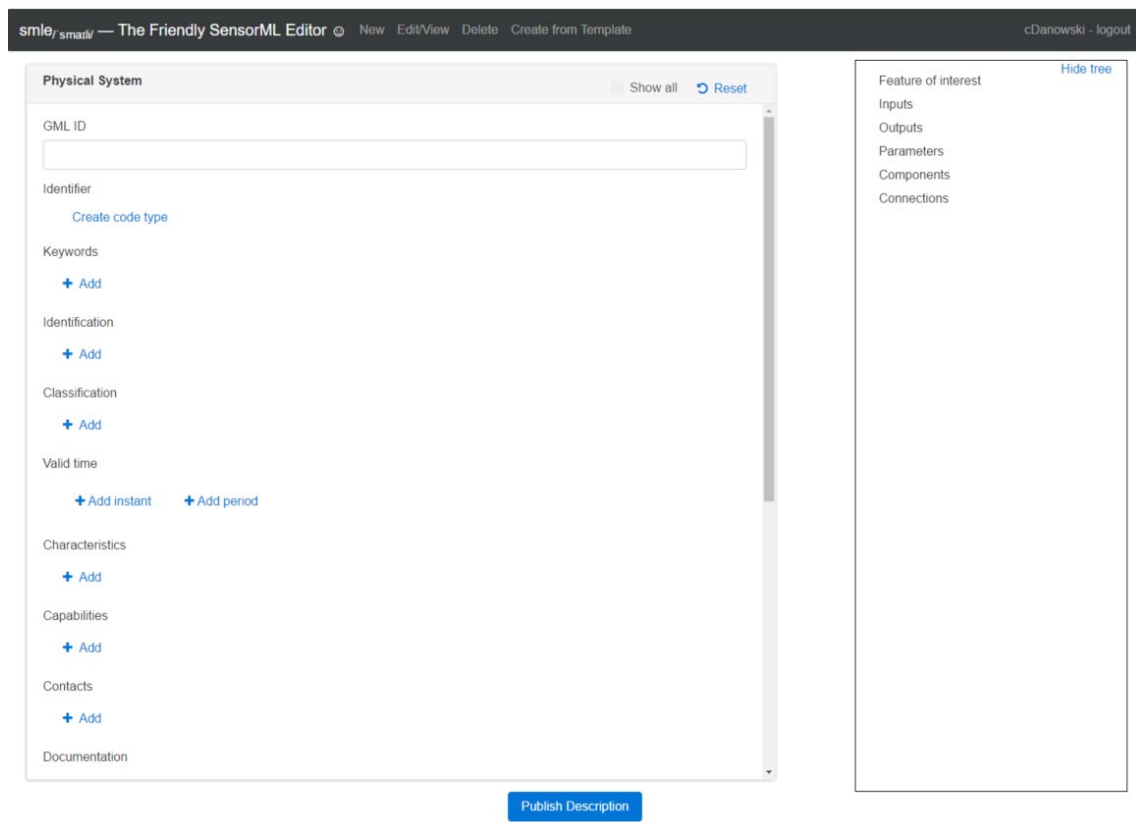


FIGURE 2.33: CREATE NEW SENSOR 2

As you notice, the created document is still empty, as you start editing from the scratch. Whether you decide to edit an empty description or base on an existing template is up to you. However, it is recommended to start using a template, if possible. The pre-set structure generated by the template may allow better comparability between similar SensorML documents of similar sensor settings.

2.8.7) View existing SensorML Descriptions

To view existing sensor descriptions, you do not have to be logged in. Navigate to the menu **Edit/View** from the navigation bar to find a dropdown with the available SensorML documents.

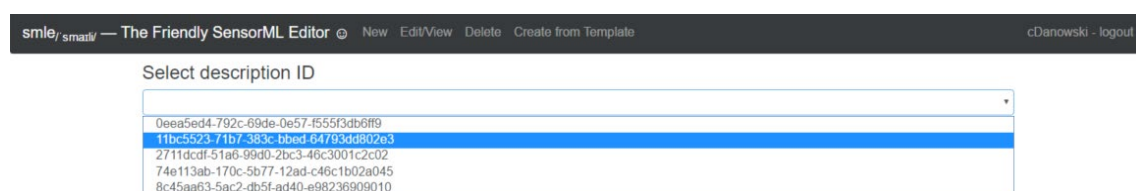


FIGURE 2.34: VIEWING SENSOR INSTANCE 1

After selecting a certain document, *SMLE* provides you with a collapsed tree structure of the elements of the SensorML document, as shown in the subsequent figure.

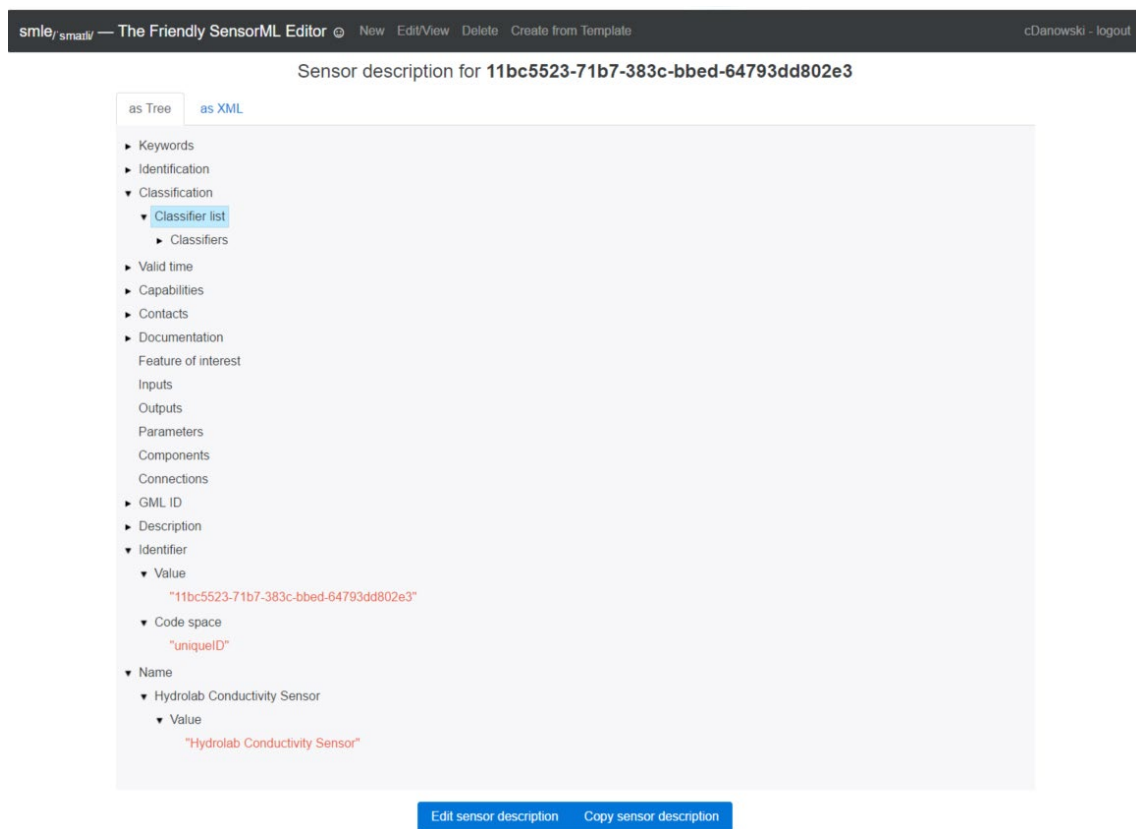


FIGURE 2.35: VIEWING SENSOR INSTANCE 2

You can inspect each element with a leading horizontal arrow by clicking on it to reveal the next level of the tree structure. Leaf node values are marked using orange colour and surrounding quotes while attribute values are marked using blue colour. Alternatively, the XML equivalent of the SensorML document can be viewed by switching to the **as XML** tab.

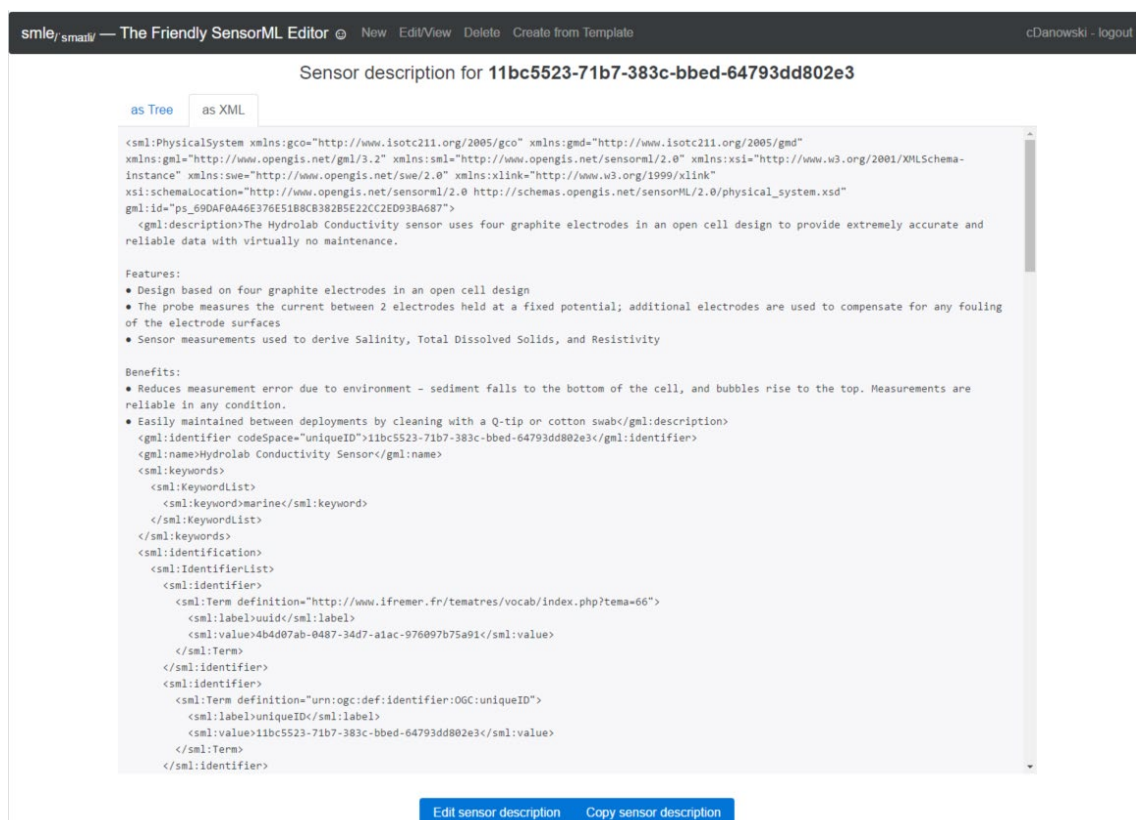


FIGURE 2.36: VIEWING SENSOR INSTANCE 3

Below both views you find two buttons **Edit sensor description** and **copy sensor description**. While the former allows you to edit the existing document (as identified by its unique ID) the latter creates a copy of the currently selected SensorML instance but removes its *unique identifier*, causing you to provide a new *identifier* within the edit view. Note that although you might not be logged in, you still see both buttons allowing you to switch to the corresponding views.

2.8.8) Edit your own SensorML Descriptions

You need to be logged in in order to edit an existing SensorML description. Also you are only allowed to edit sensor descriptions that you have created with your account.

To edit a previously created SensorML document, click on the menu **Edit/View** from the navigation bar and select the **id** of the document you intend to edit. This will provide you with the view window as already described in the previous section. To edit the document click on **Edit sensor description**, which opens the edit view, where all stored information of the selected document is displayed. Here you can apply changes like add missing information or override/remove existing elements/properties. The process of editing has already been described in section Create SensorML Document from Template and is skipped here. Please refer to that section for details. After editing you can re-publish the document by updating the document using the same *identifier*.

Role of the unique Identifier

Each SensorML document stored in the SOS is referenced by its unique *identifier*. When you intend to only update an existing sensor description, it is vital that you leave the *identifier* untouched. When you change it and then publish the document, you will create a whole new additional document, under the assumption that the new identifier does not exist yet. In a worse case the modified identifier references a different existing SensorML instance describing a different sensor. As a consequence, you should always carefully check the *identifier* property and associated notification messages of *SMLE* to prevent false publishing of a document. RDA group on sensor PIDs is available at : <https://www.rd-alliance.org/groups/persistent-identification-instruments-wg>, that aims to facilitate globally unique sensor ids.

Restriction to only edit your own Content

You are enabled to view and edit any available SensorML document of the underlying SOS. However, when trying to re-publish/update a modified SensorML document, which is not linked to your account, *SMLE* will inform you that you miss the appropriate permission, as the underlying SOS rejects the update from the non-authorized user.

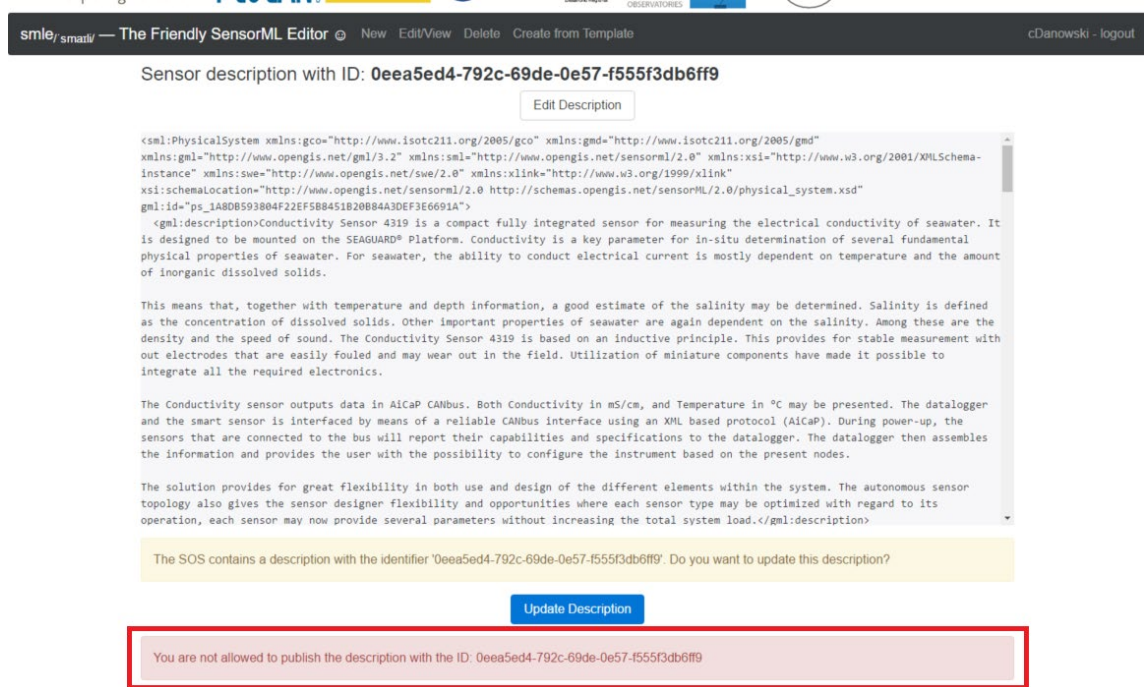


FIGURE 2.37: EDIT SENSOR INSTANCE

As a result, you are only allowed to update sensor descriptions that you have created with your current account. To still save the modifications of a certain sensor description, for which you do not have the sufficient permission, you might change its *identifier* and thus create an additional sensor description. Mind that this process preserves the *old* sensor description and creates a modified *new* version using a different *identifier*. Whether this approach is applicable is up to you and your working environment.

2.8.9) Delete your own SensorML Descriptions

You need to be logged in in order to delete a SensorML description. Also you are only allowed to delete sensor descriptions that you have created with your account.

To delete a previously created sensor description you have to navigate to menu **Delete**. Note that this menu is only available after successful login. Within the delete view, you choose the desired document via a dropdown list.

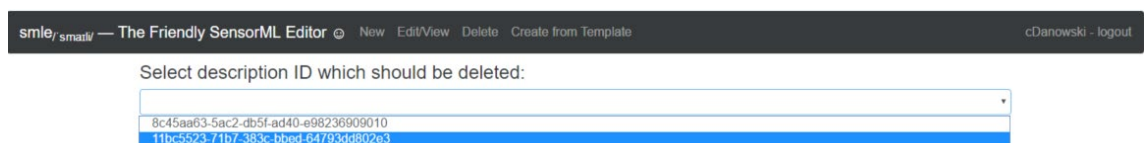


FIGURE 2.38: DELETE SENSOR INSTANCE 1

After selection you see the XML structure of the selected document. Inspect it to verify that this is the desired document, which should be deleted. By clicking the **Delete sensor description** button, you will irretrievably delete the document! *SMLE* will not prompt you with an additional conformation request.

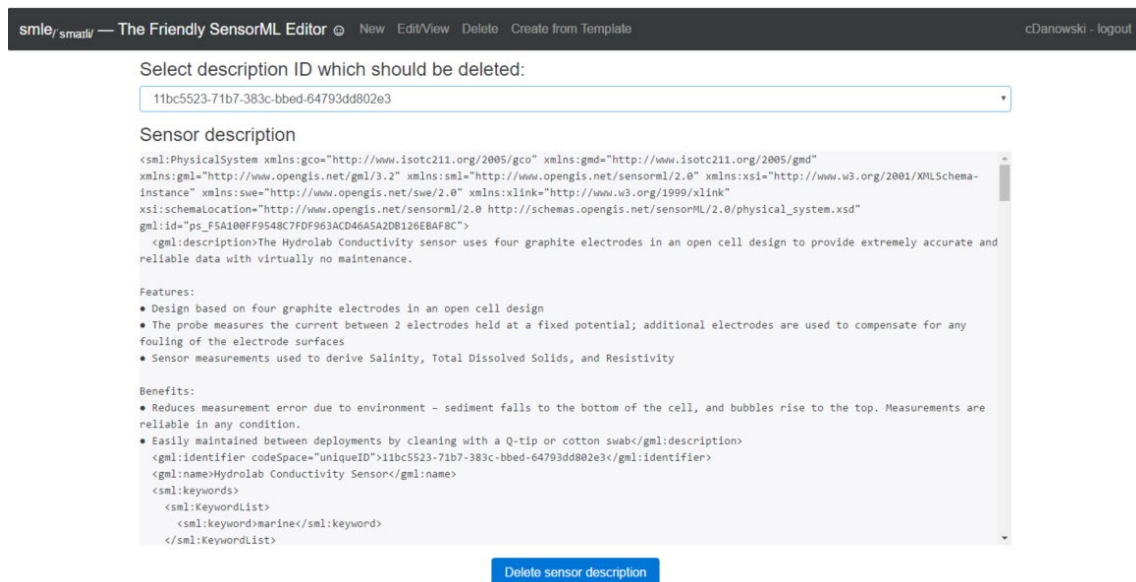


FIGURE 2.39: DELETE SENSOR INSTANCE 2

As a result, the document is deleted from the underlying SOS and SMLE displays a success message.

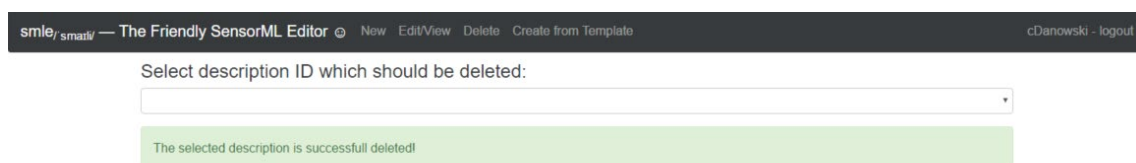


FIGURE 2.40: DELETE SENSOR INSTANCE 1

2.8.10) Restriction to only delete your own Content

Similar to the restriction of editing your own content, you are also only authorized to delete any SensorML document that is linked to your current account. You are not allowed to delete the descriptions of another user.

2.9) Outlook of SENSOR WEB ENABLEMENT

The Sensor Web Enablement (SWE) framework shows potential in delivering a new method for harmonization of ocean observing systems. Because several initiatives are willing to test the framework, there is a critical need to develop and test tools for real-world implementation. This starts with the proper registration of sensors, in SensorML format, which is the basis for all other SWE services in the framework. A user-friendly editor (SMLE, see above picture) was therefore developed, which has consisted in the core of the reported work. The document includes an exhaustive user manual in order to foster its use across the community.

SMLE is open source and is available at the following URL:

- <https://github.com/52North/sml-e>

Its latest deployment for demonstration is available at the following URL:

- <http://pilot.52north.org:3000/#/templates>

3) Inter-RI METHODOLOGY TO SUPPORT SENSOR MANAGEMENT

3.1) Introduction

The sensor web enablement provides a robust management tool for Research Infrastructures over the whole life of the sensor. In many infrastructures, sensors are liable to move from one implementation site to another or at least have to follow calibration procedures on a regular basis. The following section shows how the implementation of standard smart sensor interfaces may provide a cost advantage for the owner of large number of sensors (and other equipment).

The solution provided has been initiated by the use case TC4 selected by Theme 2 of ENVRIPLUS in order to focus development efforts on RI needs and cooperation with other Themes. The prototype implementation has been performed in context provided by the French Ministry of Research in its policy to cope with the Environment Research Infrastructure of EU supported by France and more especially ESFRI. The Research Data Management Policy of the French Ministry is to be able to coordinate the Data Bases of the various Environment RI Domain by Domain and also more globally. **Ifremer** launched the prototype implementation thanks to its partnership in most French Marine RIs and leadership of many: SeaDataNet, EMODnet, Copernicus, EURO-ARGO, EMSO, EUROFLEETS, JERICO-RI (French branch ILICO) and MaRINET2.

The management and operation of the associated observation networks requires rigorous management of the equipment (models on the market, stock status, deployed systems) and is based on so called **Sensor Registry** concept. The quality of the data acquired for science but also for surveillance depends on this monitoring. This management thus benefits data infrastructures and services, notably with the national land data poles, DCSMM, SeaDataNet, EMODnet, the Copernicus Marine Environment Monitoring Service (CMEMS), and as well the fisheries information system when existing, etc.

To be effective and rigorous, observation monitoring requires professional computerized management. To be optimized, this support service must also be shared between the observation networks. They share resources, methodologies or equipment. The challenges of organizing this integrated support are: **1)** control of monitoring costs and **2)** quality and traceability of observation, in favour of downstream services. Multiple entries of complex information and associated errors are thus avoided. Preservation and re-usability of the data will also benefit from this approach.

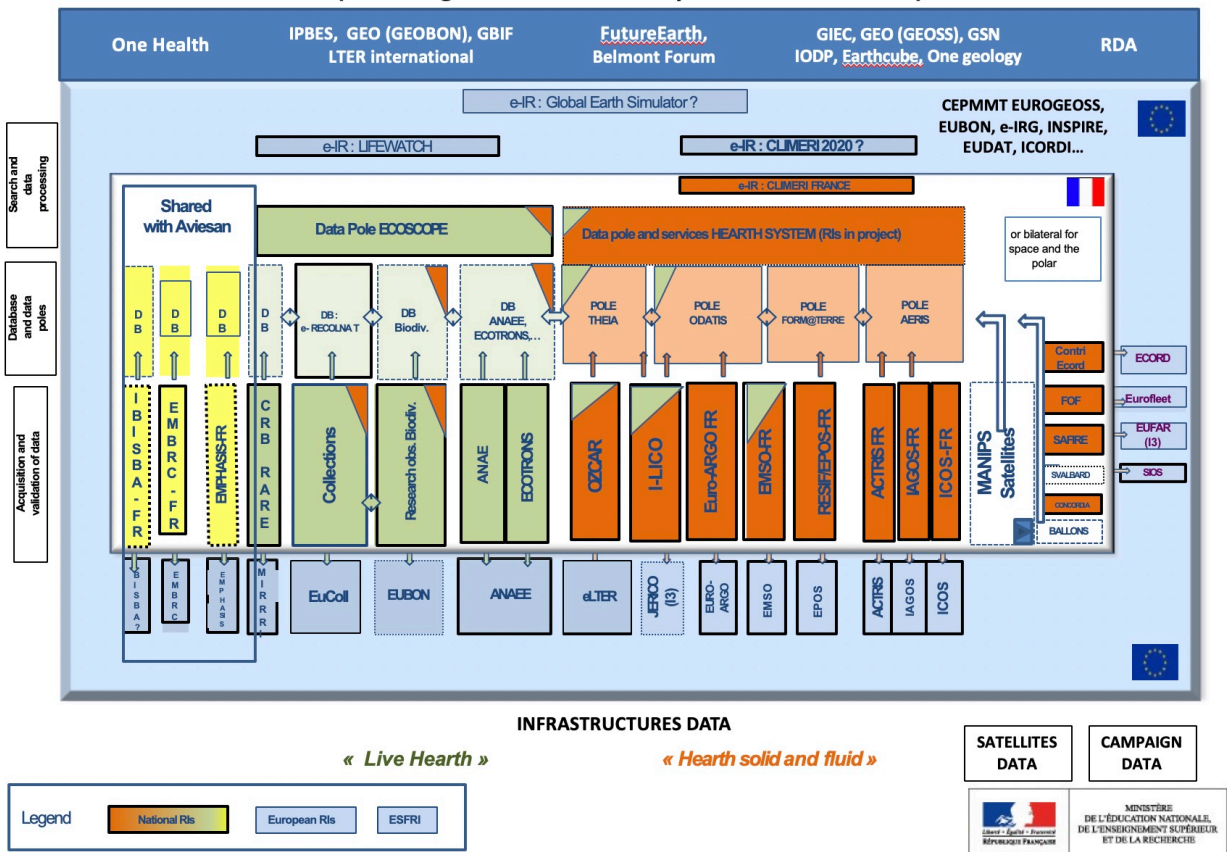


Figure 3.1: in blue, context in the landscape of the RI system earth and national environment, consider in addition the fleet and activities related to the study of biodiversity (e.g. EMBRC)

3.2) Present situation

Traditionally, data services developed are designed for data users, for monitoring and research. The services thus target downstream applications of observation: near-real-time monitoring of the water column (Copernicus MEMS), coastal environment (MSFD, EMODNET), long-term preservation of the data (SeaDataNet), etc. These emblematic initiatives provide targeted quality services by homogeneously integrating observations acquired in heterogeneous contexts. This effort of integration and collection of information is most often done in "back office", for each of the applications, according to their needs (more or less detailed context descriptions, specific vocabularies) and their planning (long term preservation or specific real-time service) (see Figure 3.2).

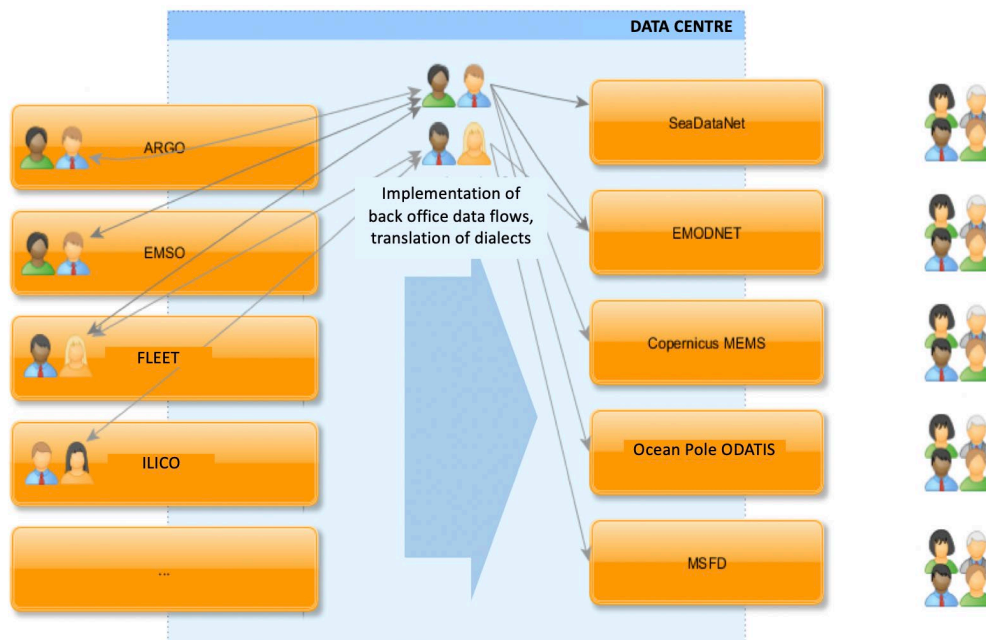


Figure 3.2: Integration of data from research infrastructures for downstream services, in back-office mode

It is recognized that **1)** the best sources of information regarding the context of the observation are the actors of the implementation of observation networks, scientists or operators of maintenances and deployments and **2)** the best moment to collect this information is when activities are conducted, especially in the field. On the contrary, from experience, the collection of context metadata at the time of long-term archiving is laborious and unreliable due to the distance from the facts described (Figure 3.2).

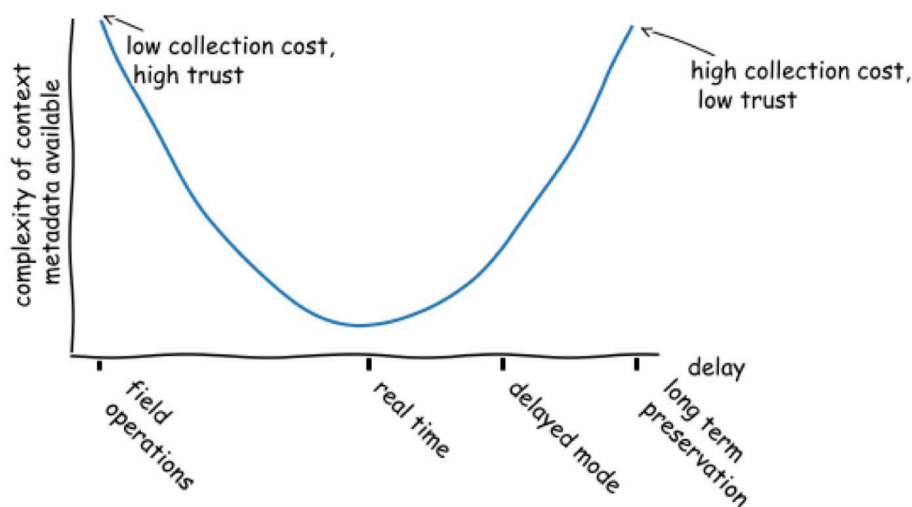


Figure 3.3: Costs and timelines for collecting context metadata on observation

In addition, observers are in need of computerized management tools for their piloting or maintenance activities (e.g. periodic calibration). In this area, for the RECOPECA observation network [Leblond, E. et al. ICES 2008 Annual Science Conference, Halifax, Canada, 2008], an integrated tool has been used since 2012 to manage in an integrated way deployment, network management and downstream data services. The tool was designed by targeting network managers as users and reusing the information they manage for the downstream data service (Figure 3.3). For example, the monitoring of the deployment of a temperature sensor on this or that ship by network operators allows the temperature measurement to be associated with the geo-referencing of the vessel, without additional input.

These support services for monitoring observation networks improve the coordination and productivity of the teams responsible.

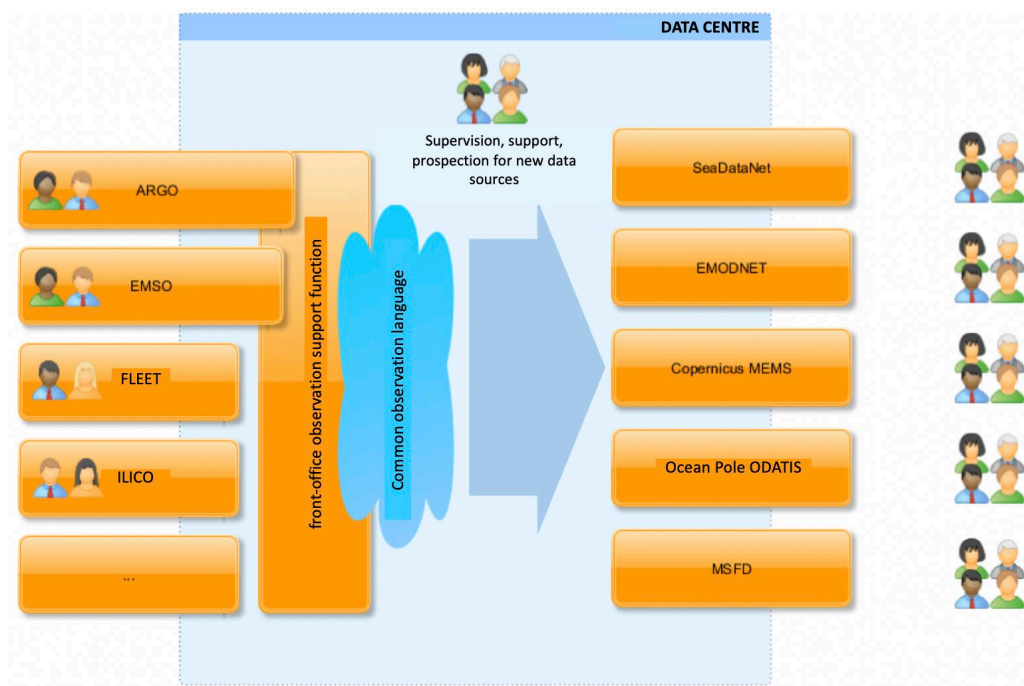


Figure 3.4: Development of support functions for observation

The provision of support services dedicated to observation actors, upstream of data centers, is therefore an opportunity to collect more precise, less noisy information on the context of observation without additional information collection costs. associated for downstream data services (see Figure 3.4). This precise information on the context of the observation is strategic for the certification of the observation, in the field of the surveillance, and for the reproducibility and the transparency of the results of research.

Some examples: the falling equation establishing the depth of measurements of the XBT lossy probes depends on the instrument model that sometimes, in the past, was lost. Today, the calibration parameters of the "biology" sensors of the ARGO floats are established in the test basin and necessary for the data processing when they are received at the data center. In both contexts, it is easy to understand the benefit of an integrated instrument and deployment tracking system. At the network level, since 2013 RECOPECA (marine observation system based on fishing boats) has been maintained thanks to an operator console from which the stock, the deployments, the data and their processing are managed in an integrated way (Figure 3.5).

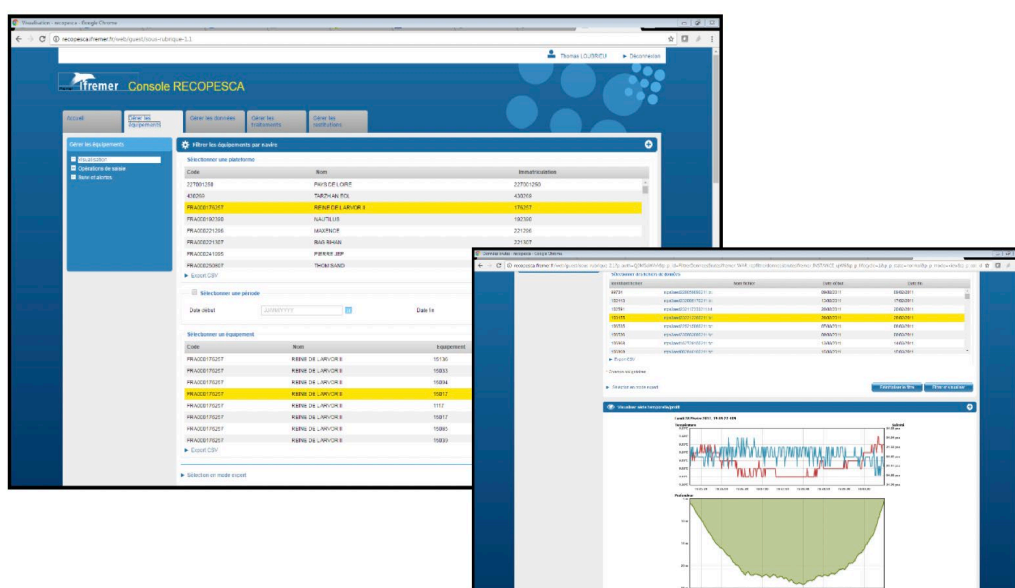


Figure 3.5: RECOPECA Network Monitoring Console

Ifremer's various research and monitoring activities require some flexibility in implementing the integration of support services to monitoring acquisition systems.

Two complementary strategies are considered:

1. To propose ergonomic, effective tools shared by the observation networks and the laboratories implementing them.
2. Establish standardized interfaces for the exchange of information between applications or systems sometimes dedicated to certain observation networks.

3.3) Effective tools

This application will provide monitoring system manager with a tool for monitoring their sensors and will allow the steering of this activity at the institutes and Research Infrastructure level. The life records of the equipment managed in this system will also be accessible for downstream data services using the data acquired by these instruments: date and calibration parameter, age of the sensors, for example, and etc.

The ESONET and Fixo3 projects developed the Yellow Pages containing the manufacturer's specifications for several hundred equipment models (see Fig. 2.6). The measurement ranges of the parameters are for example documented and can contribute to the qualification of the data acquired. This tool is hosted at Ifremer and its maintenance and use beyond the context of the projects that developed it is planned by EMSO-ERIC and EMSO-Link.

3.4) Standardized interfaces

Some embedded systems or network monitoring consoles (RECOPECA, ARGO) also contain primary information about the context of the observation. In these cases, the use of standardized interfaces is preferred to share relevant information without re-entry. Many projects, especially Europeans, are encouraging and proposing a framework for the standardization of these interfaces (ODIP2, AtlantOS, ENVRIplus, EMODNET-Ingestion, SeaDataCloud, NeXOS, Fixo3, Etc.). Experiments are carried out or planned for the following observations: fixed platforms (Figure 3.6), ARGO, RECOPECA, HF radar, flow cytometry, fixed instruments research ship, CTD, seismic, test basin tests, MASTODON network, this, especially in the context of European projects such as JERICO-NEXT.

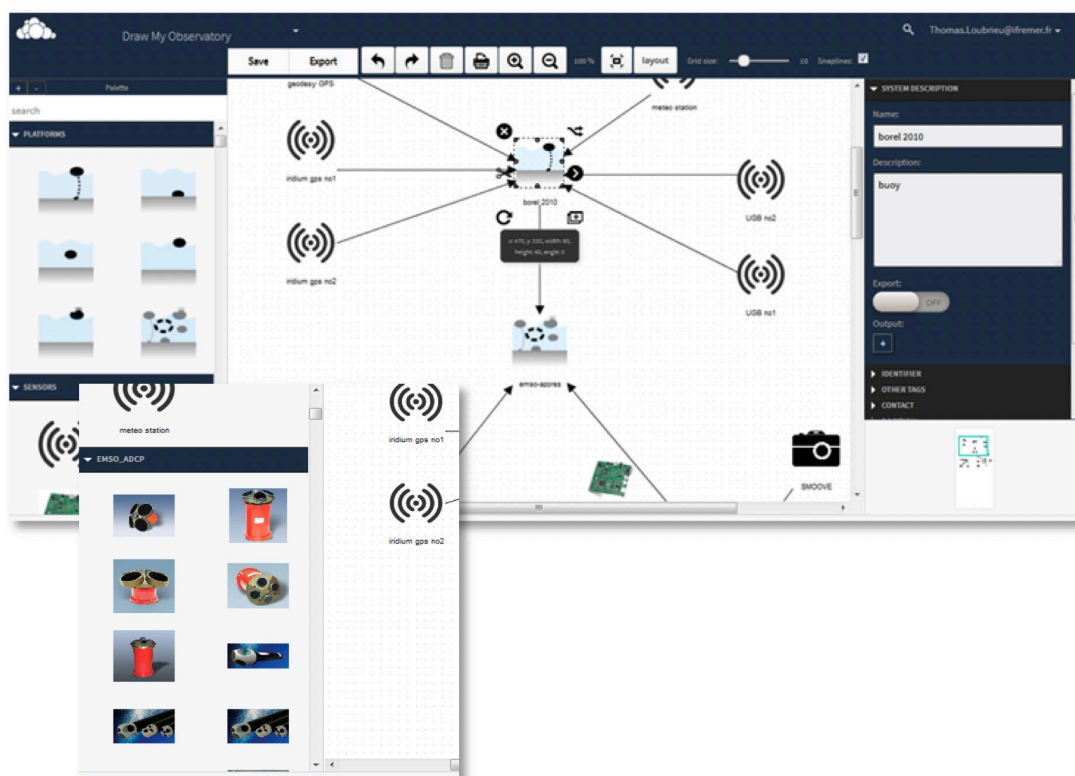


Figure 3.6: Fixed deployment graphic editor reusing Fixo3 equipment model catalog (see icons), for EMSO-Azores

In addition to the two complementary strategies presented, the accessibility and therefore the re-use of the information collected requires powerful indexing. Downstream data services, in an interdisciplinary and transverse context, will be able to exploit this information despite their heterogeneity and complexity. Technical feasibility has been demonstrated in the Sensor Nanny portal (Figure 3.7).

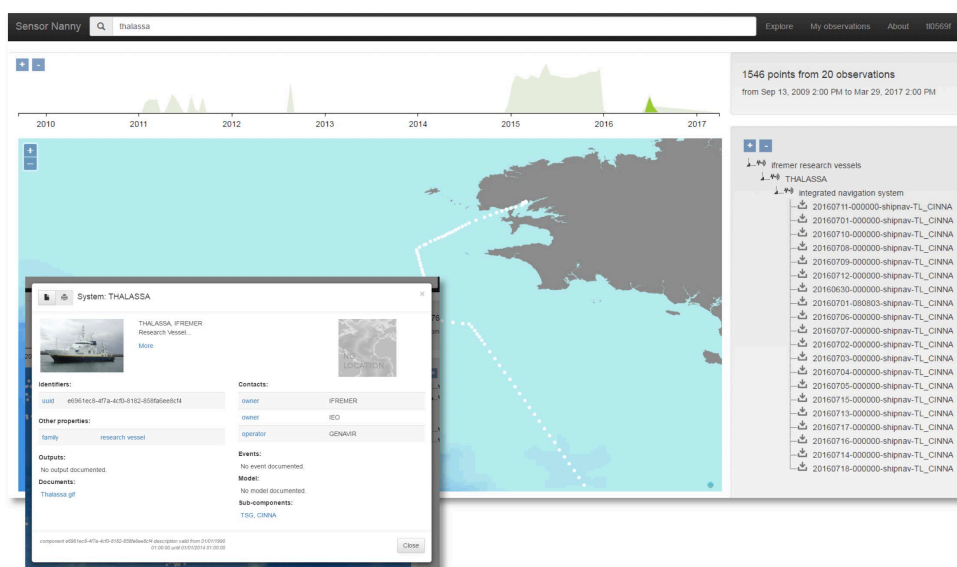


Figure 3.7: Index and portal of observations and acquisition systems, here navigation of a research vessel (Sensor Nanny)

3.5) Conclusion

The usage of such concept and tools should be encouraged for environmental monitoring activities based on sensors in order to propose to the operational teams some standardized and efficient tool to insure the traceability of the measurement produced. Metadata needs to be populated by the various users that deal with the equipment all along the life cycle of the measurement systems. This step by step and collaborative tasks can only be achieved in an efficient way by using such interfaces and tools.

4) NETWORK SECURITY OF SENSORS in a cross-domain infrastructure under project. Secure Communication Protocol for Large Number of Distributed Sensors

The cybersecurity was not a dominating topic in science research at the time ENVRIPLUS proposal was submitted. It is now a large concern when the democratic access to science data is more and more a driver of Research Infrastructure objectives. Several participants to ENVRIPLUS recently published with US and Australian colleagues a paper underlining this need for “quality and security” of Ocean Research data. (Buck, JJH, et al.). It was decided early in the project to address this cybersecurity issue inside the Task 1.4 in order to provide input to the sensor interface specifications. The case to be solved has been taken from the Task 4.3 Task 4.3 of ENVRIPLUS: “Solid Earth-marine common operation of platforms – benthic stations ». One driving aspect is that it covers a common interest between EPOS and EMSO and consequently solid earth and marine domains. Seismic data in general and acoustic data from the ocean are likely to be the target of hacking or cyberconflicts. It has been acknowledged after the Tohoku-Sendai megathrust in 2011 that economic consequences were dependent of the fast reaction of the stock exchange rates. What should occur if a fake seismic alert would be ever broadcasted from a research infrastructure open database?

4.1) Context of the study

The Joint Task Force (ITU-WMO-UNESCO IOC) (see <https://www.itu.int/en/ITU-T/climatechange/task-force-sc/Pages/default.aspx>) investigates the potential of using submarine telecommunication cables for ocean and climate monitoring and disaster warning. The general concept is to add sensors to the repeaters already installed on these cables. These sensors can measure temperature, absolute pressure, acceleration of the soil and other useful data which can be used for research purposes as well as detecting natural disasters such as tsunamis and earthquakes. Those cables are called “smart cables”.

This type of underwater cable infrastructure is already available for scientific monitoring: in the North East Pacific around Juan de Fuca tectonic plate by Neptune Canada¹ and OOI Regional Scale Nodes US² component along the east coast of Hondo, in Japan by S-net in the North³ and DONET 1 and 2⁴ in the central area. Cabled observatories in Europe are part of EMSO-ERIC, in East Sicily, Southern France Ligurian Sea, West Peloponnese and Canarias (PLOCAN). The security issues have been raised by the legal group of the Joint Task Force because of the reaction of telecom subsea cable owners when they were asked to offer an opportunity for scientists to use a minor part of the bandwidth of their trans-oceanic cables to transmit sensor data.

Large projects are expected in a near future; we can mention the CAM project (standing for Continent, Azores Madeira) (see <https://www.anacom.pt/render.jsp?contentId=1468767> and Figure 4.1) . Its cabled architecture and its ambition to measure seismic events along thousands of km corresponds to the kind of risks we address.

¹ Neptune Canada: <https://www.oceannetworks.ca/article-tags/neptune>

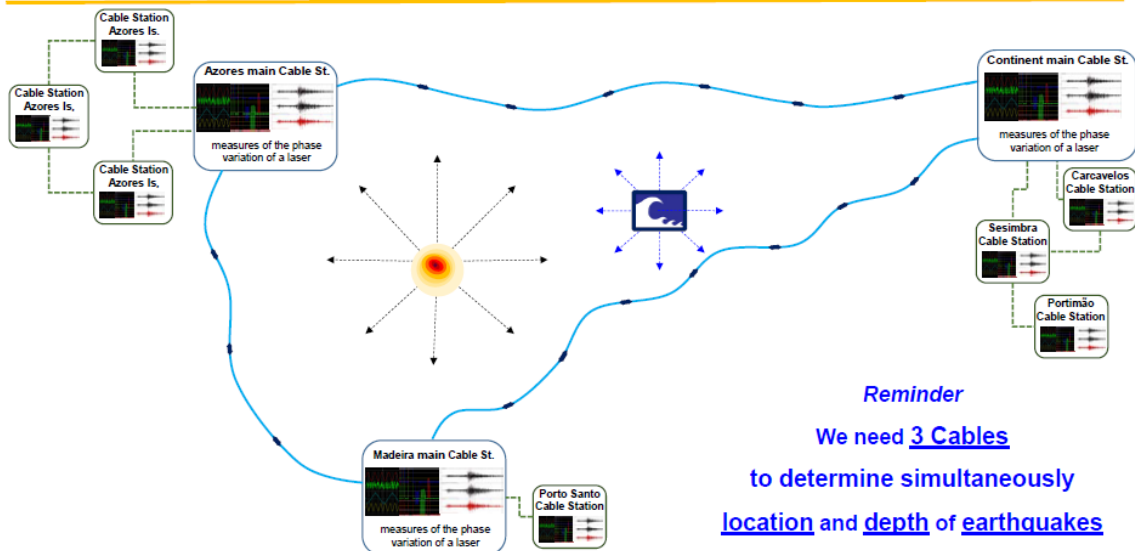
² OOI Regional Scale Nodes US:

https://interactiveoceans.washington.edu/story/The_OOI_RSN_Cable_System

³ S-net: <https://meetingorganizer.copernicus.org/EGU2018/EGU2018-12000.pdf>

⁴ DONET 1 and 2 : <https://www.jamstec.go.jp/donet/e/donet/operation.html>

Smart, Green & Blue approach - Detection of seismic vibrations caused in the fiber all along repeated and unrepeatd systems



Smart, Green & Blue approach - Data storage and Data processing, a possible way to go

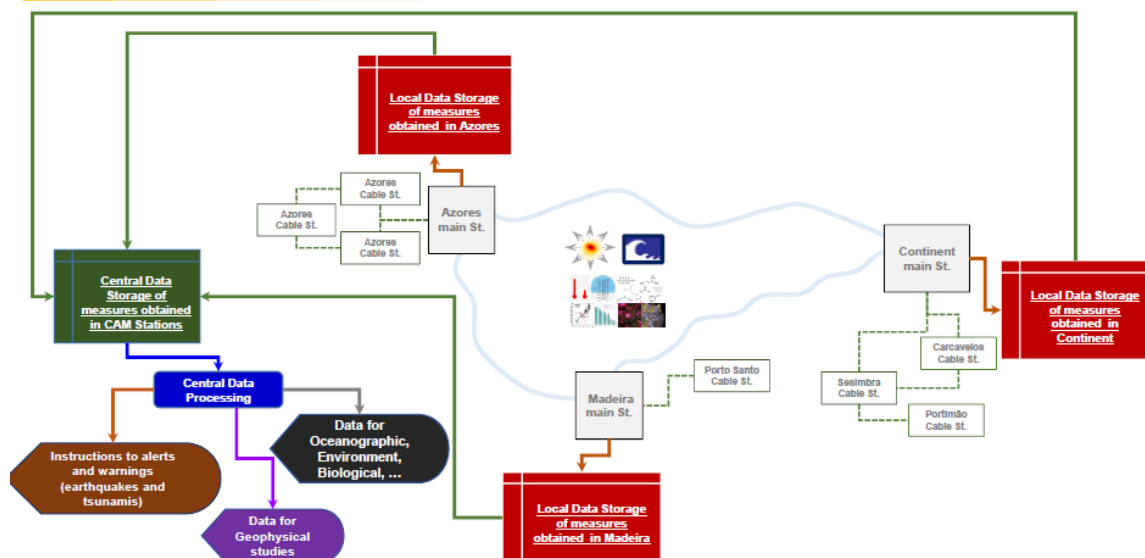


Figure 4.1: CAM project by ANACOM

This type of project potentially providing extremely important scientific information from the mid-Atlantic natural phenomena in the Ocean and Solid Earth could be potentially endangered by cyber-attacks.

EMSO-ERIC partnership with the cable owners in the future is limited by the security and responsibility topics. That is why it has been addressed as an ENVRIPLUS study, to our knowledge for the first time.

The objective of this contribution of ENVRIPLUS is to provide baseline requirements that will improve the security of data communication between sensors and database hosts. The communication needs to have specific security properties as confidentiality, authenticity, integrity, availability, Interruptibility. The additional security should have a minimum effect on the performance of the already installed devices. In addition, the computations required by security measures should expect the rough physical conditions which render complicated computations more likely to produce errors.

We investigate an appropriate data encoding to secure data transmission. Instead of classic cryptography such as AES256, we propose a different idea of how messages are coded ensuring secure communication with high performance. We therefore propose the RITA protocol, which is expected to have a satisfying security level with a very low cost (power, time, and physical space). Our approach does not need specific requirements from the other parts of the system. It can be adjusted to respect different models and different communication protocols accordingly. The protocol was already studied, simulated, and improved. However, additional robustness verifications such as patterns searching and other analysis.

We start in section 4.2 by a background of the security protocol we propose. In section 4.3, we present the protocol. Section 4.4 demonstrates how to apply our protocol in the project for a large number of distributed sensors. In section 4.5, we debate the robustness of the protocol regarding the requirements of the application. In section 4.6 we provide some experimentations and results. Finally, we summarize our work and present future plans in section 4.7.

4.2) Background in computer science

Our protocol is based on the concept of not using the same secret information (such as a key) twice. This is considered to be unbreakable since an attacker would not have useful information about the encryption.

The only cryptosystem that is considered perfectly secure is the Vernam cipher, also called the one-time pad. The one-time pad is based on a list of shared keys that can only be used once. These keys should be randomly generated and never reused. The main requirement of the one-time pad is the ability to share a list of random keys. Using a truly random key (K), the one-time pad can generate a truly random cipher-text (C). The simplest implementation of the one-time pad is $C = K \oplus M$ where M is the plain-text. Since an intruder can only know C the function becomes mathematically unsolvable (1 equation with 2 unknowns).

Gilbert Vernam patented this invention in the USA in July 1919 [Vernam, G.S., 1919]. A variation of the one-time pad was patented a few years later in Germany by Siemens and Halske [Siemens, 1923].

In 1949, Claude Shannon proved that the one-time pad is unbreakable and that any unbreakable system must respect the same properties as the one-time pad [Shannon, C.L., 1949].

To the best of our knowledge, other than the one-time pad, there are no other cryptosystems that were considered unbreakable.

4.3) Principles of the Protocol

Our protocol consists of two entities sharing and maintaining a secret. We call these entities security boxes, and once they share a secret they become twins. In comparison with the one-time pad, the secret boxes represent the agents and the shared secret represents the lists of secret keys.

The operations are very simple to make sure the performance is almost intact. We limit the operations to binary compositions (exp. binary XOR) and substitutions. The operations of our algorithm take place after sending/receiving a message and not before. This means that messages are sent and processed with almost no security related latency.

The 2 security boxes can be initialized together and then each is placed on the entry/outlet of any type of device that needs its communication secured. Another method is to have a common default architecture, then, using a key (used only once), they can each generate the common secret table and additional options. A security box can be an embedded device or an algorithm implemented on another already existent device (exp. computer, FPGA, etc.). Any communication between these devices should pass through both security boxes. The first security box translates the communicated message to a secret message and the second security box would reveal the secret.

We consider the following example. We have a smart phone with many applications, one of which would control a home automation box. We need to make sure the communication between the automation box and the application is secured from any attack. The attacks may come from the insecurity of the wireless communication, or by another application installed on the smart phone. We create 2 security boxes, the first is an embedded device that is installed on the automation box's input/output. The second is an algorithm on the smart phone and controls the input/output of the home automation application. Any communicated message between the automation box and the application translates into a secret message that has no meaning to an attacker. Changing this message would make it unreadable to the other side, making sure the integrity of the communication is intact.

The security boxes can be used as a middleware between 2 devices, the translation from the plain message to a secret message and vice versa can be different from one side to the other to make sure both devices send and receive messages they understand. The security boxes can also be used as a middleware between the devices and the communication channel, no matter what form of message the device produces, the security box can transform this message to a secret message that respects the protocol used in the communication channel without the need to change or adjust the secured device itself, the only modifications are the ones we do to the security box itself.

A very important aspect of the security boxes is that they do not require any changes from the devices being secured. It is up to the security box to adjust itself depending on the secured devices. To do so, the security device can either be a general box which would require configuration depending on the secured devices, this would increase the flexibility of the box but reduce its performance and security level. The second method is to have boxes specially designed depending on the requirements of the secured devices. Although the second method provides less flexibility, it ensures maximum performance and additional security. While the second method seems extreme with its need to recreate a security box depending on the requirements, its feasibility is real since only a small part of the security box is modified.

In the following we study the initial concept of the protocol followed by an advanced version where many security concerns are considered.

4.3.1) Initial Concept

If we want to secure the communication between 2 devices A and B, we proceed as follows: The twins SA and SB are placed each on a device, allowing these devices to communicate with each other. The first security box (sender) matches the communicated message with a value from the secret table. The second security box (receiver) would reveal the actual message by matching the value in its own version of the table. After sending or receiving a value, each security box should update its copy (before processing the next message) of the secret table based on the value it sent or received.

Figure 4.2 demonstrates an example of how this works. We have 2 devices A and B communicating with each other using the twins SA and SB. Each security box uses its version of table T to send and receive messages. Note that values in the same table should never be equal otherwise the when a value is received the index cannot be identified. Additional specifications about these values can be considered, for example, we can exclude specific values considered to reduce security, such as zero or powers of 2.

The value of the message is used as the index of the secret in the table. For example, if the possible messages are ON, OFF, and STAT, we associate each word with an index. The secret on index 1 in T is y1, therefore, SA sends y1 to SB. SB finds locates the index of y1 in T forwards it to device B. Both SA and SB update their version of T after sending or receiving a message.

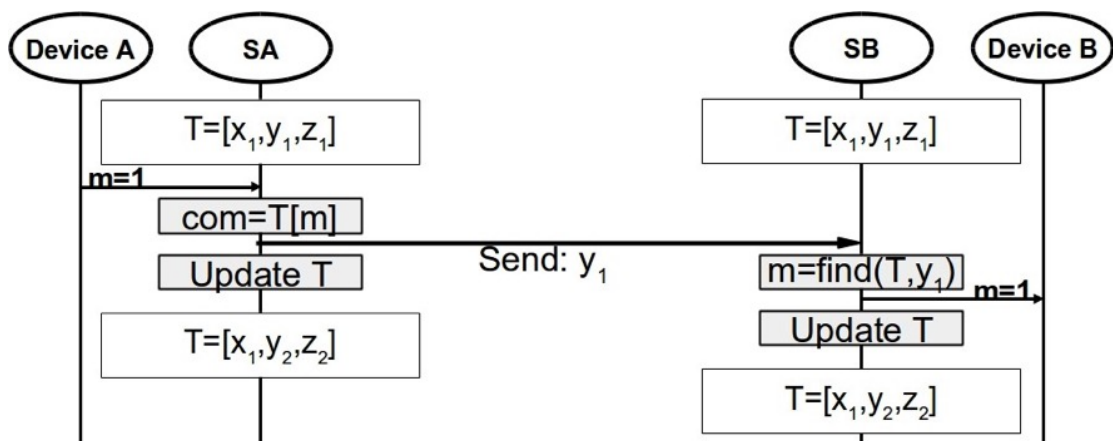


Figure 4.2: Simple Protocol

Table 4.1 demonstrates how the tables change using a function after each communication between SA and SB. We notice that at each moment, the tables on both sides are equal.

SA	Transmission	SB																																										
<table border="1"> <tr><td>x_1</td><td>y_1</td><td>z_1</td></tr> <tr><td colspan="3">$\downarrow f(x_1, x_1)$</td></tr> <tr><td>x_2</td><td>y_1</td><td>z_1</td></tr> <tr><td colspan="3">$\downarrow f(z_1, z_1)$</td></tr> <tr><td>x_2</td><td>y_1</td><td>z_2</td></tr> <tr><td colspan="3">$\downarrow f(y_1, y_1)$</td></tr> <tr><td>x_2</td><td>y_2</td><td>z_2</td></tr> </table>	x_1	y_1	z_1	$\downarrow f(x_1, x_1)$			x_2	y_1	z_1	$\downarrow f(z_1, z_1)$			x_2	y_1	z_2	$\downarrow f(y_1, y_1)$			x_2	y_2	z_2	<p>$\rightarrow \text{Send } (A, B, Aseq_1,) \rightarrow$</p> <p>$\leftarrow \text{Send } (B, A, Bseq_1, z_1) \leftarrow$</p> <p>$\rightarrow \text{Send } (A, B, Aseq_2, y_1) \rightarrow$</p>	<table border="1"> <tr><td>x_1</td><td>y_1</td><td>z_1</td></tr> <tr><td colspan="3">$\downarrow f(x_1, x_1)$</td></tr> <tr><td>x_2</td><td>y_1</td><td>z_1</td></tr> <tr><td colspan="3">$\downarrow f(z_1, z_1)$</td></tr> <tr><td>x_2</td><td>y_1</td><td>z_2</td></tr> <tr><td colspan="3">$\downarrow f(y_1, y_1)$</td></tr> <tr><td>x_2</td><td>y_2</td><td>z_2</td></tr> </table>	x_1	y_1	z_1	$\downarrow f(x_1, x_1)$			x_2	y_1	z_1	$\downarrow f(z_1, z_1)$			x_2	y_1	z_2	$\downarrow f(y_1, y_1)$			x_2	y_2	z_2
x_1	y_1	z_1																																										
$\downarrow f(x_1, x_1)$																																												
x_2	y_1	z_1																																										
$\downarrow f(z_1, z_1)$																																												
x_2	y_1	z_2																																										
$\downarrow f(y_1, y_1)$																																												
x_2	y_2	z_2																																										
x_1	y_1	z_1																																										
$\downarrow f(x_1, x_1)$																																												
x_2	y_1	z_1																																										
$\downarrow f(z_1, z_1)$																																												
x_2	y_1	z_2																																										
$\downarrow f(y_1, y_1)$																																												
x_2	y_2	z_2																																										

Table 4.1: Simple Rita Protocol

From the initial concept, many improvements and additions need to be taken in order to meet some security and safety requirements.

For instance, the initial concept does not consider asynchronous communication. If we use one table for both sending and receiving messages, asynchronous communication poses an issue since both sides may send a message at the same time before updating the table and these messages become unreadable. Therefore, we use on each side, one table to send messages and one table to receive them. The equivalent of the sending table on one side is the receiving table on the other side.

Additionally, we use public and private values in each table. The public values are the ones being sent and received, while the private values are the input of the table updating function. This ensures that an attacker does not have information about the input of the function since the private values are never used.

Finally, instead of updating only one value in the table, we can update the whole table each time. This technique is only efficient with small tables, meaning we need to have a very limited vocabulary in the communication. This is not always needed, and depends on other aspect such as the implementation of the protocol. For instance, if the public value is mixed later with the sequence, we do not need this complication.

Figure 4.3 demonstrates how the advanced protocol functions. T_s in SA is equivalent to T_r in SB and vice versa. When SA receives m , it identifies (in table T_s) and sends the secret which is in this case y_1 . SB receives the secret and locates the index of y_1 in T_r to find the original message. Both sides update their version of the table, T_s in SA and T_r and SB.

Table 4.2 gives an example of how the tables change in the case of the advanced protocol. We have a different table for sending and receiving messages on each side. Here we only study the SA side and what happens when a message is received or sent. We use the same function f on both sent and received messages, however, a different function can be used for each case.

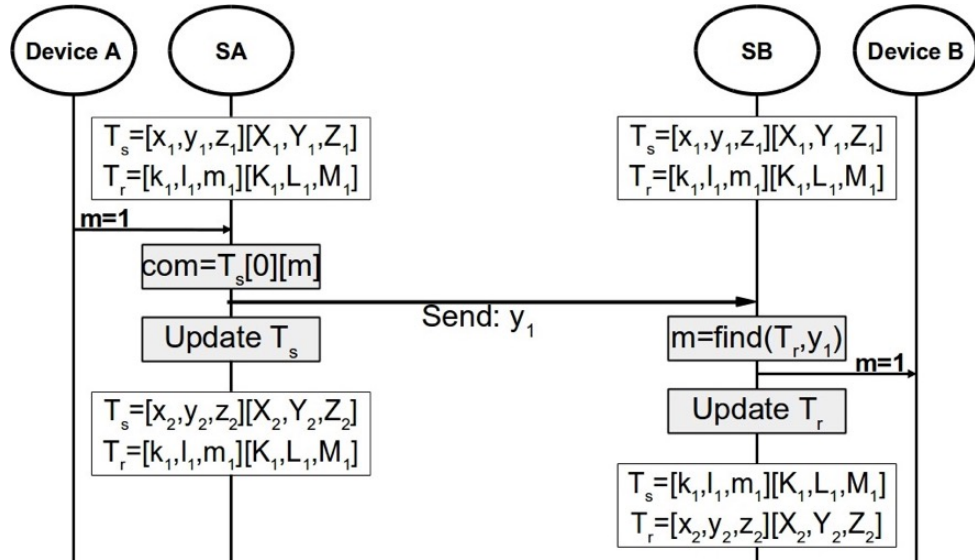


Figure 4.3: Advanced Protocol

SA						Transmission					
x_1	y_1	z_1	k_1	l_1	m_1	$\leftarrow \text{Send } (B, A, Bseq1, k_1) \leftarrow$					
X_1	Y_1	Z_1	K_1	L_1	M_1						
\downarrow			$\downarrow f(K_1, T_r)$			$\rightarrow \text{Send } (A, B, Aseq1, y_1) \rightarrow$					
x_1	y_1	z_1	k_2	l_2	m_2	$\leftarrow \text{Send } (B, A, Bseq2, k_2) \leftarrow$					
X_1	Y_1	Z_1	K_2	L_2	M_2						
$\downarrow f(Y_1, T_s)$			$\downarrow f(K_2, T_r)$			$\rightarrow \text{Send } (A, B, Aseq2, z_2) \rightarrow$					
x_2	y_2	z_2	k_3	l_3	m_3						
X_2	Y_2	Z_2	K_3	L_3	M_3						
$\downarrow f(Z_2, T_s)$			\downarrow								
x_3	y_3	z_3	k_3	l_3	m_3						
X_3	Y_3	Z_3	K_3	L_3	M_3						

Table 4.2: Advanced Rita Protocol

Finally, we use substitutions and permutations inside the updating function to create confusion and diffusion between the input and the output of the function. Figure 4.4 demonstrates how the advanced updating function works.

For example, the message being sent here is y_1 , therefore, we modify the table based on Y_1 . It is important that the modifications differ based on the private value related to the sent message. If not, the modifications become static and the security is reduced.

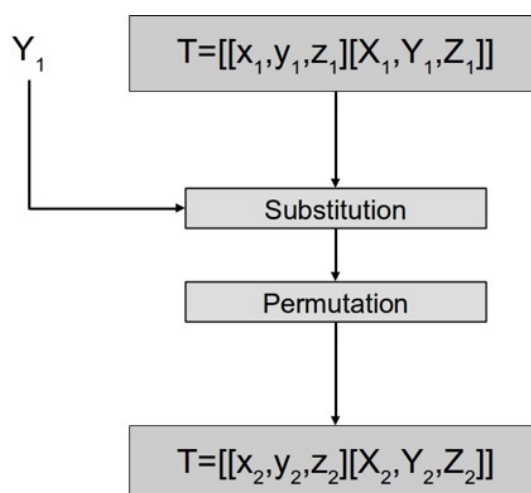


Figure 4.4: Table Updating Function

4.4) Application

In our application, we consider the specifications of the project regarding the distributed sensors. We start with the initial model and its security requirements before showing how the protocol can be introduced into the model and how the requirements can be fulfilled.

4.4.1) Initial Insecure Model

Figure 4.5 demonstrates a general view of how sensors are installed on the repeaters of smart cables. The communication is established as follows: The sensor D_i sends data to the communication panel C_i . The communication panel prepares a message to be sent through the repeater R_i which would then forward the message through the cable until it reaches a host (A or B). The receiving host forwards this message to the scientific centre SC.

In some cases, it is possible to have data sent through only one host. Additionally, the communication between the hosts and the scientific centre may have a different protocol than between the hosts and the communication panels.

We consider a worst-case scenario where communication between the hosts and the scientific centre is insecure. We also consider the communication between a host and a communication panel to be insecure. Our concept is to secure messages between the sensors and the communication panel. The application of the security measures can be either on the sensor output or the panel input. However, we consider it safer to be on the panel input so it would not disrupt or affect the sensor measurements.

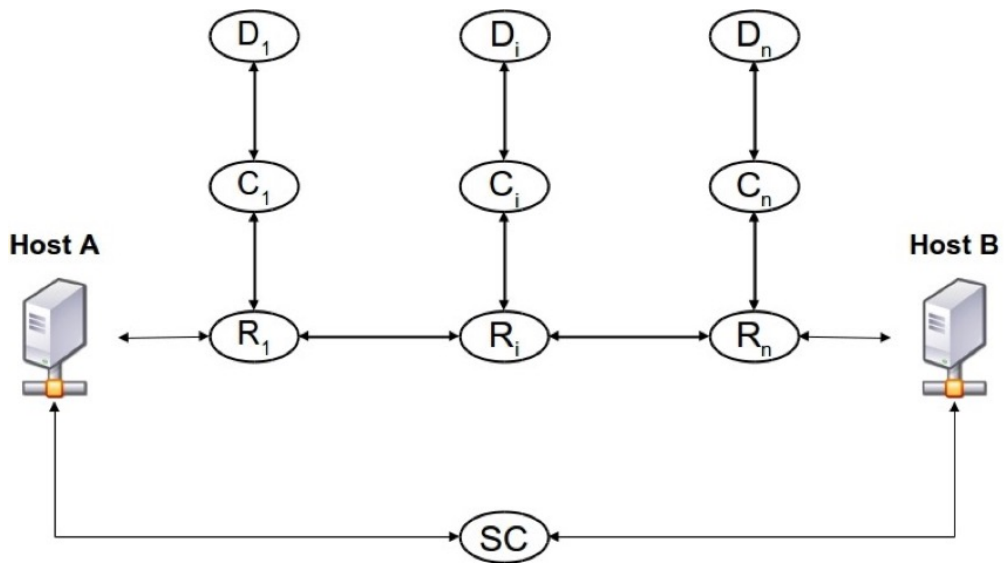


Figure 4.5: Insecure Model for distributed sensors

The security properties of this system are:

Confidentiality: Messages from D_i to SC can only be read by SC, and messages from SC to D_i can only be read by D_i .

Authenticity: When SC receives a message from S_i , SC is sure that the message originates from the exact sensor and was not sent by another sensor, or forged by a third party. Additionally, when D_i receives a configuration message from SC, D_i is sure that the message originates correctly from SC

Integrity: Once a message is created by SC or D_i , it cannot be modified. The format itself can be modified according to the used protocols in the hosts and the communication panels, however the data and its meaning are untouchable, and if changed, the receiving party would detect that.

Availability: SC and D_i can detect any Interruptibility of the service.

Interruptibility: At any point of time, a host may want to interrupt the communication for political or other reasons. It is important that messages stuck at the host, can later be forwarded to SC and can still be read in their original order.

4.4.2) Secure Model

In order to secure messages sent from the sensors, we use a security box on each sensor's input/output (or the communication panel's). For each one of these security boxes, we have a twin on the side of the scientific centre SC. Figure 4.6 shows the placement of the security boxes.

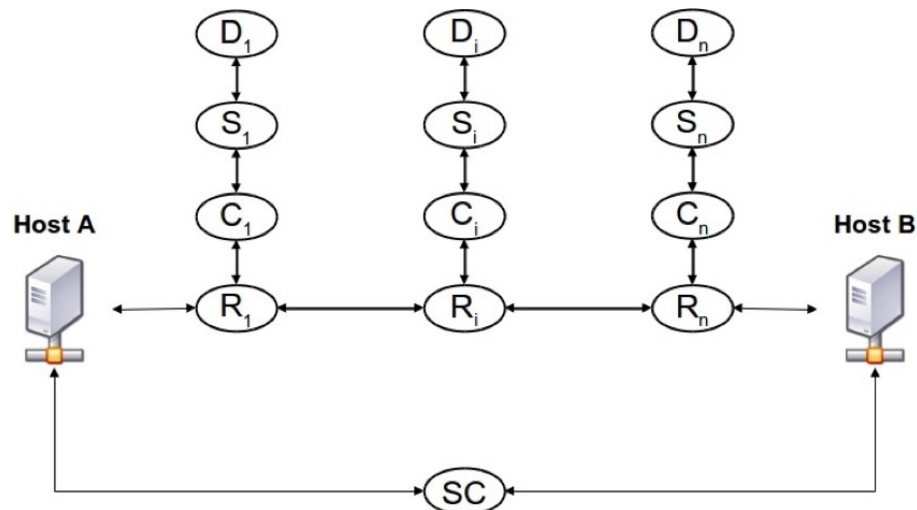


Figure 4.6: Secure Model for distributed sensors

The RITA protocol is best used in the case of limited vocabulary; therefore, we consider it in cases such as SCADA and IoT. While it is not limited to small size messages, limited signals, nor m2m communications, the protocol is at its maximum capabilities when applied on a limited vocabulary in m2m communications. In other cases, we can use multiple boxes for the same message, and base our tables on binary codes. This has not been tested, it is however expected to be less effective in terms of cost than classic cryptography.

In the case of distributed sensors, the limitations of the changing values to have a very limited vocabulary. For example, if we have a sensor that can change its value by 0.1 point each 1ms, and send its state each 1ms. Then, we can have a security box with tables of 3 values representing the following codes: +0.1, -0.1 and 0. The first information can be sent with a one-time key, or without any security measures. From this point forward, we only send the changes, which highly reduces the vocabulary and the size of a message. The security box should be able to update the table in less than 1ms, if not, messages would have to wait in the input pipeline of the security box. If the speed is high enough, this means that each message can be sent almost instantly, and, before the next message arrives at the security box, it would have already updated its table.

The SC can use multiple security boxes, either physically implemented as for the security boxes, or computer simulated depending on its needs. In case the SC needs to communicate back with the sensors, we need tables depending on the different possible configurations and commands.

4.5) Robustness

The one-time pad uses a key only once, reusing a key is a vulnerability, this is due to the possibility to attack the protocol by detecting the periodic reuse of the key. the key in RITA is the combination between the last state of the table (which is secret) and the initial algorithm (whether known or not). The last state would depend on all the previously sent messages, more precisely, on the corresponding secret for each message. It also depends on the initial table which is a secret.

For the moment, we cannot be sure whether it is important for the updating algorithm to be secret or not. If so, it can be done by generating a different combination of operations in the algorithm itself, which would be initialized when the security boxes are initialized. For example, we have a randomly generated substitution table

Consider the following case: at some point, we want to send the signal m_j from A to B using SA and SB as the security boxes. The current state of the sending table in SA is s_k , at another point of time, the table state happens to be s_k as well, and the signal to send is also m_j . In this case, both times SA is using the same key which breaks the property of the one-time pad that makes it perfect. However, from an attacker's point of view, he can never know at which states this happens, and the dynamics of the data being communicated prevents this from happening periodically.

The protocol ensures the security properties as follows:

Confidentiality: If an attacker C intercepts the message m_i corresponding to signal x , sent from SA to SB, C does not know the meaning of this message, therefore the property is respected. If in the future, if C intercepts the same message m_i , this time m_i can correspond to x , but can also be a different signal, the attacker has no clue which case it is.

Authenticity: Messages between twins have very limited choices, for instance, if we have a table of 3 messages each encoded in 32 bits, at any point of time, having a correct message has a chance of $3/2^{32}$. This means that when a message is received by a security box, and it happens to be correct/readable, we can be sure it is authentic.

Authenticity: If a message is forged, it has a probability of success of $n/|ttb| = n/(2^b)$ where n is the number of different possible signals 3 in the example above, and b is the default length used for the coded messages (32 for example). $3/|tt32| = 3/2^{32} \approx 7 \cdot 10^{-10}$ is about fifth the probability of gaining the jackpot Mega Millions multi-state lottery in the United States.

Integrity: For the same reason, modifying a message, or the order of messages, or replaying a message, also has a chance of $3/232$ of using a correct message. Also, even if the attacker considers already sent messages, each time he forges/modifies/resends a message, he would have the same chance of success.

Availability: An attacker C can interrupt the communication between SA and SB, however, he cannot replace the interrupted messages. If SA and SB use a time constraint based on a continuous checkup messages when no other data needs to be exchanged, any interruption would be detected and cause an alert.

Interruptibility: If a host interrupts communication, then it can collect messages at the host, without being able to read their contents. Once the interruption is over, messages can then be forwarded and received in the intended order.

4.6) Experimentations

In order to study and improve the algorithm, we created different computer-based simulations with 2 or more devices communicating using the security boxes. Although this is not ideal for performance measurements, our experimentations helped us improve the algorithm from its initial form to the security level it currently has.

In some cases, we did some simple pattern searches on tables states, and the values being sent. We did not find any pattern even when applied on small tables with only 16-bit values. Even in cases where the original messages from the devices were static, there was no pattern in the tables states nor the coded messages being sent. We also noticed that, values sent recently may or may not reappear. This is important because it means that an attacker cannot improve his guess based on the last sent messages.

Additional experimentations are required, especially using embedded devices to simulate a more realistic environment. It is also possible to use real values from actual systems in the simulations and evaluate the results to find possible problems or/and vulnerabilities.

In figure 4.7 we have a device generating messages as fast as possible (no wait between messages). These messages are then treated by a security box, we have two cases, one in which only the sent value from the table is updated after sending, and the second is where all values are updated. As expected, updating the whole table costs more time than updating only one value, however additional information can be useful in this area. For this, we use figure 4.7 where we simulate different latencies in messages production. In our case, if we send less than 20 000 messages per second (wait more than 0.05 ms between messages), using the protocol with one variable update does not affect the performance of the communication. Also, if we send less than around 7000 messages per second (wait more than 0.00015 ms between messages), having full table updates does not affect the performance of the communication.

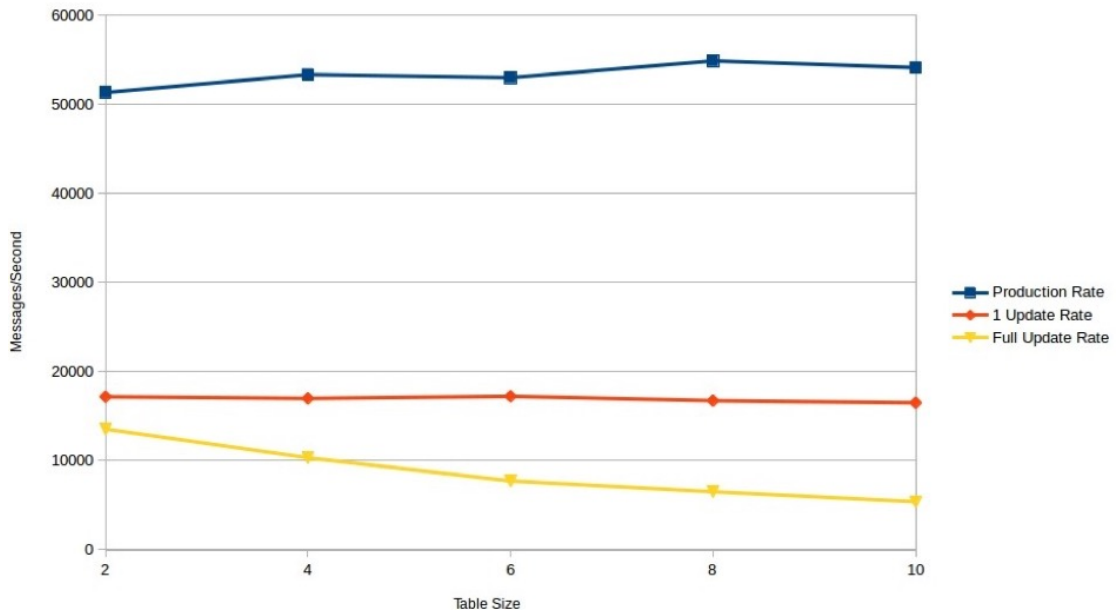


Figure 4.7: Simulation With no Wait Between Messages

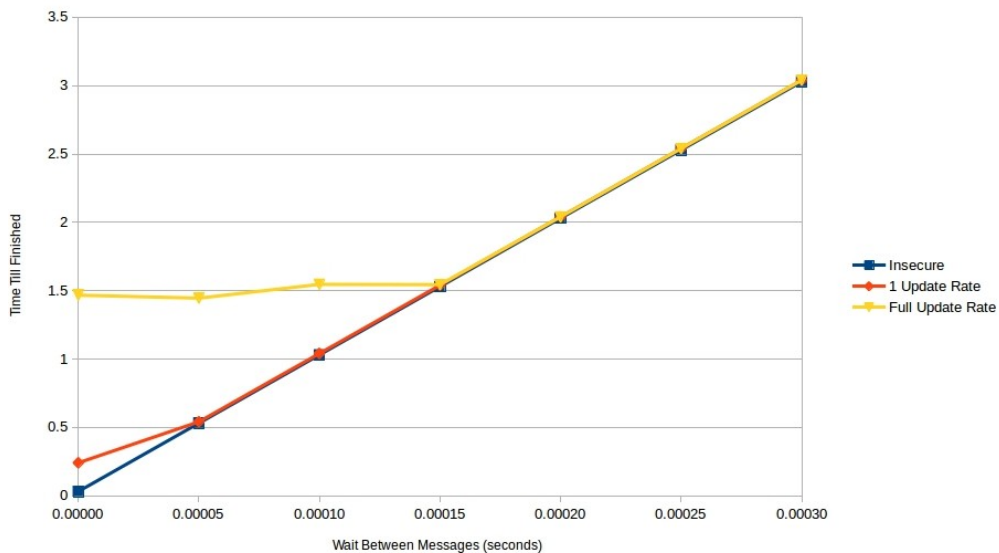


Figure 4.8: Simulation With no Wait Between Messages

4.7) Achievements

The RITA protocol is a new concept based on old techniques that were rarely used such as the one-time pad. We believe that given the right implementation area, the protocol can be very efficient, and even better than advanced cryptosystems.

The initial RITA protocol (without the advanced security schemes) was already published in NCS2016 [Obeid, F., Dhaussy, P., 2016a], presented in multiple conferences such as ACO2016 [Obeid, F., Dhaussy, P., 2016b], [Obeid, F., Dhaussy, P., 2018], and legally protected (without patent).

Although many improvements were done since the original concept, the protocol is still however in its early stages, and additional simulations and tests are required to improve it. An important test for the protocol would be its resistance to attacks and analyzes, especially checking for possible patterns (only simple patterns were tested for the moment).

We aim to have additional experimentations using multiple embedded devices simulating the sensors, the security boxes, and the communication panel

Finally, outside the aspects of the project, the protocol has other applications, such as low-level SCADA devices, and IoT communications.

5) CONCLUSIONS

The ambition of the task was to ensure converging efforts to support the growing potential of sensor interfacing engineering to the benefit of Environmental RIs. Indeed, sensor network enabling is providing a perspective for a better spatial coverage of natural phenomena, more cost-efficient deployments, homogeneous metrology and further data-processing. The management of many instruments either owned by RIs or part of citizen science programs may be envisaged in a near future under industrial methods and Quality Assurance.

5.1) Limitations

Unfortunately, it has not been possible to work with the citizen science experiments deployed in the Solid Earth Domain by Seismology Center Euro-Mediterranean (Rémy Bossu and colleagues) . The cell phone app LastQuake allows to use simple individual phones as sensors of great value for seismology. LastQuake automatically merges seismic data, direct (crowdsourced) and indirect eyewitnesses' contributions, damage scenarios and tsunami alerts to provide information on felt earthquakes and their effects in a time ranging from a few tens of seconds to 90 minutes. Only informal exchanges have been possible because of the dense schedules of the ENVRIWEEK meetings. The input of Solid Earth community would be necessary to conclude in the name of all ENVRI.

A second limitation is that digital interfaces are not implemented on all the sensors in environmental research. The transducers transforming the physical phenomenon into a quantitative parameter are based on electronics or optics. A strong tendency during the last decades has been to include an electronic analogic to digital converter level and produce a sensor integrating these functions and sometimes additional ones (self-compensation, self-calibration, pre-processing, energy management, ...). Our studies have assumed that it is the general case and that the digital interface is the rule! But in fact, two reasons are limiting this strong tendency and the larger implementation of standards presented in §2.5 and §2.6: a scientific one and a commercial/industrial one.

- In complex detectors or instruments with large bandwidth signal, it may be relevant to process several analogic signals before any digitalization. Then data from many sensors is merged to provide imagery type scientific data. In such case, the large number of sensors is not synonym of large number of digital interfaces.
- The market for manufacturers of scientific sensors is not large. They are either small companies or branches of high-tech groups with niche market specialization. Their benefit is often larger when they promote their own interface, their own communication protocols and their own sensor driver software. Then the sensor park of Research Infrastructures is either heterogeneous or manufacturer dependent. The case of Smart Cable presented in Task 4.3 and studied for its cybersecurity implications in §4 is significant. In the Joint Task Force Brest meeting on November 13th 2017, where the ENVRIPLUS cybersecurity study was presented, a new proposal by some telecom subsea cable major companies was presented. It was built on a concept of sensors provided without standard interface. These sensors would communicate with proprietary protocols, limiting scientific Quality Checking but with a maximum benefit for the providers.

Because of these limitations, this task of ENVRIPLUS addressed bottlenecks and ongoing technological innovations at the level of RIs, cluster of RIs, or cross-domain mainly in agriculture, air pollution and marine monitoring.

5.2) Main RESULTS

The developments in sensor interface engineering are stimulated by:

- The improvement in electronics (lower power consumption, higher capacities in bandwidth, CPU and storage) as shown in Deliverable D1.6,
- The launching and improvement of standards as presented in §2. They come from the industry and the perspective of Internet of Things. They are well adapted to many RIs. A community of RI engineers is strongly pushing for versions adapted to the Marine Domain RIs and applicable to many others.

A drawback of providing access to the sensors in an open data community of scientific RIs may be overcome by the use of suited protocols of security such as RITA as demonstrated in §4.

Even if they are heterogeneous in provider sources, funding, PI scientists, metrology sequencing, etc, it is possible to include sensors and associated equipment in operational databases (see §3). This includes a sensor registry, standard data interfaces and a provider data base. This opens the way to a complete follow up of the instruments whatever change in deployment place, in calibration procedures, in upgrading, in repair, in teams, PIs and technician moves, RI organization and RI growth. It will make the distributed Infrastructures of Research more robust in the management of their equipment, optimize their operational costs and, therefore, make them more sustainable.

In a decadal perspective, RIs will be able to democratize environment data in a context of mastered quality and security. We can only envisage a science of excellence with this Big Data if the chain from sensor hardware to publication is ensured and documented, including calibration. To build trust in the source data, the use of recognized standards instead of proprietary interfaces, the implementation of secured and hardened communication protocols (cybersecurity) are necessary.

6) REFERENCES

- ISO 11898-1:2003 Road vehicles — Controller area network — Part 1: Data link layer and physical signalling
- ISO 11898-2:2003 Road vehicles — Controller area network — Part 2: High-speed medium access unit
- ISO 11898-3:2006 Road vehicles — Controller area network — Part 3: Low-speed, fault-tolerant, medium dependent interface
- ISO 11898-4:2004 Road vehicles — Controller area network — Part 4: Time-triggered communication
- ISO 11898-5:2007 Road vehicles — Controller area network — Part 5: High-speed medium access unit with low-power mode
- ISO 11898-6:2013 Road vehicles -- Controller area network (CAN) -- Part 6: High-speed medium access unit with selective wake-up functionality
- IEEE1451 -Standard for a Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats, IEEE 1451.0, IEEE, 2007.
- ISO 19115 - International Organization for Standardization (ISO), “ISO 19115 metadata standard for geographic information,” Jul. 25, 2011 [Online]. Available: <http://www.iso.org>
- Cox, Simon (2011). OGC Implementation Specification: Observations and Measurements (O&M) - XML Implementation 2.0 (10-025r1). Wayland, MA, USA, Open Geospatial Consortium Inc.
- M. Botts, G. Percivall, C. Reed, and J. Davidson, “OGC sensor web enablement: Overview and high level architecture,” in *GeoSensor Networks*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, 2008, pp. 175–190.
- T. O’Reilly, “OGC® PUCK Protocol Standard,” Open Geospatial Consortium (OGC), Wayland, MA, USA, OGC Candidate Encoding Standard 09-127r2, 2014 [Online]. Available: <http://www.opengeospatial.org/standards/puck>
- ANSI/IEEE 488.2 IEEE Codes, Formats, Protocols, and Common Commands, and Standard Commands for Programmable Instruments
- Strain, L., Rajabifard, A., Williamson, I. - Marine administration and spatial data infrastructure, *Marine Policy*, 30 (4) (2006), pp. 431–441
- Botts, Mike and Alexandre Robin (2014). OGC Implementation Specification: Sensor Model Language (SensorML) 2.0.0 (12-000). Wayland, MA, USA, Open Geospatial Consortium Inc.
- Echterhoff, Johannes and Thomas Everding (2008). OGC Discussion Paper: OGC Sensor Event Service (SES) 0.3.0 (08-133). Wayland, MA, USA, Open Geospatial Consortium Inc.
- Bröring, Arne and Stefan Below (2010). OGC Discussion Paper: Sensor Interface Descriptors (OGC 10-134). Wayland, MA, USA, Open Geospatial Consortium Inc.
- Toma, D.M., O’Reilly, T., del Rio, J., Headley, K., Manuel, A., Bröring, A., and Edgington, D., “Smart sensors for interoperable smart ocean environment,” in *Proc. IEEE OCEANS Conf.*, Santander, Spain, Jun. 6–9, 2011, DOI: 10.1109/Oceans-Spain.2011.6003654.
- Bröring, Arne, Christoph Stasch and Johannes Echterhoff (2012). OGC Implementation Specification: Sensor Observation Service (SOS) 2.0 (12-006). Wayland, MA, USA, Open Geospatial Consortium Inc.
- GEOWOW Consortium (2014). OGC Discussion Paper: OGC Sensor Observation Service 2.0 Hydrology Profile (OGC 14-004). Wayland, MA, USA, Open Geospatial Consortium Inc.
- Jirka, Simon, Christoph Stasch and Arne Bröring (2011). OGC Discussion Paper: Lightweight SOS Profile for Stationary In-Situ Sensors (OGC 11-169). Wayland, MA, USA, Open Geospatial Consortium Inc.
- Simonis, Ingo, Echterhoff, J. (2011). OGC Implementation Specification: Sensor Planning Service (SPS) 2.0.0 (09-000). Wayland, MA, USA, Open Geospatial Consortium Inc.
- Robin, Alexandre (2011). OGC Implementation Specification: SWE Common Data Model 2.0.0 (08-094r1). Wayland, MA, USA, Open Geospatial Consortium Inc.
- Echterhoff, Johannes (2011). OGC Implementation Specification: SWE Service Model 2.0.0 (09-001). Wayland, MA, USA, Open Geospatial Consortium Inc.



- Coppola, L., Ntoumas, M., Bozzano, R., Bensi, M., Hartman, S., Charcos Llorens, M., Craig, J., Rolin, J-F. et al., "FixO3 Handbook Of Best Practices," 2016.
- Obeid, F., Dhaussy, P. **Rita secure communication protocol: Application to scada**. In 8th *International Conference on Network and Communications Security* (NCS 2016). Editor(s): David Wyld et al, 2016.a
- Obeid, F., Dhaussy, P. **A secured data communication protocol**. In Conf. *A Connected Ocean* (ACO 2016), SeaTech Week, Brest, october 2016.b
- Obeid, F., Dhaussy, P. **Secure Communication Protocol: Application to Large Number of Distributed Sensors**. 17th International Conference on Security and Management (SAM'18), Las Vegas, USA, July 30 - August 2 2018.
- Shannon, Claude E., **Communication theory of secrecy systems***. *Bell system technical journal*, 28(4):656–715, 1949.
- Vernam, G.S. **Secret signaling system**, July 22 1919. URL <http://www.google.com/patents/US1310719>. US Patent 1,310,719.
- Siemens and Halske, 1923. **Verfahren, vorrichtung und schaltungsanordnung zur nachrichtenuebermittlung in geheimschrift method, device and circuit for news about averaging in cipher**, March 10 1923. URL <https://www.google.com/patents/DE371087C?cl=en>. DE Patent 371,087.
- Buck JJH, Bainbridge SJ, Burger EF, Kraberg AC, Casari M, Casey KS, Darroch L, Rio JD, Metfies K, Delory E, Fischer PF, Gardner T, Heffernan R, Jirka S, Kokkinaki A, Loeb M, Buttigieg PL, Pearlman JS and Schewe I (2019) **Ocean Data Product Integration Through Innovation-The Next Level of Data Interoperability**. *Front. Mar. Sci.* 6:32. doi: 10.3389/fmars.2019.00032
- Cannat Mathilde, Sarradin Pierre-Marie, Blandin Jerome, Ballu Valérie, Barreyre Thibaut, Chavagnac Valérie, Colaco Ana, Crawford Wayne, Daniel Romuald, Escartin Javier, Legrand Julien, Matabos Marjolaine, Rommevaux Céline, Rouillet Guillaume, Reverdin Gilles, Sarrazin Joze (2016). **EMSO-Azores : Monitoring seafloor and water column processes at the Mid-Atlantic Ridge**. *FixO3 - Project Newsletter. Service Activities Special*, 3(1), 11. Open Access version : <https://archimer.ifremer.fr/doc/00353/46444/>
- Delory, E., Toma, D. M., Del Rio, F. J., Cervantes, P., Ruiz, P., & Memè, S. (2019). A New Generation of Interoperable Oceanic Passive Acoustics Sensors With Embedded Processing. In Cambridge: Elsevier (Ed.), *Challenges and Innovations in Ocean In Situ Sensors* (pp. 117–134).
- [Huber, R., Gilbert, O., Vermeulen, A., Papale, D., Rolin, J-F., Waldmann, C., Rohs, S., 2017](#). Report for best practices on robust telecom/datatransmission . **ENVRIPLUS Deliverable D3.3**.- <http://www.envriplus.eu/wp-content/uploads/2015/08/D3.3.pdf>
- Lanteri, N., Legrand, J., Moreau, B., Lagadec J.-R., Rolin J.-F. (2017). The EGIM, a generic instrumental module to equip EMSO observatories. In *OCEAN 2017 –Aberdeen* .IEEE. [Pages 1-5](#). <http://dx.doi.org/10.1109/OCEANSE.2017.8084824>
- Toma, D. M., Masmitja, I., del Río, F. J., Martinez, E., Artero-Delgado, C., Casale, A., ... Delory, E. (2018). Smart embedded passive acoustic devices for real-time hydroacoustic surveys. *Measurement: Journal of the International Measurement Confederation*, 125, 592–605, <https://doi.org/10.1016/j.measurement.2018.05.030>
- Leblond, E., Berthou, P., Laurans, M., Woerther, P., and Quemener, L., "The Recopesca project: a new example of participative approach to collect in-situ environmental and fisheries data using fisheries vessels of opportunity," in ICES 2008 Annual Science Conference, Halifax, Canada, 2008.

7) ACRONYMS

FixO³: Fixed point Open Ocean Observatories Network

SDI: Spatial Data Infrastructure

IEEE: Institute of Electrical and Electronics Engineers

OGC: Open Geospatial Consortium

ISO: International Organization for Standardization

O&M: Observations and Measurements

SOS: Sensor Observation Service

SML: Sensor Model Language (SensorML)

TML: Transducer Markup Language

SPS: Sensor Planning Service

ODIP: Ocean Data Interoperability Platform

IOOS: Integrated Ocean Observing System

EMSO: European Multidisciplinary Seafloor and Water Column Observatory

ESONET: European Seas Observatory Network

NeXOS: Next Generation Web-Enabled Sensors for the Monitoring of a Changing Ocean

O3YP's: Open Ocean Observatories Yellow Pages (a.k.a ESONET Yellow Pages)

SMLE: The newly developed SensorML Editor, pronounced "*SMiLE*"

MSFD: Marine Strategy Framework Directive

ⁱ https://ies-svn.jrc.ec.europa.eu/projects/download-services-tg/wiki/SOS_sub-group

ⁱⁱ <http://www.eurofleets.eu>

ⁱⁱⁱ <http://www.esonet-noe.org>

^{iv} <http://www.seadatanet.org>

^v <http://www.groom-fp7.eu>

^{vi} <http://www.nexosproject.eu> - NeXOS has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 614102

^{vii} <https://www.earthcube.org/group/x-domes>

^{viii} <http://www.esonet-emso.org>

^{ix} <http://www.oceansites.org>

^x <http://www.odip.org>

^{xi} <http://www.ioos.noaa.gov>

^{xii} <http://www.opengeospatial.org/standards/pubsub>

^{xiii} <http://www.opengeospatial.org/standards/sensorthings>